Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Microprocesseur de conduite d'une centrale solaire isolée ou en appoint à un réseau alternatif

Jadot, André

Award date:
1981

Awarding institution:
Universite de Namur

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 02. May. 2026

FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX

NAMUR

INSTITUT D'INFORMATIQUE



RUE GRANDGAGNAGE, 21, B - 5000 NAMUR (BELGIUM)

FN B 16/1981/5

FACULTES
UNIVERSITAIRES
N.-D. DE LA PAIX
NAMUR

Bibliothèque
FN B 16
1981/5

FACULTES UNIVERSITAIRES
N.D. DE LA PAIX - NAMUR
INSTITUT D'INFORMATIQUE

MICROPROCESSEUR DE CONDUITE D'UNE CENTRALE SOLAIRE ISOLEE OU EN APPOINT A UN RESEAU ALTERNATIF

Mémoire présenté par
A n d r é J A D O T
en vue de l'obtention
du grade de licencié et
maître en informatique

Année académique 80-81

831546
LBS 3430810

P R E L I M I N A I R E : Historique. [1]

Depuis l'aube des civilisations, la chaleur et la lumière du soleil furent la principale source d'énergie à laquelle les hommes puisèrent. Abandonnée pendant des siècles au profit du charbon, du pétrole et de l'uranium, l'énergie solaire nous revient sous le nom paradoxal d'énergie nouvelle.

L'énergie solaire est connue depuis que le monde est monde. Ce sont les Egyptiens, semble-t-il, qui les premiers découvrent l'effet de serre et leurs obélisques servent à mesurer le temps, grâce à leur ombre projetée sur des cadrans solaires. Héron d' Alexandrie construit un dispositif pour pomper l'eau à l'aide de la radiation solaire. Archimède incendie la flotte romaine à Syracuse en concentrant les rayons solaires sur les navires ennemis. En 1915, Salomon de Gaus construit une pompe solaire. La force motrice est fournie par de l'air chauffé à l'aide du rayonnement solaire. Lavoisier réalise le premier four solaire en concentrant l'énergie à l'aide d'une lentille à liquide.

Le soleil est également un symbole pour de nombreuses sociétés plus ou moins évoluées. L'inquisition du Moyen âge condamne Galilée. Il a osé dire, après Copernic, que la terre n'est pas le centre du monde, mais qu'elle tourne autour du soleil et sur elle-même. Onze théologiens du Saint-Office rédigent l'acte d'accusation : " C'est là une proposition absurde et fausse en philosophie et pour le moins erronée du point de vue théologique ".

Avec le temps qui passe, le soleil disparaît peu à peu des préoccupations humaines. Pendant des siècles, le courant des rivières et le vent animent les moulins, le bois des forêts procure le chauffage. Plus tard, la civilisation industrielle prend une position

radicale : les énergies fossiles triomphent sur les énergies renouvelables. On puise dans les mines de charbon puis dans les gisements de pétrole. Aujourd'hui, à l'heure de l'énergie atomique, le soleil nous revient, dépouillé de ses mythes et réduit au statut d'énergie.

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

Préliminaire.
Remerciements.

INTRODUCTION

PREMIERE PARTIE : Régulation d'une centrale solaire.

1. OBJET DU TRAVAIL.
1.1. DESCRIPTION D'UNE INSTALLATION SOLAIRE.
    1. Schéma général d'une installation solaire.
    2. La photopile et son organisation.
    3. Caractéristique courant/tension des modules solaires.
    4. Point de fonctionnement idéal.
    5. Caractéristiques de charge.
1.2. PROBLEME DE L'OPTIMISATION PRIMAIRE.
1.3. PROBLEME DE LA TRANSFORMATION DE LA PUISSANCE CONTINUE EN PUISSANCE ALTERNATIVE.
    1. Charge à courant alternatif.
    2. Principe de la transformation.
1.4. PROBLEME DE LA VARIATION DU RAYONNEMENT SOLAIRE.
    1. Variation du rayonnement.
    2. Première solution.
    3. Deuxième solution.
1.5. SYSTEME DE GESTION ET DE CONTROLE.

2. LES ASPECTS PHYSIQUES.
2.1. OPTIMISATION PRIMAIRE.
2.1.1. Le convertisseur à découpe continu-continu.
2.1.2. Le processus d'optimisation.
2.1.3. Algorithme.
2.1.4. Configuration.

2.2. TRANSFORMATION DE LA PUISSANCE CONTINUE EN PUISSANCE ALTERNATIVE.
2.2.1. Le hacheur.
2.2.2. Le processus de transformation.
2.2.3. Configuration.
2.3. ACCUMULATION TEMPORAIRE D'ENERGIE ET GESTION DE PRIORITE DANS LES CHARGES.
2.4. REGULATION DU COS φ.
2.4.1. Amélioration du cos φ.
2.4.2. Mesure du déphasage entre U et I.
2.5. INTERCONNEXION AVEC LE RESEAU PUBLIC.
2.5.1. Régulation de la puissance réactive par le réseau.
2.5.2. Principe de la régulation.
2.5.3. Configuration.
2.6. COUPLAGE DE LA CENTRALE SOLAIRE AU RESEAU.
2.6.1. Processus.
2.6.2. Configuration.

3. CONCEPTION.
3.1. CLASSIFICATION DES PROGRAMMES DE REGULATION.
1. Comportement passif de l'environnement.
2. Comportement déterministe de l'environnement.
3. Comportement aléatoire de l'environnement.
3.2. LA DECOUPE FONCTIONNELLE.
1. Optimisation primaire.
2. Transformation du courant continu en alternatif.
3. Optimisation générale.
4. Accumulation temporaire d'énergie et gestion de priorité dans les charges.
5. Mesure du déphasage entre U et I.
6. Amélioration du cos φ.
7. Régulation de la puissance réactive par le réseau.
8. Couplage au réseau de distribution.
3.3. REPARTITION ENTRE MATERIEL ET LOGICIEL.
1. Lecture de la puissance dans le bloc d'optimisation primaire.
2. Le timer relatif au moment d'enclenchement des vannes du hacheur.
3. L'évaluation du temps de déphasage.

3.4. ETUDE DU MATERIEL.

3.4.1. Architecture.

1. Architecture de la centrale solaire à courant continu.
2. Architecture de la centrale solaire autonome.
3. Architecture de la centrale solaire d'appoint à un réseau alternatif.

3.4.2. Adressage et interfaçage du matériel.

1. Adressage : généralités.
2. Interfaçage de la centrale solaire à courant continu.
3. Interfaçage de la centrale solaire autonome et de la centrale solaire d'appoint à un réseau alternatif.

3.5. CONCEPTION DU LOGICIEL.

3.5.1. Optimisation primaire.

3.5.2. Transformation du courant continu en courant alternatif.

3.5.3. Optimisation générale.

3.5.4. Accumulation temporaire et gestion de la priorité des charges.

3.5.5. Mesure du déphasage entre U et I.

3.5.6. Amélioration du cos φ.

3.5.7. Régulation de la puissance réactive par le réseau.

3.5.8. Couplage au réseau de distribution.

4. REALISATION.

4.1. PROGRAMME DU SYSTEME D'OPTIMISATION PRIMAIRE.

4.2. PROGRAMMES DU SYSTEME DE LA CENTRALE SOLAIRE AUTONOME.

4.2.1. Programme du système de la transformation du courant continu en courant alternatif.

4.2.2. Programmes du système qui assure les autres fonctions.

4.3. PROGRAMMES DU SYSTEME DE LA CENTRALE SOLAIRE D'APPOINT A UN RESEAU ALTERNATIF.

4.3.1. Programme du système de la transformation du courant continu en courant alternatif.

4.3.2. Programme du système qui assure les autres fonctions.

DEUXIEME PARTIE : Réseau d'interconnexion entre équipements de traitement et de transmission de données à microprocesseur.

5. RESEAU D'INTERCONNEXION ENTRE EQUIPEMENTS DE TRAITEMENT ET DE TRANSMISSION DE DONNEES A MICROPROCESSEUR.

5.1. LES OBJECTIFS DU SYSTEME.
5.1.1. Analyse de la situation.
5.1.2. Analyse des besoins.
5.1.2.1. Analyse des besoins au niveau des états.
5.1.2.2. Analyse des besoins au niveau des ordres.
5.1.2.3. Analyse des besoins au niveau des contrôles et des signalisations.
5.2. RESEAUX D'INTERCONNEXION DE PLUSIEURS MICROPROCESSEURS SITUES DANS DES SITES DIFFERENTS.
5.2.1. La structure des informations.
5.2.1.1. La synchronisation au niveau du caractère.
5.2.1.2. La synchronisation au niveau du bloc.
5.2.1.3. La synchronisation au niveau de la trame.
5.2.2. La discipline des voies.
5.2.2.1. Liaison symétrique-dissymétrique.
5.2.2.2. Type d'échanges.
1. Etoile traitement de données.
- Simple.
- à multiplexeur.
- à concentrateur.
2. Etoile processus industriel [TRACEC 130].
- Les options d'organisation de TRACEC 130.
- Les options de transmission de TRACEC 130.
- La structure des messages de TRACEC 130.
3. Point à point.
4. Réseaux multipoints.
5. Réseaux bouclés.
- Réseau en boucle.
- Réseau en anneau.
6. Réseaux maillé.
5.2.3. Procédure d'établissement de la ligne.
5.2.3.1. Procédure "je suis".
5.2.3.2. Procédure "je suis/qui êtes vous?".
5.2.3.3. Procédure "qui êtes vous".
5.2.4. Protocole de supervision des échanges.
5.2.4.1. Par code détecteur et correcteur.
5.2.4.2. Par recouvrement.
5.3. RESEAUX D'INTERCONNECTION MULTIPROCESSEURS SITUES DANS LE MEME SITE.
5.3.1. Introduction.

5.3.2. Réseaux à bus unique ou à bus multiple.
5.3.2.1. Bus unique ou jeu unique de bus.
5.3.2.1.1. Solution mixte - bus multiplexé.
5.3.2.1.2. Bus arbitre.
5.3.2.1.3. Les deux types de bus.
- Bus synchrone.
- Bus asynchrone.
5.3.3. Bus multiple ou a plusieurs jeux de bus.
5.3.3.1. Bus hiérarchisés à signalisation électrique.
5.3.3.2. Bus hiérarchisés à messages.

CONCLUSION.

BIBLIOGRAPHIE.

ANNEXE 1 : Fiches techniques de générateurs solaires.

ANNEXE 2 : Fiche technique du processeur 8080 A.

ANNENE 3 : Fiche technique de la mémoire commune 74170.

ANNEXE 4 : Le temporisateur programmable 8253.

ANNEXE 5 : Le contrôleur d'interruptions programmable 8259.

REMERCIEMENTS.

Je remercie tout particulièrement Monsieur B. PASSAU pour avoir dirigé le projet traité et Monsieur le Professeur J. BRUNIN pour m'avoir suivi dans mon travail et conseillé quant au contenu de ce rapport.

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

Les problèmes posés par l'utilisation des sources d'énergie actuelles sont multiples. En effet, songeons au prix du pétrole continuellement en augmentation, au souci d'indépendance énergétique vis-à-vis d'autres pays, à la sécurité et aux controverses relatives à l'utilisation de l'énergie nucléaire, à la pollution de l'environnement ... En conséquence, les recherches relatives à l'emploi d'énergies nouvelles s'intensifient un peu partout et, parmi celles-ci,l'énergie solaire est à l'ordre du jour.

En matière d'énergie solaire, diverses filières de mise en application sont possibles. Citons à cet égard la filière thermodynamique, la filière photovoltaïque et la filière biochimique. C'est dans la filière photovoltaïque que se situe ce travail. Dans une centrale photovoltaïque, un dispositif appelé panneau solaire, faisant appel à un phénomène physique, est à même de transformer directement l'énergie lumineuse en énergie électrique.

La conduite d'une centrale solaire à effet photovoltaïque requiert l'optimisation du point de fonctionnement entre les panneaux solaires et la charge. Cette optimisation générale doit se réaliser dynamiquement dans le temps. Par ailleurs, la plus grande partie des charges étant à courant alternatif, c'est dans un but de standardisation de la source mais également parce que nous envisageons la possibilité de couplage à un réseau de distribution qu'il faut transformer la puissance continue en puissance alternative. La variation du rayonnement solaire constitue le point faible des centrales solaires. Nous envisagerons deux solutions. Premièrement, l'accumulation temporaire d'énergie dans des supports physiques et, deuxièmement, l'interconnexion au réseau de distribution public.

Par ailleurs, la réflexion peut dépasser le cadre d'une centrale solaire pour s'étendre à un ensemble de centrales solaires, car, en principe, la production d'énergie photovoltaïque peut se faire dans des sites à grande surface.

A titre d'exemple, un projet d'irrigation pour la mise en valeur soit d'un territoire, soit d'une grande propriété, voire même d'un état peut faire appel à plusieurs centrales réparties. Il y aura des interconnexions entre équipements hydrauliques parce que le réseau d'irrigation demande la répartition de l'eau et le niveau des nappes peut varier. Par contre, la nature du projet ne réclame pas d'interconnexion électrique. Dans ce cas, toutes les centrales restent autonomes.

Au contraire, si nous constituons un réseau d'électricité pour une communauté, il y a lieu d'interconnecter les différents réseaux solaires. Il faudra par conséquent établir un système de concertation entre les centrales solaires pour réaliser une charge harmonieuse dans le réseau de puissance. Il y a donc un réseau d'informations entre les différentes centrales solaires.

Ce travail comprend deux parties. La première traite des problèmes relatifs à une centrale solaire. La seconde ébauche les problèmes du réseau de télécontrôle de la production d'énergie photovoltaïque dans des sites à grande surface.

P R E M I E R E P A R T I E.

REGULATION D'UNE CENTRALE SOLAIRE

1. O B J E T D U T R A V A I L

1.1 DESCRIPTION D'UNE INSTALLATION SOLAIRE.

1. Schéma général d'une installation solaire.

L'énergie qui assure la vie sur la terre et qui permet sa continuité nous parvient sous forme d'énergie rayonnante émise par le soleil dont une partie nous apparaît sous forme de lumière visible. Différents moyens plus ou moins complexes permettent d'en capter une partie et de la transformer directement en électricité. Parmi ceux-ci : le générateur photovoltaïque appelé communément photopile.

Nous trouvons à la figure 1 le schéma de principe d'une installation solaire photovoltaïque. Nous distinguons les quatre composants essentiels : la source, le panneau solaire, la charge et une boite électronique qui jouera le rôle d'élément d'adaptation et de régulation.

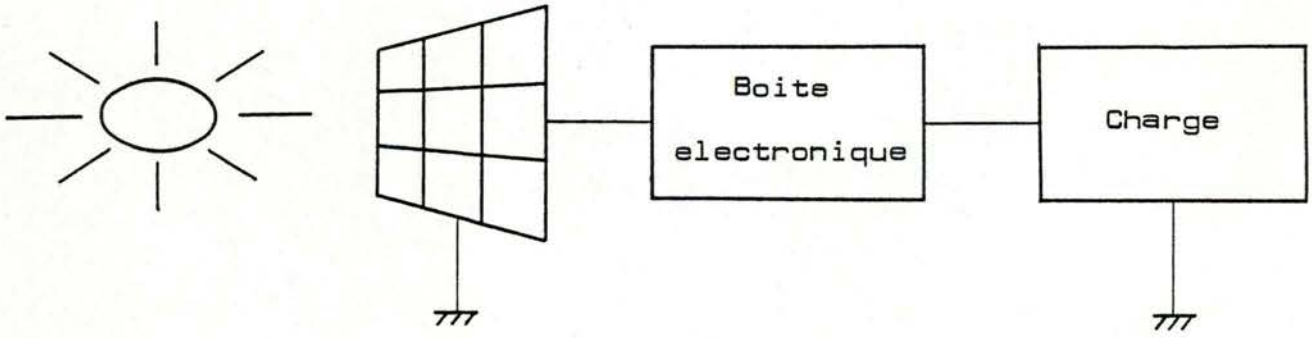


FIG 1

2. La photopile et son organisation.

La photopile est donc un dispositif qui est à même de convertir directement l'énergie lumineuse en énergie électrique. Dans tous les cas, la tension par élément et le courant par unité de surface sont faibles. En conséquence, si l'on désire obtenir une énergie exploitable, il est nécessaire de mettre en oeuvre un grand nombre de récepteurs connectés en parallèle et en série. La puissance totale obtenue vaut la somme de la puissance des cellules utilisées.

$$P_t = n \quad P_C$$

Du point de vue pratique, la photopile se présente sous la forme d'une cellule de 5 à 7 cm. de diamètre. Un module de photopile est un assemblage d'un certain nombre de cellules unitaires, interconnectées entre elles sur une surface plane et rectangulaire. Il forme un tout du point de vue utilisateur qui le considère comme un composant complexe. Un ensemble de modules solaires constitue un panneau solaire.

3. Caractéristique courant/tension des modules solaires.

Un module solaire possède des caractéristiques données de puissance et de tension. Branchons une charge aux bornes du module et envisageons deux cas. Dans le premier cas, la résistance de la charge est nulle. En pratique, on ne met pas les bornes des cellules en court-circuit sinon la puissance serait nulle. En effet, dans ce cas, un courant de court-circuit circule, mais la tension mesurée aux bornes du module est nulle. Dans le second cas, la résistance de charge est significative. Dès lors, une tension apparaît aux bornes du module. Si l'on fait croître la résistance de cette charge alors la tension augmente d'abord rapidement, plus lentement ensuite pour enfin tendre vers une tension limite qui correspond à une charge de résistance infinie. Cette tension limite est la tension de circuit ouvert pour laquelle comme pour le courant de court-circuit, la puissance est nulle.

Nous trouvons représentée à la figure 2, dans un graphique de I en fonction de V, une caractéristique courant/tension que nous désignerons également par courbe I [v]. Il s'agit d'une courbe ^{I} [v] d'un module solaire avec éclairement et température constants.

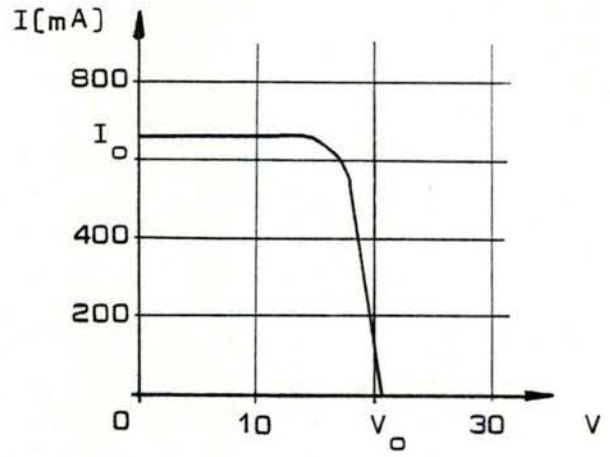


Fig 2

Deux paramètres, l'éclairement et la température, influent sur cette courbe I (v). Nous trouvons à la figure 3 les graphiques de plusieurs courbes I (v), en fonction d'un paramètre dans chaque graphique. En ce qui concerne le premier graphique, l'éclairement est variable pour une température donnée; tandis que dans le second, la température est variable pour un éclairement donné.

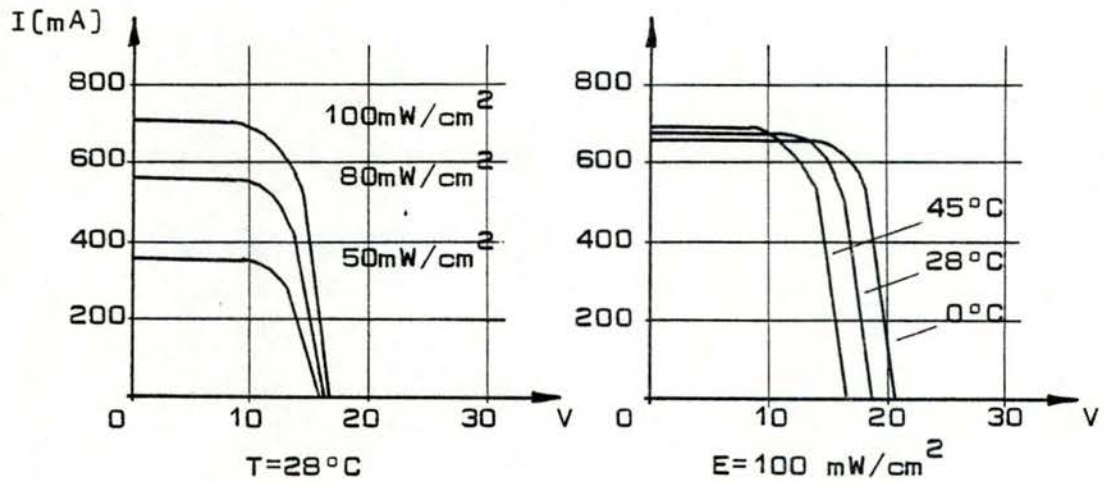


Fig 3

Par exemple, les nuages imprévisibles et inévitables peuvent influencer fortement la courbe I (v) en l'espace de quelques secondes. Par contre, les pertes calorifiques, surtout en fonction de la force du vent, déplacent la courbe I (v) en l'espace de plusieurs minutes.

4. Point de fonctionnement idéal.

On conçoit que si la puissance est nulle en I_o (valeur du courant de court-circuit) et en V_o (valeur de la tension à vide) elle passe par un maximum en (V_m, I_m). C'est ce point qui définit

le fonctionnement optimal en puissance du module; point sur lequel il faut se placer lors de l'utilisation du module pour obtenir le meilleur rendement possible.

Traçons sur le graphique d'une courbe I (v) pour un éclairement et une température donnée des hyperboles de même puissance. Le point de fonctionnement idéal se situe au point de tangence de la courbe I (v) donnée avec l'hyperbole de puissance la plus élevée.

Remarquons que si la courbe I (v) varie dans le temps, le point de fonctionnement idéal se déplace aussi dans l'espace I (v).

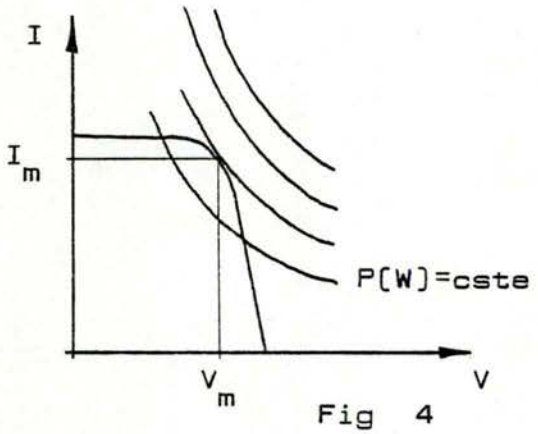


Fig 4

5. Les caractéristiques de charge.

Soit une charge Z quelconque ayant une caractéristique de charge linéaire. Si le courant est exprimé en ampères, si la tension est exprimée en volts et si la résistance est exprimée en ohms, alors nous pouvons écrire la relation

$$U = R\ I$$

Nous représentons cette caractéristique associée à la charge Z par une droite dans un diagramme I (v) et nous la désignons par la lettre Z. Un cas particulier peut être dégagé parmi les droites de charge Z. Une charge Z est optimale pour une courbe I (v) donnée si et seulement si sa droite de charge Z coupe la courbe I (v) en son point optimal.

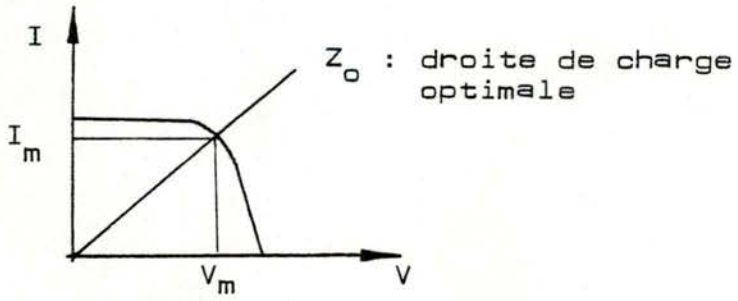


Fig 5

I.2 PROBLEME DE L'OPTIMISATION PRIMAIRE.

La droite de charge Z est dimensionnée par la charge de l'utilisateur. Il n'est par conséquent pas possible de paramétriser sa caractéristique de charge pour qu'elle coïncide à chaque instant avec la droite de charge optimale Z_o d'une courbe I (v) donnée.

Le but de l'optimisation primaire est de remédier à cette situation. A cet effet, un convertisseur continu-continu à découpe est interposé entre le panneau solaire et sa charge. Il doit transposer le point (V_m, I_m) de l'optimum du panneau solaire au point (V,I) sur la caractéristique de charge, ayant le même produit que (V_m, I_m).

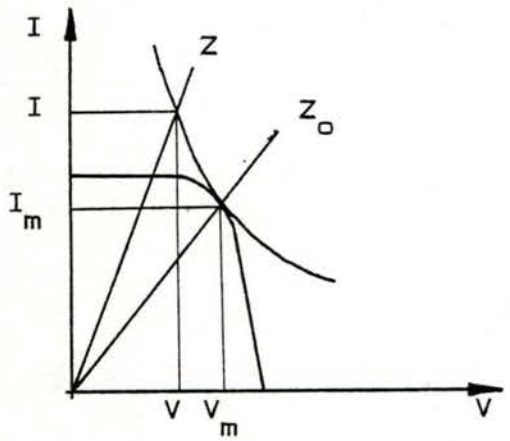


Fig 6

I.3 PROBLEME DE LA TRANSFORMATION DE LA PUISSANCE CONTINUE EN PUISSANCE ALTERNATIVE.

I. Charge à courant alternatif.

Actuellement, l'énergie solaire est généralement utilisée sous forme de courant continu. Cependant, dans la plupart des cas, les charges que nous pouvons rencontrer sont à courant alternatif. Si nous voulons aborder le plus vaste marché possible pour l'avenir, nous ne pouvons donc pas exclure que les applications requérant l'énergie solaire soient des applications à courant alternatif. Il serait dès lors élégant d'adjoindre un mécanisme de génération de courant alternatif à l'élément d'adaptation qui maximise la puissance.

C'est donc par souci de standardisation mais également parceque nous envisagerons dans la suite la possibilité d'interconnexion à un réseau public que la fonction de transformation de la puissance continue en puissance alternative est imputée à l'élément d'adaptation.

2. Principe de la transformation.

On divise chacune des demi-périodes en un nombre impair de parties égales; tout en calculant le nombre de volt-seconde nécessité par une sinusoïde dans la même division. Au cours de chaque division de période, une vanne de puissance est conduisante sous l'amplitude de tension disponible du panneau solaire pendant un temps qui donnera le même produit de volt-seconde que la quantité théorique associée à cette division. Une fois lissés, les crénaux de tension donnent la sinusoïde de tension alternative.

I.4 PROBLEME DE LA VARIATION DU RAYONNEMENT SOLAIRE.

1. Variation du rayonnement.

Les cellules solaires peuvent fournir un courant électrique aussi bien à partir de rayons solaires que de rayons provenant d'autres sources. C'est bien entendu le soleil comme source d'énergie primaire qui présente le maximum d'intérêt.

Cependant, l'utilisation du soleil comme source d'énergie primaire n'est pas sans inconvénients. Sa faiblesse fondamentale est la variation de son rayonnement. La source ne peut donc pas répondre à une demande continue. La variation de l'énergie solaire est animée par trois rythmes dissemblables. Le premier rythme est celui des nuages. En effet, en cas de ciel couvert, les rayons du soleil sont interceptés et affaiblis. Le second rythme est celui de l'alternance du jour et de la nuit. Enfin il y a le rythme plus lent des saisons.

Le rythme des saisons, de l'ordre de plusieurs mois, demande un stockage intersaisonnier pour conserver une partie de l'énergie de l'été jusqu'en hiver. Ce travail n'envisage pas ce problème.

Les deux autres rythmes posent un problème immédiat. L'alternance du jour et de la nuit entraîne la présence et l'absence d'énergie et les nuages provoquent une source essentiellement variable

Deux solutions sont envisagées. La première, pour une centrale solaire totalement autonome de son environnement. La seconde, particulièrement efficace quand il y a possibilité d'interconnexion à un réseau électrique public voisin.

2. Première solution.

Pour rendre possible le fonctionnement des appareils branchés sur les circuits alimentés par l'énergie solaire 24 heures sur 24, il est nécessaire de recourir aux batteries d'accumulateurs qui restituent pendant les heures d'absence ou de manque de soleil l'énergie emmagasinée auparavant.

La charge des accumulateurs ne pourra s'effectuer, à partir des cellules solaires, que pendant les périodes d'ensoleillement. Afin d'assurer un maximum d'autonomie aux accumulateurs, les charges sont réparties en deux classes : les charges prioritaires ne pouvant en aucun cas être déconnectées et les charges non-prioritaires déconnectées en cas de manque d'énergie primaire.

En outre, l'amélioration du cos φ peut être prise en considération et régulée au moyen d'une batterie de condensateurs.

3. Deuxième solution.

Une façon différente d'envisager le problème de la variabilité de la source permet d'éviter le stockage. Pour autant que la puissance continue soit transformée en puissance alternative, l'unité solaire peut être branchée au réseau de distribution électrique public moyennant un appareil de raccordement.

Lors des pointes d'insolation, l'unité solaire produit totalement ou partiellement le besoin en électricité des charges. Le réseau de distribution électrique public préserve toutes les charges des creux d'ensoleillement, garantit l'alimentation énergétique à tout moment et évite la mise en oeuvre d'accumulateurs ainsi que les problèmes dérivés. Si l'énergie solaire produite excède le besoin, l'excédent est injecté dans le réseau public.

Il y a lieu avant l' interconnexion de prendre le synchronisme avec le réseau, puis d' assurer le transfert maximum d' énergie active dans le réseau après interconnexion.

1.5 SYSTEME DE GESTION ET DE CONTROLE.

La mise en oeuvre de cette application à énergie solaire nécessite pour satisfaire aux objectifs assignés un système de régulation complexe dont le but est de gérer et de contrôler la bonne marche du processus.

Le microprocesseur est une solution rentable dans toutes les applications avec des algorithmes élaborés de régulation nécessitant le stockage de plusieurs données ou de nombreux organes d' entrée-sortie ou des calculs. Il dispose d' une grande souplesse d' adaptation mais il faut que sa vitesse soit compatible avec les exigences de l' application. Un système à microprocesseur n' exige que très peu de maintenance tout en offrant une fiabilité excellente. De plus, son prix de production est très abordable.

Il est cependant très difficile de choisir objectivement un microprocesseur dans la variété actuellement disponible sur le marché en tenant compte de tous les paramètres à la fois du microprocesseur et de l' application. Voici donc quelques critères permettant d' orienter ce choix.

Le choix se pose à deux niveaux :

1. Choix de la catégorie.
2. Choix dans la catégorie.

En effet, les microprocesseurs se répartissent en catégorie en fonction du nombre de bits d' unité d' information traitée et quelquefois par leur technologie de construction. Et par ailleurs, nombreux sont les microprocesseurs qui se concurrencent dans une catégorie donnée. C' est ainsi que l' on distingue la catégorie des 4 bits, des 8 bits et des 16 bits. Dans la catégorie des 8 bits, il y a entre autre le 8080 d' Intel, le 6800 de Motorola, le 2650 de Signetics.

Le choix d'une catégorie est en général assez simple. La gamme des 4 bits a été complètement supplantée par celle des 8 bits qui à l'heure actuelle ne revient pas plus cher. Deux critères qui découlent directement de l'application vont influencer notre choix. Il s'agit de la précision et de la vitesse.

1. En ce qui concerne la catégorie des 8 bits :
   - La précision est de 0,4 %
   - Le calcul en longueur double a l'inconvénient de ralentir le temps d'exécution du programme.
2. En ce qui concerne la catégorie des 16 bits :
   - La précision est de 0,2 °/ooo
   - Il offre une facilité plus grande que les 8 bits pour travailler avec des temporisateurs 16 bits
   - La limitation du nombre de broches oblige le multiplexage de certains signaux, ce qui fait perdre du temps.

Le choix du microprocesseur dans la catégorie est beaucoup plus délicat, tant sont voisines les performances. Nous retiendrons deux critères :

- Le coût du matériel et du développement.
- La performance.

1. Les coûts sont caractérisés par :

1.1. Le coût du matériel.
   - Le prix du microprocesseur.
   - Le prix des boîtiers annexes.
   - La disponibilité des secondes sources.

1.2. Le coût du développement.
   - Le prix du système de développement.
   - Le prix du support logiciel.

2. Les performances :
   - La vitesse (cycle d'horloge et nombre de cycle/instruction).
   - Le jeu d'instruction et le mode d'adressage.
   - Les Benchmarks.

A défaut d'être le critère le plus décisif, ce sont encore les performances qui diffèrent le plus dans une catégorie.

L'organe de conduite que nous nous proposons d'adopter pour la configuration de base est le microprocesseur 8080 A d'Intel.

2. LES ASPECTS PHYSIQUES.

2.1 OPTIMISATION PRIMAIRE.

Nous avons vu que la mise en oeuvre de panneaux solaires suppose pour assurer le débit maximum, le contrôle du transfert de l'énergie électrique entre la source et la charge.Elle demande de pouvoir faire varier la tension ou le courant débité dans un large domaine en consommant une puissance de commande aussi faible que possible. Les panneaux solaires ne sont pas utilisés directement :l'alimentation de la charge se fait par l'intermédiaire d'un convertisseur à découpe continu-continu.

2.1.1.LE CONVERTISSEUR A DECOUPE CONTINU-CONTINU.

1. Schéma de principe.

Un convertisseur à découpe fonctionne en établissant et en coupant les connexions entre le générateur et sa charge. Nous distinguons dans le schéma de principe de la figure 7, à gauche, le panneau solaire, suivi d'une cellule de filtrage L 1 , C 1 et enfin le bloc convertisseur.

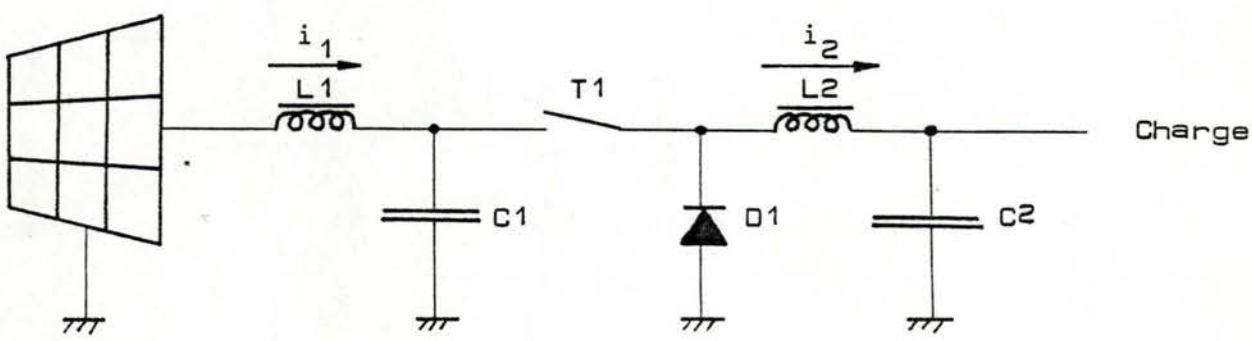


Fig 7

Le bloc à découpe est essentiellemt composé d'un transistor T 1, d'une diode D 1, d'une self L 2 et d'une capacité C 2.

Le panneau solaire délivre un courant i_1 à travers la self L 1. Ce courant charge C 1 lorsque T 1 est ouvert et se supperpose au courant de décharge de C 1 lorsque T 1 est fermé, pour donner un courant i_2 à travers L 2.

2. Fonctionnement. [3]

Lorsque le transistor T 1 est conduisant, un courant i_2 circule de la source V_p vers la self L 2. Lorsque T 1 est bloqué, il retient à ses bornes une tension $V_p - V_d$. Lorsque T 1 est bloqué, le courant i_2 dans la self L 2 ne peut varier de manière discontinue. La tension aux bornes de L 2 varie jusqu'à ce que D 1 devienne conduisante. La self L 2 tient une tension V_s à ses bornes lorsque T 1 est bloqué et $V_p - V_s$ lorsque T 1 est conduisant, abstraction faite des chutes dans les éléments. La capacité C 2 intègre le courant fourni par la self et délivre la tension V_s. Nous pouvons établir les relations suivantes : soit θ le temps de conduction d'une période et $T-\theta$ le temps de non conduction de cette même période.

$$L \frac{di}{dt} = E$$

$$L \frac{di}{dt} = V_p - V_s \quad \text{pour } 0 < t < \theta$$

$$L \frac{di}{dt} = - V_s \quad \text{pour } \theta < t < T$$

d'où

$$I[t] = I_o + \int_0^t \frac{V_p - V_s}{L} dt \quad \text{pour } 0 < t < \theta$$

$$I[t] = I_o + \int_0^\theta \frac{V_p - V_s}{L} dt + \int_\theta^t \frac{V_s}{L} dt \quad \text{pour } \theta < t < T$$

I_o étant le courant I à l'instant t=0
à l'équilibre, on a I[T] = I[0]

$$\int_0^\theta \frac{V_p - V_s}{L} dt = \int_\theta^T \frac{V_s}{L} dt$$

ou

$[V_p - V_s]\ \theta = V_s[T-\theta]$

si l'on admet que V_s demeure sensiblement constant tout au long de la période T

$$\frac{V_p - V_s}{T - \theta} = \frac{V_s}{\theta}$$

$$\frac{V_p}{T} = \frac{V_s}{\theta}$$

d'ou la relation fondamentale

$$\frac{\theta}{T} = \frac{V_s}{V_p}$$

Remarque : Lorsque le courant I s'annule, ces équations ne sont plus les seules à considérer.

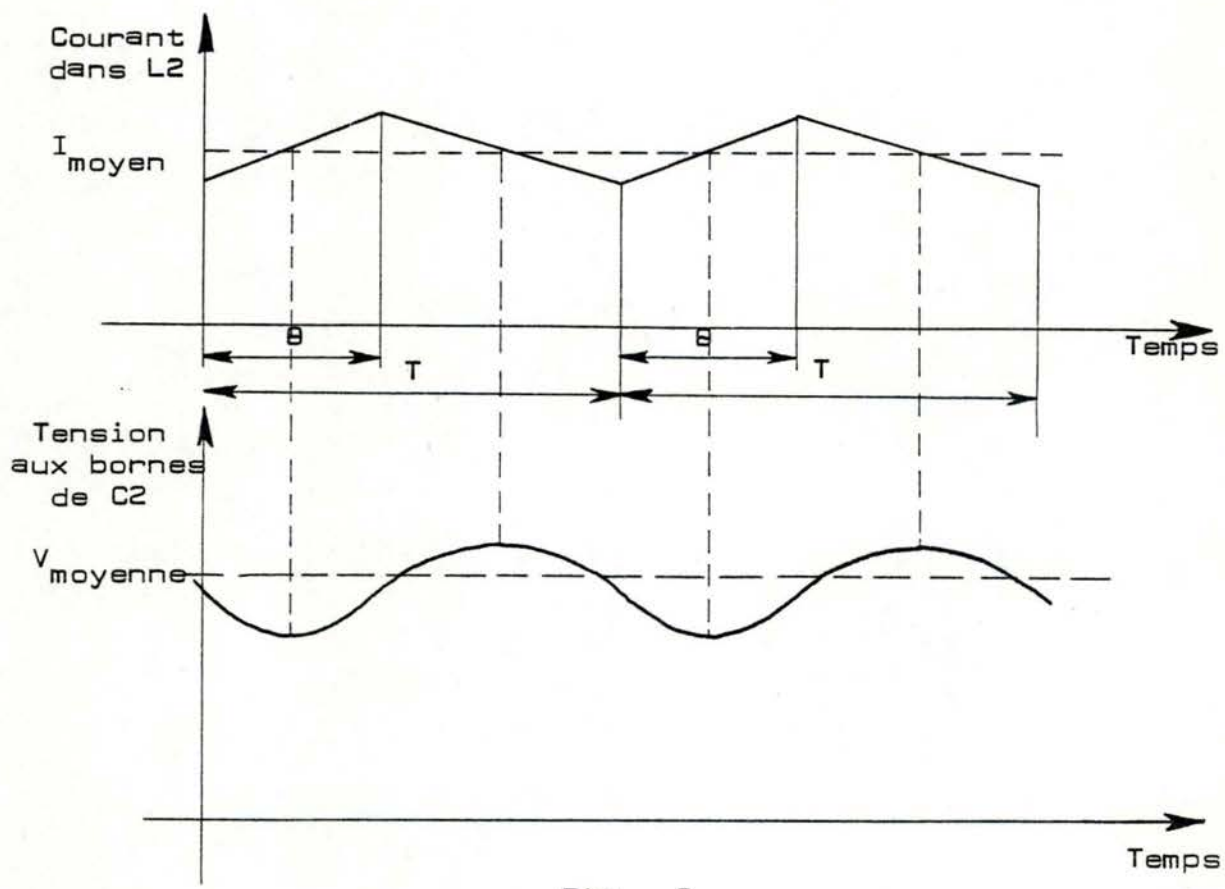


Fig 8

2.1.2. Le processus d'optimisation.

1. La droite de charge fictive.

Une charge Z alimentée par l'intermédiaire du convertisseur à découpe continu-continu peut être assimilée à une charge fictive Z_f alimentée sans l'intermédiaire de ce mécanisme.

Il existe une relation qui unit les deux droites de charge Z et Z_f. Pour une courbe I (v), une charge Z et un rapport $\frac{\theta}{T}$ donnés, nous avons les relations suivantes.

$$\frac{V_s}{V_{al}} = \frac{\theta}{T} = K$$

$$Z_f = \frac{V_{al}}{I_{al}}$$

$$Z = \frac{V_s}{I_s}$$

$$V_{al}\ I_{al} = V_s\ I_s$$

Nous pouvons en déduire que

$$Z_f = \frac{V_{al}}{I_{al}} = \frac{V_{al}^2}{V_s\ I_s} = \frac{V_s^2}{K^2\ V_s\ I_s} = K^{-2}\ \frac{V_s}{I_s}$$

$$Z_f = K^{-2}\ Z$$

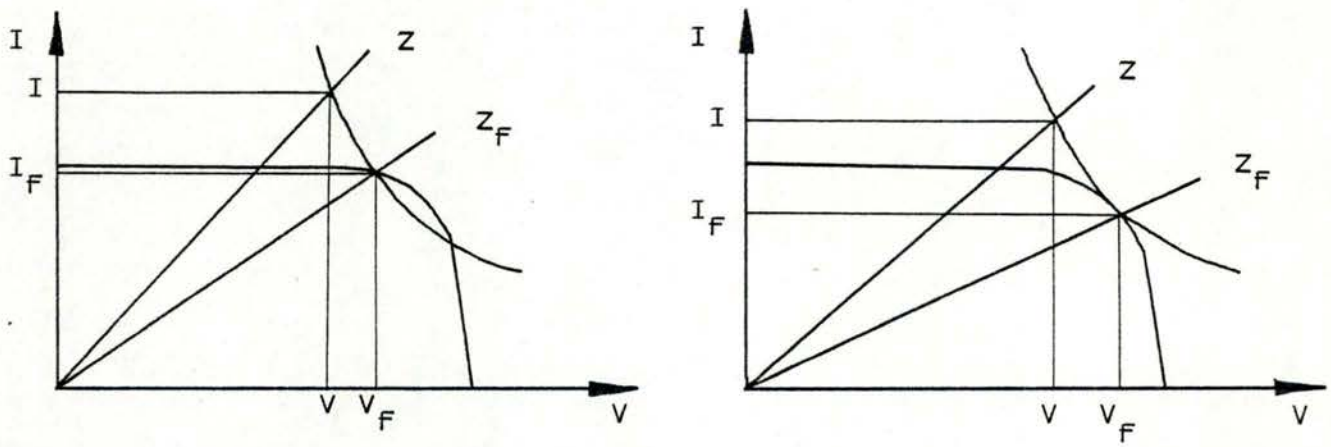


Fig 9

Nous pouvons dégager trois positions particulières de la droite de charge fictive en fonction de trois valeurs de $\frac{\theta}{T} = K$.

1. Lorsque $\theta = T$ $[K=1]$, les deux droites de charge sont confondues.

2. Lorsque $\theta = o$ $[K = o]$, la droite de charge fictive est confondue avec l'axe V.

3. Lorsque $0 < \theta < T$ $[0 < k < 1]$, la droite de charge se situe entre la droite de charge réelle et l'axe V.

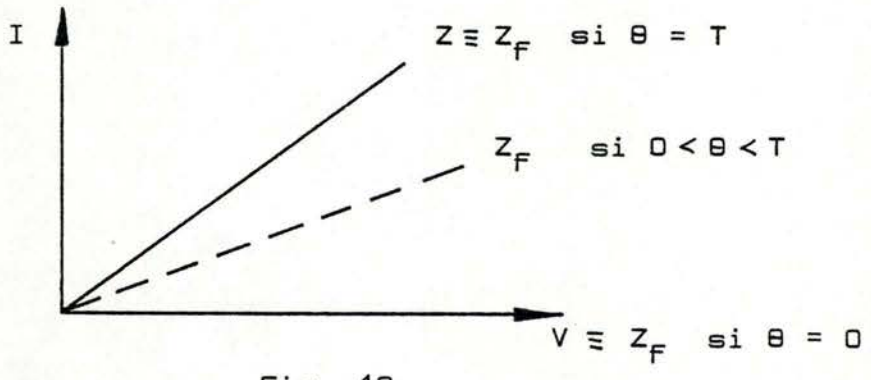


Fig 10

Nous pouvons décrire également le cheminement de la droite de charge fictive Z_F dans le graphique I (v) en fonction d'une variation $\Delta\theta$ du temps de conduction de T_1.

1. Si $\Delta\theta > 0$ (avec $\theta + \Delta\theta \leqslant T$) alors la droite de charge fictive se rapproche de la droite de charge réelle.

2. Si $\Delta\theta < 0$ [avec $\theta + \Delta\theta \geqslant 0$] alors la droite de charge fictive se rapproche de l'axe V.

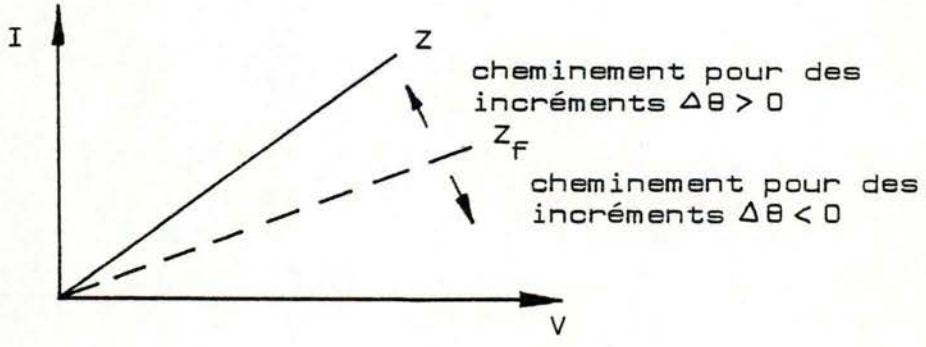


Fig 11

2. Relation entre une variation du temps de conduction et une variation du niveau de la puissance.

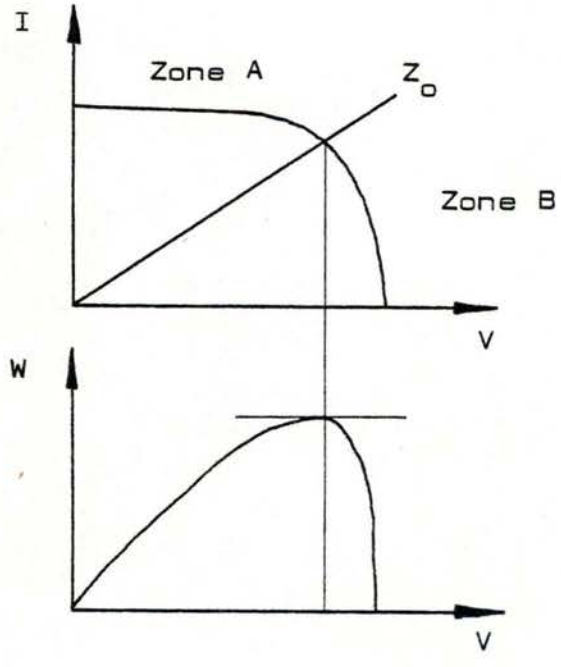
Le principe de convergence est basé sur une loi qui unit une variation de l'angle de fermeture de la vanne de puissance et une variation du niveau de puissance.

Au cours de chaque nouvelle période le microprocesseur doit modifier au coup par coup l'angle de fermeture pour maintenir la droite de charge fictive aussi voisine que possible de la droite de charge optimale. Deux phénomènes contrarient ses actions.

1. Pour une période de conduction donnée une variation de la charge modifie la position de la droite de charge fictive.

2. Une variation de la courbe I (v) modifie la position de la droite de charge optimale.

L'influence d'une variation de l'angle de fermeture sur une variation de puissance est différente selon la position de Z_f par rapport à Z_o. En effet : pour des incréments positifs, si Z_f est au delà de Z_o le niveau de puissance augmente; si au contraire Z_f est en deça de Z_o le niveau de puissance diminue. Pour des incréments négatifs, c'est l'inverse.



	$Z_F \in$ Zone A	$Z_F \in$ Zone B
$\Delta\theta < 0$	$\Delta W > 0$	$\Delta W < 0$
$\Delta\theta > 0$	$\Delta W < 0$	$\Delta W > 0$

Fig 12

2.1.3. Algorithme.

Au cours de chaque période, le microprocesseur assure l'enclenchement et le déclenchement de la vanne de puissance. Il calcule en outre l'écart de puissance entre la puissance lue à la période précédente et celle de la période en cours. Il décide en fonction de l'écart de puissance du signe de l'incrément à appliquer à l'angle de fermeture de la vanne à la prochaine période.

Cela suppose l'initialisation de l'angle de fermeture de la puissance de référence et du sens de l'incrément.

Remarques.

1. Il faut que la variation de Z ou de Z_o soit lente par rapport aux calculs.
2. D'une part, si les panneaux solaires ne sont pas tous dans les mêmes conditions, leur connection en série et en parallèle peut donner lieu à une courbe I (v) avec plusieurs optima voisins. D'autre part, le courant I n'est pas tout à fait constant. De ce fait, la puissance instantannée est différente de la puissance moyenne sur la période. En cas de variation négative de l'écart de puissance, le microprocesseur ne modifie le signe de l'incrément que si elle excède un certain seuil.

2.1.4. Configuration.

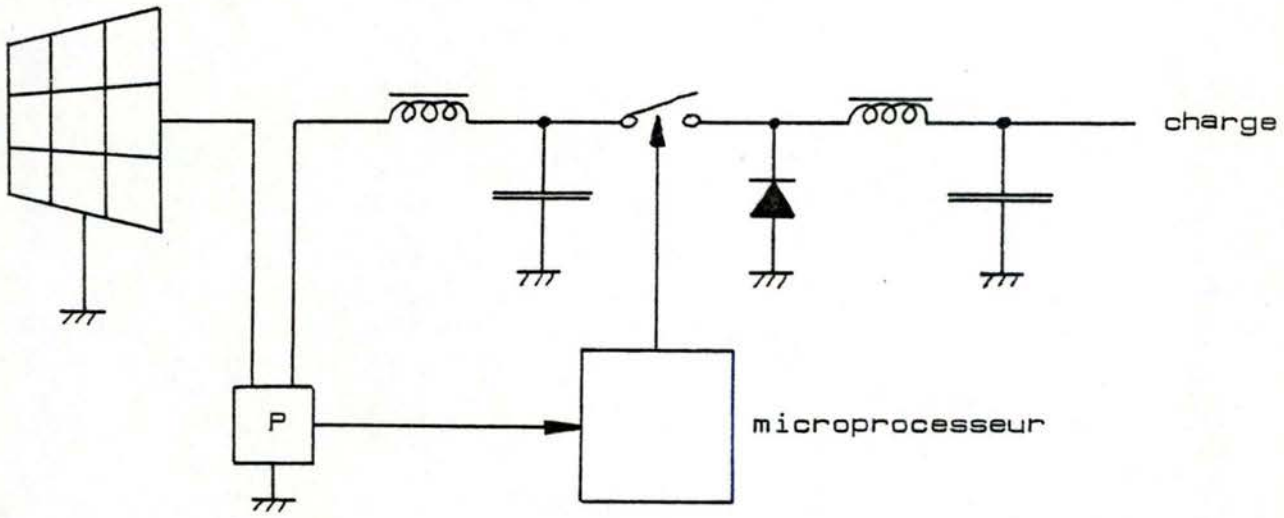


Fig 13

2.2 Transformation de puissance continue en puissance alternative.

Puisque nous ne devons pas exclure les applications à courant alternatif et que par ailleurs nous envisageons la possibilité de couplage de la centrale solaire à un réseau alternatif public, il faut être en mesure de produire une tension alternative à partir de la tension solaire continue.

Pour alimenter une charge à courant alternatif à partir d'une source continue, il faut passer par l'intermédiaire d'un onduleur . Son rôle est de commuter en alternance la source continue dans chacune des deux polarités possibles sur la charge alternative. L'onde ainsi produite aux bornes de la charge, à la fréquence du réseau que l'on veut reconstituer, est une onde carrée, riche en harmoniques et généralement mal adaptée à la charge.

Une amélioration possible à la pollution par harmoniques de fréquence est possible par l'usage d'un hacheur utilisé avec modulation de la largeur de hachage.

2.2.1. Le hacheur.

I. Schéma de principe.

Les hacheurs sont des interrupteurs unidirectionnels commandés aussi bien à l'ouverture qu'à la fermeture. Le panneau solaire délivre une tension V_p à l'entrée du hacheur. Les actions de fermeture puis d'ouverture sont exécutées sur T 1 pendant la première demi-période et sur T 2 pendant la seconde.

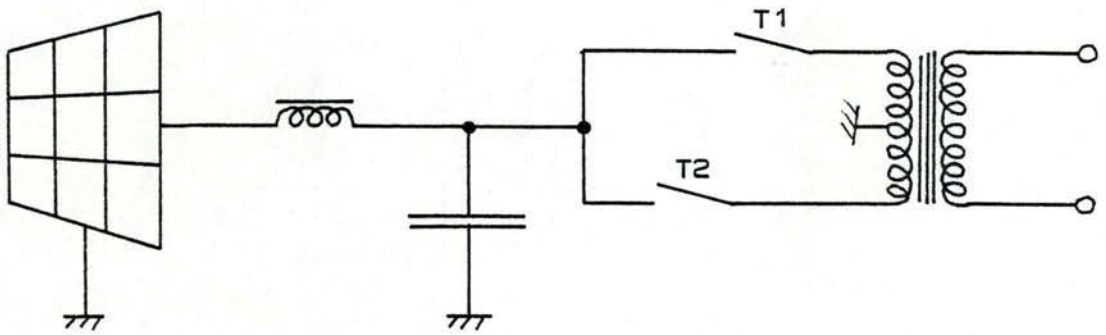


Fig 14

2. Principe de la modulation.

La transformation de la puissance continue en puissance alternative repose sur une transformation d'aire. Divisons la demi-période de la fonction sinusoïdale à constituer en un nombre n entier impair de portions identiques que nous appellerons divisions de période. Il faut pour chaque division de période délivrer un créneau de tension V_p pendant un temps θ_i centré sur l'intervalle $[i - 1 , i]$, de telle sorte que le produit $V_p\ \theta_i$ soit aussi proche que possible de la surface S_i de la division de période.

Soit une demi-période de la sinusoïde $\sin \omega t$ avec $\omega = 2\pi f$ scindée en n divisions de périodes identiques. Les surfaces comprises entre la sinusoïde, l'axe T et les limites des divisions de période valent

$$s_i = \int_{(i-1)(\frac{\pi}{n})}^{i(\frac{\pi}{n})} \sin \omega t \quad d[\omega t] \qquad \omega = 2\pi f \qquad i = 1, 2, \ldots, n$$

La fonction sinusoïdale que l'on désire reconstituer est $V_{max} \sin \omega t$ avec $\omega = 2\pi f$. La surface S_i relative à la division de période i vaut $V_{max}\ S_i$ et s'exprime en volt-seconde.

Soit le rapport entre la tension solaire V_p et la tension à obtenir V_{max} égal à k.

$$k = \frac{V_p}{V_{max}}$$

La règle de la transformation d'aire veut qu'au cours d'une division de période on délivre le produit $V_p\ \theta_i$ égal à $V_{max}\ S_i$

$$V_p\ \theta_i = V_{max}\ S_i$$

L'instant initial de θ_i correspond à la fermeture d'une des deux vannes; l'instant final de θ_i correspond à l'ouverture de cette même vanne. Nous pouvons déduire la formule qui donne l'instant de fermeture des vannes relatif au début de la demi-période.

$$t_i = \frac{2i+1}{2}\frac{\pi}{n} - \frac{1}{2}\frac{1}{k}\ S_i$$

2.2.2. Le processus de transformation.

Le microprocesseur mémorise la table des quantités de volt-seconde et des moments d'enclenchement associés à chacune des divisions de période. Il assure en outre le repérage du temps au cours de la période. Son rôle est de commander la fermeture d'une des deux vannes au moment pré-calculé; d'attendre que la quantité de volt-seconde délivrée soit aussi proche que possible de la valeur théorique. Il commande alors la fermeture de la vanne.

1. Table des quantités théoriques de volt-seconde et des moments d'enclenchement.

Le domaine de tout ordinateur est celui du fini et du discret. En simple longueur, un microprocesseur 8 bits peut présenter 256 objets différents.

On normalise la valeur des surfaces théoriques de telle sorte que la plus grande soit égale à 255. Si k' est le coefficient de normalisation, on a:

$$S_{max}\ k' = 255$$

Le temps est repéré au cours d'une période par un compteur qui progresse de 0 à 255 instants élémentaires entre le début et la fin de la période.

Exemple chiffré : 22 divisions de période

512 temps élémentaires par période

$k = V_p/V_{max} = 1.$

Divisions de période	Aire	Moment d'enclenchement	Début de la période en temps élémentaires
1	36	10	0
2	106	30	23
3	167	51	47
4	215	72	70
5	245	94	93
6	255	116	116
7	245	140	140
8	215	165	163
9	167	190	186
10	106	216	209
11	36	243	233
12	36	10	0
13	106	30	23
14	167	51	47
15	215	72	70
16	245	94	93
17	255	116	166
18	245	140	140
19	215	165	163
20	167	190	186
21	106	216	209
22	36	243	233

Tab 15

Remarque :

Chacune des 22 divisions de période comprend 23 temps élémentaires. Il en résulte alors qu'une période de sinusoïde ne fait que 506 temps élémentaires au lieu de 512. Par conséquent, pour réaliser ces 512 temps élémentaires, il faut ajouter 6 temps "de bourrage" à raison de 3 entre la 11 ième et la 12 ième division et 3 entre deux périodes.

2. Détection du moment de déclenchement.

On déclenche les vannes au terme de chaque θ_i. L'instant $t_e + \theta_i$ est en relation avec t_d par l'égalité

$$V_p \; \theta_i = V_{max} \; S_i$$

A chaque instant élémentaire, on délivre une certaine quantité de volts-seconde.

$$\theta_i = t_1 + \ldots + t_n = n \; t_i$$

$$V_p = \frac{1}{n} \left[V_{p_1} + \ldots + V_{p_n} \right]$$

$$V_{max} \; S_i = \frac{1}{n} \left[V_{p_1} + \ldots + V_{p_n} \right] n \; t_i$$

La quantité à déduire à chaque instant élémentaire n'est autre que la tension solaire.

Exemple: soit à délivrer 36 unités de Volts-seconde par à coup de 11 unités.

à $t = t_1$ 36 - 11 il en reste 25
à $t = t_2$ 25 - 11 il en reste 14
à $t = t_3$ 14 - 11 il en reste 3
à $t = t_4$ 3 - 11 il y en a 7 de trop

Nous pouvons prendre en considération à la nouvelle division de période le surplus délivré lors de la division de période antérieure. Nous le déduisons de la nouvelle quantité théorique.

Reconstitution d'une sinusoïde à partir d'un hacheur: 22 divisions de période

512 moments élémentaires

$$k = \frac{V_p}{V_{max}} = 1$$

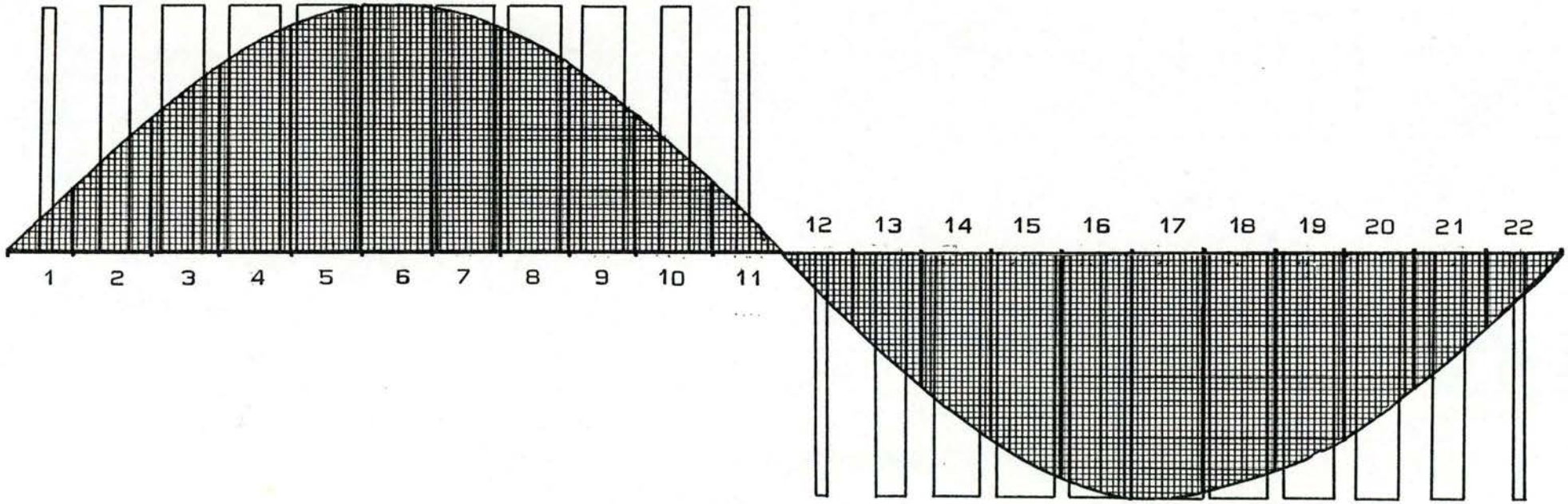


Fig 16

3. Optimisation générale.

Nous avons vu que l'algorithme de transformation de la puissance continue en puissance alternative utilise une table de quantités de volt-seconde et des moments d'enclenchement pour commander l'ouverture et la fermeture des vannes du hacheur. Plusieurs tables peuvent être calculées pour plusieurs valeurs de k [$k = \frac{V_p}{V_{max}}$]
Une seule table sert à chaque période. C'est l'algorithme d'optimisation générale qui décide de la table à prendre en considération.

L'algorithme d'optimisation générale assure l'optimum de puissance. A cet effet, le microprocesseur calcule l'écart signé entre la puissance lue à la période précédante et celle de la période en cours. Il décide en fonction de cet écart de changer de table des quantités de volt-seconde et des moments d'enclenchement.

Le seuil de variation de k est déterminé de telle sorte qu'il corresponde à une variation d'un instant élémentaire sur la valeur de θ_i normalisée. Le nombre de tables à prendre en considération dépendra de l'expérience.

2.2.3. Configuration.

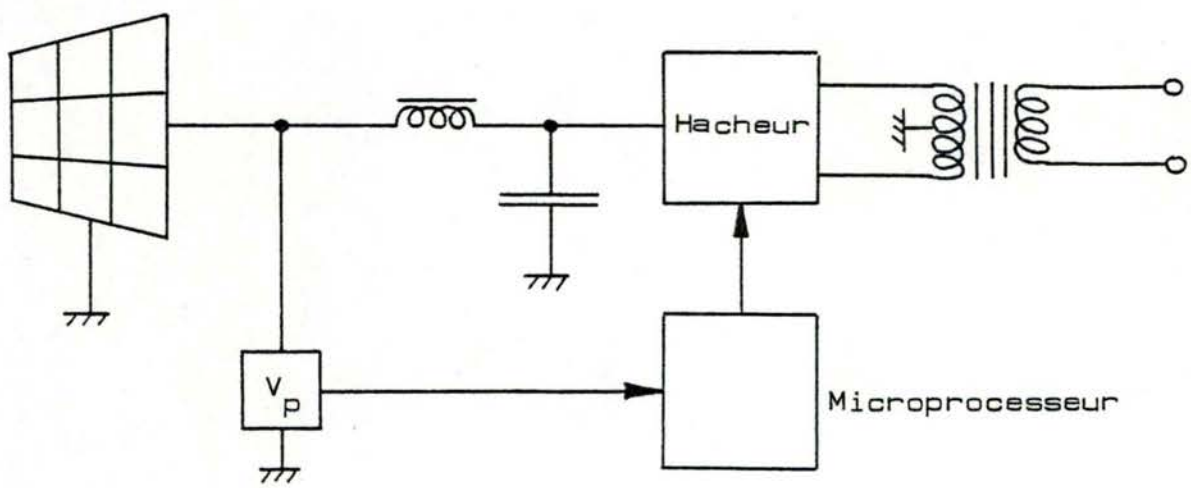


Fig 17

2.3. ACCUMULATION TEMPORAIRE D'ENERGIE ET GESTION DE PRIORITE DANS LES CHARGES.

On a vu la nécessité de faire appel aux accumulateurs dans les installations solaires. Ceux-ci serviront:

- comme source d'alimentation durant la période d'impossibilité d'obtenir un courant des cellules solaires.
- comme source d'appoint pendant les périodes de manque d'ensoleillement.

La charge des accumulateurs ne pourra s'effectuer, à partir des cellules solaires, que pendant les périodes d'ensoleillement.

Le schéma de la figure 18 donne l'allure de deux courbes de puissance échelonnées sur une période d'une journée. L'une est la puissance disponible aux panneaux solaires, l'autre est celle demandée par les charges de l'installation.

——— panneaux solaires

- - - - charges

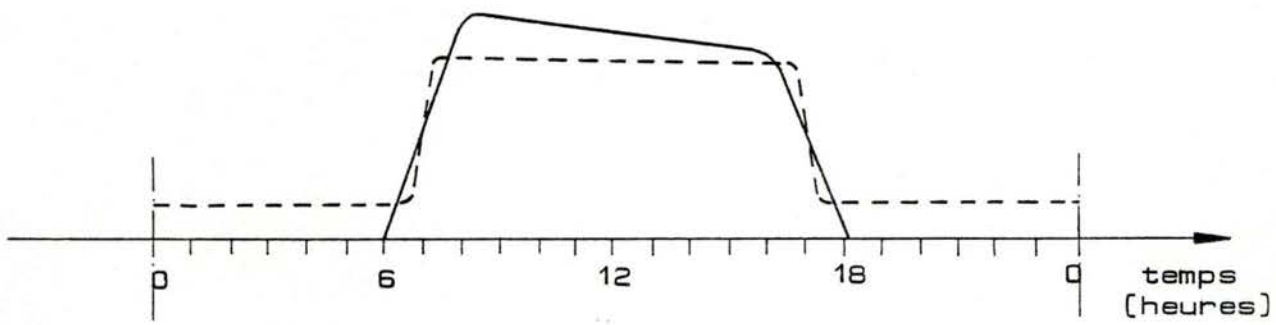


Fig 18

L'accumulateur emmagasine l'énergie électrique disponible aux panneaux solaires qui excède le besoin. Il la restitue à volonté quand l'énergie disponible aux panneaux solaires vient à manquer ou à faire défaut. Lorsque l'accumulateur restitue l'énergie, une classe de charges appelée charges non-prioritaires est automatiquement déconnectée pour assurer le maximum d'autonomie aux charges prioritaires.

Configuration.

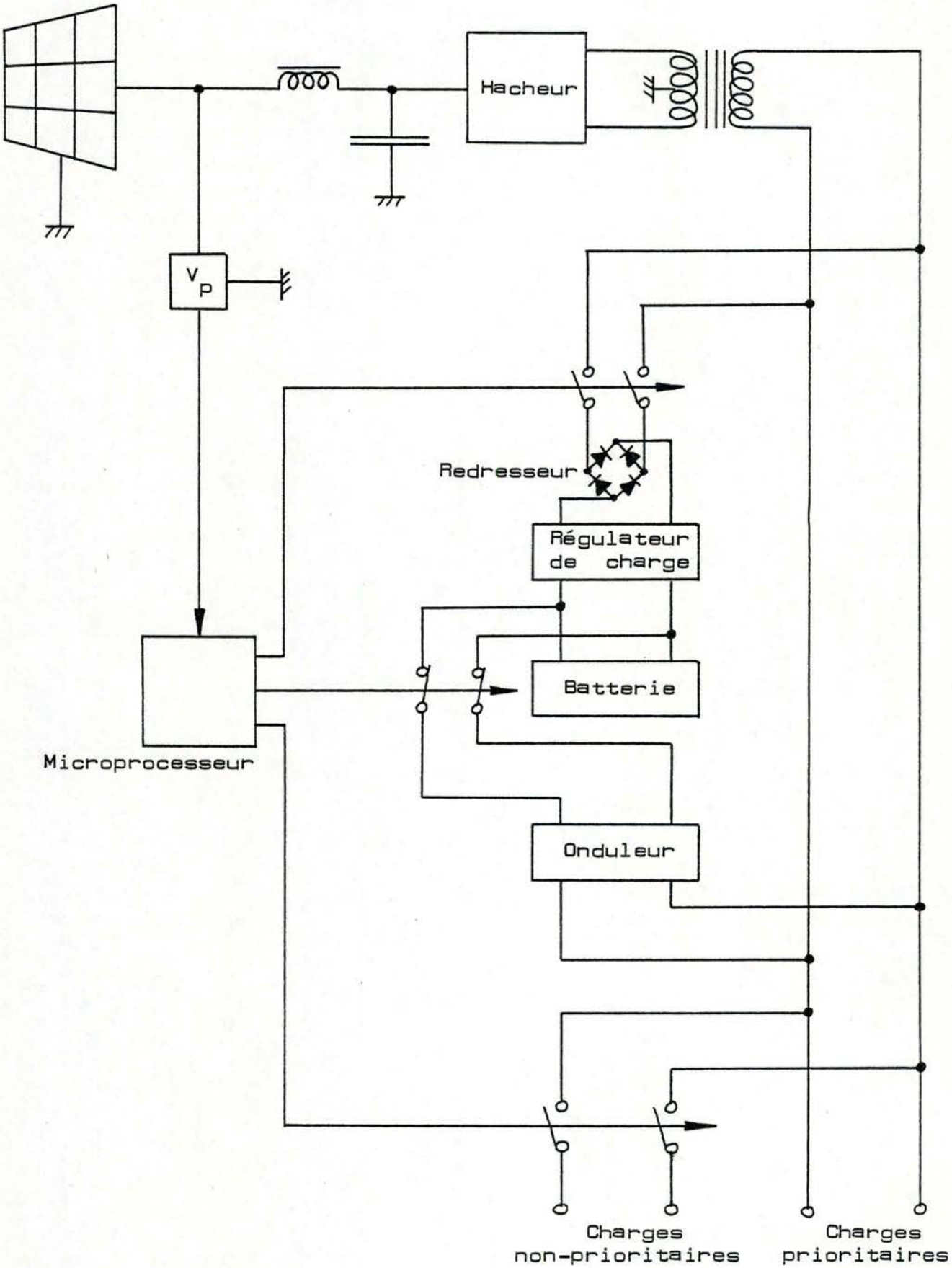


Fig 19

2.4 REGULATION DU COS φ.

2.4.I Amélioration du cos φ.

Dans le cas d'un réseau à courant alternatif, les appareils purement résistifs ne consomment que de la puissance active. Au contraire, les appareils avec une composante selfique ou capacitive véhiculent des proportions variables de puissance active et réactive.

Aussi, la distribution de puissance est caractérisée dans le cas général par le triangle des puissances tel celui de la figure 20.

P_r = puissance réactive.
P_a = puissance active.
P = puissance apparente.

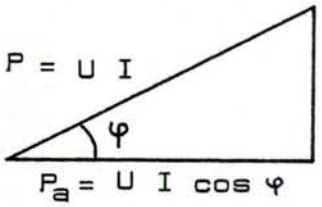


Fig 20

Il en résulte :

- qu'une puissance active déterminée est fournie à partir d'une puissance apparente d'autant plus faible que le cos φ est éleve.
- qu'une puissance apparente déterminée peut donner lieu à une puissance active d'autant plus élevée que le cos φ est élevé.

De plus, sous une tension constante, la puissance active sera fournie avec un courant d'autant plus intense que le cos φ est faible. Or, ce qui limite généralement la puissance d'une machine, c'est son échauffement qui est précisément fonction du carré du courant. Par ailleurs, une distribution d'énergie électrique est limitée par la chute de tension dans la ligne d'alimentation, laquelle est aussi fonction du courant. Par conséquent, le cos φ doit être aussi proche que possible de 1, pour éviter les pertes préjudiciables d'énergie.

Une solution pour améliorer le cos φ consiste à produire à l'endroit voulu une énergie réactive nécessaire à la compensation en faisant appel à un générateur d'énergie réactive convenable, le plus classique et le plus simple étant le condensateur statique.

Boucle de régulation.

Le cos φ est ramené à une valeur proche de 1 par l'action du microprocesseur qui introduit ou retire des capacités. Celles-ci sont disposées en parallèle sur la charge. La tentative du microprocesseur pour annuler φ s'effectue au coup par coup. A chaque itération, l'action consiste en se basant sur la dernière mesure de soit d'ajouter ou de retirer une capacité suivant le sens du dernier déphasage connu.

Configuration.

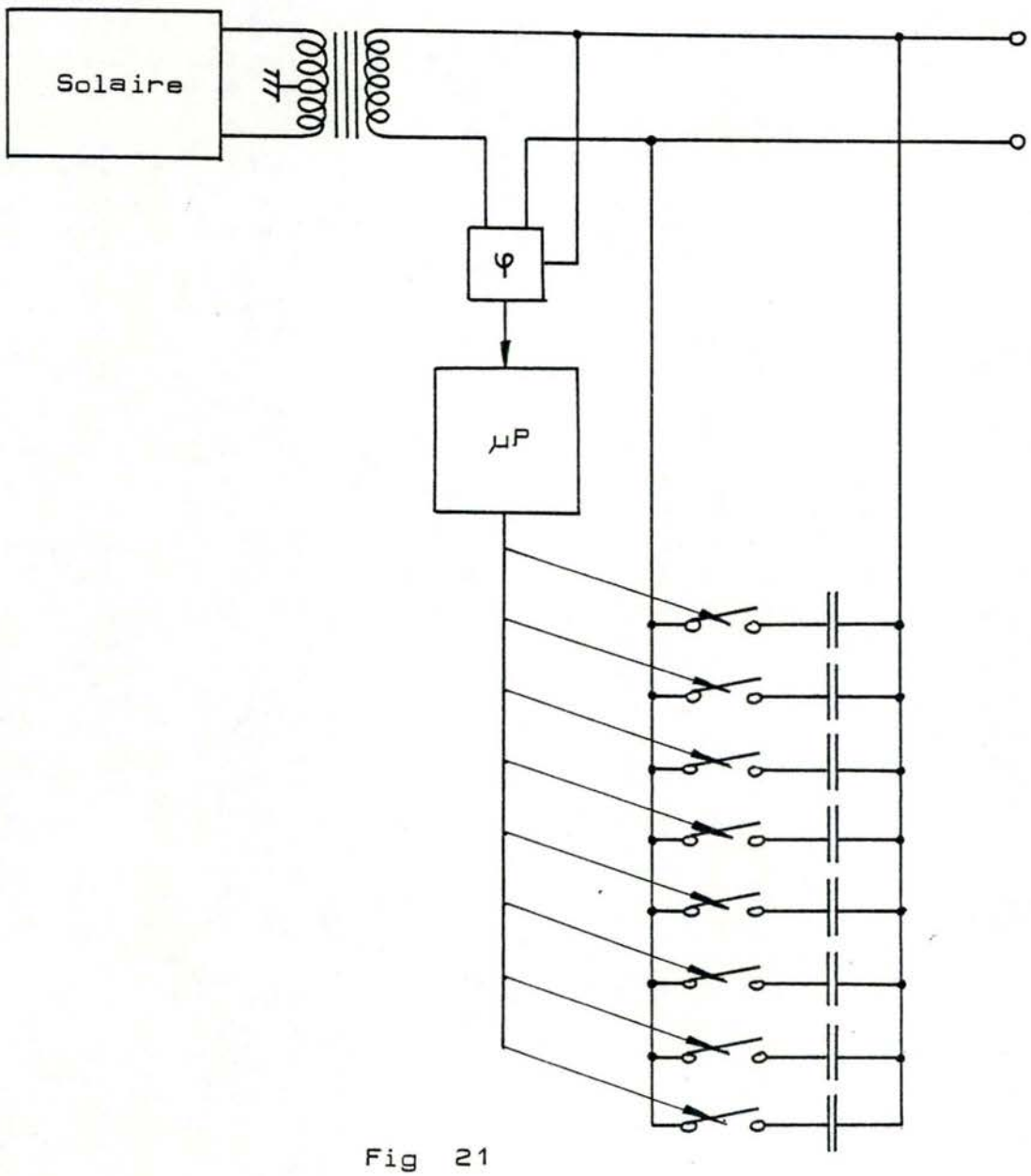


Fig 21

2.4.2. Mesure du déphasage entre U et I.

L'amélioration du cos φ suppose à chaque nouvelle période la mesure de l'angle de déphasage φ. Considérons deux vecteurs $\overrightarrow{OA}$ et $\overrightarrow{OB}$ faisant entre eux un angle φ et tournant ensemble dans le même sens arbitraire d'un mouvement circulaire uniforme de vitesse angulaire ω. Les deux sinusoïdes représentatives obtenues par projection sur l'axe des ordonnées des extrémités des deux vecteurs sont décalées l'une par rapport à l'autre de l'angle φ.

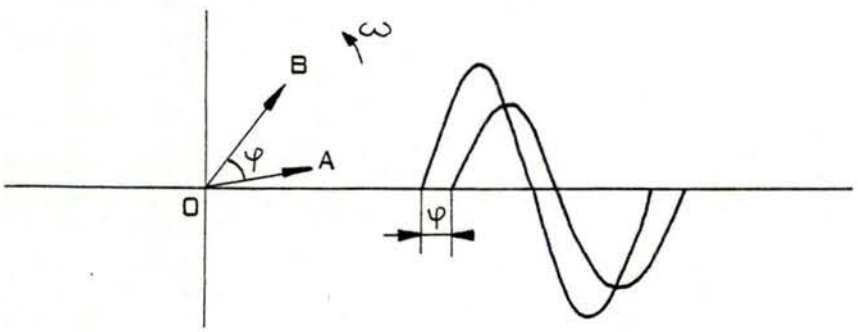


Fig 22

Les deux fonctions

- ne sont pas toujours de même signe.
- s'annulent à des temps différents.
- deviennent maximum, positives et négatives, à des moments différents.

La mesure du déphasage consiste à évaluer le temps qui s'écoule entre deux annulations consécutives montantes (ou descendantes) l'une de U et l'autre de I. Ces points particuliers peuvent être repérés à l'aide de deux amplis à seuil.

1. Schéma de principe de l'ampli à seuil.

Pour une sinusoïde tension.

Nous distinguons trois parties dans le schéma de principe de la figure 23. A gauche, la source alternative; au centre l'ampli à seuil et à droite la logique de l'impulsion de sortie.

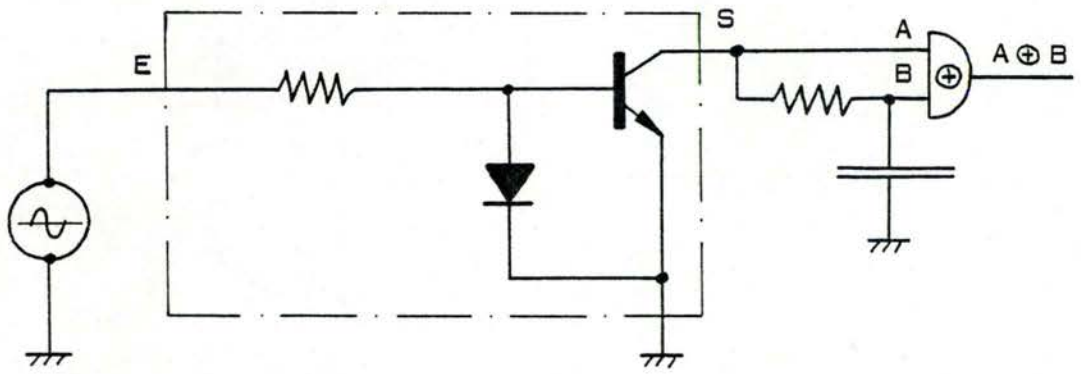


Fig 23

La source alternative délivre une tension sinusoïdale à l'entrée E de l'ampli à seuil. Celui-ci a en sortie une variable logique à l'image de la tension sinusoïdale d'entrée. Cette variable est à 1 pendant l'alternance positive et à 0 pendant l'alternance négative.

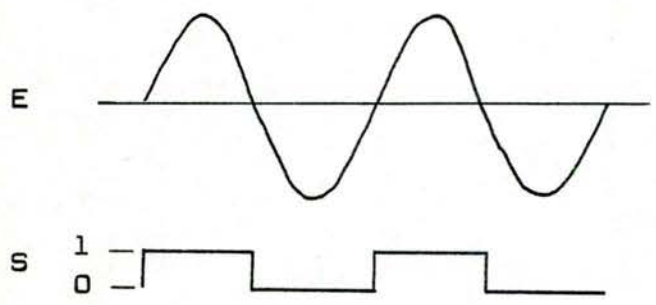


Fig 24

La logique de sortie est construite autour d'un OU exclusif entre d'une part la variable logique de l'ampli à seuil et d'autre part, cette même variable logique légèrement décalée dans le temps.

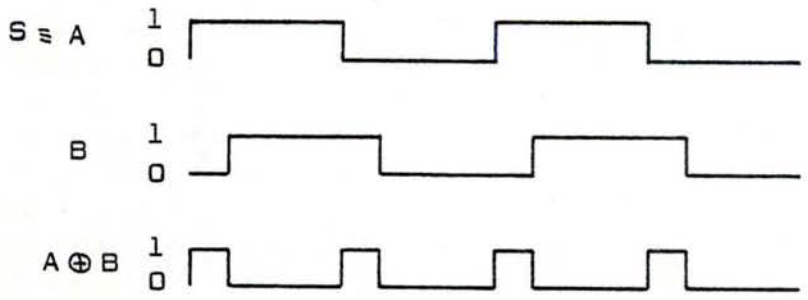


Fig 25

Chaque impulsion est l'image d'un passage par 0 de la sinusoïde. Un relevé de l'état de la variable logique S directement consécutive à l'impulsion permet de décider si la transition est de négatif vers positif ou positif vers négatif. Nous appellerons impulsion positive une impulsion relative à une transition de la sinusoïde de négatif vers positif et impulsion négative l'inverse.

Pour une sinusoïde courant.

Un courant peut toujours être ramené à une tension par l'intermédiaire d'un shunt. Nous sommes alors ramenés au cas précédent.

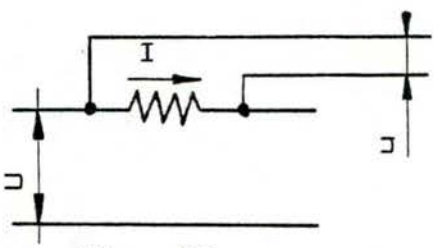


Fig 26

2. Principe de la mesure de φ.

Les impulsions à la sortie des amplis à seuil V et I se suivent en alternance et de façon asynchrone. Il faut être en mesure de repérer deux implusions positives consécutives (l'une de V et l'autre de I) tout en évaluant le temps qui s'écoule entre ces deux impulsions. Or, rien ne dit à priori laquelle va se présenter la première.

Admettons que l'on n'a pas encore reçu les deux impulsions positives consécutives V et I et que la dernière impulsion reçue,

si toutefois il y en a une, était négative. Nous sommes dirons nous dans un état d'attente initial dénommé " Etat 0". La nouvelle impulsion se présente, peu importe qu'elle provienne de V ou de I. Si elle est négative, nous sommes par hypothèse ramenés au cas précédent. Si au contraire, elle est positive, nous nous trouvons dans un nouvel état d'attente " Etat 1". En effet, à ce stade, on n'attend plus qu'une impulsion positive. Si l'impulsion suivanteest négative, de nouveau nous retournons dans l'état 0. Mais si elle est positive, nous passons à l'état 2; état de fin d'attente des deux impulsions positives.

Il suffit dès lors pour évaluer le déphasage de compter le temps entre le passage de l'état I à l'état 2.

Nous nous rendons compte que l'algorithme peut être formalisé par la notion d'automate. Un automate est défini par :

1. Un ensemble E d'états de l'automate.
2. Un ensemble fini S de symboles d'entrée.
3. Un état initial E_o
4. Une fonction de transition F, qui fait correspondre à chaque couple (E_i, S_i) un état.
5. Une fonction de sortie qui fait correspondre à chaque état un symbole de sortie.

- L'automate de la mesure de φ comporte 3 états:
  Etat 0, Etat 1 et Etat 2.
- Les symboles d'entrées sont les impulsions (positives et négatives) de V et de I.
- La fonction de transition

	Impulsion +	Impulsion −
E_o	E_1	E_o
E_1	E_2	E_o
E_2	−	E_o

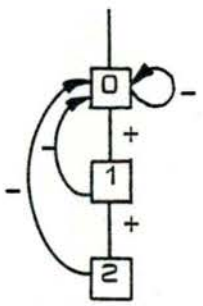


Fig 27

- Les symboles de sortie : l'initialisation et la lecture du compteur de temps.
- Fonction de sortie.

E 1 initialiser le compteur.

E 2 lire le compteur.

Pour connaître le sens du déphasage, il suffit de savoir d'où vient la dernière impulsion positive. Si elle provient de V , alors I est en avance sur V ; si par contre elle vient de I c'est l'inverse.

2.5 INTERCONNEXION AVEC LE RESEAU PUBLIC.

2.5.I. Régulation de la puissance réactive par le réseau.

Lors du couplage au réseau public, l'énergie du réseau est appelée à suppléer la carence éventuelle de l'énergie solaire soit partiellement, soit totalement. L'énergie du réseau vient se greffer directement aux bornes de la charge. Dans ce cas, la source solaire n'est plus le seul générateur à prendre en considération.

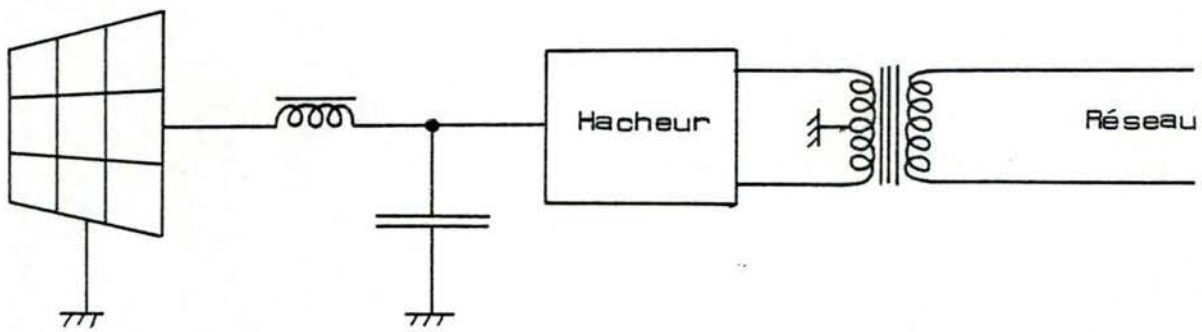


Fig 28

Le schéma électrique équivalent du couplage est représenté par la figure 29.

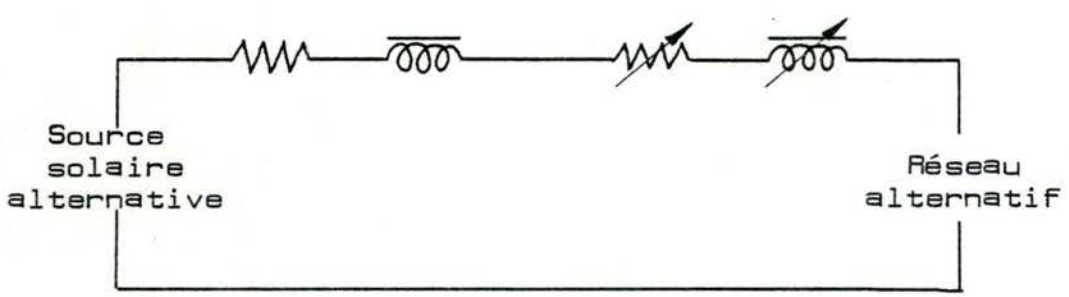


Fig 29

Différents éléments parasites du couplage sont à prendre en considération :

- les éléments de la cellule de tête
- les éléments du hacheur
- les éléments du transformateur

Ces éléments sont de nature résistive et / ou selfique.

Le diagramme vectoriel correspondant au schéma électrique équivalent est le suivant :

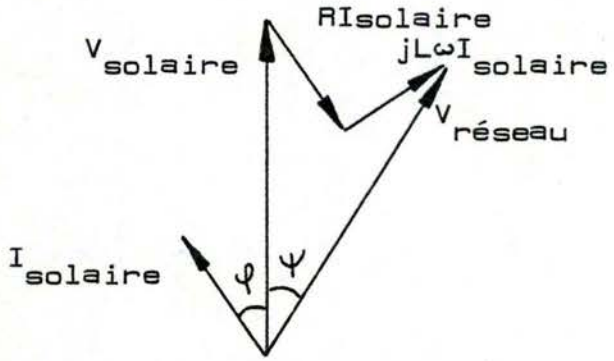


Fig 30

2.5.2. Principe de la régulation.

Il faut transférer de la puissance active et uniquement de la puissance active du générateur du réseau. La puissance active transférée au réseau [V_r I_a cos φ] est d'autant plus importante que l'angle φ est petit. L'idéal est que l'angle φ soit nul; de cette manière toute la puissance solaire injectée au réseau alternatif est de la puissance active. Dans ce cas, le diagramme vectoriel du schéma électrique de couplage est le suivant :

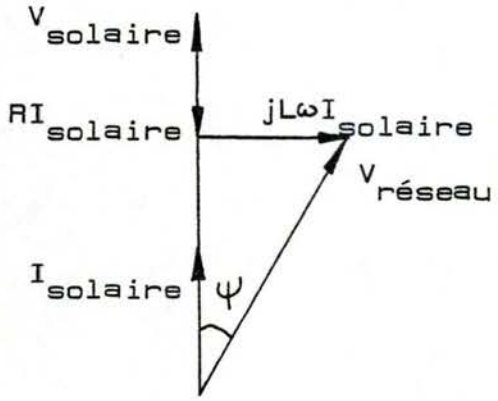


Fig 31

Il y a toujours un angle Ψ de décalage entre V solaire et V réseau. Cet angle est le miroir de l' impédance de couplage qui n' est pas purement résistive. Si l' angle Ψ est idéal, il y a transfert de puissance idéal; c' est-à-dire uniquement un transfert de puissance active de la source solaire au réseau. Tandis que dans le cas où l' angle Ψ n' est pas idéal, il y a quand même un transfert de puissance de la source solaire du réseau, mais ce transfert de puissance n' est pas idéal. En effet, il y a une proportion de puissance active et une proportion de puissance réactive.

Le problème est donc de créer l' angle Ψ tel que le courant solaire soit en phase avec la tension solaire. Autrement dit: créer l' angle Ψ tel que l' angle φ soit nul. Une manière d' agir sur l' angle Ψ est d' agir sur le bourrage de temps qui normalise à 50 Hz la fréquence de la sinusoïde solaire produite. Nous avons la possibilité soit de diminuer le bourrage de temps, ce qui revient à augmenter transitoirement la fréquence; soit d' augmenter le bourrage de temps, ce qui revient à diminuer transitoirement la fréquence. Si la fréquence augmente, alors l' angle φ augmente. Si au contraire la fréquence diminue, alors l' angle φ diminue.

2.5.3. Configuration.

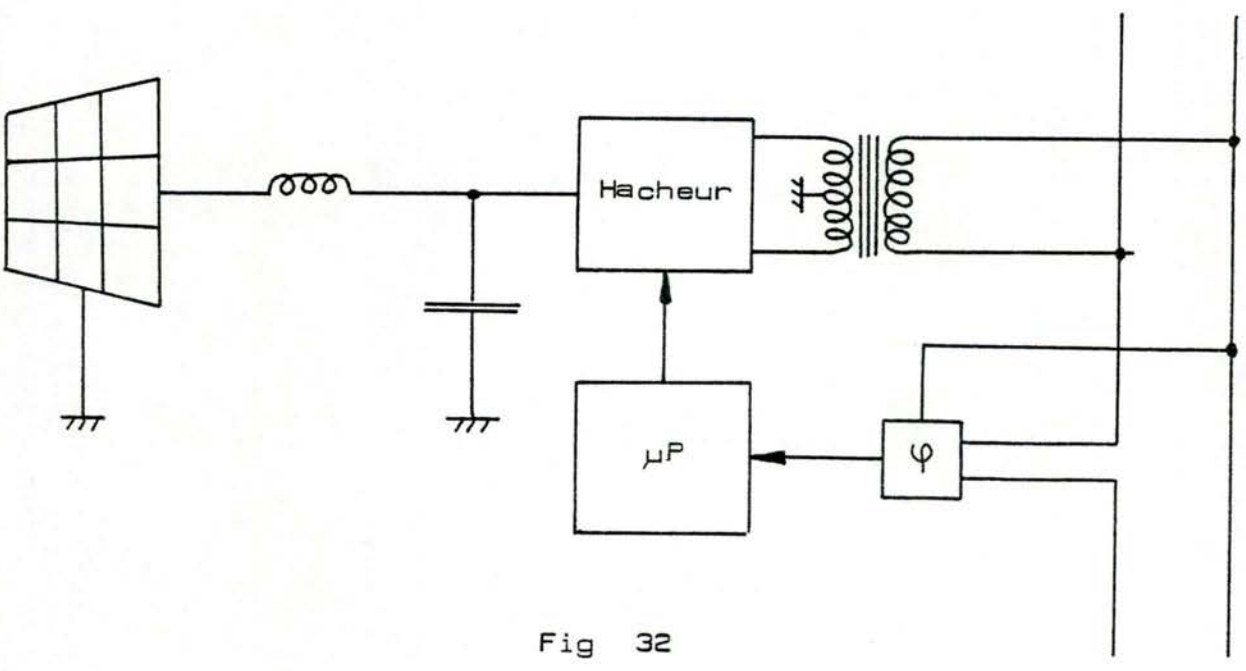


Fig 32

2.6. COUPLAGE DE LA CENTRALE SOLAIRE AU RESEAU.

L' interconnexion entre la centrale solaire et le réseau alternatif suppose deux choses. Il faut que premièrement, les deux tensions soient en concordance de phase et que deuxièmement l' amplitude de la tension solaire soit plus grande que celle du réseau.

Initialement, il n'y a aucune interaction entre la centrale solaire et le réseau. L' organe de couplage va donc:

1. amener les deux tensions en phase par une modification transitoire de la fréquence de la tension alternative solaire
2. ajuster l' amplitude de la tension solaire avant de coupler.

2.6.1. Processus.

Supposons que la fréquence du réseau soit différente de la fréquence de la tension solaire. Les deux vecteurs représentatifs des deux fonctions alternatives sinusoïdales ne sont pas du repos relatif: le déphasage n' est pas constant.

Nous connaissons d' une part la phase de V solaire puisque à chaque début de la période le système de transformation du continu en alternatif émet une interruption. D' autre part, un ampli à seuil connecté sur la tension sinusoïdale du réseau nous envoie une impulsion positive au début de chaque période et une impulsion négative à chaque demi-période. Lorsque l' impulsion solaire coïncide avec l' impulsion positive du réseau, il y a concordance de phase. Comme les impulsions ne peuvent être traitées que séquentiellement, il suffit de repérer deux impulsions consécutives telles que l' une corresponde à la tension solaire et que l' autre soit une impulsion positive du réseau. Il y a concordance de phase si le décalage temporel entre les deux est nul.

Deux cas favorables sont possibles:

1. l' impulsion solaire se présente avant l' impulsion positive du réseau.
2. l' inverse.

Toute autre combinaison est à rejeter. Nous pouvons déduire un automate à trois états avec sa fonction de transition.

- Etat 0: la dernière impulsion reçue est une impulsion négative du réseau.
- Etat 1: état d' attente soit d' une impulsion solaire, ou d' une impulsion positive du réseau.
- Etat 2: les deux impulsions attendues ont été reçues.

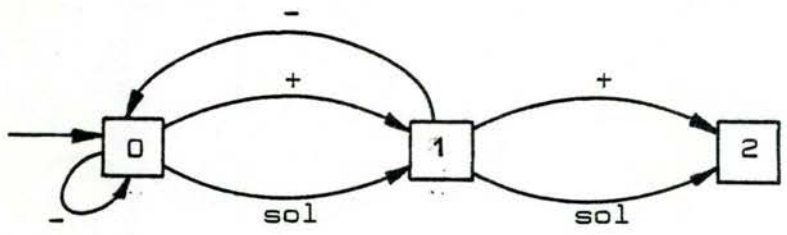


Fig 33

Il faut en outre que le temps qui s' écoule entre le passage de l' état 1 à l' état 2 soit bref. Au minimum ce sera le temps d' exécution des instructions entre le passage de 1 à 2. Nous ajoutons un troisième état, dont on transite de l' état 2, si le temps est jugé suffisamment court. Dans le cas contraire, il y a retour à l' état 0. A l' état 3, il faut directement réétablir la fréquence solaire à la même fréquence que le réseau.

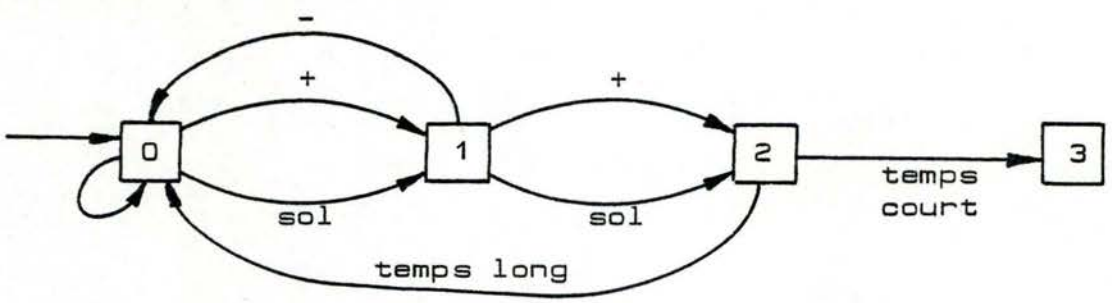


Fig 34

2.6.2. Configuration.

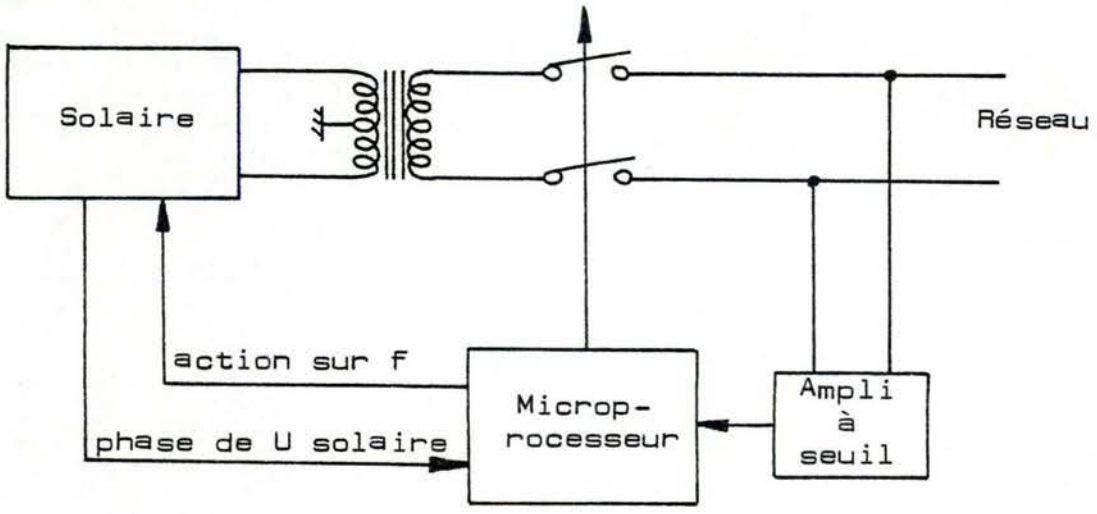


Fig 35

3. C O N C E P T I O N

3.I. CLASSIFICATION DES PROGRAMMES DE REGULATION.

Le but de l'action d'un programme est d'apporter une modification définie à son environnement. Par ailleurs, le comportement de l'environnement qui résulte des actes prévus au programme peut entrer dans l'une des trois catégories suivantes : [11]

1. Comportement passif de l'environnement
2. Comportement déterministe de l'environnement.
3. Comportement aléatoire de l'environnement.

Il en résulte que tout programme peut être classifié en fonction du comportement de son environnement sous l'action qu'il subit.

En particulier, tous les programmes de régulation de ce travail ainsi que tout programme de régulation en général entrent dans la deuxième catégorie.

I. Comportement passif de l'environnement.

La réaction du milieu extérieur est celle que présume le programme.Pour un comportement passif de l'environnement, on peut prédire la liste d'actions chronologiques à exécuter pour atteindre le but.

Telle est la réaction par exemple des éléments d'un lave-vaisselle séquencé par microprocesseur.Ils se mettent en oeuvre docilement dans l'ordre préétabli une fois pour toutes par le programme d'action. Il en résulte l'exécution de la tâche définie par le programme au moment prévu.

2. Comportement déterministe de l'environnement.

La réaction du milieu extérieur n'est pas nécessairement celle que présume le programme; mais elle est liée aux actes possibles des agents d'exécution par des lois connues. Dans le cas d'un comportement déterministe, on ne peut pas prédire une liste d'actions chronologiques,mais on peut prédire quel acte doit être effectué pour corriger le défaut des actes précédents.

A titre d'exemple, l'algorithme de l'optimisation primaire traque l'optimum de puissance du panneau solaire.Cette puissance mise en oeuvre dépend non seulement de l'angle d'ouverture de la vanne et de la charge mais aussi de l'éclairement et de la température du panneau. Un programme définissant l'angle d'ouverture de la vanne dans le temps est inconvenable pour la bonne raison que la loi de correspondance de la puissance maximum avec le temps n'est pas connue d'avance. Cependant, il existe une loi connue qui unit le sens d'une variation de l'angle d'ouverture avec une variation du niveau de puissance. Dès lors, il suffit à chaque itération de repéter la variation de l'angle d'ouverture dans le même sens qu'antérieurement si la variation de puissance est positive et dans le sens inverse si la variation de puissance est négative.

3. Comportement aléatoire de l'environnement.

La réaction du milieu extérieur n'est pas nécessairement celle que présume le programme et l'on ne connait aucune loi qui la relie aux agents d'exécution.

Dans ce cas, on ne peut ni prédire une liste chronologique d'actions ni même quel acte doit être effectué pour corriger le défaut des actes précédents.

3.2 LA DECOUPE FONCTIONNELLE.

Le système est découpé en blocs fonctionnels tels que chaque bloc fonctionnel corresponde à une grande fonction du système. Pour chaque bloc, sa fonction, ses entrées, ses sorties et ses relations avec les autres blocs doivent être définies.

Dans les aspects physiques, nous avons distingué trois types de régulation selon que la centrale solaire était utilisée en continu, en alternatif pour un usage autonome ou encore couplé à un réseau alternatif.

1. La régulation d'une centrale solaire à courant continu ne comprend que le bloc
   - optimisation primaire.
2. La régulation d'une centrale solaire autonome à courant alternatif comprend les blocs suivants :
   - transformation du continu en alternatif.
   - optimisation générale.
   - accumulation temporaire avec gestion de priorité dans les charges.
   - mesure de déphasage entre U et I.
   - amélioration du cos φ.

3. La régulation d'une centrale solaire d'appoint à un réseau alternatif comprend les blocs suivants :
   - transformation du continu en alternatif.
   - optimisation générale.
   - mesure de déphasage entre U et I
   - régulation de la puissance réactive par le réseau.
   - couplage au réseau de distribution.

I. Bloc fonctionnel "optimisation primaire"

Fonction.

Assurer l'optimisation du point de fonctionnement entre les panneaux solaires et la charge. Cette optimisation est réalisée dynamiquement dans le temps.

Entrée.

La puissance consommée au panneau solaire.

Sortie.

Les impulsions d'enclenchement et de déclenchement de la vanne de puissance du transformateur à courant continu.

Relation avec d'autres blocs.

Aucune : c'est le seul bloc fonctionnel utilisé.

2. Bloc fonctionnel " transformation du courant continu en alternatif ".

Fonction.

Transformer la puissance continue en puissance alternative en évitant les pertes par harmonique de fréquence.

Entrée.

1- La valeur de la tension continue en provenance du panneau solaire.

2- Le numéro de la table des aires théoriques relatives à la valeur de $k = V_p/V_{max}$.

3- Dans le cas d'une centrale couplée au réseau : une valeur du bourrage de temps.

Sortie.

Des impulsions d'ouverture et de fermeture à destination des vannes du hacheur.

Relation avec d'autres blocs.

Envoi d'une interruption qui synchronise tous les autres blocs excepté celui de mesure du déphasage entre U et I

3. Bloc fonctionnel " optimisation générale ".

Fonction.

Assurer l'optimisation du point de fonctionnement entre les panneaux solaires et la charge.

Entrée.

La puissance consommée du panneau solaire.

Sortie.

Le numéro de la table des aires théoriques.

Relation avec d'autres blocs.

Ce bloc est synchronisé par le bloc " transformation de la puissance continue en puissance alternative ".

4. Bloc fonctionnel " accumulation temporaire d'énergie et gestion de priorité dans les charges ".

Fonction.

1- Accumuler le trop plein d'énergie et restaurer l'énergie accumulée quand il y a pénurie.

2- Déconnecter les charges non-prioritaires quand la batterie est en restitution.

Entrée.

La valeur de la tension continue en provenance du panneau solaire.

Sortie.

La commande d'interrupteurs pour la batterie et pour les charges non-prioritaires.

Relation avec d'autres blocs.

Ce bloc est synchronisé par le bloc " transformation de la puissance continue en puissance alternative ".

5- Bloc fonctionnel " mesure de déphasage entre U et I ".

Fonction.

Evaluer le temps entre deux impulsions consécutives relatives à l'annulation des sinusoïdes U et I pour une transition de négatif vers positif.

Entrée.

Variable booléenne image soit de la tension sinusoïdale soit du courant sinusoïdal.

Sortie.

Le temps écoulé entre les deux impulsions et le sens du déphasage.

Relation avec d'autres blocs.

Ce bloc est synchronisé par des interruptions en provenance des amplis à seuil U et I.

6- Bloc fonctionnel " amélioration du cos φ ".

Fonction.

Assurer la meilleure valeur possible du cos

Entrée.

Le temps relatif du déphasage entre U et I ainsi que le sens de ce déphasage.

Sortie.

La commande d'ouverture ou de fermeture sur les interrupteurs de la batterie de capacité.

Relation avec d'autres blocs.

Ce bloc est synchronisé par le bloc " transformation de la puissance continue en puissance alternative ".

7- Bloc fonctionnel " régulation de la puissance réactive par le réseau.

Fonction.

Assurer le transfert d'énergie active maximum de la source solaire vers le réseau alternatif.

Entrée.

Le temps relatif du déphasage entre U et I ainsi que le sens de ce déphasage.

Sortie.

Une valeur du bourrage de temps pour le bloc " transformation du courant continu en alternatif ".

Relation avec d'autres blocs.

Ce bloc est synchronisé par le bloc " transformation du continu en alternatif ".

8- Bloc fonctionnel " couplage du réseau distribution ".

Fonction.

1 Mettre la tension solaire en concordance de phase avec la tension du réseau.

2 Ajuster la tension solaire à une valeur très supérieure à la tension du réseau.

3. Coupler.

Entrée.

Variable booléenne image de la tension alternative solaire et de la tension du réseau.

Sortie.

I. Initialisation de la table des aires théoriques relative à la plus grande valeur de k.

2. Initialisation de la valeur du bourrage de temps.

Relation avec d'autres blocs.

La fin de ce bloc déclenche les blocs synchronisés par la transformation du courant continu en alternatif.

3.3. REPARTITION ENTRE MATERIEL ET LOGICIEL.

Avant d'aborder l'étude du matériel et du logiciel, la question de la répartition entre les deux se pose.Qu'est-ce qui est réalisé par le matériel et qu'est-ce qui est pris en charge par le logiciel. Certaines fonctions peuvent effectivement être réalisées aussi bien par des circuits logiques que par programme. C'est le cas par exemple d'un calcul ou d'une temporisation.

L'utilisation d'un circuit logique est nécessaire pour des calculs à effectuer à des temps courts, impossibles au microprocesseur. Il y a également intérêt à réaliser les temporisations fréquentes ou relativement longues par le biais du matériel. Cependant, il ne faut pas perdre de vue que dans les grandes séries, on minimise le coût du matériel en imputant un maximum de fonctions au logiciel.

Nous allons examiner la question pour :

1. La puissance du bloc d'optimisation qui résulte de la multiplication de U par I .
2. Le timer relatif aux moments d'enclenchement des vannes du hacheur.
3. L'évaluation du temps de déphasage entre U et I.

I. Lecture de la puissance dans le bloc d'optimisation primaire.

Le débit d'acquisition est d'une fois par période. De plus, la puissance lue ne doit pas être d'une précision excessive. En effet il suffit de tester si la puissance croit ou décroit au cours des périodes successives.

Il existe deux possibilités pour l'obtention de la puissance. Premièrement le produit de U par I est fait en numérique. Cela suppose deux convertisseurs A/D et trois opérations : la conversion A/D avec la lecture de V, la conversion A/D avec la lecture de I et le produit des deux valeurs digitalisées. L'autre méthode consiste à la prise directe de la puissance avec un seul convertisseur A/D et une seule opération de conversion et lecture. C'est la solution la plus efficace compte tenu de son minimum de matériel et d'opération logicielle.

2. Le timer relatif au moment d'enclenchement des vannes du hacheur.

A un moment bien précis dans chacunes des n divisions de la demi-période, il faut enclencher une vanne du hacheur. Cela suppose bien sûr un système de repérage du temps.

La solution matérielle utilise un temporisateur programmable que l'on initialise à chaque division de période. Celui-ci renvoie une interruption au moment choisi de la division suivante. L'avantage de ce système est de ne pas monopoliser tout le temps le processeur. Cependant, il faut initialiser en deux fois parce-que le temporisateur travaille sur un registre de I6 bits et d'autre part il faut empêcher que l'interruption ne soit prise en considération avant la fin de l'exécution relative à la division antérieure si elle déborde.

Dans l'optique logicielle, on incrémente périodiquement un compteur de temps. Cela signifie que le nombre de cycles requis par l'exécution de l'ensemble des instructions relatives à une période multipliée par le temps de cycle doit donner la durée de la période. Le microprocesseur est par conséquent toujours occupé et il faut satisfaire la contrainte qui pèse sur le logiciel.

La solution logicielle a été retenue parce qu'elle ne nécessite aucun matériel et qu'il est plus facile de s'assurer à priori du fonctionnement correct du système par la bonne connaissance que l'on a de ce que fait le programme à chaque instant.

3. L'évaluation du temps de déphasage.

La mesure du déphasage entre U et I est chiffrée par l'écoulement du temps entre deux impulsions positives consécutives des amplis à seuil U et I.

De nouveau c'est un choix entre la temporisation programmable et le compteur logiciel qui se pose. Ici, la solution matérielle décharge le processeur de compter le temps et permet de l'allouer à autre chose.

3.4. ETUDE DU MATERIEL.

L'étude du matériel consiste à choisir et à définir les composants et les circuits annexes associés au microprocesseur. Il faut en premier établir l'architecture du système puis dénombrer les interfaces et déterminer leur nature. Ensuite, il restera à définir leur mode d'adressage puis à établir le schéma d'interfaçage.

3.4.I. Architecture.

Nous venons de définir les fonctions que le système doit prendre en charge. Nous avons défini les relations entre ces fonctions ainsi que les capteurs et les actionneurs associés à ces fonctions.Trois achitectures correspondent aux trois types de centrale solaire. Le nombre de microprocesseurs en jeu dans chacune de ces trois architectures découle du nombre de fonctions incompatibles entre elles et devant être exécutées en même temps. Les relations entre les différents microprocesseurs, s'il y en a plusieurs, sont déduites des relations entre les différents blocs fonctionnels répartis entre ces mircoprocesseurs.

L'architecture de la centrale solaire continue, n'ayant qu'un seul bloc fonctionnel, est construite autour d'un seul microprocesseur.

L'architecture de la centrale solaire alternative autonome nécessite deux microprocesseurs. Le programme de transformation du continu en alternatif demande le contrôle d'un microprocesseur à chaque instant. Le deuxième microprocesseur est affecté aux autres blocs. Ces fonctions ne sont pas incompatibles : le programme de mesure du déphasage, essentiellement asynchrone, doit répondre directement aux interruptions et les autres programmes peuvent être différés du temps d'exécution du premier. L'architecture de la centrale solaire couplée au réseau reste du même type.

Les relations entre ces deux microprocesseurs sont très restreintes. Outre les appels de synchronisation, il y a un passage de paramètres :

- le numéro de la table des aires théoriques relatives à la valeur de $k = V_p / V\ max$.
- la valeur de l'indice de bourrage de temps dans l'architecture de la centrale couplée au réseau.

L'architecture est construite autour de deux microprocesseurs avec un unibus et une mémoire commune.

1. Architecture de la centrale solaire à courant continu.

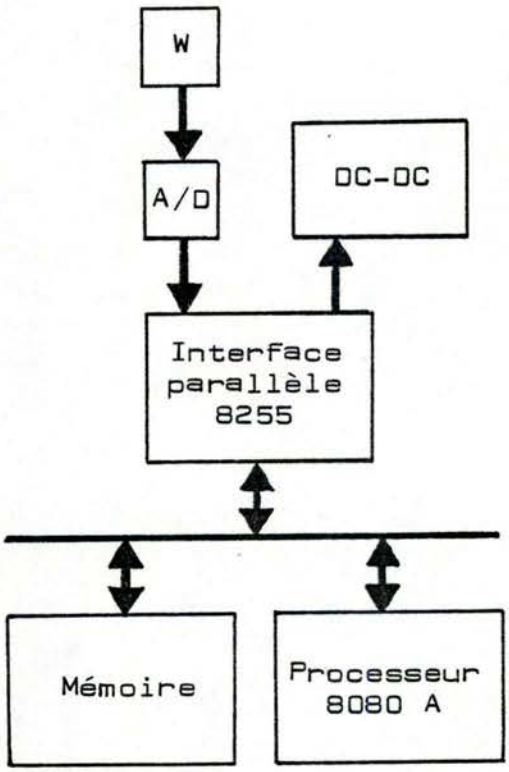


Fig 36

2. Architecture de la centrale solaire autonome.

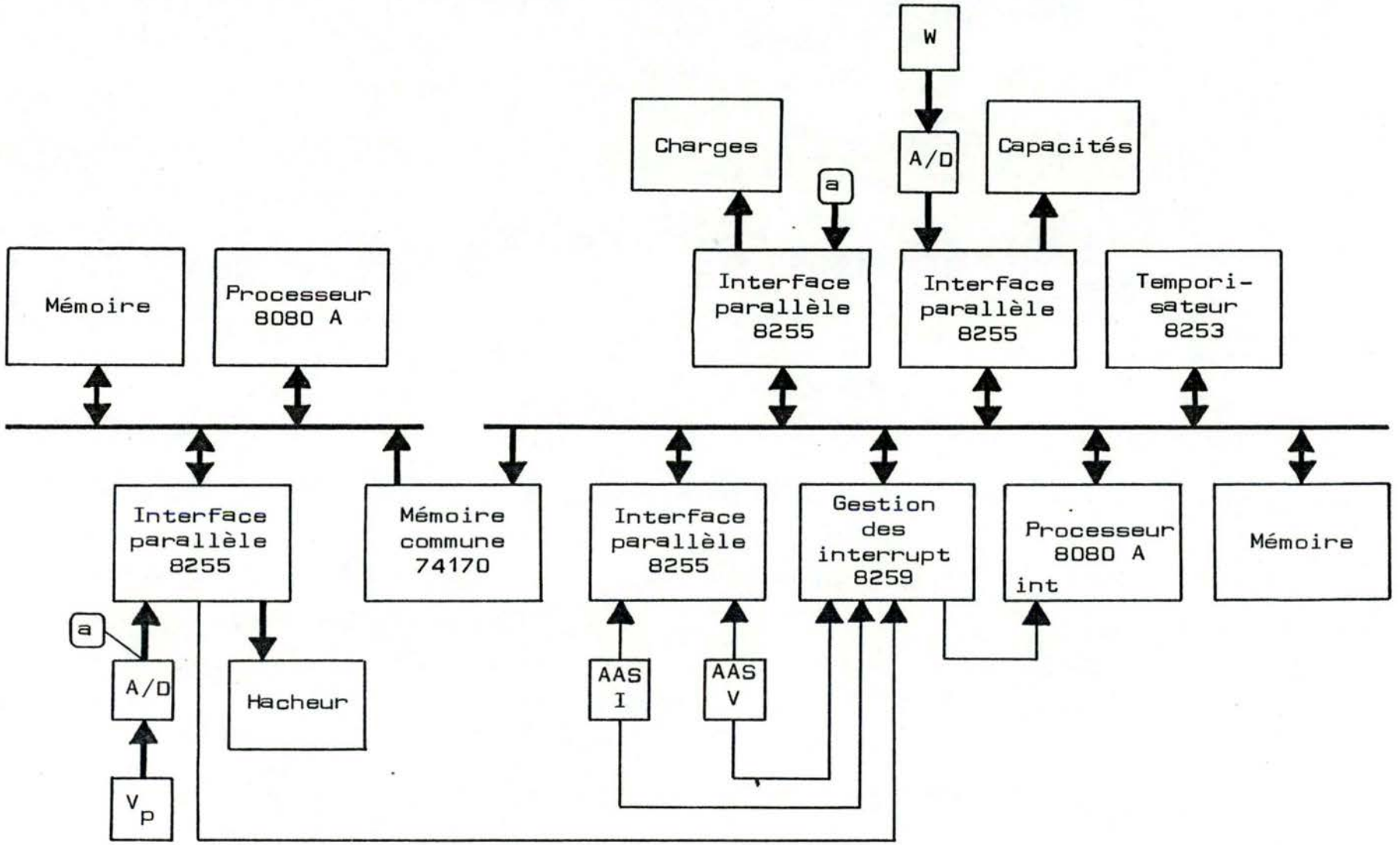


Fig 37

3. Architecture de la centrale solaire d'appoint à un réseau alternatif.

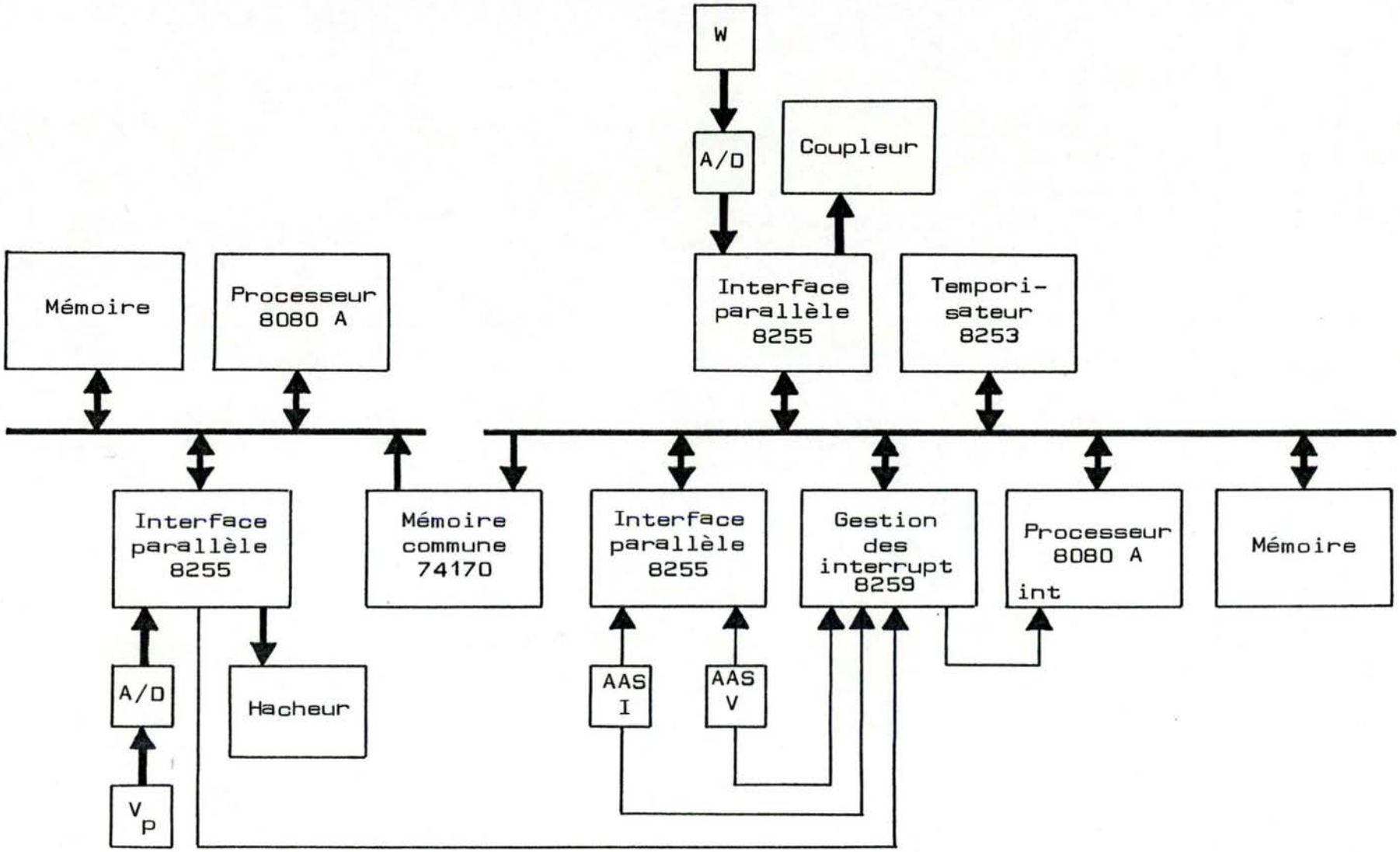


Fig 38

3.4.2. Adressage et interfacage du matériel.

I. Adressage : généralités.

Pour adresser une mémoire ou un périphérique, il faut sélectionner la position mémoire ou le registre du périphérique. Il faut par ailleurs choisir le boîtier mémoire ou périphérique quand il y en a plusieurs.

Sélection de la position mémoire ou registre.

Pour adresser 2^n positions, il faut disposer de n bits d'adresse. La règle universelle est de prendre les n bits de plus faible poids du bus adresse.

Exemples.

1. soit à adresser 1 k octet au maximum : on prendra les bits de A_o à A_9 du bus adresse.
2. soit à adresser 4 registres au plus : on prendra les bits Ao et A_1 du bus adresse.

Sélection du boîtier.

La méthode de sélection du boîtier résulte du choix d'un mode d'adressage (soit par décodage ou bien par sélection linéaire) et du choix du type de structure E/S (soit par instruction mémoire ou par instruction E/S)

Il y a 4 combinaisons possibles :

1. sélection linéaire avec instruction mémoire.
2. sélection linéaire avec instruction E/S.
3. décodage avec instruction mémoire.
4. décodage avec instruction E/S.

Dans l'adressage par décodage, un décodeur sélectionne un boîtier parmi 2^n sur base de la codification de n bits. C'est le mode le plus performant en contrepartie du prix du décodeur.

L'adressage par sélection linéaire supprime le décodeur en affectant un bit exclusif du bus d'adresse aux boîtiers. Il ne permet cependant qu'un nombre très restreint de boîtiers.

Dans la structure E/S par instruction mémoire, les signaux lecture mémoire et écriture mémoire servent aussi bien à la mémoire qu'aux périphériques. Il en résulte qu'une adresse ne doit valider qu'un et un seul organe à la fois.

Pour une structure E/S par instruction E/S, il faut disposer d'instructions spéciales aux E/S. Dans ce cas, une adresse ne doit valider qu'un et un seul boîtier périphérique et mémoire. C'est la commande lecture/écriture spécifique soit à la mémoire ou au périphérique qui fait la discrimination.

En général, la sélection d'un boîtier interface chez INTEL se fait par signal complémenté. Ceci a pour conséquence que le niveau actif est le niveau zéro.

Le microprocesseur 8080 A d'INTEL dispose des instructions spéciales aux E/S (IN et OUT) et, le nombre de boîtier interface à adresser ne dépasse pas six. Nous utiliserons l'adressage par sélection linéaire avec instruction E/S.

2. Interfaçage de la centrale solaire à courant continu.

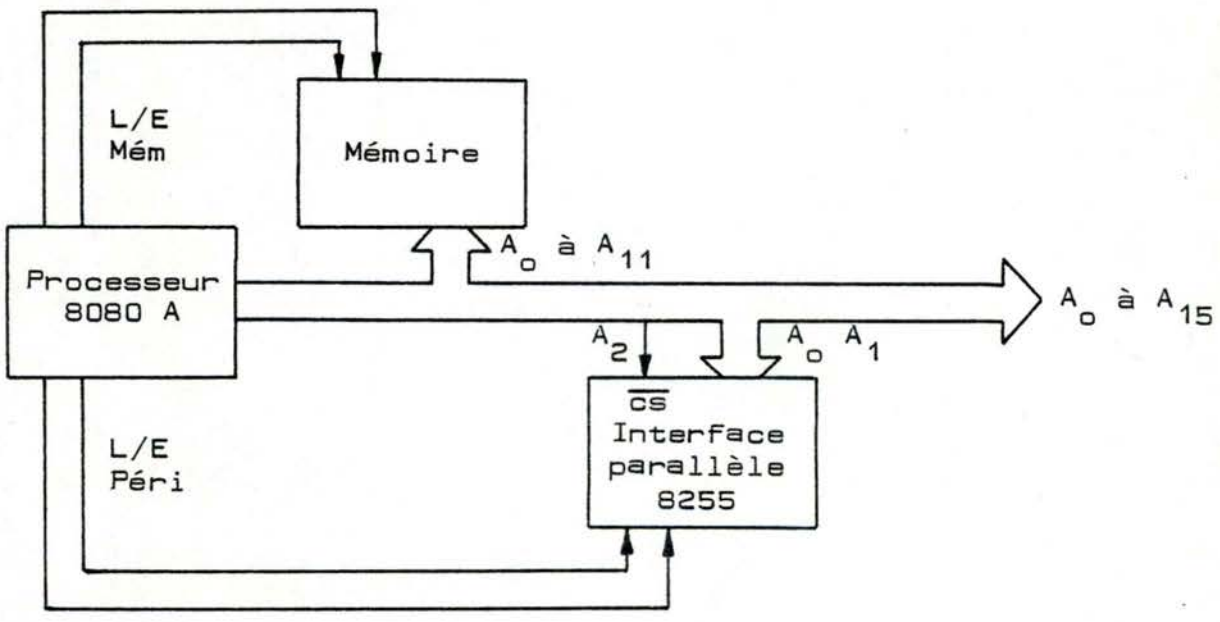


Fig 39

1] La mémoire.

Le boîtier mémoire est validé par les signaux de lecture et d'écriture mémoire. Les bits de A_o à A_{11} du bus d'adresse nous permettent d'adresser 4K au maximum.

2] L'interface parallèle 8255.

- L'interface parallèle 8255 est validé par le bit A_2 en ce qui concerne le boîtier et par les signaux de lecture et d'écriture périphérique pour les échanges.

- Le registre A de l'interface est utilisé pour la lecture de la puissance. Le registre B sert pour la commande de la vanne du convertisseur DC-DC. Nous pouvons en déduire le mot de mode de fonctionnement:

1	mode de A		A E/S	C4-7 E/S	mode de B	B E/S	C0-3 E/S
1	0	0	1	0	0	0	0

L'initialisation de l'interface se fait par l'envoi du mot de mode de fonctionnement dans le registre de commande:

```
MVI  A,90
OUT  11111011 b
```

- La commande de la lecture de la puissance se fait par l'instruction d'entrée/sortie:

```
IN   11111000 b
```

- Pour la commande de la vanne du convertisseur DC-DC, seul le bit de poids faible du registre B est significatif: 0 commande l'ouverture de la vanne et 1 commande sa fermeture.

```
ouverture:   MVI  00000000 b
             OUT  11111001 b

fermeture:   MVI  00000001 b
             OUT  11111001 b
```

3. Interfaçage de la centrale solaire autonome et de la centrale solaire d'appoint à un réseau alternatif.

A un boîtier interface parallèle près, la centrale solaire autonome a la même architecture que la centrale solaire d'appoint à un réseau alternatif. Cette architecture est construite autour de deux microprocesseurs. Le premier assure la transformation du continu en alternatif et, le second assure les autres fonctions.

Interfaçage du système de transformation du continu en alternatif.

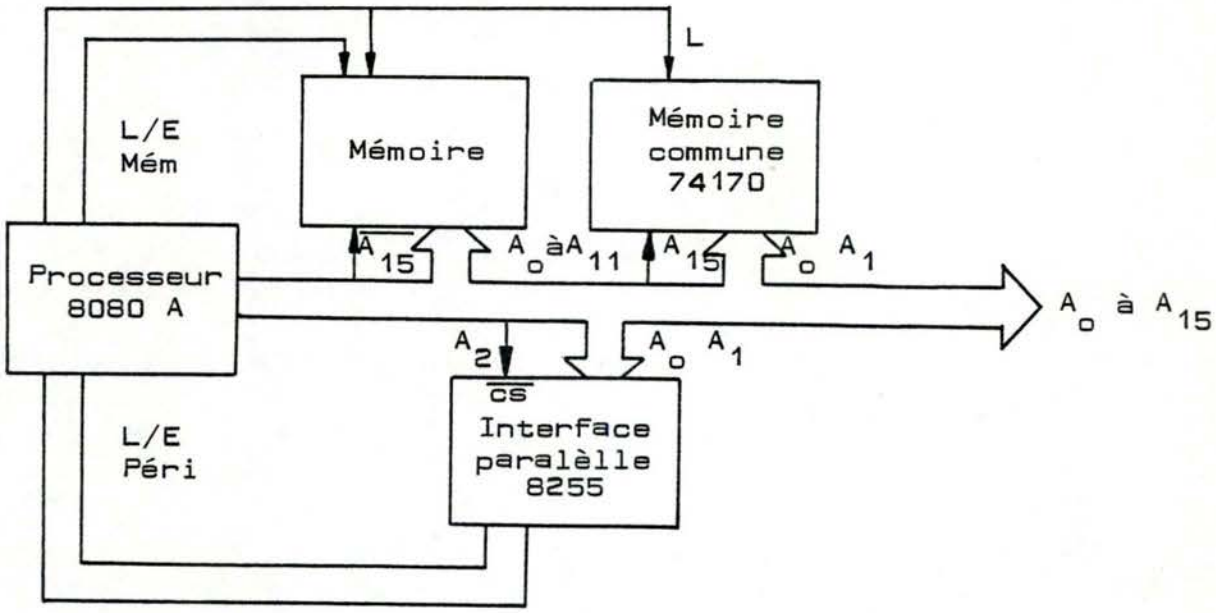


Fig 40

1) La mémoire.

Les échanges avec la mémoire sont validés par les signaux de lecture et d'écriture mémoire. Le bit A_{15} du bus d'adresse est utilisé pour discriminer le bloc mémoire de la mémoire commune.

2) La mémoire commune 74170.

Le bloc 74170 est une mémoire de quatre mots de quatre bits. Il faut en utiliser deux en parallèle pour constituer une mémoire de quatre mots de huit bits. Cette mémoire est validée par le signal de lecture mémoire et le bit A_{15} du bus d'adresse.

3) L'interface parallèle 8255.

L'interface 8255 est validé par le bit A_2 du bus d'adresse et les signaux de lecture et d'écriture périphérique.

Le registre B est utilisé en sortie pour la commande des deux vannes du hacheur. C'est par l'intermédiaire du registre C que le programme de transformation du continu en alternatif émet son interruption de début de période. Le registre A sert pour lire la tension à soustraire de la quantité théorique de volt-seconde à délivrer. Le mot de mode de fonctionnement est le suivant:

1	mode de A	A E/S	C4-7 E/S	mode de B	B E/S	C0-3 E/S
1	0 0	1	0	0	0	0

Initialisation de l'interface:

```
MVI  A,90
OUT  11111011 b
```

- Les deux bits de poids faibles du registre B de l'interface sont chacuns associés à une vanne de puissance du hacheur.

00 correspond à l'ouverture des deux vannes.

01 correspond à la fermeture de la vanne 1 et à l'ouverture de la deuxième vanne.

10 correspond à l'inverse de 01.

11 est une configuration interdite. Aucune protection n'est prévue. Il faut donc veiller à ce que le logiciel n'envoie pas un mot qui se termine par deux 1 dans le registre B de l'interface.

ouverture des vannes:
```
MVI  A,00
OUT  11111001 b
```

fermeture vanne 1 :
```
MVI  A,01
OUT  11111001 b
```

fermeture vanne 2 :
```
MVI  A,02
OUT  11111001 b
```

- L'envoi d'une interruption se fait par la mise à 1 de C_1

Mise au niveau 1 :
```
MVI  A,01
OUT  11111010 b
```

Mise au niveau 0 :
```
MVI  A,00
OUT  11111010 b
```

- Lecture de la tension V_p :
```
IN   11111000 b
```

4) Remarque.

Il est cependant plus commode d'accéder à la tension V_p à soustraire de la quantité théorique de Volts-seconde par une instruction mémoire (SUB). Pour cela, le convertisseur A/D doit être validé par la lecture mémoire et son adressage doit être spécifique.

On a :

A_{15}	A_{14}	
0	0	Mémoire.
0	1	Convertisseur A/D.
1		Mémoire commune.

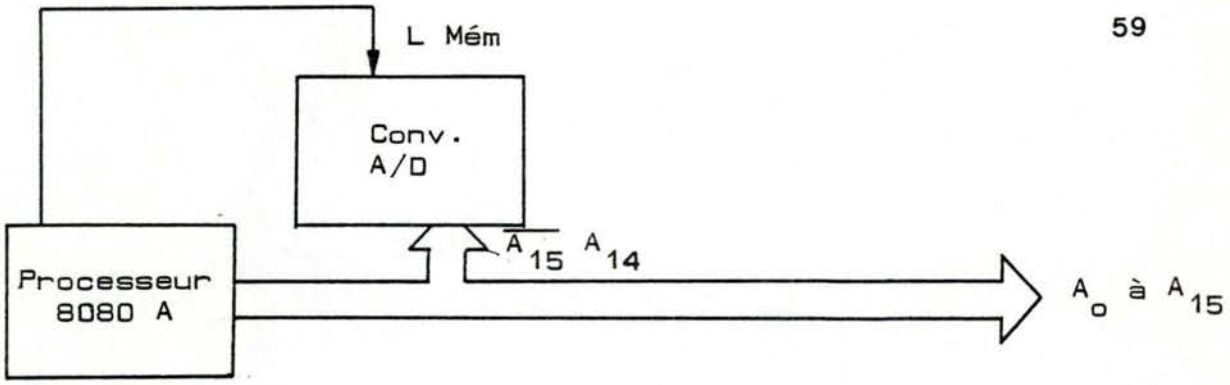


Fig 41

Interfaçage du système qui assure les autres fonctions de régulation.

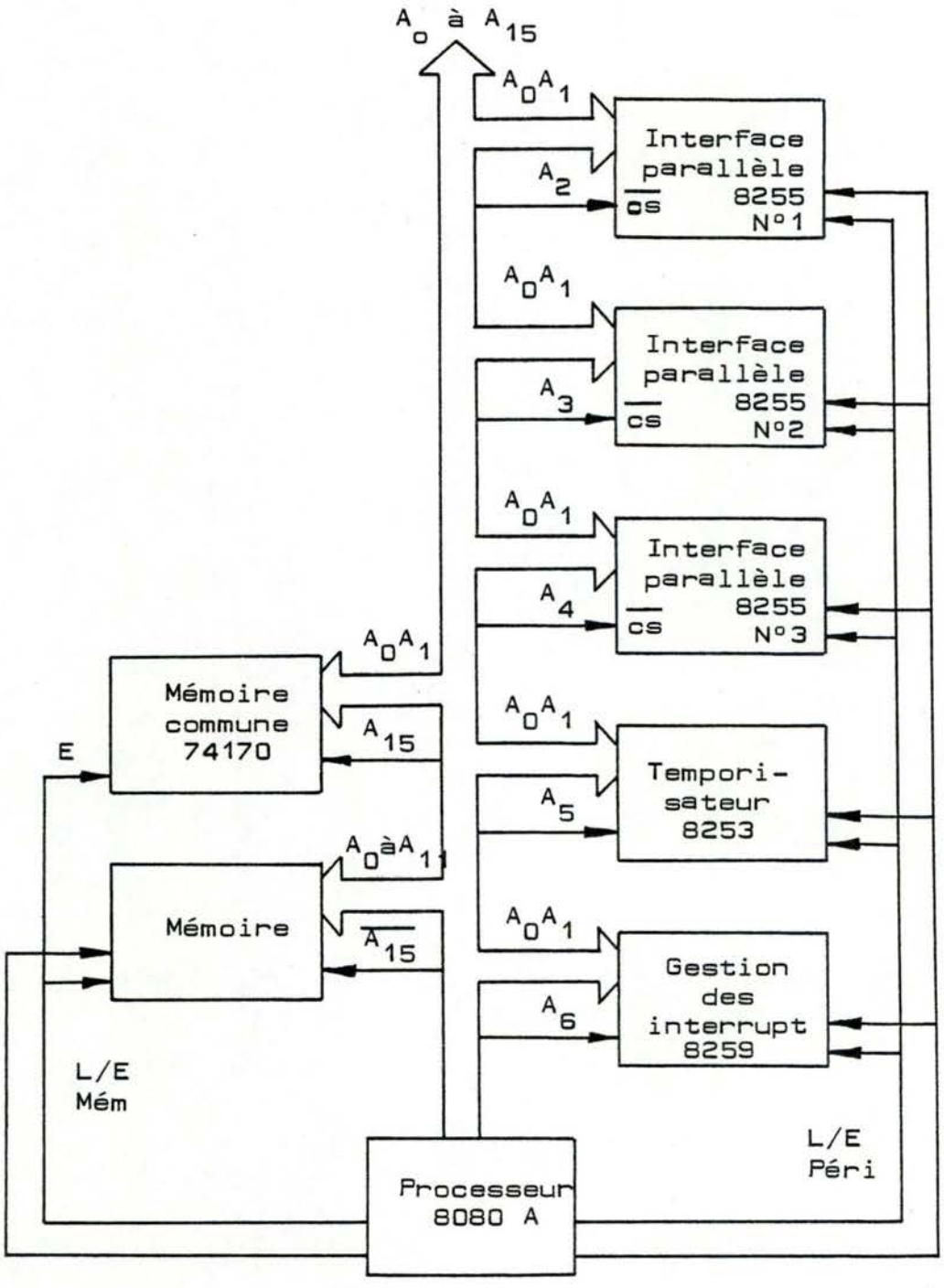


Fig 42

1) La mémoire.

Les échanges avec la mémoire sont validés par les signaux de lecture et d'écriture mémoire. Le bit A_{15} du bus d'adresse est utilisé pour discriminer le bloc mémoire de la mémoire commune.

2) La mémoire commune 74170.

Il s'agit de la porte d'entrée de la mémoire commune. Cette mémoire est validée par le signal d'écriture mémoire et le bit A_{15} du bus d'adresse.

3) L'interface parallèle 8255 N°1.

- Le bit A_2 du bus d'adresse valide le boîtier interface et les signaux de lecture et d'écriture périphérique valident les échanges.

- Le registre A est utilisé pour lire la variable logique image de la tension sinusoïdale et le registre B pour lire la variable logique image du courant sinusoïdal. Le mot de mode de fonctionnement est 92H. (10010010). Initialisation de l'interface:

```
MVI  A,92
OUT  11111011 b
```

- La commande d'une lecture du signe d'une impulsion relative à un passage par zéro de la sinusoïde courant:

```
IN   11111000 b
```

- La commande d'une lecture du signe d'une impulsion relative à un passage par zéro de la sinusoïde tension:

```
IN   11111001 b
```

4) L'interface parallèle 8255 N°2.

- Uniquement dans l'architecture de la centrale solaire autonome.

- L'interface parallèle est validé par le bit A_3 du bus d'adresse et les signaux de lecture et d'écriture périphérique.

- Le registre A sert pour lire la tension V_p. Le registre B

pour la commande des charges. Le mot de mode de fonctionnement est 92H et l'initialisation se fait par:

```
MVI  A,90
OUT  11110111 b
```

- La commande de la lecture de la tension:

```
IN   11110100 b
```

- Pour la commande des charges, seuls les trois bits de poids faible du registre B sont significatifs:
Celui de droite commande la batterie (0: déconnectée, 1: connectée).
Celui du centre commande l'onduleur (0: déconnecté, 1: connecté).
Celui de gauche commande les charges non-prioritaires. (0: déconnectées, 1: connectées). L'envoi d'une commande aux charges:

```
OUT  11110101 b
```

5) L'interface parallèle 8255 N°3.

- Le bit A_4 du bus d'adresse valide le boîtier interface et les signaux de lecture et d'écriture périphérique valident les échanges.

- Le registre A de l'interface est utilisé pour la lecture de la puissance. Le registre B sert pour la commande de la batterie de condensateurs dans l'architecture de la centrale solaire autonome. Il sert pour la commande du coupleur dans l'architecture de la centrale solaire d'appoint à un réseau alternatif. Dans les deux cas, l'initialisation se fait par l'envoi de 90H dans le registre de commande de l'interface:

```
MVI  A,90
OUT  11101111 b
```

- La commande de la lecture de la puissance se fait par l'instruction d'entrée/sortie:

```
IN   11101100 b
```

- La commande des condensateurs ou du coupleur au réseau alternatif, selon le cas se fait par:

```
OUT  11101101 b
```

6) La gestion des interruptions 8259.

- Il suffit de savoir déterminer et utiliser quatre registres

[ICW1, ICW2, OCW1 et OCW2] pour gérer jusqu'à huit entrées de demande d'interruption. Ces registres sont répartis en deux groupes:

Premièrement, les registres d'initialisation.

ICW1 et ICW2 servent à fournir la première adresse d'une zone de quatre ou de huit mots relative à l'interruption de niveau zéro. Le registre ICW3 n'est utile que dans le cas où il y a plusieurs contrôleurs d'interruptions.

Deuxièmement, les registres de commande.

OCW1 masque les entrées d'interruption.

OCW2 signale au 8259 la fin d'exécution du sous-programme d'interruption.

OCW3 ne sert que s'il faut modifier le mode de priorité qui normalement est fixe et hiérarchisée.

Un seul interface de gestion des interruptions 8259 avec mode de priorité fixe et hiérarchisée suffit.

- Le bit de sélection linéaire affecté à l'interface de gestion des interruptions 8259 est le bit A_5.

- Pour initialiser l'interface des interruptions 8259, on donne la première adresse de la zone de huit mots aux registres ICW1 et ICW2:

```
MVI  A,(valeur hexa pour ICW1)
OUT  11011111 b
MVI  A,(valeur hexa pour ICW2)
OUT  11011111 b
```

```
ICW2     ICW1
-------- ---1DFS0
           └─► utilisé seulement si F = 0
```

S = 0 : dans le cas de plusieurs 8259.
    1 : dans le cas d'un seul 8259.
F = 0 : si le sous-programme d'appel est de quatre bytes.
    1 : si le sous-programme d'appel est de huit bytes.

- La forme générale des sous-programmes d'interruptions dépend du fait qu'ils permettent ou non les interruptions et qu'ils sauvent ou non l'environnement du programme interrompu.

Une prise en considération d'une interruption inhibe toute autre interruption. Si le sous-programme ne permet pas les interruptions, on a :

```
 -
[sous-programme]
 -
DI
MVI  A,20
OUT  rég 3 du 8259
EI
RET
```

Si le sous-programme permet les interruptions, on a :

```
EI
 -
[sous-programme]
 -
DI
MVI  A,20
OUT  rég 3 du 8259
EI
RET
```

Si l'on désire sauver l'environnement du programme interrompu, il faut le faire par l'instruction PUSH au début du sous-programme et restaurer dans l'ordre inverse par l'instruction POP à la fin du sous-programme.

- L'envoi d'un 0 ou d'un 1 dans chacun des bits du registre de masque des demandes d'interruption valide ou non chacune des huit demandes correspondantes.

```
MVI  A,xxxxxxxx b
OUT  masque du 8259
```

7] Le temporisateur programmable 8253.

Outre sa fonction de temporisateur programmable, le 8253 peut-être programmé pour assurer d'autres fonctions que celle de temporisation. Citons par exemple: un monostable, un astable, un compteur d'évènement, un générateur d'impulsion. Pour utiliser ce circuit en temporisateur programmable, il suffit de lui envoyer le mot de commande adéquat et de charger la valeur N dans le décompteur pour une durée de temporisation de N T , T étant la période de l'horloge. Au terme de la temporisation, le 8253 génère une interruption.

Cette fonction du temporisateur peut facilement être utilisée pour réaliser un compteur. Pour ce, il suffit d'initialiser le registre 16 bits de décomptage à sa valeur maximale [FFFF en hexadécimal]

Chaque impulsion d'horloge arrivant au temporisateur décrémente d'une unité le contenu du décompteur. Lorsque l'on désire arrêter le comptage, il suffit de demander une lecture du décompteur. Le complément à 1 donne le nombre d'impulsions comptées. Toutefois, si l'on veut éviter d'arriver à une interruption de fin de décomptage, il faut interposer un diviseur adéquatement calculé entre l'horloge et son entrée dans le temporisateur.

Pour mesurer le déphasage, le temporisateur décrémente son compteur pendant un quart de période au maximum, soit 5000 cycles. L'interruption de fin de décomptage se produit au terme de 2^{16} cycles (65536 cycles). Il n'y a en principe pas besoin d'interposer un diviseur de cycles. Cependant, le décomptage de 5000 cycles affecte 13 bits sur les 16 du décompteur et le microprocesseur ne travaille que sur huit bits. Si l'on désire que les 5000 cycles n'affectent pas plus de 8 bits, il faut un diviseur pour ralentir les impulsions. Avec un diviseur par 32, 5000 cycles affectent 8 bits.

- Le bit de sélection linéaire affecté au temporisateur programmable est le bit A_6.

- La remise à zéro du compteur et un démarrage s'effectue par les instructions:

```
MVI   A,00110000 b
OUT   10111111 b
MVI   A,FF
OUT   10111100 b
OUT   10111100 b
```

- La lecture du compteur:

```
MVI   A,00000000 b
OUT   10111111 b
MVI   A,00010000 b
OUT   10111111 b
IN    10111100 b
IN    10111100 b
CMA
```

3.5. CONCEPTION DU LOGICIEL.

3.5.1. Optimisation primaire.

1] Les variables.

Instant élémentaire : c'est le temps qui s'écoule entre deux passages consécutifs sur l'instruction d'incrémentation du compteur C.

Période : une période est un ensemble de 256 instants élémentaires.

C : compteur d'instants élémentaires modulo 256.

D : indice tel que pour toutes valeurs du compteur C inférieure, la vanne de puissance du DC-DC est enclenchée et, pour toutes valeurs supérieures, elle est déclenchée.

Pref : puissance de référence qui prend la valeur de la dernière puissance lue en cas de modification de l'indice D du moment de déclenchement.

Deltap : différence entre la puissance lue au temps t et la puissance de référence.

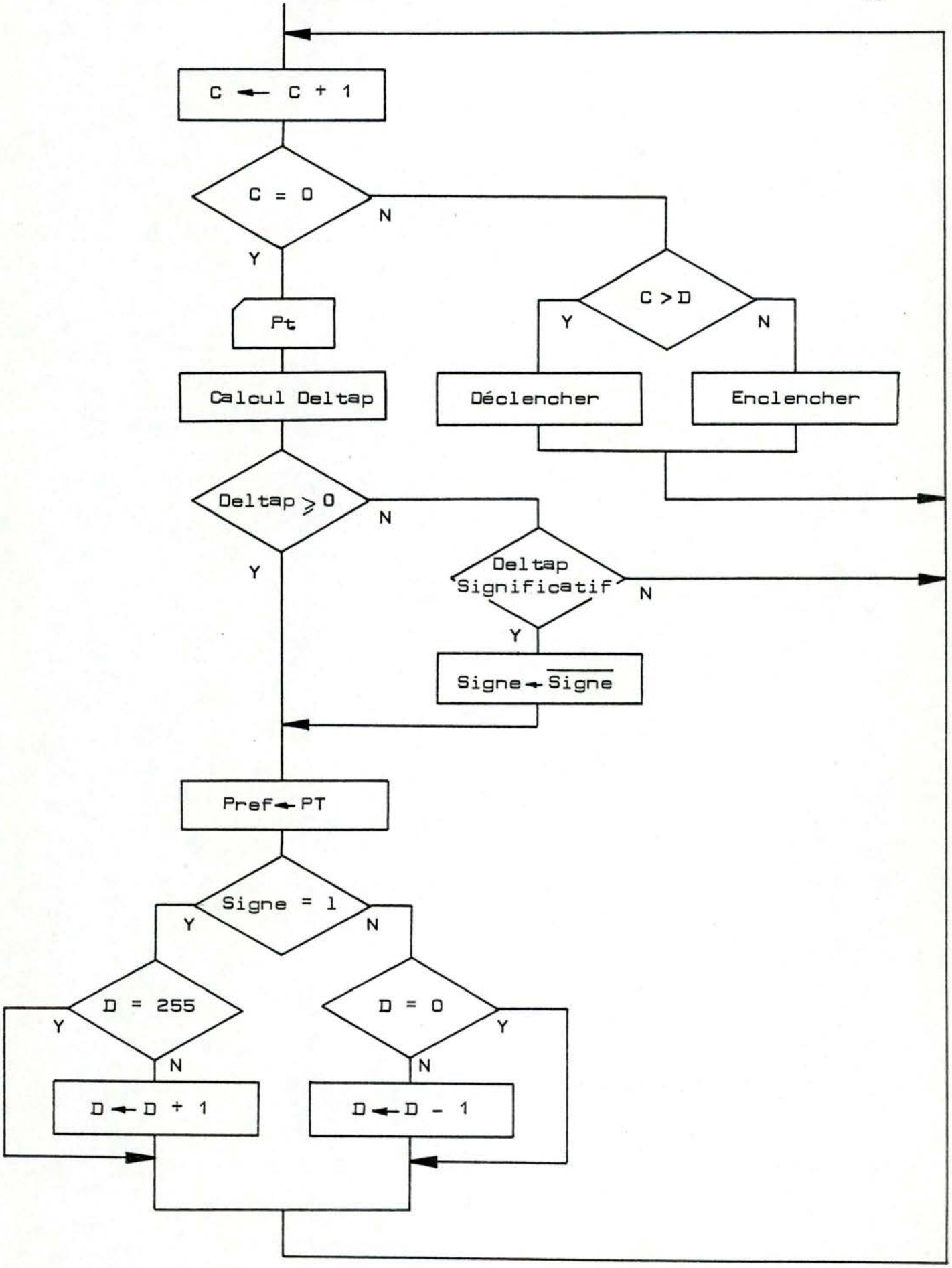
Pt : puissance lue à chaque période.

Signe : variable booléenne qui indique le sens de l'action antérieure sur l'indice D de déclenchement. 0 pour une décrémentation de D et 1 pour une incrémentation.

2] Initialisation de l'optimisation primaire.

- initialisation de l'interface parallèle 8255.
- enclenchement de la vanne du convertisseur DC-DC.
- initialisation des variables :
  - Pref à 0 [la plus mauvaise valeur possible]
  - D à 0
  - Signe à 1 [il faut augmenter l'angle d'ouverture]
  - C à 0

3] Boucle de régulation.



3.5.2. Transformation du courant continu en courant alternatif.

Un programme principal boucle sur lui-même en l'espace d'une période de la sinusoïde à produire. Il appelle N fois au cours de la première demi-période un sous-programme relatif à l'enclenchement et au déclenchement de la première vanne du hacheur. Un deuxième sous-programme relatif à la seconde vanne du hacheur est appellé de façon symétrique pendant la deuxième demi-période. Le programme principal initialise les registres de travail du sous-programme avant chaque appel.

1) Les variables.

C : compteur variant de 0 à 255 au cours d'une demi-période de la sinusoïde à produire.

S01 ... S11 : tableau des quantités théoriques de volt-seconde à délivrer au cours de chacunes des 11 divisions de période des deux demi-alternances de sinusoïde.

T01 ... T11 : tableau des valeurs à partir desquelles il faut enclencher les vannes du hacheur.

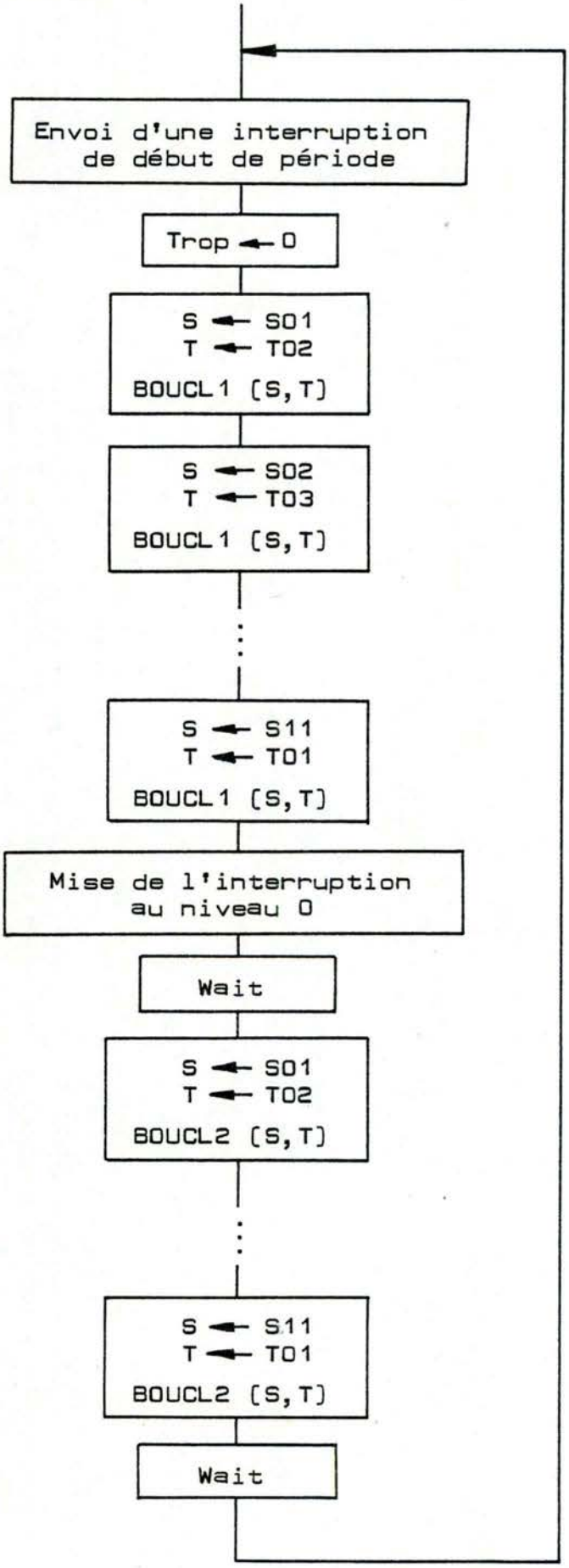
Trop : c'est la quantité de volt-seconde délivrée en plus que la quantité théorique nécessaire à chaque division de période.

Vp : c'est la tension lue au panneau solaire qui est à soustraire à chaque instant élémentaire de la quantité de volt-seconde à délivrer pendant une division de période.

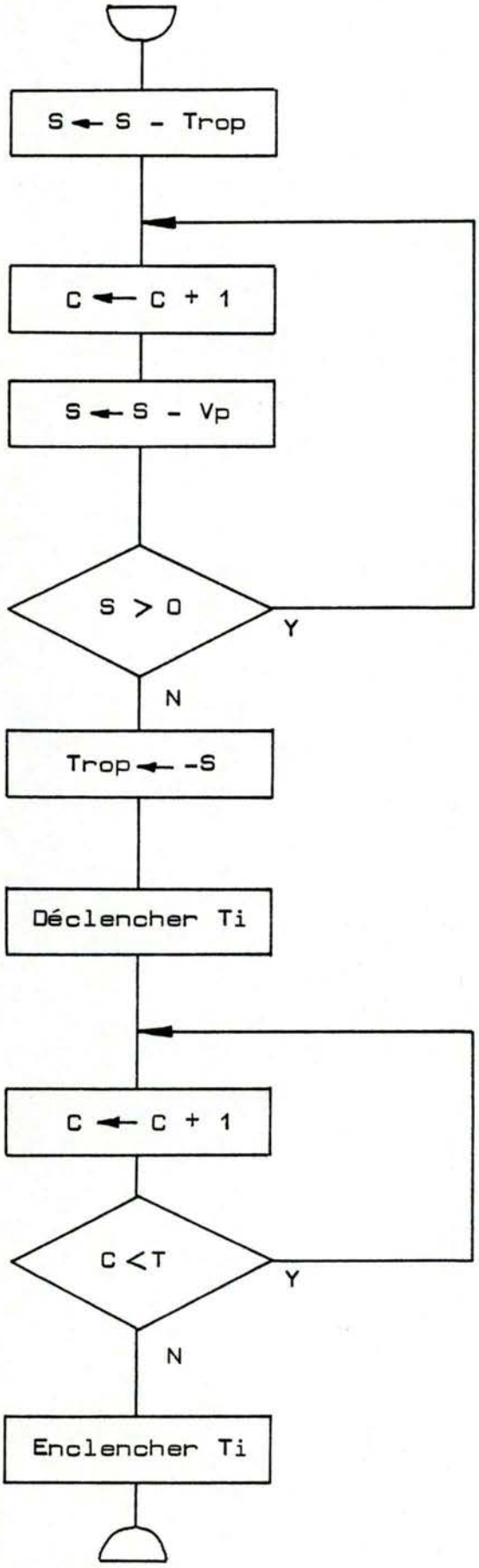
2) Initialisation du programme principal.

- initialisation de l'interface parallèle 8255.
- initialisation des variables
  - C à 0
  - Trop à 0

3] Programme principal pour une seule valeur de $k = \frac{V_p}{V_{max}}$



4] Sous-programme BOUCLi [S,T]



3.5.3. Optimisation générale.

1) Les variables.

Pref : puissance de référence qui prend la valeur de la dernière puissance lue en cas de modification du numéro de la table des quantités théoriques de volt-seconde et des moments d'enclenchement des vannes.

Deltap : différence entre la puissance lue au temps t et la puissance de référence

Pt : puissance lue à chaque période.

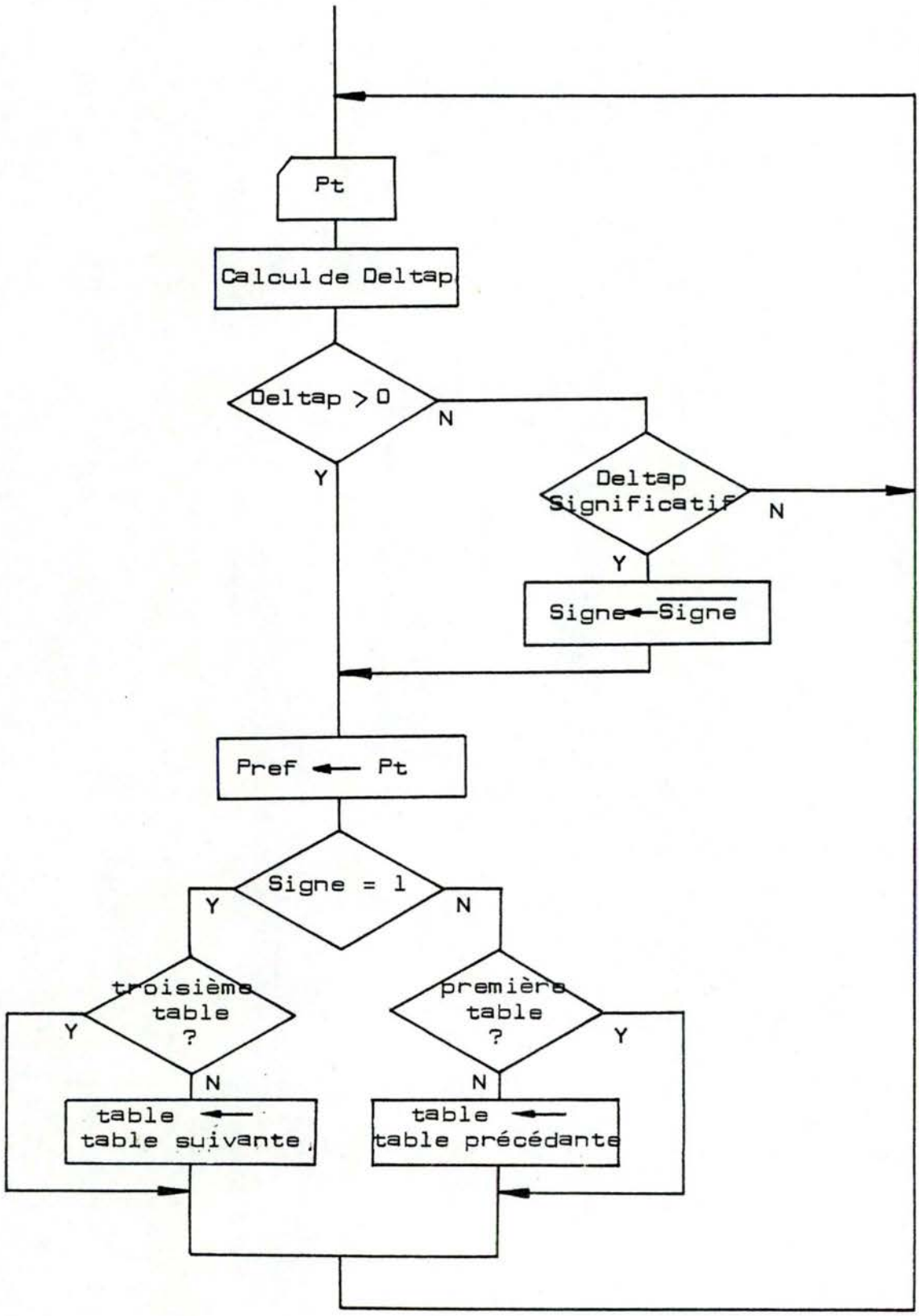
Signe : variable booléenne qui indique le sens de l'action antérieure sur la variable de l'adresse 00 de la mémoire commune 74170. 0 pour une décrémentation et 1 pour une incrémentation.

Adresse 00 de la mémoire commune 74170 : contient le numéro de la table des quantités théoriques de volt-seconde à délivrer et des moments d'enclenchement que doit utiliser le programme de transformation du continu en alternatif lorsqu'il y a plusieurs valeurs de $k = V_p/V_{max}$.

2) Initialisation de l'optimisation générale.

- initialisation de l'interface parallèle 8255.
- initialisation des variables :
  - Pref à 0
  - Signe à 0
  - Troisième table des quantités de Volts-seconde et des moments d'enclenchement.

3] Boucle de régulation.



3.5.4. Accumulation temporaire et gestion de la priorité des charges.

1] Les variables.

Vp : c'est la tension lue à chaque passage dans la boucle .

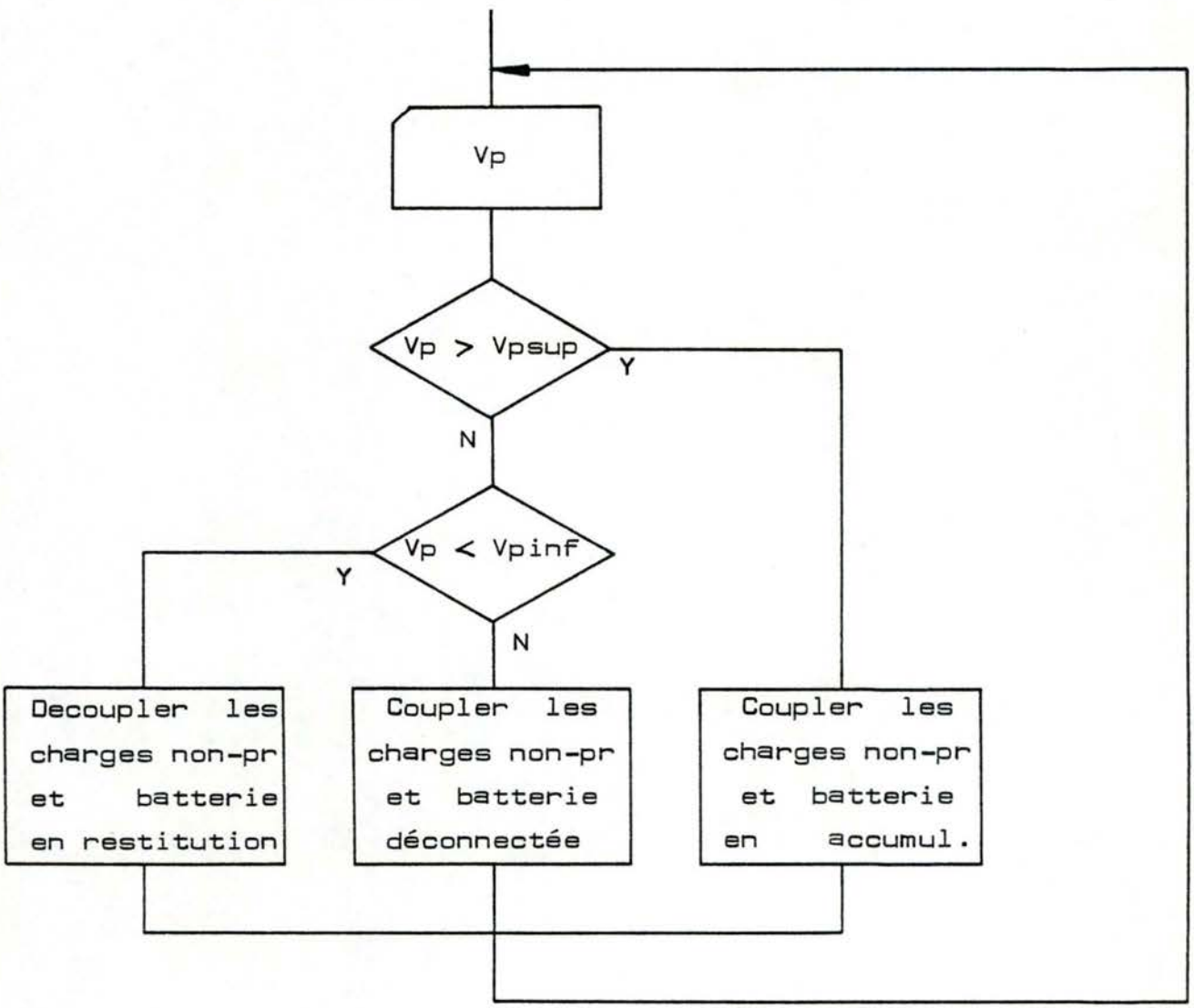
Vpsup : seuil supérieur pour la variation de Vp au-delà duquel une action est exécutée.

Vpinf : seuil inférieur pour la variation de Vp en-deçà duquel une action est exécutée.

2] Initialisation.

- initialisation de l'interface parallèle 8255.

3] Boucle de régulation.



3.5.5. Mesure du déphasage entre U et I.

1] Les variables.

U : variable booléenne qui permet de dire dans la routine d'interruption de quel ampli à seuil provient l'interruption. Cette variable est initialisée lors de l'appel de la routine à 0 si l'interruption provient de l'ampli a seuil I, à 1 si elle provient de l'ampli à seuil U.

Va : variable booléenne image de la sinusoïde tension. Cette variable est à 1 pendant l'alternance positive, à 0 pendant l'alternance négative.

Ia : variable booléenne image de la sinusoïde courant.

Etat : variable qui indique dans quel état se trouve l'automate.

Fi : variable qui contient la grandeur du déphasage entre U et I.

Av : variable booléenne qui indique le sens du déphasage. Cette variable est à 0 si I est en arrière sur U, à 1 si I est en avance sur U.

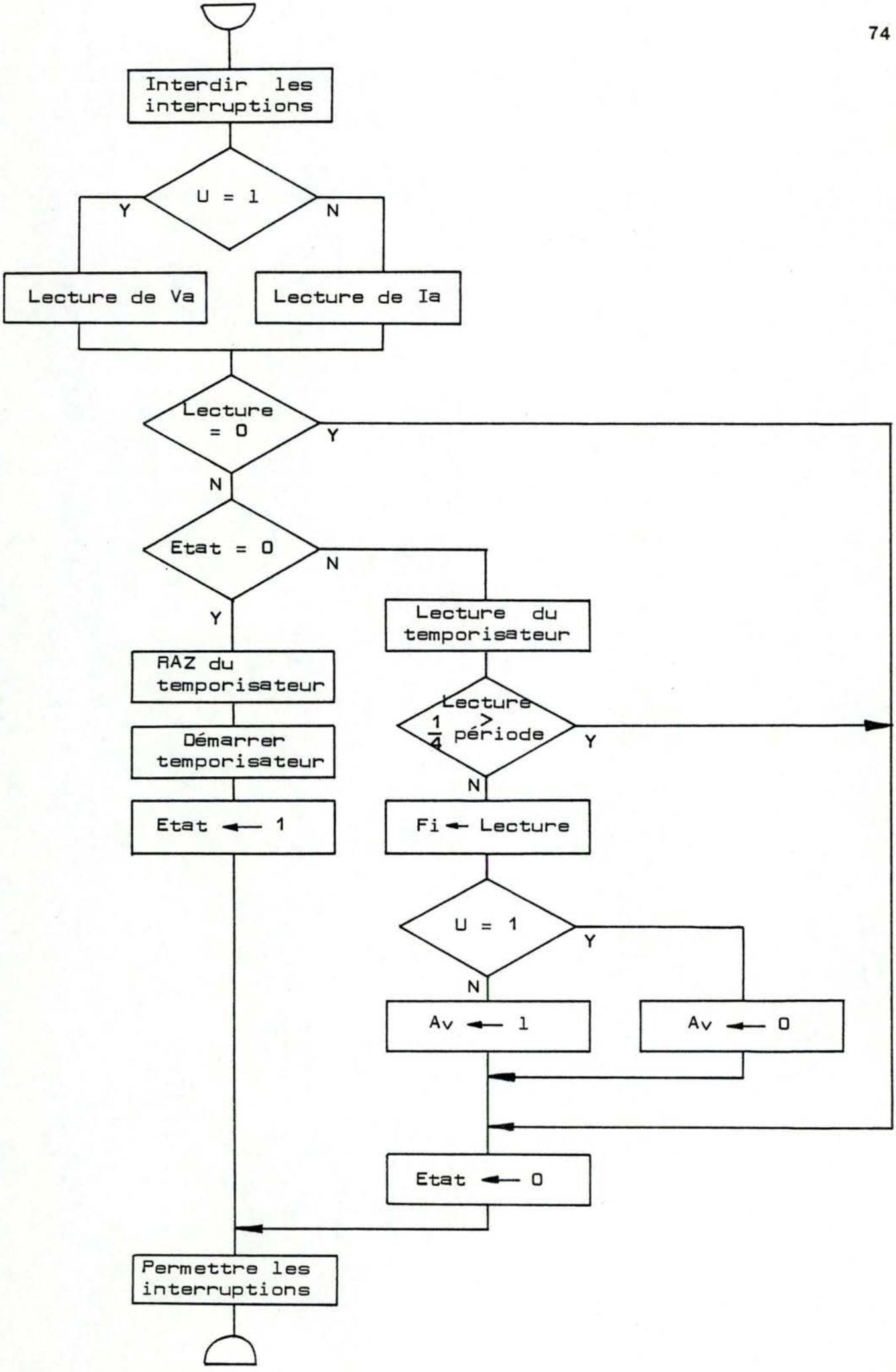
Quart : temps de cycle du quart de la période de sinusoïde.

2] Initialisation de la mesure du déphasage entre U et I.

- initialisation de l'interface parallèle 8255.
- initialisation de l'interface de gestion des interruption 8259.
- initialisation du temporisateur 8253.

Lors de l'appel de la routine :

- si l'interruption provient de l'ampli a seuil i alors U := 0.
- si l'interruption provient de l'ampli a seuil u alors U := 1.


Interdir les interruptions
U = 1
Y
N
Lecture de Va
Lecture de Ia
Lecture = 0
Y
N
Etat = 0
N
Y
Lecture du temporisateur
RAZ du temporisateur
Lecture 1/4 > période
Y
N
Démarrer temporisateur
Etat ← 1
Fi ← Lecture
U = 1
Y
N
Av ← 1
Av ← 0
Etat ← 0
Permettre les interruptions

3.5.6. Amélioration du cos φ.

1) Les variables.

Fi : variable qui contient la grandeur du déphasage entre U et I.

Av : variable booléenne qui indique le sens du déphasage. Cette variable est à 0 si I est en arrière sur U, à 1 si I est en avance sur U.

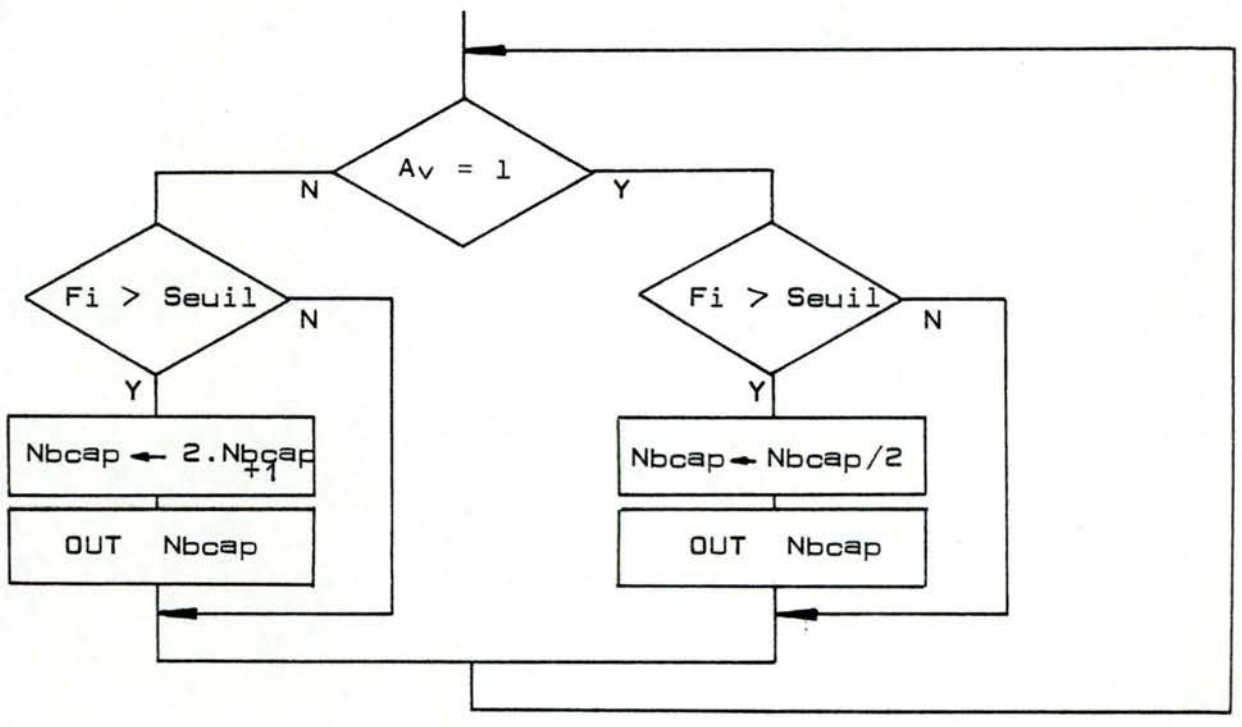
Seuil : constante qui représente la grandeur au delà de laquelle le déphasage doit être pris en considération.

Nbcap : chacun des huit bits de cette variable est associé à une capacité. Si le bit est à 1 la capacité est connectée, sinon elle ne l'est pas.

2) Initialisation de l'amélioration du cos φ.

- initialisation de l'interface parallèle 8255.
- initialisation des variables :
  - Nbcap à 0
  - Fi à 0
  - Av à 0

3) Boucle de régulation.



3.5.7. Régulation de la puissance réactive par le réseau.

1] Les variables.

Fi : variable qui indique la grandeur du déphasage entre U et I.

Av : variable qui indique le sens du déphasage.

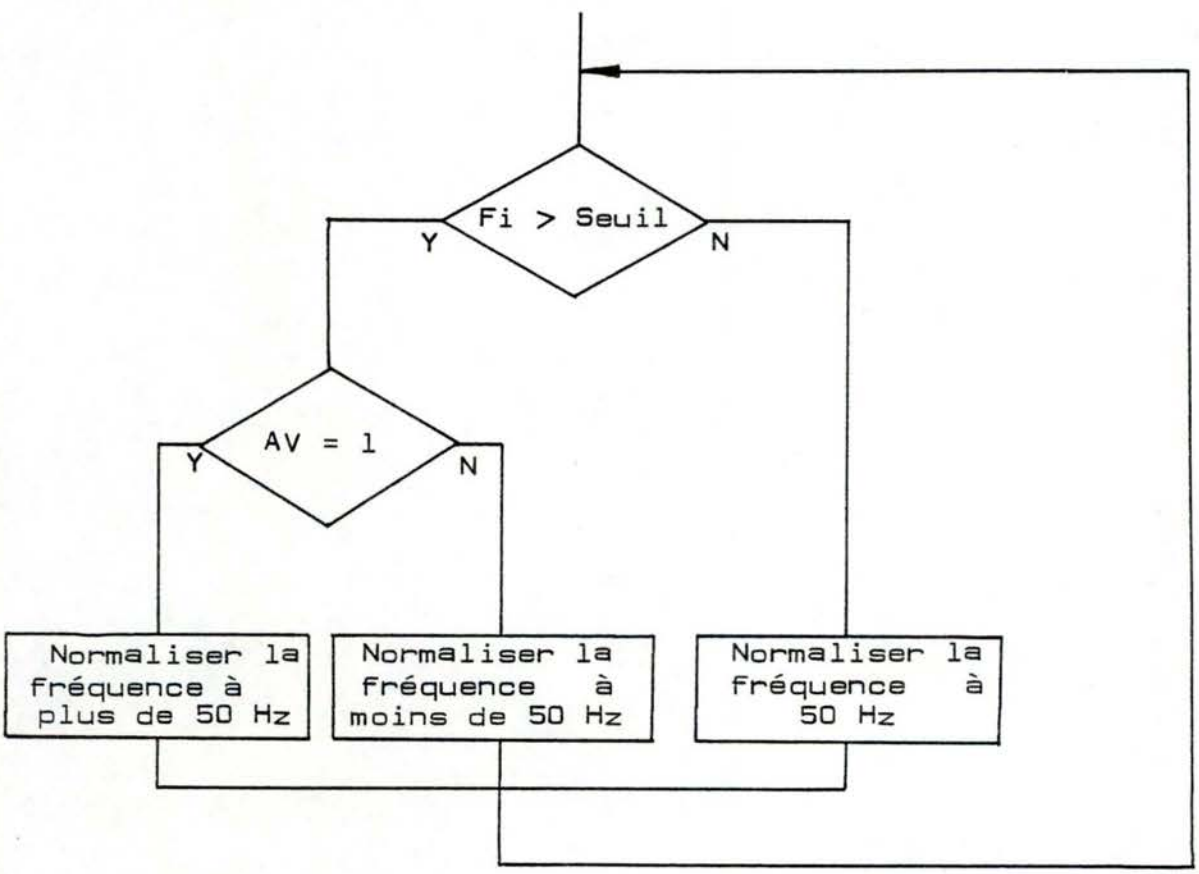
Seuil : constante qui représente la grandeur au delà de laquelle le déphasage doit être pris en considération.

Adresse 01 de la mémoire commune 74170 : contient l'indice de bourrage de temps à prendre en considération lors d'une modification de la fréquence.

2] Initialisation.

- initialisation de l'interface parallèle 8255.
- initialisation des variables :
  - Fi à 0
  - Av à 0

3] Boucle de régulation.



3.5.8. Couplage au réseau de distribution.

1] Les variables.

R : variable booléenne qui permet de dire dans la routine d'interruption qui a provoqué cette interruption. Cette variable est initialisée lors de l'appel de la routine à 0 si l'interruption provient du début de période du programme de transformation du continu en alternatif, à 1 si elle provient de l'ampli à seuil U.

Va : variable booléenne image de la sinusoïde tension. Cette variable est à 1 pendant l'alternance positive, à 0 pendant l'alternance négative.

Etat : variable qui indique dans quel état se trouve l'automate.

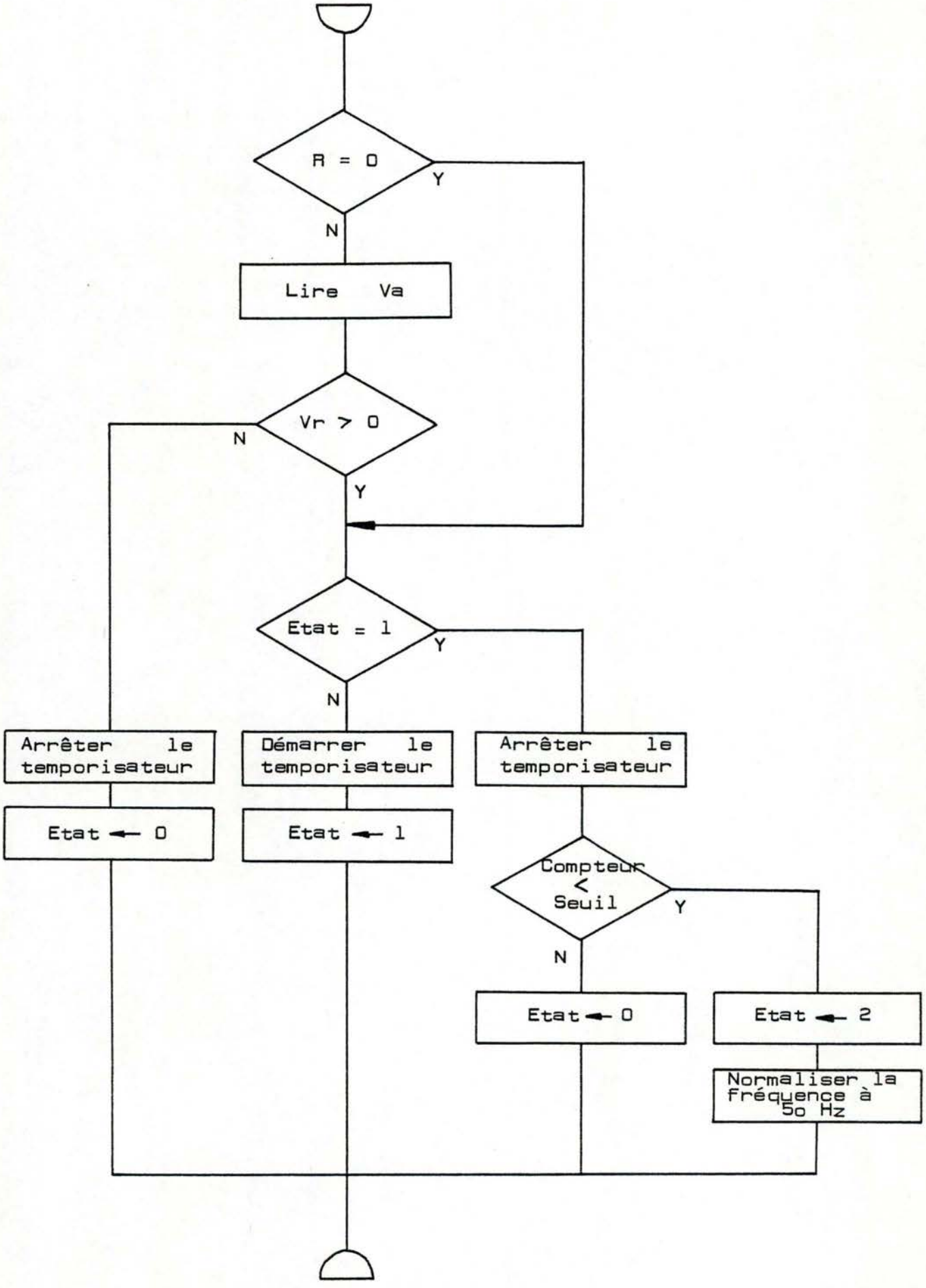
Adresse 00 de la mémoire commune 74170 : contient le numéro de la table des quantités théoriques de volt-seconde à délivrer et des moments d'enclenchement que doit utiliser le programme de transformation du continu en alternatif lorsqu'il y a plusieurs valeurs de $k = V_p/V_{max}$.

2] Initialisation du couplage au réseau de distribution.

- initialisation de l'interface parallèle 8255.
- initialisation des variables :
  - Adr 00 de 74170 à 3 (plus grande possible)
  - Adr 01 de 74170 à un indice de bourrage de temps tel que la fréquence soit différente de 50 Hz.
  - état à 0.

Lors de l'appel de la routine :

- si l'interruption provient de l'ampli à seuil U alors R := 1
- si l'interruption provient du programme principal de transformation du continu en alternatif alors R := 0.


R = 0
Y
N
Lire Va
Vr > 0
N
Y
Etat = 1
Y
N
Arrêter le temporisateur
Etat ← 0
Démarrer le temporisateur
Etat ← 1
Arrêter le temporisateur
Compteur < Seuil
Y
N
Etat ← 0
Etat ← 2
Normaliser la fréquence à 50 Hz

4. R E A L I S A T I O N.

4.1. PROGRAMME DU SYSTEME D'OPTIMISATION PRIMAIRE.

```
; initialisations
  ...............
; initialisation de l'interface parallèle
        MVI   A,90          [7,2]
        OUT   F8           [10,2]
; enclenchement de la vanne du DC-DC
        MVI   A,00          [7,2]
        OUT   F9           [10,2]
; initialisation des variables
        MVI   A,00          [7,2]
        STA   PREF         [13,3]
        STA   SIGNE        [13,3]
        MOV   C,A           [5,1]
        MVI   A,FF          [7,2]
        MOV   D,A           [5,1]
; boucle de régulation
  ....................
BCL :   INC   C             [5,1]
; test de fin de boucle
        MVI   A,00          [7,2]
        CMP   C             [4,1]
        JZ    CALCUL       [10,3]
; enclenchement/déclenchement
        MOV   A,C           [5,1]
        CMP   D             [4,1]
        JMP   ENCL         [10,3]
        MVI   A,00          [7,2]
        OUT   F9           [10,2]
        JMP   BCL          [10,3]
ENCL :  MVI   A,01          [7,2]
        OUT   F9           [10,2]
        JMP   BCL          [10,3]
; calcul de delta-P
CALCUL: IN    F8           [10,2]
        STA   PT           [13,3]
        LXI   H,PREF       [10,3]
        SUB   M             [7,1]
        JP    ACTION       [10,3]
        CPI   xx            [7,2]
        JP    BCL          [10,3]
```

```
; inversion du signe de l'incrément
        LDA   SIGNE          [13,3]
        CMA                   [4,1]
        STA   SIGNE          [13,3]

; modification de l'angle d'ouverture
ACTION: LDA   PT             [13,3]
        STA   PREF           [13,3]
        LDA   SIGNE          [13,3]
        CPI   00              [7,2]
        JZ    DECR           [10,3]

        MOV   A,D             [5,1]
        CPI   FF              [7,2]
        JZ    BCL            [10,3]
        INR   A               [5,1]
        MOV   D,A             [5,1]
        JMP   BCL            [10,3]

DECR:   MOV   A,D             [5,1]
        CPI   00              [7,2]
        JZ    BCL            [10,3]
        DCR   A               [5,1]
        JMP   BCL            [10,3]

                [nbre de cycles, nbre de bytes]
```

4.2. PROGRAMMES DU SYSTEME DE LA CENTRALE SOLAIRE AUTONOME.

4.2.1. Programme du système de la transformation du courant continu en alternatif.

1) programme principal.

```
; initialisations du programme principal
;       initialisation de l'interface paralèlle
        MVI   A,90          (7,2)
        OUT   FB            (10,2)
;       initialisation des variables
        MVI   C,00          (7,2)
        LXI   H,4000        (10,3)
        MVI   D,xx          (7,2)
; boucle relative à la période
BCL :   MVI   B,00          (7,2)
; mise de l'interruption au niveau 1
        MVI   A,01          (7,2)
        OUT   FA            (10,2)
; initialisation du pointeur de table (phase +)
        LDA   8000          (13,3)
        ADI   yy            (7,1)
        MOV   E,A           (5,1)
; 1° appel
        LDAX  D             (7,1)
        INX   D             (5,1)
        CALL  BOUCL1        (17,1)
; 2° appel
        INX   D             (5,1)
        LDAX  D             (7,1)
        INX   D             (5,1)
        CALL  BOUCL1        (17,1)
; 3° appel
        INX   D             (5,1)
        LDAX  D             (7,1)
        INX   D             (5,1)
        CALL  BOUCL1        (17,1)
; appels N°4 à N°11
        :
; normalisation
        LDA   8001          (13,3)
        CPI   01            (7,2)
        JZ    FRENO1        (10,3)
        JM    FRESU1        (10,3)
        NOP                 (4,1)
        NOP                 (4,1)
```

```
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]

FRENO1: NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]

; mise de l'interruption au niveau 0
FRESU1: MVI  A,00           [7,2]
        OUT  FA            [10,2]

; initialisation du pointeur de table (phase -)
        LDA  8000          [13,3]
        ADI  yy             [7,1]
        MOV  E,A            [5,1]

; 1° appel
        LDAX D              [7,1]
        INX  D              [5,1]
        CALL BOUCL2        [17,1]

; 2° appel
        INX  D              [5,1]
        LDAX D              [7,1]
        INX  D              [5,1]
        CALL BOUCL2        [17,1]

; 3° appel
        INX  D              [5,1]
        LDAX D              [7,1]
        INX  D              [5,1]
        CALL BOUCL2        [17,1]

; appels N°4 à N°11
          :

; normalisation
        LDA  8001          [13,3]
        CPI  01             [7,2]
        JZ   FRENO2        [10,3]
        JM   FRESU2        [10,3]

        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]

FRENO2: NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
```

```
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
FRESU2: JMP   BCL           [10,3]
```

2] sous-programme.

```
        ; correction de la surface théorique
BOUCLi: ADD   B             [7,1]
        ; boucle de la vanne fermée
BCLFER: INC   C             [5,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        SUB   M             [7,1]
        JP    BCLFER        [10,3]
        ; mémorisation de la quantité délivrée en surplus
        MOV   B,A           [5,1]
        ; ouverture de la vanne
        MVI   A,00          [7,2]
        OUT   F9            [10,2]
        ; boucle de la vanne ouverte
        MOV   A,C           [5,1]
        XCHG                [4,1]
BCLOUV: INR   A             [5,1]
        NOP                 [4,1]
        NOP                 [4,1]
        NOP                 [4,1]
        CMP   M             [7,1]
        JNZ   BCLOUV        [10,3]
        MOV   C,A           [5,1]
        XCHG                [4,1]
        ; fermeture de la vanne
        MVI   A,0i          [7,2]
        OUT   F9            [10,2]
        ; fin du sous-programme
        RET                 [10,1]
```

3] Calcul du nombre de cycles.

Pour le programme principal

initialisation de B		7 cycles
envoi de l'interruption	2 . 17 cycles	= 34 cycles
initialisation du pointeur de table		50 cycles
1° appel	2 . 29 cycles	= 58 cycles
appels N°2 à N°11	20 . 34 cycles	=680 cycles
JMP de fin de boucle		10 cycles
		839 cycles

Pour le sous-programme

- sans bourrage de temps :

[512 . 22 cycles] + [22 . 74 cycles] = 12 892 cycles

évaluation de la temporisation :

20 000 cycles - [839 + 12 892] cycles ⁒ 512 = 12 cycles

il faut ajouter 12 cycles soit 3 NOP de 4 cycles.

- avec le bourrage de temps :

[512 . [22 cycles + 12 cycles]] + [22 . 74 cycles] = 19 036 cycles

Total programme principal et secondaire

839 cycles + 19 036 cycles = 19 875 cycles

Normalisation du programme à 20 000 cycles

Il faut ajouter deux temporisations de 62 cycles chacunes

test de l'indice de normalisation : 30 cycles

On normalise le 50 Hz par 32 cycles soit huit NOP de 4 cycles

Variation de la fréquence en fonction de l'indice de normalisation N.

- N = 0 : 19 955 cycles soit 50,2 Hz
- N = 1 : 19 999 cycles soit approximativement 50 Hz.
- N = 2 : 20 083 cycles soit 49,7 Hz

4] Table des quantités de Volts-seconde et des moments d'enclenchement.

Le programme de transformation du courant continu en alternatif utilise trois tables des quantités théoriques de Volts-seconde à délivrer et des moments d'enclenchement des vannes. Ces trois tables sont implantées consécutivement à partir d'une adresse mémoire connue. Le registre DE pointe successivement par pas de 1 sur une quantité théorique de Volts-seconde à délivrer, ensuite sur un moment d'enclenchement. Pour changer de table, le programme d'optimisation générale ajoute ou retire 22 au registre E en s'assurant de ne pas déborder ni inférieurement à la première table, ni supérieurement à la troisième.

4.2.2. Programmes du système qui assure les autres fonctions.

Ce système comprend un programme principal et deux programmes d'interruption. Le programme principal initialise les interfaces et les variables des programmes d'interruption. Il boucle ensuite sur la même instruction en attendant que des interruptions demandent l'exécution des autres programmes.

L'interruption de début de période commande l'exécution du programme de régulation. Celui-ci est constitué d'une boucle d'optimisation générale, d'une boucle d'accumulation temporaire et d'une boucle d'amélioration du cos φ.

Le programme de régulation peut-être interrompu par le programme de mesure du déphasage. Ce dernier programme, ininterruptible, est exécuté à chaque interruption en provenance des ampli à seuil, c'est à dire quatre fois par période.

1] Programme principal.

```
        ; initialisations
          ...............
        ; inhiber les interruptions
                DI                      [4,1]
        ; initialisation du prog. d'optimisation générale
        ;       initialisation de l'interface parallèle
                MVI   A,90              [7,2]
                OUT   EF                [10,2]
        ;       initialisation des variables
                MVI   A,00              [7,2]
                STA   PREF              [13,3]
                STA   SIGNE             [13,3]
                MVI   A,03              [7,2]
                STA   8000              [13,3]
        ; initialisation du prog. d'accumulation temporaire
        ;       initialisation de l'interface parallèle
                MVI   A,90              [7,2]
                OUT   F7                [10,2]
        ;       initialisation des charges
                MVI   A,04              [7,2]
                OUT   F5                [10,2]
        ; initialisation du prog. d'amélioration du cos
        ;       initialisation des capacités
                MVI   A,00              [7,2]
                OUT   ED                [10,2]
        ;       initialisation des variables
                STA   FI                [13,3]
                STA   AV                [13,3]
        ; initialisation du prog. de mesure du déphasage
        ;       initialisation de l'interface parallèle
                MVI   A,92              [7,2]
                OUT   FB                [10,2]
        ;       initialisation de la gestion des interruptions
                MVI   A,xx              [7,2]
                OUT   DF                [10,2]
                MVI   A,xx              [7,2]
                OUT   DF                [10,2]
        ;       initialisation du masque des interruptions
                MVI   A,F8              [7,2]
                OUT   DF                [10,2]
        ;       initialisation des variables
                MVI   A,00              [7,2]
                STA   ETAT              [13,3]
        ; validation des interruptions
                EI                      [4,1]
        ; boucle d'attente des interruptions
          ..................................
        W :     JMP   W                 [10,3]
```

2) Sous-programmes d'appel des routines d'interruption.

```
        ; prog du début de période
        ..........................
xxxx :  JMP   REGUL          (10,3)
        ; prog relatif à l'interruption de l'ampli à seuil I
        ....................................................
        PUSH PSW             (11,1)
        MVI  A,00             (7,2)
        JMP  DEPHAS          (10,3)
        ; prog relatif à l'interruption de l'ampli à seuil U
        ....................................................
        PUSH PSW             (11,1)
        MVI  A,FF             (7,2)
        JMP  DEPHAS          (10,3)
```

3) Programmes de régulation.

```
        ; 1. programme de l'optimisation générale
        ........................................
        ; calcul de delta-p
REGUL : IN   F8              (10,2)
        STA  PT              (13,3)
        LXI  H,PREF          (10,3)
        SUB  M                (7,1)
        JP   ACTION          (10,3)
        CPI  xx               (7,2)
        JP   BCL             (10,3)
        ; inversion du signe de l'incrément
        LDA  SIGNE           (13,3)
        CMA                   (4,1)
        STA  SIGNE           (13,3)
        ; changement de table
ACTION: LDA  PT              (13,3)
        STA  PREF            (13,3)
        LDA  SIGNE           (13,3)
        CPI  00               (7,2)
        JZ   DECR            (10,3)
        LDA  8000            (13,3)
        CPI  2C               (7,2)
        JZ   ACCTEM          (10,3)
        ADI  16               (7,2)
        STA  8000            (13,3)
        JMP  ACCTEM          (10,3)
DECR:   LDA  8000            (13,3)
        CPI  00              (7,2)
        JZ   ACCTEM          (10,3)
        SBI  16               (7,2)
        STA  8000            (13,3)
        ; 2. programme d'accumulation temporaire
        .......................................
        ; lecture de la tension
ACCTEM: IN   F6              (10,2)
        STA  VP              (13,3)
```

```
; test de dépassement de la borne supérieure
        LXI   H,VP          (10,3)
        LDA   VPSUP         (13,3)
        CMP   M              (7,1)
        JM    TSTINF        (10,3)

; action associée à une tension trop grande
        MVI   A,05           (7,2)
        OUT   F5            (10,2)
        JMP   AMECOS        (10,3)

; test de dépassement de la borne inférieure
TSTINF: LXI   H,VP          (10,3)
        LDA   VPINF         (13,3)
        CMP   M              (7,1)
        JP    ACTINF        (10,3)

; action à une tension comprise entre les bornes
        MVI   A,04           (7,2)
        OUT   F5            (10,2)
        JMP   AMECOS        (10,3)

; action associée à une tension trop petite
ACTINF: MVI   A,02           (7,2)
        OUT   F5            (10,2)

; 3. programme d'amélioration du cos φ
;.....................................

; test du sens du déphasage
AMECOS: LDA   AV            (13,3)
        CPI   FF             (7,2)
        JNZ   DEPHAV        (10,3)

; test du dépassement de seuil (AV=0)
        LXIH  SEUIL         (10,3)
        LDA   FI            (13,3)
        CMP   M              (7,1)
        JM    FIN           (10,3)

; adjonction d'une capacité
        LDA   NBCAP         (13,3)
        RLC                  (4,1)
        INR   A              (5,1)
        STA   NBCAP         (13,3)
        OUT   ED            (10,2)
        JMP   FIN           (10,3)

; test du dépassement de seuil (AV=1)
DEFAV:  LXIH  SEUIL         (10,3)
        LDA   FI            (13,3)
        CMP   M             (7,1)
        JM    FIN           (10,3)

; retrait d'une capacité
        LDA   NBCAP         (13,3)
        RRC                  (4,1)
        STA   NBCAP         (13,3)
        OUT   ED            (10,2)

; fin du sous-programme de régulation
;.....................................
FIN :   DI                   (4,1)
        MVI   A,20          (10,1)
        OUT   BF            (10,2)
        EI                   (4,1)
        RET                 (10,1)
```

4] Programme de mesure du déphasage.

```
        ; test de la provenance de l'interruption
DEPHAS: LDA   U              [13,3]
        CPI   00              [7,2]
        JZ    I              [10,3]
        IN    F8             [10,2]
        JMP   SUITE          [10,3]
I :     IN    F9             [10,2]

        ; test du signe de l'impulsion
SUITE   CPI   00              [7,2]
        JM    IMPNEG         [10,3]

        ; test de l'état de l'automate
        LDA   ETAT           [13,3]
        CPI   00              [7,2]
        JZ    INICL          [10,3]

        ; lecture du temporisateur
        MVI   A,00            [7,2]
        OUT   DF             [10,2]
        MVI   A,10            [7,2]
        OUT   DF             [10,2]
        IN    DC             [10,2]
        IN    DC             [10,2]
        CMA                   [4,1]

        ; test du dépassement du quart de période
        CPI   9C              [7,2]
        JP    IMPNEG         [10,3]

        STA   FI             [13,3]

        ; test du sens de déphasage
        LDA   U              [13,3]
        CPI   00              [7,2]
        JZ    ARR            [10,3]

        MVI   A,00            [7,2]
        STA   AV             [13,3]
        JMP   IMPNEG         [10,3]

ARR :   MVI   A,FF            [7,2]
        STA   AV             [13,3]
        JMP   IMPNEG         [10,3]

        ; remise à zéro du temporisateur
        MVI   A,30            [7,2]
        OUT   DF             [10,2]
        MVI   A,FF            [7,2]
        OUT   DC             [10,2]

        ; mise de l'état de l'automate à 1
        MVI   A,FF            [7,2]
        STA   ETAT           [13,3]
        JMP   FIN            [10,3]

        ; mise de l'état de l'automate à 0
IMPNEG: MVI   A,00            [7,2]
        STA   ETAT           [13,3]
```

```
; fin du programme de mesure du déphasage
FIN :   MVI   A,20          (7,2)
        OUT   BF            (10,2)
        POP   PSW           (10,1)
        EI                   (4,1)
        RET                 (10,1)
```

Conformément à la séquence d'instructions de saut de la zone mémoire affectée à l'interface de gestion des interruptions, la hiérarchie fixe des différents niveaux d'interruptions s'établit comme suit :

IR0 : Interruption de la phase de U solaire.
IR1 : Interruption de l'ampli à seuil I.
IR2 : Interruption de l'ampli à seuil U.
IR3, IR4, IR5, IR6, IR7 } Non affectés.

4.3. PROGRAMMES DU SYSTEME DE LA CENTRALE SOLAIRE D'APPOINT A UN RESEAU ALTERNATIF.

4.3.1. Programme du système de la transformation du courant continu en alternatif.

Même programme que pour la centrale solaire autonome.

4.3.2. Programme du système qui assure les autres fonctions.

Ce système comprend un programme principal et trois programmes d'interruption. Le programme principal comprend deux phases : la phase de couplage au réseau et la phase de régulation proprement dite. Pendant la première phase, le programme principal boucle en attendant que le programme de couplage, appellé par interruption, assure sa fonction. Il signale la fin au programme principal qui entre dans sa deuxième phase. Il boucle ensuite sur la même instruction en attendant les interruptions qui commandent l'exécution des autres programmes.

L'interruption de début de période commande l'exécution du programme de régulation. Celui-ci est constitué d'une boucle d'optimisation générale et d'une boucle de régulation de la puissance réactive par le réseau.

Le programme de régulation peut-être interrompu par le programme de mesure du déphasage. Ce dernier programme, non-interruptible est exécuté à chaque interruption en provenance des ampli à seuil, c'est à dire quatre fois par période.

1] Programme principal.

```
; initialisations du prog. de couplage
  ....................................

; inhiber les interruptions
        DI                  [4,1]

; initialiser le numéro de table
        MVI  A,03           [7,2]
        STA  8000          [13,3]

; initialiser la fréquence
        MVI  A,00           [7,2]
        STA  8001          [13,3]

; initialisation du prog. de couplage au réseau
;       initialisation de l'interface parallèle N°3
        MVI  A,90           [7,2]
        OUT  EF            [10,2]
;       initialisation de l'interface parallèle N°1
        MVI  A,92           [7,2]
        OUT  FB            [10,2]
;       initialisation de la gestion des interruptions
        MVI  A,xx           [7,2]
        OUT  DF            [10,2]
        MVI  A,xx           [7,2]
        OUT  DF            [10,2]
;       initialisation du masque des interruptions
        MVI  E7             [7,2]
        OUT  DF            [10,2]
;       initialisation des variables
        MVI  A,00           [7,2]
        STA  ETAT          [13,3]

; validation des interruptions
        EI                  [4,1]

; boucle d'attente de fin de couplage au réseau
  ............................................

W1 :    MOV  A,C            [5,1]
        CPI  02             [7,2]
        JNZ  W1            [10,3]

; initialisation des prog. de régulation
  .....................................

; inhiber les interruptions
        DI                  [4,1]

; initialisation du prog. d'optimisation générale
;       initialisation des variables
        MVI  A,00           [7,2]
        STA  PREF          [13,3]
        STA  SIGNE         [13,3]

; initialisation du prog. de régulation de la P réact
;       initialisation des variables
        MVI  A,00           [7,2]
        STA  FI            [13,3]
        STA  AV            [13,3]

; initialisation du prog. de mesure du déphasage
;       initialisation du masque des interruptions
        MVI  A,F8           [7,2]
        OUT  DF            [10,2]
```

```
;       initialisation des variables
        MVI   A,00          (7,2)
        STA   ETAT          (13,3)

; validation des interruptions
        EI                  (4,1)

; boucle d'attente des interruptions
W2 :    JMP   W2            (10,3)
```

2) Sous-programmes d'appel des routines d'interruption.

```
; prog du début de période (configuration 2)
xxxx :  JMP   REGUL         (10,3)
        DI                  (4,1)

; prog relatif à l'interruption de l'ampli à seuil I
; (configuration 2)
        PUSH  PSW           (11,1)
        MVI   A,00          (7,2)
        JMP   DEPHAS        (10,3)

; prog relatif à l'interruption de l'ampli à seuil U
; (configuration 2)
        PUSH  PSW           (11,1)
        MVI   A,FF          (7,2)
        JMP   DEPHAS        (10,3)

; prog relatif à l'interruption de début de période
; (configuration 1)
        MVI   A,00          (7,2)
        STA   R             (13,3)
        JMP   COUPL         (10,3)

; prog relatif à l'interruption de l'ampli à seuil U
; (configuration 1)
        MVI   A,FF          (7,2)
        STA   R             (13,3)
        JMP   COUPL         (10,3)
```

3) Programme de couplage au réseau.

```
; test de la provenance de l'interruption
COUPL : CPI   00            (7,2)
        JZ    E             (10,3)
        IN    F8            (10,2)

; test du signe de l'impulsion
        CPI   00            (7,2)
        JM    ETAT0         (10,3)
```

```
; test de l'état de l'automate
E :     MOV  A,C            [5,1]
        CPI  01             [7,2]
        JZ   LECT           [10,3]

; remise à zéro du temporisateur
        MVI  A,30           [7,2]
        OUT  DF             [10,2]
        MVI  A,FF           [7,2]
        OUT  DC             [10,2]

; mise de l'état de l'automate à 1
        MVI  A,FF           [7,2]
        MOV  C,A            [5,1]
        JMP  FIN            [10,3]

; lecture du temporisateur
LECT :  MVI  A,00           [7,2]
        OUT  DF             [10,2]
        MVI  A,10           [7,2]
        OUT  DF             [10,2]
        IN   DC             [10,2]
        IN   DC             [10,2]
        CMA                 [4,1]

; test du dépassement de seuil
        CPI  xx             [7,2]
        JP   ETAT0          [10,3]

; normalisation de la fréquence à 50 Hz
        MVI  A,             [7,2]
        STA  8001           [13,3]

; commande du coupleur
        MVI  A,01           [7,2]
        OUT  ED             [10,2]

; mise de l'état de l'automate à 2
        MVI  A,02           [7,2]
        MOV  C,A            [5,1]
        JMP  FIN            [10,3]

; mise de l'état de l'automate à 0
ETAT0 : MVI  A,00           [7,2]
        MOV  C,A            [5,1]

; fin du programme de couplage
FIN :   MVI  A,20           [7,2]
        OUT  BF             [10,2]
        EI                  [4,1]
        RET                 [10,1]
```

4] Programmes de régulation.

```
; 1. programme de l'optimisation générale
;.....

; calcul de delta-p
REGUL : IN   F8             [10,2]
        STA  PT             [13,3]
```

```
        LXI   H,PREF        (10,3)
        SUB   M              (7,1)
        JP    ACTION        (10,3)
        CPI   xx             (7,2)
        JP    BCL           (10,3)

; inversion du signe de l'incrément
        LDA   SIGNE         (13,3)
        CMA                  (4,1)
        STA   SIGNE         (13,3)

; changement de table
ACTION: LDA   PT            (13,3)
        STA   PREF          (13,3)
        LDA   SIGNE         (13,3)
        CPI   00             (7,2)
        JZ    DECR          (10,3)

        LDA   8000          (13,3)
        CPI   2C             (7,2)
        JZ    PUISRE        (10,3)
        ADI   16             (7,2)
        STA   8000          (13,3)
        JMP   PUISRE        (10,3)

DECR:   LDA   8000          (13,3)
        CPI   00             (7,2)
        JZ    PUISRE        (10,3)
        SBI   16             (7,2)
        STA   8000          (13,3)

; 2. programme de régulation de la puissance réactive
;.....

; test du dépassement de seuil
PUISRE: LDA   SEUIL         (13,3)
        LXI   H,FI          (10,3)
        CMP   M              (7,1)
        JP    FRENOR        (10,3)

; test du sens de déphasage
        LDA   AV            (13,3)
        CPI   FF             (7,2)
        JZ    FRESUP        (10,3)

; normalisation de la fréquence à moins de 50 Hz
        MVI   A,02           (7,2)
        STA   8001          (13,3)
        JMP   FIN           (10,3)

; normalisation de la fréquence à plus de 50 Hz
FRESUP: MVI   A,00           (7,2)
        STA   A,8001        (13,3)
        JMP   FIN           (10,3)

; normalisation de la fréquence à 50 Hz
FRENOR: MVI   A,01           (7,2)
        STA   8001          (13,3)

; fin du sous-programme de régulation
FIN :   DI                   (4,1)
        MVI   A,20          (10,1)
        OUT   BF            (10,2)
        EI                   (4,1)
        RET                 (10,1)
```

5] Programme de mesure du déphasage.

Même programme que pour la centrale solaire autonome.

Conformément à la séquence d'instructions de saut de la zone mémoire affectée à l'interface de gestion des interruptions, la hiérarchie fixe des différents niveaux d'interruptions s'établit comme suit :

IR0 : Interruption de la phase de U solaire.
IR1 : Interruption de l'ampli à seuil I.
IR2 : Interruption de l'ampli à seuil U.
IR3 : Interruption de la phase de U solaire.
IR4 : Interruption de l'ampli à seuil U.
IR5, IR6, IR7 } Non affectés.

D E U X I E M E P A R T I E.

RESEAU D'INTERCONNEXION ENTRE EQUIPEMENTS DE TRAITEMENT ET DE TRANSMISSION DE DONNEES A MICROPROCESSEUR.

5. RESEAU D'INTERCONNEXION ENTRE EQUIPEMENTS DE TRAITEMENT ET DE TRANSMISSION DE DONNEES A MICROPROCESSEUR.

L'étude antérieure portait sur un système de conduite d'une centrale solaire par microprocesseur pour une application limitée en puissance ou pour un jeu de charges peu important. Il apparaît d'une part, que la puissance développée pour l'énergie solaire peut, par multiplication des modules non seulement dans le même site mais également dans des sites différents, augmenter de façon théoriquement illimitée. D'autre part, chaque centre de conversion de l'énergie assurera de plus en plus de fonctions de contrôle, d'aide à la maintenance, de statistiques de gestion. Une telle configuration nécessite un réseau d'interconnexion entre équipements de traitement et de transmission de données à microprocesseur.

Si les sites d'implantation des centrales solaires sont différents, la coordination des différents centres avec répartition des puissances fournies, des charges et de la gestion des aléas, des pannes se fera par l'intermédiaire d'un réseau présenté au point 5.2. Si au contraire plusieurs modules sont utilisés dans un même site, nous nous référerons au point 5.3. qui examine les possibilités de choix d'architecture entre ces modules.

5.1. LES OBJECTIFS DU SYSTEME.

5.1.1. Analyse de la situation.

L'implantation des différentes centrales solaires a lieu dans des sites géographiquement éloignés. Chaque centrale constitue une entité qui possède son propre mécanisme de régulation. Toutes les centrales solaires fonctionnent en parallèle et débitent dans un réseau de puissance.

Une telle interconnexion a l'avantage d'offrir une grande souplesse et une sécurité. En particulier, elle permet le choix entre les centrales disponibles pour assurer à chaque instant la production. Elle diminue l'irrégularité de la puissance demandée, car des compensations interviennent sur la consommation portant sur l'ensemble des demandeurs.

Par contre, elle rend nécessaire un contrôle permanent des échanges d'énergie. La surveillance des transferts d'énergie, les ordres de réglage aux différentes centrales solaires et la conduite du réseau sont confiés à un poste central, véritable centre de décision.

Le poste central doit être en mesure :

1. D'effectuer des relevés d'état sur les différentes centrales solaires.
2. D'envoyer des ordres à chacune des différentes centrales solaires pour réaliser une charge harmonieuse dans le réseau de puissance.
3. D'effectuer des contrôles et des signalisations diverses.

5.1.2. Analyse des besoins.

5.1.2.1. Analyse des besoins au niveau des relevés d'état.

- Les changements d'état des diverses unités seront communiqués spontanément des postes secondaires vers le poste central.
- Il faut permettre au poste central de prendre connaissance de la table des paramètres des différentes centrales solaires.
- Il faut également permettre au poste central de rafraîchir la

table des paramètres des centrales solaires. Ces paramètres sont relatifs : premièrement, aux tensions nominales et aux puissances effectives. Les seuils seront fixés à $\pm$ 2%, $\pm$ 5%, $\pm$ 10%. Deuxièmement, aux facteurs de puissance, c'est-à-dire le cos φ des différentes charges des divers postes secondaires. Troisièmement, aux priorités des postes secondaires et, dans ceux-ci, les priorités des unités connectées.

5.1.2.2. Analyse des besoins au niveau des ordres.

Le poste central peut soit commander une gestion anticipée de ses postes satellites ou bien commander la gestion en temps réel des postes satellites selon des circonstances de prévisions météorologiques ou de maintenance dans une centrale solaire.

1. Le poste central peut ordonner de charger ou d'utiliser les batteries.
2. Le poste central peut ordonner de connecter ou de déconnecter les charges non-prioritaires.
3. Le poste central doit permettre le réglage de la tension et du cos φ dans chaque station.
4. Le poste central peut ordonner de consommer ou de transférer de l'énergie d'un poste satellite à un autre.

5.1.2.3. Analyse des besoins au niveau des contrôles et des signalisations.

1. Le poste central peut envoyer un message d'alerte en cas d'un important dépassement ou sous-passement prolongé de seuils donnés.
2. Le poste central peut signaler les unités de charges hors d'usage.
3. Le poste central peut mettre en service ou hors service les charges à distance pour moduler la puissance fournie en fonction des possibilités.

5.1.2.4. Analyse des besoins au niveau des mesures.

Les résultats de mesure des tensions [continue] à l'entrée [continue ou alternative] à la sortie peuvent être communiqués par

les postes satellites au poste central. Ces résultats seront communiqués à la demande, spontanément ou cycliquement.

5.2. RESEAUX D'INTERCONNECTION DE PLUSIEURS MICROPROCESSEURS SITUES DANS DES SITES DIFFERENTS.

Notre préférence ira plutôt pour un réseau privé que public. Les réseaux publics commutés (téléphone, télex) n'offrent pas une grande fiabilité. D'autre part, les centrales solaires peuvent être implantées en un lieu défavorisé et mal désservi par de tels réseaux.

Les procédures d'accès au lien entre deux stations sont essentiellement composées de quatre éléments.

1. La structure des informations.
2. La discipline des voies.
3. L'établissement de la connexion.
4. La supervision des échanges.

5.2.1. La structure des informations.

Il y a plusieurs structures possibles pour l'information. Ces structures d'information dépendent du hardware utilisé et donc du niveau de synchronisation utilisé pour la reconnaissance des chaines de bits.

Les différents niveaux de synchronisation sont les suivants :

- La synchronisation au niveau du caractère.
- La synchronisation au niveau du bloc.
- La synchronisation au niveau de la trame.

5.2.1.1. La synchronisation au niveau du caractère.

Les bits d'un même caractère sont régulièrement espacés et l'intervalle séparant deux caractères peut avoir une durée quelconque. La transmission est donc synchrone par élément binaire mais asynchrone par caractère.

Les caractères transmis en mode start-stop sont précédés d'un élément de départ (START) et suivis d'un élément d'arrêt (STOP).

A l'état de repos, la ligne a une certaine tension. Le départ d'un caractère est donné par la transition de cette tension à la tension zéro. La ligne reste à cette tension pendant la durée prévue pour un bit; c'est le bit de départ. Les bits représentant le caractère proprement dit suivent le bit de départ. Pour terminer le caractère, la ligne retourne à l'état de repos où elle reste un temps équivalent à deux fois la durée du bit de départ : c'est le bit d'arrêt.

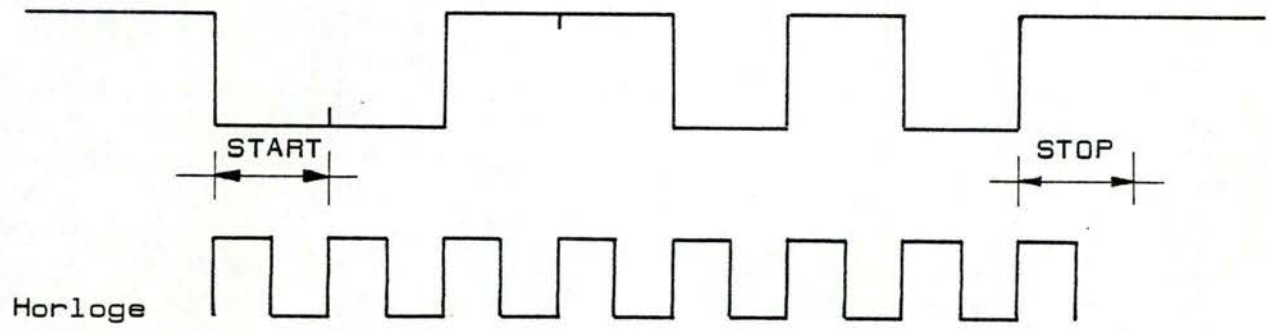


Fig 43

L'établissement de la ligne ainsi que la supervision des échanges se fait à l'aide de caractères spéciaux qui se trouvent dans le flot de caractères.

5.2.1.2. La synchronisation au niveau du bloc.

Contrairement à la synchronisation au niveau du caractère, il n'y a aucun délai entre deux caractères dans la synchronisation au niveau du bloc. Les bits d'un même message sont régulièrement espacés pendant toute la durée de la transmission.

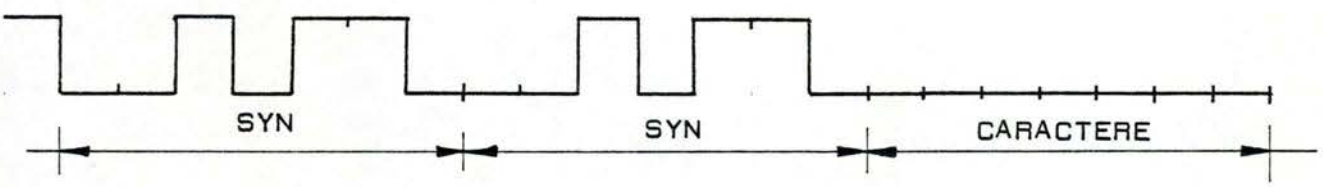
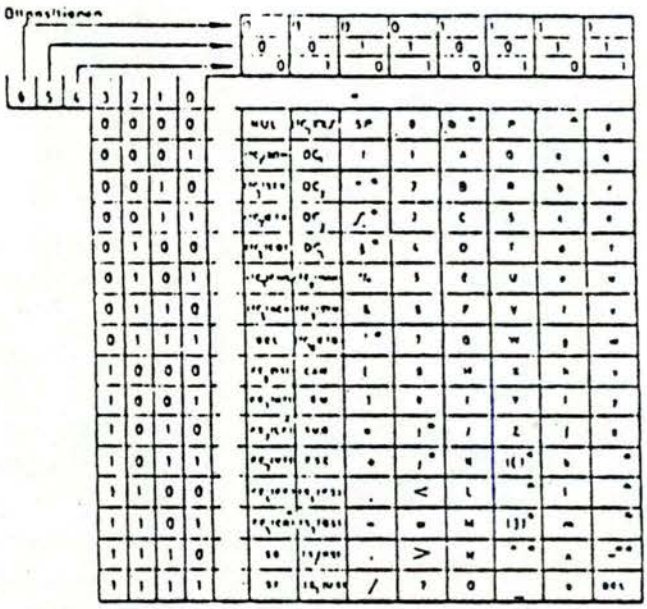


Fig 44

L'établissement de la ligne ainsi que la supervision des échanges se fait à l'aide de caractères spéciaux qui se trouvent

dans le flot de caractères. Sauf procédé pour obtenir la transparence du code, l'utilisateur ne peut utiliser que les autres combinaisons de l'alphabet donné.



ISO-7-Bit-Code (CCITT Nr. 5)

Tab 45

Désignation des commandes

ACK *(Acknowledge)*: Accusé de réception

BEL *(Bell)*: Sonnerie

BS *(Backspace)*: Retour arrière

CAN *(Cancel)*: Annulation

CR *(Carriage return)*: Retour de chariot

DC *(Device control)*: Commande d'appareil auxiliaire

DEL *(Delete)*: Oblitération

DLE *(Data link escape)*: Echappement transmission

EM *(End of medium)*: Fin de support

ENQ *(Enquiry)*: Demande

EOT *(End of transmission)*: Fin de communication

ESC *(Escape)*: Echappement

ETB *(End of transmission block)*: Fin de bloc de transmission

ETX *(End of text)*: Fin de texte

F *(Function)*: Fonction

FE *(Format effector)*: Commande de mise en page

FF *(Form feed)*: Présentation de formule

FS *(File separator)*: Séparateur de fichier

GS *(Group separator)*: Séparateur de groupe

HT *(Horizontal tabulation)*: Tabulation horizontale

IS *(Information separator)*: Séparateur d'information

LF *(Line feed)*: Interligne

NAK *(Negative acknowledge)*: Accusé de réception négatif

NL *(New line)*: Retour à la ligne

NUL *(Null)*: Nul

RS *(Record separator)*: Séparateur d'article

SI *(Shift-in)*: En code

SO *(Shift-out)*: Hors code

SOH *(Start of heading)*: Début d'en-tête

SP *(Space)*: Espace

STX *(Start of text)*: Début de texte

SUB *(Substitute)*: Substitution

SYN *(Synchronous idle)*: Synchronisation

TC *(Transmission control)*: Commande de transmission

US *(Unit separator)*: Séparateur de sous-article

VT *(Vertical tabulation)*: Tabulation verticale

Il y a deux types possibles de blocs transmis. Premièrement, les blocs d'information.Un bloc d'information se compose d'un en-tête éventuel [numérotage des blocs, nature du message, ...] suivi d'un texte. L'en-tête est précédé de SOH et le texte commence par STX et se termine par ETX. Si le message est trop long, il est découpé en plusieurs blocs. Chaque bloc est complété par l'adjonction du caractère ETB sauf le dernier qui est terminé par ETX. Les blocs sont précédés par STX ou SOH suivant qu'il s'agit d'une partie de l'en-tête ou du texte.

Exemples :

1] SOH en-tête STX texte ETX

2] SOH en-tête STX en-tête STX texte ETB STX texte ETX

Deuxièmement, les fonctions de supervision de la liaison [acquitement, invitation à émettre, invitation à recevoir, etc...] sont réalisées par les caractères de commandes EOT, ENQ, ACK et NAK. Ces caractères sont utilisés en conjonction avec des séquences d'adresse.

5.2.1.3. La synchronisation au niveau de la trame.

Une trame est un bloc structuré d'une manière particulière. Les bits d'une même trame sont régulièrement espacés.

L'organisation de la trame est conforme au format suivant :

F	A	C	données	FCS	F

F : Fanion de délimitation de trame 01111110.

Toutes les trames doivent commencer et finir par un fanion. Un même fanion peut-être utilisé à la fois comme fanion de fermeture pour une trame et fanion d'ouverture pour la trame suivante. Le fanion est utilisé pour la synchronisation de trame.

A : Adresse sur 8 bits.

L'adresse identifie la ou les stations secondaires qui sont impliquées dans l'échange de trame considérée. Si le premier bit de l'adresse est à zéro alors l'octet suivant est

une extension de l'adresse de base. Son format est le même que le précédent de telle sorte que le champ d'adresse peut être étendu par récurrence.

C : Champ de commande sur 8 bits.

Le champ de commande contient les commandes et les réponses ainsi que les numéros de séquence.

Données.

les données se présentent sur un nombre indéterminé de bits.

FCS : séquence de contrôle de trame.

La séquence de contrôle de trame est un dispositif de contrôle de parité basé sur des codes polynômiaux.

Pour assurer la transparence entre les deux fanions d'une trame, l'émetteur doit insérer un élément 0 après toutes les séquences de cinq 1 consécutifs afin de s'assurer qu'une séquence de fanion n'est pas simulée. Le récepteur élimine tout élément 0 qui suit immédiatement cinq éléments 1.

5.2.2. La discipline des voies.

5.2.2.1. Liaison symétrique-dissymétrique.

L'information circule toujours de la source vers le puits. L'accusé de réception est toujours transmis du puits vers la source.

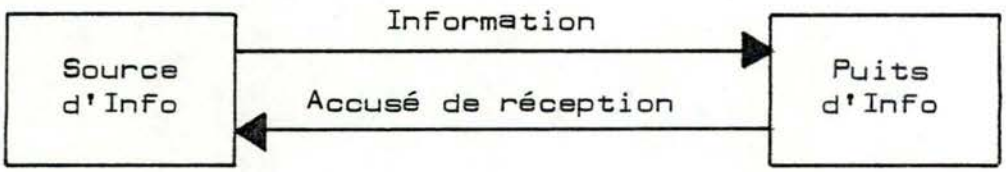


Fig 46

Lorsque la fonction primaire, responsable de la liaison, est associée à la source et la fonction secondaire au puits; c'est la fonction primaire qui prend l'initiative de transférer l'information. Dans le cas contraire, la fonction primaire demande le transfert de l'information. On parle dans le premier cas d'un mode de transfert du type invitation à recevoir et du type invitation à émettre dans le second cas.

On obtient une configuration point à point symétrique en mettant dos à dos deux sous-liaisons de type invitation à recevoir.

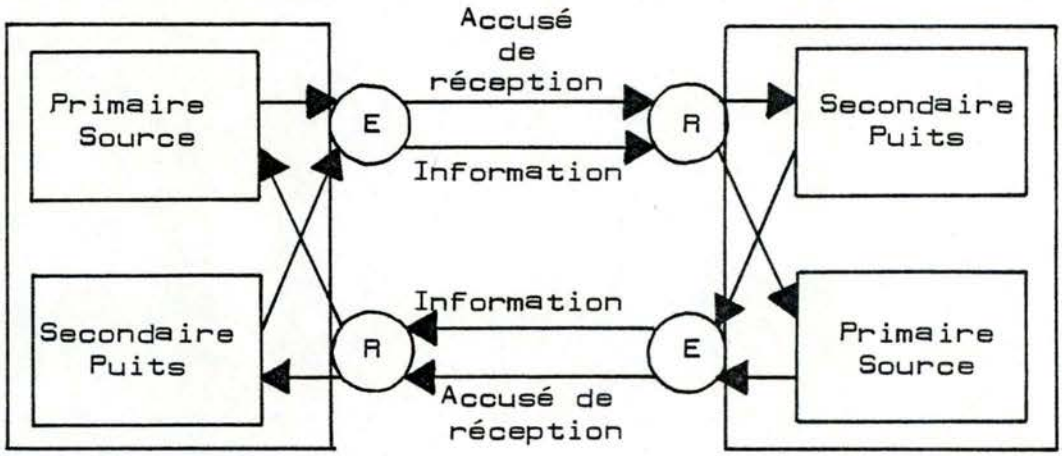


Fig 47

En mettant dos à dos deux sous-liaisons, l'une de type invitation à recevoir et l'autre de type invitation à émettre, on obtient une configuration point à point d'apparence dissymétrique. Dans ce cas, les deux fonctions primaires se trouvent dans la même station et inversément, les deux fonctions secondaires sont dans l'autre station.

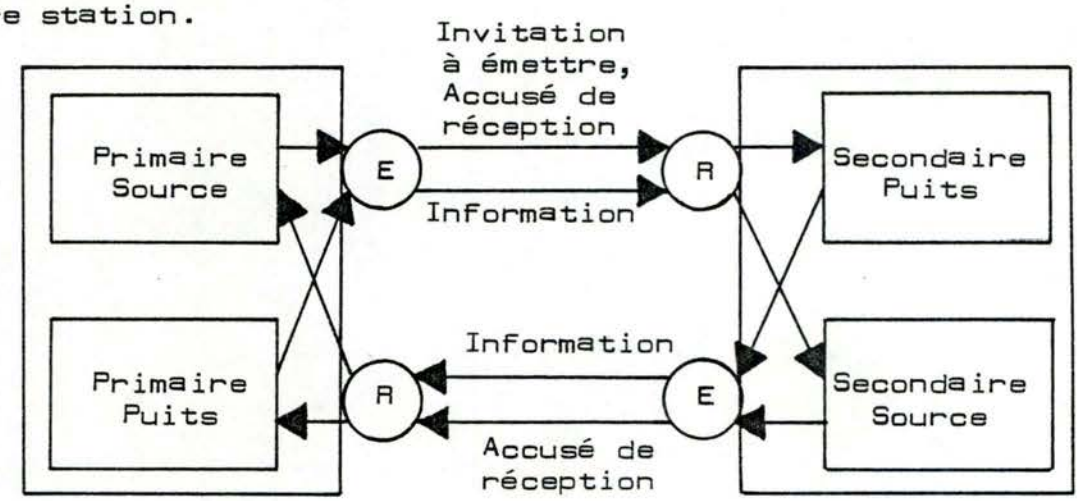


Fig 48

5.2.2.2. Types d'échanges.

1) Etoile traitement de données.

- Simple.

Toutes les stations sont reliées physiquement à un commutateur central. Les stations échangent des messages en passant par le commutateur central. Donc pour chaque station, le commutateur central est la source et la destination apparente des messages.

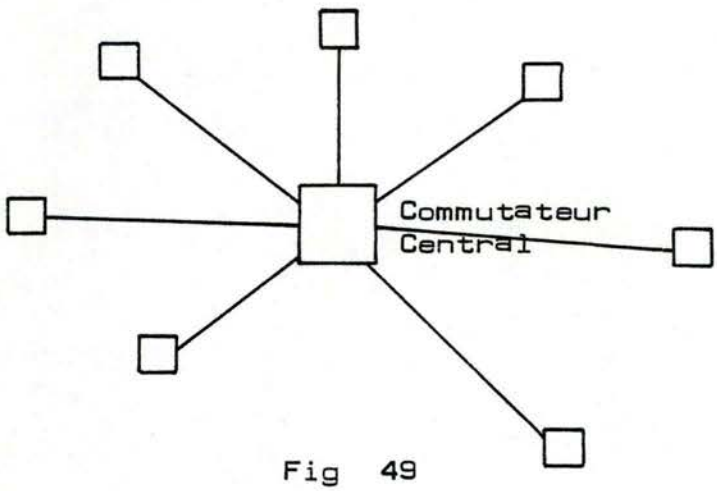


Fig 49

- à multiplexeur.

Le multiplexeur est une unité de contrôle à accès multiple et simultané. Il gère plusieurs liaisons indépendantes. Le multiplexeur explore systématiquement et à fréquence constante les lignes qui lui sont connectées. Lors de l'émission, le multiplexeur doit transmettre l'information sur la ligne convenable. Il doit par ailleurs assurer le débit de transmission aux possibilités de chacunes des lignes.

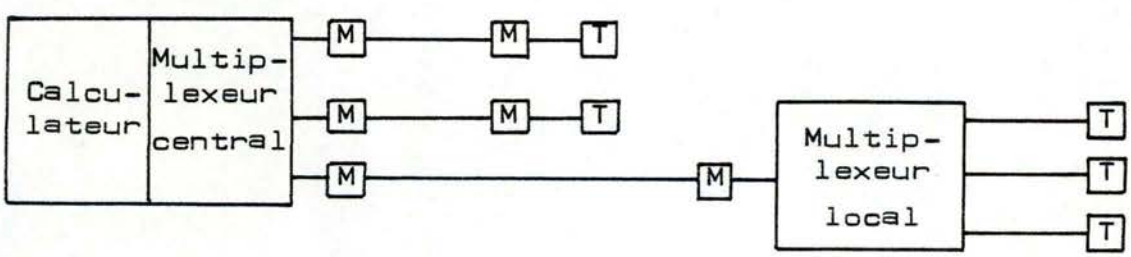


Fig 50

- à concentrateur.

Le concentrateur joue un rôle analogue à celui d'un multiplexeur. Le concentrateur opère une conversion de vitesse. Cette conversion de vitesse exige que le concentrateur possède une mémoire tampon suffisante. Le concentrateur doit également assurer des fonctions d'adressage comparables à celles d'un multiplexeur.

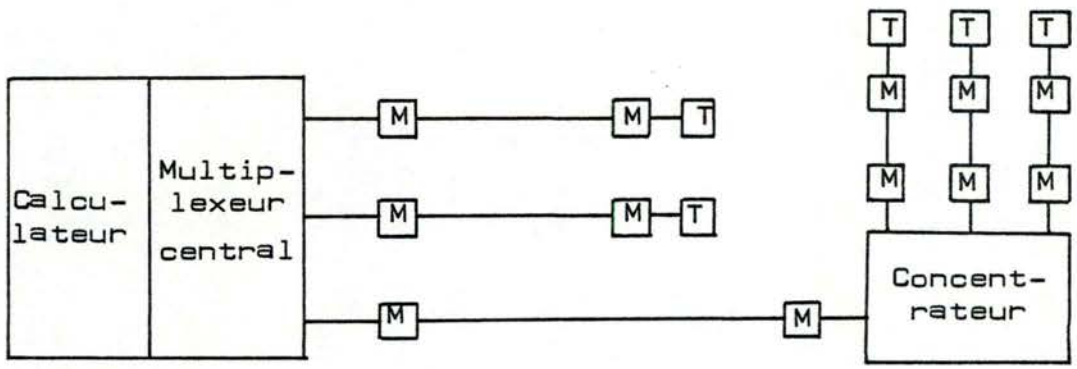


Fig 51

2) Etoile processus industriel (TRACEC 130).

Un système de télécontrôle assure la transmission des télécommandes, télésurveillances, téléréglages et télémesures,nécessaire à la conduite à distance des installations industrielles.

Les options fonctionnelles du système TRACEC 130 sont réparties en deux classes :

- Les options d'organisation.
- Les options de mode de transmission.

- Les options d'organisation de TRACEC 130.

Les options d'organisation sont relatives à l'organisation des échanges d'informations entre le poste chef (PC) et l'ensemble des postes secondaires (PS). Elles ont trait principalement à la structure point à point ou multipoint de cette organisation.

Organisation point à point.

Dans cette option, un équipement au poste chef est associé à chaque poste secondaire. Le PS peut transmettre sans interruption

des messages.

Les temps de transmissions sont courts; mais l'organisation n'est pas économique quand il y a beaucoup de PS.

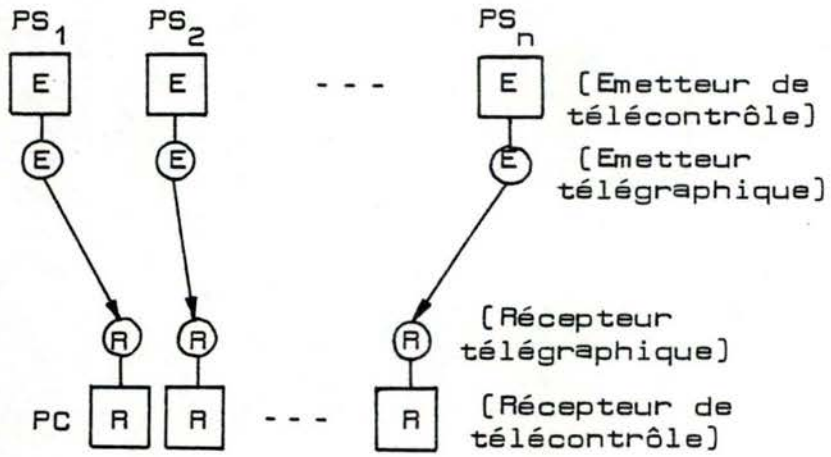


Fig 52

Organisation multipoint à interrogation.

Dans cette option, le PC est commun à l'ensemble des PS. Le PS ne transmet ses messages que lorsqu'il y a été invité par un message d'interrogation en provenance du PC. Les différents PS sont interrogés à tour de rôle.

Les temps de transmission sont allongés par rapport à l'organisation point à point mais cette organisation est économique quand il y a beaucoup de PS. Par ailleurs, l'unicité du PC est un facteur important d'économie de prix dès qu'il y a beaucoup de PS ET son couplage à un calculateur est facile.

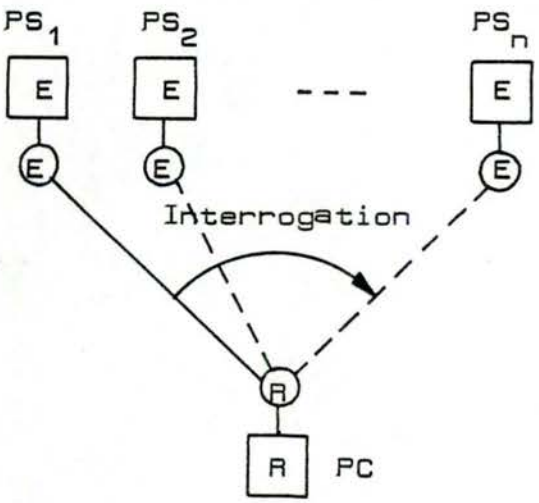


Fig 53

Organisation multipoint avec démultiplexeur.

Il s'agit d'un compromis entre l'organisation point à point et l'organisation multipoint à interrogation. Dans ce type d'organisation, chaque PS transmet ses messages sans interruption. Au PC, un démultiplexeur scrute à grande vitesse les sorties des différents PS. Le démultiplexeur restitue vers les circuits d'exploitation les messages complets et reconnus valables tant au point de vue de la synchronisation entre PS et PC qu'à celui de la satisfaction aux règles du codage redondant de ces messages.

Ce compromis conserve les avantages résultant de l'unicité du PC; supprime la tendance à l'accroissement des temps de transmission; permet des extensions à la fois de la capacité d'un PS sans allongement du temps de transmission que celui relatif à ce PS et de nouveaux PS pour autant que l'on ne dépasse pas la capacité en PS du démultiplexeur.

Il s'agit d'une organisation particulièrement efficace pour les applications comportant beaucoup de PS de grande capacité avec exploitation des informations par calculateur.

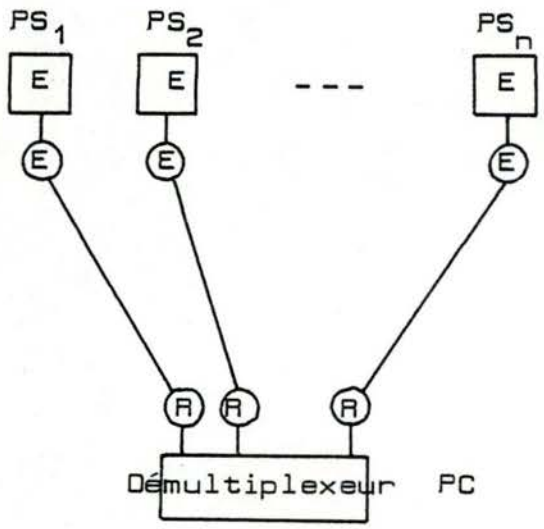


Fig 54

- Les options de transmission de TRACEC 130.

Les options de transmission sont relatives à la composition des séquences de message échangés entre le PC et chacun des PS.

Le système TRACEC 130 distingue trois catégories d'informations: des états de contacts, des mesures et des informations numériques. Pour chacunes de ces catégories d'information, plusieurs modalités de transmission sont possibles.

Pour des changements d'états : 1) la transmission cyclique.

Dans le mode de transmission cyclique, le PS transmet toutes ses informations dans une séquence de messages. Cette séquence de messages est cyclique ininterrompue dans les options point à point et multipoints démultiplexées; elle a lieu une seule fois lors de chaque interrogation dans l'option multipoints à interrogation.

Le temps de transmission d'un changement d'état dépend du temps qui sépare l'apparition de ce changement et la plus proche transmission cyclique du message porteur de cette signalisation.

Pour des changements d'états : 2) la transmission prioritaire.

En transmission prioritaire, s'il n'y a pas de changement d'état, le PS ne transmet rien excepté d'éventuelles informations d'une autre catégorie. Quand survient un changement d'état, l'éventuelle transmission en cours est interrompue au profit du message d'information contenant la signalisation de ce changement d'état.

Ce mode de transmission diminue considérablement le temps qui s'écoule entre le changement d'état et le moment où le message d'information est transmis.

Pour des changements d'états : 3) la transmission chronologique.

Lorsque survient un changement d'état, l'heure est enregistrée avec une résolution de x msec et transmise en même temps que l'adres- et l'état de contact concerné.

Pour les mesures : 1) la transmission cyclique.

Voir la transmission cyclique de la rubrique précédente.

Pour les mesures : 2) la transmission prioritaire.

Les valeurs des mesures sont comparées à des seuils fonctionnels. Ce n'est qu'en cas de dépassement d'un seuil que la transmission de la mesure correspondante s'effectue.

Pour les mesures : 3) la transmission à deux cadences.

Les mesures sont réparties en deux groupes : les mesures rapides et les mesures lentes. Au cours de chaque cycle de transmission, toutes les mesures rapides et une fraction seulement des mesures lentes sont transmises. C'est une autre fraction des mesures lentes qui sera transmise au cycle suivant et ainsi de suite.

Les cycles de transmission sont ininterrompus dans les options point à point et multipoints avec démultiplexeur; un cycle répond à une interrogation dans l'option multipoints à interrogation.

Pour les mesures : 4) la transmission conditionnelle.

Dans la transmission conditionnelle, les valeurs des mesures sont comparées à des seuils échelonnés entre 0 et 100%. Une mesure n'est transmise qu'après passage d'un de ces seuils dans l'un ou l'autre sens.

Pour les valeurs numériques : 1) la transmission spontanée périodique.

Dans ce mode de transmission, les valeurs numériques sont périodiquement introduites dans les cycles de transmission des autres informations.

Pour les valeurs numériques : 2) la transmission sur appel.

Les valeurs numériques à transmettre sont toutes mémorisées avant d'être transmises globalement ou par fraction dans les cycles

de transmission des autres informations. Et cela, à la réception d'un message de service provenant du poste chef.

- La structure des messages de TRACEC 130.

Signalisation

Classe				H
Adresse				H
				H
				H
24 signalisations				H
				H
				H
				H
V	V	V	V	H

Mesure

Classe				H
Adresse				H
				H
Mesure 1				H
		±	A1	H
				H
Mesure 2				H
		±	A1	H
V	V	V	V	H

Comptage

Classe				H
Adresse				H
				H
				H
6 Décades de comptage				H
				H
				H
				H
V	V	V	V	H

Commande

A1	B2	C3	D4	H
A4	B1	C2	D3	H
A3	B4	C1	D2	H
A2	B3	C4	D1	H
$\overline{D3}$	A'1	B'2	C'4	H
D'4	A'3	B'1	C'3	H
$\overline{D2}$	A'4	B'3	C'1	H
D'1	A'2	B'4	C'3	H
V	V	V	V	H

Synchronisation

1	1	0	1	0
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	1	0	1	0

Fig 55

Tous les messages ont un format de 45 bits. Il y a trois classes de messages.

Les messages de synchronisation.

Les messages de synchronisation sont insérés dans les télégrammes pour permettre au récepteur d'identifier correctement le début et la fin des messages d'information.

Les messages d'information transmis des PS vers le PC.

Les messages d'information transmis des PS vers le PC contiennent 45 bits dont 13 bits de redondance (9 bits de parité horizontale des 9 mots et 4 bits de parité verticale) et 32 bits répartis en 2 bits de classe d'information (signalisation, mesure, comptage), 6 bits d'adresse du message et 24 bits d'information.

Les messages d'information transmis du PC vers les PS.

Les messages d'information transmis du PC vers les PS (essentiellement des télécommandes) pour lesquels une très grande sûreté de transmission est requise sont dotés de redondances supplémentaires outre les 13 bits de parité.

Le contenu des 32 bits est réparti en quatre groupes de 8 bits. Chacun de ces groupes comprend : les quatre bits d'une décade et quatre bits donnant une répétition de la décade si les bits de leur somme est pair ou le complément de la décade si les bits de leur somme est impair.

La distance de Hamming de tout le message est encore augmentée si les 4 bits d'une décade et les 4 bits de sa répétition ne se trouvent ni sur une même ligne ni sur une même colonne.

3] Point à point.

La liaison point à point est la plus simple qui soit : elle ne comporte que deux extrémités. Dans l'échange de l'information, on distingue deux voies d'acheminement d'information. La voie principale achemine les données tandis que la voie secondaire permet le transfert des informations de service, généralement en sens inverse.

Si le circuit n'est utilisable que dans un seul sens, on parle de circuit simplex. S'il est utilisable dans les deux sens à l'alternat, il s'agit d'un circuit halfduplex. Un circuit full-duplex est utilisable dans les deux sens simultanément.

4] Réseaux multipoints.

Une liaison multipoints réside dans la mise en commun d'une ligne de télécommunication entre un équipement central et plusieurs terminaux situés dans une même direction géographique.

Dans une telle configuration, les données émises par l'équipement central sont reçues par tous les terminaux. Ils doivent être en mesure de reconnaître, à la réception, leur caractère d'identification. Les transmissions centrales ne peuvent s'effectuer que l'une après l'autre, de manière ordonnée.

5] Réseaux bouclés.

Des noeuds de transmission se greffent dans une interface d'adaptation en différents endroits d'une boucle unique et unidirectionnelle. Deux stratégies de contrôle sont possibles. Soit le contrôle est centralisé en un noeud, on parle dans ce cas de boucle; ou bien le contrôle est décentralisé et se déplace de noeuds en noeuds, on parle alors d'anneau.

La transmission des informations de noeuds en noeuds est fonction du contrôle adopté ainsi que de la technique de transmission utilisée.

- Réseau en boucle.

Le polling.

Le noeud à qui le contrôle est imputé interroge successivement chacunes des différentes stations afin de leur demander si elles ont un message à transmettre à une autre station.

La boucle de Pierce.

La boucle est parcourue par une suite de blocs. Chaque bloc possède un indicateur libre ou occupé. Quand une station désire envoyer un message, elle attend le passage d'un bloc libre. Elle positionne l'indicateur sur occupé et introduit son message dans le bloc. La station réceptrice reconnaît qu'un message lui est adressé; lit ce message et positionne l'indicateur sur libre.

Si le message est plus petit que le champ de donnée standard, celui-ci est mal utilisé. Si au contraire il est plus grand, il y a répartition sur plusieurs blocs avec indication de chaînage et de fin de message.

- Réseau en anneau.

Anneau de Newhall.

Lors de la mise en service de l'anneau, une délégation de contrôle est générée et affectée à une station. Cette station devient maîtresse de l'anneau et peut envoyer ses messages. Si elle n'a rien à transmettre ou quand elle a terminé de transmettre, elle donne la délégation de pouvoir à la station suivante et ainsi de suite. A un moment donné, seuls les messages appartenant à une même station d'origine peuvent circuler dans l'anneau. L'utilisation de l'anneau est médiocre.

Anneau de Liu.

L'interface contient un registre à décalage-tampon qui fonctionne en entrée ou en sortie. Sa longueur est au moins supérieure à

la longueur du plus long bloc à transmettre ou à recevoir.

L'interface réceptionne les appels et positionne le commutateur du noeud sur l'une des trois positions suivantes :
La position 1 permet le transfert d'éventuelles données pour d'autres noeuds.
La position 2 permet au contenu du registre du noeud de parcourir le réseau.
La position 3 permet au registre de la boucle qui s'était rempli en position 2 de se vider dans le réseau.

6) Réseaux maillé.

Un réseau maillé est composé d'un ensemble de noeuds reliés entre eux par des liaisons point à point.

La topologie d'un réseau maillé est caractérisée par le fait qu'il existe en général plusieurs chemins dans le réseau pour la transmission de données entre deux points du réseau. De ce fait, la fiabilité des communications en est considérablement augmentée et la répartition de la charge sur l'ensemble du réseau est harmonieuse. L'existence de plusieurs chemins permet en effet de choisir les chemins les plus sûrs ou les moins encombrés.

Chaque noeud assure les fonctions de stockage intermédiaire et d'acheminement des blocs de données. La limitation de la zone de stockage intermédiaire entraîne des phénomènes de blocage ou de perte de blocs. Dans le cas de blocage, le noeud refuse les blocs provenant de l'extérieur au delà de sa capacité de stockage. Ce refus est communiqué aux noeuds qui lui transmettent des blocs et la transmission est interrompue. En cas de pertes, le noeud refuse l'acquitement des blocs perdus. C'est au noeud qui les a transmis qu'il incombe de les retransmettre.

5.2.3. Procédure d'établissement de la ligne.

La liaison de donnée peut-être établie à travers un réseau commuté ou un réseau maillé ou bouclé. Il s'agit d'établir la liaison par un appel au réseau. L'appel au réseau inclut l'identification des stations connectées par la commutation. Une liaison établie à travers un réseau commuté doit être libérée quand la communication est terminée. La dernière phase d'une procédure est donc la libération de la liaison.

Il y a plusieurs types de procédure de vérification d'identité.

Procédure "je suis".

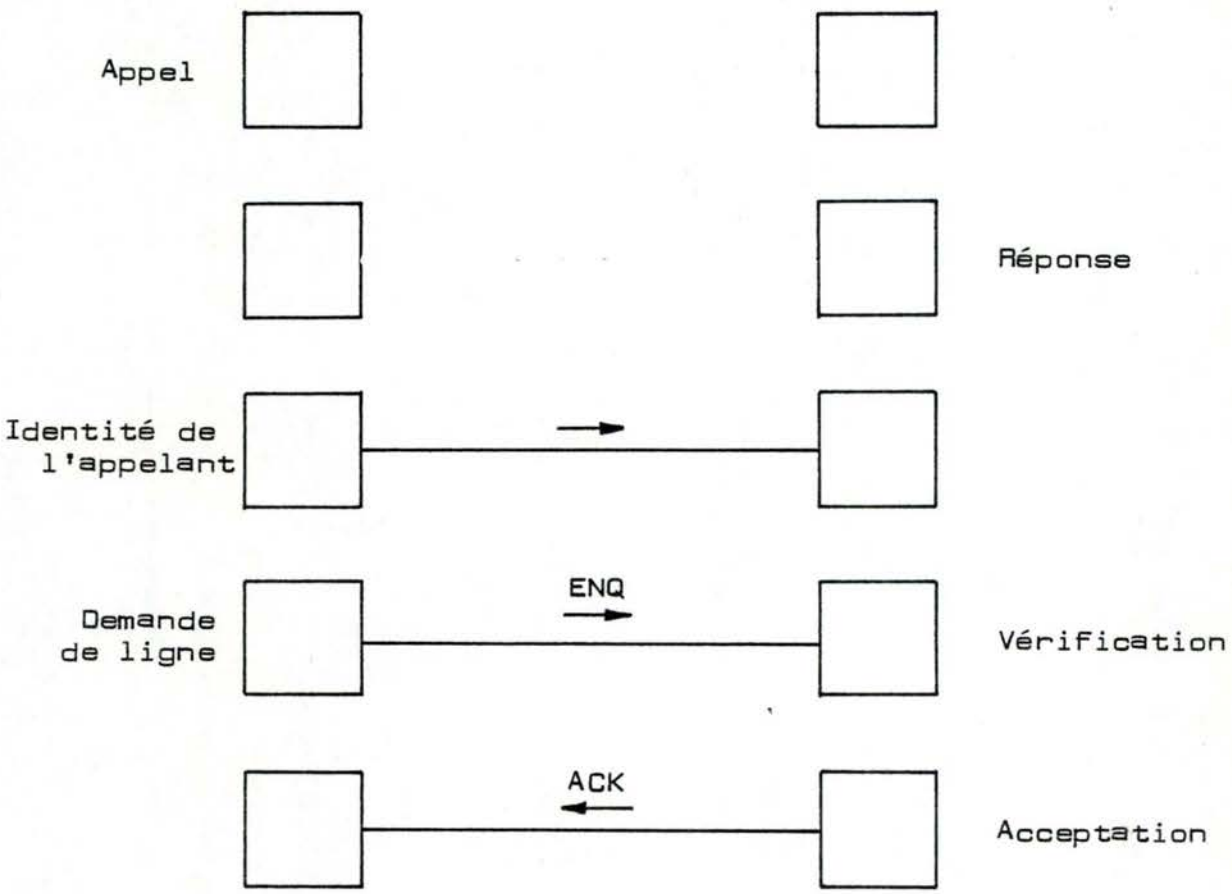


Fig 56

Procédure "je suis / qui êtes-vous".

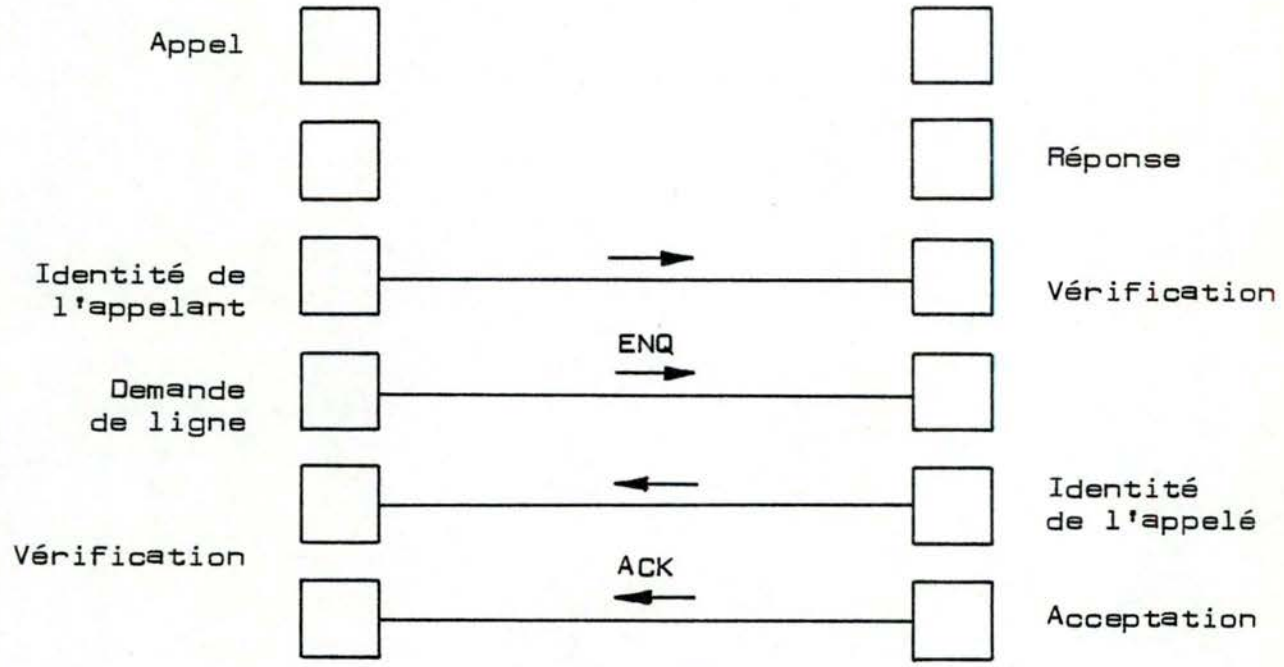


Fig 57

Procédure "qui êtes-vous".

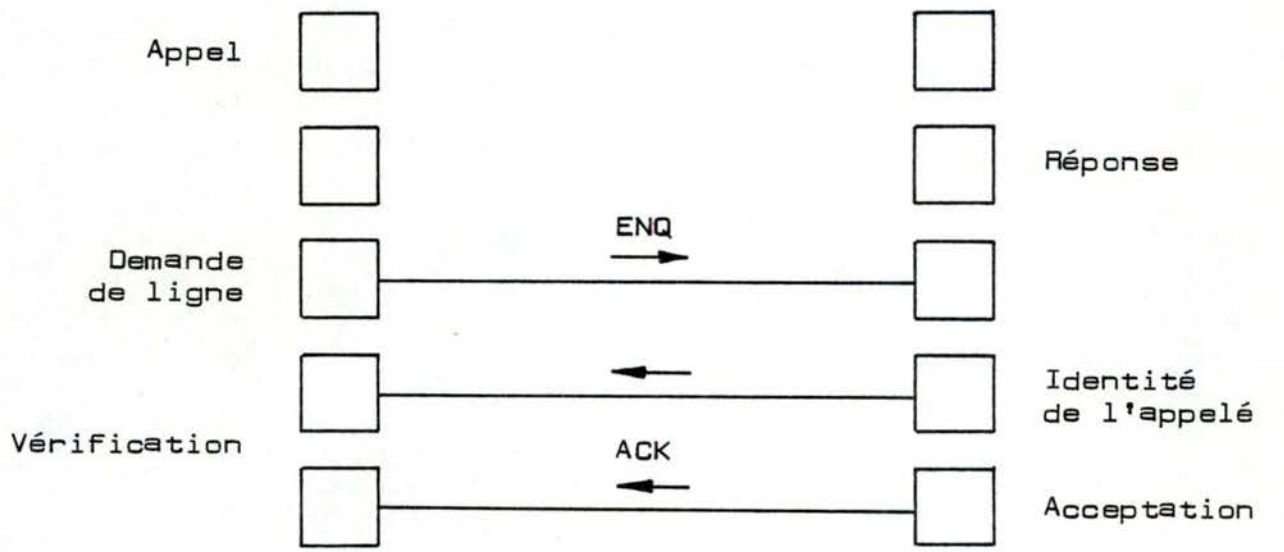


Fig 58

L'identification étant réalisée, les échanges se feront comme il a été spécifié plus haut.

5.2.4. Protocoles de supervision des échanges.

Principalement, deux techniques de transmission assurent de traiter des messages dépouillés d'erreurs lorsqu'ils sont transmis à distance. Ces techniques se basent sur : le choix d'un code détecteur et correcteur d'erreurs et une procédure de recouvrement.

5.2.4.1. Le choix d'un code détecteur et correcteur d'erreurs.

Le choix d'un code détecteur et correcteur d'erreurs est caractérisé par un paramètre que l'on appelle "distance". La distance correspond au nombre de bits qu'il s'agit de modifier pour passer d'un symbole du code à l'autre. Si d est la distance, pour un tel code, le nombre d'erreurs détectées pourra être [d-1] tandis que le nombre d'erreurs susceptibles d'être corrigées pourra être $[\frac{d}{2}-1]$ si d est pair et $[\frac{d-1}{2}]$ si d est impair.

Le choix d'un tel code, basé sur la redondance, n'est pratiquement utilisable que pour des messages courts. Le contenu principal d'un message [Fig 55] est de 32 bits répartis en 4 groupes de 8 bits. Chacun de ces groupes comprend : les 4 bits d'une décade et 4 bits qui donnent une répétition de la décade soit sans inversion des bits si leur somme est paire ou bien avec inversion des bits si leur somme est impaire. Par exemple :

1100 - 1100 [pair]
1011 - 0100 [impair]

D'autre part, la distance de Haming est encore augmentée en répartissant les 32 bits du contenu principal de telle manière que les 4 bits d'une décade et les 4 bits de sa répétition se trouvent aussi peu que possible dans une même ligne horizontale et une même colonne verticale.

Les bits des 4 décades et leurs répétitions sont représentés à la Fig 55 par $A_i\ B_i\ C_i\ D_i - A'_i\ B'_i\ C'_i\ D'_i$ [avec i = 1 à 4]. Ce code pourrait assurer une transmission fiable si l'on adoptait pour notre application un réseau de type étoilé tel celui de la Fig 59.

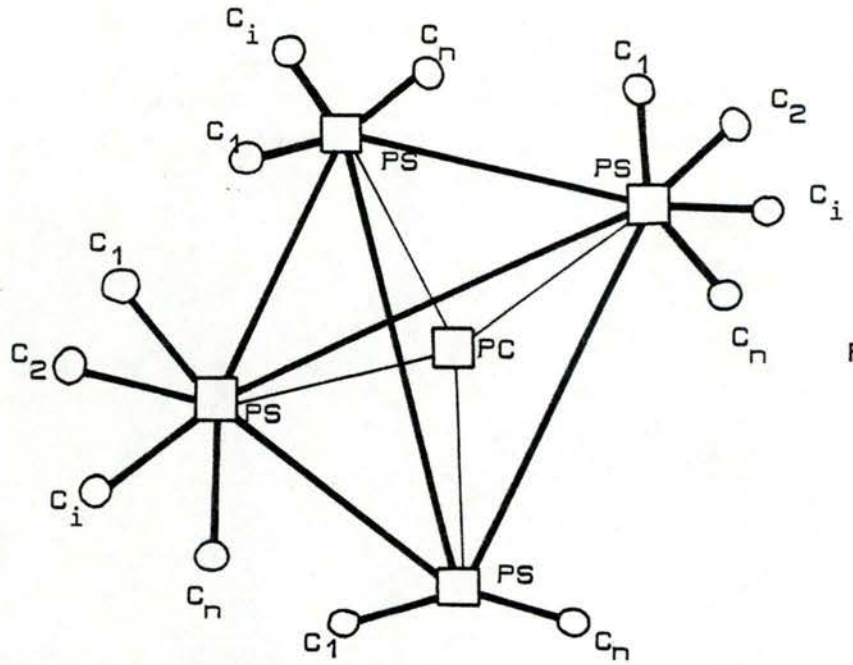


Fig 59

PC : Poste chef de contrôle et de dispatching (liaisons PC - PS_i ou PS_i - PS_j).

PS_i : Postes secondaires avec sources d'énergie solaire.

C_x : Charges - Moteurs à courant continu ou alternatif.
- Eclairage.
- Chauffage (eau)
- Force motrice pour, par exemple, exploitation fermière (broyeuse, moulin à blé, barrateuse, trayeuse).

━━━ Lignes électriques.

—— Lignes permettant les opérations de signalisations, de mesures et de commandes.

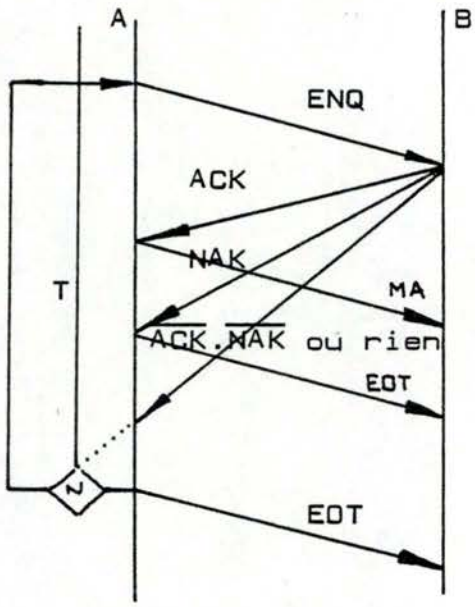
5.2.4.2. Procédure de recouvrement.

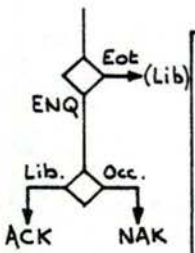
Le protocole mis en oeuvre par cette procédure pourra être soit basé sur le protocole de base illustré par les Fig 60 et 61 ou bien par le protocole HDLC.

Protocole élémentaire à un ACK - normal.

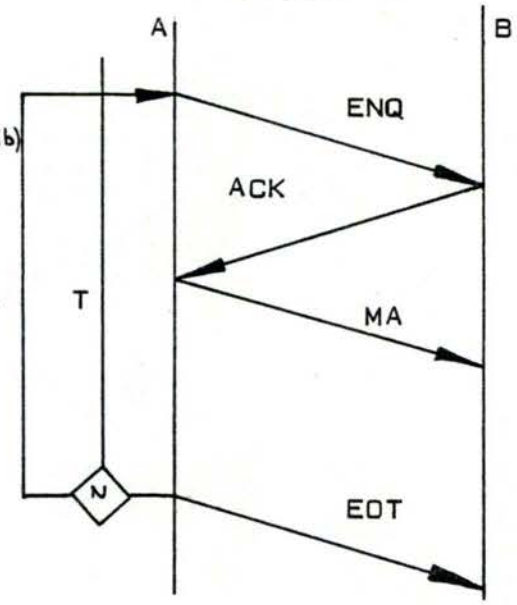
1) Appel - libération. [Appelant=A, Appelé=B]

version 1.

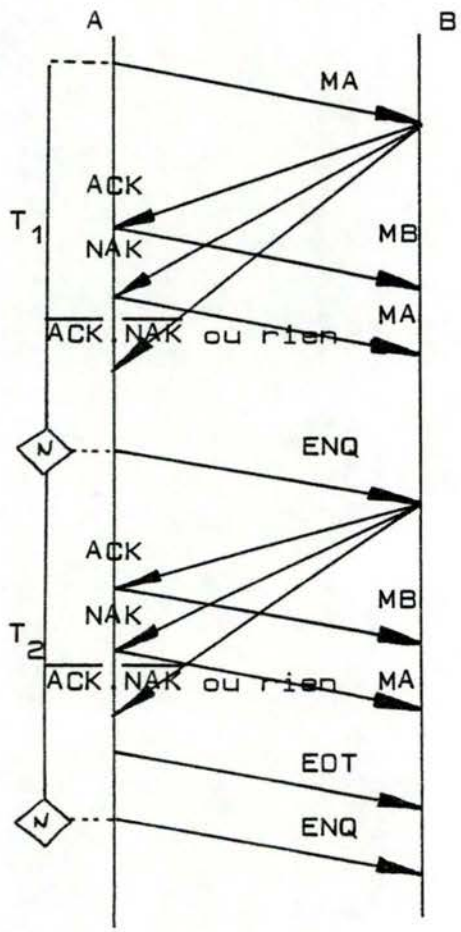




version 2.



2) Transfert de données.



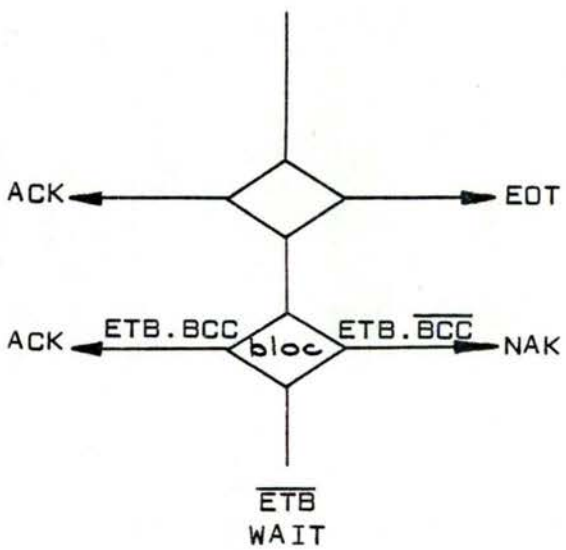
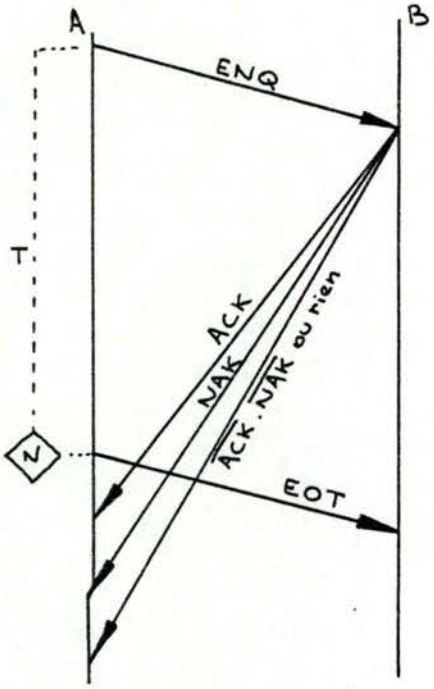


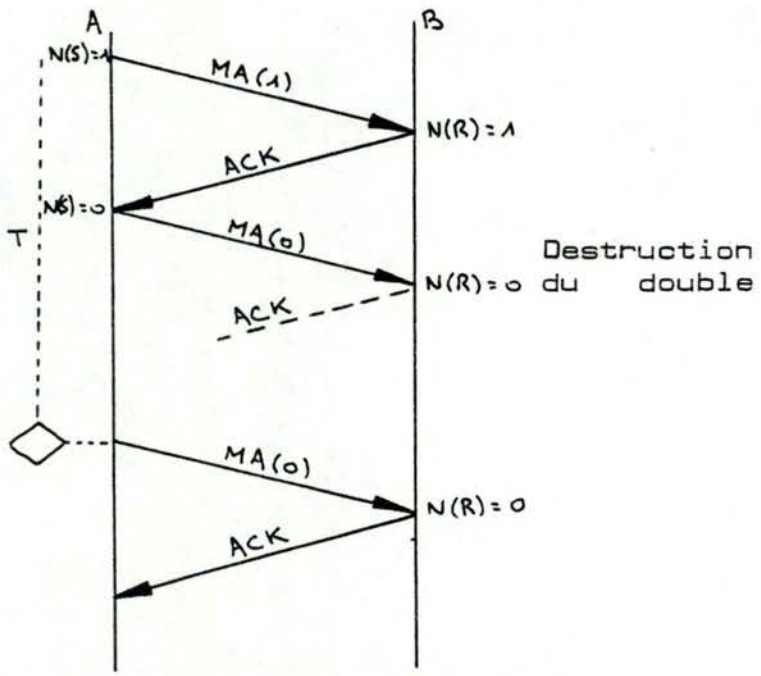
Fig 60

Protocole élémentaire à deux ACK - erreur.

1] Appel - libération. [ACK]

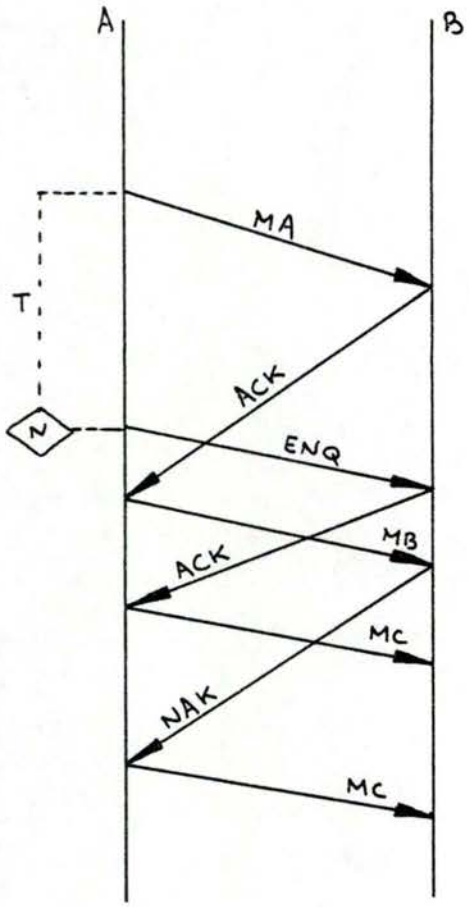


2] Transfert de données. [$MESS_0$-$MESS_1$, ACK perdu]

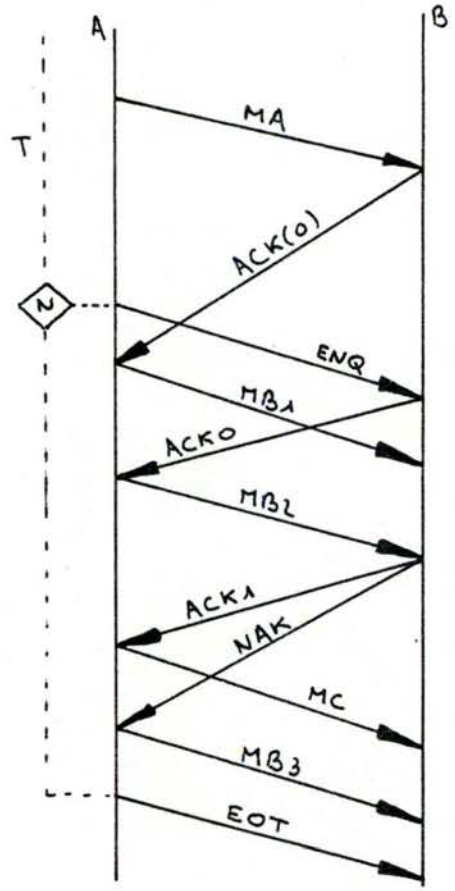


La perte d'un ACK ⟶ la numérotation des messages.

Transfert de données.
[ACK]



Transfert de données.
[ACK_0-ACK_1, ACK en retard]



Un ACK en retard ⟶ la numérotation des ACK.

Transfert de données.

- Numérotation modulo 2 des messages et des ACK.
- Une temporisation T pas trop longue.

Fig 61

Le PROTOCOLE DE BASE contient malgré lui deux types d'erreurs possibles. Sa forme la plus connue porte le sigle "BSC".

1. Appel ENQ tardif vis-à-vis du time-out. Il en résulte EOT et relâchement alors que la liaison était possible. Pour y remédier, la temporisation du timer doit être suffisamment longue.

2. Dans le transfert de données on a rupture du dialogue soit par un ACK perdu d'où à l'émetteur $\overline{ACK}.\overline{NAK}$, d'où numérotation modulo 2 des messages; soit par un ACK en retard vis-à-vis du time-out d'où numérotation modulo 2 des ACK.

Comme remède, une numérotation associe messages et accusés de réception pairs/impairs. Si on adopte le principe de la numérotation incrémentée à l'émission N(S) et à la réception N(R) avec renvoi des accusés de réception portant cette numérotation, on peut procéder par anticipation en renvoyant un ACK pour un ensemble de paquets. Une fenêtre limitant le nombre de paquets à transmettre est nécessaire.

Le PROTOCOLE HDLC a le gros avantage d'être très performant. En effet, l'information transmise est habillée d'une trame dont le format permet la synchronisation (champ Fanion), l'adressage contrôlant la source et la destination du message, le type d'opération, les paramètres de contrôle des échanges et le CRC. La gestion de cette trame est actuellement implémentée dans un circuit intégré (Fig 62) compatible avec d'une part, les modems de ligne et d'autre part, un micro-ordinateur avec ou sans DMA tandis que l'interface du microprocesseur est elle-même compatible avec toutes les gammes des miniordinateurs classiques.

Les miniordinateurs recevront donc les informations déshabillées de la trame qui aura été contrôlée de telle sorte qu'elles pourront être présentées au traitement de façon fiable sous forme de blocs physiques ou de paquets.On peut même adjoindre à ceux-ci des paramètres d'émission et de transmission constituant les références (modulo 8) des ACK d'un protocole de type de base qui exécutera un supercontrôle.

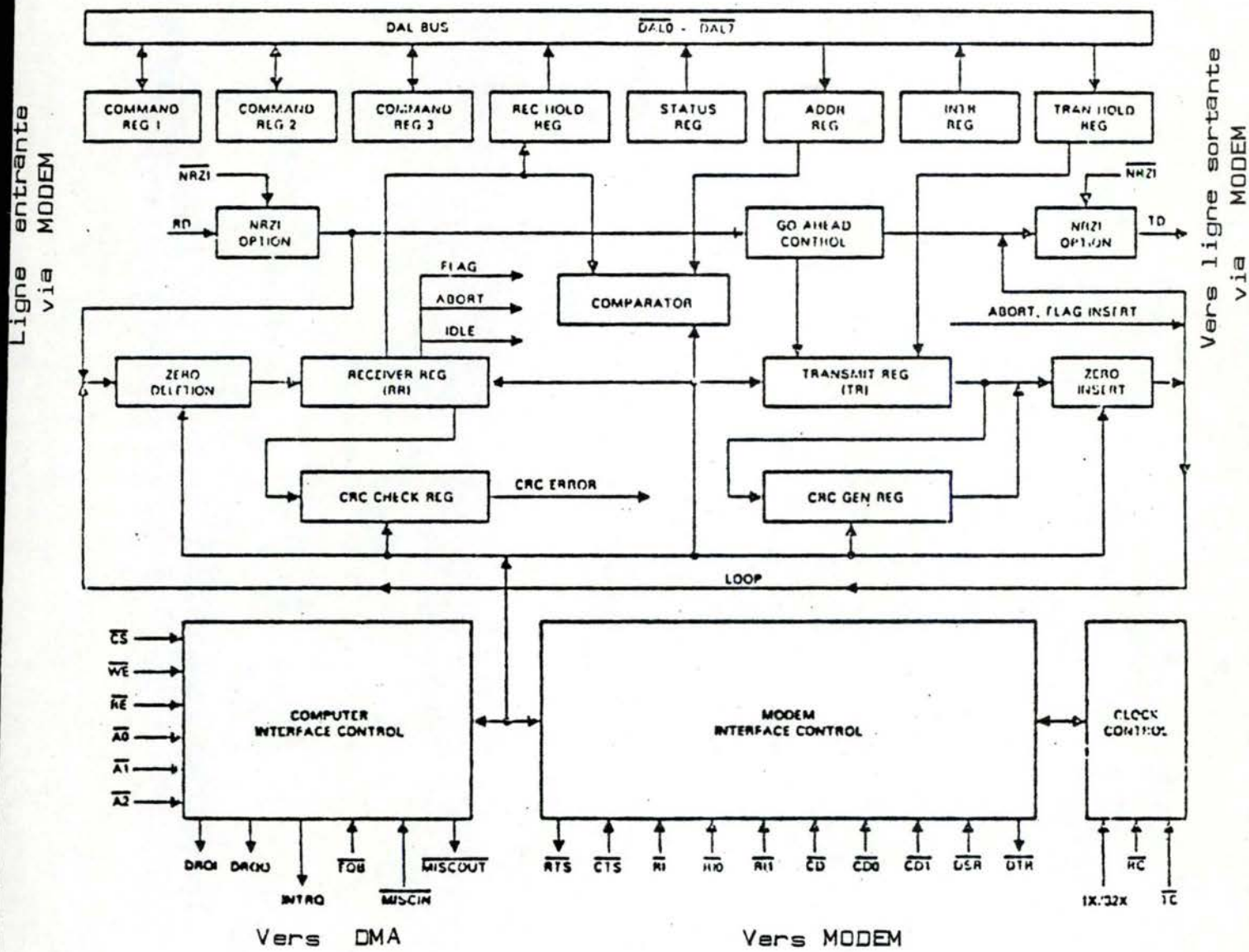


Fig 62

Pour enrichir ce protocole, de temps à autre les informations laisseront la place à des commandes (accusés de réception, appels, libérations, réinitialisations, rejets, ...). Un schéma général d'assemblage des différents blocs fonctionnels cités plus haut est représenté à la Fig 63. Des schémas particuliers de ces blocs sont représentés aux Fig 64, 65 et 66.

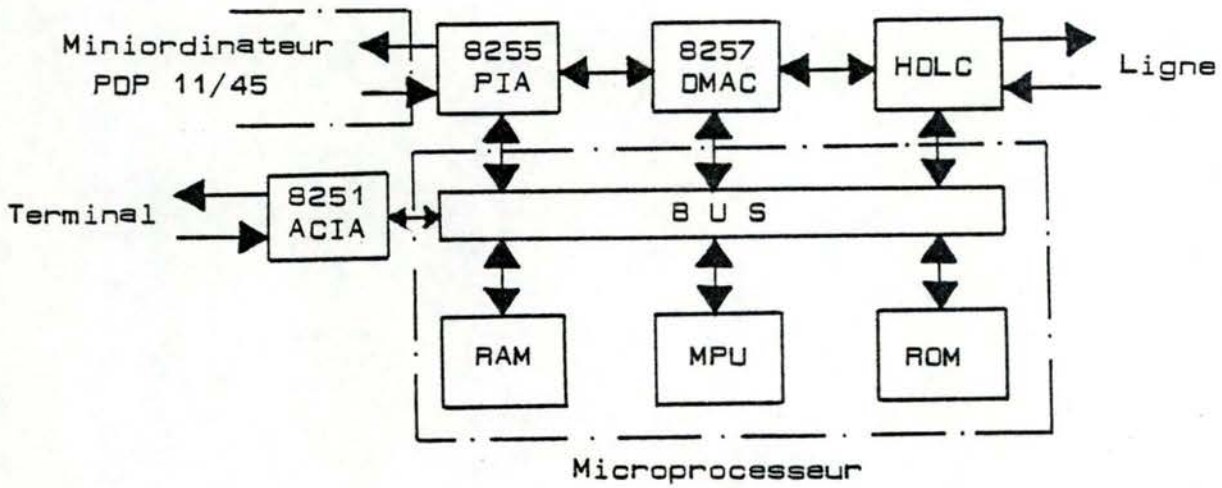

Miniordinateur
PDP 11/45
8255
PIA
8257
DMAC
HDLC
Ligne
Terminal
8251
ACIA
B U S
RAM
MPU
ROM
Microprocesseur

Fig 63

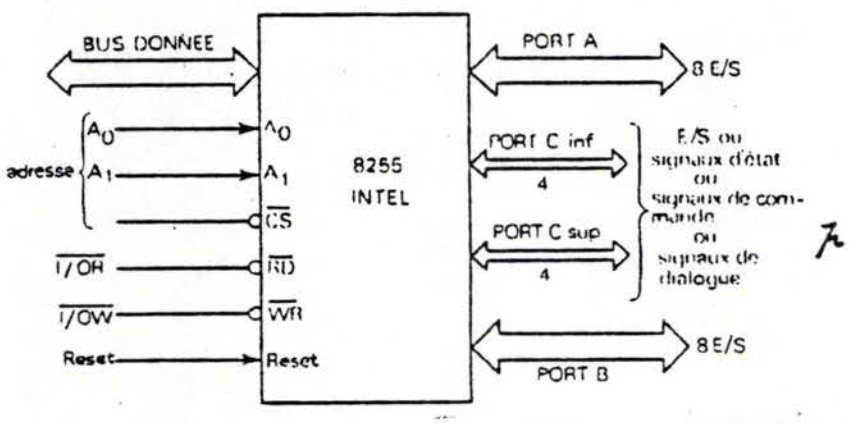

BUS DONNEE
PORT A
8 E/S
adresse
A0
A1
8255
INTEL
PORT C inf
4
PORT C sup
4
E/S ou
signaux d'état
ou
signaux de com-
mande
ou
signaux de
dialogue
I/OR
I/OW
Reset
8 E/S
PORT B

Fig 64

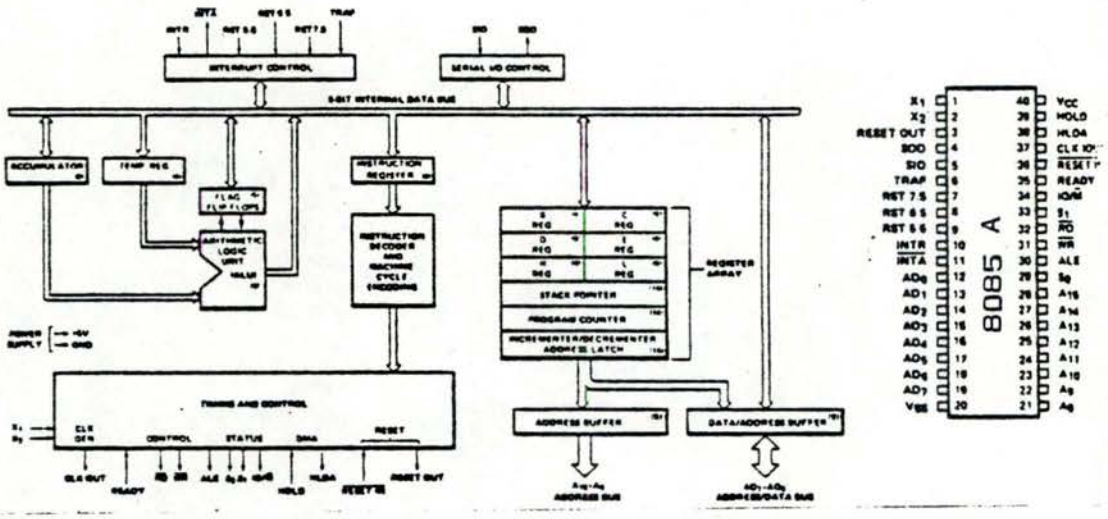


Fig 65

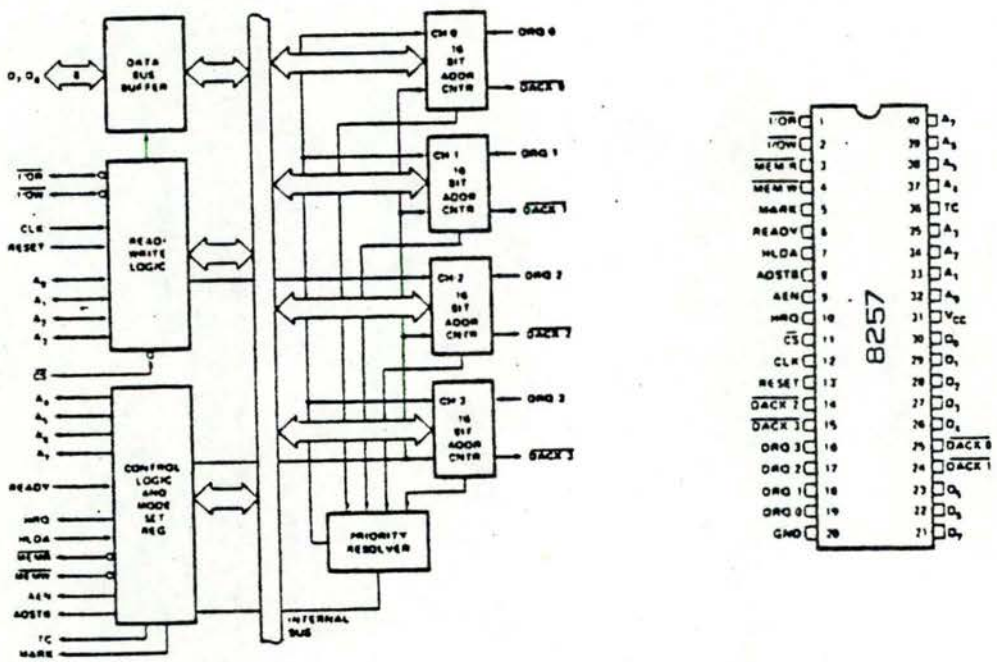


Fig 66

5.3. RESEAUX D'INTERCONNEXION MULTIPROCESSEURS SITUE DANS LE MEME SITE.

5.3.1. Introduction.

L'architecture préconisée par von Neumann interconnecte les quatre blocs fonctionnels principaux d'un ordinateur [CPU, MC, I/O et SPM] comme indiqué à la figure 67. On devrait espérer par ces interconnexions, améliorer la rapidité et la simultanéité des échanges. En pratique, toutefois, on constate d'une part, qu'il y a déséquilibre dans les flux d'échanges et d'autre part, qu'il faut prévoir l'organisation de ceux-ci.

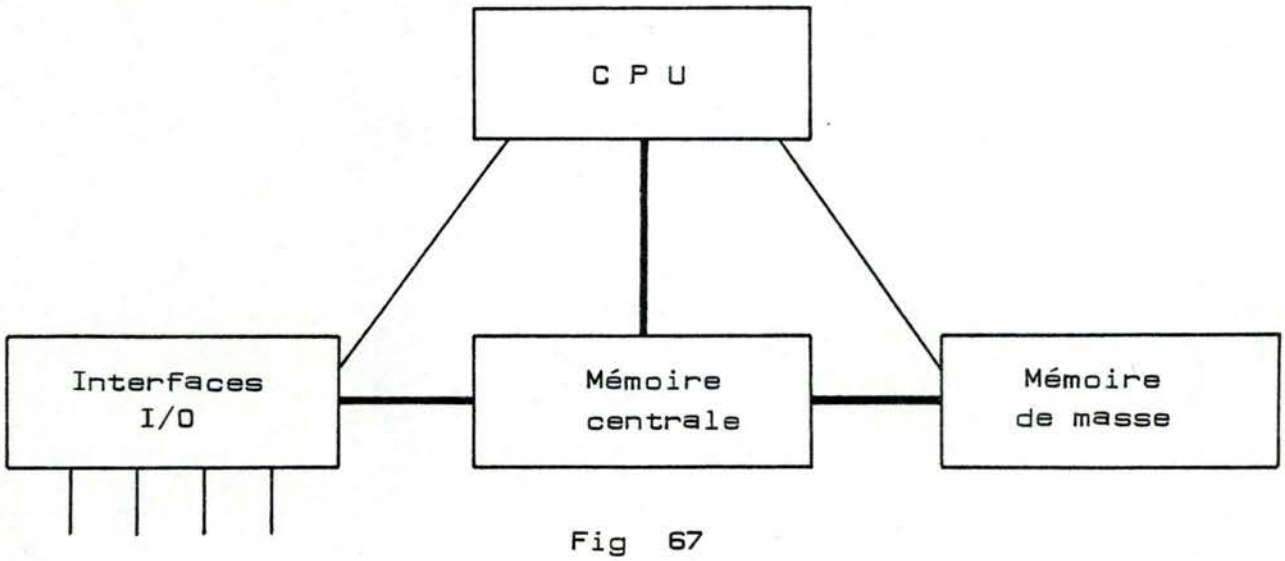


Fig 67

En effet, les échanges entre la MC et les I/O, le CPU et la SPM sont bien plus importants que ceux entre le CPU et les I/O et la SPM. Par ailleurs, l'organisation des échanges c'est-à-dire le séquencement des opérations, d'abord et la gestion des conflits d'accès ou de dépendance, ensuite, imposeraient la transformation de la structure maillée en structure étoile [Fig 68]. Le module central appelé communément "Scheduler" devra remplir toutes les fonctions susdites. Il demeure quand même que ces architectures ont le désavantage de la complexité des blocs fonctionnels de connexions au bus dues aux câblages très différents des unités d'accès. Toutes modifications de la configuration peut poser dans certains cas de sérieux problèmes.

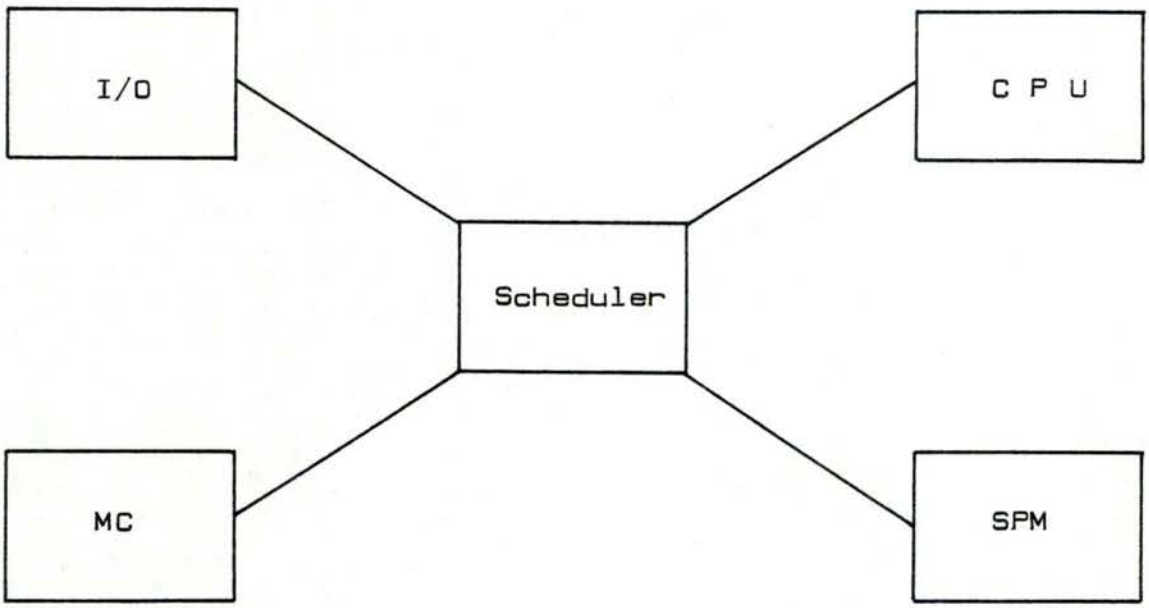


Fig 68

Ce sont là les raisons qui ont fait adopter d'autres architectures parmi lesquelles nous citerons les architectures à mémoire partagée ou à bus unique ou multiple.

5.3.2. Réseaux à bus unique ou à bus multiple.

Les transferts d'informations ou de commandes peuvent être classés en trois catégories : les transferts des données, des adresses et des contrôles.

5.3.2.1. Bus unique ou jeu unique de bus.

On peut concevoir une architecture à bus unique en acheminant sur celui-ci data et adress. L'ouverture des portes d'accès aux modules permettant le transfert de l'une ou l'autre des informations sera assurée par un microprogramme en logique cablée ou enregistrée; des ordres ou commandes (incrémentation, R/W, ...) seront donnés par ce même microprogramme (Fig 69).

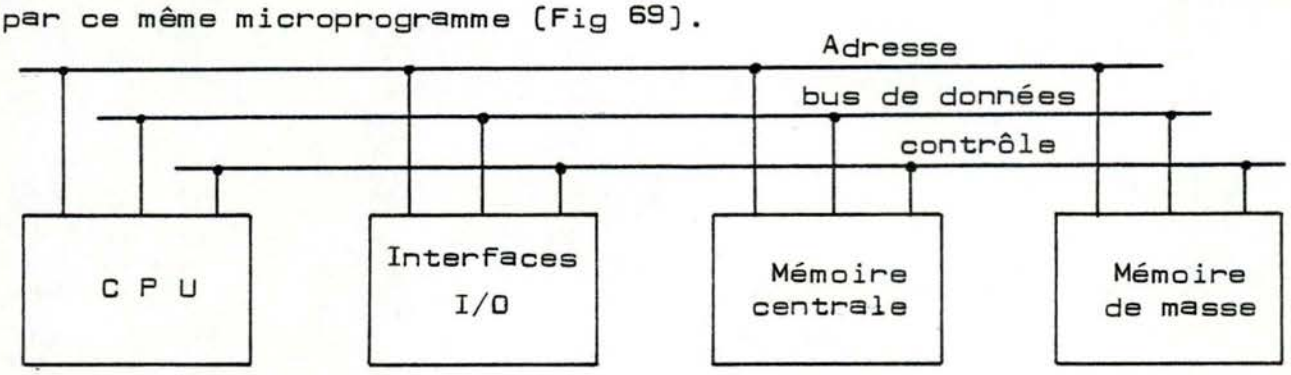


Fig 69

C'est ainsi que l'on pourra exécuter en séquence les opérations : PC[CPU] → SM[MC] puis READ puis DATA[MC] → RTM. Par contre, les opérations suivantes qui pourraient s'exécuter simultanément ne le pourront pas car le bus est commun pour les adresses et les données : il s'agit de PC[CPU] → SM[MC] et DATA[SPM] → RTM avec un WRITE.

Ce simple exemple peut illustrer l'architecture comprenant un jeu de deux bus où les DATA et ADRESS seront séparées. L'on retrouve ici les avantages, du moins en partie de la rapidité et de la simultanéité des échanges. De plus, le câblage des accès des modules aux bus sont simplifiés et banalisés.

L'architecture devient alors celle de la Fig 69 où l'on peut voir apparaître un autre avantage, celui de la facilité d'extension en blocs compatibles de fonctions quelconques.

Le contrôle des opérations élémentaires est toujours réalisé par microprogramme. Toutefois, l'organisation et la synchronisation des échanges ne peuvent plus s'appuyer sur de simples formes d'onde de tops d'horloge mais imposeront l'étude et l'élaboration de nombreux chronogrammes à créneaux judicieusement calculés et entrelacés.

Pour soulager ce contrôle tant du point de vue microprogramme que de celui des signaux de synchronisation, il y a intérêt à introduire dans le jeu des bus précédents un bus de contrôle qui comportera un certain nombre de fils de signalisation commandant directement, à un instant donné l'unité concernée. Ces contrôles consistent à :

- transmettre des ordres de transfert émis par le maître à l'esclave. S'il n'y a pas de maîtres, ces ordres seront enregistrés dans un blocs "bus arbiter" qui agira selon les priorités.
- Déterminer la direction des transferts.
- Permettre aux esclaves de demander au maître un délai supplémentaire pour terminer un transfert, s'il n'est pas possible de l'exécuter dans l'intervalle prévu.
- Transmettre les demandes d'entrée / sortie qui sont des interruptions destinées à la MC [par exemple : DMA] ou à un éventuel contrôleur des interruptions [PCI]

- Transmettre à l'interface d'entrée/sortie ou au PCI la réponse de l'unité centrale.
- Opérer des "reset".
- Signaler que le receveur est "ready".
- Synchroniser via le bloc d'horloge.

Une des fonctions importantes de certains fils du bus de contrôle que nous avons indiqués plus haut est la gestion des demandes d'unités pour transférer des informations sur les autres bus. Le mécanisme le plus simple a donc pour principe celui illustré à la Fig 72.

5.3.2.1.1. Solution mixte - bus multiplexé.

Principe.

Les bus des microprocesseurs décrits ci-dessus sont caractérisés par un grand nombre de lignes d'interconnexions, ce nombre peut poser des problèmes au niveau du dessin des cartes des circuits imprimés et également pour le nombre des broches du boîtier microprocesseur.

Pour différentes raisons il n'est pas souhaitable de fabriquer des boîtiers pour circuits intégrés avec plus de 40 broches. Certains fabricants, principalement ceux qui fabriquent des microprocesseurs 16 bits, limitent le nombre des lignes d'un bus en utilisant une même ligne pour deux fonctions distinctes. Par exemple, les mêmes lignes peuvent être utilisées pour les adresses et pour les données.

Le maître du bus place en premier lieu l'adresse de l'esclave sur le bus et ensuite la donnée correspondante est placée sur le même bus. Une ligne particulière du bus des contrôles indique en permanence la fonction actuelle du bus multiplexé.

Dans ce cas, chaque module connecté au bus doit comporter les circuits nécessaires au démultiplexage, mais dans la majorité des cas ce désavantage est négligeable comparé aux avantages du bus multiplexé.

Exemple.

Le microprocesseur 8085 d'INTEL utilise un bus multiplexé. Dans ce processeur on trouve 16 lignes [AD_0 à AD_7 et A_8 à A_{15}] qui remplissent les fonctions des bus des adresses et des données. Les lignes marquées de AD_0 jusqu'à AD_7 servent à la fois à transférer les huit bits les moins significatifs d'une adresse et à transférer les données. Les lignes de A_8 jusqu'à A_{15} sont uniquement utilisées pour les huit bits les plus significatifs des adresses.

Une ligne de contrôle particulière, marquée ALE [Address Latch Enable] indique la fonction des lignes AD_i. Quand ALE est à 1, ces lignes portent une adresse, dans l'autre cas elles portent une donnée. Comme représenté à la Fig 70, il suffit d'un tampon [Latch] de huit bits [par exemple, le circuit 8212 d'INTEL], commandé par la ligne ALE, pour démultiplexer le bus AD_i et obtenir un bus "classique" à 16 lignes d'adresses et 8 lignes de données.

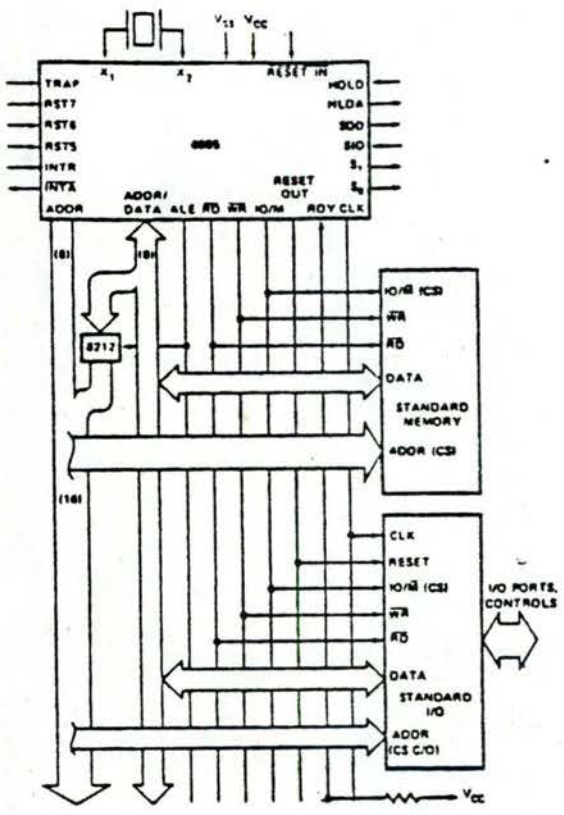
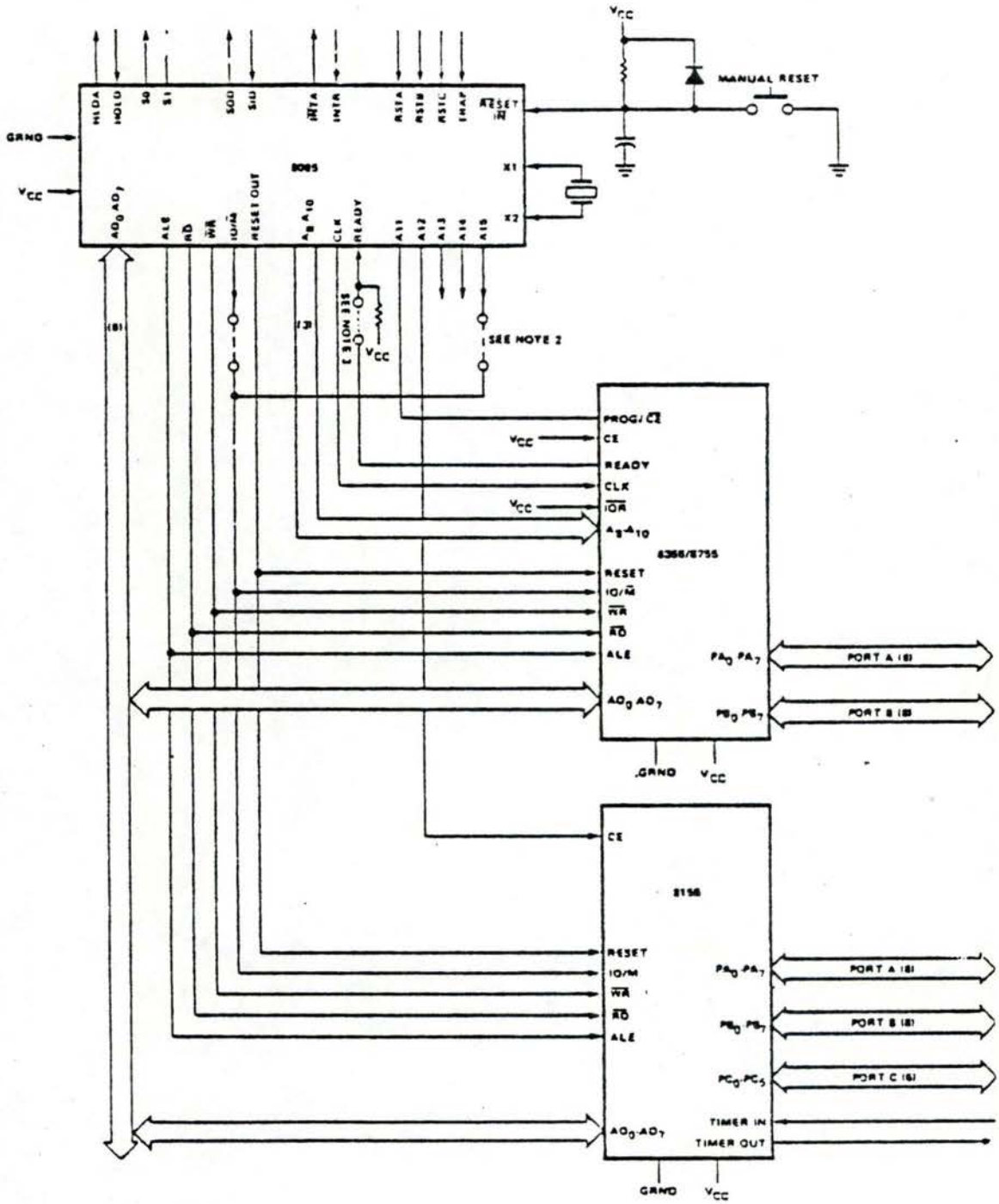


Fig 70



NOTE 1: TRAP, INTR, AND HOLD MUST BE GROUNDED IF THEY AREN'T USED.
NOTE 2: USE IO/M FOR STANDARD I/O MAPPING. USE A15 FOR MEMORY MAPPED I/O.
NOTE 3: CONNECTION IS NECESSARY ONLY IF ONE T_{WAIT} STATE IS DESIRED.

Fig 71

Il n'est pas toujours nécessaire de démultiplexer ce bus. En effet, le fabricant du 8085 produit une série de mémoires et d'interfaces d'entrée/sortie compatibles 8085, en ce sens qu'ils comportent un tampon interne pour le démultiplexage. Dans ces conditions, le bus multiplexé peut être connecté directement à ces circuits.

La figure 71 donne le schéma d'un ordinateur complet construit à l'aide de seulement trois ciruits intégrés : le microprocesseur 8085, le circuit 8355 qui comporte 16 lignes d'entrée/sortie et 2 K octets de mémoires mortes et le circuit 8155 qui comporte 22 lignes d'entrée/sortie, 256 octets de RAM et un compteur à 14 bascules. Ces circuits sont interconnectés à l'aide du bus multiplexé propre au microprocesseur 8085.

Les lignes d'adresses A_{11} et A_{12} sont utilisées pour effectuer une sélection linéaire des circuits 8355 et 8155. Les transferts d'informations sont contrôlés par les signaux des lignes $\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$ et READY. Ces signaux ont des fonctions analogues à celles pour le bus du 8080. Les relations entre les signaux propres au 8080 et ceux du 8085 sont :

$$\overline{MEMR} = \overline{RD} + IO/\overline{M}$$
$$\overline{MEMW} = \overline{WR} + IO/\overline{M}$$
$$\overline{IOR} = \overline{RD} + \overline{IO/\overline{M}}$$
$$\overline{IOW} = \overline{WR} + \overline{IO/\overline{M}}$$
$$RDYIN = READY$$

Sur la figure 71 on voit que, au choix, les lignes IO/$\overline{M}$ des circuits 8355 et 8155 peuvent être connectés à la ligne IO/$\overline{M}$ ou à ligne A_{15} du 8085. Dans le premier cas, les transferts vers les ports d'entrée/sortie devront être commandés par les instructions IN et OUT. Dans le second cas, les ports d'entrée/sortie seront adressés comme des registres de la mémoire, avec des adresses dont le bit le plus significatif vaut 1.

5.3.2.1.2. Le bus arbitre.

A chaque instant, parmi toutes les unités connectées à un bus, une seule peut placer de l'information sur celui-ci.

L'arbitre de bus déterminera l'unité qui peut jouer ce rôle. Une unité qui a besoin du bus demandera sa mise à disponibilité, à l'arbitre (bus request). Celui-ci, en fonction des demandes et de certaines règles de priorité, attribuera, à un des demandeurs, la disponibilité du bus pour un certain intervalle de temps. Pendant cet intervalle de temps, l'unité en question peut disposer du bus pour effectuer un transfert de données avec une des autres unités connectées au bus.

L'unité qui a le contrôle du bus s'appelle alors le maître, l'unité avec laquelle ce maître échange des informations s'appelle l'esclave (Fig 73). Dans certains cas, le maître peut autoriser des échanges directs d'informations entre deux esclaves (Fig 74).

Remarquons que souvent, l'arbitre de bus n'existe pas comme unité distincte. C'est le cas, par exemple, quand parmi les unités connectées au bus il n'en existe qu'une seule capable de jouer le rôle de maître et quand la fonction d'arbitre du bus est distribuée dans la circuiterie des différentes unités connectées.

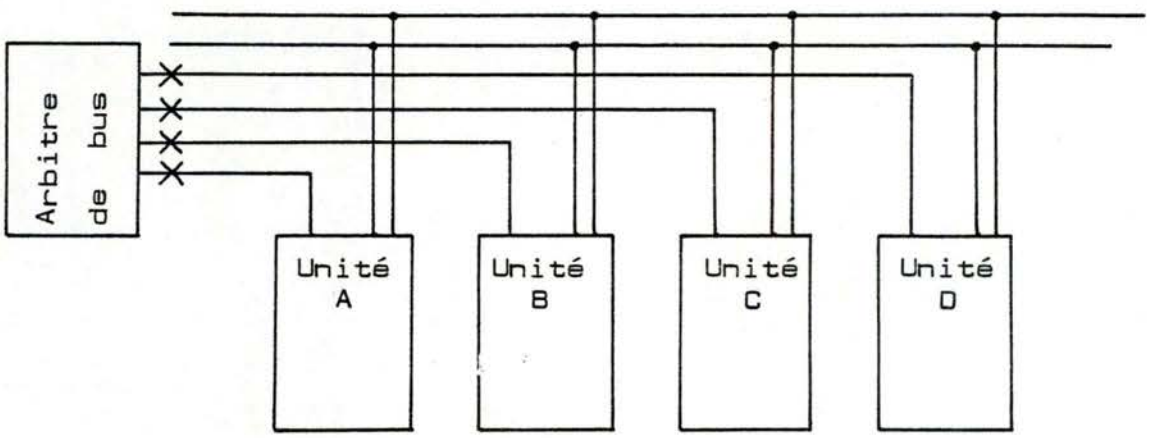


Fig 72

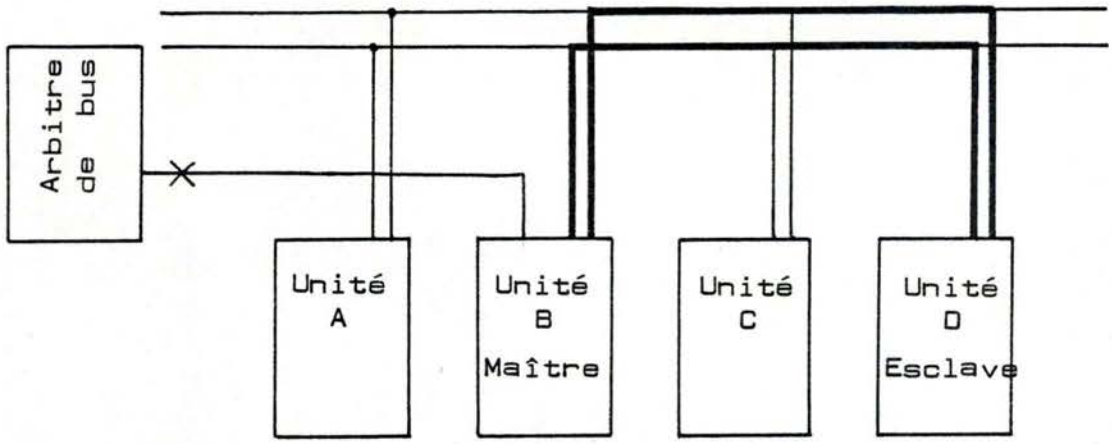


Fig 73

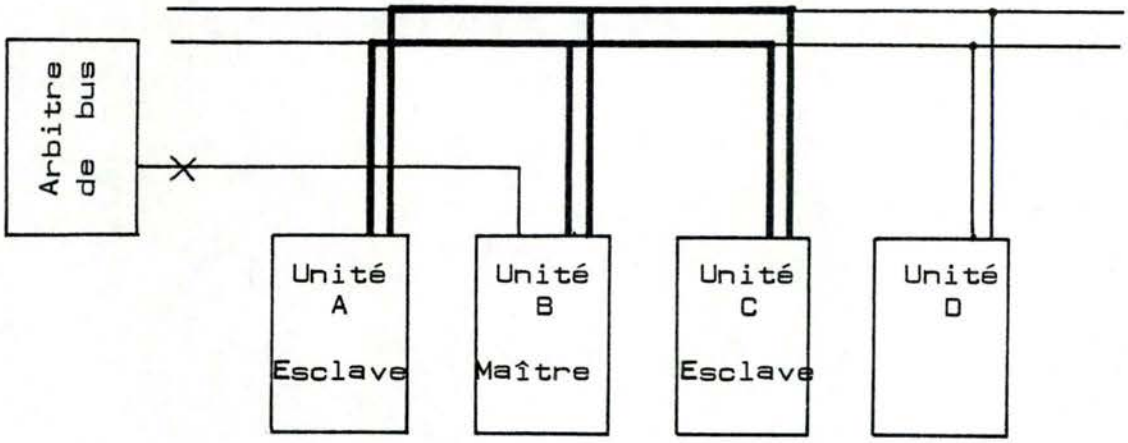


Fig 74

5.3.2.1.3. Les deux types de bus.

On peut classer les bus en fonction des diagrammes de temps qui indiquent quel est leur conditionnement lors des transferts des données.

- Bus synchrone

Dans un bus synchrone, l'esclave dispose d'un laps de temps prédéterminé pour exécuter un ordre du maître. Par exemple lors d'un ordre de lecture, le maître place l'adresse de l'esclave sur le bus et le maître suppose qu'au bout d'un certain temps l'esclave aura placé la donnée correspondante sur le bus.

Les bus synchrones conviennent parfaitement à la communication entre une unité centrale et une mémoire statique. En effet, le temps

d'accès des mémoires statiques est constant et connu. La figure 75 illustre une opération de lecture sur un bus synchrone. L'utilisation des mémoires dynamiques, indispensables lors des opérations de rafraîchissement, pose des problèmes avec un bus synchrone.

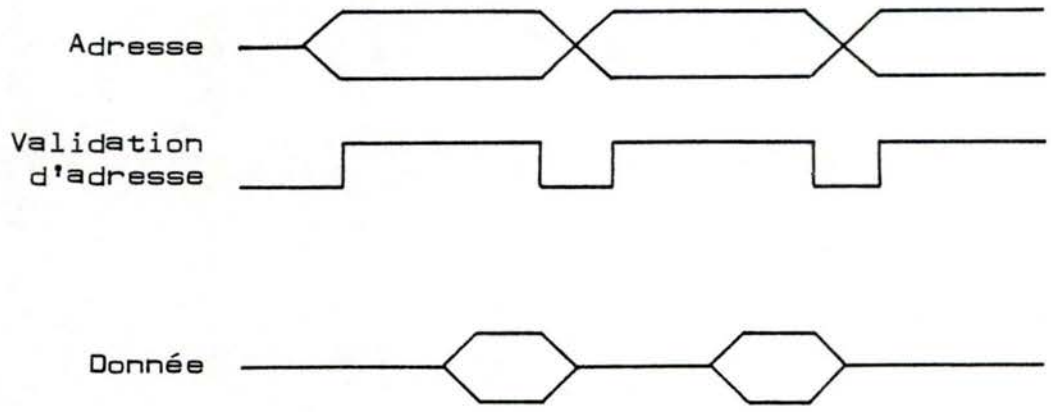


Fig 75

- Bus asynchrone (avec transfert indivisible).

Dans les bus asynchrones à transferts indivisibles, l'esclave informera le maître de la fin de l'exécution de l'ordre de transfert. Le bus reste alors réservé au transfert en cours, aussi longtemps que l'esclave n'a pas informé le maître de ce transfert.

Un tel diagramme de temps peut, par exemple, être utilisé avec des mémoires dynamiques. Normalement l'unité centrale trouvera sur le bus, après un temps égal au temps d'accès de la mémoire, un signal qui l'informera sur la disponibilité, sur le bus des données, de l'information demandée. Quand la mémoire est en rafraîchissement, ce signal tardera à être émis, et obligera ainsi l'unité centrale à attendre la disponibilité de l'information. La figure 76 illustre une opération de lecture pour un bus asynchrone.

Si le bus est asynchrone avec transfert partiel, le maître enverra un ordre à son esclave et libèrera immédiatement le bus afin que celui-ci puisse être utilisé pour d'autres transferts. Quand l'esclave a terminé sa tâche, il demandera le bus et jouera lui-même le rôle de maître pour renvoyer au premier maître le résultat de l'opération demandée.

Chacuns de ces transferts peuvent s'effectuer en mode synchrone ou asynchrone. Des transferts partiels seront utilisés dans des grands systèmes où différents flux d'informations doivent coexister, sur le bus, quasi simultanément. Dans de tels systèmes, le bus sera dimensionné de façon telle qu'un transfert individuel soit beaucoup plus court que, par exemple, le temps d'accès des mémoires. La figure 77 illustre un transfert de données de C à B demandé par B lors de l'échantillonnage temporel du bus.

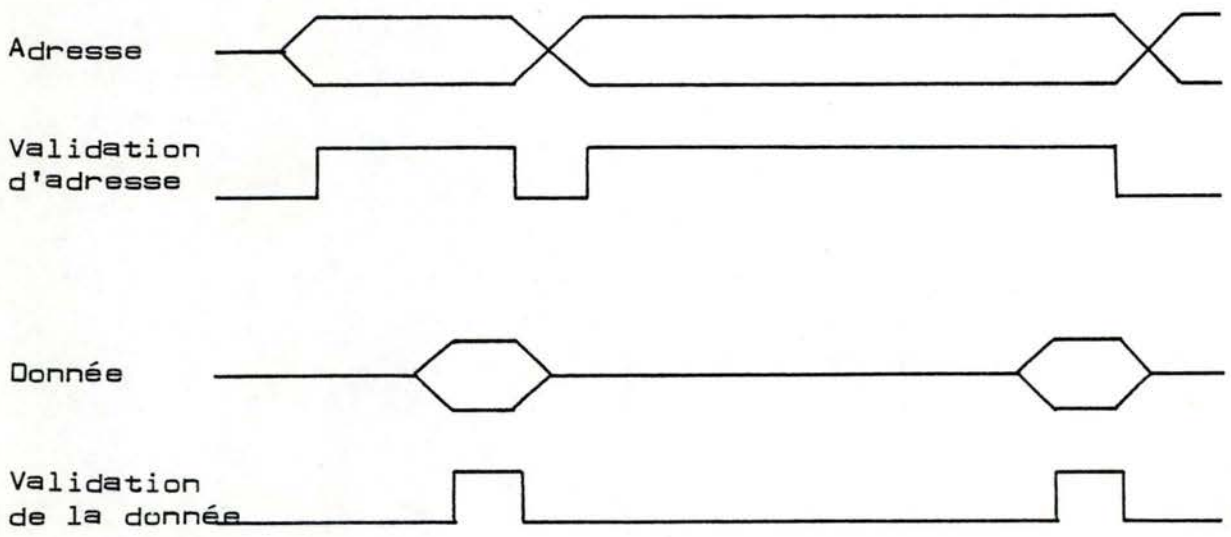


Fig 76

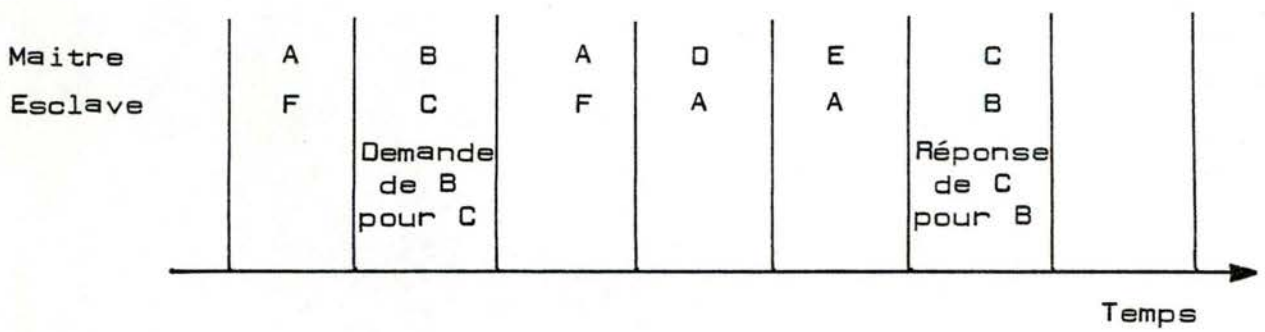


Fig 77

5.3.3. Bus multiple ou à plusieurs jeux de bus.

Lorsqu'on a affaire à un système à plusieurs microprocesseurs, l'architecture figurée en 78 présente un inconvénient majeur. En effet, comme un processeur doit utiliser le bus durant une bonne partie de son activité, son fonctionnement est considérablement ralenti par celui des autres microprocesseurs.

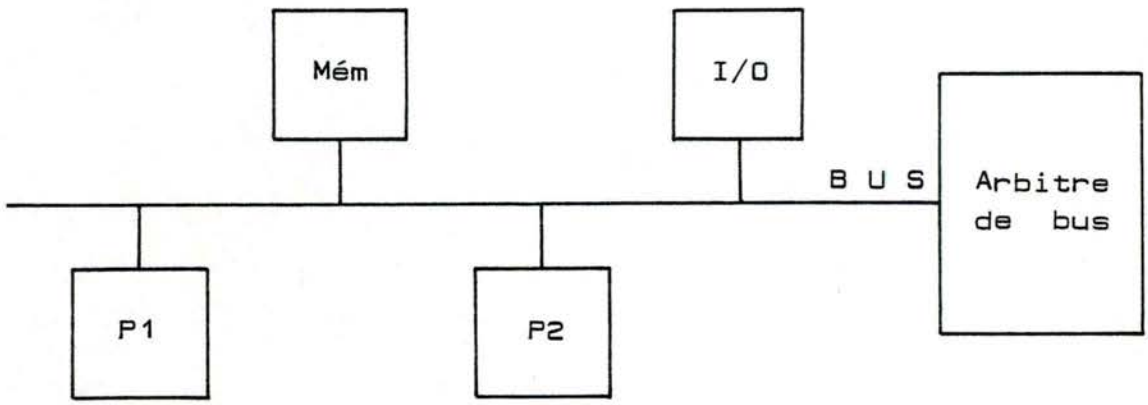


Fig 78

On constate sur la figure 79 que chaque processeur dispose de son propre bus, de sa propre mémoire et éventuellement de ses propres interfaces d'entrée/sortie. Un processeur fait uniquement appel au bus commun au moment où il veut échanger des informations avec une des ressources communes, par exemple, la mémoire commune (MC) ou les interfaces communs d'entrée/sortie. Ces ressources communes sont uniquement utilisées lorsque les processeurs doivent communiquer entre eux ou avec le monde extérieur. Dans ces conditions, plusieurs processeurs peuvent être connectés sur bus commun sans provoquer le ralentissement important de leur fonctionnement respectif.

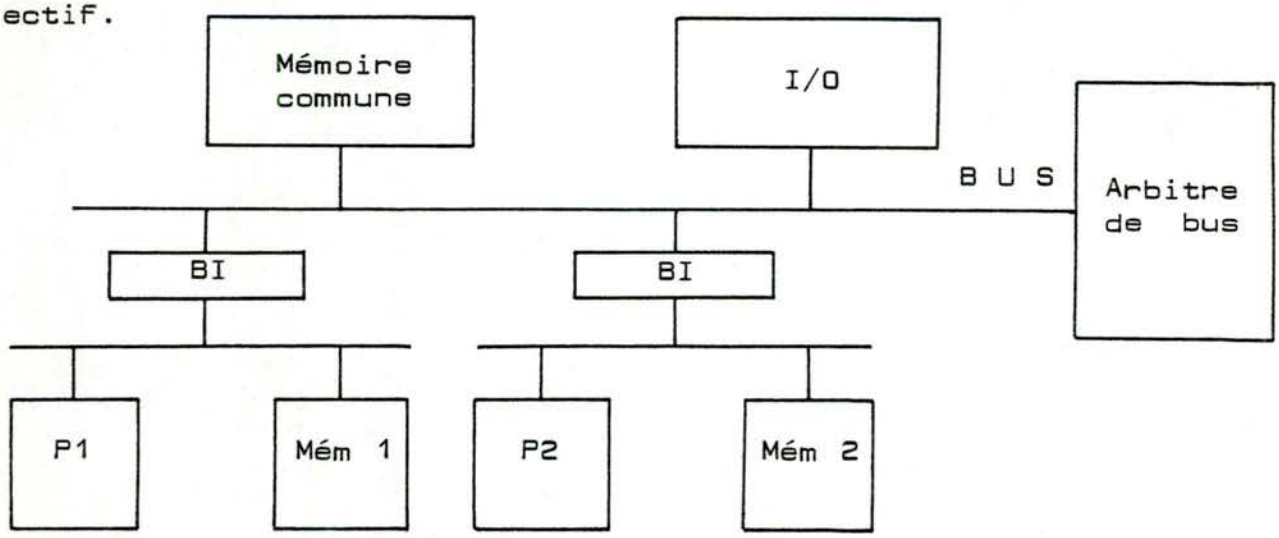


Fig 79

Entre les bus propres à chaque processeur et le bus commun, on trouve les interfaces de bus. Ces interfaces sont essentiellement constitués de tampons qui interconnectent ou non les bus auxquels ils sont reliés. Dans les multiprocesseurs les plus simples, la mémoire et les interfaces d'entrée/sortie communs ont une même adresse dans l'espace d'adresse de chaque processeur. Quand un des processeurs doit échanger des informations avec une ressource commune, l'interface de bus reconnaît l'adresse correspondante et demande à l'arbitre de bus l'accès au bus commun. En attendant l'accès proprement dit, le processeur est placé dans un état d'attente. Quand l'accès au bus est autorisé, celui-ci est connecté, via l'interface de bus, au processeur qui a fait la demande d'accès, de manière à ce que le transfert puisse s'effectuer.

Il est certain que l'arbitrage est plus compliqué pour un système multiprocesseurs qu'il ne l'est pour un système à accès direct à la mémoire. De plus, les circuits spéciaux, prévus dans chaque processeur pour permettre les transferts en accès direct à la mémoire ne sont pas suffisants dans un environnement multiprocesseurs. Ces circuits doivent être complétés par de la logique externe, de manière à régler correctement les priorités et à contrôler les interfaces de bus d'une façon appropriée. Nous indiquerons au paragraphe suivant la façon dont INTEL a résolu les problèmes posés par l'architecture de bus hiérarchisés de la Fig 80.

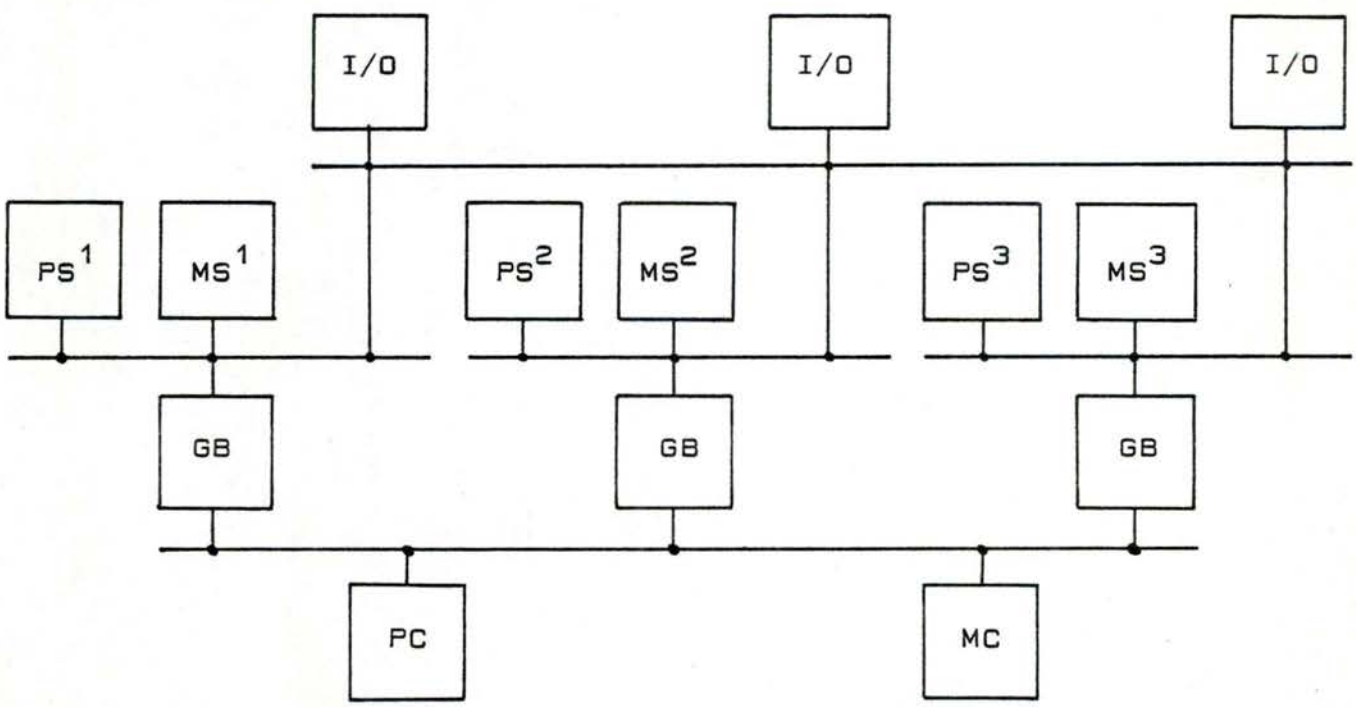


Fig 80

5.3.3.1. Bus hiérarchisés à signalisation électrique.

L'architecture simplifiée est illustrée à la figure 81. Il existe un microprocesseur principal que nous appellerons μP_p et plusieurs microprocesseurs secondaires que nous appellerons μP_s.

Dans ce mode, le 8089 partage le bus du système avec le microprocesseur central (Fig 81). L'accès à ce bus est contrôlé par des circuits de gestion de bus (8289). Le processeur d'entrée/sortie 8089 est, cependant, physiquement autonome et dispose de son propre bus ou bus local, qu'il peut partager avec un autre 8089. Plusieurs 8089 peuvent être connectés de cette manière, seuls ou par groupe de deux, au bus du système et leur bus n'a pas besoin d'opérer sur la même longueur de bus que le microprocesseur central. Le 8089 peut ainsi jouer le rôle d'interface entre des périphériques travaillant sur des mots de huit bits et le 8086. En mode autonome, le 8089 dispose de sa propre mémoire, qu'il partage éventuellement avec un autre 8089, et de ses propres coupleurs de périphériques. Les communications avec le microprocesseur central se réduisent à l'initialisation du 8089 et à la transmission des commandes. Mis à part ces deux types d'activités, pendant lesquelles les deux microprocesseurs doivent dialoguer, ces derniers ont un fonctionnement complètement autonome. En particulier, le 8089 peut aller chercher les instructions qu'il doit exécuter dans sa propre mémoire, sans avoir accès au bus du système. L'avantage d'une configuration entièrement décentralisée est de permettre des opérations en parallèle entre le microprocesseur central et le 8089. Cet avantage est contrebalancé par le prix des circuits additionnels que l'on doit ajouter pour assurer la gestion du bus système.

Les mécanismes de demande de tâche par un μP_s ou d'une sous-traitance d'une tâche (entrée/sortie) par le μP_p à un μP_s ainsi que l'acquisition des bus commun ou principal et particulier ou secondaire par l'un ou l'autre μP sont basés sur les techniques suivantes :

1) La prise de main par un μP_s vis-à-vis d'un autre μP_s s'opère au moyen d'un échange RQ/GT, soit une demande (impulsion RQ) par l'appelant et une acceptation éventuelle (impulsion GT) de l'appelé. Eventuellement, un bloc d'arbitrage est nécessaire pour mémoriser les occupations et les priorités (Fig 81).

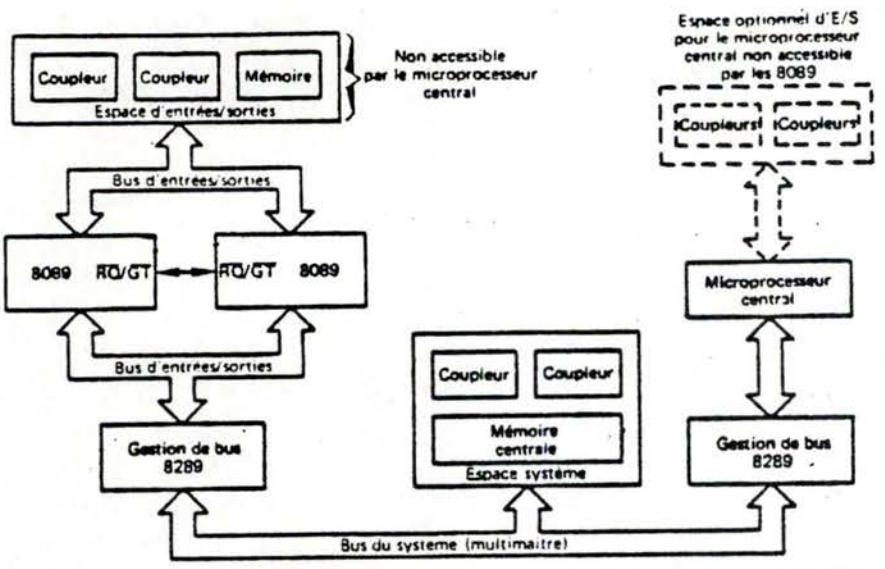


Fig 81

2] Le partage du bus commun pour assurer la liaison entre le μP_p et l'un ou l'autre μP_s se fait en deux temps. En premier lieu, au début de la phase d'initialisation le μP_p et le μP_s demandé échangent des messages par l'intermédiaire d'une boîte à lettres, située en mémoire centrale [Fig 82]. Les seules communications qui se font directement entre les deux processeurs sont d'une part, les demandes d'attention formulées par le microprocesseur central à l'adresse du processeur d'entrée/sortie; d'autre part, les demandes d'interruption du microprocesseur central par l'un des canaux du 8089.

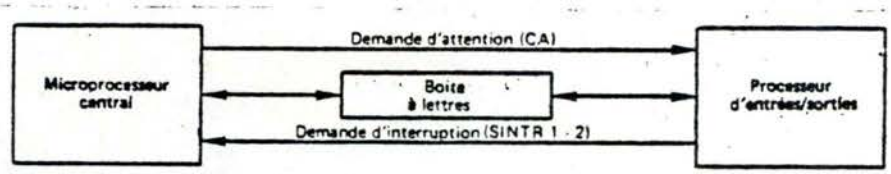


Fig 82

Dans un second temps, le partage du bus commun est conditionné par des priorités parallèles, sérielles ou tournantes lors de l'apparition du signal $\overline{BREQ}$ [Fig 83].
Premièrement, la gestion de priorités parallèles : le 8289 dispose d'une broche de demande d'accès au bus $\overline{BREQ}$ qui doit être, dans ce cas, connectée à un encodeur de priorité. Dans un environnement multimicroprocesseurs, cet encodeur génère en sortie l'adresse binaire

correspondant à la broche $\overline{\text{BERQ}}$, qui a demandé l'accès au bus, et dont le niveau de priorité est le plus élevé. Après décodage dans le circuit spécialisé 74138, cette adresse permet de sélectionner la ligne BPRN de validation du 8289, dont le niveau de priorité est le plus élevé. Le circuit de gestion de bus, ainsi validé, autorise le microprocesseur qui lui est associé à accéder au bus du système, dès que le bus est disponible. Il est important de noter que lorsqu'un microprocesseur obtient l'autorisation d'accès au bus-système, il n'en dispose pas immédiatement. Il doit attendre que le microprocesseur, en possession du bus, ait terminé son cycle de transport. Cette fin de cycle est signalée sur la broche $\overline{\text{BUSY}}$ du microprocesseur concerné.

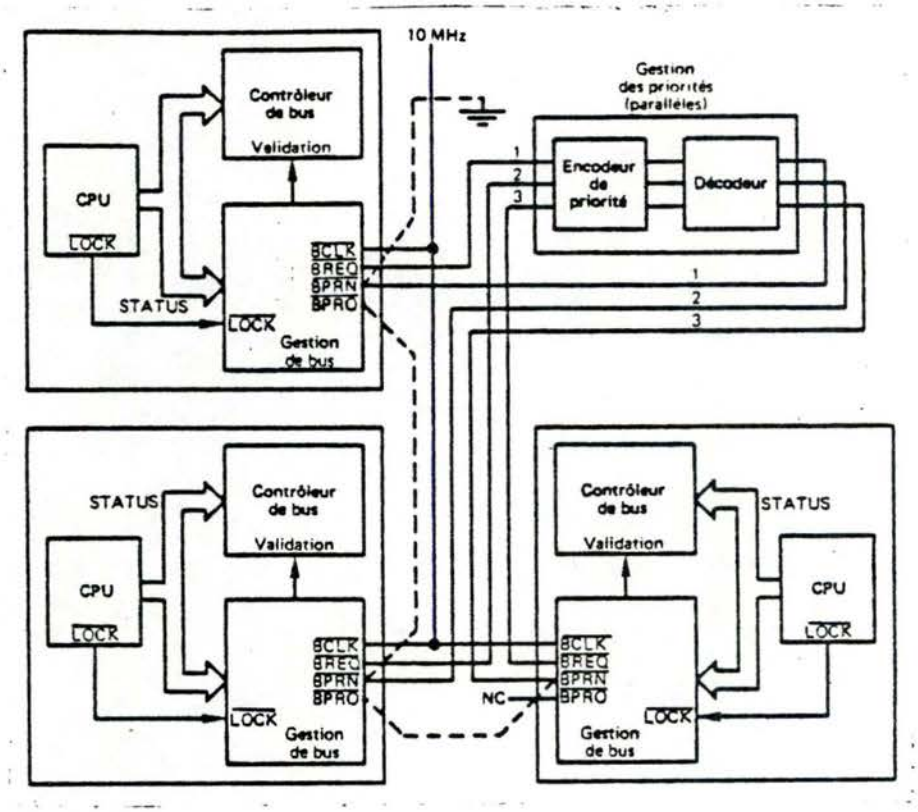


Fig 83

Deuxièmement, la gestion de priorités sérielles élimine le besoin d'un encodeur/décodeur de priorités, et s'effectue par la connexion en série des circuits de gestion. La broche $\overline{\text{BPRO}}$ du circuit le plus prioritaire est connectée à la broche $\overline{\text{BPRN}}$ du circuit, dont le niveau de priorité est juste en dessous, et ainsi de suite, jusqu'au circuit dont le niveau de priorité est le plus bas. La broche $\overline{\text{BPRN}}$ du circuit le plus prioritaire est reliée à la masse, ce qui indique le niveau de priorité le plus élevé pour l'accès au bus.

Lorsque le circuit le plus prioritaire n'a pas besoin du bus, il envoie au circuit suivant un niveau bas sur la broche $\overline{BPRN}$, et ainsi de suite. Le circuit qui désire avoir accès au bus génère un niveau haut sur la broche $\overline{BPRN}$, inhibant ainsi l'accès au bus par les circuits suivants dans la chaîne.
Troisièmement, la gestion de priorités tournantes s'effectue de la même manière que la gestion des priorités parallèles, en dehors du fait que le niveau de priorité de chaque circuit est assigné dynamiquement. L'encodeur de priorité est remplacé par un circuit plus complexe, qui affecte les priorités de manière rotative après chaque accès au bus par un circuit [le circuit qui vient d'accéder au bus a ensuite le niveau de priorité le plus bas].

3] Le choix pour un μP_s d'occuper le bus commun ou le bus-système pour assurer des fonctions respectivement entrée/sortie et en liaison avec le μP_p [Fig 84].

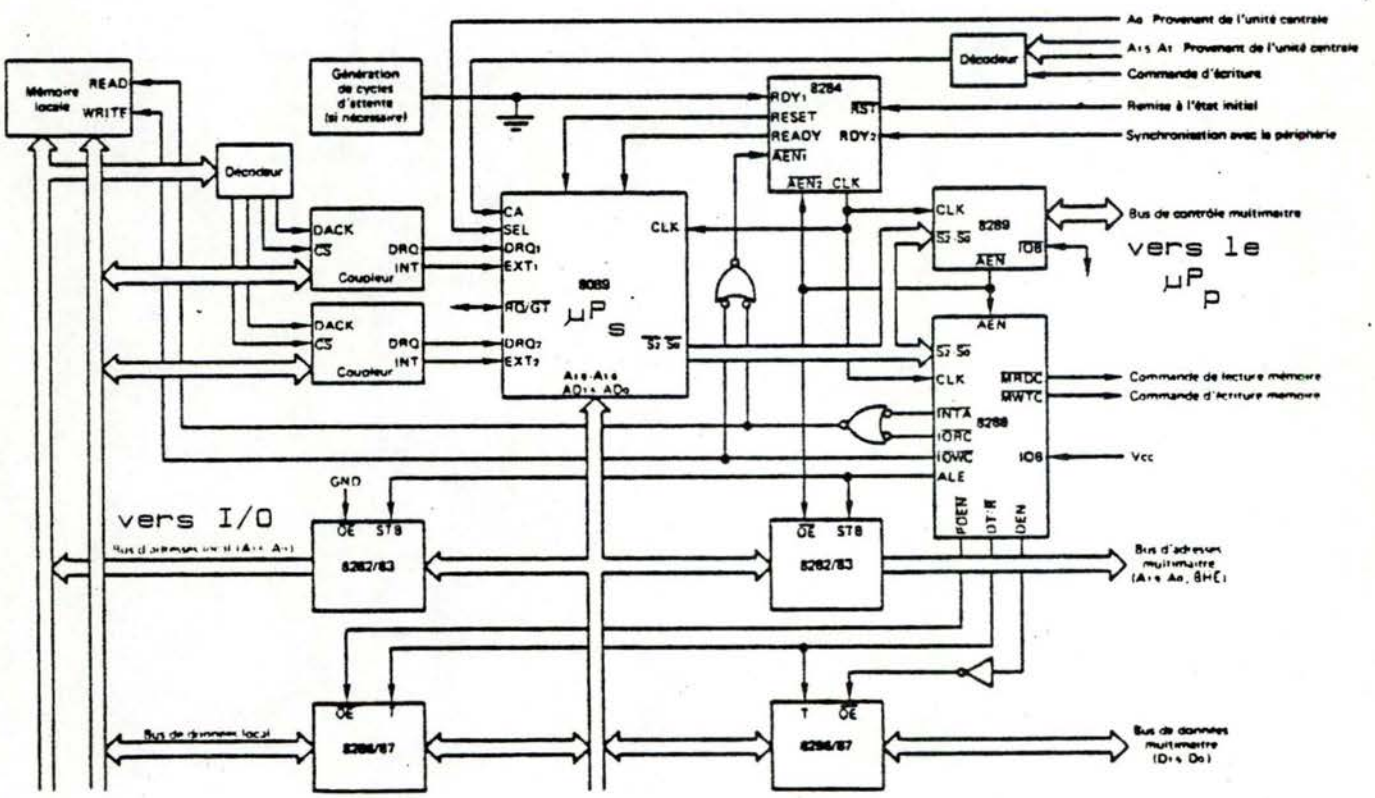


Fig 84

C'est par le décodage de l'instruction que la combinaison binaire des broches $\overline{S}_0$ $\overline{S}_1$ $\overline{S}_2$ (Tab 85) transcodée en commande DE ou $\overline{OE}$ ouvre les portes des bus adresses et de données soit vers le μP_p, soit vers les interfaces d'entrée/sortie via les blocs logiques 8288 et 8282/83 pour une configuration de microprocesseurs INTEL. Le microprocesseur principal est l'un des microprocesseurs classiques 8085, 8086, ... tandis que le microprocesseur secondaire qui a été choisi par INTEL est le 8089.

Etat des broches			Cycle de bus indiqué	Sortie d'un contrôleur de bus 8288
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$		
0	0	0	Recherche d'une instruction dans l'espace d'E/S	$\overline{INTA}$
0	0	1	Lecture d'une donnée dans l'espace d'E/S	$\overline{IORC}$
0	1	0	Ecriture d'une donnée dans l'espace d'E/S	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Non utilisé	RIEN
1	0	0	Recherche d'une instruction dans la mémoire centrale	$\overline{MRDC}$
1	0	1	Lecture d'une donnée dans la mémoire centrale	$\overline{MRDC}$
1	1	0	Ecriture d'une donnée dans la mémoire centrale	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Non utilisé	RIEN

Tab 85

Notons également que le μP_p peut par I/OB ou SYSB accéder également à la mémoire et au μP_s associé (Fig 86). L'organisation externe et interne du 8089 est illustrée très schématiquement à la figure 87. On remarque que toutes les requêtes sont formulées explicitement par des signaux électriques. On y distingue :

- Les signaux CA et SINTR de et vers le μP_p ainsi que SEL.
- Les signaux de requêtes et d'accusé de réception $\overline{RQ}/\overline{GR}$ entre μP_s.
- Un signal de verrouillage $\overline{LOCK}$ empêchera le 8089 d'être interrompu par un autre μP_s.
- Les signaux binaires décodés vers les portes 8289 donnera accès soit au bus commun soit à celui des périphériques.
- Les pseudo-interruptions d'accusé de réception (DRQ) pour un fonctionnement en mode DMA ainsi que la borne EXT (fin de DMA lorsque BC = 0).

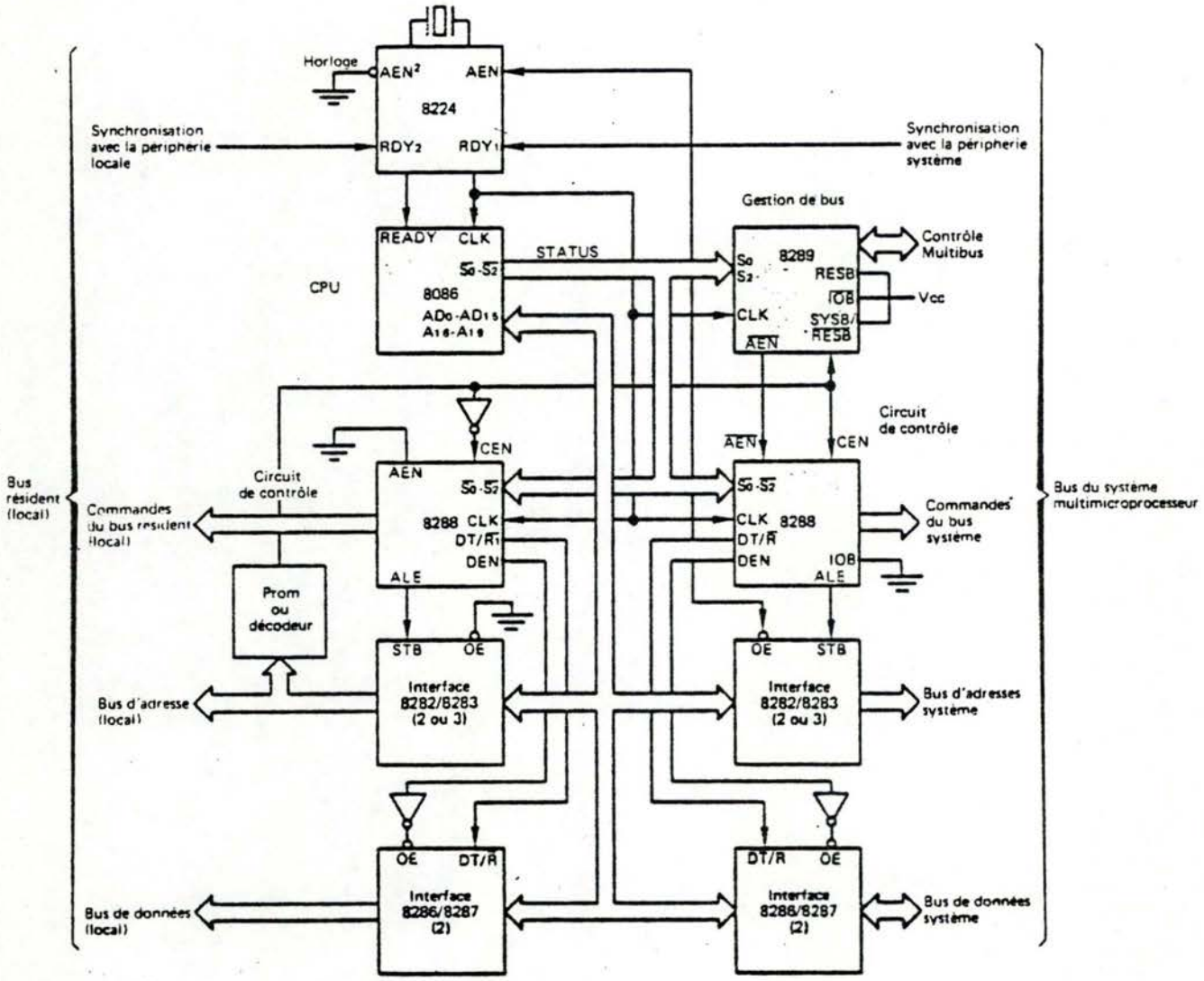

Horloge
AEN²
AEN
8224
RDY₂
RDY₁
Synchronisation avec la périphérie locale
Synchronisation avec la périphérie système
Gestion de bus
READY
CLK
CPU
8086
STATUS
S̄₀-S̄₂
AD₀-AD₁₅
A₁₆-A₁₉
8289
S₀
S₂
RESB
IOB
CLK
SYSB/RESB
AEN
Contrôle Multibus
Vcc
Circuit de contrôle
CEN
Bus résident (local)
Commandes du bus résident (local)
8288
CLK
DT/R̄
DEN
ALE
Commandes du bus système
Bus du système multimicroprocesseur
Prom ou décodeur
STB
OE
Interface 8282/8283 (2 ou 3)
Bus d'adresse (local)
Bus d'adresses système
DT/R
Interface 8286/8287 (2)
Bus de données (local)
Bus de données système

Fig 86

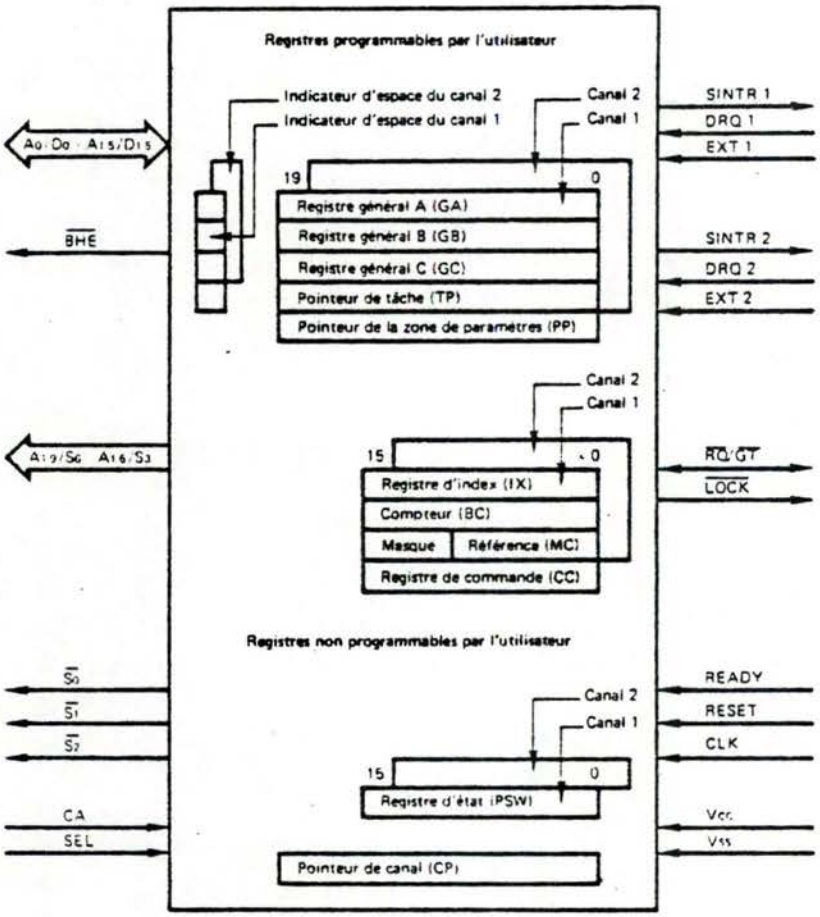

Registres programmables par l'utilisateur
Indicateur d'espace du canal 2
Indicateur d'espace du canal 1
Canal 2
Canal 1
SINTR 1
DRQ 1
EXT 1
A0/D0 - A15/D15
19
0
Registre général A (GA)
Registre général B (GB)
Registre général C (GC)
Pointeur de tâche (TP)
Pointeur de la zone de paramètres (PP)
BHE
SINTR 2
DRQ 2
EXT 2
Canal 2
Canal 1
A19/S6 A16/S3
15
0
RQ/GT
LOCK
Registre d'index (IX)
Compteur (BC)
Masque
Référence (MC)
Registre de commande (CC)
Registres non programmables par l'utilisateur
S0
S1
S2
READY
RESET
CLK
Canal 2
Canal 1
15
0
Registre d'état (PSW)
CA
SEL
Vcc
Vss
Pointeur de canal (CP)

Fig 87

5.3.3.2. Bus hiérarchisés à messages.

Des architectures peuvent être constituées sur la base de modules empilés dont le prototype est représenté à la figure 80. Ces architectures pyramidales à plusieurs couches de microprocesseurs à fonctions diverses (1 jeu de bus entre 2 couches) sont destinées à des machines spécialisées dans la commutation de données protégées par protocoles; cette commutation pouvant se faire par circuits, par messages, par paquets, ...

Cette architecture pyramidale est hiérarchisée avec envoi de messages d'étage pour acheminer et faire exécuter des commandes de traitements.

La gestion des messages et leur génération peut être organisée et créée par un miniordinateur situé à la tête de la pyramide. Toutefois, ce dernier peut sous-traiter cette tâche à des processeurs maîtres des couches inférieures.

Comme nous venons de le voir, des microprocesseurs de nature et de fonctions diverses peuvent également être disposés sur un jeu unique de trois types de bus comportant les données, les adresses et les contrôles.

Les variétés et l'importance des échanges et, de ce fait, des acquisitions et des libérations des bus ne se satisfont plus, en général, des requêtes et des accusés à l'aide de signaux électriques.

La liaison entre les microprocesseurs demandeurs fait appel à des envois de messages de communications et de commandes. La signalisation peut parfois être complètement abolie. Nous citerons deux exemples à ce sujet :
1) Un microprocesseur maître doit entrer en communication avec plusieurs microprocesseurs secondaires de type terminal c'est-à-dire connectés à des périphériques et/ou à des lignes (Fig 88). L'acquisition de données par les microprocesseurs secondaires sont transférées vers le microprocesseur principal pour y être traitées et inversément des résultats obtenus au microprocesseur principal doivent être renvoyés au microprocesseur secondaire correspondant.

Ces opérations d' échanges sont nombreuses et répétitives et seront gérées par le microprocesseur principal en mode polling. C'est ainsi que le microprocesseur maître enverra périodiquement des messages vers chacun des microprocesseurs secondaires; chacun des messages comportant en plus de l'adresse de ce dernier une commande WRITE, READ ou WRITE CONTROL (demande d' un état, bouclage de ligne, mise hors service, changement de paramètres, ...). Cette procédure réitérative et systématique n' imposera pas une réquisition préalable du bus.

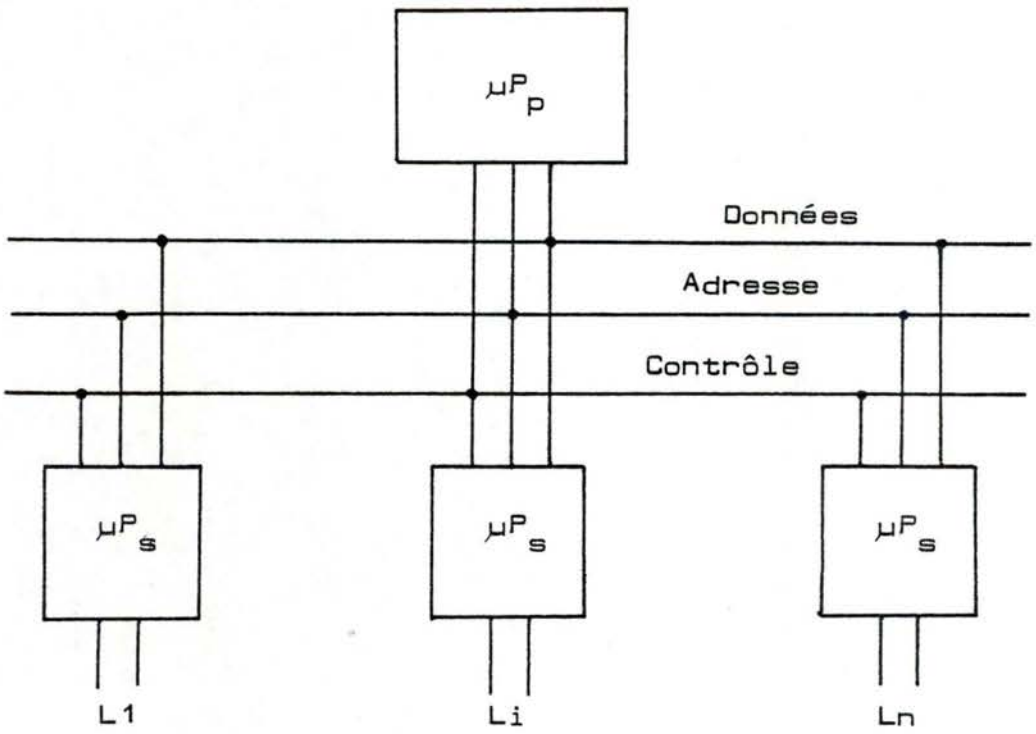


Fig 88

2)Sur le jeu de bus toujours décomposés en bus de données, d' adresses ou de contrôle on trouve deux catégories de microprocesseurs, le groupe P_1 à P_m et le groupe Q_1 à Q_m. En principe, un Q_i fournit les données à un P_j choisi qui avec les données acquises les traite pour les renvoyer à un autre Q_k qui exploitera les résultats.
Les processeurs P sont choisis en fonction des disponibilités tandis que les processeurs Q sont spécialisés en fonction de l' opération leur réclamée. Un premier groupe de Q exécutera les opérations simples et répétitives (commutation) et un second, les opérations complexes et peu fréquentes (appels, libération). Les opérations étant nombreuses et variées d' une part et d' autre part, l' optimalisation

des bus étant capitale, il faudra pour satisfaire ce second point un "BUS ARBITER " pour traiter les requêtes et les libérations selon les priorités ou la chronologie. Pour permettre d' assurer au mieux le premier point, il sera nécessaire de procéder par envoi de messages qui pourront comporter plusieurs mots dont la structure sera: AD. SOURCE (pour réponse), AD. DESTINATION , TYPE D'OPERATION (W,R,WC), PARAMETRES. Exemple: P_i désirera lire dans la mémoire de R_j à l' adresse 0R1F.

Une modification de cette architecture améliorant considérablement la rapidité mais aussi plus complexe, donc plus onéreuse, consiste à utiliser deux bus, l' un pour l' adresse de la source et l' autre de la destination. (Sélection dans un sens ou dans l' autre plus rapide, allègement des messages).

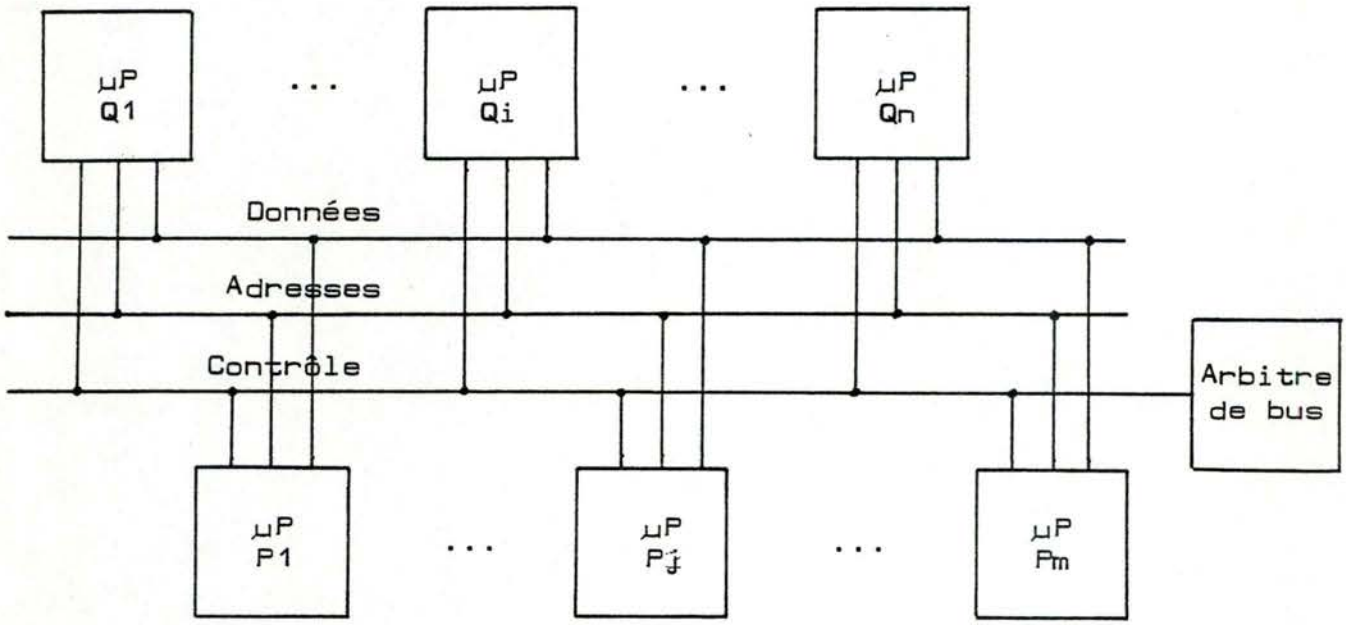


Fig 89

C O N C L U S I O N.

Nous avons décrit l'application d'un système à microprocesseur pour une centrale solaire isolée ou en appoint à un réseau alternatif. L'évolution du produit est facile avec un microprocesseur si bien que d'autres fonctions secondaires peuvent être développées. Imaginons par exemple une fonction de poursuite du soleil avec des panneaux solaires orientables. Encore faut-il en peser la nécessité : l'accroissement de rendement en vaut-il la peine en contre-partie d'une mécanique d'orientation moins fiable et plus sensible aux pannes que tout le système de régulation?

A quel avenir est appelée une telle installation solaire? L'approvisionnement en énergie électrique dépend de deux facteurs : l'état de la démographie et le niveau d'industrialisation ou niveau de vie matériel atteint. La production d'énergie électrique fait appel à des sources d'énergie, telles les combustibles fossiles, l'hydraulique, le nucléaire... Nos régions ont l'avantage sinon de trouver sur place des énergies primaires, de les acheminer facilement. C'est ainsi que nous sommes approvisionnés en électricité par des centrales thermo-électriques à combustible solide (houille), à combustible gazeux (gaz naturel), à combustible liquide (pétrole); des centrales hydro-électrique à débit limité et enfin des centrales nucléaires. Ces dernières constitueront probablement la source d'avenir pour notre production massive d'électricité. A l'inverse, tous les villages et toutes les villes des régions désertiques du monde, dépourvues d'énergie primaire et difficilement approvisionnables, tant du point de vue technologique que du point de vue financier, offrent des perspectives de premier choix pour l'énergie solaire. C'est également le cas de tous les lieux isolés tels les balises de navigation aérienne ou maritime, les stations météorologiques automatiques, les stations de pompage d'eau dans toutes les

régions du monde où l'ensoleillement annuel dépasse un certain seuil.

Pour des installations de grosse puissance, nous avons proposé le fractionnement des points de production sur un réseau maillé de puissance. Dans ces conditions, un réseau informatique bouclé ou en étoile doit permettre à un centre de décision de contrôler, de surveiller et de commander l'ensemble de l'installation. Une telle configuration, rendue possible grâce à la souplesse des microprocesseurs et à leur possibilité de dialogue, a l'avantage d'optimiser l'ensemble du réseau de puissance par l'usage de procédures simples. Trois critères vont nous permettre d'orienter notre choix en ce qui concerne le type de réseau informatique. Considérons premièrement le critère "vitesse de transmission". La vitesse de transmission dans une ligne étant donnée, nous pouvons dire qu'un réseau en étoile dont les n stations périphériques sont saturées à un moment donné est n fois plus performant qu'un réseau en anneau. Plaçons-nous ensuite du point de vue économique. Le réseau en anneau qui totalise moins de longueur de ligne que le réseau en étoile offre le moindre coût de placement et d'entretien des lignes. Examinons enfin la fiabilité du réseau. Les conséquences d'une panne dans le réseau en étoile ne sont pas graves pour les stations car, elles n'éliminent que la ligne fautive. Par contre, une panne dans l'anneau est assez désastreuse car elle empêche toute communication. En résumé, le réseau en étoile l'emporte sur le plan de la vitesse de transmission et de la fiabilité. Un réseau en étoile semble donc bien adapté par sa performance de vitesse de transmission et son protocole simple à un système pour lequel la rapidité de réaction des différents postes est recquise. De plus, sa fiabilité est un avantage certain dans les lieux isolés ou mal désservis sur le plan de l'assistance technologique.

Une centrale solaire est donc facilement adaptable aux équipements modernes et, son application dépend de conditions économiques et d'une volonté politique de mise en oeuvre.

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

[1] P. AUDIBERT — Les énergies du soleil. [Editions du Seuil, 78]

[2] F. JUSTER — Les cellules solaires. [Editions Techniques et Scientifiques Françaises, 80]

[3] B. PASSAU — Alimentation stabilisée à modulation. [ACEC]

[4] B. PASSAU — Système électronique à microprocesseur de commande et régulation d'un moteur CA branché sur réseau solaire. [ACEC]

[5] D.J. DAVID — Choix d'un microprocesseur. [Micro Systèmes, Mars 79]

[6] R.G. CRAPPE — Microprocesseurs. [Faculté polytechnique de Mons]

[7] M. OUAKNINE, R. POUSSIN — Le HARDSOFT ou la pratique des microprocesseurs. [Editions Techniques et Scientifiques Françaises, 78]

[8] M. AUMIAUX — L'emploi des microprocesseurs. [Masson, 79]

[9] M. AUMIAUX — Les systèmes à microprocesseurs. [Masson, 80]

[10] P. MELUSSON — Le microprocesseur. [Edition Techniques et Scientifiques Françaises, 80]

[11] L. COUFFIGNAL — La cybernétique. [P.U.F. 63]

[12] C. MACCHI, J.F. GUILBERT — Téléinformatique [Dunod-informatique, 79]

[13] TRACEC 130. [ACEC]

[14] DOCUMENTATION INTEL — Catalogue 80.

[15] PUBLICATIONS PERIODIQUES : - Minis-Micros.
- Micro Systèmes.

A N N E X E 1.

FICHES TECHNIQUES DE GENERATEURS SOLAIRES.

Fiche technique

Generateurs Solaires TSG MQ 32/0

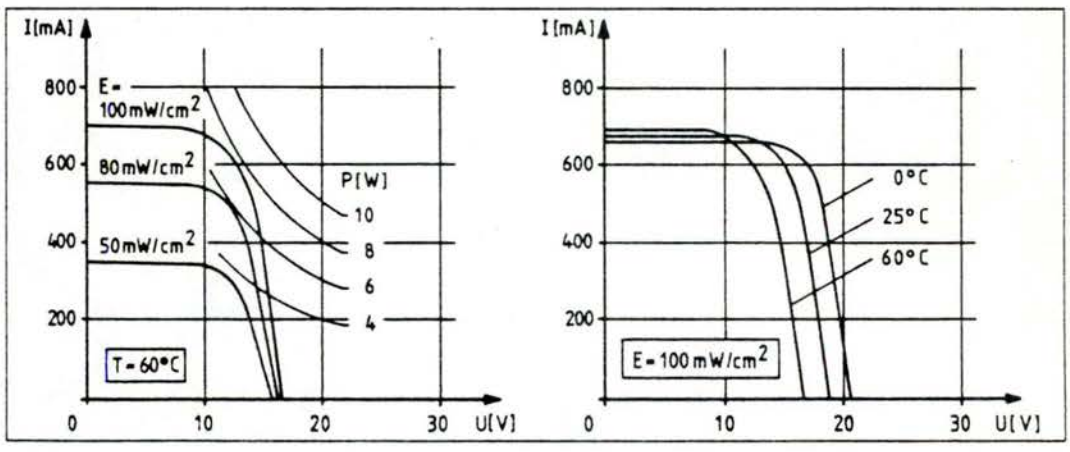
AEG

Données Electriques

Données électriques en fonction de la température de service.

Caractéristiques (AM1 – 100 mW/cm²)	Temperature de service		
	0 °C	25 °C	60 °C
Tension à vide (V)	20,5	18,7	16,3
Courant de court-circuit (mA)	679	685	694
Courant à la puissance maximale (mA)	624	630	639
Puissance maximale (W)	10,2	9,2	7,8

Courbes courant/tension



Données de tension, de courant et de puissance en fonction de la température			
La tension	augmente diminue	par 70,4 mV/°C	au-dessous de au-dessus de 25 °C
Le courant	augmente diminue	par 0,25 mA/°C	au-dessus de au-dessous de 25 °C
La puissance	augmente diminue	par 0,5 %/°C	au-dessous de au-dessus de 25 °C

Donnees Mechaniques
Eléments du module

Cellules Solaires
Matière de base:
Silicium monocristallin

Résistance spécifique
2 Ω cm

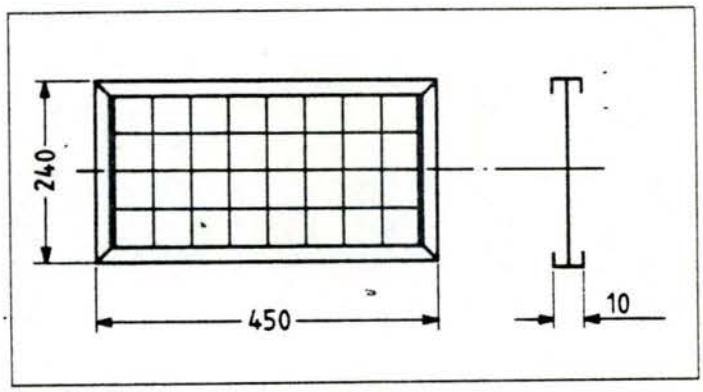
Conductivité
n/p

Dimensions
5 x 5 cm^2

Capsules
Verre

Cadre
Aluminium

Masse
1150 g



Encombrement

AEG-TELEFUNKEN
Nachrichten- und Verkehrstechnik
Fachbereich Flugwesen und Sondertechnik
Industriestraße 29 · D-2000 Wedel (Holst.)
Téléphone: (04103) 702-1 · Telex: 2189520

AEG-TELEFUNKEN

N42 2.7006/0478 F

Fiche technique

Generateurs Solaires TSG MQ 36/0

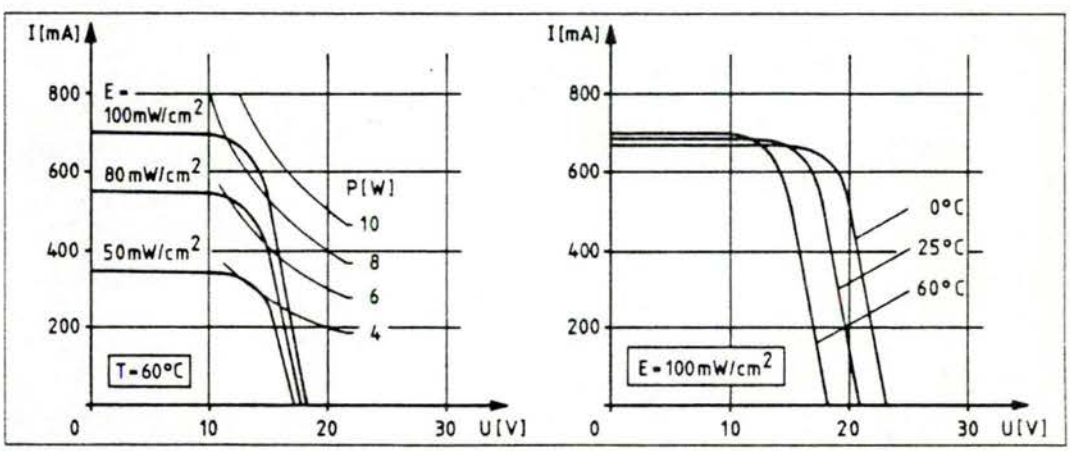
AEG

Données Electriques

Données électriques en fonction de la température de service.

Caractéristiques (AM1 – 100 mW/cm²)	Temperature de service		
	0 °C	25 °C	60 °C
Tension à vide (V)	23	21	18.2
Courant de court-circuit (mA)	679	685	694
Courant à la puissance maximale (mA)	624	630	639
Puissance maximale (W)	11,5	10,3	8,8

Courbes courant/tension



Données de tension, de courant et de puissance en fonction de la température			
La tension	augmente diminue	par 79,2 mV/° C	au-dessous de au-dessus de 25 °C
Le courant	augmente diminue	par 0,25 mA/°C	au-dessus de au-dessous de 25 °C
La puissance	augmente diminue	par 0,5 %/°C	au-dessous de au-dessus de 25 °C

Donnees Mechaniques
Eléments du module

Cellules Solaires
Matière de base:
Silicium monocristallin

Résistance spécifique
2 Ω cm

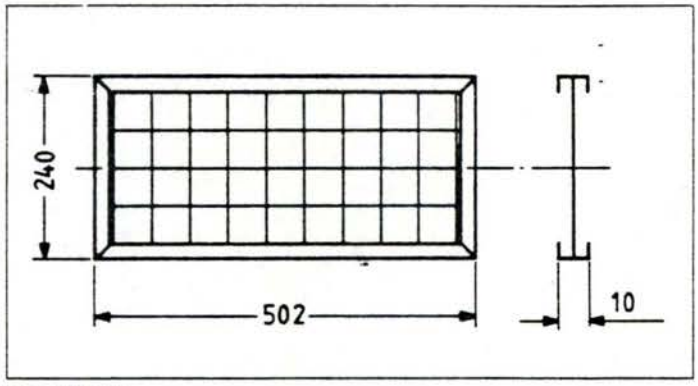
Conductivité
n/p

Dimensions
5 x 5 cm^2

Capsules
Verre

Cadre
Aluminium

Masse
1280 g



Encombrement

AEG-TELEFUNKEN
Nachrichten- und Verkehrstechnik
Fachbereich Flugwesen und Sondertechnik
Industriestraße 29 · D-2000 Wedel (Holst.)
Téléphone: (04103) 702-1 · Telex: 2189520

AEG-TELEFUNKEN

N42 2709/0478 F

Fiche technique

Generateurs Solaires TSG MQ 45/0

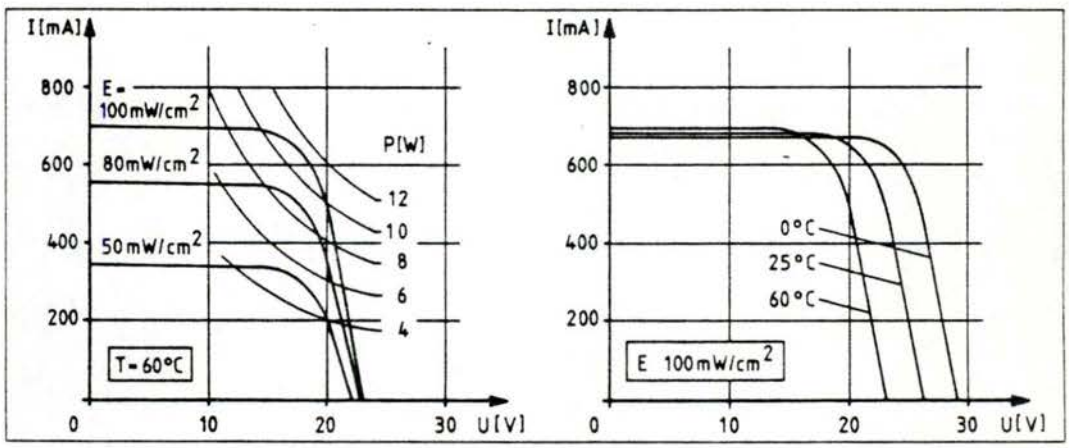
AEG

Données Electriques

Données électriques en fonction de la température de service.

Caractéristiques (AM1 – 100 mW/cm²)	Temperature de service		
	0 °C	25 °C	60 °C
Tension à vide (V)	28,9	26,3	23,0
Courant de court-circuit (mA)	679	685	694
Courant à la puissance maximale (mA)	624	630	639
Puissance maximale (W)	14,4	12,9	10,9

Courbes courant/tension



Données de tension, de courant et de puissance en fonction de la température			
La tension	augmente diminue	par 99 mV/° C	au-dessous de au-dessus de 25 °C
Le courant	augmente diminue	par 0,25 mA/°C	au-dessus de au-dessous de 25 °C
La puissance	augmente diminue	par 0,5 %/°C	au-dessous de au-dessus de 25 °C

Donnees Mechaniques

Elements du module

Cellules Solaires
Matière de base:
Silicium monocristallin

Résistance spécifique
2 Ω cm

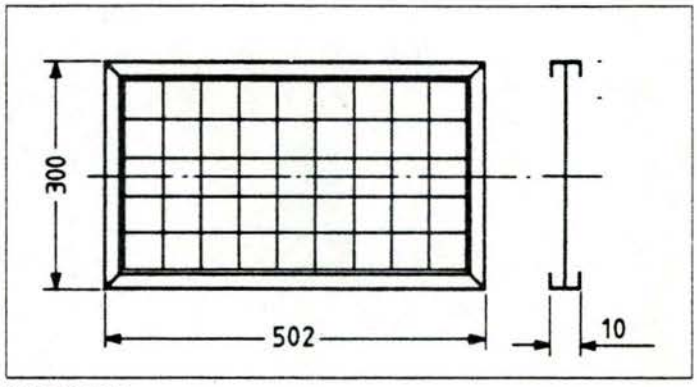
Conductivité
n/p

Dimensions
5 x 5 cm^2

Capsules
Verre

Cadre
Aluminium

Masse
1600 g



Encombrement

AEG-TELEFUNKEN
Nachrichten- und Verkehrstechnik
Fachbereich Flugwesen und Sondertechnik
Industriestraße 29 · D-2000 Wedel (Holst.)
Téléphone: (04103) 702-1 · Telex: 2189520

AEG-TELEFUNKEN

N42.2.7007/0478 F

A N N E X E 2.

FICHE TECHNIQUE DU PROCESSEUR 8080 A.

Pub. No. 420305226-001A

National Semiconductor

INS8080A 8-Bit N-Channel Microprocessor

general description

The INS8080A is an 8-bit microprocessor housed in a standard, 40-pin dual-in-line package. The chip, which is fabricated using N-channel silicon gate MOS technology, functions as the central processing unit (CPU) in National Semiconductor's N8080 microcomputer family.

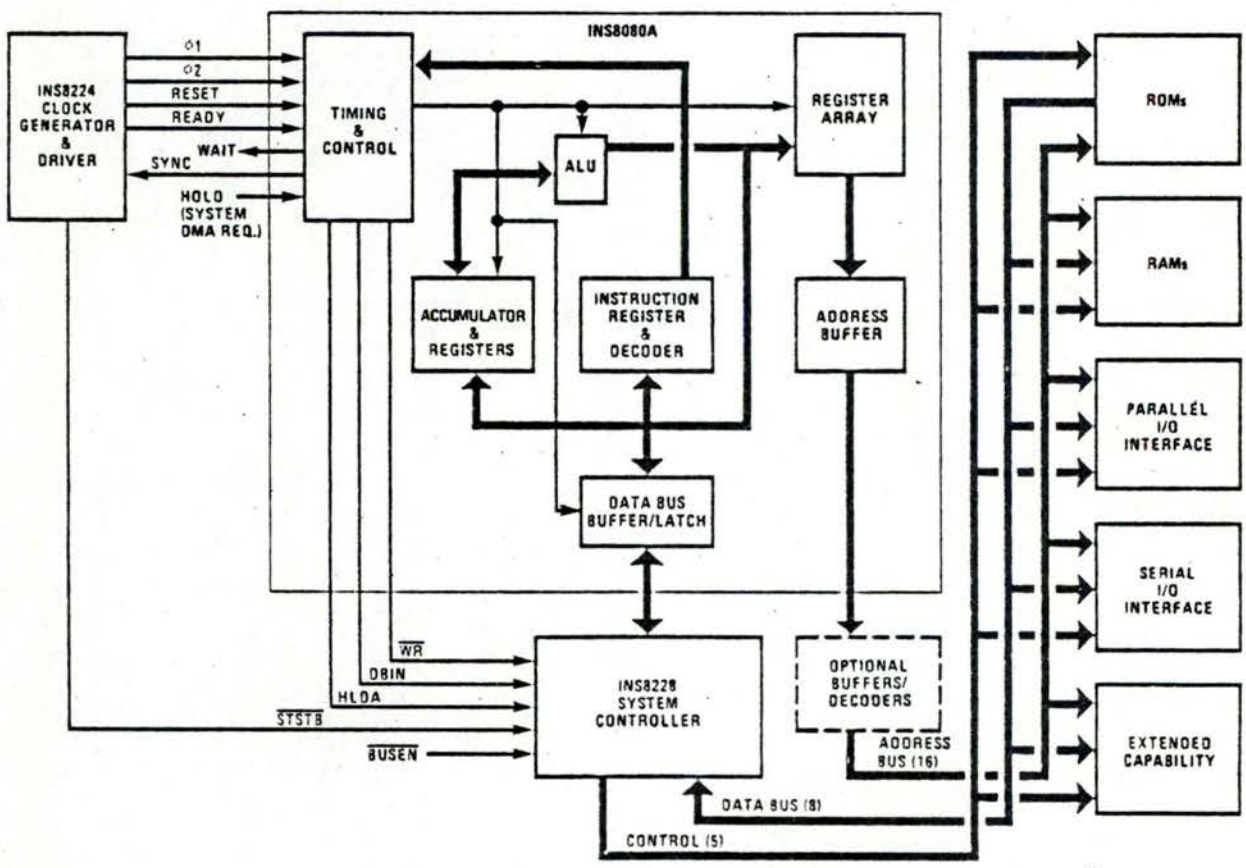
The INS8080A has a 16-bit address bus that is capable of addressing up to 65k bytes of memory and up to 256 input and 256 output devices. Data is routed to and from the INS8080A on a separate bidirectional 8-bit bus. This data bus is also TRI-STATE®, making direct memory addressing (DMA) and multiprocessing applications possible. The INS8080A directly provides signals to control the interface to memory and I/O ports. All buses, including control, are TTL compatible.

An asynchronous interrupt capability is included in the INS8080A to allow external signals to change the instruction sequence. The interrupting device may vector the program to a particular service routine location (or some other direct function) by specifying an interrupt instruction to be executed.

features

- 74 Instructions – Variable Length
- General Purpose Registers – Six plus an Accumulator
- Direct Addressing up to 65k Bytes
- Variable Length Stack Accessed by 16-bit Stack Pointer
- Addresses 256 Input and 256 Output Ports
- Provisions for Vectored Interrupts
- TRI-STATE® Bus for DMA and Multiprocessing Capability
- TRI-STATE TTL Drive Capabilities for Address and Data Buses
- Decimal Arithmetic Capability
- Multiple Addressing Modes
  - Direct
  - Register
  - Register Indirect
  - Immediate
- Direct Plug-in Replacement for Intel 8080A

N8080A microcomputer family block diagram



absolute maximum ratings

Temperature Under Bias . . . . . 0°C to +70°C
Storage Temperature . . . . . -65°C to +150°C
All Input or Output Voltages
with Respect to V_{BB} . . . . . -0.3V to +20V
V_{CC}, V_{DD} and V_{SS} with Respect to V_{BB} . . -0.3V to +20V
Power Dissipation 1.5W

Note: Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under dc electrical characteristics.

dc electrical characteristics

T_A = 0°C to +70°C, V_{DD} = +12V ± 5%, V_{CC} = +5V ± 5%, V_{BB} = -5V ± 5%, V_{SS} = 0V, unless otherwise noted.

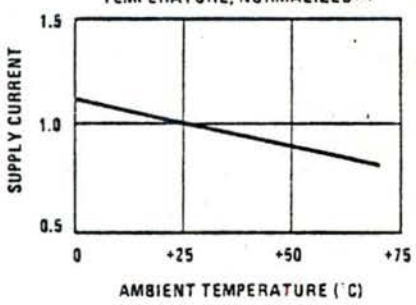
Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	$V_{SS}-1$		$V_{SS}+0.8$	V	
V_{IHC}	Clock Input High Voltage	9.0		$V_{DD}+1$	V	
V_{IL}	Input Low Voltage	$V_{SS}-1$		$V_{SS}+0.8$	V	
V_{IH}	Input High Voltage	3.3		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 1.9mA on all outputs,
V_{OH}	Output High Voltage	3.7			V	I_{OH} = 150μA.
$I_{DD\,(AV)}$	Avg. Power Supply Current (V_{DD})		40	70	mA	Operation
$I_{CC\,(AV)}$	Avg. Power Supply Current (V_{CC})		60	80	mA	t_{CY} = 0.48μs
$I_{BB\,(AV)}$	Avg. Power Supply Current (V_{BB})		0.01	1	mA	
I_{IL}	Input Leakage			±10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{CL}	Clock Leakage			±10	μA	$V_{SS} \leq V_{CLOCK} \leq V_{DD}$
I_{DL}[2]	Data Bus Leakage in Input Mode			-100	μA	$V_{SS} \leq V_{IN} \leq V_{SS} + 0.8V$
				-2.0	mA	$V_{SS} + 0.8V \leq V_{IN} \leq V_{CC}$
I_{FL}	Address and Data Bus Leakage			+10	μA	$V_{ADDR/DATA} = V_{CC}$
	During HOLD			-100	μA	$V_{ADDR/DATA} = V_{SS} + 0.45V$

capacitance

T_A = 25°C, V_{CC} = V_{DD} = V_{SS} = 0V, V_{BB} = -5V

Symbol	Parameter	Typ.	Max.	Unit	Test Condition
C_ϕ	Clock Capacitance	17	25	pF	f_c = 1MHz
C_{IN}	Input Capacitance	6	10	pF	Unmeasured Pins
C_{OUT}	Output Capacitance	10	20	pF	Returned to V_{SS}

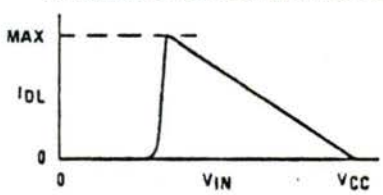
TYPICAL SUPPLY CURRENT VS. TEMPERATURE, NORMALIZED[3]



Notes:

1. The RESET signal must be active for a minimum of 3 clock cycles.
2. When DBIN is high and $V_{IN} > V_{IH}$ an internal active pullup will be switched onto the Data Bus.
3. $\Delta I_{supply} / \Delta T_A$ = -0.45%/°C.

DATA BUS CHARACTERISTIC DURING DBIN



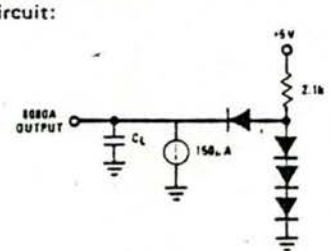
ac electrical characteristics

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CY} [3]	Clock Period	0.48	2.0	μs	
t_r, t_f	Clock Rise and Fall Time	0	50	ns	
$t_{\phi 1}$	ϕ_1 Pulse Width	60		ns	
$t_{\phi 2}$	ϕ_2 Pulse Width	220		ns	
t_{D1}	Delay ϕ_1 to ϕ_2	0		ns	
t_{D2}	Delay ϕ_2 to ϕ_1	70		ns	
t_{D3}	Delay ϕ_1 to ϕ_2 Leading Edges	80		ns	
t_{DA} [2]	Address Output Delay from ϕ_2		200	ns	C_L = 100pF
t_{DD} [2]	Data Output Delay from ϕ_2		220	ns	
t_{DC} [2]	Signal Output Delay from ϕ_1 or ϕ_2 (SYNC, $\overline{WR}$, WAIT, HLDA)		120	ns	C_L = 50pF
t_{DF} [2]	DBIN Delay from ϕ_2	25	140	ns	
t_{DI} [1]	Delay for Input Bus to Enter Input Mode		t_{DF}	ns	
t_{DS1}	Data Setup Time During ϕ_1 and DBIN	30		ns	

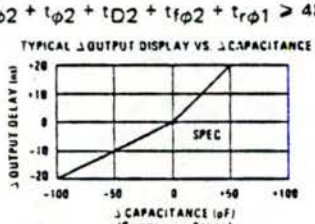
Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{DS2}	Data Setup Time to ϕ_2 During DBIN	150		ns	
t_{DH} [1]	Data Hold Time from ϕ_2 During DBIN	1		ns	
t_{IE} [2]	INTE Output Delay from ϕ_2		200	ns	C_L = 50pF
t_{RS}	READY Setup Time During ϕ_2	120		ns	
t_{HS}	HOLD Setup Time to ϕ_2	140		ns	
t_{IS}	INT Setup Time During ϕ_2 (During ϕ_1 in Halt Mode)	120		ns	
t_H	Hold Time from ϕ_2 (READY, INT, HOLD)	0		ns	
t_{FD}	Delay to Float During Hold (Address and Data Bus)		120	ns	
t_{AW} [2]	Address Stable Prior to $\overline{WR}$	5		ns	C_L = 100pF : Address, Data C_L = 50pF: $\overline{WR}$, HLDA, DBIN
t_{DW} [2]	Output Data Stable Prior to $\overline{WR}$	6		ns	
t_{WD} [2]	Output Data Stable from $\overline{WR}$	7		ns	
t_A [2]	Address Stable from $\overline{WR}$	7		ns	
t_{HF} [2]	HLDA to Float Delay	8		ns	
t_{WF} [2]	$\overline{WR}$ to Float Delay	9		ns	
t_{AH} [2]	Address Hold Time After DBIN During HLDA	-20		ns	

Notes:

1. Data input should be enabled with DBIN status. No bus conflict can then occur and data hold time is assured. t_{DH} = 50ns or t_{DF}, whichever is less.
2. Typical load circuit:



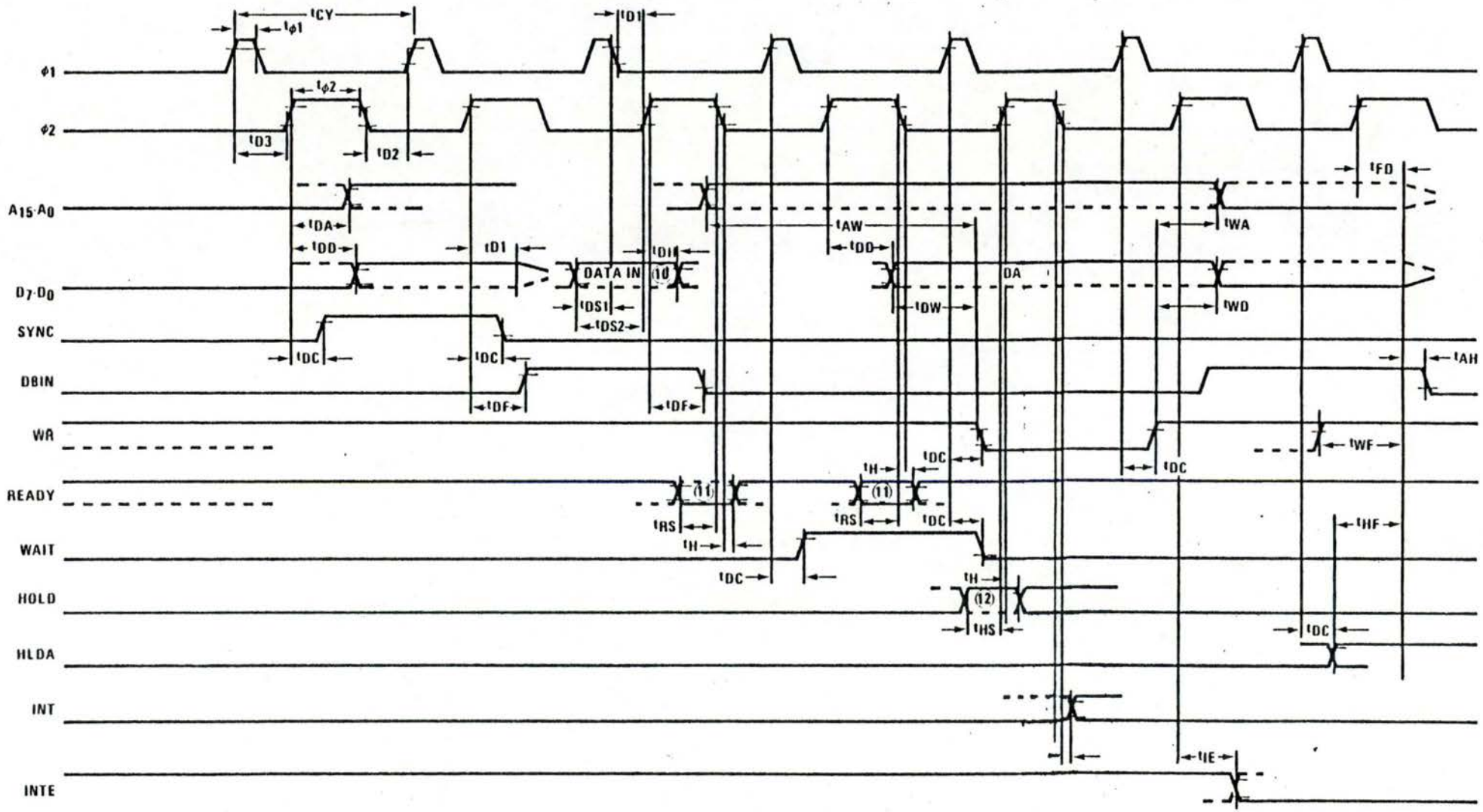
3. $t_{CY} = t_{D3} + t_{r\phi 2} + t_{\phi 2} + t_{D2} + t_{f\phi 2} + t_{r\phi 1} \geq 480$ns.



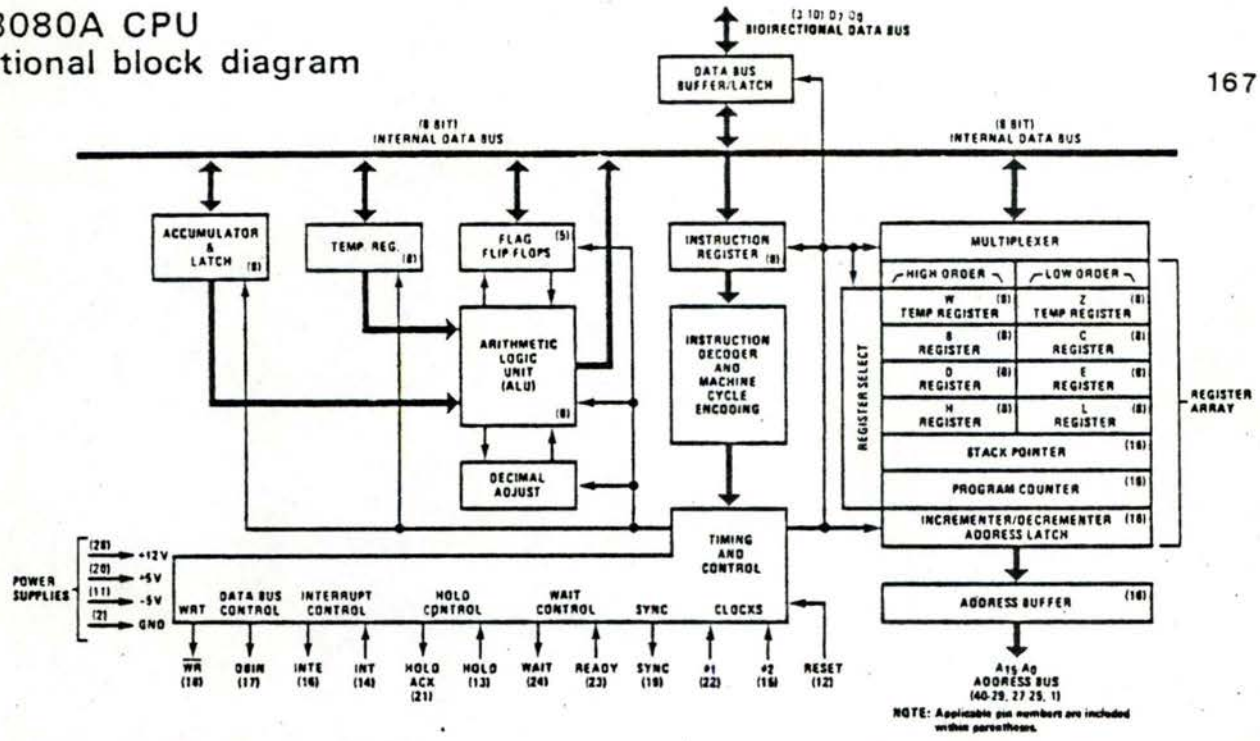
4. The following are relevant when interfacing the INS8080A to devices having V_{IH} = 3.3V:
   a) Maximum output rise time from 0.8V to 3.3V = 100ns @ C_L = SPEC.
   b) Output Delay when measured to 3.0V = SPEC + 60ns @ C_L = SPEC.
   c) If $C_L \neq$ SPEC, add 0.6ns/pF if $C_L > C_{SPEC}$, subtract 0.3ns/pF (from modified delay) if $C_L < C_{SPEC}$.
5. $t_{AW} = 2t_{CY} - t_{D3} - t_{r\phi 2} - 140$ns.
6. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi 2} - 170$ns.
7. If not HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi 2} + 10$ns. If HLDA, $t_{WD} = t_{WA} = t_{WF}$.
8. $t_{HF} = t_{D3} + t_{r\phi 2} - 50$ns.
9. $t_{WF} = t_{D3} + t_{r\phi 2} - 10$ns.
10. Data in must be stable for this period during DBIN · T_3. Both t_{DS1} and t_{DS2} must be satisfied.
11. Ready signal must be stable for this period during T_2 or T_W. (Must be externally synchronized.)
12. Hold signal must be stable for this period during T_2 or T_W when entering hold mode, and during T_3, T_4, T_5, and T_{WH} when in hold mode. (External synchronization is not required.)
13. Interrupt signal must be stable during this period of the last clock cycle of any instruction in order to be recognized on the following instruction. (External synchronization is not required.)
14. This timing diagram shows timing relationships only; it does not represent any specific machine cycle.

timing waveforms

Note: Timing measurements are made at the following reference voltages: CLOCK '1' = 8.0V, '0' = 1.0V; INPUTS '1' = 3.3V, '0' = 0.8V; OUTPUTS '1' = 2.0V, '0' = 0.8V.



INS8080A CPU functional block diagram



INS8080A functional pin definition

The following describes the function of all of the INS8080A input/output pins. Some of these descriptions reference internal timing periods.

INPUT SIGNALS

Ready: When high (logic 1), indicates that valid memory or input data are available to the CPU on the INS8080A data bus. The READY signal is used to synchronize the CPU with slower memory or input/output devices. If the INS8080A does not receive a high READY input after sending out an address to memory or an input/output device, the INS8080A enters a WAIT mode for as long as the READY input remains low (logic 0). The CPU may also be single stepped by the use of the READY signal.

Hold: When high, requests that the CPU enter the HOLD mode. When the CPU is in the HOLD mode, the CPU address and data buses both will be in the high-impedance state. The HOLD mode allows an external device to gain control of the INS8080A address and data buses immediately following the completion of the current machine cycle by the CPU. The CPU acknowledges the HOLD mode via the HOLD ACKNOWLEDGE (HLDA) output line. The HOLD request is recognized under the following conditions:

- The CPU is in the HALT mode.
- The READY signal is active and the CPU is in the t_2 or t_W microcycle.

Interrupt (INT) Request: When high, the CPU recognizes an interrupt request on this line after completing the current instruction or while in the HALT mode. An interrupt request is not honored if the CPU is in the HOLD mode (HLDA = logic 1) or the Interrupt Enable Flip-flop is reset (INTE = logic 0).

Reset: When activated (high) for a minimum of three clock periods, the content of the Program Counter is cleared and the Interrupt Enable and Hold Acknowledge Flip-flops are reset. Following a RESET, program execution starts at memory location 0. It should be noted that the status flags accumulator, stack pointer, and registers are not cleared during the RESET sequence.

ϕ_1 and ϕ_2 Clocks: Two non-TTL compatible clock phase which provide nonoverlapping timing references for interna storage elements and logic circuits of the CPU.

+12 Volts: V_{DD} Supply.

+5 Volts: V_{CC} Supply.

–5 Volts: V_{BB} Supply.

Ground: V_{SS} (0 volt) reference.

OUTPUT SIGNALS

Synchronizing (SYNC) Signal: When activated (high), th beginning of a new machine cycle is indicated and th status word is outputted on the Data Bus.

Address (A_{15} – A_0) Bus: This bus comprises sixtee TRI-STATE output lines. The bus provides the address t memory (up to 65k bytes) or denotes the input/outpu device number for up to 256 input and 256 output per pherals.

Wait: When high, acknowledges that the CPU is in th WAIT mode.

Write ($\overline{WR}$): When low, the data on the data bus are stab for WRITE memory or output operation.

Hold Acknowledge (HLDA): Goes high in response to logic 1 on the HOLD line and indicates that the data ar address bus will go to the high-impedance state. The HLD begins at one of the following times:

- The t_3 microcycle of a READ memory input operatio
- The clock period following the t_3 microcycle of WRITE memory output operation.

In both cases, the HLDA signal starts after the rising ed of the ϕ_1 clock, and high impedance occurs after the risi edge of the ϕ_2 clock.

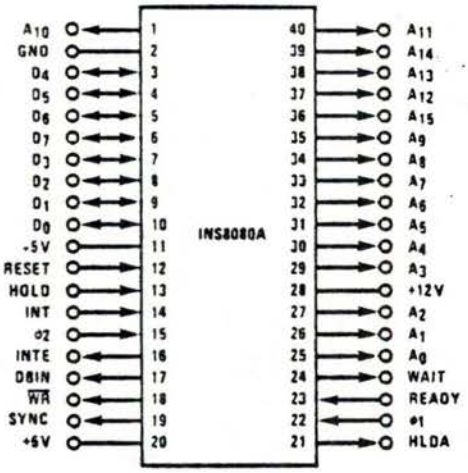
Interrupt Enable (INTE): Indicates the content of the internal Interrupt Enable Flip-flop. The Enable and Disable Interrupt (EI and DI) Instructions cause the Interrupt Enable Flip-flop to be set and reset, respectively. When the flip-flop is reset (INTE = logic 0), it inhibits interrupts from being accepted by the CPU. In addition, the Interrupt Enable Flip-flop is automatically reset (thereby disabling further interrupts) at the t_1 microcycle of the instruction fetch cycle, when an interrupt is accepted; it is also reset by the RESET Signal.

Data Bus In (DBIN): When high, indicates to external circuits that the data bus is in the input mode. The DBIN Signal should be used to gate data from memory or an I/O device onto the Data Bus.

INPUT/OUTPUT SIGNALS

Data ($D_7 - D_0$) Bus: This bus comprises eight TRI-STATE input/output lines. The bus provides bidirectional communication between the CPU, memory, and input/output devices for instructions and data transfers. A status word (which describes the current machine cycle) is also outputted on the data bus during the first microcycle of each machine cycle (SYNC = logic 1).

pin configuration



8080A status

Instructions for the 8080A require from one to five machine cycles for complete execution. The 8080A sends out 8 bits of status information on the data bus at the beginning of each machine cycle (during SYNC time). The following table defines the status information.

Status Information Definition

Symbols	Data Bus Bit	Definition
INTA*	D_0	Acknowledge signal for INTERRUPT request. Signal should be used to gate a restart instruction onto the data bus when DBIN is active.
$\overline{WO}$	D_1	Indicates that the operation in the current machine cycle will be a WRITE memory or OUTPUT function (WO = 0). Otherwise, a READ memory or INPUT operation will be executed.
STACK	D_2	Indicates that the address bus holds the pushdown stack address from the Stack Pointer.
HLTA	D_3	Acknowledge signal for HALT Instruction.
OUT	D_4	Indicates that the address bus contains the address of an output device and the data bus will contain the output data when WR is active.
M_1	D_5	Provides a signal to indicate that the CPU is in the fetch cycle for the first byte of an instuction.
INP*	D_6	Indicates that the address bus contains the address of an input device and the input data should be placed on the data bus when DBIN is active.
MEMR*	D_7	Designates that the data bus will be used for memory read data.

*These three status bits can be used to control the flow of data onto the INS8080A data bus.

Status Word Chart

Machine Cycle	Type	Data Bus Bit							
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Instruction Fetch	1	1	0	1	0	0	0	1	0
Memory Read	2	1	0	0	0	0	0	1	0
Memory Write	3	0	0	0	0	0	0	0	0
Stack Read	4	1	0	0	0	0	1	1	0
Stack Write	5	0	0	0	0	0	1	0	0
Input Read	6	0	1	0	0	0	0	1	0
Output Write	7	0	0	0	1	0	0	0	0
Interrupt Acknowledge	8	0	0	1	0	0	0	1	1
Halt Acknowledge	9	1	0	0	0	1	0	1	0
Interrupt Acknowledge While Halt	10	0	0	1	0	1	0	1	1

Mnemonic		Description	Operation	Op Code								No. of Bytes	No. of Machine (M) Cycles	No. of μcycles (T)	Condition Flags				
				D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				S	Z	AC	P	CY
DATA TRANSFER GROUP																			
LDA		Load Accumulator Direct	(A) ← ((byte 3)(byte 2))	0	0	1	1	1	0	1	0	3	4	13					
LDAX	B	Load Accumulator Indirect	(A) ← ((B))	0	0	0	0	1	0	1	0	1	2	7					
LDAX	D	Load Accumulator Indirect	(A) ← ((D))	0	0	0	1	1	0	1	0	1	2	7					
LHLD		Load H and L Direct	(L) ← ((byte 3)(byte 2)) (H) ← ((byte 3)(byte 2) + 1)	0	0	1	0	1	0	1	0	3	5	16					
LXI	B	Load Immediate, Registers B and C	(B) ← (byte 3) (C) ← (byte 2)	0	0	0	0	0	0	0	1	3	3	10					
LXI	D	Load Immediate, Registers D and E	(D) ← (byte 3) (E) ← (byte 2)	0	0	0	1	0	0	0	1	3	3	10					
LXI	H	Load Immediate, Registers H and L	(H) ← (byte 3) (L) ← (byte 2)	0	0	1	0	0	0	0	1	3	3	10					
LXI	SP	Load Immediate, Stack Pointer	(SPH) ← (byte 3) (SPL) ← (byte 2)	0	0	1	1	0	0	0	1	3	3	10	(Flags Not Affected)				
MOV	M, r	Move to Memory	((H)(L)) ← (r)	0	1	1	1	0	S	S	S	1	2	7					
MOV	r, M	Move from Memory	(r) ← ((H)(L))	0	1	D	D	D	1	1	0	1	2	7					
MOV	r1, r2	Move Registers	(r1) ← (r2)	0	1	D	D	D	S	S	S	1	1	5					
MVI	M	Move to Memory Immediate	((H)(L)) ← (byte 2)	0	0	1	1	0	1	1	0	2	3	10					
MVI	r	Move Immediate	(r) ← (byte 2)	0	0	D	D	D	1	1	0	2	2	7					
SHLD		Store H and L Direct	((byte 3)(byte 2)) ← (L) ((byte 3)(byte 2) + 1) ← (H)	0	0	1	0	0	0	1	0	3	5	16					
STA		Store Accumulator Direct	((byte 3)(byte 2)) ← (A)	0	0	1	1	0	0	1	0	3	4	13					
STAX	B	Store Accumulator Indirect	((B)) ← (A)	0	0	0	0	0	0	1	0	1	2	7					
STAX	D	Store Accumulator Indirect	((D)) ← (A)	0	0	0	1	0	0	1	0	1	2	7					
XCHG		Exchange H and L with D and E	(H) ↔ (D) (L) ↔ (E)	1	1	1	0	1	0	1	1	1	1	4					
ARITHMETIC GROUP																			
ACI		Add Immediate with Carry	(A) ← (A) + (byte 2) + (CY)	1	1	0	0	1	1	1	0	2	2	7	↕	↕	↕	↕	↕
ADC	M	Add Memory with Carry	(A) ← (A) + ((H)(L)) + (CY)	1	0	0	0	1	1	1	0	1	2	7	↕	↕	↕	↕	↕
ADC	r	Add Register with Carry	(A) ← (A) + (r) + (CY)	1	0	0	0	1	S	S	S	1	1	4	↕	↕	↕	↕	↕
ADD	M	Add Memory	(A) ← (A) + ((H)(L))	1	0	0	0	0	1	1	0	1	2	7	↕	↕	↕	↕	↕
ADD	r	Add Register	(A) ← (A) + (r)	1	0	0	0	0	S	S	S	1	1	4	↕	↕	↕	↕	↕
ADI		Add Immediate	(A) ← (A) + (byte 2)	1	1	0	0	0	1	1	0	1	2	7	↕	↕	↕	↕	↕
DAA		Decimal Adjust Accumulator	8-bit number in Accumulator is converted to two 4-bit BCD digits	0	0	1	0	0	1	1	1	1	1	4	↕	↕	↕	↕	↕
DAD	B	Add B and C to H and L	(H)(L) ← (H)(L) + (B)(C)	0	0	0	0	1	0	0	1	1	3	10	·	·	·	·	↕
DAD	D	Add D and E to H and L	(H)(L) ← (H)(L) + (D)(E)	0	0	0	1	1	0	0	1	1	3	10	·	·	·	·	↕
DAD	H	Add H and L to H and L	(H)(L) ← (H)(L) + (H)(L)	0	0	1	0	1	0	0	1	1	3	10	·	·	·	·	↕
DAD	SP	Add Stack Pointer to H and L	(H)(L) ← (H)(L) + (SP)	0	0	1	1	1	0	0	1	1	3	10	·	·	·	·	↕
DCR	M	Decrement Memory	((H)(L)) ← ((H)(L)) - 1	0	0	1	1	0	1	0	1	1	3	10	↕	↕	↕	↕	·
DCR	r	Decrement Register	(r) ← (r) - 1	0	0	D	D	D	1	0	1	1	1	5	↕	↕	↕	↕	·
DCX	B	Decrement Registers B and C	(B)(C) ← (B)(C) - 1	0	0	0	0	1	0	1	1	1	1	5	·	·	·	·	·
DCX	D	Decrement Registers D and E	(D)(E) ← (D)(E) - 1	0	0	0	1	1	0	1	1	1	1	5	·	·	·	·	·
DCX	H	Decrement Registers H and L	(H)(L) ← (H)(L) - 1	0	0	1	0	1	0	1	1	1	1	5	·	·	·	·	·
DCX	SP	Decrement Stack Pointer	(SP) ← (SP) - 1	0	0	1	1	1	0	1	1	1	1	5	·	·	·	·	·
INR	M	Increment Memory	((H)(L)) ← ((H)(L)) + 1	0	0	1	1	0	1	0	0	1	3	10	↕	↕	↕	↕	·
INR	r	Increment Register	(r) ← (r) + 1	0	0	D	D	D	1	0	0	1	1	5	↕	↕	↕	↕	·
INX	B	Increment Registers B and C	(B)(C) ← (B)(C) + 1	0	0	0	0	0	0	1	1	1	1	5	·	·	·	·	·
INX	D	Increment Registers D and E	(D)(E) ← (D)(E) + 1	0	0	0	1	0	0	1	1	1	1	5	·	·	·	·	·
INX	H	Increment Registers H and L	(H)(L) ← (H)(L) + 1	0	0	1	0	0	0	1	1	1	1	5	·	·	·	·	·
INX	SP	Increment Stack Pointer	(SP) ← (SP) + 1	0	0	1	1	0	0	1	1	1	1	5	·	·	·	·	·
SBB	M	Subtract Memory with Borrow	(A) ← (A) - ((H)(L)) - (CY)	1	0	0	1	1	1	1	0	1	2	7	↕	↕	↕	↕	↕
SBB	r	Subtract Register with Borrow	(A) ← (A) - (r) - (CY)	1	0	0	1	1	S	S	S	1	1	4	↕	↕	↕	↕	↕
SBI		Subtract Immediate with Borrow	(A) ← (A) - (byte 2) - (CY)	1	1	0	1	1	1	1	0	2	2	7	↕	↕	↕	↕	↕
SUB	M	Subtract Memory	(A) ← (A) - ((H)(L))	1	0	0	1	0	1	1	0	1	2	7	↕	↕	↕	↕	↕
SUB	r	Subtract Register	(A) ← (A) - (r)	1	0	0	1	0	S	S	S	1	1	4	↕	↕	↕	↕	↕
SUI		Subtract Immediate	(A) ← (A) - (byte 2)	1	1	0	1	0	1	1	0	2	2	7	↕	↕	↕	↕	↕
LOGICAL GROUP																			
ANA	M	AND Memory	(A) ← (A) ∧ ((H)(L))	1	0	1	0	0	1	1	0	1	2	7	↕	↕	·	↕	0
ANA	r	AND Register	(A) ← (A) ∧ (r)	1	0	1	0	0	S	S	S	1	1	4	↕	↕	·	↕	0
ANI		AND Immediate	(A) ← (A) ∧ (byte 2)	1	1	1	0	0	1	1	0	2	2	7	↕	↕	·	↕	0
CMA		Complement Accumulator	(A) ← ($\overline{A}$)	0	0	1	0	1	1	1	1	1	1	4	·	·	·	·	·
CMC		Complement Carry	(CY) ← ($\overline{CY}$)	0	0	1	1	1	1	1	1	1	1	4	·	·	·	·	↕
CMP	M	Compare Memory	(A) — ((H)(L))	1	0	1	1	1	1	1	0	1	2	7	↕	↕[a]	↕	↕	↕
CMP	r	Compare Register	(A) — (r)	1	0	1	1	1	S	S	S	1	1	4	↕	↕[b]	↕	↕	↕
CPI		Compare Immediate	(A) — (byte 2)	1	1	1	1	1	1	1	0	2	2	7	↕	↕[c]	↕	↕	↕
ORA	M	OR Memory	(A) ← (A) ∨ ((H)(L))	1	0	1	1	0	1	1	0	1	2	7	↕	↕	0	↕	0
ORA	r	OR Register	(A) ← (A) ∨ (r)	1	0	1	1	0	S	S	S	1	1	4	↕	↕	0	↕	0
ORI		OR Immediate	(A) ← (A) ∨ (byte 2)	1	1	1	1	0	1	1	0	2	2	7	↕	↕	0	↕	0
RAL		Rotate Left through Carry	(A_{n+1}) ← (A_n); (CY) ← (A_7) (A_0) ← (CY)	0	0	0	1	0	1	1	1	1	1	4	·	·	·	·	↕
RAR		Rotate Right through Carry	(A_n) ← (A_{n+1}); (CY) ← (A_0) (A_7) ← (CY)	0	0	0	1	1	1	1	1	1	1	4	·	·	·	·	↕
RLC		Rotate Left	(A_{n+1}) ← (A_n); (A_0) ← (A_7) (CY) ← (A_7)	0	0	0	0	0	1	1	1	1	1	4	·	·	·	·	↕
RRC		Rotate Right	(A_n) ← (A_{n-1}); (A_7) ← (A_0) (CY) ← (A_0)	0	0	0	0	1	1	1	1	1	1	4	·	·	·	·	↕
STC		Set Carry	(CY) ← 1	0	0	1	1	0	1	1	1	1	1	4	·	·	·	·	1
XRA	M	Exclusive OR Memory	(A) ← (A) ∀ ((H)(L))	1	0	1	0	1	1	1	0	1	2	7	↕	↕	0	↕	0
XRA	r	Exclusive OR Register	(A) ← (A) ∀ (r)	1	0	1	0	1	S	S	S	1	1	4	↕	↕	0	↕	0
XRI		Exclusive OR Immediate	(A) ← (A) ∀ (byte 2)	1	1	1	0	1	1	1	0	2	2	7	↕	↕	0	↕	0

Notes: a. Z = 1 if (A) = (H)(L); CY = 1 if (A) < (H)(L)

b. Z = 1 if (A) = (r); CY = 1 if (A) < (r)

c. Z = 1 if (A) = (byte 2); CY = 1 if (A) < (byte 2)

instruction set (cont'd.)

Mnemonic	Description	Operation	Op Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	No. of Bytes	No. of Machine (M) Cycles	No. of μcycles (T)	Condition Flags S	Z	AC	P	CY
BRANCH GROUP																		
CALL	Call Unconditional	((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	0	0	1	1	0	1	3	5	17					
CC	Call on Carry	If CY = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	0	1	1	1	0	0	3	3/5	11/17					
CM	Call on Minus	If S = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	1	1	1	1	0	0	3	3/5	11/17					
CNC	Call on No Carry	If CY = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	0	1	0	1	0	0	3	3/5	11/17					
CNZ	Call on Not Zero	If Z = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	0	0	0	1	0	0	3	3/5	11/17	(Flags Not Affected)				
CP	Call on Positive	If S = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	1	1	0	1	0	0	3	3/5	11/17					
CPE	Call on Parity Even	If P = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	1	0	1	1	0	0	3	3/5	11/17					
CPO	Call on Parity Odd	If P = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	1	0	0	1	0	0	3	3/5	11/17					
CZ	Call on Zero	If Z = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3) (byte 2)	1	1	0	0	1	1	0	0	3	3/5	11/17					
JC	Jump on Carry	If CY = 1, (PC) ← (byte 3) (byte 2)	1	1	0	1	1	0	1	0	3	3	10					
JM	Jump on Minus	If S = 1, (PC) ← (byte 3) (byte 2)	1	1	1	1	1	0	1	0	3	3	10					
JMP	Jump Unconditional	(PC) ← (byte 3) (byte 2)	1	1	0	0	0	0	1	1	3	3	10					
JNC	Jump on No Carry	If CY = 0, (PC) ← (byte 3) (byte 2)	1	1	0	1	0	0	1	0	3	3	10					
JNZ	Jump on Not Zero	If Z = 0, (PC) ← (byte 3) (byte 2)	1	1	0	0	0	0	1	0	3	3	10					
JP	Jump on Positive	If S = 0, (PC) ← (byte 3) (byte 2)	1	1	1	1	0	0	1	0	3	3	10					
JPE	Jump on Parity Even	If P = 1, (PC) ← (byte 3) (byte 2)	1	1	1	0	1	0	1	0	3	3	10					
JPO	Jump on Parity Odd	If P = 0, (PC) ← (byte 3) (byte 2)	1	1	1	0	0	0	1	0	3	3	10					
JZ	Jump on Zero	If Z = 1, (PC) ← (byte 3) (byte 2)	1	1	0	0	1	0	1	0	3	3	10					
PCHL	H and L to Program Counter	(PCH) ← (H) (PCL) ← (L)	1	1	1	0	1	0	0	1	1	1	5					
RC	Return on Carry	If CY = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	1	1	0	0	0	1	1/3	5/11					
RET	Return	(PCL) ← ((SP)); (PCH) ← ((SP) + 1); (SP) ← (SP) + 2;	1	1	0	0	1	0	0	1	1	3	10					
RM	Return on Minus	If S = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	1	1	0	0	0	1	1/3	5/11					
RNC	Return on No Carry	If CY = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	1	0	0	0	0	1	1/3	5/11					
RNZ	Return on Not Zero	If Z = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	0	0	0	0	0	1	1/3	5/11					
RP	Return on Positive	If S = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	1	0	0	0	0	1	1/3	5/11					
RPE	Return on Parity Even	If P = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	0	1	0	0	0	1	1/3	5/11					
RPO	Return on Parity Odd	If P = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	0	0	0	0	0	1	1/3	5/11					

Mnemonic		Description	Operation	Op Code								No. of Bytes	No. of Machine (M) Cycles	No. of μcycles (T)	Condition Flags				
				D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				S	Z	AC	P	CY
BRANCH GROUP (continued)																			
RST		Restart	((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← 8 * (NNN)	1	1	N	N	N	1	1	1	1	3	11					
RZ		Return on Zero	If Z = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	0	1	0	0	0	1	1/3	5/11					
STACK, I/O, AND MACHINE CONTROL GROUP																			
DI		Disable Interrupts	The interrupt system is disabled following the execution of the DI instruction.	1	1	1	1	0	0	1	1	1	1	4	·	·	·	·	·
EI		Enable Interrupts	The interrupt system is enabled following the execution of next instruction.	1	1	1	1	1	0	1	1	1	1	4	·	·	·	·	·
HLT		Halt	Processor is stopped; registers and flags are unaffected.	0	1	1	1	0	1	1	0	1	1	7	·	·	·	·	·
IN		Input	(A) ← (data)	1	1	0	1	1	0	1	1	2	3	10	·	·	·	·	·
NOP		No Operation	No operation is performed; registers and flags are unaffected.	0	0	0	0	0	0	0	0	1	1	4	·	·	·	·	·
OUT		Output	(data) ← (A)	1	1	0	1	0	0	1	1	2	3	10	·	·	·	·	·
POP	B	Pop Registers B and C off Stack	(C) ← ((SP)) (B) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	0	0	0	0	1	1	3	10	·	·	·	·	·
POP	D	Pop Registers D and E off Stack	(D) ← ((SP)) (E) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	1	0	0	0	1	1	3	10	·	·	·	·	·
POP	H	Pop Registers H and L off Stack	(H) ← ((SP)) (L) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	0	0	0	0	1	1	3	10	·	·	·	·	·
POP	PSW	Pop Accumulator and Flags off Stack	(CY) ← $((SP))_0$ (P) ← $((SP))_2$ (AC) ← $((SP))_4$ (Z) ← $((SP))_6$ (S) ← $((SP))_7$ (A) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	1	0	0	0	1	1	3	10	‡	‡	‡	‡	‡
PUSH	B	Push Registers B and C on Stack	((SP) - 1) ← (B) ((SP) - 2) ← (C) (SP) ← (SP) - 2	1	1	0	0	0	1	0	1	1	3	11	·	·	·	·	·
PUSH	D	Push Registers D and E on Stack	((SP) - 1) ← (D) ((SP) - 2) ← (E) (SP) ← (SP) - 2	1	1	0	1	0	1	0	1	1	3	11	·	·	·	·	·
PUSH	H	Push Registers H and L on Stack	((SP) - 1) ← (H) ((SP) - 2) ← (L) (SP) ← (SP) - 2	1	1	1	0	0	1	0	1	1	3	11	·	·	·	·	·
PUSH	PSW	Push Accumulator and Flags on Stack	((SP) - 1) ← (A) $((SP) - 2)_0$ ← (CY) $((SP) - 2)_1$ ← 1 $((SP) - 2)_2$ ← (P) $((SP) - 2)_3$ ← 0 $((SP) - 2)_4$ ← (AC) $((SP) - 2)_5$ ← 0 $((SP) - 2)_6$ ← (Z) $((SP) - 2)_7$ ← (S) (SP) ← (SP) - 2	1	1	1	1	0	1	0	1	1	3	11	·	·	·	·	·
SPHL		Move H and L to Stack Pointer	(SP) ← (H) (L)	1	1	1	1	1	0	0	1	1	1	5	·	·	·	·	·
XTHL		Exchange Top of Stack with H and L	(L) ↔ ((SP)) (H) ↔ ((SP) + 1)	1	1	1	0	0	0	1	1	1	5	18	·	·	·	·	·

condition flags and standard rules

There are five condition flags associated with the execution of instructions on the INS8080A. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and each flag is represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1, "reset" by forcing the bit to 0. The bit positions of the flags are indicated in the PUSH and POP PSW instructions.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

ZERO (Z): If the result of an instruction has the value 0, this flag is set; otherwise, it is reset.

SIGN (S): If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise, it is reset.

PARITY (P): If the modulo 2 sum of the bits of the result of the operation is 0 (that is, if the result has even parity), this flag is set; otherwise, it is reset (that is, if the result has odd parity).

CARRY (CY): If the instruction resulted in a carry (from addition) or a borrow (from subtraction or a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

AUXILIARY CARRY (AC): If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value the auxiliary carry is set; otherwise, it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations; however, AC is used principally with additions and increment preceding a DAA (Decimal Adjust Accumulator) Instruction.

symbols and abbreviations

The following symbols and abbreviations are used in the subsequent description of the INS8080A instructions:

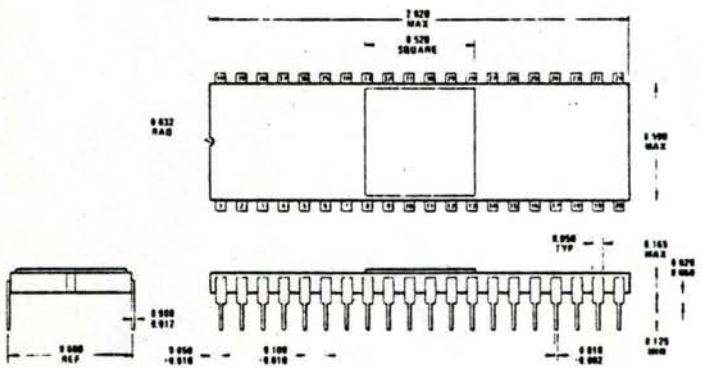
Symbols	Meaning
A	Register A (Accumulator)
B	Register B
C	Register C
D	Register D
H	Register H
L	Register L
DDD, SSS	The bit pattern designating one of the registers A, B, C, D, E, H, L (DDD = destination, SSS = source):

DDD or SSS	Register Name
111	A
000	B
001	C
010	D
011	E
100	H
101	L

byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r, r1, r2	One of the registers A, B, C, D, E, H, L

Symbols	Meaning
PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively.)
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively.)
( )	The contents of the memory location or registers enclosed in the parentheses
←	"Is replaced by"
∧	Logical AND
∀	Exclusive OR
∨	Inclusive OR
+	Addition
−	Twos complement subtraction
*	Multiplication
↔	"Exchange"
‾	The ones complement (for example, $(\overline{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7 respectively
•	"Not affected"
0	"Reset"
1	"Set"
x	Unknown
↕	Flags affected according to Standard Rules

physical dimensions



A N N E X E 3.

FICHE TECHNIQUE DE LA MEMOIRE COMMUNE 74170

4 by 4 Register Files

General Description

These 16-bit TTL register files are organized as 4 words of 4 bits each, and separate on-chip decoding is provided for addesssing the four word locations to either write-in or retrieve data. This permits writing into one location and reading from another word location, simultaneously.

Four data inputs are available to supply the 4-bit word to be stored. Location of the word is determined by the write-address inputs A and B, in conjunction with a write-enable signal. Data applied at the inputs should be in its true form. That is, if a high-level signal is desired from the output, a high level is applied at the data input for that particular bit location. The latch inputs are arranged so that new data will be accepted only if both internal address gate inputs are high. When this condition exists, data at the D input is transferred to the latch output. When the write-enable input, G_W, is high, the data inputs are inhibited and their levels can cause no change in the information stored in the internal latches. When the read-enable input, G_R, is high, the data outputs are inhibited and remain high.

The individual address lines permit direct reading of data stored in any four of the latches. Four individual decoding gates are used to complete the address for reading a word. When the read address is made in conjunction with the read-enable signal, the word appears at the four outputs.

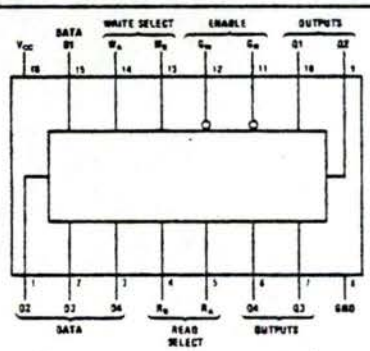
This arrangement—data entry addressing separate from data-read addressing and individual sense line—eliminates recovery times, permits simultaneous reading and writing, and is limited in speed only by the write time (30 ns typical) and the read time (25 ns typical). The register file has a nondestructive readout in that data is not lost when addressed.

All 170 inputs and all inputs except the read enable and write enable of the LS170 are buffered to lower the drive requirements to one standard load. Input-clamping diodes minimize switching transients to simplify system design. High-speed, double-ended AND-OR-INVERT gates are employed for the read-address function and drive high-sink-current, open-collector outputs. Up to 256 of these outputs may be wire-AND connected for increasing the capacity up to 1024 words. Any number of these registers may be paralleled to provide n-bit word length.

Features

- Separate addressing permits simultaneous reading and writing
- Fast access times typically 20 ns
- Organized as 4 words of 4 bits
- Expandable to 1024 words of n-bits
- For use as:
  - Scratch-pad memory
  - Buffer storage between processors
  - Bit storage in fast multiplication designs
- Open-collector outputs with low maximum off-state current:
  - 170 30μA
  - LS170 20μA
- DM54LS670 and DM74LS670 are similar but have TRI-STATE outputs

Connection Diagram



74170(J), (N);
54LS170/74LS170(J), (N), (W)

Truth Tables

WRITE TRUTH TABLE (SEE NOTES A, B, AND C)

WRITE INPUTS			WORD			
W_B	W_A	G_W	0	1	2	3
L	L	L	Q = D	Q_0	Q_0	Q_0
L	H	L	Q_0	Q = D	Q_0	Q_0
H	L	L	Q_0	Q_0	Q = D	Q_0
H	H	L	Q_0	Q_0	Q_0	Q = D
X	X	H	Q_0	Q_0	Q_0	Q_0

READ TRUTH TABLE (SEE NOTES A AND D)

READ INPUTS			OUTPUTS			
R_B	R_A	G_R	Q1	Q2	Q3	Q4
L	L	L	W01	W02	W03	W04
L	H	L	W11	W12	W13	W14
H	L	L	W21	W22	W23	W24
H	H	L	W31	W32	W33	W34
X	X	H	H	H	H	H

Notes

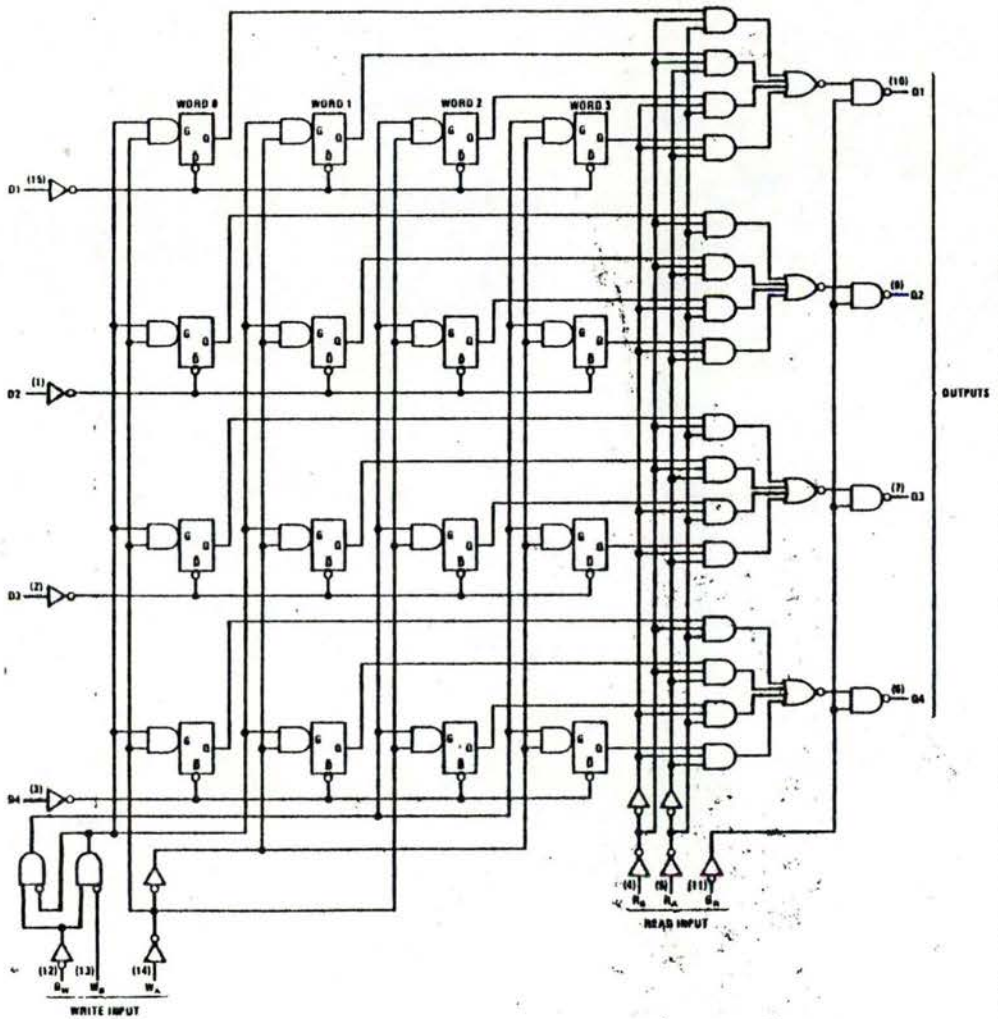
(A) H = High Level, L = Low Level, X = Don't Care

(B) (Q = D) = The four selected internal flip-flop outputs will assume the states applied to the four external data inputs.

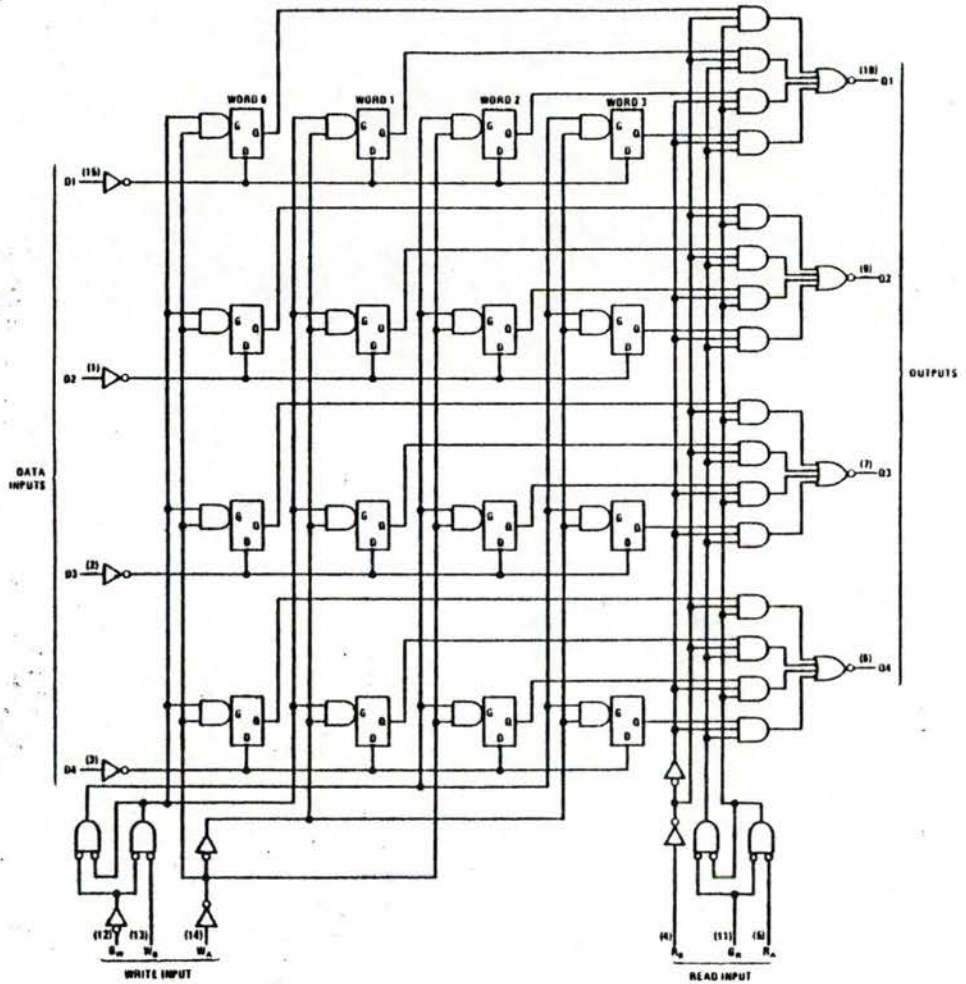
(C) Q_0 = The level of Q before the indicated input conditions were established.

(D) WORD = The first bit of word 0, etc.

170


WORD 0
WORD 1
WORD 2
WORD 3
D1
D2
D3
D4
OUTPUTS
WRITE INPUT
READ INPUT

LS170


WORD 0
WORD 1
WORD 2
WORD 3
DATA INPUTS
OUTPUTS
WRITE INPUT
READ INPUT

A N N E X E 4.

LE TEMPORISATEUR PROGRAMMABLE 8253.

intel

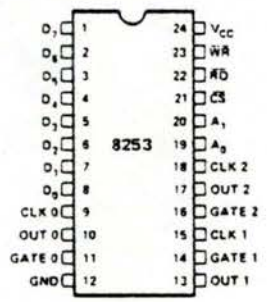
8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- MCS—85™ Compatible 8253-5
- 3 Independent 16-Bit Counters
- DC to 2 MHz
- Programmable Counter Modes
- Count Binary or BCD
- Single +5V Supply
- 24-Pin Dual In-Line Package

The Intel® 8253 is a programmable counter/timer chip designed for use as an intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.

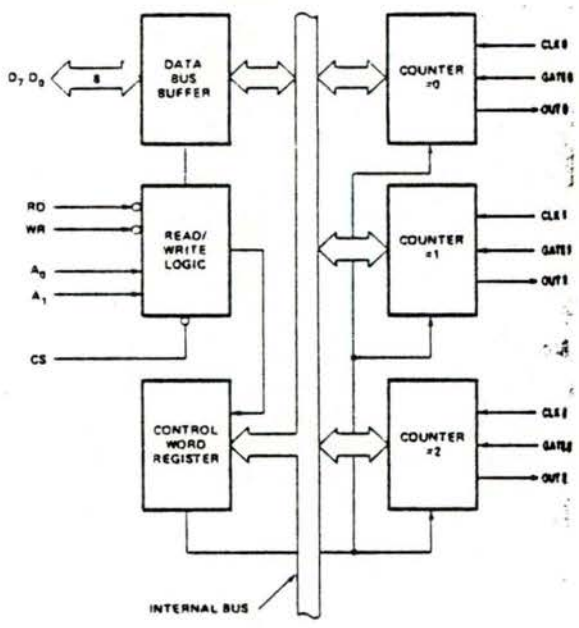
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (8 BIT)
CLK N	COUNTER CLOCK INPUTS
GATE N	COUNTER GATE INPUTS
OUT N	COUNTER OUTPUTS
RD	READ COUNTER
WR	WRITE COMMAND OR DATA
CS	CHIP SELECT
A_0, A_1	COUNTER SELECT
V_{CC}	+5 VOLTS
GND	GROUND

BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value.

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

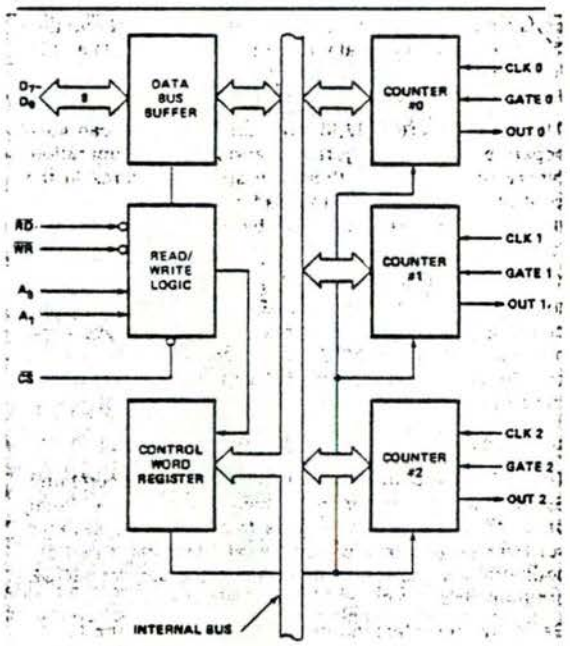


Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described. Each Counter consists of a single, 16-bit, pre-settable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate Mode configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel® 8205 for larger systems.

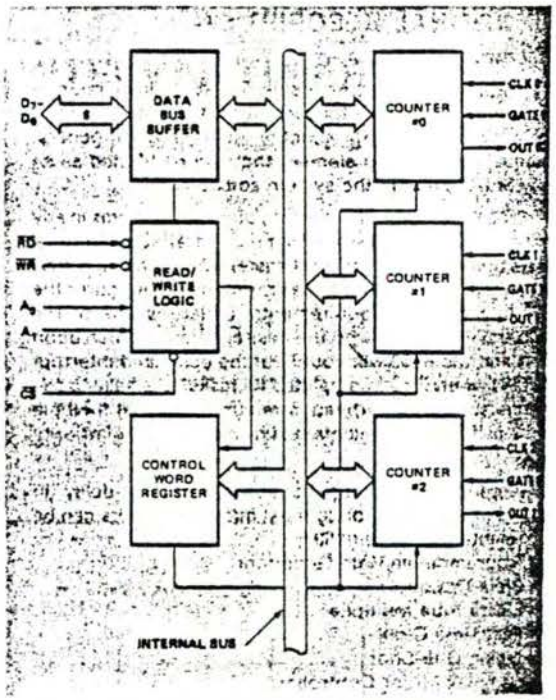


Figure 2. Block Diagram Showing Control Word Register and Counter Functions

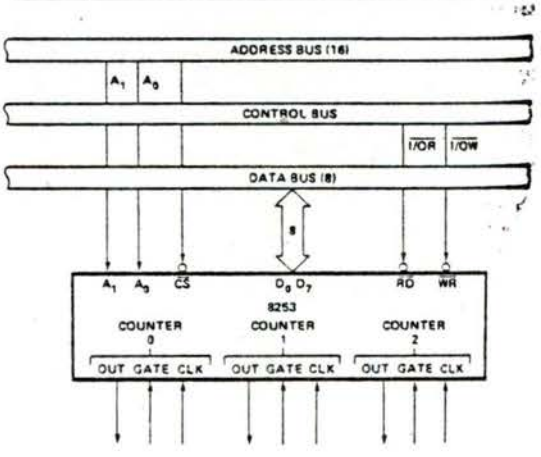


Figure 3. 8253 System Interface

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. These control words program the MODE, counting sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever counting tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided internally so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the same software by simple I/O operations.

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register. (A1, A0 = 11)

Control Word Format

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

Read/Load:

RL0	
0	Counter Latching operation (see READ/WRITE Procedure Section)
0	Read/Load most significant byte only.
1	Read/Load least significant byte only.
1	Read/Load least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE Definition

MODE 0: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.
(2) Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for $(N+1)/2$ counts and low for $(N-1)/2$ counts.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

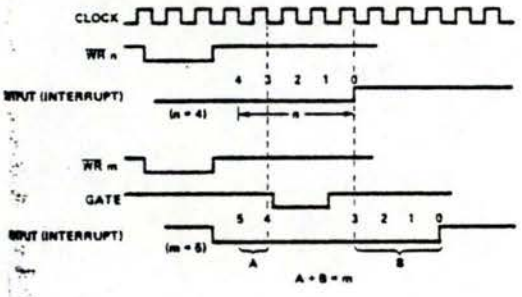
If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is low. Reloading the counter register will restart counting beginning with the new number.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

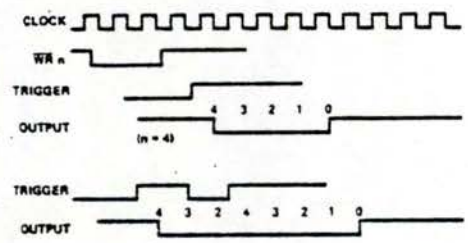
Modes \ Signal Status	Low Or Going Low	Rising	High
0	Disables counting	— —	Enables counting
1	— —	1) Initiates counting 2) Resets output after next clock	— —
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	— —	Enables counting
5	— —	Initiates counting	— —

Figure 4. Gate Pin Operations Summary

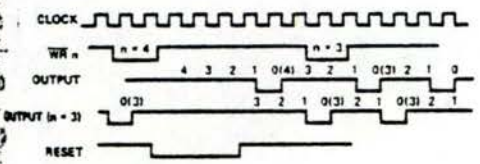
MODE 0: Interrupt on Terminal Count



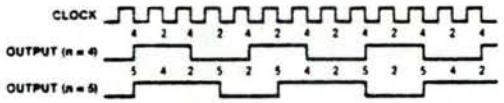
MODE 1: Programmable One-Shot



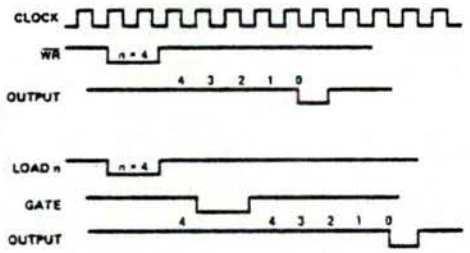
MODE 2: Rate Generator



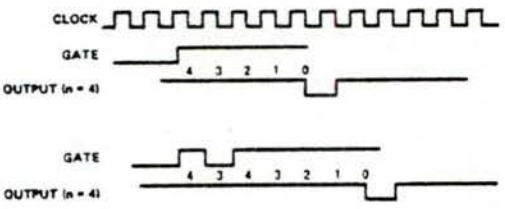
MODE 3: Square Wave Generator



MODE 4: Software Triggered Strobe



MODE 5: Hardware Triggered Strobe



& 8253 Timing Diagrams

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Count Register byte Counter n
MSB	Count Register byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 6. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Figure 7. Alternate Programming Formats

...ad Operations

...most counter applications it becomes necessary to read ... value of the count in progress and make a ...mputational decision based on this quantity. Event ...unters are probably the most common application that ...es this function. The 8253 contains logic that will allow ... programmer to easily read the contents of any of the ...ree counters without disturbing the actual count in ...ogress.

...ere are two methods that the programmer can use to ...d the value of the counters. The first method involves ... use of simple I/O read operations of the selected ...unter. By controlling the A0, A1 inputs to the 8253 the ...ogrammer can select the counter to be read (remember ...t no read operation of the mode register is allowed A0, ...=11). The only requirement with this method is that in ...der to assure a stable count reading the actual operation ... the selected counter must be inhibited either by ...ntrolling the Gate input or by external logic that inhibits ... clock input. The contents of the counter selected will ... available as follows:

...st I/O Read contains the least significant byte (LSB).

...cond I/O Read contains the most significant byte ...SB).

...e to the internal logic of the 8253 it is absolutely ...cessary to complete the entire reading procedure. If two ...tes are programmed to be read then two bytes must be ...d before any loading WR command can be sent to the ...me counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter and the contents of the latched register is available.

MODE Register for Latching Count

A0, A1 = 11

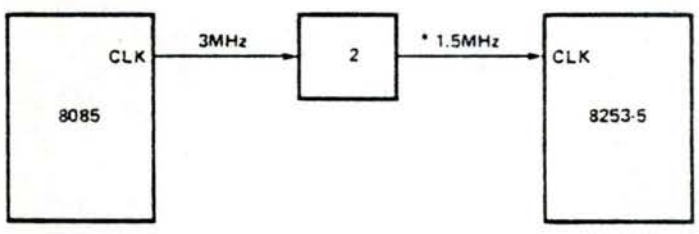
D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1,SC0 — specify counter to be latched.

D5,D4 — 00 designates counter latching operation.

X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less.

Figure 8. MCS-85™ Clock Interface*

A N N E X E 5.

LE CONTROLEUR D'INTERRUPTIONS PROGRAMMABLE 8259.

intel

8259A
PROGRAMMABLE INTERRUPT CONTROLLER

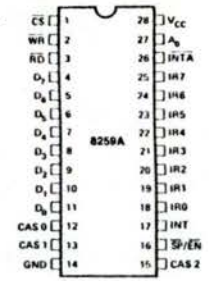
- MCS-86™ Compatible
- MCS-80/85™ Compatible
- Eight-Level Priority Controller
- Expandable to 64 Levels
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Single +5V Supply (No Clocks)
- 28-Pin Dual-In-Line Package

The Intel® 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the Intel® 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

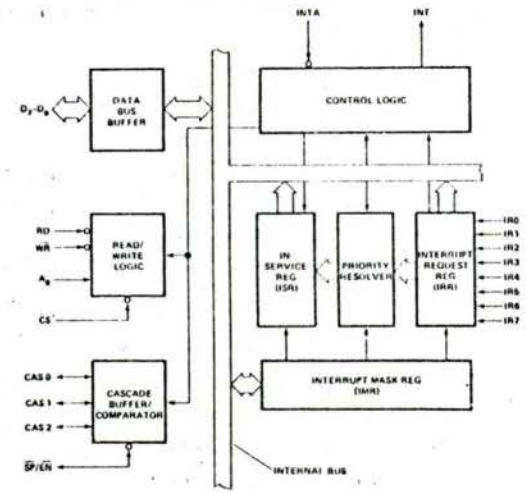
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI DIRECTIONAL)
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A_0	COMMAND SELECT ADDRESS
$\overline{CS}$	CHIP SELECT
CAS2 CAS0	CASCADE LINES
$\overline{SP}/\overline{EN}$	SLAVE PROGRAM INPUT/ENABLE
INT	INTERRUPT OUTPUT
$\overline{INTA}$	INTERRUPT ACKNOWLEDGE INPUT
IR0-IR7	INTERRUPT REQUEST INPUTS

BLOCK DIAGRAM



© Intel Corporation, 1978

INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

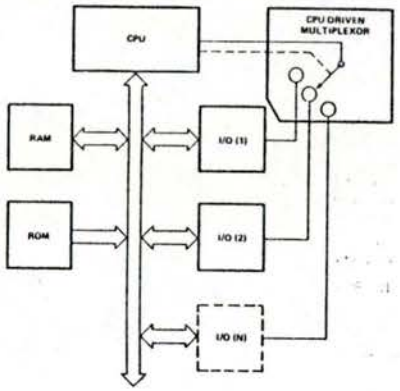
The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

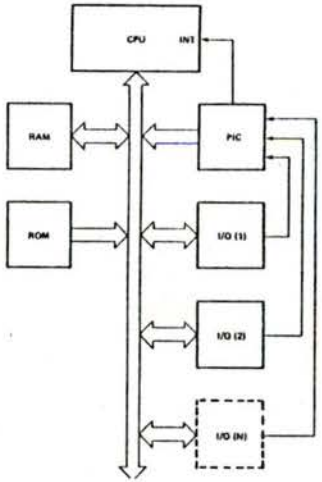
8259A BASIC FUNCTIONAL DESCRIPTION

GENERAL

The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.



Polled Method



Interrupt Method

INTERRUPT REQUEST REGISTER (IRR) AND IN SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

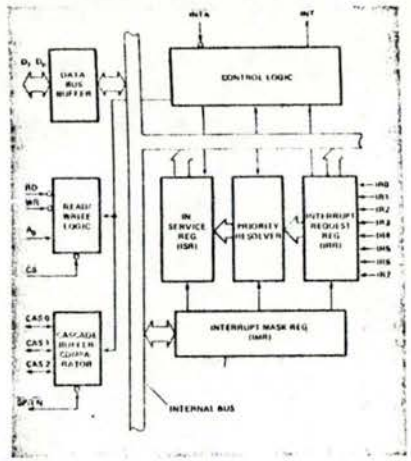
A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

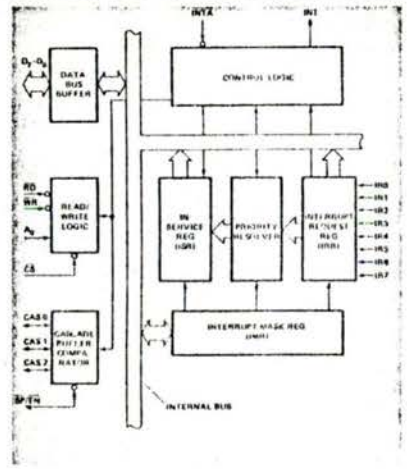
A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

$\overline{RD}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.



8259A Block Diagram



8259A Block Diagram

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 8259A".)

INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

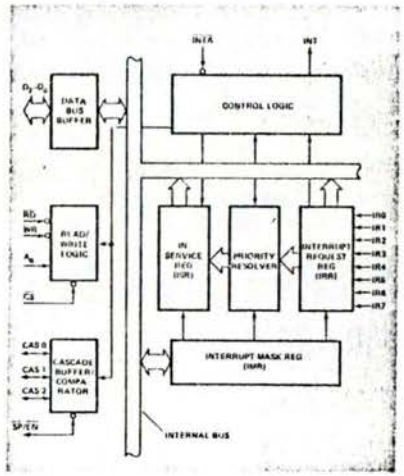
The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 8259A from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

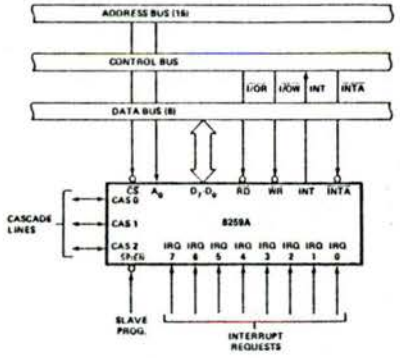
The events occurring in an MCS-86 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The MCS-86 CPU will initiate a second $\overline{INTA}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.



8259A Block Diagram



8259A Interface to Standard System Bus

INTERRUPT SEQUENCE OUTPUTS

MCS-80/85 SYSTEM

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5-A_7 are programmed, while A_0-A_4 are automatically inserted by the 8259A. When Interval = 8 only A_6 and A_7 are programmed, while A_0-A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third $\overline{INTA}$ pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8-A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

MCS-86 SYSTEM

MCS-86 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80/85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the $\overline{INTA}$ pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in MCS-86 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5-A_{11} are unused in MCS-86 mode):

Content of Interrupt Vector Byte for MCS-86 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	A15	A14	A13	A12	A11	1	1	1
IR6	A15	A14	A13	A12	A11	1	1	0
IR5	A15	A14	A13	A12	A11	1	0	1
IR4	A15	A14	A13	A12	A11	1	0	0
IR3	A15	A14	A13	A12	A11	0	1	1
IR2	A15	A14	A13	A12	A11	0	1	0
IR1	A15	A14	A13	A12	A11	0	0	1
IR0	A15	A14	A13	A12	A11	0	0	0

PROGRAMMING THE 8259A

The 8259A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 8259A in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses. This sequence is described in Figure 1.

2. *Operation Command Words (OCWs):* These are the command words which command the 8259A to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259A anytime after initialization.

INITIALIZATION

GENERAL

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

a. The Interrupt Mask Register is cleared.

b. IR 7 input is assigned priority 7.

c. The slave mode address is set to 7.

d. Special Mask Mode is cleared and Status Read is set to IRR.

e. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80/85 system, non SFNM).

*Note: Master/Slave in ICW4 is only used in the buffered mode.

A_0	D_4	D_3	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0			0	1	0	IRR, ISR or Interrupting Level → DATA BUS (Note 1)
1			0	1	0	IMR → DATA BUS
						OUTPUT OPERATION (WRITE)
0	0	0	1	0	0	DATA BUS → OCW2
0	0	1	1	0	0	DATA BUS → OCW3
0	1	X	1	0	0	DATA BUS → ICW1
1	X	X	1	0	0	DATA BUS → OCW1, ICW2, ICW3, ICW4 (Note 2)
						DISABLE FUNCTION
X	X	X	1	1	0	DATA BUS — 3-STATE (NO OPERATION)
X	X	X	X	X	1	DATA BUS — 3-STATE (NO OPERATION)

Notes: 1 Selection of IRR, ISR or Interrupting Level is based on the content of OCW3 written before the READ operation.
2 On-chip sequencer logic queues these commands into proper sequence.

8259A Basic Operation

INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

A_{5}-A_{15} *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0-A_{15}). When the routine interval is 4, A_0-A_4 are automatically inserted by the 8259A, while A_5-A_{15} are programmed externally. When the routine interval is 8, A_0-A_5 are automatically inserted by the 8259A, while A_6-A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an MCS-86 system A_{15}-A_{11} are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. A_{10}-A_5 are ignored and ADI (Address Interval) has no effect.

LTIM: If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set — ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8 bit slave register. The functions of this register are:

a. In the master mode (either when $\overline{SP}$ = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for MCS-86 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for MCS-86) are released by it on the Data Bus.

INITIALIZATION COMMAND WORD 4 (ICW4)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for MCS-80/85 system operation, μPM = 1 sets the 8259A for MCS-86 system operation.

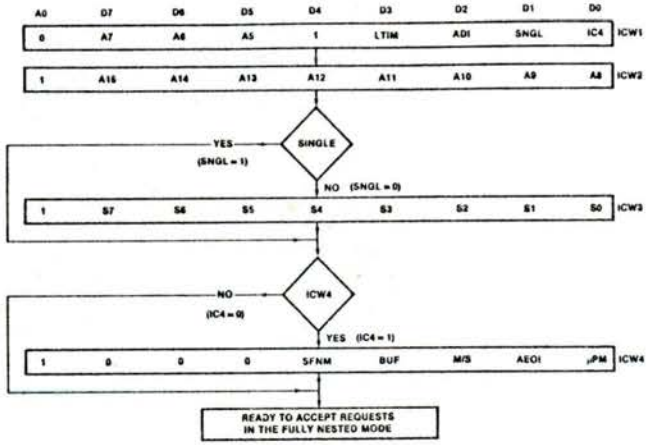
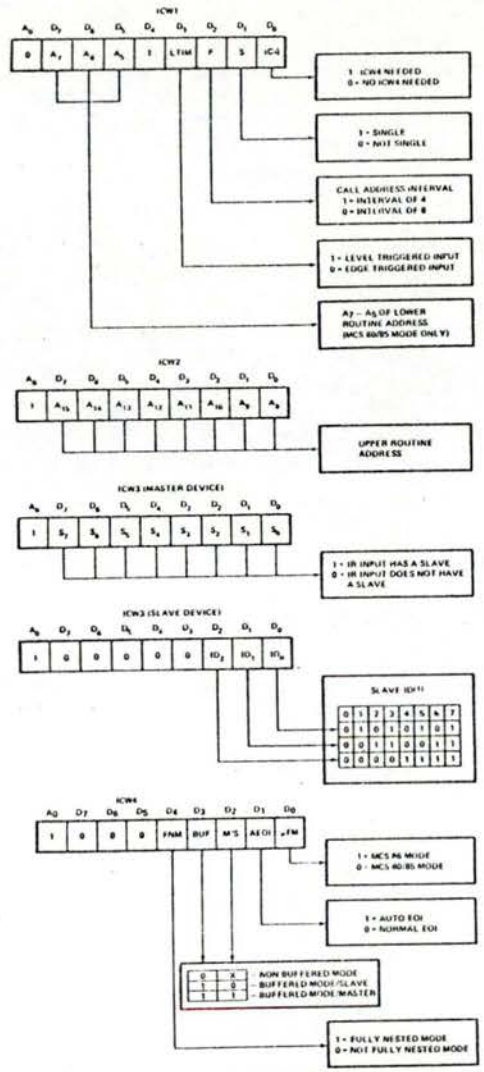


Figure 1. Initialization Sequence

PRELIMINARY



NOTE 1: SLAVED ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.
NOTE 2: X INDICATED "DON'T CARE".

Initialization Command Word Format

OPERATION COMMAND WORDS (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

OPERATION CONTROL WORDS (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SEOI	EOI	0	0	L2	L1	L0

OCW3

0	0	SSMM	SMM	0	1	P	SRIS	RIS

OPERATION CONTROL WORD 1 (OCW1)

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). M_7–M_0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited). M = 0 indicates the channel is enabled.

OPERATION CONTROL WORD 2 (OCW2)

R, SEOI, EOI — These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

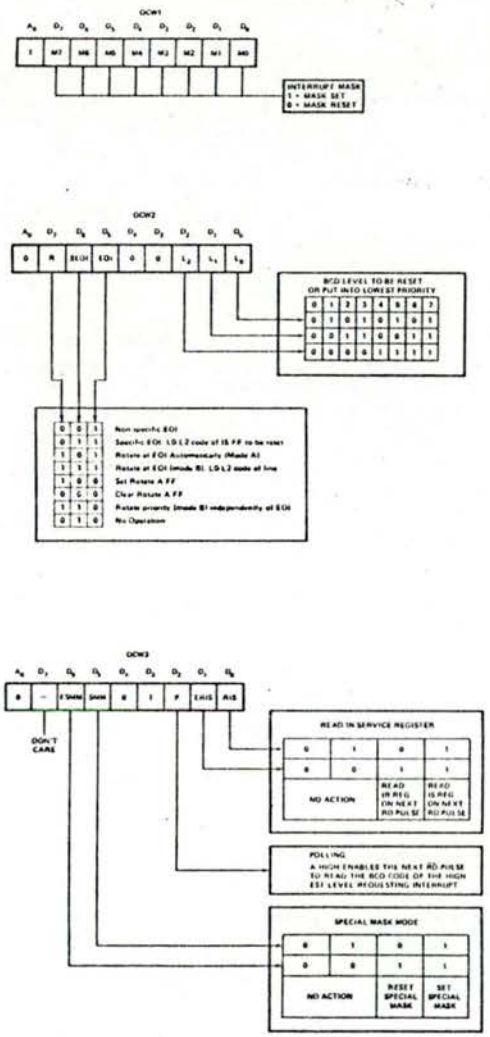
L_2, L_1, L_0 — These bits determine the interrupt level acted upon when the SEOI bit is active.

OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

PRELIMINARY
Note: This is not a final specification. Some parametric limits are subject to change.



Operation Command Word Format

INTERRUPT MASKS

Each interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

BUFFERED MODE

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (IS0-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last $\overline{INTA}$. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microporcessor internal interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

POLL

In this mode the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by programmer initiative using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The 8259A treats the next $\overline{RD}$ pulse to the 8259A (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	–	–	–	–	W2	W1	W0

W0-W2: Binary code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.

END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{INTA}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice, once for the master and once for the corresponding slave if slaves are in use.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the nested mode the highest IS level was necessarily the last level acknowledged and serviced.

However, when a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt (SEOI) must be issued which includes as part of the command the IS level to be reset. EOI is issued whenever EOI = 1, in OCW2, where L0-L2 is the binary level of the IS bit to be reset. Note that although the Rotate command can be issued together with an EOI where EOI = 1, it is not necessarily tied to it.

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

AUTOMATIC END OF INTERRUPT (AEOI) MODE

If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in MCS-86). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

To achieve automatic rotation (Rotate Mode A) within AEOI, there is a special rotate flip-flop. It is set by OCW2 with R = 1, SEOI = 0, EOI = 0, and cleared with R = 0, SEOI = 0, EOI = 0.

ROTATING PRIORITY MODE A (AUTOMATIC ROTATION) FOR EQUAL PRIORITY DEVICES

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" Status	0	1	0	1	0	0	0	0
	Lowest Priority							Highest Priority
Priority Status	7	6	5	4	3	2	1	0

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" Status	0	1	0	0	0	0	0	0
	Highest Priority							Lowest Priority
Priority Status	2	1	0	7	6	5	4	3

The Rotate command mode A is issued in OCW2 where: R = 1, EOI = 1, SEOI = 0. Internal status is updated by an End of Interrupt (EOI or AEOI command. If R = 1, EOI = 0, SEOI = 0, a "Rotate-A" flip-flop is set. This is useful in AEOI, and described under Automatic End of Interrupt.

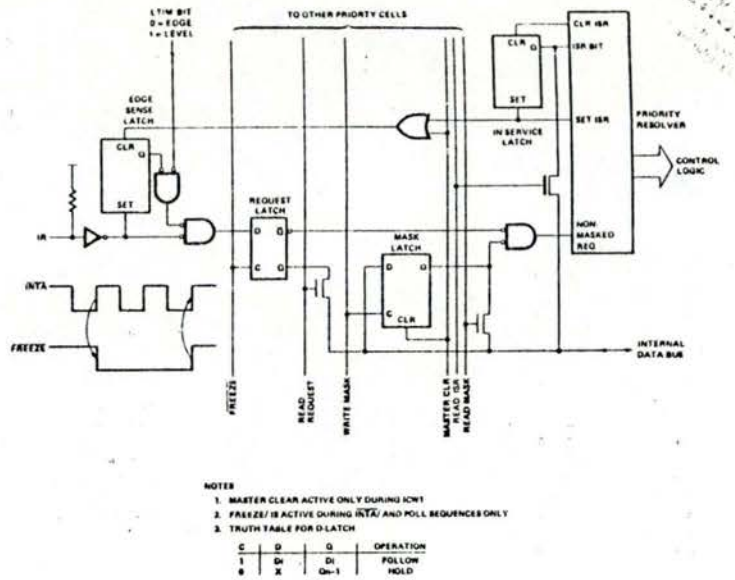
ROTATING PRIORITY MODE B (ROTATION BY SOFTWARE)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Rotate command is issued in OCW2 where: R = 1, SEOI = 1; L0-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command or independently.

PRELIMINARY



NOTES

1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/ IS ACTIVE DURING $\overline{INTA}$ AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	Di	Di	FOLLOW
0	X	Qn-1	HOLD

Priority Cell — Simplified Logic Diagram

LEVEL TRIGGERED MODE

This mode is programmed using bit 3 in ICW1.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The above figure shows a conceptual circuit to give the reader an understanding of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

READING THE 8259A STATUS

The input status of several internal registers can be read to update the user information on the system. The following registers can be read by issuing a suitable OCW3 and reading with $\overline{RD}$.

Interrupt Mask Register: 8-bit register whose content specifies the interrupt request lines being masked.

acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the $\overline{RD}$ pulse, a $\overline{WR}$ pulse is issued with OCW3 (ERIS = 1, RIS = 0.)

The ISR can be read in a similar mode when ERIS = 1, RIS = 1 in the OCW3.

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and A0 = 1.

Polling overrides status read when P = 1, ERIS = 1 in OCW3.

SUMMARY OF 8259A INSTRUCTION SET

Inst. #	Mnemonic	A0	D7	D6	D5	D4	D3	D2	D1	D0	Operation Description	
1	ICW1 A	0	A7	A6	A5	1	0	1	1	0	Byte 1 Initialization	Format = 4, single, edge triggered
2	ICW1 B	0	A7	A6	A5	1	1	1	1	0		Format = 4, single, level triggered
3	ICW1 C	0	A7	A6	A5	1	0	1	0	0		Format = 4, not single, edge triggered
4	ICW1 D	0	A7	A6	A5	1	1	1	0	0		Format = 4, not single, level triggered
5	ICW1 E	0	A7	A6	0	1	0	0	1	0	No ICW4 Required	Format = 8, single, edge triggered
6	ICW1 F	0	A7	A6	0	1	1	0	1	0		Format = 8, single, level triggered
7	ICW1 G	0	A7	A6	0	1	0	0	0	0		Format = 8, not single, edge triggered
8	ICW1 H	0	A7	A6	0	1	1	0	0	0		Format = 8, not single, level triggered
9	ICW1 I	0	A7	A6	A5	1	0	1	1	1	Byte 1 Initialization ICW4 Required	Format = 4, single, edge triggered
10	ICW1 J	0	A7	A6	A5	1	1	1	1	1		Format = 4, single, level triggered
11	ICW1 K	0	A7	A6	A5	1	0	1	0	1		Format = 4, not single, edge triggered
12	ICW1 L	0	A7	A6	A5	1	1	1	0	1		Format = 4, not single, level triggered
13	ICW1 M	0	A7	A6	0	1	0	0	1	1		Format = 8, single, edge triggered
14	ICW1 N	0	A7	A6	0	1	1	0	1	1		Format = 8, single, level triggered
15	ICW1 O	0	A7	A6	0	1	0	0	0	1		Format = 8, not single, edge triggered
16	ICW1 P	0	A7	A6	0	1	1	0	0	1		Format = 8, not single, level triggered
17	ICW2	1	A15	A14	A13	A12	A11	A10	A9	A8	Byte 2 initialization	
18	ICW3 M	1	S7	S6	S5	S4	S3	S2	S1	S0	Byte 3 initialization — master	
19	ICW3 S	1	0	0	0	0	0	S2	S1	S0	Byte 3 initialization — slave	
20	ICW4 A	1	0	0	0	0	0	0	0	0	No action, redundant	
21	ICW4 B	1	0	0	0	0	0	0	0	1	Non-buffered mode, no AEOI, MCS-86	
22	ICW4 C	1	0	0	0	0	0	0	1	0	Non-buffered mode, AEOI, MCS-80/85	
23	ICW4 D	1	0	0	0	0	0	0	1	1	Non-buffered mode, AEOI, MCS-86	
24	ICW4 E	1	0	0	0	0	0	1	0	0	No action, redundant	
25	ICW4 F	1	0	0	0	0	0	1	0	1	Non-buffered mode, no AEOI, MCS 86	
26	ICW4 G	1	0	0	0	0	0	1	1	0	Non-buffered mode, AEOI, MCS 80/85	
27	ICW4 H	1	0	0	0	0	0	1	1	1	Non-buffered mode, AEOI, MCS-86	
28	ICW4 I	1	0	0	0	0	1	0	0	0	Buffered mode, slave, no AEOI, MCS-80/85	
29	ICW4 J	1	0	0	0	0	1	0	0	1	Buffered mode, slave, no AEOI, MCS-86	
30	ICW4 K	1	0	0	0	0	1	0	1	0	Buffered mode, slave, AEOI, MCS 80/85	
31	ICW4 L	1	0	0	0	0	1	0	1	1	Buffered mode, slave, AEOI, MCS-86	
32	ICW4 M	1	0	0	0	0	1	1	0	0	Buffered mode, master, no AEOI, MCS 80/85	
33	ICW4 N	1	0	0	0	0	1	1	0	1	Buffered mode, master, no AEOI, MCS-86	
34	ICW4 O	1	0	0	0	0	1	1	1	0	Buffered mode, master, AEOI, MCS 80/85	
35	ICW4 P	1	0	0	0	0	1	1	1	1	Buffered mode, master, AEOI, MCS-86	
36	ICW4 NA	1	0	0	0	1	0	0	0	0	Fully nested mode, MCS-80, non-buffered, no AEOI	
37	ICW4 NB	1	0	0	0	1	0	0	0	1	ICW4 NB through ICW4 ND are identical to ICW4 B through ICW4 D with the addition of Fully Nested Mode	
38	ICW4 NC	1	0	0	0	1	0	0	1	0		
39	ICW4 ND	1	0	0	0	1	0	0	1	1		
40	ICW4 NE	1	0	0	0	1	0	1	0	0	Fully Nested Mode, MCS 80/85, non-buffered, no AEOI	
41	ICW4 NF	1	0	0	0	1	0	1	0	1	ICW4 NF through ICW4 NP are identical to ICW4 F through ICW4 P with the addition of Fully Nested Mode	
42	ICW4 NG	1	0	0	0	1	0	1	1	0		
43	ICW4 NH	1	0	0	0	1	0	1	1	1		
44	ICW4 NI	1	0	0	0	1	1	0	0	0		
45	ICW4 NJ	1	0	0	0	1	1	0	0	1		
46	ICW4 NK	1	0	0	0	1	1	0	1	0		
47	ICW4 NL	1	0	0	0	1	1	0	1	1		
48	ICW4 NM	1	0	0	0	1	1	1	0	0		
49	ICW4 NN	1	0	0	0	1	1	1	0	1		
50	ICW4 NO	1	0	0	0	1	1	1	1	0		
51	ICW4 NP	1	0	0	0	1	1	1	1	1		
52	OCW1	1	M7	M6	M5	M4	M3	M2	M1	M0	Load mask register, read mask register	
53	OCW2 E	0	0	0	1	0	0	0	0	0	Non-specific EOI	
54	OCW2 SE	0	0	1	1	0	0	L2	L1	L0	Specific EOI. L0-L2 code of IS FF to be reset	
55	OCW2 RE	0	1	0	1	0	0	0	0	0	Rotate at EOI Automatically (Mode A)	
56	OCW2 RSE	0	1	1	1	0	0	L2	L1	L0	Rotate at EOI (mode B). L0-L2 code of line	
57	OCW2 R	0	1	0	0	0	0	0	0	0	Set Rotate A FF	
58	OCW2 CR	0	0	0	0	0	0	0	0	0	Clear Rotate A FF	
59	OCW2 RS	0	1	1	0	0	0	L2	L1	L0	Rotate priority (mode B) independently of EOI	
60	OCW3 P	0	0	0	0	0	1	1	0	0	Poll mode	
61	OCW3 RIS	0	0	0	0	0	1	0	1	1	Read IS register	

SUMMARY OF 8259A INSTRUCTION SET (Cont.)

Inst. #	Mnemonic	A0	D7	D6	D5	D4	D3	D2	D1	D0	Operation Description
62	OCW3 RR	0	0	0	0	0	0	1	0	1	Read request register
63	OCW3 SM	0	0	1	1	0	1	0	0	0	Set special mask mode
64	OCW3 RSM	0	0	1	0	0	1	0	0	0	Reset special mask mode

Note: 1. In the master mode $\overline{SP}$ pin = 1, in slave mode $\overline{SP}$ = 0

Cascading

The 8259A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

A typical MCS-80/85 system is shown in Figure 2. The master controls, through the 3 line cascade bus, which one of the slaves will release the corresponding address.

As shown in Figure 2, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of INTA. (Byte 2 only for MCS-86). The IR0 input should not be connected to a slave 8259A unless IR1-IR7 also have slaves attached.

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first $\overline{INTA}$ pulse to the trailing edge of the third pulse. It is obvious that each 8259A in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select ($\overline{CS}$) input of each 8259A.

The cascade lines of the Master 8259A are activated for any interrupt input, even if no slave is connected to that input.

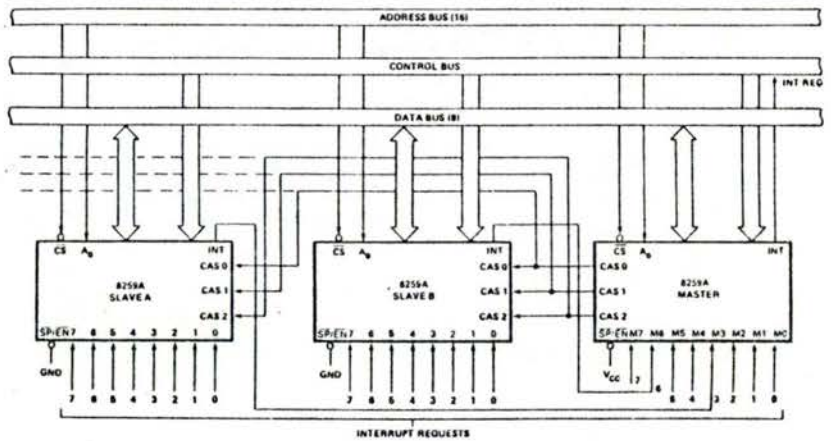


Figure 2. Cascading the 8259A

PIN FUNCTIONS

Name	I/O	Pin #	Function
V_{CC}		28	+5V supply.
GND		14	Ground.
D_{0-7}	I/O	11-4	Bidirectional data bus, used for: a) programming the mode of the 8259A (programming is done by software); b) the microprocessor can read the status of the 8259A; c) the 8259A will send vectoring data to the microprocessor when an interrupt is acknowledged.
IR_{0-7}	I	18-25	Interrupt Requests: These are asynchronous inputs. A positive-going edge will generate an interrupt request. Thus a request can be generated by raising the line and holding it high until acknowledged, or by a negative pulse. In level triggered mode, no edge is required. These lines are active HIGH.
$\overline{RD}$	I	3	Read (generally from 8228 in MCS-80 system or from 8086 in MCS-86 system).
$\overline{WR}$	I	2	Write (generally from 8228 in MCS-80 sytem or from 8086 in MCS-86 system).
$\overline{INTA}$	I	26	Interrupt Acknowledge (generally from 8228 in MCS-80 system, 8086 in MCS-86 system). The 8288 generates three distinct $\overline{INTA}$ pulses when a CALL is inserted, the 8086 produces two distinct $\overline{INTA}$ pulses during an interrupt cycle.
$\overline{CS}$	I	1	Chip Select: $\overline{RD}$ and $\overline{WR}$ are enabled by Chip Select, whereas Interrupt Acknowledge is independent of Chip Select.
A0	I	27	Usually the least significant bit of the microprocessor address output. When A0 = 1 the Interrupt Mask Register can be loaded or read. When A0 = 0 the 8259A mode can be programmed or its status can be read. $\overline{CS}$ is active LOW.
INT	O	17	Goes directly to the microprocessor interrrupt input. This output will have high V_{OH} to match the 8080 3.3V V_{IH}. INT is active HIGH.
C0-C2	I/O	12 13 15	Three cascade lines, outputs in master mode and inputs in slave mode. The master issues the binary code of the acknowledged interrupt level on these lines. Each slave compares this code with its own.
$\overline{SP}/\overline{EN}$	I/O	16	$\overline{SP}/\overline{EN}$ is a dual function pin. In the buffered mode $\overline{SP}/\overline{EN}$ is used to enable bus transceivers ($\overline{EN}$). In the non-buffered mode $\overline{SP}/\overline{EN}$ determines if this 8259A is a master or a slave. If $\overline{SP}$ = 1 the 8259A is master; $\overline{SP}$ = 0 indicates a slave.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... −40°C to 85°C
Storage Temperature ..... −65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1 Watt

*COMMENT
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C V_{CC} = 5V ± 5% (8259A-8) V_{CC} = 5V ± 10% (8259A)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	−.5	.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + .5V	V	
V_{OL}	Output Low Voltage		.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
$V_{OH(INT)}$	Interrupt Output High Voltage	3.5 2.4		V V	I_{OH} = −100 μA I_{OH} = −400 μA
I_{LI}	Input Load Current		10	μA	V_{IN} = V_{CC} to 0V
I_{LOL}	Output Leakage Current		−10	μA	V_{OUT} = 0.45V
I_{LOH}	Output Leakage Current		10	μA	V_{OUT} = V_{CC}
I_{CC}	V_{CC} Supply Current		85	mA	

8259A A.C. CHARACTERISTICS

T_A = 0°C to 70°C V_{CC} = 5V ± 5% (8259A-8) V_{CC} = 5V ± 10% (8259A)

TIMING REQUIREMENTS

Symbol	Parameter	8259A-8 Min.	8259A-8 Max.	8259A Min.	8259A Max.	Units	Test Conditions
TAHRL	A0/$\overline{CS}$ Setup to $\overline{RD}$/$\overline{INTA}$↓	50		0		ns	
TRHAX	A0/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$↑	5		0		ns	
TRLRH	$\overline{RD}$ Pulse Width	420		235		ns	
TAHWL	A0/$\overline{CS}$ Setup to$\overline{WR}$↓	50		0		ns	
TWHAX	A0/$\overline{CS}$ Hold after $\overline{WR}$↑	20		0		ns	
TWLWH	$\overline{WR}$ Pulse Width	400		290		ns	
TDVWH	Data Setup to $\overline{WR}$↑	300		240		ns	
TWHDX	Data Hold after $\overline{WR}$↑	40		0		ns	
TJLJH	Interrupt Request Width (Low)	100		100		ns	See Note 1
TCVIAL	Cascade Setup to Second or Third $\overline{INTA}$↓ (Slave Only)	55		55		ns	
TRHRL	End of $\overline{RD}$ to Next Command	160		160		ns	
TWHRL	End of $\overline{WR}$ to Next Command	190		190		ns	

Note: 1. This is the low time required to clear the input latch in the edge triggered mode.

TIMING RESPONSES

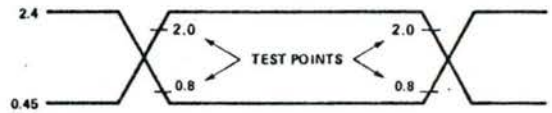
Symbol	Parameter	8259A-8 Min.	8259A-8 Max.	8259A Min.	8259A Max.	Units	Test Conditions
TRLDV	Data Valid from $\overline{RD}$/$\overline{INTA}$↓		300		200	ns	C of Data Bus Max. test C = 100 pF Min. test C = 15 pF C_{INT} = 100 pF C_{ENABLE} = 15pF
TRHDZ	Data Float after $\overline{RD}$/$\overline{INTA}$↑	10	200		100	ns	
TJHIH	Interrupt Output Delay		400		350	ns	
TIALCV	Cascade Valid from First $\overline{INTA}$↓ (Master Only)		565		565	ns	
TRLEL	Enable Active from $\overline{RD}$↓ or $\overline{INTA}$↓		160		125	ns	
TRHEH	Enable Inactive from $\overline{RD}$↑ or $\overline{INTA}$↑		325		150	ns	
TAHDV	Data Valid from Stable Address		350		200	ns	
TCVDV	Cascade Valid to Valid Data		300		300	ns	

CAPACITANCE

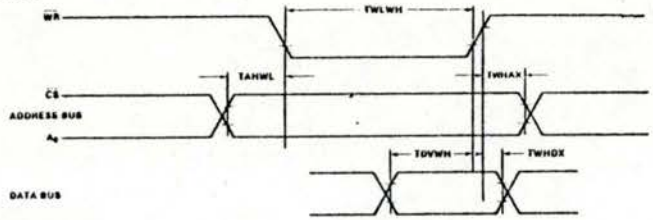
T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

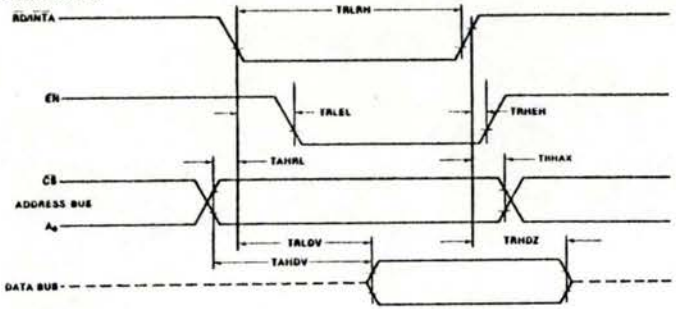
Input and Output Waveforms for A.C. Tests



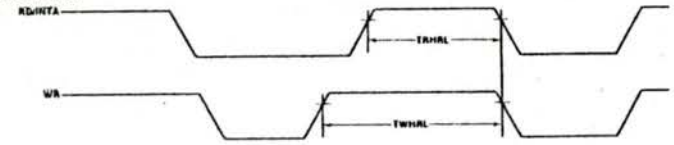
WRITE MODE



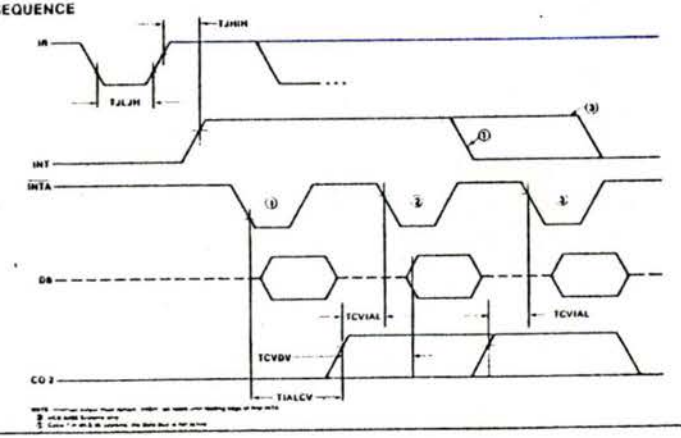
READ/INTA MODE



OTHER TIMING



INTA SEQUENCE



BUMP
0034308 10
*FM B16/1981/05