

11/87
الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLICUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

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

ECOLE NATIONALE POLYTECHNIQUE

المكتبة
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT : D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

En vue de l'obtention d'un diplôme d'ingénieur d'Etat

SUJET

**Conception et Réalisation
d'un Système Minimal à Base
du Microprocesseur
MC 6809**

Proposé par :

Mme HAMMAMI

Etudié par :

Mr KHOULALENE A.
Mr HAMMOU T.

Dirigé par :

Mme HAMMAMI

PROMOTION : JANVIER 1987

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

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

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

ECOLE NATIONALE POLYTECHNIQUE

المكتبة الوطنية للتعليم والبحث العلمي
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT : D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

En vue de l'obtention d'un diplôme d'ingénieur d'Etat

SUJET

Conception et Réalisation d'un Système Minimal à Base du Microprocesseur MC 6809

Proposé par :

Mme HAMMAMI

Etudié par :

Mr KHOULALENE A.

Mr HAMMOU T.

Dirigé par :

Mme HAMMAMI

PROMOTION : JANVIER 1987

E.N.P. 10, Avenue Hacén Badi - EL-HARRACH — ALGER

DEDICACES

الدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Je dédie ce modeste travail à :

- La mémoire de mon père
- La mémoire de mon oncle Mohand
- Ma chère mère
- Mes frères et sœurs
- Toute la famille
- Tous mes amis (es)

LOUHAB

Je dédie ce modeste travail à :

- Mes chers parents
- Mon frère
- Mes sœurs
- Tous mes amis(es)

TEWFIK

REMERCIEMENTS

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Au terme de cette étude, nous tenons à remercier
vivement :

Notre promotrice M^{me} HAMMAMI pour ses conseils et orienta-
tions qui nous ont été d'un grand apport pour mener à bien
ce travail.

Nous remercions également pour son aide précieuse :
M^E HAMMAMI , chef du département HARD à l'ENSI.

Nous remercions d'une manière générale tous les
professeurs qui ont contribué à notre formation.

Nos respects aux membres du jury qui nous feront
l'honneur d'apprécier notre travail.



4°- Traitement de \overline{IRQ}	20
5°- " " "l'interruption SWI	22
6°- Les interruption SWI2 et SWI3	22
7°- L'instruction attente d'interruption " CWAY"	25
8°- " " synchronisation interne " SYNC"	25
9°- Instruction de retour d'interruption RTI	25
B Les interfaces d'entrées / Sorties	29
B-1 Le PIA MC 6821 .	
B1 -1 signification des différentes broches	30
B1-2 Organisation interne du PIA.....	32
B1-3 Fonctionnement du PIA.....	34
B2- Interface asynchrone à E/S series: l' ACIA ^{MC} 6850	34
B2-1 Organisation externe de l'ACIA	34
B2-2 " interne de l'ACIA.....	36
B2-3 Fonctionnement de l'ACIA.....	39
CHAPITRE III -REALISATION DU SYSTEME	42
A - l'unité centrale (le MC 6809)	42
B - Le système d'adressage	44
1)° Par sélection linéaire d'adresse	44
2)° Par decodage complet du bus d'adresses.....	44
C - le système mémoire	48
1°) LES mémoires vives (RAM)	48
2°) " " mortes	50
3°) Les CIRCUITS mémoires utilisés.....	50
D-Les interfaces d'E/S à l'usage de l'utilisateur.....	52
1° L'interface à E/S parallèles (MC 6821°).....	52
2° " " " series(MC 6850).....	52
E-Le système de gestion du clavier et des afficheurs.....	55
1° Le système d'affichage.....	55
2° " " de gestion du clavier.....	55
F-Utilisation du clavier	57
CONCLUSION.....	59
ANNEXES.....	60

La conception d'un système à microprocesseur passe avant tout par la définition du cahier de charges qui est une évaluation des besoins et des contraintes du système.

Cette évaluation est conditionnée par les données du problème et les résultats attendus. Ces besoins sont généralement classés en deux catégories :

- Les besoins en "matériel" : Ce sont en général les circuits d'Entrées/Sorties à commander et la capacité mémoire à utiliser.
- Les besoins en "logiciels" : C'est le nombre et la nature des fonctions qui seront réalisées par le système.

Parmi les contraintes rencontrées en général, nous avons :

- La vitesse de traitement qui caractérise le temps d'exécution des fonctions à réaliser (lecture et écriture des données, opérations arithmétiques, envoi des signaux de commande etc...).
- L'encombrement des circuits et du système ; sa consommation en énergie et son coût.

Pour notre réalisation, destinée à une utilisation au laboratoire dans le but d'initier les étudiants à l'assembleur du microprocesseur MC 6809 et aux systèmes à microprocesseurs en général, nous avons prévu une capacité mémoire de 4 K octets (*) en mémoire morte ROM et de 4 K octets en RAM (Mémoire vive) ainsi que trois ^{CIRCUITS} d'interface dont l'un sériel et deux parallèles. Un des deux circuits d'interface parallèles est utilisé pour la gestion du clavier et des afficheurs ; le deuxième (parallèle) ainsi que le sériel sont destinés à l'usage de l'utilisateur.

Une étude détaillée de ces circuits sera présentée au niveau des chapitres qui vont suivre.

Toutefois nous allons axer notre travail sur le côté "matériel" de la réalisation car la partie "logiciel" fait l'objet d'un autre projet.

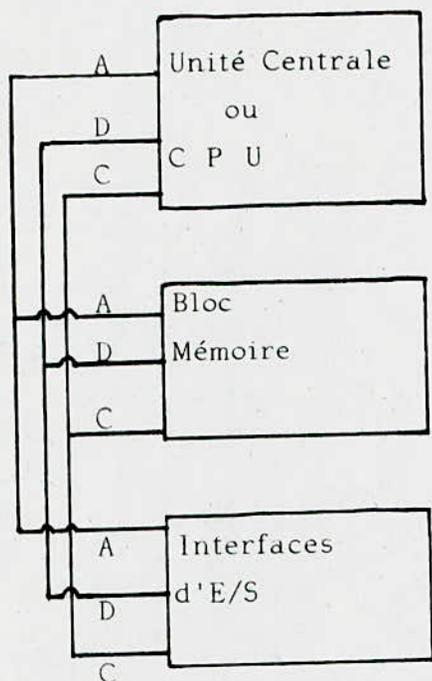
* 1 octet = 1 mot de huit bits en binaire.

CHAPITRE 1

Comme tout système à microprocesseur notre carte est composée de trois blocs fondamentaux qui sont :

- L'unité centrale ou CPU
- Le bloc mémoire
- Les interfaces d'Entrées/Sorties.

Ces blocs sont liés entre eux selon le schéma synoptique suivant :



A, D et C sont respectivement les bus d'adresses, de données et de contrôle.

A- L'unité centrale ou CPU : C'est la partie " pensante " du système, elle est réalisée par le microprocesseur 6809. C'est à son niveau qu'ont lieu les opérations arithmétiques et logiques;

B- Le bloc mémoire : C'est un ensemble de circuits capables de stocker des informations, ceux-ci sont de deux types : Ceux capables de stocker temporairement les informations qui sont appelés les circuits de mémoire vive (RAM) dans lesquels nous pouvons lire et écrire ; et ceux capables de stocker des informations de façon permanente, ce sont les circuits de mémoire morte (ROM, EPROM, PROM) dans lesquels nous ne pourrions que lire les informations.

Pour la réalisation de notre système nous avons utilisé, deux boîtiers EPROM qui sont les MC 2516 d'une capacité de 2 K.O chacun et deux boîtiers RAM, les **SJ** 2128 d'une capacité de 2 K.O chacun.

Nous avons prévu une possibilité d'extension mémoire externe en cas de nécessité.

... / ...

C-Les interfaces d'Entrées/Sorties : Ce sont les circuits permettant la communication entre l'unité centrale et le monde extérieur . Nous avons un interface à E/S parallèles ; le PIA MC 6821 destiné à la gestion du clavier et des afficheurs , un autre PIA et un interface à E/S séries l'ACIA MC 6850 dont les fonctions seront choisies par l'utilisateur.

Un clavier de 24 touches à effleurement est utilisé pour l'introduction des programmes et des données utiles ; un système de 6 afficheurs à 7 segments est aussi utilisé pour permettre l'affichage des différentes adresses et de leurs contenus ainsi que le contenu des registres internes du microprocesseur.

Pour une bonne exploitation des différents blocs, nous avons utilisé un système de selection de boitiers, des buffers pour l'amplification des signaux de bus destinés aux connections avec les systèmes extérieurs et un "BAUD RATE" qui détermine la vitesse de transmission /reception de l'ACIA.

Nous pouvons résumer le contenu de notre carte par le schéma synoptique de la "FIG-1".

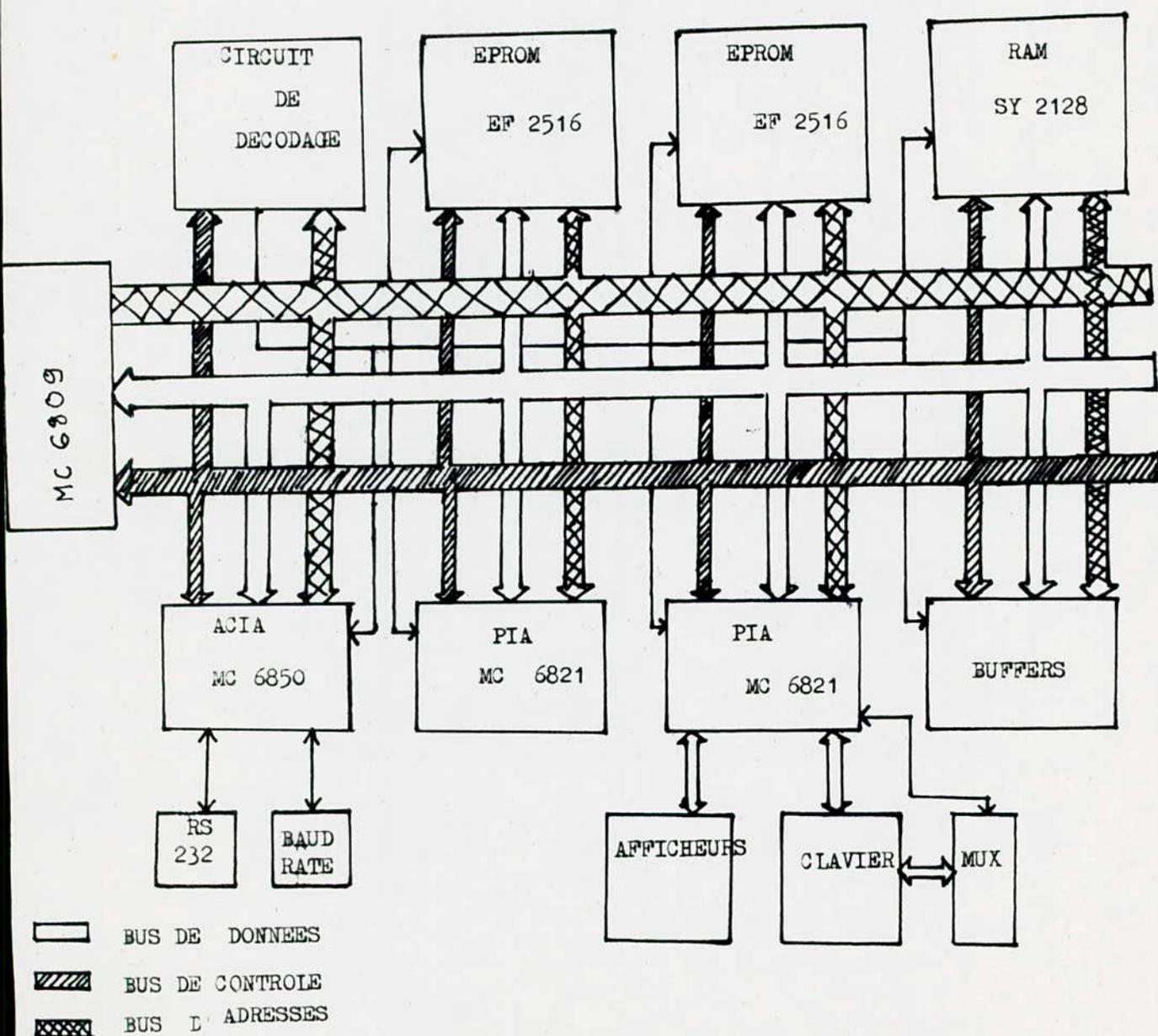


FIG- 1 SCHEMA SYNOPTIQUE DU SYSTEME REALISE

après la présentation de la carte dans le chapitre précédent, ce chapitre sera consacré à l'étude détaillée des différents circuits utilisés dans la réalisation, de même que nous définirons le rôle de chacun d'eux.

A- L'UNITE CENTRALE :

L'unité centrale est réalisée par le microprocesseur MC 6809 dont l'architecture interne le mode d'adressage et le jeu d'instruction en font un processeur orienté logiciel.

Il fait partie de la famille 6800 de MOTOROLA, il est compatible TTL et est compatible avec tous les circuits périphériques du MC 6800.

Ce processeur se présente sous deux versions, l'une à horloge interne le MC 6809 l'autre à horloge externe le MC 6809 E. Dans notre étude nous nous limiterons à l'examen du MC 6809 car c'est ce dernier que nous avons utilisé.

A1- ORGANISATION MATERIELLE DU 6809 :

Le MPU 6809 est fabriqué en technologie MOS à canal N, il est monotension (+5V) et est présenté dans un boîtier DIL à 40 broches.

La figure 2 donne le schéma de brochage du MC 6809.

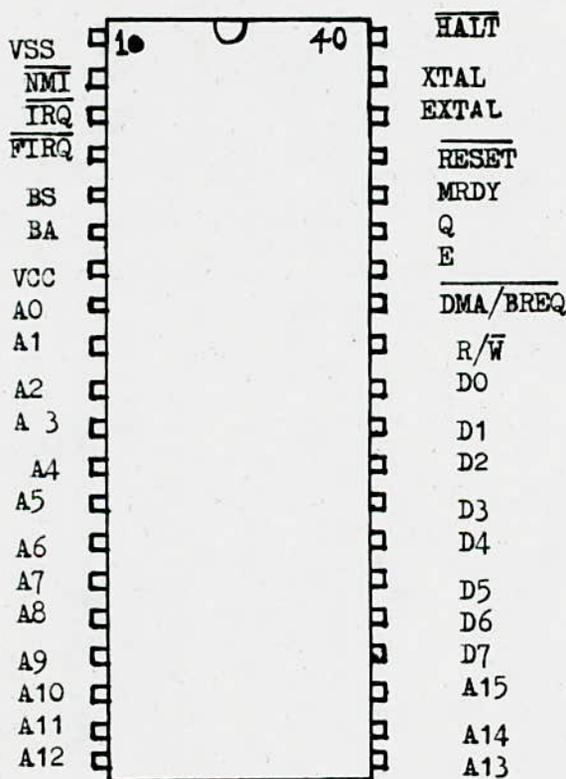


FIG-2 : SCHEMA DE BROCHAGE DU MC 6809

A1-1 SIGNIFICATION DES DIFFERENTES BROCHES :

- L'ALIMENTATION :

Une tension unique de $5\text{ V} \pm 5\%$ est utilisée pour l'alimentation, elle est appliquée à l'entrée Vcc qui correspond à la broche n° 7 du boîtier. La broche n° 1 qui correspond à Vss est liée à la masse.

- LE BUS D'ADRESSES:

Le MPU communique avec le bus d'adresses par l'intermédiaire de 16 sorties (A0 - A15) correspondant aux broches allant de 8 à 23, sa capacité d'adressage est donc de 64 K.O.

Ces lignes sont unidirectionnelles et comportent des drivers en logique trois états (état haut, état bas et état haute impédance) pouvant piloter chacune une charge TTL plus 8 circuits de la famille 6800. On notera que le bus est positionné à l'adresse FFFF avec la ligne R/\overline{W} à l'état haut et l'entrée BS à l'état bas lorsqu'il n'est pas occupé par le processeur pour gérer un transfert de données.

- LE BUS DE DONNEES :

La communication avec le bus de données s'effectue grâce à 8 lignes d'E/S (D0 - D7) correspondant aux broches du boîtier allant de 24 à 31. Ces lignes sont bidirectionnelles et peuvent piloter chacune une charge TTL plus 8 circuits de la famille 6800 et comportent des drivers en logique 3 états. Le changement du sens de fonctionnement se fait sans problèmes grâce à la période $E.\overline{Q}$ (horloge) qui maintient le bus à l'état haute impédance.

- LE BUS DE CONTROLE :

Il contient tous les signaux du MPU qui définissent son mode de fonctionnement et régissent son dialogue avec les circuits qui l'entourent (mémoire, interfaces etc...). Ces lignes permettent aussi de contrôler le fonctionnement du MPU et de coordonner ses échanges avec les autres circuits.

Les broches communiquant avec le bus de contrôle sont :

* \overline{NMI} : (no masquable interruption)

C'est une ligne d'entrée de demande d'interruption non masquable par logiciel et est activée par un niveau bas sur la broche n°2 du boîtier.

* \overline{TRQ} : (interrupt request)

C'est une ligne d'entrée de demande d'interruption masquable par logiciel qui est activée par un niveau bas sur la broche n°3 du boîtier.

* \overline{FIRQ} : (fast interrupt request)

C'est une ligne de demande d'interruption rapide masquable par logiciel qui est activé par un niveau bas sur la broche n°4 du boîtier.

Notons que l'étude du fonctionnement en interruption du MPU sera traitée ultérieurement.

* BA : (bus available)

Ce signal à l'état haut indique la mise à l'état haute impédance des bus de données et d'adresses ainsi que ^{de} la ligne de commande de lecture/écriture.

Il correspond à la broche n°6 du boîtier.

* BS : (bus state)

Cette ligne, en conjonction avec la ligne BA indique l'état de fonctionnement du MPU. Elle correspond à la broche n° 5 du MPU. Les configurations possibles des états des lignes BA et BS et leurs significations sont données par le tableau suivant :

BA	BS	FONCTIONNEMENT DU MPU
0	0	Normal (le MPU gère lui même les bus d'adresses et de données)
0	1	Reconnaissance d'interruption (recherche matérielle des vecteurs d'interruption $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, $\overline{\text{FIRQ}}$, SWI, SWI 2, SWI 3)
1	0	Reconnaissance de synchronisation externe (le MPU attend une synchronisation externe sur une ligne d'interruption). Dans ce cas les bus sont en état haute impédance.
1	1	Arrêt ou bus accordé (le MPU autorise la gestion des bus d'adresses de données par un circuit annexe). Dans ce cas aussi les bus sont dans l'état haute impédance.

* $\overline{\text{R/W}}$: (read/wright)

Cette sortie précise le sens des transferts de données entre le MPU et les circuits associés, elle est en logique 3 états et correspond à la broche n° 32 du MPU.

$\overline{\text{R/W}} = 1$: Indique que le MPU est en phase de lecture donc les lignes(D0 - D 7) du bus de données sont des entrées.

$\overline{\text{R/W}} = 0$: Indique que le MPU est en phase d'écriture donc les lignes(D0-D7) sont des sorties.

* $\overline{\text{DMA}} / \overline{\text{BREQ}}$: (direct acces memory / bus request)

Cette entrée permet de faire un accès direct mémoire ou un rafraichissement mémoire (cas des mémoires dynamiques) en suspendant l'utilisation des bus par le MPU. Sa lecture se fait sur un front descendant de Q et est activée par un niveau bas sur la broche n° 33 du MPU.

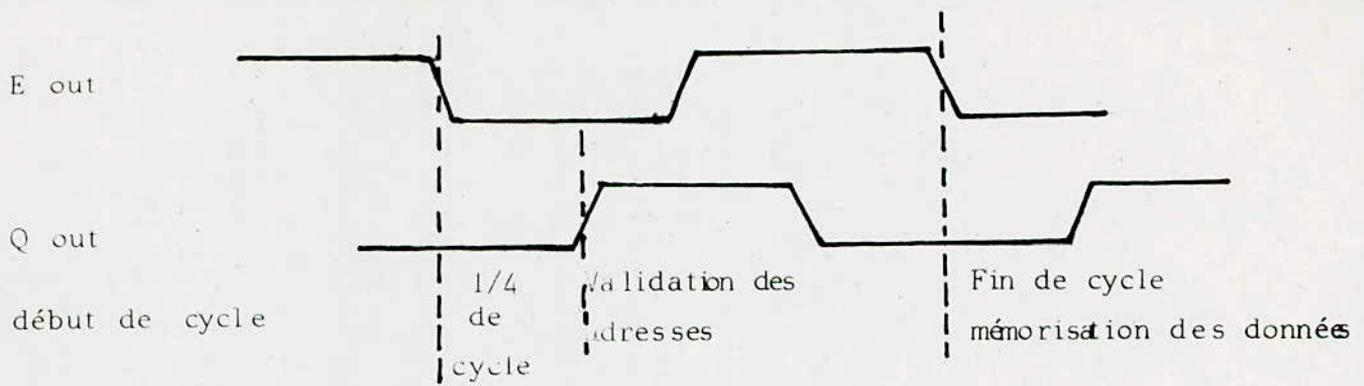
Une activation de cette entrée met, après avoir terminé l'instruction en cours, BA et BS à l'état haut, alors le dispositif demandeur disposera de 15 microcycles avant que le MPU ne reprenne les bus pour son auto-rafraichissement, necessitant un cycle précédé et suivi par un cycle mort.

* SORTIES D'HORLOGES :

Le MPU dispose de deux sorties d'horloges pour synchroniser ses échanges avec les circuits qui l'entourent. Ces sorties sont :

- La sortie E : Ce signal constitue le timing des bus, sa fréquence est celle de base du microprocesseur. Cette sortie correspond à la broche n° 34 du MPU.

- La sortie Q : C'est une sortie d'horloge en quadrature avec E. Elle correspond à la broche n° 35 du MPU. Les adresses seront validées sur un front montant de Q par contre les données seront verouillées sur un front descendant de E.



* MRDY : (allongement de l'horloge E)

Cette entrée de commande est utilisée pour l'allongement du signal d'horloge E afin de permettre l'emploi de mémoires lentes. L'allongement de E est un multiple du quart de cycle, sa valeur maximale est de 10 microsecondes. Elle correspond à la broche n° 36 du MPU et est mise à l'état haut lorsqu'elle n'est pas utilisée.

* RESET :

Cette entrée activée par un niveau bas sur la broche n° 37 du MPU, permet la réinitialisation complète de ce dernier et son démarrage après une mise sous tension ou une panne d'alimentation. Pour s'assurer que tous les périphériques de la famille 6800 ont terminé leur phase de mise sous tension, lorsque le 6809 commence le programme d'initialisation cette entrée possède un trigger de schmitt dont la tension de seuil est supérieure à celle des périphériques. Ainsi l'entrée RESET sera active plus rapidement sur les périphériques et un simple circuit RC peut être utilisé pour initialiser l'ensemble du système.

* L'ENTREE HALT :

Cette entrée ^{est} activée par un niveau bas sur la broche n° 40 du MPU et permet alors l'interruption matérielle du déroulement d'un programme. Quand elle revient à l'état haut le processeur reprend l'exécution du programme sans qu'il y ait perte d'informations.

Le passage de HALT à l'état bas entraîne après l'exécution de l'instruction en cours le positionnement de BA et BS à l'état haut, l'inhibition des demandes d'interruptions IRQ et FIRQ et l'autorisation des demandes d'accès direct mémoire. Les demandes d'interruptions non masquables RESET et NMI sont prises en compte mais leur traitement est différé.

* LES ENTREEES D'HORLOGE XTAL ET EXTAL :

Ces deux entrées sont connectées à un quartz dont la fréquence est quatre fois celle de la fréquence bus. La fréquence de fonctionnement du MPU diffère selon le type utilisé. Elle est de 1 MHz pour le 6809, de 1,5 MHz pour le 68A09 et de 2 MHz pour le 68B09. Les fréquences des quartz à utiliser avec ces microprocesseurs sont respectivement de 4 MHz, 6 MHz et 8 MHz. Notons que les deux bornes de ces quartz sont reliées à la masse à travers deux capacités dont la valeur est donnée par le constructeur. Les valeurs de ces capacités sont de 18 pF pour le 68 B 09, de 20 pF pour le 68 A 09 et de 24 pF pour le 6809.

A2- ORGANISATION INTERNE DU 6809 :

Nous étudierons dans cette partie l'ensemble des registres internes du MC 6809 ainsi que le rôle joué par chacun d'eux. Le MC 6809 est un microprocesseur 8 bits dont l'architecture interne est orientée 16 bits, ce qui fait de lui un intermédiaire puissant entre les processeurs 8 bits et les processeurs 16 bits. En effet il possède deux accumulateurs A et B qui, concatenés, forment un accumulateur D à 16 bits.

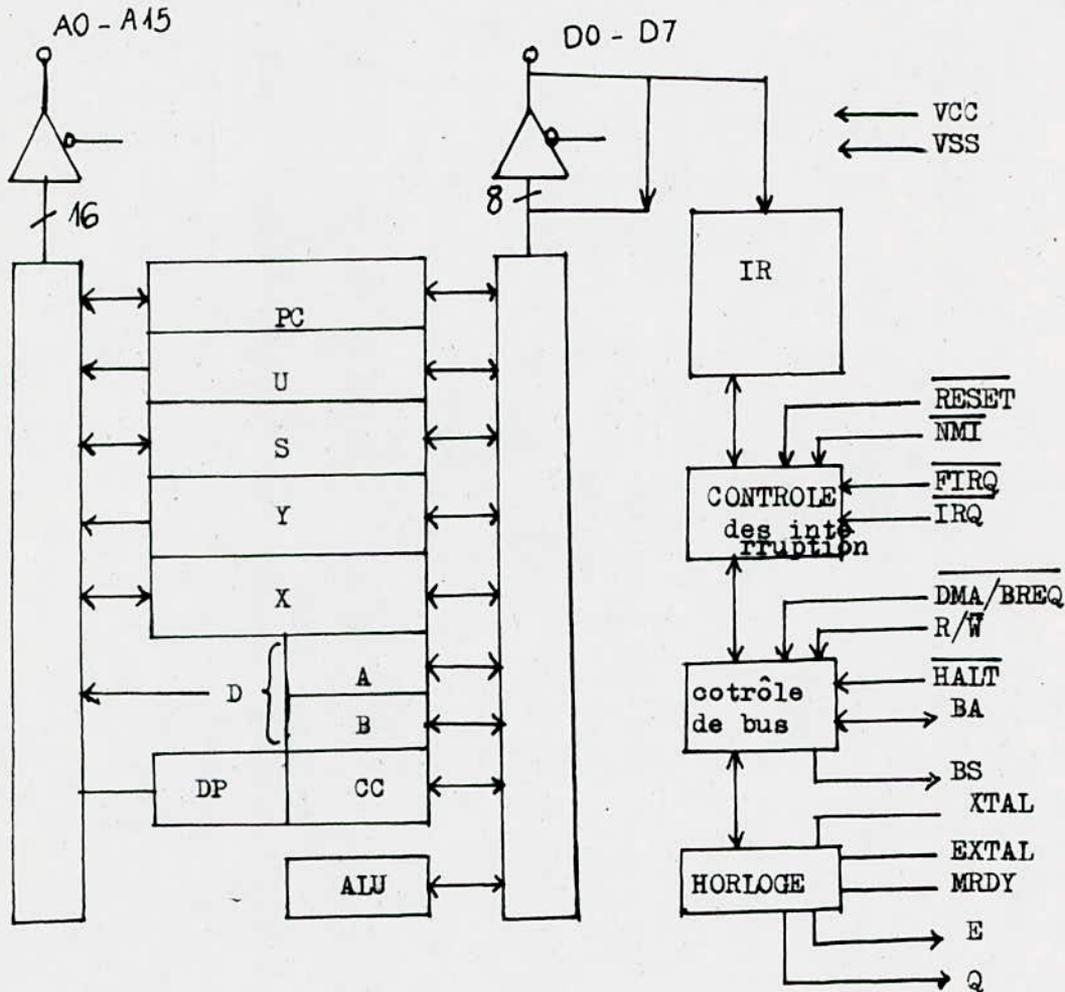


FIG-3 : SCHEMA DE L'ORGANISATION INTERNE DU 6809

Comme il apparaît sur le schéma de la FIG-3, le MC 6809 possède 9 registres internes, programmables et accessibles à l'utilisateur.

Ces registres sont :

... / ...

1- LES ACCUMULATEURS A, B ET D :

A et B sont des accumulateurs à 8 bits, pouvant être concatenés pour former D, qui est un accumulateur à 16 bits, ayant comme octet de poids forts l'accumulateur A et comme octet de poids faibles l'accumulateur B.

Les opérations arithmétiques et logiques, se font grâce aux accumulateurs A et B interchangeables, sauf pour les opérations portant sur 16 bits et pour les instructions ABX et DAA.

2 - LES REGISTRES D'INDEX X ET Y :

Ce sont deux registres de 16 bits, utilisés dans les modes d'adressages indexés, pour pointer des données se trouvant dans une position mémoire. Le contenu de ces registres peut être modifié par addition d'un déplacement codé sur 8 ou 16 bits ou d'une constante optionnelle, ou encore par une poste incrementation ou une décrémentation.

3 - LES REGISTRES POINTEURS DE PILES U ET S :

Ce sont des registres de 16 bits, pointant le sommet de la pile. Le pointeur de pile S, est utilisé par le processeur pour gérer la sauvegarde de ses registres internes, lors d'une interruption ou d'un appel de sous programme.

Le pointeur de pile U est quant à lui, réservé à l'utilisateur pour sauvegarder les données de son choix.

La FIG-4 montre la disposition des registres dans la pile dans le cas d'une sauvegarde.

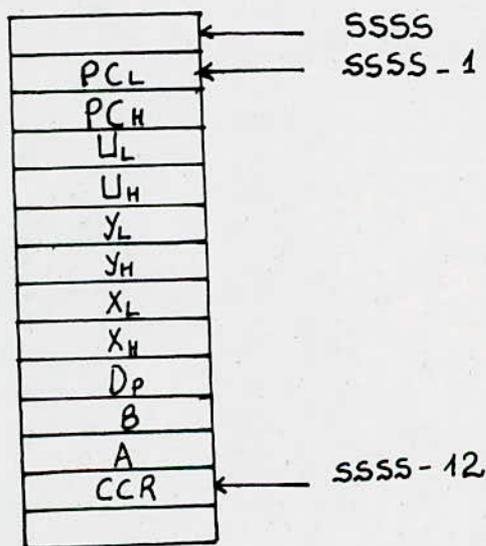


FIG-4 DISPOSITION DES REGISTRES DANS LA PILE DANS LE CAS D'UNE SAUVEGARDE.

La gestion des piles, utilisateur ou système, est identique. Leur dépilement se fait selon le mode "last in first out" (LIFO), c'est à dire que le premier registre empilé sera le dernier à être dépilé. Le dernier registre dépilé est suivi d'une incrémentation du pointeur.

Ces deux pointeurs, peuvent être aussi utilisés, comme des registres d'index.

4 -LE REGISTRE COMPTEUR PROGRAMME PC :

C'est un registre de 16 bits contenant l'adresse courante du programme. Il est initialisé à l'adresse de départ du programme et pointe en permanence l'adresse de la prochaine instruction à exécuter. Ce registre peut être utilisé dans certains cas comme index.

5 - LE REGISTRE DE PAGE DP :

C'est un registre de 8 bits, dont le contenu apparaît sur les lignes A8 à A15 du bus d'adresses, permettant d'étendre les possibilités d'adressage direct à tout l'espace mémoire en le divisant en 256 pages de 256 octets chacune et ceci sous contrôle du logiciel. Pendant le RESET son contenu est effacé pour être compatible avec le 6800.

6 - LE REGISTRE DE CODES CONDITION CCR :

C'est un registre dont les 8 bits, jouent le rôle d'indicateurs pour le microprocesseur. L'état de ces indicateurs, est positionné, pour certains d'entre eux en fonction des résultats des instructions exécutées et pour d'autres il dépend du fonctionnement en interruption.

La FIG-5 donne la disposition de ces indicateurs et leur signification.

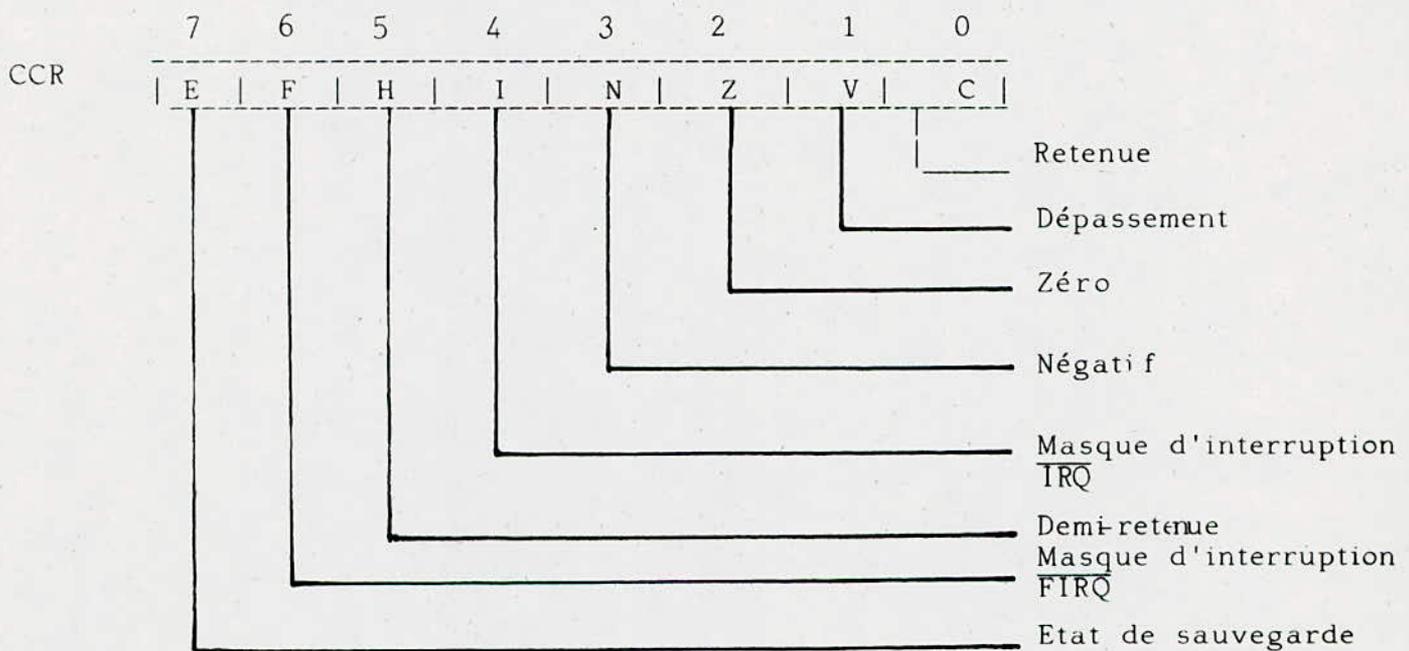


FIG-5 :SIGNIFICATION DES BITS DU CCR

* ROLE DU BIT C (b0) : (carry)

Ce bit indicateur de retenue peut jouer différents rôles suivant les opérations effectuées. Il est positionné lors d'une opération arithmétique et n'est pas affecté lors des opérations logiques et des manipulations de données.

* ROLE DU BIT V (b1) : (overflow)

Ce bit indicateur de débordement est mis à l'état haut si le résultat en complément à 2 d'une opération arithmétique déborde. Les instructions de chargements, stockages et les opérations logiques positionnent V à 0.

* ROLE DU BIT Z (b2) : (zéro)

Le bit Z est l'indicateur de zéro, il est mis à un si le résultat de l'opération précédente est nul. Il est affecté par les instructions de chargements, de stockages et les opérations arithmétiques et logiques.

* ROLE DU BIT N (b3) : (négative)

Ce bit indique un résultat négatif, il contient exactement la valeur du bit de poids forts de l'octet résultant de l'opération précédente. Un résultat négatif en complément à 2 positionne N à un.

* ROLE DU BIT I (b4) : (interrupt mask)

Lorsqu'il est à un ce bit masque les interruptions \overline{TRQ} , seules les interruptions plus prioritaires seront prises en compte. Les interruptions \overline{NMI} , \overline{FIRQ} , SWI et \overline{RESET} positionnent toutes I à un. SWI 2 et SWI 3 n'affectent pas ce dernier.

* ROLE DU BIT H (b5) : (half carry)

C'est le bit de demi retenue. Il est utilisé dans l'ALU comme indicateur de retenue du bit 3, pour les additions à 8 bits seulement (ADC ou ADD). Il est aussi utilisé dans une instruction DAA pour réaliser une opération d'ajustement décimal. L'état de cet indicateur est indéfini dans toutes les instructions de soustractions ou équivalentes.

* ROLE DU BIT F (b6) : (fast interrupt mask)

Ce bit lorsqu'il est à un masque les interruptions \overline{FIRQ} et seules les interruptions plus prioritaires seront prises en compte.

* ROLE DU BIT E (b7) :

C'est le bit indicateur de l'état de sauvegarde, mis à un il indique que l'état complet de la machine est sauvegardé (tous les registres internes du MPU sont emplilés). Il est utilisé sur un retour d'interruption RTI pour déterminer l'étendue du dépilement. Lorsqu'il est à zéro il indique que seuls les contenus du PC et du CCR sont sauvegardés.

D'une façon générale, une instruction spécifie une tâche élémentaire qui sera exécutée par la machine. Un ensemble d'instructions ordonnées dans un ordre bien établi constitue un programme. La longueur des instructions du 6809 en langage machine varie entre 1 et 5 octets binaires. L'ensemble des instructions que le MPU sait reconnaître constitue le jeu d'instruction.

La description des instructions fait intervenir deux entités caractéristiques, l'une spatiale qui est le nombre d'octets affectés à l'instruction caractérisant ainsi son encombrement mémoire, et l'autre temporelle qui est le nombre de cycles machine nécessaires à l'exécution des instructions.

Le 6809 possède un jeu d'instructions très performant qui fait de lui un processeur très puissant et très utilisé.

En effet, bien que les instructions du MC 6809 soient limitées à 59, contre 72 pour le 6800, leur conjonction avec les 9 modes d'adressage lui donne un jeu très performant de 1464 combinaisons.

Ces instructions sont de cinq types :

- Les instructions de traitement des données, regroupant les opérations arithmétiques, logiques, de décalage, de rotation, d'incréméntation, de décrémentation, de mise à zéro et de complémentation.
- Les instructions de transfert de données
- Les instructions de test et de branchement
- Les instructions opérant sur les pointeurs
- Les instructions de traitement d'interruptions

Le tableau donné à la fin de ce polycopé résume l'ensemble des instructions avec leur code opératoire, le nombre de cycles machine nécessaire à leur exécution et le nombre d'octets sur lequel elles portent, en fonction des différents modes d'adressage.

Le microprocesseur a pour principale tâche le traitement des données contenues en mémoire. Une instruction contient l'ensemble des informations nécessaires à l'accomplissement de cette tâche. Ces dernières portent sur :

- La nature de l'opération ;
- L'origine des données ou adresse de départ ;
- La destination du résultat ou adresse de stockage ;

Si l'information concernant la nature de l'opération ne dépend pas de l'adresse de la donnée ; les deux autres informations qui consistent à rechercher la donnée initiale puis à stocker le résultat final, concernent l'adresse de la donnée et dépendent de la façon dont l'utilisateur indique cette adresse.

Ce dernier peut, soit :

- Spécifier que les données se trouvent dans les registres internes du MPU ;
- Spécifier leur numéro de case dans l'espace mémoire ;
- Indiquer le déplacement en octets de l'adresse à trouver par rapport à une autre adresse de référence ;
- Indiquer une adresse intermédiaire où se trouve l'adresse effective de la donnée.

Le MC 6809 présente plusieurs possibilités pour spécifier une adresse, l'ensemble de ces possibilités constitue les modes d'adressage de ce microprocesseur.

En effet le MC 6809 possède 9 modes d'adressage différent qui sont :

1° - Le mode d'adressage immédiat :

Ce mode d'adressage permet de charger les registres internes du MPU par la valeur de l'opérande. Le code opératoire est suivi de l'opérande. La longueur de l'instruction dépend du registre sur lequel elle s'applique et varie de 2 à 4 octets mémoire.

2° - Le mode d'adressage direct :

Ce mode d'adressage permet d'accéder à l'adresse mémoire contenant l'opérande. Le code opératoire est sur 1 ou 2 octets, l'octet suivant représente le poids faible de l'adresse effective dont les 8 bits de poids fort sont donnés par le contenu du registre de page (DP).

3° - Le mode d'adressage inhérent :

L'utilisation de ce mode d'adressage est réservée aux instructions portant sur les registres internes du MPU . En effet, toutes les informations nécessaires à l'exécution de l'instruction se trouvent groupées dans le code-opération. Le champ opérande est absent de l'instruction. Il y a deux modes d'adressage inhérents :

a - Adressage inhérent simple :

Toute l'information nécessaire à l'exécution de l'instruction est contenue dans le code opératoire.

b - Adressage inhérent paramétré :

Les instructions sont sur 2 octets dont le premier constitue le code opératoire pur le second détermine les registres concernés et peut aussi masquer ou valider l'interruption.

4° - Mode d'adressage étendu :

Dans ce mode l'adresse de la donnée est contenue dans le champ opérande de l'instruction. L'adressage étendu désigne donc une adresse effective de la donnée dans l'espace mémoire.

5° - Le mode d'adressage étendu indirect :

Dans ce mode l'adresse pointée contient l'adresse effective de la donnée à chercher.

6° - Le mode d'adressage relatif court :

Dans ce mode on donne le déplacement nécessaire par rapport au contenu du compteur ordinal (PC) pour effectuer un branchement. Le déplacement étant codé sur un octet on pourra alors se déplacer de 127 octets en avant et de - 128 octets en arrière.

7° - Le mode d'adressage relatif long :

Comme le mode relatif court, il est réservé aux branchements mais dans ce mode le déplacement est codé sur deux octets. Le déplacement peut être alors de 32767 octets en avant et de - 32768 en arrière par rapport au contenu du compteur ordinal.

8° - Les modes d'adressage indexés :

Dans ces modes d'adressage, le processeur fait appel à une adresse de base contenue dans l'un des registres à 16 bits (X, Y, S ou PC) et à un déplacement appelé offset.

L'adresse effective est alors donnée par la somme de l'adresse de base et du déplacement. On notera qu'il y a plusieurs possibilités pour le déplacement, celui-ci peut être nul ou codé sur cinq, huit ou seize bits ou encore variable (dans le cas où l'index est le PC) en utilisant l'un des accumulateurs A, B ou D.

De plus nous avons les modes d'indexation à autoincrémentation ou à autodécrémentation de 1 ou 2 .

9° - Les modes d'adressage indexés indirects :

Dans ces modes on accède à l'adresse effective en transitant par une adresse intermédiaire. Ainsi la somme de l'index et du déplacement nous donne l'adresse de base dont le contenu est l'adresse effective.

Tous les modes d'adressage indexés à l'exception des modes à autoincrémentation et à auto-décrémentation peuvent travailler en indirection.

Pour ses échanges avec le monde extérieur le MPU possède des entrées de demande d'interruption, lui permettant de changer d'état en fonction des priorités relatives de l'opération en cours et de celle demandée. Le MPU possède trois lignes d'entrées de demande d'interruptions matérielles qui sont activées par un front actif sur l'une des entrées suivantes :

- $\overline{\text{NMI}}$ (No maskable interrupt) ;
- $\overline{\text{FIRQ}}$ (Fast interrupt request) ;
- $\overline{\text{IRQ}}$ (Interrupt request).

Il possède aussi une séquence d'initialisation ($\overline{\text{RESET}}$) qui est traitée comme l'interruption la plus prioritaire du système. Ainsi qu'un ensemble d'interruptions logicielles qui viennent du programme lui-même. Elles lui permettent d'arrêter un programme, d'exécuter le programme instruction par instruction, de faire une lecture ou une écriture sur un circuit périphérique et d'attendre une interruption ou une synchronisation externe. Ces interruptions sont :

- SWI (Software interrupt)
- SWI2(" " n° 2)
- SWI3(" " n° 3)
- CWAY (Attente d'une interruption)
- SYNC (Attente d'une synchronisation externe).

Pour un traitement efficace de ces interruptions, le MPU introduit la notion de priorité qui est généralement liée à sa structure matérielle.

Les méthodes de traitement d'échanges avec les circuits extérieurs présentent des similitudes :

- Le programme principal est toujours interrompu ;
- Il y a une sauvegarde totale ou partielle du contexte du MPU dans une pile ;
- L'interruption n'est prise en compte qu'à la fin de l'instruction en cours.

Les adresses des programmes de traitement des différentes interruptions qui correspondent aux vecteurs d'interruption, sont rangées dans l'espace mémoire allant de FFF2 à FFFF. Ces adresses sont automatiquement pointées par le MPU lors de la prise en compte d'une interruption, ceci dans le but d'un traitement rapide et simple. Le tableau suivant donne les adresses contenant les vecteurs d'interruption :

L'INTERRUPTION	ADRESSE CONTENANT LE VECTEUR D'INTERRUPTION
$\overline{\text{RESET}}$	FFFF / FFFE
$\overline{\text{NMI}}$	FFFD / FFFC
SWI	FFFB / FFFA
$\overline{\text{IRQ}}$	FFF9 / FFF8
$\overline{\text{FIRQ}}$	FFF7 / FFF6
SWI2	FFF5 / FFF4
SWI3	FFF3 / FFF2

TRAITEMENT DES DIFFERENTES INTERRUPTIONS :

1° - Traitement du $\overline{\text{RESET}}$:

Un niveau bas sur cette entrée du microprocesseur entraîne l'initialisation complète de ce dernier. L'organigramme de la figure 6 montre les différentes phases qu'exécute le MPU avant le traitement du programme d'initialisation proprement dit. Dès l'apparition d'un niveau bas durant plus d'un cycle machine sur cette entrée, le MPU exécute les instructions suivantes :

- a - Initialisation du registre de page à zéro ;
- b - Masquage des interruptions matérielles $\overline{\text{FIRQ}}$ et $\overline{\text{TRQ}}$ en mettant les bits I et F du CCR à un ;
- c - Mise à l'état un de la ligne $\text{R}/\overline{\text{W}}$, permettant ainsi au microprocesseur de lire le vecteur d'interruption correspondant au $\overline{\text{RESET}}$;
- d - Désactivation de l'interruption $\overline{\text{NMI}}$ qui ne sera validée que par une initialisation du pointeur de pile S ;
- e - Test des lignes $\overline{\text{DMA/BREQ}}$ et $\overline{\text{HALT}}$ et le MPU confirme, si l'une d'entre-elles est à l'état bas, qu'il est arrêté ou que son bus est géré par un système externe, en mettant BA et BS à l'état haut. Le traitement du $\overline{\text{RESET}}$ sera dans ce cas différé ;
- f - Dans le cas où les lignes $\overline{\text{DMA/BREQ}}$ et le $\overline{\text{HALT}}$ sont à l'état haut, le MPU confirme le fonctionnement normal en mettant BA et BS à l'état bas (0) ;
- g - Le processeur teste le $\overline{\text{RESET}}$ et reste en attente tant qu'il est à l'état bas ;
- h - Quand le $\overline{\text{RESET}}$ passe à l'état haut, le processeur met BA à zéro et BS à un indiquant qu'il recherche un vecteur d'interruption ;
- i - Le MPU prend en compte le contenu des adresses FFFF et FFFE ;
- j - La prise en compte du vecteur d'interruption étant terminée, le MPU met BS à zéro indiquant qu'il revient au fonctionnement normal et le PC contient alors l'adresse du programme d'initialisation.

Poids forts = contenu de FFFE

Poids faibles = contenu de FFFF

- k - Le processeur exécute le programme d'initialisation.

La durée de la séquence de pré-initialisation dure environ 10 cycles machine.

2° - Traitement de la $\overline{\text{NMI}}$:

C'est un front descendant au lieu d'un niveau de tension qui active cette interruption, c'est la plus prioritaire de toutes les interruptions du MPU. Elle ne peut être inhibée par logiciel et ne sera validée que par une initialisation du pointeur de pile système. L'organigramme de la figure 7 montre les différentes instructions exécutées par le MPU avant le traitement du programme de la $\overline{\text{NMI}}$. Ces instructions durent environ une vingtaine de cycles machine .

- a - La prise en compte d'un front descendant sur $\overline{\text{NMI}}$ a lieu à la fin de l'instruction en cours ;
- b - L'indicateur d'état de sauvegarde sera mis à un pour indiquer qu'il y aura sauvegarde totale du contexte microprocesseur dans la pile.
- c - Tous les registres du MPU sont sauvegardés à partir de l'adresse SP contenue dans la pile moins un. A la fin de la sauvegarde le pointeur S indique l'adresse du registre code condition.

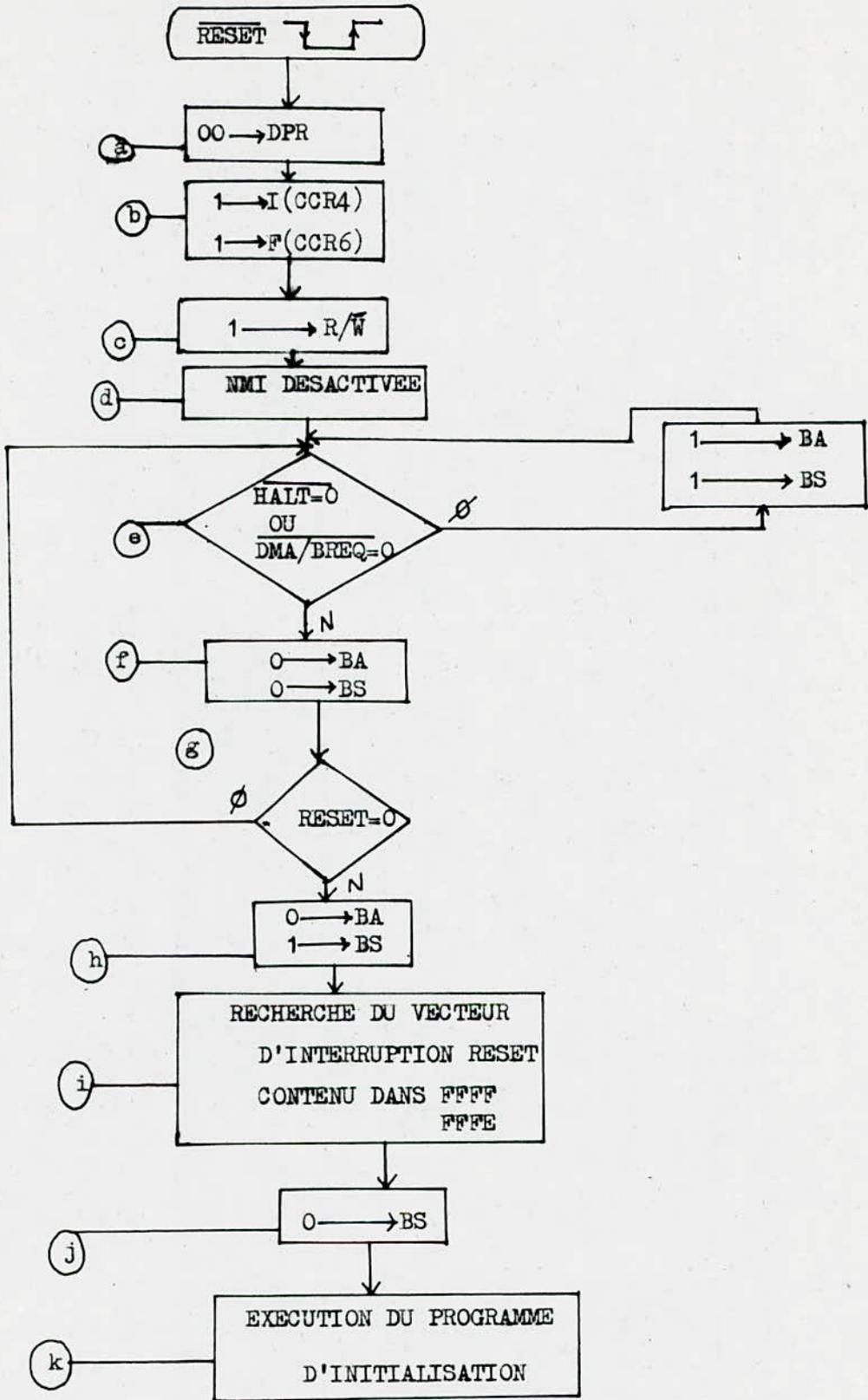


FIG-6 ORGANIGRAMME DE TRAITEMENT DU $\overline{\text{RESET}}$

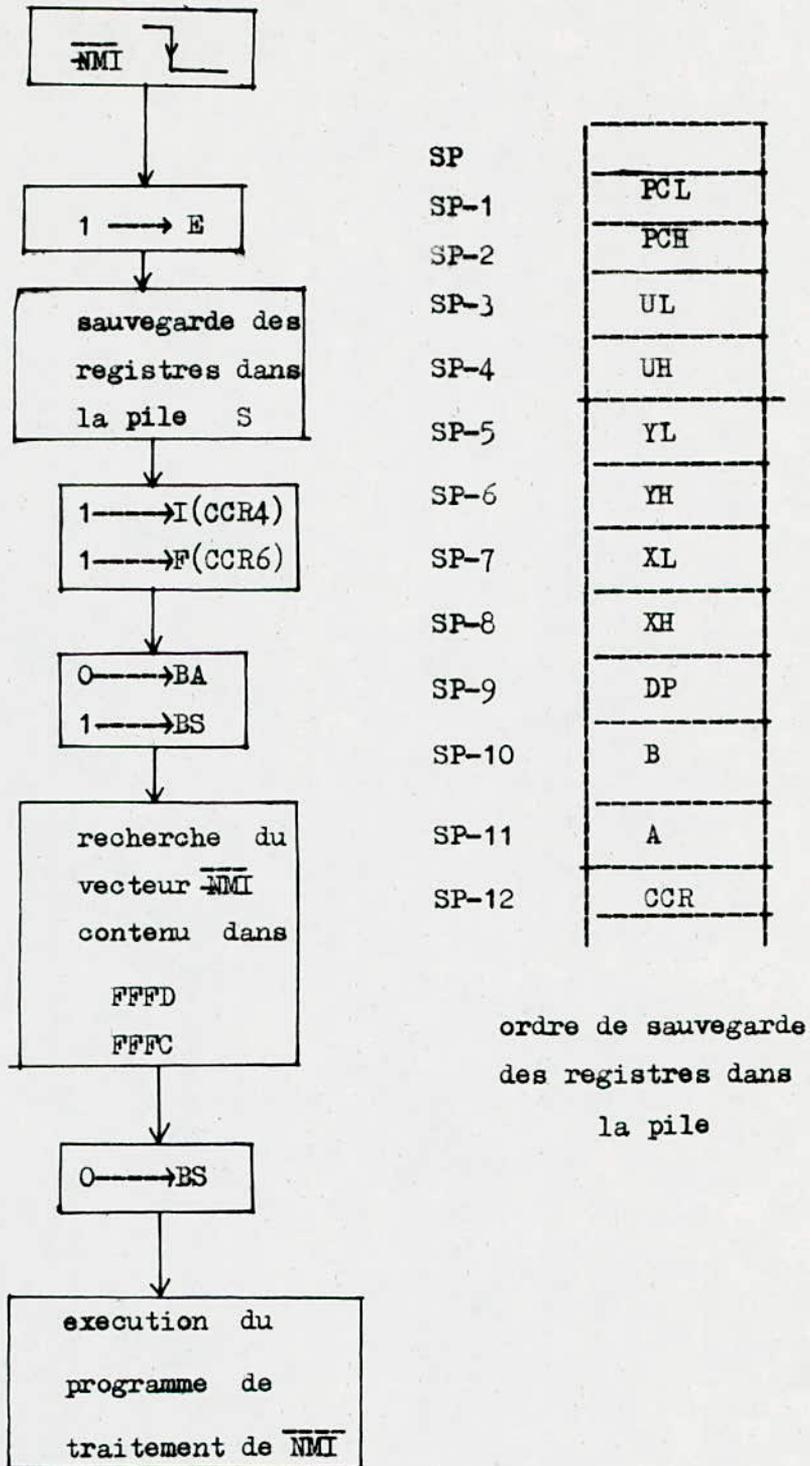


FIG-7 ORGANIGRAMME DE TRAITEMENT DE LA $\overline{\text{NMI}}$

PCL	SP-1
PCH	SP-2
CCR	SP-3

Ordre de sauvegarde des registres dans la pile

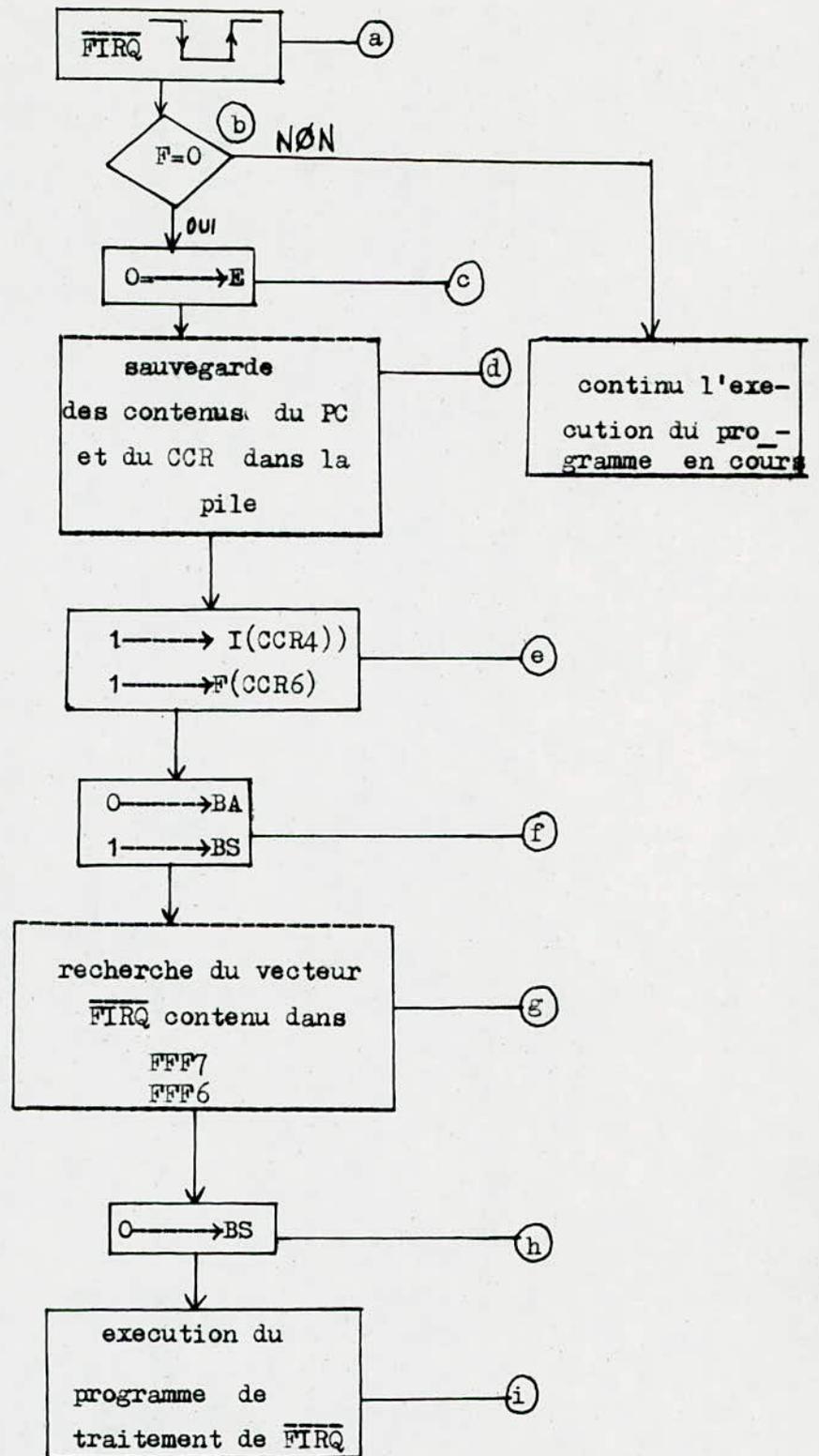


FIG-8 ORGANIGRAMME DE TRAITEMENT DE $\overline{\text{FIRQ}}$

- d- Les interruptions matérielles sont inhibées par la mise à un des masques I et F.
 - e -Pour indiquer qu'il recherche le vecteur d'interruption le MPU met BA à zéro et BS à un ;
 - f - Le MPU prend en compte le contenu des adresses FFFC et FFFD ;
 - g - Le MPU remet BS à zéro pour indiquer qu'il revient au fonctionnement normal. Le PC contient alors l'adresse du programme de traitement de la $\overline{\text{NMI}}$;
- Poids forts = contenu de FFFC
Poids faibles = contenu de FFFD
- h - Le MPU commence l'exécution du programme de traitement de la $\overline{\text{NMI}}$.

3° Traitement de la $\overline{\text{FIRQ}}$:

C'est une séquence d'interruption matérielle rapide , masquable qui est activée par un niveau bas. L'organigramme de la figure 8 décrit les différentes instructions que le MPU exécute avant le traitement du programme de la $\overline{\text{FIRQ}}$. Ces instructions durent environ 11 cycles machine.

- a - La prise en compte de la $\overline{\text{FIRQ}}$ n'a lieu qu'à la fin de l'instruction en cours ;
- b - Le processeur teste le masque d'interruption F du CCR, s'il est à un l'interruption est masquée sinon elle sera prise en compte puis traitée ;
- c - L'indicateur E est mis à zéro par le MPU pour indiquer qu'il y a une sauvegarde partielle du contenu du microprocesseur dans la pile ,
- d - Le compteur programme et le registre d'état sont sauvegardés dans la pile à partir de l'adresse SP - 1. Le pointeur de pile pointe le CCR à la fin de la sauvegarde ;
- e - Les interruptions matérielle $\overline{\text{FIRQ}}$ et $\overline{\text{TRQ}}$ seront masquées en mettant les indicateurs I et F à un ;
- f - L'indication de la recherche du vecteur d'interruption se fait par la mise de BA à zéro et BS à un ;
- g - Le MPU prend en compte le contenu des adresses FFF6 et FFF7 ;
- h - Le MPU indique le fonctionnement normal en remettant BS à zéro. L'adresse du début du programme de traitement de la $\overline{\text{FIRQ}}$ est alors dans le PC.

Poids forts = contenu de FFF6 * Poids faibles = contenu de FFF7

- i - Exécution du programme de traitement de la $\overline{\text{FIRQ}}$ par le MPU .
- L'utilisateur pourra sauvegarder dans la pile d'autres registres que le PC et le CCR grâce à l'instruction PSHS.

4° Traitement de l' $\overline{\text{TRQ}}$:

C'est l'interruption matérielle la moins prioritaire. Elle est activée par un niveau bas et est masquable par l'intermédiaire du bit I du CCR. L'organigramme de la figure 9 décrit les différentes phases que le processeur exécute avant le traitement du programme de l' $\overline{\text{TRQ}}$. Cette séquence dure environ 20 cycles machine.

- a - La prise en compte d'un niveau bas sur l'entrée $\overline{\text{TRQ}}$ se fait à la fin de l'instruction en cours ;
- b - Le masque d'interruption I de l' $\overline{\text{TRQ}}$ est testé, s'il est à un elle est masquée sinon elle est prise en compte puis traitée ;
- c - Le processeur met l'indicateur E à un pour indiquer qu'il y a sauvegarde totale du contexte microprocesseur dans la pile ;

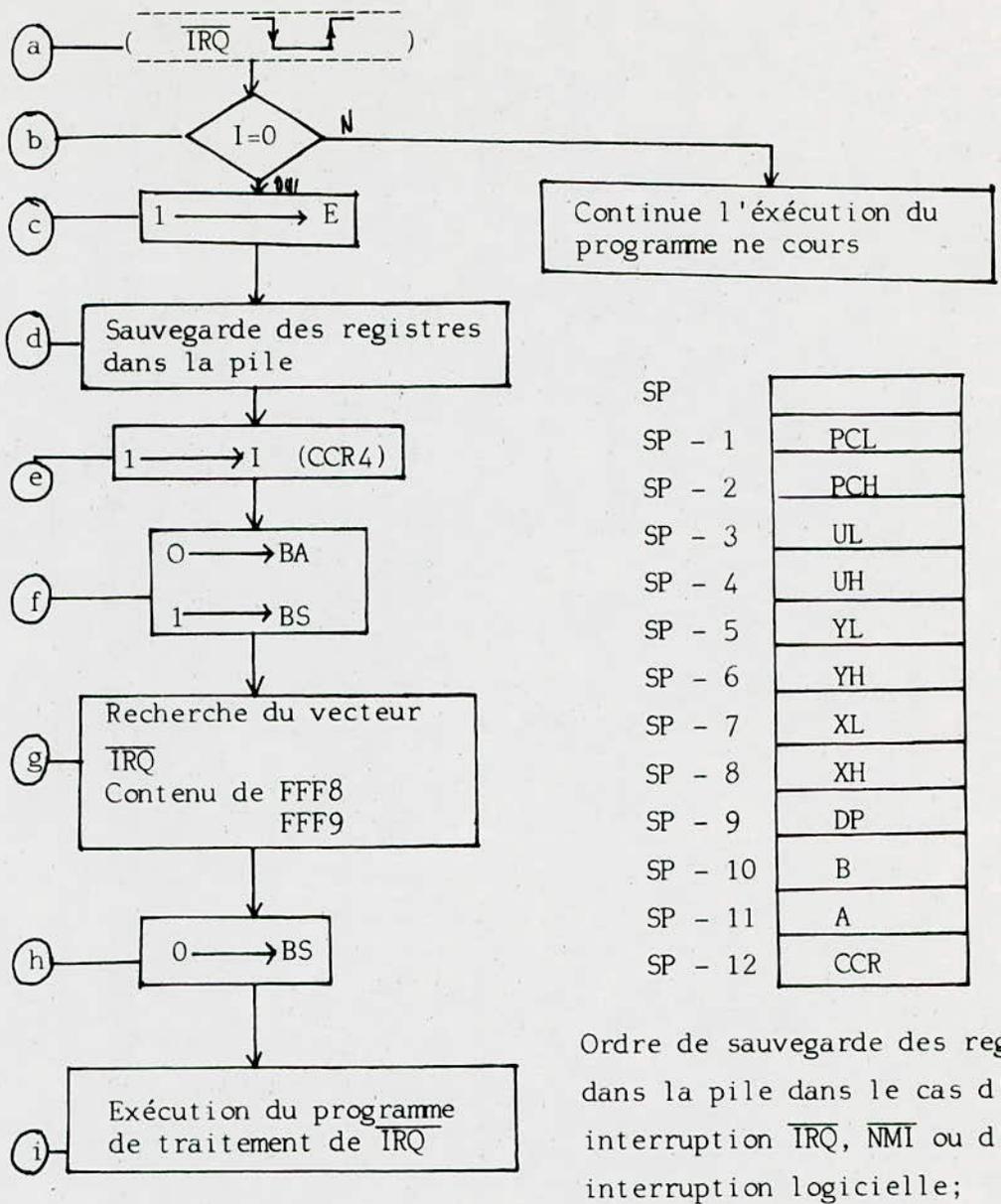


FIG-9 ORGANIGRAMME DE TRAITEMENT DE L' $\overline{\text{TRQ}}$

- d- Après la sauvegarde de tous les registres internes dans la pile, le pointeur indique l'adresse SP -12 ;
- e- Masquage de l'interruption \overline{TRQ} en mettant I à un ;
- f - Le MPU met BA à zéro et BS à un pour indiquer qu'il recherche un vecteur d'interruption ;
- g - Le MPU prend en compte le contenu des adresses FFF8 et FFF9 ;
- h - Le MPU indique qu'il revient au fonctionnement normal, en remettant BS à zéro. Le PC contient alors l'adresse du début du programme de traitement de l'interruption \overline{TRQ} .
Poids forts = contenu de FFF8 * Poids faibles = contenu de FFF9
- i - Exécution du programme de traitement de l' \overline{TRQ} par le MPU.

5° Traitement de l'interruption SWI :

C'est une interruption logicielle qui impose l'arrêt de l'exécution du programme qui la contient. L'organigramme de la figure 10 décrit les différentes instructions que le MPU exécute avant le traitement du programme de SWI.

- a - SWI impose l'arrêt du programme en cours ;
- b - Pour indiquer qu'il y a sauvegarde totale dans la pile, l'indicateur E du registre CCR sera mis à un ;
- c - Tous les registres internes du MPU sont sauvegardés dans la pile sauf le pointeur de pile S ;
- d - Inhibition des interruptions matérielles \overline{FTRQ} et \overline{NMI} en mettant les indicateurs I et F du CCR à un ;
- e - Le processeur met BA à zéro et BS à un pour indiquer qu'il recherche le vecteur d'interruption ;
- f - Le PC sera chargé par le contenu des adresses FFFA et FFFB ;
- g - Le MPU revient au fonctionnement normal et met BS à zéro ;
- h - Le processeur exécute le programme de traitement de la SWI.

6° Les interruptions SWI2 et SWI3 :

Ces interruptions dont le fonctionnement est identique sont les moins prioritaires de toutes celles du MPU. L'organigramme de la figure 11 décrit le séquençement des différentes opérations internes du MPU avant l'exécution du programme de traitement par ce dernier.

- a- SWI2 et SWI3 imposent l'arrêt du programme ;
- b - L'indicateur E sera mis à un pour indiquer la sauvegarde totale du contexte microprocesseur dans la pile ;
- c - Tous les registres du MPU sauf le pointeur de pile S sont sauvegardés ;
- d - Le MPU met BA à zéro et BS à un pour indiquer qu'il recherche le vecteur d'interruption correspondant à SWI2 et SWI3 ;
- e - Le PC est chargé par le contenu des adresses :
FFF5 et FFF4 pour la SWI2 * FFF3 et FFF2 pour la SWI3
- f- Le MPU remet BS à zéro pour indiquer qu'il revient au fonctionnement normal ;
- g - Exécution du programme de traitement de SWI2 et SWI3 .

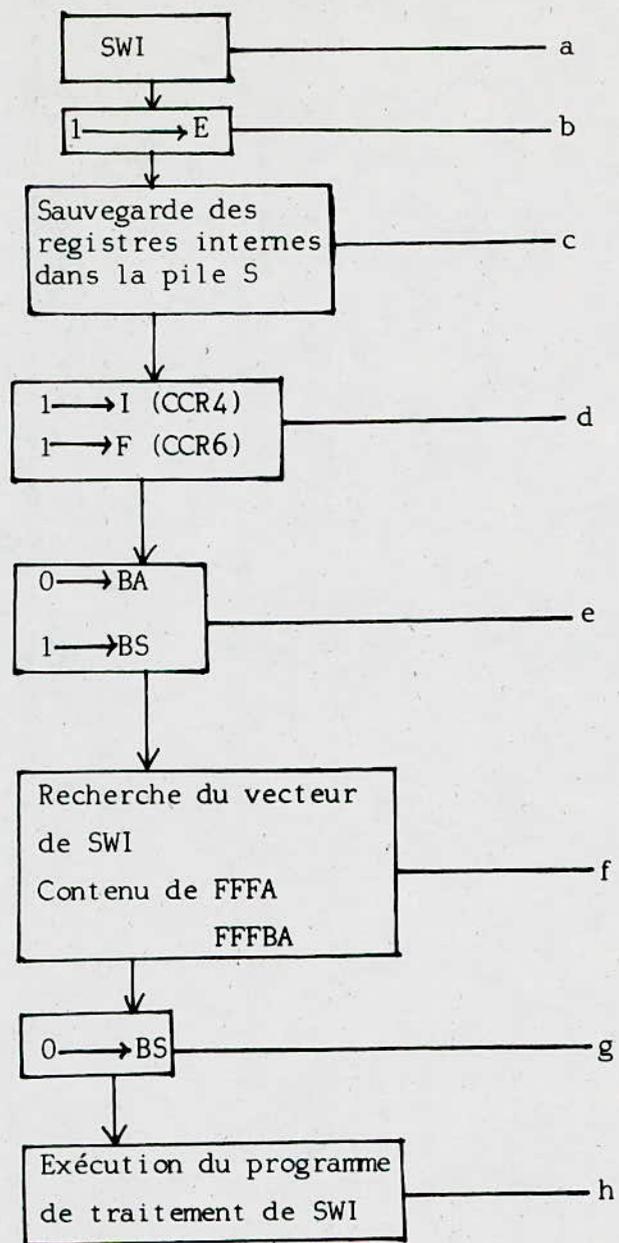


FIG-10 ORGANIGRAMME DE TRAITEMENT DE SWI

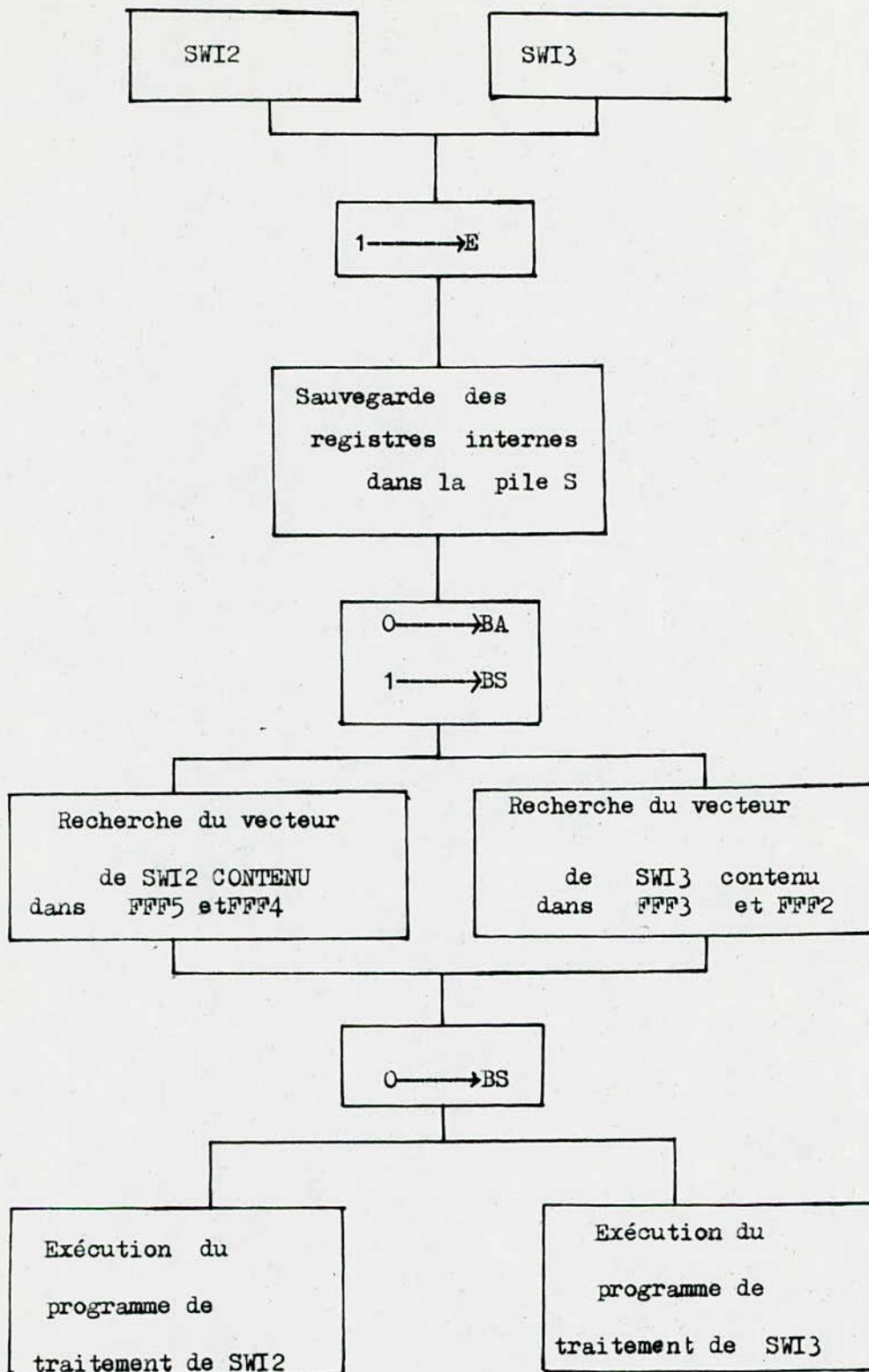


FIG- 11 ORGANIGRAMME DE TRAITEMENT DE SWI2 ET SWI3

7° L'instruction attente d'interruption CWAY :

Cette instruction permet d'effacer les masques d'interruptions, car elle effectue un **et** logique entre l'octet mémoire immédiat et le registre code conditions.

On notera que lors de l'exécution de cette instruction les bus ne sont pas mis à l'état haute impédance et BA et BS restent à l'état bas.

L'organigramme de la figure 12 décrit les différentes instructions qu'exécute le MPU lorsqu'il rencontre CWAY.

a- L'instruction CWAY provoque l'arrêt de l'exécution du programme qui la contient ;

b - Le MPU masque ou valide les interruptions matérielles.

CCR = FF $\overline{\text{TRQ}}$ et $\overline{\text{FIRQ}}$ sont masquées

CCR = EF $\overline{\text{TRQ}}$ autorisée, $\overline{\text{FIRQ}}$ masquée

CCR = BF $\overline{\text{TRQ}}$ masquée, $\overline{\text{FIRQ}}$ autorisée

CCR = AF $\overline{\text{TRQ}}$ et $\overline{\text{FIRQ}}$ sont autorisées

c- Le MPU indique la sauvegarde totale du contexte en mettant l'indicateur E à un ;

d - Tous les registres internes sauf S sont sauvegardés dans la pile ;

e - Le MPU se met en attente d'une interruption $\overline{\text{TRQ}}$, $\overline{\text{FIRQ}}$ (suivant le contenu du CCR) ou $\overline{\text{NMI}}$. Lorsque l'une de ces interruptions est activée, il n'y a plus sauvegarde d'un autre état du MPU, avant la prise en compte du vecteur d'interruption.

8° Instruction de synchronisation externe : "SYNC"

Cette instruction permet grâce aux lignes d'interruptions matérielles, la synchronisation rapide de l'exécution du programme avec les périphériques. Cette méthode permet éventuellement d'éviter l'utilisation d'un circuit d'accès direct mémoire.

L'organigramme de la figure 13 donne les différentes étapes par lesquelles passe le MPU lorsqu'il rencontre cette instruction.

a- L'instruction SYNC arrête le déroulement du programme en cours ;

b- Les bus d'adresses et de données du MPU sont à l'état haute impédance ;

c- Le MPU met BA à un et BS à zéro pour indiquer qu'il est en attente de synchronisation ;

d- Le MPU reste en attente de synchronisation tant qu'il n'a pas reçu de signal d'interruption $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$ ou $\overline{\text{TRQ}}$;

e- Si une interruption se présente sur l'une des entrées $\overline{\text{TRQ}}$ ou $\overline{\text{FIRQ}}$ le fonctionnement du MPU redevient normal, alors BA et BS sont mis à zéro. Si l'interruption est validée et que le niveau bas reste plus de 3 cycles machine, le MPU exécute le traitement approprié de cette interruption et reprend le déroulement normal du programme au retour de celui-ci. Par contre si le niveau bas dure moins de 3 cycles machine ou que l'interruption est masquée, le MPU continue l'exécution du programme principal sans traiter l'interruption ;

f- Si une interruption est activée sur l'entrée $\overline{\text{NMI}}$ et que le niveau bas dure plus de 3 cycles machine, le MPU exécute son traitement, par contre si le niveau bas dure moins de 3 cycles il continue l'exécution du programme principal en ignorant l'interruption.

9° Instruction de retour d'interruption RTI :

Pour éviter toute mauvaise utilisation de la pile l'instruction RTI termine toutes les séquences de traitement d'interruptions.

L'organigramme de la figure 14 montre les séquences de traitement de cette instruction.

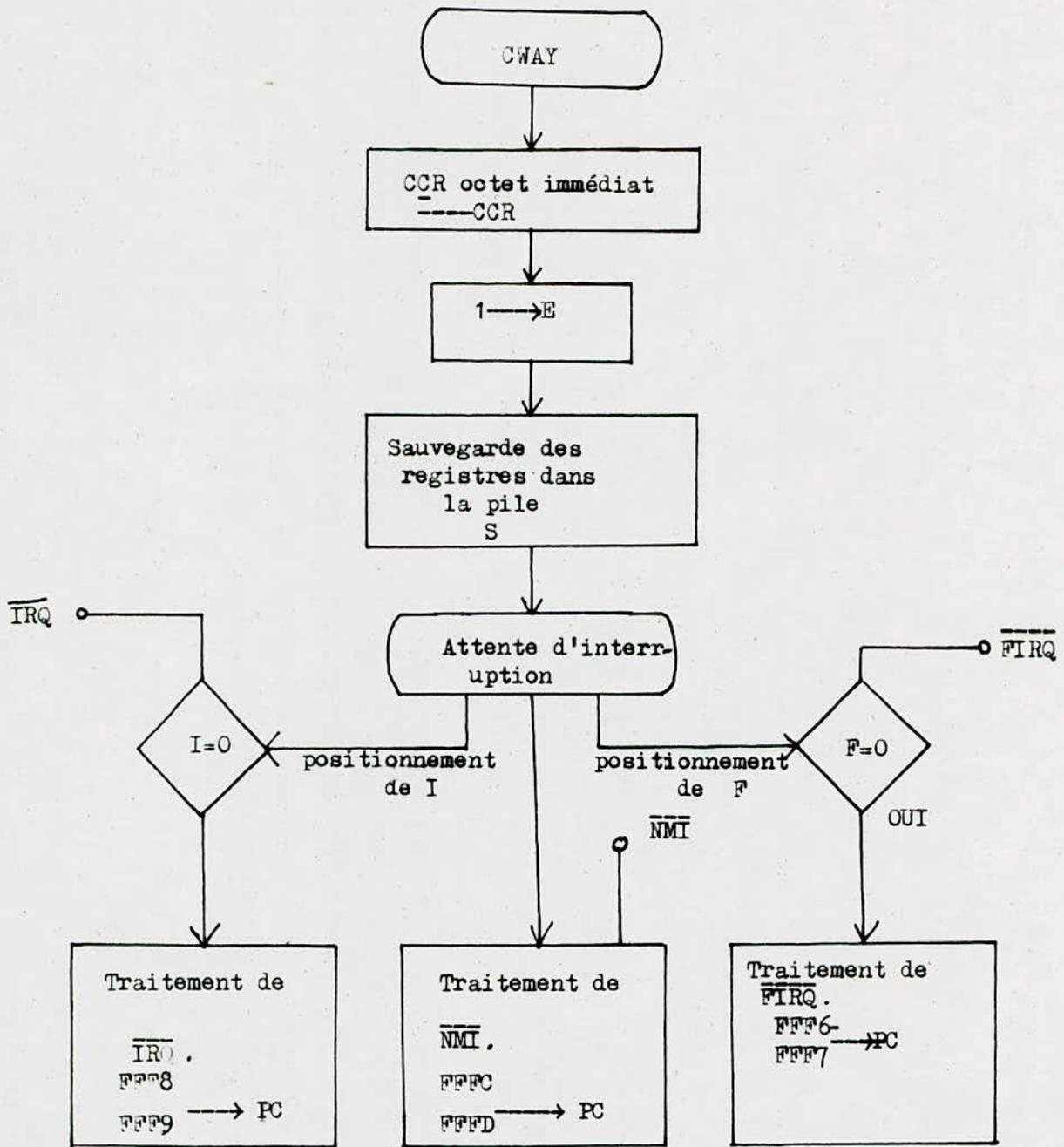


FIG-12 ORGANIGRAMME DE TRAITEMENT DE CWAY

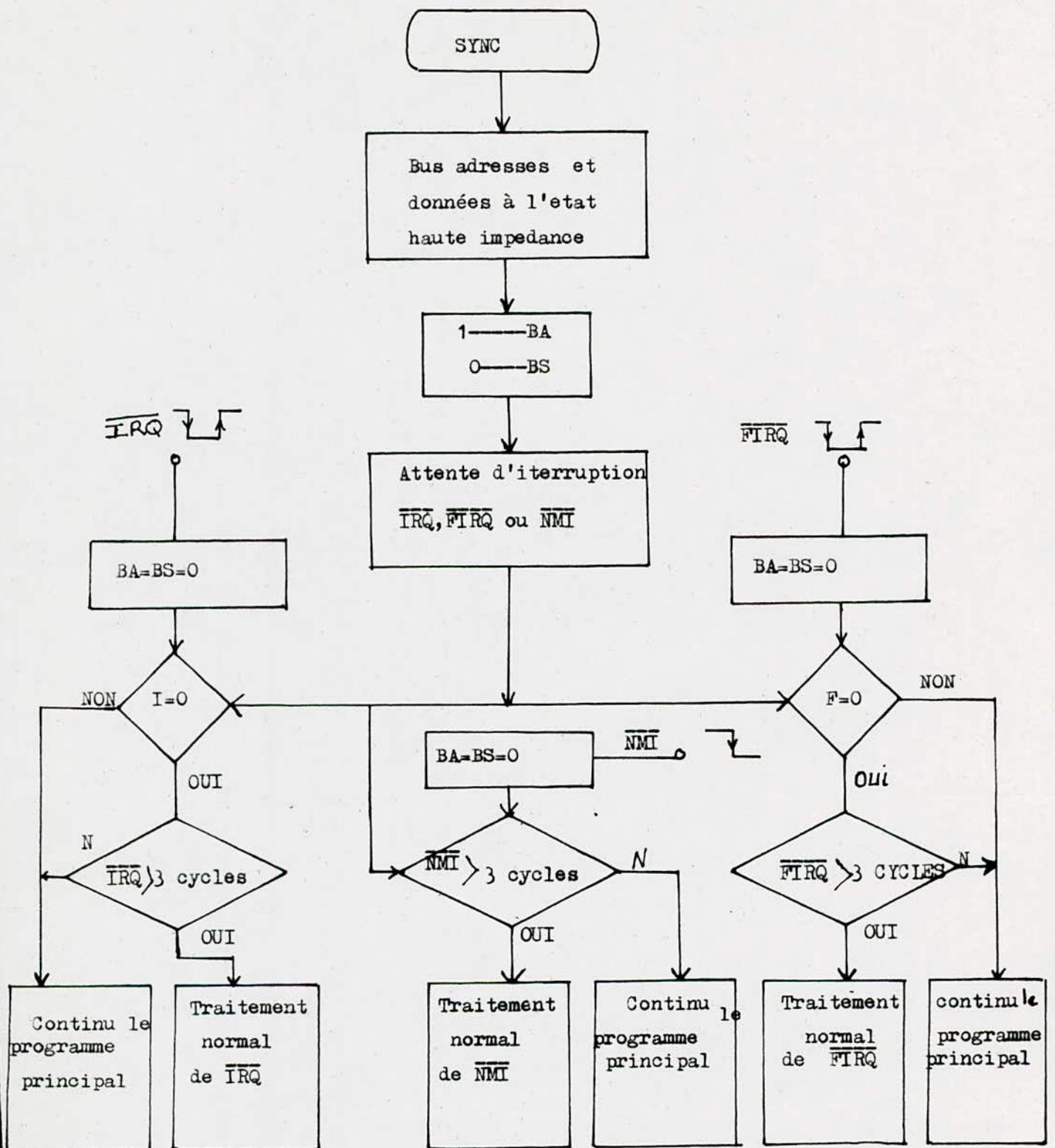


FIG-13 ORGANIGRAMME DE TRAITEMENT DE SYNC

- a- cette instruction est rencontrée à la fin des programmes de traitement d'interruptions ;
- b- Le registre d'état est chargé par le contenu de l'adresse donnée par le pointeur de pile ;
- c- Le MPU teste l'indicateur E, s'il est à un la sauvegarde est totale sinon elle est partielle ;
- d- Restitution des contenus des registres internes du MPU et continuité du déroulement normal du programme.

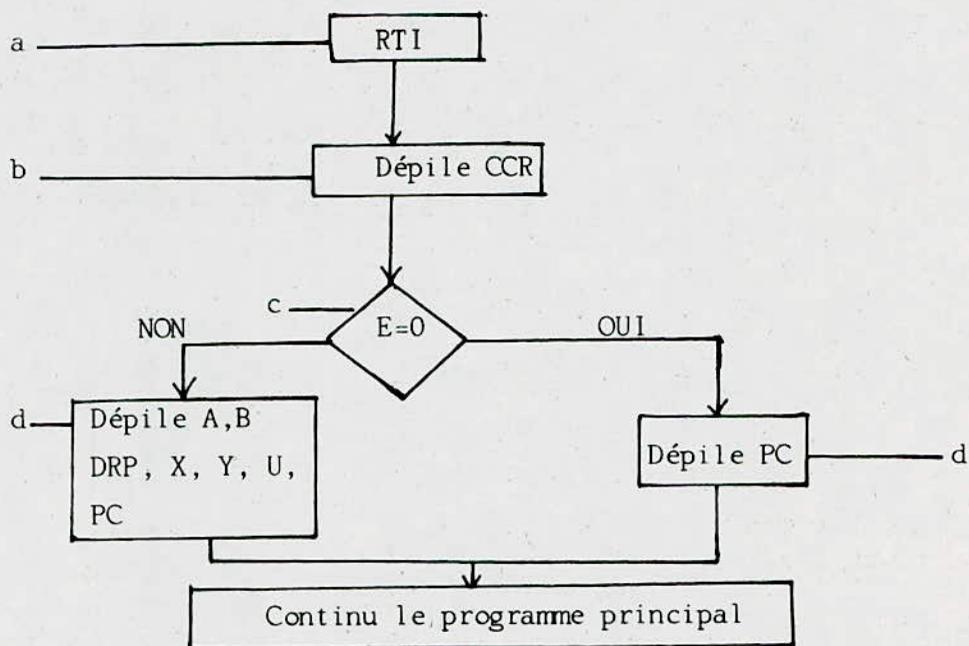


FIG-14 : ORGANIGRAMME DE TRAITEMENT DE RTI

Le dialogue entre l'unité centrale et les différents périphériques, nécessite des unités d'échanges qui peuvent assurer :

- Soit la fonction Entrée, permettant l'acquisition d'informations en provenance d'un clavier, d'un capteur, d'un convertisseur analogique-numérique, d'une mémoire de masse etc...

- Soit la fonction Sortie, permettant la transmission d'informations vers un convertisseur numérique-analogique, un écran, une mémoire de masse etc...

Ces unités sont des interfaces dont le rôle est fonction du périphérique avec lequel ils sont connectés. Ces interfaces communiquent avec le MPU par l'intermédiaire des bus de contrôle, de données et d'adresses. Ils sont alors vus par le MPU comme des positions mémoires et sont programmés en fonction du périphérique connecté. On notera que ces interfaces sont de deux catégories :

- Les interfaces actifs (ou intelligents), qui gèrent eux-mêmes les échanges avec la périphérie et permettent ainsi au MPU de ne plus perdre de temps dans la gestion de tâches complexes. Seulement leur structure matérielle complexe rend difficile leur mise en oeuvre ;

- Les interfaces passifs dont la principale caractéristique est la gestion des transferts par le MPU. Leur facilité de mise en oeuvre et leur souplesse d'utilisation font d'eux des interfaces très utilisés même si la gestion des échanges est régie par le MPU.

L'unité centrale étant dans notre cas le microprocesseur MC 6809 qui est d'une part, un microprocesseur 8 bits compatible avec le MC 6800 d'autre part un intermédiaire vers les performances des microprocesseurs 16 bits ; nous permet d'utiliser tous les interfaces de la famille 6800 et certains de la famille 68000 en plus de deux qui lui sont spécifiques (Le circuit de gestion mémoire MMU et les programmes mathématiques en ROM).

Nous étudierons ci-dessous les deux circuits d'interfaces que nous avons utilisé dans la réalisation. Ce sont des interfaces passifs de la famille 6800 dont l'un est à entrées / sorties parallèles l'autre à entrées / sorties séries.

B1- L'INTERFACE A ENTREES / SORTIES PARALLELES : LE PIA 6821

Le MC 6821 est un circuit intégré en boîtier DIL à 40 broches, monotension (5 V), compatible TTL fabriqué en technologie MOS à canal N.

C'est un circuit d'interface qui est destiné aux applications pour lesquelles les données sont organisées en mode parallèle. Il sert de coupleur de périphériques et permet la réalisation de liaison parallèle entre le microprocesseur et ses périphériques. Il permet de transmettre des données de 8 bits à travers un bus bidirectionnel.

Il communique avec le MPU par l'intermédiaire du bus de données bidirectionnel (8bits), de deux lignes d'adresses pour la sélection de registres, de trois lignes de sélection de boîtier et d'un bus de contrôle contenant une ligne d'écriture/lecture (R/W), une ligne d'horloge (E), une ligne de mise à l'état initial (RESET) ... / ...

et deux lignes de demande d'interruptions.

L'organisation interne du PIA est divisée en deux parties indépendantes A et B comportant chacune : deux lignes de contrôle, trois registres internes de 8 bits chacun et un port de huit lignes bidirectionnelles. Le dialogue avec la périphérie se fait par l'intermédiaire des deux ports A et B et de quatre lignes de demande d'interruptions. La figure 15 donne le brochage du MC 6821

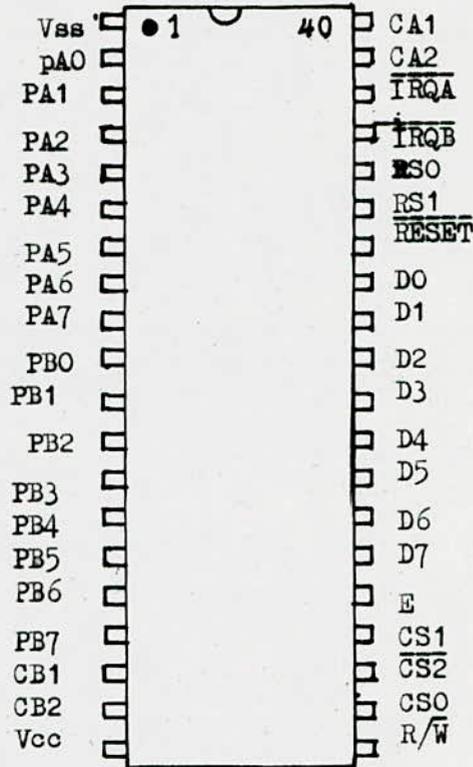


FIG-15 : SCHEMA DE BROCHAGE DU MC 6821

B1-1 SIGNIFICATION DES DIFFERENTES BROCHES DU MC 6821 :

a- Le bus de données : (D0-D7)

Ce sont huit lignes de données bidirectionnelles qui sont directement reliées au bus de données du MPU, elles permettent le transfert de données entre ce dernier et le PIA. Elles sont en logique trois états et sont à l'état haute impédance lorsqu'elles ne sont pas utilisées.

b- Entrées de sélection de boîtier : CS0, CS1 et $\overline{CS2}$ (chip select)

Ces trois lignes permettent selon leur état de sélectionner le PIA ou non. Pour le sélectionner les lignes CS0 et CS1 doivent être à l'état haut et $\overline{CS2}$ à l'état bas.

c- Entrées de sélection de registres : RS0 et RS1

Sachant que le PIA est vu par le MPU comme quatre positions mémoire les lignes RS0 et RS1 permettent la sélection de ses registres.

d- Bus de contrôle :

* Le signal de synchronisation E : c'est le signal d'horloge fournit par le MPU afin de synchroniser toutes les opérations du PIA et ses signaux.

* L'entrée de réinitialisation RESET : Un état bas sur cette entrée permet l'initialisation du PIA en mettant tous ses registres internes à zéro.

* Ligne d'écriture/lecture (R/W) : c'est un signal produit par le MPU pour spécifier le sens des transferts de données suivant qu'il y ait une lecture ou une écriture du PIA.

$R/\overline{W} = 1$ indique la lecture, du registre adressé du PIA, par le MPU.

$R/\overline{W} = 0$ indique que le MPU écrit dans le registre adressé du PIA.

e- Les lignes de dialogue avec la périphérie :

* CA1 et CB1 : Ces deux lignes d'entrées sont activées sur des fronts montant ou descendant selon l'état du registre de contrôle associé et permettent de positionner les indicateurs d'interruption des registres de contrôle (bit B7)

* CA2 et CB2 : Ces deux lignes peuvent être programmées en entrée ou en sortie par l'intermédiaire des bits b3, b4, et b5 des registres de contrôle associés et positionnent les indicateurs d'interruption (b6) de ces registres quand elles sont

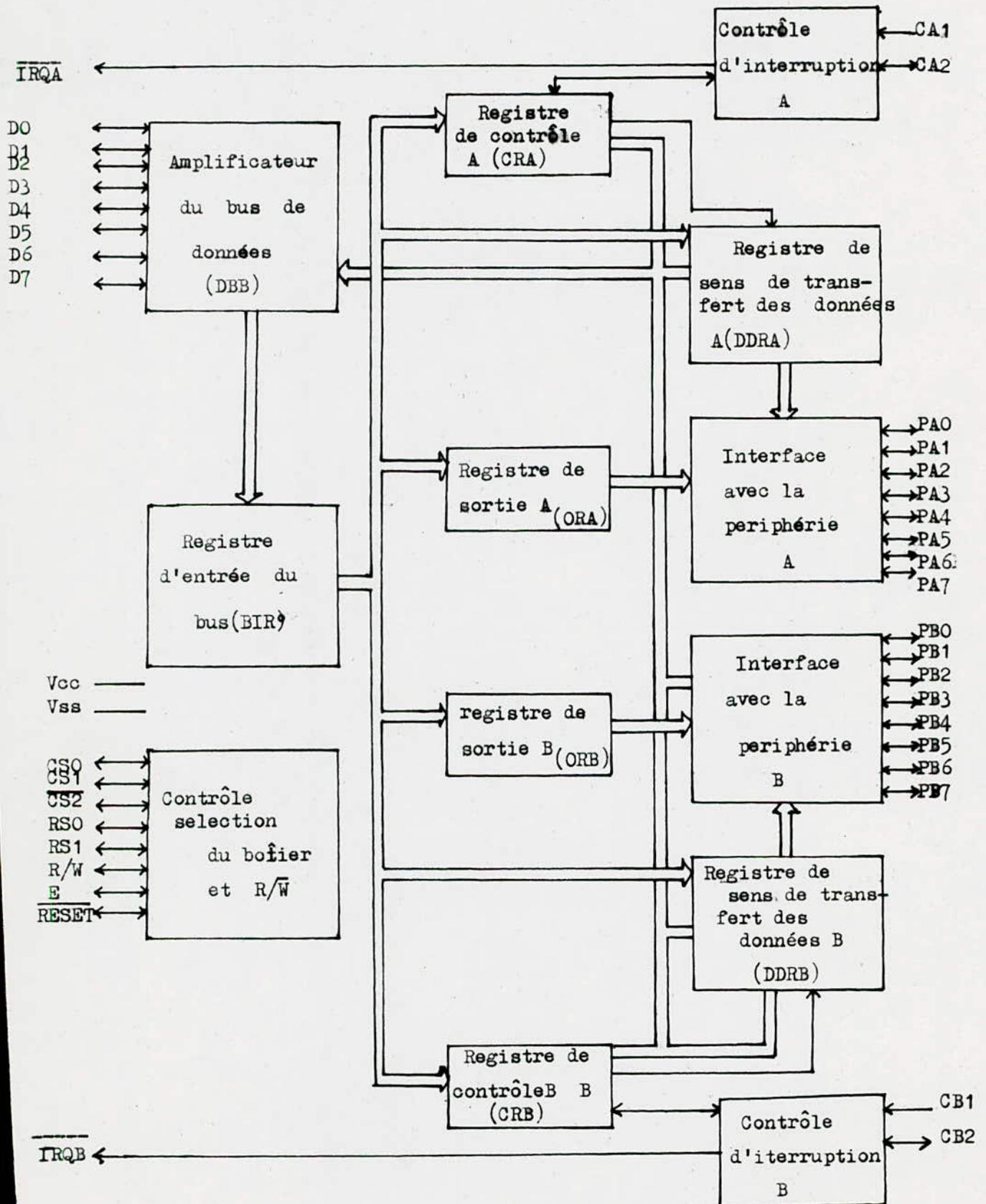


FIG-16 DIAGRAMME E FONCTIONNEL DU MC 6821

programmées en entrées. La ligne CB2 peut servir de source de courant de 1 mA sous 1,5V de tension .

f- Les sorties de liaison avec la périphérie :

* Lignes du port A (PA0 - PA7) : Ces huit lignes programmables en entrées ou en sorties par l'intermédiaire du registre de sens de transfert de données (DDRA), permettent de transmettre ou de recevoir un mot de 8 bits.

* Lignes du port B (PB0-PB7) : Comme les lignes du port A elles sont programmables en entrées ou en sorties par l'intermédiaire du registre de transfert de données (DDRB), à la différence qu'elles sont en logique trois états.

B1-2 ORGANISATION INTERNE DU PIA :

Le PIA possède six registres internes, qui sont vus comme quatre positions mémoire par le MPU et qui lui sont accessibles. Ils se divisent en deux registres de données de la périphérie, en deux registres de sens de transfert de données et en deux registres de contrôle. La figure 16 donne le schéma de l'organisation interne du PIA.

a- Les registres de données : (ORA et ORB)

Les données présentes sur les ports A et B en entrées sont prises en comptes par une lecture de ORA ou ORB mais ne sont pas mémorisées dans ces registres. Par contre ces registres permettent de mémoriser une donnée en sortie lors d'une écriture.

b- Les registres de sens de transfert de données : (DDRA et DDRB)

Ces registres de huit bits à lecture et écriture permettent de définir le sens de transfert de données de chacune des lignes des ports A et B. Ainsi un zéro dans un bit de ces registres, positionne la ligne correspondante en entrée et un "un" la positionne en sortie;

c- Les registres de contrôle CRA et CRB :

Ces deux registres permettent au MPU de contrôler le fonctionnement des quatre lignes de contrôle / interruption CA1, CA2, CB1 et CB2. De même qu'ils permettent d'autoriser ou non une interruption sur les lignes \overline{TRQA} et \overline{TRQB} et de contrôler l'état des indicateurs d'interruption.

Les bits de b0 à b5 de ces registres peuvent être lus ou écrits par le MPU, mais l'écriture n'est pas possible sur les bits b6 et b7 qui ne peuvent être modifié que par des interruptions externes sur les lignes contrôle / interruption.

Le rôle de l'état des différents bits de ces registres est donné par le tableau de la figure 17.

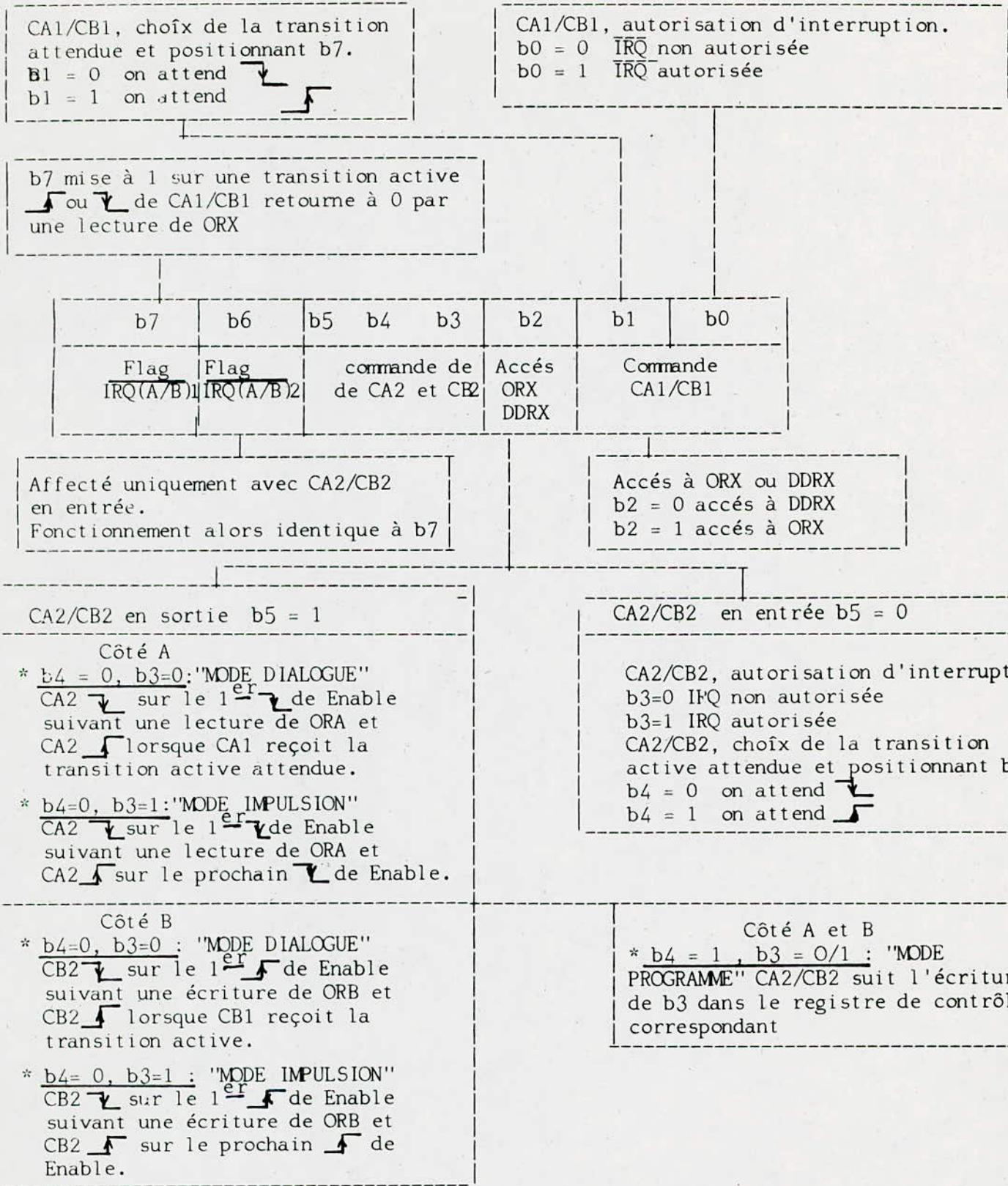


FIG-17 : SYNTHÈSE DE LA PROGRAMMATION DU PIA

B1- 3 FONCTIONNEMENT DU PIA :

Comme nous l'avons décrit précédemment, le type de liaison entre le processeur et la périphérie dépend du contenu des registres internes du PIA. Le MPU accède à ces registres par l'intermédiaire des lignes de sélection de boîtier (CS0, CS1 et $\overline{CS2}$) et des lignes de sélection de registres (RS0, RS1).

Le tableau suivant résume l'adressage des registres internes lorsque CS0 et CS1 sont à l'état haut et $\overline{CS2}$ à l'état bas.

BUS	ADRESSE	A1 RS1	A0 RS0	CRA2	CRB2	Adresses	
R E G I S T R E S	A	CRA	0	1	-	-	ADR + 1
		DDRA	0	0	0	-	ADR
		ORA	0	0	1	-	ADR
	B	CRB	1	1	-	-	ADR + 3
		DDRB	1	0	-	0	ADR + 2
		ORB	1	0	-	1	ADR + 2

Les lignes d'adresse A0 et A1 étant reliées respectivement à RS0 et RS1, la combinaison 01 ou 11 donne accès à CRA ou CRB ce qui permet de positionner les bits de l'un de ces registres. Lorsque b2 du CRX est à zéro les combinaisons 00 ou 10 permettent d'accéder respectivement à DDRA ou DDRB, par contre lorsqu'il est à un les combinaisons donnent accès à ORA ou ORB.

La donnée disponible sur le port (A ou B) est directement transmise à l'ampli de bus de données par l'intermédiaire du bus de sortie sans transiter par les registres de sortie ORA et ORB, donc il n'y aura pas de mémorisation de données en entrée.

Par contre l'information disponible sur le bus de données est chargée dans les registres de sortie ORA ou ORB, elle est donc mémorisée et sera disponible tant qu'une nouvelle écriture n'est pas intervenue sur le port en sortie.

B2- INTERFACE A SYNCHRONE A ENTREES/SORTIES SERIES : ACIA MC 6850

L'ACIA MC 6850 (asynchronous communication interface adapter) est le coupleur de périphérique qui permet de réaliser la liaison série entre le 6809 et ses périphériques. Il réalise la mise en format des données et le contrôle de la transmission.

Sa programmation est fonction de la vitesse de transmission, du format du mot envoyé et du périphérique connecté. C'est un circuit intégré en boîtier DIL à 24 broches, mono-tension (5V) et fabriqué en technologie MOS à canal N.

B2-1 ORGANISATION EXTERNE DE L'ACIA :

L'ACIA dialogue avec l'unité centrale par l'intermédiaire du bus de données, de trois lignes de sélection de boîtier, d'une ligne de sélection de registres et de trois lignes de contrôle. Sa liaison avec la périphérie a lieu par l'intermédiaire de deux lignes de transfert, de trois lignes de contrôle et de deux lignes d'horloge.

a- Le bus de données : (D0-D7)

Ces huit lignes bidirectionnelles reliées directement au bus de données du MPU permettent l'échange de données entre le microprocesseur et l'ACIA. Quand elles ne sont pas utilisées, elles sont dans l'état haute impédance.

b- Lignes de selection de boîtier: (CS0 - CS1 et $\overline{CS2}$)

Ces trois entrées à haute impédance, compatibles TTL, sont utilisées pour sélectionner l'ACIA. Ce dernier est sélectionné lorsque CS0 et CS1 sont à l'état haut et $\overline{CS2}$ est à l'état bas.

c- Ligne de selection de registre : (RS)

Cette entrée à haute impédance, compatible TTL, est utilisée pour sélectionner les registres internes qui occupent deux octets mémoire.

d- Ligne de demande d'interruption : (\overline{IRQ})

La ligne IRQ est une sortie à drain ouvert, compatible TTL et de niveau actif bas. Elle permet d'int interrompre le MPU lorsqu'elle est reliée à l' \overline{IRQ} , la \overline{FIRQ} ou la \overline{NMI} de ce dernier.

e- Ligne de lecture / écriture : (R/ \overline{W})

Cette ligne est une entrée à haute impédance, compatible TTL et est utilisée pour contrôler le sens de transfert des données. A l'état haut elle indique que le MPU est en phase de lecture du registre adressé. A l'état bas elle indique que le MPU est en phase d'écriture sur le registre adressé de l'ACIA.

f- Ligne d'horloge : (E)

Cette entrée à haute impédance, compatible TTL reçoit le signal de synchronisation des échanges du MPU.

g- Lignes de transfert de données : (RxD et TxD)

* La ligne RxD est la ligne de réception des données séries en provenance du périphérique. C'est une entrée à haute impédance, compatible TTL, dont la synchronisation avec l'horloge de reception est réalisée intérieurement par l'ACIA, si la fréquence de reception des données est de 16 ou 64 fois moindre que la fréquence de l'horloge de reception. Par contre pour une fréquence de reception des données égale à celle de l'horloge de reception, la synchronisation doit être faite extérieurement, ce qui permet d'avoir une fréquence de reception variant de 0 à 500 Kbps (*).

* La ligne TxD est une sortie de transmission séries des données. La synchronisation est réalisée avec une horloge interne dont la fréquence peut être de 1/1, 1/16 ou 1/64 de la fréquence appliquée à l'entrée de l'horloge de transmission. Pour avoir une fréquence de transmission allant de 0 à 500 Kbps il faut utiliser une synchronisation externe.

h- Lignes d'horloge : (TxClk et RxClk)

* TxClk est l'entrée de l'horloge de transmission. Les données sont transmises sur un front descendant de celle-ci.

* RxClk est l'entrée de l'horloge de reception. Les données sont prises en compte sur un front montant de celle-ci.

Ces deux entrées sont à haute impédance et sont compatibles TTL.

(*) Kbps = Kilo bits par seconde.

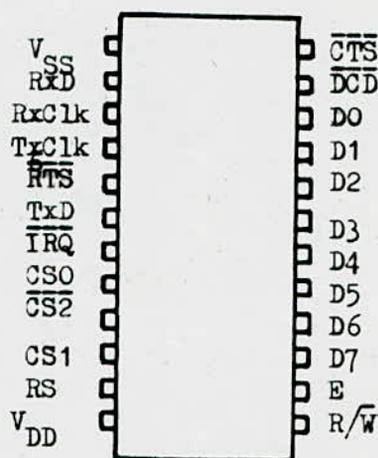
i- Lignes de contrôle du périphérique : ($\overline{\text{RTS}}$, $\overline{\text{CTS}}$ et $\overline{\text{DCD}}$)

* $\overline{\text{RTS}}$ (Request to send) : C'est une entrée de demande d'émission qui permet de piloter un périphérique ou un modem. Elle peut aussi être utilisée pour un périphérique utilisant un signal DTR (Terminal prêt).

* $\overline{\text{CTS}}$ (Clear to send) : C'est une entrée à haute impédance compatible TTL permettant le contrôle automatique de la fin de transmission par un modem ; non utilisée, cette entrée doit être à l'état bas.

* $\overline{\text{DCD}}$ (Data carrier detect) : Cette entrée à haute impédance, compatible TTL permet le contrôle automatique du récepteur de données par un modem. A l'état bas cette entrée indique la présence de la porteuse de données. Une transition de l'état bas à l'état haut sur cette entrée, génère une interruption vers le MPU, pour indiquer la perte de la porteuse de données. A l'état haut, elle inhibe et initialise le récepteur de l'ACIA.

Le schéma de brochage de l'ACIA est donné ci-dessous :



B2-2 ORGANISATION INTERNE DE L'ACIA :

Comme il apparaît sur le schéma de la figure 18 l'ACIA possède 4 registres internes à 8 bits, accessibles à l'utilisateur. Deux d'entre-eux, gèrent le fonctionnement de l'ACIA les deux autres sont associés aux lignes de transmission-réception.

a- Le registre de contrôle : (CR)

Ce registre à écriture seule, contient les paramètres de fonctionnement en transmission et en réception ; et contrôle les autorisations d'interruption et la sortie $\overline{\text{RTS}}$ de contrôle du périphérique. Il est sélectionné quand $\text{RS}=0$ et $\text{R}/\overline{\text{W}} = 0$.

La signification des bits de ce registre est donnée par les tableaux de la figure 19. Les deux bits CR0 et CR1 permettent d'indiquer le rapport de division pour l'émetteur et le récepteur entre la fréquence d'horloge et la fréquence de transfert série des données. Ils sont aussi utilisés pour l'initialisation programmée de l'ACIA. Celle-ci met à zéro le registre d'état sauf les bits liés à $\overline{\text{CTS}}$ et $\overline{\text{DCD}}$ et initialise l'émetteur et le récepteur.

Les trois bits CR2, CR3 et CR4 indiquent le format des mots reçus ou transmis. Les deux bits CR5 et CR6 permettent le contrôle des interruptions en transmission. Le bit CR7 permet d'autoriser ou de masquer les interruptions générées par le récepteur.

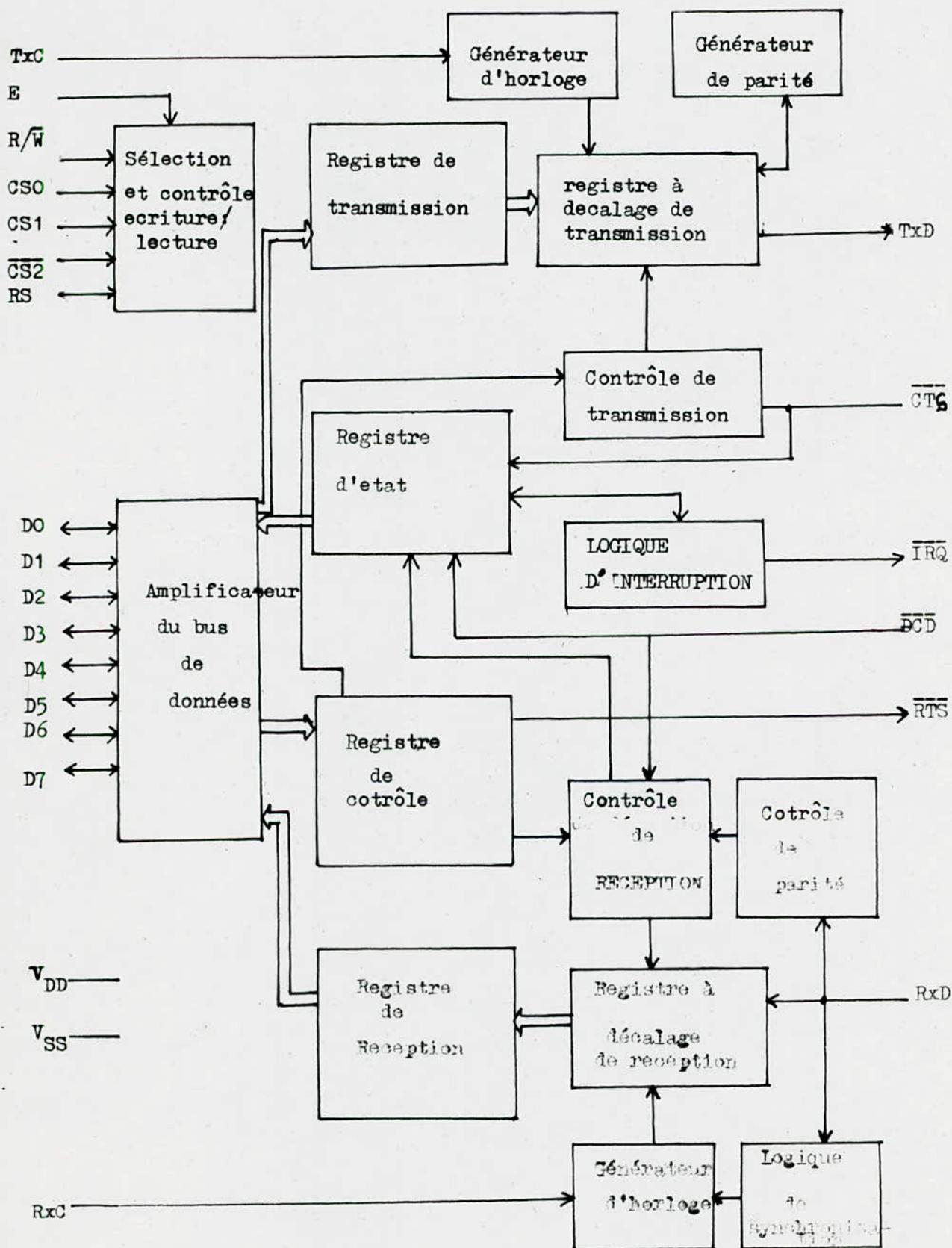


FIG-18 SCHEMA FONCTIONNEL DU MC 6850 (ACIA)

CR4	CR3	CR2	LONGUEUR DU MOT	PARITE	NOMBRE DE BITS STOP
0	0	0	7 bits	Paire	2
0	0	1	7 bits	Impaire	2
0	1	0	7 bits	Paire	1
0	1	1	7 bits	Impaire	1
1	0	0	8 bits	Sans	2
1	0	1	8 bits	Sans	1
1	1	0	8 bits	Paire	1
1	1	1	8 bits	Impaire	1

Transmetteur et receveur

CR1	CR0	RAPPORT DE DIVISION D'HORLOGE
0	0	÷ 1
0	1	÷ 16
1	0	÷ 64
1	1	Initialisation programmée

CR6	CR5	FONCTION
0	0	$\overline{RTS} = 0$ et interruptions du transmetteur inhibées
0	1	$\overline{RTS} = 0$ et interruptions du transmetteur validées
1	0	$\overline{RTS} = 1$ et interruption du transmetteur inhibées
1	1	$\overline{RTS} = 0$ et interruption du transmetteur inhibées émission d'un BREAK

Transmetteur

CR7	RECEPTEUR
0	Interruptions masquées
1	Interruptions autorisées

FIG-19 SIGNIFICATIONS DES BITS DU CCR DE L'ACIA

b- Le registre d'état : (SR)

Ce registre de 8 bits à lecture seule indique les états des registres de transmission et de réception, indique les erreurs logiques ainsi que l'état des lignes \overline{CTS} et \overline{DCD} . Le tableau de la figure 20 donne la signification des bits de ce registre. Notons que : Lorsque SR2 est à un, SR0 est forcé à 0 et quand SR3 est à un SR1 est forcé à zéro.

Fonction	bit	niveau	signification
Indicateur d'interruption \overline{IRQ}	SR 7	0	Pas d'interruption
		1	ACIA a envoyé une interruption
Erreur de parité P.E	SR6	0	Il n'y a pas d'erreur de parité
		1	Il y a une erreur de parité
Erreur de recouvrement OVRN	SR5	0	Reception correcte
		1	Perte de caractère en reception
Erreur de format F.E	SR4	0	Pas d'erreur de format
		1	Il y a une erreur de format
Ligne de contrôle en transmission \overline{CTS}	SR3	0	Transmission possible
		1	Transmission impossible
Porteuse de données en reception \overline{DCD}	SR2	0	Présente
		1	Absente
Registre de transmission TDRE	SR1	0	Plein
		1	Vide
Registre de reception RDRF	SR0	0	Vide
		1	Plein

FIG-20 SIGNIFICATION DES BITS DU REGISTRE D'ETAT (SR)

c- Registre de reception : (RDR)

Ce registre de 8 bits à lecture seule reçoit le mot transféré du registre à décalage lors d'une réception .

d- Registre de transmission : (TDR)

C'est un registre de 8 bits à écriture seule, dans lequel sera écrit le mot de 8 bits à transmettre par le MPU sur un front descendant de l'horloge E.

B2-3 FONCTIONNEMENT DE L'ACIA :

La communication série gérée par l'ACIA utilise la procédure de start-stop. Chaque caractère transmis ou reçu a un format de sept ou huit bits avec ou sans contrôle de parité. Il est précédé d'un bit start et suivi d'un ou deux bits stop. Chacun de ces bits est synchronisé par une horloge, mais la suite des caractères est à synchrone.

A la différence du PIA qui possède une entrée $\overline{\text{RESET}}$, l'ACIA à son propre circuit de mise sous tension qui le maintient dans un état inhibé jusqu'à son initialisation programmée, ceci afin d'éviter la transmission d'informations erronées. Cet état est obtenu par CR5=0 et CR6=1 du registre de contrôle, ce qui impose $\overline{\text{RTS}}$ à l'état haut donc sans action, et les interruptions du transmetteur sont inhibées.

D'autre part tous les bits sauf SR2 et SR3 du registre d'état sont mis à zéro.

Avant de programmer un mode de contrôle complet, la mise sous tension doit être suivie d'un master $\overline{\text{RESET}}$ ou initialisation programmée. Ceci est obtenu en imposant la combinaison 11 dans CR0 et CR1 du registre de contrôle. Après le master, les bits CR5 et CR6 ne sont plus inhibés. Le registre de contrôle peut alors être programmé.

a- Adressage de l'ACIA :

Bien que l'ACIA a quatre registres internes, il est vu par le MPU comme s'il occupait deux positions mémoire. Comme deux de ces registres sont à lecture seule (SR et RDR) et les deux autres à écriture seule (CR et TDR), la ligne $\text{R}/\overline{\text{W}}$ est utilisée en conjonction avec la ligne de sélection de registre pour permettre l'adressage des quatre registres de l'ACIA après validation du boîtier par les entrées CS0, CS1 et $\overline{\text{CS2}}$.

Le tableau suivant donne les combinaisons des lignes RS et $\text{R}/\overline{\text{W}}$ et les registres qu'elles sélectionnent :

RS(A0)	$\text{R}/\overline{\text{W}}$	REGISTRES SELECTIONNES
0	0	Ecriture du registre de contrôle CR
0	1	Lecture du registre d'état SR
1	0	Ecriture dans le registre de transmission
1	1	Lecture du registre de reception

b- Fonctionnement en transmission de l'ACIA :

En général, la séquence de transmission commence par la lecture du registre d'état de l'ACIA et ceci soit après une interruption soit pendant une boucle de test.

Si la lecture du registre d'état indique que le registre de transmission est vide (bit TDRE à 1), un caractère peut être écrit dans ce dernier. Le caractère est transféré dans un registre à décalage puis transmis en série par la ligne de transmission de données TxD, précédé par un bit de départ (start) suivi d'un ou deux bits d'arrêt (stop). Un bit de parité peut être optionnellement ajouté au caractère et sera transmis entre le dernier bit de données et le premier bit d'arrêt.

Le registre d'état peut être à nouveau lu pour tester le bit TDRE, après lecture du premier caractère dans le registre de transmission. S'il est vide un autre caractère peut être écrit même si le premier est en cours de transmission, ceci grâce au double registre.

En attendant la fin de la transmission du premier caractère le second caractère sera automatiquement transféré dans le registre à décalage pour être transmis. Le processus continue jusqu'à la fin de la transmission.

c- Fonctionnement en reception de l'ACIA :

Dans un système de transmission à synchrone, les données sont transmises de façon aléatoire, sans signal de synchronisation. Ce sont les bits start et stop qui vont permettre une synchronisation des bits du caractère reçu par rapport à l'horloge RxClk.

Une séquence de reception commence en général par la lecture du registre d'état afin de tester si un caractère a été reçu et si le registre de reception est plein (RDRF à un). Les données sont données en série par l'entrée de reception RxD. La possibilité de suppression des faux bits de départ causant des perturbations dans le fonctionnement, est assurée par le fait que la synchronisation des bits est effectuée après la reception de la moitié du bit de départ.

La validité du caractère reçu est contrôlée pendant la reception et positionne les bits concernés du registre d'état. Nous avons ainsi un test du format, de parité et de débordement. le caractère reçu est transféré dans le registre RDR après suppression des bits de départ, de parité et de stop. A ce moment les indicateurs d'état sont positionnés. La ligne d'horloge est reliée en interne avec une horloge dont le rapport de division est de 1/16 ou de 1/64. Ce rapport est choisi par programmation du registre de contrôle. La synchronisation est alors assurée par la première transition négative suivant une période de repos. Le bit de départ (start) est échantillonné durant les transitions positives de l'horloge externe pour vérifier qu'il ne s'agit pas d'une fausse indication.

Dans le cas d'un rapport de division 1/1 la synchronisation doit être externe.

L'horloge externe sera alors synchronisée par la donnée de la manière suivante :

Dés l'apparition de la première transition négative, après une période de repos du signal reçu, l'échantillonnage se produit sur le front montant de l'horloge externe RxClk et le bit start est chargé sur le front descendant suivant de l'horloge.

Après avoir présenté la carte et étudié en détail le microprocesseur MC 6809, les deux circuits d'interface utilisés ; nous passerons dans cette partie à l'étude du système réalisé.

Le schéma électrique général de la réalisation est donné sur la figure 21.

Pour faciliter cette étude nous la diviserons en cinq parties :

- L'unité centrale ;
- Le système de décodage ;
- Le système mémoire ;
- Les interfaces d'E/S destinés à l'usage de l'utilisateur ;
- Le système de gestion du clavier et des afficheurs.

A- L'UNITE CENTRALE (OU LE MC 6809)

Nous avons utilisé comme unité centrale le microprocesseur 68B09 de MOTOROLA de fréquence interne 2MHz, piloté par un quartz de 8MHz (LeMDX 080) dont les bornes sont reliées à la masse par des capacités de 18pF.

Comme il apparaît sur le schéma électrique, pour être désactivées, les entrées $\overline{\text{NMI}}$, $\overline{\text{TRQ}}$, $\overline{\text{FTRQ}}$, $\overline{\text{HALT}}$, $\overline{\text{MRDY}}$ et $\overline{\text{DMA/BREQ}}$ sont mises à l'état haut par des lignes les reliant à la tension d'alimentation à travers des résistances de 4,7 K Ω qui permettent, de réduire le courant d'entrée et d'imposer un niveau bas sur ces entrées quand on veut les activer.

Les broches de A0 à A15 sont directement reliées aux lignes du bus d'adresses et les broches de D0 à D7 aux lignes du bus de données.

Le bus de contrôle comprend les lignes suivantes :

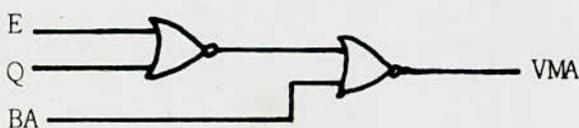
- La lignes $\overline{\text{TRQ}}$, reliée à la sortie de demande d'interruption $\overline{\text{TRQB}}$ du PIA système ;
- Les lignes $\overline{\text{NMI}}$ et $\overline{\text{FTRQ}}$ reliées à des Switchs (cavaliers) qui permettent la liaison de l'une ou l'autre avec les sorties de demande d'interruption de l'ACIA ou du PIA utilisateur ;
- La ligne $\overline{\text{R/W}}$ reliée aux entrées de validation de lecture / écriture des RAM et des interfaces ;
- La ligne E reliée aux entrées de synchronisation du trace timer et des interfaces ;
- La ligne du signal VMA :

Comme le signal de validation des adresses mémoire n'existe pas sur le MC 68B09 nous l'avons réalisé à l'aide d'une logique externe en utilisant les signaux d'horloge E et Q et le signal BA de disponibilité du bus. Le signal étant relié au circuit de décodage d'adresses, il permet de désactiver celui-ci lorsque l'adresse présente sur le bus n'est pas stable ou lorsque les bus d'adresses et de données ne sont ^{Pas} gérés par le MPU.

La fonction logique de ce signal est donnée par :

$$\text{VMA} = (\text{E} + \text{Q}) \cdot \overline{\text{BA}} = \overline{(\overline{\text{E} + \text{Q}}) + \text{BA}}$$

Sa réalisation s'effectue comme suit :

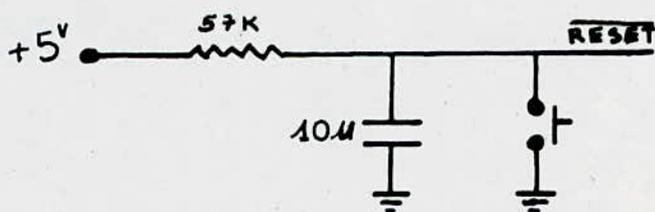


Cette configuration est choisie car les adresses présentes sur le bus sont stables à partir d'un front montant de Q jusqu'au prochain front descendant de E avec BA à l'état bas comme le montre les chronogrammes de lecture et d'écriture des figures 22a et 22b.

- La ligne $\overline{\text{RESET}}$:

L'entrée de réinitialisation $\overline{\text{RESET}}$ du 68B09 est reliée à un simple circuit RC comme il apparaît sur le schéma de la figure 21. Ce circuit permet de maintenir à un niveau bas cette entrée pendant un temps nécessaire à la stabilisation du quartz après la mise sous tension et à la prise en compte de cet état par le système de réinitialisation. Dès la mise sous tension, le condensateur entame sa charge à travers une résistance, avec une constante de charge τ , permettant de maintenir à ses bornes une tension inférieure à 0,8V qui est la tension de seuil de l'état bas de l'entrée $\overline{\text{RESET}}$ durant environ 100 ms.

La borne du condensateur qui est reliée à l'entrée $\overline{\text{RESET}}$ est reliée aussi à un interrupteur marche/arrêt permettant lorsqu'il est fermé de la mettre à la masse et de générer par conséquent un niveau bas sur la ligne $\overline{\text{RESET}}$ qui activera le processus de réinitialisation du système.



L'équation de charge du condensateur étant donnée par :

$$V_c = V_0 (1 - e^{-t/\tau}) \quad \text{avec} \quad \tau = RC \quad \text{et} \quad V_0 = 5V$$

permet pour $V_c = 0,8V$ (seuil) et $t = 100$ ms de déduire

$$\tau = \frac{t}{\ln \frac{50}{42}} = 0,573 \text{ s}$$

Pour $R = 57,3 \text{ K}$ on a $C = 10 \text{ uF}$.

Pour d'éventuelles liaisons avec des systèmes extérieurs, le bus d'adresses, le bus de données et toutes les lignes de contrôle sauf la ligne $\overline{\text{TRQ}}$, sont reliés à un connecteur à travers des amplificateurs de puissance appelés buffers.

Nous avons utilisé 3 buffers à 8 entrées unidirectionnelles, les 74 LS 244 qui sont des circuits intégrés en boîtier DIL à 20 broches, monotension (+5V). Deux d'entre-eux sont utilisés pour les 16 lignes du bus d'adresses et le troisième pour amplifier les lignes R/\overline{W} , E, Q, BA, BS et le signal VMA. Les autres lignes de contrôle sont directement reliées au connecteur. Les deux entrées enable de ces circuits étant reliées à la masse, ils ne seront jamais dans l'état haute impédance.

Pour les lignes du bus de données, nous avons utilisé un buffer à 8 entrées bidirectionnelles, de 74 LS 245 qui est un circuit intégré en boîtier DIL à 20 broches, monotension; son entrée DIR de contrôle de direction est reliée à la ligne R/\overline{W} , permettant de sélectionner la direction de transfert de données. L'entrée enable de ce circuit est reliée à la ligne d'adresse A 14 pour permettre de valider le buffer quand elle est à l'état bas et de le mettre à l'état haute impédance dans le cas contraire. Elle permet ainsi l'adressage d'une zone mémoire externe allant de l'adresse | 0000 à BFFF |.

B- LE SYSTEME D'ADRESSAGE

Les seize lignes du bus d'adresses permettent au MC 68 B 09 d'avoir une capacité mémoire adressable de 64 KO (2^{16}). Comme les entrées d'adresses des différents circuits mémoire et d'interface sont toutes reliées directement au bus d'adresse, une combinaison sur ce bus sélectionne une position mémoire dans chacun de ces circuits. Il en résultera une interférence d'informations sur le bus de données perturbant ainsi le fonctionnement du système. Pour palier à ce problème, il est nécessaire de réaliser un système de décodage validant un seul circuit à la fois pour une combinaison présente sur le bus d'adresse. Un tel système est réalisable, car tous présentent des entrées de validation de boîtier CS et \overline{CS} en nombre variable. Ce sont ces entrées qui commandent l'état haute impédance des sorties tant qu'elles n'ont pas reçu la bonne configuration, c'est à dire un état haut sur les entrées CS et un état bas sur les entrées \overline{CS} .

Pour cette raison l'espace mémoire du microprocesseur est réparti en zones, réservées chacune à un circuit.

Il y a deux modes d'adressage possibles :

1° par sélection linéaire d'adresses :

Cette méthode est généralement utilisée dans les petits systèmes où la mémoire implantée est très inférieure à 64 KO. Certaines lignes du bus d'adresses sont directement reliées aux entrées de sélection de boîtiers en plus de celles reliées aux entrées d'adresses, de manière à ne sélectionner qu'un seul circuit à la fois.

2° Par décodage complet du bus d'adresses :

Dans cette méthode le décodage est réalisé au moyen de portes logiques, de décodeurs ou de réseaux logiques programmables.

Les seize lignes du bus d'adresses seront utilisées pour valider un seul boîtier par l'intermédiaire de circuits à portes ou de décodeurs. Ainsi chaque octet mémoire n'aura qu'une seule adresse, ce qui permettra d'adresser effectivement 64 KO, avec possibilité d'organiser la mémoire en pages.

Pour notre réalisation, bien que nous n'avons pas utilisé entièrement les 64 KO mémoire adressables, nous avons un nombre assez important de boîtier à décoder. Ce qui nous a amené à utiliser un système de décodage à l'aide de portes logiques et un double décodeur.

Avant d'étudier le système de décodage, rappelons d'abord les différents circuits à adresser, c'est à dire qui sont vus par le MPU comme des positions mémoire. Nous avons deux circuits EPROM d'une capacité de 2 KO chacun, deux circuits RAM d'une capacité de 2 KO chacun, deux PIA vus chacun comme quatre positions mémoire et un ACIA vu comme deux positions mémoire.

Ainsi une répartition de l'espace mémoire adressable du 68 B 09, suivant la capacité mémoire de chaque circuit s'impose et doit tenir compte du fait que les positions mémoire d'un circuit doivent être sélectionnées par des adresses consécutives.

La table suivante nous donne les zones mémoire allouées aux différents circuits :

FFFF	EPROM système
F800	
F7FF	EPROM utilisateur
F000	
EEEE	Buffer de l'ACIA
EFFE EFFA	Reservée
EFF9	
EFF8	ACIA
EFF7	
EFF4	PIA Utilisateur
EFF3	
EFF0	PIA Système
EFEF	
E800	Reservée
E7FF	
E000	RAM Système
DFFF	
D800	RAM d'extension
D7FF	
C000	Reservée
BFFF	
0000	Extension mémoire externe

Nous déduisons alors le tableau des différentes configurations des lignes d'adresses pour les différents circuits. Figure 23.

Le système d'adressage que nous avons réalisé fait intervenir toutes les lignes du bus d'adresses dans l'adressage de chaque circuit.

Nous remarquons sur le tableau des configurations des lignes d'adresses que certaines lignes restent dans le même état durant l'adressage d'un circuit. Nous exploitons alors cette caractéristique pour mettre au point notre système de décodage.

C'est ainsi que nous remarquons que les lignes A15, A14 et A13 sont à l'état haut dans l'adressage de tous les circuits sauf celui de la RAM d'extension. Nous les utilisons alors pour la validation du décodeur en conjonction avec le signal de validation d'accès mémoire VMA. La configuration des lignes A11 et A12 qui changent selon la zone mémoire à adresser fait que ces lignes attaquent directement les entrées A1 et B1 de sélection de sorties du premier décodeur.

Ainsi chaque configuration sur les lignes A11 et A12 sélectionnera un seul circuit en portant la sortie du décodeur reliée à son entrée \overline{CS} à l'état bas. C'est alors que la combinaison 11 sur ces lignes validera l'EPROM système, la combinaison 10 l'EPROM utilisateur, la combinaison 00 la RAM système et la combinaison 01 la zone mémoire qui est réservée aux interfaces.

Pour sélectionner les différents circuits d'interfaces on remarque qu'il n'y a que les configurations des lignes A3 et A2 qui changent, suivant l'interface à adresser. Les lignes A4, A5, A6, A7, A8, A9 et A10 restent à l'état haut, et sont utilisées en conjonction avec la sortie du premier décodeur pour valider le deuxième décodeur. Les lignes A2 et A3 sont reliées aux entrées de sélection de sorties 2A et 2B du deuxième décodeur. Pour sélectionner selon la configuration qu'elles présentent l'un des interfaces. Ainsi la combinaison "00" valide le PIA système, la combinaison "01" le PIA utilisateur et la combinaison "10" l'ACIA.

Pour l'adressage de la RAM d'extension les lignes A15 et A14 restent à l'état haut tandis que, A13 passe à l'état bas. L'état de ces lignes sera utilisé en conjonction avec le signal VMA pour valider cette RAM.

Pour l'adressage de la mémoire externe d'extension, seul l'état de la ligne A14 change, elle sera donc utilisée pour la validation de cette zone mémoire. Les circuits utilisés pour réaliser ce système de décodage sont :

- 3 portes "and" à trois entrées (Le 74 LS 11) ;
- 1 porte "nand" à quatre entrées (le 74 LS 20) ;
- 1 porte "and" à deux entrées (Le 74 LS 08) ;
- 4 portes "nand" à deux entrées (Le 74 LS 00)
- 1 double décodeur (Le 74 LS 139) dont la table de vérité est la suivante :

INPUTS			OUTPUTS			
ENABLE G	SELECT B A		Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H: Représente un état haut

L: Représente un état bas

X: Représente un état indéterminé.

FIG-23 TABLEAU DES CONFIGURATIONS DES LIGNES D'ADRESSES

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Circuit sélectionné
1	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	ROM système (FFFF-F800) 2 K. Octets
1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	ROM utilisateur (F000-F7FF) 2 K.O
1	1	1	0	1	1	1	1	1	1	1	1	0	0	X	X	PIA Système (EFF0 - EFF3)
1	1	1	0	1	1	1	1	1	1	1	1	0	1	X	X	PIA Utilisateur (EFF4 - EFF7)
1	1	1	0	1	1	1	1	1	1	1	1	1	0	X	X	ACIA (EFF8 - EFF9)
1	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	RAM Système (E000 - E7FF) 2 K.O
1	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	RAM d'extension (800 - DFFF) 2 K.O
1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	Zône d'extension mémoire externe (BFFF- 0000) 48 K.O
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

X : Représente l'état indéterminé.

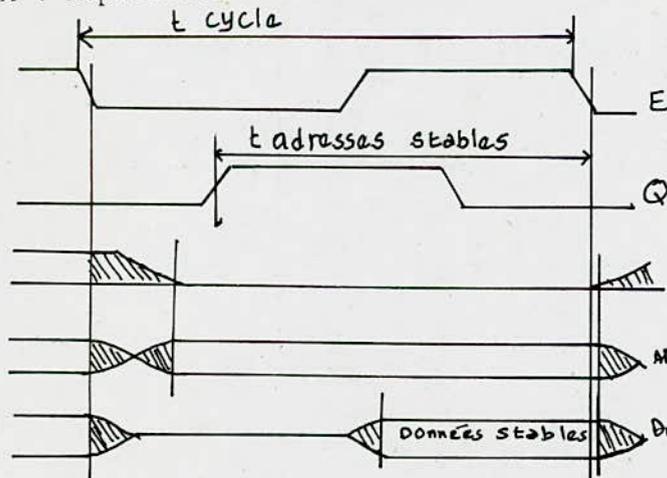


FIG-22b : CHRONOGRAMME DE LECTURE DE DONNEES

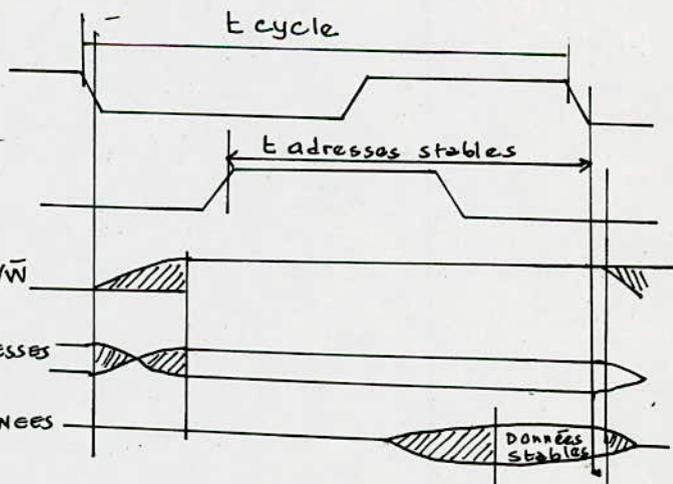


FIG-22a : CHRONOGRAMME D'ECRITURE DE DONNEES

Tous les systèmes à microprocesseur necessitent des programmes de gestion qui sont stockés dans des mémoires d'une manière définitive et ne peuvent être effacés par programme. L'ensemble de ces programmes constitue le moniteur. De même que sont indispensables les circuits mémoire dont le contenu peut être changé au gré de l'utilisateur et dont l'information se perd en cas de coupure de l'alimentation.

Le premier type de mémoire est appelé "mémoire morte" et comprend les ROM, les PROM, les EPROM, etc... Tandis que le deuxième type est appelé mémoire vive et comprend les RAM dynamiques et statiques.

Les circuits mémoire RAM et ROM sont des mémoires matricielles caractérisées par le fait que le temps d'accès à une position mémoire est indépendant du numéro de la cellule adressée.

1° Les mémoires vives : (RAM)

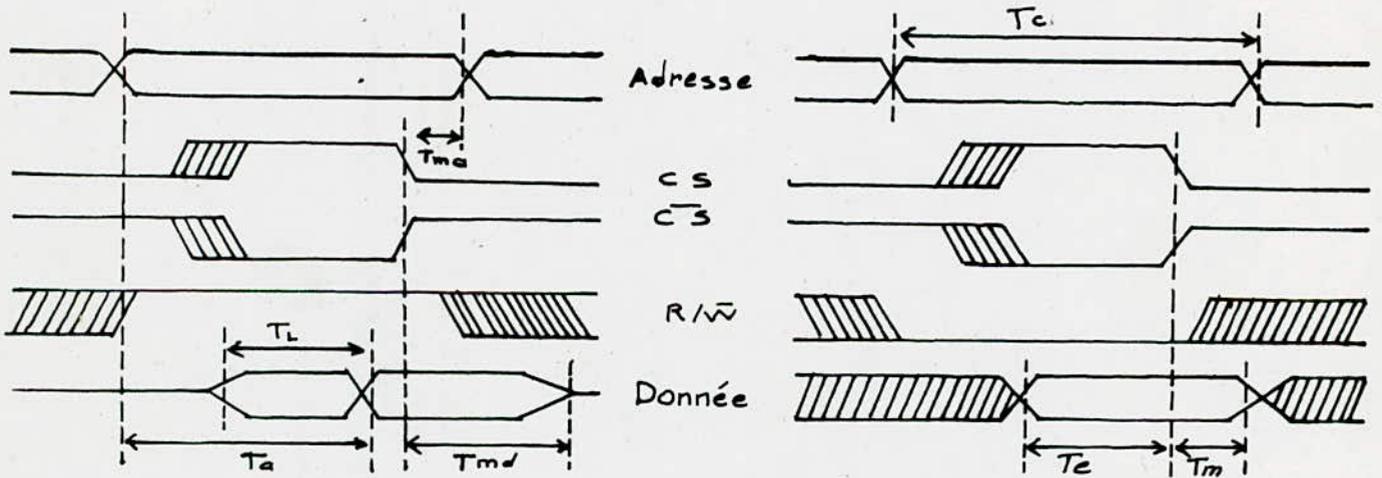
a- Les RAM statiques :

Ce sont des mémoires volatiles, c'est à dire qui perdent l'information qu'elles contiennent en cas de coupure de l'alimentation. Elles sont réalisées en plusieurs technologies notamment en technologie MOS, C MOS, bipolaire ou ECL.

Ces circuits sont toujours monotension (+5V), compatibles TTL et à sorties trois états ou à collecteurs ouverts (Exclusivement pour certaines bipolaires et ECL).

Du fait de leur fonctionnement statique ces circuits n'ont besoin d'aucun signal d'horloge ou de circuit de rafraîchissement. L'inconvénient majeur de ces types de RAM est leur intégration limitée.

Nous donnons ici les chronogrammes de lecture / écriture de ces RAM.



CHRONOGRAMME DE LECTURE

CHRONOGRAMME D'ECRITURE

* Tma : Temps de maintien de l'adresse.

Tl : Temps de lecture.

Ta : Temps d'accès.

Tmd : Temps de maintien de la donnée.

* Tc : Temps de cycle.

Tm : Temps de maintien

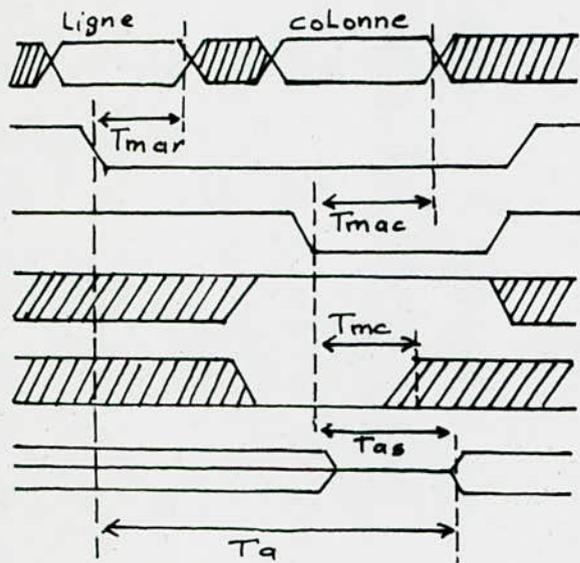
Te : Temps d'établissement de la donnée avant la fin de CS

Notons que les circuits fabriqués en technologie bipolaire ont un temps d'accès inférieur à celui des circuits fabriqués en technologie MOS.

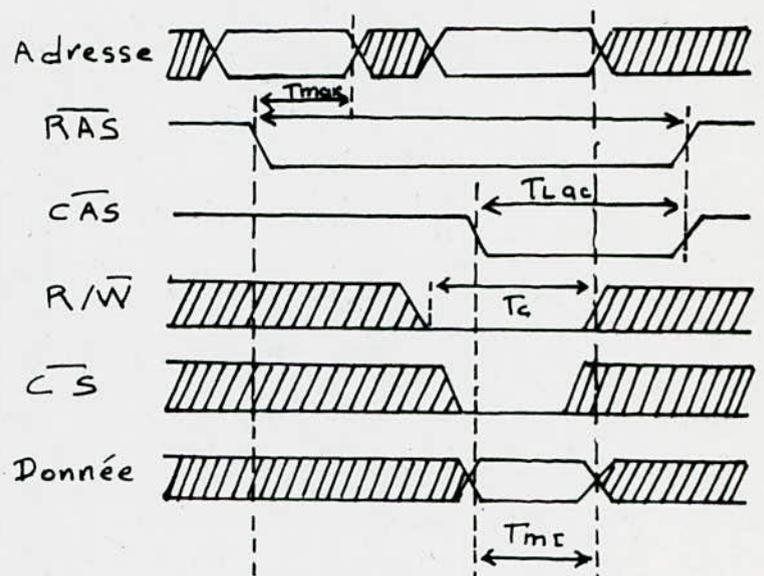
b- Les RAM dynamiques :

La limitation d'intégration des RAM statiques de grandes capacités, a poussé à la création de RAM dynamiques, dont la fabrication est plus simple et la densité d'intégration plus grande. L'élément mémoire est la capacité de gachette-source d'un transistor MOS, mais la résistance gachette-source n'étant pas infinie cette capacité se déchargera ; il faudra donc la recharger pour maintenir l'information, on dit qu'il faut rafraîchir la mémoire. L'inconvénient de ces circuits est qu'ils nécessitent une logique de rafraîchissement qui rend leur utilisation complexe.

Notons que ces mémoires ne stockent que des mots de 1 bit. Les schémas suivants donnent les chronogrammes de lecture et d'écriture de ces types de RAM.



CHRONOGRAMME DE LECTURE



CHRONOGRAMME D'ECRITURE

T_{mar} : Temps de maintien des adresses de rangées.

T_{mac} : Temps de maintien des adresses de colonnes.

T_{mc} : Temps de maintien du chip select.

T_{as} : Temps d'accès depuis le strobe CAS.

T_a : Temps d'accès.

T_{lsl} : Longueur de l'impulsion de strobe des lignes.

T_{lac} : Largeur de l'impulsion d'adressage des colonnes.

T_c : Longueur de l'impulsion de commande.

T_{mi} : Temps de maintien de l'information.

Pour ces circuits, chaque cycle de lecture/écriture est suivi d'un cycle de rafraîchissement mémoire.

2° Les mémoires mortes :

Ce sont des mémoires non volatiles, c'est à dire qui ne perdent l'information qu'elles contiennent même après la coupure de l'alimentation.

a- Les ROM (Read only memory):

Ce sont des mémoires dont l'information est enregistrée à demeure par le constructeur, et ne peut donc être modifiée.

Le coût élevé pour la réalisation des masses, impose leur utilisation dans des applications en grandes séries. Elles sont réalisées en technologie MOS et en bipolaire. Comme pour les RAM la technologie bipolaire est plus rapide que la MOS.

b- Les PROM : (Programmable read only memory)

Ce sont des mémoires mortes mortes programmables par l'utilisateur de manière irréversible. Elles sont conçues pour la mise au point de systèmes et la réalisation de programmes identiques en faible nombre d'exemplaires.

Elles sont réalisées en technologie MOS et bipolaire avec des sorties trois états ou à collecteur ouvert. La programmation s'effectue par claquage de fusibles ou de jonction ou par stockage de charges.

c- Les EPROM : (Erasable programmable read only memory)

Ce sont des mémoires programmables et effaçables par l'utilisateur. Elles sont utilisées dans des systèmes qui demandent des modifications relativement fréquentes du contenu de la mémoire ou lorsque le programme définitif ne peut être figé en ROM ou PROM qu'après la mise au point finale.

Elles sont réalisées en technologie MOS. Leur programmation s'effectue par stockage de charges et leur effacement s'effectue par exposition au rayonnement d'une lampe ultra-violet. Notons qu'il existe des circuits effaçables électriquement, permettant un effacement sélectif et plus rapide, ces circuits sont appelés EAROM.

3° Les circuits mémoires utilisés :

Le système mémoire comprend, une EPROM système contenant le moniteur (La EF 2516), une RAM (La SY 2128) dont 128 octets sont utilisés par le système (Moniteur) et le reste laissé à l'usage de l'utilisateur.

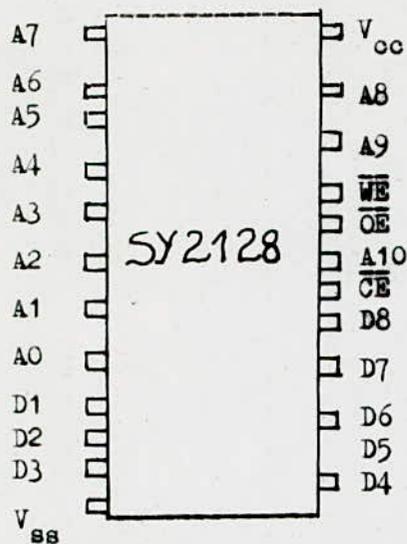
Nous avons aussi prévu l'utilisation d'une RAM d'extension ainsi qu'une EPROM laissées à l'usage de l'utilisateur. Tous ces circuits mémoires sont d'une capacité de 2 K.O chacun. La RAM d'extension est de même type que celle du système (SY 2128) et l'EPROM utilisateur est de même type que l'EPROM système (EF 2516). Ces circuits mémoires sont choisis car ils présentent un temps d'accès inférieur au temps d'accès du MPU (qui est de 320 ns) et rend possible une lecture ou une écriture sur mémoire.

a- La RAM SY 2128 et ses caractéristiques :

- Temps d'accès maximum $t_{ac} = 120$ ns
- RAM statique sans entrée d'horloge
- Elle est validée automatiquement par CE
- Elle a le même temps de cycle et d'accès
- Tension d'alimentation +5V ($\pm 10\%$)
- Compatible TTL
- Sorties à trois états
- Brochage compatible avec la 2716

- Fabriquée en technologie MOS à canal N
- Organisée ne 2048 mots de 8 bits

Son schéma de brochage est le suivant :



A : Adresse

D : donnée E/S

\overline{WE} : Lecture / Ecriture

\overline{OE} : Validation des sorties

\overline{CE} : Selection de boîtier.

Son branchement est réalisé en reliant les entrées d'adresses A0, ... A10 au bus d'adresses et les lignes de données au bus de données. L'entrée \overline{WE} est reliée à la ligne R/\overline{W} du bus de contrôle.

b- Caractéristiques de l'EPROM EF 2516 :

- Alimentation en monotension (5V \pm 10 %)
- Mode de basse consommation automatique
- Organisée ne 2048 mots de 8 bits
- Dissipation en puissance, faible
- Compatible TTL à la lecture et à la programmation
- Brochage compatible avec la EF 2716
- Temps d'accès maximum pour la EF 25 A 16 $t_{ac} = 350$ ns
- Fabriquée en technologie MOS à canal N

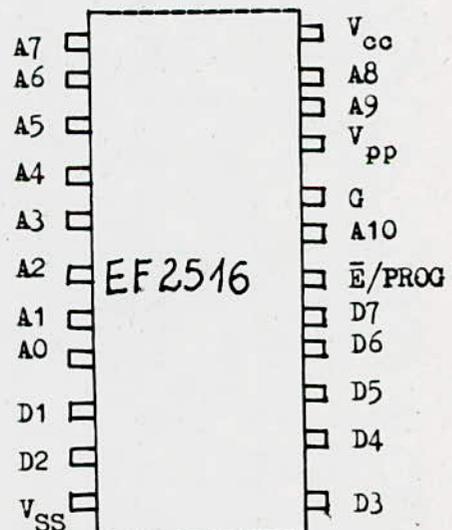
Son schéma de brochage est le suivant :

A : Entrée d'adresse

D : Données en E/S

$\overline{E}/Prog$: Validation du circuit / programme

\overline{G} : Validation du boîtier



pour la programmation, VPP doit être à plus 25 V et la tension d'alimentation à + 5V.

L'entrée G à l'état haut.

Les données à programmer sont entrées en mots de 8 bits à travers les broches de données. Comme les bits de la mémoire sont à un après chaque opération d'effacement, seuls des zéros seront programmés quand des zéros et des uns sont entrés dans le mot de donnée.

Les adresses peuvent être programmées individuellement, séquentiellement ou par accès aléatoire en appliquant une impulsion programme maximale de 55 ms, après le pré-établissement des données, sur l'entrée \bar{E}/Prog .

Son branchement est réalisé en reliant les broches d'adresses aux lignes du bus d'adresses et les broches de données au bus de données. \bar{G} est relié à la ligne de sélection du boîtier (Sortie du décodeur).

D- LES INTERFACES D' E/S A L'USAGE DE L'UTILISATEUR

1° L'interface à E/S parallèles : (Le MC 6821)

Pour le dialogue avec des périphériques externes à E/S parallèles, nous avons prévu le PIA MC 6821, dont le fonctionnement sera choisi par l'utilisateur dans ses applications nécessitant une organisation parallèle des données.

Comme il apparaît sur le schéma électrique de la figure 23, ce PIA est relié au MPU par l'intermédiaire du bus de données, de la ligne d'horloge, de la ligne R/\bar{W} et de deux lignes d'adresses A0 et A1 qui permettent de donner aux registres internes des adresses consécutives. L'entrée de réinitialisation est reliée directement à la ligne $\overline{\text{RESET}}$ du bus de contrôle. Les deux entrées CS0 et CS1 de sélection de boîtier sont mises à l'état haut par liaison à +5V. Ainsi seule l'entrée $\overline{\text{CS2}}$, reliée à la sortie du décodeur, permettra de valider ou non ce PIA.

Les deux sorties de demande d'interruption $\overline{\text{TRQA}}$ et $\overline{\text{TRQB}}$, sont reliées à des switches qui permettront de les connecter à l'une ou l'autre des entrées de demande d'interruption $\overline{\text{NMI}}$ ou $\overline{\text{FIRQ}}$ du MPU.

Les lignes des ports A et B, ainsi que les lignes CA1, CA2, CB1 et CB2 sont reliées à un connecteur qui permettra la liaison avec les périphériques extérieurs.

2° L'interface à E/S séries : (Le MC 6850)

Afin de mettre à la disposition de l'utilisateur, un moyen permettant le dialogue entre le MPU et les périphériques à entrées/sorties séries, nous avons prévu l'ACIA MC 6850 communicant avec l'extérieur par l'intermédiaire d'un connecteur utilisant la norme RS 232 C.

L'ACIA est relié au MPU par l'intermédiaire du bus de données, de la ligne d'horloge E, de la ligne R/\bar{W} et de la ligne d'adresse A0. La sortie de demande d'interruption est reliée à un switch qui permettra de la connecter à l'une des deux entrées de demande d'interruption $\overline{\text{NMI}}$ ou $\overline{\text{FIRQ}}$ du MPU. Les entrées CS0 et CS1 de sélection de boîtier étant mises à l'état haut, seule l'entrée $\overline{\text{CS2}}$ reliée au décodeur permettra de valider ou non l'ACIA.

Pour avoir la même fréquence de transmission et de réception, les deux entrées d'horloge RxClk et TxClk sont connectées à un switch qui permettra de sélectionner l'une des fréquences délivrée par un BAUD-RATE.

Le BAUD RATE utilisé est le MC 14411 qui est un circuit intégré à 24 broches, compatible TTL, monotension qui est fabriqué en technologie CMOS.

Il possède un oscillateur piloté par un quartz externe de 1,8432 MHz, qui délivre la fréquence principale f_0 . Il utilise un diviseur de fréquence qui permet d'avoir 16 sorties de fréquences différentes.

Deux entrées RSA et RSB permettent de sélectionner l'un des quatre modes de fonctionnement du BAUD RATE.

ETATS DES ENTREES		MODE SELECTIONNE
RSB	RSA	
0	0	x 1
0	1	x 8
1	0	x 16
1	1	x 64

Nous avons choisi le fonctionnement en mode "x1" en mettant les entrées RSA et RSB à la masse. L'entrée de réinitialisation est connectée à la ligne RESET du bus de contrôle. Le tableau suivant donne les fréquences en Hertz selon le mode sélectionné des 16 sorties du BAUD RATE /

SORTIES	FREQUENCE DE SORTIE EN Hz			
	x 64	x 16	x 8	x 1
F1	614,4 K	153,6 K	76,8 K	9600
F2	460,8 K	115,2 K	57,6 K	7200
F3	307,2 K	76,8 K	38,4 K	4800
F4	230,4 K	57,6 K	28,8 K	3600
F5	153,6 K	38,4 K	19,2 K	2400
F6	115,2 K	28,8 K	14,4 K	1800
F7	76,8 K	19,2 K	9600	1200
F8	38,4 K	9600	4800	600
F9	19,2 K	4800	2400	300
F10	12,8 K	3200	1600	200
F11	9600	2400	1200	150
F12	8613,2	2153,3	1076,6	134,5
F13	7035,5	1758,8	879,4	109,9
F14	4800	1200	600	75
F15	921,6 K	921,6 K	921,6 K	921,6 K
F16	1,843 M	1,843 M	1,843 M	1,843 M

On notera que seules les sorties F1, F2, F3, F4, F5, F7, F9, F11 et F14 sont utilisées.

Afin de pouvoir utiliser la carte comme équipement terminal de données (D T E) ou comme équipement de communication de données (DCE), nous avons réalisé un système utilisant un swich qui permet de décaler les sorties de l'ACIA sur le connecteur .

Les liaisons entre l'ACIA et le connecteur qui peuvent être obtenues sont les suivantes:

- L'entrée de réception de données RxD de l'ACIA est reliée à la ligne de réception de données ou à la ligne d'émission de données du connecteur.
- La sortie de transmission de données TxD de l'ACIA est reliée à la ligne d'émission de données ou à la ligne de réception de données du connecteur .
- La sortie $\overline{\text{RTS}}$ de demande d'émission est reliée à la ligne équipement de données prêt ou à la ligne de demande à émettre du connecteur .
- L'entrée inhibition de l'émetteur $\overline{\text{CTS}}$ est reliée à la ligne de demande à émettre ou à la ligne équipement de donnée prêt du connecteur .

Ces liaisons s'effectuent à travers des portes NAND de conversion de logique : Les MC 1488 pour les sorties de l'ACIA et les MC 1489 pour ses entrées.

Pour obtenir une plus grande souplesse d'utilisation de l'ACIA , la programmation de son registre de contrôle s'effectue par l'intermédiaire d'un buffer relié à un switch. Ainsi la valeur à charger dans ce registre est lue sur les sorties du buffer à l'adresse EFFF. Les entrées de ce dernier sont mises à l'état haut à travers des résistances de $4,7 \text{ K}\Omega$ et sont reliées à un switch qui permet de les mettre à la masse .

1° Le système d'affichage :

Le système d'affichage comprend six afficheurs à sept segments, à anode commune.

Quatre de ces afficheurs sont utilisés pour visualiser soit l'adresse d'une case mémoire sélectionnée ou bien le contenu des registres internes du MPU. Les deux autres affichent soit le contenu de l'adresse mémoire soit le nom du registre interne visualisé. Les sorties PB0 à PB5 du port B d'un PIA attaquent chacune la base d'un transistor PNP (Le 2N 2905) dont, le collecteur attaque l'anode d'un des six afficheurs et l'émetteur relié à la tension d'alimentation (+5V).

Un état haut appliqué à la base d'un transistor le bloquera, tandis que qu'un état bas polarisera sa jonction base-émetteur en direct et entraînera sa conduction validant ainsi l'afficheur qui lui correspond.

Les afficheurs sont validés à tour de rôle par un décalage du niveau bas sur les sorties du port B.

Les cathodes d'un même segment sur chaque afficheur, sont reliées au collecteur d'un même transistor NPM (Le 2N 2219). La base de ce transistor est attaquée par l'une des sept sorties (De PA0 à PA6) du port A du PIA et son émetteur est relié à travers une résistance à la masse. Ainsi chaque ligne du port A, validera lorsqu'elle est à l'état haut, un même segment, sur chaque afficheur, en entraînant la conduction du transistor qui lui correspond. Par contre lorsque une ligne du port A (PA0 à PA6) est à l'état bas, elle bloquera le transistor et permettra d'éteindre le segment qui lui correspond quel que soit l'état des anodes des afficheurs. Les sorties PA0 à PA6 sont donc utilisées pour le transfert du code sept segments de la valeur à afficher. Le code transféré varie à chaque fois, en fonction de la donnée à afficher. Ainsi l'état bas, qui ne doit être présent que sur l'une des six lignes du port B est maintenu pendant environ 25 ms avant d'être décalé vers la ligne suivante. On obtient alors l'impression d'un affichage continu permettant la visualisation d'une donnée (de 3 octets), sur les six afficheurs.

2° La gestion du clavier :

Parmi les vingt touches du clavier organisées en quatre et cinq lignes, seize sont des touches numériques et quatre des touches de commande. Nous avons ajouté quatre boutons poussoirs, comme touches de commande, constituant une sixième ligne. Chaque ligne est reliée à l'une des sorties PB0 à PB5 du PIA de manière à ce que chacune des six sorties du port B corresponde à une seule ligne. Les quatre colonnes mises à l'état haut à travers des résistances de 4,7 K Ω , sont reliées à une porte "nand" à quatre entrées (La 74 LS 20) dont la sortie attaque la ligne d'interruption CB1 programmée pour être activée par un front montant. Ces colonnes sont aussi reliées aux entrées de données d'un multiplexeur (Le 74 LS 153), dont les deux entrées de sélection A et B sont reliées aux sorties PB6 et PB7 du port B, son entrée de validation est mise à la masse et sa sortie attaque l'entrée PA7 du port A. L'envoi d'un niveau bas sur toutes les lignes engendre ; lorsqu'une touche est enfoncée, un état bas sur une colonne et entraîne par l'intermédiaire de la porte "nand" un front actif sur l'entrée CB1.

Comme la sortie de demande d'interruption \overline{TRQB} du PIA est reliée à l'entrée \overline{TRQ} du MPU, la transition sur CBI provoquera l'interruption du programme en cours et exécutera un programme de reconnaissance de la touche enfoncée qui consiste à rechercher le code de sa ligne et celui de sa colonne.

Recherche du code de la colonne et du code de la ligne :

L'état étant présent sur l'une des entrées de données du multiplexeur, des combinaisons sont envoyées sur les lignes PB6 et PB7 en testant à chaque fois l'état de l'entrée PA7. Lorsque cette dernière passe à l'état bas, la combinaison présente sur PB6 et PB7, donne le code de la colonne et sera maintenue pour la reconnaissance du code de la ligne. Cette dernière commence, par l'envoi d'un niveau bas sur la première ligne et d'un niveau haut sur les cinq autres (PB0 à PB5), suivi du test de l'entrée PA7. Si cette dernière est à l'état haut, un décalage du niveau bas sur les lignes est effectué, jusqu'à ce que l'entrée PA7 passe à l'état bas.

La combinaison présente sur les lignes PB0 à PB5 du port B représente alors le code de la ligne de la touche enfoncée.

L'information ligne et l'information colonne constituent le code de cette touche. Ce code est utilisé pour adresser une table à partir de laquelle la valeur ou la fonction de la touche est déterminée.

Comme l'affichage et la gestion du clavier utilisent un même PIA, ces deux fonctions sont exécutées en deux séquences distinctes. Ainsi après chaque période de 15 ms de maintien d'une valeur sur un afficheur, une séquence de quelques ms () sera consacrée à la gestion du clavier.

E - UTILISATION DU CLAVIER

Le clavier possède 16 touches hexadécimales numérotées de 0 à F et 4 touches de commande (RD, M, GO, CL). A droite du clavier sont disposés 5 boutons poussoir de commande (RS, P/B, OF, MV, FU)

Rôle des différentes touches:

RD : Touche de visualisation du contenu des registres.

Le passage d'un registre à l'autre a lieu après l'appui sur la touche GO. On pourra visualiser dans l'ordre: PC, D, DP, X; Y, U, S, CC. Le contenu apparaît sur les 4 afficheurs de droite et le nom du registre sur les 2 afficheurs de gauche.

M : Touche de visualisation du contenu des adresses mémoire.

Après l'introduction d'une adresse, l'appui sur cette touche fera apparaître son contenu sur les 2 afficheurs de droite.

GO : Touche de lancement d'un programme.

Après l'introduction de l'adresse de début de programme, l'appui sur cette touche fera exécuter le programme. Le contenu du PC sera alors affiché.

CL : Touche d'effacement.

Elle permet de sortir d'une commande et d'effacer les points d'arrêt après l'introduction de leurs adresses.

P/B : Touche d'exécution pas à pas d'un programme et d'insertion des points d'arrêt.

L'incrémentation lors d'une exécution pas à pas s'effectue par l'appui sur la touche GO.

L'insertion d'un point d'arrêt (5 points au maximum) s'effectue par l'introduction de l'adresse où il sera exécuté.

OF : Touche de calcul d'offset.

L'exécution de ce calcul s'effectue comme suit :

- on introduit l'adresse de départ;
- on appuie sur cette touche

- on introduit l'adresse d'arrivée;
- on appuie sur GO;

On verra alors affiché le déplacement et le type de branchement relatif (long ou court).

MV : Touche de transfert d'un programme d'une zone mémoire à une autre. L'exécution du transfert s'effectue comme suit :

- on introduit l'adresse de départ du programme;
- on appuie sur la touche MV;
- on introduit l'adresse de la fin du programme;
- on appuie sur la touche OF;
- on introduit l'adresse de destination;
- on appuie sur la touche GO.

FU : Touche de commande utilisateur.

L'appui sur cette touche puis sur une touche hexadécimale branchera vers une table contenant les commandes programmées par l'utilisateur.

RS : Touche de réinitialisation (Reset) .

L'appui sur cette touche permettra la réinitialisation complète du système.

- on introduit l'adresse d'arrivée;
- on appuie sur GO;

On verra alors affiché le déplacement et le type de branchement relatif (long ou court).

MV : Touche de transfert d'un programme d'une zone mémoire à une autre. L'exécution du transfert s'effectue comme suit :

- on introduit l'adresse de départ du programme;
- on appuie sur la touche MV;
- on introduit l'adresse de la fin du programme;
- on appuie sur la touche OF;
- on introduit l'adresse de destination;
- on appuie sur la touche GO.

FU : Touche de commande utilisateur.

L'appui sur cette touche puis sur une touche hexadécimale branchera vers une table contenant les commandes programmées par l'utilisateur.

RS : Touche de réinitialisation (Reset) .

L'appui sur cette touche permettra la réinitialisation complète du système.

CONCLUSION

La réalisation d'un système à microprocesseur demande une profonde connaissance des circuits à utiliser, ceci afin d'optimiser leur exploitation et parer à toutes contraintes pouvant survenir pendant leur utilisation.

Pour cette raison, nous^{avons} jugé utile d'étudier en détail les circuits essentiels de la réalisation, c'est à dire les deux circuits d'interface et le microprocesseur .

La réalisation du système conçu, se fait en utilisant la technique de "wrapping" qui permet des modifications faciles du système, le cas échéant.

vu la complexité des systèmes à microprocesseur, ils sont réalisés par cette technique avant la mise au point finale, c'est à dire sur circuit imprimé.

La réalisation de ce système nous a permis, d'approfondir nos connaissances dans le domaine de la micro-électronique dont l'utilisation est très répandue de nos jours et toucher de près aux problèmes pratiques rencontrés lors d'une réalisation.

Enfin, nous espérons avoir été assez clairs dans notre écrit et souhaitons que ce travail sera utile pour d'autres études dans le domaine.

ANNEXES

Mnémo.	Inhér.		Imméd.		Etendu		Direct		Ind.		Res. d'état								
	Op.	e #	Op.	e #	Op.	e #	Op.	e #	Op.	e #	E	F	H	I	N	Z	V	C	
Mise à 0 (0) --> (A ou B ou M)																			
CLR					7F	7 3	0F	6 2	6F	6+ 2+								0 1 0 0	
CLRA	4F	2 1																0 1 0 0	
CLRB	5F	2 1																0 1 0 0	
Comparaison et action sur (CC). Registres intacts. (A ou B) - (M) (D ou S ou U ou X ou Y) - (MM)																			
CMPA			B1	2 2	B1	5 3	91	4 2	A1	4+ 2+		?						X X X X	
CMPB			C1	2 2	F1	5 3	D1	4 2	E1	4+ 2+		?						X X X X	
CMPD			10B3	5 4	10B3	8 4	1093	7 3	10A3	7+ 3+								X X X X	
CMP5			118C	5 4	118C	8 4	119C	7 3	11AC	7+ 3+								X X X X	
CMPU			11B3	5 4	11B3	8 4	1193	7 3	11A3	7+ 3+								X X X X	
CMPX			8C	4 3	8C	7 3	9C	6 2	AC	6+ 2+								X X X X	
CMPY			108C	5 4	108C	8 4	109C	7 3	10AC	7+ 3+								X X X X	
Complémentation logique des registres B bits																			
COM					73	7 3	03	6 2	63	6+ 2+								X X 0 1	
COMA	43	2 1																X X 0 1	
COMB	53	2 1																X X 0 1	
Ajustement décimal de l'accumulateur A																			
DAA	19	2 1																X X X X	
Décrémenter (M ou A ou B) - 1 --> (M ou A ou B)																			
DEC					7A	7 3	0A	6 2	6A	6+ 2+								X X X .	
DECA	4A	2 1																X X X .	
DECB	5A	2 1																X X X .	
OU exclusif (A ou B) ⊕ (M) --> (A ou B)																			
EORA			8B	2 2	8B	5 3	9B	4 2	AB	4+ 2+								X X 0 .	
EORB			CB	2 2	FB	5 3	DB	4 2	EB	4+ 2+								X X 0 .	
Echange de deux registres																			
EXG R1,R2	1E..	B 2	R1 et R2 doivent être de même type, B ou 16 bits. L'instruction complète s'obtient en ajoutant le post-octet au code 1E. Le post-octet est formé de 2 demi-octets dont les valeurs possibles sont: D: 0 X: 1 Y: 2 U: 3 S: 4 PC: 5 A: 8 B: 9 CC: A DF: B Exemple: EXG A,B est traduit par 1E 89 ou 1E 9B EXG X,Y est traduit par 1E 12 ou 1E 21 Aucun bit du registre d'état n'est affecté à moins que le registre échangé soit CC lui-même																
Incrémenter (M ou A ou B) + 1 --> (M ou A ou B)																			
INC					7C	7 3	0C	6 2	6C	6+ 2+								X X X .	
INCA	4C	2 1																X X X .	
INCB	5C	2 1																X X X .	
Chargement des registres (M) --> (A ou B) ; (MM) --> (D ou S ou U ou X ou Y)																			
LDA			86	2 2	B6	5 3	96	4 2	A6	4+ 2+								X X 0 .	
LDB			C6	2 2	F6	5 3	D6	4 2	E6	4+ 2+								X X 0 .	
LDD			CC	3 3	FC	6 3	DC	5 2	EC	5+ 2+								X X 0 .	
LDS			10CE	4 4	10FE	7 4	10DE	6 3	10EE	6+ 3+								X X 0 .	
LDU			CE	3 3	FE	6 3	DE	5 2	EE	5+ 2+								X X 0 .	
LDX			8E	3 3	BE	6 3	9E	5 2	AE	5+ 2+								X X 0 .	
LDY			10BE	4 4	10BE	7 4	109E	6 3	10AE	6+ 3+								X X 0 .	
Instructions d'empilement ou de dépilement																			
PSHS	34..	5+ 2	7 6 5 4 3 2 1 0										No. du bit					
PSHU	36..	5+ 2											Res.	voir commentaire					
PULS	35..	5+ 2	PC U/S Y X DP B A CC										Res.	voir commentaire					
PULU	37..	5+ 2											Res.	voir commentaire					
Pour trouver le post-octet correspondant dans une instruction d'empilement ou de dépilement, inscrire un 1 dans la case correspondante, puis calculer la valeur globale de l'octet. Exemples: PSHS U,X,A,CC a pour code 34 53 PSHS A a pour code 34 02 Pour PULS et PULU, le registre d'état n'est pas affecté à moins que le dépilement invoque lui-même CC Pour connaître le nombre total de cycles requis, ajouter à la valeur de base 5, 1 cycle pour chaque registre 8 bits et 2 cycles pour chaque registre 16 bits																			
Décalage circulaire à gauche (C) <-- b7b6b5b4b3b2b1b0 <-- (C)																			
ROL					79	7 3	09	6 2	69	6+ 2+								X X X X	
ROLA	49	2 1																X X X X	
ROLB	59	2 1																X X X X	
Décalage circulaire à droite (C) --> b7b6b5b4b3b2b1b0 --> (C)																			
ROR					76	7 3	06	6 2	66	6+ 2+								X X . X	
RORA	46	2 1																X X . X	
RORB	56	2 1																X X . X	
Soustraction avec retenue (A ou B) - (M) - (C) --> (A ou B)																			
SBCA			B2	2 2	B2	5 3	92	4 2	A2	4+ 2+		?						X X X X	

Mnémono.	Inhér.	Imméd.	Étendu		Direct		Ind.		Res. d'état							
	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	E	F	H	I	N	Z	V	C
SBCB		C2 2 2	F2 5 3	D2 4 2	E2 4+ 2+	. . . ? . X X X X										
Extension de signe. FF --> (A) si b7 de (B) égal à 1, 0 --> (A) si b7=0																
SEX	1D 2 1				 X X . .										
Mise en mémoire de registres (A ou B) --> (M)																
(D ou S ou U ou X ou Y) --> (MM)																
STA			B7 5 3	97 4 2	A7 4+ 2+ X X 0 .										
STB			F7 5 3	D7 4 2	E7 4+ 2+ X X 0 .										
STD			FD 6 3	DD 5 2	ED 5+ 2+ X X 0 .										
STS			10FF 7 4	10DF 6 3	10EF 6+ 3+ X X 0 .										
STU			FF 6 3	DF 5 2	EF 5+ 2+ X X 0 .										
STX			BF 6 3	9F 5 2	AF 5+ 2+ X X 0 .										
STY			10BF 7 4	109F 6 3	10AF 6+ 3+ X X 0 .										
Soustraction (A ou B) - (M) --> (A ou B) ; (D) - (MM) --> (D)																
SUBA		B0 2 2	B0 5 3	90 4 2	A0 4+ 2+	. . . ? . X X X X										
SUBB		C0 2 2	F0 5 3	D0 4 2	E0 4+ 2+	. . . ? . X X X X										
SURD		B3 4 3	B3 7 3	93 6 2	A3 6+ 2+ X X X X										
Transfert de registres 8 et 16 bits																
TFR R1,R2	1F.. 7 2					voir commentaire										
Les registres R1 et R2 doivent être de même longueur.																
Le registre d'état n'est affecté que si R2 est CC lui-même																
Le calcul du post-octet est identique à celui exposé pour l'instruction EXG R1,R2																
Test et action sur (CC) (M ou A ou B) - (O)																
TST			7D 7 3	OD 6 2	6D 6+ 2+ X X 0 .										
TSTA	4D 2 1				 X X 0 .										
TSTB	5D 2 1				 X X 0 .										

INSTRUCTIONS IMPLIQUANT UNE RUPTURE DE SEQUENCE

Mnémono.	Inhér.	Imméd.	Étendu		Direct		Ind.		Res. d'état							
	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	Op. @ #	E	F	H	I	N	Z	V	C
Saut inconditionnel vers une adresse																
JMP			7E 4 3	OE 3 2	6E 3+ 2+										
Saut vers un sous-programme avec sauvegarde de l'adresse de retour																
JSR			BD 8 3	9D 7 2	AD 7+ 2+										
Retour de sous-programme d'interruption																
RTI	3B 6/15 1					X X X X X X X X										
Si E=0, le nombre de cycles est 6																
Si E=1, le nombre de cycles est 15																
Retour de sous-programme																
RTS	39 5 1														
Interruptions logicielles																
SWI	3F 19 1					1 1 . 1										
SWI2	103F 20 2					1										
SWI3	113F 20 2					1										
Synchronisation avec la ligne d'interruption																
SYNC	113 2 1														
ET logique de (CC) avec un opérande immédiat puis attente d'interruption																
CHAI	3C 20 2					X X X X X X X X										

INSTRUCTIONS DE BRANCHEMENTS RELATIFS COURTS ET LONGS. Le mode d'adressage correspondant est le mode relatif.

Mnémono.	Op. @ #	Mnémono.	Op. @ #	Conditions
BCC	24 3 2	LBCC	1024 5/6 4	C = 0
BCS	25 3 2	LBCS	1025 5/6 4	C = 1
BEQ	27 3 2	LBEQ	1027 5/6 4	Z = 1
BGE	2C 3 2	LBGE	102C 5/6 4	$N \oplus V = 0$
BGT	2E 3 2	LBGT	102E 5/6 4	$Z + (N \oplus V) = 0$
BHI	22 3 2	LBHI	1022 5/6 4	$Z + C = 0$
BHS	24 3 2	LBHS	1024 5/6 4	C = 0
BLE	2F 3 2	LBLLE	102F 5/6 4	$(N \oplus V) + Z = 1$
BLO	25 3 2	LBLO	1025 5/6 4	C = 1
BLS	23 3 2	LBLS	1023 5/6 4	C + Z = 1
BLT	2D 3 2	LBLT	102D 5/6 4	$N \oplus V = 1$
BMI	2B 3 2	LBMI	102B 5/6 4	N = 1
BNE	26 3 2	LBNE	1026 5/6 4	Z = 0
BPL	2A 3 2	LBPL	102A 5/6 4	N = 0
BRA	20 3 2	LBRA	16 5 3	Aucune
BRN	21 3 2	LBRN	1021 5 4	voir note
BSR	8D 7 2	LBSR	17 9 3	Aucune
BVC	28 3 2	LBVC	1028 5/6 4	V = 0
BVS	29 3 2	LBVS	1029 5/6 4	V = 1

Note: 5/6 signifie 6 cycles quand le branchement s'opère, 5 cycles quand la condition de branchement n'est pas réalisée.
BRN équivaut à 2 NOP. LBRN équivaut à 4 NOP

ADRESSAGE INDEXE. SIGNIFICATION DES BITS DU POST-OCTET.

Bit du registre post-octet								Mode d'adressage indexé					
7	6	5	4	3	2	1	0						
0	r	r	S	X	X	X	X	R + 4 bits déplacement				R: Registre de base	
1	r	r	0	0	0	0	0	,R+					
1	r	r	I	0	0	0	1	,R++					
1	r	r	0	0	0	1	0	,-R					
1	r	r	I	0	0	1	1	,--R					
1	r	r	I	0	1	0	0	R + 0 déplacement					
1	r	r	I	0	1	0	1	R + Acc.B déplacement					
1	r	r	I	0	1	1	0	R + Acc.A déplacement					
1	r	r	I	1	0	0	0	R + 7 bits déplacement					
1	r	r	I	1	0	0	1	R + 15 bits déplacement					
1	r	r	I	1	0	1	1	R + Acc.D déplacement					
1	X	X	I	1	1	0	0	PC + 7 bits déplacement					X: indifférent
1	X	X	I	1	1	0	1	PC + 15 bits déplacement					
1	X	X	1	1	1	1	1	Mode indirect type [n]					

Champs définissant le mode d'adressage

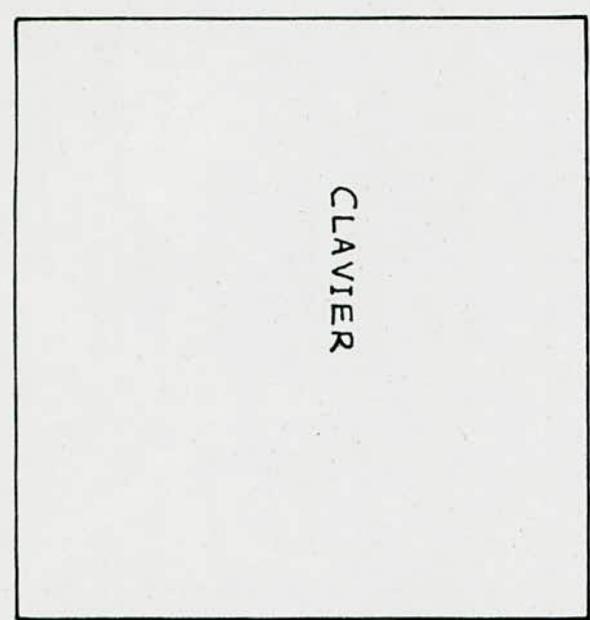
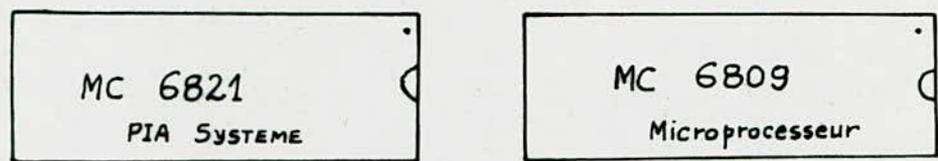
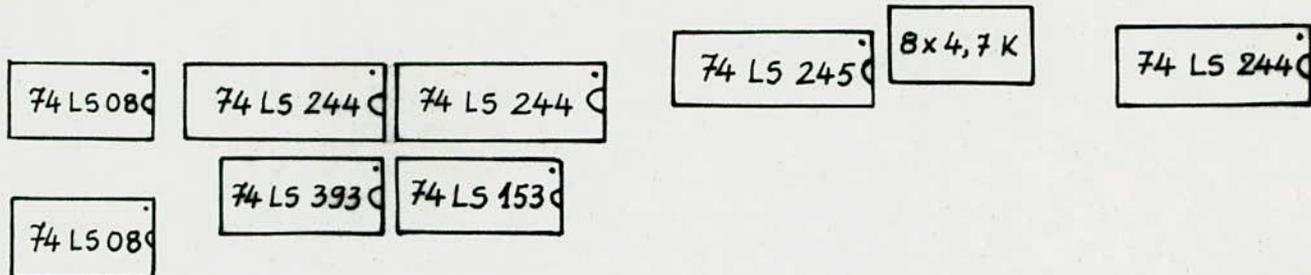
Bit d'indirection. Si I.=1, mode indirect
 Dans le 1er cas où b7=0, ce bit devient le bit de signe pour l'offset. Si S.=1, l'offset codé sur 4 bits se retranche de l'adresse de base

Les 2 bits b6b5 du post-octet définissent la nature du registre de base du mode indexé.
 Si b6b5 = 00 le registre est X
 Si b6b5 = 01 le registre est Y
 Si b6b5 = 10 le registre est U
 Si b6b5 = 11 le registre est S

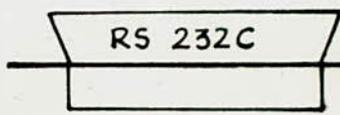
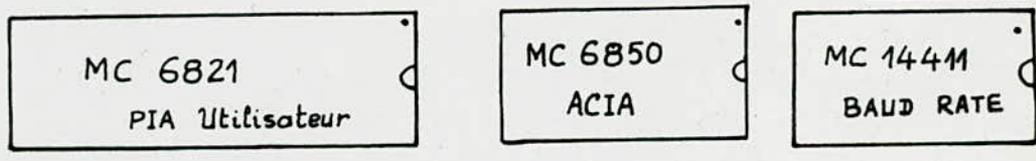
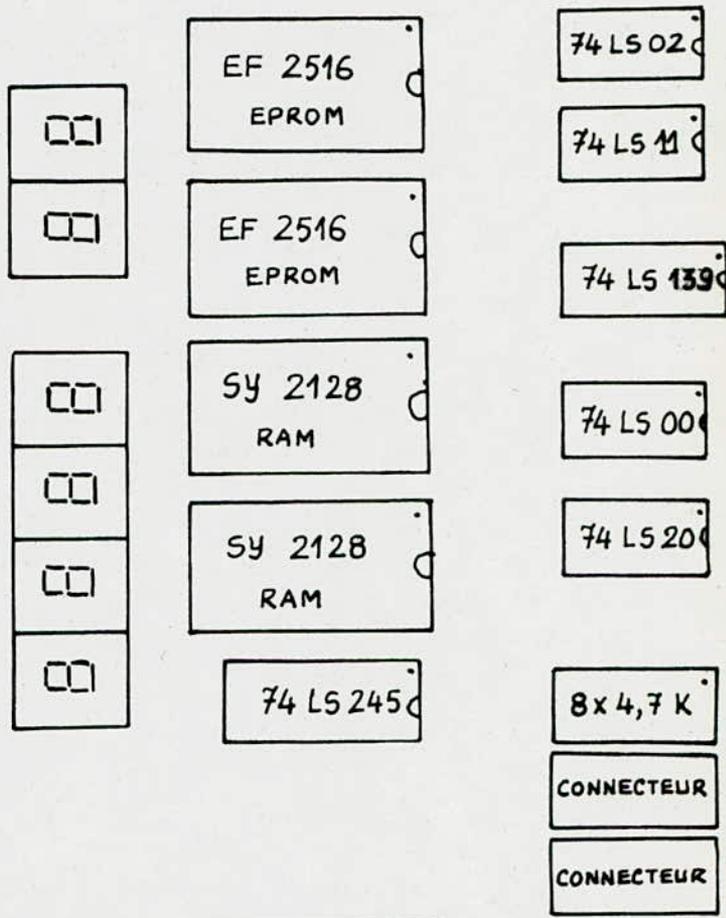
NOMBRE DE CYCLES ET D'OCTETS ADDITIONNELS POUR LES MODES INDEXES OU INDIRECTS

Formes	Non indirect			Indirect		
	Assembleur	Post-octet	@ #	Assembleur	Post-octet	@ #
sans déplacement	,R	1rr0 0100	0 0	[,R]	1rr1 0100	3 0
deplac. 4 bits	n,R	0rrS XXXX	1 0	par défaut, mode 7 bits		
deplac. 7 bits	n,R	1rr0 1000	1 1	[n,R]	1rr1 1000	4 1
deplac. 15 bits	n,R	1rr0 1001	4 2	[n,R]	1rr1 1001	7 2
deplac. Acc.A	A,R	1rr0 0110	1 0	[A,R]	1rr1 0110	4 0
deplac. Acc.B	B,R	1rr0 0101	1 0	[B,R]	1rr1 0101	4 0
deplac. Acc.D	D,R	1rr0 1011	4 0	[D,R]	1rr1 1011	7 0
Incram. par 1	,R+	1rr0 0000	2 0	impossible		
Incram. par 2	,R++	1rr0 0001	3 0	[,R++]	1rr1 0001	6 0
decrem. par 1	,-R	1rr0 0010	2 0	impossible		
decrem. par 2	,--R	1rr0 0011	3 0	[,--R]	1rr1 0011	6 0
depl. 7 bits PCR	n,PCR	1XX0 1100	1 1	[n,PCR]	1XX1 1100	4 1
depl. 15 bits PCR	n,PCR	1XX0 1101	5 2	[n,PCR]	1XX1 1101	8 2
Indirect étendu	—	—	—	[n]	1001 1111	5 2

n: nombre ou étiquette assembleur
 R: Registre de base des modes indexés S ou U ou X ou Y



Boutons Poussoires.



BIBLIOGRAPHIE

-M. AUMIAUX

Les systèmes à microprocesseurs

"MASSON" 1982

-BUI MINH DUC

Programation en assembleur 6809

"EYROLLES" 1986

-CLAUDE DARDANNE

Le microprocesseur 6809 ses peripheriques et le processeur
graphique 936566

"EYROLLES" 1984

-G. REVELLIN

Microprocesseurs du 6800 au 6809

Modes d'interfaçage

"DUNOD" 1981

-STEVE LEIBSON

Manual des interfaces

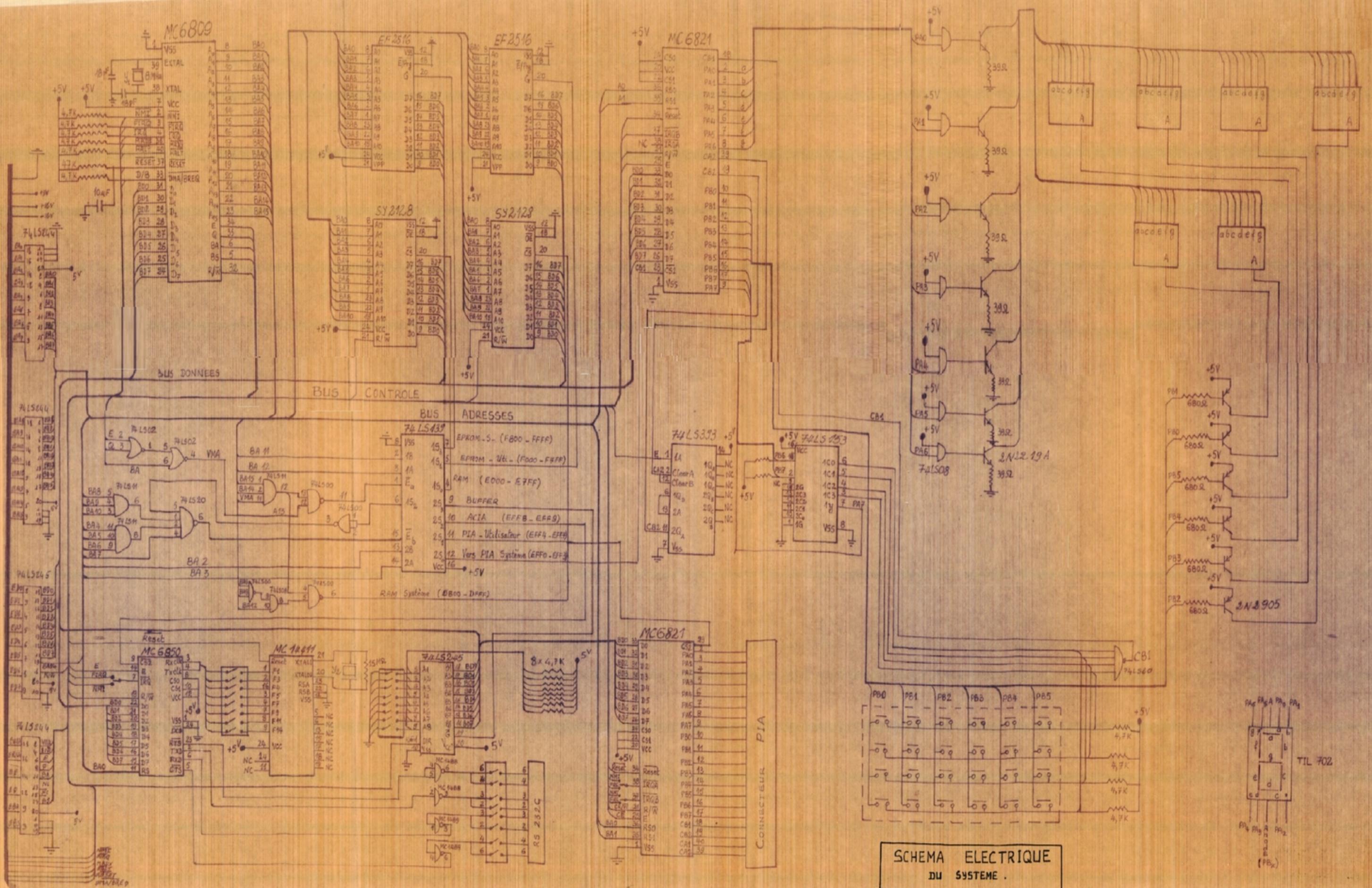
"Mc GRAW HILL" 1984

- MEK 6802 D5E

Microcomputer evaluation board user's manual

"MOTOROLA INC"

.....



SCHEMA ELECTRIQUE
DU SYSTEME.

