

Université des Sciences et Technologies d'Alger

14/80

Aex

DEPARTEMENT ELECTRONIQUE ET ELECTROTECHNIQUE

المدرسة الوطنية للعلوم الهندسية
المكتبة
PROJET DE FIN D'ETUDES

ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

SYSTEME MICROINFORMATIQUE " TEMPS REEL "

المدرسة الوطنية للعلوم الهندسية

المكتبة

ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

Proposé par :

H. TEDJINI

Docteur Ingénieur

Etudié par :

M. BOUSETTA

M. LAMARA

JANVIER 1980

Université des Sciences et Technologies d'Alger

DEPARTEMENT ELECTRONIQUE ET ELECTROTECHNIQUE



SYSTEME MICROINFORMATIQUE " TEMPS REEL "

Proposé par :

H. TEDJINI

Docteur Ingénieur

Etudié par :

M. BOUSETTA

M. LAMARA

JANVIER 1980

REMERCIEMENTS

En ces moments mémorables, où notre travail prend sa dernière forme, nous ne pouvons rester sans une pensée à tous ceux qui nous ont permis d'abord de commencer et ensuite de bien terminer.

Nos remerciements vont, tout d'abord, à MM. B. SANSAL et H. TEJINI; responsables de la division V au CSTN, pour l'accès qu'ils nous ont permis et l'accueil qu'ils nous y ont réservés.

Nos remerciements vont également à Mr. H. TEJINI, ainsi qu'à tous ses collaborateurs MM. HALIMI, A. BOURKEB, R. OUIGUINI, GHRIB et SMARA et BOUMGHAR, pour leur fructueuse collaboration.

Notre gratitude va enfin à Mr ABDEDDAIM, directeur de l'école supérieure de chimie d'Alger, pour sa formidable compréhension.

INTRODUCTION

Depuis l'avènement des circuits intégrés, l'électronique en général et l'informatique en particulier ont accompli d'énormes progrès. En effet, ces circuits réduisent au minimum l'encombrement des systèmes et apportent de ce fait un surplus de fiabilité.

L'informatique, il en est de même de la micro-informatique, s'est alors accaparé de nouveaux domaines de travail aussi divers que l'industrie, l'économie, le scientifique...

Particulièrement, l'informatique dite "temps réel" y trouve une place de choix. Son prestige prend sa source dans le fait qu'elle répond parfaitement aux exigences des utilisateurs.

Au sein de la division V (simulation et contrôle) du CSTN, divers projets utilitaires sont confiés aux ingénieurs stagiaires et réalisés périodiquement. Un de ceux-là est traité dans cet ouvrage.

Il développe un système "temps réel" (système micro-informatique) à travers deux de ses éléments essentiels: l'horloge et le moniteur temps réel.

SOMMAIRE

<u>Chapitre I : Système temps réel</u>	4
1-Notion de système temps réel	4
2-Emploi d'un système temps réel	4
3-Structure d'un système temps réel	5
<u>Chapitre 2 : Moniteur temps réel</u>	8
1-Les interruptions	8
2-Généralités et définitions	13
3-Organigrammes et programmes	18
<u>Chapitre 3 : Le MC 6840</u>	33
1-Etude des lignes d'entrée/sortie	33
2-Etude des registres internes	35
3-Modes de fonctionnement	40
4-Initialisation	44
<u>Chapitre 4 : Horloge temps réel</u>	47
1-Description de la carte	48
2-Applications	53
<u>Chapitre 5 : Applications du système temps réel</u>	
1-Application industrielle	74
2-Application de gestion	76

CHAPITRE I: SYSTEME TEMPS REEL.

1°)-Notion de système temps réel

Un système informatique temps réel est un ensemble d'éléments informatiques regroupés autour d'une unité centrale.

Il gère une situation donnée en recevant des informations, en les traitant et en retournant les résultats sous forme d'informations de commande ou de contrôle, suffisamment vite pour agir à temps sur l'évolution du phénomène.

DE CETTE définition il ressort que:

a- l'ordinateur devient une partie intégrante du processus ou de la situation gérée puisqu'il traite immédiatement des données issues du processus à des temps aléatoires (c-à-d non définis à l'avance), : on dira que l'ordinateur est en ligne.

b- Le traitement des données risque d'introduire un retard dans l'élaboration des informations de commande. Si la situation évolue vite, les données prélevées à l'instant T_0 vont déclencher, après traitement, des signaux de commande à l'instant T_1 . Mais ceux-ci vont agir sur une situation qui a du évoluer de nouveau. Il est donc impératif de réduire le temps de traitement des données de sorte que la commande de l'ordinateur se fasse suffisamment à temps.

En conclusion, si la donnée à traiter n'est pas exploitée dans un temps minimum, dont la limite supérieure est fixée par la vitesse d'évolution du système, alors elle se dégrade et est irrémédiablement perdue. L'ordinateur temps réel remédie à cet inconvénient.

Aussi le système temps réel ne "voit" aucune limite dans les domaines d'applications, pourvu que les conditions sus-citées soient respectées. Notons, enfin, que le temps de réponse imposé au système temps réel varie selon la nature du phénomène surveillé: gestion de stocks d'un magasin, relevé d'un compte bancaire, surveillance d'un processus chimique ou nucléaire, guidage d'une fusée...

Remarque

Un système micro-informatique temps réel est défini de la même manière que précédemment, sauf que l'unité centrale est un micro-processeur.

2°)-Emploi d'un système en temps réel

Dans l'utilisation classique (en temps différé) de l'ordinateur les données lui sont fournies dans un ordre fixe et le résultat de leur traitement est indépendant du temps dans lequel ces

données ont été introduites.

L'information et son traitement ont donc des valeurs indépendantes du temps.

Mais il en est tout autrement dans l'utilisation de l'ordinateur en temps réel. Dans cette optique, les conditions sont différentes et ceci parceque:

a-les données à traiter sont caractérisées par:

1°)-un ordre d'arrivée aléatoire

2°)-une dégradation de leur valeur si elles ne sont pas exploitées en temps voulu.

b-l'évolution du phénomène peut exiger la multiplicité des programmes de traitement.

L'utilisation d'un système temps réel nécessite donc:

-une structure technologiques adaptée

-une fiabilité extrême du matériel vue la dépendance entre la valeur de l'information traitée et l'instant où elle est produite.

-un logiciel adapté, le moins encombrant possible : le moniteur temps réel en constitue l'essentiel

3°)-Structure d'un système temps réel

La structure minimale d'un système temps réel comporte:

a-une horloge temps réel: elle assure et contrôle, en association avec le moniteur temps réel, et par temporisation le déroulement des travaux; elle relance automatiquement certaines tâches et élabore des mesures en temps réel.

Cette horloge permet également les opérations suivantes:

-l'activation périodique

-l'activation sur délai

-l'attente sur délai

Cependant, la gestion d'une horloge peut alourdir le temps d'exécution général du système. En effet, l'élaboration des données temporelles se fait à partir des interruptions délivrées par une horloge matérielle, et il faut prendre en compte et gérer toutes ces interruptions jusqu'à des cadences très élevées si l'on veut traiter des quantums de temps très petits. Il n'est donc pas judicieux de fixer de façon inamovible les unités de temps délivrées par une horloge matérielle. Par contre, une manière bien souple de procéder est l'utilisation d'horloges logicielles: à partir d'une horloge matérielle, le moniteur élabore par comptage plusieurs horloges logicielles, ou unités de travail à partir desquelles

l'utilisateur exprime ses données temporelles. Les seuils de ces horloges logicielles seront fixés par l'utilisateur à la génération du système. L'avantage de cette méthode est l'adaptation de l'horloge aux besoins de l'application et une optimisation du temps de traitement correspondant.

b- Les interruptions

Le système d'interruption permet à l'unité centrale de fonctionner avec un rendement acceptable, quand elle est reliée à un grand nombre d'éléments périphériques. Une interruption (du programme en cours) déclenche la sauvegarde des contenus des registres de travail (ACC.A, ACC.B, X, P.C., C.C.R.) dans un emplacement mémoire (pile) repéré par un registre (Pointeur de Pile: S.P.). L'interruption permet au terminal, qui la provoque, de suspendre le déroulement du programme en cours en vue d'exécuter le programme associé ou de dialoguer avec l'unité centrale.

c- Les priorités

Les systèmes de priorités sont essentiellement de deux types:

-le système de priorités hardware (matériel) défini par connexion des éléments périphériques avec l'unité centrale. Cette méthode exige la multiplication des circuits codeurs de priorités (type 8214A de INTEL ou MC 6828 de MOTOROLA) ainsi que celle des vecteurs d'interruption implantés en mémoires.

-le système de priorité software (logiciel) permettant de gérer le déroulement des travaux stockés préalablement en mémoires.

Dans les deux cas, la tâche de priorité supérieure, et seulement dans ce cas, suspend toujours celle de moindre priorité.

d- Les périphériques

Les périphériques doivent être adaptés au système temps réel. Ils doivent travailler avec une grande vitesse et de ce fait les mémoires à temps d'accès élevé sont généralement éliminées (mémoires à tores ou à tambour).

e- structure d'un logiciel adapté

Un système temps réel exécute, dans le temps, un certain nombre de travaux. Pour les exécuter, il faut écrire des programmes correspondant à des tâches puis les loger en mémoires. L'ordinateur travaillant en série (c-à-d n'exécute qu'un programme à la fois), il se pose alors un problème de déroulement de ces programmes en fonction de leur priorité, des sollicitations des périphériques, des sollicitations de l'opérateur ou des tâches elles-mêmes.

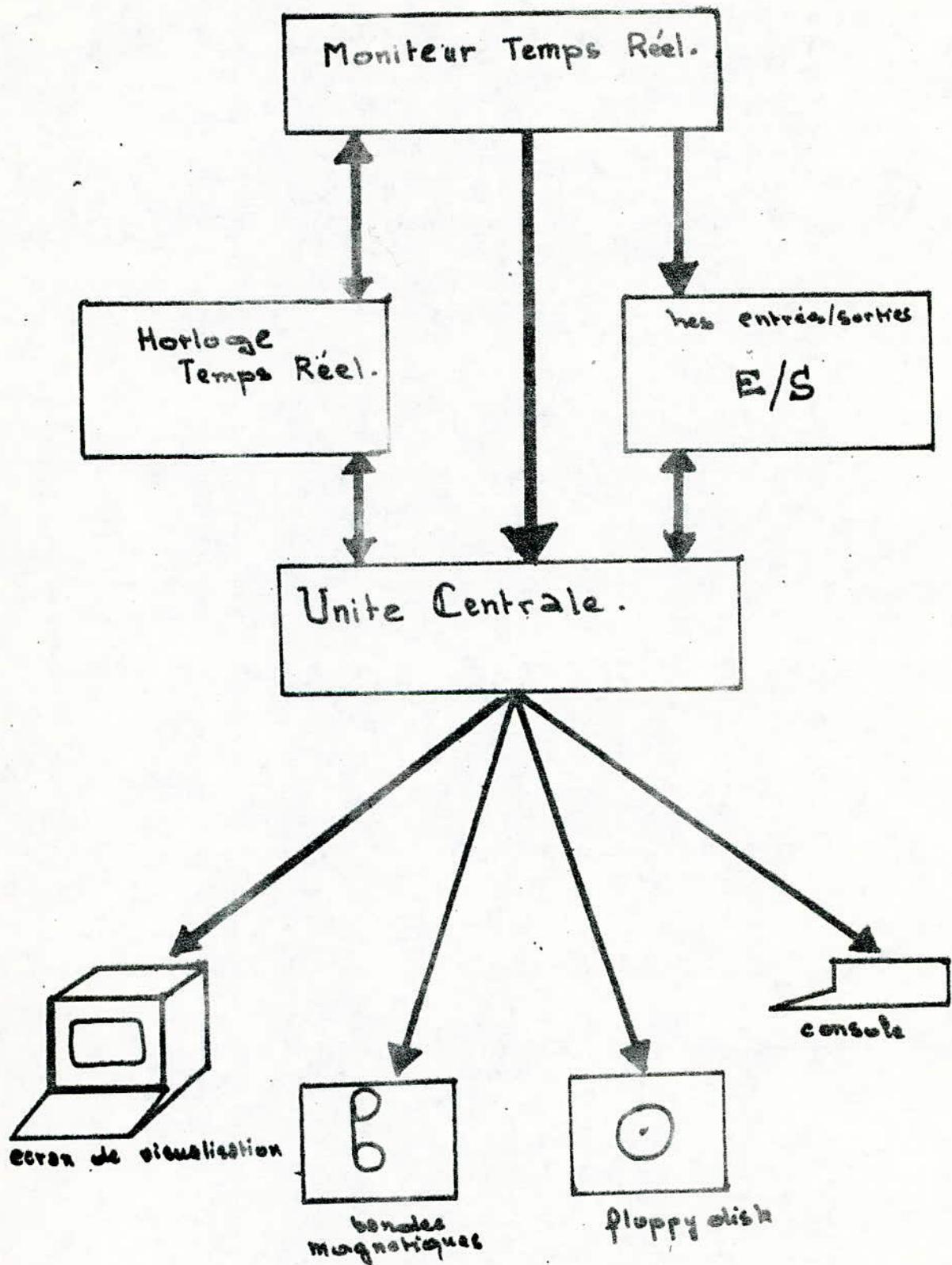
L'outil logiciel qui va gérer l'ordonnancement des tâches, tout en

optimisant l'occupation du C.P.U., est le moniteur temps réel. Pour mener à bien sa mission, le moniteur dispose d'une série d'informations sur chaque tâche: la priorité, le mot d'état, le numéro de la tâche et son adresse en mémoire en constituent l'essentiel. Nous reviendrons plus en détail sur les aspects techniques dans le chapitre 2.

Conclusion:

Le système temps réel, nouveau concept de l'informatique, constitue une évolution du logiciel par rapport au matériel. Cette évolution est surtout sensible dans les micro et mini-ordinateurs dont l'efficacité tient aux soins apportés dans l'élaboration du moniteur et de l'horloge temps réel.

Une structure-type d'un micro-système temps réel.



1 : Les INTERRUPTIONS.

Les interruptions sont des événements qui provoquent l'arrêt d'un programme en cours de traitement pour permettre de traiter un autre programme. Elles sont de différents types :

- Les interruptions d'entrée/sortie.

Les interruptions permettent le transfert de données entre le M P U et les périphériques (téléimprimeur, écran de visualisation, clavier, capteur, etc...) et cela dans les deux sens Exemple: $\overline{\text{NMI}}$ et $\overline{\text{IRQ}}$.

- Les interruptions programmées.

Certaines instructions du MC.6800 génèrent des interruptions de programme, comme par exemple, l'instruction SWI (software interrupt).

- Les interruptions internes.

Ce sont des interruptions dont l'utilisateur du MC 6800 n'a pas accès. Elles sont déclenchées lors d'une panne d'alimentation, d'erreur de parité en mémoire, mauvaise disposition d'une carte fonctionnelle, etc... Exemple: $\overline{\text{HALT}}$.

1.1. NATION DE " PILE "

Il se peut qu'au moment de l'apparition de l'interruption, les registres internes du microprocesseur contiennent des résultats partiels d'un programme. Afin de ne pas perdre ces informations contenues dans l'ensemble des registres du MC.6800, une opération de sauvegarde de ces informations dans une zone mémoire spéciale appelée " pile " est déclenchée automatiquement par l'apparition d'une interruption.

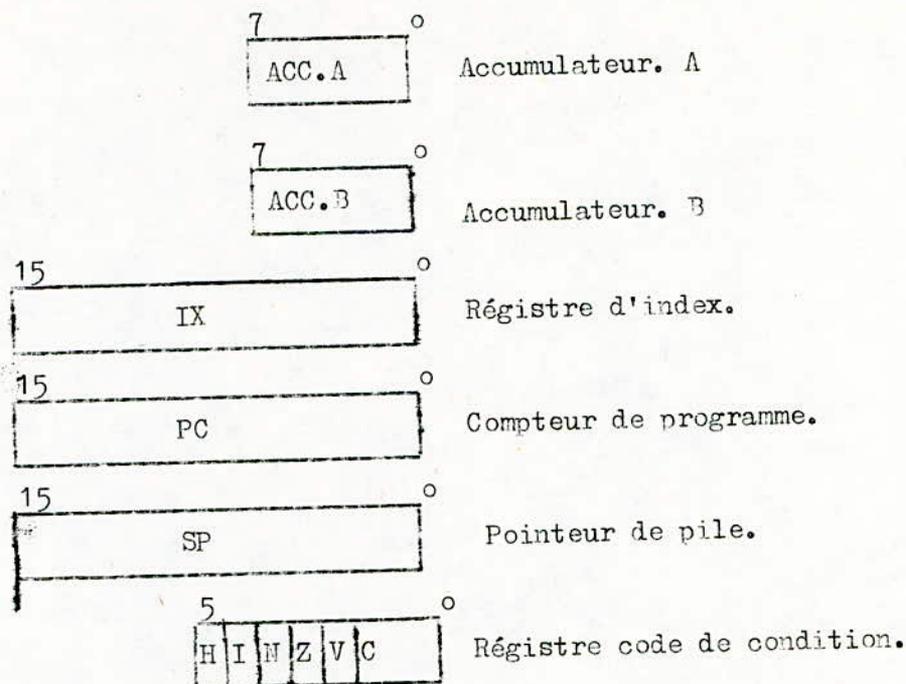


Fig 2.1 : Régistres internes du MC.6800.

La pile associée au microprocesseur MC 6800 est du type software du fait qu'elle est réalisée en réservant une zone dans la mémoire vive du système (RAM). La position de cette pile dans la mémoire vive est donc choisie par le programmeur. Pour définir l'adresse de cette pile, il existe à l'intérieur du microprocesseur un compteur spécial (régistre de 16 bits) appelé pointeur de pile.

Remarque 1 : Le pointeur de pile indique toujours la position mémoire vide de la pile, juste au dessus de la position mémoire du dernier mot "empilé."

Remarque 2 : Lors d'une opération de chargement (ou empilement), le contenu de l'accumulateur est d'abord transféré dans la pile à l'adresse indiquée par le pointeur de pile. Ensuite le pointeur de pile est décrementé (diminue d'une unité).

Remarque 3 : Lors d'une opération d'extraction, le pointeur de pile est d'abord incrementé (augmente d'une unité) puis le contenu de la position mémoire de la pile ~~repointé par le~~ pointeur pile, est transféré dans l'accumulateur.

1.2 Les interruptions d'entrée/sortie et leurs séquences.

Ce sont des interruptions qui nous intéressent dans la mesure où elles jouent un très grand rôle dans le moniteur temps réel (de même dans le software de l'horloge temps réel).

1.2.1 L'interruption non masquable: NMI.

Une interruption masquable est une interruption qui n'a lieu que si une

bascule interne au microprocesseur, appelée masque d'interruption, est à l'état zero . L'état de ce masque peut être programmé par instruction. Dans le cas de l'interruption $\overline{\text{NMI}}$, l'état du masque est ignoré : c'est en quelque sorte une interruption prioritaire. $\overline{\text{NMI}}$ est généralement réservée aux périphériques de type télécype. Elle est reconnue au front descendant du signal issu de la ligne $\overline{\text{NMI}}$ (entrée du MC. 6800).

Après avoir achevé l'instruction en cours , le MPU sauvegarde les registres internes, met à 1 le masque d'interruption, charge le compteur de programme avec l'information contenue dans les positions mémoires d'adresses FFFF et FFFF (2 OCTETS) Puis se branche au programme dont l'adresse est contenue dans le compteur d'adresse. (Voir figure 2.2.A.)

1.2. Interruption $\overline{\text{IRQ}}$ masquable ($\overline{\text{IRQ}}$: interrupt request)

Contrairement à l'interruption $\overline{\text{NMI}}$, l'interruption $\overline{\text{IRQ}}$ ne peut être prise en compte que si le masque d'interruption est à zero. Auquel cas sequence, est identique à celle de $\overline{\text{NMI}}$. si le masque est à "un", alors le programme initial continue. (Voir figure 2.2.B.).

CONCLUSION : Les deux interruptions que nous venons d'étudier, provoquent un branchement directement à l'adresse du programme de traitement de celles-ci. Les interruptions sont dites vectorisées. La vectorisation consiste donc à associer à une interruption donnée une adresse fixe qu'est l'adresse du programme à exécuter. Cette adresse fixe est implantée en mémoire morte par le constructeur afin qu'elle ne s'efface pas . (Voir figure 2.3.).

- $\overline{\text{RESET}}$ est l'interruption d'initialisation (pour le démarrage).
- SWI est une interruption programmée qui arrête le programme afin d'effectuer des opérations spéciales dites " de mise au point (Debugging) d'un programme .

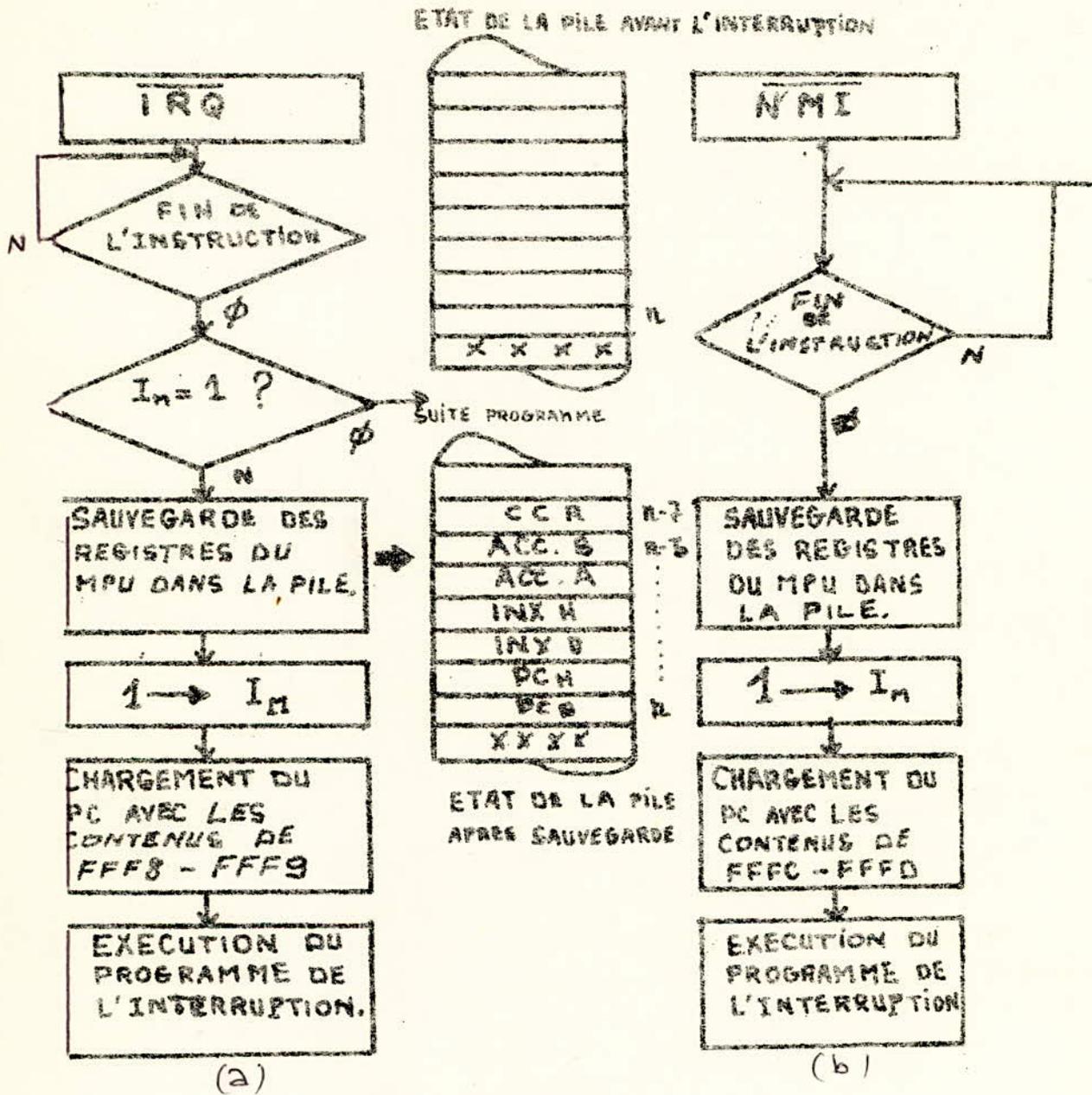


FIG 2.2. SEQUENCE DE TRAITEMENT DES INTERRUPTIONS IRQ ET NMI

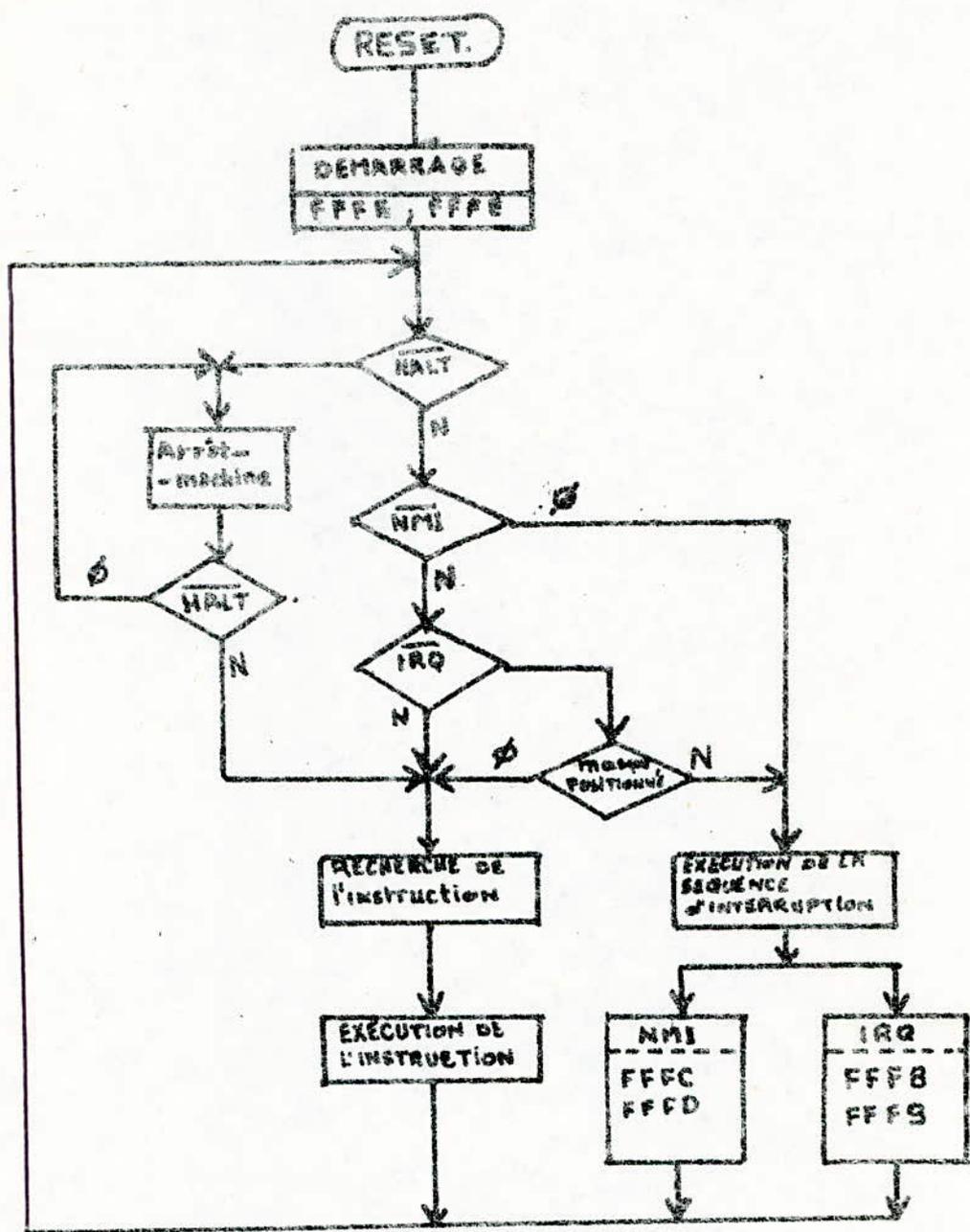


FIGURE 2.4 :

Organigramme des principaux chemins de décision et les vecteurs d'interruption du microprocesseur MC. 6800

(CHAPITRE.2)

MONITEUR TEMPS REEL.

Plan d'étude:

- 2.1) Généralités et définitions.
- 2.2) Tables de vérités
- 2.3) Algorithmes.
- 2.4) Registres et tables de travail du moniteur.
- 2.5) Organigrammes et programmes.

-----000-----

CHAPITRE II LE MONITEUR "TEMPS REEL"

2°)-Généralités et définitions

1-) Dans tout micro-ordinateur, on conserve en permanence des tâches susceptibles d'être activées à tout moment.

Etant donnée que la machine ne peut exécuter qu'un seul programme à la fois, son choix se posera dès que plusieurs demandes se font simultanément. L'établissement de priorités et leur assignation aux différentes tâches, en fonction de l'urgence des travaux à faire, sera une solution. Le moniteur, ensemble de programmes logés en mémoires, aura pour fonctions

de:

- rechercher une tâche activée, susceptible d'être exécutée,
- comparer les priorités des différentes tâches à divers niveaux d'exécution (tâche la plus prioritaire prête, tâche en exécution, et éventuellement tâche la plus récemment suspendue),
- suspendre ou redémarrer l'exécution d'une tâche.

Ceci nous amène à parler des quatre états possibles d'une tâche:

a)-état inactif ou mort:

Ce type de tâche ne demande pas l'exécution. C'est une tâche pratiquement inexistante.

b)-état prêt:

La tâche sort de l'état inactif, et demande à être exécutée. Son passage effectif sous contrôle du CPU (Unité Centrale) dépendra de son niveau de priorité.

c)-état exécution:

La tâche est effectivement sous contrôle du CPU.

d)-état suspendu:

Il reflète l'état d'une tâche dont l'exécution est interrompue au profit d'une tâche plus prioritaire.

En conclusion, on remarque, qu'à un instant donné, il ne peut y avoir plusieurs tâches dans divers états, mais UNE SEULE sera en exécution.

Si toutes les tâches sont dans l'état inactif, le CPU reste en attente (état WAIT) jusqu'à ce qu'une interruption transforme une tâche inactive en tâche prête.

L'exécution d'une tâche par le CPU ne peut être arrêtée que par interruption. Ces interruptions sont prises en compte par le CPU seulement pendant la phase "exécution" d'une tâche ou dans l'état "WAIT". Elles sont donc ignorées (c-à-d masquées) pendant le déroulement des programmes du moniteur.

2-)Table de vérité

Notations

Je : tache à l'état exécution

Jr : " " prêt

Js : " " suspendu

Pe : priorité de Je

Pr : " Jr

Ps : " Js

Remarques:

Une tache en exécution est prioritaire sur toutes les taches pretes et suspendues; ces dernières étant stockées dans une pile type LIFO (dernière tache empilée étant la première sortie) dans l'ordre de priorité décroissant à partir du haut de la pile.

La reprise en exécution d'une tache suspendue commence toujours au point d'interruption.

Une tache prete, plus prioritaire que celle en exécution, provoque toujours la suspension de celle-ci, son rangement dans la pile LIFO, puis le début du cycle d'exécution de la tache prete.

La table de vérité est donc:

Jr	Je	Js	Pe \geq Pr	Ps \geq Pr	Commentaires
0	0	0	-	-	fin du Moniteur
0	0	1	-	-	poursuite d'exécution
0	1	-	-	-	impossible
1	0	0	-	-	début d'exécution
1	0	1	-	0	début d'exécution
1	0	1	-	1	poursuite d'exécution
1	1	-	0	-	suspension de la tache en exécution exécution de la tache prete
1	1	-	1	-	fin du moniteur

Commentaires

1° Pas : Jr = 0 , Je = 0 , Js = 0.

alors fin du moniteur et attente (état WAIT) d'une interruption.

2° Pas : Jr = 0 , Je = 0 , Js = 1.

apres la fin de programme de la tache en exécution, le CPU reprend l'exécution de la tache précédemment suspendue.

3° Pas : Jr = 1 : recherche des taches en exécution et suspendue

4°Pas : $J_e=0, J_s=0.$

la tache prete passe en execution.

5°Pas : $J_e=1, J_s=0.$

comparaison des priorités PE et Pr des taches en execution et prete

a-) $Pr > Pe$

arret de la tache en execution sauvegardée dans la pile LIFO, puis début d'execution de la tache prete.

b-) $Pr < Pe$

la tache prete n'est pas prise en consideration, et poursuite en execution de J_e .

6°Pas : $J_e=0, J_s=1.$

comparaison des priorités Pr et Ps des taches prete et suspendue.

a-) $Pr > Ps$

la tache suspendue garde son état, et execution de la tache prete.

b-) $Pr < Ps$

la tache prete garde son état, et reprise en execution de la tache suspendue.

7°Pas : $J_e=1, J_s=1.$

dans ce cas $Pe > Ps$. Aller en 5a ou en 5b

8°Pas : fin du moniteur et début d'execution de la tache sélectionnée.

3-) Algorithme du moniteur

De la table de vérité découlent les étapes successives qui mènent à l'execution de la tache la plus prioritaire du moment.

4-) Registres et tables de travail du moniteur

Les paramètres essentiels utilisés par le moniteur sont les états et les priorités affectés aux taches existantes. Ces renseignements (numéro de tache, état et priorité correspondants) sont notés dans une table dite des taches possibles (TSKTBL). Le numéro de la tache est noté sur un octet, donc on recense au maximum $2^8=256$ taches possibles.

Les quatre états possibles pour une tache sont fixés par 2 bits de l'octet immédiatement au-dessous du numéro de tache. Il nous reste alors six (6) bits pour coder la priorité de la tache, soit :

$2^6=64$ priorités possibles.

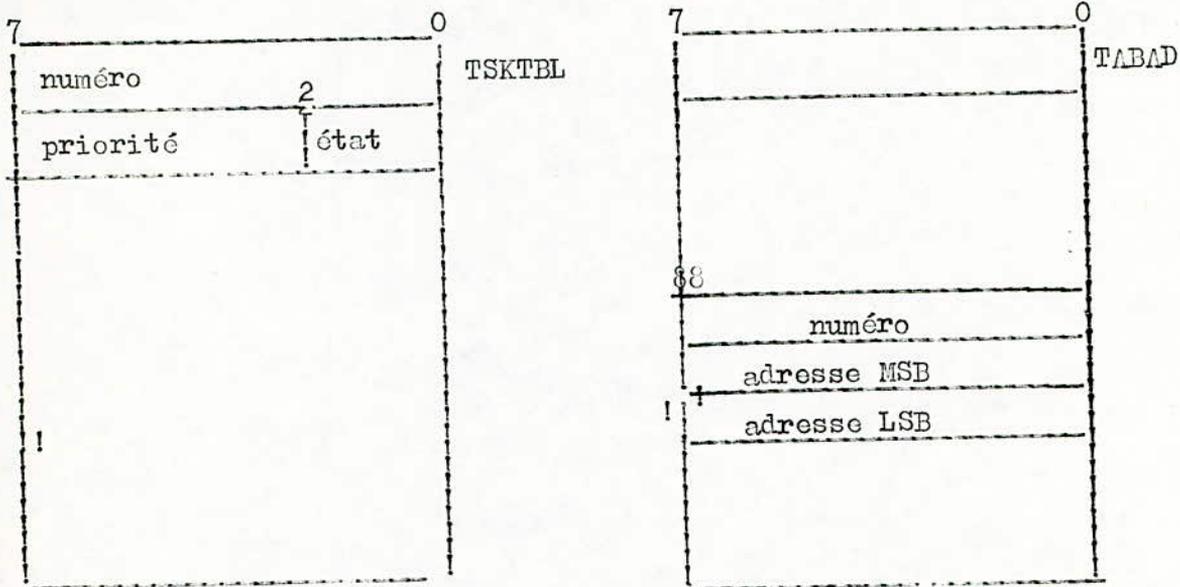
On déduit donc qu'il peut exister plusieurs taches de même niveau de priorité.

Sachant qu'il peut exister, à un instant donné, plusieurs taches à l'état pret, seule celle disposant du niveau de priorité supérieure sera prise en consideration.

La table TSKTBL sera remplie par le programmeur dans l'ordre des prio-

rités croissantes; et l'inspection de la table par le moniteur se fait à partir de la plus haute priorité.

Dans une seconde table dite des adresses (TABAD) on loge les adresses de début de programmes de ces taches.



Une 3^o table, arrangée en pile LIFO, sert au stockage des informations relatives aux taches suspendues (numéro de tache, état et priorité correspondants) La tache la plus récemment suspendue sera donc la plus prioritaire par rapport à toutes celles déjà existantes dans la pile. Cette pile peut contenir jusqu'à 255 taches suspendues.

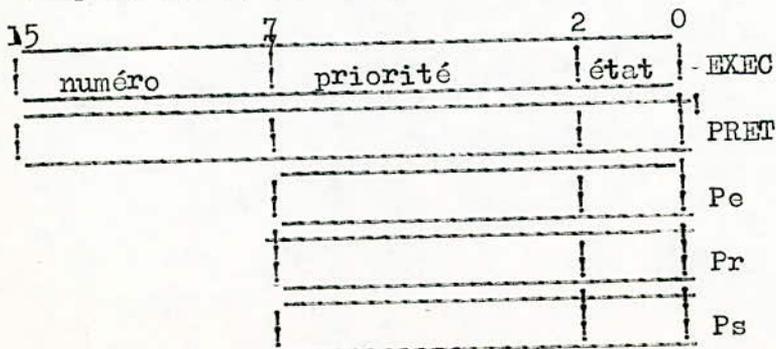
La prochaine position libre de la pile est toujours repérée par un registre de 16 bits dit SPTR.

Les autres registres du moniteur sont:

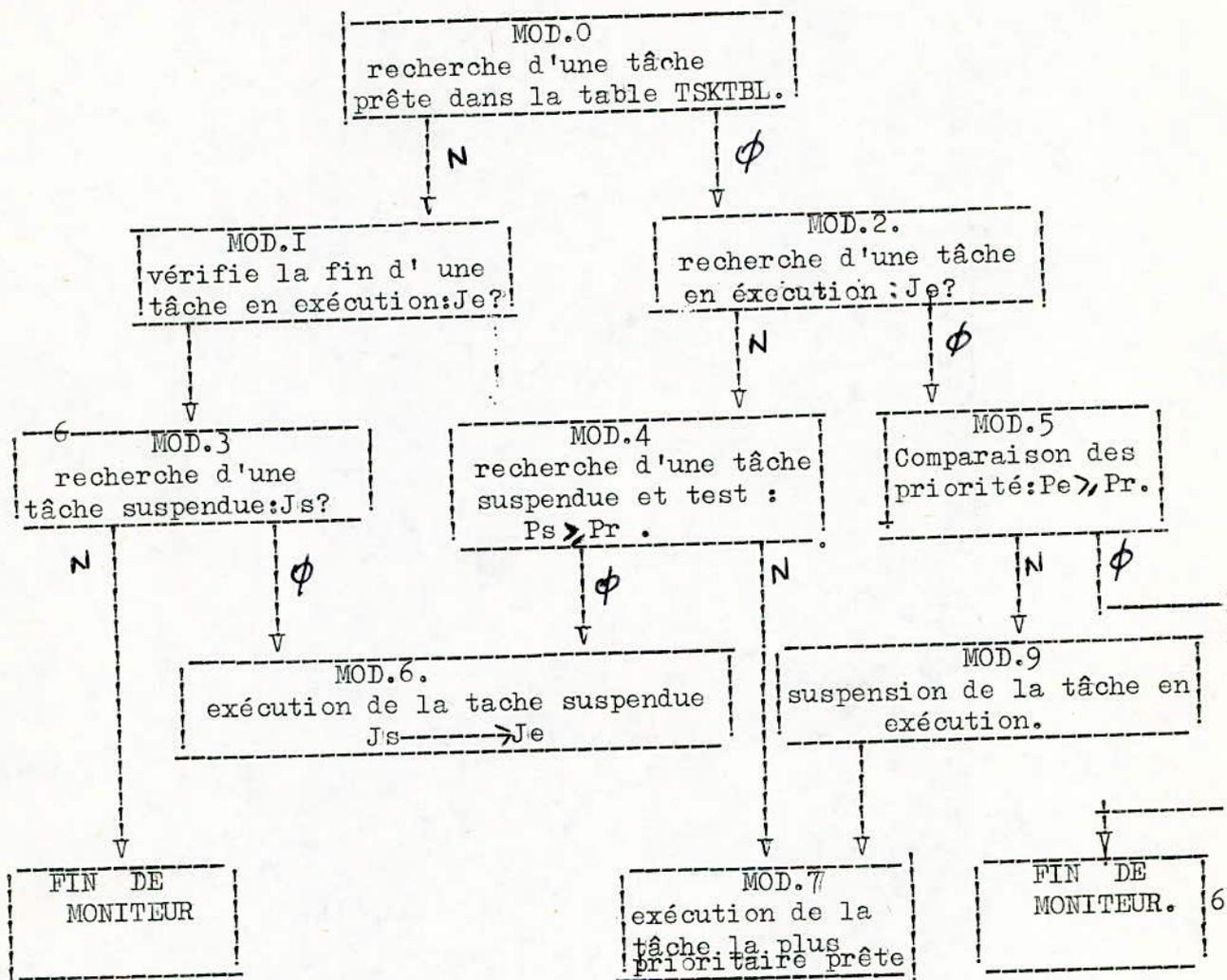
- un registre de 16 bits pour la tache la plus prioritaire prete (PRET)
- un " " " " en exécution (EXEC)

Le MSB de ces registres contient le numéro de la tache, leur LSB renferme la priorité et le mot d'état.

- trois registres de 8 bits (Pe, Pr, Ps) servant à stocker les priorités et les mots d'état des taches en exécution, la plus prioritaire prete et la plus récemment suspendue respectivement.



ALGORITHME DU MONITEUR TEMPS REEL.



Module MOD.0 :

Il recherche, dans la table des taches TSKTBL, la tache la plus prioritaire prete. L'entrée, dans ce module, se fait par interruption ou lors de la mise sous tension du micro-ordinateur.

Module MOD.1 :

Activé par une fin de tache et en l'absence d'interruption, ce module vérifie l'inexistence d'une tache en exécution, ou constate sa fin en remettant à zéro tous les registres de la tache en exécution et en mettant à jours l'état de la tache dans la table des taches TSKTBL.

Module MOD.2 :

Après activation d'une tache morte, ce module cherche une tache en exécution.

Module MOD.3 :

Après inexistence des taches prete et en execution, ce module cherche une tache suspendue.

Module MOD.4 :

Après activation d'une tache morte et inexistence de tache en execution, ce module cherche une tache suspendue et compare les priorités des taches prete et suspendue (s'il y en a une).

Module MOD.5 :

Il compare les priorités de la tache en execution et de la tache prete.

Module MOD.6 :

Il passe en execution la tache la plus recemment suspendue, et remet à jours le mot d'état de cette tache dans la table TSKTBL.

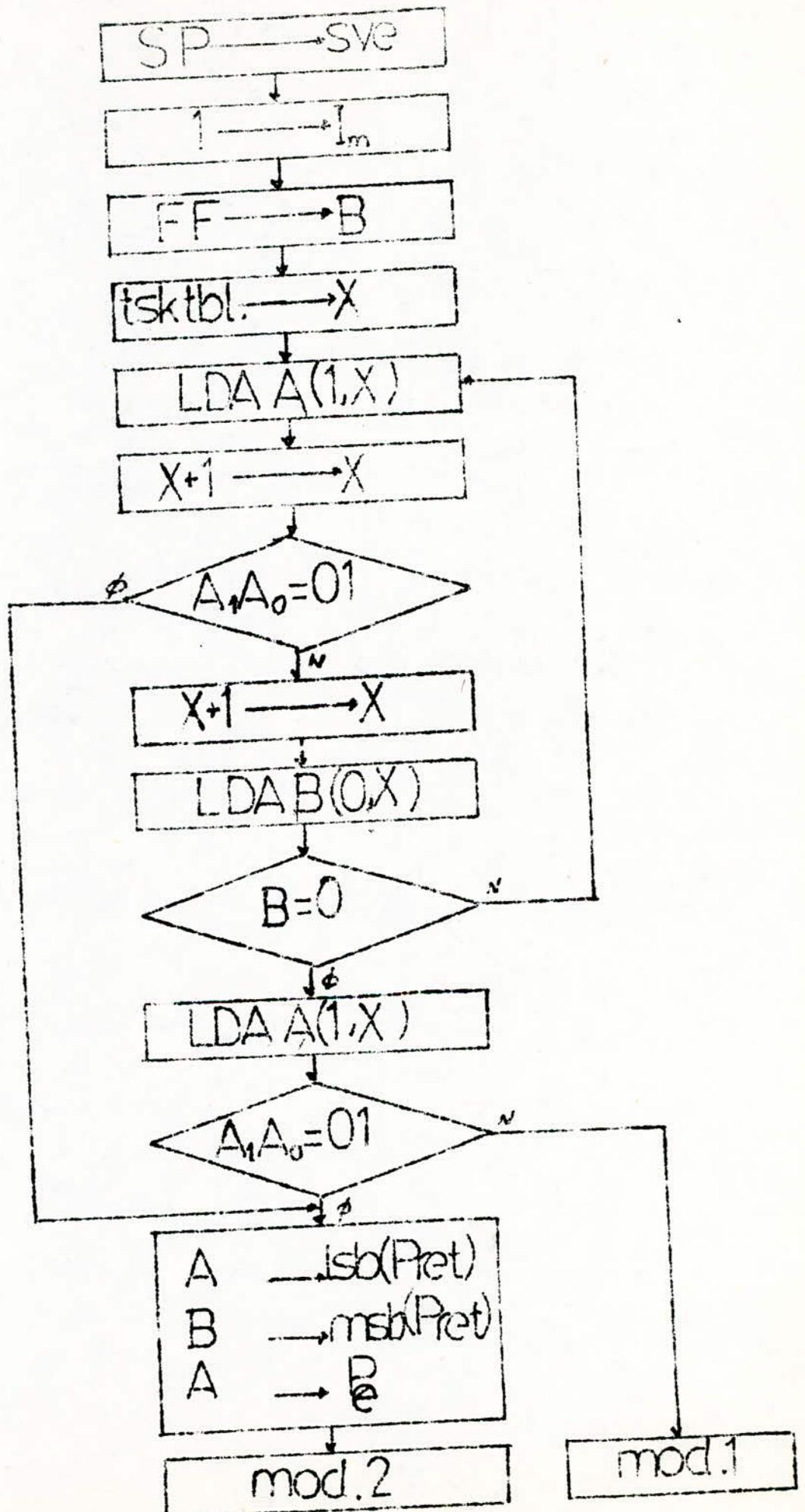
Module mod.7 :

Il passe sous controle du CPU la tache la plus prioritaire prete et met à jour son mot d'état dans la table des taches TSKTBL.

Module MOD.9 :

Il suspend une tache en cours d'exécution après l'existence d'une tache prete plus prioritaire. Il remet également à jour le mot d'état de la tache suspendue dans la table des taches TSKTBL.

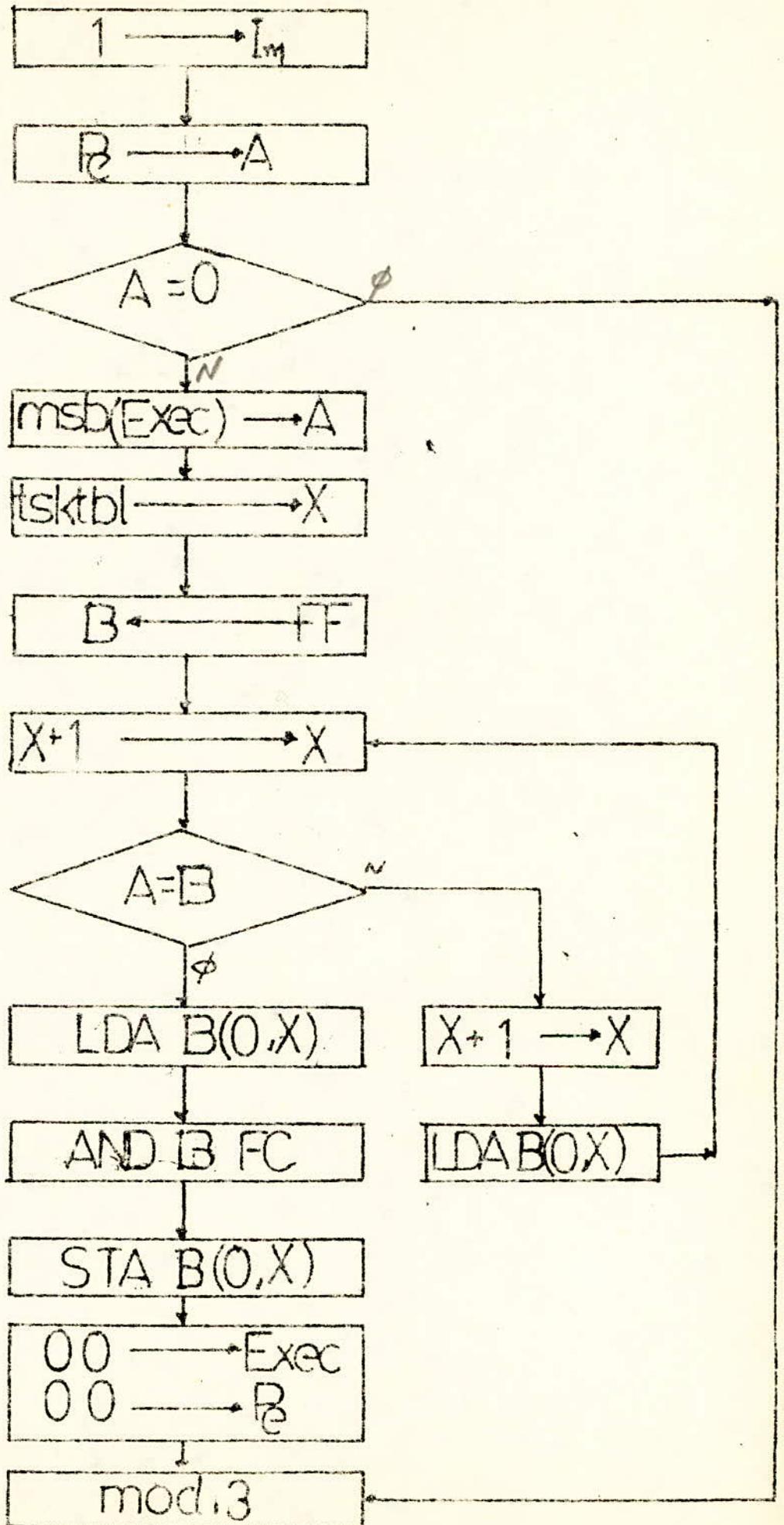
MODULE mod.0



Programme du Moniteur: module MOD. 0

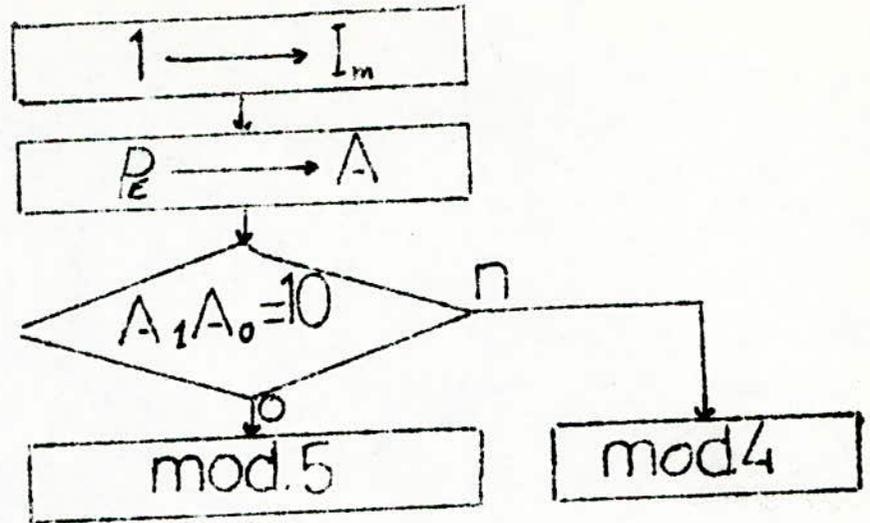
0010	BF	0500		STS	SVE
0013	0F			SEI	
0014	C6	FF		LDAB	
0016	CE	0300		LDX	TSKTBL
0019	46	01	RO3	LDAA	
001B	08			INX	
001C	46			RORA	
001D	24	11		BCC	RO1
001F	46			RORA	
0020	25	0E		BCS	RO2
0022	49		PS	ROLA	
0023	49			ROLA	
0024	B7	0801		STAA	Lsb(PRET)
0027	B7	0804		STAA	Pr
002A	F7	0800		STAB	Msb(PRET)
002B	7E	0080		JUMP	Mod. 2
0030	08		RO(1,2)	INX	
0031	E6	00		LDAB	
0033	5D			TSTB	
0034	26	E3		BNE	RO3
0036	46	01		LDAA	
0038	46			RORA	
0039	24	06		BCC	RO4
003B	46			RORA	
003C	25	03		BCS	RO5
003E	7E	0022		JUMP	PS
0041	7E	0050	RO(4,5)	JUMP	Mod. 1

MODULE mod.1

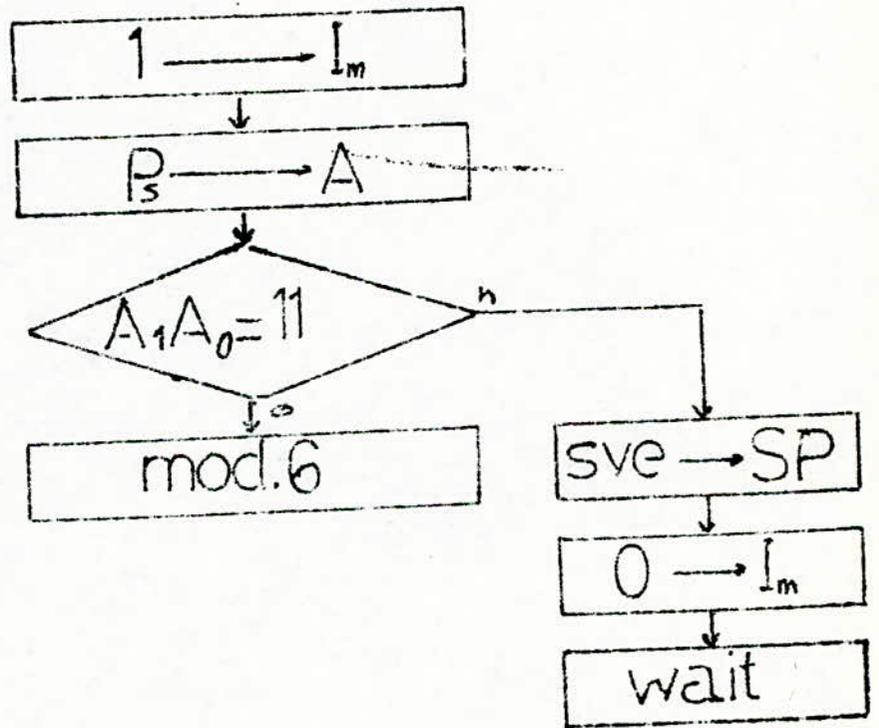


Programme du Moniteur: Module MOD.1

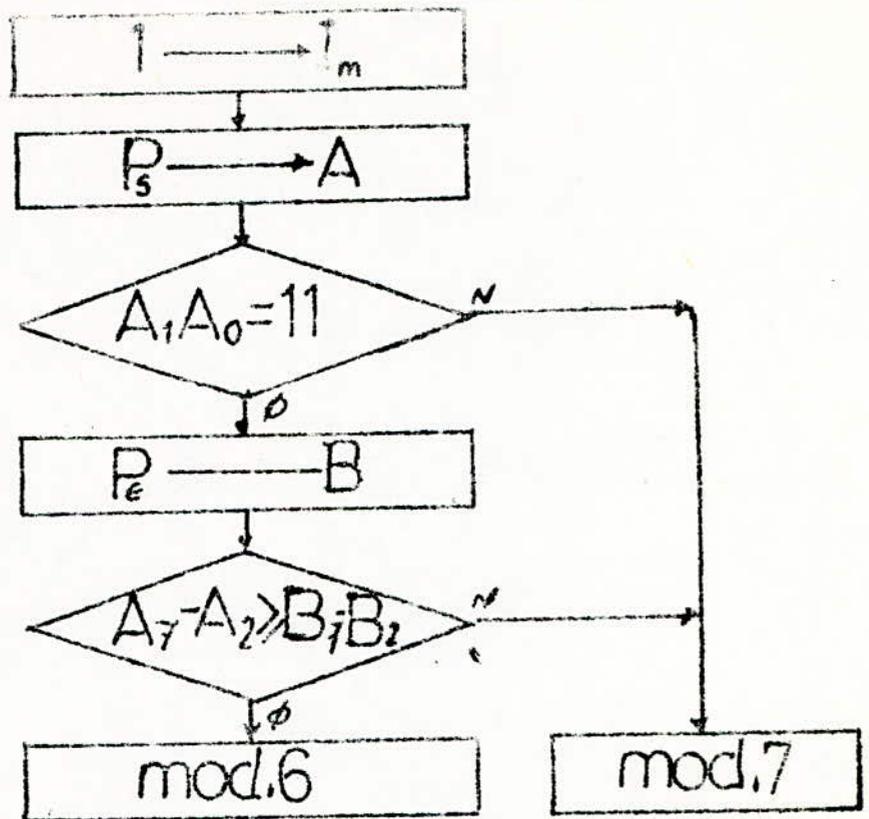
0050	0F			SEI	
0051	B6	0805		LDAB	Pe
0054	4D			TSTA	
0055	27	1B		BEQ	ROO
0057	E6	0802		LDAA	Msb(EXEC.)
005A	CE	0300		LDX	TSKTBL
005D	C6	FF		LDAB	
0060	08		PS1	INX	
0060	11			CBA	
0061	24	12		BNE	RO1
0063	E6	00		LDAB	
0065	C4	FC		ANDB	
0067	E7	00		STAB	
0069	7F	0802		CLR	Msb(EXEC.)
006C	7F	0803		CLR	Lsb(EXEC.)
006F	7F	0805		CLR	Pe
0072	7E	0090	ROO	JUMP	Mod.3
0075	08		RO1	INX	
0076	E6	00		LDAB	
0078	7E	005F		JUMP	PS1



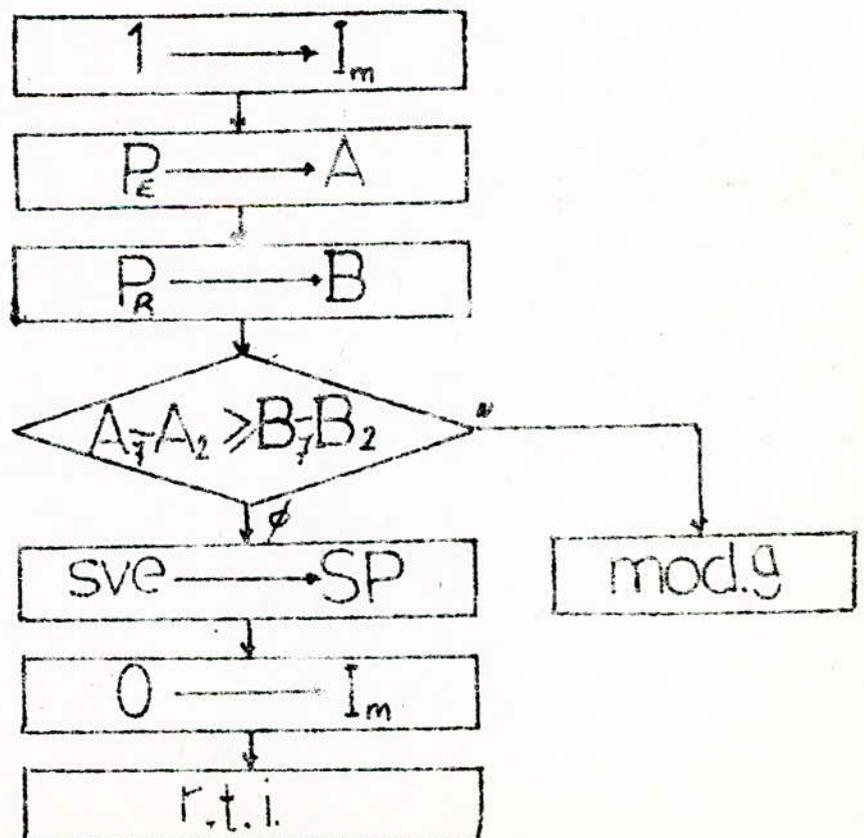
MODULE mod.2



MODULE mod.3



MODULE mod.4



MODULE mod.5

PROGRAMME DU MONITEUR: MOD.2

0080	0F			SEI	
0081	B6	0805		LDAA	Pe
0084	44			LSRA	
0085	25	10		BCC	RO1
0087	44			LSRA	
0088	24	03		BCC	RO2
008A	7E	00C0		JUMP	MOD.5
008D	7E	00A5	RO(1,2)	JUMP	MOD.4

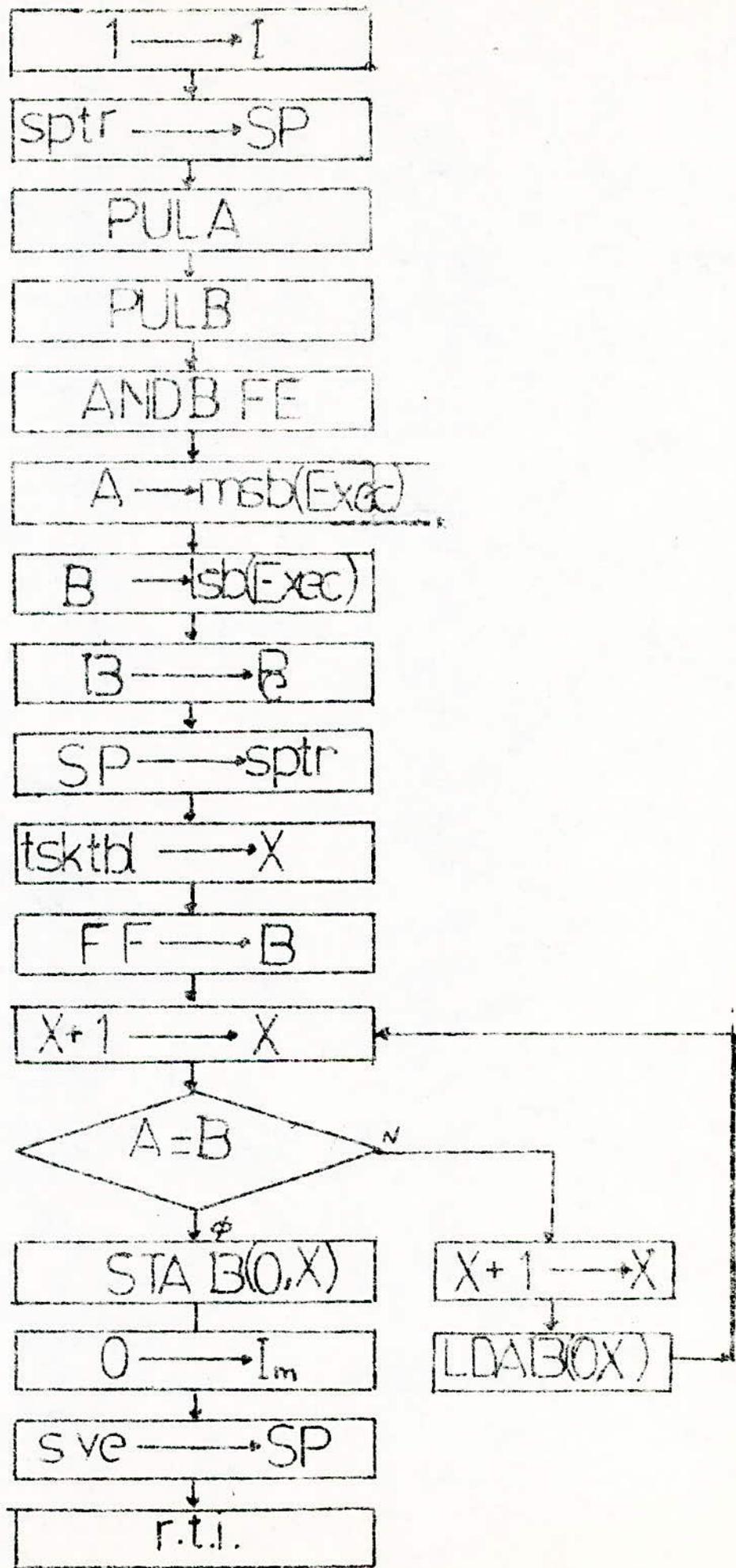
PROGRAMME DU MONITEUR: MOD.3

0090	0F			SEI	
0091	B6	0806		LDAA	Ps
0094	44			LSRA	
0095	24	06		BCC	RO1
0097	44			LSRA	
0098	24	03		BCC	RO2
009A	7E	00E0		JUMP	MOD.6
009D	BE	0500	RO(1,2)	LDS	SVE
00A0	0E			CKR	
00A1	3E			WAI	

PROGRAMME DU MONITEUR: MOD.4

00A5	0F			SEI	
00A6	B6	0806		LDAA	Ps
00A9	44			LSRA	
00AA	24	0E		BCC	RO1
00AC	44			LSRA	
00AD	24	0B		BCC	RO2
00AF	F6	0804		LDAB	Pr
00B2	54			LSRB	
00B3	54			LSRB	
00B4	11			CBA	
00B5	2D	03		BLT	RO3
00B7	7E	00E0		JUMP	MOD.6
00BA	7E	0110	RO(1,2,3)	JUMP	MOD.7

MODULE mod.6



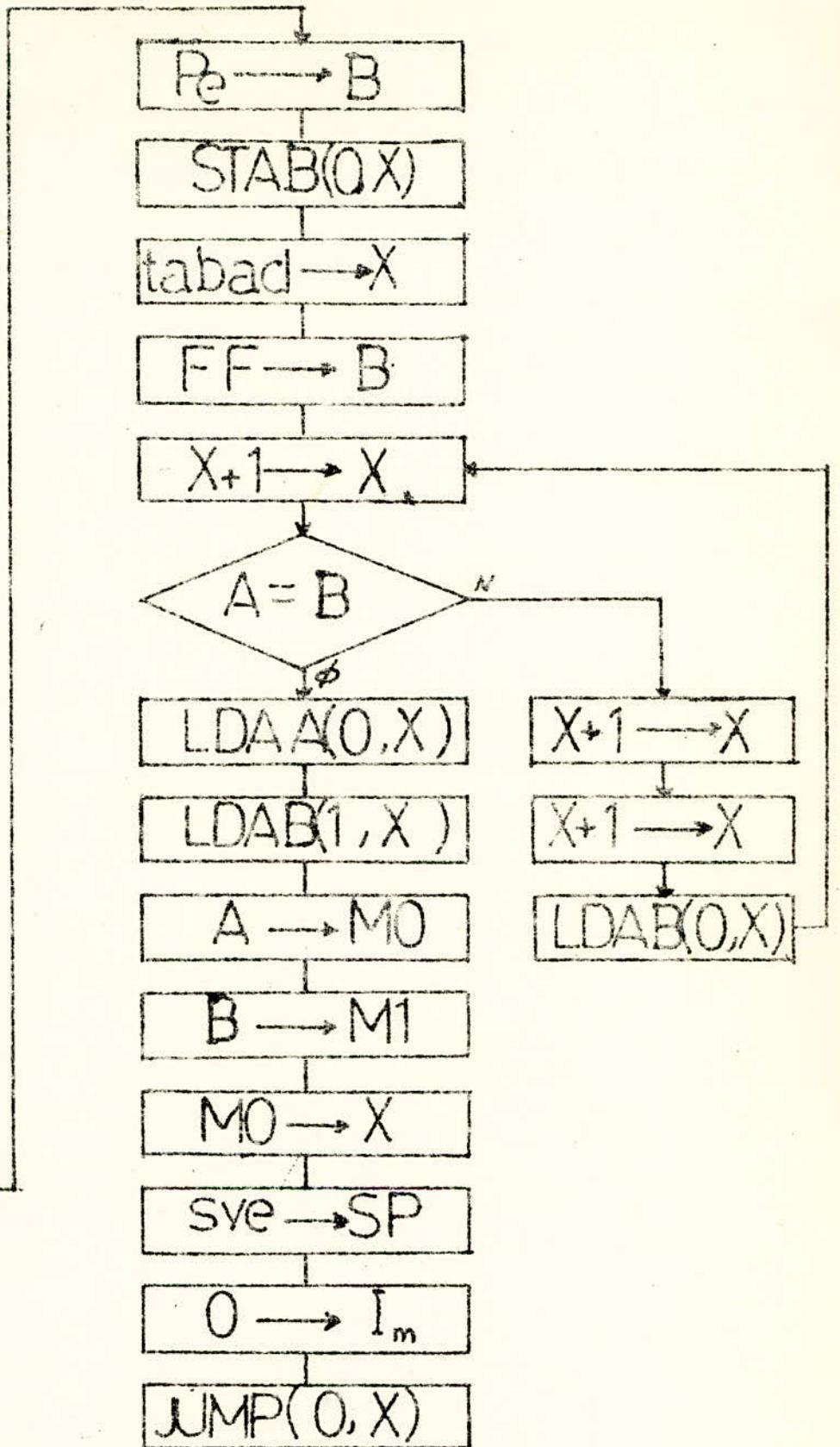
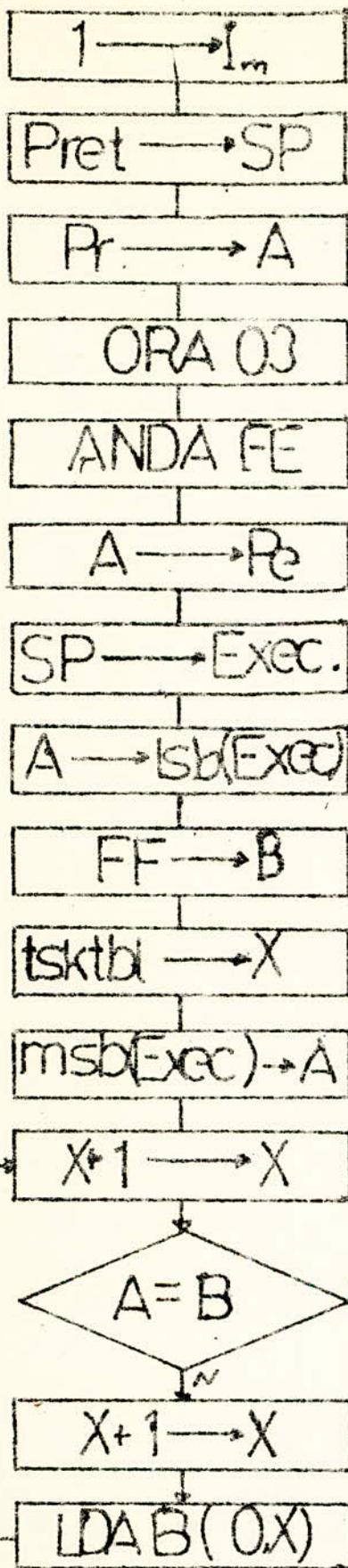
PROGRAMME DU MONITEUR/ MOD.5

00C0	0F			SEI	
00C1	B6	0805		LDAA	Pe
00C4	F6	0804		LDAB	Pr
00C7	54			LSRB	
00C8	54			LSRB	
00C9	44			LSRA	
00CA	44			LSRA	
00CB	11			CBA	
00CC	2D	05		BLT	ROO
00CE	BE	0500		LDS	SVE
00D0	0E			CLRI	
00D2	3B			RTI	
00D3	7E	0160	ROO	JUMP	MOD.9

PROGRAMME DU MONITEUR:MOD.6

00E0	0F			SEI	
00E1	BE	0807		LDS	SPTR
00E4	32			PULA	
00E5	33			PULB	
00E6	C4	FE		ANDB	
00E8	B7	0802		STAA	Msb(Exec.)
00EB	F7	0803		STAB	Lsb(Exec.)
00EE	F7	0805		STAB	Pe
00F1	BF	0807		STS	SPTR
00F4	C6	FF		LDAB	
00F6	CE	0300		LDX	Tsktbl
00F9	08		PSO	INX	
00FA	11			CBA	
00FB	26	07		BNE	RO1
00FD	F6	0805		LDAB	Pe
0100	E7	00		STAB	
0102	0F			CLRI	
0103	BE	0500		LDS	SVE
0106	3B			RTI	
0107	08		RO1	INX	
0108	E6	00		LDAB	
010A	7E	00F9		JUMP	PSO

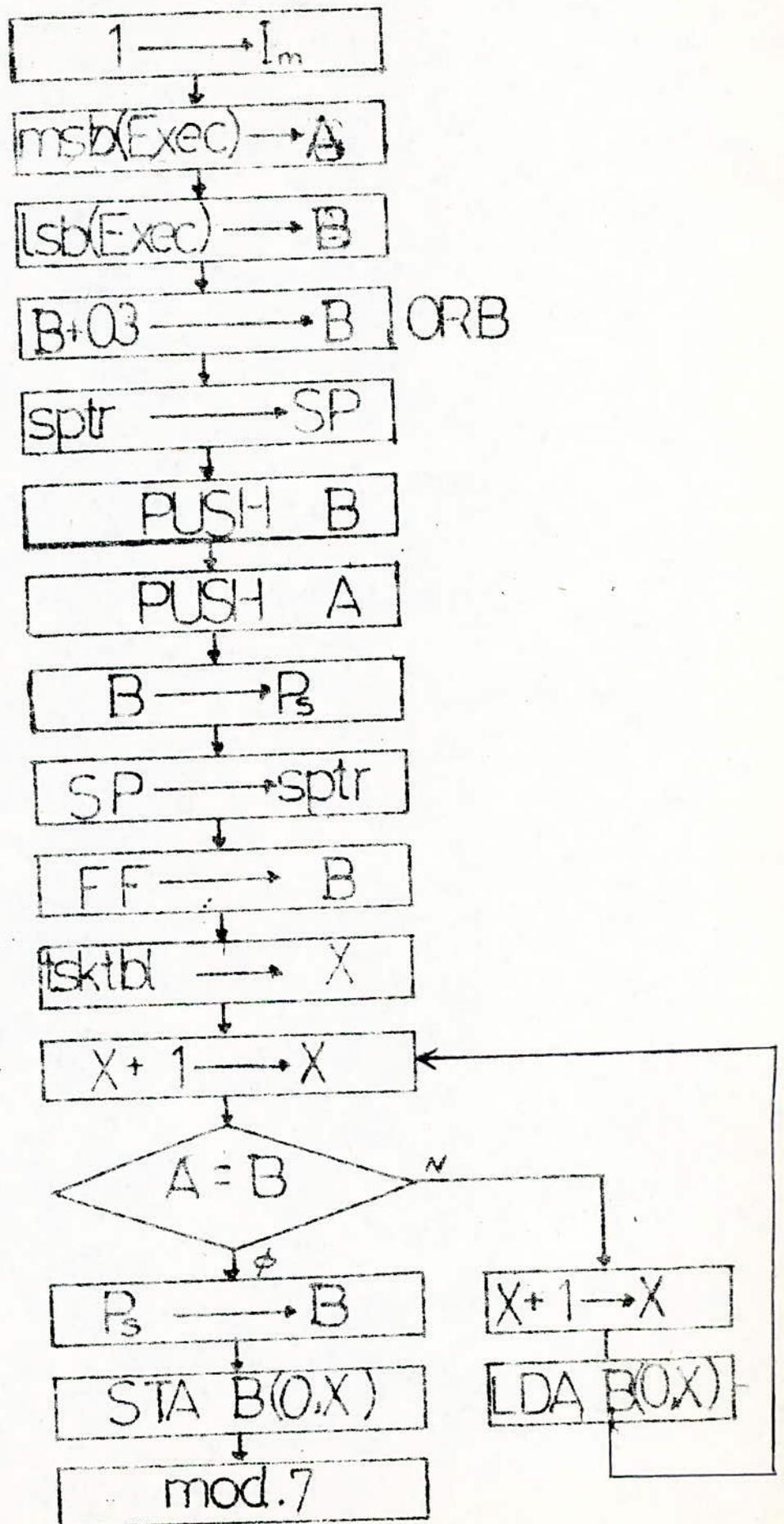
MODULE mod.7



PROGRAMME ■ DU MONITEUR:MOD.7

0110	0F			SET	
0111	B6	0804		LDAA	Pr
0114	8A	02		ORA	
0116	84	FE		ANDA	
0118	BE	0800		LDS	PRET
011B	B7	0805		STAA	Pe
011E	BF	0802		STS	EXEC
0121	B7	0803		STAA	Lsb(EXEC)
0124	C6	FF		LDAB	
0126	CE	0300		LDX	TSKTPBL
0129	B6	0802		LDAA	Msb(EXEC)
012C	08		PS1	INX	
012D	11			CBA	
012E	26	25		BNE	RO1
0130	F6	0805		LDAB	Pe
0133	E7	00		STAB	
0135	CE	0400		LDX	TABAD
013 8	C6	FF		LDAB	
013A	08		PS0	INX	
013B	11			CBA	
013C	26	10		BNE	RO2
013E	A6	00		LDAA	
0140	E6	01		LDAB	
0142	B7	0809		STAA	MO
014 5	F7	080A		STAB	M1
0148	BE	0500		LDS	SVE
014B	0E			CLI	
014C	6E	00		JUMP	
014E	08		RO2	INX	
014F	08			INX	
0150	E6	00		LDAB	
0152	7E	013A		JUMP	PS0
0155	08		RO1	INX	
0156	E6	00		LDAB	
0158	7E	012C		JUMP	PS1

MODULE mod.9



PROGRAMME DU MONITEUR: MOD.9

0160	0F			SEI	
0161	B6	0802		LDAA	Msb(EXEC)
0164	F6	0803		LDAB	Lsb(EXEC)
0167	CA	03		ORA	
0169	BE	0807		LDS	SPTR
016C	37			PUSHB	
016D	36			PUSHA	
016E	F7	0806		STAB	Ps
0171	C6	FF		LDAB	
0173	BF	0807		STS	SPTR
0176	CE	0300		LDX	TSKTBL
0179	08		PSO	INX	
017A	11			CBA	
017B	26	08		BNE	ROO
017D	F6	0806		LDAB	Ps
0180	E7	00		STAB	
0182	7E	0110		JUMP	MOD.7
0185	08		ROO	INX	
0186	E6	00		LDAB	
0188	7E	0179		JUMP	PSO

CHAPITRE IV, LE M.C. 6840

Introduction

Le M.C. 6840 est un composant programmable de la famille M.6800 de MOTOROLA. Il se compose de trois compteurs binaires de seize bits chacun, trois registres de contrôle leur correspondant et d'un registre d'état. Ces compteurs, contrôlés par logiciel, peuvent générer des interruptions et (ou) des signaux de sortie.

Le M.C. 6840 (P.T.M.: Programmable Timer Module) peut être utilisé comme fréquence-mètre, compteur d'événements, générateur de signaux de durée et de largeur programmable.

A-) Description externe.

Le PTM se présente sous la forme d'un boîtier à 28 broches se répartissant comme suit:

1) Huit lignes bidirectionnelles DATA (D0-D7), servant à l'échange de données entre PTM et MPU seulement. En dehors des opérations d'écriture ou de lecture, le bus Données du PTM est déconnecté (état haute impédance).

2) Une ligne d'entrée R/\bar{W} (Lecture/Ecriture): ce signal, issu du MPU, contrôle le sens de transfert des données. Ainsi:
- $R/\bar{W}=1$ le PTM est sollicité en lecture et le transfert des données se fait du PTM vers le MPU.

- $R/\bar{W}=0$ le PTM est sollicité en écriture et le transfert a lieu dans le sens inverse du précédent.

3) Deux lignes d'entrée "Chip Select" ($\overline{CS0}$, $CS1$):

Ces deux lignes, issues du MPU (bus d'adresses), permettent dans une combinaison unique la sélection du PTM. Ainsi, les opérations de Lecture/Ecriture ne peuvent avoir lieu que si cette combinaison ($\overline{CS0}=0$, $CS1=1$) est satisfaite, en accord avec une autre condition sur $\phi 2$.

4) Trois lignes (d'entrée) de sélection de registres internes:

Ces lignes ($RS0$, $RS1$, $RS2$) permettent la sélection des registres internes, latches et compteurs sur lesquelles portent les opérations d'écriture/Lecture.

5) Une ligne (d'entrée) de validation $\phi 2$ (Enable):

Ce signal permet la synchronisation des transferts de données entre PTM et MPU. C'est sur le front descendant de $\phi 2$ que s'effectuent ces échanges.

Il synchronise également les signaux injectés sur les entrées Gate, horloges externes, et le signal RESET.

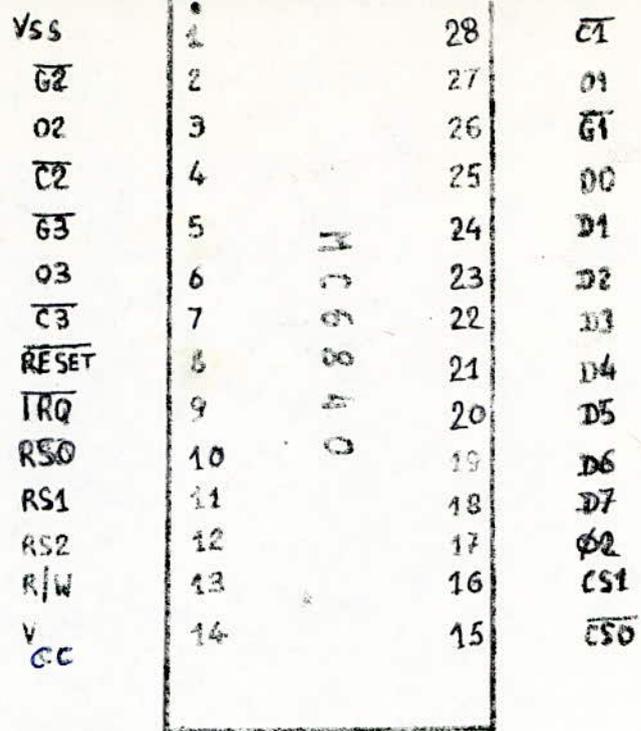


FIGURE 4.1 Schema des broches

REGISTRES INTERNES DU P.T.M (MC.6860)

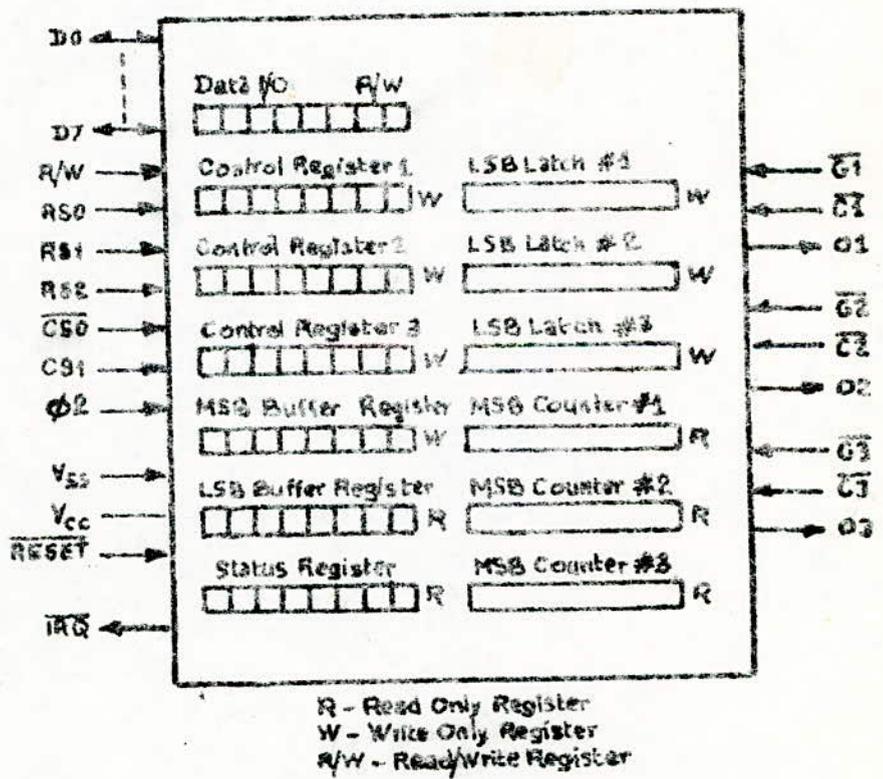


FIGURE 4.3

6°) Une ligne de sortie \overline{IRQ} (demande d'interruption)
Cette ligne, activée au niveau bas, permet si elle est validée d'adresser une demande d'interruption au MPU en vue d'effectuer un certain travail programmé par l'utilisateur.

7) Une ligne d'entrée \overline{RESET} :

Activée au niveau bas ($\overline{Reset}=0$), cette ligne issue du MPU provoque dans le PTM, les conséquences suivantes:

- tous les latches sont chargés à leur valeur maximale (2^{16})
- tous les compteurs sont initialisés de même
- tous les bits des 3 registres de contrôle sont mis à zéro, excepté le bit 0 du registre de contrôle 1 qui est mis à 1.
- toutes les horloges, toutes les sorties des compteurs, sont invalidées.

- tous les bits du registre d'état sont mis à "0"

8) Les lignes d'entrée horloges ($\overline{C1}, \overline{C2}, \overline{C3}$):

Les signaux injectés sur ces entrées sont synchronisés par $\phi 2$. Leurs fréquences peuvent varier du continu à la fréquence maximale $\phi 2$ de 1 MHz. Seule la 4ème impulsion de $\phi 2$ permet la décrémentation du compteur interne, les 3 autres impulsions précédentes étant utilisées pour la "mise au point" du signal et sa reconnaissance interne par le PTM. Par ailleurs, l'entrée d'horloge externe $\overline{C3}$ offre une possibilité supplémentaire au programmeur d'utiliser le mode optionnel prédivision par 8 ($\div 8$), en injectant sur $\overline{C3}$ une fréquence maximale de 4 MHz.

9) Trois lignes d'entrée Gate ($\overline{G1}, \overline{G2}, \overline{G3}$):

Ces lignes servent à injecter des signaux de validation des entrées horloge ($\overline{C1}, \overline{C2}, \overline{C3}$) respectivement. Elles sont compatibles TTL et acceptent des signaux asynchrones. Les signaux d'entrées Gate sont reconnus par le PTM de la même façon que pour les entrées horloges et affectent directement les compteurs internes à 16 bits. Cependant, le mode $\div 8$ ne peut être utilisée pour l'entrée $\overline{G3}$.

10) Trois lignes de sortie des compteurs ($O1, O2, O3$):

Elles permettent, si elles sont validées, de visualiser des signaux définis seulement pour chacun des modes Continu et Impulsionnel (single-shot).

Nous reviendrons plus en détail sur le type de signaux obtenus en sortie lors de l'étude des modes de fonctionnement.

B-) Description interne: étude des registres

(voir figures 4-2 et 4-3)

Le MC 6840 dispose de:

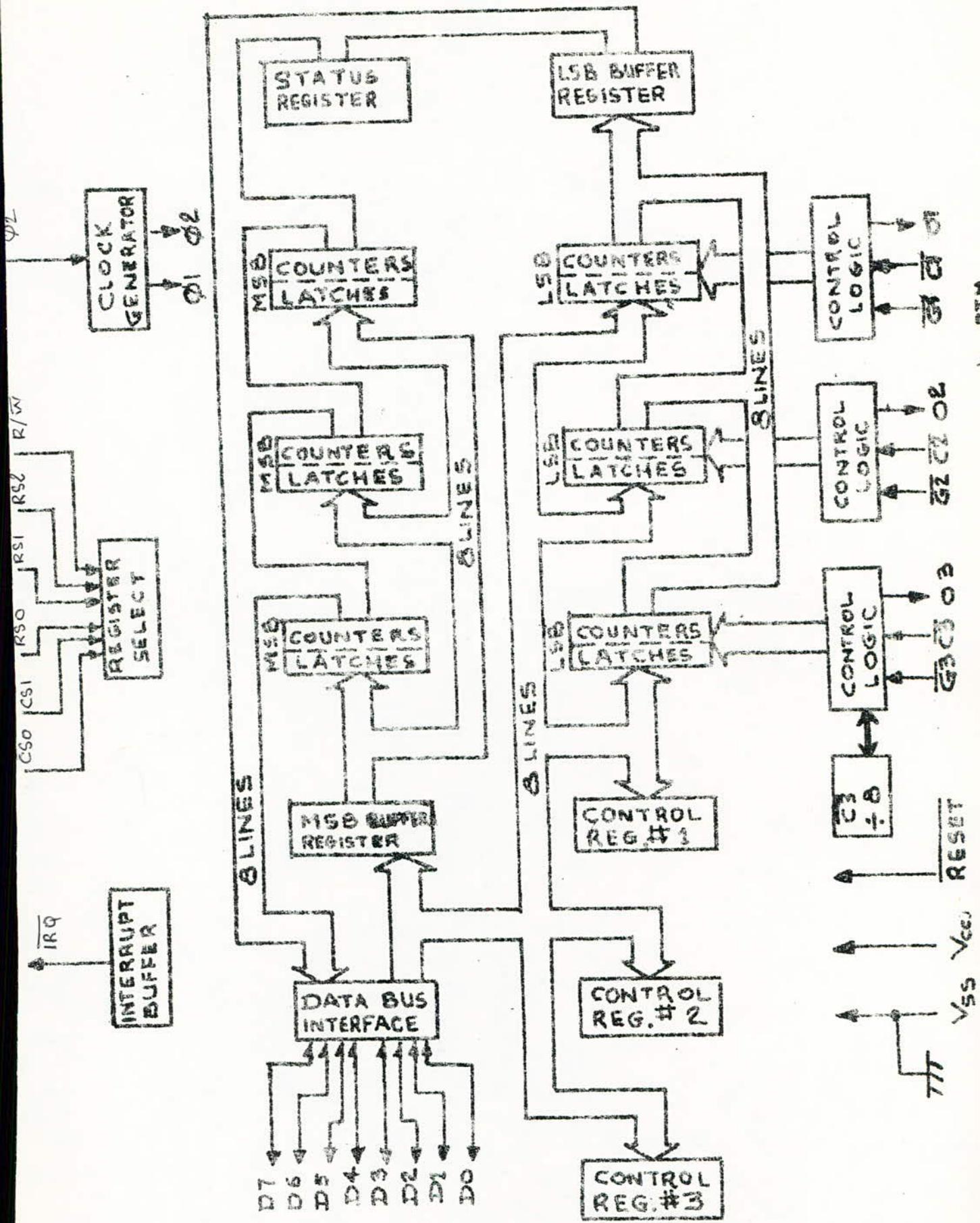


FIGURE 4.2 : Bloc-diagramme du P.T.M.

-Trois registres de controle (notés: CR1, CR2, CR3) affectés respectivement aux Timers 1, 2 et 3.

-Un registre buffer (temporaire) MSB

-Un " " " LSB

-Trois latches LSB (LSB1, 2, 3) affectés respectivement aux Timers 1, 2 et 3.

-Trois latches MSB (MSB1, 2, 3) affectés de meme.

-Un registre d'état.

1) Les registres de controle

Ces 3 registres, de 8 bits chacun, sont à écriture seulement. Ils définissent, chacun pour le timer auquel il est affecté,

-le mode de fonctionnement

-la validation de la ligne \overline{IRQ} et de la sortie

-le choix de l'horloge à utiliser (externe ou interne $\phi 2$)

Une exception existe toutefois pour les bits 0 des registres de controle 1 et 2, lesquels servent aux 3 Timers.

1-a) Registre de controle 1: bit 0 (CR10)

Ce bit sert de Reset interne du PPM.

-CR10=0: il permet aux 3 compteurs de fonctionner selon les contenus des autres bits des registres de controle.

CR10=1: il initialise les compteurs avec les contenus des latches leur correspondant, invalide les horloges et annule toutes les sorties et tous les bits du registre d'état. Les latches et les registres de controle ne sont pas affectés par l'état de ce bit et peuvent donc être écrits.

1-b) Registre de controle 2 / bit 0 (CR20)

Ce bit est utilisé comme bit d'adressage supplémentaire des registres de controle 1 et "3, en coordination avec la combinaison (RS0=0, RS1=0, RS2=0).

1-c) Registre de controle 3: bit 0 (CR30)

Ce bit permet la sélection de la prédivision par 8 ($\div 8$) de l'horloge utilisé par le Timer 3.

En mode ($\div 8$), la fréquence maximale permise sur l'entrée horloge $\overline{O3}$ est de 4 MHz.

1-d) Bit 1 des "3 registres de controle (CR11, CR21, CR31)

Il sélectionne le type d'horloge (interne $\phi 2$ ou externe) utilisé

par le Timer correspondant.

1-e) Bit 2 des 3 registres de controle (CR12, CR22, CR32)

Il détermine si le mot contenu dans les latches LSB et MSB d'un meme Timer est traité dans le compteur du Timer comme un mot binaire unique de 16 bits (mode normal 16 bits) ou comme 2 mots distincts de 8 bits (mode Dual 8 bits).

1-f) Bits 3,4,5 des 3 registres de controle (CRX3, CRX4, CRX5- X=1,2,3)

Par leur combinaison, ces bits définissent le mode de fonctionnement du Timer et seront détaillés ultérieurement.

1-g) Bit 6 des 3 registres de controle (CR16, CR26, CR36)

Il représente le bit de masque de la ligne de sortie \overline{IRQ} .

1-h) Bit 7 des 3 registres de controle (CR17, CR27, CR37)

Il permet la validation des sorties des compteurs, lesquelles n'affectent pas le fonctionnement des Timers.

La figure 4-4 résume l'essentiel de la programmation des registres de controle

B-2) Le registre d'état (noté RS)

C'est un registre à lecture seulement, contenant 4 bits indicateurs (flags) d'état-les 4 autres bits sont forcés à 1 et reviennent à 0 en lecture.

Les bits 0,1 et 2 sont assignés respectivement aux Timers 1,2 et 3 et ils traduisent l'état de ces compteurs.

Par exemple le bit 0 du registre d'état se met à 1 quand le contenu du compteur 1 devient nul (on dit alors que le compteur a atteint le Time Out -T.O.--)

Le bit 7 est un indicateur d'état, d'interruption composite; il se met à 1 si un des indicateurs d'interruption est à 1 ainsi que le bit 6 du registre de controle correspondant au Timer (c-à-d que la ligne \overline{IRQ} est validée).

L'expression mathématique du bit 7 peut s'écrire:

$$\text{Bit 7} = \text{Bit 0} \cdot \text{CR16} + \text{Bit 1} \cdot \text{CR26} + \text{Bit 2} \cdot \text{CR36}$$

La remise à zéro des flags individuels d'interruption peut se faire par:

-la Reset interne (CR10=1) ou externe $\overline{\text{Reset}}=0$

-la séquence:

- Lecture bit registre d'état / Lecture compteur correspondant

- l'initialisation compteur

Voir table 4-5 : résumé des bits de RS.

B-3) Le registre temporaire MSB

TABLE DE PROGRAMMATION DES REGISTRES DE CONTROLE

CR10 : bit de Reset interne		CR20 : bit d'adressage		CR30 : bit de prédivi- sion	
0	fonctionnement des Timers	0	écriture de CR3	0	prédivi- sion par 1
1	initialisation des Timers	1	écriture de CR1	1	prédivi- sion par 8
CRX1		choix de l'horloge à utiliser			
0		le Timer X utilise l'horloge externe \overline{CX}			
1		le Timer X utilise l'horloge interne $\phi 2$			
CRX2		choix du mode de comptage			
0		le Timer X est programmé en mode normal 16 bits			
1		le Timer X est programmé en mode dual-8-bits			
CRX3, CRX4, CRX5		définissent le mode de fonctionnement			
CRX6		validation de la ligne d'interruption \overline{IRQ}			
0		ligne d'interruption \overline{IRQ} masquée			
1		ligne d'interruption validée			
CRX7		validation de la sortie du compteur			
0		ligne de sortie du compteur invalidée			
1		ligne de sortie du compteur validée			

X = 1, 2, 3.

figure 4-4

Registre de huit bits, à écriture seulement, il sert au stockage temporaire de l'octet MSB de la donnée à écrire. Il est repéré par 3 adresses différentes (voir table d'adressage: figure 4-6) . La séquence d'écriture des latches d'un Timer est la suivante:

- écriture du registre buffer MSB
- écriture du latch LSB du Timer provoquant simultanément le transfert du contenu du registre MSB dans le latch MSB du Timer considéré.

B-4) Le registre temporaire LSB

Ce registre de 8 bits, à lecture seulement, sert au transit de l'octet LSB de la donnée à lire. La séquence de lecture est la suivante:

- lecture directe de l'octet MSB du compteur, plaçant simultanément l'octet LSB de la donnée dans le registre buffer LSB.
- une commande de lecture de ce registre nous délivre alors son contenu.

C-) Modes de fonctionnement des Timers

Chacun des Timer peut être programmé individuellement pour fonctionner selon un mode choisi parmi 4 possibles.

Les modes de fonctionnement sont sélectionnés par positionnement des bits 3, 4 et 5 des registres de contrôle, dont la table 4-7 en donne le résumé.

Registre de contrôle			Mode de fonctionnement
CRX3	CRX4	CRX5	
0	∅	0	Continu
0	∅	1	Impulsionnel
1	0	∅	Comparateur de fréquence
1	1	∅	Comparateur de largeur d'impulsion

X=1, 2 ou 3.

∅ définit d'autres variantes du même mode.

C-1) Le mode continu (CRX3=0=CRX5)

-existe en deux versions: Normal 16 bits (CRX2=0) ou en Dual -8-bits (CRX2=1).

En normal 16 bits, les octets contenus dans les latches MSB et LSB d'un Timer forme un mot unique de valeur N traité comme tel dans le compteur. Le T.O. compteur a lieu après (N+1) impulsions d'horloge.

La sortie, si elle est validée, change d'état à chaque T.O. compteur. En mode Dual-8-Bits (CRX2=1), les contenus des latches LSB et MSB du Timer sont affectés de poids différents et sont traités dans le compteur comme 2 mots distincts.

TABLE DU REGISTRE D'ETAT DU PTM

SR0	Indicateur d'interruption du Timer 1
0	SR7 et \overline{IRQ} invalidés
1	SR7 et \overline{IRQ} validés si CR16 = 1
SR1	Indicateur d'interruption du Timer 2
0	SR7 et \overline{IRQ} invalidés
1	SR7 et \overline{IRQ} validés si CR26 = 1
SR2	Indicateur d'interruption du Timer 3
0	SR7 et \overline{IRQ} invalidés
1	SR7 et \overline{IRQ} validés si CR36 = 1
SR3 à SR6	Inutilisés
SR7	Bit d'état composite
0	ligne d'interruption \overline{IRQ} masquée pour tous les Timers
1	ligne d'interruption validée

figure 4-5

TABLE D'ADRESSAGE DU PTM

lignes de sélections			opérations	
RS2	RS1	RS0	R/ \overline{W} = 0 : écriture	R/ \overline{W} = 1 : lecture
0	0	0	CR20 = 0 CR3	Pas d'opérations
			CR20 = 1 CR1	
0	0	1	CR2	Registre d'Etat
0	1	0	buffer MSB	MSB Timer 1 (compteur)
0	1	1	latches Timer 1	buffer LSB
1	0	0	buffer MSB	MSB Timer 2 (compteur)
1	0	1	latches Timer 2	buffer LSB
1	1	0	buffer MSB	MSB Timer 3 (compteur)
1	1	1	latches Timer 3	buffer LSB

figure 4-6

Le T.O. compteur a lieu apres (L+1) (M+1) To où :

L: valeur chargée dans le latch LSB

M: valeur chargée dans le latch MSB

To: période de l'horloge utilisée

Le latch MSB décremente de 1 apres (L+i) To

A L=0, le MSB reste inchangé.

A la lère impulsion suivant l'état L=0, le compteur LSB se reinitialise à la valeur du latch LSB (c-à-d L) et le compteur MSB diminue de 1; la sortie, si elle est validée, reste à l'état bas tant que le compteur MSB est différent de zéro.

A LA lère impulsion d'horloge suivant l'état des compteurs MSB=0 et LSB=0 la sortie passe à l'état haut et le garde pendant L impulsions d'horloge. La sortie reprend l'état bas au T.O. suivant au bout de ces L impulsions d'horloge.

Les flags d'interruption sont mis à 1 à chaque T.O.

Cas particuliers

a-L=0: le Timer aura un fonctionnement semblable au Normal 16 bits excepté que le T.O. compteur a lieu seulement apres (M+1) To.

b-) L=0=M :

La sortie, si elle validée, change d'état avec moitié de celle du signal d'horloge.

Voir table récapitulative figure 4.8

C-2) Le mode Impulsionnel (Single-Shot) - CRX3=0, CRX5=1-

A 3 exceptions pres, ce mode ressemble au précédent:

1) la sortie validée (CRX7=1) retourne au niveau bas au T.O. initial, et le gardera jusqu'au prochain cycle d'initialisation.

2) la validation des compteurs ne dépend pas du signal d'entrée Gate

3) dans le cas où L=M=0 (mode Dual-8-Bits), où N=0 (Normal 16 bits), la sortie validée reste à l'état bas. Le T.O. continue à avoir lieu à la fréquence d'horloge appliquée.

Voir résumé figure 4.9

C-3) Le mode comparateur de fréquence (CRX3=1, CRX4=0)

Chaque Timer du PTM peut être programmé pour comparer la période de l'impulsion appliqué à l'entrée Gate avec la période requise par le T.O. compteur.

Dans ce mode les signaux de sortie ne sont pas définis.

La transition négative du signal de l'entrée Gate valide le compteur et démarre le cycle d'initialisation. Le compteur décremente alors à chaque période d'horloge jusqu'à ce qu'une interruption soit générée, ou une commande d'écriture des latches envoyée ou en-

MODE CONTINU DUAL 8-BITS

DANS CET EXEMPLE : CONTENU DU LATCH MSB : $M = 03$
 CONTENU DU LATCH LSB : $L = 04$

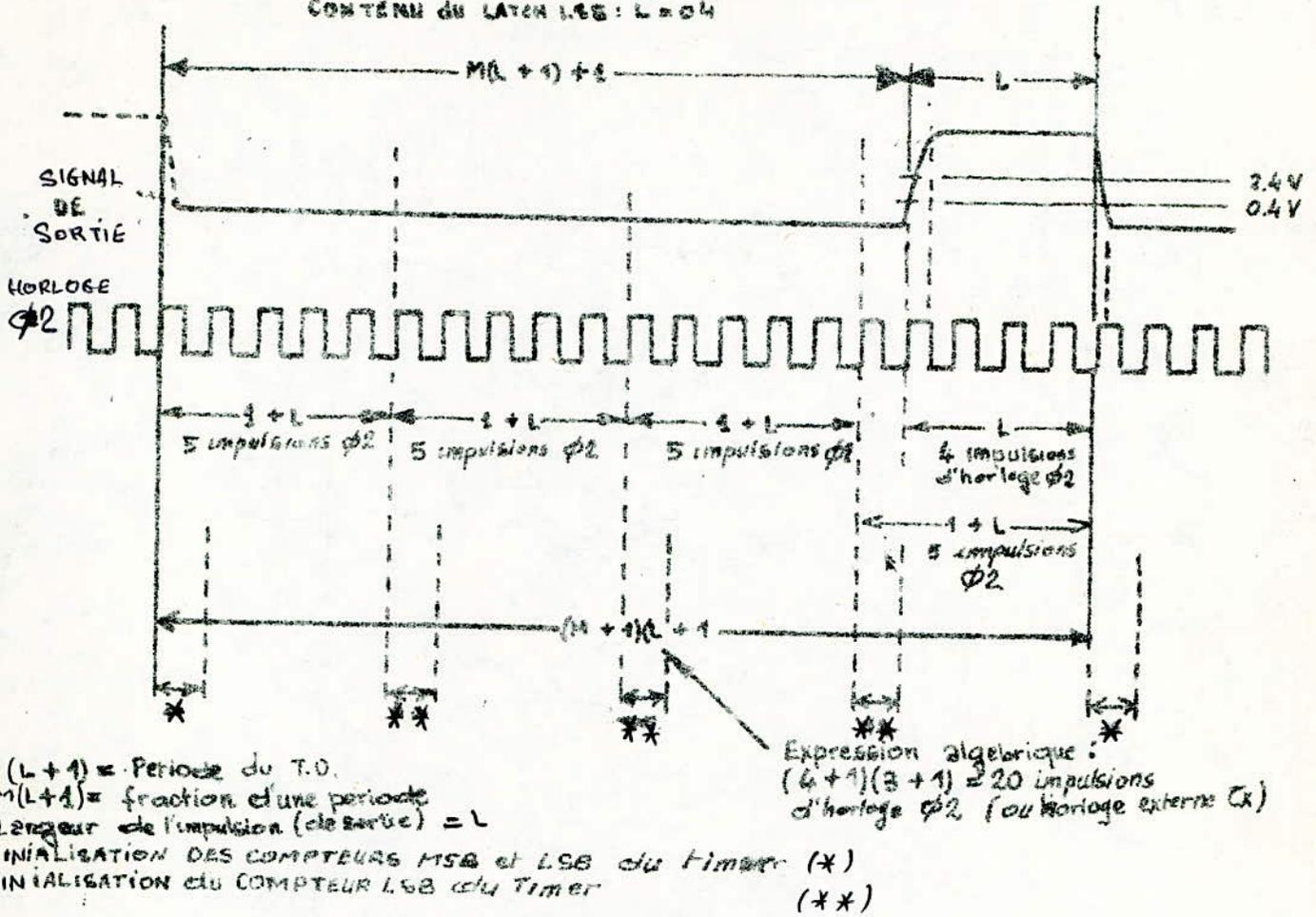


FIGURE 4.10

core une Reset ait lieu.

Si CRX5=1:

Si le T.O. a lieu avant la lère transition négative de \bar{G} ($\bar{G}\downarrow$) et après le cycle d'initialisation du compteur, le flag d'interruption est mis à 1, invalidant le compteur. Aussi, un nouveau cycle d'initialisation ne peut commencer sans l'effacement préalable de ce flag et la reconnaissance par le Timer de la transition négative de $\bar{G}\downarrow$.

Si CRX5=0:

L'interruption est générée si l'entrée Gate retourne au niveau bas avant le T.O. L'horloge est alors invalidée. La réinitialisation se fait sur le front négatif de $\bar{G}\downarrow$ si et seulement si le flag d'interruption a été préalablement effacé.

Voir table récapitulative figure 4.12

C-4) Le mode comparateur de largeur d'impulsion : (CRX3=1=CRX4)

Tous les compteurs peuvent être programmés pour décrémenter à chaque période d'horloge à partir de $\bar{G}\downarrow$ jusqu'au prochain $\bar{G}\uparrow$. Le front négatif de $\bar{G}\downarrow$ démarre le cycle d'initialisation et à la lère impulsion, le compteur commencera à décrémenter jusqu'au prochain front montant de $\bar{G}\uparrow$.

Voir table récapitulative figure 4.13

D-) Initialisation

D-a) Initialisation des latches

Chacun des Timers dispose de latches, à écriture seulement, d'une capacité totale de 16 bits. Leur chargement (initialisation) peut se faire par:

-une commande d'écriture des latches

-une Reset externe ($\overline{\text{Reset}}=0$): dans ce cas les latches sont

chargés à leur valeur maximale ($\text{LSB}=2^8$, $\text{MSB}=2^8$)

D-b) Initialisation des compteurs

Chaque Timer du PPM dispose d'un compteur, à lecture seulement, d'une capacité de 16 (seize) bits. Son initialisation consiste en un transfert de données des latches au compteur avec remise à 0 du flag d'interruption correspondant. Elle peut se faire par:

-Reset externe ($\overline{\text{Reset}}=0$): dans ce cas le compteur stocke sa valeur maximale (soit $2^{16}=65536$).

-Reset interne ($\text{CR10}=1$): le compteur se charge à la valeur stockée dans les latches lui correspondant.

-commande d'écriture des latches

-reconnaissance de la transition négative G du signal d'entrée Gate

-la séquence: lecture registre d'état / lecture Timer correspondant

Ces 3 dernières méthodes ne sont valables que pour les modes continu

et impulsions.

Modes Intervalle de temps

CRX3 = 1			
CRX4	CRX5	Application	Mise à 1 des flags d'interruption
0	0	comparateur de fréquence	interruption générée si période de Gate est inférieure au TO du compteur
0	1	comparateur de fréquence	interruption générée si période de Gate est supérieure au TO
1	0	comparateur de largeur d'impulsion	interruption générée si le niveau bas du Gate est inférieur au TO
1	1	comparateur de largeur d'impulsion	interruption générée si niveau bas de Gate est supérieur au TO

figure 4.11

Mode comparateur de fréquence

CRX3 = 1 , CRX4 = 0				
CRX5	Initialisation compteur	valid. compt. flip-flop=1	valid. compt. flip-flop=0	flag interruption=I=1
0	$\bar{G} \downarrow \cdot \bar{I} \cdot (CE \cdot TO + \overline{CE}) + R$	$\bar{G} \downarrow \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$	W+R+I	$\bar{G} \downarrow$ avant TO
1	$\bar{G} \downarrow \cdot \bar{I} + R$	$\bar{G} \downarrow \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$	W+R+I	$\bar{G} \downarrow$ après TO

CE : validation compteur

figure 4.12

Mode comparateur de largeur d'impulsion

CRX3 = 1 , CRX4 = 1				
CRX5	Initialisation compteur	valid. compt. flip-flop=1	valid. compt. flip-flop=0	flag interruption=I=1
0	$\bar{G} \downarrow \cdot \bar{I} + R$	$\bar{G} \downarrow \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$	W+R+ \bar{I} +G	$\bar{G} \uparrow$ avant TO
1	$\bar{G} \downarrow \cdot \bar{I} + R$	$\bar{G} \downarrow \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$	W+R+ \bar{I} +G	$\bar{G} \uparrow$ après TO

fig 4.13

FONCTIONNEMENT EN MODE IMPULSIONNEL.

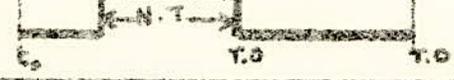
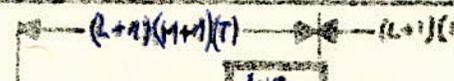
Mode impulsionnel (fixé par $CRX_3=0$, $CRX_5=1$ et $CRX_7=1$)			
REGISTRE DE CONTROLE		INITIALISATION / SIGNAL DE SORTIE	
CRX_2	CRX_4	INITIALISATION DU COMPTEUR	SIGNAL DE SORTIE du TIMER
0	0	$G \downarrow + W + R$	
0	1	$\overline{G} \downarrow + R$	
1	0	$\overline{G} \downarrow + W + R$	
1	1	$\overline{G} \downarrow + R$	

Fig. 4.9

FONCTIONNEMENT EN MODE CONTINU.

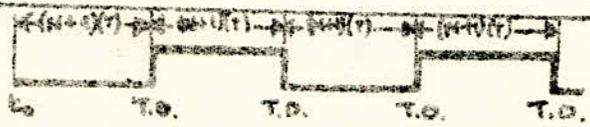
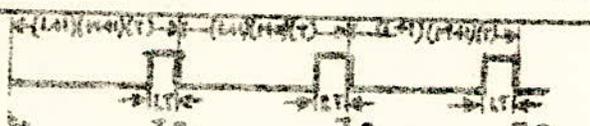
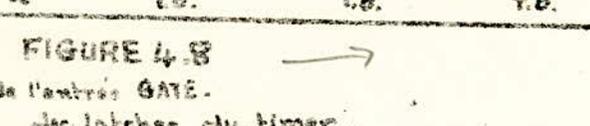
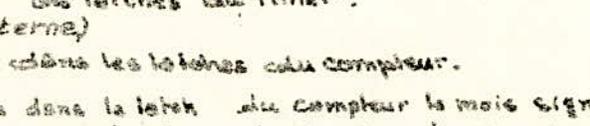
Mode Continu (fixé par $CRX_3=0$, $CRX_5=0$ et $CRX_7=1$)			
REGISTRE DE CONTROLE		INITIALISATION / SIGNAL DE SORTIE	
CRX_2	CRX_4	INITIALISATION DU COMPTEUR	Signal de sortie du timer
0	0	$\overline{G} \downarrow + W + R$	
0	1	$\overline{G} \downarrow + R$	
1	0	$\overline{G} \downarrow + W + R$	
1	1	$\overline{G} \downarrow + R$	

FIGURE 4.8

$\overline{G} \downarrow$ = Transition négative de l'entrée GATE.

W = Commande d'écriture des latches du timer.

R = RESSET (interne ou externe)

N = mot de 16 bits contenu dans les latches du compteur.

L = mot de 8 bits contenu dans la latch du compteur la moins significative (LSB).

M = mot de 8 bits contenu dans la latch du compteur la plus significative (MSB).

T = période de l'horloge d'entrée du compteur.

T_0 = Début du cycle d'initialisation.

$T.O.$ = Fin de compte provoquant une interruption, et elle est valide.

MODE DE FONCTIONNEMENT ET CONTROL DES INTERRUPTIONS.

MODE CRYS CRYN CRYZ	INITIALISATION "CI"	VALIDATION COMPTEUR "CE"	FLAG DE L'INTERRUPTION MIS À 1	MIS À 0
0 0 0 (mode normal)	$\bar{G}_1 + W + R$	$\bar{G} \cdot \bar{R}$	T.O.	RS-RT ou CI
1 0 0 (mode impulsionnel)	$\bar{G}_1 + W + R$	\bar{R}	T.O.	RS-RT ou CI
0 1 0 (mode continu)	$\bar{G}_1 + R$	$\bar{G} \cdot \bar{R}$	T.O.	RS-RT ou CI ou W
1 1 0 (mode impulsionnel)	$\bar{G}_1 + R$	\bar{R}	T.O.	RS-RT ou CI ou W
0 0 1 (mode comparateur de fréquence)	$\bar{G}_1 \cdot \bar{I} \cdot (\bar{C} + T.O.) + R$	$CE_{SET} = \bar{G}_1 \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$ $CE_{RESET} = W + R + I$	\bar{G}_1 avant le T.O.	RS-RT ou CI ou W
1 0 1 (mode comparateur de fréquence)	$\bar{G}_1 \cdot \bar{I} + R$	$CE_{SET} = \bar{G}_1 \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$ $CE_{RESET} = W + R + I$	T.O. avant \bar{G}_1	RS-RT ou CI ou W
0 1 1 (mode comparateur de largeur d'impulsion)	$\bar{G}_1 \cdot \bar{I} + R$	$CE_{SET} = \bar{G}_1 \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$ $CE_{RESET} = W + R + I + S$	\bar{G}_1 avant le T.O.	RS-RT ou CI ou W
1 1 1 (mode comparateur de largeur d'impulsion)	$\bar{G}_1 \cdot \bar{I} + R$	$CE_{SET} = \bar{G}_1 \cdot \bar{W} \cdot \bar{R} \cdot \bar{I}$ $CE_{RESET} = W + R + I + S$	T.O. avant \bar{G}_1	RS-RT ou CI ou W

R = RESET INTERNE ou EXTERNE

W = Commande d'écriture des latchs

I = FLAG d'INTERRUPTION

\bar{G} = Reconnaissance interne du signal Set

C = Reconnaissance interne du signal d'horloge

\bar{G}_1 = Reconnaissance de la transition négative de \bar{G}

\bar{G}_1 = Reconnaissance de la transition positive de \bar{G}

RS-RT = Commande de lecture du registre d'état suivie de la lecture du compteur

T.O. = fin de Comptage

FIGURE 4.14

CH. IV. L'HORLOGE TEMPS REEL.

A-) Description de la carte:

- 1) Le buffer adresses et décodeur d'adresse
- 2) Le buffer données
- 3) Les décades
- 4) Retour sur le M.C. 6840

B-) Applications de la carte

- 1) le calendrier
- 2) le chronomètre

A-) DESCRIPTION DE LA CARTE

1) Le buffer d'adresse

Il comprend quatre circuits "MC 8T26" montés en circuits unidirectionnels par fixation des potentiels de leurs pins de commande. Chaque circuit peut faire transiter quatre lignes d'adresse en assurant une amplification appréciable en courant et une protection contre les court-circuits.

Le décodeur d'adresse, quant à lui, comprend:

- une porte NAND à 8 entrées (SN 7430)
- deux portes NAND à 4 entrées chacune (SN 7420)
- des inverseurs (SN 7404)

LE P.T.M., quand il est adressé, doit pouvoir répondre à 8 adresses seulement, lesquelles correspondent à ses registres internes (voir table d'adressage du MC 6840). Ces registres sont sélectionnés par une combinaison de (RS0, RS1, RS2 et R/\bar{W}). Conformément à ce principe on relie directement les 3 lignes de plus faibles poids du bus adresses du M.P.U. (donc A0, A1, A2) respectivement à (RS0, RS1, RS2). Les treize lignes restantes (A3 à A15), ainsi que la ligne de contrôle VMA, serviront à définir une combinaison d'adressage du P.T.M. lui-même. L'adresse choisie, dans notre cas, est 2620 (en écriture hexadécimale).

L'unicité de cette adresse nous oblige à:

- n'avoir qu'une suite de "1" sur les entrées des portes NAND
- et donc à utiliser, suivant le bit présent à l'entrée du "8T26" la sortie inverseuse correspondante (si ce bit d'entrée est nul) ou la sortie non inverseuse dans le cas contraire.

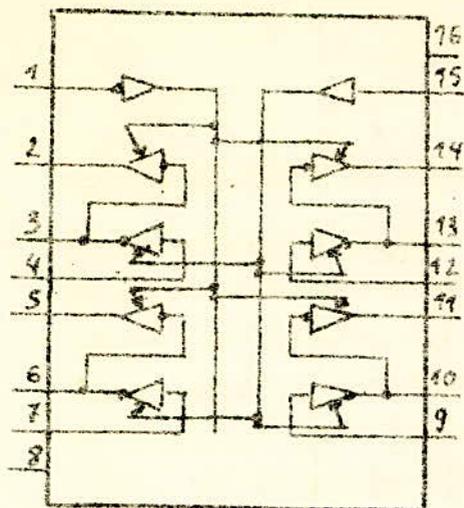
Dans notre cas, le décodage d'adresse nous fournit la pin CS0, l'autre pin CS1 étant prise par inversion de $\overline{CS0}$.

En dehors de l'ensemble des huit adresses (de 2620 à 2627) le PTM n'est plus adressé.

2) Le buffer données

Il comprend deux "8T26" dont la mission est de faciliter l'échange de données entre P.T.M. et M.P.U. Le sens de transfert de ces données sera déterminé par des potentiels (variables) appliqués aux pins de commande. Ces potentiels sont issus des pins 6 et 8 d'un "SN 7420", comme combinaison des signaux ($CS1, R/\bar{W}, \phi 2$), et définissent les trois états possibles du buffer de données:

- 1)-état basse impédance Ecriture ($R/\bar{W}=0$)
- 2)-état basse impédance Lecture ($R/\bar{W}=1$)
- 3)-état haute impédance: buffer données déconnecté

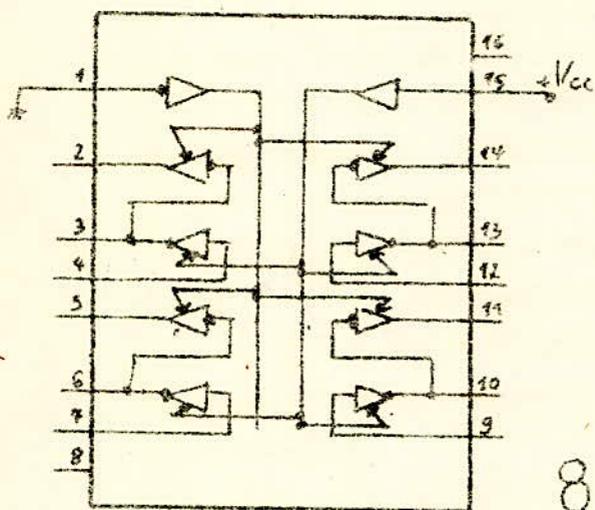


pin 1, 15 : commande

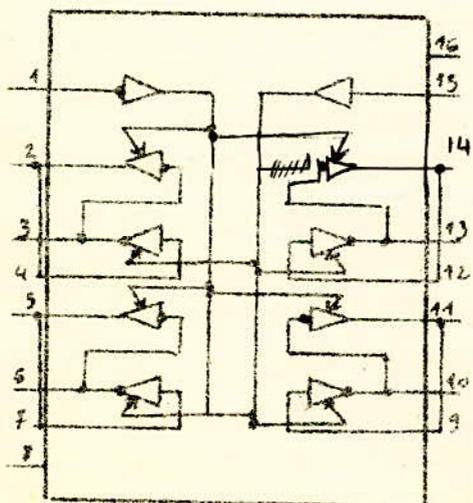
pin 16 : Vcc

8 : masse

schema interne du "8T26"



8T26 monté en unidirectionnel



montage en bidirectionnel

Table de vérité du buffer de Données

R/ \overline{W}	$\phi 2$	CS1	Pin1	Pin15	état du buffer
0	0	0	1	0	haute impédance
0	0	1	1	0	"
0	1	0	1	0	"
0	1	1	1	1	écriture
1	0	0	1	0	haute impédance
1	0	1	1	0	"
1	1	0	1	0	"
1	1	1	0	0	lecture

3) les décades

Les décades sont constitués de circuits "SN 7490". Elles permettent de disposer d'une large gamme de fréquences (1 MHz à 10 Hz) susceptibles d'être utilisées sur les entrées Clock des Timers. Ceci facilite le choix de l'intervalle de temps mis entre deux interruptions successives

Exemple: en mode continu 16 bits, les limites de comptage sont:

$$f=1 \text{ MHz/} \quad N=1 \quad (N+1)T=(1+1) \cdot 10^{-6} = 2 \text{ microsecondes}$$

$$N=65536 \quad (N+1)T=65537 \cdot 10^{-6} \text{ secondes} = 65,537 \text{ millisecondes}$$

$$f=100 \text{ KHZ} \quad N=1 \quad (N+1)T=20 \text{ microsecondes}$$

$$N=65536 \quad (N+1)T=655,37 \text{ millisecondes}$$

$$f=10 \text{ Hz} \quad N=1 \quad (N+1)T=200 \text{ millisecondes}$$

$$N=65536 \quad (N+1)T=1 \text{ h. } 49 \text{ mn. } 13,7 \text{ sec.}$$

L'intervalle entre deux interruptions successives dépend donc de la valeur chargée dans le latch et de la période du signal d'horloge appliqué.

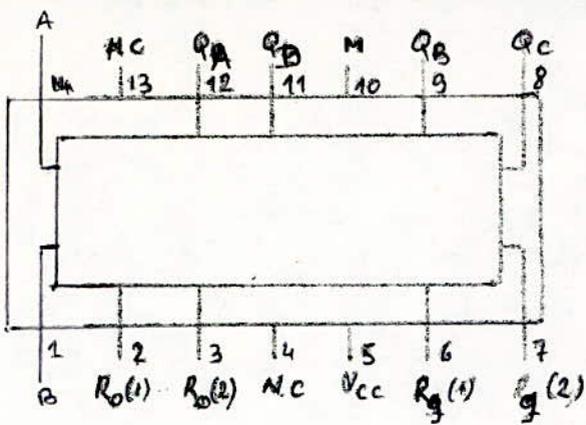
4)-Retour sur le "MC 6840"

Avant d'aborder l'application du Timer qu'est le calendrier, il est bon de préciser quelques détails quant au montage du "MC 6840".

On a vu que les signaux injectés sur les entrées Gate avaient un double rôle:

-valider, sur le niveau bas de ce signal, les horloges des Timers et par conséquent le comptage.

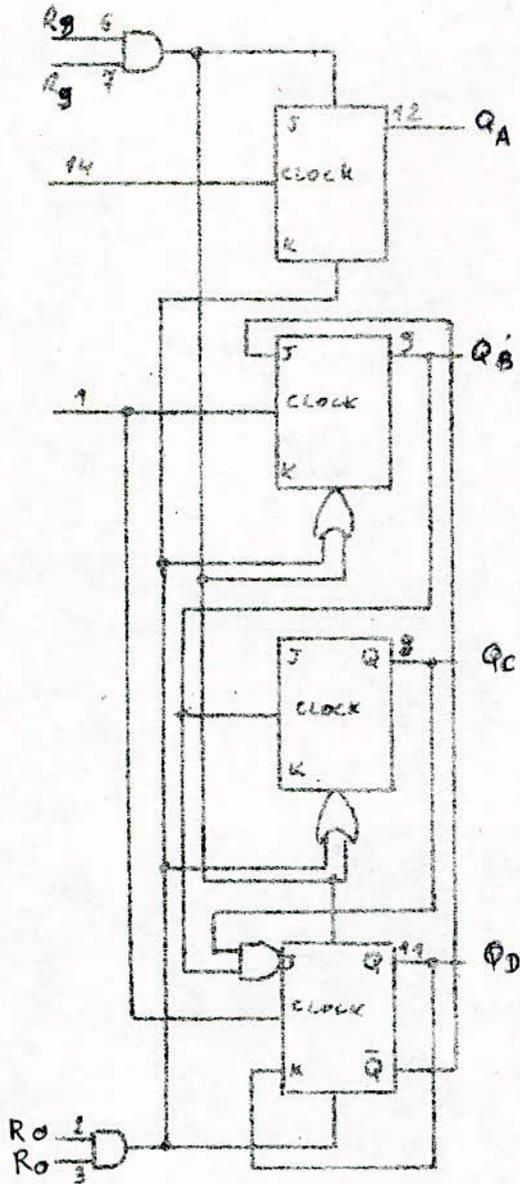
-réinitialiser, sur le front descendant du signal, les compteurs arrivés au T.O. avec remise à zéro des flags d'interruption



SN 74LS90

Table de Verite des entrees RESET

Entrees RESET				Sorties			
R ₀ (1)	R ₀ (2)	R ₉ (1)	R ₉ (2)	Q _D	Q _C	Q _B	Q _A
H	H	B	φ	B	B	B	B
H	H	φ	B	B	B	B	B
φ	φ	H	H	H	B	B	H
φ	B	φ	B	COMPTAGE			
B	φ	B	φ	COMPTAGE			
B	φ	φ	B	COMPTAGE			
φ	B	B	φ	COMPTAGE			



COMPTAGE BCD

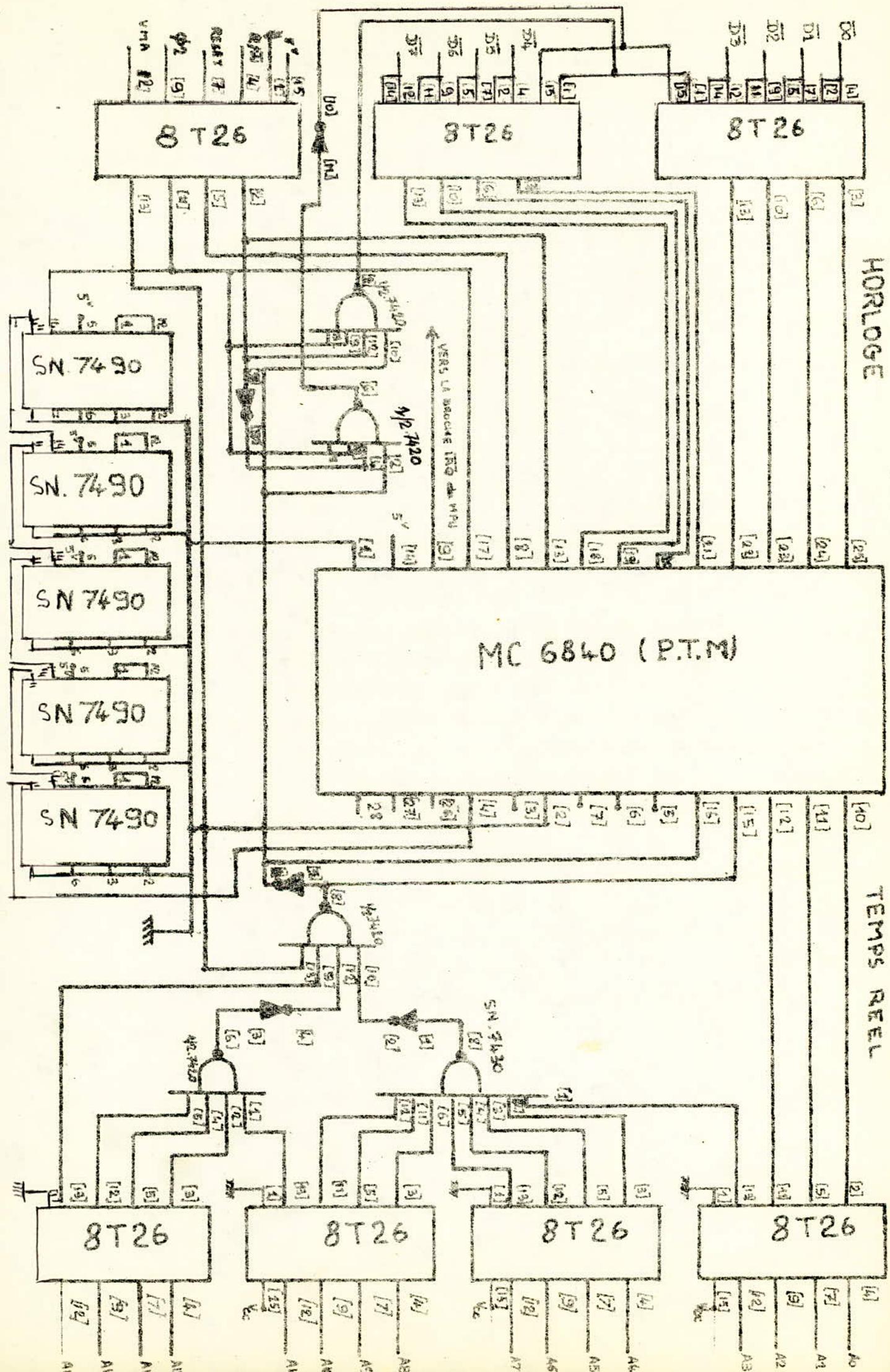
N	SORTIES			
	Q _D	Q _C	Q _B	Q _A
0	B	B	B	B
1	B	B	B	H
2	B	B	H	B
3	B	B	H	H
4	B	H	B	B
5	B	H	B	H
6	B	H	H	B
7	B	H	H	H
8	H	B	B	B
9	H	B	B	H

Comptage BCD : Relier Q_D à entrée B.

H : Niveau Haut

B : Niveau Bas

φ : Indifferent



associés et donc levée du signal d'interruption envoyée au M.P.U. Or pour l'élaboration du calendrier, seule la 1^o conséquence nous est indispensable. La réinitialisation du compteur au T.O. se faisant automatiquement (du moins pour les modes continu et impulsional), il reste donc à remettre à zéro le flag d'interruption associé par un autre moyen que le signal Gate : c'est la séquence RS/RT (lecture registre d'état-lecture compteur). Pour toutes ces raisons l'entrée Gate du Timer utilisé pour le calendrier est mise à la masse.

B) - APPLICATIONS

L'élaboration du calendrier a nécessité la mobilisation d'un seul Timer du P.T.M. Il aura pour horloge, l'horloge interne $\phi 2$ DE 1 MHz, et son entrée Gate sera mise à la masse. Il enverra des interruptions au MPU toutes les millisecondes. La mise à jour des autres paramètres du calendrier (seconde, minute, heure ...) se fait par programme. Sachant que l'adressage des registres de contrôle 1 et 3 dépend du bit "0" du registre de contrôle 2, il est nécessaire de ne pas perturber le fonctionnement du calendrier lors d'une utilisation ultérieure, et à d'autres fins, des deux autres Timers du PTM. Un choix s'est donc posé et l'on a pris le Timer 1. Il contiendra dans ses latches LSB et MSB, pour le comptage d'une milliseconde, le nombre suivant :

soit $(N+1)T=t$ avec : N: nombre chargé dans les latches
 T: période du signal d'horloge
 t: intervalle entre 2 T.O. voisins

$$\text{donc } (N+1) \cdot 10^{-6} = 10^{-3} \rightarrow N=999$$

L'initialisation du Timer 1 consiste à charger ses latches et à positionner son registre de contrôle comme suit :

- valider le comptage (CR10=0)
- Valider l'horloge interne $\phi 2$ (CR11=1)
- utiliser le mode normal 16 bits (CR12=0)
- sélectionner le mode continu (CR13=0=CR15)
- invalider l'initialisation compteur par écriture des latches (CR14=1).
- valider la ligne d'interruption (CR16=1)
- valider la sortie (CR17=1)

Le registre de contrôle 1 aura donc la configuration suivante :

!	1	!	1	!	0	!	1	!	0	!	0	!	1	!	0	!	0	!	
bit	7		6		5		4		3		2		1		0				

Les signaux d'interruption, issus du P.T.M. (pin 9) toutes les millisecondes, sont envoyés sur la ligne NMI du M.P.U. Cette ligne a l'avantage de prendre en considération l'interruption quelque soit l'état du bit de masque Im.

A la réception du signal, le C.P.U. termine l'exécution de l'instruction en cours, sauvegarde dans la pile le contenu de ses registres internes et saute à l'adresse FFFC-FFED. Là, le compteur de programme (P.C.) se charge avec le contenu de ses positions mémoires, préalablement positionnées à l'adresse (adca) du calendrier.

Une séquence de programme calendrier (comptage des millisecondes, secondes, minutes...) se termine toujours par une instruction RTI (retour d'interruption), laquelle permet au C.P.U. de recharger ses registres avec les contenus de la pile et de poursuivre l'exécution de son programme initial, dans l'attente d'une autre interruption.

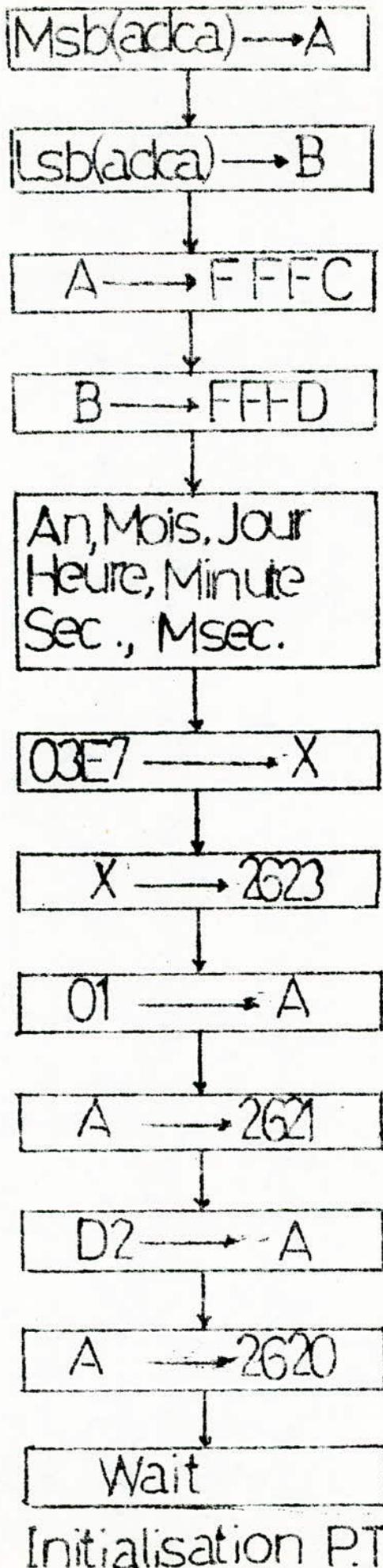
2)-LE CHRONOMETRE

Une seconde application, découlant du calendrier, est les programmes "chronomètres". Ceux-ci permettent d'estimer la durée d'un phénomène à la microseconde près.

Les trois séquences du "chronomètre" sont:

- 1)-début du phénomène: enregistrement de la configuration du temps de départ (heure, minute, seconde, milliseconde, microseconde).
- 2)-fin du phénomène: enregistrement de la configuration du temps à cet instant
- 3)-calcul par l'ordinateur du temps effectif mis par le phénomène, à partir des temps initial et final.

Ces trois programmes seront gérés par le moniteur temps réel, les programmes "départ" et "fin" chronomètre étant affectés d'un même niveau de priorité relativement élevé.



Initialisation P.T.M.

DEROUTEMENT
DE L'ADRESSE NAI
DU N.P.U. VERS L'ADRESSE
DU CALENDRIER -

POSITIONNEMENT DES
PARAMETRES DU TEMPS

charger 999 dans X
stocker l'index
dans latches T≠1

charger 01 dans AccA.
stocker A dans le
registre de controle 2

charger A avec 178
stocker l'accumulateur A
dans le registre controle 1.

Attente d'une
interruption.

INITIALISATION DU PTM

0010	86	01	LDAA	
0012	B7	FFFC	STAA	$\overline{\text{NMI}}_H$
0015	86	00	LDAA	
0017	B7	FFFD	STAA	$\overline{\text{NMI}}_L$
001A	CE	0000	LDX	
001D	FF	0600	STX	Milsec
0020	86	00	LDAA	
0022	B7	0602	STAA	Sec
0025	86	01	LDAA	
0027	B7	0603	STAA	Min
002A	86	01	LDAA	
002C	B7	0604	STAA	H
002F	86	01	LDAA	
0031	B7	0605	STAA	J
0034	86	01	LDAA	
0036	B7	0606	STAA	M
0039	CE	03BC	LDX	
003C	FF	0607	STX	An
003F	CE	03E7	LDX	
0042	FF	2622	STX	Latches Timer 1
0045	86	01	LDAA	
0047	B7	2621	STAA	CR2
004A	86	D2	LDAA	
004C	B7	2620	STAA	CR1
004F	3E		WAI	

$\overline{NMI}=0$

Le calendrier

$RS \rightarrow A$

$I0=1$

$I1=1$

$I2=1$

$T-2 \rightarrow X$

JUMP PS

$T-3 \rightarrow X$

JUMP PS

RTI

$F1 \rightarrow X$

$M_{s+1} \rightarrow M_s$

$M_s=1000$

RTI

$M_s=0$

$Sec+1 \rightarrow Sec$

$Sec=60$

RTI

$Sec=0$

$M_{n+1} \rightarrow M_n$

$M_n=60$

RTI

$M_n=0$

$H+1 \rightarrow H$

$H=24$

RTI

$H=0$

a suivre

RS : registre d'état du PTM

I0 : bit 0 de RS

F1 : compteur du Timer 1

PS : programme special

A : accumulateur A

X : registre d'Index

PROGRAMME DU CALENDRIER.

Remarque:

Chaque instruction du programme est représentée sur six (6) colonnes dont la signification est:

1° colonne: indique les (ou la) locations mémoires où est logée l'instruction (code opération + adresse ou opérande).

2° colonne: contient le code opération de l'instruction.

3° Colonne: renferme l'adresse (pour un adressage étendu) ou l'opérande (adressage immédiat) sur lequel porte l'opération; dans le cas de l'adressage étendu l'adresse sus citée est évidemment celle de l'opérande.

4° colonne: sert de label ou étiquette et permet de repérer les adresses de branchements.

5° colonne: contient le mnémonique du code opération qu'il traduit en clair.

6° colonne: contient l'opérande.

Cependant seules les trois premières colonnes sont indispensables.

Programme

0100	BF	0500		STS	
0103	B6	2621		LDAA	
0106	46			RORA	
0107	25	10		BCS	RO1
0109	46			RORA	
010A	24	06		BCC	RO2
010C	FE	2624		LDX	
010F	7E	0800		JUMP	PS1
0112	46		RO2	RORA	
0113	FE	2626		LDX	
0116	7E	0850		JUMP	PS2
0119	FE	2622	RO1	LDX	
011C	FE	0600		LDX	
011F	08			INX	
0120	FF	0600		STX	
0123	8C	03E8		CPX	
0126	27	03		BEQ	RO3
0128	8D	45		BSR	RO4
012A	3B			RTI	
012B	CE	0000	RO3	LDX	
012E	FF	0600		STX	
0131	B6	0602		LDAA	

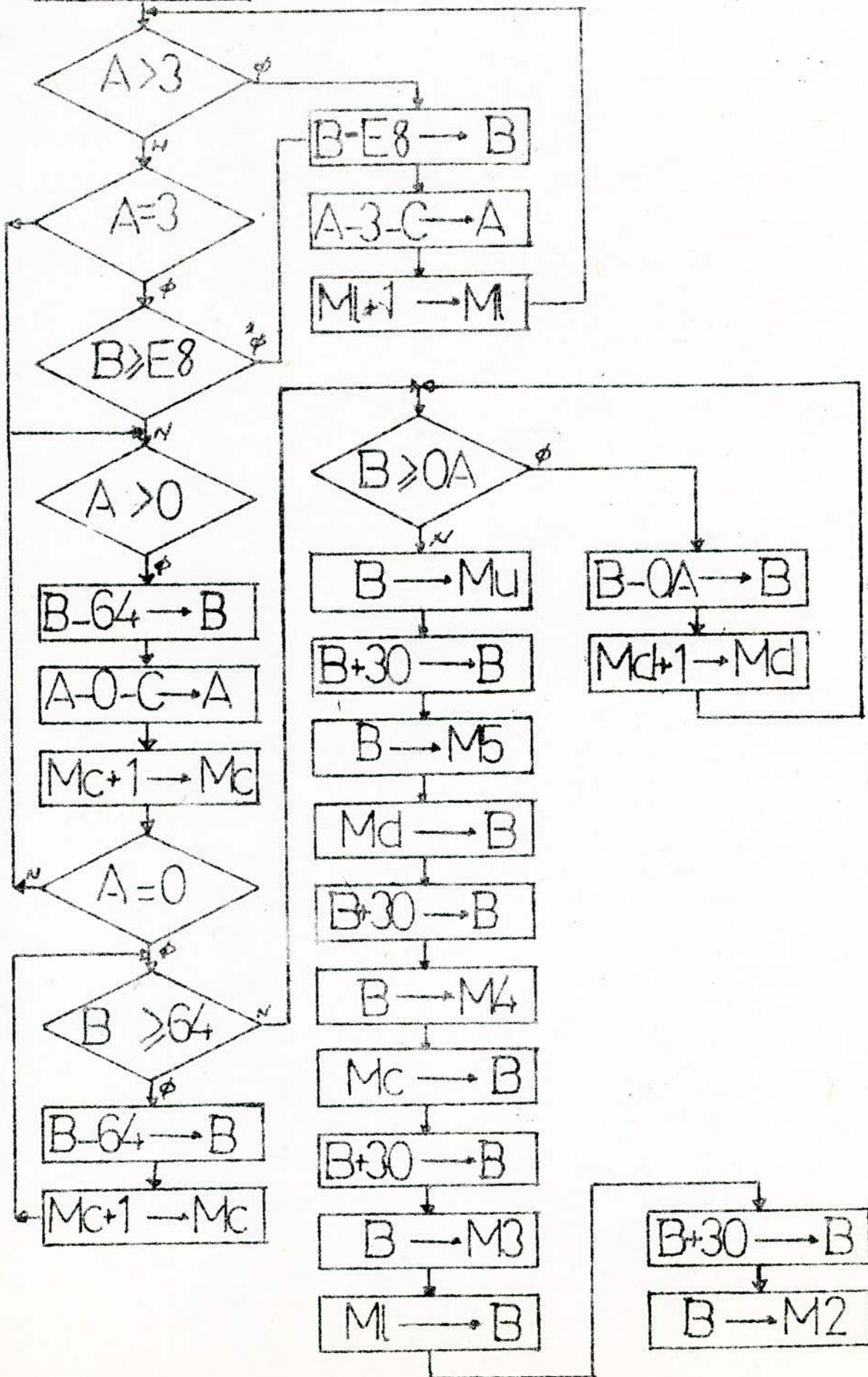
0134	4C			INCA
0135	81	3C		CMPA
0137	27	06		BEQ RO5
0139	B7	0602		STAA
013C	8D	31		BSR RO6
013E	3B			RTI
013F	7F	0602	RO5	CLR
0142	B6	0603		LDAA
0145	4C			INCA
0146	81	3C		CMPA
0148	27	06		BEQ RO7
014A	B7	0603		STAA
014D	8D	20		BSR RO8
014F	3B			RTI
0150	7F	0603	RO7	CLR
0153	B6	0604		LDAA
0156	4C			INCA
0157	81	18		CMPA
0159	27	06		BEQ RO9
015B	B7	0604		STAA
015E	8D	0F		BSR ROA
0160	3B			RTI
0161	7F	0604	RO9	CLR
0164	B6	0605		LDAA
0167	4C			INCA
0168	B7	0605		STAA
016B	8D	02		BSR ROE
016D	20	02		BRA ROG
016F	20	67	RO(4,6,8 A,B)	BRA ROD
0171	B6	0605		LDAA
0174	81	1D		CMPA
0176	2C	01		BGE ROE
0178	3B			RTI
0179	B6	0606	ROE	LDAA
017C	81	02		CMPA
017E	26	1D		BNE ROF
0180	B6	0608		LDAA
0183	44			LSRA
0184	25	0B		BGS R10

0186	44			LSRL	
0187	25	08		BCS	R11
0189	B6	0605		LDAA	
018C	81	1E		CMPA	
018E	27	01		BEQ	R12
0190	3B			RTI	
0191	7C	0606	R1(0,1,2 8,B)	INC	
0194	7F	0604		CLR	
0197	86	01		LDAA	
0199	B7	0605		STAA	
019C	3B			RTI	
019D	81	04	ROF	CMPA	
019F	27	0C		BEQ	R13
01A1	81	06		CMPA	
01A3	27	08		BEQ	R14
01A5	81	09		CMPA	
01A7	27	04		BEQ	R15
01A9	81	0B		CMPA	
01AB	26	0A		BNE	R16
01AD	F6	0605	R1(3,4,5)	LDAB	
01B0	C1	1F		CMPB	
01B2	27	01		BEQ	R17
01B4	3B			RTI	
01B5	20	0A DA	R17	BRA	R18
01B7	F6	0605	R16	LDAB	
01BA	C1	20		CMPB	
01BC	27	01		BEQ	R19
01BE	3B			RTI	
01BF	81	0C	R19	CMPA	
01C1	27	02		BEQ	R1A
01C3	20	CC		BRA	R1B
01C5	FE	0607	R1A	LDX	
01C8	08			INX	
01C9	FF	0607		STX	
01CC	C6	01		LDAB	
01CE	F7	0606		STAB	
01D1	F7	0605		STAB	
01D4	7F	0604		CLR	
01D7	3B			RTI	
01D8	20	26		BRA	R1C

Conversion binaire-ascii

$M0 \rightarrow A$ $M0$: octet msb du mot

$M1 \rightarrow B$ $M1$: octet lsb



Programme de conversion binaire-ascii

Ce programme se compose, en fait, de 3 parties:

- La 1^o, allant de l'adresse 0200 à 0249, consiste à transférer le contenu des mémoires du calendrier dans d'autres locations mémoires où tous les paramètres seront sur 2 octets.

Exemple: les secondes, initialement codées sur un octet, seront après transfert et avant conversion inscrites sur deux octets, l'octet MSB ayant toujours un contenu égal à zéro.

Pour éviter cette partie du programme, il suffit de coder, dans le programme initial du calendrier, tous les paramètres sur deux octets dont certains auront toujours un contenu nul.

- La 2^o partie, de l'adresse 0240 à 0287, transforme le contenu binaire des mémoires précédentes en son équivalent décimal.

- La 3^o partie enfin, de l'adresse 028A à 02F8, convertit en code ASCII LES DIGITS décimaux précédents.

Programme

0200	CE	0666		LDX	
0203	FF	060E		STX	SVE1
0206	CE	0600		LDX	MS
0209	A6	00		LDAA	
020B	E6	01		LDAB	
020D	A7	64		STAA	
020F	E7	65		STAB	
0211	CE	0602		LDX	SEC
0214	86	00	ROL	LDAA	
0216	E6	00		LDAB	
0218	FF	0610		STX	SVE2
021B	FE	060E		LDX	SVE1
021E	A7	00		STAA	
0220	08			INX	
0221	E7	00		STAB	
0223	08			INX	
0224	FF	060E		STX	SVE1
0227	FE	0610		LDX	SVE2
022A	08			INX	
022B	8C	0607		CPX	MSB(AN)
022E	26	E4		BNE	ROL
0230	A6	00		LDAA	
0232	E6	01		LDAB	

0234	FE	060E		LDX	SVE1
0237	A7	00		STAA	
0239	08			INX	
023A	E7	00		STAB	
023C	08			INX	
023D	FF	060E		STX	SVE1
0240	CE	0664		LDX	
0243	A6	00	ROE	LDAA	
0245	E6	01		LDAB	
0247	08			INX	
0248	08			INX	
0249	FF	0610		STX	SVE2
024C	81	03	RO3	CMPA	
024E	2F	09		BLS	RO2
0250	00	E8	RO5	SUBB	
0252	82	03		SBCA	
0254	7C	060A		INC	ML
0257	20	F3		BRA	RO3
0259	81	03	RO2	CMPA	
025B	26	04		BNE	RO4
025E	C1	E8		CMPB	
0260	2C	EF		BGE	RO5
0262	81	00	RO(4,7)	CMPA	
0264	2F	07		BLS	RO6
0266	00	64		SUBB	
0268	82	00		SBCA	
026A	7C	060B		INC	MC
026D	81	00	RO6	CMPA	
026F	26	F1		BNE	RO7
0271	C1	64	RO9	CMPB	
0273	2B	07		BMI	RO8
0275	00	64		SUBB	
0277	7C	060B		INC	MC
027A	20	F5		BRA	RO9
027C	C1	0A	RO(8,B)	CMPB	
027E	2B	07		BMI	ROA
0280	00	0A		SUBB	
0282	7C	060C		INC	MD
0285	20	F5		BRA	ROB
0287	F7	060D	ROA	STAB	MU

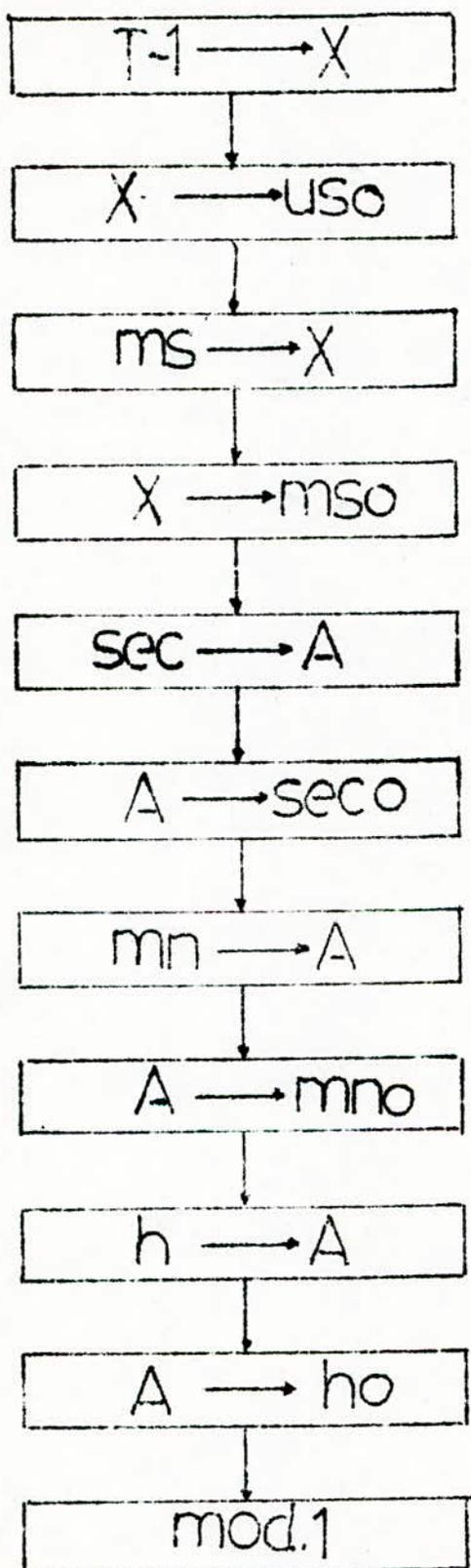
028A	FE	060E		LDX	SVEL
028D	8C	0672		CPX	MS
0290	27	28		BEQ	ROC
0292	8C	0686		CPX	AN
0295	27	23		BEQ	ROD
0297	F6	060C		LDAB	MD
029A	CB	30		ADDB	
029C	E7	00		STAB	
029E	08			INX	
029F	F6	060D		LDAB	MU
02A2	CB	30		ADDB	
02A4	E7	00		STAB	
02A6	08			INX	
02A7	86	20		LDAA	
02A9	A7	00		STAA	
02AB	08			INX	
02AC	FF	060E		STX	SVEL
02AF	7F	060C		CLR	MD
02B2	7F	060D		CLR	MU
02B5	FE	0610		LDX	SVE2
02B8	20	8A	ROO	BRA	ROE
02BA	F6	060A	RO(C,D)	LDAB	ML
02BD	CB	30		ADDB	
02BF	E7	00		STAB	
02C1	08			INX	
02C2	F6	060B		LDAB	MC
02C5	CB	30		ADDB	
02C7	E7	00		STAB	
02C9	08			INX	
02CA	F6	060C		LDAB	MD
02CD	CB	30		ADDB	
02CF	E7	00		STAB	
02D1	08			INX	
02D2	F6	060D		LDAB	MU
02D5	CB	30		ADDB	
02D7	E7	00		STAB	
02D9	08			INX	
02DA	86	20		LDAA	
02DC	A7	00		STAA	
02DE	08			INX	

02DF	FF	060E	STX	SVE1
02E2	7F	060A	CLR	ML
02E5	7F	060B	CLR	MC
02E8	7F	060C	CLR	MD
02EB	7F	060D	CLR	MU
02EE	FE	0610	LDX	SVE2
02F1	8C	0672	CPX	MS
02F4	26	C2	BNE	ROO
02F6	86	04	LDAA	
02F8	B7	068A	STAA	

Ici s'arrete le programme de conversion, que l'on fera suivre du programme d'affichage sur l'unité de visualisation.

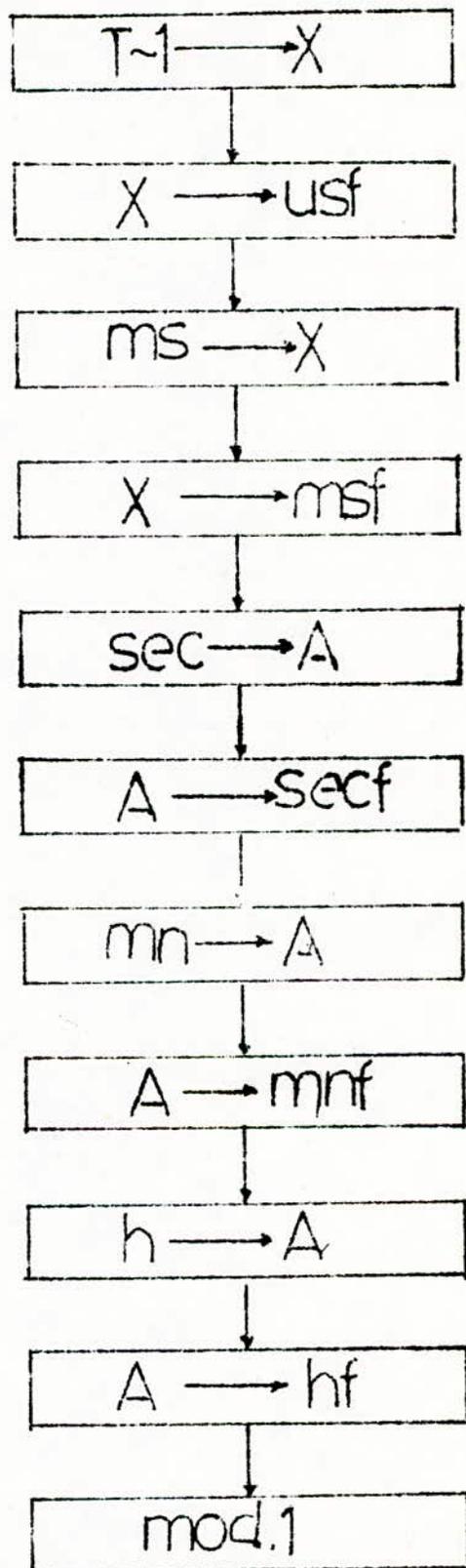
02FB	CE	0672	LDX	MS
02FE	BD	FA14	JSR	
0301	39		RTS	

Chronometre:



debut

et



fin

du phenomene

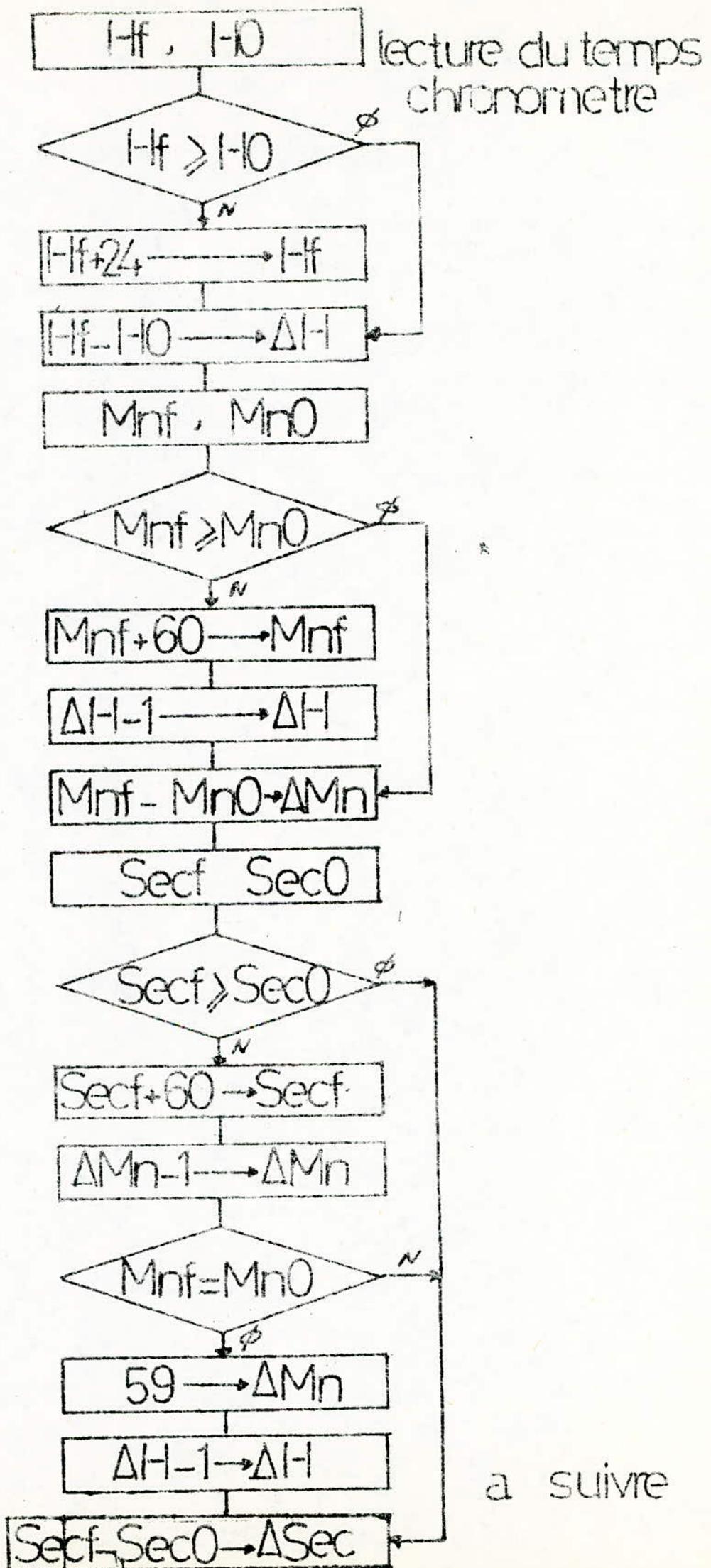
PROGRAMMES CHRONOMETRES

DEBUT CHRONOMETRE

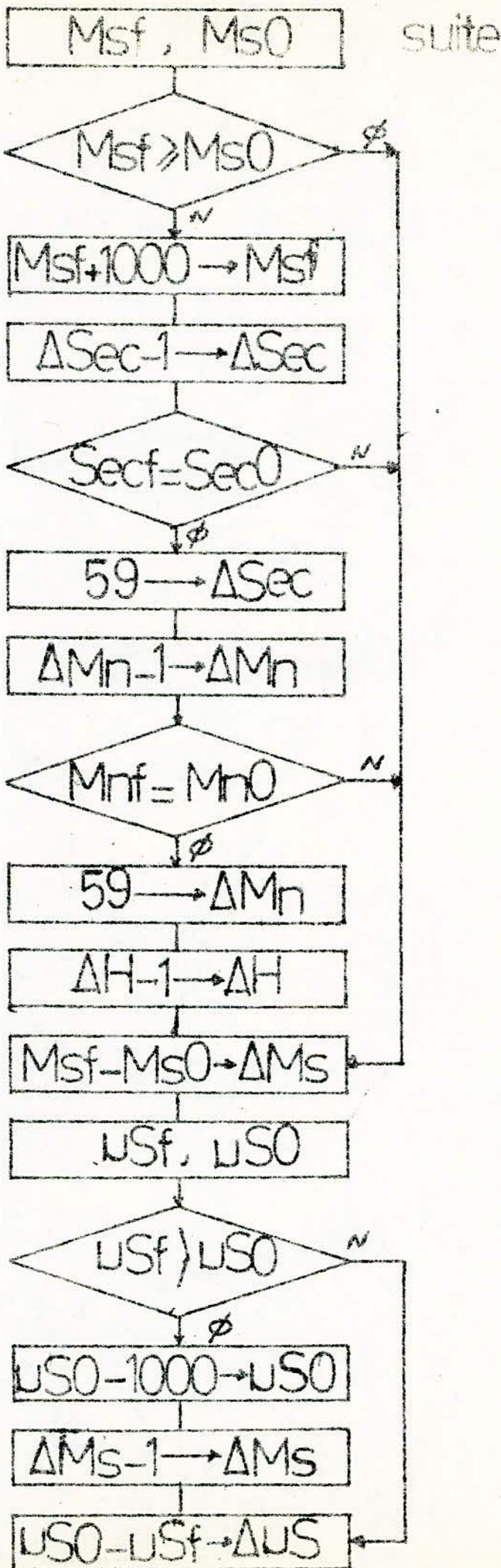
0100	FE	2622	LDX	Timer 1
0103	FF	0200	STX	Micsecf
0106	FE	0600	LDX	Milsec
0109	FF	0202	STX	Milsecf
010C	B6	0602	LDAA	Sec
010F	B7	0205	STAA	Secf
0112	B6	0603	LDAA	Min
0115	B7	0204	STAA	Minf
0118	B6	0604	LDAA	H
011B	B7	0206	STAA	Hf
011E	7E	0050	JUMP	MOD.1

FIN CHRONOMETRE

0121	FE	2622	LDX	Timer 1
0124	FF	0207	STX	Micseco
0127	FE	0600	LDX	Milsec
012A	FF	0209	STX	Milseco
012D	B6	0602	LDAA	Sec
0130	B7	020C	STAA	Seco
0133	B6	0603	LDAA	Min
0136	B7	020B	STAA	Mino
0139	B6	0604	LDAA	H
013C	B7	020D	STAA	Ho
013E	7E	0050	JUMP	MOD.1



a suivre



CHRONOMETRAGE D'UN PHENOMENE

Le calcul de la durée d'un phénomène se fait à partir des relevés des temps initial et final. Le résultat du calcul est donné à la microseconde près.

Programme:

0030	B6	0011		LDAA	Hf
0033	B1	0010		CMPA	Ho
0036	2C	02		BGE	RO1
0038	8B	18		ADDA	
003A	B0	0010	RO1	SUBA	Ho
003D	B7	001E		STAA	DH
0040	B6	0013		LDAA	Mnf
0043	B1	0012		CMPA	Mno
0046	2C	05		BGE	RO2
0048	8B	3C		ADDA	
004A	7A	001E		DEC	DH
004D	B0	0012	RO2	SUBA	Mno
0050	B7	001F		STAA	DMn
0053	B6	0015		LDAA	Secf
0056	B1	0014		CMPA	Seco
0059	2C	15		BGE	RO3
005B	8B	3C		ADDA	
005D	7A	001F		DEC	DMn
0060	F6	0013		LDAB	Mnf
0063	F1	0012		CMPB	Mno
0066	26	08		BNE	RO4
0068	C6	3B		LDAB	
006A	F7	001F		STAB	DMn
006D	7A	001E		DEC	DH
0070	B0	0014	RO(3,4)	SUBA	Seco
0073	B7	0020		STAA	DSec
0076	B6	0018		LDAA	Msb(Msf)
0079	B1	0016		CMPA	Msb(Mso)
007C	2E	3A		BGT	RO5
007E	26	08		BNE	RO6
0080	F6	0019		LDAB	Lsb(Msf)
0083	F1	0017		CMPB	Lsb(Mso)
0086	2C	30		BGE	RO7
0088	F6	0019	RO6	LDAB	Lsb(Msf)

008B	CB	E8		ADDB	
008D	89	03		ADCA	
008F	7A	0020		DEC	DSec
0092	B7	0018		STAA	Msb(Msf)
0095	F7	0019		STAB	Lsb(Msf)
0098	B6	0015		LDAA	Secf
009B	B6	0014		CMPA	Seco
009E	26	<u>18</u>		BNE	RO8
00A0	86	3B		LDAA	
00A2	B7	0020		STAA	DSec
00A5	7A	001F		DEC	DMn
00A8	B6	0013		LDAA	Mnf
00AB	B1	0012		CMPA	Mno
00AE	26	08		BNE	RO9
00B0	06	3B		LDAB	
00B2	F7	001F		STAB	DMn
00B5	7A	001E		DEC	DH
00B8	B6	0019	RO(5,7,8,9)	LDAA	Lsb(Msf)
00BB	B0	0017		SUBA	Lsb(Mso)
00BE	F6	0018		LDAB	Msb(Msf)
00C1	F2	0016		SBCB	Msb(Mso)
00C4	B7	0022		STAA	Lsb(DMs)
00C7	F7	0021		STAB	Msb(Msf)
00CA	B6	001C		LDAA	Msb(uSf)
00CD	B1	001A		CMPA	Msb(uSo)
00D0	2E	A		BGT	ROA
00D2	26	1F		BNE	ROB
00D4	F6	001D		LDAB	Lsb(uSf)
00D7	F1	001B		CMPB	Lsb(uSo)
00DA	2F	17		BLE	ROC
00DC	B6	001B	ROA	LDAA	Lsb(uSo)
00DF	8B	E8		ADDA	
00E1	F6	001A		LDAB	Msb(uSo)
00E4	C9	03		ADCB	
00E7	B7	001B		STAA	Lsb(uSo)
00EA	F7	001A		STAB	Msb(uSo)
00ED	FE	0021		LDX	Msb(DMs)
00EF	09			DEX	
00FC	FF	0021		STX	Msb(DMs)
00F3	B6	001B	RO(B,C)	LDAA	Lsb(uSo)

00F6	B0	001D	SUBA	Lsb(uSo)
00F9	F6	001A	LDAB	Msb(uSo)
00FC	F2	001C	SBCB	Msb(uSf)
00FF	B7	0024	STAA	Lsb(DU DuS)
0102	F7	0023	STAB	Msb(DuS)
0105	7E	0050	JUMP	MOD.i

CHAPITRE V : APPLICATIONS DE L'ORDINATEUR

TEMPS REEL

Après avoir introduit la notion de temps réel et étudié les deux éléments fondamentaux (Moniteur et horloge temps réel) d'un système temps réel, nous aborderons dans ce chapitre quelques applications.

1°)-Application industrielle: le contrôle de processus.

La surveillance d'une unité industrielle, qui se faisait et se fait encore à l'aide d'instruments analogiques, a été révolutionnée par l'introduction des ordinateurs. Particulièrement ceux de la troisième génération qui ont permis à l'informatique temps réel de s'imposer dans ce domaine.

Le contrôle industriel, quant à lui, comporte quatre stades:

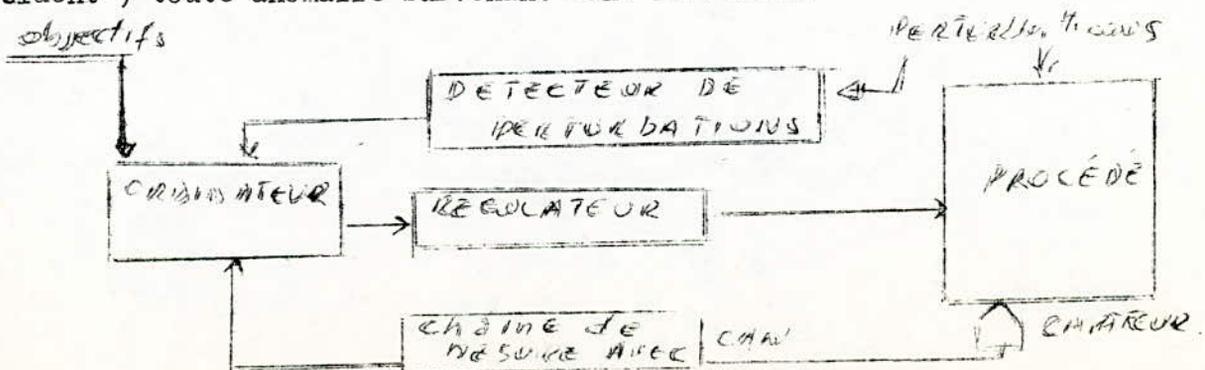
- a-la surveillance centralisée
- b-la conduite centralisée
- c-la régulation centralisée
- d-la commande dynamique optimale

L'étude détaillée sera faite sur les deux stades les plus favorables à l'utilisation de système temps réel: la conduite centralisée et la commande optimale.

1-1)La conduite centralisée

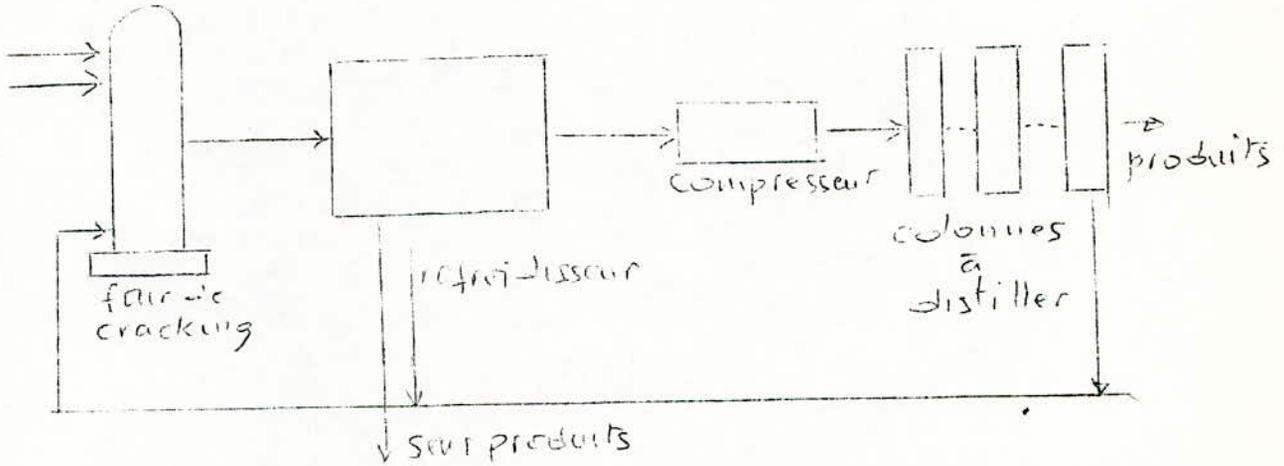
Le contrôle d'un processus de production industrielle implique de nos jours l'acquisition d'un très grand nombre de mesures; lesquelles sont prises par des appareils de type analogique. Il faut donc prévoir une conversion numérique-analogique pour que l'ordinateur puisse utiliser ces données. Le traitement des données terminé il faut encore convertir les résultats du numérique à l'analogique avant de pouvoir les appliquer aux appareils de commande ou de contrôle.

Outre les fonctions de centralisations des données et de leur traitement l'ordinateur aura pour tâche particulière de signaler très précisément (en inscrivant, par exemple, les moment et lieu de l'incident) toute anomalie survenant dans la chaîne.



1-2) La commande dynamique optimale

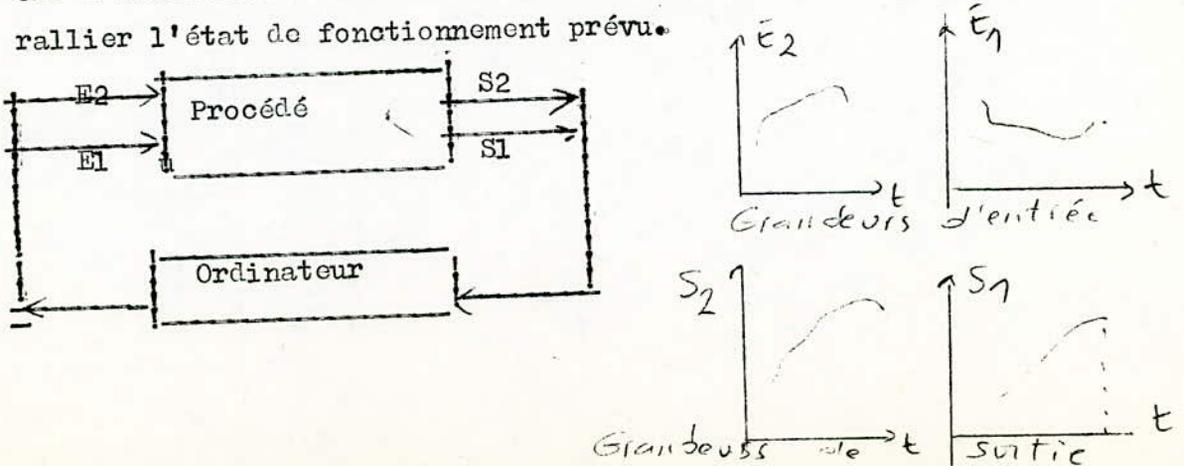
C'est le stade le plus élevé du contrôle industriel. Un exemple typique est donnée par le processus de fabrication de l'éthylène, processus qui compte jusqu'à 1500 paramètres de mesures (température, pressions volumes, débits...)



Dans un contexte figé, au cours du fonctionnement continu et stable, la valeur optimale des diverses grandeurs de sortie a été déterminée une fois pour toute. Cependant, lors de la phase de démarrage, il faut atteindre rapidement le meilleur régime. De plus, au cours du temps, diverses raisons imposent la modification du régime de fonctionnement. Il faut donc pouvoir commander l'action sur le procédé afin d'obtenir l'évolution souhaitée de la production.

La commande optimale consiste donc à faire évoluer les consignes de commande de façon à optimiser à chaque instant le rendement de la production, soit à minimiser le coût du produit.

Il est donc indispensable de connaître à chaque instant la valeur la meilleure à donner aux grandeurs de sortie (comme dans le cas de la conduite centralisée) mais encore il faut savoir déterminer quelle est l'évolution la meilleure qu'il faut leur faire suivre pour rallier l'état de fonctionnement prévu.



To

2°)-Application dans le domaine de la gestion: banque des données

Dans la conception classique de la gestion, l'ordinateur résoud les problèmes de gestion indépendamment les uns des autres et en considérant les données figées au cours du temps.

Actuellement, le système de traitement est constitué telle une banque de données c-a-d que l'ensemble des connaissances nécessaires à la totalité d'un traitement à effectuer sera stocké, disponible à tout instant et remis à jour constamment ou régulièrement .

Cette centralisation des données, adaptée au traitement et même au multitraitement temps réel, permet l'interrogation à distance par des périphériques d'un ordinateur puissant répartis dans l'espace.

Autres exemples:

-Réservation de billets dans les compagnies aériennes

Le système temps réel permet de connaître en quelques instants la disponibilité des places, pour un vol donné. Les agences de voyage, répartis dans une ville, sont reliés à l'ordinateur au moyen de lignes téléphoniques. Le dialogue, entre l'opérateur et l'ordinateur, s'effectue grâce à des claviers reliés à des consoles de visualisation.

-Utilisation du système temps réel dans les banques, compagnies d'assurances, polices.

Le système temps réel est également utilisée dans les banques pour connaître l'état d'un compte bancaire et y faire les opérations nécessaires.

La police peut également utiliser le système temps réel pour l'élaboration et la mise à jour de son fichier.

CONCLUSION

Le moniteur et l'horloge temps réel donnent à tout mini ou micro-ordinateur le pouvoir de contrôler toute situation en temps réel. Bien que de mise au point délicate, l'utilisation du moniteur est fort avantageuse, ne serait-ce que dans l'optimisation de l'occupation de l'unité centrale.

L'emploi de systèmes micro-informatiques temps réel tend à se généraliser. Ainsi des domaines, aussi divers que la médecine, le nucléaire, l'aéronautique... et bien sûr "le grand public" utilisent de tels systèmes.

Toutefois, leur rentabilité dépend du degré de complexité des phénomènes supervisés.

ESSAI DU CALENDRIER

Remarque

La variation des secondes aura un pas de 2 ou 3 unités du au fait que l'imprimante ne peut écrire que 10 caractères à la seconde et que notre affichage en comporte 25 *

ms	s	m	h	j	m	an
0000	28	58	01	01	01	1980
0000	31	58	01	01	01	1980
0000	34	58	01	01	01	1980
0000	37	58	01	01	01	1980
0000	39	58	01	01	01	1980
0000	42	58	01	01	01	1980
0000	45	58	01	01	01	1980
0000	48	58	01	01	01	1980
0000	51	58	01	01	01	1980
0000	54	58	01	01	01	1980
0000	56	58	01	01	01	1980
0000	59	58	01	01	01	1980
0000	02	00	01	01	01	1980
0000	05	59	01	01	01	1980
0000	08	59	01	01	01	1980
0000	10	59	01	01	01	1980
0000	13	59	01	01	01	1980
0000	16	59	01	01	01	1980
0000	19	59	01	01	01	1980
0000	22	59	01	01	01	1980
0000	24	59	01	01	01	1980
0000	27	59	01	01	01	1980
0000	30	59	01	01	01	1980
0000	33	59	01	01	01	1980
0000	36	59	01	01	01	1980
0000	38	59	01	01	01	1980
0000	41	59	01	01	01	1980
0000	44	59	01	01	01	1980
0000	47	59	01	01	01	1980
0000	50	59	01	0 ₁	01	1980
0000	52	59	01	01	01	1980

ACCUMULATOR AND MEMORY		ADDRESSING MODES															BOOLEAN/ARITHMETIC	COND. CODE REG						
		IMMED			DIRECT			INDEX			EXTND			INTHER			OPERATION	S	4	3	2	1	0	
		OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	(All register labels refer to contents)	H	I	N	Z	V	C	
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3			A + M → A	†	•	†	†	†	†		
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3			B + M → B	†	•	†	†	†	†		
Add Acmltrs	ABA													1B	2	1	A + B → A	†	•	†	†	†	†	
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3			A + M + C → A	†	•	†	†	†	†		
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3			B + M + C → B	†	•	†	†	†	†		
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3			A · M → A	•	•	†	†	R	•		
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3			B · M → B	•	•	†	†	R	•		
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3			A · M	•	•	†	†	R	•		
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3			B · M	•	•	†	†	R	•		
Clear	CLR							6F	7	2	7F	6	3			00 → M	•	•	R	S	R	R	R	
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3			A - M	•	•	†	†	†	†	†	
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3			B - M	•	•	†	†	†	†	†	
Compare Acmltrs	CBA													11	2	1	A - B	•	•	†	†	†	†	†
Complement, 1's	COM							63	7	2	73	6	3			M → M	•	•	†	†	R	S	•	
	COMA													43	2	1	A → A	•	•	†	†	R	S	•
	COMB													53	2	1	B → B	•	•	†	†	R	S	•
Complement, 2's (Negate)	NEG							60	7	2	70	6	3			00 - M → M	•	•	†	†	①	②	•	
	NEGA													40	2	1	00 - A → A	•	•	†	†	①	②	•
	NEGB													50	2	1	00 - B → B	•	•	†	†	①	②	•
Decimal Adjust, A	DAA												19	2	1	Converts Binary Add. of BCD Characters into BCD Format	•	•	†	†	†	③	•	
Decrement	DEC							6A	7	2	7A	6	3			M - 1 → M	•	•	†	†	④	•	•	
	DECA													4A	2	1	A - 1 → A	•	•	†	†	④	•	
	DECB													5A	2	1	B - 1 → B	•	•	†	†	④	•	
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3			A ⊕ M → A	•	•	†	†	R	•	•	
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3			B ⊕ M → B	•	•	†	†	R	•	•	
Increment	INC							6C	7	2	7C	6	3			M + 1 → M	•	•	†	†	⑤	•	•	
	INCA													4C	2	1	A + 1 → A	•	•	†	†	⑤	•	
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3			M → A	•	•	†	†	R	•	•	
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3			M → B	•	•	†	†	R	•	•	
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	8A	4	3			A + M → A	•	•	†	†	R	•	•	
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3			B + M → B	•	•	†	†	R	•	•	
Push Data	PSHA													36	4	1	A → M _{SP} , SP - 1 → SP	•	•	•	•	•	•	•
	PSHB													37	4	1	B → M _{SP} , SP - 1 → SP	•	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M _{SP} → A	•	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M _{SP} → B	•	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3			M	•	•	†	†	⑥	†	•	
	ROLA													49	2	1	A	•	•	†	†	⑥	†	•
	ROLB													59	2	1	B	•	•	†	†	⑥	†	•
Rotate Right	ROR							66	7	2	76	6	3			M	•	•	†	†	⑥	†	•	
	RORA													46	2	1	A	•	•	†	†	⑥	†	•
	RORB													56	2	1	B	•	•	†	†	⑥	†	•
Shift Left, Arithmetic	ASL							68	7	2	78	6	3			M	•	•	†	†	⑥	†	•	
	ASLA													48	2	1	A	•	•	†	†	⑥	†	•
Shift Right, Arithmetic	ASLB													58	2	1	B	•	•	†	†	⑥	†	•
	ASR							67	7	2	77	6	3			M	•	•	†	†	⑥	†	•	
Shift Right, Logic	ASRA													47	2	1	A	•	•	†	†	⑥	†	•
	ASRB													57	2	1	B	•	•	†	†	⑥	†	•
Store Acmltr	LSR							64	7	2	74	6	3			M	•	•	†	†	⑥	†	•	
	LSRA													44	2	1	A	•	•	†	†	⑥	†	•
	LSRB													54	2	1	B	•	•	†	†	⑥	†	•
Subtract	STAA				97	4	2	A7	6	2	B7	5	3			A → M	•	•	†	†	R	•	•	
	STAB				D7	4	2	E7	6	2	F7	5	3			B → M	•	•	†	†	R	•	•	
Subtract Acmltrs	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3			A - M → A	•	•	†	†	†	†	†	
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3			B - M → B	•	•	†	†	†	†	†	
Subtr. with Carry	SBA													10	2	1	A - B → A	•	•	†	†	†	†	†
	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3			A - M - C → A	•	•	†	†	†	†	†	
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3			B - M - C → B	•	•	†	†	†	†	†	
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	†	†	R	•	•
	TBA													17	2	1	B → A	•	•	†	†	R	•	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3			M - 00	•	•	†	†	R	R	•	
	TSTA													4D	2	1	A - 00	•	•	†	†	R	R	•
	TSTB													5D	2	1	B - 00	•	•	†	†	R	R	•

INDEX REGISTER AND STACK		IMMED						DIRECT						INDEX						EXTND						INHER						BOOLEAN/ARITHMETIC OPERATION							
POINTER OPERATIONS		MNEMONIC		OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	5	4	3	2	1	0
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3							09	4	1																	
Decrement Index Reg	DEX																																						
Decrement Stack Pntr	DES																			34	4	1																	
Increment Index Reg	INX																			08	4	1																	
Increment Stack Pntr	INS																			31	4	1																	
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3																										
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3																										
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3																										
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3																										
Indx Reg → Stack Pntr	TXS																			35	4	1																	
Stack Pntr → Indx Reg	TSX																			30	4	1																	

JUMP AND BRANCH OPERATIONS		MNEMONIC		RELATIVE		INDEX		EXTND		INHER		BRANCH TEST							
		OP	~	≠	OP	~	≠	OP	~	≠	OP	~	≠	5	4	3	2	1	0
Branch Always	BRA	20	4	2															
Branch If Carry Clear	BCC	24	4	2															
Branch If Carry Set	BCS	25	4	2															
Branch If = Zero	BEQ	27	4	2															
Branch If ≥ Zero	BGE	2C	4	2															
Branch If > Zero	BGT	2E	4	2															
Branch If Higher	BHI	22	4	2															
Branch If ≤ Zero	BLE	2F	4	2															
Branch If Lower Or Same	BLS	23	4	2															
Branch If < Zero	BLT	2D	4	2															
Branch If Minus	BMI	2B	4	2															
Branch If Not Equal Zero	BNE	26	4	2															
Branch If Overflow Clear	BVC	28	4	2															
Branch If Overflow Set	BVS	29	4	2															
Branch If Plus	BPL	2A	4	2															
Branch To Subroutine	BSR	8D	8	2															
Jump	JMP				6E	4	2	7E	3	3									
Jump To Subroutine	JSR				AD	8	2	BD	9	3									
No Operation	NOP										01	2	1						
Return From interrupt	RTI										3B	10	1						
Return From Subroutine	RTS										39	5	1						
Software Interrupt	SWI										3F	12	1						
Wait for Interrupt	WAI										3E	9	1						

CONDITIONS CODE REGISTER		INHER		BOOLEAN		5 4 3 2 1 0					CONDITION CODE REGISTER NOTES					
OPERATIONS		OP	~	≠	OPERATION	H	I	N	Z	V	C	(Bit set if test is true and cleared other wise)				
Clear Carry	CLC	0C	2	1	0 → C						R	① (Bit V) Test Result = 10000000 ?				
Clear Interrupt Mask	CLI	0E	2	1	0 → I		R					② (Bit C) Test Result = 00000000 ?				
Clear Overflow	CLV	0A	2	1	0 → V					R		③ (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared it previously set)				
Set Carry	SEC	0D	2	1	1 → C						S					
Set Interrupt Mask	SEI	0F	2	1	1 → I		S									
Set Overflow	SEV	0B	2	1	1 → V						S	④ (Bit V) Test Operand = 10000000 prior to execution ?				
Acmltr A → CCR	TAP	06	2	1	A → CCR						⑫	⑤ (Bit V) Test Operand = 01111111 prior to execution ?				
CCR → Acmltr A	TPA	07	2	1	CCR → A							⑥ (BIT V) Test Set equal to result of N ⊕ C after shift has occurred				

- LEGEND**
- OP Operation Code (Hexadecimal)
 - ~ Number of MPU Cycles
 - ≠ Number of Program Bytes
 - + Arithmetic Plus
 - Arithmetic Minus
 - Boolean AND
 - M_{SP} Contents of memory location pointed to be Stack Pointer
 - + Boolean Inclusive OR
 - ⊕ Boolean Exclusive OR
 - M Complement of M
 - Transfer Into
 - 0 Bit = Zero
 - 00 Byte = Zero
 - H Half carry from bit 3
 - I Interrupt mask
 - N Negative (sign bit)
 - Z Zero (byte)
 - V Overflow, 2's complement
 - C Carry from bit 7
 - R Reset Always
 - S Set Always
 - ‡ Test and set if true cleared otherwise
 - Not Affected
 - CCR Condition Code Register
 - LS Least Significant
 - MS Most Significant
- ⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
- ⑧ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?
- ⑨ (Bit N) Test Result less (han zero) ? (Bit 15 = 1)
- ⑩ (ALL) Load Condition Code Register from Stack (See Special Operations)
- ⑪ (Bit I) Set when interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.
- ⑫ (ALL) Set according to the contents of Accumulator A

BIBLIOGRAPHIE:

Livres:

- Microprocesseurs et microordinateurs.
par R.LYON-CAEN et J-M.CROZET.
(ED. Masson.)
- Programmation des microprocesseurs.
par H. LILEN
(ED.Radio.)
- Informatique en temps réel
par R.JOUMARD.
(ED. Dunod.)

REVUES ET BROCHURES SPECIALISEES :

- MC6800 MICROPROCESSOR APPLICATIONS MANUEL
par MOTOROLA INCORPORATION.
- 6. A scheduler for real-time task control in microcomputer
par EDWARD A.PARRISH ET VICTOR K.L.HUANG
(IEEE. VOL IECI-25? N°1, FEB.1978)
- MICROS ET MONITEURS TEMPS REEL
par H.HANNE
(Minis et micros N°63,65,66)
- MC6840 . Advance Information
(MOTOROLA semiconductors.)

