

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ALGER  
DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

4/82

2 ex

FILIERE D'INGENIEUR EN



**PROJET DE FIN D'ETUDES**

Sujet : **ETUDE D'UN GESTIONNAIRE  
DE TACHES D'UN PROCESSUS  
SEMI-INDUSTRIEL**

Proposé par : Mme R.RAMDANI  
Mme L.BENALLEGUE

Réalisé par : A.AIT ARKOUB

JANVIER 82

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ALGER  
DEPARTEMENT D'ELECTRONIQUE ET D'ELECTROTECHNIQUE

**FILIERE D'INGENIEUR EN  
ELECTRONIQUE**

**PROJET DE FIN D'ETUDES**

Sujet : **ETUDE D'UN GESTIONNAIRE  
DE TACHES D'UN PROCESSUS  
SEMI-INDUSTRIEL**

Proposé par : Mme R.RAMDANI  
Mme L.BENALLEGUE

Réalisé par : A.AIT ARKOUR

JANVIER 82

## REMERCIEMENTS

Le présent travail a été réalisé au sein de la Division V du CSTN .

Je remercie Monsieur H. TEDJINI pour m'avoir admis dans sa division et pour ses conseils tout au long de cette étude.

Je remercie Madame R. RAMDANI et Madame L. BENALLEGUE pour m'avoir encadré et suivi durant tout le semestre.

Je remercie toutes les personnes qui m'ont aidé lors de la réalisation de cette étude, en particulier Monsieur LAZIB et Monsieur ABDI pour le tirage, Monsieur HALIMI pour l'attention qu'il m'a prêtée.

Je remercie enfin tous mes amis et collègues.

## SOMMAIRE

### INTRODUCTION

#### Chapitre 1 : RAPPELS SUR L'UNITE CENTRALE MC6800

- 1- Caractéristiques du MC6800
- 2- Différents registres adressables
- 3- Différents modes d'adressage
- 4- Brochage du 6800 et signification des différentes lignes

#### Chapitre 2 : CONTROLE DE PROCESSUS EN TEMPS REEL

- 1- Processus temps réel
- 2- Le moniteur

#### Chapitre 3 : LES DIFFERENTES TACHES DU PROCESSUS

- 1- Les différentes tâches du processus
- 2- Tables du gestionnaire
- 3- Le gestionnaire

#### Chapitre 4 : LE GESTIONNAIRE ORGANIGRAMMES ET PROGRAMMES

- 1- Programmes du gestionnaire
- 2- Définition des tables du gestionnaire
- 3- Simulation
- 4- Organigrammes et programmes

#### Chapitre 5 : HORLOGE TEMPS REEL : LE TEMPORISATEUR PROGRAMMABLE MC6840

- 1- Généralités
- 2- Interconnexion du boîtier et adressage
- 3- Organisation interne du temporisateur
  - 3.1- Registres de commande
  - 3.2- Registre d'état

- 3.3- Initialisation des registres
- 3.4- Entrées asynchrones et sorties
- 4- Mode intervalle de temps
  - 4.1- Mode mesure de periode
  - 4.2- Mode comparaison de largeurs d'impulsions

#### Chapitre 6 : PROGRAMMATION DU TEMPORISATEUR MC 6840

- 1- Logique d'adressage
- 2- Branchement des timers
- 3- Programmation des timers
- 4- Exemple de mise en œuvre du timer
- 5- Programme de l'horloge

#### Chapitre 7 : REALISATION D'UNE CARTE EPROM

- 1- Etude de l'EPROM 2708
- 2- Etude des circuits utilisés
- 3- Obtention du -5V

#### CONCLUSION

## INTRODUCTION

Prolongement de l'outil primitif, la machine a rendu possible notre civilisation. Entre toutes les machines, la dernière née, l'ordinateur, est porteuse d'extraordinaires promesses.

Machine universelle, capable d'automatiser un laminoir ou d'analyser un texte ancien.

Cette révolution initiale est maintenant suivie d'une évolution technologique qui va multiplier son importance: les microprocesseurs.

Ces petits ordinateurs, petits par le coût et par la taille, donnent le jour à de nombreuses applications : conduites d'unités industrielles, d'expériences , d'appareils d'analyse, d'engins spatiaux, surveillance automatique de grands malades, etc...

Ce projet est l'étude d'un gestionnaire de tâches et d'une horloge temps réel, éléments essentiels à la conduite et au contrôle d'un processus.

Dans notre cas il s'agit d'un processus électrochimique.

CHAPITRE 1 :      RAPPELS SUR L'UNITE CENTRALE  
MC 6800

Le processus nécessite un microprocesseur pour la gestion de l'information. On utilisera le MC 6800 de MOTOROLA.

On rappelle dans ce chapitre les caractéristiques et performances de cette unité centrale.

C'est la première unité de la famille 6800, aux performances modestes, mais encore largement utilisée dans les applications simples.

1- CARACTERISTIQUES DU MC 6800 :

Le 6800 est un microprocesseur monolithique 8 bits, réalisant la fonction d'unité centrale, compatible TTL, ne demande qu'une alimentation de +5V et n'a besoin d'aucun circuit externe TTL pour l'interface avec le bus.

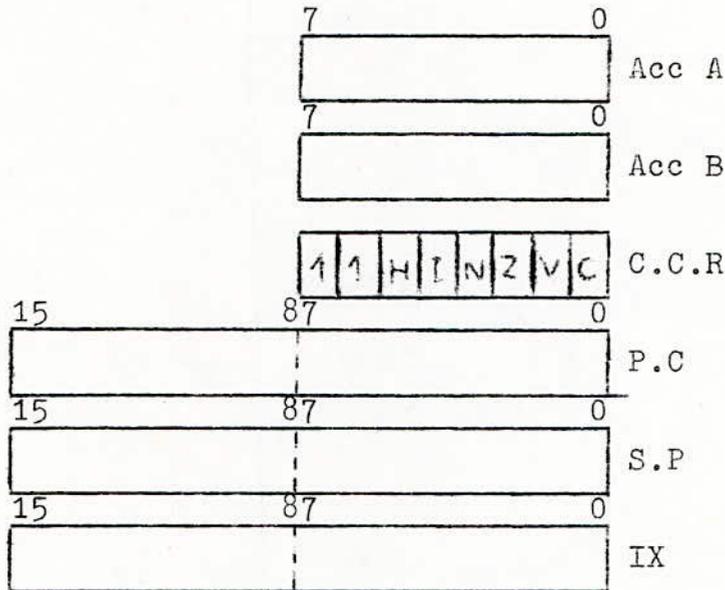
Ses principales caractéristiques sont les suivantes :

- traitement sur 8 bits en parallèle
- bus de données bidirectionnel
- bus d'adresses 16 bits; espace d'adressage 64 Koctets
- 72 instructions, longueur variable
- redémarrage vectorisé
- 7 modes d'adressage
- pile externe de longueur variable
- vecteur d'interruption masquable
- interruption non masquable séparée, registres internes sauvegardés dans la pile.
- 6 registres internes
- possibilités d'accès direct en mémoire et de configuration multiprocesseurs.
- caractéristiques d'horloge simplifiées

- fréquence d'horloge jusqu'à 2MHz
- interfaçage avec le bus simple, sans circuit TTL
- possibilité d'arrêt et d'exécution pas à pas.

2- DIFFERENTS REGISTRES ADRESSABLES :

La figure suivante montre les six registres internes :



On rappelle également leur utilisation :

- le compteur de programme (Program Counter) contient l'adresse courante du programme. Il est initialisé à l'adresse de départ du programme à exécuter.

- le pointeur de pile (Stack Pointer) contient en permanence la première adresse disponible de la pile. La pile est une zone de sauvegarde des registres, de certaines données, toujours située en mémoire vive. Elle est utilisée automatiquement lors de procédures d'interruption et lors d'appel à un sous programme par le système.

- le registre d'index (Index Register) sert de pointeur d'adresse dans le mode d'adressage indexé. Accessoirement il peut servir de registre intermédiaire 16 bits.

- le registre d'état (Code Condition Register) est un registre 8 bits dont les six premiers seulement sont utilisés. Les deux derniers sont fixés à un. Il comporte les indicateurs d'état (Flags) suivants :

+ H, half carry ou demi retenue. Il s'agit de la retenue du bit 4 sur le bit 5. Cet indicateur est exploité par l'instruction DAA pour formater en BCD des additions de nombres BCD.

+ I, interrupt flag ou indicateur d'interruption. Il s'agit du masque pour la prise en compte des demandes d'interruption masquables IRQ.

+ N, negative flag ou indicateur de signe. Il recopie le bit de plus fort poids d'un accumulateur.

+ Z, zero flag ou indicateur de zéro. Celui-ci indique simplement si un résultat est nul.

+ V, overflow ou débordement de capacité. Ce bit permet en conjonction avec les indicateurs N et (ou) Z, d'interpréter des résultats sur des nombres signés.

+ C, carry ou retenue. C'est la retenue du bit 8 sur le bit 9 .

Les indicateurs d'état sont positionnés à 1 , si la proposition qu'ils représentent est vraie. Le registre d'état est positionné, tout ou en partie, après chaque instruction de traitement arithmétique ou logique.

### 3- DIFFERENTS MODES D'ADRESSAGE :

Le 6800 dispose de 72 instructions et de 7 modes d-adressage :

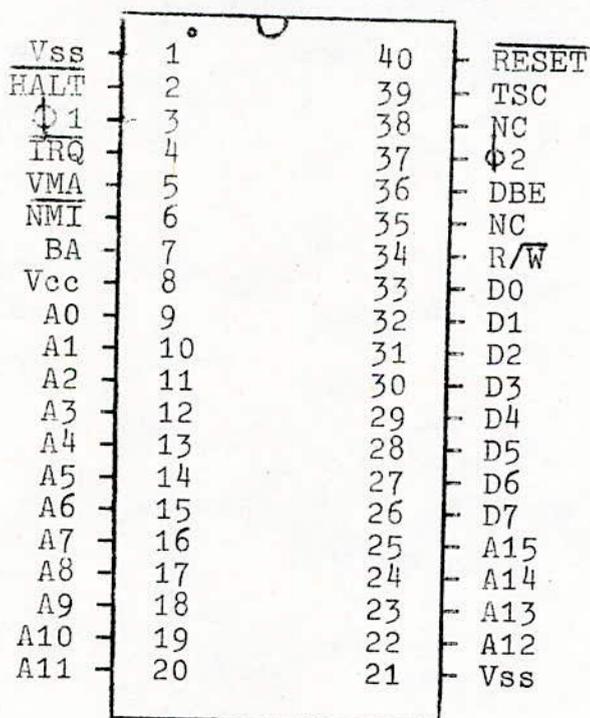
- adressage immédiat sur un octet : le champ adresse qui suit l'instruction contient l'opèrande.
- adressage direct sur un octet: le champ adresse qui suit l'instruction contient l'adresse effective de l'opèrande.
- adressage étendu sur deux octets: le champ adresse qui suit l'instruction contient l'adresse effective, sur deux octets, de l'opèrande.

- adressage indexé sur un octet: l'adresse effective de l'opèrande est obtenue en additionnant le contenu du champ adresse qui suit l'instruction,déplacement, au contenu du registre d'index .

- adressage implicite et adressage sur accumulateur ou inhèrent: le code opèration de l'instruction contient à la fois l'opèration et l'adresse du registre dèstinataire. Ce type d'adressage est réservé aux registres internes.

- adressage relatif: au contenu du compteur de programme.Le contenu du champ adresse qui suit l'instruction,uniquement de test et (ou) de branchement,indique la valeur relative signée de combien de pas le compteur de programme P.C doit avancer.

#### 4- BROCHAGE DU 6800 ET SIGNIFICATION DES DIFFERENTES LIGNES:



La signification des differentes lignes est la suivante:

- Bus d'adresse A0-A15: lignes de sortie à trois états.
- Bus de données D0-D7: bus bidirectionnel à sorties trois états.
- R/W : ligne de lecture écriture qui précise le sens des transferts entre microprocesseur et périphériques.

- VMA (Valid Memory Address): sortie qui signale une adresse valide sur le bus d'adresses.

- Entrées d'horloge  $\phi 1$  et  $\phi 2$ : l'unité centrale exige deux phases d'horloge sans recouvrement.

- RESET : cette entrée, active à l'état bas, initialise le microprocesseur et permet son démarrage après une mise sous tension ou une panne d'alimentation.

- IRQ : ligne d'entrée de demandes d'interruptions masquable active au niveau bas.

- NMI : ligne d'entrée de demandes d'interruptions non masquable, active au niveau bas.

- HALT : ligne d'entrée permettant, sur un niveau bas, d'arrêter le fonctionnement du microprocesseur. Cette commande est prévue pour faire de l'exécution de programme pas à pas.

- TSC (Three State Control) : ligne permettant sur un niveau haut de mettre le bus d'adresses et la ligne R/W à l'état haute impédance. Elle s'utilise éventuellement pour faire un DMA de brève durée.

- DBE (Data Bus Enable): ligne permettant sur un niveau bas, de mettre le bus de données à l'état haute impédance.

- BA (Bus Available) : ligne indiquant, si activée, que le microprocesseur est arrêté et que le bus d'adresses est disponible.

#### 4- DIFFERENTES VERSIONS :

Les différentes versions ont le même jeu d'instructions mais des vitesses différentes:

- 6800 : horloge 1 MHz , temps minimum pour une instruction  $2\mu s$ .

- 68A00 : horloge 1,5 MHz , temps minimum pour une instruction  $1,3\mu s$ .

- 68B00 : horloge 2MHz , temps minimum pour une instruction  $1\mu s$ .

ACCUMULATOR AND MEMORY		ADDRESSING MODES										BOOLEAN/ARITHMETIC		COND. CODE REG						
		IMMED		DIRECT		INDEX		EXTND		INTHER		OPERATION (All register labels refer to contents)		S	4	3	2	1	0	
OPERATIONS	MNEMONIC	OP	~	OP	~	OP	~	OP	~	OP	~	OP	~		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													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	↑	↑	↑	↑	↑	↑
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3	B · M → B	↑	↑	↑	↑	↑	↑
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3	A · M → B	↑	↑	↑	↑	↑	↑
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3	B · M → B	↑	↑	↑	↑	↑	↑
Clear	CLR							8F	7	2	7F	6	3	00 → M	↑	↑	↑	↑	↑	↑
	CLRA										4F	2	1	00 → A	↑	↑	↑	↑	↑	↑
	CLRB										5F	2	1	00 → B	↑	↑	↑	↑	↑	↑
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 Complement, 1's	CBA										11	2	1	A - B	↑	↑	↑	↑	↑	↑
	COM							63	7	2	73	6	3	M → M	↑	↑	↑	↑	↑	↑
	COMA										43	2	1	A → A	↑	↑	↑	↑	↑	↑
Complement, 2's (Negate)	COMB										53	2	1	B → B	↑	↑	↑	↑	↑	↑
	NEG							60	7	2	70	6	3	00 - M + M	↑	↑	↑	↑	↑	↑
Decimal Adjust, A	NEGA										40	2	1	00 - A → A	↑	↑	↑	↑	↑	↑
	NEGB										50	2	1	00 - B → B	↑	↑	↑	↑	↑	↑
Decrement	DAA										19	2	1	Converts Binary Add. of BCD Characters into BCD Format	↑	↑	↑	↑	↑	↑
	DEC							6A	7	2	7A	6	3	M - 1 → M	↑	↑	↑	↑	↑	↑
Exclusive OR	DECA										4A	2	1	A - 1 → A	↑	↑	↑	↑	↑	↑
	DECB										5A	2	1	B - 1 → B	↑	↑	↑	↑	↑	↑
Increment	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3	A ⊕ M → A	↑	↑	↑	↑	↑	↑
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3	B ⊕ M → B	↑	↑	↑	↑	↑	↑
Load Acmltr	INC							6C	7	2	7C	6	3	M + 1 → M	↑	↑	↑	↑	↑	↑
	INCA										4C	2	1	A + 1 → A	↑	↑	↑	↑	↑	↑
Or, inclusive	INCB										5C	2	1	B + 1 → B	↑	↑	↑	↑	↑	↑
	LDAA	66	2	2	96	3	2	A6	5	2	B6	4	3	M → A	↑	↑	↑	↑	↑	↑
Push Data	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3	M → B	↑	↑	↑	↑	↑	↑
	ORA	8A	2	2	9A	3	2	AA	5	2	BA	4	3	A + M → A	↑	↑	↑	↑	↑	↑
Pull Data	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3	B + M → B	↑	↑	↑	↑	↑	↑
	PSHA										36	4	1	A → M <sub>SP</sub> , SP - 1 → SP	↑	↑	↑	↑	↑	↑
Rotate Left	PSHB										37	4	1	B → M <sub>SP</sub> , SP - 1 → SP	↑	↑	↑	↑	↑	↑
	PULA										32	4	1	SP + 1 → SP, M <sub>SP</sub> → A	↑	↑	↑	↑	↑	↑
Rotate Right	PULB										33	4	1	SP + 1 → SP, M <sub>SP</sub> → B	↑	↑	↑	↑	↑	↑
	ROL							69	7	2	79	6	3	M	↑	↑	↑	↑	↑	↑
Shift Left, Arithmetic	ROLA										48	2	1	A	↑	↑	↑	↑	↑	↑
	ROLB										58	2	1	B	↑	↑	↑	↑	↑	↑
Shift Right, Arithmetic	ROR							66	7	2	76	6	3	M	↑	↑	↑	↑	↑	↑
	RORA										46	2	1	A	↑	↑	↑	↑	↑	↑
Shift Right, Logic	RORB										56	2	1	B	↑	↑	↑	↑	↑	↑
	ASL							68	7	2	78	6	3	M	↑	↑	↑	↑	↑	↑
Store Acmltr	ASLA										48	2	1	A	↑	↑	↑	↑	↑	↑
	ASLB										58	2	1	B	↑	↑	↑	↑	↑	↑
Subtract	ASRA							67	7	2	77	6	3	M	↑	↑	↑	↑	↑	↑
	ASRB										47	2	1	A	↑	↑	↑	↑	↑	↑
Subtract Acmltrs	LSR							64	7	2	74	6	3	M	↑	↑	↑	↑	↑	↑
	LSRA										44	2	1	A	↑	↑	↑	↑	↑	↑
Subtr. with Carry	LSRB										54	2	1	B	↑	↑	↑	↑	↑	↑
	STAA							97	4	2	A7	6	3	A → M	↑	↑	↑	↑	↑	↑
Transfer Acmltrs	STAB							D7	4	2	E7	6	3	B → M	↑	↑	↑	↑	↑	↑
	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3	A - M → A	↑	↑	↑	↑	↑	↑
Test, Zero or Minus	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3	B - M → B	↑	↑	↑	↑	↑	↑
	SBA										10	2	1	A - B → A	↑	↑	↑	↑	↑	↑
Test, Zero or Minus	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	↑	↑	↑	↑	↑	↑
Test, Zero or Minus	TAB										16	2	1	A → B	↑	↑	↑	↑	↑	↑
	TBA										17	2	1	B → A	↑	↑	↑	↑	↑	↑
Test, Zero or Minus	TST							6D	7	2	7D	6	3	M - 00	↑	↑	↑	↑	↑	↑
	TSTA										4D	2	1	A - 00	↑	↑	↑	↑	↑	↑
	TSTB										5D	2	1	B - 00	↑	↑	↑	↑	↑	↑

INDEX REGISTER AND STACK POINTER OPERATIONS		MNEMONIC										BOOLEAN/ARITHMETIC OPERATION		5 4 3 2 1 0 H I N Z V C							
		IMMED		DIRECT		INDEX		EXTND		INNER											
		OP	~	OP	~	OP	~	OP	~	OP	~			H	I	N	Z	V	C		
Compare Index Reg	CPX	8C	3	3		9C	4	2	AC	6	2	BC	5	3	(X <sub>H</sub> /X <sub>L</sub> ) - (M/N + 1)	.	.	⑦	⑧	.	.
Decrement Index Reg	DEX											09	4	1	X - 1 → X	.	.	.	.	.	.
Decrement Stack Ptr	DES											34	4	1	SP - 1 → SP	.	.	.	.	.	.
Increment Index Reg	INX											08	4	1	X + 1 → X	.	.	.	.	.	.
Increment Stack Ptr	INS											31	4	1	SP + 1 → SP	.	.	.	.	.	.
Load Index Reg	LDX	CE	3	3		DE	4	2	EE	6	2	FE	5	3	M → X <sub>H</sub> (M + 1) → X <sub>L</sub>	.	.	⑨	.	.	R
Load Stack Ptr	LDS	BE	3	3		9E	4	2	AE	6	2	BE	5	3	M → SP <sub>H</sub> (M + 1) → SP <sub>L</sub>	.	.	⑩	.	.	R
Store Index Reg	STX					DF	5	2	EF	7	2	FF	6	3	X <sub>H</sub> → M, X <sub>L</sub> → (M + 1)	.	.	⑪	.	.	R
Store Stack Ptr	STS					9F	5	2	AF	7	2	BF	6	3	SP <sub>H</sub> → M, SP <sub>L</sub> → (M + 1)	.	.	⑫	.	.	R
Indx Reg → Stack Ptr	TXS														X - 1 → SP	.	.	.	.	.	.
Stack Ptr → indx Reg	TSX														SP + 1 → X	.	.	.	.	.	.

JUMP AND BRANCH OPERATIONS		MNEMONIC										BRANCH TEST		5 4 3 2 1 0 H I N Z V C							
		RELATIVE		INDEX		EXTND		INNER													
		OP	~	OP	~	OP	~	OP	~	OP	~	H	I	N	Z	V	C				
Branch Always	BRA	20	4	2								None	.	.	.	.	.	.			
Branch If Carry Clear	BCC	24	4	2								C = 0	.	.	.	.	.	.			
Branch If Carry Set	BCS	25	4	2								C = 1	.	.	.	.	.	.			
Branch If = Zero	BEQ	27	4	2								Z = 1	.	.	.	.	.	.			
Branch If > Zero	BGE	2C	4	2								N ⊕ V = 0	.	.	.	.	.	.			
Branch If > Zero	BGT	2E	4	2								Z + (N ⊕ V) = 0	.	.	.	.	.	.			
Branch If Higher	BHI	22	4	2								C + Z = 0	.	.	.	.	.	.			
Branch If < Zero	BLE	2F	4	2								Z + (N ⊕ V) = 1	.	.	.	.	.	.			
Branch If Lower Or Same	BLS	23	4	2								C + Z = 1	.	.	.	.	.	.			
Branch If < Zero	SLT	2D	4	2								N ⊕ V = 1	.	.	.	.	.	.			
Branch If Minus	BMI	2B	4	2								N = 1	.	.	.	.	.	.			
Branch If Not Equal Zero	BNE	26	4	2								Z = 0	.	.	.	.	.	.			
Branch If Overflow Clear	BVC	28	4	2								V = 0	.	.	.	.	.	.			
Branch If Overflow Set	BVS	29	4	2								V = 1	.	.	.	.	.	.			
Branch If Plus	BPL	2A	4	2								N = 0	.	.	.	.	.	.			
Branch To Subroutine	BSR	8D	8	2									.	.	.	.	.	.			
Jump	JMP					BE	4	2	7E	3	3	} See Special Operations	.	.	.	.	.	.			
Jump To Subroutine	JSR					AD	8	2	BD	9	3		.	.	.	.	.	.			
No Operation	NOP											01	2	1	Ad Prog Cntr. Only	.	.	.	.	.	.
Return From interrupt	RTI											3B	10	1	} See special Operation	.	.	.	.	.	.
Return From Subroutine	RTS											39	5	1		.	.	.	.	.	.
Software Interrupt	SWI											3F	12	1	.	.	.	.	.	.	
Wait for Interrupt	WAI											3E	9	1	.	.	.	.	.	.	

CONDITIONS CODE REGISTER		MNEMONIC										CONDITION CODE REGISTER NOTES					
		INNER		BOOLEAN		5 4 3 2 1 0				5 4 3 2 1 0 H I N Z V C							
		OP	~	OP	~	OP	~	OP	~	OP	~	H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1		0 → C	.	.	.	.	.	.	.	.	.	.	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	.	.	.	.	.	.	.	.	.	.	④ (Bit V) Test Operand = 10000000 prior to execution ?
Set Interrupt Mask	SEI	0F	2	1		1 → I	.	S	.	.	.	.	.	.	.	.	⑤ (Bit V) Test Operand = 01111111 prior to execution ?
Set Overflow	SEV	0B	2	1		1 → V	.	.	.	.	S	.	.	.	.	.	⑥ (BIT V) Test Set equal to result of N ⊕ C after shift has occurred
Acmltr A → CCR	TAP	06	2	1		A → CCR	.	.	.	.	.	.	.	.	.	.	⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
CCR → Acmltr A	TPA	07	2	1		CCR → A	.	.	.	.	.	.	.	.	.	.	⑧ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?

- LEGEND**
- OP Operation Code (Hexadecimal)
  - ~ Number of MPU Cycles
  - ≠ Number of Program Bytes
  - + Arithmetic Plus
  - Arithmetic Minus
  - Boolean AND
  - M<sub>sp</sub> 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

- CONDITION CODE REGISTER NOTES**  
(Bit set if test is true and cleared other wise)
- ① (Bit V) Test Result = 10000000 ?
  - ② (Bit C) Test Result = 00000000 ?
  - ③ (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared it previously set)
  - ④ (Bit V) Test Operand = 10000000 prior to execution ?
  - ⑤ (Bit V) Test Operand = 01111111 prior to execution ?
  - ⑥ (BIT V) Test Set equal to result of N ⊕ C after shift has occurred
  - 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 than 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

## CHAPITRE 2 :    CONTROLE DE PROCESSUS EN                           TEMPS REEL

Dans diverses applications les micro-ordinateurs servent de contrôleur sophistiqué, utilisant les ressources du système pour commander un processus.

Divers capteurs l'informent constamment de l'évolution du processus, d'une part et divers actionneurs sont sous son contrôle d'autre part.

Le but de ce chapitre est d'établir un lien entre hardware et software en décrivant un moniteur en temps réel.

### 1- PROCESSUS TEMPS REEL :

Généralement un processus se décompose en trois parties: un moniteur temps réel ,des tâches primaires et des tâches secondaires.

Le moniteur est nécessaire pour la programmation des tâches et d'autres fonctions d'entretien ou de maintenance.

Les tâches primaires sont les tâches principales ou urgentes du processus, tel que traitement d'événements extérieurs ou situations d'alarme.

Il ya donc des cas très critiques, bénéficiant de la plus haute priorité et exécutés immédiatement.

Les tâches secondaires bénéficient d'une priorité plus faible et d'un traitement normal.

La classification des tâches dépend tout d'abord du processus, de l'application, mais le moniteur est plus général et peut être traité plus en détail.

Le moniteur était destiné à être utilisé dans un système de contrôle où certains paramètres doivent être estimés à une valeur prédéterminée. En plus, des situations d'alarme pourraient, comme précisé, attirer immédiatement l'attention du contrôleur.

Le système doit fonctionner en boucle fermée. Tout le logiciel (software) nécessaire réside en mémoire et chaque fonction est traitée suivant un algorithme. La structure du logiciel du système est représentée à la fig1 . Les programmes de traitement fonctionnent d'après les programmes de contrôle qui sont au même nombre que les tâches. Ils se divisent en deux parties: les tâches et les programmes communs.

Les tâches sont prioritaires ou secondaires et leur séquence d'exécution est fonction de la priorité attribuée par les programmes de contrôle.

Les programmes communs sont ceux que se partagent plusieurs tâches. Ca peut être des tâches d'entrée-sortie avec les différents périphériques ou d'autres processus.

Les programmes de contrôle se divisent en trois catégories: gestion de tâches, gestion de fonctions et gestion de données. Ces programmes constituent le moniteur temps réel.

## 2- LE MONITEUR :

Les programmes de gestion des tâches sont qualifiés, généralement, de superviseur des requêtes d'interruptions, des fonctions programmées et de commande du temporisateur (Timer).

Les programmes de gestion de fonctions traitent les commandes de l'opérateur, transmettent des données du système à l'opérateur et gèrent les divers périphériques.

Les programmes de gestion des données préparent généralement les interfaces aux programmes de traitement, aux mémoires auxiliaires et aux périphériques.

Parmi les diverses versions, les programmes suivants constituent un moniteur:

Gestion de tâches

- 1- programme d'utilisation des interruptions
- 2- programmes du temporisateur (Timer)
- 3- le gestionnaire

## Gestion de fonctions

- 1- programme du pupitre de commande
- 2- programme d'activation des tâches

Il n'y a pas de programmes de gestion des données. Toutes les opérations d'entrée-sortie pouvant être gérées par des tâches spécifiques. En considérant les opérations d'entrée-sortie comme des tâches avec différentes priorités, le moniteur sera plus compacte.

Un bloc diagramme du moniteur est représenté à la fig 2. Le contrôle du CPU est attribué, par le moniteur, aux tâches prêtes. Au même instant, il pourrait y avoir plusieurs tâches à divers stades d'exécution. Seulement, une seule tâche peut être activée à la fois. Ainsi, pour chaque tâche on peut définir quatre états :

- 1- INACT : tâche inactive
- 2- PRET : tâche prête
- 3- EXECT : tâche en exécution
- 4- SUSPD : tâche suspendue

Une tâche est à l'état inactif quand elle n'est pas sollicitée à l'exécution ou n'a pas été programmée pour être exécutée.

Une tâche est prête quand celle-ci a été sollicitée et demande à être exécutée. Son activation dépend de sa priorité. L'état en exécution est attribué à une tâche en cours d'exécution.

Enfin, une tâche est suspendue si elle est arrêtée par une autre plus prioritaire.

Si toutes les tâches sont inactives, le CPU se met en état d'attente jusqu'à ce qu'une interruption vienne solliciter une des tâches.

Il y a plusieurs types d'interruptions, pouvant parvenir pendant l'état d'attente ou en cours de fonctionnement.

Les interruptions parvenant en cours d'exécution du programme du moniteur sont inhibées jusqu'à ce que l'exécution de celui-ci soit terminée.

Dans certaines situations, il est impératif que le moniteur ne suspende pas les interruptions pendant un long moment.

Donc une optimisation du temps d'exécution du moniteur est

essentielle.

Si une interruption prioritaire arrive au cours de l'exécution d'une tâche, les registres, les compteurs et autres informations d'état ou de contrôle sont transférés en mémoire et la tâche passe à l'état suspendue. Quand la tâche sera réactivée, elle reprendra au point d'interruption si aucune tâche plus prioritaire n'est sollicitée.

Les types d'interruptions sur la fig 2 sont: interruptions temporaires, interruptions extérieures et interruptions d'activation des tâches.

Les interruptions temporaires sont générées par un temporisateur. Les interruptions extérieures englobent toutes celles, sollicitées par les périphériques, les commandes de l'opérateur et tout les systèmes d'entrée-sortie.

Les interruptions d'activation de tâches ~~sont~~ sollicitées ~~par~~ les divers actionneurs. Donc le moniteur doit être, constamment, informé des changements d'état des actionneurs des diverses tâches.

Une tâche s'exécutant périodiquement passera à l'état prête, à chaque cycle d'interruption du temporisateur, grâce au programme de mise à jour des tâches.

Après exécution de toutes les opérations de traitement des interruptions, le gestionnaire est exécuté.

Le gestionnaire est un algorithme très spécifique au processus considéré. C'est une série de règles qui déterminent dans quel cas une tâche sera activée.

L'algorithme est dit statique, si à chaque tâche est attribuée une priorité qui restera toujours la même. Il est dit dynamique si les priorités évoluent.

Les paramètres essentiels traités par le gestionnaire sont les états et les priorités des diverses tâches. Ces informations seront stockées dans une zone mémoire spécialement réservée à cet effet et est dite table.

Le rendement du moniteur peut être mesuré à l'aide du temps nécessaire à un cycle d'interruptions. Ce temps doit être acceptable dans certaines situations en temps réel du processus.

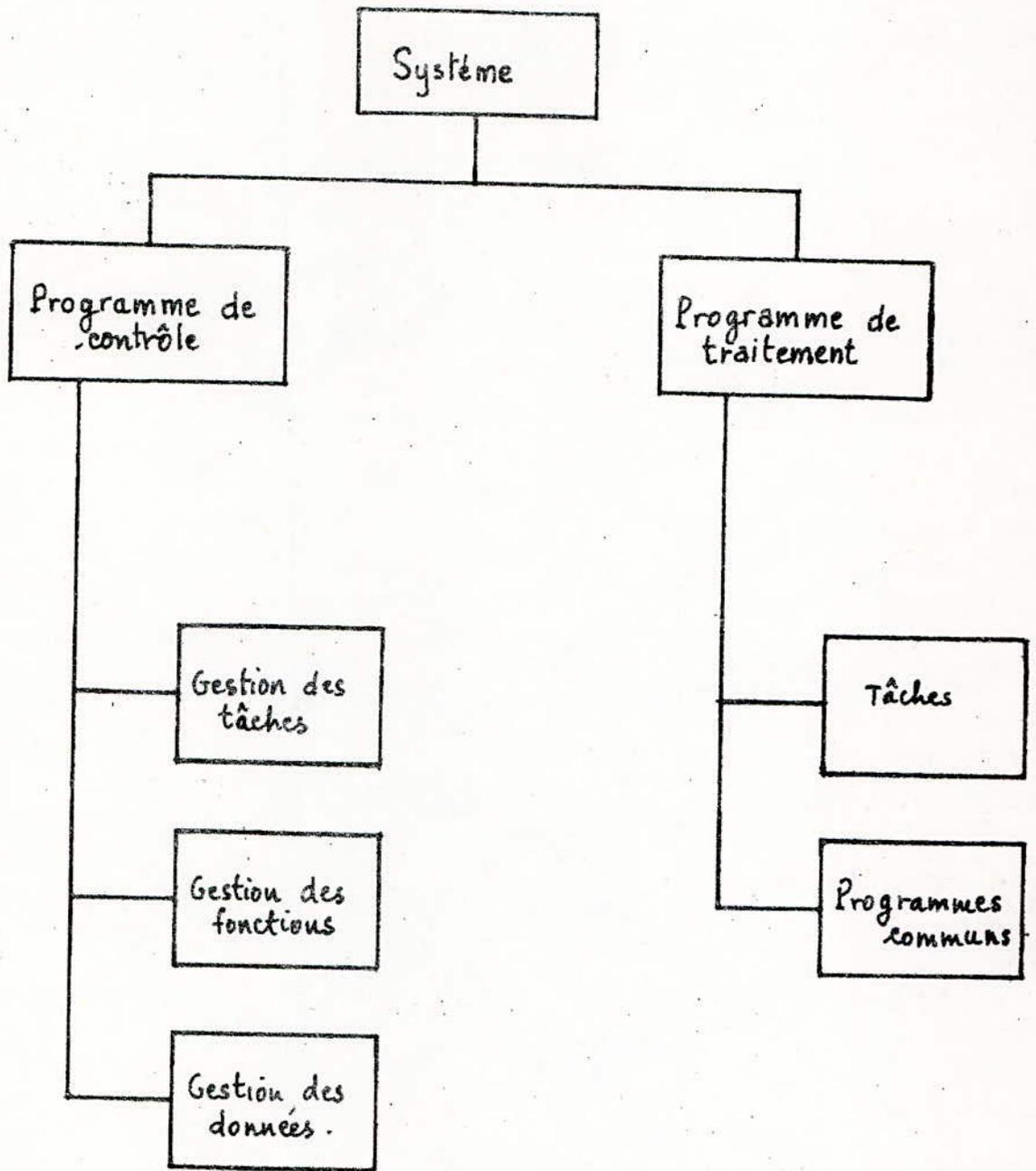


fig 1: Structure du software du système

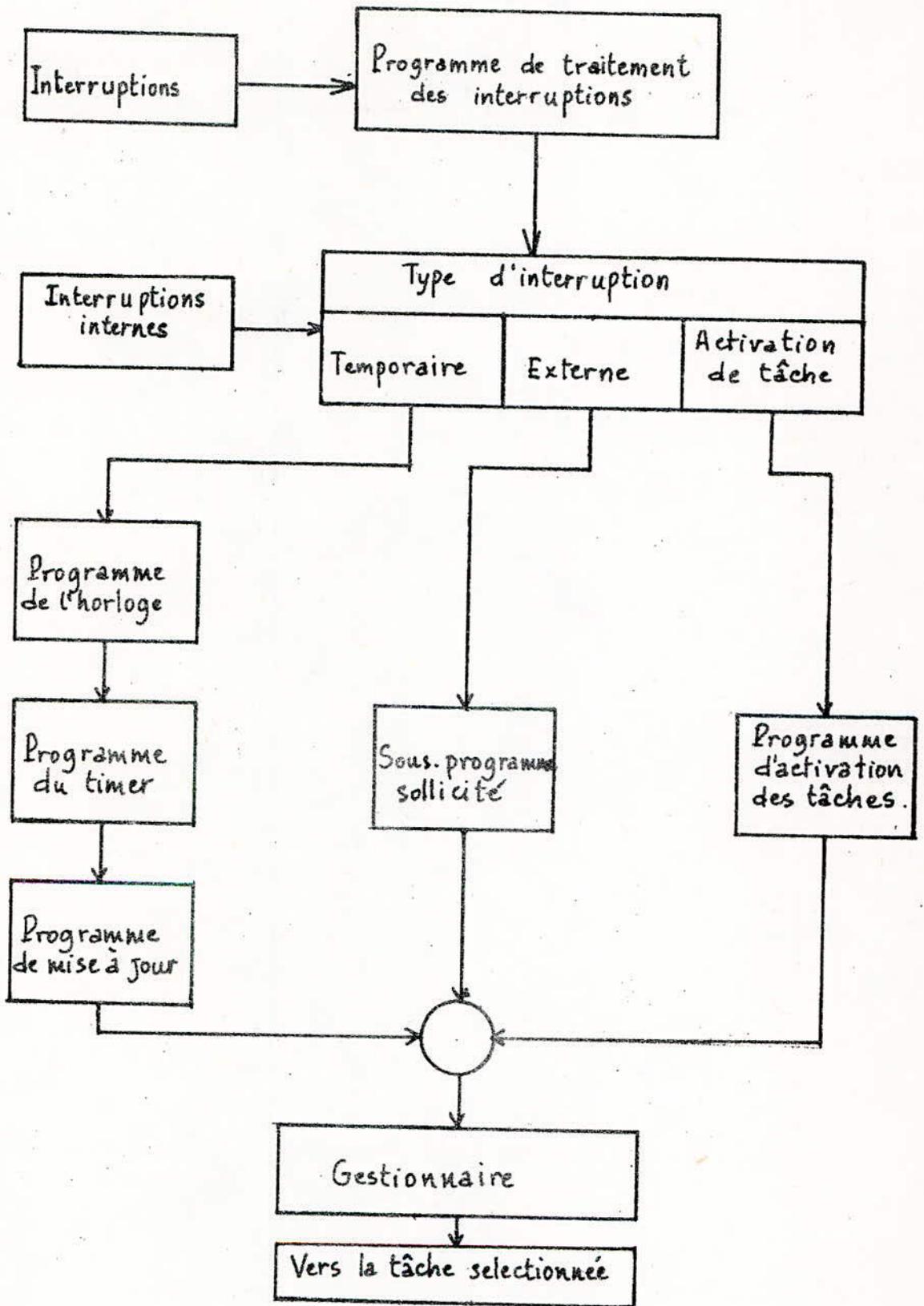


fig 2: Bloc diagramme du moniteur temps réel

### CHAPITRE 3 : LES DIFFERENTES TACHES DU PROCESSUS

Ce chapitre sera un exposé des différentes tâches du processus de leur complexité, et de la manière dont elles seront traitées par le système.

Le gestionnaire traitera les états et les paramètres essentiels des diverses tâches. Ces informations seront stockées dans une zone mémoire spécialement réservée à cet effet.

#### 1- TACHES DU PROCESSUS :

Le processus à contrôler comprend quatre tâches T1, T2, T3 et T4 .

Des capteurs informent constamment le système de l'évolution du processus et des actionneurs sont sous son contrôle pour réagir sur le processus.

Il s'agit d'un processus du type électrochimique où T1, T2, T3 et T4 seront respectivement des tâches d'étalonnage, de prélèvement, de communication des résultats vers l'extérieur et de mélange des diverses solutions utilisées.

Les tâches T1, T2 et T3 utilisent, chacune, un compteur; soit respectivement C1, C2 et C3.

Ces trois tâches s'exécutant périodiquement, les compteurs seront chargés avec les périodes d'exécution des diverses tâches, seront décrementés à chaque passage du gestionnaire et chaque fois qu'ils seront à zéro on exécutera la tâche correspondante.

La tâche T4 utilise trois compteurs C4, C5 et C6 qui contiendront les temps durant lesquels des actionneurs maintiendront l'écoulement des solutions chimiques nécessaires.

A chaque fois q'un de ces compteurs sera à zéro , on arrêtera l'écoulement de la solution correspondante. L'exécution d'une tâche, signifie exécution du programme correspondant à cette tâche.

Taches $T_j$	Compteurs $C_j$	SS. Prog. $P_j$
T1	C1	P1
T2	C2	P2
T3	C3	P3
T4	C4, C5, C6	P4.

On définira plusieurs états possibles pour chacune des tâches

- tâche en exécution (TE) : tâche en cours d'exécution.
- tâche prête (TP) : tâche sollicitant l'exécution.
- tâche en sommeil : tâche non sollicitée à l'exécution.

## 2- TABLES DU GESTIONNAIRE :

La figure 1 <sup>montre</sup> toutes les tables du gestionnaire.

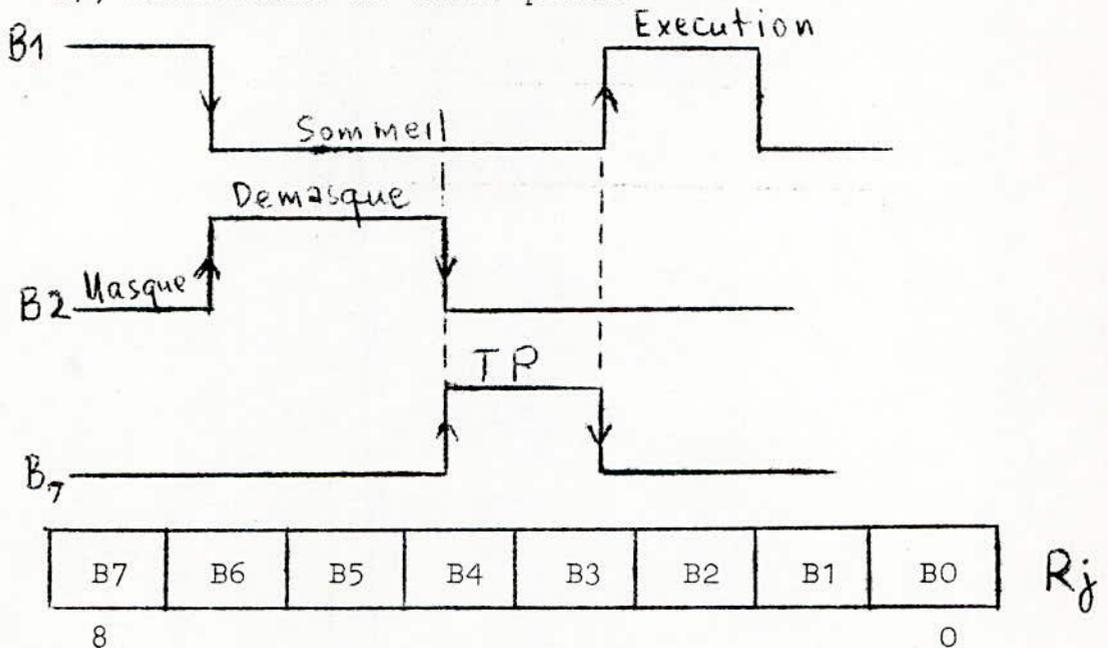
On conservera en mémoire l'état des tâches, les adresses de leurs sous-programmes et les temps de décrementation des compteurs.

On utilisera pour chacune des tâches, un mot de huit bits comme registre d'état.

Signification de chacun des bits du registre d'état :

- B0 = 1 tâche active
- B0 = 0 tâche inactive

- B1 = 1 tâche en exécution
  - B1 = 0 tâche en sommeil
  - B2, masque de la mise à jour des compteurs.
- B2 = 1 démasque MISCO  
 B2 = 0 masque MISCO
- B3, indicateur de fin de tâche.
- Mis à 1 par fin d'exécution de la tâche  
 Mis à 0 par fin d'activité
- B4, indicateur de début et d'exécution.
- Mis à 1 par fin d'exécution  
 Mis à 0 par début d'exécution
- B5 et B6, identiques à B2 mais pour tâche T4 seulement.
  - B7, indicateur de tâche prête.



Les six temps de décrémentation sont en virgule flottante et occupent ,chacun, trois octets; un pour l'exposant et les deux autres pour la mantisse sur 16 bits.

### 3- LE GESTIONNAIRE :

C'est un programme qui gérera toutes ces tâches et qui

s'exécute périodiquement grâce à un système d'interruptions. Ces interruptions seront générées par une horloge programmable permettant une grande souplesse d'utilisation. Le temps séparant deux interruptions doit être acceptable dans certaines situations temps réel du processus.

N° de tâche
MSB du sous programme
LSB du sous programme

Table des tâches en ROM

R1
R2
R3
R4

Table des registres d'état en RAM

M1 de T1
M2 de T2
M3 de T3

Table des tâches prêtes périodiques en RAM

t1
t2
t3
t4
t5
t6

Table des 6 temps de décrémentation en RAM.

fig 1: TABLES DU GESTIONNAIRE

Le gestionnaire sera composé d'un programme principal et de plusieurs sous programmes.

Les tables du moniteur comportant les données, les registres d'état et les resultats intermédiaires seront stockés en mémoire, en page zéro, pour permettre un accès plus rapide.

1- PROGRAMMES DU GESTIONNAIRE :

Les différents sous-programmes du gestionnaire sont les suivants :

- MISCO , mise à jour des compteurs :

teste les bits  $B_2^1$  ,  $B_2^2$  ,  $B_2^3$  ,  $B_2^4$  ,  $B_5^4$  et  $B_6^4$  des registres d'état.

$B_{-1}^j = 0$  décrémentation masquée des  $C_j$ .

$B_1^j = 1$  décrémentation autorisée des  $C_j$ .

- RECTEX , recherche d'une tâche en exécution :

teste les bits  $B_2$  et  $B_1$  de chacune des tâches T1, T2 et T3.

- RECTP : recherche d'une tâche prête ::

consulte le contenu des mémoires M1 , M2 et M3 contenant le numero de la tâche prête <sup>et appelle</sup> un des sous-programmes PROG 1 ou PROG 2 ou PROG 3, si une tâche est prête.

- PROG 1,2,3 , réveil d'une tâche :

+ remet dans  $C_j$  la valeur du temps à décrémentation

+ met à 1 ,  $B_0^j$  et  $B_1^j$  du registre d'état  $R_j$

+ met à 0 ,  $B_7^j$  et  $B_2^j$

+ met à zéro  $M_j$

+ la tâche prête passe à l'exécution.

- RECTAP , recherche d'une autre tâche prête .
- DECCO , décrémente les compteurs Cj .
- COMIND K , envoie dans le PIA le mot qui basculerait l'indicateur K , qui pour T4 correspond à trois actionneurs et pour les tâches T1 , T2 et T3 au programme correspondant.

## 2- TABLES DU GESTIONNAIRE :

Les tables du gestionnaire et les résultats intermédiaires seront stockés en RAM en page zéro.

Nous commencerons à l'adresse 0061 , la zone précédente étant occupée par d'autres tables du système.

0061	R1	}	Registres d'état
0062	R2		
0063	R3		
0064	R4		
0065	M1	}	Table des tâches périodiques prêtes
0066	M2		
0067	M3		
0068	EC1	}	Compteur C1
0069	C1H		
006A	C1L		
006B	EC2	}	Compteur C2
006C	C2H		
006D	C2L		
006E	EC3	}	Compteur C3
006F	C3H		
0070	C3L		
0071	EC4	}	Compteur c4
0072	C4H		
0073	C4L		

0074	EC5	}	Compteur C5
0075	C5H		
0076	C5L		
0077	EC6	}	Compteur C6
0078	C6H		
0079	C6L		

Table des temps de décrémentation :

007A	Et1	}	t1
007B	t1H		
007C	t1L		
007D	Et2	}	t2
007E	t2H		
007F	t2L		
0080	Et3	}	t3
0081	t3H		
0082	t3L		
0083	Et4	}	t4
0084	t4H		
0085	t4L		
0086	Et5	}	t5
0087	t5H		
0088	t5L		
0089	Et6	}	t6
008A	t6H		
008B	t6L		

008C }  
 008D } utilisées par DECCO

### 3- SIMULATION :

Ne disposant pas d'un processus prêt à fonctionner pour tester le logiciel du système , nous avons eu recours à une simulation.

En effet, diverses lampes représentent les divers actionneurs et leur allumage ou extinction correspond au fonctionnement ou à l'arrêt de l'actionneur dont il est question.

Le sous-programme COMIND K envoie dans le PIA le mot qui provoquerait l'allumage de l'un des indicateurs selon la valeur de K :

- K = 1 , 2 , 3 : correspond aux trois actionneurs de la tâche T4 .

- K = 4 : tâche T3.

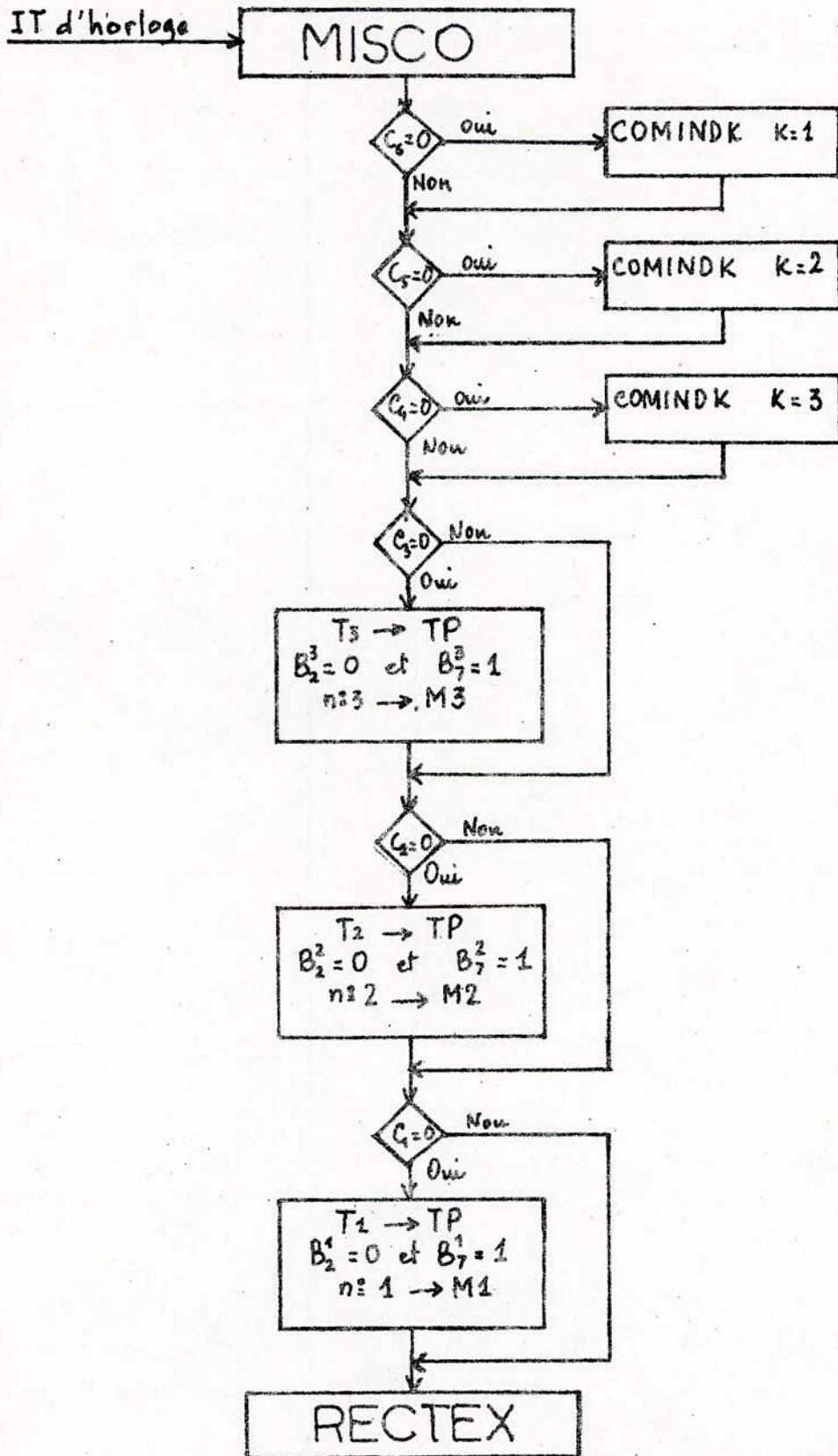
- K = 5 : tâche T2.

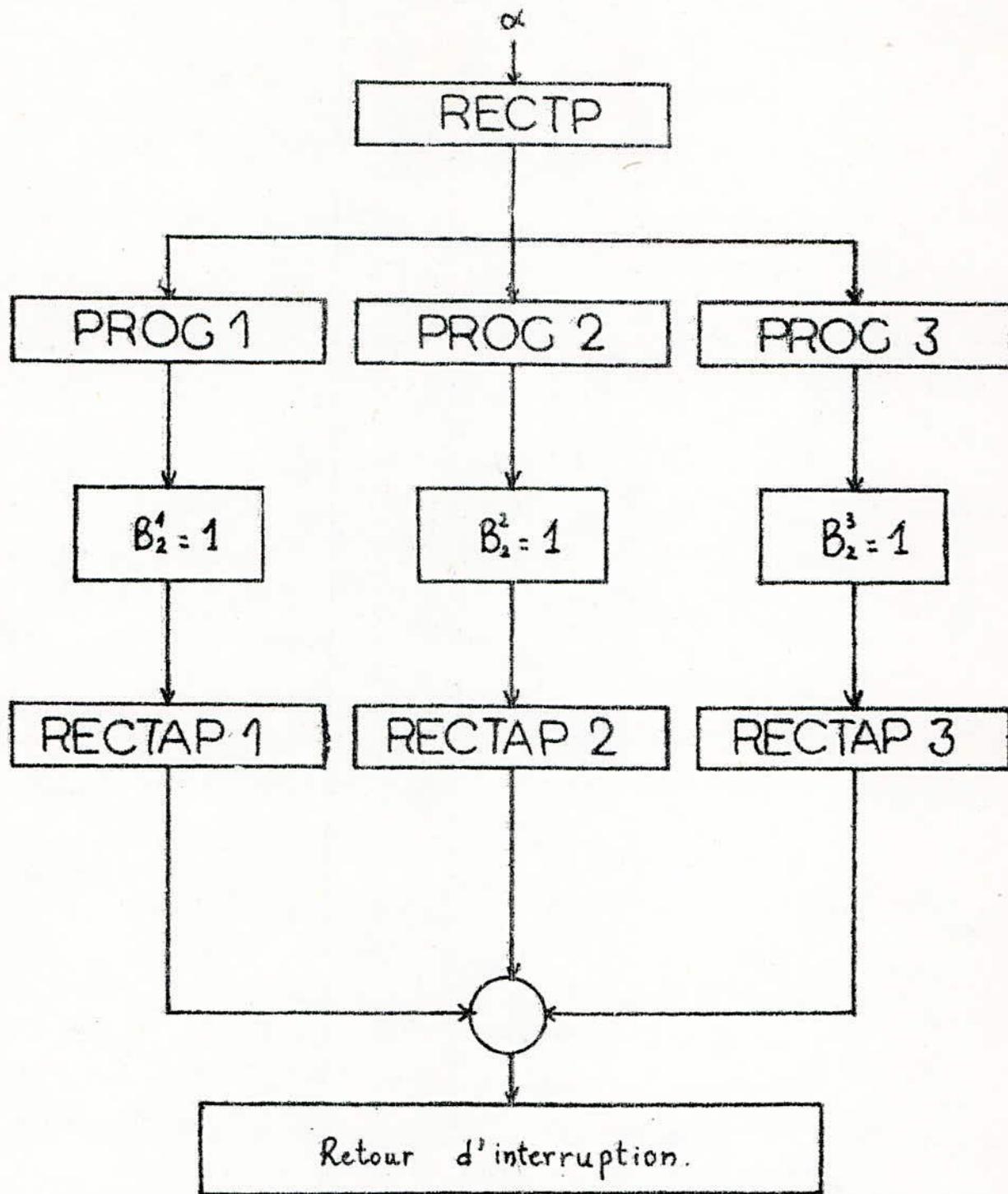
- K = 6 : tâche T1.

Ce programme ne figure pas dans ce qui suit . Il figure dans le projet de MM. BOUZA N. et BOUCETTA Y. . Il s'intitule PROPIA et se trouve à l'adresse 00E9.

### 3- ORGANIGRAMMES ET PROGRAMMES :

# PROGRAMME PRINCIPAL





PROGRAMME PRINCIPAL :

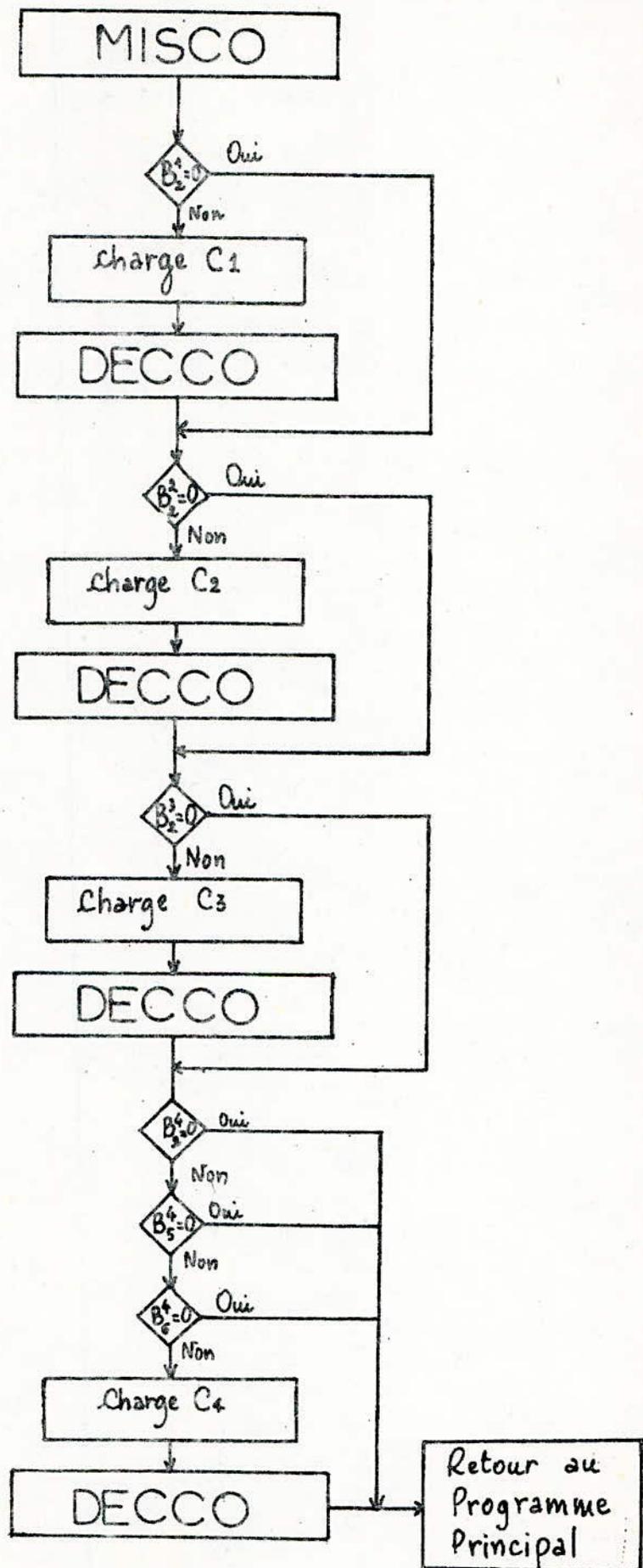
C117	BD	C000	JSR	MISCO
C11A	96	77	LDAA	EC6
C11C	26	0A	BNE	<del>§</del> 0A
C11E	DE	78	LDX	C6
C120	8C	0000	CPX	<del>§</del> 0000
C123	26	03	BNE	<del>§</del> 03
C125	BD	(XXXX)	JSR	COMINDK (K = 1)
C128	96	74	LDAA	EC5
C12A	26	0A	BNE	<del>§</del> 0A
C12C	DE	75	LDX	C5
C12E	8C	0000	CPX	<del>##</del> <del>§</del> 0000
C131	26	03	BNE	<del>§</del> 03
C133	BD	(XXXX)	JSR	COMINDK (K = 2)
C136	96	71	LDAA	EC4
C138	26	0A	BNE	<del>§</del> 0A
C13A	DE	72	LDX	C4
C13C	8C	0000	CPX	<del>##</del> <del>§</del> 0000
C13F	26	03	BNE	<del>§</del> 03
C141	BD	(XXXX)	JSR	COMINDK (K = 3)
C144	96	6E	LDAA	EC3
C146	26	13	BNE	<del>§</del> 13
C148	DE	6F	LDX	C3
C14A	8C	0000	CPX	<del>##</del> <del>§</del> 0000
C14D	26	0C	BNE	<del>§</del> 0C
C14F	96	63	LDAA	R3
C151	84	FB	ANDA	<del>##</del> % 11111011
C153	8A	80	ORAA	<del>##</del> % 10000000
C155	97	63	STAA	R3

Suite programme principal :

C157	86	03	LDAA	## § 03
C159	97	67	STAA	M3
C15B	96	6B	LDAA	EC2
C15D	26	13	BNE	§ 13
C15F	DE	6C	LDX	C2
C161	8C	0000	CPX	## § 0000
C164	26	0C	BNE	§ 0C
C166	96	62	LDAA	R2
C168	84	FB	ANDA	## % 11111011
C16A	8A	80	ORAA	## % 10000000
C16C	97	62	STAA	R2
C16E	86	02	LDAA	## § 02
C170	97	66	STAA	M2
C172	96	68	LDAA	EC1
C174	26	13	BNE	§ 13
C176	DE	69	LDX	C1
C178	8C	0000	CPX	## § 0000
C17B	26	0C	BNE	§ 0C
C17D	96	61	LDAA	R1
C17F	84	FB	ANDA	## % 11111011
C181	8A	80	ORAA	## % 10000000
C183	97	61	STAA	R1
C185	86	01	LDAA	## § 01
C187	97	65	STAA	M1
C189	BD	C05A	JSR	RECTEX
C18C	7E	C101	JMP	RECTP
C18F	96	61	LDAA	R1
C191	8A	04	ORAA	## % 00000100
C193	97	61	STAA	R1

Suite programme principal :

C195	BD	COCE	JSR	RECTAP1
C198	3B		RTI	
C199	96	62	LDAA	R2
C19B	8A	04	ORAA	## % 00000100
C19D	97	62	STAA	R2
C19F	BD	CODF	JSR	RECTAP2
C202	3B		RTI	
C203	96	63	LDAA	R3
C205	8A	04	ORAA	## % 00000100
C207	97	63	STAA	R3
C209	BD	COFO	JSR	RECTAP3
C20C	3B		RTI	

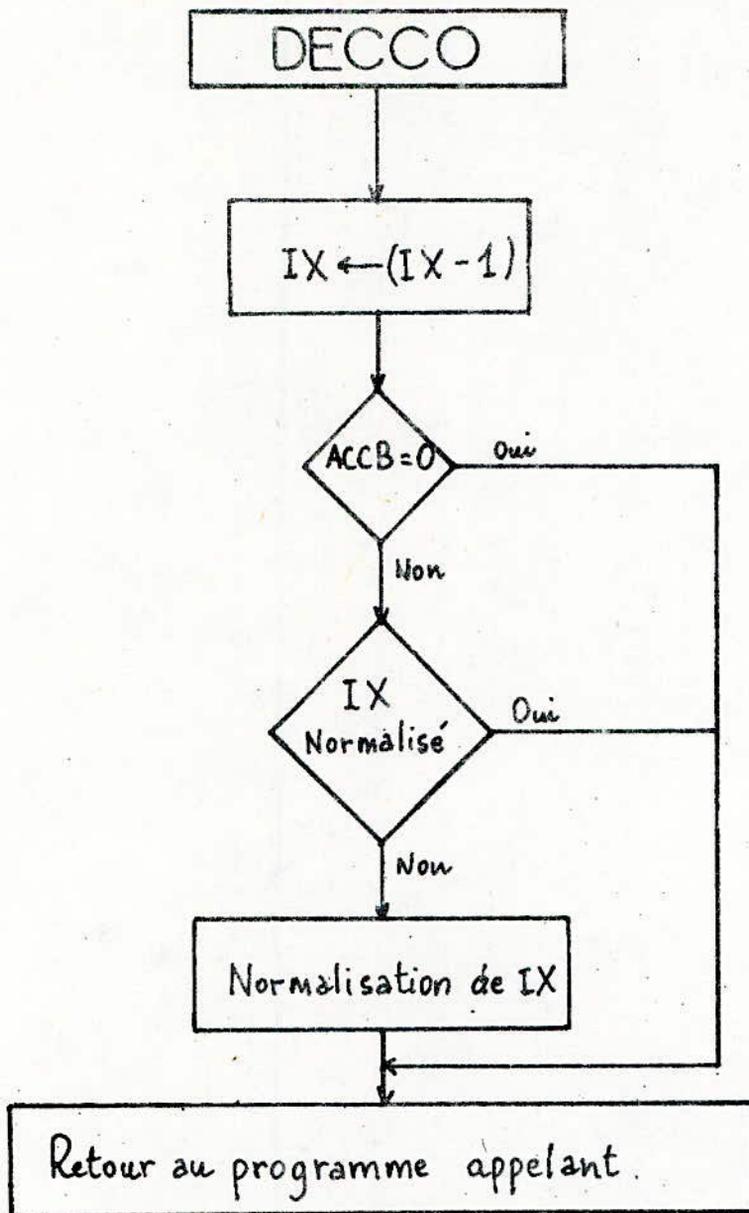


MISCO

C000	96	61	LDAA	R1
C002	06		TAP	
C003	26	0B	BNE	§ 0B
C005	D6	68	LDAB	EC1
C007	DE	69	LDX	C1
C009	BD	C048	JSR	DECCO
C00C	D7	68	STAB	§ 68
C00E	DF	69	STX	§ 69
C010	96	62	LDAA	R2
C012	06		TAP	
C013	26	0B	BNE	§ 0B
C015	D6	6B	LDAB	EC2
C017	DE	6C	LDX	C2
C019	ED	C048	JSR	DECCO
C01C	D7	6B	STAB	§ 6B
C01E	DF	6C	STX	§ 6C
C020	96	63	LDAA	R3
C022	06		TAP	
C023	26	0B	BNE	§ 0B
C025	D6	6E	LDAB	§ EC3
C027	DE	6F	LDX	C3
C029	BD	C048	JSR	DECCO
C02C	D7	6E	STAB	§ 6E
C02E	DF	6F	STX	§ 6F
C030	96	64	LDAA	R4
C032	44		LSRA	
C033	44		LSRA	
C034	44		LSRA	

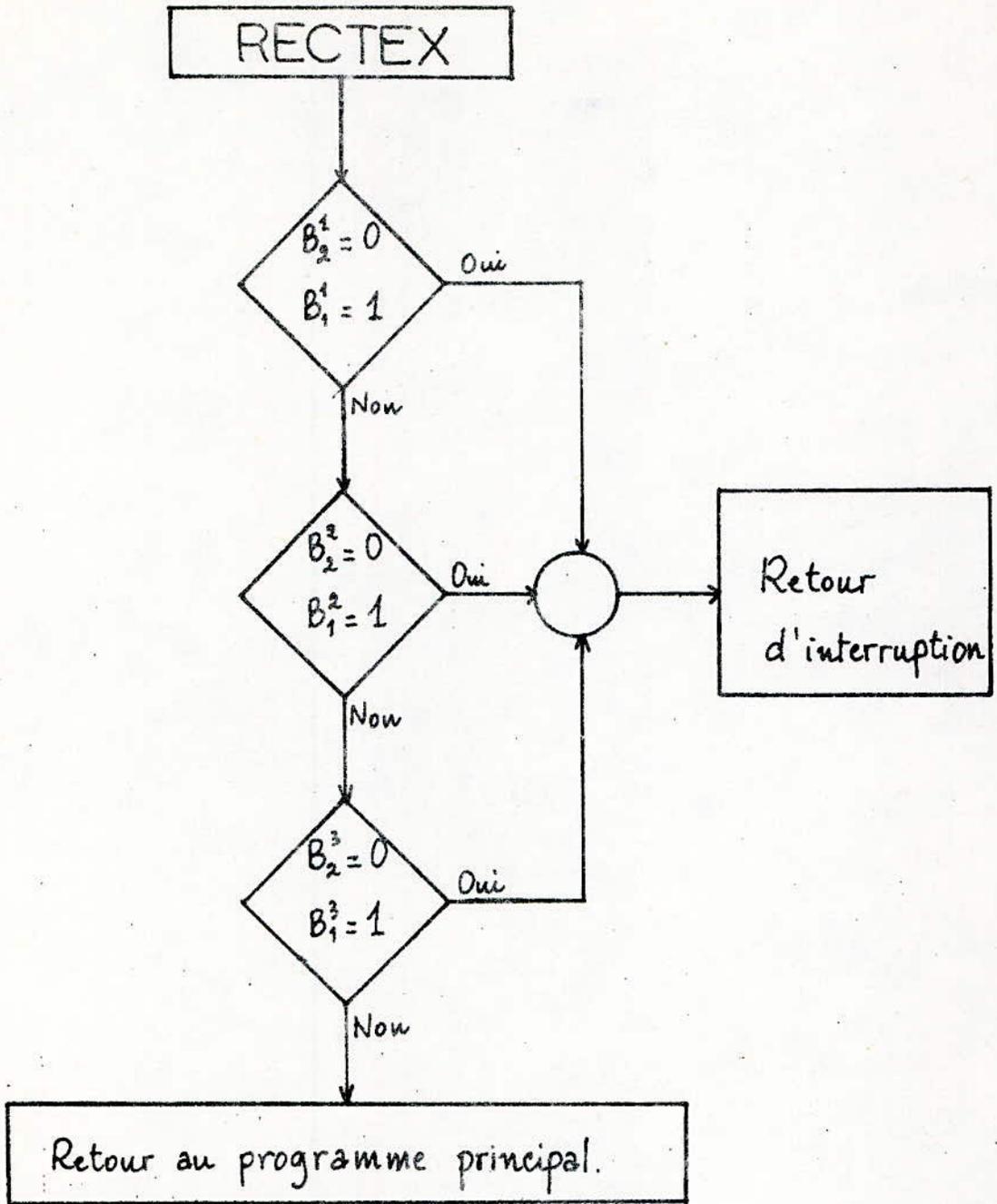
Suite MISCO :

C035	24	10	BCC	§ 10
C037	06		TAP	
C038	26	0D	BNE	§ 0D
C03A	2A	0B	BPL	§ 0B
C03C	D6	71	LDAB	EC4
C03E	DE	72	LDX	C4
C040	BD	C048	JSR	DECCO
C043	D7	71	STAB	§ 71
C045	DF	72	STX	§ 72
C047	39		RTS	



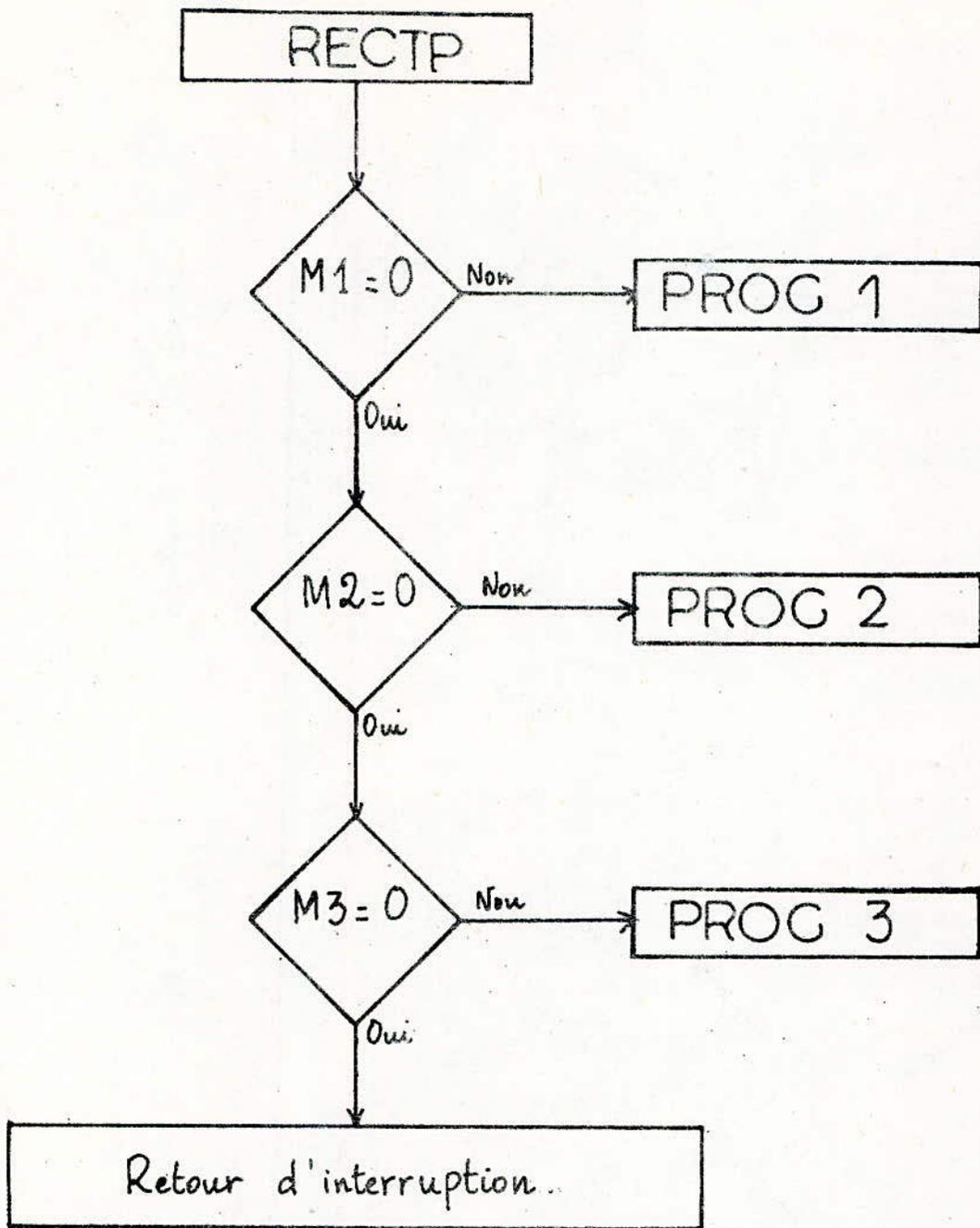
DECCO :

C048	09		DEX	
C049	17		TBA	
C04A	27	0D	BEQ	Ø 0D
C04C	DF	8C	STX	Ø 8C
C04E	78	008D	ASL	Ø 008D
C051	79	008C	ROL	Ø 008C
C054	26	03	BMI	Ø 03
C056	DE	8C	LDX	Ø 8C
C058	5A		DECB	
C059	39		RTS	



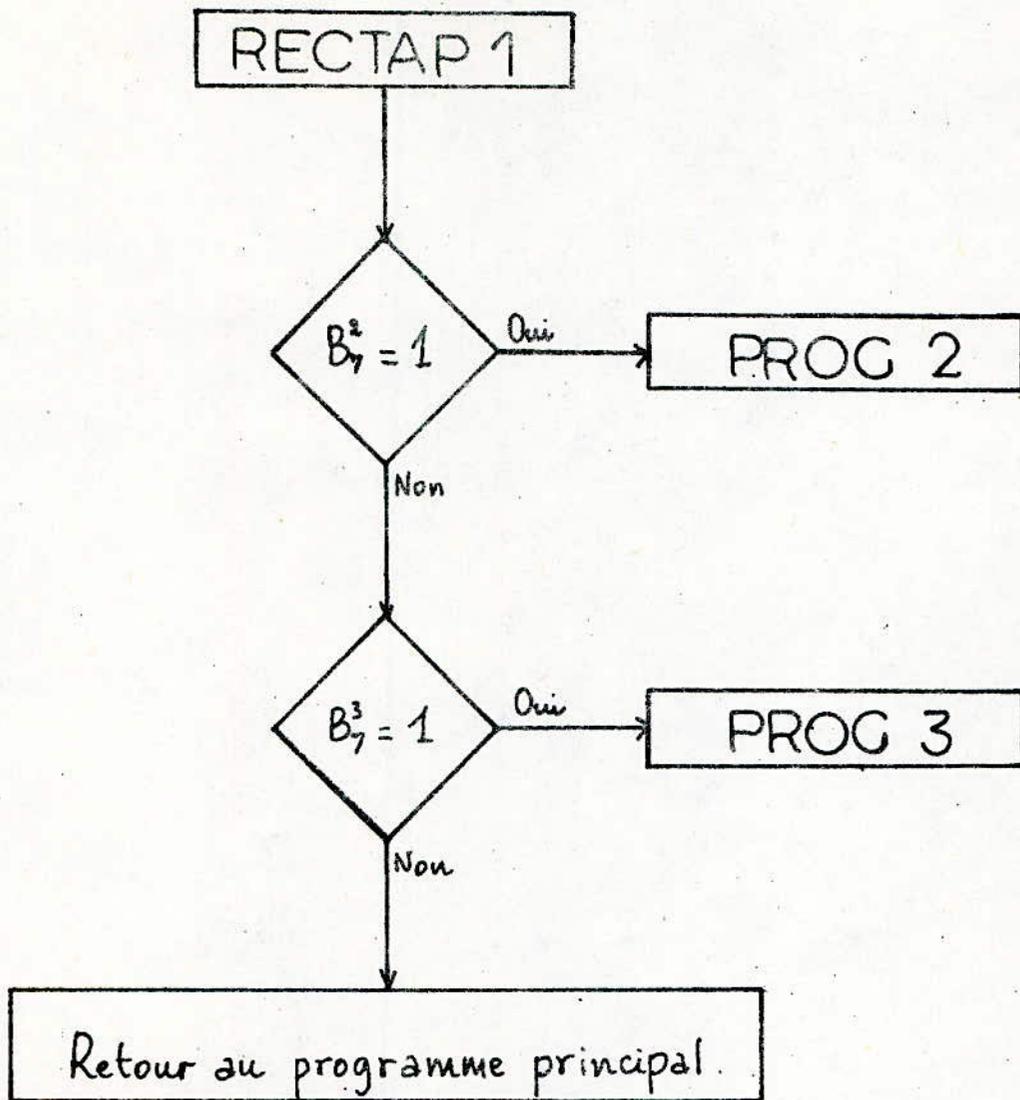
RECTEX :

C05A	96	61	LDAA	R1
C05C	06		TAP	
C05D	28	02	BVC	§ 02
C05E	26	0F	BNE	§ 0F
C060	96	62	LDAA	R2
C062	06		TAP	
C063	28	02	BVC	§ 02
C065	26	08	BNE	§ 08
C067	96	63	LDAA	R3
C069	06		TAP	
C06A	28	02	BVC	§ 02
C06C	26	01	BNE	§ 01
C06E	39		RTS	
C06F	3B		RTI	



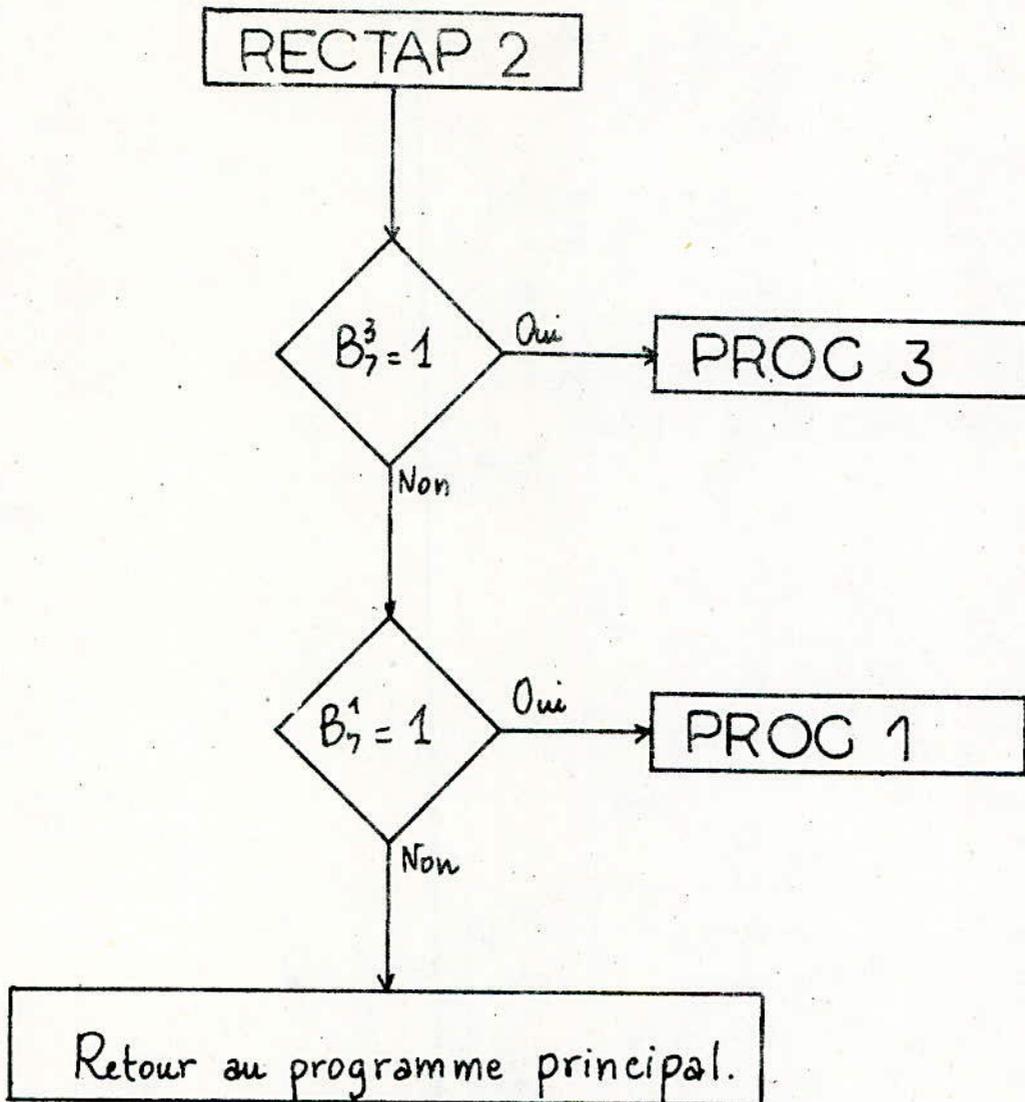
RECTP :

C101	96	65	LDAA	M1
C103	26	09	BNE	Ø 09
C105	96	66	LDAA	M2
C107	26	08	BNE	Ø 08
C109	96	67	LDAA	M3
C10B	26	07	BNE	Ø 07
C10D	3B		RTI	
C10E	7E	C070	JMP	PROG1
C111	7E	C08F	JMP	PROG2
C114	7E	COAE	JMP	PROG3



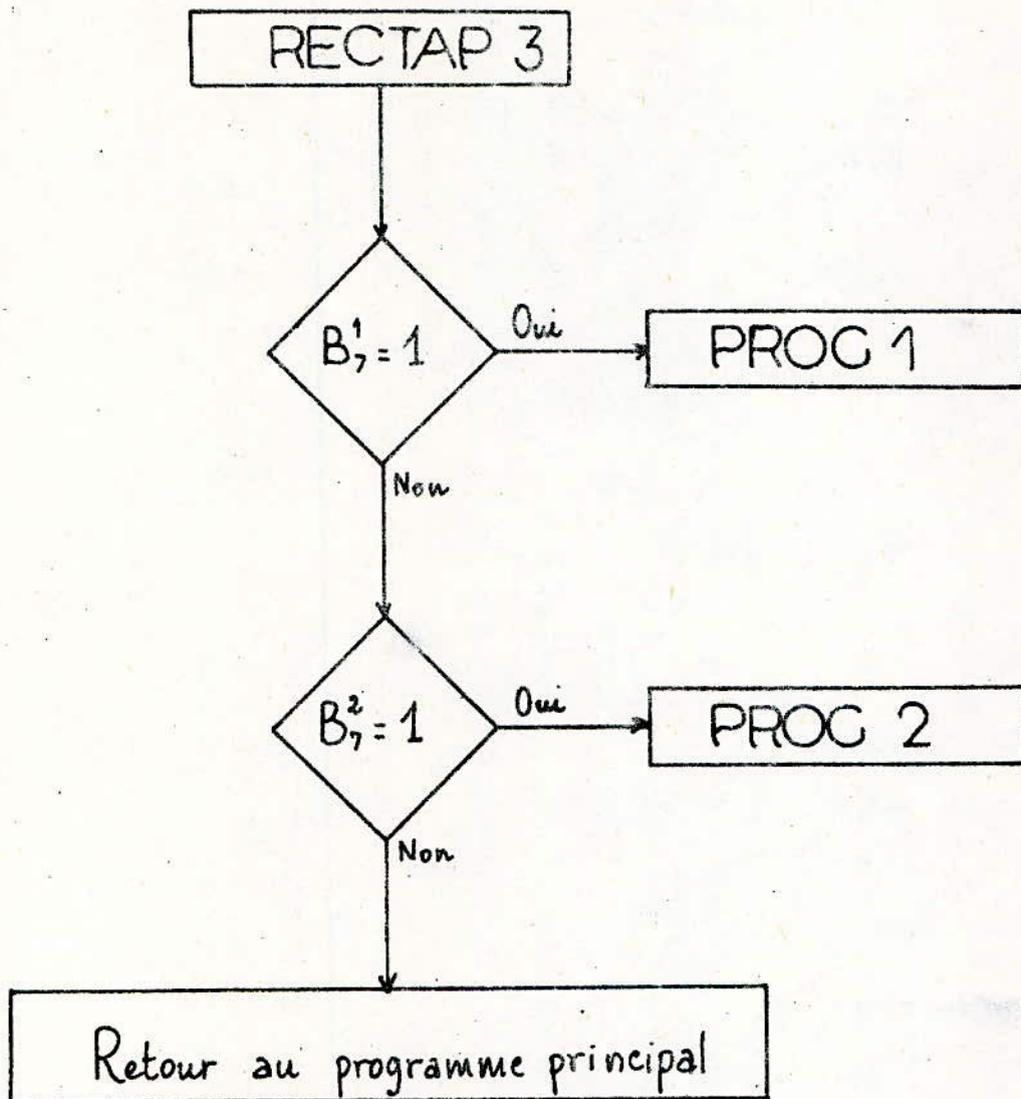
RECTAP1 :

COCE	96	62	LDAA	R2
COD0	49		ROLA	
COD1	25	06	BCS	06
COD3	96	63	LDAA	R3
COD5	49		ROLA	
COD6	25	04	BCS	04
COD8	39		RTS	
COD9	7E	C08F	JMP	PROG2
CODC	7E	COAE	JMP	PROG3



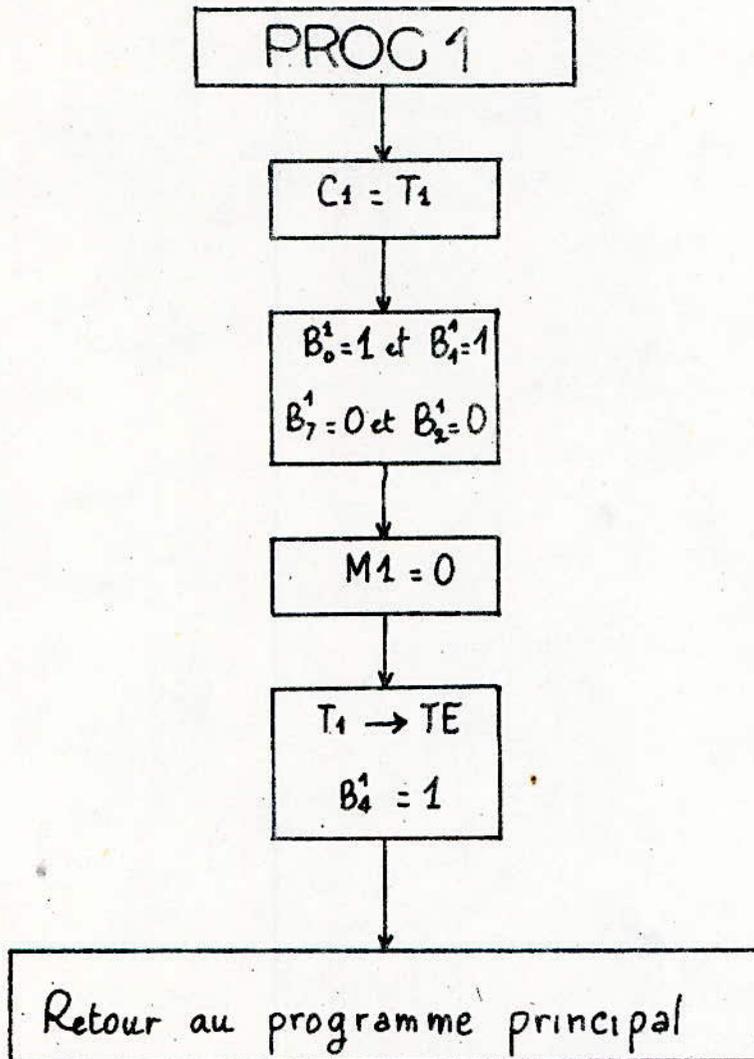
RECTAP2 :

CODF	96	63	LDAA	R3
COE1	49		ROLA	
COE2	25	06	BCS	8 06
COE4	96	61	LDAA	R1
COE6	49		ROLA	
COE7	25	04	BCS	8 04
COE9	39		RTS	
COEA	7E	COAE	JMP	PROG3
COED	7E	C070	JMP	PROG1



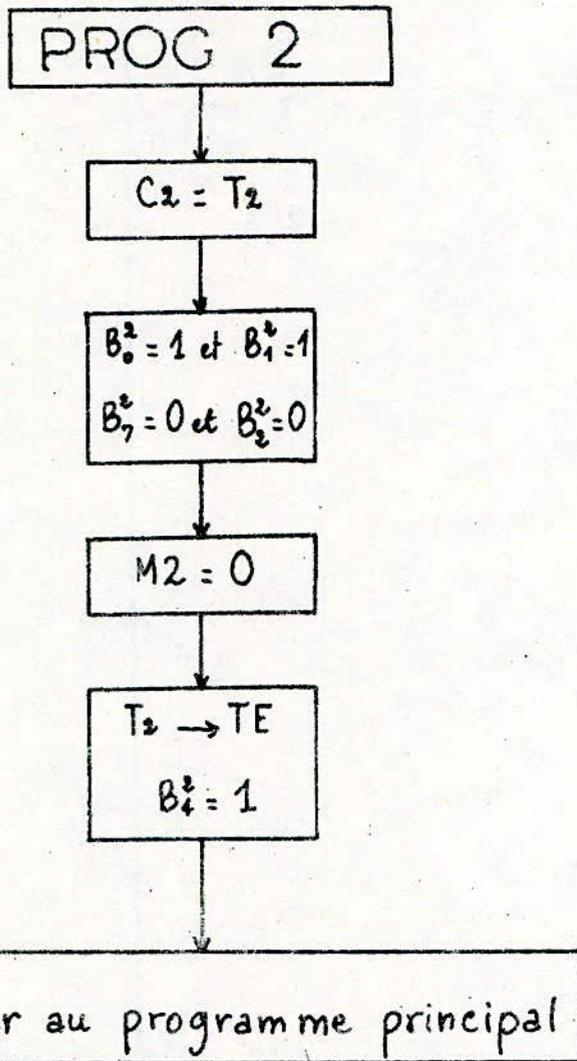
RECTAP3 :

COF0	96	61	LDAA	R1
COF2	49		ROLA	
COF3	25	06	BCS	§ 06
COF5	96	62	LDAA	R2
COF7	49		ROLA	
COF8	25	04	BCS	§ 04
COFA	39		RTS	
COFB	7E	C070	JMP	PROG1
COFE	7E	C08F	JMP	PROG2



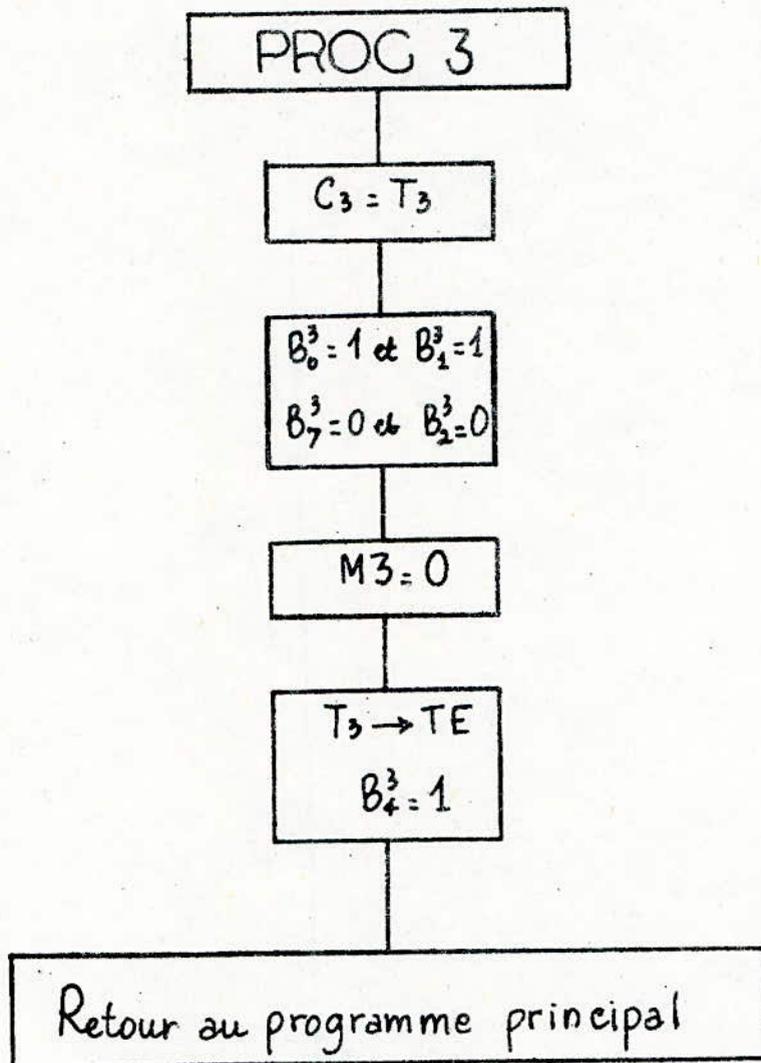
PROG1 :

C070	96	7A	LDAA	ET1
C072	97	68	STAA	EC1
C074	DE	7B	LDX	T1
C076	DF	69	STX	C1
C078	96	61	LDAA	R1
C07A	84	7B	ANDA	<del>##</del> % 01111011
C07C	8A	03	ORAA	<del>##</del> % 00000011
C07E	97	61	STAA	R1
C080	7F	0065	CLR	M1
C083	BD	(XXXX)	JSR	COMINDK (K = 6)
C086	96	61	LDAA	R1
C088	8A	10	ORAA	<del>##</del> % 00010000
C08A	97	61	STAA	R1
C08C	7E	C18F	JMP	% C18F Retour au prog. princip.



PROG2 :

C08F	96	7D	LDAA	ET2
C091	97	6B	STAA	EC2
C093	DE	7E	LDX	T2
C095	DF	6C	STX	C2
C097	96	62	LDAA	R2
C099	84	7B	ANDA	## % 01111011
C09B	8A	03	ORAA	## % 00000011
C09D	97	62	STAA	R2
C09F	7F	0066	CLR	M2
COA2	BD	(XXXX)	JSR	COMINDK (K = 5)
COA5	96	62	LDAA	R1
COA7	8A	10	ORAA	## % 00010000
COA9	97	62	STAA	R1
COAB	7E	C199	JMP	§ C199 Retour au prog.princip.



PROG3 :

COAE	96	80	LDAA	ET3
COB0	97	6E	STAA	EC3
COB2	DE	81	LDX	T3
COB4	DF	6F	STX	C3
COB6	96	63	LDAA	R3
COB8	84	7B	ANDA	<del>##</del> %01111011
COBA	8A	03	ORAA	<del>##</del> %00000011
COBC	97	63	STAA	R3
COBE	7F	0067	CLR	M3
COC2	BD	(XXXX)	JSR	COMINDK (K = 4)
COC5	96	63	LDAA	R3
COC7	8A	10	ORAA	<del>##</del> % 00010000
COC9	97	63	STAA	R3
COCB	7E	C203	JMP	§ C203 Retour au prog. princip.

CHAPITRE 5 : HORLOGE TEMPS REEL : LE  
TEMPORISATEUR PROGRAMMABLE MC 6840

Pour permettre au calculateur de se synchroniser sur le processus à contrôler et d'être immédiatement informé des événements extérieurs susceptibles de se produire, un système d'interruption est utilisé.

Ces interruptions seront générées par un temporisateur programmable, le MC 6840 de MOTOROLA.

Ce chapitre sera une présentation de ce timer.

1- GENERALITES :

Le 6840 est un temporisateur programmable de la famille 6800 qui peut être utilisé comme:

- générateur d'interruptions
- générateur de signaux périodiques: multivibrateur, trains d'impulsions de durée et de période programmable.
- générateur de signaux non périodiques: monostable.
- chronomètre: mesures d'intervalles de temps.
- fréquencemètre: mesure de durée d'impulsions.
- compteur d'événements.

Il comporte essentiellement trois compteurs à 16 bits dont le fonctionnement est commandé par trois registres de commande.

Les trois temporisateurs peuvent fonctionner simultanément, ce qui donne beaucoup de souplesse à ce circuit.

On peut aussi simultanément générer un signal carré, générer un signal unique et faire une mesure de durée.

C'est un circuit 28 broches, monotension (+5V), réalisé en technologie NMOS et entièrement compatible TTL dont les autres caractéristiques essentielles sont:

- fonctionnement à partir de l'horloge du microprocesseur ou d'une horloge externe.

- trois entrées  $\overline{C}$  pour horloges externes et trois entrées  $\overline{G}$  de déclenchement qui sont synchronisées à l'intérieur du temporisateur.

- fréquence maximum externe 4 MHz, uniquement sur le timer n° 3.

- trois sorties masquables.

- les compteurs accessibles par lecture indiquent le temps qui sépare de la fin de la période programmée.

Les échanges avec le microprocesseur se font par l'intermédiaire:

- du bus de données D0-D7 pour programmer les registres de contrôle CR1, CR2 et CR3; les registres tampon LSB et MSB de chaque timer; lire les registres tampon ou le registre d'état.

- de deux lignes de validation de boîtier  $\overline{CS0}$  et CS1 qui permettent l'adressage physique du boîtier.

- de trois entrées de sélection de registre RS0, RS1, RS2 qui permettent de distinguer les registres internes en conjonction avec le fil  $R/\overline{W}$ , puisque 7 registres sont à lecture seule et 9 registres à écriture seule. Ces entrées reçoivent nécessairement les bits A0, A1 et A2 du bus d'adresses pour que le microprocesseur voit le temporisateur comme 8 positions mémoire consécutives.

- de l'entrée ENABLE qui reçoit l'horloge  $\phi 2$  du système, afin de synchroniser les échanges.

- de l'entrée  $R/\overline{W}$  qui fixe le sens des transferts, écriture du temporisateur ou lecture.

- d'une ligne d'interruption  $\overline{IRQ}$  à drain ouvert, donc supportant le OU câblé, et qui permet d'interrompre le programme en cours.

## 2- INTERCONNEXION DU BOITIER ET ADRESSAGE :

La fig 1 montre les interconnexions du temporisateur.

Adressage du temporisateur:

Les trois entrées RSx ne suffisent pas pour adresser les 9 registres à écriture seule, le bit 0 du registre de contrôle 2 (CR 20) permettra de différencier les registres de contrôle 1 et 3 qui ont donc même adresse.

Entrée de selection des registres			Opérations	
RS2	RS1	RS0	R/W = 0	R/W = 1
0	0	0	CR20=0. Ecriture du registre de commande n° 3 (CR 3). CR20=1. Ecriture du registre de commande n° 1 (CR1).	Pas de lecture possible
0	0	1	Ecriture du registre de commande n°2 (CR2)	Lecture du registre d'état
0	1	0	Ecriture du registre tampon MSB 1.	Lecture du compteur MSB 1.
0	1	1	Ecriture du registre tampon LSB 1.	Lecture du compteur LSB 1.
1	0	0	Ecriture du registre tampon MSB 2.	Lecture du compteur MSB 2.
1	0	1	Ecriture du registre tampon LSB 2.	Lecture du compteur LSB 2.
1	1	0	Ecriture du registre tampon MSB 3.	Lecture du compteur MSB 3.
1	1	1	Ecriture du registre tampon LSB 3.	Lecture du compteur LSB 3.

### ADRESSAGE DES REGISTRES INTERNES DU TEMPORISATEUR

Les 7 autres registres à lecture seule, auront même adresse mais sont différenciés par R/W qui sera alors à 1 (lecture). Les adresses étant consécutives, RS0, RS1 et RS2 reçoivent respectivement A0, A1 et A2 du bus d'adresses.

## 3- ORGANISATION INTERNE DU TEMPORISATEUR :

Il comprend essentiellement trois compteurs à 16 bits, pouvant fonctionner en 2 x 8 bits, ce qui permet de générer des signaux de rapport cyclique variable.

Les données sont transférées des registres tampon dans les compteurs proprement dits lors d'un cycle d'initialisation des compteurs.

Les compteurs sont décrémentés à chaque impulsion d'horloge (interne ou externe). Suivant le mode de fonctionnement spécifié, le compteur s'arrête ou recommence un nouveau cycle lorsqu'il arrive à zéro.

Trois registres de commande définissent le mode de fonctionnement de chacun des compteurs: mode astable, mode monostable, comparateur de fréquence, comparateur de largeurs d'impulsions. Ces registres de commande sont accessibles par un ordre d'écriture comme pour une position mémoire, par l'intermédiaire du bus de données.

Un registre d'état à lecture seule nous fournit les indications d'interruption de chacun des temporisateurs (interruptions indépendantes) et de n'importe lequel d'entre eux.

### 3.1- Registres de commande :

Ils précisent le mode de fonctionnement de chaque temporisateur. Ce sont des registres à écriture seule, de 8 bits. Tous les bits homologues ont la même signification dans chacun des registres excepté pour le bit n° 0. De plus, comme la table des adresses le montre, seul le registre de commande n° 2 est adressable directement; pour écrire dans les registres de commande n° 1 et n° 3, il faut auparavant positionner le bit 0 du registre de commande n° 2.

Signification et programmation des bits CRx1 à CRx7 (x: n° du registre considéré):

- CRx1 définit si l'on utilise une horloge externe, appliquée sur l'entrée Cx correspondante, ou l'horloge du micro-système.

CRx1 = 0 utilisation d'une horloge externe

CRx1 = 1 utilisation de l'horloge du microprocesseur

- CRx2 définit si le compteur correspondant fonctionne sur

16 bits ou sur 2 x 8 bits .

CRx2 = 0 compteur 16 bits

CRx2 = 1 compteur 2 fois 8 bits

- CRx6 valide ou non les interruptions.

CRx6 = 0 n'autorise pas l'envoi d'IRQ

CRx6 = 1 autorise l'envoi d'un IRQ

- CRx7 valide ou non la sortie correspondante.

CRx7 = 0 sortie masquée

CRx7 = 1 sortie validée

Les bits CRx5 , CRx4 , CRx3 définissent le mode de fonctionnement du temporisateur avec :

CRx3	CRx4	CRx5	Mode de fonctionnement
0	0	0	Multivibrateur astable
0	0	1	Monostable
1	0	0	Comparaison de fréquences
1	1	0	Comparaison de largeurs d'impulsions.

Le bit indiqué "0" est utilisé pour modifier l'initialisation du compteur et sa validation, ou les conditions d'interruption.

Signification et programmation de CRx0 :

- Registre de commande 1 : CR10 bit de remise à zéro interne avec :

CR10 = 0 tous les temporisateurs sont autorisés à fonctionner

CR10 = 1 tous les temporisateurs sont figés dans leur état présent.

- Registre de commande 2 : CR20 permet l'adressage des registres de contrôle 1 OU 3 avec :

CR20 = 0 accès à CR3

CR20 = 1 accès à CR1

- Registre de commande 3 : CR30 précise le facteur de division de l'horloge du temporisateur 3 avec :

CR30 = 0 facteur de division :1

CR30 = 1 facteur de division :8

Ceci autorise une horloge externe de fréquence maximum 4 MHz.

### 31.1- Mode multivibrateur astable :

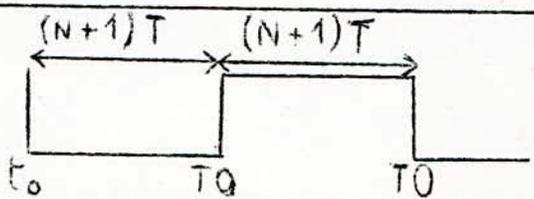
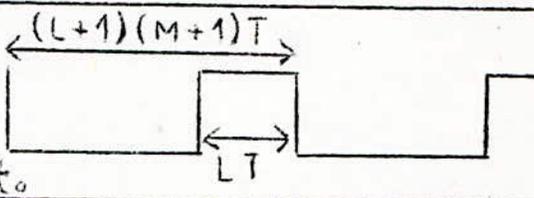
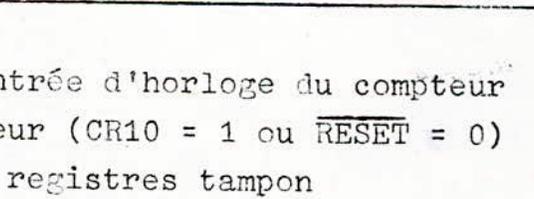
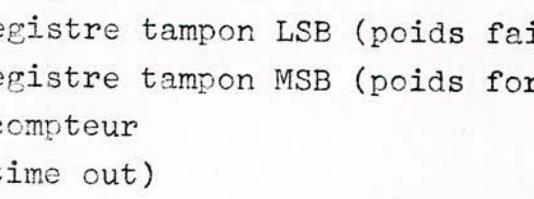
On obtiendra sur la sortie Ox un signal carré si l'on travaille sur 16 bits ou un signal rectangulaire en 2 fois 8 bits.

L'initialisation du compteur peut se faire par remise à zéro du temporisateur des trois façons suivantes:

- niveau bas sur  $\overline{\text{RESET}}$
- CR10 = 1
- transition descendante sur l'entrée GATE :  $\overline{\text{Gx}} \downarrow$ .

D'autre part si CRx4 = 0 on aura une initialisation du compteur à chaque commande d'écriture dans le registre tampon.

Mode astable CRx3 = 0 et CRx5 = 0

CRx2	CRx4	Initialisation du compteur	Signal en sortie si CRx7 = 1
0	0	$\overline{\text{G}} \downarrow$ ou W ou R	
0	1	$\overline{\text{G}} \downarrow$ ou R	
1	0	$\overline{\text{G}} \downarrow$ ou W ou R	
1	1	$\overline{\text{G}} \downarrow$ ou R	

T : transition négative sur l'entrée d'horloge du compteur

R : remise à zéro du temporisateur (CR10 = 1 ou  $\overline{\text{RESET}} = 0$ )

N : donnée sur 16 bits dans les registres tampon

L : donnée sur 8 bits dans le registre tampon LSB (poids faibles)

M : donnée sur 8 bits dans le registre tampon MSB (poids forts)

to : cycle d'initialisation du compteur

TO : fin du temps de comptage (time out)

#### REMARQUES :

- Il est indispensable pour le fonctionnement des compteurs que l'entrée  $\overline{\text{GATE}}$  soit maintenue à l'état bas.
- Toujours charger les registres MSB avant les registres LSB.

### 31.2- Mode monostable :

Dans ce cas après le premier time out la sortie reste à l'état bas jusqu'au prochain cycle d'initialisation.

Mode monostable CRx3 = 0 et CRx5 = 1

CRx2	CRx4	Initialisation du compteur	Signal en sortie si CRx7=1
0	0	$\bar{G} \downarrow$ ou W ou R	
0	1	$\bar{G} \downarrow$ ou R	
1	0	$\bar{G} \downarrow$ ou W ou R	
1	1	$\bar{G} \downarrow$ ou R	

Chaque time out du compteur positionne à "1" l'indicateur d'interruption et reinitialise le compteur.

La validation du compteur est indépendante de l'entrée  $\bar{G}$ , il suffit d'y appliquer une transition négative pour déclencher le monostable.

$L = M = 0$  ou  $N = 0$  inhibe la sortie .

### 3.2- Registre d'état : (Status register)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
-----	-----	-----	-----	-----	-----	-----	-----

C'est un registre à lecture seule comportant quatre indicateurs d'interruptions.

- les bits SR0 , SR1 et SR2 sont respectivement affectés aux temporisateurs 1 , 2 et 3 en tant que bits d'interruption indépendants.

- le bit SR7 est le bit d'interruption commun aux trois temporisateurs, il est positionné à un lorsque n'importe quel bit indicateur indépendant le sera, à condition que

le bit d'interruption correspondant (CRx6) valide les interruptions.

Un indicateur d'interruption est remis à zéro soit par un niveau actif sur la ligne RESET soit par CR10 = 1 . IL peut aussi être remis à zéro par une lecture du compteur du temporisateur à condition que le registre d'état ait été lu auparavant et l'indicateur d'interruption positionné.

### 3.3- Initialisation des registres :

Les compteurs sont chargés par les données binaires contenues dans les registres tampons. Le transfert se produit lors de l'initialisation des compteurs. Comme les compteurs sont à 16 bits et le bus de données à 8 bits, il est nécessaire de stocker dans un buffer les 8 bits de plus fort poids ; ceux-ci seront transférés dans le registre tampon correspondant lorsqu'on écrira les 8 bits de poids faible.

C'est pour cela que l'on doit d'abord écrire le MSB puis le LSB. Cet ordre est le même pour la lecture.

L'initialisation d'un compteur peut se faire :

- en appliquant un niveau actif sur RESET ou CR10 = 1 .
- lors d'une commande d'écriture des registres tampons .
- par application d'une transition descendante sur l'entrée GATE .

### 3.4- Entrées asynchrones et sorties :

#### 34.1- Entrées horloges externes C1 , C2 , C3 :

Ces entrées sont compatibles avec les niveaux TTL. La fréquence appliquée peut aller du continu à la fréquence d'horloge appliquée sur ENABLE. Trois périodes de l'horloge du microprocesseur sont nécessaires pour synchroniser et traiter l'horloge externe.

L'horloge du temporisateur 3 peut être divisée par 8, mais elle est traitée de façon identique à C1 ou C2.

#### 34.2- Entrées GATE G1 , G2 , G3 :

Ces entrées sont compatibles avec des signaux TTL non synchrones de l'horloge du microprocesseur.

Trois impulsions d'horloge sont nécessaires pour les prendre en compte. Ces entrées sont directement liées aux compteurs 16 bits.

34.3- Sorties des temporisateurs 01 , 02 , 03 :

La sortance de chaque circuit est de deux charges TTL. Seuls les fonctionnements en mode astable ou monostable permettent d'obtenir des signaux définis. En mode chronomètre ou fréquencemètre, des signaux apparaissent en sortie si CRx7 = 1 mais leur forme est imprévisible.

4- MODE INTERVALLE DE TEMPS :

Dans ce mode les indicateurs d'interruptions indépendants sont fonction à la fois du time out des compteurs et de l'entrée  $\overline{\text{GATE}}$ . Dans chacun de ces modes le signal de sortie n'est pas défini mais malgré cela le compteur peut travailler soit en 16 bits soit en 2x8 bits.

Un front descendant sur l'entrée  $\overline{\text{G}}$  active le compteur et commence un cycle d'initialisation. Le compteur est alors décrémenté à chaque coup d'horloge pendant ou après l'initialisation du compteur et jusqu'à ce qu'une interruption soit engendrée.

PROGRAMMATION DU REGISTRE DE CONTROLE :

CRx3	CRx4	CRx5	Application	Condition de positionnement du flag d'interruption.
1	0	0	Comparaison de fréquence	Une interruption est engendrée si la période de l'entrée GATE est inférieure au time out du compteur.
1	0	1	Comparaison de fréquence	Une interruption est engendrée si la période de l'entrée GATE est supérieure au TO du compteur
1	1	0	Comparaison de largeurs d'impulsions	Une interruption est engendrée si la durée de l'état bas sur l'entrée GATE est inférieure au TO du compteur.
1	1	1	Comparaison de largeurs d'impulsions	Une interruption est générée si la durée de l'état bas sur l'entrée GATE est supérieure au TO du compteur.

#### 4.1- Mode mesure de période :

CRx3 = 1 , CRx4 = 0

Suivant CRx5 , l'indicateur indépendant d'interruption sera positionné si le time out du compteur s'écoule avant l'arrivée du deuxième front descendant sur  $\bar{G}$  ou bien l'inverse.

Si CRx5 = 1 , les signaux sont ceux de la figure 2.

Si  $TO > T$  , l'indicateur Ix restera à zéro, le deuxième front descendant du signal d'entrée reinitialisant le compteur.

Si CRx5 = 0 , les signaux sont ceux de la figure 3.

Si  $TO < T$  , l'indicateur Ix restera à zéro et le compteur recommence son cycle de décomptage.

#### 4.2- Mode comparaison de largeurs d'impulsions :

CRx3 = 1 , CRx4 = 1

Une transition sur l'entrée  $\bar{G}$  lance le compteur. Le compteur est décrémenté à chaque impulsion d'horloge et sera bloqué sur un front montant du signal d'entrée.

- Si CRx5 = 1 , l'interruption est générée si la durée de l'état bas sur l'entrée  $\bar{G}$  est supérieure au TO du compteur.
- Si CRx5 = 0 , l'interruption est générée si la durée de l'état bas sur l'entrée  $\bar{G}$  est inférieure au TO du compteur.

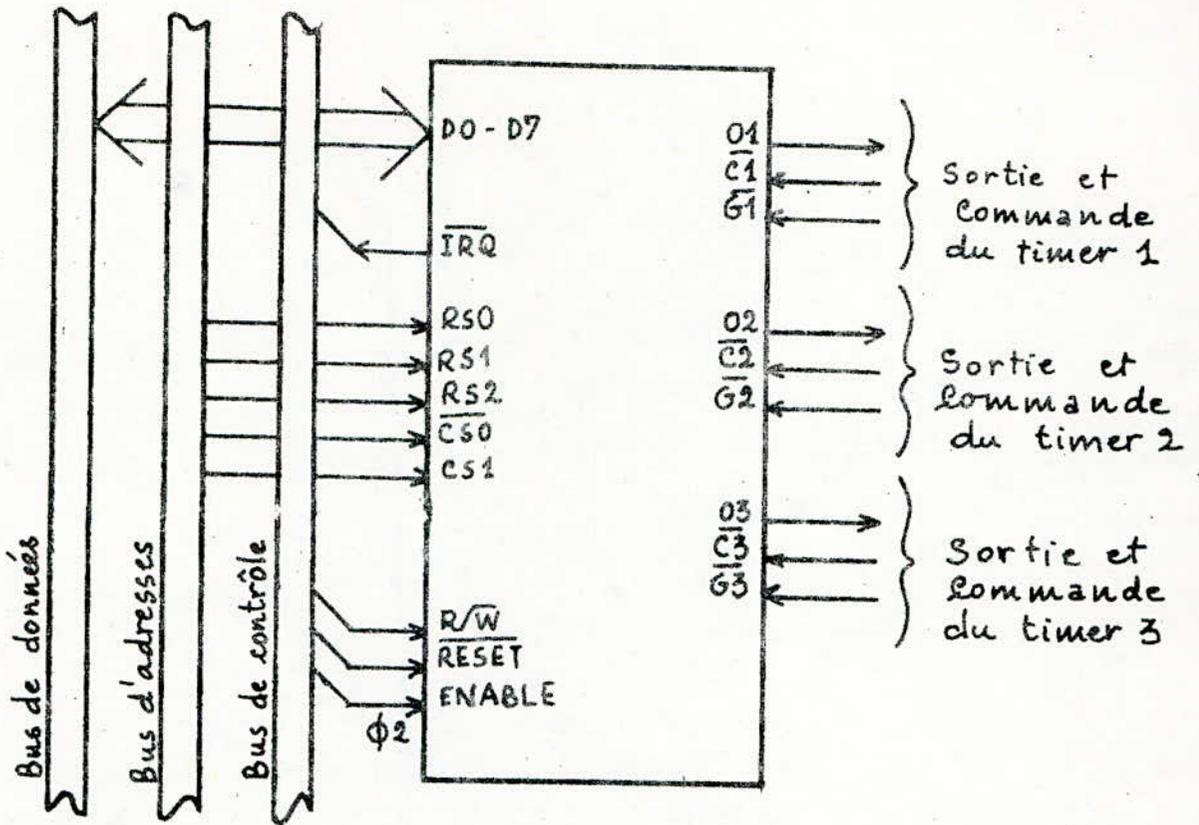


fig 1: INTERCONNEXION DU TIMER 6840.

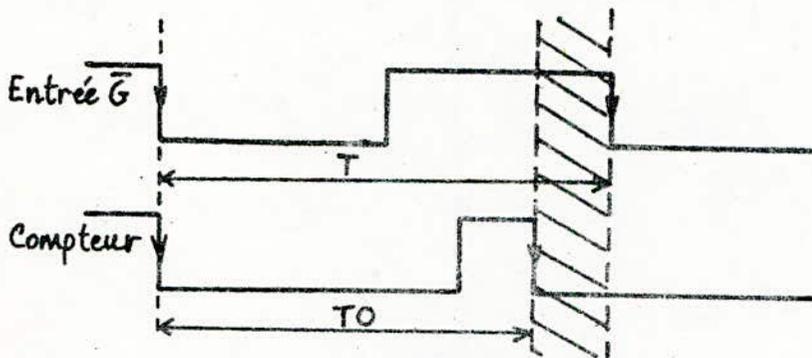


fig 2: MESURE DE DUREE PLUS GRANDE QUE LE  $T_0$ .

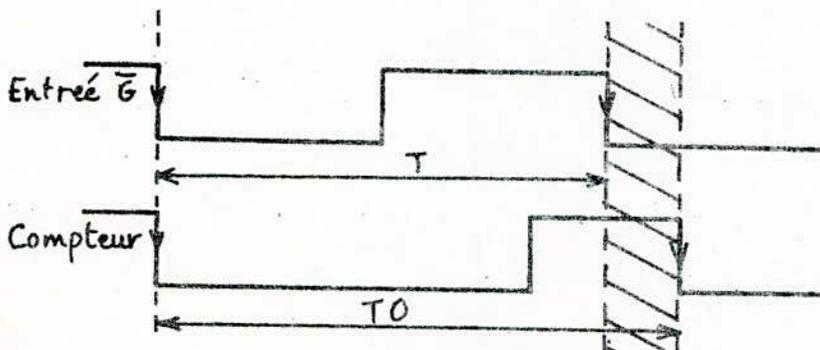


fig 3: MESURE DE DUREE PLUS PETITE QUE LE  $T_0$ .

Le concept de travail en temps réel correspond à la nécessité pour le calculateur d'élaborer ses résultats dans des temps cohérents avec l'évolution du processus contrôlé.

Pour cela des interruptions sont utilisées. Celles-ci seront générées par un temporisateur programmable, le MC6840 de MOTOROLA, permettant une grande souplesse d'utilisation.

#### 1- LOGIQUE D'ADRESSAGE :

Le timer occupant huit positions mémoire, la sélection des registres internes se fait par l'intermédiaire des trois lignes RS0 , RS1 et RS2 ; de la ligne  $R/\bar{W}$  du système pour la lecture ou l'écriture et du bit zéro du registre de contrôle CR2 .

Les lignes RS0 , RS1 et RS2 seront donc reliées aux lignes de plus faible poids du bus d'adresses soit respectivement A0 , A1 et A2 (fig 2).

Le signal d'horloge  $\phi_2$  permet la synchronisation des transferts entre le timer et le CPU.

#### 2- BRANCHEMENT DES TIMERS :

Nous utiliserons le timer n°3 en mode multivibrateur sur 16 bits pour servir d'horloge, intermédiaire, au timer n°2. Sur le timer n°3 nous avons la possibilité de prédiviser par huit le signal d'horloge, ce qui permet d'avoir une large gamme de fréquences sur l'horloge intermédiaire.

Le timer n°2 sera programmé en mode multivibrateur sur 2 fois 8 bits pour avoir en sortie des signaux à rapport cyclique variable.

Une configuration du branchement de ces timers est représentée à la figure 2.

### 3- PROGRAMMATION DES TIMERS :

Le timer n°3 délivre un signal carré de période variable.

Sa période sera :  $2(N + 1)T$ , où N est le contenu du registre interne du timer sur 16 bits et T la période de l'horloge du système.

La sortie sera validée et la ligne d'interruptions masquée. On utilisera l'horloge  $\phi_2$  du système.

Le registre de commande du timer n°3 aura donc la configuration suivante :  $CR3 = 10010010$

Le timer n°2 travaillera en multivibrateur sur 2 fois 8 bits et la période du signal à sa sortie sera :  $(M + 1)(L + 1)T'$  où M est l'octet de poids forts du registre du timer et L celui des poids faibles. T' est la période du signal en sortie du timer n°3.

La figure 3 donne la forme des signaux en sortie de ces deux timers.

Le signal obtenu en sortie du timer n°2 sera inversé puis envoyé, vers le MPU, sur la ligne  $\overline{IRQ}$  du système qui est active à l'état bas.

### 4- EXEMPLE DE MISE EN OEUVRE DU TEMPORISATEUR :

Nous désirons avoir un signal de période 1 ms.

Programmation du timer n°3 :

prenons  $N = 24$ , ce qui donne  $T' = 2(N + 1)T = 50 \mu s$ .

Nous aurons ainsi une horloge intermédiaire de période

$T' = 50 \mu s$ , qu'utilisera le timer n°2.

Programmation du timer n°2 :

prenons  $M = 9$  et  $L = 1$

nous aurons  $T'' = (M + 1)(N + 1)T' = 1\text{ms}$ .

$T''$  étant la période du signal en sortie du timer n°2.

Le démarrage du comptage s'effectuera en réalisant un reset interne en mettant le bit CR10 à 1 .

CR1 = 00000001

Le comptage démarrera juste après l'écriture de CR1.

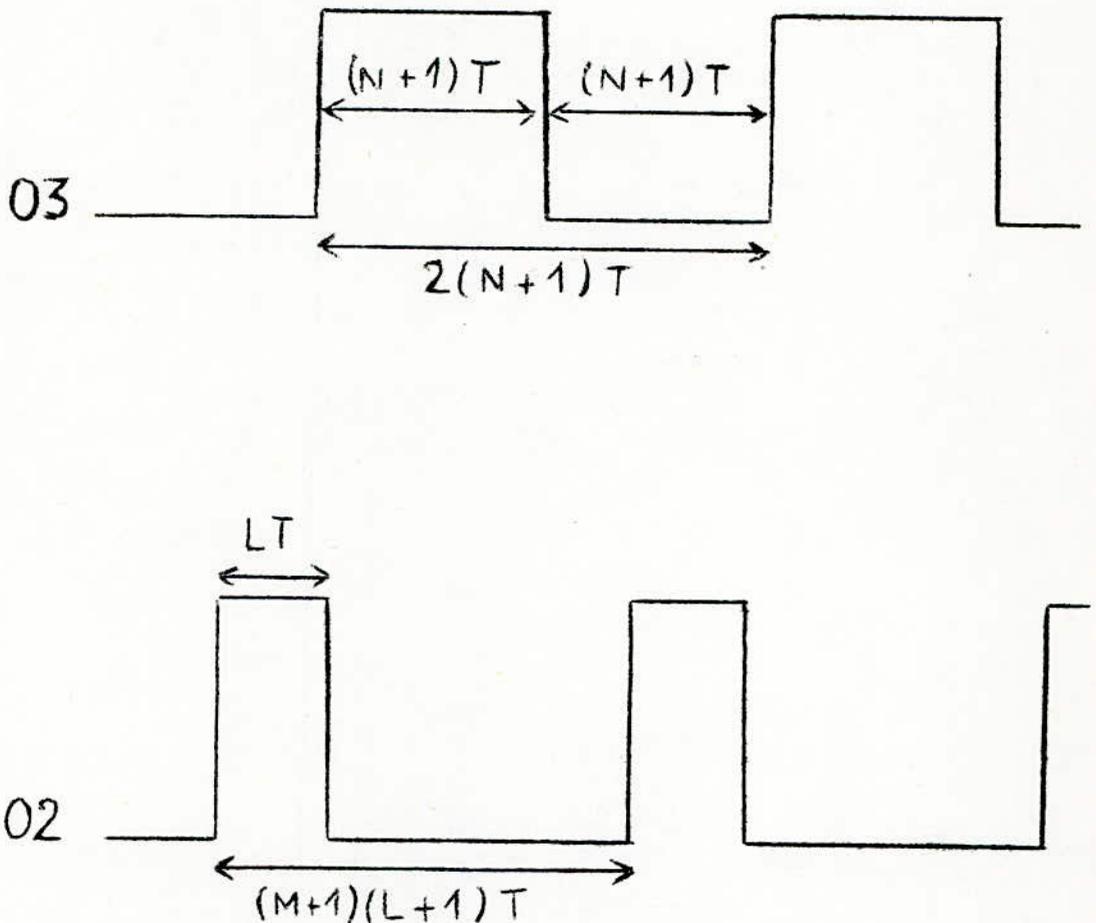


Fig 3 : Signaux aux sorties des timers n°2 et n°3 .

## 5- PROGRAMME DE L'HORLOGE :

### Adresses des registres internes du timer:

8800	TIMCR1	EQU	§ 8800	Registre de commande du timer 1.
8801	TIMCR2	EQU	§ 8801	Registre de commande du timer 2.
8800	TIMCR3	EQU	§ 8800	Registre de commande du timer3.
8802	TMREG1	EQU	§ 8802	Registre de chargement du timer1
8804	TMREG2	EQU	§ 8804	Registre de chargement du timer2
8806	TMREG3	EQU	§ 8806	Registre de chargement du timer3

### Parametres:

008E	EQU	§ 92	CR3: mode 16 bits, horloge interne
008F	EQU	§ 95	CR2: 2x8 bits, horloge externe, accès à CR1
0090	EQU	§ 01	CR1: reset interne
0091	EQU	§ 00	N: MSB demi periode de l'horloge intermediaire
0092	EQU	§ 18	N: LSB
0093	EQU	§ 09	M: poids forts de TMREG2
0094	EQU	§ 01	L: poids faibles de TMREG2

### Programme de l'horloge:

C20D	7F	8801	CLR	TIMCR2	accès à CR3
C210	96	8E	LDAA	§ 8E	pog. de CR3
C212	B7	8800	STAA	§ 8800	
C215	96	8F	LDAA	§ 8F	PROG. de CR2 avec accès à CR1
C217	B7	8801	STAA	§ 8801	
C21A	DE	91	LDX	§ 0091	chargement de TMREG3
C21C	FF	8806	STX	§ 8806	
C21F	DE	93	LDX	§ 0093	chargement de TMREG2
C221	FF	8804	STX	§ 8804	
C224	96	90	LDAA	§0090	démarrage du comptage
C226	B7	8800	STAA	§ 8800	
C229	3F		SWI		software interrupt

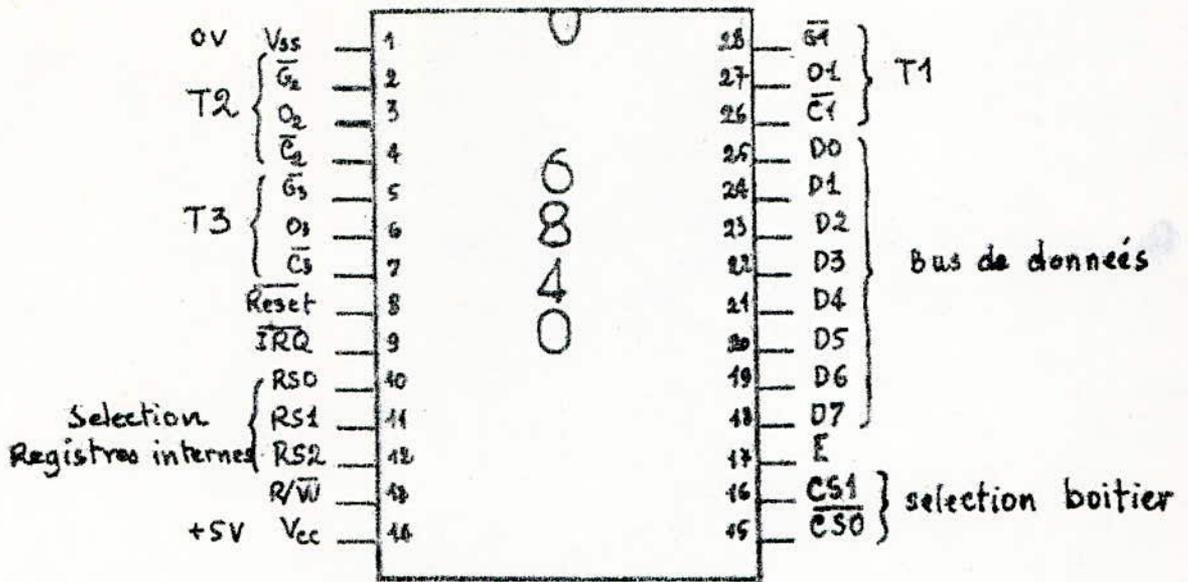


fig 1: Brochage du 6840

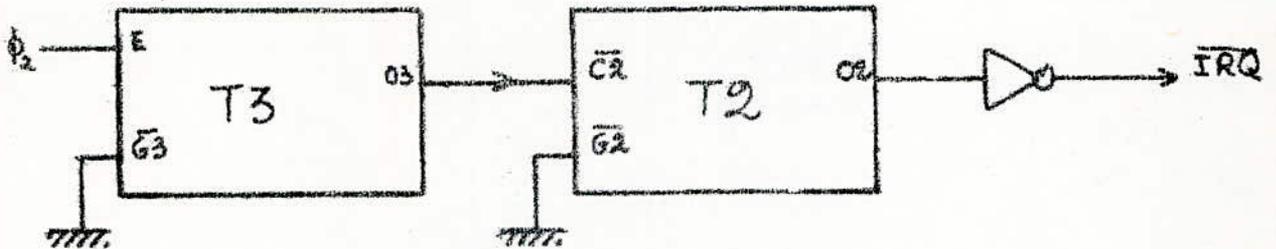


fig 2: Configuration de branchement des timers internes du 6840

## CHAPITRE 7 : REALISATION D'UNE CARTE EPROM

Cette carte EPROM est destinée à servir de support hardware au logiciel du système.

Celle-ci comportera 16 K octets de mémoire morte et le temporisateur programmable MC 6840. Elle sera wrappée sur une carte standard MOTOROLA.

Les 16 K de mémoire se répartissent comme suit : 8 K occupant la zone de A000 à BFFF et les autres 8 K de D000 à EFFF. La zone médiane étant occupée par le moniteur du système.

### 1- ETUDE DE L'EPROM 2708 :

On utilisera sur la carte seize boîtiers EPROM 2708 pour obtenir les 16 K de mémoire voulus.

L'EPROM 2708 est une mémoire effaçable par ultra-violets et reprogrammable électriquement. Elle est utilisable pour la mise au point de systèmes et pour des applications similaires demandant une mémoire non volatile qui doit être reprogrammée.

La fenêtre transparente sur le boîtier permet d'effacer aux rayons ultra-violets le contenu de la mémoire.

Les principales caractéristiques de cette mémoire sont les suivantes :

- organisée en 1024 octets (mots de 8 bits)
- fonctionnement statique
- tensions d'alimentation standards : +12V, +5V et -5V.

- temps d'accès maximum : 450 ns.
- entrée de selection de boitier pour l'extension de la mémoire.
- compatible TTL .
- sorties trois états .

## 2- ETUDE DES CIRCUITS UTILISES :

- Le bus d'adresses est amplifié par trois circuits amplificateurs unidirectionnels du type 8T95. Ils amplifient également les lignes VMA et  $\phi$  2 du système. On utilisera la ligne BA (Bus available) du système pour la validation de ces circuits.

- Le bus de données doit être amplifié dans les deux sens, écriture et lecture, puisqu'on écrit dans le temporisateur MC6840 et on lit dans les EPROM ou le temporisateur. Pour cela on utilise deux circuits 8T26, amplificateurs bidirectionnels. Ces circuits sont validés dans le sens lecture si un zéro logique est présent sur le pin1 et dans le sens écriture si un un logique est présent sur le pin 15. La logique de validation de ces circuits est représentée sur le schéma global de la carte.

- Puisque les 16 K de mémoire sont répartis en deux zones , on utilisera deux décodeurs , un pour chaque partie. Les décodeurs utilisés sont des 7442, décodeurs 1 parmi 10. On utilisera seulement huit des dix sorties de chacun des deux décodeurs.

- L'horloge programmable MC6840 est également wrappée sur cette carte. Son brochage et le branchement des différents timers est clairement exposés dans les chapitres précédents.

Sur le schéma global de la carte figurent toutes les logiques d'adressage de ces circuits ainsi que leur brochage.

### 3- OBTENTION DU -5V :

La carte MOTOROLA utilisée ne comporte pas d'alimentation -5V, nécessaire aux EPROM 2708. Par contre on trouve sur la carte une alimentation en -12V. Il suffira donc d'abaisser et de stabiliser cette dernière. Pour cela on utilise un montage dit DARLINGTON. (le schéma du dit montage se trouve sur le schéma global de la carte).

Les transistors utilisés sont des PNP puisque les tensions sont négatives.

#### Détermination de la capacité de l'alimentation :

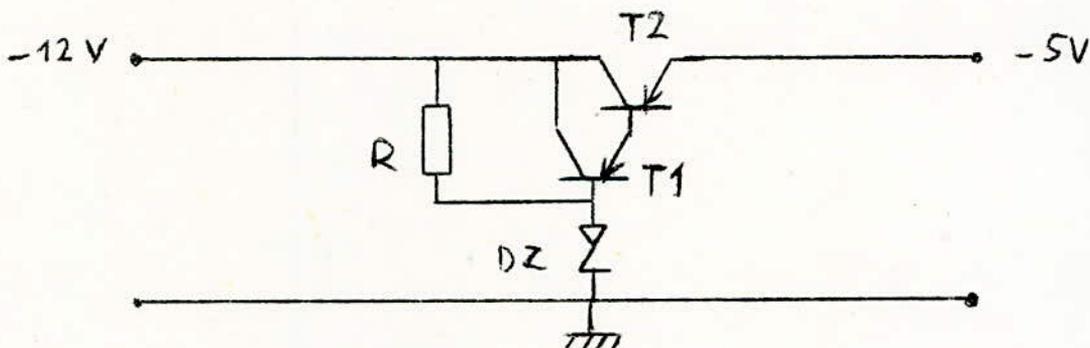
Le courant maximal sur l'entrée -5V des EPROM est de 45 mA.

Soit pour 16 boitiers :

$$45 \times 16 = 720 \text{ mA.}$$

Nous calculerons notre alimentation pour un débit maximal de 1 ampère. Le transistor de puissance utilisé doit donc pouvoir supporter un tel courant.

#### Schéma de principe du montage :



Le transistor de puissance (T2) utilisé est un 2N4919 , débitant un courant maximal de 3 A et ayant un  $\beta$  compris entre 30 et 150.

Le transistor driver ( T1) est un 2N4032, ayant un  $\beta$  minimal de 40.

## C O N C L U S I O N

La présente étude m'a permis d'aborder le domaine des micro-systèmes qui m'était inconnu, de définir les systèmes logiques à micro-processeurs, d'étudier les constituants d'un tel système ainsi que la façon de le réaliser.

Dans la mise au point des micro-processeurs on distingue deux parties : une partie software, abordée lors de l'étude du gestionnaire et une partie hardware abordée lors de la réalisation de la carte EPROM.

L'emploi des micro-processeurs et des micro-ordinateurs, de plus en plus performants, est tout à fait judicieux pour la commande d'un processus. Cet outil informatique présente aussi l'avantage de demander un investissement financier limité.

## BIBLIOGRAPHIE

- Daniel-Jean DAVID et Rodney ZAKS "Programmation du 6800" Edition Sybex 1981.
- D. GIROD et R. DUBOIS "Au cœur des microprocesseurs" Edition Eyrolles 1979.
- MICRO SYSTEMES N° 15 Janvier/Février 81.
- SESCOSEM "Microprocesseur SF.F96800. Manuel de programmation" .
- EFCIS "Microprocesseurs et mémoires" 1980.
- Edward A. PARRISH and Victor K.L. HUANG "A scheduler for real time task control in microcomputers" .
- MOTOROLA Inc. "Microprocessor course" .
- MOTOROLA Inc . "Microcomputer components" .

# SCHEMA DE LA CARTE EPROM

