

Travaux dirigés
de
microprogrammation
sur

MITRA

15

A. GUYOT



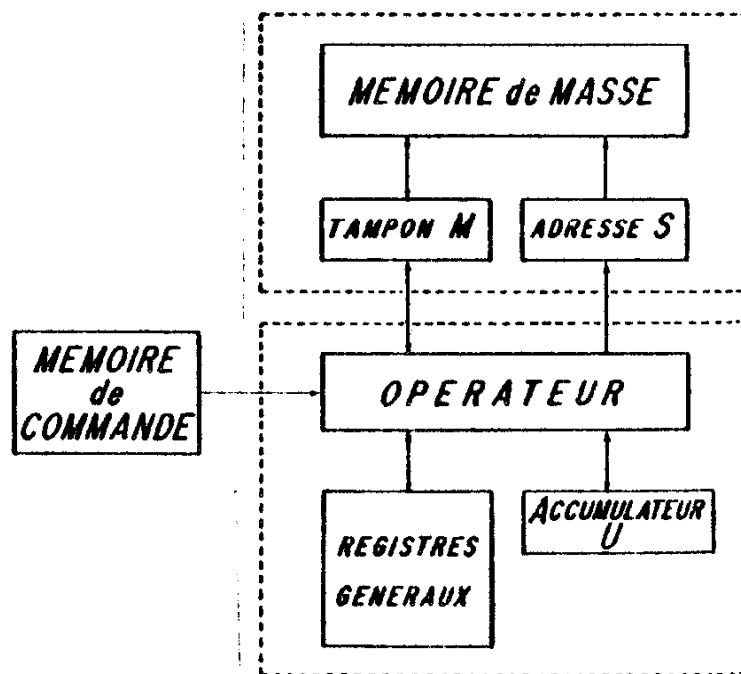
I. <u>Minimachine fonctionnelle</u>	1
I.1.1 Opérateur	1
I.1.2 Registres généraux	1
I.1.3 Compteur Ordinal	2
I.1.4 Mémoire de commande	2
I.1.5 Séquencement	2
I.2 Mémoire de masse	2
I.3 Mini-instruction	3
I.3.1 Découpage de la mini-instruction	4
I.3.2 Types de mini-instructions	4
I.3.3 Zones de la mini-instruction	5
I.3.4 Branchements	6
I.4 Schémas de principe	10
I.4.1 Partie opérative (planche 1)	10
I.4.2 Partie contrôle (planche 2)	10
II. <u>Micromachine</u>	11
II.1 Parties opératives	11
II.1.1 Description de l'opérateur	11
II.1.2 Registres	12
II.1.3 Indicateurs de microprogramme	13
II.1.4 Indicateurs de programme	13
II.2 Partie commande	14
II.2.1 Mémoire des exécutifs	14
II.2.2 Découpage d'un exécutif	14
II.2.3 Commande du multiplexeur μ	15
II.2.4 Commande du multiplexeur γ	15
II.2.5 Commandes du multiplexeur α	15
II.2.6 Commandes du multiplexeur β	15
II.2.7 Commandes de l'opérateur et de la retenue entrante	17
II.2.8 Commandes des transferts de σ	18
II.2.9 Commandes de modification de T	18
II.2.10 Commandes de modification de J	19
II.2.11 Commandes de modification des indicateurs de programme	19
II.2.12 Commandes de modification des indicateurs de microprogramme	20
II.2.13 Commandes des périphériques	20
II.2.14 Commandes du système d'interruption et suspension	20

II.3	Synoptique de la micromachine (planche 3)	22
II.4	Format des exécutifs (planche 4)	23
III.	<u>Macromachine et minimachine</u>	24
III.1	Macromachine	24
III.2	Mode d'adressage des instructions	24
III.3	Liste des instructions	25
III.4	Réalisation physique de la macromachine	26
III.5	Algorithme d'interprétation des instructions	26
III.6	Liste des exécutifs utilisés dans la macromachine	28
IV.	<u>Interruption et suspensions</u>	31
IV.1	Suspensions	31
IV.1.1	Mémoire associée aux suspensions	31
IV.1.2	Communications Micromachine-Périphérique	32
IV.2	Interruptions	32
IV.3	Format du mot de M.A.S (planche 5)	34
IV.4	Schéma du système d'interruption et suspension (planche 6)	34

I. MINIMACHINE FONCTIONNELLE

La minimachine est un petit ordinateur assez spécial. Ses éléments de mémorisation peuvent être séparés en deux classes : la mémoire de commande, qui contient les instructions, appelées mini-instructions d'une part, et les registres généraux la mémoire de masse et quelques registres associés qui contiennent les données.

La mémoire de masse peut être considérée comme un périphérique de la minimachine qui communique avec celle-là par l'intermédiaire d'un registre tampon appelé M et d'un registre adresse appelé S. Il y a, de plus, dans la minimachine, un registre accumulateur appelé U. Une opération diadique s'effectue entre M ou U comme premier opérande, un registre général comme deuxième opérande, et le résultat est rangé dans ce registre général, ou dans M, S ou U.



I.1.1 Opérateur

Physiquement, l'opérateur peut exécuter un très grand nombre de fonctions différentes, très simples (plusieurs milliers). Ces fonctions seront appelées microfonctions. Seul un petit nombre de ces microfonctions, quatre vingt huit, ont été sélectionnées pour former un jeu standard de microfonctions. On peut modifier ce jeu standard pour ajouter des microfonctions spécifiques à un problème.

I.1.2 Registres généraux

D'un bloc de soixante quatre registres, on tire huit registres généraux. Dans une mini-instruction, le numéro du registre opérande ou résultat

est donné par une zone de trois bits, appelée zone r de la mini-instruction. Pour une opération de registre général à registre général cette zone indique un registre qui est à la fois opérande et résultat.

I.1.3 Compteur ordinal

Il existe dans la machine un registre spécial qui contient en permanence l'adresse de la mini-instruction en cours d'exécution. Nous appellerons T, ce registre compteur ordinal, qui a onze bits.

I.1.4 Mémoire de commande

Elle est organisée en trente deux pages de trente deux mots, soit mille vingt quatre mots de seize bits. Elle contient des mini-instructions dont les séquences forment des mini-programmes.

I.1.5 Séquencement

Chaque mini-instruction comprend des informations qui permettent de calculer l'adresse de la mini-instruction suivante. La distinction habituelle entre opération et branchement n'est pas faite dans la minimachine. Le regroupement de ces deux fonctions dans une mini-instruction permet d'augmenter la rapidité d'exécution, et de diminuer la taille des mini-programmes.

I.2 Mémoire de masse

La mémoire de masse ne contient que des données. Elle peut être considérée comme un périphérique de la minimachine car la mémoire et la minimachine travaillent simultanément, donc plus rapidement.

L'unité élémentaire d'information de la mémoire est le mot. Chaque mot comprend 18 bits dont 16 d'information, 1 pour le contrôle de parité, 1 pour la protection mémoire.

Sa capacité peut varier par blocs de 4 k-mots jusqu'à 32 k-mots.

La mémoire fonctionne par demi-cycle séparé (c'est-à-dire que le cycle de réécriture n'est pas lancé immédiatement après le cycle de lecture).

Pour lire un mot, il faut effectuer un demi cycle de lecture avec effacement suivi d'un demi-cycle de réécriture. Pour écrire, il faut, après un demi-cycle d'effacement effectuer un demi-cycle d'écriture.

Il y a trois commandes de la mémoire :

- Demande Mémoire (DM)

A la fin de l'exécution d'une mini-instruction, il y a chargement du registre S, remise à zéro du registre M et envoi d'une demande d'accès en lecture à la mémoire. Si la mémoire est libre, elle commence automatiquement un cycle de lecture qui efface la case mémoire à l'adresse S et remplit le registre M avec le contenu de cette case mémoire.

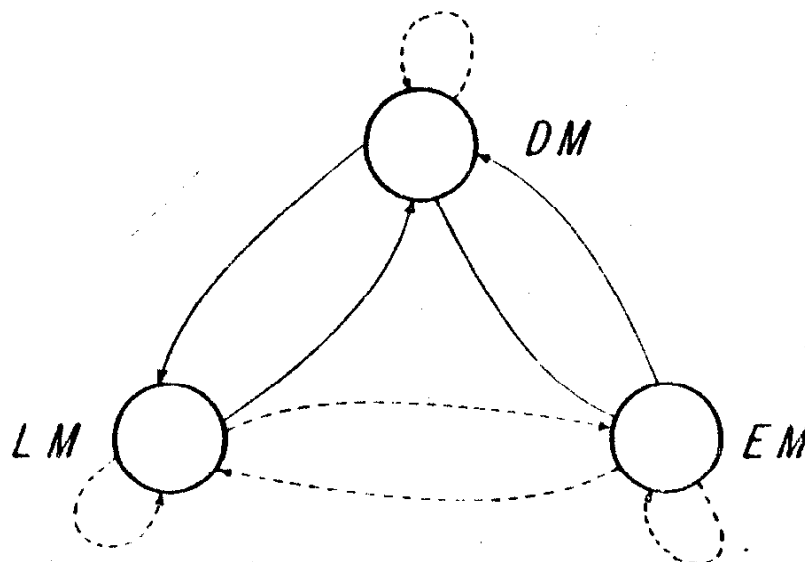
- Lecture Mémoire (LM)

Si le registre M a été garni, cette action lance un cycle de réécriture de la mémoire et force le transfert du registre M.

- Ecriture Mémoire (EM)

A la fin de l'exécution d'une mini-instruction, cette action accomplit le transfert de la sortie de l'opérateur dans le registre M. En même temps, elle lance le cycle d'écriture de la mémoire.

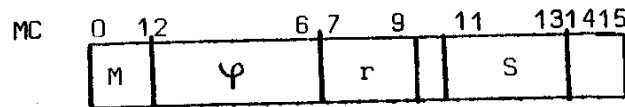
Si une demande de mémoire suit une autre demande, le fonctionnement mémoire fait qu'elle n'est pas prise en compte. De même pour une lecture et écriture. Les transitions non prises en compte sont indiquées en pointillé sur la figure.



1.3 Mini-instruction

Les mini-instructions sont classées en 6 types, notées 0 à 5, qui correspondent à des découpages en champs et à des significations des champs différentes. Ces types se distinguent par la valeur du code de la microfonction notée ϕ .

I.3.1 Découpage de la mini-instruction



- Zone M : commande de la mémoire
- Zone φ : code de la micro-fonction
- Zone r : $\begin{cases} \text{soit : Zone registre} \\ \text{soit : } \varphi_c : \text{extension éventuelle de } \varphi \end{cases}$
- Zone S : extension éventuelle de φ
- MC10 à MC15 : Signification variable suivant φ .

Quel que soit le type de la mini-instruction, les bits MC0, MC1 auront la même signification : c'est la commande de la mémoire. Leur notation et leur action est la suivante :

MC0	MC1		
1	1		aucune action
0	1	DM	demande d'accès ($\sigma \rightarrow S$, demi-cycle de lecture)
1	0	LM	lecture ($M \rightarrow \mu$, 1/2 cycle d'écriture)
0	0	EM	écriture ($\sigma \rightarrow M$, 1/2 cycle d'écriture)

σ est le résultat de l'opération exécutée par la mini-instruction en cours, c'est-à-dire la sortie de l'opérateur, et μ vient remplacer U comme opérande, c'est-à-dire que l'entrée de l'opérateur normalement connectée à U est connectée à M dans le cas d'une lecture.

Dans tous les types de mini-instructions, les bits MC2 à MC6 représentent le code de la microfonction φ .

I.3.2 Types de mini-instruction

Dans ce tableau sont regroupés les découpages et signification des zones des mini-instructions, ainsi que les valeurs des codes φ .

Type	découpage	Registre	Valeur de φ	φ complément.	Valeur Immédiate	Modificat de T
0	<div> <div>0 2 6 9 15</div> <div> <div>M</div> <div>φ</div> <div>r</div> <div>mt</div> </div> </div>	r	$4 \leq \varphi < 28$	-	-	Longue (6 bits)
1	<div> <div>0 2 6 9 11 13 15</div> <div> <div>M</div> <div>φ</div> <div>r</div> <div>S</div> <div></div> </div> <div>mt</div> </div>	r	$28 \leq \varphi < 32$	S	-	Courte (3 bits)
2	<div> <div>0 2 6 9 15</div> <div> <div>M</div> <div>φ</div> <div>φ_c</div> <div>mt</div> </div> </div>	-	$0 \leq \varphi < 4$	r	-	Longue (5 bits)
3	<div> <div>0 2 6 15</div> <div> <div>M</div> <div>φ</div> <div>I</div> </div> </div>	-	$4 \leq \varphi < 28$	-	Longue (9 bits)	Implicite (+ 1)
4	<div> <div>0 2 6 9 15</div> <div> <div>M</div> <div>φ</div> <div>φ_c</div> <div>I</div> </div> </div>	-	$0 \leq \varphi < 4$	r	Courte (6 bits)	Implicite (+ 1)
5	<div> <div>0 2 6 9 15</div> <div> <div>M</div> <div>φ</div> <div>r</div> <div>mt</div> </div> </div>	r	$4 \leq \varphi < 28$	-	-	Spéciale (6 bits)

Remarque : - signifie absence de la zone.

Les types 0 et 5 ont le même découpage, mais la zone modification de T (m t) n'a pas la même signification.

I.3.3 Zones de la mini-instruction

Zone r : Lorsque cette zone est présente, elle désigne soit un registre, soit une bascule dans le cas de tests.

Zone φ : Les valeurs de φ , code de la microfonction, données dans le tableau ne sont pas suffisantes pour définir le type, cependant pour chaque type la valeur de φ est nécessairement entre les bornes données. Nous verrons plus loin quels critères définissent le type d'une mini-instruction.

Zone φ_c : La correspondance entre la valeur du code φ et la position du champ φ_c a été choisie de manière à faciliter le décodage.

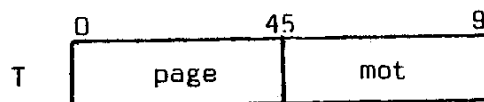
En effet, si φ est compris entre 0 et 4, les trois premiers bits de φ valent 0. Si φ est compris entre 28 et 32, les trois premiers bits de φ valent 1. Dans les autres cas, c'est-à-dire si φ est compris entre 4 et 28, les trois premiers bits ne sont pas tous égaux.

Zone valeur Immédiate : (notée I dans le tableau). Dans la mini-instruction de type 3, les bits MC_{7-15} , complétés à gauche par des zéros sont envoyés dans l'opérateur. Dans le type 4, ce sont les bits MC_{10-15} .

I.3.4 Branchements

Un branchement est une modification de la valeur du compteur ordinal T de la mini-machine. Nous ne nous intéresserons qu'aux dix derniers bits de T, notés de 0 à 9.

La mémoire de contrôle étant divisée en trente deux pages de trente deux mots, les cinq bits poids fort de T indiquent la page, tandis que les cinq bits poids faible indiquent le mot dans cette page.



Comme on le verra, on ne modifie que certains bits de T, les autres conservent évidemment leur valeur.

Cette modification peut être de plusieurs types. Les différents types sont :

- modification implicite

La valeur de T est obtenue par incrémentation de 1 modulo 16.
 $(T+1) \text{ modulo } 16 \rightarrow T$.

Seuls les quatre bits poids faible de T sont modifiés, les autres restant inchangés. Cette modification ne permet pas de changer de page.

On trouve les mini-instructions de type 3 et de type 4 qui produisent cette modification.

De plus les mini-instructions de test font soit la modification implicite, soit la modification normale suivant le résultat du test.

- modification normale

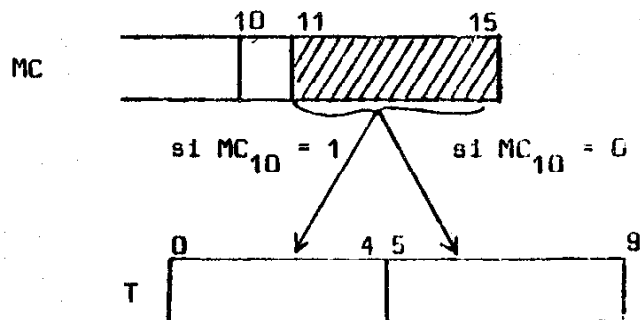
On distingue 2 types de modification normale : court et long.

Les informations servant à la modification viennent de la mini-instruction en cours d'exécution.

Type long (5 bits + 1)

Si $MC_{10} = 0$ $MC_{11-15} \rightarrow T_{5-9}$, T_{0-4} inchangé

Si $MC_{10} = 1$ $MC_{11-15} \rightarrow T_{0-4}$, T_{5-9} inchangé



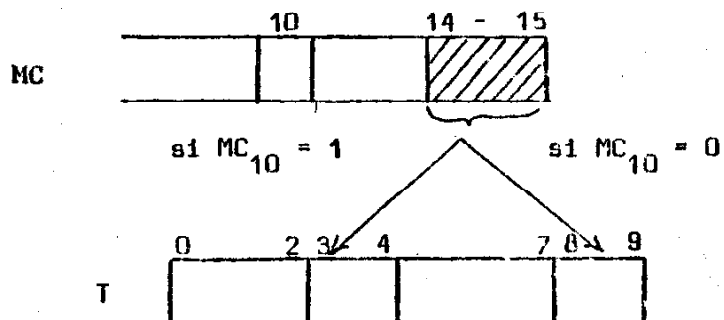
On trouve les mini-instructions de type 0 et 2

Cette modification de T permet soit de changer de mot, en restant dans la même page, soit de changer de page, mais en restant à la même adresse de mot.

Type court (2 bits + 1)

Si $MC_{10} = 0$ $MC_{14-15} \rightarrow T_{8-9}$, T_{0-7} inchangé

Si $MC_{10} = 1$ $MC_{14-15} \rightarrow T_{3-4}$, T_{0-2} et T_{5-9} inchangé



elle correspond aux microinstructions de type 1.

Cette modification de T est une restriction de la précédente.

Il existe cependant un cas particulier :

Si $MC_{10} = 0$ et que $MC_{11-15} = T_{5-9}$ dans la type 1

ou que $MC_{14-15} = T_{8-9}$ dans les types 0 et 2

le compteur ordinal T est remis à zéro.

Cela permet le regroupement sur un point commun situé au mot d'adresse 0 de la page 0, à partir d'une mini-instruction située n'importe où.

- modification spéciale

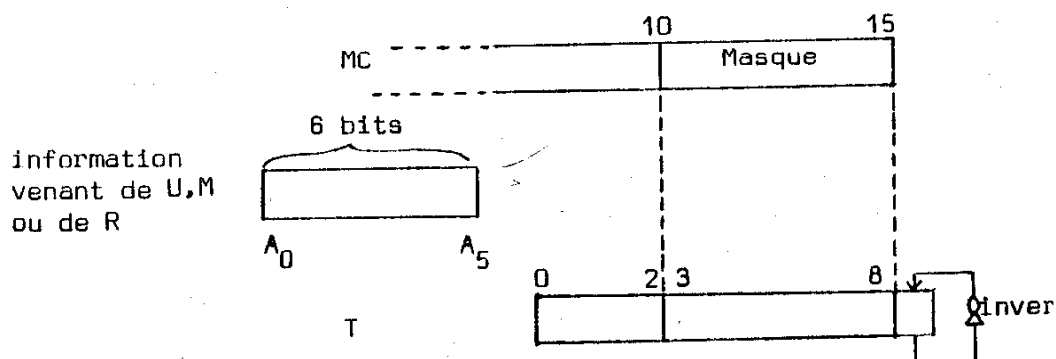
C'est l'éclatement qui se trouve dans les mini-instructions de type 5.

Cette modification porte sur les bits 3 à 9 du compteur ordinal T, les bits 0 à 2 restant inchangés.

La nouvelle valeur des bits 3 à 9 est calculée à partir de l'ancienne valeur, d'une information venant de l'accumulateur U, ou de M, ou d'un registre général, et d'un masque donné par les bits MC₁₀ à MC₁₅ de la mini-instruction.

Dans le cas ou on utilise un registre général, son adresse est déterminée par le champ r.

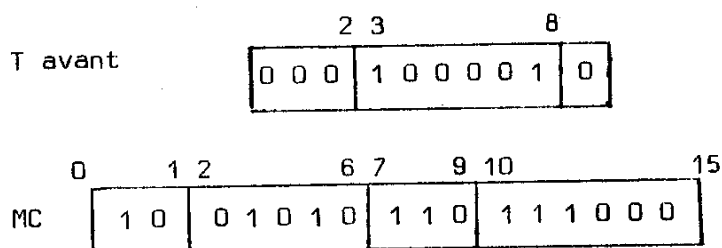
L'algorithme de modification en pseudo-algol est le suivant



```

pour I := 0 jusqu'à 5 faire
    si MC(10+I) = 1 alors T(3+I) := A(I) ;
si T(9) = 1 alors T(9) := 0 sinon T(9) := 1 ;
    
```

Exemple : Soit



En supposant que la micro-fonction commande la modification à partir des bits 0 à 5 d'un registre contenant 5 0 1 4 base 16.

Cela nous conduit à la valeur de T suivante :

				2	3					8	9	
T	0	0	0	0	1	0	0	0	1	1		

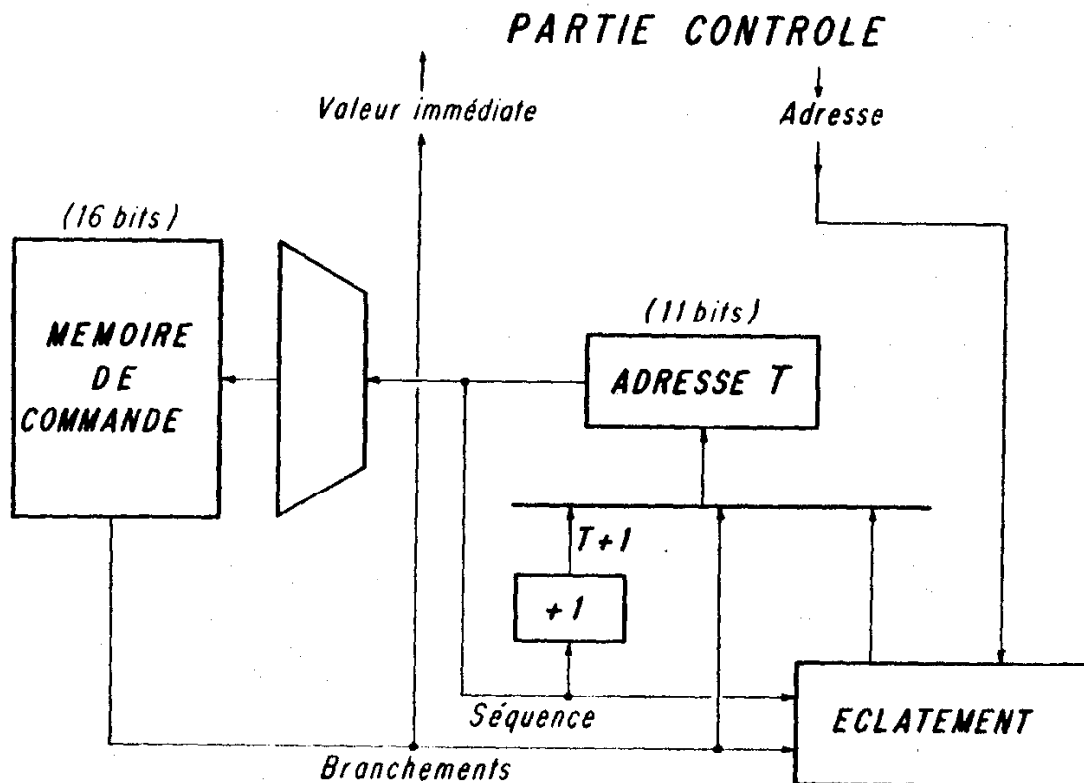
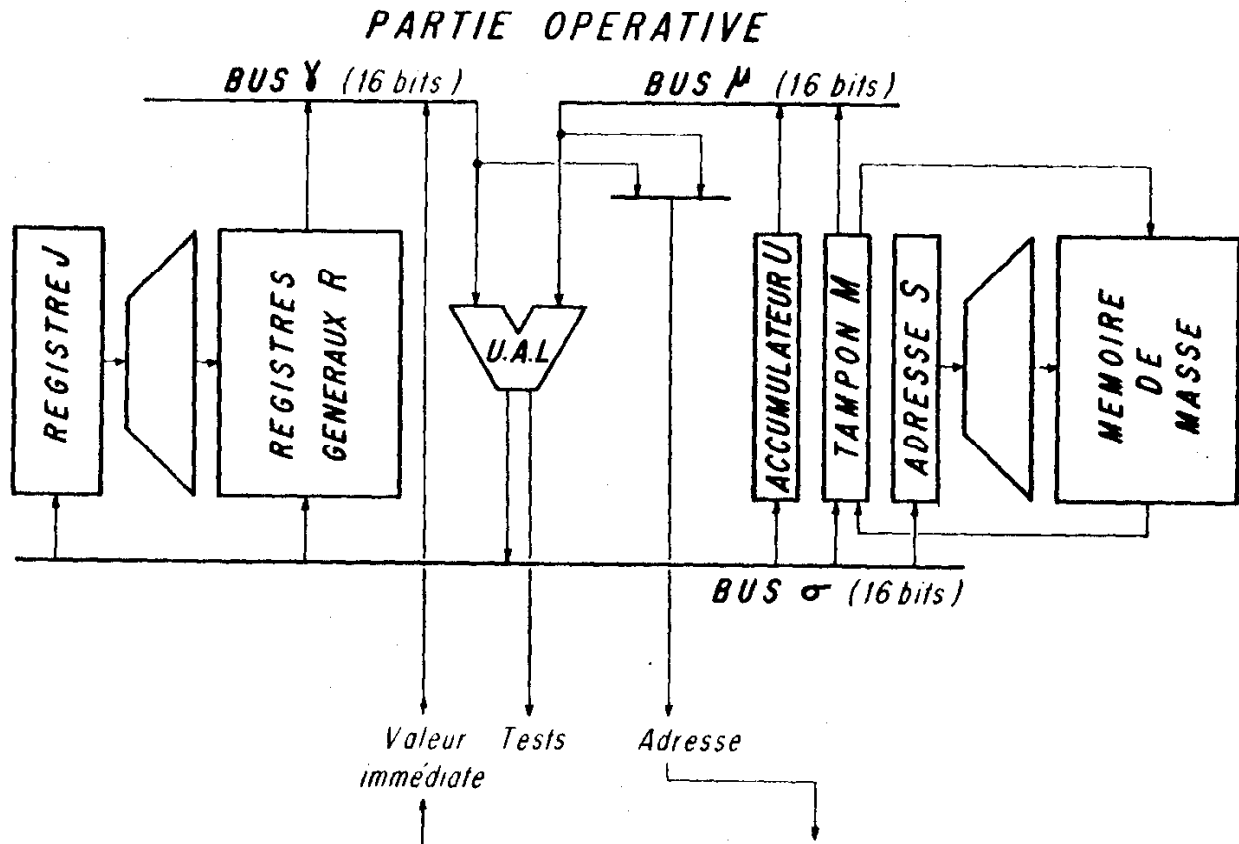
Ce branchement permet de faire un décodage, et le masque associé à cette modification de choisir parmi les six bits de l'information à décoder.

On inverse le neuvième bit pour ne rien perdre des différentes valeurs de T. On a n bits à un dans le masque, on peut donc décoder deux puissance n valeurs, donc pouvoir se brancher à deux puissance n adresses. Parmi ces adresses il y en a une qui serait forcément déjà prise si on n'inversait pas le dernier bit.

I.4 Schémas de principe

I.4.1 Partie opérative (planche 1)

I.4.2 Partie contrôle (planche 2)



II. MICROMACHINE

Nous appellerons Micro-machine la réalisation cablée de la mini-machine. Dans ce chapitre sera donnée une description de cette réalisation cablée qui permet de comprendre comment sont réalisées les microfonctions, et donc d'augmenter ou modifier la liste des microfonctions disponibles.

La planche 3 (page 22) peut être divisée de haut en bas en trois ensembles :

- la mémoire de masse et ses registres d'accès
- la partie opérative avec ses registres et indicateurs
- la mémoire de commande, la mémoire des exécutifs et le séquençement.

II.1 Partie Opérative

Elle comprend :

- un opérateur et les circuits associés
- des registres généraux et autres
- des indicateurs.

II.1.1 Description de l'opérateur

L'opérateur utilisé travaille en parallèle sur 16 bits.

Il reçoit en entrée

- . 2 informations de 16 bits : α et β
- . 1 information de 1 bit : CIN (retenue entrante)

Il délivre en sortie

- . 1 information de 16 bits : σ
- . 1 information de 1 bit : COUT (Report)

Cet opérateur est formé de deux paires de circuits intégrés SN 74181 qui traitent chacun huit bits. Les huit bits poids faible et huit bits poids fort peuvent être traités différemment. L'opérateur fonctionne suivant deux modes :

- . logique :

Il réalise les 16 fonctions booléennes de deux variables ($\sigma = \varphi(\alpha, \beta)$). Dans ce mode, la CIN n'intervient pas.

- . Arithmétique :

Il réalise alors 16 fonctions des 3 variables α , β , CIN.

La valeur de l'opérande α peut venir :

- soit d'un périphérique (voir chapitre 4)
- soit de gamma directement ($\gamma_{0-15} \rightarrow \alpha_{0-15}$)
- soit de gamma inversé ($\gamma_{0-7} \rightarrow \alpha_{8-15}$, $\gamma_{8-15} \rightarrow \alpha_{0-7}$)
- soit des indicateurs de microprogramme.

Le choix d'une provenance d'un opérande constitue également une fonction de la micromachine.

La valeur de l'opérande γ peut venir

- soit d'un registre général R, dont l'adresse est r
- soit de la mémoire de commande dans le cas d'une valeur immédiate.

La valeur de l'opérande β est fonction de μ . Sont réalisées un grand nombre de fonctions qui ne sont pas toutes représentées sur le synoptique.

La valeur de l'opérande μ peut venir

- soit du registre accumulateur U
- soit du registre mémoire M.

La sortie de l'opérateur : σ peut aller :

- soit vers le registre S
- soit vers le registre U
- soit vers un registre général R, dont l'adresse est r.

II.1.2 Registre

Registre S :

Il sert à adresser les mots mémoire, il ne possède que 15 positions binaires et reçoit les 15 bits poids fort de la sortie de l'opérateur. Le bit de poids faible sert à positionner une bascule (S_0) qui permet de traiter l'octet par microprogramme.

Registre M :

Il possède 16 positions binaires. C'est le registre de donnée de la mémoire (lecture, écriture).

Registre U :

16 positions binaires. C'est l'accumulateur de la micro-machine. Ses poids forts peuvent être remis à zéro indépendamment de ses poids faibles. ($0 \rightarrow U_{0-7}$, $\sigma_{8-15} \rightarrow U_{8-15}$)

Registres généraux :

Ils sont au nombre de 64 en 8 blocs de 8 registres de 16 bits chacun.

Ils sont adressés par une fonction du contenu du registre J et de 3 bits du champs r de la mémoire de commande. Ils peuvent être étendus jusqu'à 32 blocs de 8 registres soit 256 registres.

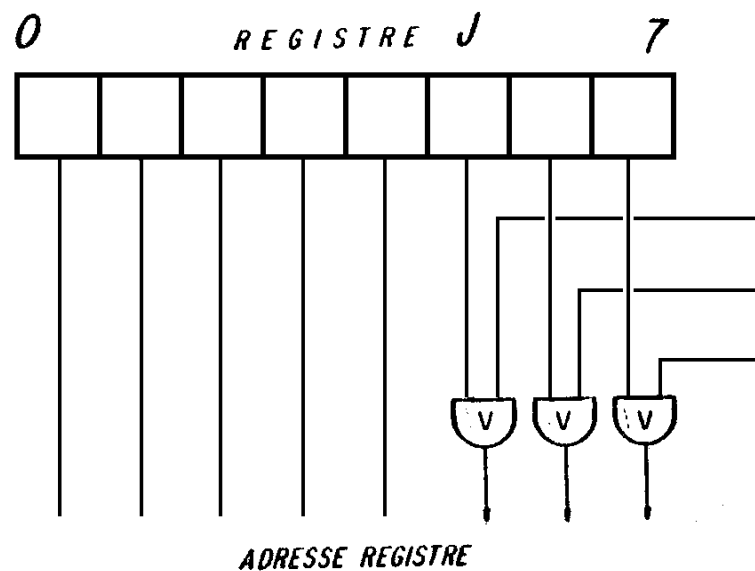
Registre J :

Sert à adresser un registre général, possède 8 positions binaires :

L'adresse du registre étant définie de la façon suivante :

$$A_r = J_{2-4} \times 8 + (r \vee J_{5-7}) \quad (\vee = \text{"ou logique"})$$

où r est une information provenant d'une microinstruction.



II.1.3 Indicateurs de microprogramme

Ils sont au nombre de 4

B : la bascule B sert dans les rotations, décalages, addition avec retenue, etc...

S0 : Select-octet, indique la parité de l'octet traité. Elle est positionnée, lors du chargement de S par le bit poids faible de la sortie de l'opérateur, ou bien peut être forcée de l'extérieur.

SZ : Elle est positionnée à la fin de chaque microinstruction. Elle indique que le résultat de l'opérateur est nul (ZERO).

SN : Elle est également positionnée à la fin de chaque microinstruction. Elle indique que le résultat de l'opérateur est inférieur à zéro (NEGATIF).

II.1.4 Indicateurs de Programme

Ils sont au nombre de 5. Ils peuvent être positionnés et testés par l'utilisateur de la micromachine.

PM : Cette bascule inhébe, quand elle est à 1, la protection mémoire.
Elle peut être positionnée par σ_{11} , 11ème bit de la sortie de l'opérateur σ .

MA : Sert à masquer les interruptions. Elle peut être positionnée par σ_{12} .

MS : Indique le mode maître ou esclave. Elle peut être positionnée par σ_{13} .

I1 : Indicateur général peut recevoir :

- la même valeur que SZ quand SZ est chargé.
- la retenue sortante de l'UAL (COUT)
- σ_{14} 14ème bit de σ .

I2 : peut recevoir :

- la même valeur que SN
- le résultat du calcul de débordement (*qui est câblé*)
- σ_{15} , 15ème bit de σ
- la valeur de la bascule B

II.2 Partie Commande

Elle fait le lien entre le code φ , éventuellement étendu du code φ_c , et les commandes des diverses fonctions de la micro-machine, appelées micro-fonctions au chapitre précédent.

Elle est formée d'une mémoire dite mémoire des exécutifs et d'une certaine logique de décodage qui n'est pas représentée sur la planche 3.

II.2.1 Mémoire des exécutifs

C'est une mémoire à lecture non destructive de 96 mots de 48 bits.
On a 88 exécutifs adressables par microinstructions. Les 8 supplémentaires sont uniquement accessibles en mode maintenance (du pupitre).
C'est cette mémoire qui fournit les commandes de micro-machine.
L'adresse A_x d'un exécutif est calculée à partir de φ et de φ_c par un organe câblé..

	Types
Si $0 \leq \varphi < 4$ alors $A_x = 32 + \varphi \times 8 + \varphi_c$	2,4
Si $4 \leq \varphi < 28$ alors $A_x = \varphi$	0,3,5
Si $28 \leq \varphi < 32$ alors $A_x = 64 + (\varphi - 28) \times 8 + \varphi_c$	1

II.2.2 Découpage d'un exécutif

On distingue plusieurs groupes de bits, appelés champs, dans un exécutif.

Chaque champ se rapporte à un organe différent de la micro-machine, comme indiqué sur la planche 4. L'échelle à droite, qui regroupe les bits opérants par quatre, facilite l'écriture en hexadécimal des exécutifs.

Les quarante huit bits d'un exécutif ont reçu des noms, dont la première lettre est N, puis un mnémonique, et les différents bits sont numérotés C0, C1 ..., lorsqu'un champ est composé de plusieurs bits.

II.2.3 Commande du multiplexeur μ

(voir planches 3 et 4)
pages 22 et 23

1 bit : NMUCO

NMUCO = 0 U \rightarrow μ

NMUCO = 1 M \rightarrow μ

N.B. : la commande mémoire LM force NMUCO = 1

II.2.4 Commande du multiplexeur γ

1 bit : NGACO

NGACO = 0 R \rightarrow γ

NGACO = 1 D \rightarrow γ_{0-6} MC $_{7-15} \rightarrow \gamma_{7-15}$

II.2.5 Commandes du multiplexeur α

2 bits : NALCO

NALCI

NALCO	NALCI	α
0	0	γ
0	1	PDG (γ) ($\gamma_{8-15} \rightarrow \alpha_{0-7}$, $\gamma_{0-7} \rightarrow \alpha_{8-15}$)
1	0	Bus lecture périphérique
1	1	Indicateurs programme

II.2.6 Commandes du multiplexeur β

On peut avoir des actions différentes sur le bit poids fort, le bit poids faible et les 14 autres bits.

Ce qui implique 3 groupes de 3 commandes :

NBTOCO, NBTOC1, NBTOC2 pour le bit 0

NBTFCO, NBTFC1, NBTFC2 pour le bit 15

NBTCO, NBTC1, NBTC2 pour les bits 1 - 14

NBTOC0	NBTOC1	NBTOC2	β_0
0	0	0	μ_0
0	0	1	μ_8
0	1	0	0
0	1	1	μ_1
1	0	0	B
1	0	1	μ_0
1	1	0	μ_{15}
1	1	1	1

NBTFC0	NBTFC1	NBTFC2	β_{15}
0	0	0	μ_{15}
0	0	1	μ_7
0	1	0	μ_{14}
0	1	1	0
1	0	0	B
1	0	1	γ_0
1	1	0	Non utilisé
1	1	1	1

Exemple d'opération:

Transparence

Rotation 8 bits

Décalage droit ouvert

Décalage gauche ouvert

décalage gauche, B entrant

etc...etc...

NBTC0	NBTC1	NBTC2	β_{1-14}
0	0	0	μ_{1-14}
0	0	1	$\left\{ \begin{array}{l} \mu_{1-7} \rightarrow \beta_{8-14} \\ \mu_{8-15} \rightarrow \beta_{1-7} \end{array} \right.$
0	1	0	μ_{0-13}
0	1	1	μ_{2-15}
1	x	x	0

Le x indique que la valeur de cette position est indifférente.

II.2.7 Commandes de l'opérateur et de la retenue entrante

- Il y a 2 bits de commande pour la retenue entrante CIN

NCINCO, NCINC1

NCINCO	NCINC1	CIN
0	0	0
0	1	1
1	0	bit de poids faible d'un Registre R
1	1	B

- 1 bit pour indiquer le mode dans lequel va travailler l'opérateur

NSGLA = 0 mode arithmétique

NSGLA = 1 mode logique

- 2 groupes de 4 bits

NSGDC0, NSGKC1, NSGDC2, NSGDC3 pour la commande des bits 8 à 15 de l'opérateur.

NSGGC0, NSGGC1, NSGGC2, NSGGC3 pour la commande des bits 0 à 7 de l'opérateur.

Fonctions réalisées : ($V \equiv$ "ou logique", $\wedge \equiv$ "et logique")

NSGD NSGG	C0	C1	C2	C3	en mode arithmétique	en mode logique
	0	0	0	0	$\alpha - 1 + \text{CIN}$	$\bar{\alpha}$
	0	0	0	1	$(\alpha \wedge \beta) - 1 + \text{CIN}$	$\alpha \wedge \beta$
	0	0	1	0	$(\alpha \wedge \bar{\beta}) - 1 + \text{CIN}$	$\bar{\alpha} \vee \beta$
	0	0	1	1	$-1 + \text{CIN}$	1 (tous les bits)
	0	1	0	0	$\alpha + (\alpha \wedge \bar{\beta}) + \text{CIN}$	$\bar{\alpha} \vee \bar{\beta}$
	0	1	0	1	$(\alpha \wedge \beta) + (\alpha \vee \bar{\beta}) + \text{CIN}$	$\bar{\beta}$
	0	1	1	0	$\alpha - \beta - 1 + \text{CIN}$	$\alpha \ominus \beta$
	0	1	1	1	$(\alpha \vee \bar{\beta}) + \text{CIN}$	$\alpha \vee \bar{\beta}$
	1	0	0	0	$\alpha + (\alpha \vee \beta) + \text{CIN}$	$\bar{\alpha} \wedge \beta$
	1	0	0	1	$\alpha + \beta + \text{CIN}$	$\alpha \ominus \beta$ "disjonction"
	1	0	1	0	$(\alpha \wedge \bar{\beta}) + (\alpha \vee \beta) + \text{CIN}$	β
	1	0	1	1	$(\alpha \vee \beta) + \text{CIN}$	$\alpha \vee \beta$
	1	1	0	0	$\alpha + \alpha + \text{CIN}$	0
	1	1	0	1	$(\alpha \wedge \beta) + \alpha + \text{CIN}$	$\alpha \wedge \bar{\beta}$
	1	1	1	0	$(\alpha \wedge \bar{\beta}) + \alpha + \text{CIN}$	$\alpha \wedge \beta$
	1	1	1	1	$\alpha + \text{CIN}$	α

Parmi ces nombreuses opérations, certaines ne servent jamais.

II.2.8 Commandes des transferts de σ

Elles sont au nombre de 3 :

NRUC1 pour la commande U_{0-7}

NRUC0 pour le transfert de σ dans U

NRUC2 pour le transfert de σ dans R.

NRUC0	NRUC1	U
0	0	inchangé
0	1	$0 \rightarrow U_{0-7}, U_{8-15}$ inchangé
1	0	$\sigma \rightarrow U$
1	1	$0 \rightarrow U_{0-7}, \sigma \xrightarrow{\quad} U_{8-15}$

NRUC2 = 0 aucune action sur R

NRUC2 = 1 $\sigma \rightarrow R$

II.2.9 Commandes de modification de T

Ce sont :

NRTCO, NRTC1, NRTC2, NRTC3, NRTC4, NPARAM

NRT	C0	C1	C2	C3	C4	action sur T
0	0	1	0	0		recopie
0	x	x	1	1		implicite $(T+1)_{\text{modulo } 16} \rightarrow T$
0	1	1	0	0		retour pile
0	0	0	0	0		adressage normal
0	0	0	1	0		test bascule vraie
0	0	0	0	1		test bascule fausse
1	0	x	1	1		informations extérieur U_{0-5}
1	1	x	1	1		informations extérieur R_{0-5}
1	1	1	0	0		informations extérieur R_{2-7}

Les autres codes ne sont pas employés. Pour les codes désignant un test nous aurons une modification normale de T, si le test est vrai et une modification implicite, si le test est faux. La bascule testée est indiquée par la zone r de la microinstruction, en fonction du tableau suivant :

r = 0	indicateur externe
r = 1	bascule SN
r = 2	bascule SZ
r = 3	bascule SØ

r = 4	bascule MS
r = 5	indicateur programme I ₂
r = 6	indicateur programme I ₁
r = 7	bascule B

Quand NRTCO vaut 1, il y a éclatement et les bits restant indiquent la provenance de l'information extérieure à la partie contrôle.

Il existe en outre la commande NPARAM

NPARAM = 0 aucune action

NPARAM = 1 inhibition du comparateur servant à voir si

$$T_{5-9} = MC_{11-15}$$

En effet, si lors de la modification de T, on a $T_{5-9} = MC_{11-15}$, T-après serait égal à T-avant. Dans ce cas, on remet T à 0, ce qui permet un retour en un point fixe, à l'adresse 0 de la page 0.

Cependant dans les microinstructions ayant une valeur immédiate dans le champ MC_{11-15} , ce retour pourrait se faire fortuitement si la microinstruction est implantée en une adresse dont les cinq derniers bits ont justement la même valeur que la constante.

II.2.10 Commandes de modifications de J

2 bits : NRJCO, NRJC1

NRJCO	NRJC1	J
0	0	inchangé
0	1	sortie de pile de J, U et des indicateurs.
1	0	Y_{10-15}
1	1	J_{0-4} inchangé, $0 \rightarrow J_{5-7}$ (Remise à zéro poids faibles).

II.2.11 Commandes de modification des indicateurs de programmes

Ce sont NMACO pour l'horloge du masque MA

NMSCO pour l'horloge des indicateurs MS et PM

NIDCO pour l'horloge des indicateurs I₁ et I₂

Lorsque l'entrée horloge d'une bascule est à zéro, il n'y a pas chargement de cette bascule. Son contenu est donc conservé.

NIDC1, NIDC2 pour le choix des entrées de I₁ et I₂.

NMACO = 0 MA inchangé

NMACO = 1 $\sigma_{12} \rightarrow MA$

NMSCO = 0 MS et PM inchangé

NMSCO = 1 $\sigma_{13} \rightarrow MS$, $\sigma_{11} \rightarrow PM$

NIDCO	NIDC1	NIDC2	I1	I2
1	0	0	S Z	S N
1	0	1	CØUT	Ø F (Débordement)
1	1	0	σ_{14}	σ_{15}
1	1	1	S Z	B
0	x	x	inchangés (horloge inhibée)	

II.2.12 Commandes de modification des indicateurs de microprogramme

NSØCO pour l'indicateur SØ (Parfois nommé PØ)

NSØCO = 0: SØ inchangé

NSØCO = 1: $\sigma_{15} \rightarrow SØ$

NIBCO, NIBC1, NIBC2 pour la bascule B.

NIBCO	NIBC1	NIBC2	B
0	0	0	inchangé
0	0	1	CØUT
0	1	0	I1
0	1	1	I2
1	0	0	σ_0
1	0	1	σ_{15}
1	1	0	0
1	1	1	1

II.2.13 Commandes des périphériques

2 bits NESCO, NESCI

NESCO	NESCI	
0	0	aucune action
0	1	lecture
1	0	écriture (horloge)
1	1	acquiescement (horloge)

II.2.14 Commandes du système d'interruption et suspension

Ce sont NTITCO, NSUCO

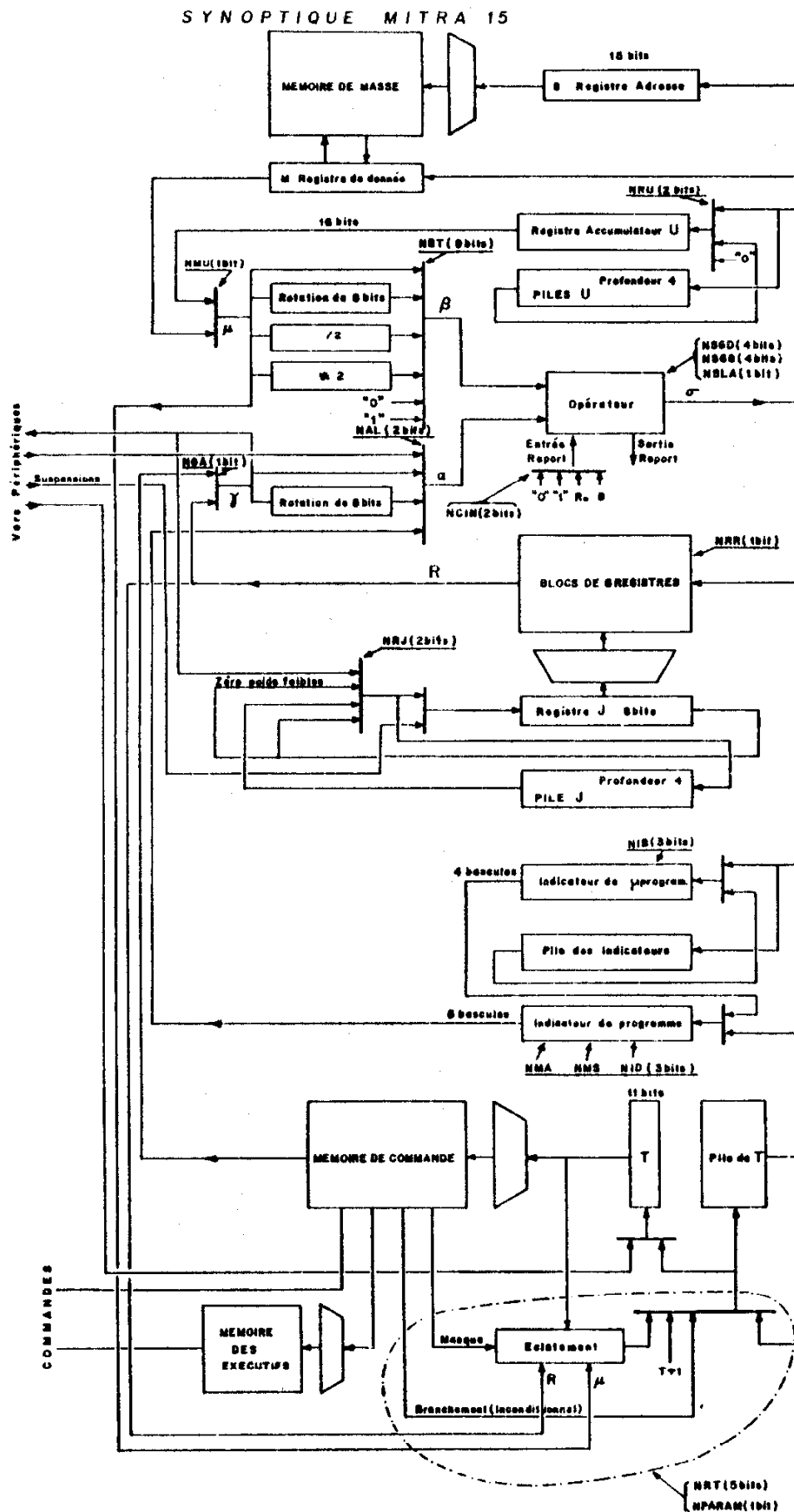
NTITCO = 0 aucune action.

NTITCO = 1 permet, si une interruption de niveau supérieur à celui en cours est présente, de la prendre en comp

NSUCD = 0	aucune action.
NSUCD = 1	interdit la prise en compte d'une suspension lors de l'horloge de la microinstruction.

La signification des 48 bits d'un exécutif est résumée dans le tableau de la page suivante. Les bits ont été regroupés quatre par quatre pour en faciliter l'écriture en hexadécimal.

II.3 Synoptique de la micromachine (planche 3)



II.4 Format des exécutifs (planche 4)

(9 bits) Commande de L'opérateur	bits poids faible	NSGD	$\begin{cases} C0 \\ C1 \\ C2 \\ C3 \end{cases}$	1
	MODE	NSGLA		
	bits poids fort	NSGG	$\begin{cases} C0 \\ C1 \\ C2 \\ C3 \end{cases}$	2
(9 bits) Commande du Multiplexeur β	β_0	NBTO	$\begin{cases} C0 \\ C1 \\ C2 \end{cases}$	3
	β_{15}	NBTF	$\begin{cases} C0 \\ C1 \\ C2 \end{cases}$	4
	β_{1-14}	NBT	$\begin{cases} C0 \\ C1 \\ C2 \end{cases}$	5
Commande de la retenue entrante (2 bits)		NCIN	$\begin{cases} C0 \\ C1 \end{cases}$	6
Commande des transferts (3 bits) $\sigma \rightarrow R, \sigma \rightarrow U$		NRU	$\begin{cases} C0 \\ C1 \\ C2 \end{cases}$	7
Commande du système (2 bits) d'interruption et de suspension		NSU NTIT	$\begin{cases} C0 \\ C0 \end{cases}$	8
Commande de la bascule B (3 bits)		NIB	$\begin{cases} C0 \\ C1 \\ C2 \end{cases}$	9
Commandes de I_1, I_2 (3 bits)		NID	$\begin{cases} C0 \\ C1 \\ C2 \end{cases}$	10
Commandes de MA, MS (2 bits)		NMACO NMSCO		11
Commande du registre J (2 bits)		NRJ	$\begin{cases} C0 \\ C1 \end{cases}$	12
Commande des périphériques (2 bits)		NES	$\begin{cases} C0 \\ C1 \end{cases}$	13
Commande du multiplexeur α (2 bits)		NAL	$\begin{cases} C0 \\ C1 \end{cases}$	14
Commande du multiplexeur μ		NMU	C0	15
Commande du multiplexeur γ		NGA	C0	16
Commande de modification de T (6 bits)		NRT	$\begin{cases} C0 \\ C1 \\ C2 \\ C3 \\ C4 \end{cases}$	17
		NPARAM		18
Commande de la bascule S0		NS0	C0	19

III. MACROMACHINE ET MICROMACHINE MITRA 15

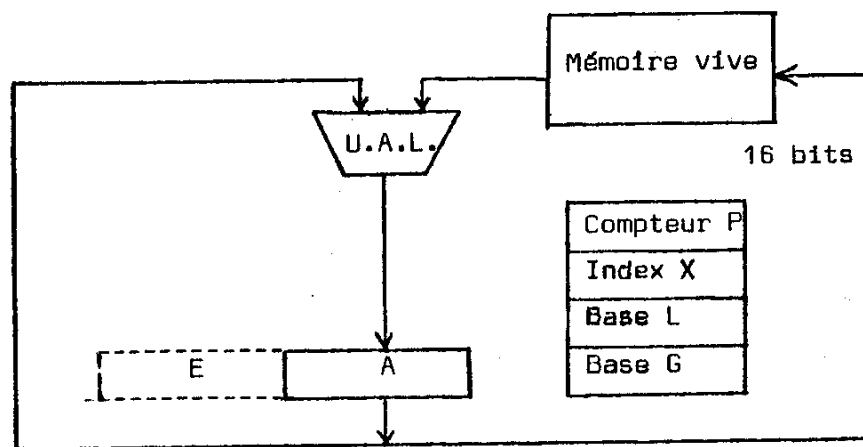
III.1 Macromachine

Modulaire et microprogrammée, elle est de type tourbillonnaire à accumulateur. La mémoire principale est composée de blocs de 4096 mots de 16 bits.

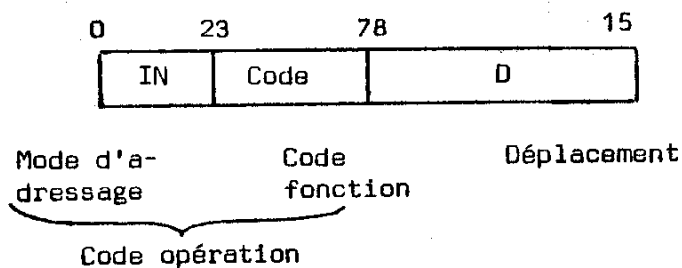
Six registres ont des fonctions particulières :

A	Accumulateur
E	Extension de l'accumulateur
P	Compteur de Programme (compteur ordinal)
X	Index
L	Base Locale
G	Base Générale

La structure interne est grossièrement la suivante :



La macromachine peut exécuter 86 instructions de format fixe 16 bits.



III.2 Mode d'adressage des instructions

Les valeurs des registres Index, Base-Locale et Base-Générale, servent, avec le déplacement D, à calculer une adresse. Cette adresse peut être celle d'un opérande, celle où ranger les résultats, la nouvelle valeur du compteur de programme P ou peut également servir de valeur immédiate. L'adressage peut donc être immédiat, local, général, indexé, indirect ou combinaison des cinq. Dans le cas de branchement il peut être relatif aval ou amont. Ces modes d'adressage permettent d'écrire des programmes translatables.

III.3. Liste des instructions de la macromachine

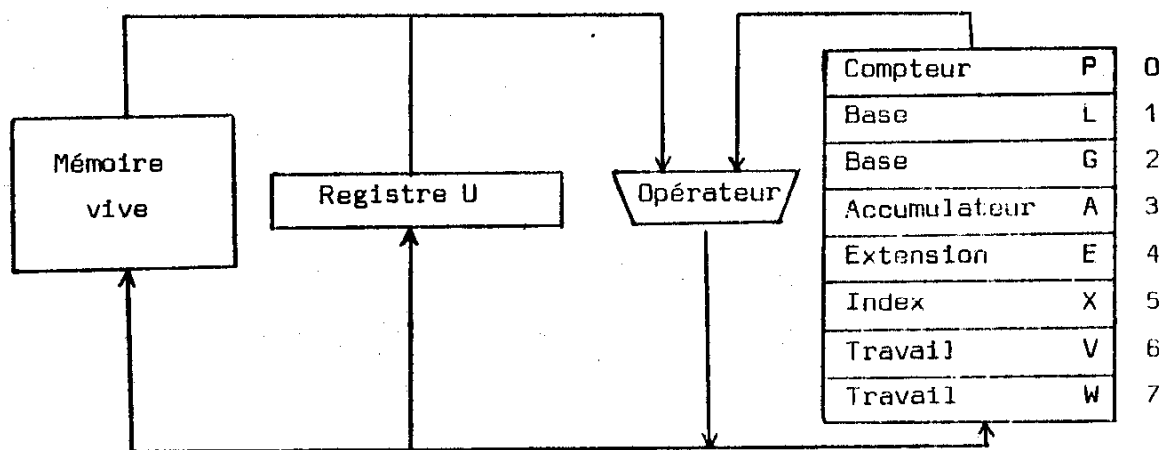
code symbole	fonction	durée en µs	code symbole	fonction	durée en µs
1 - Instructions de chargement					
Instructions de rangement					
LDA	Chargement de A	2,1	XAE	Echange de A et de E	4,3
LDE	Chargement de E	2,1	XAX	Echange de A et de X	4,3
LDX	Chargement de X	2,1	XEX	Echange de E et de X	4,3
LBL	Chargement de l'octet gauche de A	2,6	XAA	Echange des octets droit et gauche de A	2,8
LBR	Chargement de l'octet droit de A	2,6	CCA	Complémentation logique de A	2,8
LBX	Chargement d'un octet dans X	2,6	CCE	Complémentation logique de E	2,8
LLD	Chargement de A et de E	3,4	AAE	Intersection de A et de E dans A	3,1
LEA	Chargement de l'adresse effective dans A	2,7; 3,3	AIE	Réunion de A et de E dans A	3,1
STA	Rangement de A	2,2	AEE	Disjonction de A et de E dans A	3,1
STE	Rangement de E	2,2	CNA	Complémentation algébrique dans A	3,1
STX	Rangement de X	2,2	LNE	Chargement de -1 dans E	2,8
STS	Rangement sélectif	3,4	CNX	Complémentation algébrique dans X	3,1
SBL	Rangement de l'octet gauche de A	2,5	ACE	Addition du report dans E	3,3
SBR	Rangement de l'octet droit de A	2,5	CHX	Division de X par deux	3,1
DST	Rangement de A et de E	3,5	LDR	Chargement d'un registre dans A	3,7
SPA	Rangement de l'adresse-programme	2,8; 3,1	STR	Rangement de A dans un registre	4,3
2 - Instructions arithmétiques et logiques			6 - Instructions de branchements		
ADD	Addition dans A	2,1	BRU	Branchement inconditionnel	1,3
SUB	Soustraction dans A	2,1	BRX	Branchement indexé	1,6
MUL	Multiplication entière microprogrammée	40,-	BCT	Branchement si indicateur C vrai	1,9/2,9
DIV	Division entière microprogrammée	45,-	BOT	Branchement si indicateur O vrai	1,9/2,9
OR	Réunion dans A	2,1	BCF	Branchement si indicateur C faux	1,9/2,9
EOR	Disjonction dans A	2,1	BOF	Branchement si indicateur O faux	1,9/2,9
AND	Intersection dans A	2,1	BAZ	Branchement si A nul	2,2/3,2
CMP	Comparaison	3,2; 4,1; 5	BAN	Branchement si A négatif	2,2/3,2
ADM	Incrémentation en mémoire	2,4	BE	Branchement si égal { équivalent à BCT	1,9/2,9
FAD	Addition flottante		BZ	Branchement si nul { équivalent à BCT	1,9/2,9
FSU	Soustraction flottante		BL	Branchement si négatif { équivalent à BOT	1,9/2,9
FMU	Multiplication flottante		BLZ	Branchement si inférieur { équivalent à BOT	1,9/2,9
FDV	Division flottante		BNE	Branchement si différent / équivalent à BCF	1,9/2,9
3 - Instructions sur chaînes de caractères			BNZ	Branchement si non nul { équivalent à BCF	1,9/2,9
MVS	Déplacement d'une chaîne de caractères	2,5 + 3,7 n	BGE	Branchement si supérieur ou égal / équivalent à BOF	1,9/2,9
TRS	Transcodage d'une chaîne de caractères		BPZ	Branchement si positif ou nul { à BOF	
CPS	Comparaison à une chaîne de caractères		de l'instruction précédente		
4 - Instructions de décalage			7 - Instructions de lecture-écriture		
SHR	Décalage		RD	Lecture directe	3,5
SHC	Décalage spécial		WD	Écriture directe	3,5
SLLS	Décalage logique simple à gauche	4,3 + 1,2 n	8 - Instructions de communication		
SLLD	Décalage logique double à gauche	4,3 + 2,1 n	CLS	Call section	8,3/8,7
SRLS	Décalage logique simple à droite	4,3 + 1,5 n	CSV	Call superviseur	8,8
SRLD	Décalage logique double à droite	4,3 + 1,5 n	RTS	Retour section	4,9/4,5
SAS	Décalage arithmétique simple à droite	4,3 + 2,1 n	RSV	Retour superviseur	6,2
SAD	Décalage arithmétique double à droite	4,3 + 1,5 n	9 - Instructions sur interruptions		
SLCS	Décalage circulaire simple à gauche	4,3 + 1,2 n	STM	Masquage des interruptions	3,4
SLCD	Décalage circulaire double à gauche	4,3 + 2,1 n	CLM	Démasquage des interruptions	3,3
SRCS	Décalage circulaire simple à droite	4,3 + 1,5 n	DIT	Désactivation d'interruption et changement de contexte	33,2
SRCD	Décalage circulaire double à droite	4,3 + 2,7 n	10 - Instructions de protection		
5 - Instructions sur registres			TES	Test et marquage	4,-
SRG	Opération sur registre		LDP	Chargement de la protection mémoire	4,- + 1,4 n
ICX	Incrémentation de X	2,2	11 - Instructions spéciales de décalage		
DCX	Décrémentation de X	2,2	NLZ	Normalisation double longueur	4,6 + 1,8 n
ICL	Incrémentation de L	2,2	PTY	Calcul de parité	4,9 + 1,2 n
DCL	Décrémentation de L	2,2			

III.4 Réalisation Physique de la macromachine

La macromachine MITRA 15, qui n'a pas d'existence physique, est simulée par la micromachine qui est réalisée par câblage.

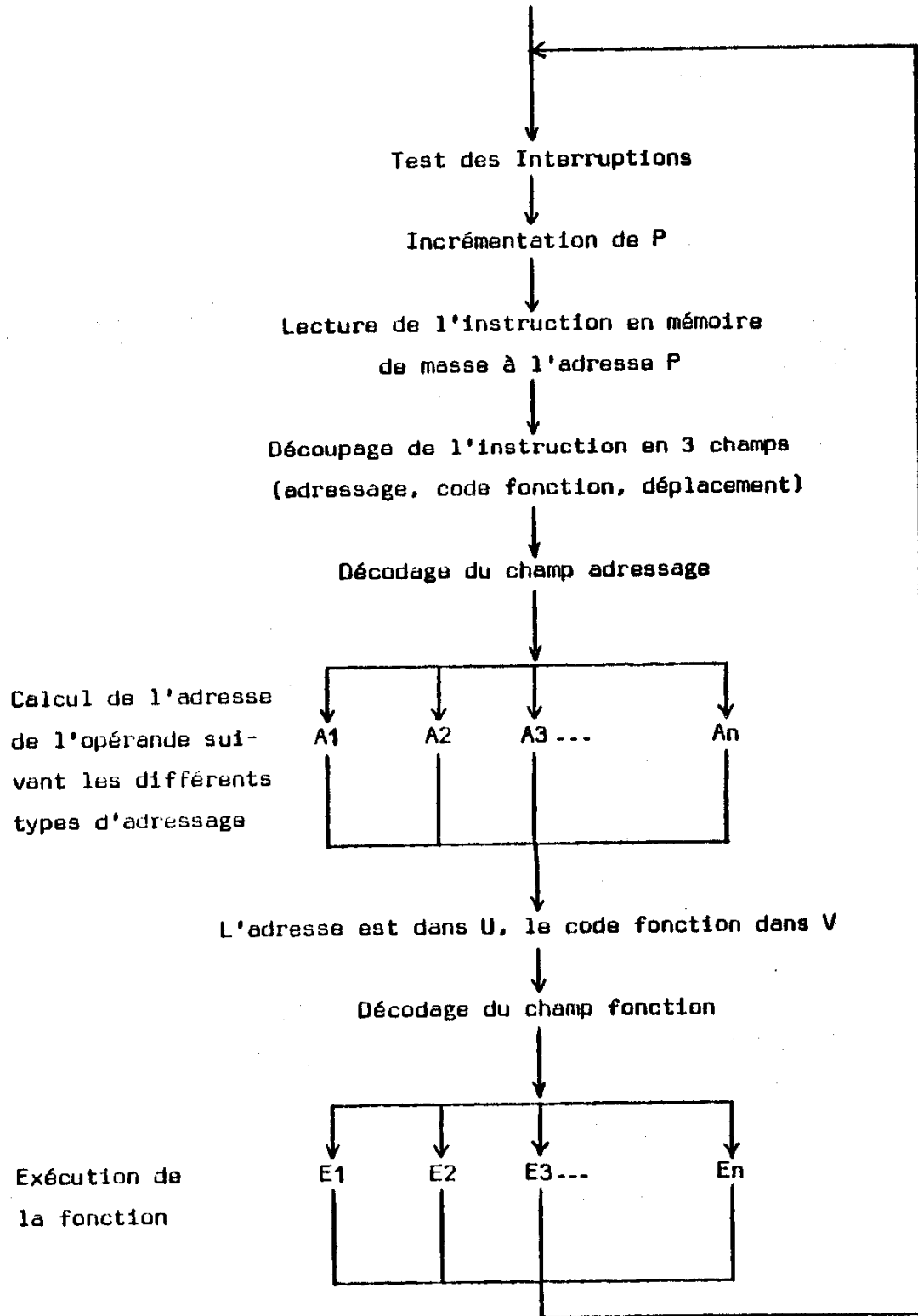
(pour les T.P. de microprogrammation, on ne se sert pas d'une micromachine câblée, mais d'un modèle, rigoureusement équivalent programmé sur le CII IRIS80)

Les registres de la macromachine sont logés dans le bloc zéro des registres généraux de la micromachine (Registres 0 à 7).



III.5 Algorithme de Simulation

L'algorithme de simulation de la macromachine est évidemment microprogrammé. L'organigramme très simplifié de ce microprogramme est donné en pages suivantes, ainsi que la liste des exécutifs utilisés.



Ce microprogramme boucle indéfiniment. Il n'y a pas de moyen "interne" d'en arrêter le déroulement.
 La liste des exécutifs utilisés est donné en pages suivantes.

FONCTIONNEMENT			MNEMONIQUE		TYPE	ϕ_c	Tc	
lecture indirecte	LP \rightarrow α \rightarrow σ \rightarrow U	R \rightarrow γ \rightarrow coupleur	LI		1c	0	0	08000C00
écriture indirecte	$\mu \rightarrow \beta \rightarrow \sigma \rightarrow$	U, coupleur	EI		,	1	0	08001000
R + $\mu \rightarrow$ U	C \rightarrow B		R+UB		,	2	0	08100000
R + $\mu \rightarrow$ R, U			R+UR		,	3	0	0A000000
R - $\mu \rightarrow$ R, U			R-JR		,	4	0	1A000000
R - $\mu \rightarrow$ R, U			R-JI		,	5	0	1A0A0000
R + 2 $\mu \rightarrow$ R, U	C \rightarrow I1	0 \rightarrow I2	R+NR		,	6	0	CA000000
R - 2 $\mu \rightarrow$ R, U			R-NR		,	7	0	DA000000
R $\oplus \mu \rightarrow$ R, U			RJR		1D	0	0	0A000000
R $\oplus \mu \rightarrow$ R, U	Z \rightarrow I1	N \rightarrow I2	RJI		,	1	0	0A080000
R $\wedge \mu \rightarrow$ U			R+MJ		,	2	0	08000100
R $\wedge \mu \rightarrow$ U			R-JU		,	3	0	08000000
(R = 0) \rightarrow Z	(R < 0) \rightarrow N		=RZ		,	4	0	01000000
R \rightarrow R, U			#RR		,	5	0	0A000000
R \rightarrow R, U			=OR		,	6	0	0A000000
R x 2 \rightarrow R, U	C \rightarrow B décalage logique gauche		RXR		,	7	0	0A100000
R x 2 \rightarrow RO \rightarrow	R, U	C \rightarrow B décalage circulaire gauche	RYSR		1E	0	0	2A100000
$\mu/2 \rightarrow$ R, U	$\mu_0 \rightarrow$ RO, UO	$\mu_{15} \rightarrow$ B arithmétique droit	U/SR		,	1	0	8A500000
$\mu/2 \rightarrow$ R, U	$\mu_{15} \rightarrow$ B	décalage logique droit	U/OR		,	2	0	8A500000
$\mu/2 \rightarrow$ R, U	$\mu_{15} \rightarrow$ RO, UO	$\mu_{15} \rightarrow$ B rotation droite	U/PR		,	3	0	8A500000
RG \rightarrow UG	MD \rightarrow UD		RMJO		,	4	0	08000100
RD \rightarrow UG	RG \rightarrow UD		MRU2		,	5	0	08000300
RD \rightarrow UG	MD \rightarrow UD		RMJ1		,	6	0	08000300
MG \rightarrow UG	RD \rightarrow UD		MRU0		,	7	0	08000100
RG \rightarrow RG, UG	$\mu G \rightarrow$ RD, UD		RUR2		1F	0	0	4A000000
O \rightarrow RG, UG	$\mu G \rightarrow$ RD, UD		OUR2		,	0	0	4A000000
O \rightarrow RG, UG	$\mu D \rightarrow$ RD, UD		OURO		,	2	0	0A000000
RG \rightarrow RD, UD	RD \rightarrow RG, UG		RRR3		,	3	0	0A000200
R8 - 15 \rightarrow J	$\mu \rightarrow$ U		=RJ		,	4	0	08004000
R + 1 \rightarrow R, U			R+1R		,	5	0	1A000000
acquiescement indirect $\mu \rightarrow$ U retour pile			ACIR		,	6	0	08003832
R + $\mu \rightarrow$ U			RJU		,	7	0	08000000

Remarque: La code de ces exécutifs n'est donné qu'à titre d'exemple.

FONCTIONNEMENT

B → Bit de protection mémoire (BP)
 arrêt $\mu \rightarrow U$ recopie de T
 lecture directe périphérique $\rightarrow U$
 écriture directe $\mu \rightarrow$ coupleur U
 acquitement suspension avec retour pile
 acquitement suspension sans retour pile
 positionnement du déroutement maître-esclave
 test des interruptions $\mu \rightarrow U$
 P+2 $\rightarrow P, U$ et test des interruptions
 2 $\times \mu \rightarrow U$
 0 $\rightarrow B, \mu \rightarrow U$
 1 $\rightarrow B, \mu \rightarrow U$
 uo $\rightarrow B, \mu \rightarrow U$
 u15 $\rightarrow B, \mu \rightarrow U$
 1 $\rightarrow S0$
 u14 $\rightarrow I1, u15 \rightarrow I2, \mu \rightarrow U$
 R(J) $\rightarrow U, O \rightarrow J$
 u13 $\rightarrow MS, \mu \rightarrow U, u11 \rightarrow PM$

u12 $\rightarrow MA, \mu \rightarrow U$
 B $\rightarrow I2, Z \rightarrow I1$
 3 $\rightarrow I1, N \rightarrow I2, \mu \rightarrow U$
 0 $\rightarrow U$
 1 $\rightarrow U$
 MA $\rightarrow U12, MS \rightarrow U13, I1 \rightarrow U14, I2 \rightarrow U15$
 u $\rightarrow R(J), U, O \rightarrow J$
 M+1 $\rightarrow U$
 M $\rightarrow U$
 u12 $\rightarrow MA, u13 \rightarrow MS, u14 \rightarrow I1, u15 \rightarrow I2$
 MC8-15 $\rightarrow J, \mu \rightarrow U$
 0 $\rightarrow UD, UG$ inchangé
 0 $\rightarrow UG, UD$ inchangé
 uD $\rightarrow UG, uG \rightarrow UD$

MNEMONIQUE	TYPE	ψ	ψ_c	Tc	
-BBP	2	0	0	0	00000000
HLT	2	'	1	1	08000012
LD	4	'	2	0	08000C3E
ED	4	'	3	0	0800108E
AC	4	'	4	0	0800188E
ACR	4	'	5	1	090038E2
DMS	4	'	6	1	08001892
TI	2	0	7	0	88000000
P+2P	'	1	0	0	1A800000
UXOU	'	'	1	0	C8000000
-OB	'	'	2	0	08600000
-1B	'	'	3	0	08700000
-SB	'	'	4	0	08400000
-PB	'	'	5	0	08500000
-1 ϕ	'	'	6	0	01000010
-UP	'	1	7	0	080C0000
-RJU	'	2	0	0	08006000
-UE	'	'	1	0	08008000
-UA	'	'	2	0	08010000
-BP	'	'	3	0	080E0000
-ZP	'	'	4	0	08080000
-OU	'	'	5	0	08000000
-IU	'	'	6	0	08000000
-IU	'	2	7	0	0800FF80
-URJ	'	3	0	0	0A000600
M+IU	'	'	1	0	18000500
-MU	'	'	2	0	08000100
-UIP	'	'	3	0	080D8000
MCJ	4	'	4	0	0800408E
UOUO	2	'	5	0	08000000
OUUO	2	'	5	0	08000000
UUU3	2	3	7	0	48000000

POINCTIONNEMENT	MNEMONIQUE	TYPE	φ	φ _c	T _c	
-			0	Néant		
-			1			
-			2			
-			3			
-			4			
-			5			
-			6			
-			7			
-			8			
-			9			
-			A			
-			B			
-			C			
-			D			
-			E			
-			F			
-			10			
-			11			
-			12			
-			13			
-			14			
-			15			
-			16			
-			17			
-			18			
-			19			
-			1A			
-			1B			
-			1C			
-			1D			
-			1E			
-			1F			
Test bascule vraie, μ → U	TV	0			0	0100AD00 08000008
Test bascule faux, μ → U	TF	0			0	0100AD00 08000004
MC 7-15 / → U7-15, o → U0-6	MCUD	3			0	0000FF80 08000008E
MC 8-15 → U0-7, U8--15 inchangé	MCUG	3			0	0000AF80 0800028E
éclatement suivant R2-7, μ → U	BRD	5			1	0000AD00 08000072
éclatement suivant R0-5, μ → U	BRG	5			1	0000AD00 0800006E
éclatement suivant μ0-5, μ → R, μD → UD, O → UG	MURT	5			1	0000AD00 0F00004E
R + 2 → U	R+2U	0			0	000094AF 18000000
R + 2 → R, U	R+2R				0	000094AF 1A000000
R + μ → U	R+UU				0	00009480 08000000
R + μ → R, U, C → I1, φ → I2	R+UI				0	00009480 0A080000
R - U U → U	R-UU				0	00006300 18000000
MC 8-15 → R9-15, U8-15, O → R0-7, U0-7	MCR	3			0	0000FE00 0A00008E
R - 1 → R, U	R-1R	0			0	00000000 0A000000
R μ → R, U	R. UI				0	0000EF00 0A080000
R V μ → R, U	R/ UI				0	0000BD80 0A080000
μ/2 → R, U	U/BR				0	0000AD44 85A00000
R x 2 → R, U	RXBR				0	0000C600 3A100000
μ → R, U	= UR				0	0000AD00 0A000000
μ → R, U	= UI				0	0000AD00 0A080000
R → R, U	= RU				0	0000FF80 08000000
μD → RG, UG	URR1				0	0000FD12 4A000000
μG → RG, UG	URRO				0	0000FD00 0A000000
RG → RG, UG	RURO				0	0000AF80 0A000000

IV. INTERRUPTIONS ET SUSPENSIONS

IV.1 Suspensions

Une suspension est à la micro-machine ce que l'interruption est à la macromachine. Voyons le déroulement d'une suspension.

La gestion d'un périphérique est assurée par un micro-programme de la mémoire de commande de la micromachine. Le périphérique demande que son micro-programme s'exécute, pour cela il positionne à "1" une bascule de suspension qui lui est réservée. (voir planches 5 et 6 page 34)

La priorité de cette demande est le numéro de sa bascule, si cette demande est la plus prioritaire, on lit dans la Mémoire Associée aux Suspensions (M.A.S) un mot de seize bits qui donne tous les renseignements sur le micro-programme dont le périphérique demande l'exécution.

(0 début du pg et 0 des données)

IV.1.1 Mémoire Associée aux Suspensions

Signification des différents champs d'un mot de la M.A.S.

Bits 0.1 On a regroupé les trente deux suspensions possibles en quatre niveaux. Il y a quatre niveaux de suspension, indiqués par un registre de deux bits, N, qui peut recevoir les deux premiers bits de la M.A.S.

Bit 2 Ce bit commande la sauvegarde des registres J et U, ainsi que des indicateurs, dans les piles câblées.

Bit 3 Ce bit interdit la prise en compte d'une suspension si une demande mémoire est en cours. Le signal α est mis à "1" si on fait une demande mémoire (DM) et remis à zéro si on fait une lecture ou écriture (LM, EM).

On ne prend pas en compte une suspension soit parce que son niveau est inférieur au niveau en cours, soit parce que le registre M de communication avec la mémoire contient une valeur à sauvegarder (après DM), soit enfin parce que dans l'exécutif en cours, le bit NSU (II.2.14) est positionné.

La bascule de suspension est mise à zéro par le périphérique lorsqu'il a satisfaction, aucune demande n'est donc perdue.

non associé à une pile de sauvegarde

Bits 4.7 Chargés dans les bits 2.5 du registre J, indiquent donc un bloc de quatre ou huit registres affecté au micro-programme dont le périphérique demande l'exécution.

Bits 9.15 Chargés dans les bits 0.6 du registre compteur ordinal T, donnent donc l'adresse du microprogramme. Cette adresse est toujours multiple de huit (trois derniers bits à zéro).

IV.1.2 Communication Micro-machine - Périphérique

La micromachine peut envoyer des ordres aux périphériques, lecture, écriture ou acquittement, bits NES (II.2.13). Pour désigner le périphérique qui doit exécuter l'ordre, on donne son adresse sur les cinq bits poids faible de Y. (voir synoptique). (page 22)

Le périphérique peut envoyer des requêtes à la micro-machine, sous la forme de suspensions.

Enfin la micromachine et les périphériques peuvent échanger des données de seize bits, l'entrée des données se faisant sur α et la sortie sur σ (fil non représenté).

IV.1.3 Interruptions

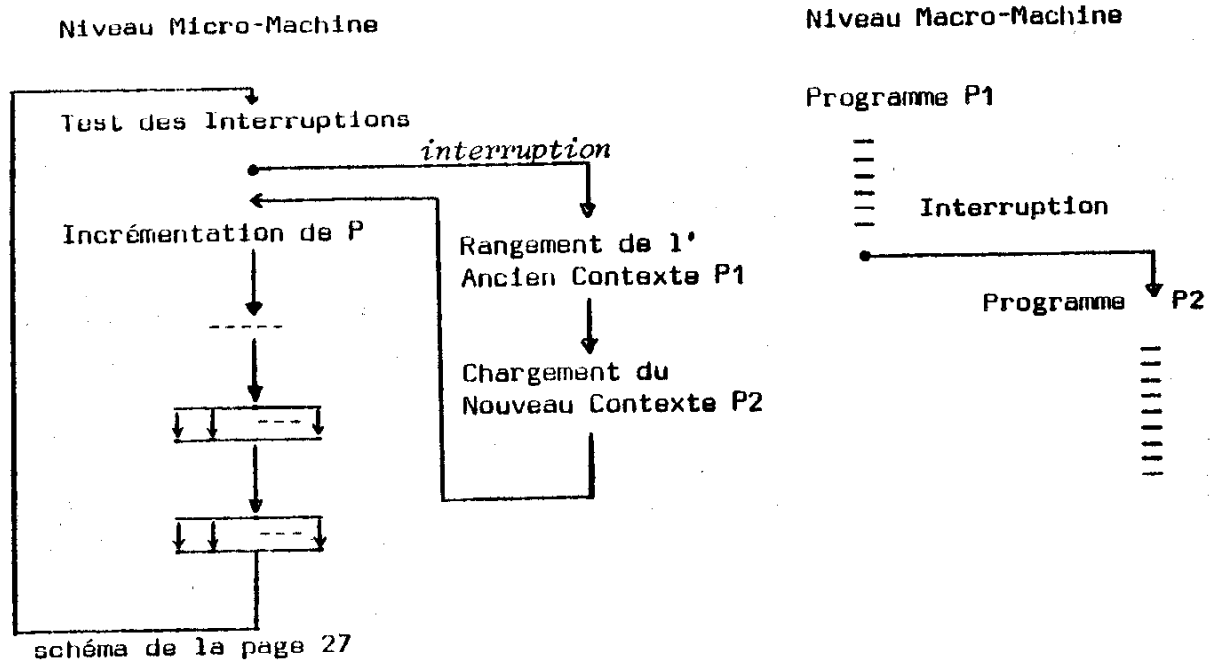
Une interruption s'adresse à la macromachine. Si une interruption a un niveau de priorité supérieur au niveau en cours et que les interruptions sont testées par le bit NTIT (II.2.14), alors elle déclenche une suspension.

Une interruption est une suspension de priorité trente.

Le microprogramme associé à cette interruption va trouver dans le registre huit le code binaire du niveau de priorité de l'interruption et calcule deux adresses grâce à ce code. Dans une zone de la mémoire de masse donnée par la première adresse, le microprogramme sauvegarde le contexte de la macromachine, c'est-à-dire les registres P, L, G, A, E, X et les indicateurs de programme. Puis il charge dans ces registres et indicateurs le contexte du programme d'interruption qui se trouve dans la mémoire de masse à la deuxième adresse.

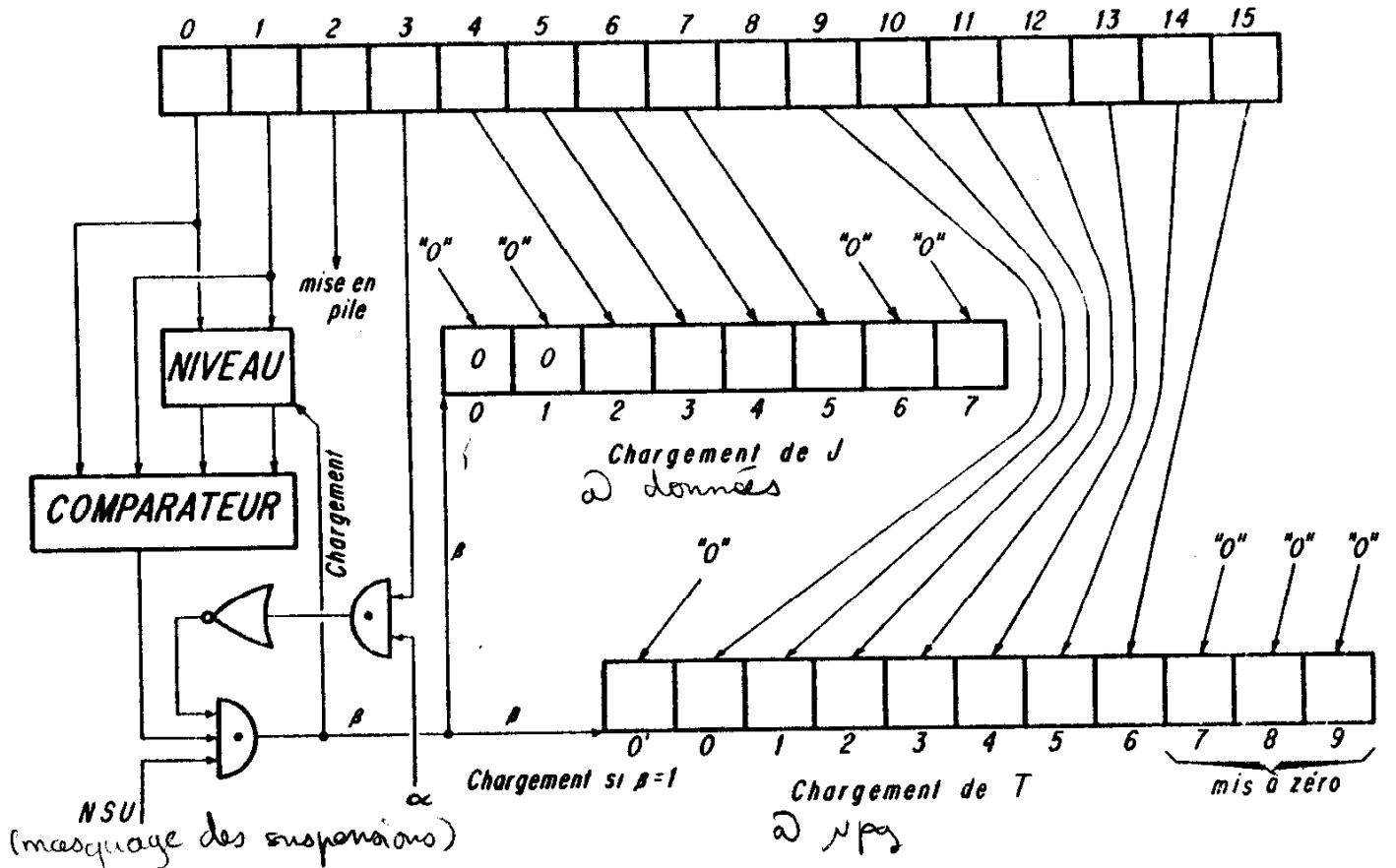
Le microprogramme d'interprétation déroule alors les macro-instructions du macro-programme de traitement des interruptions.

Schéma d'une interruption



L'exécution de chaque instruction, notée — du niveau macro-machine, équivaut à un tour complet du niveau micro-machine.

IV.3 Format du mot de M.A.S (planche 5)



IV.4 Schéma du système d'interruption et suspension (planche 6)

