209 p. hob

FACULTÉ DES SCIENCES

Se N

# STRUCTURE DU CODE DE PROGRAMMATION



# THÈSE

présentée

à la FACULTÉ DES SCIENCES de l'UNIVERSITÉ de NANCY
pour l'obtention

du DOCTORAT de SPÉCIALITÉ (Mathématiques 3me cycle)

par

### Marion CRÉHANGE

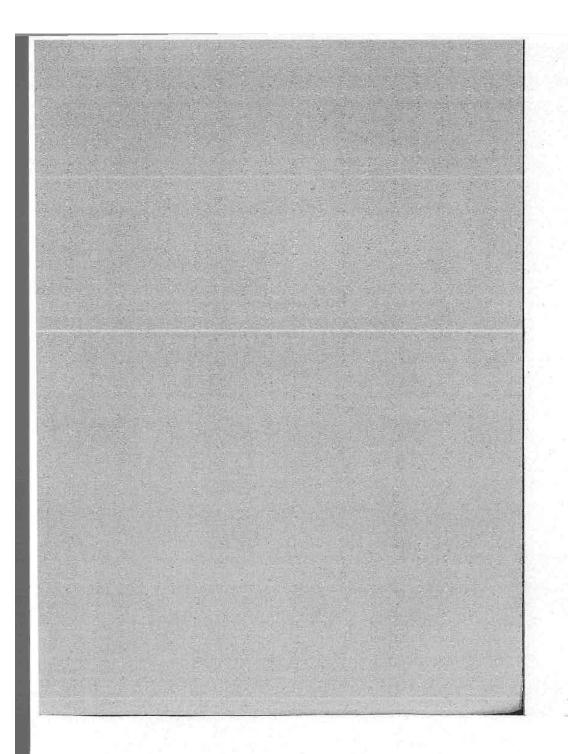
et soutenue en mars 1961 devant la Commission d'examen

Jury: MM. J. LEGRAS, Président.

M. HERVÉ

J. GOSSE

Examinateurs.



#### UNIVERSITE DE NANCY

#### FACULTE DES SCIENCES



#### STRUCTURE

DU CODE DE PROGRAMMATION

par

Marion CREHANGE

#### UNIVERSITE DE NANCY --- FACULTE DES SCIENCES

Doyen: M. URION
Assesseur: M. ECHEVIN

Doyens honoraires: MM. CORNUBERT, DELSARTE

Professeurs honoraires: MM. GUTTON, CROZE, RAYBAUD, LAFFITTE, LERAY, JOLY, LAFORTE, EICHHORN, CAPELLE, GODEMENT, DUBREIL, L. SCHWARTZ, DIEUDONNE, de MALLEMANN, LONGCHAMBON, LETORT, DODE, GAUTHIER, GOUDET, OLMER, CORNUBERT, CHAPELLE, GUERIN.

Maîtres de conférences honoraires : MM. RAUX, LIENHART.

#### PROFESSEURS

1/1/

IVLIVI.			
URION	Chimie biologique	DUVAL	Chimie
DELSARTE	Analyse supérieure	COPPENS	Radiogéologie
ROUBAULT	Géologie	FRUHLING	Physique
VEILLET	Biologie animale	LIONS	M.M.P.
ECHEVIN	Botanique	SUHNER	Physique expérimentale
W.Y.	Chimic org. industr.	HILLY	Géologie
BARRIOL	Chimie théorique	LE GOFF	Génie Chimique
BIZETTE	Physique	CHAPON	Chimie biologique
GUILLIEN	Electronique	HEROLD	Chimie industrielle
GIBERT	Chimie physique	SCHWARTZ	Exploitation minière
HERVE	Calcul diff. et intégral	GAYET	Physiologie
LEGRAS	Mécanique rationnelle	MANGENOT	Phytopathologie
DAVID	Chimie organique	MALAPRADE	Chimie
BOLFA	Minéralogie	HVDNI	Physique
NICLAUSE	Chimie	DELYWYIE	Zoologie
FAIVRE	Physique appliquée	DEBOUTEVILLE	
AUBRY	Chimie minérale	BONVALET	Mécanique physique

	MAÎTRES	DE CONFERENCE	ES
MM.			
WERNER	Botanique	RENARD	Physique théor, et Nucl.
GARNIER	Agronomie	CONDE	Zoologie
BRUHAT	Mathématiques génér.	GOSSE	Génie chimique
GUDEFIN	Physique	CHAMPIER	Physique
Mme ROIZEN	Physique	GAY	Chimie biologique
REGNIER	Physico-chimie	CLIN	Paléontologie
KERN	Minéralogie	ROCCI	Géologie
WEPPE	Minéralogie appliquée	WEISS	Physique M.P.C.
BERNARD	Géologie	Mme BASTICK	Chimie M.P.C.
ARAGNOL	Mathématiques (prop.)	VUILLAUME	Psychophysiologie
BONVALET	Mécanique Appliquée	PIERRET	Maître de conf. adjoint
BASTICK	Chimie	PLAN	Mathématiques
MARI	Chimie (ISIN)	LAFON	Physique (ISIN)

Secrétaire : M. CARON

Je tiens tout d'abord à exprimer mes plus vifs remerciements à Monsieur le Professeur LEGRAS pour la façon attentive et bienveillante avec laquelle il a dirigé mon travail et pour les conseils et suggestions qu'il n'a cessé de me prodiguer.

A STATE OF THE STA

Je remercie également Messieurs les Professeurs M. HERVE et J. GOSSE qui ont bien voulu me faire l'honneur de composer le Jury.

#### TABLE des MATIERES

I	-	INTRODUCTION	
ΙΙ		LOGICUE EXTERIEURE ET AUTOPROGRAMMATION	69
		II - A - Utilité d'un langage autre que le langage machine	
		II - B - Différence entre logique extérieure et autoprogrammation	4
		<ul> <li>II - C - Avantages et inconvénients de la logique extérieure et de l'autoprogrammation</li> </ul>	
		II - D - Avantages de la coexistence des 2 procédés	(
		NOTE OF THE PROPERTY.	
III	-	ORGANISATION GENERALE DU C.D.P.	8
		III - A - Structure	8
		III - A - 1 - Partie centrale	
			8
		III - A - 2 - Sous-programmes - Divers C.D.P.	9
		III - B - Paquets de cartes	10
ΙV	-	INSTRUCTIONS DU C.D.P. (tronc commun)	11
ć		IV - 1/2 - Structure des ordres	11
		IV - A - 1 - Ordres à 2 mémoires	11
		IV - A - 2 - Ordres à 1 mémoire	12
		IV - B - Détail des ordres	13
		4 Total Colores and Alexander and Colores	
		IV - B - 1 - Ordres à 2 mémoires	13
		IV - B - 2 - Ordres à 1 múmoire (ordres de service)	16
37		ECRIPHED DEC E/PRIES INTERDED ÉTATIONS	10

#### VI - PARTIE INTERFRETATIVE EN LOGIQUE EXTERIEURE :

GENERALITES	7	0
	LATE AND THE STATE OF	
VI - A - Schéma général	2	0
VI - B - /_nalyse	2	1
VI - C - Organigramme	2	2
VI - C - 1 - Instructions à 1 mémoire	2	2
VI - C - 2 - Instructions à 2 mémoires	2.	ć.
a) Partie commune	ed 2	5
ø) Découpage des ordr		
(b) Indexage	2.	
() Partie commençant		
δ) Λiguillage "aller à	GOOOP" 2	G
b) Branche particulière à cha	eque valeur de P 2	7
$\alpha$ ) $P = 0$	2	7
(β) P = 1 à 5	2.	8
(Y) P = 8	2	9
J) P = 9	2	9
c) Analyse	30	)
4 x 14 A 4 T	The second second	
VII - ETUDE DETAILLEE DE LA PARTIE INTERPRE	GTATIVE 31	L
VII - A - Partie commune à tous les ordres	33	1
VII - B - Ordres à 1 mémoire ou ordres de serv	ice 31	Ĺ
VII - B - 1 - Partie commune à ces ord	dres 31	Ĺ
VII - B - 2 - Partie propre à chaque co		2
VII -B - 3 - Remarque sur les ordres	de service 37	7
VII - C - Ordres à 2 mémoires	38	3
VII - C - 1 - Fartic commune à tous ce	es ordres 38	3
a) "Dóbut	38	3
b) Index et fin de la partie co	ommune 40	)
c) Analyse	43	3

VII - C - 2 - $P = 0$ : sous-programmes à 2 adresses. Opérat	ions
en virgule fixe - Tests	45
a) PP' = 00 : sous-programmes à 2 adresses	
(standardisation et programme	
interprétatif relatif à ces ordres)	45
b) PP' = 01 : addition on virgule fixe	48
c) PP' = 02: soustraction en virgule fixe	49
d) PP' = 03 et PP' = 05 : multiplication sans ou ave	С
dócalage	50
e) PP' = 04 et PP' = 06 : division sans ou avec	
décalage	53
f) PP' = 07: test de signe à 3 branches	56
g) PP' = 08 : test sur l'existence d'un 8 ou d'un 9	
on position n° B de la mémoire C	60
VII - C - 3 - $P = 1, 2, 3, \frac{1}{2}, 5$ : sous programmes à	
3 adresses	61
- Standardisation	
- Programme interprétatif relatif à ces ordres	
5 4 5 4 7	
VII - C - 4 - P = 9: bouclage	63
*****	
VII - D - Sous-programmes placés à poste fixe	73
VII - D - 1 - Perforation (49 N iA jB kC)	73
VII - D - 2 - Modification de bouclage (00 N 01937 jB kC)	79
7.000	
VIII - ASSEMBLAGE ET DISPOSITION DE LA LOGIQUE EXTERIEURE	81

IX ·	- AUTOPERFORATION - GENERALITES	85
	IX - 1 Emplacement du programme en langage machine obtenu.	
	Séquence 1 <sub>min</sub> , 1 <sub>max</sub>	86
	IX - B - Notations	87
	IX - C - Préparation des cartes à autoperforer	87
	IX - D - Schéma général	88
	IX - E - Organigramme 39 c	t 89 bi
	IX - E - 1 - Perforation	89
	IX - E - 2 - Instructions à 1 mémoire	90
	IX - E - 3 - Instructions à 2 (ou 3) mémoires	91
	a) Partie commune	91
	b) Branche particulière à chaque valeur de P	91
	al An	91
	β) P = 1 2 5	92
	<b>8</b> ) P = 8	92
	5)P=9 ·	92
X -	ETUDE DETAILLEE DE L'AUTOPROGRAMMATION	93
	X - A - Indexage	93
	X - A - 1 - Indexage des opérations en virgule fixe	94
	X - A - 2 - Indexage des entrées dans les sous-programm	es
	à 2 adresses	3.
	X - A - 3 - Indexage des entrées dans les sous-programm à 3 adresses	es 9
	X - B - Perforation	10:
	X - C - Partie commune à tous les ordres	10:
	X - D - Ordres à 1 mémoire ou ordres de service	10
	X - D - 1 - Pártie commune à ces ordres	10
	X - D - Z - Partic propre à chaque code PP'	10
	X - E - Ordres à 2 (ou 3) mémoires	118
	X - E - 1 - Partie commune à ces ordres	113
	X - E - 2 - P = 0: sous-programmes à 2 adresses. Opére en virgule fixe. Tests.	ations 12

,

## X - E - Ordres à 2 (ou 3) mémoires (suite)

			120
			123
c) Opération	ons e	n virgule fixe et tests, non indexés	123
	ok )		123
	B)	Partie commune aux ordres OP' non	
100	8)	$\mathbb{P}^1 = 1$ : addition on virgule fixe	126
	5)		127
		fixe (sans index)	128
		(sans index)	128
(en	يعثان	(sans index)	129
	0)	P' = 4: division sans décalage (sans index)	131
	Partic commune aux ordres OP' (P' \neq 0)  c) Opérations en virgule fixe et tests, non indexés  d) Programmes en langage machine dlaborés par l'autoprogrammation  partic commune aux ordres OP' non indexés  d) F' = 1: addition en virgule fixe (sans index)  f) P' = 2: soustraction en virgule fixe (sans index)  p' = 3: multiplication sans décalage (sans index)  p' = 5: multiplication avec décalage (sans index)  p' = 6: division sans décalage (sans index)  h) P' = 6: division avec décalage (sans index)  h) P' = 7: test SI (sans index)  p' = 8: test SD (sans index)  Opérations en virgule fixe et tests, indexés  d') Programmes en langage machine Glaborés par l'autoprògrammation  partic commune aux ordres OP' indexés  y') P' = 1: addition en virgule fixe	131	
		132	
	2)	P' = 8 : test SD (sans index)	133
d) Opératio		1 DE - N SOLE DEN 1 A - 14 OF NO X R - X	134
	≪ )		134
	3)		120
	X)		139
	51	' Carlotte and the carl	140
		fixe (avec index)	140
	c)	(avec index)	141
			142
	$\theta$ )		143
	λ)	P' = 6: division avec décalage (avec	143
	M 1		145
	5)		147

X - E - 3 - P = 1, 2, 3, 4, 5: sous-programmes à 3 adresses	148
이 모르는 사람이 그 이렇게 이 사람이 가를 통해 통해를 통해를 통해 들어 들어 보니 때에 나는 이 보니 아니다. 아니라 아니라 아니라 나를 하는데 다 다 다 하는데 그리다.	15
	152
	15
c) Sous-programme de bouclage pour l'exécution	
en langage machine	154
X - F - Sous-programmes placés à poste fixe	16
X-F-1-Perforation (ordre 49 N iA jB kC)	160
X-F-2-Modification de bouclage (ordre 00 N 01937 jB kC)	16
XI - DISPOSITION DE L'AUTOPROGRAMMATION ET DES SOUS-	
PROGRAMMES DES "PAQUETS VERTS"	164
XI = A - Autoprogrammation	164
	165
YIL SOUS-PROCEAMMES ADAPTES ALL D. P. DIVERS C.D. P. FORM	VÉ.S
MI - BOOD I NOCHAMANIED IDIL IDD II O G. P. I	166
XII - Λ - Généralités	160
XII - B - C.D.P. en virgule flottante simple précision	169
XII - C - C.D.P. en virgule flottante double précision	17
XII - D - C.D.P. en virgule fixe	171
XII - D - 1 - Sous-programme auxiliaire de décalage	172
a) Sous-programme auxiliaire de décalage pour	
l'exécution en langage machine	175
b) Sous-programme auxiliaire de décalage en	
logique extérieure.	177
X - E - 4 - P = 9: bouclage a) instructions produites par l'autoprogrammation b) Autoprogrammation de l'ordre de bouclage c) Sous-programme de bouclage pour l'exécution en langage machine  X - F - Sous-programmes placés à poste fixe  X - F - 1 - Perforation (ordre 49 N iA jB kC) X - F - 2 - Modification de bouclage (ordre 00 N 01937 jB kC)  XI - DISPOSITION DE L'AUTOPROGRAMMATION ET DES SOUS- PROGRAMMES DES "PAQUETS VERTS"  XI - A - Autoprogrammation XI - B - Exécution en langage machine (paquets verts)  XII - SOUS-PROGRAMMES ADAPTES AU C. D. P DIVERS C. D. P. FORM  XII - A - Généralités  XII - B - C. D. P. en virgule flottante simple précision XII - C - C. D. P. en virgule flottante double précision XII - D - C. D. P. en virgule fixe  XII - D - 1 - Sous-programme auxiliaire de décalage a) Sous-programme auxiliaire de décalage pour l'exécution en langage machine b) Sous-programme auxiliaire de décalage en	180
XIII - REMARQUES DIVERSES	182
XIII - A - Chargement	182
	18
	188
	188
VIV CONCLUCION	
AT A - CONCTROION	18

#### CODE DE PROGRAMMATION

-----------

#### INTRODUCTION

part of the first of the control of the first of the firs

Le code de programmation ou "C. D. P." est à la fois une logique extérieure et un dispositif d'autoprogrammation ou compilateur. Il est adapté . à l'ordinateur I. B. M. 650 standard mais se généralisera probablement aux autres 650.

La logique extérieure et l'autoprogrammation constituaient chacune déjà un gros progrès sur les simples ordres en langage machine. Mais un procédé comportant une juxtaposition des deux méthodes possède de nets avantages supplémentaires, Le C. D. P. a sur les logiques extérieures classiques la supériorité de permettre une exécution bien plus rapide des programmes grâce à l'autoprogrammation. Par rapport aux dispositifs d'autoprogrammation habituels, il offre la commodité de permettre une analyse en langage "extérieur" et non en langage machine, ce qui rend la mise au point aisée et les modifications peu risquées.

Le C. D. P. a pour autre caractéristique d'être constitué par un tronc commun constant (mais susceptible d'améliorations et d'agrandissements) autour duquel on peut grouper toutes sortes de sous-programmes. Ceci permet, par le choix des sous-programmes, de construire très facilement des codes de programmation adaptés à des problèmes

particuliers. Pour l'instant, 3 C.D.P. ont été créés: le C.D.P. en virgule fixe, le C.D.P. en virgule flottante simple précision, le C.D.P. en virgule flottante double précision. Chacun de ces C.D.P. peut être modifié et augmenté à volonté (dans la limite de capacité du tambour). D'autre part il est en général très aisé de passer d'un programme écrit en C.D.P. virgule fixe à un programme en C.D.P. virgule flottante (et vice versa) ou d'un programme écrit en C.D.P. virgule flottante simple précision à un programme en C.D.P. virgule flottante double précision (et vice versa). Ceci permet de faire des vérifications de la position de la virgule, ou des améliorations de précision.

Enfin le C.D.P. simule 9 registres d'index. Ses instructions, pour la plupart, occupent chacune 2 mémoires consécutives. Il comporte des ordres logiques commodes, en particulier un ordre de bouclage assez puissant qui exécute une modification automatique d'un registre d'index choisi.

# II - LOGIQUE EXTERIEURE e

#### II - A - UTILITE D'UN LANGAGE AUTRE CUE LE LANGAGE MACHINE :

Les ordres en langage màchine sont des ordres élémentaires, peu pratiques pour la programmation à cause surtout de leur manque de concision. Pour commander une addition en virgule fixe, par exemple, il faut 3 ordres élémentaires; pour calculer une fonction, il faut, dans les meilleures conditions, entrer dans un sous-programme, ce qui exige plusieurs ordres en langage machine. La logique extérieure ou l'autoprogrammation permettent de faire interpréter par la machine des ordres plus compacts que les ordres en langage machine. Une addition, par exemple, s'exprimera par un seul ordre, de même que le calcul d'une fonction classique. De plus, dans le langage plus concis que comprend alors la machine, peuvent entrer des ordres logiques commodes. On peut très bien, d'autre part, créer des logiques extérieures ou des compilateurs adaptés à des programmes particuliers (par exemple le FLEC, spécialement adapté aux problèmes électriques : THOMAS, ingénieur E. N. S. E.M docteur de 3ème cycle de Mathématiques Appliquées).

Si par exemple nous faisons la convention que l'ordre complexe 11 A B C représente l'addition en virgule flottante des nombres contenus respectivement en mémoires A et B pour placer le résultat en C, il est facile de concevoir un programme qui interprétera 11 comme "entrée dans le sous-programme d'addition en virgule flottante" et utilisera les valeurs de A, B et C pour fournir à ce sous-programme les renseignements nécessaires. Ainsi à chaque ordre complexe la partie interprétative fait correspondre une suite d'ordres en langage machine. Ce processus peut se dérouler de 2 façons différentes : en logique extérieure et en autoprogrammation.

#### II - B - DIFFERENCE ENTRE LOGIQUE EXTERIEURE ET AUTOPROGRAMMATION

En logique extérieure, la machine traduit chaque ordre complexe en instructions "langage machine" qu'elle exécute aussitôt. Elle passe ensuite à l'ordre suivant qu'elle interprète (c'est-à-dire traduit en langage machine) puis exécute. Il ne reste aucune trace des instructions en langage machine formées. En particulier, si par bouclage on revient à un ordre déjà traité, la machine le retraduit, gnorant le travail effectué auparavant.

En autoprogrammation au contraire, la machine traduit chaque ordre complexe par des ordres en lang ge machine, mais, pour conserver ces ordres, les perfore sur des cartes aprèls leur avoir attribué des adresses. Le succession des événements est donc la suivante : interprétation d'un ordre puis perforation; interprétation de l'ordre suivant et perforation; etc,..... A la suite de cette phase (dans laquelle les bouclages ne sont pas exécutés) qui est l'autoprogrammation proprement dite ou i "autoperforation", on obtient un paquet de cartes portant le programme en langi ge machine équivalent au programme écrit initialement en ordres complexes.

Remarque: en C. D. T., la plupart des ordres sont en fait des entrées dans des sous-programmes. L'autoprogrammation ne provoque pas la perforation des sous-programmes eux-mômes, car cela risquerait de donner un il pencombrement énorme au programme en langage machine obtenu : certains sous-programmes seraient en effet reperforés un grand nombre de fois. Seules les instructions nécessairas à l'entrée dans ces sous-programmes sont perforés. Pour exécuter le programme en langage machine obtenu il faudra donc lui adjoindre les sous-programmes utiles (cf paragraphe III - B).

every series of the law supplied that the

Après avoir effectué l'autoprogrammation, il reste dans une 2ème phase, à exécuter le programme obtenu. Pour cela, il faut charger les cartes qui ont été perforées par la machine, ainsi que, éventuellement, les sous-programmes utiles. On dispose alors d'un programme en langage machine, sans partie interprétative donc d'exécution beaucoup plus rapide qu'un programme en logique extérieure.

## II - C - AVANTAGES ET INCONVENIENTS DE LA LOGIQUE EXTERIEURE ET DE L'AUTOPROGRAMMATION

La logique extérieure a le gros avantage de permettre une analyse très commode du programme et par conséquent une mise au point aisée. En effet l'analyse d'un programme en logique extérieure donne lieu à la perforation d'une carte d'analyse par ordre en langage extérieur. L'analyse est donc beaucoup plus concise et par conséquent bien plus facile à lire et à vérifier qu'une analyse en langage machine. En particulier la suppression de l'analyse des sous-programmes internes est automatique. Le fait que la mise au point soit aisée entraîne la facilité et le peu de risque des modifications.

Par contre, la logique extérieure a un gros inconvénient : la lenteur d'exécution. Pour chaque ordre complexe, la machine doit exécuter une partie interprétative, même si cet ordre a déjà été effectué auparavant. En particulier, dans un bouclage, si l'on passe 1000 fois par l'exécution de chaque ordre, on passe 1000 fois par sa partie interprétative, ce qui cause une grande perte de temps (voir le paragraphe II - B).

.../..

. . . /. . .

L'autoprogrammation, elle, se prête mal à la mise au point. En effet l'analyse d'un programme se fait lors de son exécution. Or c'est au cours de la 2ème phase que les programmes s'exécutent. Ils sont alors en langage machine et il serait très délicat de prévoir pour eux une analyse aussi concise que celle en logique extérieure.

Par contre, pour les programmes répétitifs ou d'utilisation fréquente, l'autorprogrammation représente un gain de temps considérable (qui peut aller de 75 % à 25 % environ suivant le genre de programme et l'importance des sous-programmes) par rapport à la logique extérieure : chaque ordre n'est interprété qu'une seule fois, quel que soit le nombre de fois où il doit être exécuté. En effet, ce n'est que dans la lère phase, celle de perforation, qu'il y a un programme interprétatif. Or pendant cette phase les bouclages ne fonctionnent pas. Ainsi les ordres qui doivent, par bouclage, être exécutés 1000 fois ne sont interprétés qu'une seule fois. C'est pendant la 2ème phase, celle où le programme est en langage machine et ne nécessite plus de partie interprétative, que les bouclages fonctionnent. Un autre gain de temps est dû au fait qu'un programme une fois autoprogrammé peut être réutilisé à volonté, toute phase interprétative étant supprimée; ceci n'est absolument pas le cas en logique extérieure où il faut tout recommencer à chaque utilisation du programme.

#### II - D - AVANTAGES DE LA COEXISTENCE DES 2 PROCÉDÉS.

Le C. D. P. est l'association d'une logique extérieure et d'un procédé d'autoprogrammation (compilateur). Les ordres complexes propres au C. D. P. peuvent en effet être interprétés soit en logique extérieure soit en autoprogrammation. Ceci permet de bénéficier pour un même programme de la commodité de mise au point en logique extérieure et de la rapidité

d'exécution après l'autoprogrammation. La plupart des programmes seront traités d'abord en logique extérieure pour être vérifiés puis, s'ils sont répétitifs ou d'utilisation fréquente, seront transformés par l'autoprogrammation en programmes en langage machine.

-------

Marie 114 to State Miles in the Miles

A second of the second

THE PARTY OF THE PROPERTY.

2

g ann - Sianes

2 3 Contract

and engage

CALL STREET, SHELL

P Week at the country of the second

The college person at the property

water and a color of a state of

real transfer of the second transfer of the s

\_\_\_\_

#### III - ORGANISATION GENERALE DU C.D.P.

#### III - A - STRUCTURE

Dans ce chapitre, on entend par C. D. P. aussi bien le C. D. P. "logique extérieure" que le C. D. P. "autoprogrammation".

Le C. D. P. est constitué d'une part d'une partie centrale (tronc commun) comportant les 4 opérations en virgule fixe et les ordres logiques ou de routine, et d'autre part de sous-programmes à 2 ou 3 adresses. Cette structure permet une très grande souplesse du code de programmation et une possibilité énorme d'extension, limitée seulement par la capacité du tambour.

#### III - A - 1 - Partie centrale

La partie centrale, qui sera étudiée en détail ultérieurement, comporte principalement :

- les opérations en virgule fixe : addition, soustraction, multiplication avec ou sans décalage, division avec ou sans décalage. Ces ordres sont indexables.
- des ordres logiques et de routine : arrêt, saut inconditionnel, retour en langage machine, lecture, perforation, tests (dont un test de signe à 3 branches), ordre de bouclage puissant pouvant modifier automatiquement un registre d'index, etc....
- des ordres d'entrée dans des sous-programmes. Ces ordres sont indexables,

#### III - A - 2 - Sous-programmes - Divers C. D. P.

Pour obtenir un C. D. P. utilisable, il faut ajouter au tronc commun les sous-programmes utiles. Ce sont par exemple des sous-programmes de calcul de fonctions, ou d'opérations en virgule flottante simple ou double précision, etc.... Mais il serait peu commode d'être obligé de s'occuper de la mise en place des sous-programmes courants chaque fois qu'ils sont utilisés. On a donc constitué divers groupes de sous-programmes pour former, en les ajoutant au tronc commun, des C. D. P. d'utilisation facile. Il en existe trois actuellement (voir le chapitre XII):

- le C. D. P. en virgule fixe simple précision groupant autour du tronc commun les sous-programmes de calcul en virgule fixe de fonctions courantes: racine carrée, sinus, cosinus, logarithmes, exponentielle, Arc sinus. Ces calculs ont lieu avec ou sans décalage.
- le C. D. P. en virgule flottante simple précision groupant autour du tronc commun les sous-programmes d'opérations et de calcul de fonctions courantes en virgule flottante FLAIR (dans cette convention, 1 est noté 5112400000).

Chacun de ces C. D. P. est extensible à volonté dans la limite de capacité du tambour. D'autre part si un des sous-programmes n'est pas utilisé dans un programme, sa place peut être récupérée.

#### III - B - PAQUETS DE CARTES . . .

Quand un programme a été écrit en C. D. P., il est utilisable en logique extérieure et en autoprogrammation. Si l'on veut l'exécuter en logique extérieure, on le charge avec le paquet interprétatif "logique extérieure". Si l'on veut l'interpréter en autoprogrammation, on le charge avec le paquet "autoprogrammation". C'est le choix du paquèt qui fixe le choix du procédé.

Comme on l'a vu dans la 2ème partie du chapitre II - B, pour exécuter un programme perforé par l'autoprogrammation, il faut lui ajouter les sous-programmes utiles. En instituant 3 C. D. P. (virgule fixe, virgule flottante simple et double précision) on a établi 3 paquets de sous-programmes à joindre aux programmes autoperforés. Ces paquets sont appelés paquets pour "l'exécution en langage machine" ou "paquets verts".

En résumé, pour chaque modèle de C. D. P. (virgule fixe, virgule flottante, etc...) existent 3 paquets de cartes :

- le paquet "logique extérieure" comportant la partie interprétative en logique extérieure et les sous-programmes.
- le paquet "autoprogrammation" ou "autoperforation" comportant la partie interprétative en autoprogrammation.
- le paquet "exécution en langage machine" comportant les sous-programmes.

.../..

#### IV INSTRUCTIONS DU C. D. P. (tronc commun)

#### IV - A - STRUCTURE DES ORDRES

CHAIL COLLEGE TO THE PARTY OF T

The state of the s

La plupart des logiques extérieures adaptées à l'I. B. M. 650 utilisent, comme le langage machine, des ordres à 10 chiffres. Mais ce format a été jugé insuffisant, les ordres étant alors trop pauvres en informations. En particulier les ordres ne peuvent avoir 3 adresses et une varieté suffisante de codes, et l'indexage est peu pratique. Si 10 chiffres sont jugés insuffisants, il est à peu près obligatoire de passer à 20 chiffres, ce qui a été fait pour la plupart des ordres du C. D. P. Ces ordres occupent 2 mémoires consécutives. Cependant certains ordres dits "ordres de service" ne nécessitent pas l'encombrement de 2 mémoires : ces ordres n'ont que 10 chiffres. Enfin, un petit nombre d'ordres à 2 mémoires nécessitent des indications supplémentaires (décalages). On place alors ces indications dans une 3ème n.émoire qui suit les 2 autres.

#### IV - A - 1 - Ordres à 2 mémoires

Ces ordres cont des ordres à 3 adresses, chaque adresse étant indexable. Le C. D. P. simule 9 registres d'index numérotés de I à 9. Les registres CTR1, CTR2,...., CTR9 sont respectivement les mémoires 1968, 1959,....., 1976. Leur contenu est toujours positif ou nul et inférieur à 1000. Il est de la forme 00 0000 0xxx. Un 10ème registre d'index, CTR0, contient toujours zéro. Il est simulé par la mémoire 1967. Après le chargement du C. D. P., les registres d'index ne sont pas à zéro. Ils le seront par l'ordre d'initialisation (voir plus loin). Seul le CTR0, c'est-àdire la mémoire 1967, est dès le début à zéro.

Lorsque nous considérerons un ordre quelconque en C.D.P., nous supposerons que son adresse est n. Si c'est un ordre à 2 mémoires, il sera contenu en n et n + 1. La composition des ordres à 2 mémoires est la suivante :

PP' N i A	j B k C		
x x xxx x xxxx	x xxxx x xxxx		
Positif	Signe ignoré		
Contenu en n	Contenu en n + 1		

- PP' est un code d'opération
- N est l'adresse de l'instruction suivante (comprise entre 000 et 999)
- A, B et C sont des adresses (indexables)
- i, j, k, compris entre 0 et 9, indiquent respectivement qu'il faut ajouter à l'adresse A le contenu du registre d'index n° i, à B celui du n° j, à C celui du n° k.

Remarque : les ordres indexés sont exécutés en tenant compte des index mais ne sont pas modifiés eux-mêmes (ils restent intacts dans leurs mémoires),

#### IV - A - 2 - Ordres à 1 mémoire

Ils sont négatifs et de la forme

-	PP	U	V
	x x	xxxx	xxxx

- PP' est un code pouvant varier entre 00 et 15 Pour l'instant, seuls les codes 00 à 09 ont été utilisés (sauf 08).
- U et V sont des adresses non indexables, assimilables dans la plupart des cas respectivement à l'adresse facteur et à l'adresse instruction suivante d'un ordre en langage machine.

.../...

#### IV - B - DETAIL DES ORDRES

#### IV - B - 1 - Ordres à 2 mémoires

00 N 0A jB kC Entrée dans un sous-programme en langage machine à 2 adresses (calcul de fonction).

A est l'adresse de la lère instruction du sous-programme. La donnée se trouve en B; le résultat est envoyé en C (au sujet de la standardisation de ces sous-programmes, voir le paragraphe VII - C - 2 - a).

Remarque: le sous-programme peut très bien être conçu pour utiliser des données placées en B, B + 1, ...., et envoyer des résultats en C, C + 1, .....

Pour ces 2 ordres, si le résultat a plus de 10 chiffres, il y a "dépassement de capacité" comme en langage machine.

#### 03 N iA jB kC (A) x (B) en C

- Si le produit comporte plus de 10 chiffres (la machine fait un test GNN), la machine s'arrête avec au pupitre 01 1333 xxxx. En logique extérieure, on peut repartir en appuyant sur "départ programme". Après ce départ, le contenu de l'accumulateur droit (produit incomplet) est transféré dans la mémoire C et le programme continue. L'ordre est analysé normalement.
- Il est commode de pouvoir commander des décalages sur le produit avant le test. Dans ce cas on utilise l'ordre 05 au lieu de 03.

#### 04 N iA jB kC (A): (B) en C

- Pour effectuer la division, le programme met le dividende (A) dans l'accumulateur droit.

- Si le quotient est trop grand, il y a un "dépassement de capacité" comme en langage machine.
- -Il est commode de pouvoir commander des décalages sur le dividende. Dans ce cas, on utilise l'ordre 06 au lieu de 04.

#### 05 N iA jB kC (A) x (B) en C avec décalage. Ordre à 3 mémoires.

- L'ordre de décalage à exécuter sur le produit doit être placé dans la mémoire qui suit celle contenant jB kC. Il doit avoir la forme suivante :

  3 x 000 m 1939; cet ordre, à part l'adresse instruction suivante, est exactement l'ordre de décalage en langage machine qui s'exécute (x = 0 ou 1 ou 5 ou éventuellement 6).
- L'arrêt 01 1333 xxxx peut aussi se produirepour l'ordre 05.
- 06 N iA jB kC (A): (B) en C avec décalage. Ordre à 3 mémoires.

  L'ordre de décalage (en langage machine) doit être placé dans la mémoire qui suit celle contenant jB kC, et doit avoir la forme suivante: 3x 000m 1991. Il s'exécute avant la division.

#### 07 N OA OB kC Aiguillage à 3 branches (test SI)

Si (C) < 0 Aller à N

Si (C) = 0 aller à A

Si (C) > 0 aller à B

L'ordre d'exécution des tests est : 1° ANG ; 2° GNN.

#### 08 N 0A 0B kC Saut conditionnel suivant une position de la mémoire C.

B a la forme suivante : 000B. La machine amène le contenu de C dans le distributeur et fait un test SD sur la position n° B de ce distributeur. B = 0 désigne la position 10. Si c'est un 8, le programme est aiguillé en N; si c'est un 9, il va en A; si ce n'est ni un 8 ni un 9, le programme s'arrête sur le test SD. L'adresse C peut être 8000.

.../...

#### PP' N iA jB kC $P = 1, 2, 3, 4, 5 - PP' \neq 49$ . Entrée dans un sous-programme à 3 adresses.

Les données se trouvent en A et B, le résultat est envoyé en C. L'adresse de la lère instruction du sous-programme doit être placée, pour la logique extérieure et l'autoprogrammation, en 1875 + PP' sous la forme 00 0000 xxxx. (Pour la standardisation, voir le paragraphe VII - C - 3).

Remarque: le sous-programme peut être conçu pour utiliser des données placées en A, A + 1, . . . . . et B, B + 1, . . . . , et envoyer les résultats

# en C, C + 1, ..... 9i N of H 0B 0C Bouclage

Cet ordre commande : exécuter H fois (H)1) le programme commençant en B, en ajoutant chaque fois C (et non son contenu) au contenu du registre d'index n° i et en utilisant le compteur de boucles n° a. A la sortie de boucle, aller à N.

La boucle a alors été décrite H fois (répétée H - 1 fois). A la sortie de boucle, le compteur de boucles n° a et le registre d'index n° i sont remis à zéro.

Grâce à l'existence de 10 compteurs de boucles, on peut exécuter jusqu'à 10 boucles intérieures les unes aux autres. Des boucles intérieures les unes aux autres doivent avoir des conférents. Si une boucle possède une boucle intérieure, il ne faut pas lui attribuer conférence et aux autres doivent avoir des conférence et aux autres doivent avoir des conférence et aux autres doivent avoir des conférences et aux autres doivent aux autres de conférences et aux autres de conférences e

Remarque 1: si l'on sort de la boucle avant l'exécution des H tours (ceci peut être causé par un test ou un aiguillage à distance), il faut remettre à zéro le compteur de boucles n°  $\chi$  et le registre d'index n° i (si l'on doit ré-utiliser ceux-ci sans refaire d'initialisation). De plus, dans ce cas, il ne faut pas utiliser  $\chi = 0$ .

Remarque 2 : si une boucle ne possède pas de boucle intérieure, il est recommandé de lui attribuer  $\alpha=0$  (car alors les calculs sont plus courts pour l'exécution en langage machine) sauf dans le cas cité en remarque l.

Remarque 3 : si i = 0, aucun registre d'index n'est modifié, ce qui évite de détruire le registre d'index n° 0 qui est par définition à zéro : l'adresse C est alors ignorée.

Remarque 4: si, par suite d'une erreur, le compteur de boucles n° of n'a pas été remis à zéro avant l'ordre de bouclage et contient une valeur supérieure ou égale à H, la boucle n'est pas répétée : le compteur de boucles et le registre d'index sont remis à zéro et le programme continue en N.

- OO N, 01937 jB kC Modification d'un ordre de bouclage.

  Cet ordre est un sous-programme à 2 adresses placé à poste fixe. Il remplace le nombre H de boucles de l'ordre de bouclage placé en C et C + 1 par H' contenu en B sous la forme 00 0000 H'.
- 49 N iA jB kC Perforation d'une séquence en n mots par carte. Cet ordre, qui est un sous-programme à 3 adresses placé à poste fixe, commande la perforation à A mots par carte du contenu des mémoires B à C comprise.

  Il existe un autre ordre de perforation, 06 U V.

□ 株 (m) × (m) (m) (m) (m)

#### IV - B - 2 - Ordres à 1 mémoire (ordres de service)

- 00 U V Saut inconditionnel à V.
- -01 U V Arrêt inconditionnel repéré, c'est-à-dire que U apparaît dans le registre programme (01 U xxxx). On peut repartir à V.
- 02 U V Initialisation. Cet ordre provoque la remise à zéro des compteurs de boucles et registres d'index (ainsi que les mémoires 1950 à 1966). En exécution en langage machine, il restaure aussi le contenu de la mémoire  $\Delta_{\text{bouclage}}$ . U est ignoré.
- 03 U V Remise à zéro de la mémoire U.

- 04 U V Entrée dans le distributeur du contenu de U et arrêt par l'ordre 01 U xxxx, Cet ordre sert à vérifier un résultat en cours de calcul sans faire d'arrêt prédéterminé.

about the difference in the

proceduring at . to discovering a

- 05 U V Sortie de code. Cet ordre envoie en langage machine à l'instruction U. En autoprogrammation, la machine perfore le programme en langage machine (voir le paragraphe X - D - 2). Pour rentrer en code, il faut affecter de l'adresse instruction 1800 le dernier ordre en langage machine.

- 06 U V Perforation. Ordre équivalent à 71 U V en langage machine. Il existe aussi 49 N iA jB kC.
- 07 U V Lecture. Ordre équivalent à 70 U V à condition que les cartes lues soient "non chargement".
- -09 U V Fin de séquence (voir le paragraphe VII B 2). Cet ordre envoie à U en logique extérieure et exécution en langage machine, mais à V en autoprogrammation.

.../..

П

... ...

residence a starre of the

#### V ECRITURE DES PARTIES INTERPRETATIVES

En logique extérieure et en autoprogrammation, la partie interprétative a été écrite en PASØ II, qui se prête beaucoup mieux aux transformations de programmes et s'écrit plus facilement que le vrai langage machine. De nombreuses modifications qui auraient été pénibles et dangereuses en langage machine se sont faites très aisèment en PASØ. Dans la suite, chaque adresse sera désignée par son nom en PASØ, suivi de l'adresse réelle attribuée par l'assemblage. Le numéro de l'instruction dans l'assemblage PASØ permet de repérer plus rapidement les instructions sur la liste située à la fin de cette brochure. Ce numéro, précédé de "n°", sera quelquefois indiqué à la suite des adresses.

Certaines mémoires ont la même utilisation en logique extérieure et en "exécution en langage machine" ou en autoprogrammation. Les principales de ces mémoires sont :

- DEBUT = 1800, adresse de la lère instruction de la partie interprétative en logique extérieure comme en autoprogrammation.
- MEMNN = 1884, mémoire contenant l'adresse de l'instruction suivante 00 0000 N.
- -huit mémoires de travail, W 0000 = 1990 à W 0007 = 1997, employées pour placer des résultats ou des indications intermédiaires. Ces mémoires de travail sont utilisables dans les sous-programmes mais, d'un ordre en C. D. P. à un autre, ne peuvent pas conserver d'indication. Les mémoires W 0008 = 1998 et W 0009 = 1999, non utilisées par les parties interprétatives, mais disponibles, peuvent au contraire conserver des indications d'un ordre à l'autre. Cependant, si l'on exécute une analyse, on ne peut pas utiliser 1999 comme adresse C car alors l'analyse doit chercher aussi le contenu de C + 1 = 2000 et cela provoque une "erreur adresse".

- les mémoires P 0001 = 1977 à P 0010 = 1986 qui sont des mémoires de perforation utilisées aussi bien par la logique extérieure pour l'analyse que par l'autoperforation.
- les compteurs de boucles H 0000 = 1940 à H 0009 = 1949 et les registres d'index C 0000 = 1967 à C 0009 = 1976.

Les programmes ont été écrits et surtout vérifiés morceau par morceau.

....

#### VI - PARTIE INTERPRETATIVE EN LOGIQUE EXTERIEURE :

#### GENERALITES

Interpréter un ordre en logique extérieure consiste à former, en tenant compte des indications données par cet ordre, des instructions en langage de machine permettant d'effectuer les opérations demandées par l'ordre.

Après les avoir exécutées, la machine va chercher l'ordre suivant sur lequel elle recommence ce processus.

#### VI - A - SCHEMA GENERAL.

L'interprétation d'un ordre en C.D.P. commence toujours par l'instruction placée en DEBUT = 1800 (n° 41). La seule indication demandée par la machine à ce moment est l'adresse n de l'ordre à interpréter. Cette adresse doit être placée sous la forme 00 0000 n dans la mémoire MEMNN = 1884. L'exécution en logique extérieure d'un ordre se passe donc suivant le schéma suivant :

- recherche de l'ordre dans sa ou ses mémoires (commence en 1800)
- découpage de l'ordre et séparation des diverses indications.
- mise en place de l'adresse de l'instruction suivante en MEMNN = 1884
- formation des ordres en langage machine et exécution
- retour à DEBUT = 1800.

La succession de ces phases n'est pas toujours exactement celle-ci.

.../...

#### VI - B - ANALYSE.

Pour tous les ordres, si le signe du pupitre est -, la machine perfore une carte d'analyse. Cette carte porte les renseignements suivants :

- Pour les ordres à 1 mémoire :

Mot 3:

Mot 1: 00 n 0000

Mot 2: instruction (en a)

Pour ces ordres, la carte d'analyse sert à signaler la présence de l'ordre dont la fonction est en général évidente.

- Pour les ordres à 2 mémoires sauf le bouclage :

Mot 2: 
$$\begin{cases} \text{instruction (en n et n + 1)} \end{cases}$$

(CTRi) représente le contenu du registre d'index n°i. Ce motipermet de connaître le contenu de tous les registres d'index. Ce contenu doit toujours être positif et inférieur à 1000.

Mot 5: Contenu de A

Mot 6: Contenu de B

Mot 7: Contenu de C Résultat

Mot 8: Contenu de C + 1

Les mots 5, 6, 7, 8 représentent le contenu des mémoires après l'exécution le l'ordre.

#### - Pour le bouclage

Mot 1: 00 n 0000

Mot 2:

instruction (en n et n + 1)

Mot 3:

Mot 4: Contenu du compteur de bouclesn°≪ après sa

modification.

Mot 5: Contenu du registre d'index n° i après sa modification.

La perforation de la carte d'analyse se fait dans la bande 1950. Autrement dit, le mot 1 wient de 1977 que nous appellerons P 0001, le mot 2 de 1978 ou P 0002, ......, le mot 8 de 1984 ou P 0008.

#### VI - C - ORGANIGRAMME (voir figure page 22bis)

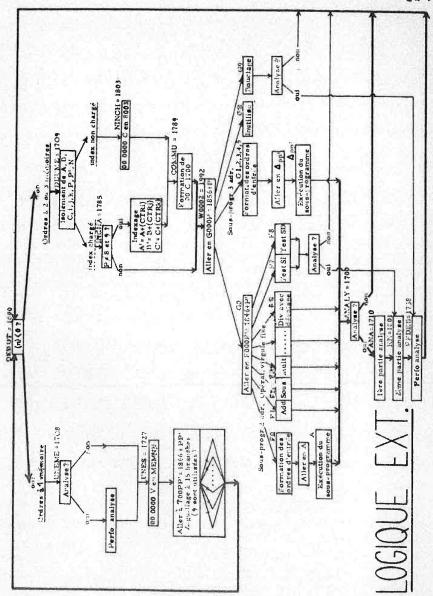
Nous noterons n l'adresse de l'instruction considérée (contenue en n; ou n et n + 1; où n, n + 1 et n + 2) et N l'adresse de l'instruction suivante.

La première opération que fait la machine est d'amener dans l'accumulateur droit le contenu (n) de n et de tester son signe. Dès cet endroit se séparent les trajets suivis pour les instructions à 1 mémoire et celles à 2 mémoires. C'est en effet le signe de (n) qui les différencie.

VI - C - 1 - Instructions à 1 mémoire (cas où le contenu de n est négatif).

La lère instruction de cette branche est l'instruction UNEME = 1708

(n° 186). Elle commence une partie dans laquelle ont été groupées toutes les opérations valables à la fois pour tous les ordres de service. Cette partie comporte :





- D'abord l'analyse: pour les ordres de service, les seules indications intéressantes sont l'adresse de l'instruction et l'instruction elle-même.

Or, dès l'exécution de UNEME = 1708 ces indications sont en place, la première en P 0001 = 1977, la seconde en P 0002 = 1970 (voir le détail de la partie interprétative). On peut donc dès le début traiter le problème de l'analyse. Pour cela, le contenu de la mémoire "pupitre" est amené dans l'accumulateur et testé. S'il est négatif, cas où l'on désire l'analyse, la carte d'analyse est perforée. Sinon aucune carte n'est perforée.

- Ces 2 cas se rejoignent à l'instruction UNES = 1727 où débutent l'envoi de 00 0000 V en MEMNN = 1864, c'est-à-dire la mise en place de l'instruction suivante, et la préparation du test à branches multiples différenciant les branches propres à chaque code PP'. En fait, V n'est pas toujours l'instruction suivante. Mais on a intérêt pour gagner des mémoires à placer son envoi dans la partie commune.

- Le test est un test à 15 branches, laissant ainsi la possibilité d'utiliser 15 codes PP' différents (9 seulement ont été utilisés jusqu'à présent). Ce test est très simple. Il en existe de nombreux autres du même type dans les parties interprétatives, que ce soit en logique extérieure ou en autoperforation. Il consiste à faire débuter la branche relative à PP' = 00 à une instruction d'adresse: "constante + 00"; celle relative à PP' = 01

à "constante + 01"; etc ..... Cette forme peut être notée par des adresses régionales en PASØ. La région des lères instructions des branches relatives aux ordres à 1 mémoire sera par exemple désignée par la lettre T. La branche relative au code PP' commencera donc en T 00PP' = 1866 + PP'. Pour former l'ordre d'aiguillage (aller à T 00PP') il suffit d'ajouter à 00 0000 00PP' la constante 00 0000 T0000 = 00 0000 1866. Le mot ainsi formé représente alors l'ordre d'aiguillage.

- Exécution du test : il suffit pour cela d'exécuter l'ordre formé, c'est-àdire 00 0000 T 00PP' = 00 0000 1866 + PP'.
- Parties propres à chaque code PP': elles seront passées en revue au cours de l'étude détaillée de la partie interprétative.
- Toutes les parties particulières se terminent par un retour à DEBUT = 1800. La seule exception apparente est celle du code 05 (retour en langage machine). Cette exception n'est que fictive, car la fin de son exécution (c'est-à-dire, au point de vue de la logique extérieure, la fin de l'exécution du programme en langage machine) doit être marquée par un retour à 1800, c'est-à-dire à DEBUT (voir étude détaillée).
- VI C 2 Instructions à 2 mémoires : (cas où le contenu de n est positif).

  La lère instruction de cette branche est l'instruction DEUME = 1709
  (n° 48)

a) Partie commune:

La partie commune se décompose en 4 parties principales.

- α) Le découpage des ordres. Ce découpage sera envisagé dans l'étude détaillée de la partie interprétative.
- extérieure, si aucun ordre d'un programme n'est indexé, on peut supprimer une petite portion de la partie interprétative (matérialisée par un paquet d'indicatif spécial, le paquet "index").
- Si le paquet "index" n'est pas chargé, il ne reste, après le découpage des ordres, rien d'essentiel à faire par la partie interprétative commune à tous les ordres. Par commodité, la machine fabrique, en COMMU = 1789 (n° 73) le mot 00 C 1700 très utilisé par la suite. Avant d'aller à GOMMU = 1789, la machine doit exécuter une instruction (NINCH = 1803) qui permet d'être à des états équivalents, lorsqu'on arrive à COMMU = 1789, que le paquet "index" soit chargé ou non.
- Si le paquet "index" est chargé, la partie interprétative doit indexer les adresses. Mais l'indexage n'est pas valable pour les ordres tels que P = 8 ou 9. Pour ces ordres, la partie interprétative commune n'a plus rien à faire après le découpage, si bien que pour P = 8 ou 9, le programme va directement à l'aiguillage gouverné par P.

.../...

.....

Pour les ordres tels que  $P \neq S$  et 9, l'indexage consiste à ajouter à l'adresse A le contenu du registre d'index n° i, à B celui du n° j et è C celui du n° k. Si un, deux ou les trois indices sont nuls, l'indexage a quand même lieu, Mais, pour cela, il est essentiel que le contenu du registre n° 0 soit nul.

Remarque: il serait possible de faire un test sur la nullité simultanée de i, j, k pour ne pas inutilement exécuter les opérations d'indexage.

Mais le gain de temps serait assez minime et la perte de place assez sensible. Or, en logique extérieure, puisqu'il existe une autoprogrammation, le temps n'est qu'un facteur secondaire par rapport à l'encombrement. En autoprogrammation, par contre, ce test a lieu.

- () Après l'indexage, la branche "index chargé" rejoint la branche "index non chargé" en CØMMU = 1789. Après la formation de 00 ( 1700 (c'est-à-dire en PASO: 00 ( ANALY), c'est à l'aiguillage multiple gouverné par P que l'on arrive. L'ordre d'aiguillage a été formé dans la mémoire W 0002 = 1992 au cours du découpage des ordres.
- 5) Aiguillage gouverné par P. Il serait trop encombrant d'effectuer, comme pour les ordres de service, un aiguillage ayant autant de branches qu'il existe de valeurs de PP'. On a donc commandé d'abord un aiguillage suivant P permettant de distinguer les différents genres

d'ordres. Puis, pour certaines valeurs de P, un aiguillage suivant P'
est nécessaire. L'aiguillage suivant P envoie à l'instruction d'adresse
régionale G 000P = 1856 ÷ P. Ainsi la branche commençant en
G 0000 = 1856 est relative aux sous-programmes à 2 adresses et aux
opérations en virgule fixe. Les branches commençant en G 0001 = 1857,
G 0002 = 1858,....., G 0007 = 1863 se rejoignent et sont relatives
aux sous-programmes à 3 adresses. En fait P = 6 et 7 ne sont pas
utilisables; cette restriction a été apportée pour gagner de la place
sur le tambour, car le nombre de 50 sous-programmes à 3 adresses
utilisables simultanèment a été jugé tout-à-fait suffisant. La branche
commençant en G 0008 = 1864 n'existe pas encore. Elle est réservée à
d'éventuels ordres de forme spéciale. Quant à la branche commençant
en G 0009 = 1865, elle est relative à l'ordre de bouclage.

Remarque: Il peut paraître curieux qu'un aiguillage à 10 branches contienne 7 branches se rejoignant aussitôt. C'est cependant le procédé de différenciation qui a paru le plus économique et le plus souple. S'il n'avait pas été employé, il aurait fallu avoir recours à des tests.

- b) Branche particulière à chaque valeur de P.
- a) P = 0: la machine fait immédiatement un aiguillage multiple sur P'.

Les branches particulières à chaque P' commencent à l'adresse régionale F 000P' = 1846 ÷ P'. Les branches commençant aux adresses F 0001 = 1847 à F 0006 = 1852, c'est-à-dire correspondant aux codes PP' = 01 à 06, sont relatives aux ordres d'opérations en virgule fixe. Elles rejoignent la plupart des autres branches avant la partie "analyse". Les branches F 0007 = 1853 et F 0003 = 1854 correspondant aux ordres de test PP' = 07 et 08 ont leur propre début d'analyse. Elles rejoignent la plupart des autres branches en cours d'analyse ou à DEBUT = 1800 suivant que l'on désire ou non l'analyse. La branche P' = 0, correspondant aux ordres d'entrée dans les sousprogrammes à 2 adresses, commence par la formation des ordres d'entrée dans le sous-programme et se poursuit par l'exécution effective du sous-programme. Elle rejoint la majorité des branches à l'analyse.

β) P = 1 à 5 : il s'agit des ordres d'entrée dans les sous-programmes à 3 adresses. La branche commune à tous ces ordres consiste en la formation de l'entrée dans le sous-programme voulu. Elle est suivie de l'exécution de celui-ci, exécution commençant à l'instruction placée en Δ<sub>pp'</sub> (cf chapitreVII-C-3); cette adresse représente le facteur de translation du sous-programme, si la lère instruction est en Δ+0. L'adresse Δ<sub>pp'</sub> ne peut pas, comme pour les sous-programmes à

.../...

2 adresses, figurer dans l'ordre. Elle est indiquée sous la forme

00 0000 \$\Delta\_{\text{pp'}}\$ dans la mémoire 1875 \(\div \text{PP'}\). Il y a ainsi une correspondance facile entre le code et l'adresse de la première instruction du sous-programme. L'exécution du sous-programme terminée, le programme rejoint la plupart des autres branches au début de l'analyse.

- 8) P = 8 n'est pas encore créé.
- §) P = 9 correspond à l'ordre de bouclage. Cet ordre à structure très spéciale ne peut pas s'adapter à l'analyse commune. Il possède donc son propre programme d'analyse et ne rejoint l'ensemble des branches qu'à l'ordre de perforation de l'analyse ou à DEBUT = 1800 selon que l'on désire ou non l'analyse. Le programme de bouclage a été écrit sous forme d'un sous-programme (non standard) translatable, ce qui a permis, au cours des divers assemblages de la partie interprétative, de garder en formation compacte la partie réservée au bouclage. Le facteur de translation de ce sous-programme est placé dans la mémoire 1875 + 61 = 1936 sous la forme 00 0000 Δ = 00 0000 1596. Ce mot représente d'ailleurs l'ordre "aller au sous-programme de bouclage".

.../...

c) Analyse: pour la plupart des ordres, le même programme d'analyse est utilisé. Ce programme, qui sera étudié en détail plus loin, commence par un test sur le signe du pupitre. Si ce signe est +, le programme d'analyse ne doit pas être déroulé et l'on est envoyé en DEBUT = 1800 où commencera l'interprétation de l'ordre suivant. Si ce signe est -, le programme d'analyse se déroule. Il se termine par la perforation d'une carte et le retour à DEBUT = 1800.

VII - ETUDE DETAILLEE DE LA PARTIE INTERPRETATIVE

Les N° placés en tête des tableaux suivants représentent le numéro de chaque ligne dans l'assemblage.

VII - A - PARTIE COMMUNE A TOUS LES ORDRES : commence en DEBUT = 1800 Au début de cette partie, se trouve dans la mémoire MEMNN = 1884 la quantité 00 0000 n indiquant l'adresse n de l'ordre à interpréter.

No Adresse	C. 0.	A. F.	A. I.	(8003)	(8002)	(8001)	
41 DIEBUT	RIAD	M EMNN		00	0Cn		
42	D, AG	10004	1		00 n 0000		
43 1	T RD	P 0001	10.00		100	00 n 0000	en Pl (pour analyse)
44	EIDI	GIG	G I	ON -1.		65 xxxx GH	
45 G G	65	0000	G H				
46 GII	SIBF	glg ,	18001			65 n GH	en GG
(8001	R AD	n	GН	00	(n)		
47 G.H	A, NG	U NEME	D EUME		10 1	1 1	Test sur (n)
E. F.		(UNE ME	noire) (	DEUx MEmoire	(s)		

Cette séquence de programme sert à former l'ordre 65 n GH qui amène le contenu de n dans l'accumulateur, puis à exécuter cet ordre et à faire le test qui envoie : en UNEME = 1708 si l'ordre est à l mémoire et en DIUME = 1709 s'il est à 2 mémoires.

#### VII - B - ORDRES / 1 MEMOIRE OU ORDRES DE SERVICE

VII - B 1 ) Partie commune à ces ordres : débute à UNEME = 1708 (n° 186)

		100	-PP'UV	-PP'U V	
186 U NEME	T RD P10002			-PP'UV	en P2 (pour analyse)
187	R AG 8000				tu - 3 0
1.88	A!NG U!NES				Analyse ?
189	P FØ P 0001 U NES		1		Perfo si analyse
190 U NES	R AV P 0002	00	PP' U V		2
191	E DI 8003			00	
192	S BI M, EMNN			00 0000 V	en MEMNN = 1884
193	D AG 0002	0OPP'	U V 00		
194	A AG C TEA 8003	00(TOOPP')	U V 00	0OTOOOO	Formation aiguillage
(8003	) SIPP 10000 T100PP				Aiguillage

- La carte d'analyse porte en mot 1 le contenu de P 0001 = 1977, soit 00 n 0000 et en mot 2 le contenu de P 0002 = 1978, soit PP' U V.
- V, dans la plupart des cas, représente l'adresse de l'instruction suivante.
  Le programme envoie donc 00 0000 V en MEMNN = 1884 dans la partie commune car cela est plus rentable au point de vue de l'encombrement.
- La mémoire CTE 4 = 1775 contient 00 0000 T0000 = 00 0000 1866.

#### VII - B - 2 - Partie propre à chaque code PP' :

= Code - 00 U V : sans opération, aller à V.

217 T 0001 S OP 0000 DEBUT

Le seul but de cette partie de programme est d'envoyer à DEBUT = 1800. En effet, lorsqu'on arrive à DEBUT = 1800, c'est l'adresse V qui est dans la mémoire MEMNN = 1884 car elle a été placée par la partie commune aux ordres à 1 mémoire. C'est donc bien l'instructionV qui est exécutée ensuite.

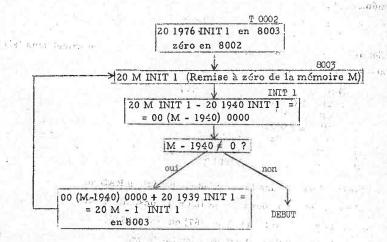
= Code - 01 U V : arrêt ; après l'arrêt, aller éventuellement à V.

199 T 0001	E DI i	A RDEB	D ADSB	0-0 TOOO1	U V 00 01 2000 DEBUT
197 D ADSB	D'AD	,0002	S BFW1	0 0xx	
198   S BFW1	SBF	W,0001	8001	A 4	01 U DEBUT
(8001)	ART	U	D. EBUT		

- Les 3 premiers ordres forment l'ordre d'arrêt 01 U DEBUT. Cet ordre a comme adresse facteur U pour pouvoir être repéré. En effet, il est très utile lorsque la machine s'arrête de savoir quel ordre d'arrêt a fonctionné. L'adresse instruction suivante est DEBUT = 1800 pour que, si l'on veut faire repartir le programme, l'instruction V soit exécutée.
- La mémoire ARDEB contient la constante 01 0000 DEBUT = 01 0000 1800.
- Remarque: L'ordre de décalage pourrait s'exécuter avant la mise en place de 01 0000 DEBUT dans le distributeur. Mais cela supprimerait la possibilité d'utiliser l'ordre DADSB pour d'autres codes que 01 U V.
- = Code 62 U V : Initialisation,

Cet ordre sert à remettre à zéro les mémoires 1940 à 1976, c'est-à-dire les compteurs de boucles et d'index plus les mémoires 1950 à 1966. On verra qu'en exécution en langage machine (après autoprogrammation) cet ordre a un rôle supplémentaire.

L'organigramme suivant montre le déroulement de la partie interprétative propre à l'ordre - 02 U V -



						1	100000000000000000000000000000000000000
206	T 0002	RAG	1	8003	20 1976 INIT1 0 0		
207	1	20	1976	I NITL	30 pm 1 2		1 1 1
	(8003)	TRD	l M	I NIT1		0 0	en M
208	I NITI	SAG	I NIT2	ी भिन्ने खेलर	00 M-1940 0000	20 1940 INI T 1	a solativity St
209	1,50 1	GNN		DEBUT	E 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		M-1940≠0?
210		AAG	1	8003	20 M-1 INIT 1	20 1939 INIT 1	Transformation
211		20	11939	INITI	29 P - 24 2 20 P		de l'ordre de R./
	(8003)	TRD	M-1	I NIT1	BURNEY OF THE PARTY OF		100
212	I NIT2	20	1940	I NIT1		L v sustil	Constante
		,					

≥ Code - 03	U V : remise à	zéro de la mémoire U	10 to 2 - 10
214 T 0003 SAG	1 00001	0OT0003 U V 00	
214 1 0003 SAG 215 EDI	DIADSB	0, 0	21 0000 DEBUT
215 E DI 216	DiADSB 0000 DiEBUT		6 THE 1 THE

Les ordres DADSB = 1679 (n° 197) et SBFW1 = 1825 (n° 198) déjà utilisés pour l'ordre - 01 U V permettent de former dans le distributeur l'ordre 21 U DEBUT qui s'exécute alors, envoyant le contenu de 8003, c'est-à-dire zéro, dans la mémoire U.

= Code - 04 U V : entrée dans la distributeur du contenu de U at arrêt par l'ordre 01 U xxxx. Aller ensuite à V.

Cet ordre est utile pour contrôler un résultat en cours de calcul sans faire

	d'arrê	t prédé	terminé,		O OT 0004	u v oo	F		-7-11
2 18	T 0004	D'AD	10002		00 xx				
219		E DI	R GOW2	71 -2 3	1344		60 0000	W0002	
220		SBF	W 0001	(6) (2)			60 U	W0002	en W0001
221		E DI	ARDEB				01 0000	DEBUT	
222		SBF	W, 0002	W,0001	Acres 1		01 U	DEBUT	en W0002
	(W 0001)	RAG	, n	W 0002	00	(U)	(U)		
	(W10002)	ART	U .	D! EBUT					

- L'ordre d'entrée dans le distributeur estrici un RAG au lièu de EDI. La seule raison est que cela permet d'utiliser la constante 60 0000 1992 contenue dans la mémoire RGO W2 = 1810 (n° 167) qui est utilisée une autre fois dans la partie interprétative. Il est donc inutile de créer une constante supplémentaire.
- Cette partie de programme pourrait comporter un ordre de moins. Il suffirait pour cela de former l'ordre 01 U DEBUT avant l'ordre 60 U W0002. Le dernier ordre de substitution serait alors SBF W.0001 W 0001 ou SBF W0001 8001, c'est-à-dire l'ordre contenu en SBFW1 = 1825 (n° 198) et déjà utilisé. Une mémoire serait alors libérée. Cette amélioration sera apportée ultérieurement au C. D. P.
- = Code 05 U V : aller an langage machine à U.

Le dernier ordre en langage machine doit avoir comme adresse instruction suivante : 1800, c'est-à-dire DEBUT. Le programme continuera alors à l'ordre en code placé en V.

Le seul ordre à former par la partie interprétative est 00 xxxx U c'est-à-dire: aller à U. En effet, le programme en langage machine commençant à U se

déroule jusqu'à ce qu'une adresse instruction suivante 1800 se présente. A ce moment, l'adresse V se trouve dans la mémoire MEMNN (placée par la partie interprétative commune à tous les ordres à 1 mémoire), et le programme continue comme souhaité.

Le seul ordre à écrire est donc :

Lorsqu'on est sûr de n'utiliser un programme qu'en logique extérieure, on peut exécuter des tests et des aiguillages à l'intérieur du programme en langage machine et prévoir des rentrées en codes à des adresses différentes.

Par exemple, si la rentrée pour une branche doit se faire en W, il suffit que cette branche place 00 0000 W en 1884 = MEMNN puis s'adresse à 1800 = DEBUT. Mais ceci n'est pas possible en autopoogrammation, à moins de prendre de très grosses précautions.

= Code - 06 U V : Perforation simple.

Cet ordre est équivalent à l'ordre en langage machine 71 U V.

- La mémoire PFDEB = 1733 (n° 40) est aussi utilisée pour l'analyse. Elle contient 71 P 0001 DEBUT c'est-à-dire 71 1977 1800.
- En DADSB = 1679 (n°197) la machine place U dans l'accumulateur droit en position d'adresse facteur puis fait un SBF pour former l'ordre 71 U DEBUT qu'elle exécute.
- = Code 07 U V : lecture.

Cet ordre est équivalent à l'ordre 70 U V en langage machine à condition que les cartes lues soient des cartes "lecture". En effet la partie interprétative forme l'ordre 70 U DEBUT et l'exécute. Si la carte est une carte

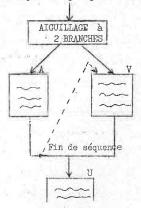
.../..

"lecture", elle est lue et le programme continue en DEBUT = 1800 où il commence à interpréter l'ordre V (car 00 0000 V est en MEMNN = 1884). Mais si la carte est "chargement", après la lecture le programme va à U qu'il essaye d'interpréter en langage machine. Cette propriété peut au besoin être utilisée à condition que le programme ne soit traité qu'en logique extérieure. Si ce n'est pas le cas on peut se tirer d'affaire en prenant de grosses précautions.

196 | T|0007 | E|DI L'CDEB | D|ADSB | 0...0T0007 | U V 00 | 70 0000 DEBUT

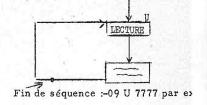
- La mémoire LCDEB = 1726 (n° 294) contient 70 0000 1800 c'est-à-dire 70 0000 DEBUT.
- En DADSB = 1679 (n°197), U est placé en position d'adresse facteur dans 8002; ensuite, l'ordre 70 U DEBUT est formé par un SBF puis exécuté.
- = Code 09 U V : Fin de sequence.
  - Utilité de cet ordre : nous la montrerons sur un exemple d'organigramme.

En autoperforation, il faut parcourir successivement les 2 branches de l'aîguillage pour pouvoir perforer les programmes en langage machine correspondants. S'il n'y avait pas d'ordre de fin de séquence dans l'organigramme ci-contre, la branche A serait autoperforée mais pas la branche V. En autoperforation, il faut donc aller à V à la fin de la branche A. Par contre, en logique extérieure et en exécution de programme autoperforé, c'est à U que l'on doit aller après la branche A. L'ordre de fin de



séquence - 09 U V envoie donc à U en logique extérieure et en exécution de programme autoperforé, mais à V en autoprogrammation. Cet ordre doit en général figurer à la fin d'une des branches des tests à 2 branches et à la fin de 2 de celles des tests à 3 branches. Il doit également être utilisé lorsqu'on arrive à un endroit qui doit marquer la fin de l'autoperforation et non calle du

déroulement du programme. Par exemple: La fin de séquence sert ici à stopper l'autoprogrammation. Lorsque le programme s'exécute(en logique extérieure ou en langage machine), le programme s'arrêtera quand il n'y aura plus de cartes à lire.



D'une façon générale, la fin de séquence est utilisée lorsque les trajets suivis par la logique extérieure et l'autoperforation sont différents (sauf pour le bouclage).

- Réalisation : En logique extérieure, cet ordre est un saut inconditionnel à U.

200 T,000	9 S'AG	1 18003	:.	ooToog9	U V 00		1,1	2 108	
201		10006	1 1		0O U				
202	TIPD	MEMNN	DEBU	rT()		0O U	en	MEMNN =	1884

Le programme envoie U à la place de V comme adresse instruction suivante en MEMNN = 1884 puis va en DEBUT = 1800.

#### VII - B - 3 - Remardue sur les ordres de service :

Pour l'exécution de tous ces ordres à 1 mémoire ou ordres de service, apparaissent nettement :

- La partie commune, d'abord à tous les ordres, ensuite aux ordres de service.

  Cette partie prépare tout ce qui est utile à l'ensemble des ordres de service

  (analyse et adresse V de l'instruction suivante) et effectue l'aiguillage à branches multiples qui différencie les ordres
- La formation des ordres à exécuter
- L'exécution de ces ordres
- Le retour à DEBUT.

Pour les ordres à 2 mémoires, les plus fréquents et les plus complexes, apparaissent deux faits nouveaux : l'indexage et l'existence de sous-programmes.

#### VII - C - CRDRES A DEUX MEMOIRES.

Chaque adresse représentée par une lettre (par exemple N) sera notée indiféremment par cette lettre (N) ou par la lettre écrite autant de fois que l'adresse a de chiffres (NNN).

#### VII - C - 1 - Partie commune à tous ces ordres : débute en DEUME = 1709 (n°48)

a) Début : Sa principale fonction est de séparer les diverses indications contenues dans l'ordre.

Harris TVI.	5	11-1-1	, 1	do o	PP'NNNIAAAA		
48 D EUME	TRD	P 0002	11		II MINIMANA	PP' NNNi AAAA	en P2 (pour analyse)
49	EDI	18003				00	1
50	SBI	P10005		100		0 0 A	en P5 = 1981
51	SAD	18001	1		PP'NNNioocc		
52 1	DAG	10001		0 OP	P'NNNL00000		
53	AIAG	CITEL		0OGOOOP	PANNIOO	00G0000	
54	T RG	W10002	11			0OGOOOP	en W2 = 1992
55	SIAG	C TEL	11	0 OP	P'NNNioo		
56	DAG	0001		0OPP	NNNi 00		
57	S, AG	8003	111	0 0		0OPP'	
58	TDI	P PPRI				0 OPP'	en PPPRI = 1750
59	DAG	10003	2/	OONNN	io o		
60	T RG	M EMNN			4.00	0 O N	en MEMNN = 1884
61	TRD	W 0003				i0 0	en W3 = 1993
62	R AD	P 0001	1	00	00 n 0000		21 = 2 = 2
63	A AD	G,J	8002		67 n+1 GL	67 0001 GL	
64 G J	67	0001	G L				
(8002)	RAV	n+l	G L	0 0	jbbbbkcccc	j B k C	
65 G L	TDI	P 0003				j B k C	en P3 (pour analyse)
66	E DI	8003	t	A LINE		0 0	
67 1	SBI	P 0008			1 100	o 00000 c	en P8 = 1984
68	SAD	8001	1		jBBBBk0000		

69	DAG	0001		00 j	BBBBk00		
70	SAG	8003	1	0 0		00 j	
71	T DI	W 0005	į			0 j	en W5 = 1995
72	DIAG	10004	I INNIN	0ОВ	k00		

- F0005 = 1981 et P0008 = 1984 servent ici de mémoires de travail.
- La mémoire CTE 1 = 1802 (n°159) contient la constante 00 0000 G 0000 c'est-à-dire 00 0000 1856. En lui ajoutant P, on forme l'ordre d'aiguillage multiple "aller à G 000 P" qui marquera la séparation des branches correspondant aux différents P. Cet ordre est mis en réserve dans la mémoire de travail W0002 = 1992 par l'ordre n°54.
- L'ordre n°60 met en place l'adresse de l'instruction suivant e dans la mémoire MEMNN = 1884.
- Quand tout 10 parti possible a été tiré de la noitié gauche de l'instruction, c'est la partie droite qui est considérée (celle contenue en n+1). Le mot 00 0000 N a pris la place de OO 0000 n en MEMNN = 1884, mais l'adresse n se trouve encore dans la mémoire P0001 = 1977. Les ordres n°62 et 63 forment l'ordre qui va chercher la valeur absolue de la partie droite de l'instruction.
- C'est l'ordre INNIN qui enverra 00 0000 B en P0006 = 1982 utilisée comme mémoire de travail. Quant à k 0.....0, il ne sera envoyé en mémoire que si l'indexage a lieu (paquet "index" chargo).

#### b) Index et fin de la partie commune :

Le contenu de INNIN = 1759 (n° 73) est différent selon que le paquet supplémentaire "index" est chargé ou non. Si ce paquet n'est pas chargé, INNIN = 1759 contient: TRG P0006 NINCH = 21 1982.

1803. Cet ordre envoie donc à NINCH = 1803 (n°74): voir organigramme.

Si ce paquet est chargé, INNIN = 1759 contient: TRG P0006 INCHA = 21

1982 1785. Cet ordre envoie à INCHA = 1785 (n° 226).

= Cas où le paquet "index" n'est pas chargé:

		1						1
73	INNIN	T <sub> </sub> RG	P 0006	N INCH	00 B	k0 0	00 B	en P6 = 1982
74	N INCH	RAG	P 0008	C ØMMU	oo c	00		
75	.C ØMMU	DAG	0004	- 40	00 C 0000	W 250		
76		A AG	O OANA		OO C ANALY			5_500 10 10
77		T RG	P 0007	W 0002			OO C ANALY	en P7 = 1983
	(wlooo2)	SIP	10000	G 000P	ľ			Aiguillage

La mémoire PC007 = 1983 est utilisée ici comme mémoire de travail.

= Cas où le paquet "index" est chargi:

(CTRI) désigne le contanu du registre d'index n°i.

73	I NNIN	T RG	P 0006	I NCHA	00 B k00	00 в	en P6 = 1982
226	I NCHA	T RD	W 0007	J. 15		k00	en W7 = 1997
227		RAG	P PPRI		0OPP' 0 O		
228		SIAG	0 080		0OPF480		
229		A NG	I'NF	W 0.002			P ≠ 8 et 9 ?
230	INF	R AD	W 0003	14.50	0 0 i0 0		Cas où P < 8
231		DAD	10005		00000i0000	De De Sei	de,
232	in gran	ALAD	RIDCKA	8002	65CTR1 KA		
	(8002)	RIAD	CITRA	KA	00(CTRi)		

						1361 3751	
233	KA	A AG	8001		00 (CTRi)	00 (CTR1)	
234		A AG	P 0005	1	O OA1		OOA
235	1.1	T RG	P,0005				00 A' en P5 = 1981
236	J.	DIAG	0003			00(CTRi)000	
237	1	TRD	P 0004		1 6 6 50		00(CTRi) en P4 = 1980
238	1	RAD	W 0005		00	0 0 j	4 27
239	1	DIAG	.0004	= x1 =	the coffee of	00 j 0000	
240	1	A AD	R DCKE	8002		65 CTRj KB	
4	(8002)	RIAD	CITRJ	KIB	00	00 (CTRj)	
241	KB	A AG	18001		0o (CTRj)		
242		A įAG	P. 0006	1	0 0 B'		о о в
243	100	TIRG	P 0006				0 0 B' en P6 = 1982
244	1	A AD	P 0004			0000(CTR1)(CTRj)	Parker of the parker of
245	50	D AG	0003			0(CTR1)(CTRj)000	
246		T RD	P 0004	15			O(CTRi)(CTRj)en P4 = 1980
247		R AD	W 0007		00	k00	000
248	- 19	D AD	0005	5 III /		00k0000	
249	1	A AD	R DCKC	8002	1	65 CTRk KC	
	(8002)	RIAD	CTRk	Kic	0 0	00(CTRk)	3 - 1231-200
250	KIC	A AG	18001		0 O(CTRk)	20 1	<i>9</i> :
251		A IAG	P 0008		0 0 C'		o o c
252		A AD	PJ 0004	2		O(CTRi)(CTRj)	
- 1			1			(CTRk)	
253		SAD	8002	1 49 390		00	O(CTRi)(CTRj) (CTRk)
254	1	TIDI	P 0004				
274			1 0004	11250			O(CTRi)(CTRj)en P4 = 1980 (CTRk)
255		TRG	P10008	с фими	0 o c	0	0 0 C' en P8 = 1984
				1			
256	R DCKA	R AD	C 0000	KA			
257	R DCKB	R AD	G 0000	KB			10 P 2 25 - 10 10 10 10 10 10 10 10 10 10 10 10 10
258	RIDCKC	R AD	c 0000	KC			

- Les instructions n° 227, 228, 229 permettent d'effectuer le test:

  "P ≠ 8 et 9 ?". En fait, c'est le test "P < 8 ?" qui est exécuté, ce qui
  revient au même. Si P est égal à 8 ou 9, le programme va directement

  à W 0002 = 1992 où se trouve l'ordre d'aiguillage 00 0000 G 000P =

  00 0000 G 0008 ou 00 0000 G 0009 (voir organigramme). Si P est différent
  de 8 et 9, le programme va en INF = 1642 (n° 230) pour exécuter l'indexage.
- La mémoire 0080 = 1822 (n° 161) contient la constante 00 0000 0080.
- Les ordres n° 230, 231, 232 servent à former l'ordre 65 CTRi KA =
  65 CTRi 1651 qui fait entrer dans l'accumulateur droit le contenu du registre
  d'index n° i. Les ordres n° 238, 239, 240 jouent le même rôle pour
  l'index j, et les ordres n° 247, 248, 249 pour l'index k.
- L'ordre n° 235 envoie A' = A + (CTRi) à la place de A, c'est-à-dire en

  P 0005 = 1981. De même les ordres n° 243 et n° 255 envoient respectivement B' et C' dans les mémoires P 0006 = 1982 et P 0008 = 1984.
- La partie de programme "indexage" sert également à former le mot 4

  de l'analyse, c'est-à-dire 0 xxx xxx xxx
  (CIR)(CIR)

Ce mot est formé petit à petit, au fur et à mesure qu'interviennent les 3 registres d'index. L'ordre n° 254 envoic ce mot en P 0004 = 1986 (4ème mot de la zone de perforation).

"index" est chargé rejoint le programme sans paquet "index".

Remarque: l'assemblage de cette partie de programme a été fait de telle façon qu'elle remplisse une séquence compacte (1639 à 1670). Cet assemblage a donc été fait à la suite de l'assemblage du reste du programme, séparé de celui-ci par le passage d'une carte PBD 1639 1670 et d'une carte PRA 1671 1999 (de sécurité).

En CØMMU = 1789 (n° 75), la machine forme 00 C ANALY ou 00 C' ANALY (ANALY = 1700) et l'envoie en P 0007 = 1983 utilisé comme mémoire de travail à ce moment là. Ensuite, le programme arrive en W 0002 = 1992 où se trouve l'ordre d'aiguillage 00 0000 G 000P = 00 0000 1856 + P, placé par l'ordre n° 54. Dans la suite du texte, nous désignerons par Λ l'adresse Λ indexée ou non (suivant que le paquet "index" est ou non chargé) c'est-àdire Λ ou Λ'. Cela ne risque pas de prêter à confusion car lorsque Λ' est créé, Λ n'a plus aucune importance. La même remarque est applicable à B et C.

#### c) Analyse

La partie de programme relative à l'analyse de la plupart des ordres à 2 mémoires commence par un test sur le signe du pupitre. Si ce signe est -, l'analyse doit être faite. Les mémoires P 0001, P 0002, P 0003, P 0004 sont déjà pourvues quand on arrive en ANALY = 1700 (n° 23). .../...

Il reste donc à envoyer (A) en P 0005 = 1981, (B) en P 0006 = 1982, (C) en P 0007 = 1983 et (C + 1) en P 0008 = 1984.

				,	44.74	lu .
23 A NAL	R AG 8000	1				3000
24	A NG A NA	Di EBUT		,	var chart	Analyse ?
25 A NA	R, AG P, 0005	(	00 A	00		30.8 DS 709
26	A AD P 0006	A DR.		00 В	- 87% A	
27	D AG , 0004		00 A 0000	00 B 0000		He year
	A, AG E, DOKM	8003	69 A KM		69 0000 KM	
(8003	E DI A	K M			(A)	
29 KI M	T, DI P, 0005	1			(A)	en P5 = 1981
30 1	A AD E DOKN	18002		69 B KN	69 0000 KN	L THE SALE
(8002	EDI B	K, N			(B)	
	TIDI P 0006		100	- 4	(B)	en P6 = 1982
32	RIAD P 0007	S, UANA	0 0	OO C ANALY		TEL HOUSE
33 S UANA	E DI E DOKP	7	A military		69 XXXX KP	
34	S BF E DOKP	8001	es es un cito	ri ge ar illi	69 C KP	I service to
(8001	EDI C	K P			(c)	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1
35 K P	T, DI P, 0007	120	- 46 -	100 Aug	(c)	en P7 = 1983
36	A AD 0 10	1		OOC+LANALY	v 100 100	Marie P. A
	E DI E DOKR				69 xxxx KR	
38	S BF E DOKR	8001			69 C +1 KR	
	E DI C+1				(C+1).	F. 4
39 K R :	1 DI P 0008	P' FDEB	9514		(C+1)	en P8 = 1984
40 P FDEB	P FØ P 0001	DI EBUT		1		

- La mémoire EDOKM = 1701 (nº 180) contient 69 0000 KM = 69 0000 1751
  - " EDOKN = 1737 (nº181) contient 69 0000 KN = 69 0000 1801
  - " EDOKP = 1690 (nº 182) contient 69 xxxx KP = 69 xxxx 1702
  - " EDOKR = 1696 (nº 185) contient 69 xxxx KR = 69 xxxx 1752
- La mémoire 010 = 1689 (nº 160) contient 00 0001 0000

VII - C - 2 - F = 0: sous-programmes à 2 adresses - Opérations virgule fixe Tests -

Dès le début de cette partie, les programmes relatifs aux différentes valeurs de F' se séparent. Four cela, la machine forme l'ordre d'aiguillage 00 0000 F 00FP' = 00 0000 F 000F' = 00 0000 1846 + F'. Ce test est aussitôt exécuté.

				111111			1 1	
149	G 0000	RAG	PPPRI		0 0 PP'	00	A 1 =	
150	1	A AG	C (TE2	18003	0 O F OOPP'		00 F 0000	
	(8003)	SIOP	10000	F OOOP				

- La mémoire CTE 2 = 1771 (n° 177) contient 00 0000 F 0000 = 00 0000 1846
- Dans la mémoire PPPAI = 1750, l'ordre n° 58 a envoyé 00 0000 00PP!
- Remarque: Les mémoires F 0000, F 0001,..., F 0008 contiennent les premiers ordres des branches relatives respectivement à FP' = 01, 02,..., 08. La mémoire F 0009 est libre, ce qui permettra de créer sans difficulté un ordre PP' = 09.

#### a) PP' = 00: sous-programmes à 2 adresses :

La partie de programme commençant à F 0000 = 1846 forme les ordres d'entrée dans les sous-programmes à 2 adresses. Ceux-ci servent en général à calculer des fonctions. Leur standardisation est assez délicate. Certains des sous-programmes existant demandent que l'argument soit en 8002, d'autres le veulent en 8003, d'autres enfin dans les deux accumulateurs.

Cela ne permet pas de faire une entrée standard. Pour standardiser l'entrée, que l'argument soit en simple ou double précision, il faut obligatoirement faire figurer dans l'accumulateur non pas l'argument lui-même mais son adresse. Cette adresse servira à former le ou les ordres destinés à entrer l'argument lans l'accumulateur. Cette entrée sera le plus souvent faite par un ordre de RAD. C'est donc sous la forme 65 B xxxx que le C.D.P. envoie l'adresse B dans l'accumulateur. Il met de plus en position "adresse instruction suivante" de cet ordre l'adresse A + 1 qui est une adresse du sous-programme. L'ordre 65 B A ÷ 1 est donc directement utilisable et exécutable. Le C.D.P. le place en 3003. L'ordre de sortie est placé par le C.D.P. en 3002 et non en 8001 pour pouvoir, si besoin est, le transformer plus facilement. En résumé, l'entrée pratiquée par le code est donc de la forme :

Adresse	cø	F	IS	(8003)	(8002)
	RAG	X	6	65 B A + 1	00
8	AAD	(-	A		20 C N;
c:	65	В	A+1	_ 1 (e)	
(5)	20	C	NI		

Les sous-programmes employés loivent utiliser cette disposition,

Remarque 1: 20 C N' est l'ordre de sortie ou tout au moins servira à former celui-ci. N' est l'adresse de la lère instruction en langage machine à effectuer après le sous-programme. Si l'on est en logique extérieure, la lère instruction est l'ordre ANALY = 1700 (n° 23) (voir l'organigramme). On a donc N' = 1700 = ANALY.

/

Si l'on a par contre autoperforé le programme ét que on l'exécute en langage machine, la lère instruction après le sous-programme est l'instruction N. On a donc N' = N.

Remarque ?: La forme de l'entrée pratiquée par le code qui est indiquée ci-dessus est schématisée. En fait, les mots  $\alpha$  et a sont directement formés dans les accumulateurs, du moins en logique extérieure. Dans ce cas, 20 C N' n'apparaît pas dans le distributeur.

		1	1 1	1.1	OOxxxx	0 0		1
151	F10000	A,AD	P 0006	- E		OQB		
152	0.1	DIAG	0004	1	-	00 B 0000		
153	-1	AAD	P 0005	1		00 B A		
154	1	EDI	KE	1			15 P7 xxxx	
155	T	SBI	KE	1.			15 P7 A	en KE
156	1	RAG	8002	- (	00 B A	0 0	4000	
157	114 41	A AG	R DOL		65 B A+1		6500000001	100
158	1	A!AD	TI DOO	K! E		20 0	20 0	
	(K:E)	A AD	P! 0007	IA		20 C ANALY	OO C ANALY	

- L'adresse A en position l'adresse instruction dans l'accumulateur sert d'une part à former l'ordre 65 B A+1 et d'autre part à former l'ordre d'entrée dans le sous-programme : AAD PCCC7 A.
- La mémoire KE = 1821 (n° 178) contient la constante 15 PCC07 xxxx = 15 1983 xxxx.
- La mémoire RD01 = 1843 (n°179) contient 65 0000 0001.
- La mémoire TDCC = 1808 (z°171) contient 20 0000 0000.
- La mémoire F0007 = 1983 contient 00 C ANALY = 00 C 1700 envoyé par l'ordre n°77.

#### b) PPt = 01 : addition en virgule fixe .

- Les parties de programme relatives aux opérations en virgule fixe ont été construites de façon à avoir entre elles le plus possible de parties communes.
- Les ordres que construit la partie relative à l'addition sont :

	RAG	A	W0002
W0002	AAG	В	W0003
W0003	TRG	C	ANALY

- La formation des ordres RAG et TRG, pouvant avoir une partie commune avec d'autres opérations, a été placée après la formation de l'ordre AAG.

79	F 0001	RIAD	O DIXO	AIDSØ1	0 0	00 0010 0000
81		AAD	Pl 0006			00 0010 B 0 0 B
82	1	D,AG	10004			10 B 0000
83	1	A AD	010W3			10 B W0003 00 W0003
84	10	T RD	W  0002			10 B W 0003 en W2
100		RIAG	P <sub>1</sub> 0005		0 O A	0 0
85	1				1	
86	1	DAG	0004		000 A 0000	
87		A AD	T G00	K G		210 0
88	K <sub>1</sub> G	A AG	R GOW2	1	60 A W0002	60 0000 W0002
89		A AD	P 0007	i i i i i		21 C ANALY OO C ANALY
90	1	TRD	wi 0003	18003		21 C ANALY en W3
	(8003)	RAG	A	W 0002	(A)	0 0
	(WI 0002)	A AG	'В	W 0003	(A) + (B)	
	(WI 0003)			ANALY		(A) + (B) en C

.../...

- La mémoire ODIXO = 1754 (n°164) contient la constante 00 0010 000
- Pour former 10 B W0003 on aurait pu amener d'abord 0.....0 B dans l'accumulateur droit, faire un décalage de 4 positions à gauch et ajouter 10 0000 W0003. Il y aurait eu 3 ordres au lieu de 4. Mais la partie relative à la soustraction n'aurait pu rejoindre celle de l'addition qu'après ces 3 ordres, et en en comport ant aussi 3, ce qui fait 6 ordres en tout. Or, dans le cas présent (voir § c) la partie "soustraction" rejoint la partie "addition" en ADSØ1 = 1760 (n°81) après un sehl ordre. L'ensemble a donc 4 + 1 = 5 ordres.

  Les 2 parties étant séparées au début, on a gagné de la place en écrivant en premier les ordres non communs.
- La mémoire 00 W3 = 1758 (n°165) contient 00 0000 W0003 = 00 0000
- La mémoire T G 00= 1748 (n°166) contient 21 0..... 0
- La mémoire RGCW2 = 1810 (n°167) contient 60 0000 WCC02 = 60 0000 1992

#### c) $PP^1 = 02$ : soustraction en virgule fixe.

- Les ordres que construit la partie relative à la soustraction sont :

	RAG	A	W0002
W0002	SAG	В	W0003
W0003	TRG	C	ANALY

La seule différence avæc les ordres d'addition est le SAG.

La seule partie du programme particulière à la soustraction est :

80 | F10002 | RIAD | 01 ØNZO | A1 DSØ1 | 0 ...... 0 | 00 0011 0000 | 00 0011 0000 |

La mémoire 00NZC = 1804 (n°184) contient 00 0011 0000.

## d) PF' = 03 et FP' = 05 : multiplication en virgule fixe sans ou avec décalage

- Les ordres que construit la partie relative à la multiplication sans décalage pourraient être :

1	RAG	A	W0002
W0002	MUL	В	W0003
W0003	TRD	C	ANALY

Mais le produit (A) x (B) risque d'avoir 20 chiffres, c'est-à-dire que le résultat peut dépasser la capacité de l'accumulateur droit. Dans ce cas, la simple précision n'est plus respectée et il faut que la machine le signale pour que l'on ne risque pas d'envoyer dans C une valeur qui ne serait que la partie droite du résultat. On fait donc pour cela un test sur la nullité de l'accumulateur gauche après la multiplication. Les ordres à former sont donc de la forme:

1	RAG	A	W0002		
W0002	MUL	В	KF		
KF	GNN	ARETM	W0003	= 44 1815 1993	
W0003	TRD	C	ANALY		
ARETM	ART	1333	W0003	= 01 1333 1993	

Ce n'est pas encore tout à fait ces ordres là que l'on forme.

En effet, on a ajouté un ordre pour que ce programme se rapproche plus de celui relatif à la multiplication avec décalage. Pour celle-ci, les ordres que forme le C.D.P. sont : (remarquess que le décalage a lieu après la multiplication et avant le test)

,			-	**
	RAG	A	W0002	
W0002	MUL	В	M000J	v_1 = 1 = 1.5
W0001	3 x	000m	KF	Ordre de décalage
KF	GNN	ARETM	W0003	Ordre constant
W0003	TRD	C	ANALY	
ARETM	ART	1333	W0003	Ordre constant

On a donc adopté comme ordres à former pour la multiplication sans décalage :

***************************************	-		PARTY AND THE PA	
	RAG	A	W0002	
W0002	MUL	В	M0001	
W0001	00	0000	KF	
KF	GNN	ARETM	W0003	Ordre constant
W0003	TRD	C	ANALY	
ARETM	ART	1333	W0003	Ordre constant

Programme relatif à la multiplication sans décalage :

91	F 0003	TRD	Wr0001		00 xxxx	0 0	0 0 en W1 = 1991
92	1	R AG	P10006		оов	0 0	
93		D AG	,i 0004	1	00 B 0000		
94	1	A AG	M, TOMT	1	19 B W1		
95	,	TIRG	W10002	1 3			19 B W 1 en W2 = 1992
96		RAD	0! 2KF	K;I	0 0	00 0002 KF	
97	K,I	E DI	W10001				0 0
98	1	SBI	M 0001				0 0 KF en Wl = 1991
99	1 6	RAG	P! 0005	1	0 0 A	00	
300	1	DAG	10004		0000 A 000	2	Har a la supplication
101	1	A AD	T, DOO	KIG	F	20 0	N 87 - 5 - 5 - 6 - 6 - 6 - 6 - 6 - 6 - 6 - 6
88	(K,G	A AG	RIGOW2	1	60 A W0002		
89	( )	AAD	P! 0007			20 C ANALY	
30	(1)	TRD	W10003	8003	W 62 - CS		20 C ANALY   en W3 = 1993

- La mémoire KF = 1762 (1°102) contient le test GNN ARETM W003 = 44 1315 1993.
- La mémoire ARETM = 1815 (n°172) contient l'ordre d'arrêt que provoque l'apparition d'un produit de plus de 10 chiffres. Cet ordre est : 01 1333 W0003 = 01 1333 1993. Il est caractéristique de l'apparition d'un produit de plus de 10 chiffres.
- L'ordre F0003 = 1849 (n°91) prépare l'ordre KI = 1712 (n°97) en envoyant 0....0 en W0001 = 1991. Nous verrons plus loin que la partie de programme commençant en F0005 = 1851 (n°103) prépare aussi l'ordre KI en envoyant l'ordre de décalage en W0001.
- La mémoire MLCW1 = 1811 (n°168) contient 19 0000 WCCC1 = 19 0000 1991.
- La mámoire 02 KF = 1806 (n°169) contient 00 0000 KF = 00 0002 1752.

  Le 2 en position facteur ne sert à rien. Le programme sous sa forme initiale utilisait cette constante; comme le 2 n'est pas nuisible, il a été conservé.
- La mémoire TD00 = 1808 (n°171) contient 20 ..... 0.
- La partie commençant en KG = 17.6 (n°82) fait aussi partie du programme d'addition, au cours duquel elle a été écrite.
- Après l'ordre n°90 s'exécutent l'ordre 60 A W0002 et la suite des ordres de la multiplication (2ème programme encadré).

.../...

Programme relatif à la multiplication avec décalage :

La seule partie particulière se compose des 3 ordres suivants :

103	F 0005 RIA	D   P10001	1 '	0 0	00 n	0000	
104	A,A	D !	8002		65 n+2	F0003	65 0002 F0003
105		5 ,0002				135	
	(8002) RIA	0   ln +2	F10003	0 0	3x 000	m 1939	

Ces ordres servent à entrer dans l'accumulateur droit l'ordre de décalage placé en n + Z. L'ordre FOCC3 = 1849 (n°91) envoie ensuite cet ordre en W0001 = 1991. La seule différence, dans la suite, avec la multiplication sans décalage est que c'est l'ordre 3x 800m KF qui est formé en W0001 = 1991, et non 06 0000 KF. C'est donc le ler programme encadré quisera exécuté après l'ordre n° 90.

#### e) PF1 = 04 et 06 : division en virgule fixe sans ou avec décalage :

Les ordres que construit la partie relative à la division sans décalage pourraient être :

Pour permettre au programme relatif à la division de rejoindre plus tôt celui de la multiplication, et en particulier avant l'ordre KG, ce n'est pas RAD A W0002 mais RAG A W0002 qui est formé comme ler ordre. Cela oblige à ajouter un ordre qui paraît artificiel mais permet d'économiser de la place.

	RAG	A	W0002	
M0005	RAD	8003	W0001	
W0001	DRR	В	W0003	
W0003	TRD	C	ANALY	

Mais là n'est pas encore ce qui a été adopté. En effet, pour permettre une plus grande similitude entre les divisions avec et sans décalage, on a encore ajouté unordre.

Pour la division avec décalage, les ordres formés sont :

T		mental and second		*	
-		RAG	A	W0002	
1	W0002	RAD	8003	W0001	
-	W0001	3×	00Cm	KH .	
	KH	DRR	В	W0003	
	W0003	TRD	C	ANALY	

Décalage (placé avant la division)

Par analogie, les ordres formés pour la division sans décalage sont :

	RAG	A	W0002
W0002	RAD	8003	W0001
W0001	00	0000	KH
KH	DRR	В	W0003
W0003	TRD	C	AMALY

..../

## Programme relatif | la division sans décalage :

					10 0 xxxx	00		
106	F 0004	T! RD	M 0001	1270			0 0	en Wl = 1991
107		R AD	P 0006		0 0	0 ОВ	and a line	
108	1	D, AG;	0004	Fa.		00 B 0000	, M = 1	
109		E, DI	KH	14.00	Auto .		64 xxxxW0003	
110		SBF	KIH	,			64 B W0003	en KH = 1716
111		EDI	RIDAWL	1	_ 24		65 8003W0001	
112		T, DI	W   0002	1			65 8003W0001	en $W2 = 1992$
113		R, AD	0   2KH	KI	0 0-	-00 0002 KH	Sugar Normal	
97	(KII)	E DI	W 0001	-1		W 1-897	00	
98	(i)	SBI	W,0001	J. 1			о окн	en W1 = 1991
99	( , )	R AG	P 0005		0 0 A	00	B	
100	( )	D AG	0004	1	00 A 0000			- 1424
101	( )	AAD	T DOO	K G		20 0		-1 7 - 1
88	(KIG)	AAG	R GOW2	i _	60 A W0002	. 4.0424		
89	(1)	A AD	P10007	1		20 C ANALY		
90	(   )	T; RD	W 0003	18003			20 C ANALY	en $W3 = 1993$

- L'ordre F0004 = 1050 (n°106) prépare l'ordre KI = 1712 (n°97) en envoyant C...O en W0001 = 1991. Nous verrons plus loin que la partie de programme commençant en F0006 envoie, elle, l'ordre de d00alage en W0001.
- La mémoire KH = 1716 (n°173) contient 60 xxxx W0003 = 64 xxxx 1993.
- La mimoire RDAW1 = 1672 (n°174) contient 65 8003 W0001 = 65 8003 1991.
- En KI = 1712 (n°97) le programme relatif à la division rejoint celui relatif à la multiplication. Le nombre d'ordres particuliers

à la division est donc assez restreint; ceci est dû surtout au fait que l'on ait pu se ramener à avoir comme ler ordre : 60 A WCCC2

- Après l'ordre n°90 s'exécutent les ordres du 7ème programme encadre.

### Frogramme relatif à la division avec décalage :

La seule partie particulière à la division avec d'scalage se compose de :

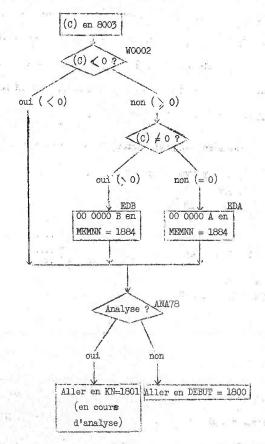
114	F 0006 R AD	P 0001		0 0	00 1	n 0000	1
115	AAD	1	8002		65 n-	+2 F0004	65 0002 F0004
116	165	0002	F, 0004		11-13		
	(8002) RIAD	1 n+2	F 0004	0 0	3x 00	00m 1991	1 - 1 - 7

Ces ordres servent à entrer dans l'accumulateur droit l'ordre de décalage placé en n + 2. L'ordre F6004 = 1650 (n°106) l'envoie ensuite dans la mémoire W6001 = 1991 pour préparer l'ordre KI = 1712 (n°97). La seule différence avec la division sans décalage est que c'est l'ordre 3x 600m KH et non 60 6000 KH qui est formé en W6001. C'est donc les ordres du ler programme encalré qui sont exécutés après l'ordre n°90.

## f) PF' = 07: Test de signe 2 3 branches.

Le programme que forme la partie interprétative pour exécutor le test SI peut être représenté par le schémic suivant :

.../...



Nous verrons au paragraphe X - E - 2 - d - a qu'à cause de l'exécution en langage machine, seule l'adresse C est indexable.

La branche correspondant à (C) < 0 doit envoyer à l'instruction d'adresse N en code . Pour cela, il faut que 00 0000 N soit placé en MEMNN = 1884, ce qui est déjà réalisé par le début de la partie interprétative. Les branches correspondant à (C) nul et à (C) positif doivent envoyer respectivement aux instructions en code contenues en A et B. Elles placent donc respectivement 00 0000 A et 00 0000 B en MEMNN = 1884.

Les ordres que forme la partie interprét ative sont les suivants :

	R AG	C	W 0002	(c)	0 0			
W 0002	A NG	ANA78			j = "-		5.7	(c) < 0 ?
	G NN	EDB	EDA					(c) ≠ o ?
EDB	E DI	P0006	EDBA		č., a	0	OB	Cas où (C) > 0
EDA	E DI	P0005	EDBA		1-1-1 P	0	OA	Cas où (C) = 0
EDBA	T DI	MEMNN	ANA78		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0	OB OA	en MEMNN = 1884
ANA78	R AG	8000						
	A NG	KN	DEBUT					

Pourquoi envoyer en ANA 78 et non pas en ANALY? Nous verrons que le programme relatif au code PP' = 08 (test suivant la position B de la mémoire C) a la même fin que le programme relatif à PP' = 07. Or, pour PP' = 08 il est dangereux d'amerer dans le distributeur le contenu de la mémoire B. B est en effet compris entre 0 et 9; il se peut que, par suite de la lecture d'une carte incomplète, certaines des mémoires d'adresse comprise entre 0 et 9

soient vides, ce qui provoquerzit une "erreur distributeur". L'analyse des ordres 07 et 08 ne comporte donc pas l'envoi du contenu de la mémoire B dans la mémoire de perforation P 0006. Elle ne commence donc qu'à l'envoi de (C) en P 0007 = 1983. En fait, c'est à l'ordre précédent qu'elle commence, c'est à dire à l'ordre KN = 1801 (n° 31). Il ne faut par conséquent pas s'adresser à ANALY = 1700 (n° 23) qui envoie à ANA, mais à ANA 78 = 1717 (n° 127) qui envoie à KN.

#### Programme interprétatif relatif au test SI

							i	
117	F. 0007	RIAD	1	F 7F8		46 ANA78 XXXX		
118		46	A NA78	1	1,51,51	- Territor 1	M cent	
119		G+NN	E DB	E DA				
120	E DB	EIDI	P10006	E DBA	ma 22 50.	2 2 22 22	O OB	
121	E DA	E,DI	P10005	E DBA			O OA	
122	E DBA	TDE	M;EMNN	1	- 4 (B)	W 1/1389 L 1/2	0 A	en MEMNN = 1884
	1			1	1	South Agent	О В	
127	A NA78	R AG		,				
128		A;NG	K'N	DEBUT	A 137 9	Projet we		
		-1		'		1.C 177 FO 1	4C 4N450	110000 3000
123	F: 7F8		W 0002	-	1		46 ANA/8 XXXX	en W0002 = 1992
124	da es	R:AD	P 0008	1 1	0 0	000		
125	į	D'AG	0004			.00, C 000D		5
126	1114	A!AD	R GOW2	8002	F # 447	60 C W0002	rus.	1
	(8002)	RIAG	1c	M; 0005	(c)	0 0	Wilking to 1 days	
	(WI 0002	AING	A NA78	1			1	
	(	G:NN	EIDB	E DA	1	į.	A 25 . D.	X v
						William Control		and the state of

10000

La succession de ces ordres, due à l'écriture en PASO, n'est pas celle de l'exécution. En effet, après l'ordre F 0007, c'est l'ordre F 7 F 8 qui s'exécute. L'exécution effective de l'ordre SI, indiquée précédemment, débute après l'ordre n° 126.

g)PP' = 08 Test sur l'existence d'un 8 ou d'un 9 en position n° B de la mémoire C.

Comme pour le test SI, seule l'adresse C est indexable à cause de l'exécution en langage machine. Les ordres que forme la partie interprétative pour l'exécution du test SD sont:

Le test 9B s'effectue sur le distributeur. Le ler ordre pourrait donc être EDI C W0002. Mais, pour permettre une plus grande partie commune avec l'ordre SI, c'est l'ordre RAG C W 6002 qui a été formé.

Si la réponse au test OUI, c'est-à-dire s'il y a un 3, le programme va en ANA78 = 1717 (n° 127). Le mot 00 0000 N se trouve en effet en MEMNN = 1884, c'est-à-dire que l'instruction suivante en code est bien N. Si la réponse est NON, c'est-à-dire s'il y a un 9, le programme va en EDA et 00 0000 A est envoyé en MEMNN = 1884. S'il n'y a ni 3 ni 9 en position B, aucune réponse au test n'est donnée et la machine s'arrête avec au registre programme :

9B ANA70 EDA = 9B 1717 1722.

Remarque: B = 0 désigne la position n° 10.

.../...

#### Programme interprétatif relatif au test SD :

129	F 0008	RIAG	P! 0006		О ОВ	0 0		
130	1	D AD	10005		0 0	ово о		
131	1	A AD	1	F 7F8		9B ANA78 EDA	90 ANA78 EDA	
132	1	90	A NA78	E, DA		la m	4 7-4	
	Ę		J			2012		
123	(F 7F8)	TRD	W: 0002			75-47	9B ANA 78 EDA	en W0002 = 1992
124	( ' )	R AD	P 0008		0 0	o oc	28.	
125	( )	DAG	0004			00 C 0000		244577.8
126	(; )	A AD	R GOW2	8002		60 C W0002	5.85	211
	(8002)	R AG	1 C	W 0002	(c)	00	(c)	
	(W; 0002)	SIDB	A! NA78	EDA	1			

Seuls les 4 premiers mots (n° 129 à 132) sont particuliers au test SD.

Les suivants sont communs aux tests SD et SI.

## VII - C - 3 - P = 1, 2, 3, 4, 5: sous-programmes 2 3 adresses.

Standardisation: Comme pour la plupart des sous-programmes à 3 adresses déjà existants (par exemple les opérations en virgule flottante), l'entrée dans le sous-programme se fait dans les conditions suivantes:

L'ordre de sortie que forme le CDP est 20 C N'. En logique extérieure, N' = ANALY = 1700. En autoprogrammation, N' = N, adresse de l'instruction suivante en code.

Comme il a été dit au paragraphe VI - C - 2 - a - 0 , les branches relatives à P = 1,2,3,4,5,6,7 se rejoignent tout de suite. En fait, nous ne nous occuperons que des P inférieurs à 6 car P = 6 et P = 7 ont été supprimés. Les mémoires G 0001 = 1057, G 0002 = 1050,...... G 0005 = 1061 contienment le même mot; c'est le 1er ordre de la partie interprétative particulière aux sous-programmes à 3 adresses. Les mémoires G 0006 = 1062 et G 0007 = 1063 sont libres.

#### Programme interprétatif relatif aux sous-programmes à 3 adresses :

33 34			P. PPRI P. PPRI				4.4	- V
10	, , ,		11000	1				
37	G. 0005	R AG	P PPRI	G COMU	0 OPP'	0 0	1	
40	G CØMU	D! AG	10004		0 OPP'0000		A PROPERTY OF THE PARTY OF THE	
41		A.AG	A DDEL	8003	15 DEL+PP' KJ		15 DEL KJ	
V	(8003)	A AD	DEL+PP'	K,J		0 0 Opp.		
42	K, J	E DI	E, DP7				111 8003 xxxx	
42 43 44 45		SBI	E DP7				11 8003 (1 <sub>PP</sub> ,	en EDP7 = 1770
44		RIAD	P10005		0 0	O OA	**	
45	11	DAG	10004			00 A 0000	1	
46		A'AD.	P,0006			00 A B		
47		A AG	P'0007	. 1	00 C ANALY			
48	,	A,AG	T D00	E DP7	20 C ANALY			
	(E:DP7)	SIAG	18003	1 Dpp	10 0	00 A B	20 C ANALY	

Les ordres n° 133 à 143 servent à former le dernier ordre, EDP 7, de ce programme interprétatif. D.7 doit effectuer l'entrée dans le sous-programme, c'est-à-dire envoyer en  $\triangle_{pp'}$ . La valeur de  $\triangle_{pp'}$  est indiquée dans la mémoire 1875 + FP' que nous appellerons aussi DEL + PF'. Les limites de cette table sont : DEL + 10 et DEL + 59 (en fait la table a été prolongée jusqu'à DEL + 61). Cette région sera

appelée en PAS $\emptyset$ : D CCCO à D CC51. Cette dénomination est donnée pour que l'on puisse faire une réservation régionale :

PRG D 1836 D 1939 (région D un peu prolongée)
PEN D 2000 1835

- Les ordres n° 140 et 141 forment l'ordre AAD DEL + PP' K J qui amène  $\Delta_{\rm pp'}$  dans l'accumulateur droit. L'ordre n° 143 envoie l'ordre SAG 8003  $\Delta_{\rm pp'}$  en EDP7 = 1770.
- La mémoire ADDEL = 1730 (n° 175) contient la constante 15 DEL KJ = 15 1875 1817.
- La mémoire EDP7 = 1770 (n° 176) contient la constante 11 8003 xxxx.

#### VII - C - 4 - P = 9 : bouclage.

Le programme de bouclage a été écrit sous forme d'un sous-programme. Ce sous-programme n'est pas standard et est d'ailleurs différent en logique extérieure et en exécution en langage machine. Il a été assemblé dans une séquence de mémoires commençant à 0000, puis mis sous forme translatable. En logique extérieure, son facteur de translation,  $\Delta_{\text{bou}}$ , a été fixé à 1596 et placé sous la forme 00 0000  $\Lambda_{\text{bou}}$  dans la mémoire DEL  $\div$  61 = D 0051 où se trouve donc l'ordre "aller à  $\Lambda_{\text{bou}}$ ". Le sous-programme de bouclage va chercher les différentes indications dont il a besoin dans les mémoires où les a mises la partie interprétative.

La seule instruction particulière au bouclage avant l'entrée dans le sous-programme est :

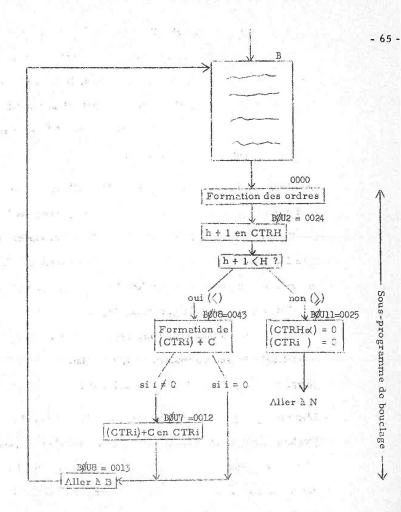
$$8 \left| \begin{array}{c} G_1^10009 \\ (D_10051) \right| S^1 / P \\ \end{array} \right| \left| \begin{array}{c} 0000 \\ 0000 \end{array} \right| \left| \begin{array}{c} D_10051 \\ 000 \end{array} \right| = 00 \ 0000 \ 1936 \\ = 00 \ 0000 \ 1596 \\ \end{array}$$

Pour que la lère instruction du sous-programme de bouclage soit bien en  $\triangle_{\rm bou}$  + 0, son adresse a été fixée à 0000 dans le programme en PAS $\phi$ .

-Voici le schéma suivant lequel s'exécute le bouclage 9 i N ≈ H O B O C.

Nous appellerons CTRH : le compteur de boucles n° , et h son
contenu. L'adresse CTRi désignera le registre d'index n° i et

(CTRi) sont contenu. Le sous-programme de bouclage débute par
la formation d'un certain nombre d'ordres destinés à l'exécution
proprement dite du bouclage.



.../...

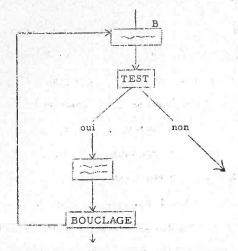
1

Il peut arriver que, par suite d'une erreur, le compteur de boucles n° of ne soit pas à zéro au ler passage par l'ordre de bouclage, et même contienne un h supérieur ou égal à H. Dans ce cas, comme on peut le voir sur l'organigramme, la boucle n'est pas répétée. Le compteur de boucles n° et le registre d'index n° i sont remis à zéro et le programme continue en N.

- Sauf dans des cas exceptionnels, le registre d'index n° i et le compteur de boucles n° x sont à zéro au moment où l'on passe pour la lère fois par l'ordre de bouclage. Chaque fois que l'on passe par cet ordre, la machine ajoute l'au compteur de boucle et C au registre d'index. Notons que si i = 0, aucun registre d'index n'est modifié, quel que soit C.

  En effet, il serait trop dangereux de risquer de modifier le registre d'index n° 0 qui doit obligatoirement contenir zéro. Lorsque la boucle à été décrite le nombre voulu de fois, le registre d'index n° i et le compteur de boucles n° x sont remis à zéro.
- L'existence de plusieurs compteurs de boucles permet d'effectuer
  plusieurs boucles intérieures les unes aux autres (On ne peut pas
  décrire des boucles intérieures les unes aux autres en utilisant le
  même compteur) Lorsque le programmeur écrit des boucles intérieures
  les unes aux autres, il ne doit absolument pas leur attribuer le même &.

- Remarque 1 : quand une boucle ne possède pas de boucle intérieure, les ordres formés au cours du ler passage par l'ordre de bouclage restent valables pour les autres passages et pourraient être conservés; cela permettrait d'éviter pour les autres passages la phase de formation d'ordres et ainsi de gagner du temps. Ce perfectionnement a été apporté en exécution en langage machine comme on le verra au paragraphe X - E - 4; on réserve la valeur 0 = 0 pour les boucles n'ayant pas de boucle intérieure. Pour ces boucles, en exécution "langage machine", la partic de formation des ordres est supprimée après le ler passage. Cela permet de gagner du temps (le temps est le facteur essentiel en exécution en langage machine) mais donne plus d'encombrement au programme de bouclage. C'est pour cette raison qu'en logique extérieure, quel que soit le compteur de boucles utilisé, la phase de formation des ordres n'est jamais supprimée. En logique extérieure en effet, c'est l'encombrement qui est le facteur le plus important.
- Remarque 2: il peut arriver que l'on sorte de la boucle sans avoir décrit le nombre H de tours. Ceci peut se présenter en particulier lorsqu'il y a un test à l'intérieur de la boucle, par exemple :



Dans ce cas, on ne passe pas par les ordres de remise à zéro du compeur de boucles & et du registre d'index i. 3'ils doivent être utilisés par la suite, il ne faut pas oublier de les remettre à zéro.

- Voici le programme qui exécute effectivement le bouclage (voir l'organigramme); a plupart de ses ordres sont formés par la phase de formation des ordres.

				0 0 h			
B, ØU13	AJAG	0101	B ØU4	0 0 h+l			
B, ØU4	T,RG	C! TRH ox	1			0 0 h+l	en CTRH 🐱
	TDI	P10004	BIØU3		9014 730 177	0 0 h+1	en P0004 = 1980
B ØU3	SIAG	P10005	1	00 h+1-H		О ОН	
			B ØU11		表(表)(1000)		h+1 < H?

.../..

	4	100								
	BI ØUS	R!AG	CITRI	WI 0006	0 0	(CTRi)				
(	MI 0006	A AG	P10008	B <sub>j</sub> ØU7	0 0	(CTRi)+C		237 437	si	$i \neq 0$
(	W, 0006	A, AG	P 0008	B ØU8				Lecond 150	si	i = 0
	B ØU7	T RG	C, TRi					0 0 (CTRi)+C	en CTR	li
		T DI	P 0005	B ØU8	le Te :			0 0 (CTRi)+C		
	B ØU8	E DI	P 0006					0 B	19	981
	1	T DI	M EMNN	A NABØ			^	0 0 B	en MEM	
	B ØU11	TIRD	CITRI	B Øug				0 0	en CTR	384 Ri
	D ØU9	T DI	C, TRH	A NABØ				0 0	en CTR	出义
	1									
	A NABØ	R AG	8000	1					3.0	
		A NG	PFDEB	D. EBUT	9			To a read	Analys	e?

La mémoire W 0006 = 1996 a un contenu différent selon que i est ou non égal à zéro. Si i ≠ 0, W 0006 adresse à l'ordre BØU 7 = 0012 (n° 294) qui renvoie l'index modifié (CTRi) + C dans le registre d'index n° i. Si i = 0, W 0006 adresse directement à BØU 8 = 0013 (n° 296), n'envoyant rien en CTRi = CTRO = 1967. C'est ainsi que le programme évite de modifier le registre d'index n° 0.

Placés par le début de la partie interprétative, H se trouve en P 0005 = 1901, C en P 0000 = 1904 et B en P 0006 = 1902.

Analyse: Les mots 1, 2, 3 de l'analyse sont déjà pourvus au moment où débute la partie propre au bouclage. Pendant cette partie, h + 1 (c'est-à-dire le contenu du compteur de boucles n° après sa modification) est envoyé en P 0004 = 1980; le contenu du registre .../...

d'index n° i après sa modification est envoyé en P 0005 = 1931.

Si i = 0, rien n'est envoyé en P 0005. Au dernier passage par l'ordre de bouclage, c'est 0.....0H qui est envoyé en P 0004 et P 0005.

L'instruction ANABØ = 0011 (n° 300) amène le contenu du pupitre dans l'accumulateur. S'il est positif, le programme va directement en DEBUT = 1800; s'il est négatif, le programme rejoint l'analyse normale en PFDEB = 1738 (n° 40) où se perfore la carte d'analyse.

#### Liste du sous-programme de bouclage :

259	11	PIRA	11500	11999				
260		PiBD	,0001	: 0042	1	_		
261		PiEN	B ØU7	10012				
262		P,EN	B ØU8	10013	/ 4 1		6 II II III	
1.0			· ·			The Part of the	- C	
263	10000	RIAD	W,0003		0 0	· · · · · · · · · · · · · · · · · · ·	9.17	
264	ŧ	DAD	0005	3	E. 174.	00000 ⋈ 0000		in "
265		A AD	O HOO	1		000TRH 4 0000		
266		E <sup>‡</sup> DI	Bløu2			- 45 & P 3v	60 xxxx BØUl3	
267		S,BF	BIØU2			19951	60CTRH & BØUL3	en BØU2 = 0024
268	,	E DI	B ØU4	,	and the same of th	Tarabili in No.	21 xxxx	
269	1	SBF	B ØU4	1		Rigin su	21CTRH &	en BØU4 = 0031
270	1	EĮDI	BIØU9	1			24 XXXX ANABØ	
271		SBF	BiØU9				24CTRHX ANABØ	en BØU9 = 0038
272		R AD	P PPRI	1-	00	009i		
273		SIAD	0,090	_		00i		
274		AINN	B ØUl	J. 27		1 1000 1800	1.3	i≠0?
275	1	A AD	O CENO	B ØU1		00 0100 0000	1 1	
276	B ØU1	A AD	01000	41	n kontis	00 0100 CTRi	CRITICAL INC.	
277		DIAG	0004	1 1	0 0 1	00 CTRi 0000		

.../...

278 279 280 281 282 283	1	E'DI S'BF E,DI S'BF E'DI S,BF	B ØU5 B ØU5 B ØU7 B ØU7 B ØU11 B ØU11	1	0		0 0 1		CTRi	0000	60 21 21 21 20	CTRi xxxx CTRi xxxx	W0006 W0006 BØU9 BØU9	en	BØU7	= 0043 = 0012 = 0029
283 284 285		A AG T, RG	B ØU6 W 0006	B1ØU2	10	P00	008 <b>3Ø</b> U	8)			10	P000	BØU7 BØU8	en	W0006	= 199
286 287 288 289	B ØU2 B ØU13 B ØU4 B ØU3	RIAG A AG TIRG T.DI S'AG	0000 0 01 0000 P 0004 P-0005	B ØU13 B ØU4 B ØU3			. ·	š								
290 291 292 293 294 295	B ØU5 B ØU6 B ØU7	A NG R AG A AG T RG T DI	B <sub>1</sub> ØU5 0000 P <sub>1</sub> 0008 0000 P <sup>1</sup> 0005	B,ØU11 W 0006 B,ØU7 B/ØU8						57			(9 (5			
296 29 <b>7</b>	B ØU8	E DI T'DI	P 0006 M EMNN	AINABØ		(4										
298 299	B ØU11 B ØU9	TRD TDI	0000	B ØU9 A, NABØ		S.	9									
300 301	A, NABØ	R AG A NG	P FDEB	DEBUT												
302 303 304 305	O HOO O O O O O CENO O COO	00	H 0000 0000 0100 0000	0000 0000 0000 0000		14			les i			5				

- La mémoire W 2003 = 1993 contient 🗷 qui a été placé par le début de la partie interprétative. De même, la mémoire PPPRI = 1750 contient PP<sup>1</sup> = 9i.
- -La mómoire CH00 = 0017 (n° 302) contient la constante :

  00 H0000 0000 = 00 1940 0000. L'adresse régionale H 0000 est l'adresse

  1940 du compteur de boucles n° 0. Le compteur n° d a comme

  adresse H 000 d = 1940 + d.

- La mémoire 0C00 = 0023 (n° 305) contient la constante : 00 0000 C0000 = 00 0000 1967. L'adresse régionale C 0000 = 1967 est l'adresse du registre d'index n° 0. Le registre d'index n° i a comme adresse C 000i = 1967 + i.
- Ce sous-programme a ctc assemblé à la suite de la partie interprétative en logique extérieure, ce qui a permis de profiter des affectations de celle-ci sans avoir à réutiliser trop de pseudo-codes.

#### VII - D - SOUS-PROGRAMMES PLACES A POSTE FIXE.

Deux sous-programmes ont été placés à poste fixe. Ce sont : la perforation d'une séquence de mémoires et la modification de bouclage. Ces sous-programmes ont été faits de façon différente pour la logique extérieure et l'exécution en langage machine : cela a permis d'économiser de la place en logique extérieure.

## VII - D - 1 - Perforation (ordre 49 N iA jB kC)

Cet ordre provoque la perforation de cartes "A mots par carte" (A \left\lambda 7).

Considérons par exemple la p ème carte. Elle contiendra le contenu de la mémoire B + (p - 1) A et des A - 1 suivantes si leurs adresses ne dépassent pas C. Appelons X la première mémoire représentée sur cette carte, soit B + (p - 1) A. La carte portera :

en mot I : 00 X A

en mot 2 : le contenu de X, soit (X)

. . . . . . . . . . . . . . .

en mot 2 + m : (X + m)

.............

en mot 2 + A - 1: (X + A - 1)

Si au contraire la carte considérée est la dernière et si par exemple X + 3 = C, la carte portera :

en mot 1 : OO X 4

en mot 2 : (X)

en mot 3: (X+1)

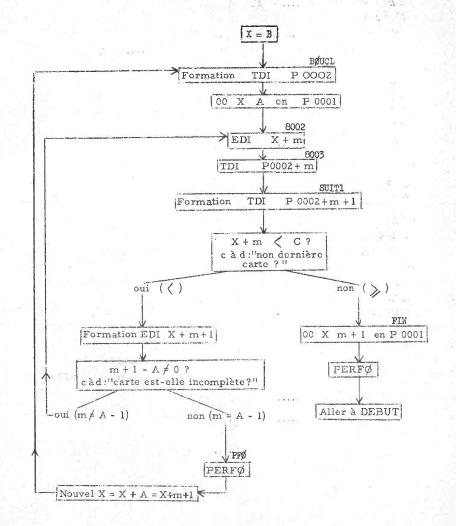
en mot 4 : (X + 2)

en mot 5 : (X + 3)

Pour la perforation, le mot l est préparé en P 0001 = 1977, le mot 2 en P 0002 = 1978, etc ....

J1971

10 450 65504



.../...

	Sous-	programme de perf	oration:		- 76 -
1 1 0000	EIDI 6, 9AG	go o	00 A B	69 xxxx 8003	1
2 1	D AG 0004	0 O AA	A B O O		
3 1	S BF W 0002			69 B 8003	en W2 = 1992
4	DIAD 0008 1	00	0 O A		
5	T RD W 0003	· ·		0 0 A	en $W3 = 1993$
6	A AG P 0008	0 0 C			
7 1	DIAG 0004	00 C 0000	0000 A 00		
8	A AD 2 4PS1		24P2 + A SUIT1		
9	T RD W, 0004			24 P2+A SUIT1	en W4 = 1994
10	T'RG W 0005	PHS at	6	00 C 0000	en W5 = 1995
11	R AD W 0002 B ØU	I. 00	69 B 8003	-	
12 B. ØUCL	A AG 2 4PS1	24P0002 SUTT1	69 X 8003		
13	E'DI W 0003			00 A	- Di 1077
14 (8002)	S. BF P 0001 800 E DI X 800	5 It 190 201 1	Section 6-1	ČO X A	en $P1 = 1977$
(8003)	T DI P'0002 S UII	ń		(x)	en P2 = 1978
15 S UIT1	A AG 0, 10	24P2+m+1 SUIT1			
16 1	S AD W 0005		68 X+m-C 8003	00 G 0000	
17	E DI ,8002			6 <sup>8</sup> X+m-C 8003	
18	SID9 FIN				X + m < C ?
19	A AD 0110		68 X+m-C+1 8003	Apple.	-
20	A AD W 0005		69 X+m+1 8003		
21	S AG W 0004	m+1-A		24 P2+A SUIT1	
22 1	G NN P FØ			8,000	m + 1 ≠ A?
23	AJAG 18001 800	2 24P2+m+l SUIT1			
(8002)		13	]	(X+m+1)	
(8003)	T DI P 2+m+1 S UIT	1	18 >	(X+m+1)	en P2 + m+l
24 P FØ	BIEN BOOT BOOC	L 0 0	69 X+m+1 8003		
25 F IN	R, AD 18003	0 0	24 P2+m+1 SUIT		111.201
26)	S AD 2 4PS1		00 m + 1 0000	00 X A	
27!!	E DI P 0001 DIAD 0004	The state of the s	Q 0 m + 1		
29	S BI P 0001	0	CONTRACTOR	00 X m+1	en Pl = 1977
30	P,FØ P 0001 180				1
31 6 9AG 32 2 4PS1	169 0000 800 24 P 0002 S UIT		10 1 2 1 1 2 W		
33 0110	100 10001 1000				

- Ce programme a été écrit indépendamment de la partie interprétative et assemblé seul, dans la bande 0000. Il a ensuite été translaté. Son facteur de translation qui a été fixé à  $\Delta_{49} = 1563$  est placé dans la mémoire DEL + 49 = 1875 + 49 = 1924. Pour l'assemblage, le programme en PAS $\phi$  a été précédé des pseudo-codes suivants :

| P'RA | 10000 | 10000 | 10000 | 10000 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 1999 | 199

- Quand l'exécution du sous-programme débute, l'accumulateur gauche contient zéro, le droit contient 00 A B et le distributeur 20 C ANALY; en effet, l'entrée se fait comme pour un sous-programme à 3 adresses ordinaire (PP' = 49). Mais ce sous-programme n'est pas standard.

  En particulier, il ne tire pas C du distributeur mais de la mémoire P 0000 = 1984 où l'a placé la partie interprétative. Le sous-programme de perforation destiné à l'exécution en langage machine est, au contraire, standard.
- Le sous-programme utilise les mémoires de travail W (1991,....) et les mémoires de perforation P (1977,...). Or, en logique extérieure, l'analyse utilise également les mémoires P (1977,...). A moins d'utiliser d'autres mémoires de perforation pour le programme de perforation (ce qui aurait 616 encombrant), il était pratiquement impossible de concilier ce programme avec l'analyse. Il a été jugé peu génant de supprimer

l'analyse des ordres de perforation. En logique extérieure, si le signe du pupitre est -, les ordres de perforation 49 fonctionnent donc normalement mais ne sont pas analysés. Remarquons qu'au contraire les ordres de perforation simple - 06 U V sont analysés. Comme l'analyse n'existe pas pour l'ordre 49, c'est par un envoi à DEBUT = 1800 et non à ANALY = 1700 que se termine le sous-programme.

- La première partie, jusqu'à BØUCL, forme un certain nombre de mots utiles dans la suite.
- Lorsqu'on arrive à BØUCL (venant du début ou de PFØ) l'accumulateur gauche contient zéro et le droit contient 69 X 3003, X étant l'adresse de la première mémoire représentée sur la carte en préparation. Pour la lère carte, X = B. Pour la 2ème, X = B + A, etc... Chaque fois que l'on commence la préparation d'une carte, on revient à BØUCL. On envoie d'abord 00 X A en P 0001 = 1977 qui sera le mot l de la carte.
- Pour exécuter le test X + m < C?, on se sert d'un test SD. Dans l'accumulateur droit, se trouve 69 X + m 2003. On en soustrait 00 C 0000. Si X + m > C, on obtient 69 X + m C 2003. Si au contraire X + m < C, on obtient 60 10000+ X + m C 3003. Ainsi, en faisant un test SD sur la '9ème position de ce mot, on fait un test sur X + m C. Pour cela, on amène le résultat de la soustraction dans le distributeur et on effectue le test SD9 (instruction n° 13).
- Pour obtenir m + 1 A, on retranche de 24 P2 + m + 1 SUIT 1 la quantité 24 P2 + A SUIT 1 formée en W 0004 = 1994 par la Tère partie.
- La branche commençant en FIN correspond à la dernière carte à perforer ; cas où X + m = C. Dans ce cas, il faut remplacer le mot 1 par 00 X m + 1 (1 < m + 1 < A). Pour former m + 1, on soustrait à 24 P2 + m + 1 SUIT 1 la quantité 24 P2 SUIT 1 contenue dans la mémoire 24 PS 1.

#### VII - D - 2 - Modification de bouclage (ordre 00 N 0 1937 jB kC):

Il arrive souvent qu'on ait à écrire un ordre de bouclage dont le nombre H de boucles soit variable. Par exemple, citons le problème suivant : on lit une carte comportant : en mot 1, x; en mot 2, h; en mot 3,0....0 n, Calculer f  $(x_0)$ , f  $(x_0 + h)$ ,..., f  $(x_0 + n-1 h)$ , f (x) étant une certaine fonction imposée. On aura, dans ce problème, à décrire n fois la boucle calculant f (x). Mais le n variera d'une carte lue à l'autre. On pourrait penser qu'il suffit d'écrire l'ordre de bouclage 9i N 

✓ 0000 O B O C et de commander l'addition en virgule fixe de la mémoire contenant 0.... 0 n à la lère mémoire de l'ordre de bouclage. Mais cela n'est pas possible à cause de l'autoprogrammation. En effet, en exécution en langage machine pour un ordre de bou clage placé dans les mémoires D et D + 1 par exemple, H se trouve sous la forme 0.... 0 H en mémoire D + 1. Ce n'est pas en mémoire D qu'il faut ajouter 0.... 0 n (comme en logique extérieure) mais en D + 1. On ne peut donc pas utiliser un ordre d'addition valable à la fois pour la logique extérieure et l'exécution en langage machine. Il a fallu créer un nouvel ordre, l'ordre de modification de bouclage. L'ordre de modification de bouclage 00 N 0 1937 ¡B kC consiste à remplacer le H de l'ordre de bouclage placé en C et C + 1 par H' placé en B sous la forme 00 0000 H'.

Le programme de modification de bouclage est différent en logique extérièure et en autoprogrammation. Il a dans les deux cas été écrit sous la forme d'un sous-programme (à 2 adresses) utilisant l'entrée standard. En logique extérieure, le facteur de translation de ce sous-programme a été fixé à  $\dot{\Delta}=1556$ . Comme c'est un sous-programme à 2 adresses, on aurait dû écrire l'ordre de modification de bouclage sous la forme 00 N 0 1556 jB kC. Or, comme le facteur de translation en exécution en langage

n'a pas pu être fixé à 1556, on ne peut pas appeler cet ordre 00 N 0 1556 jB kC. On a transféré les deux premières mémoires du sous-programme (1556 et 1557) en 1937 et 1938, disponibles en logique extérieure et en langage machine, et appelé l'ordre 00 N 0 1937 jB kC. Schblablo opération a été effectuée pour l'exécution en langage machine.

#### Sous-programme de modification de bouclage :

24.5.							
# Ba		93	il.	65 B A+1	20 C ANALY		w.= 14 °
A = 1937	15	1558	1559	10= 1	23 C ANALY	0300000000	
Δ+ 0002	03	0000	0000	21.2	19%	de the	
3	20	1991	0004	Early A-	" PERMITS ?	23 C ANALY	en 1991 = W 0001
4	69	00.05	0006		Tell # 5	69 0000 1991	
5	69	0000	1991	5 Blycu	veges in	# # 8	2 F
6	22	1938	8003		= <sup>1</sup>	69 C 1991	en 1938 = A + 1
(8003)	65	В	A+l	00	ОО Н'	1 3 m st	es" a vij "
+1 =(1938)	69	C	1991		h 16	91 N od H	, a 1 10
1991)	23	C	ANALY			9i N & H'	en C

Les mémoires  $\Delta$ = 1556 et  $\Delta$ + 1 = 1557 n'ont plus aucune utilité et ont été libérées.

VIII ASSEMBLAGE ET DISPOSITION DE LA

Voici la disposition sur le tambour des différentes parties de la logique extérieure

Numéros de mémoires	Utilisations
 1555	Sous - programmes du C.D.P. choisi
1556 1557	Libres
1558 1562	Modification de bouclage ( début en 1937 )
1563 1595	Perforation (ordre 49)
1596 1638	Bouclage
1639 1670	PARTIE SPECIALE INDEXAGE
1671 1881	PARTIE INTERPRETATIVE
1882 1883	Libres
1884	MEMNN
1885 1936	Mémoires réservées aux Δ <sub>pp</sub> , (région D )
1937 1938	Début de modification de bouclage
1939	Libre
1940 1949	Compteurs de boucles ( région H )
1950	Libre
1951 1960	Mémoires de lecture ( région L )
1961 1966	Libres
196 <b>7</b> 1976	Registres d'index ( région C )
1977 1986	Mémoires de perforation ( région P )
1987 1990	Libres mais 1987 est utilisé par le sous - programme auxiliaire de décalage
1991 1999	Mémoimes de travail (région W)

... / ..

....

Le programme de chargement (voir paragraphe VIII-A) occupe la plupart des mémoires de 1940 à 1999. Il est donc détruit par l'exécution des calculs. Inversement, si on le réutilise après un calcul, il détruit les compteurs de boucles, les registres d'index, etc....

Les compteurs de boucles, mémoires de lecture, registres d'index, mémoires de perforation et mémoires de travail ont été placés dans la zone qu'occupe le programme de chargement parce que ces mémoires n'utilisent justement pas celui-ci, rien n'y étant enregistré. Notons que les compteurs de boucles et les registres d'index ne sont pas à zéro après le chargement (sauf le registre d'index n° 0) et qu'il est indispensable, si on les utilise, de les remettre à zéro par l'ordre - 02 U V.

#### Composition du paquet de cartes PAS $\phi$ :

1°) Pseudocodes (n° 1 à 22)

Les pseudo-codes utilisés sont de 4 sortes : PRA, PRG, PEN, PER.

#### PRA: il y en a 3.

- PRA 1639 1670 | Pourraient très bien être remplacés par
- PRA 0000 1638

le seul ordre PRA 0000 1670

- PRA 1940 1999 En fait, cet ordre pourrait être PRA 1940 1990.
- PRG et PEN: Après chaque réservation sera indiquée l'utilisation des mémoires réservées.

Les 4 premiers PRG nommés servent seulement à définir des régions, ces régions étant déjà réservées par le PRA 1940 1999.

- PRG H 1941 1949 : compteurs de boucles H 0000 = 1940 est réservé par le PRA 1940 1999.
- PRG C 1968 1976: registres d'index C 0000 = 1967 est réservé par le PRA 1940 1999.
- PRG P 1977 1986 : mémoires de perforation.
- PRG W 1991 1999 : mémoires de travail.

- PRG D 1886 1939: Mémoires réservées aux Δ<sub>PP</sub>.
En fait la table de ces mémoires ne va que jusqu'à 1936, Ce PRG réserve en plus les mémoires 1937, 1938, 1939 qui pourraient aussi être réservées par une transformation du PRA 1940 1999 en PRA 1937 1999.

- PEN D 0000 1885 : Première mémoire de la table réservée aux  $\Delta_{\mathrm{PP}^1}$  . Destinée à recevoir  $\Delta_{\mathrm{10}}$  .
- PRG F 1847 1855: Premières instructions des parties interprétatives particulières aux ordres PP' = 0P' (F 000 P').cf organigramme général.
- PEN F 0000 1846: Première instruction de la partie relative à PP' = 00 (sous-programmes à 2 adresses).
- PRG G 1857 1865: Premières instructions des parties interprétatives propres à chaque P (G 000 P). cf organigramme général.
- PEN G 0000 1856: Première instruction de la partie relative à P = 0 (sous-programmes à 2 adresses, opérations en virgule fixe, et tests SI et SD).
- PRG T 1867 1883: Premières instructions des branches particulières aux PP' pour les ordres de service (T 00 PP'). Cf organigramme général, ordres à 1 mémoire.
- PEN T 0000 1866 : Première instruction de la partie relative à l'ordre 00 U V.
- PEN DEBUT 1800 : Première instruction de la partie interprétative.
- PEN MEMNN 1884 : Mémoire contenant l'adresse de l'instruction suivante : 00 0000 N.
- PEN ANALY 1700 : Première instruction de l'analyse.
- PEN PPPRI 1750 : Mémoire où est envoyé 0 .... 0PP'.
- PEN 001 1764 : Constante.
- PEN PFDEB 1738 : Ordre de perforation de la carte d'analyse.

... /...

- Ces 3 dernières réservations ont été faites pour pouvoir éviter de réassembler le sous-programme de bouclage si on réassemble le reste.
- PER: PER DEL 1875 est une affectation sans réservation. En effet la lère mémoire utile dans la famille DEL + PP' est DEL + 10 = 1885 et non DEL = 1875.

#### 2°) Partie interprétative sans index (n° 23 à 222)

- De 23 à 40 analyse
- -De 41 à 47 : partie commune à tous les ordres
- De 47 à 158: partie relative aux ordres à 2 mémoires.

  Carte n° 73: la mémoire INNIN = 1759 contient (cf VII-C-1 b) TRG

  P0006 NINCH = 21 1932 1303 si le paquet "index" n'est pas chargé et

  TRG P 0006 INCHA = 21 1932 1785 s'il est chargé. Dans la partie
  interprétative sans index, le vrai contenu de INNIN est TRG P 0006 NINCH.

  On lui a donné la valeur TRG P 0006 INCHA pour fixer INCHA dès cette
  carte. Après assemblage, il a fallu remplacer 21 1932 1735 par 21 1932 1803

  et ajouter au paquet d'index la carte chargeant 21 1932 1735 en INNIN = 1759.
- De 159 à 185 : constantes relatives à la partie précédente.
- De 186 à 222 : partie interprétative propre aux ordres de service.
- 3°) Partie "index" (224 à 253)

  Il a fallu, après assemblage, ajouter au programme obtenu pour la partie

  "index" la carte chargeant TRG P 0006 INCHA = 21 1982 1785 en INNIN=1759.
- 4°) Sous-programme de bouclage (259 à 305) Voir le chapitre VII - C - 4.

#### IX - AUTOPERFORATION - GENERALITES -

L'autoprogrammation ou compilateur provoque la formation d'un programme en langage machine équivalent au programme en CDP: pour chaque ordre en CDP est formé un groupe d'ordres en langage machine. Ce groupe sera capable de jouer le rôle de l'ordre en CDP; certains des groupes se suffisent à eux-mêines pour jouer ce rôle, d'autres ne sont que des entrées dans des sous-programmes.

Pour chaque ordre en CDP la machine, grâce au programme d'autoperforation, perfore une ou plus souvent 2 cartes "n mots par carte"; ces cartes portent le morceau de programme en langage machine correspondant à l'ordre en code, ainsi que son emplacement. Pour exécuter le programme en langage machine obtenu, il faudra charger les cartes produites par l'autoperforation, j'intes à un paquet vert constitué par les sous-programmes utiles. De même que pour l'instant il existe trois paquets de logique extérieure (virgule fixe, virgule flottante cimple précision, virgule flottante double précision), il existe au choix trois paquets verts de sous-programmes à joindre au paquet autoperforé. Au contraire le programme d'autoperforation, qui représente la partie interprétative du proédé d'autoprogrammation, est en principe unique. En fait, comme on le verra plus loin, il faut joindre à ce paquet quelques cartes destinées à placer dans les mémoires DEL + PP' = 1875 + PP' les facteurs de translation des sous-programmes constituant le paquet vert choisi. Ces cartes sont différentes pour chacun des 3 paquets verts. On a, pour plus de commodité, formé 3 paquets d'autoperforation (bleus), chacun constitué du programme interprétatif unique et des cartes chargeant les facteurs de translation voulus.

.../..

## 

Le programme en langage machine est plus encombrant que celui en CDP. Il occupera, en plus de la place qu'occupait ce programme, une séquence de mémoires choisie par le programmeur. Avant l'autoperforation, celui-ci doit indiquer à la machine les limites les la decette séquence : il doit entrer 00 0000 les en 1832 et 00 0000 les max

Pour chaque ordre en code, le morceau de programme en l'angage machine correspondant occupera la place qu'occupait l'ordre (une, deux ou trois mémoires) et, si celle-ci est insuffisante, un certain nombre de mémoires de la séquence  $l_{\min}$ ,  $l_{\max}$ . Par exemple, si le ler ordre autoperforé exige 5 mémoires dans la séquence  $l_{\min}$ ,  $l_{\max}$ , il occupera les mémoires  $l_{\min}$ ,  $l_$ 

Dans le morceau de programme en langage machine correspondant à un ordre en CDP placé en n (et éventuellement n + 1 et n + 2), le ler ordre à exécuter est toujours celui qui est en n. La correspondance entre le programme initial en CDP et le programme en langage machine obtenu est ainsi évidente, ce qui facilite la consultation de celui-ci en cas de besoin et même la modification directe du programme en langage machine. Sur nour les tests, le dernier ordre à exécuter (dans un morceau de programme correspondant à un ordre en CDP) a comme adresse instruction suivante : N.

Il se peut que la séquence l choisie soit insuffisante. Dans ce cas, la machine demande la lecture d'une carte portant en mot l'un rouveau l et en mot 2 un nouveau l (voir le paragraphe X - B).

Remarque: l'autoperforation d'un ordre ne modifie rien à cet ordre ni aux mémoires de la séquence l'min', l'max'

#### IX - B - NOTATIONS:

- Nous appellerons n l'adresse de l'ordre en CDP considéré. Celui-ci occupera la mémoire n, ou les mémoires n et n + 1, ou les mémoires n, n + 4 et n + 2.
- Nous appellerons 1 la lère mémoire libre de la séquence l'min, max pour l'ordre en CDP considéré. Ceci suppose que les mémoires l'min à 1 1 ont été utilisées pour les Ardres précédents.
- Une mémoire sert de compteur pour les 1. Elle est nominée CTRL = 1882. Le veleur initiele de 1 est 1<sub>min</sub> que l'on enregietre en 1882. La mémoire

to the speciment of the mean reserver will be a set out of the

# IX - C - PREPARATION DES CARTES A AUTOPERFORER :

Four la plupart des ordres en CDP, 2 cartes "n mots par carte" sont perforées. La tère porte les ordres qui prendront la place de l'ordre en code. Sin rict 1 a la forme suivante : 00 n 000p avec p = 1, 2 vu 3 selon que l'ordre en code occupi, una, deux ou trois mémoires. La 2ème porte les ordres à charger dans les inémpires 1, d + 1, ... de la séquence l min, l son mot 1 a la forme suivante 100 l 000q, q étant le nombre de mémoires utilisées dans la séquence d min, l par l'ordre en C.D.F. considéré.

Subject to the second of the s

.....

Le programme d'autoperforation prépare la perforation de la lère carte de la façon suivante :

en P 0001 = 1977 : 00 n 000p

en P 0002 = 1978 : Mot à charger en n

en P 0003 = 1979 : Mot à charger en n + 1 (éventuellement)

en P 0004 = 1980 : Mot à charger en n + 2 (éventuellement)

La 2ème carte est préparée de la façon suivante :

en W 0001 = 1991 : Mot à charger en 1

en W 0002 = 1992 : Mot à charger en 1 + 1

en W 0003 = 1993 ; Mot à charger en 1 + 2

en P 0005 = 1981 : Mot à charger en 1 + 3 q mots

en P 0008 = 1984 : Mot à charger en 1 + 6

Après la perforation de la lère carte, 00 l q est envoyé en P 0001 = 1977; le contenu de W 0001 = 1991 est envoyé en P 0002 = 1978; celui de W 0002 = 1992 en P 0003 = 1979; celui de W 0003 = 1993 en P 0004 = 1980. Puis la 2ème carte est perforée.

#### IX - D - SCHEMA GENERAL:

Comme on logique extérieure, l'interprétation d'un ordre en autoprogrammation débute toujours par l'instruction placée en DEBUT = 1800 (n° 23). Les seules indications demandées par la machine à ce moment sont : l'adresse n de l'ordre à interpréter (celle-ci doit être placée sous la forme 00 0000 n dans la mémoire MEMNN = 1884) et l'état du compteur CTRL = 1882. L'autoperforation se passe suivant le schéma suivant :

.../...

- recherche de l'ordre dans sa ou ses mémoires (commence en 1800)
- découpage de l'ordre et séparation des diverses indications
- mise en place de l'adresse N de l'instruction suivante en MEMNN = 1884.
- formation des ordres en langage machine dans les mémoires P 0002 = 1978, P 0003 = 1979, ..., P 0008 = 1984 et W 0001 = 1991, W 0002 = 1992, W 0003 = 1993 (voir le paragraphe IX C)
- perforation de la ou des 2 cartes portant les ordres en langage machine.
- retour à DEBUT = 1800.

Remarque: L'autoperforation s'exécute sur le programme en CDP enregistré complètement, et non carte par carte.

#### IX - E - ORGANIGRAMME - (voir figure page 89bis)

La première opération que fait la machine est d'amener dans l'accumulateur droit le contenu (n) de n et de tester son signé, comme en logique extérieure. Dès cet endroit se séparent les trajets suivis pour les instructions à 1 mémoire et celles à 2 mémoires. C'est en effet le signe de (n) qui les différencie. Avant d'examiner les branches relatives respectivement à (n) positif et négatif, voyons comment se passe la perforation, partie finale commune à tous les ordres.

#### IX - E - 1 - Perforation :

Tous les ordres à 2 ou 3 mémoires provoquent la perforation de 2 cartes sauf le t est SD non indexé). Cuand le programme parvient à l'ordre PER-F $\phi$  = 1400 (n° 72) qui débute la partie "perforation", la lère carte est entièrement préparée et n'a plus qu'à être perforée. Pour la 2ème carte,

comme il est indiqué au paragraphe IX - C, la machine envoie les contenus de W 0001 = 1991, W 0002 = 1992, W 0003 = 1993 respectivement en P 0002 = 1978, P 0003 = 1979, P 0004 = 1980. Elle le fait même si q est inférieur à 3, ce qui n'est pas gênant. La machine, pour toutes les opérations citées ci-dessus, n'a besoin d'aucune indication particulière à chaque ordre en CDP. Par contre elle a besoin de connaître la valeur de q pour former 00 l q en P 0001 = 1977 (pour la 2ème carte) et pour faire avancer le compteur CTRL = 1882. Avant d'entrer dans la partie "perforation", il faut envoyer 00 0q00 000q dans l'accumulateur droit. On a formé les constantes 00 0100 0001 en PERF 1 = 1612 (n°408), 00 0200 0002 en PERF 2 = 1662 (n° 409) ...., 00 0500 0005 en PERF 5 = 1812 (n° 412); q ne dépasse pas 5. On a également formé les ordres : RAD PERF 1 PERFØ en VAPF 1 = 1795 (n°441), RAD PERF 2 PERFØ en VAPF 2 = 1748 (n°442), ...., RAD PERF 5 PERFØ en VAPF 5 = 1613 (n° 445). Toutes les branches relatives aux ordres à 2 ou 3 mémoires (sauf le test SD) rejoignent la partie "perforation" par une de ces 5 instructions. Pour 2 ordres de service, les branches particulières se terminent par VAPF 1 = 1795 (n° 441) et, pour un autre, par VAPF 5 = 1613 (n°445). L'ordre - 05 U V, un peu particulier, sera envisagé dans l'étude détaillée. Les 5 autres ordres de service ne provoquent la perforation que d'une seule carte. Les branches relatives à ces ordres rejoignent les autres en FINPF = 1436 (n° 88).

#### IX - E - 2 - Instructions à 1 mémoire (cas où le contenu de n est négatif) :

- La lère instruction de cette branche est en UNEME = 1358 (n°462). Le début de la branche est commun à tous les ordres de service (ordres à 1 mémoire). Il comporte l'envoi de 00 n 0001 en P 0001 = 1977 (voir IX - C)

PROGRAMMATION



et de 00 0000 V en MEMNN = 1884. Il se termine par la préparation de l'aiguillage à branches multiples "aller à T 00 PP' = 1866 + PP' " identique à celui en logique extérieure (voir VI - C - 1).

- Les parties propres à chaque code PP' seront envisagées au cours de l'étude détaillée.
- Toutes ces parties, après la ou les perforations (voir IX E 1), se terminent par un retour à DEBUT = 1800 (n° 23).
- IX E 3 Instructions à 2 ou 3 mémoires (cas où le contenu de n est positif):

  La lère instruction de cette branche est en DEUME = 1359 (n° 36).

  Pour tous les ordres indexables, les cartes perforées par l'autoperforation sont différentes selon que l'ordre est indexé ou non.

### a) Partie commune :

Cette partie est essentiellement formée du découpage des ordres et de la préparation du test : "aller en G 000 P = 1856 + P". Ce test est identique à celui effectué en logique extérieure (VI - C - 2 -  $\delta$ ).

#### b) Branche particulière à chaque valeur de P :

α)P.= 0 : la machine fait tout de suite un test sur la nullité de P'. Si P' = 0, il s'agit d'un ordre d'entrée dans un sous-programme à 2 adresses; le programme continue en 3P2A = 1369 (n°326). Le rôle de cette branche sera de construire et perforer les ordres nécessaires à l'entrée dans le sous-programme. Après une petite partie commune aux ordres indexés et non indexés, les branches correspondant à ces 2 cas divergent. La branche correspondant à j = k = 0 (ordre non indexé) commence à NIN2A = 1753 (n°348) et se termine par un envoi à VAPF2. Celle qui correspond à j ou k ≠ 0 (ordre indexé) commence en IND2A = 1703 (n°334) et se termine en VAPF3.

Si P' \neq 0, il s'agit d'une opération en virgule fixe ou d'un test SI ou SD; le programme continue en VFX = 1368 (n°133). La partie du programme d'autoperforation relative aux ordres PP' = 0P' (P' \neq 0) est la plus encombrante. Mais l'encombrement en autoperforation est un facteur sans importance. Les branches correspondant à i = j = k = 0 et à i ou j ou  $k \neq 0$ se séparent tout de suite. En IND1 = 1479 (n° 136) débute la branche correspondant aux ordres indexés. La machine forme un certain nombre d'instructions communes à toutes les valeurs de P' puis effectue un aiguillage suivant la valeur de P': "aller à F000P' = 1846 + P'". Cet aiguillage est analogue à celui de la logique extérieure sauf que le cas P' = 0 n'y est pas inclus. La mémoire F0009 = 1855, non utilisée pour le moment, est libre, ce qui permettra de créer éventuellement un ordre PP' = 09. En NIND1 = 1480 (n° 236) débute la branche correspondant aux ordres non indexés. Après une courte partie commune aux diverses valeurs de P', s'effectue l'aiguillage "aller en X000P = 1837 + P' ", semblable à l'aiguillage "aller à F000P' ". X0009 = 1846 est libre.

β)P = 1 à 5 : il s'agit des ordres d'entrée dans les sous-programmes à 3 adresses. Cette partie de programme est destinée à construire et perforer les ordres en langage machine nécessaires à l'entrée dans un sous-programme à 3 adresses. Ces ordres, comme nous le verrons plus loin, sont différents selon que l'ordre en C. D. P. est indexé ou non. Après la formation de quelques instructions communes, les branches relatives aux ordres indexés ou non se séparent. En IND3A = 1806 (n°378) commence la branche relative aux ordres indexés et en NIN3A = 1660 (n°392) celle relative aux ordres non indexés.

N)P =8 : n'est pas encore créé.

5) P = 9 : ordre de bouclage. Cette partie du programme d'autoperforation forme et perfore les ordres en langage machine nécessaires à l'entrée dans le sous-programme de bouclage placé à poste fixe dans les paquets verts à joindre aux programmes autoperforés.

.../..

## X - ETUDE DETAILLEE DE L'AUTOPROGRAMMATION

Quelques détails propres à certains ordres, notés dans la description de la logique extérieure, n'ont pas été répétés dans la description de l'autoprogrammation.

#### X - A - INDEXAGE.

Ce n'est pas pendant la phase d'autoperforation que s'exécute l'indexage. En effet, aucun ordre du programme ne s'exécute pendant cette phase, en particulier pas les ordres de bouclage. Ce n'est qu'au cours de l'exécution en langage machine du programme autoperforé que l'indexage peut se faire. Il existe pour cela 3 sous-programmes d'indexage placés à poste fixe dans les paquets verts à joindre aux programmes autoperforés. L'un de ces programmes est relatif aux opérations en virgule fixe, un autre aux sous-programmes à 2 adresses et le 3ème aux sous-programmes à 3 adresses. L'indexage des 2 ordres de tests n'utilise pas ces sous-programmes; tout leur programme d'indexage (très court) est obtenu par autoperforation (voir X - E - 2 - d - \pi ).

Contrairement à ce qui se produit en logique extérieure, le programme d'autoperforation effectue pour chaque ordre indexable un test pour savoir s'il est indexé ou non. S'il ne l'est pas, la machine forme et perfore des instructions en langage machine destinées à exécuter

l'ordre directement pour ne pas perdre de temps. S'il est indexé.

la machine forme et perfore des instructions capables <u>d'exécuter</u>
l'ordre après le rancage par l'un des 3 sous-programmes d'indexage.

Ces mots formés contiennent les indications nécessaires sous la forme la plus compacte possible.

Les 3 sous-programmes d'indexage ont été rassemblés en un seul. Ce sous-programme unique a été écrit en langage machine dans les mémoires 0000 à 0074. Il a été ensuite perforé en cartes translatables 5 mots par carte, puis translaté. Son facteur de translation, placé en : EL + 52 = 1875 + 62 = 1937 pendant l'autoperforation, est égal à  $\Delta_i$  = 1807. Après chaque instruction du programme d'indexage nous écrit ons, dans la liste qui suit, les 2 indicatifs de translation :

le ler = 3 si l'adresse facteur doit être translatée = 9 sinon

le Zème = 8 si l'adresse instruction suivante doit être translatée

= 9 sinon

## X - A - 1-Indexage des opérations en virgule fixe.

Prenons pour exemple l'addition. Le cas des autres opérations est tout à fait semblable, même pour les opérations avec décalages (voir X - E - 2 - d - c X).

.../...

L'autoperforation fournit les ordres suivants pour un ordre d'addition indexé 01 N iA jB kC placé en n et n + 1:

n	RAD	n + 1	Δ <sub>i</sub>
n + 1	60	1	Δ <sub>i</sub> + 13
1	i0	000j	0001<
1 + 1	TRG	С	N
1 + 2	AAG	В	W 0002
1 + 3	RAG	A	M 0001

Ces ordres sont obtenus en langage machine pur (numérique). Comme pour tous les ordres, c'est l'inetruction placée en n qui s'exécute la lère. Elle provoque immédiatement l'entrée dans le sous-programme d'indexage.

n	65	n+1	1 di		-	0 0	60	1 0	i+ 13	60	1 Ai	+ 13			
A + 0000	24	1993	0001	9	8					60	1 4	+ 13	en '	W 0003	
100	1		0006	۱ E	- 1	1	10	1+1	4, + 25						
2	49	9998	9988	9	9				_						
3	05	0001	0001	9	9										
- 4	04	9998	2023	9	8										
5	00	0100	0000	9	9	.59.11			93 h.I						
6	20	1994	0007	9	8					10	1+1	Δ <sub>1</sub> + 25	en '	W 0004	
7	15	0003	0008	8	8		15	1+2	0, +25						
- 1	1	1	0009	ł 1	- 1				J.	15	1+2	A+ 26	en	W 0005	
9	16	0004	0010	8	8		10	1+3	8003			_			
10	20	1996	1993	9	9	- district		3)	7 6	10	1+3	8003	en	W 0006	
(1993)	\$	3		t I	- 1	i0000j000k	0 -		0	İ					
13	30	0005	0012	9	8	00000100000	jo	00k0	0						

Δ:+	12	10	0011	0014	8	8	60 CTRi Δ; + 27	j000k 00		
				0027	. 1	- 1				
	14	11	8003	0015	9	8	00		60CTRi ∆ <sub>i</sub> +27	No. of the last
	15	24	1997	0016	9	8		Trad (MI).	60CTRi A <sub>1</sub> +27	en W 0007
	16	35	0001	0017	9	8	0 0 j	000k 0 0	L'Albin'	TELL.
							000k 0 0	000j 0 0	0001000000	
	18	30	0002	0019	9	8	00000k 0000	00000 j00000		
	19	10	0020	0021	8	8	10CTRk ∆,+24		540 0 20	10° - 1
	20	10	1967	0024	9	8				
	21	21	1993	0022	9	8			10CTRk & +24	en W 0003
	22	15	0023	8002	8	9		65CTRj W3		
				1993						
	(8002)	65	CTRj	1993			0 0	0 O(CTRj)	P H	
							0 O(CTRk)		W I	
	24	35	0004	1994	9	9	00(CTRk)0000	00(CTRj)0000		
-	(1994)	10	1+1	0025		8	21 C' N		10 4.00 at	
- 1	25	21	1992	1995	9	9	5 1 2		21 C' N	en W 0002
	(1995)	15	1+2	0026		8	A	10 B, MS, .		
-1				1997			W 9		10 B' W2	en W 0001
								0 0		110
	27	35	0004	1996	9	9	00(CTRi)0000			
	(1996)	10	1+3	8003		C. Bough	60 A' W1			

- Cette partie du sous-programme d'indexage utilise les mémoires de travail W 0001 = 1991 à W 0007 = 1997.
- Les constantes placées en  $\Delta_{\hat{1}}$  + 4,  $\Delta_{\hat{1}}$  + 11,  $\Delta_{\hat{1}}$  + 20 sont translatables.
- Les ordres  $\Delta_1 + 0 \ge \Delta_1 + 10$  forment les ordres utiles par la suite pour la recherche du contenu des mémoires 1 + 1, 1 + 2 et 1 + 3.

.../...

l'accumulateur le contenu des registres d'index CTRi, CTRj, CTRk.

- Après l'ordre placé en W 0006 = 1996, s'exécute l'addition demandée.

- Les ordres  $\Delta_{i}$  + 13 à  $\Delta_{i}$  + 22 forment les ordres devant amener dans

- L'ordre qui était à l'origine placé en 1 + 3 est exécuté en 3003 aussitôt après l'indexage.
- L'ordre qui était en 1 + 2 est, après indexage, envoyé en W 0001 = 1991 pour être exécuté.
- L'ordre qui était en 1+1 est, après indexage, envoyé en W 0002 = 1992 pour être exécuté.

## X - A - 2 - Indexage des entrées dans les sous-programmes à 2 adresses.

L'autoperforation fournit les ordres suivants pour un ordre indexé d'entrée dans un sous-programme à 2 adresses 00 N OA jB kC placé en n et n+1.:

n	RAG	n+1	Δ <sub>i</sub> + 28
n + 1	60	1	Δ <sub>i</sub> + 36
1	00	000j	000k
1+1	RAD	В	A+1
1 + 2	TRD	С	N

La lère instruction qui s'exécute est, comme toujours, celle placée en n

	n	60	P'A	14:+2	8		60 1 A +30	5  0	60 1 A,+36	
٥,+	0028								60 1 A <sub>4</sub> +36	en W 0001
	29	11	0047	0030	8	8	10 1+1 W2			a san
		1	1994	1	1				10 1+1 W2	en W 0004
						9	15 1+4 A <sub>1</sub> +32	2.	27 2 4 4	
	(8003)	8		1	{	8		65 B A+1		
			0045		F	Į I		65 B A		
	33	10	0044	0034	8	8	15 1+2 0 +32	2 (3.07)		· ~ ~ *, i
	34	69	8003	0035	9	8			15 1+2 △+32	F
			1992	3	3	E L			15 1+2 A	en W 0002
							00000j000k	00	A.v.	
	36	30	0004	0037	9	8	00 j	000k00		
	31	11	8003	0038	9	8	00		00 j	co 43 Byrio 1
	- 1		0006	3		1 1		00 k		
							00 j			14,276
							0000050000	00000k0000	1	
			0020		,	5 5	10CTRj 01+24		10 1967 A <sub>i</sub> + 24	
			1993		5			1.6	10 CTRj 1+24	en W0003
			0023			9	2.4816	65CTRk W3	65 1967 W3	
	(8002)						00	00(CTRk)	1 4	1 45
	(1903)	10	CTRj	0024		8	0o(CTRj)			
	1	- 5	0004				00(CTRj)000C	00(CTRk)0000	= 04 =	
}	(1994)	10	1:1	1992			65 B' 4+1	74.		
	(1992)	15	1+2	1				20 C' N	_ g	58 58
	44	00	0001	0000	9	9		4		
	- 1	i	0000		1	- 1		1.0		
	46	04	9999	8040	9	8			Car en	
	3	1	9993			ŀ				

- Les constantes  $A_i$  + 46 et  $A_i$  + 47 sont translatables.

- Les ordres  $\Delta_i$  + 20 à  $\Delta_i$  + 35 forment les ordres qui iront chercher le contenu des mémoires 1+1 et 1+2.
- Les ordres  $\Delta_i$  + 36 à  $\Delta_i$  + 43 forment les ordres devant amener dans l'accumulateur le contenu des registres d'index CTRj et CTRk (A n'est pas indexable).
- L'ordre placé en W 0002 = 1992 provoque l'entrée dans le sous-programme (en A). A ce moment, les accumulateurs contiennent bien ce qui est demandé par le sous-programme.

## X - A - 3 - Indexage des entrées dans les sous-programmes à 3 adresses : L'autoperforation fournit les ordres suivants pour un ordre indexé d'entrée dans un sous-programme à 3 adresses PP' N iA jB kC placé en n et n + 1 :

n	RAD	n + 1	43 + 48	
n + 1	60	1	Δ <sub>PP</sub> ,	
1	iO	000j	000k	
1 + 1	00	A	В	
1 + 2	20	C	N	

La lère instruction qui s'exécute est celle placée en n.

			1 <sup>48</sup>			.0,0	60 1 App		, S
0048	9	0057	0049	18	8			118003 xxxx	
1	- 1		0050	1	1	Charles Act	4128	118003 App'	en 1 + 57
	- 1		0051		1		000060 1		A SE BOTA V.
			0052				60 1 0000		
1 1	- 1		0054	7			60 I 4,+63	HE DESCRIPTION	THE STATE OF THE
			0063			107			254.
1	-		0055					60 1 4 <sub>i</sub> +63	en W 0007
1			0058	4			15 1+1 W1	The state of	7.85 3.27
1 1	- 1	4.0	8072	1			BOUTER CITED	TALIFIE NO	Bridge of the D
1 1		1.0	0000	;	1		2 = 1 = 111	C Sec. Co.	March 31 Tall
1 - 1	- 6	Service of	0059				e, buyo jaji s	15 1+1 W1	en W 0004
1 1	- 1		0060	4 1	- 1		10 1 WO		The second second
	- 1		0026		- 1		10 1+2 4:+57	g it estilia. Tr	e Trans
61 0	0	0001	8067	9	8		_		į
( 26)2	0	1991	1997				1 2 1 2 2 2 2	10 1+2 Δ <sub>1</sub> +57	en W 0001
(1997) 6	0	1	0063	-	8	iO 000j000k	00	E THAT I SOL	zeth.
63 3	0	0005	0064	9	8	0000i0000	j000k00000	By the year of	West of a
64 1	0 0	0065	0066	8	8	15 CTRi Å, +27			170.00
65 1	5 ]	1967	0027	9	8		2 11 2 2 2 3		21.20
66 1	1 9	5003	0067	9	8	0 0		15 CTRi Δ,+27	7" Hall #
67 2	4 3	.997	0068	9	8			15 CTRi ∆,+27	
68 3	5 0	2001	0069	9	8	00j	000k00		
69 1	9/0	0005	0070	8	8	000k 00	000j00		
70 3	0	0002	C071	9	8	00000k0000	00000 j00000		
71 10	0 0	072	0073	8	8	60 CTRk W7	e yeu		
72 5	0 1	.957	1997	9	9		-5,1,0		
73 1	5 0	074	0062	8	8		15CTRj W4	2 4	
74 1	5 1	.967	1934	9	9			8.8	
62 20	0 1	996	8003	9	9			3	en W 0006
(8003) 60	0 0	TRk	1997			0 O(CTRk)	00		
(1997) 1	5 0	TRi	0027		8.	3 M W -	0O(CTRi)	in the production of	
( 27) 3	5 0	0004	1996			00(CTRk)0000	00(CTRi)0000		
(1996) 19	5 0	TRj	1994		-		oo(cTRi)(cTRj)		
(1994) 1	5 1	+1	1991				00 A' B'	5-16-4	
(1991) 10	0 1	+2			8	20 C' N			
				and deal	1	0	00 A' B'	20 C' N	
( 5%) 13					8		00 A' B'	20 C' N	/

- Les constantes  $\Delta_i$  + 53,  $\Delta_i$  + 56,  $\Delta_i$  + 61,  $\Delta_i$  + 65 sont translatables.
- Les ordres  $\Delta_i$  + 48 à  $\Delta_i$  + 60 servent à former les ordres qui amènent dans les accumulateurs le contenu des mémoires 1 + 1 et 1 + 2.
- Les ordres  $\Delta_i$  + 63 à  $\Delta_i$  + 74 servent à former les ordres qui amèneront dans les accumulateurs le contenu des registres d'index CTRi, CTRj, CTRk,
- Le dernier ordre, placé en  $\Delta_{i}$  + 57, provoque l'entrée dans le sous-programme (en  $\Delta_{ppi}$ ). A ce moment, les accumulateurs et le distributeur contiennent bien ce qu'exige le sous-programme.

## X - B - PERTORATION (voir IX - E - 1)

72 P'ERFØ P FØ P 0001	00	рооооор	i	Perfo lère carte
73 A AG C TRL	00 1		3- 1	
74 D'AG 0004	00 1 000q	00000p00000	Si .	
75 T RG P 0001			00 l q	mot 1, 2ème carte
76   A AG   18002	00 1+q 000q	49	m #	
77   L AD 10004	001 + q	р00000р000	ter stability on	
78 T RG C TRL			00 l+q	en CTRL = 1882
79 S AG L MAX	001+q-1 max	FE 32-058	4.00	
80 SIAG OLOS SUITE	001+q-1 <sub>max</sub> -2	M. mark	2	
81 S UITE E DI W OOOL				
82 T DI P 0002			mot 2	2ème carte
83   EIDI W 0002				
84     TIDII Pl0003		1	mot 3	2ème carte

85	EIDI   W   0003   1	1+q-1_max2		N. T.	
86	T DI P 0004			mot 4	2ème carte
87	A NG F INPF A RETI				$1 + q - 1_{max} - 2 < 0$ ?
88	f inpf p fø p 0001 d ebut			74. [46.4	
456	A RETL L EC 1951				Nouveaux 1 ot 1 max
457	E DI 1951			00 1'min	min max
458	TIDI CIRL		1.62	00 1' min	en CTRL = 1882
459	E DI 1952			00 1'	
460	T DI L MAX			00 1' max	en LMAX = 1883
461	RADMENN DEBUZ	100!	00 n 0000	20.00	e ign ui

- L'instruction n° 78 envoie en CTRL = 1882 l'adresse 1 + q de la lère mémoire de la séquence l<sub>min</sub>, l<sub>max</sub> disponible pour l'ordre suivant (la dernière mémoire utilisée pour l'ordre considéré est la mémoire 1 + q 1).
- Avant de perforer la deuxième carte, il faut vérifier si 1 + q 1 ne dépasse pas l Pour cela, est effectué le test : 1 + q 2 1 max. ○? Si la réponse est OUI, cela veut dire que 1 + q 1 < 1 max + 1, c'est-à-dire que 1 + q 1 < 1 max. La carte peut être perforée et on peut passer à l'ordre suivant. Si la réponse est NON, cela veut dire que 1 + q 1 > 1 max + 1, c'est-à-dire que 1 + q 1 > 1 max. La Jimite supérieure 1 de 1 est atteinte. La 2ème carte de l'ordre n'est pas perforée. La machine, en ARETL = 1737 (n° 456), demande à

connaître une nouvelle séquence  $l_{min}$ ,  $l_{max}$  par la lecture d'une carte, non chargement, portant 00 0000  $l_{min}$  en mot 1 et 00 0000  $l_{max}$  en mot 2. A ce moment, l'interprétation de l'ordre considéré recommence à DEBU2 = 1356 (n° 26) (voir X-C). La lère carte relative à cet ordre et qui a déjà été perforée perd sa valeur et est remplacée par une autre, différente car l'est différent. Elle n'a pas besoin d'être supprimée car, lors de l'enregistrement du paquet autoperforé, ce qu'elle aura chargé en n, n + 1 (et éventuellement n + 2) sera remplacé par ce que la nouvelle carte y chargera.

- Le fait qu'il n'y ait pas zéro dans l'accumulateur droit ne change rien au test "  $1 + q 1_{max} 2 < 0$  ?".
- La mémoire 002 = 1410 (n° 403) contient la constante 00 0000 0002.
- Pour les ordres de service qui ne provoquent la perforation que d'une carte, cette perforation est effectuée par l'ordre FINPF = 1436 (n°33).

## X - C - PARTIE COMMUNE A TOUS LES ORDRES.

Au début de cette partie, dont la lère instruction est en DEBUT = 1800, se trouve dans la mémoire MEMNN = 1884 la quantité CC 0000 n indiquant l'adresse n de l'ordre à autoperforer.

23 D EBUT R' AD M EMNN	00	00 n	John Co	may the		
24   DI AG 10004		00 n 0000	- 11 C In			
25 TRD M EMN DEBU2	68 5		00 n 0000	en MEMN = 1403		
26 D EBU2 T, RG P 0003	12 1 21	file letter film	00	en $P3 = 1979$		
27 T DI P 0004			QO	en P4 = 1980		
28   T DI P 0005	(X F2)	10, 10*11 33.74	00	en P5 = 1981		
29 T.DI P.0006	T-1	Charge .	00	en P6 = 1982		
30 1 T DI P 0007	100		00	en P7 = 1983		
31 T DI P'0008			00	en P8 = 1984		
32 EIDI FIF FIG	a Lugar	de siuil.	65 xxxx FH	10 m at 1		
33 F F 65 0000 F H						
34 F G S BF F F 8001	4 497	12.0	65 n FH	en FF = 1390		
(8001) R AD R F H	00	(n)		- W		
35 FIH AINGUINEME DEUME				test sur (n)		
(UNE MEmoire, DEUx Memoires)						

Cette séquence de programme sert à former et à exécuter l'ordre 65 n F H qui amène le contenu de n dans l'accumulateur, puis à exécuter le test qui adrosso: en UNEME = 1350 (n° 462) si l'ordre est à une mémoire et en DEUME = 1359 (n° 36) si l'ordre est à 2 mémoires. De plus cette séquence remet à zéro les mémoires P 3003 à P 3003 pour que les cartes perforées par l'autoperforation ne comportent pas trop de mots inutilement différents de zéro.

#### X - D - ORDRES A 1 MEMOIRE OU ORDRES DE SERVICE

X - D - 1 Partie commune à ces ordres : débute à UNEME = 1358 (n° 462)

		00	-PP' U V		
462 U NEME T RD W 10002	1			-PP'UV	en W2 = 1992
463   RIAD M.EMN			00 n 0000		
464 , A,AD O O1			00 n 0001		
465 T RD P 0001				00 n 0001	en Pl = 1977
466 RAV W10002	P.	00	PP U V	ed to a	
467   E.DI 8003	,		garge, a	00	10 m
468 S BI M EMNN				00 V	en MEMNN = 1884
469 EIDI  8003	1		5 th 10	00	
470 S.BF W 0001	1 1			00 U 0000	en W1 = 1991
471   D AG 0002		0OPP'	n n 00		
472   A AG C TE4	18003	0OTOOPP'		0OTOOOO	Formation aiguillage
(8003) s p  10000	r loopf'	2 200 11 545 1			Aiguillage

- Pour tous les ordres de service, la lère carte perforée doit avoir pour mot 1:00 n 0001. La préparation de ce mot est donc dans la partie commune. 00 n 0001 est envoyé en P 0001 = 1977 et sera ainsi perforé en mot 1.
- V, dans la plupart des cas, représente l'adresse de l'instruction suivante. L'envoi de 00 0000 V en MEMNN = 1834 se fait donc aussi pendant la partie commune. Le V en MEMNN est destiné à indiquer l'adresse de la prochaine instruction à autoperforer. Mais V est

aussi utilisé comme adresse instruction suivante du dernier ordre en langage machine à exécuter.

- La mémoire CTE4 = 1629 (n° 500) contient la constante 00 0000 T0000 = 00 0000 1066.

#### X - D - 2 - Partie propre à chaque code PP'

= Code - .00 U V: sans opération, aller à V.

L'ordre qui devra s'exécuter en langage machine est :

00 U V . Cet ordre devra être en n. L'autoperforation doit donc produire une seule carte ayant :

en mot 1 : 00 n 0001 formé par la partie commune

en mot 2 : 00 U V

Les seuls ordres du programme d'autoperforation particuliers à

- 00 U V sont donc:

531 T10000  RAV W'0002  V'APFØ	00	00	UV			0.50	ii bara ay
497 VAPFOTRD P 0002 F INPF				00	Ų	V	en P2 (puis n)
SC (FINPE PFØ PICCOL DEBUT			- 1				

- FINPF = 1436 est le dernier ordre de la partie "perforation" (cf X B)
- La mémoire W 0002 = 1992 contient PP' UV, envoyé par l'ordre UNEME (n° 462).

= Code - 01 U V : arrêt ; puis départ éventuel à V.

L'autoperforation doit produire une carte unique portant :

en mot 1 : 00 n 0001 en mot 2 : 01 Ú V

Le seul ordre particulier à - Ol U V est donc :

498 T|0001 R|AV | W|0002 | V|APEØ | 0 ... 0 | 01 U V | 01

#### = Code - 02 U V : initialisation

Cet ordre, comme en logique extérieure, a pour rôle de romettre à zéro les compteurs de boucles et registres d'index. Mais en plus, il régénère le 1-r ordre du sous - programme de bouclage. Comme nous le verrons plus loin ( paragraphe X - E - 4 - c ), pour  $\alpha=0$  le premier ordre du programme de bouclage ( 24  $\Delta_{\rm bou}^+$  0055  $\Delta_{\rm bou}^+$  0001 ) est remplacé par : 00 0000  $\Delta_{\rm bou}^+$  0035. Cet ordre est régénéré à la sortie de boucle, en même temps que sont remis à zéro le compteur de boucles et le registre d'index utilisés. Mais si l'on ne termine pas le bouclage, l'ordre n'est pas régénéré. Exemple : on effectue un problème comportant un bouclage avec  $\alpha=0$ . On doit, pour une cause quelconque, interrompre le déroulement de ce problème sans passer par la fin de boucle. Si l'on veut exécuter un autre problème sans recharger le C.D.P., le programme de bouclage ne fonctionnera pas normalement car son premier ordre sera détruit .

Pour éviter cet ennui, l'ordre - 02 U V, qui initialise les compteurs de boucles et registres d'index, initialise aussi la mémoire  $\Delta_{\rm bou}$  + 0000 en y entrant : 24  $\Delta_{\rm bou}$  + 0055  $\Delta_{\rm bou}$  + 0001.

Une partie du programme destiné à exécuter l'initialisation pendant l'exécution en langage machine est à poste fixe dans le paquet vert à joindre au paquet autoperforé. L'autre partie est fournie par l'autoperforation.

La partic que nous appellerons la "partie fixe" occupe les mémoires K = 1930 à K + 5 = 1935. Elle représente le programme de remise à zéro des compteurs, et est analogue au programme d'initialisation en logique extérieure (voir VII - B - 2).

Voici la liste du programme qui se déroule pour l'initialisation en exécution en langage machine.

						, ,	
n	RAG	1	1+1	44 1932 V	0	44 1932 V	•
1	44	1932	V	111111111111111111111111111111111111111			
1+1	TDI	1991	1+2			44 1932 V	en Wl = 1991
1+2	RAG	1+3	1+4	24 A bot 55 A bot 1		44 1932 V 24 A bot 55 Pot 1	
1+3 1+4	24 TDI	DOM	A +1 1930			24 4 <sub>bot</sub> 55 4 <sub>bot</sub> 1	en $\Delta_{\rm bou} = 1751$
1930	RAG	1933	8003	20 1976 1931	00		
3003)	TRD	М	1931			00	en M
1931	SAG	1934	1991	00 M-1940 0000			

(1991) 1932	9	1	1 1	20 M-1 1931		7.74	M-1940 ≠ 0	?
1933 1934		1976 1940	1		đ.			
1935			1931					

- Les ordres n et 1+1 servent à envoyer en W 0001 = 1991 l'ordre
  44 1932 V qui dépend de l'ordre 02 U V et ne peut par conséquent
  pas être mis à poste fixe.
- Les ordres 1+2 et 1+4 servent à initialiser la mémoire  $\Delta_{\rm bou}$ = 1751 en y envoyant 24  $\Delta_{\rm bou}$  + 55  $\Delta_{\rm bou}$  + 1 = 24 1006 1752, contenu en 1+3. Ces ordres ne dépendent que de la valeur de  $\Delta_{\rm bou}$ . Or  $\Delta_{\rm bou}$  a été fixé à 1751. On aurait pu mettre ces ordres dans la partie fixe. On ne l'a pas fait pour ne pas avoir trop de choses à modifier, avec risques d'oublis, si l'on devait modifier  $\Delta_{\rm bou}$ .
- Les ordres n et 1 ÷ 2 pourraient être des EDI mais la formation des RAG économise une mémoire.
- La scule différence (autre que des différences d'affectation d'adresses)
  entre la partie fixe de ce programme d'initialisation et celui de la
  logique extérieure (voir VII B 2) est que la réponse NON au test
  GNN envoie dans le ler cas à V et dans le 2ème cas à DEBUT.

L'autoperforation doit fournir 2 cartes : la lère doit porter :

en mot 1: 00 n 0001

en mot 2: 60 1 1+1

La 2ème doit porter, K étant égal à 1930 :

en mot 1: 00 1 0005

en mot 2: 44 K+2 V

en mot 3: 24 1991 1+2

en mot 4: 60 1+3 1+4

en mot 5: 24 4 +55 4 + 1

en mot 6: 24 A bou K

Voici la partie du programme d'autoperforation propre à l'ordre - O2 U V;

503 T1000	2 R AG CITRL	11	00 1	00	001	
504	D AG 10004 1	the Park	00 1 0000		40 0 0 0	The state of the s
505	A AG   8001	N 20 30-	00 1 1		001	
506	A, AD 8001	1.0		001	001	
507	A AG R GOL		60 1 1+1		15,10	15
508	T RG P 0002				60 1 1+1	en P2 (puis n)
509	A AG 0 33	i.z	60 1+3 1+4			
510	TIRG W10003 1			1.87	60 1+3 1+4	en W3 (puis 1+2)
511	A AD 2 4W12			24 W1 1+2		
512	T RD W 0002		and the same	2 5 8 6 5 76	24 1991 1+2	en W2 (puis 1+1)
513 1	RIAD DIO051 1		00	00 A <sub>bou</sub>	00 Abou	

514	1	D/AG 10004	le kin nj. es	00 A 600 0000		
515	1	A AD 8001		oo A bou A bou		
516	1	E DI 2 40K		200 000	24 0000 K	
517	1	SIBF P10006 I			24 A bou K	en P6 (puis 1+4)
518	1	A AD 2 4551		24 40 表55 40 表1	500	
519		T'RD P 0005			24 80±55 80±1	en P5 (puis 1+3)
520	l	R AG M EMMN	00 V	00		
521		A AG 4 4K2	44 K+2 V		18 5	
522	1	TIRG W OOOL V APF5		7.5	44 K+2 V	en Wi (puis 1)

- La mémoire RGC1 = 1510 (n° 446) contient la constante 60 0000 0001
- La mémoire 033 = 1663 (n° 451) contient la constante 00 0003 0003
- La mémoire 24 W 12 = 1713 (n° 452) contient la constante 24 W 0001 0002 = 24 1991 0002.
- La mémoire 24 0K = 1763 (n° 453) contient la constante 24 0000 K = 24 0000 1930.
- La mémoire 24551 = 1013 (n° 454) contient la constante 24 0055 0001
- La mémoire 44 K 2 = 1464 ( $n^{\circ}$  455) contient la constante 44 K + 2 0000 = 44 1932 0000.
- La mémoire D 0051 = DEL + 61 contient le facteur de translation du sous-programme de bouclage,  $\Delta_{\rm bou}$  = 1751 .

. . . / . . .

= Code - 03 U V : remise à zéro de la mémoire U

Le programme fourni par l'autoprogrammation est :

Il se perforera donc 2 cartes.

La partie d'autoprogrammation relative à l'ordre - 03 U V est :

- La mémoire TD 88 = 1638 (n° 424) contient 20 3000 0000
- La mémoire RG 00 = 1730 (n° 432) contient 60 0000 0000 .

= Code - 04 U V : entrée dans le distributeur du contenu de U et arrêt par l'ordre 01 U V.

Le programme fourni par l'autoprogrammation est :

Il se perforera donc 2 cartes.

.../...

La partie du programme d'autoprogrammation relative à l'ordre - 04 U V est la suivante :

532 TIC	0004 RIAV W 0002	0	104 U V	4	
533	SIAD 0 300		OI U V	0300	72
534	TIRD W COOL	TOTAL 1491	Charles	01 U V	en Wi (puis 1)
535	A,AG 619	6900			2 11 2 35
538	A AG C TRL	69 0000 1	M. Stock		
539	S AG 8003	00	Ol U V	69 0000 1	
540	SIBF PLOODS VIAPFA	l	The said	1 <sub>69</sub> U 1	en P2 (puis n)

The Property of the Reservation of the Property of the Propert

- La mémoire 0300 = 1778 (n° 541) contient la constante 03 0000 0000.
- La mémoire 69 = 1594 (n° 407) contient la constante 69 0000 0000.

#### = Code - 05 UV: aller en langage machine à U

L'autoprogrammation, pour cet ordre, provoque la création de l'ordre en langage machine : aller à U. Elle perfore une carte portant en mot 1:00 n 0001, et en mot 2:00 0000 U. Mais en plus elle perfore tout le programme en langage machine commençant à U et se terminant à l'instruction ayant 1800 comme adresse instruction suivante. Pour chaque ordre en langage machine, CFI (code opération, adresse Facteur, adresse Instruction) placé en J, la machine perfore une carte portant : en mot 1:00 J 0001

#### en mot 2 : CF I

Pour le dernier ordre, l'adresse I = 1800 est remplacée par V puis le programme interprétatif, après la perforation de la carte relative à cet ordre, continue en DEBUT = 1800 pour interprétér l'ordre suivant, V.

Remarque 1 : Le programme en langage machine commençant en U ne doit pas comporter de test : en effet après chaque ordre CFI, l'ordre suivant est indiqué par l'adresse I, si bien que la branche F no somit pas traitée par l'autoperforation.

On peut au besoin remédier à cela en ajoutant au paquet autoperforé des cartes portant la partie de programme qui n'a pas été obtenue. Cependant, à moins de prendre des précautions, il faut que le programme en langage machine se termine par une seule rentrée en CDP. Autrement dit, si le programme en langage machine se divise en plusieurs branches (ce qui est déconseillé), il faut que ces branches se rejoi gnent avant la rentrée en CDP. Si ce n'est pas le cas, les parties en CDP qui suivent les parties en langage machine n'ayant pas été programmées ne le sont pas non plus.

Remarque 2: Aucun ordre du programme en langage machine ne doit avoir d'adresse instruction suivante de la série 8000. En effet, comme le programme ne s'exécute pas en autoprogrammation, celle-ci, rencontrant une adresse I de la série 8000, part dans une fausse direction.

Il est en général facile d'éviter ces adresses - Exemple : la partie de programme :

200 60 0201 0205		0 0	Or was a light plant
201 69 0300 0302			terror in 1979
205 10 0206 8003			A STATE OF STREET
506 00 000 1 0000		1 10	- 159 Year Lad?
		n Fe S	acto was er a

peut très bien s'écrire :

200 60 0201 0205	69 0300 0302			
201 69 0300 0302		1 3		1
		2 yet 1. 1. 1.	Se 13 235	The transfer of the state of the state of
309 00 0001 0000	1			sa ni Ang. Ti ndaga en
207 21 0210 0210		. 'x "yild	69 0301 0302	en 210 ; at A d A T' i'd a
		A religion of	10.5	
3.00				
and the second	e in other	1-37	0.00	E STOR BY CHESTORING

Il faut, pour que l'autoprogrammation continue bien, que la mémoire 0210 contienne un ordre quelconque ayant comme adresse instruction suivante I=0302.

La partie du programme d'autoprogrammation relative à l'ordre - 05 U V est la suivante :

3 .					
473 T 0005	RIAD W 0001	00	00 0 0000	a 8	N 1000
474	D AD 0004		00 U		
475	T RD W 0001 U NE53		de fu	oo n	en W1 = 1991
1.	T RD P 0002   P FØ P 0001			$0 \cdots 0 $	en P2 PERFO
477	1 1	11	II	PaGnillia S	PERCO
478	R AD W 0001	00	oo. J	4 - 10 - 1 - 1 - 1	
479	D AG 7 0004	(6),21,88	00 J 0000	GW 3.9	
480	A AD 0 01		00 J 0001	001	
481	T RD P 0001	1		00 J 0001	en Pl
482	E DI U'NE51			67 ×××	
483	S BF U NE51 8001	1	1	67 J	en UNE51 = 1518
484 U NE51	167 10000	E 11340 35 m	ate to all	- 12 Park	
(8001	R,AV J		CFI	P.S 14	
485 1	E DI 8003			00	a, Phase Y
486	S,BI W,0001		3.7	00	en Wl = 1991
487	A AG 8001	0 0 I	P	Marine St.	T is and
488	S AG E NCØD	00I-1800		00 1800	
489	G NN UI NE52			2 0	I ≠ 1800 ?
490	A,AG   8001 U NE53	00 I	C F I		
491 U NE52	R AG 8002	C F 1800	.00		
492 1	A AD M EMNN		00 V		
493	S AG 8003	0		C F 1800	
494	S BI P 0002 F INPF		1/4	CFV	en P2
(F) INPF	PFF POOOL DEBUT	i.	1000		

- La première fois que le programme passe par l'ordre de perforation n° 477,
la mémoire P 0001 = 1977 contient 00 n 0001, placé par la partie commune aux
ordres de service. La mémoire P 0002 = 1978 contient 0 ... 0 U. La carte perforée contient donc : en mot 1 : 00 n 0001
en mot 2 : 0 ... 0 U

Le premier ordre exécuté en langage machine sera donc : "aller à U". Lorsqu'on passe à l'ordre n° 477 en venant de l'ordre n° 490, la carte perforée est de la

forme: en mot 1:00 J 0001

en mot 2 : C F I

La carte relative à l'ordre en langago machino C F I = C F 1800 est perforée par l'ordre FINPF = 1436 (n° 88) qui renvoie à DEBUT = 1800. Gette carte est de la forme :

en mot 1:00 J 0001

en mot 2 : C F V

- La mémoire 001 = 1634 (n°422) contient la constante 00 0000 0001
- La mémoire ENCØD = 1729 (n°501) contient la constante 00 0000 DEBUT = 00 0000 1800.
- = Codo 06 U V: perforation simple,

L'ordre qui devra s'exécuter en langage machine est : 7! U V. L'autoperforation doit donc fournir une carte portant :

en mot 1:00 n 0001

en mot 2:71 U V

Les seuls ordres particuliers à l'ordre - 06 U V dans le programme d'autoper-

.../...

= Code - 07 U V: lecture

Cet ordre est équivalent à l'ordre 70 U V à condition que les cartes lues ne soient pas "chargement" (voir VII - B - 2). L'autoperforation fournit une carte portant :

en mot 1:00 n 0001

en mot 2:70 U V

Pour cela, la partie propre à l'ordre - 07 U V est :

495 T1 0007 RIAD 61300 j	L!ECPF	00	630		0				17 62 1
496 (LIECPF) AIVD WIOOO2	VIAPFØ	17 tv	70	.U	A	07	U	V	- m Vg (i
497 (VI APFØ) TIRD P10002	FINPF				XX71.14	70	Ū	${\tt V}$	en P2 (puis n)

- La mémoire 6300 = 1578 (n° 502) contient 63 0000 0000.
- = Codo G9 U V: fin de séquence.

Cet ordre, en exécution en langage machine comme en logique extérieure, doit envoyer à U. Son utilité a été expliquée au chapître VII - B - 2.

L'autoprogrammation doit donc fournir une carte portant :

en mot I: 00 n 0001

en mot 2:00 xxxx U "aller à U"

Par contre, pendant l'autoprogrammation, l'instruction suivante doit être prise en V, c'est-à-dire que lors du retour à DEBUT = 1800 c'est bien V qui doit figurer en MEMNN = 1884. Or V a déjà été placé en MEMNN par la partie commune aux ordres de service. Le seul ordre spécial à - 09 U V est donc, en autoprogrammation :

499 T10009 D1AD 10006 V1APFØ 0 .....0 00 T 0009 U 497 (V1APFØ) T1RD P10002 F1 INPF

## X - E - ORDRES A 2 (OU 3) MEMOIRES. -

Certains détails valables à la fois en logique extérieure et en autoprogrammation n'ont été notés qu'au chapître VII relatif à la logique extérieure.

#### X - E - 1 - Partie commune à ces ordres : débute à DEUME = 1359 (n°36)

Cette partie comporte ledécoupage des ordres ainsi que la construction et l'exécution du test "aller à G000P = 1856 + P".

				٠.	6	00	i	PP'NNNi A	- 17	7	1			
36 D	EUME	E DI	1800	031	1				0	0	1			
37	111	S.BI	000	01		1			0		1 6	n C	0001 =	= 1968
38		SAD	800	01	1		- 1	PP' Ni 0000						
39	. 11	D, AG	1000	21		0OP		P* Ni 00000		100		9-1	inc	
40	1	A AG			1	00G000	P	~	0	OG0000				
41			W   000			01110000			0	OGOOOT	9 6	n W	0002 :	= 1992
42			C TE			0OP								
43	1	D AG	000		1	0OPP		N i 000000			- 1-			
44		SIAG	1800			00			0	OPP				
45			P PP		1	0		V	0	יממרי		n PF	PRI =	1360
46		D AG				OONNN	- 1	i 0	0	• /11			1 ****	-,
			M EM			OTTOMAN	- 1	1 0	0	ON		n MF	MNN =	1884
47					1		1		i0					= 1971
48	10.0	T RD				0	1	00 n 0000	10		1,	, L	0004	- 1711
49	1		M, EMI	N I	2111111111	00		00 n 0002	Hebrios	1 150				100
50	300	A AD		2					00 - ~	0002	1	m D1	(per	fo)
51			P 000	11	1 1	Phase in the		00 - 10		0002	15	II FI	(her	10)
52		A'AD	0 10					00 n+1 0002			- 1			/4
53		E DI			F K	13.5	-	- 44 EW	OLXX	XX FL	-	111		e al 18 8
213-	J	167	000	100	FL		1		CT	2 777		to 7	. 14	40
55 F		SBF	FJ		8001		1		67 n+	TFL	. 6	en rJ	r = 14	40
	V	R AV	]n+		FIL	00	. [	jBBBBkcccc	0		- 8	55.7		
56 F	L	E DI	800	- 1									0007	1.000
57		S BI.			vine m	Samme of the	-	Table 1	00 00	00 C		ur C	0003	= 1970
58	-	S AD	1800		1			jBBBBk0000						
59	112	DIAD	0,00	01	J. 44	ter to the		OJBBBBBk000	4	1. 1975	. 4	Land		F . V
60		E DI	800	03					0	0				
61	0.5	SBF	01000	02	1 = 14	A CHARLES	1	With Mile 1 of 1	00 B	0000	. 6	na C	0002	= 1969
62	1	S AD	800	01			- 1	0j0000k000						
63	11547	DAD	,000	03	1 4 4 1	4 (20) 585		0000 j0000k				12.1	13	
64	1	EIDI	800	03	1		f			0		100	5 00 4	
65		S BI	C 000	05	1	-	1		0	.0 k	€	en C	0005	= 1972
66		D,AD	1000	01	1		-	00000j0000						
67	)	SBF	C 000	05	,		-		00.00	000 to	Ok 6	en <u>C</u>	0005	= 1972
68		R AG	000	03		0 OC		0						20
69		DIAG	1000		11	00 C 00	00							
70			M EM	NN		00 C N			3.0					
71	i	T. RG	C. 000		W 0002				00	C N	6	en <u>C</u>	0006	= 1973
(W	10002)	SIDP	1 1000	00 1	G! 000P	1	- 1				1			
		,												

- Les mémoires C 0001 = 1968 à C 0006 = 1973 (registres d'index) servent ici de mémoires de travail. Les registres d'index n'ont en effet aucune utilité en autoperforation.
- La mémoire CTE 1 = 1388 (n° 402) contient la constante 00 0000 G 0000 = 00 0000 1856 qui, comme en logique extérieure, sert à former l'ordre "aller à G 000P" marquant la séparation entre les branches correspondant aux différentes valeurs de P. Cet ordre 00 0000 G 000P est mis enréserve en W 0002 = 1992 où il est envoyé par l'ordre n° 41.
- L'ordre n° 47 met en place l'adresse N de l'instruction suivante dans la mémoire MEMNN = 1884.
- Quand toutes les indications possibles ont été puisées de la partie gauche de l'instruction, on amène la partie droite, d'adresse n + 1, dans l'accumulateur. Pour cela on forme l'ordre 67 n+1 FL = 67 n+1 1405 qui amène dans l'accumulateur droit la valeur absolue de la partie droite de l'ordre. Avant de former cet ordre, le programme profite de la présence de n dans l'accumulateur pour former 00 n 0002, mot l de la lère carte à perforer si l'ordre est à 2 mémoires. 00 n 0002 est envoyé en P 0001 = 1977.
- Le programme d'autoperforation n'a jamais besoin d'avoir j et k séparément, mais saus la forme 00 000 j 000 k. Ce mot est envoyé en C 0005 = 1972 par l'ordre n° 67.
- Ce programme a souvent besoin du mot 00 C N. Celui-ci est donc formé par la partie commune et envoyé en C 0006 = 1973.
- Cette partie commune s'achève par l'aiguillage multiple placé en W 0002 = 1992: "aller à G 000P = 1856 + P".

X - E - 2 - P = 0 : sous - programmes à 2 adresses - Opérations en virgule fixe - Tests.

Dès le début do cette partie, le programme relatif aux sous - programmes à 2 adressos se sépare des autres . Pour cela, la machine exécute un test sur la nullité de P' :

131 | G10000 | R AD | PIPPRI | AINN VIFX SIP2A

X-E-2-a) P'=0: sous - programmes à 2 adresses (cf VII-C-2-a)

Le partie de programme propre à PF' = 00 commence à SP2A = 1369 (nº 326).

Si l'ordre d'entrée dans un sous - programme à 2 adresses n'est pas indexé, le programme en langage machine qui doit être obtenu par l'autoperforation est (voir la normalisation, au paragraphe VII - C - 2 - a):

RAG	1:	n+1	65 B A+1	00
AAD	1+1	_A_		20 C N
65	В	A+1		> 4 9 4 -
20	C	N		E he way or
	AAD 65	AAD 1+1 65 B	AAD 1+i A	65 B A+1

Si l'ordre est au contraire indexé, le programme en langage machine qui doit être obtenu est, commo il est indiqué au paragraphe X - A - 2:

n	RAG	n+1	Δ <sub>i</sub> + 28	manufacture.
n+1	60	1	4 <sub>i</sub> + 36	The second second
1	00	000j	000k	-
1+1	65	В	A+1	
1+2	20	C	N	-

Avant que se séparent les parties de programme relatives aux ordres d'entrée dans les sous-programmes à 2 adresses indexés et non indexés, se déroule une partie commune qui forme les mots 65 B A + 1 et 20 C N.

326  S   P2A   F	RIAGICIOOO1   I	0 0 A	0-0		2.7
	1,AG C,0002	00 B A			
	A AG R DOL	65 B A+1		65 0000 0001	
	RG W 0001	oy D Mis		65 B A+1	en W1 (puis 1 si
1 1	3000 D DAI	00	00 C N		j=k=0)
1	AD T DOO		20 C N		
	1 AG C 0005	00 000j 000k			
1 1	1 1 1	00 0000 0001			index ?
333	I NI I NDSA N INSA	A HALL	as é litudu y	ontar lass l'oc	
334 1'ND2A E	a bit w oods	00 000j 000k	20 C N	65 B A+1	
1	3 DE   11 000E	2000 [000 00	20 0 N	the first and the first and the	en W2 (puis 1 + 1)
1 1	DI M 0005		North No. 9		en Wl (puis 1)
	RG W OOOL Agreet	Chair Tables II			en W3 (puis 1+2)
337	RD W 0003	जिल्लामा स्टब्स्ट	ber Spill	20 0 4	on wy (page 1(E)
3 - 1	R'AD C TRL	0.	0 0 1	70. va = 36A	
1	AG 0004	35.55	00 1 0000	P. J.	
1	AG M EMN	0000 n CO		60 0003 6000	1,00015
1 1	ACR GLAG	60 n+1 8003		60 0001 8003	
342	AD D 0052		00 1 Ai	0 — 0 Δ <sub>i</sub>	
343	A JAD M EAX	130m t 63 m	00 1 1 +28	0 0 28	
. 11	E DI 8003	Wild Go. Horts.	ac(0) , si	60 n+1 8003	70 ( )
345   8	S BI P 0002	DOLLINGSON	gwern 🖁		en P2 (puis n)
346	A AD R GOS		60 1 ∆ <sub>1</sub> +36	6000000008	
347 1 19	IRD P 0003 V APF3	1		60 1 ∆ <sub>1</sub> +36	en P3 (puis n+1)
	. 19991		10	and the same of th	

348 N IN2A T RD W 0002 1	100	20 C N (20 C N	en W2 (puis 1 + 1)
349 R AD M EMN	00	00 n 0000	
350 D AG 10002	7	n 000000	
351 A AG C TRL	001	n 00 001	
352 A AD 8001 . 4	00 1	ń 00 1	
353 DIAG 10004 I	00 l n	00 1 0000	
354 A AG R GO1	60 1 n+1	60 0000 0001	
355 TRG PI 0002 C ØDEC		60 l n+1	en P2 (puis n)
356 C ØDEC A AD C 0001	4	00 1 A.	
357 A AD A D10 T DP32	100 200	15 1+1 A 15 0001 0000	1.75
358 T DP32 T RD P 0003 V APF2		15 1+1 A	en P3 (puis n + 1)

- Le mot 65 B A + 1 est envoyé en W 0001 = 1991 par l'ordre n° 329. Pour les ordres non indexés, 65 B A + 1 doit figurer en mot 2 de la 2ème carte (pour être chargé en l). Ce mot est donc bien placé en W 0001. Par contre pour les ordres indexés, 65 B A + 1 doit figurer en mot 3 (pour être chargé en l + 1). Il doit donc être envoyé en W 0002 = 1992, ce que fait l'ordre n° 335.
- Pour former l'adresse  $\Delta_{i}$  + 28, on appelle la mémoire D 0052 = DEL + 62 = 1875 + 62 = 1937 qui contient 00 0000  $\Delta_{i}$  = 00 0000 1807 puis la mémoire MEMX = 1659 (n°439) qui contient 00 0000 0028.
- La mémoire RD01 = 1701 (n°437) contient 65 0000 0001.
- La mémoire TD00 = 1638 (n°424) contient 20 0000 0000.
- La mémoire RG1AG = 1460 (n° 405) contient 60 0001 8003. Cette constante sert à former le mot 60 n + 1 8003. En fait, l'adresse instruction suivante 8003 n'a aucune importance car le mot 60 n + 1 8003 subit ensuite un SBI.
- La mémoire RG08 = 1709 (n° 438) contient 60 0000 0008.
- La mémoire RG0I = 1510 (n° 446) contient 60 0000 0001.
- La mémoire AD10 = 1777 (n° 447) contient 15 0001 0000.

## X - E - 2 - b - Partie commune aux ordres OP' (P' \neq 0)

Cette partie, qui débute en VFX = 1368 (n° 133), ne comporte que la formation du mot i0 000j 000k et de son test pour savoir si l'ordre est indexé ou non.

#### X - E - 2 - c - Opérations en virgule fixe et tests, non indexés :

Cette partie débute en NIND1 = 1480 (n°236).

make a mark farmants of the first

## d-Programmes en langage machine élaborés par l'autoprogrammation :

Passons en revue les programmes en langage machine obtenus à partir de chacun des ordres 01, 02,..., 68.

- Addition (01):

- Soustraction (02)

La seule différence avec l'addition est que le AAG est remplacé par SAG.

- Multiplication sans décalage (03) :

Le test placé en 1 + 1 vérifie que (A) x (B) n'a pas plus de 10 chiffres et qu'il occupe donc seulement l'accumulateur droit. Si ce n'est pas le cas, l'exécution en langage machine s'arrêtera par l'ordre 01 1333 8000 placé en ARETM = 1936. Ce dernier ordre figure à poste fixe dans les paquets verts à joindre aux cartes obtenues par autoperforation.

- Multiplication avec décalage (05) :

Les mots formés sont identiques à ceux relatifs à l'ordre 03 à l'exception des parties soulignées. La différence est donc légère. La lère carte perforée comportera 3 mots en plus du mot 1 qui sera 00 n 0003. En effet, l'ordre 05 occupe 3 mémoires, n, n + I et n + 2. Remarquons que, lors de l'exécution de ces ordres en langage machine, le décalage s'effectuera sur le produit, c'est-à-dire après l'exécution de la multiplication.

- Division sans décalage (04) :

.../..

- Division avec décalage (O6) :

Les mots formés sont identiques à ceux relatifs à l'ordre O4 à l'except ion des parties soulignées. Comme pour l'ordre 05, la lère carte comportera 3 mots en plus du mot 1 qui sera 00 'n 0003. Lors de l'exécution de ces ordres en langage machine, le décalage s'effectuera sur le dividende, c'est-à-dire avant la division.

- Test SI (07):

Si (C) est négatif le programme en langage machine ira en N. Si (C) est positif ou nul, un deuxième test sera fait sur lui : si (C) est différent de zéro, c'est-à-dire positif, le programme ira en B ; si (C) est nul, le programme ira en A. C peut très bien désigner le pupitre (8000).

Remarque importante: en autoprogrammation, c'est N qui (comme pour les autres ordres) désigne l'instruction suivante. Pour obtenir l'autoperforation des branches commençant en A et B il faut en général mettre un ordre de fin de séquence à la fin de la branche N et un autre à la fin de la branche A ou B.

- Test SD (00):

Si la position n° B du contenu de C (C peut très bien désigner le pupitre : C = 8000) est un 3, le programme en langage machine ira en N. Si cette position est un 9, le programme ira en A. Si cette position n'est ni 8 ni 9, le programme s'arrêtera sur l'ordre SDB N A = 9B N A.

Remarque importante: Comme pour l'ordre 07, lors de l'autoprogrammation c'est l'adresse N qui désigne l'instruction suivante. Pour obtenir l'autoperforation de la branche A, il faut en général placer un ordre de fin de séquence à la fin de la branche N.

- Remarque : l'ordre de succession des différentes instructions en langage machine formées a ét é choisi pour que le programme d'autoprogrammation soit le plus simple possible et que ses différentes branches aient le plus possible de parties communes.
- β ) Partie commune aux ordres OP' non indexés.

Cette partie qui débute à NIND 1 = 1480 (n° 236) comporte la formation de l'ordre d'aiguillage "aller à XOOOP' = 1837 + P' " et des mots 00 A n + 1 et 00 B 00 utiles pour la plupart des branches.

.../..

					0	, 00 P'		
236	N INDI	AIAD	CI TE3	1		0 OXOOOP'	o oxoooo	
237		T RD	W 0004	1	To To Harry	7 - 724	0 OXOOOP!	en W4 = 1994
238	M1 /	R AG	C 0001		OOAAAA	0 0		
239	1	DIAD	0002	1	O OAA	AA 0 0		
240	1	A AD	M, EMN		O OAA	A/. n 0000		
241		DIAG	0006		00 A n	00		- 7111811
242		AAG	0101		00 A n+1		0 01	
243		A AD	C 0002	W 0004		00 B 0000		
	(W 10004)	SIØP	10000	X 000P'				Aiguillage

- La mémoire CTE3 = 1833 (n° 414) contient la constante 00 0000 X0000 = 00 0000 1837.
- La mémoire 001 = 1634 (n° 422) contient 00 0000 0001

#### ) P' = 1: addition en virgule fixe (sans index)

Cette partie débute par le seul ordre qui soit différent de ceux de la partie "soustraction", c'est-à-dire la formation de 10 B 6000.

Nous verrons que la branche "soustraction" commence par la formation de 11 B 6000 puis rejoint la branche "addition".

244 246	X'0001 Aldsø2			A DSØ2	00 A n+1	00 B 0000 10 B 0000 10 B 1	10 0	
247		T RD	P10003				10 B 1 en P3 (puis n	+1)
248	1	AAG	R G00	15 1	60 An+1	7	60 0	
249	1	TRG	P 0002			11 EH.	60 A n+1 en P2 (puis n	1)
250		R AG	T, GOO	A, SDIV	210 0	0 0		
251	A SDIV	A AG	C 0006		21 C N		00 C N	
252	1	TIRG	W 0001	V APF1			21 C N en W1 (puis 1	.)

- La mémoire AG00 = 1492 (n° 430) contient 10 0000 0000
- La mémoire RG00 = 1730 (n° 432) contient 60 0000 0000
- La mémoire TG00 = 1588 (n° 419) contient 21,0000 0000

### $\delta$ ) P' = 2: soustraction en virgule fixe (sans index)

Cette partie est identique à la précédente, sauf que c'est 11 B l qui doit être envoyé en P 0003.

Le seul ordre particulier à la soustraction est donc :

- En ADSØ2 = 1797 (n° 246) la branche "soustraction" rejoint celle d'addition
- La mémoire SGCC = 1542 (n° 431) contient 11 0000 0000

#### 2) P' = 3: multiplication sans décalage (sans index)

253	X10003	A AG	R GOO		00 A n+1 60 A n+1	00 B 0000	60 0	
254		TIRG	P 0002	1			60 A n+1	en P2 (puis n)
255		RIAG	8002		00 B 0000	0 0		
256	1	AAG	MULOZ	1	19 B Z	1	19 0000 Z	
257		A AD	C TRL			0 01		
258	. *	A AD	0 01	8 8		00 1+1		4 79 M
259	1	E, DI	8003	RIEUN3			19 B Z	72 - T
260	R EUN3	S BI	P 0003				19 B 1+1	en' P3 (puis n+1)
261	i	R AG	C! TRL		0 01	0 0		* 4 7 - 1
262		A, AG	G NARM		44 ARETM 1		44 ARETM 0000	
263		T RG	W 0002		1	10.1978	44 ARETM 1	en W2 (puis 1+1)
261	1 1 2 3	ALAD	010006			00 G M		

.../...

- Cette partie débute par l'utilisation de CC A n+l placé dans l'accumulateur gauche, puis prépare l'ordre qui sera différent de celui formé pour la multiplication avec décalage.
- En REUN3, comme nous le verrons plus loin, les parties relatives aux ordres 03 et 05 se rejoignent.
- La mémoire MULOZ = 1402 (n° 421) contient la constante 19 0000 Z = 19 0000 1939. Cette constante a été formée pour la multiplication avec index, comme nous le verrons au paragraphe X E 2 d ε -. Dans le cas présent, l'adresse "instruction suivante" est ignorée.
- La mémoire GNARM = 1590 (n° 434) contient 44 ARETM 0000 = 44 1936 0000 (voir X E 2 C 0 ).

#### 7) P' = 5: multiplication avec décalage (sans index).

					1100 A n+1	00 B 0000	24 29 GO	
267	X10005	AIAG	RIGOO I	1	60 A n+1			
268		TIRG	P 0002	ł		-74	60 A n+1 en P2 (puis n)	
269	-1	RIAG	18002	1	00 B 0000	0 0		
270		A,AD	P,0001	1		00 n 0002		
271	,	A AD	0 01	13.5		00 n 0003	0 01	
272	1	TIRD	P!0001	1			00 n 0003 en Pl (mot 1)	

1

273   1	AIAD	10120	11	00 B 0000 00 n+2 0003	00 0002 0000	
274	A AD	CITRL	1	00 n+2 1+3		
275	SAD	0 02	1	00 n+2 1+1	0 02	
276	EIDI	E D2GC		1 .4	69 xxxx GC	THAT DIE
277	SBF	E'D2GC	8001		69 n+2 GC	en ED2GC = 1419
(8001)	E DI	n+2	glo		3x 000m Z	(Z = 1939)
278 GlC	SIBI	P,0004			3x 000m 1+1	en P4 (puis n+2)
279	AJAG	M ULOZ		19 B Z		
280	E DI	18003			19 B Z	
281	D'AD	10004	R EUN3	0000 x x 00 n+2	19 B Z	
260(R EUN3)	SIBI	P 0003			19 B n+2	en P3 (puis n+1)
	11.			0.0		

- Cette branche débute par l'utilisation de CO A n+1. Puis le mot 1 de la lère carte à perforer, qui est CO n CCO3 et non plus CO n CCO2, est envoyé en P CCO1 = 1977. Ensuite le mot CO n+2 1 + 1 est formé dans l'accumulateur droit; il servira à construire 3 instructions. La lère de ces instructions est celle qui va chercher l'ordre de décalage contenu en n + 2. Cet ordre a la forme: 3x CCCO 1939. Par un SBI, on le transforme en 3x CCCO 1+1 et on l'envoie en PCCO4 = 1980 pour qu'il se perfore en mot 4 de la lère carte. Lorsque le programme arrive à REUN3 = 1543 (n° 260), l'instruction 19 B Z est prête dans le distributeur et n + 2 est en position d'adresse instruction suivante dans l'accumulateur droit. L'ordre REUN3 forme ainsi l'ordre 19 B n+2 et l'envoie en PCCO3 = 1979. Le programme se poursuit comme pour la multiplication sans index.

- Les ordres n° 273 et 275 pourraient être réunis en un seul qui ajouterait la constante 00 0001 9998. Mais l'économie d'une ou deux mémoires n'a aucune importance en autoprogrammation.
- La mémoire ED2GC = 1419 (n° 433) contient la constante 69 0002 GC = 69 0002 1700. En fait l'adresse facteur est modifiée mais n'a aucune importance.

#### $\theta$ ) P<sup>1</sup> = 4 : Division sans décalage (sans index) :

282	x!0004	[A AD	DIROO		00	A	n+1			0000	640		0				
283	1	A AD	C TRL	1		9.3	4	64	В	1				5			
284	1	TIRD	P 0003								64	В	1	en :	P3	(puis	n+1)
285	15	A AG	R DOO	- W	65	A	n+1				650		0				
286	1	T RG	P10002	R EUN4							65	A n	+1	en	P2	(puis	n)
287	R EUN4	RAG	T,D00	A SDIV	20		0			. 1							
251	(A SDIV)	A AG	C 0006		20	C	N			e e e e	00	C	N				
252	(1)	TIRG	M 1000J	VAPFI	1			1			20	C	N	en	W2	(puis	1)

- Le programme rejoint les branches relatives à l'addition et à la soustraction en ASDIV = 1745 ( nº 251 ).

### $\lambda$ ) P' = 6 : Division avec décalage (sans index) :

288	X 0006	AAD	D'ROO	n F	00 A n+1	00 B 0000 64 B 0000	640 0	
289	'	A AD	CTRL	į		64 B 1		D. Commission of the
290	1	SIAD	18002	1		0 0	64 B 1	W
291		T DI	P,0003	1			64 B 1	en P3 (puis n+1)
292	'	A AD	P 0001			00 n 0002		
293	1 .	A AD	0 01	1 2		00 n 0003	001	
	1	, ,						/

294		TRD	[P10001	1	00 A n+1	00 n 0003	00	n 000	03	en	Pi	(mot	1)	
295	î.	AAD	0120	1	City City City City City City City City	00 n+2 0003	00	0002	0000					
296		E DI	E DOGD	1-		2 (g lg) g	69	XXXX	.GD					
297	e arm	SBF	E DOGD	1823	.615 the 21	J 3501 5	69	n+2	.GD	en,	EDC	)GD ∓	1354	+
298	1.5	RAD	8003	E DOGD	0 0	00 A n+1					Z			
	(EDOGD)	E DI	n + 2	GID			3x	000m	199	4	-			
299	GD	SIBI	PI 0004	1			3x	000m	n+1	en	P4	(puis	n+2	2)
300	× 1	A, AD	R DOL			65 A n+2	65	0000	0003	1				
301		TRD	P 0002	R EUN4			65	A	1+2	en	<b>P</b> 2	(puis	3 n)	
287	(REUN4)	RAG	T, D00	A, SDIV	20 0	00								
251	(ASDIV)	A AG	C 0006	501	20 C N		00	C :	N					
252	( )	TIRG	MIGGGT	VAPFI			20	C :	N	en	W1	(puis	3 1)	d

- L'ordre n° 294 envoie en P0001 le nouveau mot 1 de la lère carte à perforer, c'est-à-dire 00 n 0003 et non plus 00 n 0002.
- La mémoire ED0GD = 1354 (n° 436) contient 69 0000 GD = 69 0000 1651. En fait l'adresse facteur est modifiée mais n'a aucune importance. L'ordre 69 n+2 GD envoyé dans cette mémoire par l'ordre n° 297 sert à amener dans le distributeur l'ordre de décalage, qui est en n+2.
- La mémoire RD01 = 1701 (n° 437) contient 65 0000 0001.
- Pour la formation de 20 C N, cette branche rejoint celle relative à la division sans décalage à l'instruction REUN4 = 1686 (n° 287).

### M) P' = 7 : Test SI (non indexé):

302	x 10007	A AD	c 0001	00 A n+1	00 B 0000 00 B A		
303 304		A AD	4 <sup>1</sup> 400   W10001   I		44 B A	440 0 44 B A	en WI (puis 1)
							/

305	RIAD	MIEMN	1	10 0	00 n 0000		
306	D! AG	10002			n 00 0000		
307	AIAG	C10003		00.0	n 00 0000		
308	A AD	m emnn	Tro 3	00 0	n 00 N		
309	D AG	10004		00 C n	00 N 0000		
316	AIAU	R GO1		60 C n+1	1,5	60 0000 0001	
311	A AD	C'TRL			00 N J		
312	AJAD	4,600			46 N 1	46 0000 0000	
313	T RD	P 0003	T GP21			46 N l	en P3 (puis n+1)
(T GP21)	TRG	P10002	VAPFI			60 C n+1	en P2 (puis n)

- Cette partie n'utilise pas le mot 00 A n+1 .
- La mémoire 4400 = 1476 (n° 208) contient 44 0000 0000.
- La mémoire RG01 = 1510 (n° 446) contient 60 0000 0001.
- La mémoire 4600 = 1655 (n° 314) contient 46 0000 0000.
- L'ordre TGP21 = 1552 (n° 530) a été créé pour la partie relative à l'ordre de service 03 U V.

#### ) P' = 8 : test SD (non indexé)

Cet ordre est le seul ordre à 2 mémoires qui ne provoque la perforation que d'une seule carte. La partie d'autoprogrammation qui lui est relative se termine par l'envoi à VAPFØ = 1759 (n° 497) qui dirige vers FINPF = 1436 (n° 88) sans passer par PERFØ = 1400 (n° 72) : voirlorganigramme général.

						(		
315	x10008	A <sub>j</sub> AD	MI EMNN		00 A n+1	00 B 0000		B = 000B
316	1 11 11	D AG	,0004		xx nel 0000	OB N OOOO		PV . in the
317		AI AD	C 0001			OB N A		
318	16.1-4	A AD	9 00	1		9B N A	90 0	
319		TIRD	P 0003			0.04	9B N A	on P3 ( puis n+1)
320		R AD	m emn		0 0	00 n 0000		
321	1	D AG	10002	1		n 00 0000		
322		AAG	0,0003		00 0000 C	n 00 0000		
323		D AD	0006	1		00 C n		
324		ALAD	1 -	V APFØ		69 C n+1	69 0000 0001	
325		69	0000	0001		1		
497	VIAPFØ	TRD	P 0002	F'INPF			69 C n+1	en P2 (puis n)

- La mémoire "900"= 1526 (n° 218) contient 90....0.

# X-E-2-d-) Opérations en virgule fixe et tests, indexés. Cette partie débute en IND1 = 1479 (n° 136).

# Programmes en langage machine élaborés par l'autoprogrammation. Ces programmes, à l'exception de ceux relatifs aux ordres 07 et 08, constituent tous des entrées dans le sous-programme d'indexage commençant

à  $\Delta_i = 1807$  (voir X - A - 1). Ils comportent tous :

en n : RAD n + 1  $\Delta_i$ en n + 1 : 60 1  $\Delta_i$  + 13 en 1 : io 000j 000k

> en 1+1 : l'ordre à indexer par k en 1+2 : l'ordre à indexer par j

en 1+3 : l'ordre à indexer par i

.../...

A la sortie du sous-programme d'indexage, c'est l'ordre qui était placé en 1, 4, 3 qui, indexé, sera exécuté le ler; l'ordre placé en 1 + 2, indexé, se trouve alors en W 0001 = 1991 et l'ordre en 1 + 1 se trouve alors en W 0002 = 1992.

#### - Addition (01):

n	RAD	n+1	Δi	İ		
n+1	60	1	A i + 13			
1	iO	000j	000k			
1+1	TRG	C	N	isa	en	W0002
1+2	AAG	В	W0002	ira	en	W0001
1+3	RAG	A	W0001			

#### - Soustraction (02):

	Ordr	es com	muns	en n, n+1,	, 1	
29	1+1	TRG	C	N	ira e	n W0002
	1+2	SAG	В	W0002	ira e	n W0001
	1+3	RAG	A	W0001		

## - Multiplication sans décalage (03) :

Ordr	es con	mun	s en n,	n+1,	1		
1+1	TRD	C	N	1	ira	en	W0002
1+2	MUL	В	Z		ira	en	W0001
1+3	RAG	A	W0001				

En Z = 1939 se trouve à poste fixe pour l'exécution en langage machine l'ordre de test : GNN ARETM W 0002 = 44 1936 1992. L'ordre des instructions en langage machine exécutées à la sortie du sous-programme d'indexage est donc :

	RAG	A'	W0001	
WOOOl	MUL	B1	Z	
Z	GNN	ARETM	W0002	
W0002	TRD	C1	N	

#### - Multiplication avec décalage (05) :

Ordres	communs	en 'n,	n+1, 1	A 45 mg
1+1	TRD	C	N	ira en W0002
1+2	MUL	В	n + 2	ira en W0001
1+3	RAG	A	W0001	
n+2	3 x	000m	Z	Z = 1939

Les mots formés sont identiques à ceux relatifs à l'ordre 03 à l'exception des parties soulignées.

L'ordre d'exécution des instructions à la sortie du sous-programme d'indexage sera :

	RAG	A'	W0001	100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg 100 mg
WOOOl	MUL	B'	n + 2	
n + 2	3x	000m	Z	Décalage sur le produit
Z	GNN	ARETM	W0002	Z = 1939
M0003	men	01	N	

.../..

#### - Division sans décalage (04) :

Ordr	es com	muns (	en n, n+1,	1	
1+1	TRD	C	'- N	ira	en W0002
1+2	DRR	В	W0002	ira	en W0001
1+3	RAD	A	M0001.		

#### - Division avec décalage (06)

Ordres communs en n, n+1, 1									
	1+1	TRD	C	N	ira en W0002				
	1+2	DRR	В	W0002	ira en W0001				
	1+3	RAD	A	n + 2					
	n+2	3 x	000m	W0001	W0001 = 199				

Les mots formés sont identiques à ceux relatifs à l'ordre 04, à l'exception des parties soulignées.

L'ordre d'exécution des instructions à la sortie du sous-programme d'indexage seration des instructions à la sortie du sous-programme d'inde-

#### - Test SI (07):

L'indexage du test SI ne se fait pas par le sous-programme d'indexage. Pour cet ordre en effet; les adresses A, B et C, dans les instructions en langage machine à exécuter, ont une disposition qui n'est compatible avec aucune des 3 parties du sous-programme d'indexage existant. Il faudrait donc probablement créer une 4ème partie pour le sous-programme d'indexage.

Il a été décidé que, pour simplifier l'indexage, seule l'adresse C serait indexable; ainsi le programme d'indexage de l'ordre 07 est assez simple pour être obtenu par autoperforation. Les cartes perforées sont:

n	RAG	CTRk	n+1	0	. 0 (	CTRk)	0 0		į	1
n + 1	DAG	0004	1	00 (	CTRk)	0000				
1	AAG	1+1	8003	60	C t	1 + 2	174	60 C	1+2	
1 + 1	60	C	1 + 2	d.						
(8003)	RAG	C†	1 + 2		(C)		0 0	LOAK_III		
1 + 2	ANG	N	1 + 3							
1+3	GNN	В	A					74 11	- 1	

Voir la remarque relative au test SI au paragraphe X - E - 2 - C -  $\alpha$  -

# - Test SD (08):

Le cas de ce test est analogue à celui du test SI. C'est également l'autoprogrammation qui fournit le programme d'indexage, simplifié par le fait que seul C est indexable.

Les cartes obtenues par l'autoperforation sont :

n	RAG	CTRk	n + 1	0 0 (CTRk) 0 0	i
				00 (CTRk) 0000	
1	AAG	1 + 1	8003	69 C' 1 + 2	
		C			37.04
(8003)	EDI	C1 -	1 + 2	SCHOOL STATE AND DEAD	(c')
11+2					

Voir la remarque relative au test SD au paragraphe X - E - 2 - C -  $\propto$  -

B) Partie commune aux ordres 0P' indexés :

Cette partie débute à l'instruction IND1 = 1479 (n° 136). Elle est en grande partie inutile pour les ordres 07 et 08, mais n'est pas nuisible car ellen'allonge qu'extrêmement peu, ou pas, l'autoprogramm tion de ces ordres. Elle forme les 3 instructions communes aux ordres 01,02,... 06. De plus, elle construit l'aiguillage "aller à F000P" = 1846 + P'".

				10000j000k	0 0P!	3.00	
136	I! ND1	A' AD	C)TE2	1 1	0 0 FOOOP'	0 OFOOOO	A Sun
137	. 1	TI RD	W 0004			0 OFOOOP'	en W4 = 1994
138		T RG	W,0001	Alas M	Type SV	10000j000k	en Wi (puis 1)
139	'	R, AD	CTRL	0 0	0 01	0,400	
140		D AG	10004		00 1 0000	74.	27 - 10 S 1 W
141		ALAD	D 0052		00 1 A;	ο ο Δ,	
142	ļ	A AG	18001	0 οΔ.			
143		AIAG	M EMN	, 00 n A,			
144	1	A AG	R'DlO	65 n+1 Δ <sub>i</sub>		65 0001 0000	
145	Stare.	TIRG	P 0002			65 n+1 ∆ ,	en P2 (puis n)
146	i Aire	A AD	6 013	the fillip let to its	60 1 <b>4</b> ,+13	60 0000 0013	
147		TIRD	P,0003	T 41 27 West		60 1 A; + 13	en P3 (puis n+1)
148		R AD	0 0002	W 0004 b 0	00 B 0000		
	(W 10004)	SIØP	10000	F OOOP		Chair -	

- C'est en D0052 = DEL + 62 = 1875 + 62 = 1937 que se trouve; pendant l'autoprogrammation, le facteur de translation du sous-programme d'indexage : △; = 1807 (voir X - A):

S Page Nay Princip

- La mémoire CTE2 = 1532 (n°413) contient 00 0000 F 0000 = 00 0000 1846.
- La mémoire RD10 = 1560 (n° 415) contient 65 0001 0000.
- La mémoire 6013 = 1484 (n° 416) contient 60 0000 0013.

#### ) P' = 1 : Addition en virgule fixe (avec index) :

Cette partie débute par le seul ordre qui soit différent de ceux de la partie "soustraction", c'est-à-dire la formation de 10 B W0002.

		14	111111111111111111111111111111111111111		1 - 4 6 6 7 7		14	
					00	00 B 0000		
-49	F 0001	AAD	A GOW2	AIDSØ1	111	10 B W2 .	10 0000 W2	A PRESTOR OF
.51	A DSØ1	T RD	W 0003	1,			10 B W2	en W3 (puis 1 + 2)
.52		R'AD	C,0001		0 0	0 OA	4., 9	
.53		DIAG	10004	1		0000 A 00		la i
.54		A AD	R GOW1		- 1	60 A W1	60 0000 W1	
55		T RD	P 0005				60 A W1	en P5 (puis 1 + 3)
56		A,AG	T, G00	A SMUl	210 0		210 0	
57	A'SMUl	A AG	C 0006		21 C N		00 C N	7 50 7
58		TRG	W10002	VIAPF4			21 C N	en W2 (puis 1 + 1)

- La mémoire AGOW2 = 1450 (n° 417) contient 10 0000 W0002 = 10 0000 1992.
- La mémoire RG0W1 = 1536 (n° 420) contient 60 0000 W0001 = 60 0000 1991.
- La mémoire TGOO = 1588 (n° 419) contient 21 0000 0000.

# S) P' = 2 : Soustraction en virgule fixe (avec index).

Cette partie est identique à la précédente, à part la lère instruction. La seule instruction particulière est :

- C'est 11 B W0002 au lieu de 10 B W0002 qui est envoyé en W0003 = 1993 par l'ordre ADSØ1 = 1455 (n° 151).
- La mémoire SG0W2 = 1401 (n° 418) contient 11 0000 W0002 = 11 0000 1992.

.../..

#### (avec index) P' = 3: multiplication sans décalage (avec index)

159 F10003 A AD M'ULOZ   0 0 00 B 0000   19 B Z   19 0000 Z	or all
159 F 10003 A A D M ULOZ   19 B Z   19 0000 Z	100
160 TRD W 0003 R EUNI 19 B Z en W 3 (puis	1 + 2)
161 R EUNL R AG C 0001 1 0 0 A 0 0	
162   D.AG 0004 00 A 0000	
163 A,AG R'GOW1 M'UDI1 60 A W 1 60 0000 W1	
164 M UDI1 T RG P 0005 60 A W1 en P 5 (puis	1 + 3)
165 RIAG T'DOO AISMUL 200 0 0	
157 (A SMU1) A AG C 0006 20 C N 00 C N	
158 (   ) TIRG W   0002 V   APF4   20 C N   en W 2 (puis	1 + 1)

- Cette partie commence par la formation de l'instruction 19 B Z = 19 B 1939 qui n'est pas utile pour l'ordre de multiplication avec décalage. A partir de l'instruction REUN l = 1596 (n°161), le programme est valable à la fois pour les 2 ordres de multiplication,03 et 05. A partir de MUDI l = 1691 (n°164), le programme est valable à la fois pour les 2 ordres de multiplication et les 2 ordres de division. A partir de A3MU l = 1593 (n° 157), ce programme rejoint celui relatff à l'addition et la soustraction. Ces recoupements sont aisés parce que tous les ordres d'opérations en virgule fixe indexés provoquent la perforation d'une 2ème carte ayant le même nombre q = 4 de mots en plus du mot l (ces mots se chargeront en l, l + 1, l + 2, L+ 3).
- La mémoire MULOZ = 1402 (n° 421) contient 19 0000 Z = 19 0000 1939. (voir X - E - 2 - d - of )

#### 7) P' = 5: multiplication avec décalage (avec index)

					0 0	00 B 0000	1 2 11 2	
66	F10005	AJAD	MULOZ		7.01	19 B Z	19 0000 Z	
67	W.L.	T RD	W 10003				19 B Z	en W3
68		R.AD	P 0001	1		00 n 0002		
69		A AD	0 01	1	3,	00 n 0003	0 01	
70	E 10 T 1	T RD	P 0001				00 n 0003	en P1 (mot 1)
71	ı	D AD	0004			0 On	101077	
72	*	A AD	0 02	1 L		0 On+2	0 02	
73		E DI	W 0003		2,1-20,1	41	19 B Z	AND A
74		SBI	W 0003				19 B n+2	en W3(puis 1+2)
75		D AG	10004			00 n+2 000	0)	
76	1	A,AD	E DOGA	8002		69 n+2 GA	69 0000 GA	
	(8002)	E DI	n+2	GA	E IN E		3x 000m Z	Z = 1939
77	GIA	TIDI	P10004	REUNA		70 , 1	3x 000m Z	en P4 (puis n+2)

- Ce programme commence par utiliser B qui se trouve dans l'accumulateur droit pour former l'ordre 19 B Z qui sera transformé plus loin. Ensuite, le nouveau mot 1 de la lère carte à perforer est formé : 00 n 0003. Le n qui se trouve en 8002 est utilisé pour former l'ordre 19 E n + 2 ainsi que l'ordre 69 n + 2 GA destiné à aller chercher l'ordre de décalage pour l'envoyer en P 0004 = 1980 (afin qu'il soit, pour l'exécution en langage machine, placé en n + 2). En REUN 1 = 1596 (n° 161), ce programme rejoint la branche relative à l'ordre de multiplication sans décalage, 03.
- La mémoire EDOGA = 1610 (n° 423) contient 69 0000 1500.

 $\theta$ ) P' = 4: division sans décalage (avec index)

178	F 10004	AIAD	D <sup>i</sup> ROW2		0 0	00 B 0000 64 B W2	64 0000 W2	
179		T RD	W, 0003	1			64 B W2	en W3 (puis 1+2)
180		R AG	C 0001		O OA	0 0		
181	7	D'AG	10004	,	00 A 0000			
182	1	AAG	0,0W1	RIEUN2	00 A W1		0 0 W1	
183					65 A W1	-8	650 0	1 1 1 1 1
164	(M NODIL)	TIRG	P10005	1: 1	Tel 10 1 800		65 A W1	en P5 (puis 1+3)
	:		100	;				

- En REUN2 = 1791 (n° 183) les programmes relatifs aux divisions 04 et 06 se rejoignent.
- En MUDII = 1691 (n° 164), ce programme rejoint celui relatif aux 2 ordres de multiplication, pour la formation de 20 C N et son envoi en W 0002.
- La mémoire DROW2 = 1453 (n° 425) contient 64 0000 W0002 = 64 0000 1992
- La mémoire RD00 = 1794 (n° 429) contient 65 0000 0000.

## $\lambda$ ) P' = 6: division avec décalage (avec index)

	00	00 B 0000		
184 F10006 A AD D ROW2		64 B W2	64 0000 W2	
185 TRD W 0003	LA NEL CALL	1 1 1 1 1 1 1 1	64 B W2	en W3 (puis 1 + 2)
186 RIAD P10001	00	00 n 0002		
187 A AD 0 01		00 n 0003	0 01	W 1 L
188 T RD P 0001	The second second	r out	00 n 0003	en Pi (mot 1)
189 1 A AD 0 20 1		00 n+2 0003		

../...

19	0 1	EIDI EI DOGB	11	100	00 n+2 0003	69 xxxx GB	
19	1	S BF E DOGB	18001			69 n+2 GB	en $EDOGB = 1392$
	(8001)	EDI n+2	G, B			3x 000m W1	Wl = 1991
19	2 G B	T'DI P 0004	ž.	75.7	1 min =	3x 000m W1	en P4 (puis n+2)
19	3	DIAD 0004	1		00 10+2		
19	4	D AG 0006	14		n+2 000000		
19	5 1	A AG C10001	.1	00 A			
19	6	D AG 0004	R EUN2	00 A n+2	00		
18	3 (R EUN2)	A AG R DOO	M. UDII	65 A n+2		6500	7
16	4 (MIUDIL)	T   RG P   0005	1:		1	65 A n+2	en P5 (puis 1+3)

- La lère instruction formée est 64 B W CCO2 = 64 B 1992. Cette instruction pourrait être formée dans une partie commune aux 2 ordres de division, mais il est plus simple d'utiliser dès le début le mot 00 B CCCO qui est dans l'accumulateur. C'est ensuite le nouveau mot 1 de la lère carte qui est formé, puis l'ordre 69 n + 2 GB = 69 n + 2 1550 allant chercher l'ordre de décalage 3x 000m 1991. L'adresse n + 2 sert aussi à former le mot 65 A n + 2. En REUN2 = 1791 (n° 103) ce programme rejoint celui relatif à la division sans décalage.
- La mémoire EDOGB = 1392 (n° 420) contient 69 0000 GB = 69 0000 1550.

  En fait, l'adresse facteur est modifiée mais n'a aucune importance.

P1 = 7: test SI (avec index)

1 /					
	January ji Sakat S	10	00 B 0000	Attendaria	
197 F 0007	A AD C 0001		OO B A	a de la	
198	A AD 4 400 1	1	44 B A	4400	
199	T RD P 0005		S 10 40 4"	44 B A	en P5 (puis 1+3)
200	R AD M EMNN	00	0O N		
201	A, AG C, 0003	00 C	10041 22		
202	D AG 0004 ,	00 C 0000	00 N 0000	postráne i i	
203	Alag Rigoo	60 C 0000	(T)	600	
204	A AD C TRL ,		00 N 1		1 1 1, 1
205	A AD 41603		46 N 1+3	46 0000 0003	
206	E DI F 7F8	1		20 W1 VAPF4	- 12.7
207	120 W 0001 V APF4	1		- AG	
219 F 7F8	T DI W 0005	1		20 Wl VAPF4	en W5 = 1995
220	T RD W 0003	ii .		46 N 1+3	en W3 (puis 1+2)
221	A AG C TRL	60 C 1			A1 46
222 1	A,AG 0,02	60 C 1+2	211 20 1	C 44 PF	2
223	T RG W 0002			60 C 1+2	en W2 (puis 1+1)
224	R AD M ENN	0	00 n 0000	10 61 67	2 1 1
225	D AG 0002		n 00 0000		
226	A AG C 0005	00 k	n 00		. (4)
227	A AD C TRL	00 k	n 00 1		or the William Record
228	E DI D AG4	ii .		35 0004 xxxx	
229	S'BI P'0003 D'AG4		e de cid	35 0004 1	en P3 (puis n+1)
230 D AG4	D AG 0004	00 k n	00 1 0000	- u-	
231	A AG RIGCOL	60 CTRk n+1		60 C0000 0001	= 60 1967 0001
232	T RG P 0002			60 CTRk n+1	en P2 (puis n)
233	A AD   W 0005	1 200	10 1+1 8003		
234	10 0001 8003				, , , , ,
(w 10005	TIRD WIOOOL VIAPF4	1	age a si	10 1+1 8003	en W1 (puis 1)

.../..

- La lère partie de ce programme (jusqu'à F7F8) comporte la construction des mots qui sont utiles pour l'ordre 07 mais pas pour l'ordre 08.
- La 25me partie (à partir de F7F3) est commune aux ordres 07 et 00. Elle envoie à leur place certains des mots construits par la 15re partie et fabrique les mots utiles à la fois pour les deux tests (SI et SD). Elle est construite en forme de sous-programme : il faut placer une instruction de sortie dans le distributeur avant d'arriver en F7F3 = 1503 (n° 219).
- Il vaut mieux faire exécuter le plus de choses possibles par la partie commune.

  C'est pour cela que 46 N 1+3 est encore dans l'accumulateur lorsqu'on

  arrive en F7F3 et que le mot 60 C 0000, est complété par cette partie

  commune.
- L'ordre F7F8 = 1503 (n° 219) envoie l'instruction de sortie en W 0005 = 1995
- k est placé en C 9905 = 1972 par la partie commune aux ordres à 2 mémoires (ordre n° 67).
- Pour former le mot 35 0004 l par un SBI, la machine a besoin d'entrer dans le distributeur (instruction n° 228) n'importe quel mot commençant par 35 0004, par exemple le mot contenu en DAG 4 n'est pas modifiée.
- La mémoire "4400" = 1476 (n° 203) contient 44 0000 0000.
- La mémoire 4603' = 1442 (n° 209) contient 46 0000 0003.
- La mémoire RGCO1 = 1406 (n° 235 contient 60 00000 0001 = 60 1967 0001.

  C 0000 = 1967 est l'adresse du registre d'index n° 0 : CTRO.

#### $\nu$ ) P' = 8 : test SD (avec index)

			10 0	00 B 0000		
210 F 0008	A AD M EMNN	1	6 m	00 000B N		
211	A AG C 0003		0oc			
212	D AG 10004	!	00 C 0000	OB N 0000		07
213	A AG 6 9		69 C 0000		6900	
214	A AD C 0001	ŧ		OB N A	0.15	
215	A AD 9 00			9B N A	90 0	
216	EIDI I	F 7F8			20 W1 VAPF3	
217	20 M 0001	V APF3		Ø 1060		
219 (F17F8)	TIDI WIOOO5	1			20 W1 VAPF3	en W5 = 1995
220 (, )	T RD W 0003	3	ON N T		9B N A	en W3 (puis 1 + 2)
221 ( )	A AG CITRL	1	69 C 1	. 100	5	
222 (' )	A AG 0 02	10	69 C 1+2		_ =	
223 (1 )	T  RG   W   0002	1.			69 C 1+2	en W2 (puis 1 + 1)
	: :					
		•	301 6 34			

- La mémoire 69 = 1594 (n° 407) contient 69 0000 0000
- La mémoire 900 = 1526 (n° 213) contient 900000 0000
- Ce programme rejoint celui relatif au test SI en F7F0 = 1503 (n° 219). Voir le paragraphe ci-dessus, relatif à l'ordre 07.
- L'ordre de sortie envoyé en W 0005 = 1995 est ici 20 W 0001 VAPF3 et non 20 W0001 VAPF4 comme pour l'ordre 07. En effet la 2ème carte à perforer ne doit porter que q = 3 mots en plus du mot 1.

# X - E - 3) P = 1, 2, 3, 4, 5: sous-programmes à 3 adresses.

Le facteur de translation  $\Delta$  de chaque sous-programme à 3 adresses doit pp' figurer dans la mémoire DEL + PP' = 1375 + PP' = D 00PP' - 0010 au moment de l'autoprogrammation. Ainsi, pendant l'autoprogrammation, les facteurs de translations doivent être enregistrés mais pas les sous-programmes, tandis qu'en exécution en langage machine, les sous-programmes doivent être enregistrés mais pas leurs facteurs de translation.

Pour les ordres d'entrée dans les sous-programmes à 3 adresses non indexés, l'autoprogrammation fournit le programme suivant (voir la normalisation au paragraphe VII - C - 3):

n	RAD	1	n+1	00	00 A B	5 0	
n + 1	EDI	1+1	Δ <sub>pp</sub> ,	00		20 C N	
1 1 + 1	00	A	В				
1+1	20	С	N			r Washington	

Pour ces mêmes ordres indexés, comme il est noté au paragraphe X - A - 3, l'autoprogrammation fournit le programme :

21	1.4	10.1	A 100 TO 1	
n	RAD	n+1	$\Delta_{i} + 48$	
n + 4	60	1,0	Δ <sub>pp</sub> ,	
1	in,	(000 j	.000k	
1+1	00	Α	В	
1+2	20	C	N	

Comme en logique extérieure, les branches relatives à P = 1, 2, 3, 4, 5 se rejoignent aussitôt. Les mémoires G 0001 = 1057, G 0002 = 1858,....,
G 0005 = 1061 contiennent le même mot; ce mot est la lère instruction de la partie propre aux sous-programmes à 3 adresses. Cette partie débute par un morceau de programme commun aux ordres indexés et non indexés.

359 G 0001	R AD C 0002	SIUITG	41			
360 G 0002	R AD C 0002	S UITG				7
		4,(	7 11			
11:		4 5 EE				
363 G 0005	R,AD C,0002	S'UITG	00	00 B 0000 .		
364 S UITG	D AG 0002			B 00 0000		
365	A1AG C10001		00 A			
366	D AG 0004		00 A B	00		
367	TIRG WIOOOL	,			00 A B	en W1 (puis 1
368	R AD P PPRI		00	0 OPP'		si i = j = k = 0)
369	D'AG 0004	-		00 00PP'0000		
370	A AD E DDEL			69 DOOPP' HA	69 D0000 HA	
371	S'AD O'DIXO	8002		69 DEL+PP'HA	00 0010 0000	
8002	E DI DEL+PP	H; A			00 App'	
372 HIA	T DI W 0004	4	F 7.		0ο Δ <sub>pp</sub> ,	en W4
373	R AD C 0006	1	00	00 C N	PP	
374	A AD T DOO	E - 1		20 C N	200	5 - 5 - 6 - 6 - 6 - 6 - 6 - 6 - 6 - 6 -
375	A AG C 0004	1	i00			
376	A AG C 0005		i0000j000k		00000j000k	
377	G NN I ND3A	N'IN3A				index ?

.../...

Marketon Company of the Company of t

A10113 TO 10		West and the second	3 V		
378 IIND3A	E DI WIOOO1	i0 000j 000k	20 C N	00 A B	
379	T DI W 0002		and the state of	00 A B	en W2 (puis 1 + 1)
380	T RG W  0001			i0 <b>0</b> 00j 000k	en W1 (puis 1)
381	T RD W 0003 1		572 J C	20 C N	en W3 (puis 1 + 2)
382	RIAG C TRL	00 1	00		
383	D AG 0004	00 1 0000			
384	A1AG W 0004	00 1 Δ <sub>pp</sub>		00 ∆ <sub>pp</sub> ,	
385	A AG R GOO	60 1 Δ <sub>pp</sub> ,		600	
386	A   AD D 0052	PP	00 A <sub>i</sub>		
387	A AD M EMY		00 ∆,+48	0048	4.1
588	A JAD M EMN		00 n A <sub>1</sub> +48		
589	A AD R DIO		65 n+1 ∆ <sub>1</sub> +48	65 0001 0000	
1 1	T RD P 0002 T GP33			65 n+1 ∆ <sub>i</sub> +48	en P2 (puis n)
391 T GP33	T'RG P'0003 V'APF3		a Putit	60 1 ∆ <sub>pp</sub> ,	en P3 (puis n + 1)
				P.P	
192 N IN3A	T'RD W'0002	0	50 C N	20 C N	en W2 (puis 1 + 1)
1	R AD M EMN	0	00 n 0000		
	D AG 10002		n 00 0000		36
95	A AG C TRL	00 1	11	00 1	
96	AIAD 1800L		n 00 1		r 1
.97	D AG 0004	00 l n	00 1 0000	-	Land of
98	A!AG R!DOL I	65 l n+1		65 0000 0001	
99	T RG P 0002			65 l n+1	en P2 (puis n)
00	Alad W10004	- 34	00 l App'	oο Δ pp'	
	A AD E DlO T DP32		69 1+1 Δ <sub>pp</sub> ,	69 0001 0000	milder to the
58 (T   DP32)	TIRD P 0003 VIAPF2	3		69 1+1 Δ <sub>pp</sub> ,	en P3 (puis n + 1)

- La partie commune forme le mot 00 A B qu'elle envoie en réserve en W 0001 = 1991. Pour les ordres non indexés, ce mot restera en W 0001 où il est bien placé; pour les ordres indexés, il sera envoyé en W 0002. La partie commune a aussi pour rôle d'aller chercher le facteur de translation du sous-programme. Pour cela, elle forme l'ordre 69 DEL + PP' HA = 69 (D00PP' - 0010) HA, puis l'exécute. La mémoire EDDEL = 1730 (n° 448) contient 69 D0000 H A = 69 1885 1656. Le facteur de translation : Appinest envoyé en W 0004 = 1994 pour être mis en réserve. Ensuite, la partie commune forme 20 C N puis i0 000j 000k qui est testé pour savoir si l'ordre est indexé. S'il est indexé, le programme est aiguillé en IND3A = 1806 (n° 378) Pour former l'adresse  $\Delta_i$  + 48, la machine va chercher  $\Delta_i$  en D 0052 = DEL + 62 = 1875 + 62 = 1937 (voir X - A) et lui ajoute 00 0000 0048 placé dans la mémoire MEMY = 1414 (n° 440). Si l'ordre n'est pas indexé, le programme est aiguillé en NIN3/. = 1660 (n° 392). L'ordre TDP 32 = 1805 (n° 358) qui termine cette branche est utilisé sussi pour les ordres indexés d'entrée dans les sous - programmes à 2 adresses.

#### X - E - 4) P = 9: bouclage:

Le rôle de l'autoprogrammation pour un ordre de bouclage est de construire des instructions qui, lors de l'exécution en langage machine du programme, permettront d'entrer dans le sous-programme de bouclage. Après la perforation de ces cartes, l'instruction suivante est prise en N.

Le sous-programme de bouclage figure à poste fixe dans les paquets verts à joindre aux cartes autoperforées. Son facteur de translation a été fixé à 1751 pour l'exécution en langage machine. Pendant l'autoprogrammation, ce facteur est placé sous forme 00 0000  $\Delta_{\rm bou}$  dans la mémoire D 0051 = DEL + 61 = 1375 + 61 = 1936.

# a) Instructions produites par l'autoprogrammation :

State Strike

				1000
	n	RAG	ì	1+1
	n + 1	,00	0000	H
	1	69	1 + 1	Δ <sub>boα</sub>
9	1+1	EDI	n + 1	Abou
	1 + 2	00	C	N
	1+3	Oak	CTRi	! B !

Ce morceau de programme était primitivement plus simple. Il était de la forme :

Mais cette forme ne se prêtait pas à la modification de bouclage. En effet pour que H puisse être modifié par un ordre de modification de bouclage, il faut que cet ordre puisse "savoir" où est H. Or, en programmant, on ne peut pas connaître la valeur de l qui sera attribuée à l'ordre de bouclage à modifier. Il fallait donc que H soit en n ou en n + 1. Mais n doit contenir le ler ordre à exécuter pour le bouclage. Donc H ne pouvait être mis

qu'en n + 1. C'est ainsi que la lère forme de programme a été adoptée.

#### b) Autoprogrammation de l'ordre de bouclage 9i N & H OB OC

				1000		AUG 1 3	0 - 10 G
108	G 1000	9 RIAGICITEL	1	0 01	00	001	
109	1	D AG 0004		00 1 0000	1.0		
110		A AG 8001		00 1 1	-		2 - 1 × - 1 × 1
111	1	A AG R GOL		60 1 1+1		60 0000 0001	
112		T RG P 0002				60 1 1+1	en P2 (puis n)
113	1	A AD M EMN		27 90	00 n 0000	10000E 10	20
114		A AD D 0051	.9		00 n Abou	00 Abou	
115	40	A AD E DIO			69 m+1 Abou	69 0001 0000	
116	٠.	T RD W 0002	or a	TOW TO	0.00	69 m+1 A bou	en W2 (puis 1 + 4)
117		D AD 0006		000000xxxx	xx 1+1 xxxx	500	
118	•	SIBF WIOOOL	'	, N . T	1000	69 1+1 ∆ <sub>bou</sub>	en Wl (puis 1)
119	v.	E DI C 0006			80	00 C N .	
120		T DI W 0003	$\mathbf{P}'$	- 15 m		00 C N	en W3 (puis 1 + 2)
121		R AG P PPRI		009i	0		
122	1	D AD 0002		00	9i 00	LI CAR LA	ia-11
123		AIAD C10002	Mual		9i B 0000-	00 B 0000	
124	1.	D AD 0003			0009i B 0	en er ist ti	11 to 11 1
125		A AD C 0004	,		d 0091 B 0	ď0	
126	light.	D AD 0001			0 0 009i B	2.400 300	
127		A AD M CTRO			O of CTRi B	00 1877 0000	
128	J	T RD P 0005	L.	-	3"	O d CTRi B	en P5 (puis 1 + 3)
129	,	E DI C 0001		L	HC	ОО Н	
130	ş	TIDI P10003	V APF4			оо н	en P3 (puis n + 1)

.....

- Comme il est indiqué au paragraphe X E 4, la mémoire D CO51 = DEL ÷ 61 = 1875 + 61 = 1936 contient 0C 0CCC Δ tou = CO 0C00 1751.
- Pour former l'adresse CTRi, on a ajouté à PP' = 9i la constante CTR1 90 = 1967 90 = 1877. La mémoire MCTRO = 1486 (n° 406) contient 00 1877 0000.
- Remarque: dans l'assemblage de l'autoprogrammation, avant d'avoir sa forme actuelle (ci-dessus) la partie relative au bouclage était différente. Or on a omis d'enlever l'ancienne forme que représentaient les instructions n° 89 à 107. Ceci n'a aucune importance d'ailleurs car l'encombrement de l'autoprogrammation ne compte pas. Dans la liste figurant à la fin de cette brochure, les cartes inutiles ont été supprimées.
- L'organigramme de bouclage pour l'exécution en langage machine.

  L'organigramme de ce sous-programme de bouclage est semblable à celui du sous-programme utilisé en logique extérieure, à l'exception de deux points :

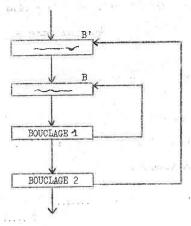
  1) pour les ordres de bouclage tels que α = 0, après le ler passage la phase de formation des ordres est supprimée. Ceci donne un peu plus d'encombrement au sous-programme mais provoque un gros gain de temps.

  La suppression de la phase de formation des ordres se fait en remplaçant le ler ordre du sous-programme de bouclage (placé en Δ<sub>bou</sub>+ 0000 = 1751) par l'ordre "aller à BØU2 = Δ <sub>bou</sub>+ 35 = 1751 + 35 = 1706" (voir l'organigramme du paragraphe VII C 4).

.../...

Seuls peuvent être dotés de de les bouclages ne comportant pas de bouclage intérieur.

Contre-exemple: considérons 2 bouclages intérieurs l'un à l'autre dont le plus extérieur soit affecté de d = 0.



Bouclage 1: 9i N 1 H OB OC

Bouclage 2: 9i' N' OH' OB' OC'

La lère fois que l'on passe par le sous-programme de bouclage, c'est pour effectuer le bouclage l. Celui-ci se déroule normalement, sans suppression de la formation des ordres. Quand les H tours du bouclage l ont été effectués, le sous-programme de bouclage est utilisé pour le ler passage par l'ordre de bouclage 2. Après ce ler passage, la phase de formation des ordres est détruite. Lorsqu'on repasse au bouclage 1, le sous-programme de bouclage (ne formant plus les ordres nécessaires) utilise les ordres formés précédemment, c'est-à-dire ceux construits

pour le bouclage 2. Cela n'est pas correct et le bouclage 1 ne fonctionne pas comme il devrait le faire.

2) pour les bouclages tels que  $\alpha=0$ , à la sortie de boucle (après la remise à zéro du compteur de boucles n° 0 et du registre d'index n° i). le mémoire  $\Delta_{bou} + 0000 = 1751 \text{ est régénérée} ; la phase de formation des ordres est donc rétablie.}$ 

Remarque: l'ordre - 02 U V fait aussi cette régénération.

Voici le sous-programme de bouclage destiné à l'exécution en langage machine. Ce programme a été écrit en langage machine. Chaque ordre est suivi de ses indicatifs de translation.

n	60	1	1 + 1	1		69 1+1 △	0	0	1	$\Delta \doteq \Delta_{\text{bou}}$
1+1	69	n + 3	Δ					Ψĸ	0 о н	bou
0000	24	0055	0001	8	8				оон	en $\Delta$ + 55
1	11	0002	8003	8	9	15 1+2 ∆+5				
2	53	9998	9995	9	9				- M &	19. 19
(8003)	15	1+2	0005		8	G 08	00 C	N		3570. 8
5	69	0042	0006	8	8				24 <b>A</b> xxxx	
6	23	0042	0003	8	8				24 A N	en <u>A</u> + 42
3	20	0004	0007	8	8		10-		00 C N	en /+ 4
7	10	0008	8003	8	9	65 1+3 <b>∆</b> +9				OII 2114
8	50	0001	0004	9	9	-24 923			ा गांवको ⇒ार-देते	
(8003)	65	1+3	0009		8	0 0	Od CTRi	В		
9	69	8003	,0010	9	8	V	4	-	0 0	THE STATE OF THE S
10	22	1991	0011	9	8			A	00 CTR1 0000	en W 0001

0011	10	8001	0012	918	00 CTRi	0000	Oc CTRi I	В	00 CTRi 0000	
12	11	0013	0014	88	00 i	0000				
13	00	1967	0000	99						3 11 7 73
14	44	0017	0015	88						i ≠ 0 ?
15	15	0016	0017	8 8			0× 1991	В		Cas où i = 0
16	00	0024	0000	99						
17	69	0047	0018	88			od CTRi 1991		10 xxxx &+ 49	
18	22	0047	0019	88			1991		to CTRi ∆+ 49	en A + 47
					00 SER 1				1991	7 2 1 K 1 1 3
19	69	0049	0020	88	. 1			- 1	CTRi	Jas W.
20	22	0049	0021	88				12	21 1991 xxxx	
21	23	0049	0022	88	40.00			12	21 CTRi 1991 B	en $\Delta + 49$
22	69	0038	0023	88	-0.5(9) 8.19			1	20 xxxx ∆+ 39	
23	22	0038	0024	88				12	20 CTRi A + 39	en A + 38
24	11	8003	0025	98	0	0		1,	00 i 0000	P
25	30	0008	0026	98			00		2 0000	
26	35	0004	0050	98		- 1	00 000 % 00			
50	45	0051	0052	88	İ				100	<b>∞</b> ≠ 0 ?
51	69	0041	0054	88			G + + + ' ' ' ' '		24. △+55. △ +1	cas où of ≠ 0
52	69	. 0053	0054	88			**		00 0000 △+35	cas où d = 0
53	00	.0000	0035	98	77 1					
54	24	0000	0027	88	0.000 1 0.00				24 A+55 A+1	
74		0000	0021		(9)(				00 0000 Δ+35	en ∆ + 0
27	15	0028	0029	88		o	O CTRHO OC	000		6.00
28	00	1940	0000	99	a de la composition della comp					
29	69	0035	0030	88				10	60 xxxx ∆+44	
30	22	0035	0031	88					60 CTRH & 4+ 44	en A + 35
31	69	0039	0032	88		1		12	24 xxxx 4+40	

.../...

. . . / . .

1	0032	122	0039	0033	:8	18		100 CTRHW OOOC	24 CTRHαΔ + 40	en Δ + 39
	33		0046	0034	1	8		1	21 xxxx A + 36	OH 24 73
	34		0046	0035	1	8				A . 46
	24	122	0046	0055	°	0	,	a N 1	21 CTRHqA + 36	en <b>4</b> + 46
	( 35	60	CTRH	0044		8	0 0 h	0 0		
	44	10	0045	0046	8	8	00 h+1		0	
	45	00	0000	0001	9	9				
	(46	) 21	CTRH	0036		8			0 0 h+1	en CTRH &
	36	11	0055	0037	8	8	00 h+l-H		0 O H	
	37	46	0048	0038	8	8				h + 1 - H < 0 ?
1	( 38	3) 20	CTRi	0039		8			0 0	en CTRi (si i ≠ 0)
1	( 38	20	1991	0039		8			00	en 1991 (si i = 0)
	39	24	CTRHO	0040		8			00	en CTRH
	40	69	0041	0042	8	8		1 200	24 △ +55 △ +1	•
٦	41	24	0055	0001	8	8				
	( 42	) 24	0000	N	8			222	24 △+55 △+1	en A + 0
			1					-51		
	48	60	0004	0043	8	8	00 C N	00		cas où h+1 < H
	43	30	0004	0047	9	8	00 C	N 00 0000		
1	(47	) 10	CTRi	0049		8	00(CTRi)+C		0 O(CTRi)	si i ≠ 0
1	(47	) 10	1991	0049		8	x x		x x	<b>si</b> i = 0
-	(49	) 21	CTRi	В			00(CTRi)+C		00(CTRi)+C	en CTRi (si i $\neq$ 0)
}	(49	) 21	1991	В			x x	200	xx	en 1991 (si i = 0)

- Les instructions  $\Delta \div 0$  à  $\Delta \div 34$  constituent la partie de formation des ordres ; cette formation a lieu à partir des indications données par les mots qui ont été fournis par l'autoprogrammation. D'abord 00 0000 H est envoyé dans la mémoire  $\Delta$ + 55. Ce n'est pas dans une mémoire de travail qu'il est envoyé car il doit être conservé d'un passage à l'autre si la phase de formation des ordres est supprimée. Il en est de même de 00 C N envoyé en A+ 4 et non dans une mémoire de travail.
- Pour tester i, on le forme dans l'accumulateur gauche. Pour cela, on appelle 00 CTRi 0000 dans cet accumulateur et on en soustrait la quantité 00 1967 0000 (1967 est l'adresse du CTRO). Si la réponse au test est non, i = 0, l'accumulateur droit contient Od CTRO B = 0d 1967 B. Pour éviter de former des ordres susceptibles de modifier le CTRO, on remplace l'adresse facteur de l'accumulateur droit par W 0001 = 1991 en lui ajoutant 00 0024 0000: 1967 + 24 = 1991. Ainsi les ordres formés à partir de cette adresse ne modifieront pas la mémoire CTRO mais la mémoire W 0001 qui peut être transformée sans inconvénient.
- \_L'ordre d'adresse  $\Delta$  + 50 exécute un test sur  $\alpha$  . Si  $\alpha$  = 0, l'ordre  $\Delta$  + 52 prépare dans le distributeur le mot 00 0000  $\Delta + 35$  que l'ordre  $\Delta + 54$  envoie en ∆+ C pour que la phase de formation des ordres soit supprimée (pour le prochain passage par l'ordre de bouclage). Si  $\neq$  0, l'ordre  $\Delta$  + 54 l'envoie en A+ 0 (ce qui est d'ailleurs inutile).

- a 00 000d 0000 la constante 00 CTRH0 000 = 00 1940 0000.
- En Δ + 35 débute l'exécution proprement dite du bouclage.
- L'instruction  $\Delta + 37$  exécute un test sur h + 1 H. Si  $h + 1 H \geqslant 0$  le programme est aiguillé en  $\Delta + 30$  pour exécuter la sortie de boucle. Le registre d'index CTRi et le compteur de boucles CTRH  $\alpha$  sont remis à zéro. En fait si i = 0 c'est la mémoire 1991 = W 0001 qui est remise à zéro. Ensuite a lieu la régénération de la mémoire  $\Delta + 0000$ , indispensable dans le cas où  $\alpha = 0$ . Le dernier ordre adresse à l'instruction N. Si h + 1 H < 0, le programme est aiguillé en  $\Delta + 48$  pour que le bouclage continue. La machine ajoute C au contenu du registre d'index CTRi si  $i \neq 0$  puis adresse à B pour boucler.

#### X - F - SOUS-PROGRAMMES PLACES A POSTE FIXE.

Comme en logique extérieure, deux sous-programmes ont été placés à poste fixe dans les paquets verts à joindre aux programmes autoperforés. Ce sont la perforation d'une séquence de mémoireset la modification de bouclage.

#### X - F - 1 - Perforation (ordre 49 N iA jB kC)

L'entrée dans ce sous-programme est standard, et est traitée en autoprogrammation comme l'entrée dans n'importe quel sous-programme. Dans les paquets d'autoprogrammation a été placée à poste fixe une carte chargeant 00 0000  $\Delta_{49}$  = 00 0000 1889 en DEL + 49 = 1875 + 49 = 1875 + 49 = 1924.

L'organigramme est semblable à celui qui a été fait en logique extérieure, à part que la branche FIN envoie, après la perforation, à l'adresse N . et non à DEBUT = 1800.

Le programme est légèrement plus long que celui en logique extérieure car ce dernicr, non standard, avait une partie de formation d'ordres moins importante. Il a été écrit en PASØ puis assemblé, perforé sous forme translatable et translaté en  $\Delta_{QQ}$  = 1889.

1	10000	T,DI	W10001	,	0 0	00 A B	20 C N 20 C N	Entrée standard en W1 = 1991
2	,	E DI	6 9AG				69 0000 8003	
3		D AG	0004		0O A A	A A B 0000		
4	, I	SIBF	W 0002				69 B 8003	en W2 = 1992
5		D AD	8000		0 0	O OA		
6		TRD	W 0003	f	in Silfe		A0	en W3 = 1993
7		D AG	0004			0000 A 000	- 1	
8		A AD	2,4PS1	ŧ	. A	24 P2+A SUITI	24 P2 SUIT1	
9	,	TIRD	W 0004				24 P2+A SUIT1	en W4 = 1994
10		R AD	W 0001		0 0	20 C N	l day	-71
11	1	EIDI	P FØFI		15 6811881		71 P1 xxxx	
12	20 1	S BI	P FØFI				71 P1 N	en PFØFI
13		E DI	,8003	1			00	
14	1	S BF	W 0005				00 C 0000	en W5 = 1995
15		R AD	W 0002	B ØUCL	0 0	69 B 8003	-0.0	
	1	1	1 1777	1	OA DO GUITMI	CO X 0007		
16	B ØUCL	A AG	2 4PS1		24 P2 SUIT1	69 X 8003	0 O A	
17		E DI	W 0003	lasse				en Pl = 1977
18		SBF	P 0001	8002			A X 00	en Fr = 1911
	(8002)	E DI	X	8003			(X)	en P2 ==1978
	(8003)	TIDI	Pl.0002	SIUITI	11	1	(X)	, en tv ==1310
								/

19 S UIT	A A G O I D		24P2+m+1 SUITI	-	24	
20	S AD W 0005		F 520 1300	68 X+m-C 8003	00 C 0000	
21	E DI 8002				68 X+m-C 8003	
22		F IN	Markey 1/L	04.57	9	X+m ( C ?
23	A AD 0 10			68 X+m-C+1 8003		15
24	A, AD W 10005			69 X+m+1 8003		
25	S AG W 0004		m+1-A		24 P2+A SUIT1	
26	G NN	P FØ				m+1 ≠ A?
27	A,AG 8001	18002	24P2+m+lSUIT1		- '	
(800	2) E DI X +m+1	8003			(X+m+1)	- 1
1800	3) T DI P2+m+1	s uiti	17 L 1 A		(X+m+1)	en $P2 + m + 1$
		1				
28 P FØ	P FØ P 0001	B ØUCL	0	69 X+m+1 8003	3.1	-1
					12111114	72. 2
29 F IN	RIAD 18003	1	00	24 P2+m+l SUIT1		
30	S AD 2 4PS1	E 141	3-4 x 2	00 m+1 0000	11% o 5 T	
31	E DI P 0001	1,5			00 X A	
32 1	D AD 0004	1		00 m+1	× 1	
33	S BI P 0001	p føfi			00 X m+1	en Pl = 1977
PIFF	I PIFØ Ploool	I N			н -	
34 0 10	00 0001	0000				
5 P FØF	71 P <sup>1</sup> 0001	0000				
6 2 4PS		- 1		ese s &		i la la la la
7 619AG	69 10000	18003				

L'ordre n°12, après avoir formé l'ordre final PFØ PO001 N., l'envoie à sa place, c'est - à - dire en PFØFI. On aurait d'ailleurs pu gagner la mémoire PFØFI en remplaçant les ordres n° 11 et 12 respectivement par EDI PFØ et SBI W0006, et en remplaçant l'adresse "instruction suivante" de l'ordre n°33 par W0006.

#### X - F - 2 - Modification de bouclage (ordre 00 N 01937 jB kC)

La modification de bouclage doit remplacer le H de l'ordre de bouclage placé
en Get C+1 par H', placé sous la forme 00 0000 H' dans la mémoire B. En
exécution en langage machine, la valeur de H correspondant à l'ordre de
bouclage d'adresse C est placée sous la forme 00 0000 H dans la mémoire C+1
( voir le paragraphe X-E-4-a ). Pour effectuer la modification de bouclage,
il suffit d'envoyer 00 0000 H' en C+1.

Le facteur de translation de ce court sous - programme a été fixé à 1885. Mais, pour les raisons expliquées au paragraphe VII - D - 2, les deux premières mémoirez ont été remplacées par 1937 et 1938 et ainsi libérées. Le sous - programme de modification de bouclage destiné à l'exécution en langage machine peut donc séécrire :

	65 B A+1	20 C N		
A = 1937 15   \( \Delta + 2 \)   \( \Delta + 3 \)		24 C+1 N	04 0001 0000	
Δ + 0002 04 0001 0000	-1- (-1)			
0003 20 1938 8003	elfentur		24 C+1 N	en $A + 1 = 1938$
(8003) 65 B A + 1	00	0О Н'	.00 0000 H'	and the state of t
A+1=(1938) 24 C+1 N		I Da	00 0000 H;	en C + 1

# XI - DISPOSITION DE L'AUTOPROGRAMMATION

# ET DES SOUS-PROGRAMMES DES " PAQUETS VERTS "

#### XI - A - AUTOPROGRAMMATION :

- Le programme d'autoprogrammation occupe les mémoires 1352 à 1881.
- Les mémoires 1882, 1883, 1884 sont respectivement les mémoires CTRL, IMAX et MEMNN (qui contient l'adresse de l'instruction suivante). Le mémoire 1800 contient la l<sup>ère</sup> instruction de l'autoprogrammation et est appelée DEBUT.
- Les mémoires DEL + 10 = 1875 + 10 = D0000 à DEL + 59 = 1934 = D0049 sont destinées à recevoir les facteurs de translation  $\Delta_{\rm pp}$ .
- DEL + 60 = 1935 peut aussi contenir un △.
- DEL + 61 = D0051 = 1936 contient le facteur de translation du sous programme de bouclage, soit 00 0000 1751.
- DEL + 62 = D0052 = 1937 contient le facteur de translation du sous programme d'indexage, soit 00 0000 1807.
- Les mémoires 1940 à 1999, utilisées pour la plupart par le programme de chargement, ne peuvent pas contenir d'instruction.
- Certaines des mémoires utilisées comme registres d'index pour l'exécution des programmes (1967 à 1976) jouent en autoprogrammation le rôle de mémoires de travail.
- Les mémoires de perforation 1977 à 1984 jouent un rôle très important. C'est à partir de ces mémoires que se fait la perforation des programmes en langage machine formés par l'autoprogrammation.
- Les mémoires 1991 à 1999 sont utilisées comme mémoires de travail. En particulier, les mots devant aller en mots 2, 3 et 4 des secondes cartes de chaque ordre en CDP sont envoyés d'abord en 1991, 1992, 1993.

#### XI - B - EXECUTION EN LANGAGE MACHINE ( paquets verts )

La partie fixe des paquets (verts) à joindre aux programmes autoperforés est composée de la façon suivante :

1751 1806	Sous - programme de bouclage
· 1807 1881	Sous - programme triple d'indexage
1882 à 1886	Libres
1887 1888	Sous-programme de modification de bouclage (début en 1937)
1889 1925	Sous - programme de perforation (ordre 49)
1926 à 1929	Libres
K = 1930 1935	Partie constante de l'initialisation (-02 U V)
1936	Mémoire ARETM . Contient : 01 1333 8000
1937 1938	Début de la modification de bouclage
1939,	Mémoire Z . Contient : 44 ARETM W0002 = 44 1936 1992
1940 1949	Compteurs de boucles ( région H )
1950	Libre
1951 1960	Mémoires de lecture (région L). Utilisables par le programmeur
1961 à 1966	Libres
1967 1976	Registres d'index (région C). 1967 contient toujours ZERO
1977 1986	Mémoires de perforation (région P). Non utilisables par programmeur
1987 1990	Libres. Mais 1987 est utilisé comme mémoire de travail par le sous - programme auxiliaire de décalage.
1991 1999	Mémoires de travail (région W)

# XII - SOUS-PROGRAMMES ADAPTÉS AU C.D.P..

### DIVERS C.D.P. FORMÉS

#### XII - A - GENERALITES

A la partie commune du C.D.P. peuvent être ajoutés des sous-programmes à 2 ou 3 adresses. C'est ainsi qu'ont été constitués les trois C.D.P. existant pour le moment : virgule fixe, virgule flottante simple et double précision. Chacun de ces C.D.P. peut être complété par d'autres sous-programmes. D'autres C.D.P. peuvent être créés en ajoutant des sous-programmes à la partie commune du C.D.P.

Les sous-programmes à ajouter au C.D.P. doivent être compatibles avec l'une des 2 formes d'entrée prévues : entrée dans les sous-programmes à 2 ou ceux à 3 adresses. Ils peuvent utiliser les mémoires W 0001 = 1991 à W 0007 = 1997 comme mémoires de travail, c'est-à-dire y envoyer des résultats ou des indications intermédiaires comme, par exemple, l'instruction de sortie.

Rappelons l'état des accumulateurs et du distributeur au moment de l'entrée dans un sous-programme :

- Sous-programme à 2 adresses :

	80	003		0002	x 550	8001
65	B	A + 1	20	<u> </u>	NII	

La lère instruction est en A.

- Sous-Programme à 3 adresses :

8303	4.5	8002	8001		
0	0.9	A B	20	C	N <sub>1</sub>

La lère instruction est en A pp! .

Les 2 formes d'entrée dans les sous-programmes sont indexables par le C. D. P. lui-même, ce qui leur donne un grand intérêt et une grande souplesse d'utilisation.

Pour adjoindre au C. D. P. un sous-programme à 2 adresses compatible avec lui, il suffit de choisir pour ce sous-programme un facteur de translation valable à la fois en logique extérieure et en exécution en langage machine. Le sous-programme, écrit sous forme translatable 5 mots par carte et précédé de son facteur de translation, doit être joint au paquet de logique extérieure ainsi qu'à celui destiné à l'exécution en langage machine. La lère instruction de ce sous-programme, après translation, se trouve alors en A. L'ordre qui commande son exécution est : 30 N 0A jB kC.

S'il est impossible de trouver une place pour le sous-programme, libre à la fois en logique extérieure et en exécution en langage machine, il faut couper le sousprogramme en 2 parties :

- 1°) Une partie constituée de 2 mémoires consécutives au moins, dont l'emplacement soit le même en logique extérieure et en langage machine; c'est dans cette partie que seront choisies les adresses A et A + 1, A devant contenir la lère instruction du sous-programme.
- 2°) L'autre partie translatable différemment en logique extérieure et en langage machine. L'exécution du sous-programme sera commandée par l'ordre 90 N OA jB kC.

.../..

- 167 -

Pour adjoindre au C.D.F. un sous-programme à 3 adresses compatible avec lui, il faut choisir pour ce sous-programme non seulement un facteur de translation mais aussi un code PP'.

En logique extérieure, il faut joindre au paquet du C.D.P. le sous-programme translatable précédé de son facteur de translation  $\Delta_{nn}$ , et une carte destinée à charger en DEL + PP' = 1875 + PP' le mot 00 0000  $\Delta_{pp'}$ . En exécution en langage machine, il faut joindre au paquet du C.D.P. et aux cartes autoperforces le sous-programme translatable précédé de son facteur de translation qui peut être différent de celui en logique extérieure (soit  $\Delta'_{nn}$ ). Il ne faut pas mettre de carte chargeant 00 0000  $\Delta'_{pp'}$  en 1875 + PP'. Par contre il est indispensable de joindre cette carte au paquet d'autoprogrammation. Il est inutile de joindre à celui-ci le sous-programme lui-même. L'ordre qui commande l'exécution du sous-programme est : PP' N iA jB kC.

#### Remarque:

Pour créer des sous-programmes logiques (tests, aiguillages,...) il faudra prendre de grandes précautions car en général ces sous-programmes ne pourront pas être les mêmes en logique extérieure et en exécution en langage machine. Cela est dû au fait que dans ces 2 procédés la façon de désigner l'adresse de l'instruction suivante N est différente : N doit être placé en MEMNN = 1884, en logique extérieure, et en position d'adresse instruction suivant e de la dernière instruction, en langage machine.

#### XII - B - C. D. P. EN VIRGULE FLOTTANTE SIMPLE PRECISION:

Le C. D. P: en virgule flottante simple précision est formé du tronc commun du C. D. P. auxquels sont joints 3 sous-programmes à 3 adresses et 6 sousprogrammes à 2 adresses,

- Les sous-programmes à 3 adresses sont les 4 opérations en virgule flottante

11	N	iA	jВ	kC	(A) + (B) en C \	2 entrées différentes dans le
12	N	iA	jB	kC	(A) - (B) en C	même sous-programme.
13	N	iA	ĵΒ	kC	(A) x (B) en C	49
14	N	iA	iB	kC	(A) : (B) en C	1 21 21 21 21 21

Ces sous-programmes ont le même emplacement en logique extérieure et en langage machine :

l'ordre 11 }	occupe le	s mémoires	1487 à 1555
12)			45-972
13	11.	11	1450 à 1486
14	0	11	1414 à 1449

En logique extérieure et en autoprogrammation, il existe une carte qui charge ;

- Les sous-programmes à 2 adresses calculent les fonctions principales.

00 N 0	1008 jB kC	log <sub>10</sub> (B) en C	
	1014	noge (D) en C /	même sous-programme
	1010	10 <sup>(B)</sup> en C	
	1016	e <sup>(B)</sup> en C	même sous-programme
19	1012	(B) en C	All grant and the particle of

Ces sous-programmes occupent respectivement les mémoires suivantes :

√(B)	138 I à 1413
sin_et cos	1201 à 1288
Arc sin	1101 à 1200
Logarithmes	1020 à 1100
Exponentielles	1289 à 1380
Valeur absolue	1012 et 1013

En fait, pour simplifier les codes de ces ordres, on a reproduit :

138 1 et	1382	en	1000 et	1001
1201 et	1202	en	1002 et	1003
1203 et	1204	en	1004 et	1005
1101 et	1102	en	1006 et	1007
1022 et	1023	en	1008 et	1009
1020 et	1021	en	1014 et	1015
1291 et	1292	en	1010 et	1011
1289 et	1290	en	1016 et	1017

Les mémoires 1381, 1382, 1201 à 1204, 1101, 1102, 1020 à 1023 et 1289 à 1292 pourraient être libérées.

.../..

#### XII - C - C. D. P. EN VIRGULE FLOTTANTE DOUBLE PRECISION. -

Ce C. D. P. est formé du C. D. P. en virgule flottante simple précision auquel sont jointes les 4 opérations en virgule flottante double précision (2,18). Chaque nombre est contenu dans 2 mémoires; par exemple le nombre désigné par son adresse D est en fait contenu en D et D + 1. Les 4 opérations sont exécutées par 3 sous-programmes à 3 adresses:

21	N i	А јВ	kC	addition	
22	N i	А јВ	kC	soustraction } même sous-progra	mme
23	N i	А јВ	kC	multiplication	
24	N i	A jB	kC	division	7.

Ces sous-programmes ont le même emplacement en logique extérieure et en langage machine :

121	occupe les	mémoires	714 à 840
122	V F .50	e g of a	
23	H	11	841 à 914
24	ti	n	915 à 999

En logique extérieure et en autoprogrammation, il existe une carte qui charge :

#### XII - D - C. D. P. EN VIRGULE FIXE. -

Les opérations en virgule fixe existent dans la partie commune du C. D. P.. Seules les fonctions ont été ajoutées à ce tronc commun pour former le C. D. P. en virgule fixe. Mais les sous-programmes de calcul de fonction en virgule fixe exigent une certaine position de virgule pour l'argument et fournissent la fonction

avec une position de virgule bien déterminée. Or, ces positionnements ne sont pas toujours compatibles avec le reste du programme. Il a donc été jugé nécessaire de prévoir des possibilités de décalage : avant l'exécution du sous-programme, décalage de l'argument pour ajuster la position de sa virgule sur la position exigée par le sous-programme; après l'exécution du sous-programme, décalage de la fonction pour l'amener au positionnement voulu pour le reste du problème.

Les calculs se font par l'intermédiaire d'un sous-programme nommé "sousprogramme auxiliaire de décalage".

#### XII - D - I - Sous-programme auxiliaire de décalage :

C'est un sous-programme à 3 adresses à 10 entrées, une pour chaque valeur de P'. Appelons D le facteur de translation de ce sous-programme.

L'entrée relative à P' = 0 se fait en D + 0, celle relative à P' = 1 se fait en D + 1, ...., celle relative à P' = 9 en D + 9. Ainsi dans la mémoire DEL + 50 = 1925 se trouve 00 0000 D, en DEL + 51 = 1926 se trouve 00 0000 D+1,....,

en DEL + 59 = 1934 se trouve 00 0000 D+9. Les entrées relatives aux divers P<sup>1</sup> se rejoingnent après que chaque branche ait introduit dans le distributeur le facteur de translation  $\Delta_{\mathbf{P}^1}$  du sous-programme devant calculer la fonction désignée par P'.

Voici la succession des ordres qui précèdent le rassemblement des branches en D + 30 :

Adresse	Code	F	I	(8001)	
D + P'	TDI EDI	W 0001 D + 20 + P'	D + 10 + P'  D + 30	20 d ) 6 M N 20 d ) 6 M N 30 0000 D p.	en Wl
D + 20 + P'	30	0000	Δ <sub>P'</sub>		

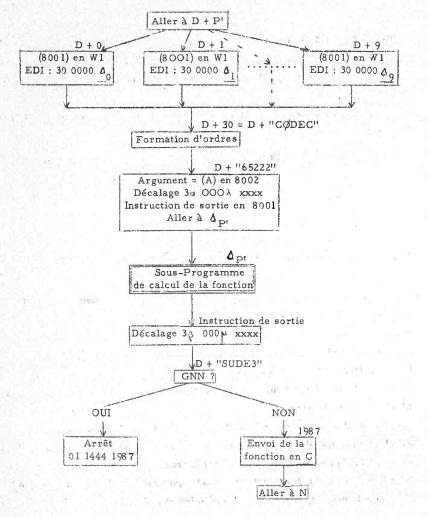
Cette partie, existant pour les 10 valeurs de P<sup>1</sup>, occupe les 30 premières mémoires du sous-programme.

Le déroulement du programme peut se représenter par le schéma suivant : (voir schéma page suivante)

Les sous-programmes de calcul de fonctions en virgule fixe adaptables au sousprogramme auxiliaire de décalage doivent prendre l'argument dans l'accumulateur
(gauche, droit ou les deux) et l'instruction de sortie dans le distributeur. Ils
remettent la fonction calculée dans l'accumulateur (gauche, droit ou les deux).

L'entrée effectuée par le programme auxiliaire de décalage prépare dans l'accumulateur l'argument x décalé et dans le distributeur l'instruction de sortie
qui n'est autre que l'ordre de décalage à effectuer sur la fonction avant de l'envoyer en B. Après ce décalage, et avant le transfert de l'accumulateur droit en
B, a lieu un test sur l'accumulateur gauche qui provoque l'arrêt 01 1444 1987
si cet accumulateur est non nul, c'est-à-dire si la fonction a plus de 10 chiffres.

.../...



Les sous-programmes auxiliaires de décalage utilisés en langage machine et en logique extérieure sont un peu différents car dans ce dernier cas le sous-programme comporte la préparation de l'analyse. En effet, une analyse spéciale a été créée pour les ordres de calculs de fonctions avec décalages. Cette analyse permet de mettre en évidence les décalages. Les cartes obtenues sont de la forme :

Mot 1: n

Mot 2: Instruction 5P' NiA jB Oak B M

Mot 4: 0 (CTRi) (CTRj) 000

Mot 5: Argument dans l'accumulateur entier après décalage

Mot 7: Fonction après décalage (telle qu'elle est envoyée en B)

Mot 8: 20 αλβμ 1700

Les mots 1,2,3,4 sont préparés par la partie commune de la logique extérieure.

#### a) Sous - programme auxiliaire de décalage pour l'exécution en langage machine :

. /...

+ CIØDEC	TIDI	W 0002	1 1	0	0	00 A B	30 0000 Δ <sub>P</sub> ,	en W2
	E DI	6,5222	1137.7				65 xxxx	
	SIBF	6 5222	100	-10	-11		65 A	en 65222
	D AG	0004		0	0	ыж В 0000		
	E DI	W 0001					20 01 /3 pm N	Lincoln St.
	SBF	1987	Large				20 B N	en 1987
1	R AD	W 0001	Y 1	0	0	30 My 13 M N		
	DIAD	0004				000020 · 1 2 m		
	D AG	0005		0	02	000000 4 K K O		
	AAD	M 0005				30 A SMO Ap.		
	R AD	8002	1	0	0	3018 MODP	3 d & 3 m 0 Dp,	
	D AD	10003				00030( ) B MOX	3.1	
	SBF	W 0001					32 0xx \ A.p.	en Wl
275	D AG	10006	1	x	x	₹ ₩0x000000	1	
100	S. AG	8003		0	0		Law To	
	DIAD	0001				0 B M 0x00000		
) mission a	A AD	3 OSU3	1			33 MXXXSUDE3	30 0000 SUDE3	
	EjDI	8002				1,	36 xxxx SUDE3	
	DAD	0003	765			0003β ► XXXX	,	
	SBF	W 0002	6 5222				3 8 Oxx A SUDE3	en W2
6 5222	65	2222				- ta	, I - 8 N	
(6 5222)	R AD	1 A	1	0	0	х		
	EIDI	M 0005	W 0001				3 0xx M SUDE3	
(W 0001)	3 d	i0xxλ	Ap,		x	décalé		ler décalage
	.1				f	(x)		
Slortie	36	10xx M	SIUDE3			f (x)		2ème décalage
S UDE3	G NN		1987					
	A RT	1444	1987				9.2	4
(1987)	TIRD	В	N	0	0	f (x)	f (x)	en B
310SU3	130	10000	SIUDE3					

- Après assemblage, ce sous-programme a été mis sous forme translatable. Son facteur de translation a été fixé à D = 1692.
- Les 3 instructions notées après les réservations sont là à titre indicatif mais ne figurent pas dans le programme PASØ.
- Les instructions de décalage formées ont comme adresse facteur respectivement  $0xx\lambda$  et  $0xx\mu$  et non  $000\lambda$  et  $000\mu$ . Mais pour les ordres de décalage, seule la position de droite de l'adresse facteur est considérée, à condition toutefois que cette adresse soit valable, ce qui est le cas pour les adresses  $0xx\lambda$  et  $0xx\mu$ .
- La mémoire 1987 est utilisée comme mémoire de travail par le sous-programme auxiliaire de décalage.
- Remarque: Si l'ordre 5P' est indexé, on passe par le sous-programme d'indexage (réservé aux sous-programmes à 3 adresses) avant d'entrer dans le sous-programme de décalage.

### b) Sous-programme auxiliaire de décalage en logique extérieure :

en Wl

C ØDEC	TIDI.	W/0002		0 0	00 A B	30 0000 A P	en W2
2 10	E DI	6 5222				65 xxxx W1	of applications
8-4	SIBF	615222	1			65 A W1	en 65222
	D AG	0004		XX	xx B 0000		
	E DI	2 0501			92 PH 754	20 0000 SUDE1	
	SIBF	1987	1			20 B SUDEL	en 1987
	R AD	W 0002		00	30 .0000 Ap,		
	E DI	6 9W2			7.00	69 W2 xxxx	
1	S,BI	619W2	1			69 W2 A P'	en 69W2
	R AD	W 0001		0 0	20 da 8 m1700	P	
	T DI	P 0008				20 m h pt 1700	en P8 (analyse)
1	DAD	10004	1	100	000020 axim	4	
	D AG	0005		2		m and mid	
100	A AD	3 OSU2			30XBHOSUDE2	30 0000 SUDE2 '	
	R,AD	18002		0 0	3dlp MOSUDE2	3 × A & MOSUDE2	
	D'AD	0003			0003 x / 64 0x		
1	S BF	W 0001				3d Oxxà SUDE2	en Wl
	D AG	10006	i_	00xxx	B M 0x00		
	SAG	8003		0 0	,	/= . \	81.4
1	D AD	0001			0 в нохоо		
1	A AD	3,0SU3	1		38 MOXOSUDE3	30 0000 SUDE3	
,	E   DI	8002				3/3 MOXO SUDE3	
	D AD	0003			0003 3 H 0x0x	1	the for the
	SBF	W,0002	6 5222			33 OxxxxiSUDE3	en W2
6 5222	65	2222	W 0001			, ,	
6 5222)	R AD	A	W 0001	0 0	x		
N 0001)	30	Oxx	S UDE2	x	décalé		ler décalage
3 UDE2	T RG	P 0005					en P5 et P6
	T RD	P 0006	6 9W2			x	(analyse)

1 2 to		State of	7 000	a star of a fi		and the second	
6 9W2	169	W,0002	2222	C 13 -0	-0 -0 F K	1 1 1 1 1 1 1	= 100 st. 50 st. 2
(6 9%2)	E DI	M 0005	Δ <sub>P</sub>	'х	décalé	33 Oxx M SUDE3.	ever Par Envis
				f	(x)		
Sortie	38	Oxx	S UDE3	W. S.	f (x)	article of	2ème décalage
STUDES	G NN		1987				
	A ET	11444	1987	4 8 0	55.86° 1 °61	Mrs " = g r =	the standard
(11987)	TED	В	S UDE1		11 7/11 2/1	f (x)	en B
S UTEL	T DI	P 0007				f (x)	en P7(analyse)
1	RIAG	18000	1	6 In 3527		0244	
	A NG	P FDEB	D EBUT		- 1	007	
		1				. 18	
3 0Su2	30	0000	S UDE2		U.W 5	I AM	
3 USU3	30	0000	S UDE3		-1.11		12,000
alosua	120	10000	SIUDEL	the second	, - C M	1 , 5	

- Après assemblage, ce sous-programme a été mis sous forme translatable. Son facteur de translation a été fixé à D=1490.

- Les 2ème, 3ème et 4ème paragraphes qui suivent la liste du sous-programme pour l'exécution en langage machine sont valables pour la logique extérieure.

- L'instruction PFDEB est l'instruction n°40 de la logique extérieure, et a comme adresse 1738. Le pseudo-code PEN PFDEB 1738 sert à affecter l'adresse 1738 à PFDEB dans l'assemblage du sous-programme.

.../...

#### XII - D - 2 - Ordres particuliers au C. D. P. en virgule fixe :

Le C. D. P. en virgule fixe comporte le sous-programme auxiliaire de décalage et 9 entrées différentes dans celui-ci (correspondant chacune à un calcul de fonction). Les codes utilisés sont : PP' = 50, PP' = 51, ..., PP' = 58.

Le code 59 reste libre, ce qui permettra d'ajouter au C. D. P. en virgule fixe un sous-programme supplémentaire nécessitant des décalages.

Voici les 9 ordres fonctionnels avec décalages :

Leur emplacement est différent en logique extérieure et en exécution en langage machine.

#### - En logique extérieure ;

1875 + 50 = 1925 contient 00 0000 D = 00 0000 1490

(D = lère instruction du sous-programme auxiliaire de décalage).

1875 + 5 P' = 1925 + P' contient 00 0000 D + P' = 00 0000 149 P'.

$$D + 20 + 0 = 1510$$
 contient 30 0000  $\Delta_0 = 30$  0000 1461

$$1 = 1511$$
  $\triangle_{1} = 1377$   
 $2 = 1512$   $\triangle_{2} = 1376$ 

.../...

#### - En autoprogrammation :

1875 = 5 P' = 1925 + P' contient 00 0000 D' + P' = 00 0000 1692 + P'

#### - En exécution en langage machine :

HAS IN THE PARTY OF THE STATE O

#### XIII - REMARQUES DIVERSES

#### XIII - A - CHARGEMENT

Chaque exemplaire du C.D.P. est précédé d'un paquet jaune de chargement. Ce programme de chargement charge non seulement les cartes n mots par carte  $(n \leqslant 7)$  ordinaires mais aussi des cartes translatables n mots par carte  $(n \leqslant 5)$  de la forme suivante :

Mot 1: 00 ABCD 000n NEGATIF

2 : Instruction 1 qui, translatée, entrera en ABCD + △

3: " 2 éventuellement " "ABCD + Δ + 1

4: " 3 " , " "ABCD +  $\Delta$  + 2

5 : " 4 " " ABCD + \( \Delta + 3

 $6 \cdot 11 \cdot 5 \cdot 11 \cdot 11 \cdot 11 \cdot 11 \cdot ABCD + \Delta + 4$ 

7: Indicatif de translation:  $\mu_F^1 \mu_I^1 \mu_F^2 \mu_I^2 \mu_I^5 \mu_I^5$ 

\( \mathbb{H}^{\text{x}}\_{\text{F}} = 8 \text{ si l'adresse F de l'instruction x doit être translatée} \)

Mr = 9 dans le cas contraire

W x = 8 si l'adresse I de l'instruction x doit être translatée

MT = 9 dans le-cas contraire.

8 : Indicatif ou blanc.

 Si, pour une carte, n < 5, le mot 7 contient les µ correspondant aux n mots et doit être obligatoirement complété à droite par n'importe quoi, par exemple des zéros. - Les mots négatifs sont également translatés, c'est-à-dire augmentés en valeur absolue.

- Le facteur de translation  $\Delta$  doit être chargé dans 3 mémoires de la façon suivante :

00 \( \Delta \) 0000 en 1997

00 0000 △ en 1998

00 \( \Delta \) \( \Delta \) en 1999

Ge chargement peut être fait par une carte n mots par carte non translatable, avec n = 3.

- Le programme de chargement utilise les mémoires 1941 à 1999 à part 1950 et 1967. Il est détruit lorsque s'exécute un programme. Inversement, si l'on veut le réutiliser au cours d'un calcul, il détruit les compteurs de boucles, registres d'index et mémoires de travail. Il existe d'ailleurs aussi sous forme translatable (1 instruction par carte).

Dans la liste ci-dessous, les notations suivantes ont été adoptées :

- ABCD est appelé A (pour abréger)
- L'instruction à envoyer en A ou A +∆ est appelée Ins 0 et non Instruction 1 comme sur la description des cartes translatables.
- L'instruction à envoyer en A + p ou A + △ + p est appelée
   Ins p.. p varie de 0 à n 1.
- A + A = A!

	A PA							
		PIRA	11967	11967				
1		P RA	1995	1996				
2		P RA	0000	1940				
3		PRA	11950	1950				
4		P RG	D 1997	1999				
5 6		PIRG	L 1951	1960				
6		P EN	S)C	1994				
				1				
44	L'ECT	LEC	S_C	8000	10 15 1 X	04 2		
18	SIC	R AD	L 0001	100	0 0	00 A 000n		
19		A NG	T RAN	N TRAN	1 10			Translatable ?
1//		12	Charle			1 10	1/2	. 6
21	T RAN	R AD	D 0001		0 0	00 <b>Δ</b> 0000		Translatable
22		E DI	L 0007	1 1	- 12		indic. da trans	
23		TDI	L 0010				indic. de trans	en L 10
24		E DI	M 2	c ømmu			65 L2 SB	
20	n TRAN	R AG	М 3	clømmu	65 L2 M1	00	65 L2 M1	Non transl.
				.03	100		an an	
25	ciømmu	T DI	L 0009		A	A. A	65 L2 SB .	en L 9
26		A VD	F 0001	Share	- 11 1	00 A+A 000n	3.7	
27	1	EIDI	M 1	c 13		4 200	20 XXXX SA	
28		S BF	Mil		,		20 A+ A SA	en M 1
29	ku - sk	D AG	0004		xx	xx n 0000		
30		A AD	18001			xx A+A+nSA	A. A.m	
31		S BF	L 0001	L 0009			20 <sup>A+Δ+n</sup> SA	en L l
W			1		Accordance to	Anne Lunio Lee		
	(I10009)		L 2+p	Ml	0 0	Ins p		Cas où non transl.
100	(M 1)	T RD	A+p	SA			Ins p	en A + p
7	SA	RIAG	MII	T	20 A+p SA	0 0		
8	- '	A AG	M ILLE		20 A+p+1 SA		00 0001 0000	
9		A AD	8001			00 0001 0000		
10	1	AlAD	L10009	1		65 L2+p+l M1	65 L2+p Ml	

11		TIRG			20 A+p+1 SA	65 L2+p+l Ml	20 A+p+1 SA	en Ml
12	194	T RD	L,0009	1			65 L2+p+l Ml	en L9
13		SIAG	L 0001	4-11	p+1-n		20 A + n SA	
14	-34	A NG	S VITE	L ECT	Service Co.	ST SEL SE		p+1 (n?
15	S, UITE	R AG	r 0010		0 0	00		
16		D! AG	0002	SID		10 m 5 m		
17	SD	T RG	L,0010	L 0009			00	en L 10
	(F 0003)		L 2 +p	SB	0 0	Ins p		Cas où transl
	SB	A NG	Ø UI	n'øn				Ins p < 0 ?
34	\$ UI	S AG	nin	T EST	0 0 1	CFI -		
35	n øn	A AG	UN	T EST	0 0 1	CFI	ת היו היו	
36	TEST	E DI	T 0010		0 0 1	CFI +	p p p+l p+l	
57		S DO	HUIT	N EUF				№ P = 8 ?
38	H'UIT	S D9	D ELDE	DELZE				MP = 8?
40	D EITE	MILL	D 0003	A DDIT	CFI	00 A ·A ±	00 A A	
1.1	D ELZE	M UL	Di0001	A DDIT	CFI	00 A 0000 ±	00 △ 0000	
39	NEUF	S D9	Z EDEL	MI				MP = 8 ?
42	Z EOUL	MIUL	D 0002	A DDIT	CFI	00 0000 A ±	00 0000 4	11
43	A DDIT	A AD	8003	MI	CFI	c F' I' ±	CFI	
	(M/1)	TIRD	A +A+p	SA		415 1	C'F'I'±	en $A + \Delta + p$
7	(S A)	R AG	MI	JA	20A'+p SA	00	V 1 1	
8	( A)	A AG	M ILLE	' '	20A'+p+l SA	1	00 0001 0000	
9	( )	A¦AD	8001		1 to 1 to 1	00 0001 0000		
10	( )	A AD	L10009		V	65 L2+p+l SB	65 L2+p SB	
11	( )	T EG	N 1		127		20A'+p+1 SA	en Ml
12	(1 ))	TIRD	L 0009	in			65L2+p+1 SB	en L 9
13	( )	SAG	L 0001		p+1-n	P 3 7 8	3	
14	(, )	A NG	SUITE	LECT	18-			p+1 < n?
15	(s'UITE)	RIAG	L 0010		p p p+l p+l	00		
16	( )	D AG	10005	SD	p p+l p+l FIF I p+l p+l 00			
17	The same of	TIRG	L 0010	L 0009	F I		μ <sup>p+1</sup> μ <sup>p+1</sup> 00	en L 10
45	M 1	20	0000	SA				
46	M,2	165	L10002	SIB				
47	И3	65	L 0002	M 1			As a second	
48	N ILLE	00	0001	0000				
40	UIN	100	10000	10001	•			

- Le PEN SC 1994 a comme principale utilité d'obliger SC à être une adresse de la bande 1950 pour que la lecture se fasse bien dans les mémoires
- Dès l'instruction n° 19 se séparent les parties relatives aux cartes translatables et non translatables. Ces parties se rejoignent à COMMU (n° 25) puis se reséparent à l'instruction n° 31 pour se retrouver en M1.
- Les mémoires L 0009 et L 0010 servent de mémoires de travail.
- Lorsque les 2 branches se rejoignent en CØMMU (n° 25), l'accumulateur droit contient : pour les cartes translatables, 00 Δ 0000, et pour les cartes non translatables, 0.....0 (grâce au fait que l'instruction NTRAN soit un RAG au lieu d'un EDI).
- Si p + 1 n ≥ 0, le chargement de la carte est terminé et la carte suivante est lue (test n° 14).
- Les instructions nº15, 16 et 17 sont in tiles pour les cartes non translatables mais sont conservées pour que les ordres qui les précèdent puissent être communs aux cartes translatables ou non.
- Pour garder une plus grande partie commune aux ordres négatifs et positifs à translater, on a entré 1 dans l'accumulateur gauche pour les ordres négatifs (ØUI) et + 1 pour les positifs (NØN).

- L'introduction des de fait alors, non pas par addition directe, mais par l'intermédiaire d'une multiplication qui affecte la quantité à ajouter à l'instruction du signe convenable (le signe de l'instruction).

#### XIII - B - FORMATIONS D'ORDRES.

Soit un ordre quelconque à 2 (ou 3) mémoires PP' N iA jB kC (à l'exception d'un ordre de bouclage) dont une des adresses, A par exemple, doive prendre des valeurs diverses, A<sub>1</sub>, A<sub>2</sub>,..., A<sub>p</sub>,... Supposons par exemple que le programme comporte un ordre de lecture et que les cartes à lire contiennent chacune une valeur A. Il est impossible de former l'ordre PP' N iA<sub>p</sub> jB kC<sub>p</sub> par substitution d'adresse ou par addition car si l'une de ces opérations est valable en logique extérieure elle ne le sera pas en exécution en langage machine et vice versa. La solution la plus simple est de procéder par indexage.

- Si l'adresse A n'est pas indexée, il suffit d'envoyer A dans les positions l à 4 d'un registre index inutilisé que nous noterons n° i, et d'écrire l'ordre sous la forme : PP' N i 0000 jB kC.
- Si l'adresse A est indexée (par un index que nous noterons i'), deux cas peuvent se présenter :
- I°) Si l'index i' n'indexe pas d'autre adresse que A, il suffit d'ajouter A à la valeur de cet index et d'écrire l'ordre : PP' N i' 0000 jB kC. Notons que lorsque l'index i' est remis à zéro et doit encore ultérieurement être utilisé pour former l'ordre PP' N iA jB kC, il faut y entrer à nouveau A p

2°) Si l'index i' indexe d'autres adresses que A, il faut envoyer A, + le contenu du registre index n° i' dans un registre index n° i'' inutilisé et écrire l'ordre: PP' N i'' 0000 jB kC. Mais ceci oblige à faire cette opération à chaque modification du registre d'index i'.

#### XIII- C - INDEXAGE DOUBLE.

Il est très aisé de faire des indexages doubles par addition d'index.

#### XIII- D -AIGUILLAGE MULTIPLE.

we see to be set with the second

Pour faire un aiguillage multiple c'est-à-dire créer un ordre "Aller à V", V étant variable, il suffit de créer dans une mémoire quelconque l'ordre - 00 0000 V, V ayant la valeur désirée, puis de s'adresser à cette mémoire. Cet ordre doit être négatif pour être valable aussi bien en logique extérieure (- 00 0000 V est considéré comme un ordre en C. D. P.) qu'en exécution en langage machine (- 00 0000 V est considéré comme un ordre en langage machine et son signe est ignoré).

XIV. CONCLUSION

Le C. D. P., méthode de programmation intermédiaire entre le FLAIR (logique extérieure où les ordres logiques sont très réduits) et le FORTRAN ou 1' A P 3 adaptés aux plus grosses machines, possède en fait maints avantages de chacune des 2 méthodes : comme le FORTRAN il effectue une autoprogrammation et comporte des ordres comparables aux ordres SI, FAIRE, ALLER A,... Mais il a aussi la commodité d'utilisation du FLAIR, et en particulier sa facilité de mise au point.

Le C. D. P. est constitué d'un noyau formé par les opérations en virgule fixe, les ordres logiques et de routine, et les ordres d'entrée dans des
sous-programmes. C'est ce noyau qui, de plus, effectue l'indexage et éventuellement l'analyse. L'adjonction aisée de sous-programmes à ce noyau donne une
grande souplesse au C. D. P. Le nombre des sous-programmes que l'on peut
lui adjoindre simultanément n'est en fait limité que par la capacité du tambour.

On a jusqu'à présent formé trois C. D. P., d'utilisation fréquente:
C. D. P. en virgule fixe, en virgule flottante simple précision, en virgule flottante double précision. A ces C. D. P. peuvent être facilement ajoutés des sousprogrammes utiles dans des problèmes particuliers.

Grâce à la souplesse du C. D. P., on peut aussi, en ajoutant à son noyau des sous-programmes particuliers, former des C. D. P. spécialisés (C. D. P. matriciel, en cours de création; C. D. P. adapté aux calculs d'impédances et aux calculs en nombres complexes, transposition du FLEC; etc...).

Le C. D. P. a ainsi de grandes possibilités d'extension. Il sera vraisemblablement aisé de l'adapter aux ordinateurs I. B. M. 650 à registres

d'index, virgule flottante, bandes et par la suite à d'autres matériels. On pourra même probablement, après avoir mis au point en logique extérieure des programmes sur la 650 I. B. M., effectuer sur cette même machine une autoprogrammation modifiée qui créera des ordres exécutables sur une autre machine.

Mais dès à présent, grâce à la co-existence de la logique extérieure et de l'autoprogrammation, grâce aussi aux ordres logiques (le bouclage en particulier), le C. D. P. est un outil de travail très commode.

-0-0-0-0-0-0-0-0-0-

state or the or think a

101 Fig. 12. - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1

e de la Maria de la Caracteria de la compansión de la compansión de la compansión de la compansión de la compa La compansión de la compansión de la compansión de la compansión de la compansión de la compansión de la compa

and the second s

 INTERPRETATIFS

========

des PROGRAMMES

1		PRA	1639	1670				
2		PRA	0000	1638				
3		PEN	PPPRI	1750				
4		PEN	ANALY	1700				
5		PEN	001	1764				
6		PEN	PFDEB	1738				
7		PRA	1940	1999				
8		PRG	H1941	1949				
9		PRG	01968	1976				
10		PRG	P1977	1986				
1 1		PRG	W1991	1999				
12		PEN	F0000	1846				
13		PRG	F1847	1855				
14		PEN	G0000	1856				
15		PRG	G1857	1865				
		PEN		1866				
16		PRG	T 0 0 0 0	1883				
1 7		PEN	T 1 8 6 7					
18		PEN	MEMNN	1884 1885				
19		PRG	00000	1939				
20		PER	D1886	1875				
			DEL					
22	ANALY	PEN	DEBUT	1800	1700	60	0000	1707
23	ANALY	RAG	8000	0.50.17	1700	60	8000	1707
2 4		ANG	ANA	DEBUT	1707	46	1710	1800
25	ANA	RAG	P0005		1710	60	1981	1685
26		AAD	P0006		1685	15	1982	1687
27		DAG	0004		1687	35	4	1697
28		AAG	EDOKM	8003	1697	1 0	1701	8003
29	KM	TDI	P0005		1751	24	1981	1684
30		AAD	EDOKN	8002	1684	15	1737	8002
31	KN	TDI	P0006		1801	24	1982	1735
32	CHANA	RAD	P0007	SUANA	1735	65	1983	1787
	SUANA	EDI	EDOKP		1787	69	1690	1693
34		SBF	EDOKP	8001	1693	22	1690	8001
35	KP	TDI	P0007		1702	24	1983	1686
36		AAD	010		1686	1 5	1689	1743
37		EDI	EDOKR	12.50	1743	69	1696	1699
38		SBF	EDOKR	8001	1699	22	1696	8001
39	KR	TDI	P0008	PFDEB	1752	2 4	1984	1738
40	PFDEB	PFO	P0001	DEBUT	1738	7 1	1977	1800
4 1	DEBUT	RAD	MEMNN		1800	65	1884	1739
42		DAG	0004		1739	35	4	1749
43		TRD	P0001		1749	20	1977	1680
44		EDI	GG	G I	1680	69	1683	1736
45	GG	RAD	0000	GН	1683	65		1705
46	GI	SBF	GG	8001	1736	22	1683	8001

47	GH	ANG	UNEME	DEUME	1705	46	1708	1709
48	DEUME	TRD	P0002		1709	20	1978	1681
49		EDI	8003		1681	69	8003	1688
50		SBI	P0005		1688	23	1981	1734
51		SAD	8001		1734	16	8001	1691
52		DAG	0001		1691	35	1	1747
53		AAG	CTE1		1747	10	1802	1757
54		TRG	W0002		1757	21	1992	1695
55		SAG	CTE1		1695	1 1	1802	1807
56		DAG	0001		1807	35	1	1713
5 7		SAG	8003		1713	1 1	8003	1671
58		TDI	PPPRI		1671	2 4	1750	1703
59		DAG	0003		1703	35	3	1711
60		TRG	MEMNN		1711	21	1884	1837
61		TRO	W0003		1837	20	1993	1746
62		RAD	P0001		1746	65	1977	1731
63		AAD	GJ	8002	1731	15	1784	8002
64	GJ	RAV	0001	GL	1784	67	1	1755
65	GL	TDI	P0003		1755	2 4	1979	1682
66		EDI	8003		1682	69	8003	1788
67		SBI	P0008		1788	23	1984	1838
68		SAD	8001		1838	16	8001	
69		DAG	0001		1745	35		1745
70		SAG	8003				1	1753
7 1		TDI			1753	1 1	8003	1761
72			W0005	7	1761	2 4	1995	1698
	T 5151 T 51	DAG	0004	INNIN	1698	35	4	1759
73	INNIN	TRG	P0006	INCHA	1759	2 1	1982	1785
7 4	NINCH	RAG	P0008	COMMU	1803	60	1984	1789
75	COMMU	DAG	0004		1789	35	4	1799
76		AAG	OOANA		1799	10	1704	1809
77		TRG	P0007	W0002	1809	21	1983	1992
78	G0009	SOP	0000	00051	1865	00		1936
79	F0001	RAD	ONICO	ADSO1	1847	65	1754	1760
80	F0002	RAD	OONZO	ADSO1	1848	65	1804	1760
8 1	ADSO1	AAD	P0006		1760	15	1982	1839
82		DAG	0004		1839	35	4	1805
83		AAD	0 0 W 3		1805	15	1758	1763
8 4		TRD	W0002		1763	20	1992	1795
85		RAG	P0005		1795	60	1981	1835
86		DAG	0004		1835	35		
8 7		AAD	TGOO	KG	1845		4 7 ( 5	1845
88	KG	AAG	RGOW2	V Q		1 5	1748	1706
89		AAD	P0007		1706	10	1810	1715
90		TRO		0.0.0.0	1715	15	1983	1740
91	F0007		W0003	8003	1740	20	1993	8003
9 5	F0003	TRD	W O O O 1		1849	20	1991	1694
		RAG	P0006		1694	60	1982	1790
93		DAG	0004		1790	35	4	1756
94		AAG	MLOW1		1756	10	1811	1765
95		TRG	W0002		1765	21	1992	1796
96		RAD	02KF	K I	1796	65	1806	1712
97	K I	EDI	W O O O 1		1712	69	1991	1744

98		SBI	W0001		1744	23	1991	1794
99		RAG	P0005		1794	60	1981	1786
100		DAG	0004		1786	35	4	1797
101		AAD	TDOO	KG	1797	15	1808	1706
102	KF	GNN	ARETM	W0003	1762	44	1815	1993
103	F0005	RAD	P0001		1851	65	1977	1781
104	F 0 0 0 0	AAD	, 0001	8002	1781	15	1834	8002
		RAD	0002	F0003	1834	65	2	1849
105				10003	1850	20	1991	1844
106	F0004	TRD	W O O O 1		1844	65	1982	1840
107		RAD	P0006		1840	35	4	1812
108		DAG	0004				1716	1719
109		EDI	KH		1812	69		1769
110		SBF	KH		1719	22	1716	
1 1 1		EDI	RDAW1		1769	69	1672	1675
112		TDI	M0005		1675	2 4	1992	1798
113		RAD	0 2 K H	K I	1798	65	1813	1712
114	F0006	RAD	P0001		1852	65	1977	1831
115		AAD		8002	1831	1 5	1836	8002
116		RAD	0002	F0004	1836	65	2	1850
117	F0007	RAD		F7F8	1853	65	1714	1819
118		46	ANA78		1714	46	1717	1718
119		GNN	EDB	EDA	1718	44	1721	1722
120	EDB	EDI	P0006	EDBA	1721	69	1982	1741
121	EDA	EDI	P0005	EDBA	1722	69	1981	1741
122	EDBA	TDI	MEMNN	ANA78	1741	2 4	1884	1717
123	F 7 F 8	TRD	W0002	AITA	1819	20	1992	1814
	F / F O	RAD	P0008		1814	65	1984	1791
124					1791	35	4	1766
125		DAG	0004	0000	1766	15	1810	8002
126		AAD	RGOW2	8002			8000	1725
127	ANA78	RAG	8000	a = B · i T	1717	60		
128	WWW.	ANG	KN	DEBUT	1725	46	1801	1800
129	F0008	RAG	P0006		1854	60	1982	1841
130		DAD	0002		1841	30	2	1816
131		AAD		F7F8	1816	1 5	1720	1819
132		90	ANA78	EDA	1720	90	1717	1722
133	G O O O 1	RAG	PPPRI	GCOMU	1857	60	1750	1767
134	G0002	RAG	PPPRI	GCOMU	1858	60	1750	1767
135	60003	RAG	PPPRI	GCOMU	1859	60	1750	1767
136	G O O O 4	RAG	PPPRI	GCOMU	1860	60	1750	1767
137	G0005	RAG	PPPRI	GCOMU	1861	60	1750	1767
138	G0006	RAG	PPPRI	GCOMU	1862	60	1750	1767
139	G0007	RAG	PPPRI	GCOMU	1863	60	1750	1767
140	GCOMU	DAG	0004		1767	35	4	1677
1 4 1		AAG	ADDEL	8003	1677	10	1730	8003
142	КJ	EDI	EDP7		1817	69	1770	1673
143	,, 0	SBI	EDP7		1673	23	1770	1723
144		RAD	P0005		1723	65	1981	1692
		DAG			1692	35	4	1768
145			0004		1768	15	1982	1742
146		AAD	P0006		1742			1792
1 4 7		AAG	P0007	E D D 7	1792	10	1983	1770
1 4 8		AAG	TDOO	EDP7		1 0	1808	
1 4 9	G0000	RAG	PPPRI		1856	60	1750	1818

150		AAG	CTE2	8003	1818	10	1771	8003
151	F0000	AAD	P0006		1846	15	1982	1842
152		DAG	0004		1842	35	4	1820
153		AAD	P0005		1820	15	1981	1793
154		EDI	KE		1793	69	1821	1674
155		SBI	ΚΞ		1674	23	1821	1724
156		RAG	8002		1724	60	8002	1733
157		AAG	RD01		1733	10	1843	1772
158		AAD	TDOO	KE	1772	15	1808	1821
159	CTE1	00	0000	G0000	1802	00	1000	1856
160	010	00	0001	0000	1689	0 0	1	00
161	0080	0.0	0001		1822		1	
162	OOANA	0.0	0000	80		00		80
		00	0000	ANALY	1704	00		1700
163	001			1	1764	00		1
164	ONIXO	00	0010	0000	1754	00	1 0	
165	00 W 3	00	0000	W0003	1758	00		1993
166	TGOO	2 1	0 0		1748	21		
167	RGOW2	60	0000	M0005	1810	60		1992
168	MLOW1	19	0000	W O O O 1	1 8 1 1	1 9		1991
169	02KF	00	0002	KF	1806	00	2	1762
170	002MI	00	0000	4000	1773	00		4000
171	TDOO	20	0		1808	20		
172	ARETM	0 1	1333	W0003	1815	01	1333	1993
173	KH	6 4	0000	W0003	1716	6 4		1993
174	RDAW 1	65	8003	W0001	1672	65	8003	1991
175	ADDEL	1 5	DEL	KJ	1730	15	1875	1817
176	EDP7	1 1	8003	0000	1770	1 1	8003	00
177	CTE2	00	0000	F0000	1771	00		1846
178	KE	1 5	P0007	0000	1821	15	1983	000
179	RD01	65	0000	0001	1843	65		1
180	EDOKM	69	0000	KM	1701	69		1751
181	EDOKN	69	0000	KN	1737	69		1801
182	EDOKP	69	0000	KP	1690	69		1702
183	02KH	00	0002	KH	1813	00	2	1716
184	OONZO	00	0011	0000	1804	00	1 1	001
185	EDOKR	69	0000	KR	1696	69	7 1	
				KK			1070	1752
186	UNEME	TRD	P0002		1708	20	1978	1732
187		RAG	8000		1732	60	8000	1823
188		ANG		UNES	1823	46	1676	1727
189		PFO	P0001	UNES	1676	7 1	1977	1727
190	UNES	RAV	P0002		1727	67	1978	1783
191		EDI	8003		1783	69	8003	1774
192		SBI	MEMNN		1774	23	1884	1824
193		DAG	0002		1824	35	2	1782
194		AAG	CTE4	8003	1782	1 0	1775	3003
195	T0005	DAD	0006	8002	1871	30	6	9002
196	T0007	EDI	LCDEB	DADSB	1873	69	1726	1679

197	DADSB	DAD	0002	SBFW1	1	679	30	2	1825
198	SBFW1	SBF	W0001	8001	1	825	22	1991	8001
199	T0001	EDI	ARDEB	DADSB	1	867	69	1776	1679
200	T0009	SAG	8003		1	875	1 1	8003	1833
201		DAD	0006		1	833	30	6	1826
202		TRO	MEMNN	DEBUT	1	826	20	1884	1800
203	CTE4	00	0000	T0000	1	775	00		1866
204	LCDEB	70	0000	DEBUT	1	726	70		1800
205	ARDEB	0 1	0000	DEBUT	1	776	01		1800
206	T0002	RAG		8003		868	60	1777	8003
207		TRD	1976	INIT1	1	777	20	1976	1729
208	INIT1	SAG	INIT2		1	729	1 1	1832	1827
209		GNN		DEBUT	1	827	44	1678	1800
210		AAG		8003	1	678	10	1728	5003
211		TRD	1939	INIT1	1	728	20	1939	1729
212	INIT2	TRD	1940	INIT1	1	832	20	1940	1729
213	T0006	EDI	PFDEB	DADSB	1	872	69	1738	1679
214	T0003	SAG	8003		1	869	1 1	8003	1778
215		EDI		DADSB	1	778	69	1828	1679
216		TRG	0000	DEBUT	1	828	21		1800
217	T0000	SOP	0000	DEBUT	1	866	00		1800
218	T 0 0 0 4	DAD	0002		1	870	30	2	1779
219		EDI	RGOW2		1	779	69	1810	1829
220		SBF	W O O O 1		1	829	22	1991	1780
221		EDI	ARDEB		1	780	69	1776	1830
222		SBF	W0002	W0001	1	830	22	1992	1991

	INDEX	LO	GIQUE	EXT				
224		PBD	1639	1670				
225		PRA	1671	1999				
226	INCHA	TRD	W0007		1785	2.0	1997	1650
227		RAG	PPPR I		1650	60	1750	1655
228		SAG	0080		1655	1 1	1822	1639
229		ANG	INF	W0002	1639	46	1642	1992
230	INF	RAD	W0003		1642	65	1993	1647
231		DAD	0005		1647	30	5	1659
232		AAD	RDCKA	8002	1659	15	1662	8002
233	KA	AAG	8001		1651	1 0	8001	1660
234		AAG	P0005		1660	10	1981	1640
235		TRG	P0005		1640	21	1981	1641
236		DAG	0003		1641	35	3	1649
237		TRD	P0004		1649	20	1980	1643
238		RAD	W0005		1643	65	1995	1652
239		DAG	0004		1652	35	4	1663
240		AAD	RDCKB	8002	1663	15	1666	8002
241	KB	AAG	8001		1653	10	8001	1661
242		AAG	P0006		1661	1 0	1982	1644
243		TRG	P0006		1644	21	1982	1645
244		AAD	P0004		1645	15	1980	1646
245		DAG	0003		1646	35	3	1656
246		TRD	P0004		1656	20	1980	1648
247		RAD	W0007		1648	65	1997	1654
248		DAD	0005		1654	30	5	1667
249		AAD	RDCKC	8002	1667	15	1670	8002
250	KC	AAG	8001		1657	10	8001	1665
251		AAG	P0008		1665	10	1984	1658
252		AAD	P0004		1658	15	1980	1664
253		SAD	8002		1664	16	8002	1668
254		TDI	P0004		1668	2 4	1980	1669
255		TRG	P0008	COMMU	1669	21	1984	1789
256	ROCKA	RAD	00000	KA	1662	65	1967	1651
257	RDCKB	RAD	00000	K B	1666	65	1967	1653
258	RDCKC	RAD	00000	KC	1670	65	1967	1657

	259		PRA	1500	1999				
1	260		PBD	0001	0042				
	261		PEN	BOU7	0012				
	262		PEN	BOUS	0013	5707 57			
4 4 (1 t)	263	0000	RAD	W0003			65	1993	0001
4 4	264		DAD	0005		1	30	5	0014
**	265		AAD	OHOO		1 4	15	17	0021
	266		EDI	POU2		21	69	2 4	0027
	267		SBF	Bouz		27	22	24	0028
	268		EDI	BOU4		28	69	3 1	0034
	269		SBF	BOU4		3 4	22	31	0035
	270		EDI	BOU9		35	69	38	0041
	271		SBF	BOU9		4 1	22	38	0042
	272		RAD	PPPRI		42	65	1750	0005
	273		SAD	0090		5	16	8	0015
	274		ANN	BOU1		15	45	18	0019
	275		AAD	OCENO	B 0 U 1	0019	15	22	0018
	276	BOU1	AAD	0000		18	1 5	23	0029
	277		DAG	0004		29	35	4	0039
	278		EDI	BOUS		39	69	43	0002
	279		SBF	BOU5		2	22	43	0003
	280		EDI	8007		3	69	12	0016
	281		SBF	BOU7		1.6	22	12	0020
	282		EDI	B 0 U 1 1		2.0	69	25	0030
	283		SBF	B O U 1 1		30	22	25	0032
	284		AAG	B0U6		32	10	36	0004
	285		TRG	W0006	B 0 U 2	0004	21	1996	0024
	286	B 0 U 2	RAG	0000	B0U13	0024	60		6
	287	BOU3	SAG	P0005		7	1 1	1981	0037
	288		ANG	BOU5	B 0 U 1 1	0037	45	43	0025
	289	B 0 U 1 3	AAG	001	B 0 U 4	0006	10	1764	0031
	290	B 0 U 4	TRG	00		3 1	21		9
OF CH	291		TDI	P0004	B0U3	0009	24	1980	0007
4	292	B O U 5	RAG	0000	W0006	0043	60		1996
the gray o	293	8006	AAG	P0008	B O U 7	0036	10	1984	0012
	294	BOU7	TRG	00		12	21		10
	295		TDI	P0005	8008	0010	24	1981	0013
	296	BOU8	EDI	P0006		13	69	1982	0040
	297		TDI	WEWNN	ANABO	0040	24	1884	0011
	298	B O U 1 1	TRD	0000	B 0 U 9	0025	20		38
	299	BOU9	TDI	0000	ANABO	0038	24		1 1
	300	ANABO	RAG	8000		1 1	60	8000	0026
	301	dollar to	ANG	PFDEB	DEBUT	0026	46	1738	1800
	302	0 H O O	00	HOO		1 7	00	1940	0.0
	303	0090		-11 July 10 PO	90	0008	00		90
	304	OCENO	0 0	0100		22	00	100	
	305	0000	0		00000	0023	00		19€7

1		PRA	0000	1 3 5 1				
2		PRA	1940	1999				
3		PRG	H1941	1949				
4		PRG	C1968	1976				
5		PRG	P1977	1986				
6		PRG	W1991	1999				
7		PRG	X1838	1846				
8		PRG	F1847	1855				
9		PEN	G0000	1856				
10		PRG	G1857	1865				
11		PEN	T0000	1866				
12		PRG	T1867	1881				
13		PEN	CTRL	1882				
14		PEN	LMAX	1883				
15		PEN	MEMNN	1884				
16		PEN	00000	1885				
17		PRG	D1886	1939				
18		PEN	DEBUT	1800				
19		PER	K	1930				
20		PER	KPLU2	1932				
21		PER	ARETM	1936				
22		PER	Z	1939				
23	DEBUT		MEMNN		1800	65	1884	1389
24	0200	DAG	0004		1389	35	4	1399
25		TRD	MEMN	DEBUZ	1399	20	1403	1356
26	DEBU2	TRG	P0003		1356	21	1979	1382
27		TDI	P0004		1382	2 4	1980	1383
28		TOI	P0005		1383	2 4	1981	1384
29		TDI	P0006		1384	24	1982	1385
30		TDI	P0007		1385	2 4	1983	1386
31		TDI	P0008		1386	2 4	1984	1387
32		EDI	FF	FG	1387	69	1390	1393
33	FF	RAD	0000	FH	1390	65		1355
34	FG	SBF	FF	8001	1393	22	1390	8001
35	FH	ANG	UNEME	DEUME	1355	46	1358	1359
36	DEUME	EDI	8003	0 2 0 1 , 2	1359	69	8003	1366
37	D L O I I L	SBI	00001		1366	23	1968	1371
38		SAD	8001		1371	16	8001	1379
39		DAG	0001		1379	35	1	1 4 3 5
40		AAG	CTE1		1 4 3 5	10	1388	1 4 4 3
41		TRG	W0005		1 4 4 3	21	1992	1395
42		SAG	CTE1		1395	1 1	1388	1 4 9 3
43		DAG	0001		1 4 9 3	35	1	1 4 4 9
44		SAG	8003		1 4 4 9	1 1	8003	1357
4 4		TDI	PPPRI		1357	24	1360	1363
45		DAG	0003		1363	35	3	1 4 2 1
47		TRG	MEMNN		1 4 2 1	21	1884	1 4 3 7
48		TRD	C O O O 4		1 4 3 7	20	1971	1374
+0		1 10	00004		2 -7 -7 /	20	- / 1 1	

49		RAD	MEMN		1374	65	1403	1407
50		AAD	002		1 4 0 7	15	1410	1365
51		TRD	P0001		1 3 6 5	20	1977	1380
52		AAD	010		1380	15	1433	1 4 8 7
53		EDI	FJ	FK	1 4 8 7	69	1440	1543
54	FJ	RAV	0000	FL	1 4 4 0		1440	
55	FK	SBF	FJ	8001		67	4 4 4 6	1 4 0 5
_				8001	1543	22	1 4 4 0	8001
56	FL	EDI	8003		1 4 0 5	69	8003	1362
57		SBI	00003		1 3 6 2	23	1970	1373
58		SAD	8001		1 3 7 3	16	8001	1381
59		DAD	0001		1 381	30	1	1537
60		EDI	8003		1537	69	8003	1394
61		SBF	00005		1 3 9 4	25	1969	1372
62		SAD	8001		1372	16	8001	1429
63		DAD	0003		1 4 2 9	30	3	1587
64		EDI	8003		1587	69	8003	1444
65		SBI	C0005		1 4 4 4	23	1972	1375
66		DAD	0001		1375	30	1	1431
67		SBF	00005		1 4 3 1	22	1972	1425
68		RAG	00003		1 4 2 5	60	1970	1 4 7 5
69		DAG	0004		1 4 7 5	35	4	1 4 8 5
70		AAG	MEMNN		1 4 8 5	10	1884	1439
7 1		TRG	00006	W0002	1 4 3 9	21	1973	1992
72	PERFO	PFO	P0001		1 400	7 1	1977	1377
73		AAG	CTRL		1377	10	1882	1637
74		DAG	0004		1637	35	4	1397
75		TRG	P0001		1397	21	1977	1430
76		AAG	8002		1 4 3 0	10		
77		DAD					8002	1489
			0004		1 4 8 9	30	. 4	1 4 9 9
78		TRG	CTRL		1 4 9 9	21	1882	1535
79		SAG	LMAX		1535	1 1	1883	1687
80		SAG	002	SUITE	1687	1 1	1 4 1 0	1415
8 1	SUITE	EDI	W O O O 1		1 4 1 5	69	1991	1494
8 2		TDI	P0002		1 4 9 4	24	1978	1481
83		EDI	W0005		1 4 8 1	69	1992	1 4 4 5
8 4		TDI	P0003		1 4 4 5	2 4	1979	1 4 3 2
85		EDI	W0003		1432	69	1993	1396
86		TDI	P0004		1396	24	1980	1483
8 7		ANG	FINPF	ARETL	1 483	46	1436	1737
88	FINPF	PFO	P0001	DEBUT	1 4 3 6	7 1	1977	1800
08	G0009	RAG	CTRL		1865	60	1882	1438
09		DAG	0004		1 4 3 8	35	4	1599
10		AAG	8001		1599	10	8001	1507
11		AAG	RGO1		1507	10	1510	1565
. 12		TRG	P0002		1565	21	1978	1581
13		AAD	MEMN		1581	15	1403	1557
14	100	AAD	D0051		1557	15	1936	1 491
15		AAD	ED10		1 4 9 1	15	1644	1649
16		TRD	W0002		1649	20	1992	1545

117		DAD	0006		1545	30	6	1409
118		SBF	W0001		1 409	22	1991	1694
119		EDI	00006		1694	69	1973	1426
120		TDI	W0003		1 4 2 6	24	1993	1496
121		RAG	PPPRI		1 4 9 6	60	1360	1615
122		DAD	0002		1615	30	2	1571
123		AAD	00002		1571	15	1969	1473
124		DAD	0003		1 4 7 3	30	3	1631
125		AAD	00004		1631	15	1971	1575
126		DAD	0001		1575	30	1	1681
127		AAD	MCTRO		1681	15	1486	1541
128		TRO	P0005		1541	20	1981	1 4 3 4
129		EDI	CO 0 0 1		1 4 3 4	69	1968	1621
130		TDI	P0003	VAPF4	1621	24	1979	1482
131	G0000	RAD	PPPRI	, , , , , ,	1856	65	1360	1665
132	00000	ANN	VFX	SPZA	1665	45	1368	1369
133	VFX	AAG	00005	0. 2.1	1368	10	1972	1 4 2 7
134		AAG	- 00004		1 4 2 7	1 0	1971	1625
135		GNN	IND1	NIND1	1625	44	1479	1 4 8 0
136	IND1	AAD	CTE2	14 7 14 10 1	1479	15	1532	1 4 8 8
137	1 14 5 1	TRD	W 0 0 0 4		1 488	20	1994	1 4 9 7
138		TRG	W 0 0 0 1		1497	21	1991	1744
139		RAD	CTRL		1744		1882	1538
140		DAG	0004		1538		1 602	1699
141		AAD	D0052		1699			
142		AAG	8001		1591	10	1937	1591
143		AAG	MEMN		1749			
144		AAG			1607	1 0	1403	1607
1 4 5		TRG	RD10			1 0	1560	1715
146		AAD	P0002		1715	21	1978	1731
147			6013			15	1484	1539
148		TRD	P0003	W O O O 4	1539	20	1979	1582
149	F0001	RAD	00002		1582	65	1969	1994
		AAD	A G O W 2	ADSO1	1847	15	1 4 5 0	1 4 5 5
150	F0002 ADS01	AAD	SGOW2	ADSO1	1848	1 5	1401	1 4 5 5
152	AUSUI	RAD	W O O O O O		1 4 5 5	20	1993	1546
153		DAG	0004		1546	65	1968	1523
154		AAD	RGOW1		1523	35	4	1583
155		TRD			1583	15	1536	1641
156			P0005	4.0.411.4	1641	20	1981	1534
157	ASMU1	AAG	TGOO	ASMU1	1534	10	1588	1593
158	ASMUI	AAG	00006		1593	1 0	1973	1 4 7 7
	50005	TRG	M O O O S	VAPF4	1 4 7 7	2.1	1992	1482
159	F0003	AAD	MULOZ		1849	1 5	1402	1657
160		TRD	W0003	REUN1	1657	50	1993	1596
161	REUN1	RAG	_C O O O 1		1596	60	1968	1573
162		DA,G	0004		1573	35	4	1633
163		AAG	R.GOW1	MUDI1	1633	10	1536	1691
164	MUDI1	TRG	P0005		1691	21	1981	1584
165	1.0	RAG	TDOO	ASMU1	1584	60	1638	1593
166	F0005	AAD	MULOZ		1851	15	1402	1707

167		TRD	W0003		1707	20	1993	1646
168		RAD	P0001		1646	65	1977	1781
169		AAD	001		1781	15	1634	1589
170		TRD	P0001		1589	20	1977	1530
171		DAD	0004		1530	30	4	1741
172		AAD	002		1741	15	1410	1765
173		EDI	W0003		1765	69	1993	1696
174		SBI	W0003		1696	23	1993	1746
175		DAG	0004		1746	35	4	1757
176		AAD	EDOGA	8002	1757	15	1610	8002
177	G A	TDI	P0004	REUN1	1500	24	1980	1596
178	F0004	AAD	DROW2	11 2 0 11 2	1850	15	1 4 5 3	1807
179	F 0 0 0 4	TRD	W0003		1807	20	1993	1796
180		RAG	00001		1796	60	1968	1623
		DAG	0004		1623	35	4	1683
181		AAG	0 0 W 1	REUN2	1683	1 0	1586	1791
182	OFUNG	AAG	RDOO	MUDI1	1791	10	1794	1691
183	REUN2	AAD	DROW2	MODII	1852	15	1453	1408
184	F0006				1 4 0 8	20	1993	1547
185		TRD	W 0 0 0 3		1547		1977	1831
186		RAD			1831	15	1634	1639
187		AAD	001		1639	20	1977	1580
188		TRD	P0001		1580	15	1733	1688
189		AAD	020				1392	1595
190		EDI	EDOGB	0.004	1688	69	1392	8001
191	Service Control	SBF	EDOGB	8001	1595		1980	1783
192	G B	TDI	P0004		1550	24	-	1643
193		DAO	0004		1783	30	6	1458
194		DAG	0006		1643	35		
195		AAG	C O O O 1	5511110	1 4 5 8	1 0	1968	1673
196	and the second	DAG	0004	REUN2	1673	35	1050	1723
197	F0007	AAD	C O O O 1		1853	15	1968	1632
198		AAG	4400		1723	10	1476	
199		TRD	P0005		1632	20	1981	1684
200		RAD	WEWNN		1684	65	1884	1689
201		AAG	00003		1689	1 0	1970	1675
202		DAG	0004		1675	35	4 7 7 6	1585
203		AAG	RGOO		1585	10	1738	1693
204		AAD	CTRL		1693	15	1882	1788
205		AAD	4603		1788	15	1442	1597
206		EDI		F7F8	1597	69	1600	1503
207		TRD		VAPF4	1600		1991	1482
208	4400	4 4	00		1 4 7 6			
209	4603	4 6	0000	0003	1 4 4 2			3
210	F0008	AAD	MEMNN		1854	1 5	1884	1739
211		AAG	00003		1739		1970	1725
212		DAG	0004		1725		4	1635
213		AAG	69		1635		1594	1799
214		AAD	C O O O 1		1799		1968	1773
215		AAD	900		1773	15	1526	1682

216		EDI		F7F8	1682	69	1685	1503
217		TRD	W0001	VAPF3	1685	20	1991	1645
218	900	90	0		1526	90		
219	F7F8	TDI	W0005		1503	24	1995	1398
220		TRD	W0003		1398	20	1993	1647
221		AAG	CTRL		1647	10	1882	1789
222		AAG	002		1789	1 0	1410	1815
223		TRG	W0002		1815	21	1992	1695
224		RAD	MEMN		1695	65	1403	1508
225		DAG	0002		1508	35	2	1416
226		AAG	00005		1416	10	1972	1527
227		AAD	CTRL		1527	15	1882	1490
228		EDI	DAG4		1490	69	1743	1697
229		SBI	P0003	DAG4	1697	23	1979	1743
230	DAG4	DAG	0004	D/CG.	1743	35	4	1553
231		AAG	RGC01		1553	1 0	1406	1361
232		TRG	P0002		1361	21	1978	1732
233		AAD	-0002	W0005	1732	15	1735	1995
234		1 0	0001	8003	1735	10	1 / 55	8003
235	RGC01	60	0000	0001	1 40 6	60	1967	
236	NINDI	AAD	CTE3	0001	1 4 8 0	15	1833	0001
237	111101	TRD	W O O O 4		1540	20	1994	
238		RAG	00001		1747	60	1968	1747
239		DAD	0002		1823	30		1823
240		AAD	MEMN		1529		2	1529
241		DAG				15	1403	1558
241		AAG	0006		1558	35	6	1 4 2 4
243			001	140004	1 4 2 4	1 0	1634	1590
	V 0 0 0 1	AAD	00002	W O O O 4	1590	15	1969	1994
244	X0001	AAD	AGOO	ADS02	1838	15	1492	1797
245	X0002	AAD	SGOO	ADS02	1839	15	1542	1797
246	ADS02	AAD	CTRL		1797	15	1882	1640
247		TRD	P0003		1640	20	1979	1782
248		AAG	RGOO		1782	10	1738	1793
249		TRG	P0002		1793	21	1978	1832
250	A C D	RAG	TGOO	ASDIV	1832	60	1588	1745
251	ASDIV	AAG	00006		1745	1 0	1973	1577
252		TRG	W0001	VAPF1	1577	2 1	1991	1795
253	X0003	AAG	RGOO		1840	1 0	1738	1 4 4 8
254		TRG	P0002		1 4 4 6	21	1978	1734
255		RAG	8002		1734	60	8002	1498
256		AAG	MULOZ		1 4 9 8	1 0	1 4 0 2	1608
257		AAD	CTRL		1608	15	1882	1690
258		AAD	001		1690	15	1634	1740
259			8003	REUNS	1740	69	8003	1548
260	REUN3	SBI	P0003		1548	23	1979	1784
261		RAG	CTRL		1784	60	1882	1790
262		AAG	GNARM		1790	1 0	1598	1603
263		TRG	W0002		1603	21	1992	1648
264		AAD	00006		1648	1 5	1973	1627

265		AAD	TDOO		1627	15	1638	1698
266		TRD	W O O O 1	VAPF2	1698	20	1991	1748
267	X0005	AAG	RGOO		1842	1 0	1738	1798
268		TRG	P0002		1798	21	1978	1834
269		RAG	8002		1834	60	8002	1650
270		AAD	P0001		1650	15	1977	1785
271		AAD	001		1785	15	1634	1592
272		TRD	P0001		1592	20	1977	1630
273		AAD	020		1630	15	1733	1642
274		AAD	CTRL		1642	15	1882	1692
275		SAD	002		1692	16	1410	1 4 6 6
276		EDI	ED2GC		1 4 6 6	69	1419	1422
277		SBF	ED2GC	8001	1 4 2 2	22	1419	8001
278	GC	SBI	P0004		1700	23	1980	1835
279		AAG	MULOZ		1835	10	1402	1658
280		EDI	8003		1658	69	8003	1364
281		DAD	0004	REUN3	1364	30	4	1548
282	X0004	AAD	DROO		1841	15	1750	1505
283		AAD	CTRL		1505	15	1882	1742
284		TRD	P0003		1742	20	1979	1636
285		AAG	RDOO		1636	10	1794	1 4 5 1
286		TRG	P0002	REUN4	1 4 5 1	21	1978	1686
287	REUN 4	RAG	TDOO	ASDIV	1686	60	1638	1745
288	X0006	AAD	DROO		1843	15	1750	1555
289		AAD	CTRL		1555	15	1882	1792
290		SAD	8002		1792	16	8002	1501
291		TDI	P0003		1501	24	1979	1736
292		AAD	P0001		1736	15	1977	1786
293		AAD	001		1786	15	1634	1551
294		TRD	P0001		1551	20	1977	1680
295		AAD	020		1680	1 5	1733	1601
296		EDI	EDOGD		1601	69	1354	1708
297		SBF	EDOGD		1708	55	1354	1758
298		RAD	8003	EDOGD	1758	65	8003	1354
299	G D	SBI	P0004		1651	23	1980	1836
300		AAD	RDO1		1836	15	1701	1605
301		TRD	P0002	REUN4	1605	20	1978	1686
302	X0007	AAD	00001		1844	15	1968	1 4 7 4
303		AAD	4400		1 4 7 4	15	1476	1751
304		TRD	W 0 0 0 1		1751	20	1991	1801
305		RAD	MEMN		1801	65	1403	1808
306		DAG	0002		1808	35	2	1516
307		AAG	00003		1516	10	1970	1775
308		AAD	MEMNN		1775	15	1884	1 4 5 2
309		DAG	0004		1 4 5 2	35	4	1413
310		AAG	RG01		1 4 1 3	10	1510	1566
311		AAD	CTRL		1566	15	1882	1502
312		AAD	4600		1502	15	1655	1 4 5 9
313		TRD	P0003	TGP21	1 4 5 9	20	1979	1552

314	4600	46	00		1655	46		
315	X0008	AAD	MEMNN		1845	15	1884	1602
316		DAG	0004		1602	35	4	1 4 6 3
317		AAD	00001		1463	15	1968	1524
318		AAD	900		1524	15	1526	1652
319		TRD	P0003		1652	20	1979	
320		RAD	MEMN		1702	65	1403	1702
321		DAG	0002		1509			1509
322		AAD	00002		1616	35	2	1616
323		DAD	0006		1825	15	1970	1825
324		AAD	0000	VAPFO	1752	30	6	1752
325			0000	0001		15	1705	1559
326	SPZA	RAG	00001	0001	1705	69	1000	1
327	JEZA	AAG			1369	60	1968	1574
328		AAG	00002		1574	1 0	1969	1624
329			RDO1		1624	1 0	1701	1755
330		TRG	W O O O 1		1755	21	1991	1802
331		RAD	00006		1602	65	1973	1677
332		AAD	TDOO		1677	1 5	1638	1653
333		AAG	00005		1653	10	1972	1727
		GNN	INDSA	NINZA	1727	44	1703	1753
334	INDZA	EDI	W O O O 1		1703	69	1991	1803
336		TDI	W0002		1803	2 4	1992	1 4 0 4
337		TRG	W O O O 1		1 4 0 4	21	1991	1 4 5 4
		TRD	W0003		1 4 5 4	20	1993	1504
338			CTRL		1504	65	1882	1554
339		DAG	0004		1554	35	4	1666
340		AAG	WEWN		1666	10	1403	1609
3 4 1		AAG	RG1AG		1.609	10	1460	1716
342		AAD	D0052		1716	15	1937	1604
343		AAD	$M \equiv M \times$		1604	15	1659	1513
3 4 4		EDI	8003		1513	69	8003	1370
3 4 5		SBI	P0002		1370	23	1978	1654
346		AAD	RGOB		1654	1 5	1709	1563
347		TRO	P0003	VAPF3	1563	20	1979	1645
348	NINSA	TRD	M0005		1753	20	1992	1704
349		RAD	MEMN		1704	65	1403	1759
350		DAG	0002		1759	35	2	1766
351		AAG	CTRL		1766	1 0	1882	1754
352		AAD	8001		1754	1 5	8001	1 4 1 1
353		DAG	0004		1 4 1 1	35	4	1671
354		AAG	RGO1		1671	10	1510	1816
355		TRG	P0002		1816	2.1	1978	1804
356		AAD	C O O O 1		1804	15	1968	1674
357		AAD	A D 1 O	TDP32	1674	15	1777	1805
358	TDP32	TRD	P0003	VAPF2	1805	20	1979	1748
359	G0001	RAD		SUITG	1857	65	1969	1724
360	G0002	RAD	00002		1858	65	1969	1724
361	G0003	RAD		SUITG	1859	65	1969	1724
362	G0004	RAD	00002	SUITG	1 5 6 0	65	1969	1724
					2 2 3 0			1 / 2 4

363	G0005	RAD COOO2	SUITG	1861	65	1969	1724
364	SUITG	DAG 0002		1724		2	1456
	001.4					. ~	
365		AAG COOO1		1 4 5 6	10	1968	1774
366		DAG 0004		1774	35	4	1506
367		TRG WOOO1		1506	21	1991	1556
368		RAD PPPRI		1556	65	1360	1367
369		DAG 0004		1 3 6 7	35	4	1827
370		AAD EDDEL		1827	1 5	1730	1606
371		SAD ODIXO	8002	1606	16	1809	8002
372	HA	TDI WOOO4		1656	24	1994	1706
373		RAD COOO6		1706	65	1973	1378
374		AAD TDOO		1378	15	1638	1756
375		AAG COOO4		1756	10	1971	1576
376		AAG COOOS		1576	10	1972	1 4 2 8
377		GNN IND3A	NINJA	1 4 2 8	44	1806	1660
378	INDJA	EDI WOOO1		1806	69	1991	1710
379		TDI WOODZ		1710	24	1992	1760
380		TRG WOOD1		1760	21	1991	1810
		TRD WOODS		1810			1 4 6 1
381					20	1993	
382		RAG CTRL		1 4 6 1	60	1882	1511
383		DAG 0004		1511	35	4	1721
384		AAG WOOO4		1721	10	1994	1561
385		AAG RGOO		1561	10	1738	1611
386		AAD DOO52		1611	15	1937	1661
387				1661			1469
		AAD MEMY			1 5	1 4 1 4	
388		AAD MEMN		1 4 6 9	1 5	1 4 0 3	1711
389		AAD RD10		1711	15	1560	1 4 1 7
390		TRD POOO2	TGP33	1 4 1 7	20	1978	1549
391	TGP33	TRG POODS	VAPF3	1549	21	1979	1645
392	NINJA	TRD WOOD2		1660	20	1992	1761
	NCNII						
393		RAD MEMN		1761	65	1403	1811
394		DAG 0002		1811	35	2	1467
395		AAG CTRL		1 4 6 7	1 0	1882	1412
396		AAD 8001		1 4 1 2	15	8001	1519
397		DAG 0004		1519	35	4	1579
398		AAG RD01		1579	1. 0	1701	1462
399							
1		TRG P0002		1 4 6 2	21	1978	1512
400		AAD WOOO4		1512	1 5	1994	1562
401		AAD ED10	TOP32	1562	1 5	1644	1805
402	CTE1	00 0000	G0000	1388	00		1856
403	002		2	1 4 1 0	00		2
404		00 0001		1 4 3 3	00	1	00
							8003
405		60 0001		1 4 6 0	60	1	
406	MCTRO	00 1877	0000	1 4 8 6	00	1877	000
407	69	69 00		1594	69		
408	PERF 1	00 0100	0001	1612	00	100	0001
409	PERF2	00 0200		1662	00	200	0002
410	PERF3	00 0300		1712	00	300	0003
7 1 0	. 5111	50 0900	0000	4 1 4 6	00	300	

	411	PERF4	00	0400	0004	1762	00	400	0004
	412	PERF5	00	0500	0005	1812	00	500	0005
	413	CTE2	00	0000	F0000	1532	00		1846
	414	CTE3	0.0	0000	X0000	1833	00		1837
	415	RD10	65	0001	0000	1560	65	1	00
	416	6013	60	0000	0013	1 4 8 4	60		13
	417	AGOW2	10	0000	W0002	1 4 5 0	1 0		1992
	418	SGOW2	1 1	0000	W0002	1 4 0 1	1 1		1992
	419	TGOO	21	00	110002	1588	2 1		1 9 9 2
	420	RGOW 1	60	0000	W O O O 1	1536	60		1 001
í	421	MULOZ	19		z				1991
	422	001	1 9	0000	1	1 4 0 2	19		1939
	423	EDOGA	6.0	0.000		1634	00		1
			69	0000	G A	1610	69		1500
	424	TDOO	20	0		1638	20		
	425	DROW2	64	0000	W0002	1 4 5 3	6 4		1992
	426	00W1	00	0000	W0001	1586	00		1991
	427	020	00	0005	0000	1733	00	2	00
	428	EDOGB	69	0000	GB	1 3 9 2	69		1550
	429	RDOO	65	0 0		1794	65		
	430	AGOO	1 0	0		1 4 9 2	10		
	431	S G O O	1 1	00		1542	1 1		
	432	RGOO	60	0		1738	60		
	433	ED2GC	69	0002	GC	1 4 1 9	69	2	1700
	434	GNARM	4 4	ARETM	0000	1598	44	1936	000
	435	DROO	6 4	00		1750	64		
	436	EDOGD	69	0000	GD	1 3 5 4	69		1651
	437	RDO1	65	0000	0001	1701	65		1
	438	RG08	60	0000	8	1709	60		8
	439	MEMX	0 +		2.8	1659	00		28
	440	MEMY	00		43	1 4 1 4	00		48
	441	VAPF 1	RAD	PERFI	PERFO	1795	65	1612	1400
	442	VAPF2	RAD	PERF2	PERFO	1748	65	1662	1400
	443	VAPF3	RAD	PERF3	PERFO	1645	65	1712	1 4 0 0
	444	VAPF 4	RAD	PERF4	PERFO	1 482	65	1762	1 4 0 0
	445	VAPF 5	RAD	PERF5	PERFO	1613	65	1812	1 4 0 0
	446	RGO1	60	000	r 1	1510	60	1012	1
	447	AD10	15	0001	0000	1777	15	1	00
	448	EDDEL		00000		1730		1885	1656
			69		HA		69		
	449	ODIXO	00	0010	0000	1809	00	1 0	00
	450	ED10	69	0001	0000	1644	69	1	00
	451	033	00	0003	0003	1663	00	3	0003
	452	24W12	2 4	W O O O 1	0002	1713	2.4	1991	0005
	453	240K	2 4	0000	K	1763	24	100	1930
	454	2 4 5 5 1	2 4	0055	0001	1813	2 4	55	0001
	455	44K2	44	KPLU2	0000	1 4 6 4	4 4	1932	000
	456	ARETL	LEC	1951		1737	70	1951	1514
	457		EDI	1951		1514	69	1951	1564

458		TDI	CTRL		1564	24	1882	1614
459		EDI	1952		1614	69	1952	1664
460		TOI	LMAX		1664	24	1883	1714
461		RAD	MEMN	DEBU2	1714	65	1403	1356
462	UNEME	TRD	W0005		1358	20	1992	1764
463		RAD	MEMN		1764	65	1403	1814
464		AAD	001		1814	15	1634	1517
465		TRD	P0001		1517	20	1977	1780
466		RAV	M0005		1780	67	1992	1567
467		EDI	8003		1567	69	8003	1824
468		SBI	MEMNN		1824	23	1884	1617
469		EDI	8003		1617	69	8003	1626
470		SBF	W0001		1626	22	1991	1667
471		DAG	0002		1667	35	2	1676
472		AAG	CTE4	8003	1676	10	1629	8003
473	T0005	RAD	W0001		1871	65	1991	1717
474		DAD	0004		1717	30	4	1478
475		TRO	W0001	UNE53	1 4 7 8	20	1991	1767.
476	UNE53	TRD	P0002		1767	20	1978	1817
477		PFO	P0001		1817	7 1	1977	1528
478		RAD	W0001		1528	65	1991	1418
479		DAG	0004		1 4 1 8	35	4	1679
480		AAD	001		1679	15	1634	1468
481		TRD	P0001		1 4 6 8	20	1977	1830
482		EDI	UNE 51		1830	69	1518	1771
483		SBF	UNE51	8001	1771	22	1518	8001
484	UNES1	RAV	00	0001	1518	67	1310	1568
485	ONEST	EDI	8003		1568	69	8003	1726
486		SBI	W O O O 1		1726	23	1991	1618
487		AAG	8001		1618	10	8001	1776
488		SAG	ENCOD		1776	1 1	1729	1668
489		GNN	ENCOD	UNE52	1668	44	1821	1472
499		-	0.001	UNE53	1821	1 0	8001	1767
	LINESO		8001	ONESS	1472	60	8002	1718
491	UNE 52	RAG	8002		1718		1884	1768
492		AAD	MEMNN			15		
493		SAG		E 1 11 E E	1768	1 1	8003	1826
494	T 0 0 0 7	SBI	P0002	FINPF	1826	23	1978	1436
495	T0007	RAD	6300		1873	65	1578	1818
496	LECPF	AVD	W0002	VAPFO	1818	17	1992	1559
497	VAPFO	TRD	P0002	FINPF	1559	20	1978	1436
498	T0001	RAV	W0005	VAPFO	1867	67	1992	1559
499	T0009	DAD	0006	VAPFO	1875	30	6	1559
500	CTE4	00	0000	TOOOO	1629	00		1866
501	ENCOD	00	0000	DEBUT	1729	00		1800
502	6300	63	0 0		1578	63		
503	T0002	RAG	CTRL		1868	60	1882	1569
504		DAG	0004		1569	35	4	1779
505		AAG	8001		1779	10	8001	1619
506		AAD	8001		1619	15	8001	1628

507		AAG	RG01		1628	10	1510	1669
508		TRG	P0002		1669	21	1978	1719
509		AAG	033		1719	10	1663	1769
510		TRG	W0003		1769	21	1993	1819
511		CAA	24W12		1819	15	1713	1420
512		TRD	W0002		1420	20	1992	1470
513		RAD	D0051		1 4 7 0	65	1936	
514		DAG	0004		1520	35		1520
515		AAD	8001		1570	15	4	1570
516		EDI	240K		1678		8001	1678
517		SBF	P0006		941	69	1763	1620
5.18		AAD	24551		1620	22	1982	1670
519		TRD	P0005		1670	15	1813	1720
520		RAG	MEMNN		1720	20	1981	1770
521		AAG	44K2		1770	60	1884	1820
522		TRG	W O O O 1	V A D E E	1820	10	1464	1522
523	T0006	RAD	RDOO	VAPF5	1522	21	1991	1613
524	T0003	SAG		LECPF	1872	65	1794	1818
525	10003		8003		1869	1 1	8003	1728
526		DAD	0002		1728	30	2	1572
527		AAD	TDOO		1572	15	1638	1622
		TRD	W 0 0 0 1		1622	20	1991	1672
528		RAG	CTRL		1672	60	1882	1722
529		AAG	RGOO	TGP21	1722	10	1738	1552
530	TGP21	TRG	P0002	VAPF1	1552	21	1978	1795
531	T0000	RAV	M0005	VAPFO	1866	67	1992	1559
532	T0004	RAV	W0002		1870	67	1992	1772
533		SAD	0300		1772	16	1778	1822
534		TRD	W O O O 1		1822	20	1991	1828
535		AAG	69		1828	10	1594	1829
538		AAG	CTRL		1829	10	1882	1352
539		SAG	8003		1 3 5 2	1 1	8003	1353
540		SBF	P0002	VAPF1	1353	22	1978	1795
541	0300	03	00		1778	03		. , 9 3

1		PRA	1995	1996				
		PRA	1967	1967				
		PRA	0000	1940				
3		PRA	1950	1950				
4		PRG	D1997	1999				
5		PRG	L 1951	1960				
6		PEN	SC	1994				
7	SA	RAG	M 1.		1961	60	1964	1969
8		AAG	MILLE		1969	1 0	1972	1977
9		AAD	8001		1977	15	8001	1985
10		AAD	L0009		1985	15	1959	1963
1 1		TRG	M 1		1963	21	1964	1967
12		TRD	L0009		1967	20	1959	1962
13		SAG	L0001		1962	1 1	1951	1965
1 4		ANG	SUITE	LECT	1965	46	1968	1970
15	SUITE	RAG	L0010		1968	60	1960	1966
16		DAG	0002	SD	1966	35	2	1973
17	SD	TRG	L0010	L0009	1973	21	1960	1959
18	sc	RAD	L0001		1994	65	1951	1971
19		ANG	TRAN	NTRAN	1971	4 6	1974	1975
20	NTRAN	RAG	M 3	COMMU	1 9 7 5	60	1978	1983
21	TRAN	RAD	D0001		1974	65	1997	1976
22		EDI	L0007		1976	69	1957	1979
23		TDI	L0010		1979	2 4	1960	1980
24		EDI	M 2	COMMU	1980	69	1984	1983
25	COMMU	TDI	L0009		1983	24	1959	1981
26		AVD	L0001		1981	1 7	1951	1982
27		EDI	M 1		1982	69	1964	1986
28		SBF	M 1		1986	22	1964	1987
29		DAG	0004		1987	35	4	1947
30		AAD	8001		1947	15	8001	1988
3 1		SBF	L0001	L0009	1988	22	1951	1959
33	SB	ANG	OUI	NON	1989	46	1942	1943
34	OUI	SAG	UN	TEST	1942	1 1	1945	1949
35	NON	AAG	UN	TEST	1943	10	1945	1949
36	TEST	EDI	L0010		1949	69	1960	1990
37		SDO	HUIT	NEUF	1990	90	1944	1946
38	HUIT	SD9	DELDE	DELZE	1944	99	1948	1991
39	NEUF	SD9	ZEDEL	M 1	1945	99	1992	1964
40	DELDE	MUL	00003	ADDIT	1948	19	1999	1993
41	DELZE	MUL	D0001	ADDIT	1991	19	1997	1993
42	ZEDEL	MUL	00002	ADDIT	1992	19	1998	1993
43	ADDIT	AAD	8003	M 1	1993	1 5	8003	1964
44	LECT	LEC	SC	8000	1970	70	1994	8000
45	M 1	TRD	0000	SA	1964	20		1961
46	M 2	RAD	L0002	SB	1984	65	1952	1989
47	м 3	RAD	L0002	M 1	1978	65	1952	1964
48	MILLE	00	0001	0000	1972	00	1	00
49	UN			1	1945	00		1



Vu et Approuvé

Nancy, le

Le Doyen,

E. URION.

Vu et Permis d'Imprimer

Nancy, le

Le Recteur :

Président du Conseil de l'Université,

P. IMBS.