ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE ET ELECTROTECHNIQUE

PROJET DE FIN D'ETUDES

INGENIORAT EN ELECTRONIQUE

CONCEPTION et REALISATION d'un SYSTEME D'EXPLOITATION sur BANDE MAGNETIQUE

Proposé par: H.TEDJINI
Docteur Ingénieur

Etudié par:

Djamal MEGUENNI Mohamed AIDJA

— JUIN 81 —

سبحانك لا علم لنا إلا ما علمتنا

صدق الله العظيم

MESRS USTHB

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE ET ELECTROTECHNIQUE

PROJET DE FIN D'ETUDES

INGENIORAT EN ELECTRONIQUE

CONCEPTION et REALISATION d'un SYSTEME D'EXPLOITATION sur BANDE MAGNETIQUE

Proposé par: H.TEDJINI
Docteur Ingénieur

Etudié par:

Djamal MEGUENNI Mohamed AIDJA

— JUIN 81 —

REMERCIEMENTS

"...En ces moments mémorables,où notre étude touche à sa fin, nous ne pouvons rester sans une pensée sincère à tous ceux qui ont contribué à notre formation,en particuliers ,nos chers professeurs de l'Ecole Nationale Polytechnique .

Ainsi,nous remercions Mr. B. SANSAL pour nous avoir accepté dans sa division,nous permettant ainsi de mener à bien ce projet de fin d'études.

Nous remercions Mr. H. TEDJINI ,Docteur-Ingénieur et chargé de cours à l'Université des Sciences et de la Technologie Houari Boumédiène,pour la confiance qu'il a placé en nous,ainsi que pour les conseils et les encouragements qu'il n'a cessé de nous prodiguer tout le long de notre étude.

Qu'il veuille bien croire à notre gratitude.

Nous tenons aussi à remercier l'ensemble du personnel de la division,en particulier Mr. S. NOUR,Mr. A. BOURKEB pour leur conseils , ainsi que le personnel de la RONEO pour leur assistance technique.

Nous ne terminerons pas sans remercier également le binôme qui nous a précédé Mrs. M. BENMILOUD et M. ALIKACEM .

Que tous ceux qui ont contribué de prés ou de loin à l'élabo-ration de ce projet trouvent ici l'expression de notre profonde reconnaissance.

DEDICACES

A mon père pour son courage et son sacrifice

A ma mère pour son soutien moral

A mes frères

A mes soeurs

A M. El Bouziri pour son aide morale

A tous mes amis

Djamal

A mon père à qui je dois tout

A ma mère pour ses sacrifices

A mes frères et soeurs

A tous mes amis

Mohamed

§§§§§§§§§§§§§
§§§§§§§§§
§§§§§
§

S O M M A I R E

II/BANDE MAGNETIQUE ET ACCES DIRECT MEMOIRE

A/Bande magnétique

A-1/Introduction

A-2/Gestion de la bande magnétique

A-3/Format d'un enregistrement sur bande magnétique

A-4/Coupleur du dérouleur de bande

B/Canal A.D.M.

B-1/Généralités

B-2/Canal A.D.M.

B-3/Organes d'entrées-sorties

III/APPLICATION : SYSTEME BOS

A/ Chargeur élémentaire

B/M.E/T

C/Editeurs de liens

D/Editeur de texte

E/Assembleur

CONCLUSION

ANNEXE

I N T R O D U C T I O N

Une machine à calculer en bon état mais non éduquée, serait totalement inutilisable;car les opérations de lecture des info--rmations et de rangement des instructions en mémoire ne peuvent se faire que grâce à un programme spécial mis préalablement en mémoire.
On voit ainsi que parmi les problèmes vitaux de l'informatique,figure celui de la communication entre l'homme et la machine.
Ce problème se présente sous deux formes:
-d'une part,l'étude des moyens utilisés pour que l'homme transmette clairement sa pensée à la machine
-d'autre part,l'exploitation temporelle optimale de la machine.
La résolution de ce problème conduit à réaliser un programme d'exploi--tation,constituant le système d'exploitation qui est le software de base de la machine;
Ces programmes permettent à l'homme d'utiliser au mieux les possibilités de l'ordinateur,tout en minimisant les tâches fastidieuses de l'opérateur.
Un tel système existe sur l'unité de disque du Multi-20,au sein du C.S.T.N.,mais comme cette dernière est une unité tres fragile et tombe souvent en panne,notre tâche consiste alors à réaliser un sys--tème similaire sur bande magnétique;cette unité est moins puissante que celle du disque,cependant,elle donne des résultats qui satisfont aux besoins de l'utilisateur.

Ainsi le présent mémoire s'inscrit dans le nouvel axe tracé au sein de la division V : "Simulation et contrôle" du Centre des Sciences et de la Technologie Nucléaire

PRESENTATION DU SUJET

Notre travail consiste à concevoir et à réaliser un système d'exploitation sur bande magnétique.

Le système d'exploitation doît comprendre au minimum un programme chargeur élémentaire et un moniteur.

Le programme chargeur élémentaire permet d'exécuter les opérations d'enregistrements de la bande magnétique en mémoire centrale, de lecture et de rangements des instructions en mémoire (commandes A.D.M.).

Ce programme permet le chargement et l'exécution du programme suivant qui est le moniteur d'explitation télétype (M.E.T.) lequel permet à son tour l'enregistrement et l'exécution de tous les autres programmes en particulier l'editeur de leins et l'assembleur.

Le problème qui se pose est que, souvent, à chaque fausse instruction de l'opérateur, le M.E.T. s'efface partiellement ou totalement de la mémoire centrale. On devait donc le recharger par son programme amorce sur le lecteur de télétype par ruban perforé (support très fragile et peu fiable) ce qui nècessitait un travail long et pas toujours satisfaisant.

De plus, un ruban perforé ne peut, aussi volumineux soit-il, contenir tout un système.

De ce fait, à chaque fois que la mémoire centrale se vidait accidentellement, on se contentait du chargement du M.E.T. seul.

A/ CALCULATEUR

Chapitre I/PRESENTATION DU CALCULATEUR

A - 1/ Généralités

Le Multi-20 est un calculateur numérique à logique microprogrammable. L'application des techniques de microprogrammation en mémoire permanente R.O.M. permet l'utilisation du Multi-20 dans de nombreux domaines. L'architecture de ce calculateur est organisée sur la base de l'octet, ce qui autorise le traîtement d'informations en longueur variable et la manipulations de chaînes de caractères de manière souple et économique. Multi-20est capable de satisfaire au besoin d'une largr gamme d'appli--cations grâce aux possibilités d'extension tant de sa mémoire principale à tores que de sa mémoire à grande vitesse.
Le choix délibéré du découpage de l'information en octets permet une utilisation optimale de la mémoire et des circuits.
Le Multi-20 est disponible en plusieurs versions stantards qui corres--pondent à des répertoires d'instructions particuliers.Le répertoirele plus complet est le M.1305 correspondant à Multi-20 / 05.

A-2-1/ Les registres programmables

Le Multi-20 dispose de 6 registres accessibles par macro-instruction:
-Registre accumulateur A contenant 16 bits
-Registre extension accumulateur B contenant 16 bits
-Registre index X contenant &§ 16 bits
-Registres compteur ordinal P contenant 16 bits
-Registre de précision W contenant 2 bits
-Registre de débordement OV contenant 1 bit

A-2-2/Mémoires à tores

Elle est constituée de modules enfichables de 4096 et 8192 octets.
Elle est extensible jusqu'à 65536 octets.L'adressage s'effectue par octet

Le cycle de lecture-écriture de la mémoire est d'une microseconde.
Bien que constituée essentiellement de blocs technologiquements dis--tincts,la mémoire ne permet que l'accés à un seul bloc à la fois;

A-2-3/Systèmes d'entrée-sortie

Il est constitué essentiellement de 2 liaisons permettant le transfert d'informations de ou vers l'extérieur.Celles-ci sont:
-la liaison standard pour les transferts par octet en mode programmé ou simultané.
-la liaison directe à la mémoire pour les transferts rapides de bloc d'informations de ou vers la mémoire.

A-2-4/Présentation des informations

L'élément d'information de base est l'octet(8bits°).Les instructions et les données occupent un nombre variable d'octets en mémoire.

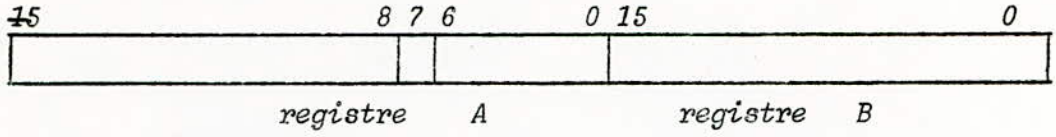
1/Format des données

Les données sont des nombres entiers signés en simple(1 octet), double (2 octets), triple(3 octets°) , ou quadruple précision(4 octets).
Les nombres négatifs sont représentés par leur complément à 2.

a/Simple précision

étendue: -2^7 0à 2^7 - 1

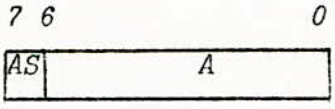
Format dans les registres



Format en mémoire

A:valeur absolue

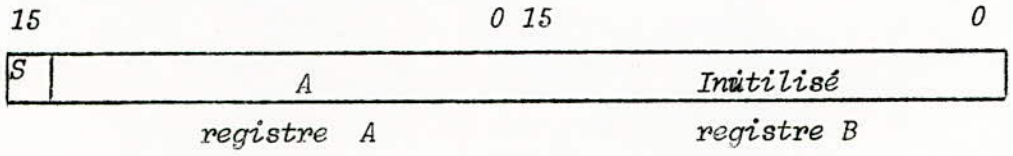
S:signe



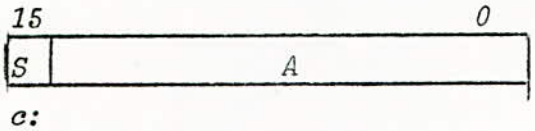
b/double précision

étendue: -2^{15} à $2^{15}-1$

Format dans les registres



Format en mémoire

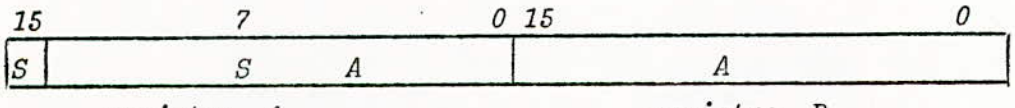


c:

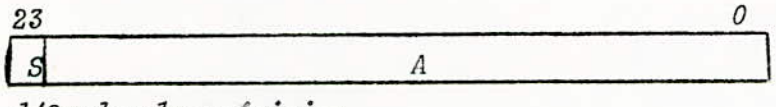
c/Triple précision

étendue: ; -2^{23} à $2^{23}-1$

Format dans les registres



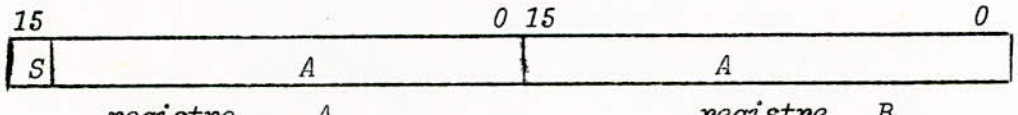
Format en mémoire



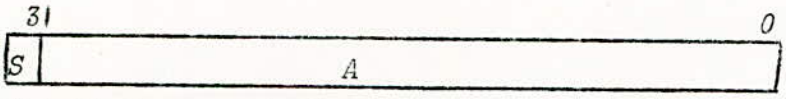
d/Quadruple précision

étendue: -é -2^{31} à $2^{31}-1$

Format dans les registres



Format en mémoire



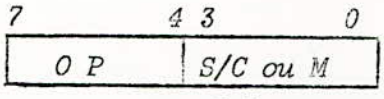
Mot d'adressage indirect

Le mot d'adressage indirect occupe 2 octets en mémoire et contient une adresse complète sur 15 bits.
Le bit 7 de L'octet 1 précise si l'adresse doît être modifiée par le registre index.

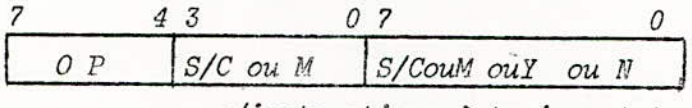
FORMAT DES INSTRUCTIONS

M: mode d'adressage
OP :code opération
S/C :sous-code
Y :adresse d'opérande
N : nombre de décalages

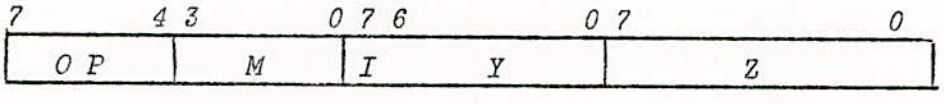
a/instructions à un octet



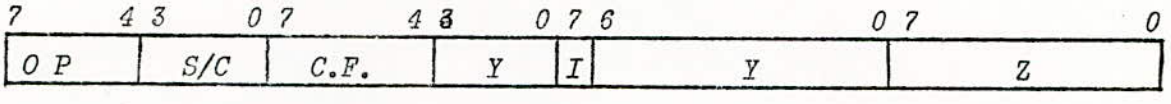
b/instructions à deux octets



c/instructions à trois octets



d/ instructions à quatre octets:



A-2-5/MODES D'ADRESSAGES

Les instructions à référence mémoire possèdent 8 modes d'adressages. Le nombre d'octets de l'instruction varie avec le mode d'adressage; on appelle:

-adresse intermédiaire (A.I.):l'adresse obtenue aprés toutes les modifications d'adresse excepté la post-indexation.

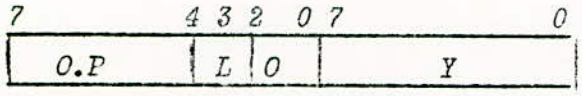
-adresse effective (A.E.):l'adresse effe de l'opérande ou l'adresse de saut obtenue aprés toutes les modifications d'adresse,y compris la post-indexation.

$$A.E. = A.I. + (X)$$

A-2-5-1/Adressage direct page zéro

M= 0 $A.E. = Y$

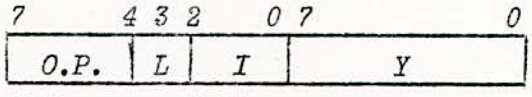
L'adresse effective est égale à l'adresse spécifiée par Y dans les 256 premiers mots.



2-5-2/Adressage direct relatif

M = 1 A.E. = (P) + Y

L'adresse effective est égale au contenu du compteur ordinal augmentée du déplacement Y qui peut êtr positif(è7° bit nul donc déplacement vers le bas) ou négatif (7° bit égal à 1 donc déplacement vers le haut).



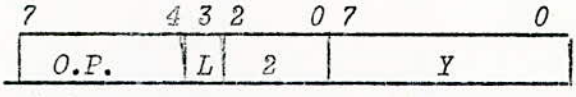
2-5-3/Adressage indirect page 0

M = 2 A.I. = (Y)

L'adresse effective est déterminée par le bit I de l'adresse indirecte(bit 15)

si I = 0 A.E. = (Y)

SI I = 1 A.E. = (Y) + X



2-5-4/Adressage indirect relatif

M = 3 A.I. = (P) + Y

L'adresse effective est déterminée par le bit I de l'adresse indirecte

si I = 0 A.E. =5((P) + Y)

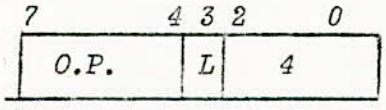
si I = 1 A.E. = ((P) + Y) + (X)

A noter que Y est un dépalcement qui peut être positif ou négatif suivant sa valeur,donc en résumé cet adressage présente quatre variantes différentes .

2-5-5/Adressage par index

M = 4 A.E. = (X)

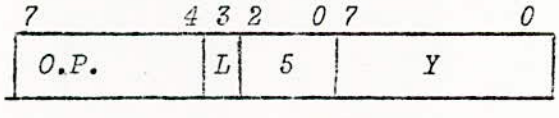
L'adresse effective est égale au contenu du registre index ,de ce fait on peut balayer l'étendue de 0 à(65.535(c'est-à-dire de X =0000 à X = FFFF)

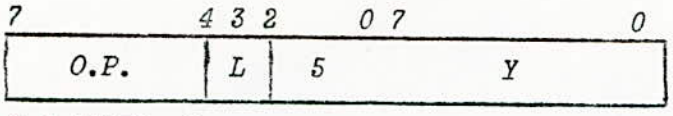


2-5-6/Adressage par index avec déplacement

M =5 A.E. = (X) + Y

L'adresse effetive est égale au contenu du registre index plus la valeur de Y; Y étant considérée comme un nombre de 8 bits sans signe.





2-5-7/ Etendu

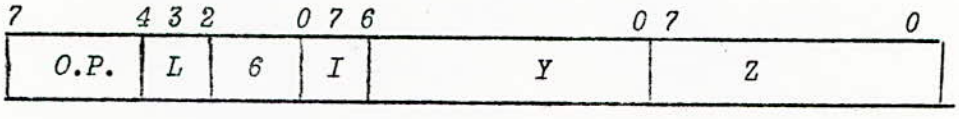
M =6 A.I. = Y , Z

L'adresse intermédiaire est égale à la valeur Y , Z.

L'adresse éffective est déterminée par le bit I ;

si I = 0 A.E. = Y , Z

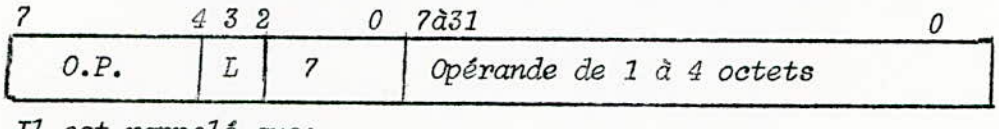
si I = 1 A.E. = Y , Z + (X)



2-5-8/ Adressage immédiat

M = 7 A.E. = (P)

L'adresse est effective est égale à la valeur courante du compteur ordinal qui rappelons-le est un registre contenant 16 bits



Il est rappelé que:

-la zone O.P. définit le code opération

-le bit L définit,pour les instructions à longueur variable la longueur de l'opérande:

si L = 0 2 octets (16 BITS)

si L = 1 suivant la valeur du registre de précision W

Pour les autres instructions,le bit L définit des instructions supplémen-taires .

A-3/Les catégories d'instructions du Multi-20

Le répertoire Multi-20 est composé des catégories d'instructions suivantes

-inqstructions de contrôle

-instructions de branchements conditionnels

instructions de décalage

-instructions sur registres

-instructions à référence mémoire

-instructions travaillant sur pile

-instructions de manipulations de chaîne de caractère

-instructions arithmétiques.

A-4/Les entrées-sorties du multi-20

Le multi-20 DISPOSE DE 3types d'entrées-sorties

-transfert d'octets en mode programmable sur liaison standard parallèle.

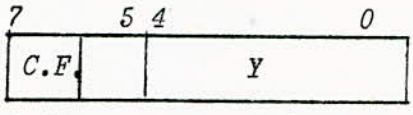
-transfert de blocs d'octets en simultané sur liaison standard parallèle.

-transfert de blocs d'octets par accés direct à la mémoire sur canal A.D.M.

La liaison standard parallèle est la voie de communication entre les organes périphériques et les registres A et B ainsi que la mémoire.
Le canal d'accés direct à la mémoire,comme son nom l'indique commu--nique directement à la mémoire ou plûtot avec la mémoire , à tores, sans le recours des registres,cette partie sera étudiée plus en détail au chapitre II .

4-1/OPERATIONS D'ENTREE-SORTIES

Adresses,commandes et données circulent sur la liaison standard.
Ce canal multiplexé permet de mêler des transferts de blocs.
Plusieurs périphériques peuvent réaliser leurs transferts de blocs de données en même temps.
Le nombre maximum de périphériques est de 32 .
Dans une instruction d'entrée-sortie,le second octet est une commande.
Il contient un code de fonction (C.F.) sur 3 bits et une adresse de périphérique sur les 5 autres bits.
Format:



Y étant cette- fois-ci l'adresse de périphérique.

4 2/ADRESSE DE PERIPHERIQUE

A chaque coupleur de périphérique placé sur la liaison stantard,est affecté une adresse de5 bits . Cette adresse est comparée à l'adresse figurant dans l'octet de commande.
Elle est transmise vers l'unité centrale lorsque le périphérique emet une demande de transfert,afin de permettre à ce dernier d'iden--tifier le demandeur.
Adresses utilisées pour notre travail:

adresse 0 pour l'interface télétype série
adresse 7 pour l'interface bande magnétique
adresse 16 pour l'interface A.D.M.

4-3/ CODE DE FONCTION

Ce code ,sur 3 bits ,précise le type d'opération à effectuer.
Il fait partie de l'octet de commande au même titre que l'adresse du périphérique concerné.
Les codes de fonction : 2,3,5,6 et 7 peuvent avoir une signification différente selon les coupleurs de périphériques.

CODE DE FONCTION

Code	Fonction	D e s c r i p t i o n
0	Transfert de donnée	Un octet de donnée est transmis entre le périphérique concerné et l'unité centrale
1	Etat/ Com--mande	Dans une instruction d'ENTREE, ce code est une deman--de de mot d'état au périphérique adressé. Dans une instruction de sortie, ce code indique que l'octet transmis de A à la mémoire est une commande
2	Entrée de bloc avec interrup	Dans une instruction de sortie, ce code initialise un transfert simultané du périphérique vers la mémoire.
3	armement de l'interruption	Dans une instruction de sortie, ce code arme le générateur d'interruption externe.
4	arrêt immédiat	Dans une instruction de sortie, ce code ordonne la déconection immédiate du coupleur, donc l'arrêt du transfert en cours.
5	Désarmer l'interru	Dans une instruction de sortie, ce code désarme le générateur d'interruptions externes.
6	Sortie de bloc	Dans une instruction de sortie, ce code initialise un transfert simultané dans le sens mémoire vers périphérique.
7	Disponible	a une signification différente suivant le coupleur

MOT D'ETAT

Les bits de 0 à 3 ont une signification standard, les autres sont particuliers à chaque périphérique.

Bits	Indicateur d'état	D E S C R I P T I O N
0	prêt	Ce bit est égal à 1 lorsque le périphérique est prêt.
1	indicateur d'entrée	ce bit est égal à 1 quand le périphérique est prêt à envoyer un octet au calculateur
2	indicateur de sortie	ce bit est égal à 1 quand le périphérique est prêt à recevoir un octet du calculateur
3	Erreur	ce bit est égal à 1 lorsqqu'une erreur s'est produite durant le transfert.

B/ SYSTEME D'EXPLOITATION N

1/ DEFINITION

Un système d'exploitation, comme son nom l'indique, permet d'exploiter d'une façon optimale les possibilités et les capacités de l'ordinateur. Il constitue le software de base de la machine.

C'est donc un ensemble de règles concernant les grandes lignes de la programmation et l'exploitation de la machine et de ses programmes avec le meilleur rendement possible.

Déroulement du fonctionnement

Toute machine comporte un bouton de démarrage, qui a pour effet, lorsqu'il est appuyé, de lancer le programme à partir d'une certaine adresse constante; (pour le Multi-20 le microprogramme-chargeur l'implante et le lance à l'adresse zéro).

Par ailleurs, une manoeuvre spéciale doit avoir pour effet de charger à partir de cette même adresse, les premiers mots lus sur la bande magnétique. Cette manoeuvre spéciale est assuré par le programme chargeur élémentaire, appelé parfois programme initial(P.I.).

Ainsi , on a introduit une amorçe de programme, dont le rôle est d'alimenter la suite, c'est-à-dire de charger un certain programme et de le lancer.

La fonction d'enchaînement se traduit par des programmes qui sont des pièces essentielles du système et jouent un rôle important qui est "entre" les programmes de l'utilisateur proprement dit. Leur ensemble porte le nom de moniteur.

Aprés avoir décrit brièvement la FONCTION que doît remplir le système d'exploitation, il en résulte les différentes parties organiques que doît comporter le système.

Nous avons [illegible] évoqué 3 fonctions qui doîvent être satisfaîtes, nécessitant 3 organes qui interviendront , chacun, à des moments différents du fonctionnement de l'ordinateur.

Constitution:

Le système est lancé par le microprogramme amorce. Ce software de base est en général constitué par:

-le chargeur initial: qui ne doît jouer son rôle qu'à la mise quotidienne du système.

-le moniteur résident: qui est présent en permanence dans la mémoire centrale dés-que l'initialisation quotidienne a été éffectuée.

-le moniteur de transition: qui est chargé en mémoire par le résident, aussitôt qu'un programme de l'utilisateur est terminé; son rôle est de préparer la mémoire pour le chargement suivant.

Pour notre étude, le système d'exlpoitation, software de base sera constitué par un groupe de programmes qui joueront le rôle "d'aide à la programmation". Il comportera:

-un chargeur : CARBAM

-un moniteur d'exploitation : M.E.T.

-un éditeur de liens :M.A. E.D.L.

-un éditeur de texte : P.E.R.

-un assembleur P.A.T.

Pour mettre en mémoire centrale le chargeur élémentaire, il faudra l'exciter par un micro-programme de chargement à l'aide des touches sur le périphérique avant du Multi-20.

En résumé pour un système d'exploitation, il faut impérativement et au minimum,

-un micro-programme de chargement

un programme initial (programme de chargement initial)

-un moniteur résident (met (MET)

Nous allons étudier tour à tour leur rôle et leur fonctionnement.

B-2/Le M.E.T.

B-2-1/Définition

Le moniteur d'exploitation télétype constitue l'ensemble des opérations chargées de l'exécution d'un programme, c'est en d'autres termes, "l'ordonnateur" du programme.

Les instructions élémentaires doivent pouvoir être comprises par l'ordonnateur du programme, au moment où il les lit une à une dans la mémoire centrale, pour les éxécuter, il faut donc qu'elles soient rédigées dans un langage que la machine comprenne.

B-2-2/ Rôle et fonctionnement

Avant tout travail sur l'ordinateur, il faut mettre en place dans la mémoire centrale, le moniteur résident, sans lequel, aucune opération ne peut avoir lieu; pour le Multi20 , c'est le M.E.T.

Mais pour charger le MET en mémoire centrale, il faudra au préalable charger un programme chargeur initial (PI) de lecture du programme à exécuter.

B-3/Le programme initial

B-3-1/Définition

Le programme initial est un processus software permettant la lecture et l'enregistrement en mémoire centrale, du programme immédiatement suivant.

Il excite le démarrage du chargement du MET.

B-3-2/Rôle et fonctionnement

On a précisé précedemment que le chargement du moniteur résident nècessitait impérativement,une éxcitation fournie par un programme chargeur élémentaire appelé P.I.

Toutes les machines comportent donc,d'une manière inamovible,un programme d'ordres initiaux,qui leur permet de commencer à travailler.Le programme chargeur élémentaire est toujours sommaire,il déclenche la lecture du premier enregistrement bande magnétique, (par exemple), en mémoire centrale.

Une fois en mémoire,à un emplacement autre que celui qu'occupera un peu plus tard le M.E.T.,le chargeur initiale prépare la mémoire, enfin il initialise le monoteur.

B-4/MICROPROGRAMME DE CHARGEMENT

B-4-1/Définition

C'est un procédé hardware qui permet la lecture du premier enregistrement sur bande magnétique,c'est-à-dire le programme élémentaire chargeur (programme initial).

Il est déclenché mécaniquement:action sur les touches du pupitre.

B-4-2/Rôle et fonctionnement

Le microprogramme de chargement permet l'entrée du chargeur élémentaire en mémoire centrale;.

Il joue donc le rôle d'un programme amorce.

C'est un dispositif cablé qui excite le dérouteur de bande magnétique et celà par les clés de test du pupitre avant.

En règle général,il assure le chargement des programmes directement par le périphérique choisi;.

Chapitre II / B A N D E M A G N E T I Q U E

E T C A N A L A.D.M.

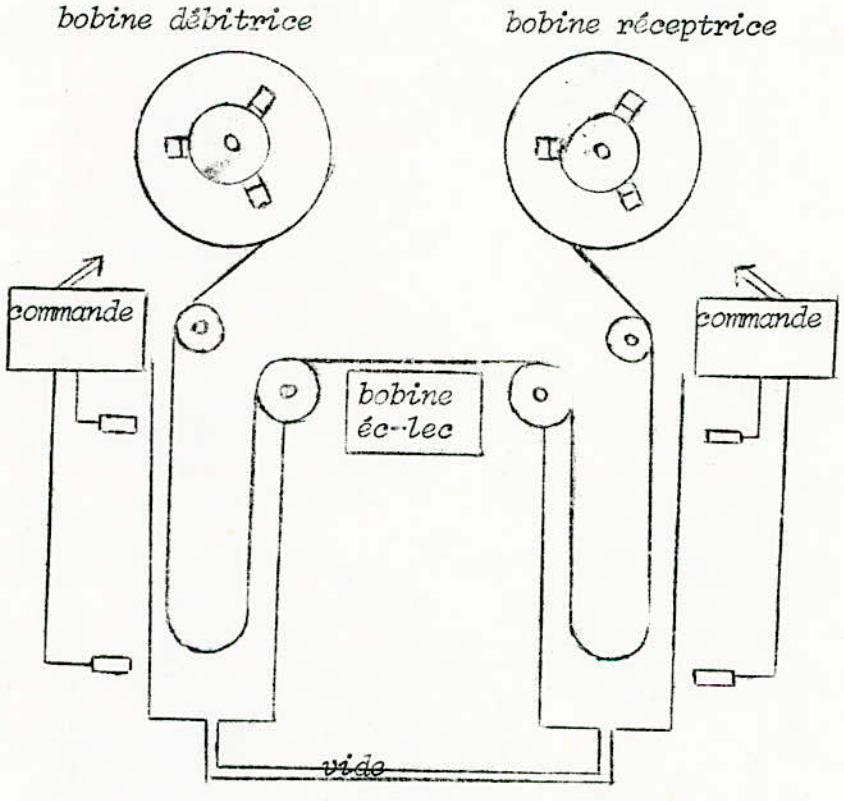
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

A/A BANDE MAGNETIQUE

A-1/Introduction

Le dérouleur de bande est un appareil permettant l'écriture ou la lecture d'une bande magnétique, c'est en quelque sorte le même principe que le magnétophone; il est l'un des périphériques les plus répandus pour la transmission des informations à l'unité centrale.

Shéma du dérouleur de bande utilisé:



Le ruban défile à vitesse constante devant la tête à lecture entre deux cabestans d'entraînement permettant le déroulement de la bande ou son rebobinage.

A-2/ GESTION DE LA BANDE MAGNETIQUE

a:/ l'utilitaire M.E.T.

Le programme utilitaire M.E.T. est autonome et permet principalement de cataloguer des fichiers programmes sur la bande.

On dispose de plusieurs commandes de l'utilitaire M.E.T.

UJ :positionne la bande au début

TZ , :annule tous les fichiers de la bande

TD , :annule le dernier fichier écrit sur la bande

TB nnn,aaa,bbb (rc) :écrit sur la bande magnétique l'enregistrement d'identificateur nnn(& 15 caractères au maximum) aaa,bbb,sont les adresses début et fin de la zone mémoire vive à écrire sur la bande;.

TF dddd (rc) :modifie le descriptif fin d'enregistrement du dernier fichier écrit sur la bande pour y incor--porer une adresse de lancement qui sera utilisée au moment du chargement en mémoire vive par la commande UB. (dddd représente cette adresse de lancement.

UB nnnn(rc) :permet de lire un enregistrement d'identificateurs nnnn

UL :imprime la liste,dans l'ordre d'enregistrement,des identificateurs des fichiers.

b/ Structure de la bande

La bande doît être bornée par un sticker début (BOT) et un sticker fin de bande (EOT)

Un fichier est constituée par un ou plusieurs enregistrements compris entre deux codes "fin de fichiers".

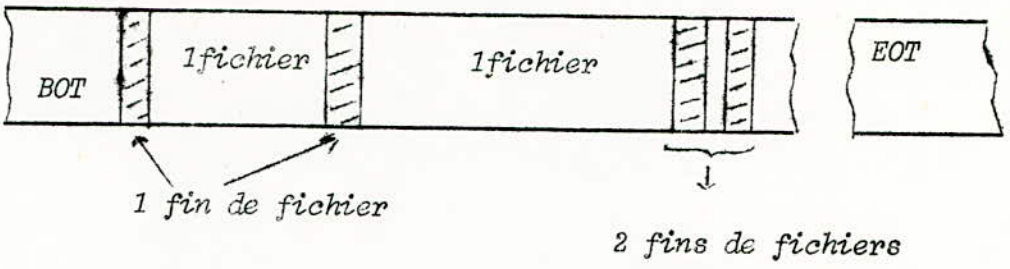
La fin des enregistrements est caractérisée par deux fins de fichiers.

Etat d'une bande aprés la commande d'annulation des fichiers:

2fins de fichiers

Tous les enregistrements débutent donc par deux "fins de fichiers" qui annoncent le début d'enregistrement.

Etat d'une bande aprés l'écriture de deux fichiers:



c/Mode d'accés à la bande

Sur une bande magnétique on ne dispose que d'un, accés séquentiel. En écriture sur bande,on ne peut écrire qu'à la suite de l'enre--gistrement précédent.Il n'est pas possible de réécrire à la place d'un enregistrement existant.
En lecture,on peut accéder à tous les enregistrements écrits sur bande magnétique.

A-3/FORMAT D'UN ENREGISTREMENT SUR BANDE MAGNETIQUE

Un enregistrement est désigné par un nom de 14 caractères ASCII au maximum,appelé identificateurs Il est composé de blocs de données et de bloc de descriptif de début d'enregistrement qui contient:
-l'identificateur de l'enregistrement
-six octets de réservés
-l'adresse de chargement mémoire du bloc de données qui suit le bloc descriptif
-l'adresse de fin du bloc de données
-la fin d'un enregistrement est indiquée par un bloc descriptif de fin d'enregistrement avec une adresse de chargement de bloc nulle. L'adresse de fin de bloc est remplacé par une adresse d'exécution du bloc de données précédent.
L'ensemble bloc descripteur de début-bloc de données et bloc descri--pteur de fin est encadré par des blocs dits blocs "fin de fichiers" Voir schéma

A-4/COUPLEUR DU DEROULEUR DE BANDE MAGNETIQUE

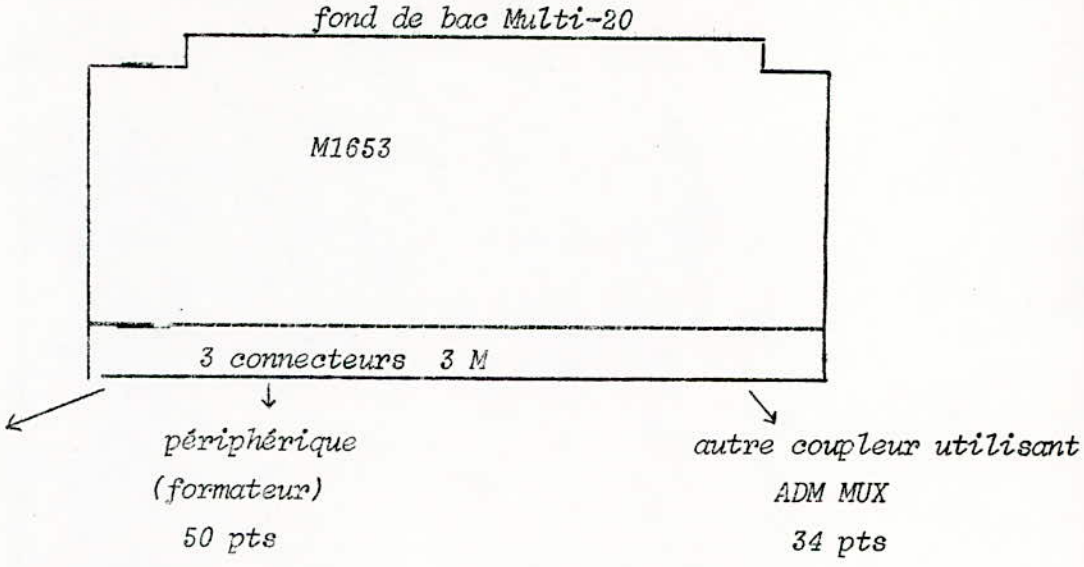
a/Description générale

Le coupleur M1653 assure la liaison Multi-20-unité de bande. L'accés au Multi-20 se fait en transfert par bloc en accés direct mémoire par le coupleur A.D.M. .1662

L'interruption est du type externe.

Le coupleur est constitué d'une carte enfichable.

FORMAT :



-alimentation + 5 V (2,5 A)

-circuits intégrés TTL-MSI

-sorties logiques à collecteurs ouvert

-les sorties vers le périphérique sont cablées au + 5 V à travers une résistance de 2 Kiloohms

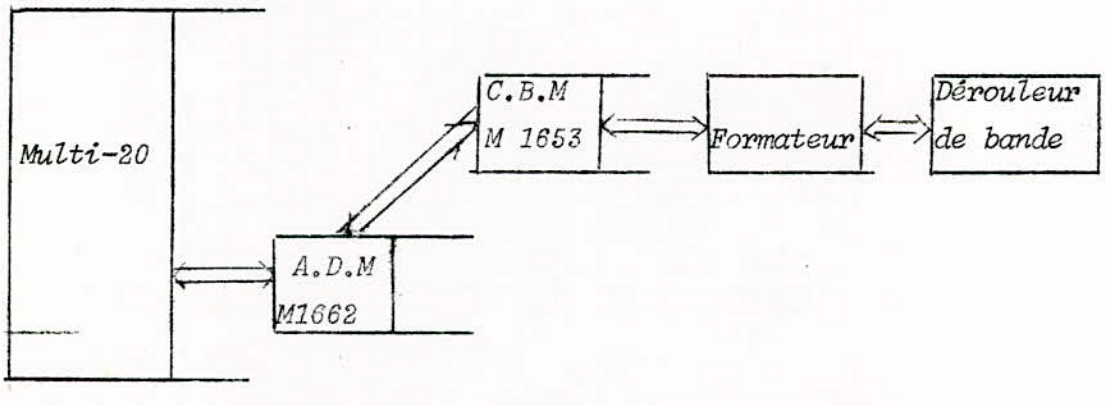
b/Spécification des unités de bande

Une unité de bande comprend un formateur et un dérouleur.

Le rôle du formateur est de créer des enregistrements au format IBM de détecter les erreurs d'enregistrements ou de lecture, de gérer les commandes du dérouleur.

La vitesse du dérouleur utilisable peut varier de 12,5 à125 micro-secondes.

c/Configuration de base d'un systeme d'enregistrement sur B.M.



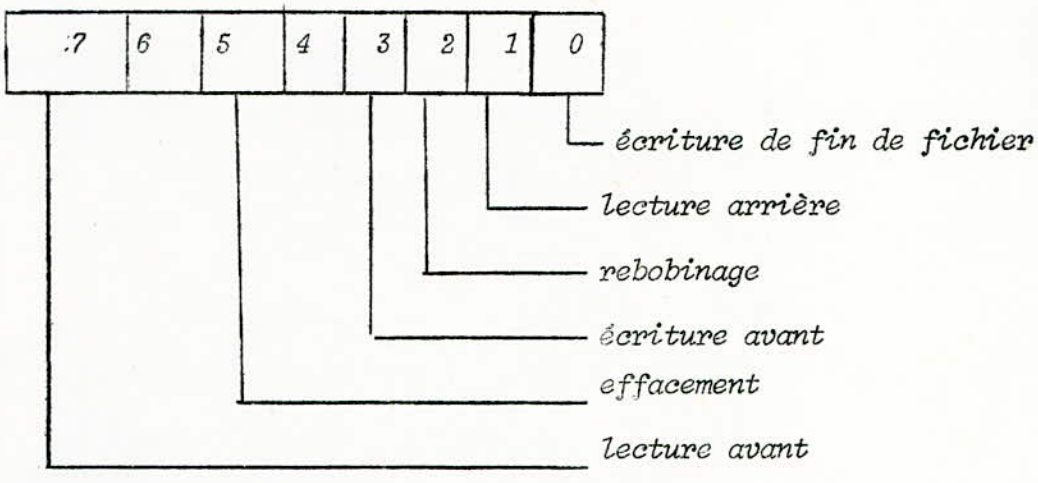
Un système comprend nècessairement une carte coupleur M1653, un formateur et un dérouleur de bande ce qui permet un transfert de bloc

entre le Multi-20 et la bande magnétique.Le coupleur M1662 est indispensable,il régit les accés direct à la mémoire .

c/Programmation du coupleur de la bande magnétique

OBA 0,7 :chargement d'adresse de bloc poids faibles

OBA 1,7 :fonction demandée au coupleur et armement de l'inter-ruption pour un transfert de donnée.

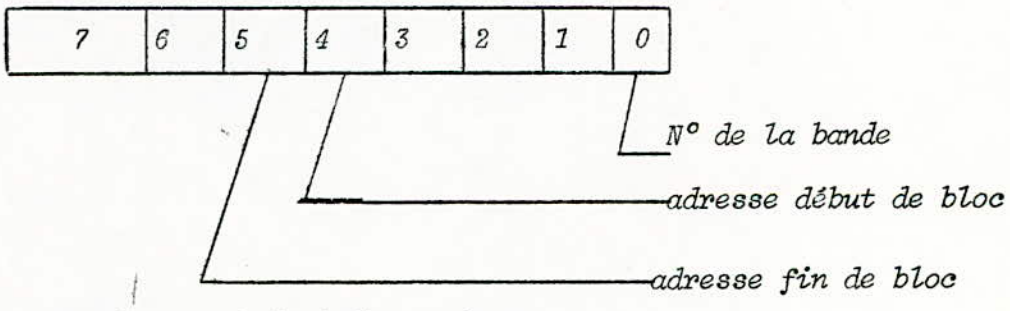


OBA 3,7 :chargement adresse de bloc poids forts

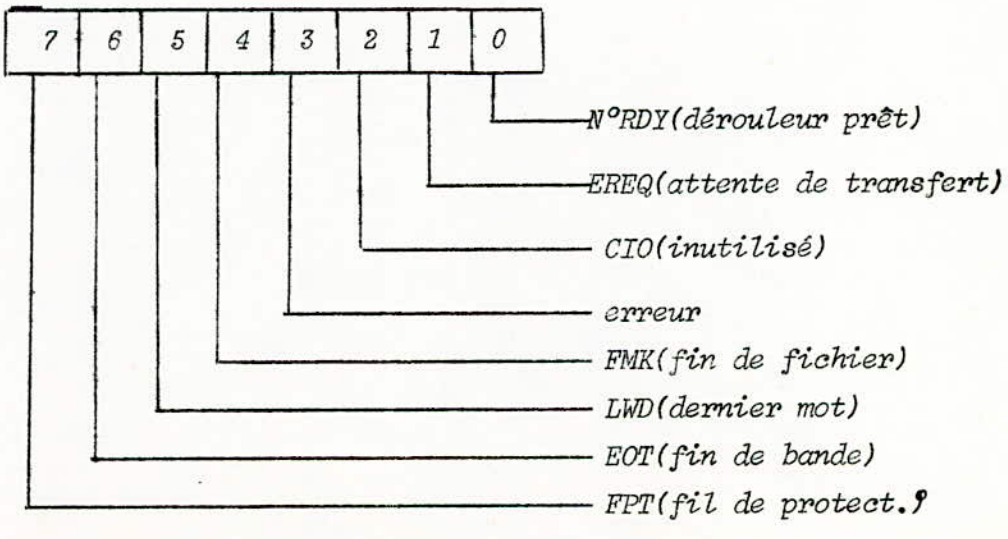
OBA 4,7 :déconnexion provoque une interruption si le coupleur est armé

OBA 5,7 :désarmement de l'interruption de fin de bloc

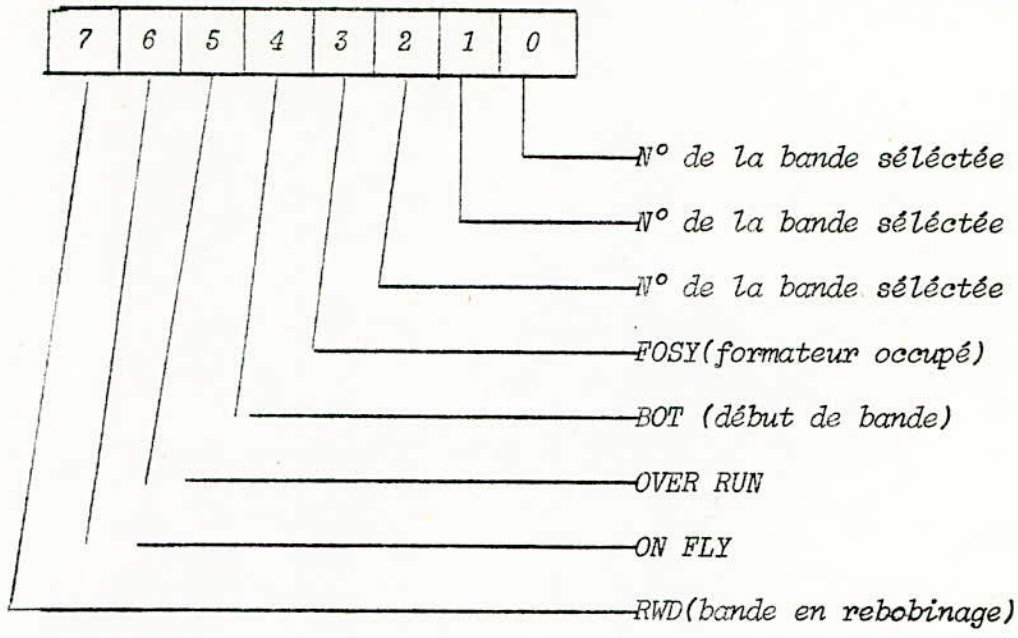
OBA 7,7 :mot de commande



IBA 1,7 :mot d'état général du coupleur



IBA7,7 :mot d'état principal



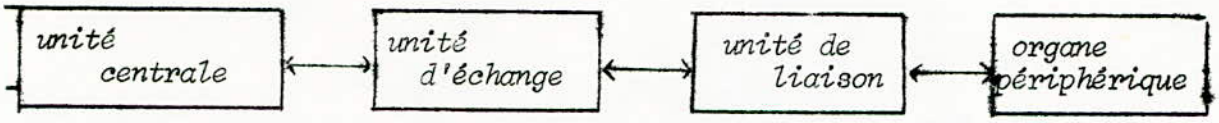
L'entrée du mot d'état général IBA 1,7 permet en particulier de savoir si le dérouleur de bande est prêt et s'il y a eu une erreur pendant le transfert.
L'entrée du mot d'état principal permet en particulier de connaître le numéro de bande selectéé (nombre de 3 bits°)dans le cas où il y a plusieurs dérouleurs de bande.

II/ORGANES D'ENTREES-SORTIES

1/GENERALITES

La communication du calculateur avec l'extérieur est assurée par les organes d'entrées-sorties.
Cet échange d'information avec l'extérieur nècessite un support inter--médiaire qui a pour rôle de stocker les informations,dans notre cas ce support est la bande magnétique.
Cependant la grande variété des vitesses de fonctionnement de l'unité centrale et de la bande magnétique pose un problème de raccordement du calculateur à la bande magnétique;ce problème est traîté au sein de l'unité d'échange,les problèmes spécifiques à chaque type de périphérique sont traîtés au sein de l'unité de liaison.

Un schéma général de raccordement d'un organe périphérique à l'unité centrale est le suivant:



L'interface entre l'unité d'échange et l'unité de liaison est normalisée :

l'organe périphérique est au contraire propre à chaque organe.

L'unité d'échange est chargée de gérer les échanges avec l'unité centrale du calculateur et de présenter ces informations sur l'inter-face normalisée avec une unité de liaison.

2/CANAL A.D.M.

Dans notre cas, l'unité d'échange est pratiquement inexistante et la gestion des transferts se fait de façon programmable. Nous devons donc tester dans nos programmes, avant chaque transfert si la bande magnéti-que est prête à fournir ou à recevoir l'information; pendant ce temps, l'unité centrale est occupée en permanence à éxécuter en répété une instruction de test et d'attente de cette disponibilité de l'organe.

Remarquons à cet effet que nous avons axé notre choix sur le mode par test d'état et non le mode bloqué, car ce dernier consiste à donner un certain délai à la bande magnétique afin de lui permettre d'être prête tandis-que le mode par test d'état consiste à déclencher le transfert dès-que le dérouleur est prêt. On voit l'avantage à cet effet qu'on améliore le pourcentage actif de l'unité centrale.

Pour augmenter l'efficacité du procédé, on effectue généralement les transferts d'information par "bloc" d'un certain nombre de mots ou de caractères.

Remarque: l'efficacité étant définie comme le rapport en pourcentage, du temps théorique nècessaire pour l'exécution d'une instruction avec le temps réèl mis par le calculateur (éxecution +attente que le périphérique soit prêt)

COUPLEUR M1662

Le canal A.D.M. M1662 est placé entre le périphérique dérouleur de bande et l'unité centrale; et ce pour réaliser les fonctions d'entrée sortie des informations sans passer par les registres et pour adapter également les vitesses de transmission nettement différentes.

Le M 1662 est transparent aux commandes d'entrée-sortie; il doit être initialisé par:

-un chargement des adresses début et fin de transfert

-un ordre de départ de transfert de bloc.

Le M 1662 contrôle l'interruption fin de bloc; si cette inter-ruption est armée; c'est une interruption interne.

Pendant le travail, en conjonction avec la mémoire du calculateur, le M 1662 doit, pour chaque mode de fonction, définir les adresses où l'information doit êrte lue ou écrite.

Les adresses de commencement et de fin des fonctions A.D.M. ainsi que le départ de la fonction elle-même sont programmées.

L'arrêt est automatique et se produit à la fin de bloc à moins qu'il ne se produise une interruption interne.

L'utilisation simultané d'un autre type A.D.M. n'est pas permis.

PROGRAMMATION DE L'A.D.M. M1662

Sur disque ou sur ruban magnétique

OBA 0,16 : démarrer un transfert de bloc

OBA 1,16 : démarrer un transfert de bloc et permettre une interruption interne à la fin du bloc

OBA 2,16 : sortie d'une zone mémoire et remise à zéro des contenus

OBA 3,16 : sortie d'une zone mémoire, remise à zéro des contenus et possibilité d'une interruption interne à la fin du bloc.

OBA 4,16 : charger l'octet de poids faible de l'adresse début

OBA 5,16 : charger l'octet de poids fort de l'adresse début

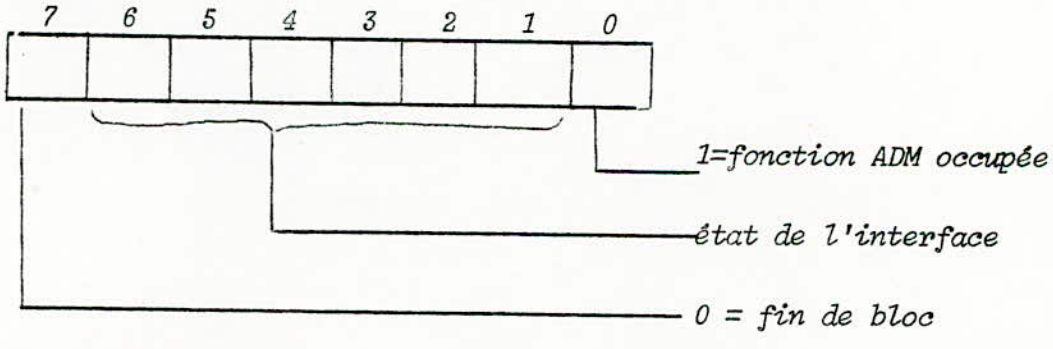
OBA 6,16 : charger l'octet de poids faible de l'adresse fin

OBA 7,16 : charger l'octet de poids forts de l'adresse fin

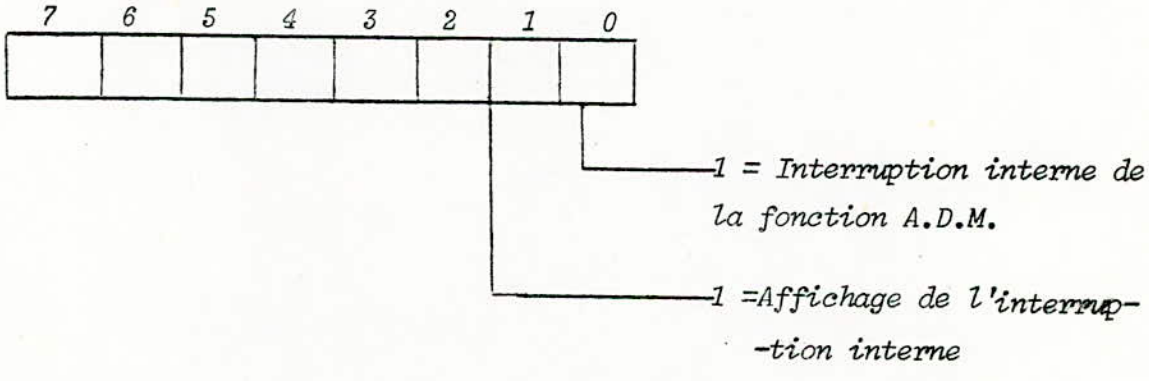
IBA 0,16 : lecture du mot d'état (1)

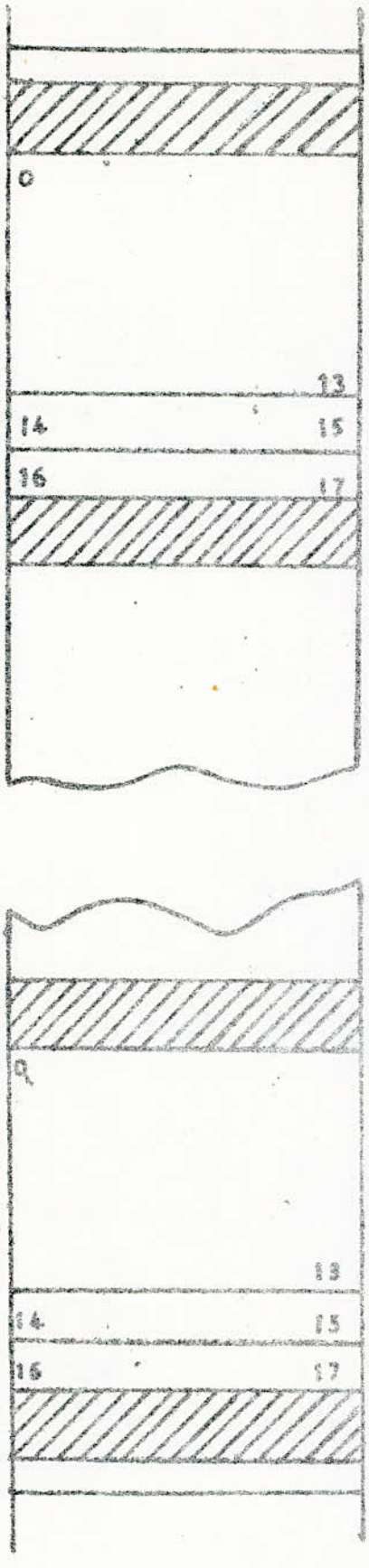
IBA 1,16 : lecture du mot d'état de l'interruption interne et annu-lation de l'interruption interne (2).

(1) configuration



(2) configuration





Bloc fin de fichier

Espace Inter-Bloc

Bloc descriptif

- 0 - 13 - Nom du programme
- 14 - 19 - 6 octets de sauvegarde
- 20 - 21 - Adresse début bloc
- 22 - 23 - Adresse fin bloc

Espace inter-bloc

Bloc de données

Espace inter-bloc

Bloc descriptif
Fin
d'Enregistrement

- 0 - 13 - Nom du programme
- 14 - 19 - 6 octets de sauvegarde
- 20 - 21 - Adresse d'exécution si différent de 0
- 22 - 23 - Adresse 0

Espace inter-bloc

Bloc fin de fichier

Format d'un enregistrement sur bande magnétique

CHAPITRE III/ S Y S T E M E B O S

A/ CHARGEUR ELEMENTAIRE BANDE MAGNETIQUE :CARBAM

A-1/Définition

La lecture d'un programme, la mémorisation des instructions en mémoire centrale nècessitent un certain nombre d'opérations séquentiel--les qui doivent obligatoirement être programmées dans des instructions car l'ordinateur ne fait qu'exécuter des opérations conformément aux ordres qui lui sont données dans les instructions.
Nous avons réalisé à cet éffet, un programme chargeur qui a pour rôle de lire le premier enregistrement sur bande magnétique, en mémoire centrale; ce premier enregistrement est l'utilitaire MET .
Seulement comment lire et faire rentrer en mémoire centrale le program--me chargeur élémentaire lui-même?
Il est impossible de faire celà par programme puisqu'on suppose que lors du chargement du système, la mémoire centrale est complètement vide (cas le plus critique); c'est par un jeu des clés de test (microprogram--me de chargement: harduare) du pupitre de l'ordinateur que l'on assure le chargement de CARBAM en mémoire centrale: c'est ce que l'on appelle initialisation.

A-2/Préliminaire sur la programmation :

a: Rappels:

Un programme est essentiellement un "horaire de changement " que vont suivre, avec ponctualité, les circuits logiques ou groupement de circuits logiques formant le calculateur.
Le pilotage du calculateur est assuré par la suite d'instructions que constitue le programme. Cette suite d'instructions peut-être rompue en différents endroits par une instruction spéciale dite de branchement ou de saut. La rupture de séquences peut être soit:
-systématique, c'est alors un branchement inconditionnel
-soit soumis à la présence d'une condition logique, c'est alors un branchement conditionnel.
Dans ce dernier cas, il est sous-entendu que lorsque la condition n'est pas remplie, l'enchaînement se fera sur l'instruction suivante dans la liste du programme. Notons à ce niveau l'importance des branchements conditionnels puisque c'est par eux que s'exerce la possibilité de faire prendre par la machine une décision logique quelconque , ne serait-ce que celle de s'arrêter en fin de travail.

Une fois le programme élaboré, on le communique à la machine où il occupe une zone mémoire choisie par le programmeur. Celui-ci prendra soin d'écrire son programme dans une zone adéquate afin d'éviter d'empiéter sur des zones occupées .

2/Notions de sous-programmes/

Un sous-programme est un programme particulier réalisant une fonction bien définie sur des données et fournissant des résultats comme conséquence de leur éxécution .La possibilité d'introduire cette structure modulaire donne toute la souplesse désirée pour construire des modèles qui permettent un travail relativement complexe .

3/ Appel d'un sous-programme:

Les sous-programmes conçus au cours de notre étude sont dits du type fermé .Les instructions de ces sous-programmes ne sont introduits qu'une seule fois et l'utilisateur pourra alors en faire appel autant de fois que nècessaire . Le transfert de la commande du programme principal vers le sous-programme se fait à l'aide de l'instruction "RTJ" qui signifie branchement vers le-dit sous-programme .
L'appel d'un sous-programme s'écrit de façon suivante:
"RTJ" adresse début du sous-programme
Outre l'appel au sous-programme cette instruction permet aussi de sauvegarder l'état du compteur ordinal ,par conséquent de l'adresse de retour, et ce au moment de l'appel.Pour ce faire ,les deux premiers octets du sous-programme sont réservès pour contenir l'adresse de retour au programmme principal.Aussi le calculateur ne commencera l'éxécution d'un sous-programme qu'à l'adresse début + 2 .
La dernière instruction du sous-programme permet le retour au programme principal.Cette instruction se présente sous la forme d'un branchement inconditionnel JMP adressé en indirect relatif de manière à permettre une translatabilité du sous-programme vers une autre région mémoire . Le contenu du compteur ordinal rétabli ,l'éxécution du programme principal peut se poursuivre .
En définitif ,l'instruction d'appel au sous-programme RTJ permet :
-de ranger le contenu du compteur ordinal dans les deux premiers octets du sous-programme (valeur du compteur ordinal au moment de l'appel).
-de recharger le compteur ordinal à la valeur adresse du sous-programme + 2 ,laquelle adresse représente l'adresse de la première instruction exécutable du sous-programme .

Le programme principal ainsi que tous les sous programmes suivants ont été écrits de manière à ne pas rester figés dans une zone mémoire donnée ;on a utilisé à cet effet l'adressage relatif afin de permettre une éventuelle translation de ces programmes vers d'autres régions mémoires .

A-3/ P R O G R A M M E C A R B A M

A-3-1/ SOUS-PROGRAMME LEBO

On désigne sous ce nom le sous-programme nècessaire pour la lecture d'un bloc de données .Ceci nous a conduit à la programmation du coupleur accés direct à la mémoire utilisé comme organe d'entrée-sortie pour dialoguer avec la mémoire centrale;c'est un des sous-programmes les plus délicats à réaliser.

Remarque:

1/En fin de lecture les cas suivants peuvent se présenter/ :

a/ La longueur du bloc sur la bande est égale à la longueur de la zone réservée en mémoire:la bande s'arrête automatiquement en fin de transfert.

b/ La longueur du bloc sur la bande est supérieure à la longueur de la zone réservée ,la lecture de bande continue jusqu'à l'espace inter-bloc .

c/LA longueur du bloc sur la bande est inférieure à la longueur de la zone réservée en mémoire;en fin de bloc ,la bande s'arrête automatiquement ainsi que le transfert.

Pour éviter tout problème de fonctionnement nous avons toujours utilisé le cas "a" ou pour plus de précautions le cas "c".

2/ L'instruction OBM 7,7 01D6 est équivalente à LDV 10 et OBA 7,7

Le listing d'un tel sous-programme est le suivant:

```
RTJ  2C          branchement au S/P état bande magnétique
OBM 7,7 01D6     .
LDA-             adresse de la table
OBA  0,7         charger adresse de bloc poids faible(début)
OBA  4,16        " " " " " " " " " " " " " " " " " " " " " "
LLA  08
OBA  3,7         charger adresse début poids forts
OBA  5,16        " " " " " " " " " " " " " " " " " " " " " "
OBM  7,7  01D6   charger 20 dans l'accumulateur A
LDA- 02          adresse de la table + 2
OBA  0,7         adresse de fin poids faibles
OBA  6,16        " " " " " " " " " " " " " " " " " " " " " "
LLA  08          décalage pour obtenir l'octet de poids fort
```

```
OBA  3,7          adresse de bloc poids forts (fin )
OBA  7,16         " " " " " " " " " " " " " "  "
OBA  0,16         début de transfert de bloc
LDV= 80           lecture avant
OBA  1,7
OBA  5,7          désarmement de l'interruption fin de bloc
RTJ/ 0264  6      branchement au S/P E.B.M.
JMP  DO           retour au programme principal
```

voir organigramme LEBO

SOUS-PROGRAMME LEKTEP

Sous-programme de lecture avec test erreur de parité.
Ce sous-programme nous permet de faire une lecture d'un bloc avec un test d'erreur de parité;il complète le S/P précédent auquel il fait appel, en effet en cas d'une erreur de parité (erreur de lecture),il permet de refaire la lecture de même bloc et ce jusqu'au quatrième essai consécutif. Si au bout de qatre essais consécutifs,l'erreur persiste toujours,il imprime "ER" sur la télétype en faisant appel au S/P de sortie de caractères "OTA", et il y aura arrêt du calculateur
L'opérateur pourra intervenir pour reprendre un nouvel essai éventuellement en refaisant les mêmes manipulations;
Ce S/P occupera la zone mémoire allant de 01E9 à 0227.
Le listing d'un tel sous-programme est le suivant:

```
          -- --          adresse du compteur ordinal
STX=      -- --         adresse du descriptif
LDV=  FD                compteur d'essais = 4
STV=  --                pointeur d'essais
LDV=  F8
RTJ   34                appel au sous-programme LECO
LDA   6B                mot d'état
ANA=  08  20
NAZ   04
LDA   64
JMP   E8                retour au P/P si pas d'érreur
LDV   EE                recharger le compteur d'essais
INA                     incrémenter le compteur d'essais
STV   EB
JAZ   04
RTJ   68
JMP   E6                refaire un autre essai
LDV=  C5                imprime "E"
RTJ   BF                appel au S/P OTA
```

```
LDV= D2          imprime " R "
RTJ  BB          appel au sous-programme OTA
LDV  8D
RTJ  B7
LDV  8A
RTJ  B3
HLT
```

Voir organigramme LEKTEP

SOUS-PROGRAMME OTA

Ce sous-programme est nécessaire à la sortie du message "ER" sur la télétype (erreur de lecture) et permet l'arrêt du calculateur.
Il occupe la zone mémoire allant de 01D9 à 01E8.
Comme ce S/P fait appel à des instructions d'entrée-sorties, on ne demande au calculateur de faire sortir un caractère sur le périphérique (télétype) que lorsque ce dernier est prêt.
On est obligé donc de
1/ tester l'état du périphérique: le bit A_2 est égal à 1 quand le périphérique est prêt à recevoir un octet (code ASCII) du calculateur;
2/ ne demander le transfert d'octet que (et dès-que) le périphérique est prêt.
Le périphérique n'est prêt que lorsque le contenu de l'accumulateur A (format sumple précision) est égal à 04 c'est-à-dire il n'y a que le bit A2 qui est égal à 1

7	6	5	4	3	2	1	0	
0	0	0	0	0	1	0	0	prêt à recevoir une commande

L'instruction ANV, consistant en un produit logique du contenu de l'accumulateur A avec l'opérande 04 , donne un résultat dans l'accumulateur
-soit nul: donc le bit A_2 EST DIFF2RENT de 1 ⟶ attente
-soit égal à 4 , dans ce cas le bit A_2 = 1 donc il est prêt.
Le listing d'un tel sous-programme est le suivant:

```
---  ---      réservation d'octets
STV= ---      sauvegarde de l'octet
IBA  1,0      demande de mot d'état
ANV  04       tester le bit A2
JAZ  FA       attente
LDV  F3       restitution de l'octet
OBA  0,0      sortie d'octet
JMP  EC       retour au programme principal
```

SOUS-PROGRAMME DBM

On a désigné sous ce nom le sous-programme qui nous permet de tseter si la bande magnétique est positionné à son début.
Ce suos-programme est utile pour faire un saut aarière, lors d'un nouvel essai de lecture en cas d'erreur de parité: on ne peut par ,exemple, rebobiner la bande si elle est à son début.
Il fait appel au sous-programme EBM (état bande magnétique) ,on teste le mot d'état principal qui est dans l'accumulateur A .
Si l'on est en début de bande le bit A4 est égal à 1 donc le contenu de A est égal à 10 .Le test de début de bande se fait par un "masquage" assuré par l'instruction ANV (intersection format simple).
Le sous-programme DBM occupe la zone mémoire allant de 0270 à 283
Le listing d'un tel sous-programme est le suivant:

```
-- --          octets réservés pour l'adresse de retour
RTJ  E4         appel au S/P EBM
ANV= 10         tester le bit A4
JAZ  05
LDV  FF
OCA
JAZ  F2
LDV= 02         lecture arrière
OBA  1,7
JMP  ED
```

voir organigramme DBM

SOUS-PROGRAMME EBM:

Ce sous-programme permet de tester l'état de la bande magnétique. Il permet ainsi de lire les mots d'état géneral et principal qui indiquent tous les cas possibles d'états du dérouleur. Il assure également le rangement des mots d'états.
Il occupe la zone mémoire allant de 0258 à 026E
Le listing d'un tel sous-programme est le suivant.

```
-- --          2 octets de réservés pour l'adresse de retour
IBA  1,7        entrée mot d'état général
LLA  08
IBA  7,7        entrée mot d'état principal
STA= -- --      rangement des mots d'états
ANA= 01 08      test
SBA= 01 00          bande
NOP                     prête
NAZ  EE         attente
LDA  F3
JMP  E8         retour au programme principal
```

Voir organigramme EBM

PROGRAMME PRINCIPAL "CARBAM"

On désigne sous le nom CARBAM le programme principal qui gère tous les sous-programmes précedents,en vue de charger le premier enregistrement,qui se trouve sur bande magnétique,en mémoire centrale Une table §(TAB) est une zone mémoire dont l'adresse début se trouve dans le registre index X ,elle contient les adresses début et fin de zone mémoire où doît être lue le bloc d'information:voir listing.

L'étendue de chaque table (nombre d'octets dans une zone mémoire réservée) varie en fonction du nombre d'octets nècessaire à la lecture de chaque bloc.Néanmoins,dans certains cas,pour plus de sécurité,cette étendue est supérieure à la zone demandée.
Des essais préliminaires,nous ont permis de remarquer,qu'un espace inter-bloc séparait les deux zones réservées pour le bloc descripteur de début d'enregistrement et celui des données,ce qui a été omis dans le manuel de référence du Multi-20
C'est cequi nècessite donc deux tables différentes pour la lecture de ces deux blocs,car à chaque bloc doît correspondre une table.
La première table (TAB1) est réservée au bloc "fin de fichier" qui est signalée par deux octets seulement (13 13).
La seconde table (TAB2)réserve la zone mémoire 6100 à 6150 pour l'écriture du mot MET en code ASCII et pour spécifier les adresses début et fin des données à enregistrer en mémoire centrale.
La troisième table (TAB3) contient les adresses début et fin du bloc de données,c'est-à-dire les adresses entre lesquelles doît êrte lu en mémoire centrale,l'utilitaire MET.
Enfin la quatrième et la dernière table (TAB4) est réservée pour le nom du programme et l'adresse de lancement du programme enregistré(MET) qui est 74FF.
Le listing de ce programme est le suivant:

```
LDX= 01  C6
RTJ  36
LDX= 01  CA
RTJ  31
LDX= 01  CE
RTJ  2C
LDX= 01  D2
RTJ  27
JMP  74  FF
TRP
```

TABLE DU DESCRIPTIF ET ZONE DE MEMOIRES RESERVEES

TAB1

01C6	
	6 1
	0 0
	6 1
	1 0

TAB2

01CA	
	6 1
	6 0
	6 1
	A 0

TAB3

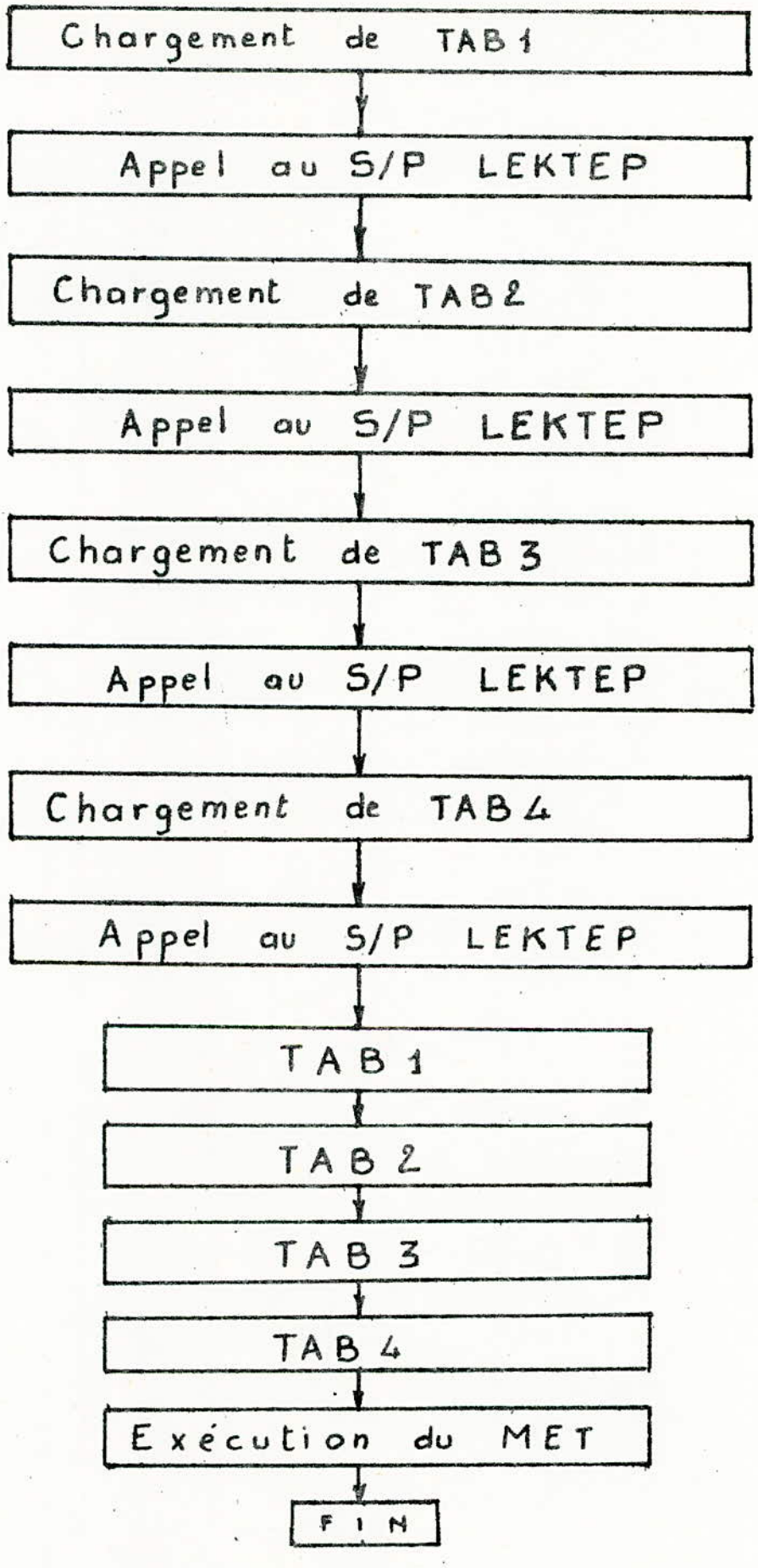
01CE	
	7 2
	0 0
	7 F
	F F

TAB4

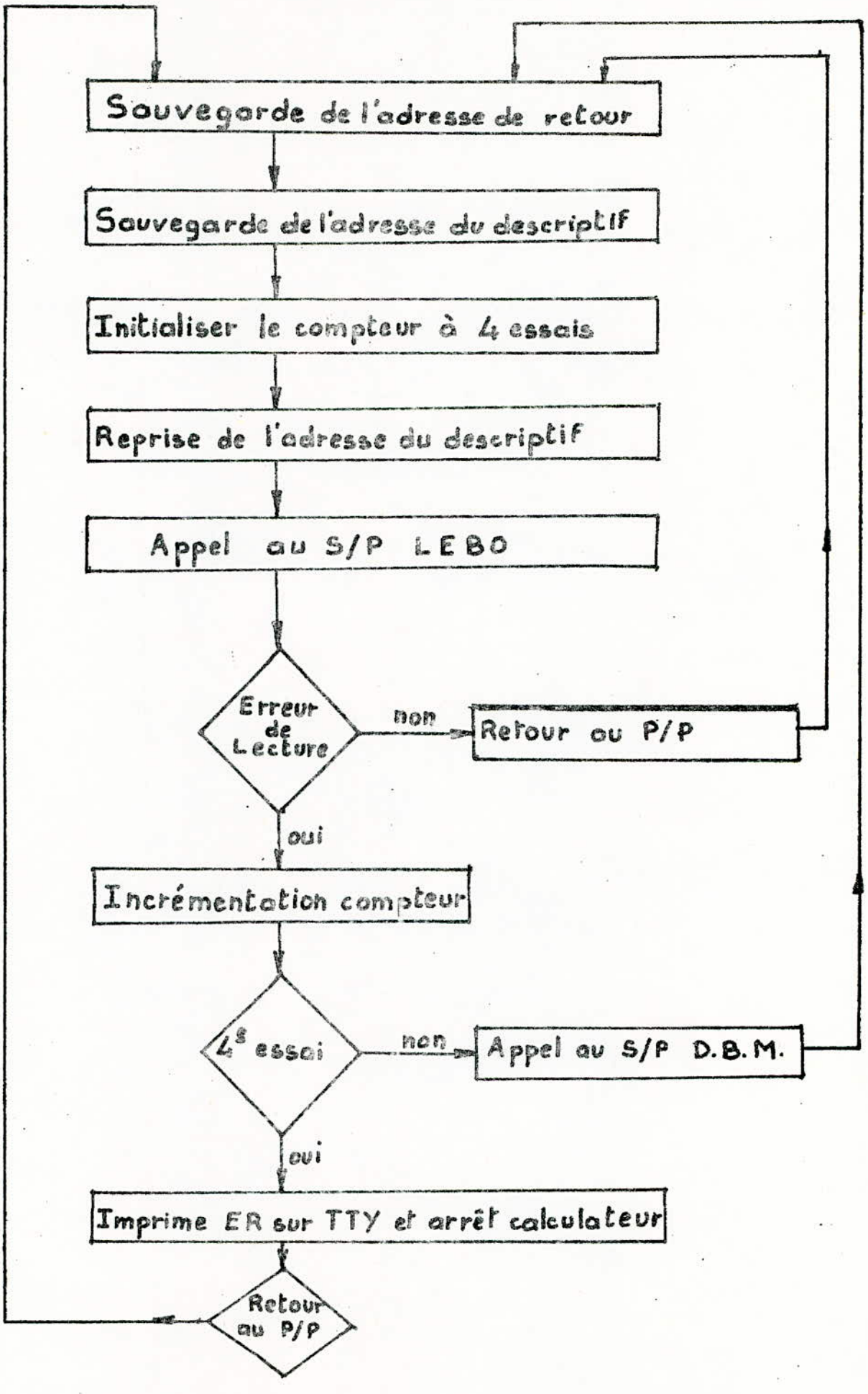
01D2	
	6 1
	7 0
	6 1
	F 0

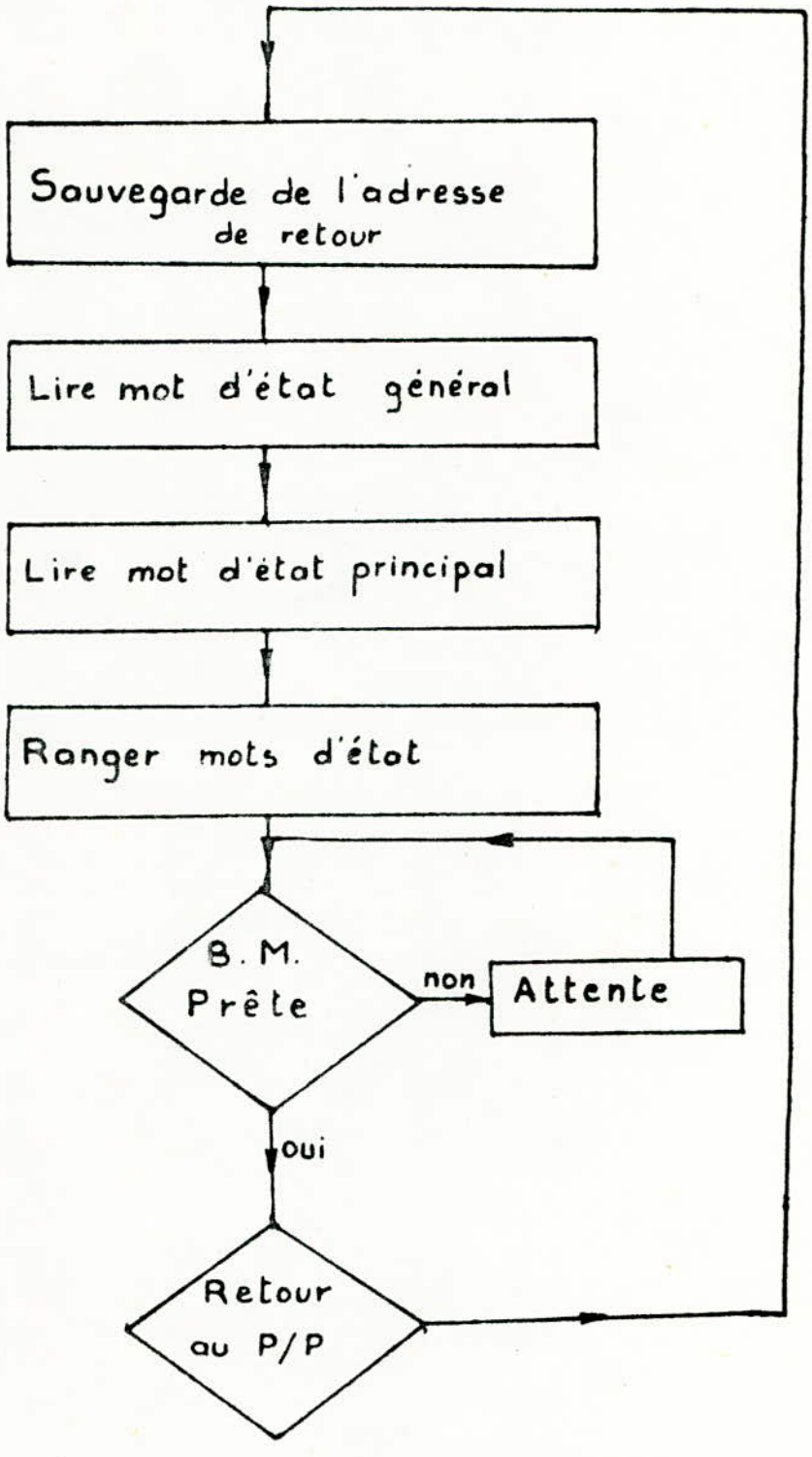
Voir organigramme page suivante

Programme Principal : carbam

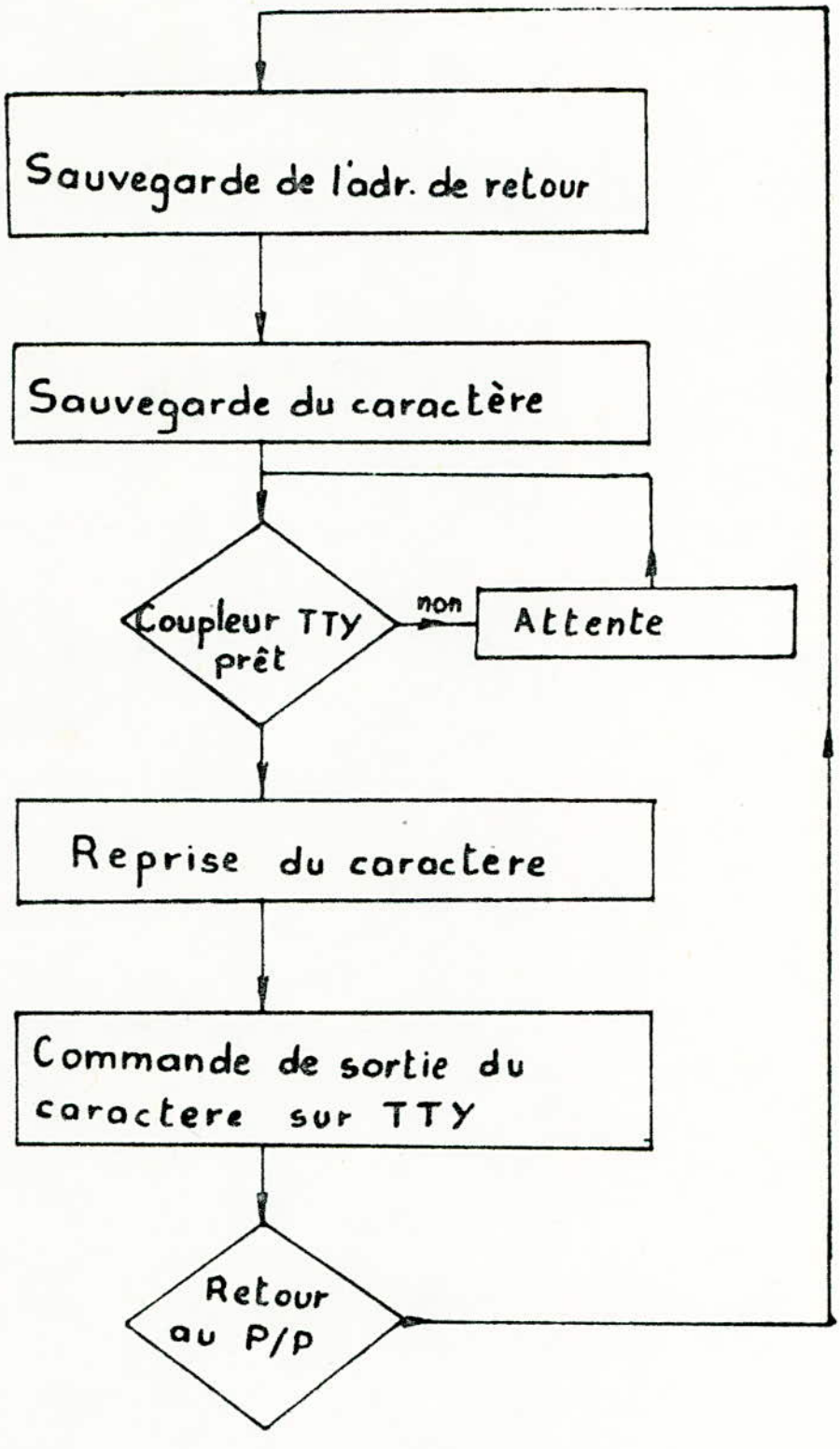


S/P LEKTEP

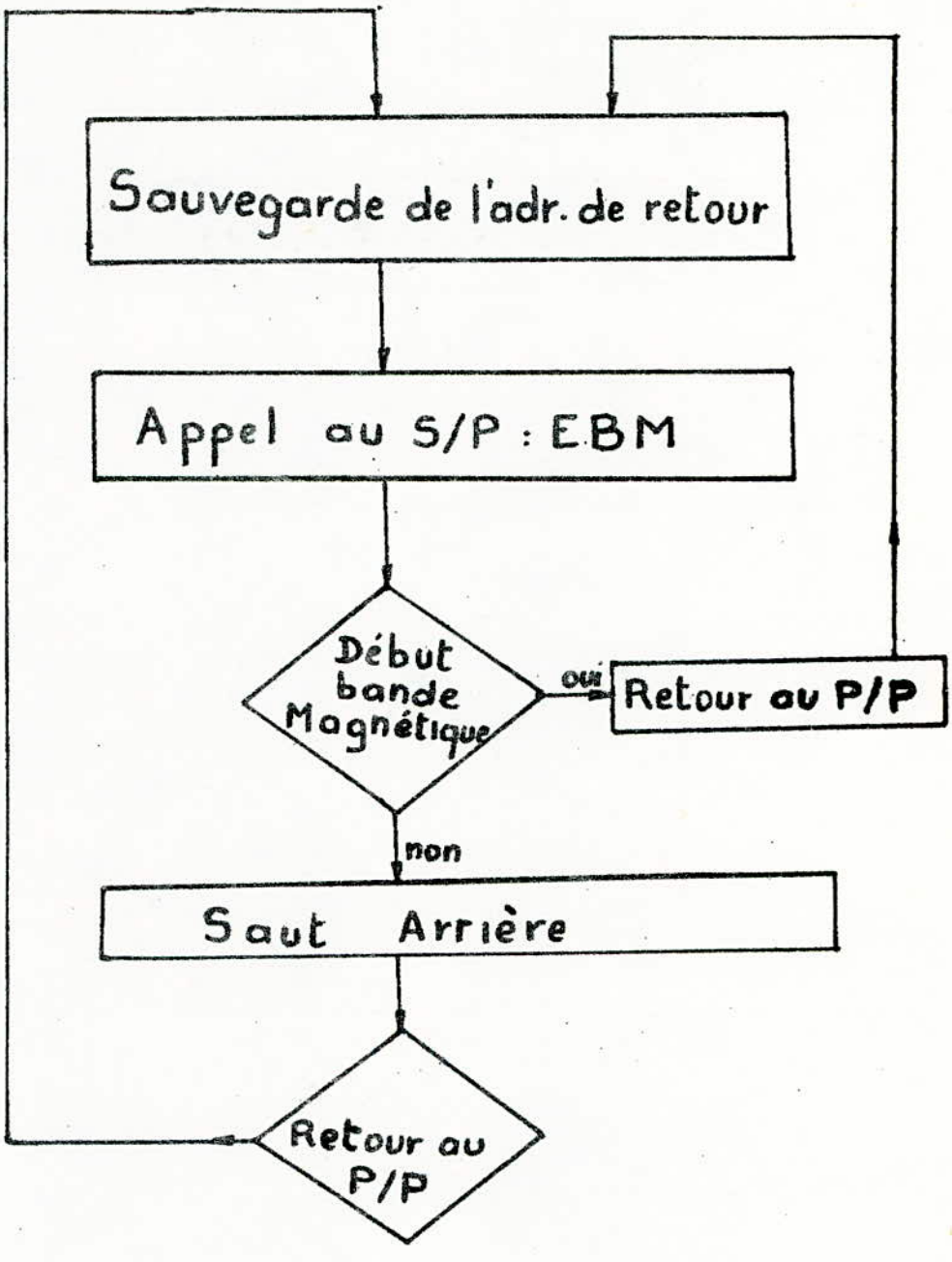

Souvegarde de l'adresse de retour
Souvegarde de l'adresse du descriptif
Initialiser le compteur à 4 essais
Reprise de l'adresse du descriptif
Appel au S/P LEBO
Erreur de Lecture
non
Retour ou P/P
oui
Incrémentation compteur
4e essai
non
Appel au S/P D.B.M.
oui
Imprime ER sur TTY et arrêt calculateur
Retour au P/P


Sauvegarde de l'adresse de retour
Lire mot d'état général
Lire mot d'état principal
Ranger mots d'état
B. M. Prête
non
Attente
oui
Retour au P/P

S/P E.B.M.

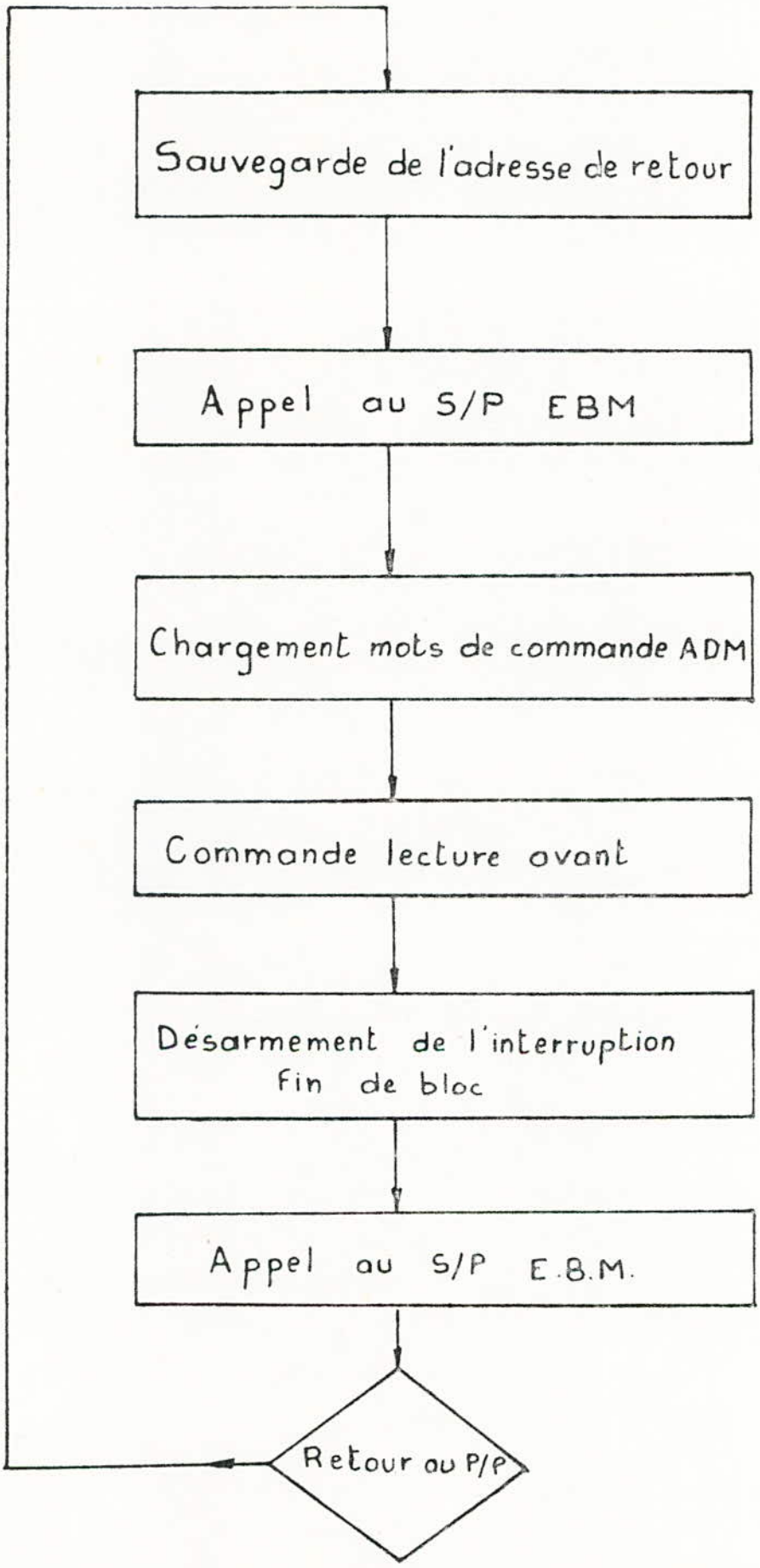

Sauvegarde de l'adr. de retour
Sauvegarde du caractère
Coupleur TTY prêt
non
Attente
Reprise du caractere
Commande de sortie du caractere sur TTY
Retour au P/P

S/P OTA


Sauvegarde de l'adr. de retour
Appel au S/P : EBM
Début bande Magnétique
oui
Retour au P/P
non
Saut Arrière
Retour au P/P

S/P D.B.M.

S/P LEBO


Sauvegarde de l'adresse de retour
Appel au S/P EBM
Chargement mots de commande ADM
Commande lecture avant
Désarmement de l'interruption
Fin de bloc
Appel au S/P E.B.M.
Retour au P/P

B/MONITEUR D'EXPLOITATION TELETYPE

C'est un programme utilitaire permettant le chargement, l'éxécution la mise au point et l'archivage de programmes en langage assembleur MULTI .

La commande du MET se fait au clavier du téléimprimeur (ou de la console de visualisation connéctée à la liaison dite "télétype" du calculateur. Seule la clé "INTERRUPTION" de la console est utilisée pour le retour au MET lors de l'éxécution d'un programme de l'utilisateur .Le MET est un logiciel de base qui permet d'éffectuer les opérations suivantes:

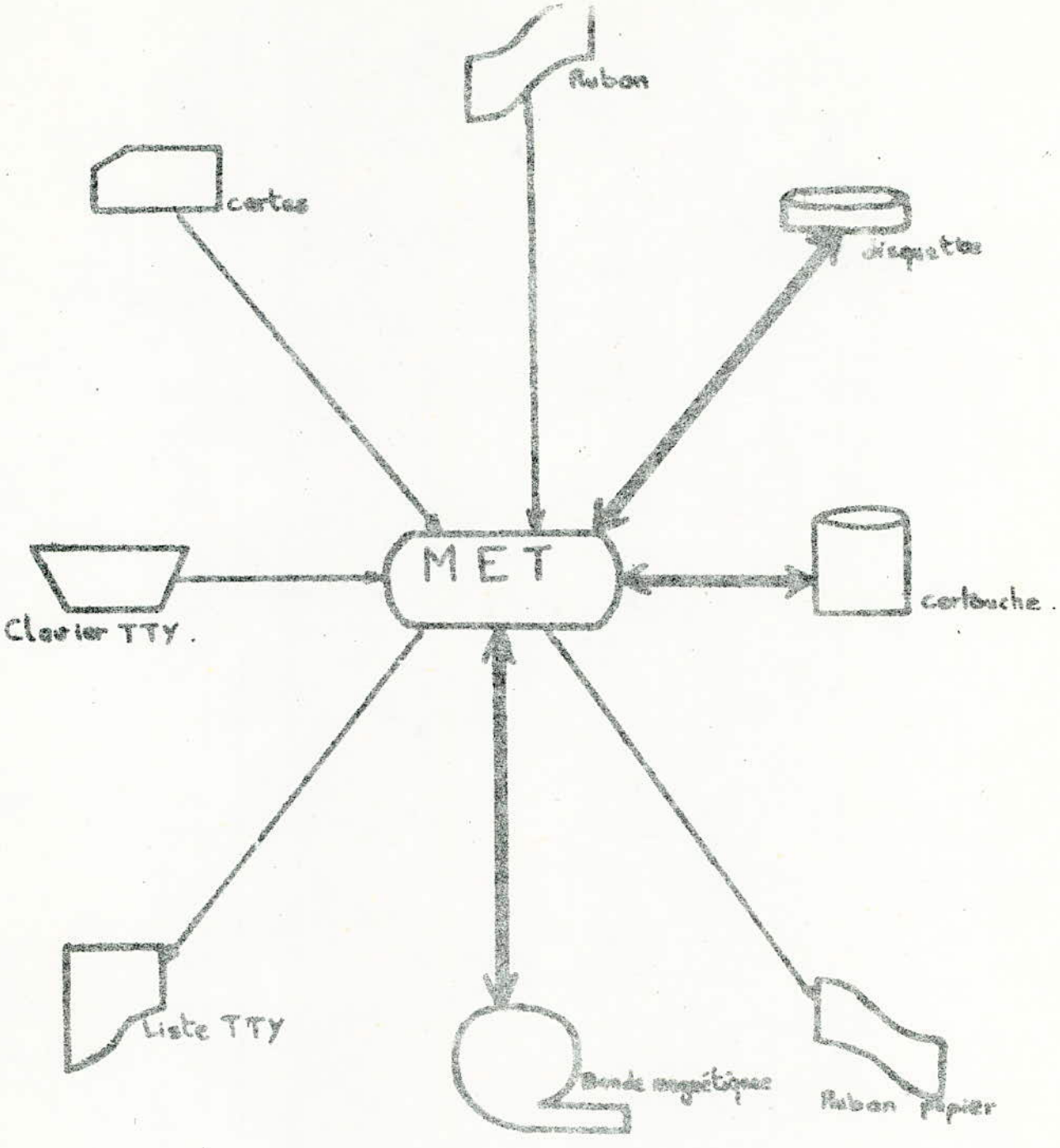
-impression du contenu d'une mémoire
-modification du contenu d'une mémoire
-déroutement d'un programme
impression du contenu d'une zone mémoire
-initialisation du contenu d'une zone mémoire
-impression du contenu des registres
-modification du contenu des registres
-éxécution du programme avec arrêt sur adresse
-calcul de la somme ou de la différente de nombres hexadécimaux.
-charger un programme binaire absolu depuis:
- un lecteur de cartes
- un lecteur télétype
- un lecteur rapide de ruban perforé
- un disque à cartouche
- une bande magnétique

-sauvegarder un programme binaire absolu sur:
ruban perforé
-disque à cartouche
-bande magnétique
-disque souple

-effectuer des transferts de données entre la mémoire et:
-un ruban perforé
-un disque à cartouche
-une bande magnétique
-un disque souple

Le MET est un logiciel de base permettant d'éviter toute manipulation au niveau de la console lors des tests et mises au point de programmes assembleurs.

L'interruption console est l'action de l'opérateur sur la clé 4"INT" du panneau avant du calculateur. Cette action permet lors du démarrage ou pendant l'exploitation de donner ou redonner le contrôle au MET .



les périphériques gérés par le mef

Lors d'une interruption console, les registres de la machine sont rangés dans une zone mémoire appelée registres virtuels.
La valeur du registre P précédée d'un astérisque, est imprimée sur le téléimprimeur au moment de l'interruption. Les autres registres peuvent être visualisés par frappe d'un espace aprés l'impression sur le téléimprimeur du registre précédent.
Si, lors de l'exploitation, l'utilisateur désire rendre la main au MET, il suffit, avant d'actionner la clé "INT", de mettre le calculateur en halte par action sur la clé "pas à pas" qui provoque un arrêt de l'ordinateur aprés éxécution de l'instruction en cours.
Si de plus, le programme effectuait des entrée-sorties, il suffit d'actionner la clé INI(initialisatio) qui réinitialise l'ensemble des coupleurs.

COMMANDES DU MET

La commande D permet de visualser une zone mémoire

La commande M permet de modifier une zone mémoire

La commande S assure le rangement des données en mémoire

La commande I permet d'initialiser un zone mémoire

La commande G permet de lancer une séquence d'instruction à une adresse donnée.

La commande K permet de restituer les adresses interruptions console et interruptions internes gérées par le MET ..

La commande H imprime la somme et la différence de deux nombres hexadécimaux.

C/ E D I T E U R D E L I E N S - C H A R G E U R

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

L'éditeur de liens-chargeur est un logiciel de base ayant pour fonction principale de résoudre les références entre modules d'un même programme et de transformer des modules translata--bles en modules objets absolus éxécutables.
Léditeur de liens est capable d'éditer soit directement en mémoire, soit sur les mémoires auxiliaires disque à cartouche,disque souple ou bande magnétique.
Les modules translatables peuvent provenir d'assemblages ou de com--pilations séparés,le chargeur-éditeur de liens assure les chaîna--ges entre les différents modules.
C'est un processeur de base indispensable pour l'écriture et l'orga--nisation de tout logiciel dépassant une centaine de lignes de prog--ramme.En effet,il permet l'écriture modulaire d'un programme,c'est-à-dire,un découpage en blocs logiques conférant au programme une li--sibilité accrue,d'où une maintenance plus aisée .Chaque module du programme constitue alors une unité d'assemblage,les liens entre ces unités étant résolus par l'éditeur de liens.
Le programme objet absolu peut être obtenu soit sur ruban perforé, soit directement en mémoire.Dans ce dernier cas,l'EDL peut lancer l'éxécution du programme généré.

L'E.D.L. assure en une passe,les fonctions suivantes:

- -chargement en mémoire de un ou plusieurs modules translatables ou absolus en effectuant les chaînages
- -mise en éxécution du programme édité en mémoire
- -création d'un programme objet absolu sur ruban perforé depuis un ou plusieurs modules translatables ou absolus.
- -édition de la table des références symboliques externes ou inter--nes accompagnée de leur adresse mémoire.
- -édition des références symboliques externes ou internes inconnus ou multi-définies.
- -plusieurs fonctions de service

Les options de l'éditeur de liens concernent le choix du support-source du module objet translatable et du support-but du module édité

- -le module objet translatable peut être sur un ruban perforé ou sur disque.
- -lédition peut s'éffectuer soit sur ruban,soit sur disque,soit directement en mémoire

DESCRIPTION DES COMMANDES DE L'E.D.L.

1/Commande RL aaaa :chargement et chaînage 5Read-Load)

La commande RL provoque la lecture d'un ruban objet produit par l'assembleur et son implantation à l'adresse aaaa (hexadécimale) -si le ruban lu est un ruban absolu,l'adresse lue sur celui-ci remplace aaaa;le chargeur vérifie que cette adresse est acceptable,car elle doit être impérativement:

-supérieure ou égale à 100_{16}
-inférieure à la table des symboles
-supérieure à l'adresse fin de l'E.D.L.

Si ce n'est pas le cas,la commande est refusée

2/Commande LR aaaa:chargement

Cette commande est identique à la commande RL aaaa, à la différence prés que aaaa ne spécifie plus l'adresse de début d'implantation,mais l'adresse de fin d'implantation
Si l'adresse aaaa est omise la lecture et l'implantation se fait à la première adresse disponible(dans le cas où le ruban est translatable car dans l'autre cas l'adresse prise esr celle lue sur le ruban).

3/ Commande RB

La commande RB permet de créer un ruban absolu à partir d'un ruban objet translatable(ou absolu),sans implantation réèlle en mémoire Cette implantation fictive,mais perforée sur le riban,est la pre-mière adresse disponible (contenue dans HIC)
Donc si on rentre un programme translatable à la suite d'un prog-ramme absolu,il ira s'implanter (implantation fictive) juste aprés le programme absolu.

4/ Commande RB aaaa

Cette commande est identique à la précédente à la différence prés que celle-là a sa première adresse disponible aaaa hexadécimale: (adresse HIC).

5/Commande EX

Commande de controle d'éxécution
La commande EX qui est une demande d'éxécution n'a de sens que si le chargeur a mémorisé une adresse d'éxécution(c'est-à-dire que le programme-source doit comporter au moins une instruction END nnnn)
Si plusieurs programmes ont été chargés,donc plusieurs instructions END,c'est la dernière adresse d'éxécution entrée qui est prise en compte.

6/ Commande RA

La commande RA permet de lire un ruban absolu et de l'implan-
-ter en mémoire.
Si le bloc lu comporte un END avec une adresse d'éxécution
différente de zéro, le chargeur effectue le lancement à cette
adresse.
Si l'adresse d'éxécution est nulle, le chargeur E.D.L. impri-
-me le symbole ! qui signifie:prêt à recevoir une nouvelle
commande.

7/Commande SM

impression de la table des symboles
En frappant SM, la télétype donne le listage des symboles de
la table ESD accompagnès de leur adresse.

8/Commande RS

La commande RS réinitialise le chargeur et efface la table des
symboles

voir schéma récapitulatif de fonctionnement de l'EDL ,page suivante.

• Définition des unités symboliques impression

Les impressions liées aux commandes SM et SI sont effectuées sur télétype (LIS = TF) ou imprimante Logabax (LIS = LC) ou imprimante Data Product (LIS = DP).

```
SCE = LR, BUT = ME (RC)

BUT = PR, SCE = DX, SCE = LR (RC)

SCE = LR, LIS = LC, BUT = DX (RC)
```

Figure A - Exemples de définition d'unités symboliques.

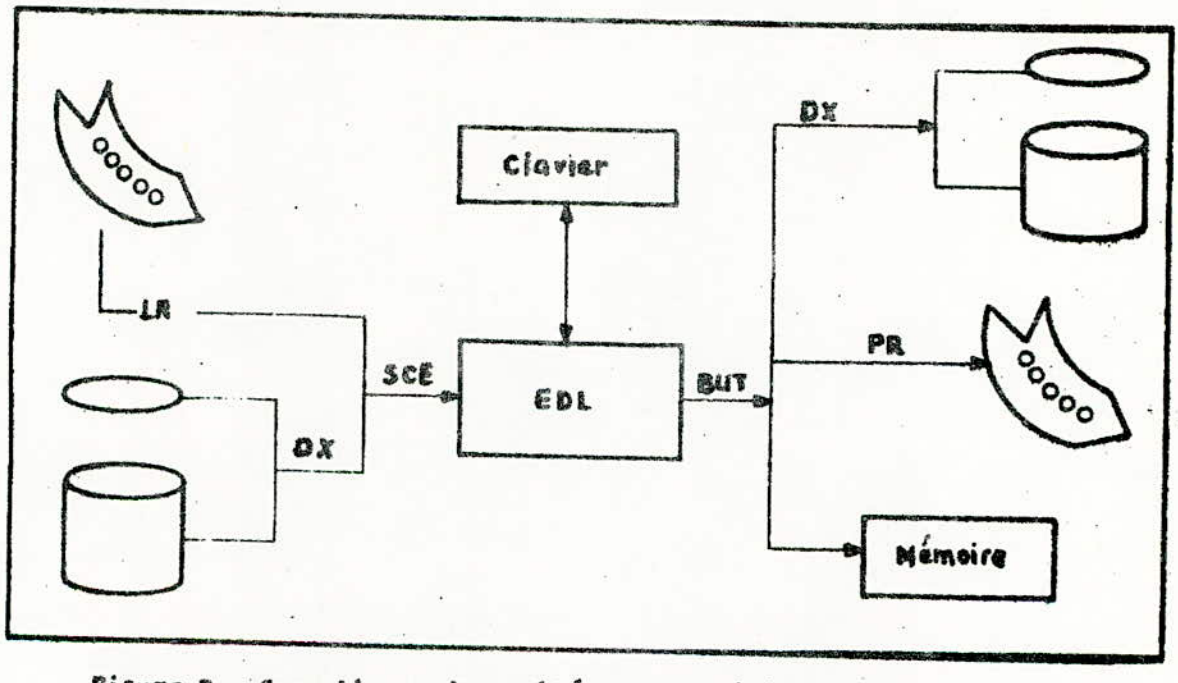


Figure B - Synoptiques des unités source et but.

ESSAIS PRELIMINAIRES ET TESTS DE L'E.D.L.

1/Chargement du MET

Le chargement du MET par ruban perforé se fait sur le lecteur rapide

* on abaisse les clés 1 et 4 du pupitre du Multi-20 qui indiquent qu'un chargement sera en cours.

*On place un programme amorce sur le lecteur de la télétype qui prévient la mémoire centrale qu'il va y avoir un chargement d'un programme sur ruban perforé à partir du lecteur rapide.

* Le lecteur rapide s'actionne automatiquement,et lit le ruban du MET. Ala fin de la lecture du ruban ,un tintement se déclenche sur la télétype: signe que le MET s'est chargé et éxécuté en mémoire centrale.

Le MET occupe la zone mémoire de 7200 à 7FFF .

2/Chargement de l'E.D.L.

Comme tout programme complexe,l'E.D.L. est un programme principal qui fait appel à de nombreux sous-programmes,et à certaines zones mémoire de données.

L'E.D.L. se présente sur ruban perforé,mais ne possède pas d'adresse d'éxécution à sa fin.(en ce qui concerne le premier bloc).

De ce fait,au début,quand l'E.D.L. défilait sur le lecteur rapide,il se chargeait seulement en mémoire centrale mais ne s'éxécutait pas.

PROBLEME:Recherche de l'adresse d'éxécution de l'E.D.L.

Ayant le listing du programme E.D.L. en mode translatable,on va utiliser ce dernier pour identifier l'adresse de lancement de l'E.D.L. absolu.Ces 2 programmes ont le même fonctionnement donc les mêmes sous-programmes mais possèdent quelques instructions différentes, bien sûr,puisque l'un est absolu et l'autre ,translatable.

L'adresse de lancement est primordial quand il s'agit d'un enregistrement sur bande;en effet la commande TB du dérouleur de bande permet seulement l'enregistrement de la zone mémoire du

programme sans adresse d'éxécution,cette dernière est enregistrée par une commande spéciale:la commande TF

Donc au moment de l'enregistrement sur bande,on doit suivre les instructions suivantes:

TB EDL ,adresse début,adresse fin

TF adresse d'éxécution

Pour celà on a été conduit à consulter le listing du programme MT.EDL(translatable).Le dernier sous-programme sur ruban fait une sortie de caractères;on a pu ainsi déterminer l'adresse d'éxécution, en refaisant le chemin inverse de la lecture d'un programme , exemple: le dernier sous-programme est appelé par un S/P A,qui lui même est appelé par un troisième B,et ainsi de suite jusqu'à aboutir à l'adresse d'éxécution.

EXEMPLE D'UTILISATION DU MA.EDL

* Chargement de l'EDL
* résultat : impression de ! (symbole qui remplace le message" prêt"
* génération d'un ruban absolu PER à partir d'un ruban translata-
-ble PER

RL 4000 (RC) pour le premier bloc

RL (RC) pour tous les blocs suivants

* on active le perforateur de ruban de la télétype par les commandes

L génération d'une amorce début de ruban

W 4000,4A00

L génération d'une amorce fin de ruban

D'où,un ruban perforé absolu du PER

D / E D I T E U R D E T E X T E : P.E.R.

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

1/ GENERALITES

Le programme d'Edition Ruban (PER 15)est un programme utilitaire permettant la génération et la correction d'un programme source sur ruban.Le programme sera ensuite assemblé.

La commande du PER s'éffectue à partir du clavier de la télétype, son lecteur-perforateur étant utilisé comme organe d'entrée-sortie. En option,un lecteur de cartes ou un lecteur-perforateur rapide peuvent être employés.

Le PER permet d'effectuer les opérations suivantes:

-lecture et numérotation des instructions du programme-source

-impression d'une ou plusieurs instructions

-suppression d'une ou plusieurs instructions

-insertion ou remplacement d'instructions dans le programme

-édition/correction d'une instruction,caractère par caractère

-saut d'un certain nombre d'instructions à la lecture du ruban .

-perforation d'une ou plusieurs instructions sous un format compact

-retour au MET

CHARGEMENT ET EXPLOITATION :

Le PER est toujours livré sous forme d'un ruban binaire format translatable comprenant 7 blocs,qui sont :

-éditeur avec entrée-sorties clavier télétype

-lecteur ruban télétype

-lecteur ruban sur le lecteur rapide

-perforateur de ruban sur le perforateur rapide

-lecteur de cartes

bloc de fin et de lancement

Le chargement de ce ruban ne peut être réalisé que par un éditeur

de liens EDL grâce auquel on pourra générer un ruban absolu, implanté selon les besoins de l'utilisateur.

* Les chargements de ce ruban absolu seront réalisée à partir du MET.
* Le lancement s'éffectue automatiquement en fin de chargement.
* La gestion des interruptions internes est laissée au MET qui peut résider simultanément en mémoire centrale.
* Chaque segment de ruban-source prend place en mémoire dans une zone tampon. Le nombre maximum de lignes admissibles par l'éditeur de texte est de 99.

ORGANES D'ENTREE-SORTIES DISPONIBLES

	T T Y	lecteur ruban	perfo. ruban	lecteur cartes
E/S clavier TTY	X			
Lecture prog. SOURCE	X	X		X
Perforateur ruban	X		X	

Toutes les opérations qui sont réalisées par le PER, débutent par la frappe d'un caractère alphabétique qui désigne l'une des commandes suivantes.

POINTS D'ENTREES DU PER

TED : séquence de contrôle, entrée des opérateurs

KTK : séquence d'initialisation, mise à zéro de la zone tampon

CSN : interruption console

Toute interruption console permet le retour à la séquence de con-trôle TED avec sortie du prochain numéro de lignes.

Les corrections du ruban-source sont éffectuées dans la zone tampon jusqu'à ce que l'utilisateur donne l'ordre de transcrire sur ruban perforé le programme définitif, cette sortie n'entraîne en aucun cas la remise à zéro de la zone tampon correspondante; ce qui nous créera des problèmes d'écritures de deux textes différents. En effet il fal-lait toujours lancer l'éxécution du PER à l'adresse de KTK qui

initialisait la zone tampon.

Le nombre maximum de lignes admissibles par l'éditeur de texte est de 99, cependant les numéros de lignes peuvent être choisis entre 0 et 99. Tout esai de débordement de ce nombre de lignes maximum, entraîne la sortie d'un message d'érreur, précédé d'une temporisation d'activation de la télétype.

EXEMPLE D'UTILISATION DU PER

Génération d'un ruban-source en langage assembleur.

Opérations à suivre:

* faire une entrée à partir du clavier télétype à l'aide de la commande T m,n,k où m est le numéro de la première ligne
  n est le numéro de la dernière ligne
  k est le pas d'incrémentation
* le programme imprime alors le N° de la ligne et l'opérateur frappe la ligne d'instruction qu'il termine par un retour chariot
* Une fois le texte terminé, l'opérateur génère une amorce début de 15 cm à l'aide de la commande L
* L'opérateur pourra ensuite perforer sur un ruban toutes les lignes à l'aide la commande W m,n (RC), toute ligne perforée est accompagnée de la perforation d'un caractère (Retour Chariot), d'un caractère LF (Line Feed) et de deux caractères d'éffacement (Rub-out).
* On termine la perforation du ruban en générant une amorce fin de ruban à l'aide de la commande L.

D'où un ruban perforé d'un texte en langage assembleur(code mnémonique) .

E / A S S E M B L E U R : M.A. P A T

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

1/Généralités :

Il permet d'assembler sur Multi-20 ,les programmes écrits en langage assembleur Multi-20 ,il nècessite deux passes du programme-source et produit un programme-objet binaire en adressage absolu.

Il utilise comme organe d'entrée-sortie la télétype-série avec lecteur et perforateur de ruban.

Au cours de la première passe,l'assembleur lit le programme source constitue une table des symboles en affectant une valeur à chaque symbole. Les symboles non définis sont détéctés au cours de cette passe et sont imprimés à la fin de la passe.

Au cours de la deuxième passe,l'assembleur lit de nouveau le prog--ramme source et génère un ruban binaire accompagné d'un listing d'assemblage.

Chaque ligne du listing comprend l'adresse héxadécimale du premier octet de l'instruction,la valeur hexadécimale des octets de l'inst--ruction,les codes d'érreurs et le programme symbolique.

Le langage d'assemblage offre les possibilités suivantes:

-calcul d'adresse

-définition des données

-contrôle du listing :possibilité d'insertion de commentaires

-messages d'érreurs

2/ LANGAGE ASSEMBLEUR

Le langage source est constitué d'une suite d'instructions symbo--liques perforée sur carte ou sur ruban papier.

Chaque ligne de programme comprend 4 zones:

-une zone étiquette

-une zone opérateur

-une zone opérande

-une zone commentaire

Une ligne de programme comprend au maximum 12 caractères.
Le ruban programme préparé sur une télétype en autonome a un format libre:un ou plusieurs espaces peuvent être placés entre chaque zone.Chaque ligne de programme doit se terminer par un retour chariot,un interligne et deux caractères d'éffacement.
Le ruban programme peut être préparé avec le programme PER qui offre plusieurs avantages.

DESCRIPTIONS DES ZONES

a/Zone étiquette

Elle contient un symbole constitué de 1 à 6 caractères,débutant à la colonne 1 et se terminant au premier caractère blanc rencontré (seuls les trois premiers caractères sont pris en compte par l'assembleur et permettant de distinguer les symboles).La présence d'un symbole en zone étiquette est optionnelle. Si le premier caractère est un astérique,la ligne de programme est un commentaire.

b/Zone opérateur

Elle contient un code opération mnémonique de 3 CARACTERES désignant une instruction-machine de l'assembleur.Pour certaines instructions machines,le code opération est muni d'un caractère spécial permettant d'indiquer le mode d'adressage.

c/Zone opérande

Elle définit les opérandes des instructions,suivant le type de ces dernières,elle peut contenir un ou plusieurs opérandes séparés par une virgule.

d/Zone commentaire

Tous les caractères disponibles peuvent être utilisés y compris les espaces.

2/FONCTION D'ASSEMBLEUR

L'assembleur est un programme conversationnel dont le lancement et la définition des unités logiques sources et but sont faits à partir

du clavier de la télétype.

Les unités logiques reconnaissables par l'assembleur sont:

SCE :définit le support du programme source

BUT :définit le support sur lequel le programme source est transféré lors de la première passe.

OBJ :définit le support de sortie du code objet au cours de la deux-ième passe.

LIS :définit l'unité logique supportant le listing et les messages d'erreur.

Les diverses unités physiques que l'on peut affecter aux unités logi-ques sont:

LC :lecteur de cartes

LR :lecteur de ruban

DX :disque souple

BM :bande magnétique

TF :le téléimprimeur

LG : imprimante logabax

DP :imprimante dataproduct

Le fonctionnement global de l'assembleur,avec les périphériques,est résumé par le schéma de la page suivante.

```
PAT 05 - PASS 1
OPTIONS                                  ← implique la définition d'un nom de fichier
SCE = LR, BUT = DX, OBJ = DX, LIS = LC (RC)
(RC)
FICHIER : EXPROG (RC)        début d'assemblage
```

Figure A - Définition du fichier binaire permettant de répertorier le code objet éditable.

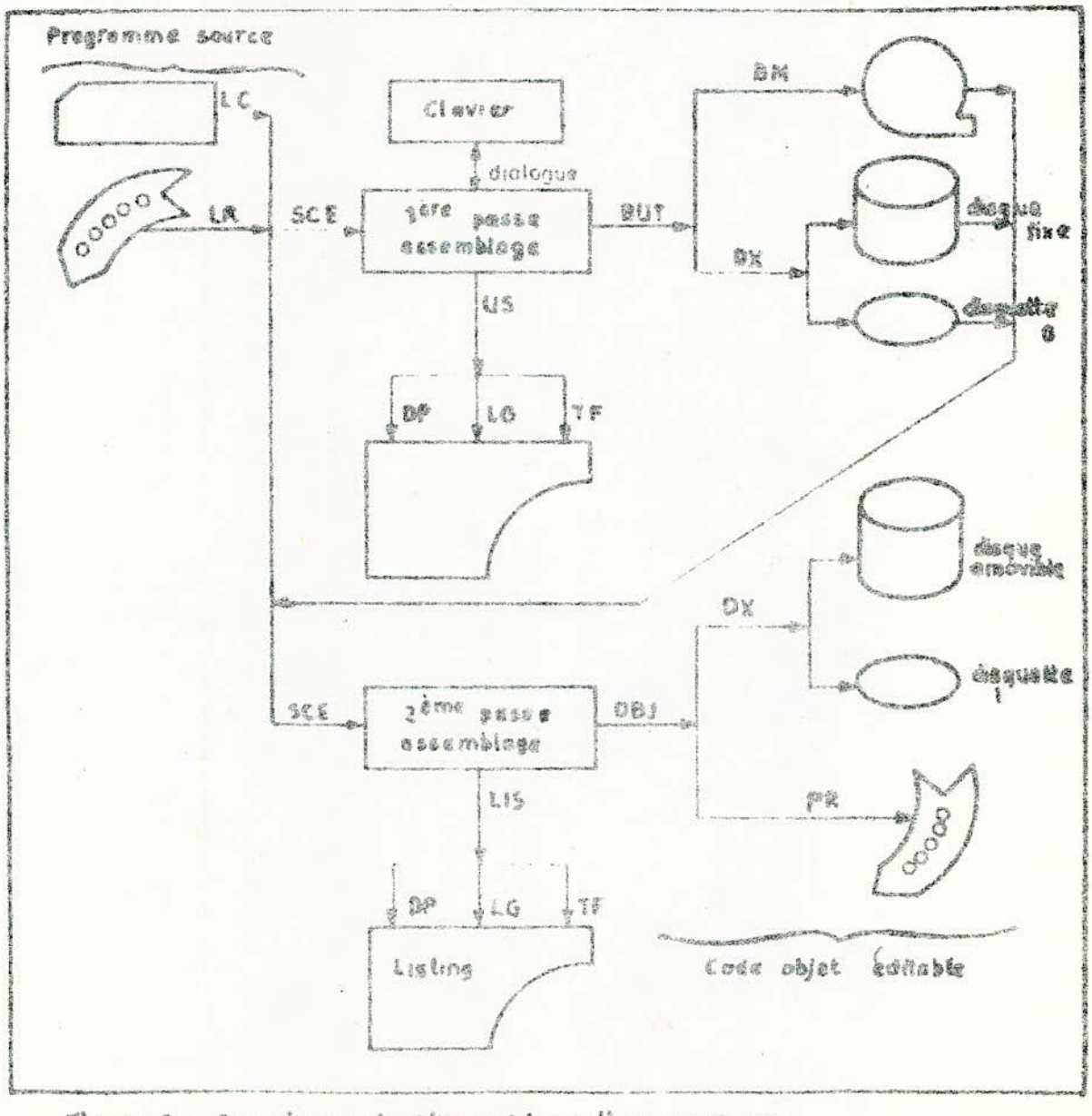


Figure B : Synoptique de l'assemblage d'un programme

Chargement de l'assembleur:

Le programme assembleur utilisé est livré sur ruban qui se compose de cinq blocs:

1/ assembleur local version 05 avec lecture et perforation par la TTY

2/additif pour version 01

3/ sous-programme de lecture de ruban par lecteur rapide

4/sous programme de perforation de ruban par le perforateur rapide

5/sous-programme de lecture de cartes

Comme on le voit, il est ainsi possible d'utiliser pour la lecture du programme-source soit le lecteur de ruban, soit le lecteur de cartes; l'introduction du sous-programme de lecture correspondant à l'un de ces deux organes supprime automatiquement la lecture du ruban par la télétype.

Egalement, la perforation peut se faire, soit par la TTY, soit par le perforateur rapide.

L'introduction du sous-programme de perforation du perfo. rapide sup--prime celui de la télétype.

Assemblage

Au début de chaque passe, un message est imprimé indiquant l'assembleur utilisé et le numéro de la passe. On place le programme source sur l'organe d'entrée choisi et frappe un retour chariot(RC) pour démar--rer la passe.

A la fin de l'assemblage, le programme assembleur est automatiquement relancé à la passe 2. Néanmoins, on peut toujours revenir à la passe 1 et ce par un jeu de clés de test du pupitre de commende du Multi-20; on peut également, si l'assemblage est arrêté, au cours de la passe 2, le relancer .

Remarque: l'assembleur qu'on a utilisé est la version translatable de PAL , comme il n'utilise ni disque ni bande magnétique, il faut, donc lire le programme source lors de la première puis de la seconde passe.

Ceci nous élimine la possibilité de définir une unité physique but qui aurait pu accélérer l'assemblage d'un programme.

INSTRUCTIONS DE L'ASSEMBLEUR

L'instruction machine est représenté en langage assembleur par un code mnémonique de 3 caractères ,placé dans la zone opérateur. Les instructions avec référence mémoire ont huit modes d'adressage, l'adressage est représenté par la valeur de l'expression de calcul d'adresse située dans la zone opérande et par un symbole spécial placé derrière le code opération.

* Symboles affectés aux différents modes d'adressages:

1/Mode direct page zéro:aucun caractère n'est placé dérrière le code opération;(la valeur de l'adresse dans la zone opérande est inférieure à 256.

2/Mode direct relatif :ancun caractère n'est placé derrière le code opération(la valeur de l'adresse doit être supérieure à 256.

3/Mode indirect page 0:le code opération est suivi d'un astérisque(*) (la valeur de l'adresse est inférieur à 256)

4/Mode indirect relatif:le code opération est suivi d'un astérisque (*),la valeur de l'adresse est supérieur à 256

5/Mode indéxè :le code opération est suivi du caractère -

6/Indéxé avec déplacement:le code opération est suivi du caractère +

7/Mode étendu :le code opération est suivi d'une bande oblique /

8/Mode immédiat :le code opération est suivi du caractère égal (=).

REPRESENTATION DES CONSTANTES

Il existe deux types de constantes:décimale et hexadécimale

a/Constante décimale:

C'est un nombre non signé dont la valeur maximale est 65535 .

b/Constantes hexadécimales

C'est un nombre hexadécimal non signé constitué de 1 à 4 chiffres hexadécimaux. Les chiffres sont écrits entre apostrophes et précédés de la lettre X. Exemple X'2C5F'

PSEUDO-INSTRUCTIONS OU DIRECTIVES D'ASSEMBLAGE

ORG :qui signifie origine

La peudo-instruction ORG modifie le contenu du compteur d'emplacement

EQU :qui signifie équivalent

Elle est utilisée pour définir un symbole en lui affectant la valeur de l'expression figurant dans la zone opérande

DC : définition de constante

DS :réservation mémoire

END:indique la fin de programme

Elle permet de terminer l'assemblage d'un programme et doit être la dernière instruction d'un programme-source.

EXEMPLE D'UTILISATION DE L'ASSEMBLEUR

1/Eciture d'un texte sous éditeur de texte:

Nous avons choisi,pour cet exemple,un programme "écho" qui rend la télétype en attente de caractères;il permet de faire entrer dans le calculateur un (ou plusieurs,suivant le compteur) caractère alpha-numérique ,puis de le faire imprimer sur la TTY.

T1,26 ouverture d'un fichiers de 26 lignes

Comme cette instruction est nècessaire avant l'écriture du texte lui-même,on peut toujours prévoir un nombre de lignes reservées supérieur à celui du programme.

La première ligne est toujours réservée à la pseudo-instruction ORG,qui indique "début de programme".

Voir texte page suivante.

```
001         ORG X'5000'          adresse début du programme
002     PRI LDB= X'FFF8'         nombre de caractères= 8
003         LDX= X'2000'         adresse de rangements des caractères
004     DEB TRJ/ ENT             entréé de caractère
005         RTJ/ OTA             sortie de caractère
006         INB                  incrémenter le compteur
007         INX                  incrémenter adresse de rangement
008         NBZ DEB              dernier caractère?
009         TRP                  si oui fin
010     ENT DC *  *
011         IBA 1,0              demande de mot d'état du coupleur TTY
012         ANV= X'02'           test si coupleur prêt
013         JAZ  ENT + 2         si non prêt attente
014         IBA  0,0              entréé du caractère
015         STV_
016.        JMP_* ENT            retour au prog.principal
017     OTA DC *  *
018         LDV_
019         STV= H'0'            sauvegarde du caractère
020         IBA  1,0             demande de mot d'état
021         ANV= X'04'           test  si TTY prêt à recevoir commande
022         JAZ  OTA + 5         si non prêt attente
023         LDV_* -7             reprise du caractère
024         OBA  0,0             sortie de caractères
025         JMP_* OTA
026         END  X'5000'         adresse d'éxécution
```

2/ACTIVATION DE L'ASSEMBLEUR

* On place le ruban généré par l'éditeur de texte sur le lecteur de la TTY.

* Retour Chariot

L'assembleur répond par :

```
PAGE  002

PRI   5000   DEB   5006   ENT   5011   OTA   501E

NO ERRORS    PASS1
```

La passe 1 est ainsi achevée, on lance la passe 2 par un jeu de clés du panneau de commande du Multi-20, la TTY imprime:

```
PAGE  001

5000                1        ORG   X'5000'
5000   97FFF8       2   PRI  LDB=   X'FFF8'
5003   872000       3        LDX=  X'2000'
5006   6E5011       4   DEB  RTJ/  ENT
5009   6E501E       5        RTJ/  OTA
500C   49           6        INB
500D   44           7        INX
500E   1AF6         8        NBZ   DEB
5010   01           9        TRP
5011   5011        10   ENT  DC    *
5013   3120        11        IBA   1,0
5015   DF02        12        ANV=  X'02'
5017   11FA        13        JAZ   ENT+2
5019   3100        14        IBA   0,0
501B   FC          15        STV-
501C   63F3        16        JMP   *
501E   501E        17   OTA  DC    *
5020   EC          18        LDV-
```

```
 5021    FF00        19          STV=  H'0'
 5023    3120        20          IBA   1,0
 5025    DF04        21          ANV=  X'04'
 5027    11FA        22          JAZ   OTA+5
 5029    E9F7        23          LDV   *  -7
 502B    3900        24          OBA   0,0
 502D    63EF        25          JMP*  OTA
 502F    5000        26          END   X'5000'
NO ERROR(S)  PASS 2
```

Le ruban perforé défile sur le perforateur de la TTY et imprime la ligne correspondante et la perfore sur le perforateur de la télétype en même temps en langage machine.

On obtient ainsi à la fin le programme "écho" en langage machine sur le perforateur qu'on pourra éxécuter à l'aide de la commande R du MET.

MISE EN OEUVRE DU SYSTEME

Nous avons sauvegarder sur bande tous ces programmes utilitaires à l'aide de la commande TB du MET;ainsi à chaque fois que l'opérateur aura à utiliser un de ces programmes,il pourra en faire appel simplement à l'aide de la commande UB du MET.

Adresses d'implantation de chaque programme

programme	adresse début	adresse fin	adresse de lancement
CARBAM	01AE	028F	01AE
MET	7200	7FFF	7200
EDL	500	100A	502
PER	4000	4A00	4000
ASSE	0000	1200	0000

OPERATIONS A SUIVRE POUR L'ENRGISTREMENT SUR BANDE MAGNETIQUE

TB CARBAM,01AE,028F

TF 01AE

TB MET,7200,7FFF

TF 74FF

TB EDL,500,100A

TF 502

TB PER,4000,4A00

TF 4000

TB ASS,0000,1200

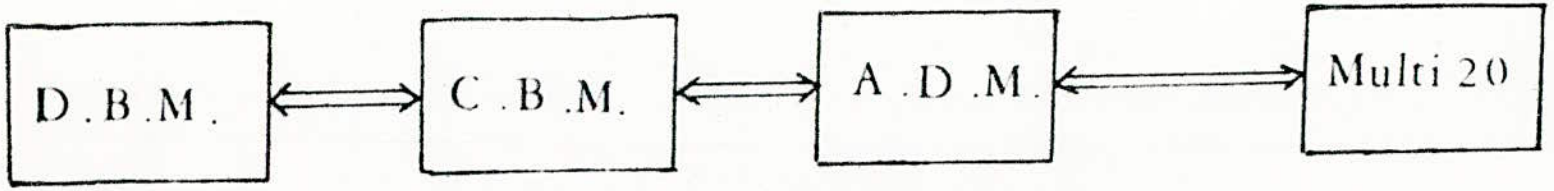
TF 0000

LANCEMENT DU SYSTEME

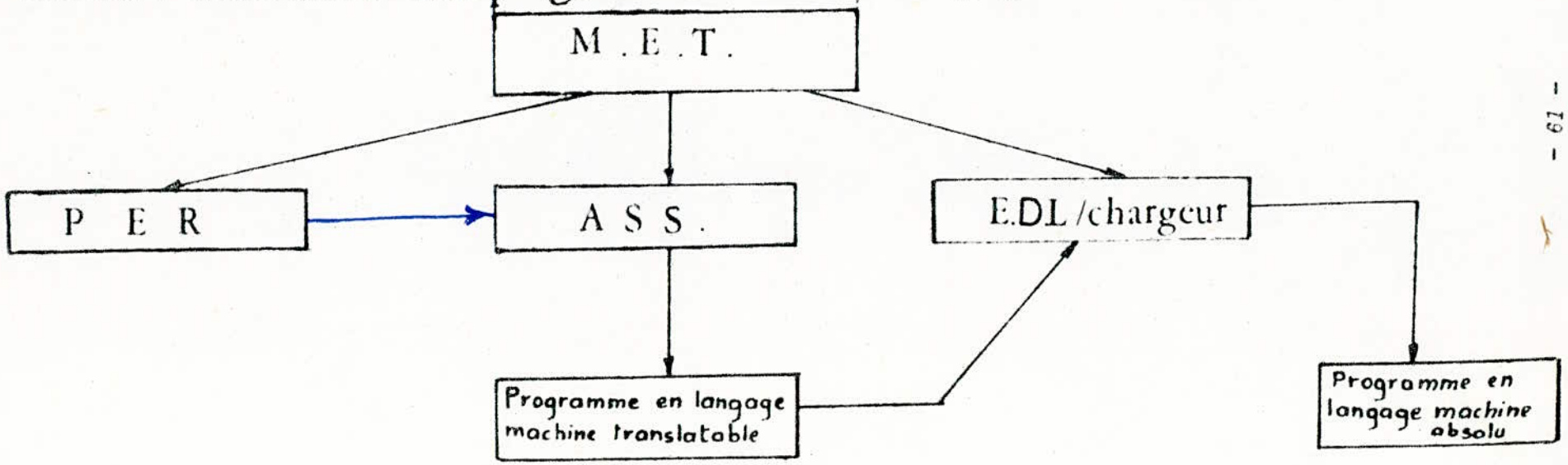
L'opérateur doit commencer par lancer le programme "CARBAM" conçu pour charger le premier enregistrement sur bande magnétique qui n'est autre que le MET.

Voir synoptique page suivante

1° PHASE: Chargement du M E T



2° PHASE: Exécution d'un programme écrit en langage assembleur



SYNOPTIQUE DE MISE EN ŒUVRE DU SYSTEME

C O N C L U S I O N

&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&

La structure technologique "hardware" des ordinateurs semble actuellement avoir atteint une stabilité relative cédant le pas à une évolution dans laquelle les efforts se portent surtout vers la program--mation" "software".

Le développement de l'informatique se base donc, surtout sur la manière de concevoir un "software" de base de la machine et de son exploi--tation optimale.

L'utilisation d'un calculateur met donc en oeuvre une véritable biblio--thèque de programmes et de sous-programmes qui facilitent son exploi--tation et qui sont utilisés au moment du passage en machine d'un programme d'utilisateur.

Le nombre considérable de solutions actuellement existantes et appliquées donne une idée de l'ampleur du problème et de l'imperfec--tion de chacune des solutions. D'un système à un autre, ce qui varie c'est l'automaticité, la souplesse des méthodes et l'encombrement total des programmes; tout système réalise un compromis entre ces trois critères i inconciliables qu'il essayera d'optimiser.

Pour mener à bien notre étude, il fallait d'abord passer par:

1/ la maitrise du fonctionnement du calculateur et de ses divers péri--phériques: TTY, dérouleur de bande, lecteur...

2/ la maitrise du fonctionnement et de la programmation des coupleurs A.D.M. et dérouleur de bande magnétique.

3/ la mise au point des programmes nècessaires à l'enregistrement d'un programme sur bande magnétique en mémoire centrale. Il faut signaler

qu'elle s'est faîte dans le soucis:

-d'avoir des programmesles plus généraux possibles (utilisation de sous-programmes)

-de permettre une éventuelle translation vers d'autres zones mémoires, (adoption d'un adressage relatif)

4/distinguer les fonctions respectives des différents programmes, moniteurs,assembleurs,chargeur,dont chacun a un rôle particulier à jouer.

Par ailleurs,on a noté l'existence de quelques perturbations d'ordre matériel (commandes de la bande magnétique sous MET)

Pour cette quatrième génération d'ordinateurs on peut dire que le software joue le rôle le plus important dans l'amélioration et le développement des qualités et performances d'un ordinateur.

Cette étude nous aura permis d'une part,de mettre à jour nos connaissances et d'autre part de nous familiariser aux machines informa-tiques aussi bien au niveau de la conception qu'à celui de l'utilisation.

\$\$\$

annexe

***

- I N D E X -

Dans le but de faciliter la consultation de cet ouvrage, nous donnons, ci-aprés, la liste des abrêviations utilisées dans ce mémoire.

CARBAM :programme principal de chargement duMET sur Bande Magnétique.

LEBO :sous-programme de lecture sur bande magnétique et rangement mots de commande A.D.M.

LEBTEK: sous-programme de lecture sur bande magnétique (4 essais en cas d'erreur)

OTA :sous-programme sortie de caractères

EBM :sous-programme testant l'état de la bande magnétique

DBM :sous-programme testant le début de la bande magnétique

PER :programme utilitaire à la génération d'un texte en langage assembleur(éditeur de textes)

EDL :programme utilitaire de l'éditeur de liens

ASS :programme assembleur

ADM :canal accés direct en mémoire

BM :bande magnétique

MC :mémoire centrale

MET :moniteur d'exploitation télétype

LR :lecteur rapide

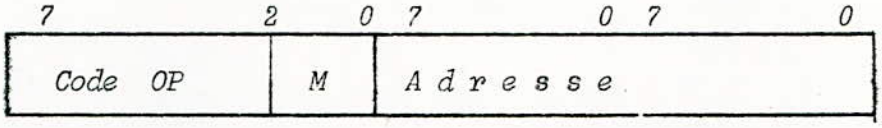
PR :perforateur rapide

TTY :téléimprimeur (télétype)

CBM coupleur bande magnétique

I N S T R U C T I O N S A R E F E R E N C E

M E M O I R E

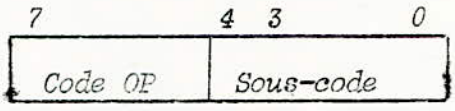


Ces 20 instructions disposent de 8 modes d'adressages et opèrent sur des formats de 8 , 16 , 24 ou 32 bits.

Les 8 modes d'adressage leur sont applicables.

Code	Mnémonique	F o n c t i o n r é a l i s é e	temps d'éxéc. en us
60	JMP	Branchement inconditionnel	3,2
68	RTJ	Appel de sous-programme	5,8
70	IWM	Incrémentation mot en mémoire	5,4
78	DWM	Décrémentation mot en mémoire	5,4
80	LDX	Chargement de X	5,4
88	STX	Rangement de X	5,4
90	LDB	Chargement de B	5,4
98	STB	Rangement de B	5,8
A0	ADA	Addition mot(16 Bits)dans A	4,8
A8	ADV	Addition format variable(8,16,24,32)	5,8 - 8,6
B0	SBA	Soustraction mot (16 bits)	5,2
B8	SBV	Scustraction format variable(8,16 ,24,32)	6,2 - 9
C0	CPA	Comparaison mot (16 bits)avec A(=)	4,8
C8	CPV	Comparaison format variable (=)	4,8 - 8,2
D0	ANA	Intersection mot (16 bits) sur A	5,2
D8	ANV	Intersection format variable(8,16,24,32)	6,2 - 9
E0	LDA	Chargement mot (16 bits)dans A	5,2
E8	LDV	Chargement de format variable en A,B	6,2 - 9
F0	STA	Rangement de A	4,2
F8	STV	Rangement de format variable(8,16,24,32)	3,4 - 9,2

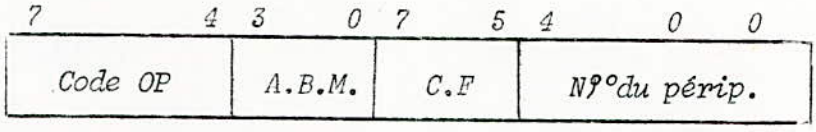
I N S T R U C T I O N S S U R R E G I S T R E S



Ce groupe d'instructions assure des transferts ou traitements arithmé-
-tiques et logiques entre contenus de registres.

Code	Mnémonique	F o n c t i o n r é a l i s é e	temps d'éxécut. en us
40	ORA	Réunion entre A et B	5,8
41	XRA	Disjonction entre A et B	5,8
42	ORB	Réunion entre B et A	6
43	XRB	Disjonction entre B et A	6
44	INX	Incrémentation de X	6,4
45	DCX	Décrémentation de X	6,4
46	AWX	Incrémentation de X suivant format	6,4
47	SWX	Décrementation de X suivant format	6,4
48	INA	Incrémentation de A	6,4
49	INB	Incrémentation de B	6,4
4A	OCA	Complément à 1 sur A	6
4B	OCB	Complément à 1 sur B	6
4C	TAX	Transfert de A sur X	6,4
4D	TBX	Transfert de B sur X	6,4
4E	TXA	Transfert de X sur A	6,6
4F	TXB	Transfert de X sur B	6,6
59	ADX	Addition de constante à X	8
5B	EBX	Echange entre B et X	6,4

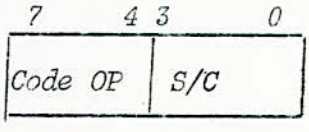
INSTRUCTIONS D'ENTREE-SORTIE



Ces instructions assurent les échanges avec l'extérieur par le bus d'entrée-sortie, par transferts d'octets.

Code	Mnémonique	Fonction réalisée	temps d'éx.
31	IBA	Entréee d'octet en A	7,6
32	IBB	Entrée d'octet en B	8
33	IBM	Entrée directe en mémoire(8b)	13
39	OBA	Sortie d'octet de A	7,6
3A	OBB	Sortie d'octet de B	8,4
3B	OBM	Sortie directe de la mémoire(8b)	13,2

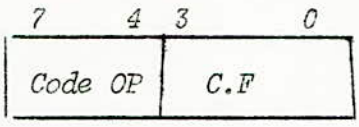
INSTRUCTIONS ARITHMETIQUES



Ce groupe d'instructions réalise l'addition et la soustraction sur des nombres décimaux signés d'une longueur variable pouvant atteindre 16 chiffres. Les registres B et X contiennent les adresses des deux opérandes. La multiplication et la division sont réalisés sur des nombres binaires de 16 bits.

Code	Mnémonique	Fonction réalisée	temps d'éxéc.
3C	DAD	Addition décimale	
3D	DSB	Soustraction décimale	
3E	MUL	Multiplication	67
3F	DIV	Division	92

INSTRUCTIONS TRAVAILLANT SUR PILES



Ce groupe d'instructions permet/

-de faire appel à un sous-programme avec rangement du contexte du programme appelant dans la pile.

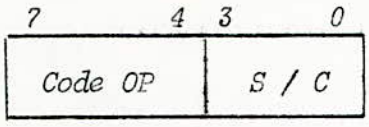
-de faire un retour de sous-programme avec restauration du contexte du programme appelant à partir de la pile.

-de ranger ou de restaurer individuellement les registres A,B et X.

Ces instructions facilitent le traitement des interruptions hiérar-chisées et les techniques de réentrance.

Code	Mnémonique	Fonction réalisée	TEMPS D'EXEC.
50	RTN	Retour de sous-programme	29,6
51	CAL	Appel de sous-programme (rangement en pile)	31,4
52	PLX	Chargement de X	13,8
53	PSW	Rangement de X	12,8
54	PLA	Chargement de A	13,8
55	PSA	Rangement de A	12,8
56	PLB	Chargement de B	13,8
57	PSB	Rangement de B	12,8

INSTRUCTIONS DE MANIPULATIONS DE CHAINES DE CARACTERES

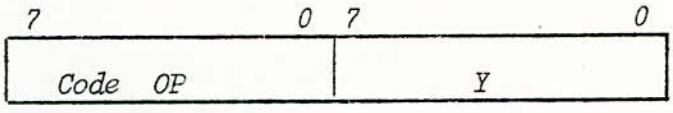


Ces 6 instructions, véritables macro-instructions, effectuent un traitement logique sur chacun des caractères d'une chaîne avec branchement suivant résultat .

Code	Mnémonique	Fonction réalisée	temps d'éxécution en us.
35	CLC	Comparaison de 2 chaînes de caractères	9,6par octet
5C	MOV	Transfert de zone	9,2par octet
5D	GCC	Génération de code cyclique	29 par octet
5E	SCH	Recherche en table	7,2par octet
5F	GAP	Génération de parité ASCII	15,2par octet

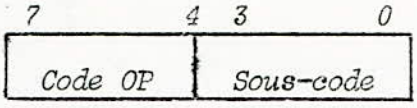
INSTRUCTIONS DE BRANCHEMENT CONDITIONNEL

Ces instructions assurent les tests et les articulations entre séquences de programmes ;elles effectuent un adressage direct relatif



Code	Mnemonique	Fonction réalisée	temps d'éxécution en us
10	JOV	Saut si OV est égal à 1	6,2- 7,8
11	JAZ	Saut si A est nul	6,8 - 7,6
12	JBZ	Saut si B est nul	6;6 -7,4
13	JXZ	Saut si X est nUl	6,4 - 7,2
14	JAN	Saut si A est négatif	6,8 - 7,6
15	JXN	Saut si X est négatif	6,6 - 7,4
16	JAB	Saut si A égale B	7 - 7,8
17	JAX	Saut si A égale X	6,8 - 7,6
18	NOV	Saut si OV égale zéro	7 - 7
19	NAZ	Saut si A différent de zéro	6,8 - 7;6
1A	NBZ	Saut si B différent de zéro	6,6 - 7,4
1B	NXZ	Saut si X différent de zéro	6,4 - 7,2
1C	NAN	SAut si A est positif ou nul	6,8 - 7,6
1D	NXN	Saut si X est positif ou nul	6,6 - 7,4
1E	NAB	Saut si A différent de B	7 - 7,8
1F	NAX	SAut si A différent de X	6,8 - 7,6
5A	JEP	SAut si parité paire sur A	7,8 - 29,6

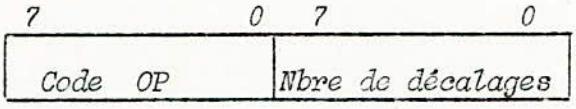
I N S T R U C T I O N S D E C O N T R O L E



Ces instructions ne nècessitant qu'un octet

CODE	Mnémonique	F o n c t i o n r é a l i s é e
00	HLT	Halte
01	TRP	Déroutement
02	ESW	Entrées des 4 clés de test
04	DIN	Désarmement des interruptions externes
05	EIN	Armement des interruptions externes
06	DRT	Arrêt de l'horloge temps réèl
07	ERT	Départ de l'horloge temps réèl
08	RO1	Mise à zéro du registre OV et choix du format 1
09	RO2	Mise à zéro du registre OV et choix du format 2
0A	RO3	Mise à zéro du registre OV et choix du format 3
0B	RO4	Mise à zéro du registre OV et choix du format 4
0C	SO1	Mise à un du registre OV et choix du format 1
0D	SO2	Mise à un du registre OV et choix du format 2
0E	SO3	Mise à un du registre OV et choix du format 3
0F	SO4	Mise à un du registre OV et choix du format 4
34	NOP	Non opération

INSTRUCTIONS DE DECALAGE



Les instructions de décalage opèrent sur le contenu de A, de B ou des deux registres

Code	Mnémonique	Fonction réalisée	temps d'éxécution en us
20	LLA	Logique circulaire à gauche sue A	5,8 + 3,2 n
21	LLB	Logique circulaire à gauche sur B	5,8 + 3,2 n
22	LLL	Logique circulaire à gauche sur A,B	5,8 + 3,4 n
24	LRA	Logique à droite sur A	5,8 + 3n
25	LRB	Logique à droite sur B	5,8 + 3,n
26	LRL	Logique à droite sur A,B	5,8 + 3,6n
28	ALA	Arithmétique à gauche sur A	5,8 +3,2 n
29	ALB	Arithmétique à gauche sur B	5,8 + 3,2n
2A	ALL	Arithmétique à gauche sur A,B	5,8 + 3,4 n
2C	ARA	Arithmétique à droite sur A	5,8 + 3 n
2D	ARB	Arithmétique à droite sur B	5,8 + 3 n
2E	ARL	Arithmétique à droite sur A,B	5,8 + 3,6 n

COMMANDES DU PER

(RC) = Retour Chariot

A m,n	Affectation du numéro n à la ligne m
D m, (RC)	Suppression de la ligne m
D m, n	Suppression des lignes comprises entre m et n inclusivement
E m	Edition/Correction de la ligne m.
I m	Insertion(ou remplacement)de la ligne m
L	Perforation d'une amorce ou fin de ruban
N m,n,s, (RC)	Renumérotation des lignes m à n à parir de la valeur s,par pas de 1
N m,n,s,k	Renumérotation des lignes m à n à parir de la valeur s,le pas d'incrémentation est défini par k.
P m (RC)	Impression de la ligne m
P m,n	Impression des lignes comprises entre m et n par valeur croissante de numéro.
R m (RC)	Lecture d'une ligne avec affectation du numéro m
C m (RC)	Lecture d'une carte avec affectation du numéro m
R m,n (RC)	Lecture de n-(m+1)lignes avec affectation de numéros de valeur initiale m et progressant avec un pas de1
C m,n (RC)	Lecture de n-(m+1)lignes avec affectation de numéros de valeur initiale m et progressant avec avec un pas de 1
M (RC)	Retour au moniteur.

COMMANDES E.D.L.

La mise en exploitation est faite par frappe au clavier de la télé-type de commandes conversationnelles constituées de 2 lettres

RA -Chargement d'un ruban absolu.

RL - Chargement d'un ruban absolu ou translatable à partir de la première placelibre ou à l'adresse frappée à la suite de la commande RL.(LR est une variante).

RB -Création d'un ruban absolu. (BR est une variante).

RT -Reprise d'un bloc aprés détection d'erreur au chargement.

RS -Ré-initialisation du système.

EX -Mise en éxécution du programme chargé ou perforation d'un bloc END.

SM -Impression de la table des symboles.

CM -Entée de la base des COMMON

SK -Saut d'un bloc sur le ruban(SKIP)

SI -Listage des symboles inconnus ou multi-définis.

B I B L I O G R A P H I E

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

INTRODUCTION A L'INFORMATIQUE STRUCTURE ET PROGRAMMATIONS DES ORDINATEURS

J.Dondoux ,P.MARANO et J.C.MERLIN

CONCEPTION DE LA PROGRAMMATION DES ORDINATEURS

J. DU ROSCOAT

INTERTECHNIQUE:

-MANUEL DE PRESENTATION DU MULTI-20

-MANUEL D'EXPLOITATION PAR TELEIMPRIMEUR

-EDITEUR DE LIENS/CHARGEUR ET ASSEMBLEUR

PROJET DE FIN D'ETUDES:

-VISUALISATION ET TRAITEMENT D'IMAGES NUMERISEES SUR UNE BANDE MAGNETIQUE

(JANVIER 1981)

§§§§§§§§§§§§§§
§§§§§§§§§§
§§§§§§
§