

- Note CEA-N-1123 -

Centre d'Etudes Nucléaires de Saclay  
Département d'Electronique Générale  
Service d'Instrumentation Nucléaire

**SYSTEME D'ACQUISITION DE DONNEES  
UTILISANT UN ORDINATEUR C 90-10**

par

Gabro SMILJANIC

- Mai 1969 -

## SYSTEME D'ACQUISITION DE DONNEES UTILISANT UN ORDINATEUR C 90-10

### I - INTRODUCTION

Le but de cette étude est de rendre possible l'acquisition de données expérimentales par la mémoire d'un ordinateur numérique. Ces données proviennent de convertisseurs analogiques-numériques qui analysent l'amplitude des impulsions fournies par des détecteurs.

En temps normal le ordinateur exécute son programme principal (traitement des données, transfert sur bande magnétique, visualisation, etc...). Lorsqu'une information est disponible en sortie d'un convertisseur, une interruption du programme principal est demandée, et après accord un sous-programme prend en charge l'accès de l'information dans le ordinateur. Nous avons également envisagé l'acquisition bi- et triparamétrique.

Le ordinateur et les convertisseurs sont des dispositifs commerciaux (ordinateur C 90-10 de C. I. I. et convertisseur CA 12 ou CA 25 d'Intertechnique), tandis que les systèmes d'adaptation de niveaux entrée-sortie et la visualisation ont été étudiés et réalisés au C.E.N. SACLAY.

Nous avons construit un dispositif d'interface pour connecter les convertisseurs ; c'est la partie câblée du système. D'autre part nous avons étudié et mis au point les programmes nécessaires au fonctionnement du ordinateur ; c'est l'aspect programmé du système.

Dans la mesure du possible l'interface est conçue de façon à être universelle c'est-à-dire qu'elle doit pouvoir fonctionner avec des appareillages d'autres marques.

L'acquisition des données s'exécute en deux phases :

a) le convertisseur exprime l'amplitude du signal d'entrée sous la forme d'un nombre binaire qui est transféré dans l'interface en même temps qu'une interruption du programme principal est demandée.

b) après acceptation de cette interruption, le sous-programme prend en charge le transfert de l'information de l'interface dans le ordinateur, puis ajoute une unité au mot situé à l'adresse déterminée à partir de l'information reçue.

Autrement dit le système se comporte comme un analyseur d'amplitudes dont le fonctionnement est bien connu. Mais il est d'utilisation beaucoup plus souple à cause des possibilités d'adaptation rapide des programmes aux besoins de l'expérience considérée, de la possibilité de traiter immédiatement les données ou de les transférer éventuellement vers d'autres systèmes, etc...

Si l'on utilise deux convertisseurs, deux interfaces et un sous-programme approprié on obtient un système biparamétrique. Nous avons également étudié jusqu'à un certain point un système triparamétrique.

Pour étendre et généraliser les possibilités du système, on peut envisager de substituer aux impulsions d'entrée, des échantillons de tension continue correspondant à des paramètres lentement variables.

## II - CONCEPTION DU SYSTEME

La manière la plus simple et la plus rapide pour transférer une information dans la mémoire du calculateur C 90-10 est d'utiliser le canal d'accès direct [1][2], commandé par l'instruction d'entrée parallèle PIN, toujours précédée d'une instruction EOM, dont le but est de sélectionner et de mettre en service l'appareil périphérique concerné qui est, dans le cas présent, "l'Interface" du convertisseur.

Le processus de transfert est déclenché automatiquement à la suite d'une fin de conversion, lorsqu'une interruption est demandée au calculateur et qu'elle est acceptée. La figure 2.1 représente le schéma de principe de "l'Interface" pour transfert d'information, dans le cas le plus simple c'est-à-dire en format monoparamétrique.

L'information analogique (amplitude d'une impulsion issue d'un détecteur ou échantillon d'une tension continue) accède au convertisseur par l'intermédiaire d'une porte. La valeur de cette amplitude est transformée en un nombre binaire de douze bits maximum. A la fin de la conversion ce nombre est transférée immédiatement dans un registre d'interface.

Dès qu'une impulsion accède au convertisseur, la porte d'entrée est fermée et le temps mort commence. Une impulsion de la durée de ce temps mort est disponible sur une sortie [4][5]. Le front avant de cette impulsion sert pour la remise à zéro du registre "d'Interface" dans lequel l'information du convertisseur sera transférée aussitôt après la fin de conversion (fig. 2.2). Au cours de ce transfert on dispose sur une sortie de convertisseur d'une impulsion "Pilote". Celle-ci est utilisée pour positionner à 1 le bistable  $B_b$  dont le but est d'appeler l'interruption du programme principal du calculateur et par suite de déclencher le sous-programme de transfert.

Cette impulsion "Pilote" sert également à positionner le bistable  $B_d$  dans le but de bloquer l'entrée du convertisseur, tant que le calculateur n'a pas fourni l'impulsion  $R_{ti}$ , indiquant qu'il a enregistré l'information. Ceci permet de tenir compte, dans certains cas, du fait que le calculateur peut mettre un temps plus ou moins long avant de répondre à la demande d'interruption, et d'éviter ainsi que le convertisseur transfère une nouvelle information dans le registre

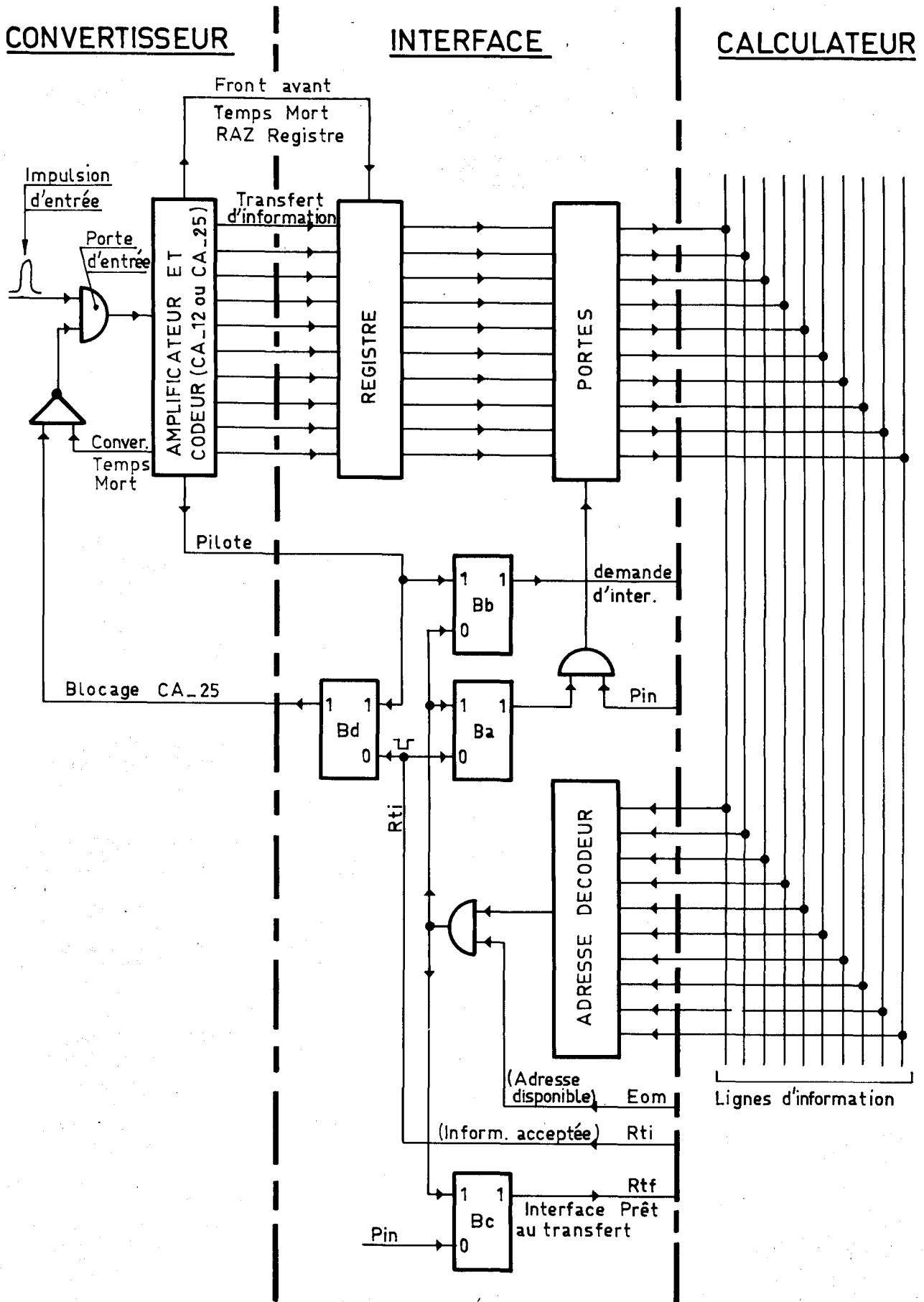


Fig. 2-1

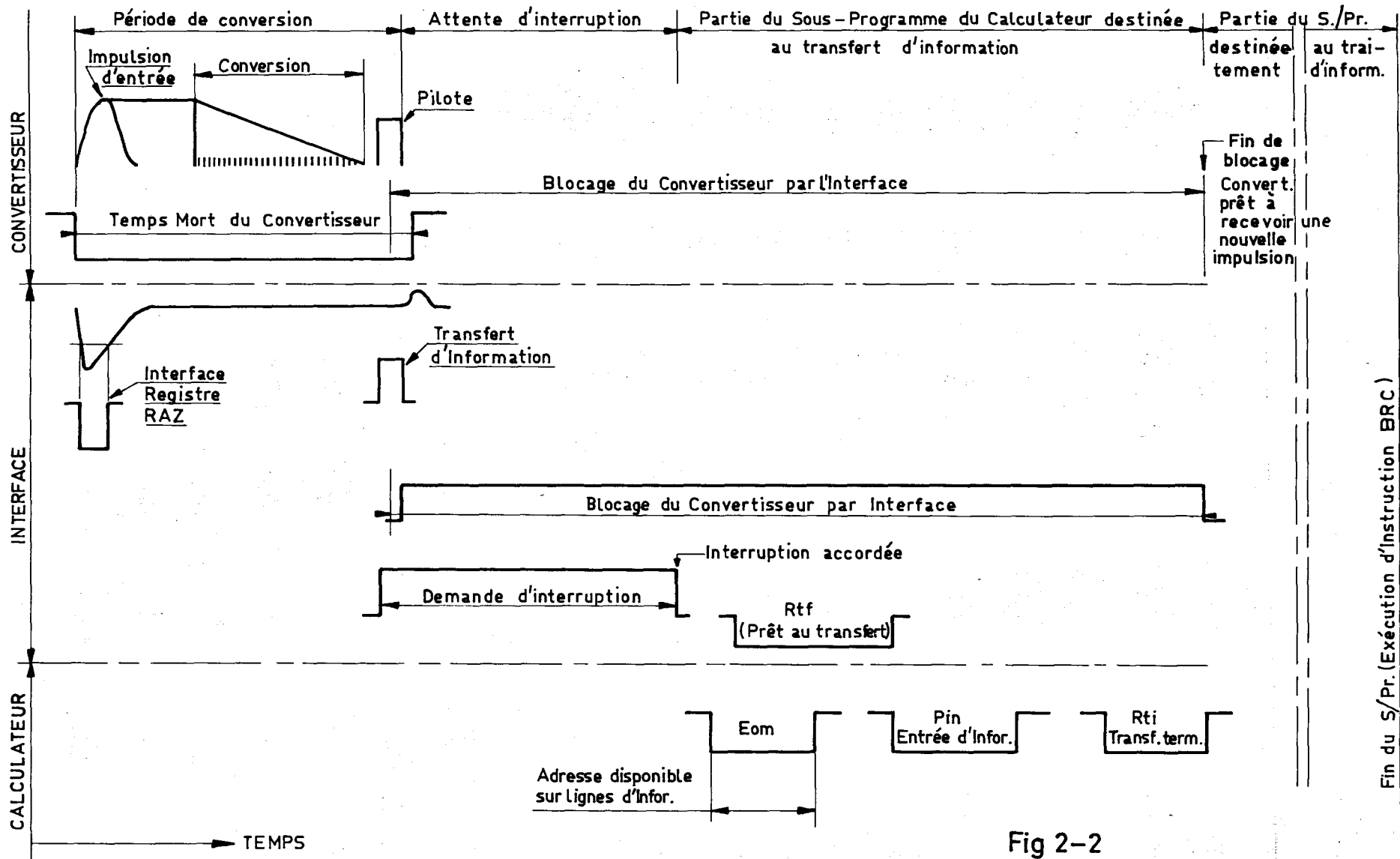


Fig 2-2

Fin du S/Pr. (Exécution d'Instruction BRC)

"d'Interface" tant que la précédente n'a pas été utilisée (sinon on risquerait de favoriser les impulsions d'entrée de faibles amplitudes ou de provoquer de faux enregistrements).

L'état des bistables  $B_a$ ,  $B_b$ ,  $B_c$  et  $B_d$  est représenté sur la figure 2.3. Quand le calculateur accepte la demande d'interruption, le sous-programme de stockage d'information commence. C'est d'abord l'instruction EOM qui est effectuée. Le calculateur présente l'adresse choisie de l'interface sur les lignes d'information et en même temps émet une impulsion Eom (fig. 2.2) dans le but d'informer l'interface que le sous-programme demandé est commencé. La demande d'interruption n'étant plus utile, le bistable  $B_b$  est remis à zéro (fig. 2.1 et 2.3). L'impulsion Eom positionne à un les bistables  $B_a$  et  $B_c$ .

Le bistable  $B_c$  délivre le signal Rtf avertissant le calculateur que l'interface est prête au transfert, le bistable  $B_a$  ayant ouvert la porte de sortie des informations. Ensuite le calculateur exécute l'instruction PIN. Pendant cette instruction il accepte l'information de l'interface par les lignes d'information échantillonnées par l'impulsion Pin. Celle-ci remet à zéro le bistable  $B_c$ , le signal Rtf n'étant plus nécessaire.

Ayant accepté l'information, le calculateur engendre l'impulsion Rti qui informe l'interface que l'information a été reçue. L'interface ouvre à nouveau la porte d'accès au convertisseur et une nouvelle impulsion peut être analysée. L'impulsion Rti ferme aussi les dix portes de transfert d'information d'interface au calculateur. Il faut déconnecter l'interface du calculateur pendant le temps où la connexion n'est pas nécessaire c'est-à-dire lorsque les autres opérations se déroulent.

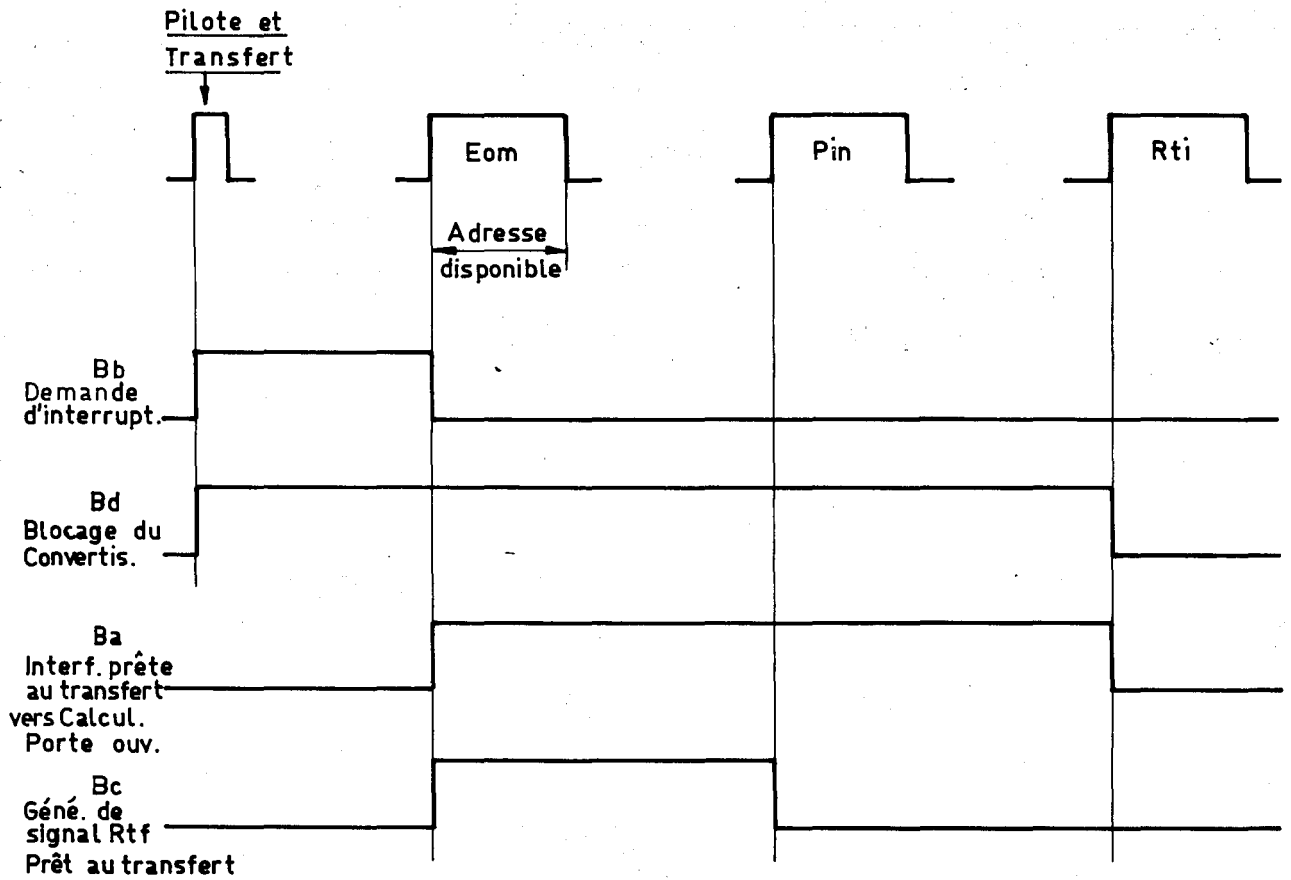
Après réception de l'information, la deuxième partie du sous-programme "traitement d'information reçue" se déroule dans le calculateur et simultanément le convertisseur peut accepter une nouvelle impulsion. Si la conversion de cette deuxième impulsion se termine plus tard que la partie du sous-programme destiné au traitement d'information, la nouvelle information sera stockée de la même façon que la première. Mais si la conversion de la deuxième impulsion se termine plus tôt que cette deuxième partie du sous-programme, la demande d'interruption envoyée par l'interface ne sera pas acceptée par le calculateur et cela tant que le sous-programme suscité par la première impulsion n'est pas terminé. La partie du sous-programme destinée au traitement de l'information

se compose surtout de la formation de l'adresse du mot mémoire du calculateur auquel il faut ajouter 1. Lorsque le sous-programme est terminé le calculateur commence à exécuter l'instruction suivante du programme principal.

Le fonctionnement exposé ci-dessus suppose un transfert automatique du convertisseur vers l'interface dès la fin de la conversion et ce transfert déclenche l'exécution de toutes les autres fonctions. Le système est tel que les autres modes de transfert permis par les convertisseurs CA 25 et CA 12 ne peuvent pas être pris en considération.

L'interface est conçue pour le calculateur C 90-10. Mais beaucoup d'ordinateurs fonctionnent avec des signaux de service équivalents à ceux utilisés ici (impulsions Eom, Pin, Rti, Rtf, lignes d'informations parallèles, et interruptions). De même les convertisseurs analogiques-numériques délivrent l'information sous forme de digits en parallèle et aussi une impulsion identique au "Pilote". Par conséquent cette interface doit pouvoir être utilisée avec d'autres types de calculateurs sous réserve d'adaptation des niveaux.





Niveau haut → "1"  
Niveau bas → "0"

Fig. 2-3

### III - TEMPS MORTS DU SYSTEME

Un système d'acquisition de données aléatoires doit avoir un temps mort le plus court possible. Le temps mort est le temps pendant lequel l'entrée du système est fermée après acceptation d'une information et pendant lequel aucune autre ne peut être acceptée.

Le temps mort de ce système se compose du temps mort propre au convertisseur, du temps nécessaire au transfert des données de l'interface dans la mémoire du calculateur et éventuellement du temps d'attente d'une interruption (fig. 3.1). On suppose que l'impulsion n-1 provoque une interruption aussitôt après la fin de conversion et que le sous-programme commence à se dérouler dès que l'instruction en cours d'exécution du programme principal est terminée. Dans ce cas le temps mort total comprend le temps mort du convertisseur et le temps nécessaire au transfert d'information de l'interface dans la mémoire du calculateur. Le transfert d'information s'exécute pendant le déroulement des instructions EOM et PIN. Après l'instruction PIN le calculateur poursuit l'exécution de la partie du sous-programme destinée à la formation d'adresse. La porte d'entrée du convertisseur s'ouvre et la conversion d'une nouvelle impulsion peut commencer.

Il faut ajouter que dans certains cas le programme principal ne peut pas être interrompu à la fin de l'instruction en cours, mais au terme de l'instruction suivante. Dans le cas de l'impulsion n-1, il a été supposé que l'interruption utilisée a une priorité maximum.

A l'apparition de l'impulsion n, qui trouve la porte du système ouverte après le transfert de l'information n-1, deux éventualités de temps morts sont possibles. Si la conversion est terminée après la fin du sous-programme du à l'impulsion n-1, l'impulsion n sera acceptée de la même façon que l'impulsion n-1. Mais si la conversion de l'impulsion n est terminée avant la fin du sous-programme déclenché par l'impulsion n-1, le calculateur n'accordera l'interruption qu'à la fin du sous-programme en cours. Il faut dans ce cas ajouter le temps d'attente au temps mort total, puisque la porte d'entrée est fermée. L'attente peut aussi être provoquée par l'exécution d'un sous-programme plus prioritaire, bien qu'il soit souhaitable d'utiliser l'interruption la plus prioritaire pour l'acquisition des données.

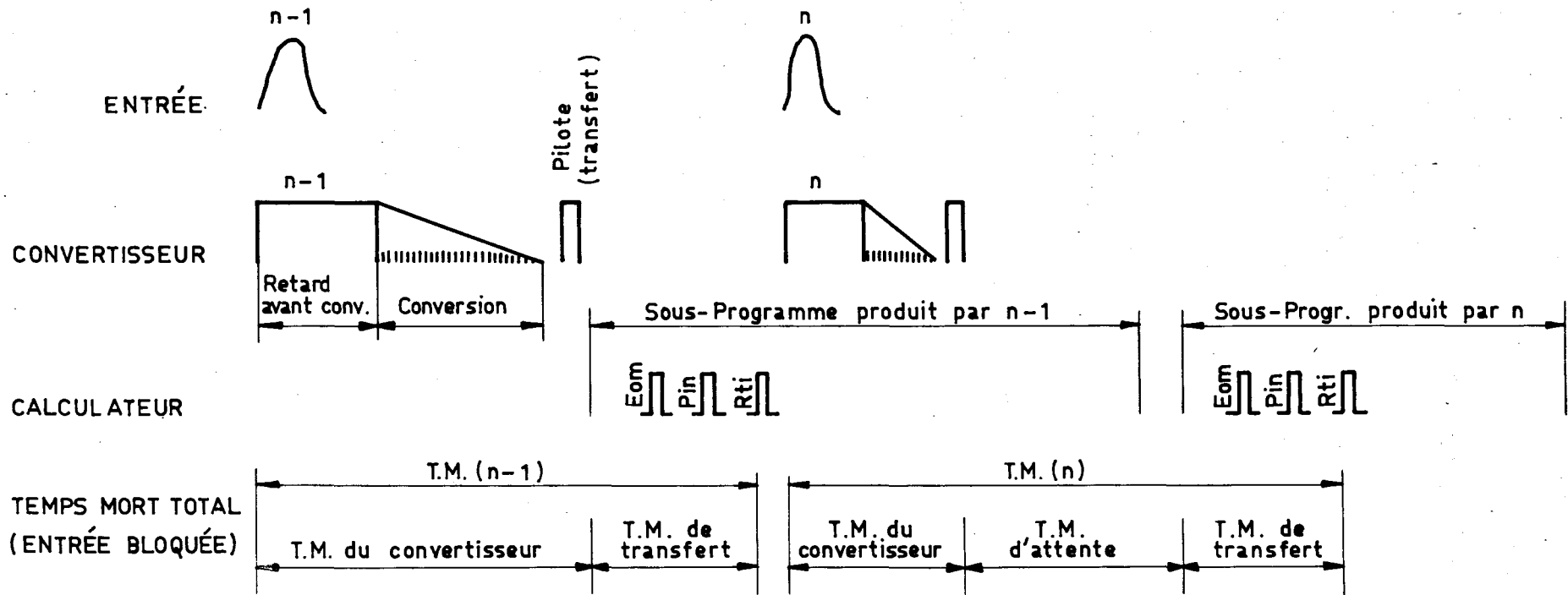


Fig. 3-1

On peut avoir plusieurs systèmes monoparamétriques non-corrélés, utilisant un seul ordinateur avec plusieurs convertisseurs, interfaces, interruptions et sous-programmes. Il faut dire, que dans ce cas, la hiérarchie des interruptions influence la quantité d'informations obtenues par les différentes voies, augmentant le temps mort des systèmes dont les interruptions sont moins prioritaires.

#### IV - PROGRAMMES DU SYSTEME MONOPARAMETRIQUE

Considérons maintenant les programmes du calculateur nécessaires au déroulement convenable des opérations. Rappelons qu'il est important de disposer de sous-programmes les plus courts possible pour diminuer le temps mort du système et pour ne pas interrompre trop longtemps le déroulement du programme principal.

Quelque soit le nombre de paramètres le sous-programme doit assurer les fonctions suivantes :

a) Interruption du programme principal et retour à l'instruction suivante de celui-ci après exécution du sous-programme.

b) Sélection de l'unité périphérique, c'est-à-dire dans le cas présent de l' "interface", et transfert des informations en mémoire du calculateur.

c) Traitement d'information reçue. Ceci consiste à déterminer une adresse dans la zone mémoire réservée au stockage d'information et à ajouter une unité au contenu du mot mémoire ainsi sélectionné.

Le mot de la mémoire du calculateur C 90-10 comportant 12 bits, un nombre de  $2^{12} = 4096$  peut être enregistré au maximum dans un mot. Pour un système monoparamétrique cela ne suffit pas le plus souvent. C'est la raison pour laquelle l'acquisition d'un spectre monoparamétrique exige la réservation de deux zones de stockage dans la mémoire. Dans la zone de poids faible (ZONE I) toutes les informations sont enregistrées. Lorsqu'un registre de la zone de poids faible a débordé, il faut ajouter une unité au mot correspondant de la zone de "poids fort" (fig. 4.1). Supposons que l'adresse du premier mot de la zone "poids faible" soit nommé ORIG et que l'écart entre les adresses respectives du premier mot de la zone "poids faible" et du premier mot de la zone "poids fort" (ou la distance de n'importe quelle paire de mots correspondants dans les deux zones) soit nommé DIST, alors un débordement du mot X (X = l'information binaire reçue du convertisseur) de la zone "poids faibles", c'est à dire le mot d'adresse ORIG + X de la mémoire, exige l'addition d'une unité dans le mot d'adresse ORIG + X + DIST dans la zone "poids forts"(fig. 4.1).

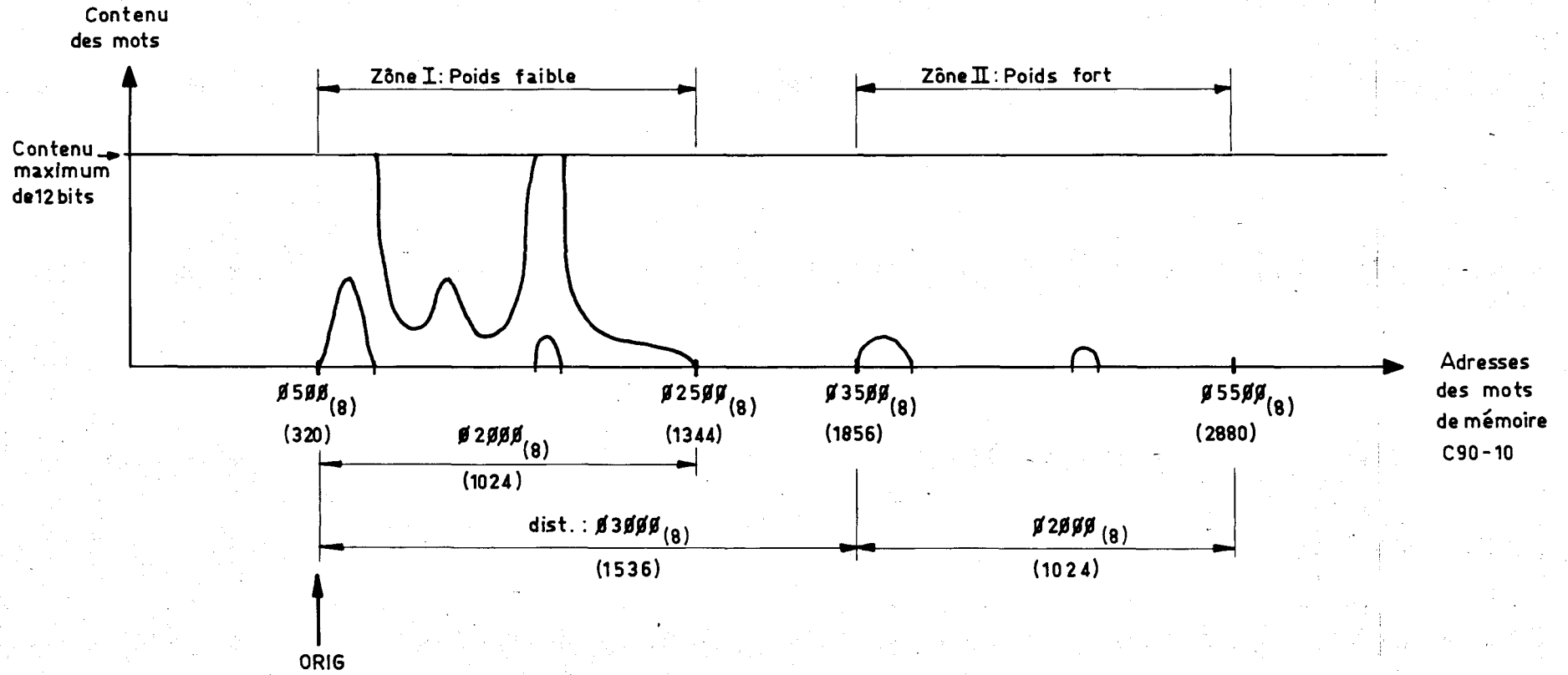


Fig. 4-1

Autrement dit le sous-programme doit vérifier si le contenu de la zone "poids faibles" a débordé. S'il en est ainsi il faut ajouter une unité dans le mot correspondant de la zone "poids forts". Sinon on peut tout de suite terminer le sous-programme. Dans la figure 4.1 on a supposé que ORIG = 0500 et DIST = 03000 en numération octale (les valeurs décimales sont entre parenthèses). Un exemple de sous-programme\* concernant un système mono-paramétrique et répondant à ce que nous avons exposé est présenté ci-dessous :

SYSTEME MONOPARAMETRIQUE  
SOUS - PROGRAMME

	AORG	0500
ORIG	RES	02000
AMPLIT	DATA	0,0
DIST	EQU	03000
	AORG	03500
DEBORIG	RES	02000
	AORG	0200
SP	PZE	
	EOM	030025
	PIN	AMPLIT + 1
	LDA	= ORIG
	MPA	AMPLIT + 1
	MPO	* AMPLIT
	BFT	DEBSP
	BRC	* SP
DEBSP	LDA	= DIST
	MPA	AMPLIT + 1
	MPO	* AMPLIT
	BRC	* SP

---

\* Ce programme est écrit dans le langage symbolique exposé dans la réf. [3]

SYSTEME MONOPARAMETRIQUE

PROGRAMME REMISE A ZERO, SIMULATION DU PROGRAMME  
PRINCIPAL ET ARMEMENT D'INTERRUPTION

	AORG	Ø25Ø
CP	PZE	
	LDA	= ORIG
	STA	CP + 1
RETOUR	LDA	= Ø
	STA	* CP
	MPO	CP + 1
	LDA	= Ø55ØØ
	CEA	CP + 1
	BFT	RETOUR
PP	EIR	
	LDA	= 1
	SUA	= 1
	BRU	PP
	AORG	Ø174
	BMC	SP
	END	

L'adresse du mot auquel il faut ajouter une unité, est formée dans le mot AMPLIT + 1, bien que deux mots soient réservés pour la formation de l'adresse (AMPLIT et AMPLIT + 1). Dans le sous-programme présenté il a été supposé que l'adresse la plus grande possible est toujours contenue dans les douze bits du mot AMPLIT + 1 et par conséquent qu'elle ne dépasse jamais  $Ø7777(8) = 4095(10)$ . Si l'adresse est plus grande il faut inclure dans le sous-programme les instructions pour le transfert du débordement éventuel de AMPLIT + 1 dans AMPLIT. Ces instructions ont été omises pour obtenir un sous-programme le plus court possible. Un tel choix est possible parce que les emplacements de mémoire utilisés par les autres types de programme n'ont pas d'importance.



On peut, bien sûr, prendre la distance  $DIST = 02000$  au lieu de  $03000$  comme dans la fig. 4.1 et avoir une zone "poids forts" aussitôt après la zone "poids faible" ou bien choisir n'importe quelle  $DIST$  plus grande que  $02000$ .

Avant de commencer le stockage une remise à zéro générale des contenus des ZONE I et II est nécessaire. Un exemple du programme de remise à zéro de la mémoire entre les adresses  $05500$  est décrit page 15, ainsi que la simulation du programme principal et l'armement d'interruption.

## V - SYSTEME BIPARAMETRIQUE

Le système biparamétrique utilisant deux convertisseurs, rend possible la mesure de deux grandeurs issues du même événement et conditionnées l'une par l'autre. Par exemple, l'énergie du rayonnement radioactif mesurée par un détecteur A est le paramètre X et l'énergie mesurée par le détecteur B est le paramètre Y. Chacune des impulsions d'entrée, d'amplitude différente, est analysée par un convertisseur. Lorsque la conversion la plus longue est terminée les informations des convertisseurs sont transférées dans les interfaces respectives (fig. 2.1) et une demande d'interruption de programme principal est effectuée. Quand celle-ci est accordée le sous-programme est déclenché et il y a transfert d'une information, puis de l'autre. Utilisant les deux informations reçues, le calculateur forme l'adresse nécessaire et ajoute une unité au contenu de cette adresse.

Le nombre de mots mémoire nécessaire est le produit des nombres  $X_{\max}$  et  $Y_{\max}$ . Par exemple deux convertisseurs à sept bits chacun (informations 0 - 127) exigent  $2^{14}$  emplacements de mémoire du calculateur (fig. 5.1).

Si le rangement dans la zone mémoire réservée pour l'acquisition des données est effectué par adresses croissantes à partir d'une origine (ORIG) comme sur la fig. 5.1, l'adresse peut être calculée selon la formule :

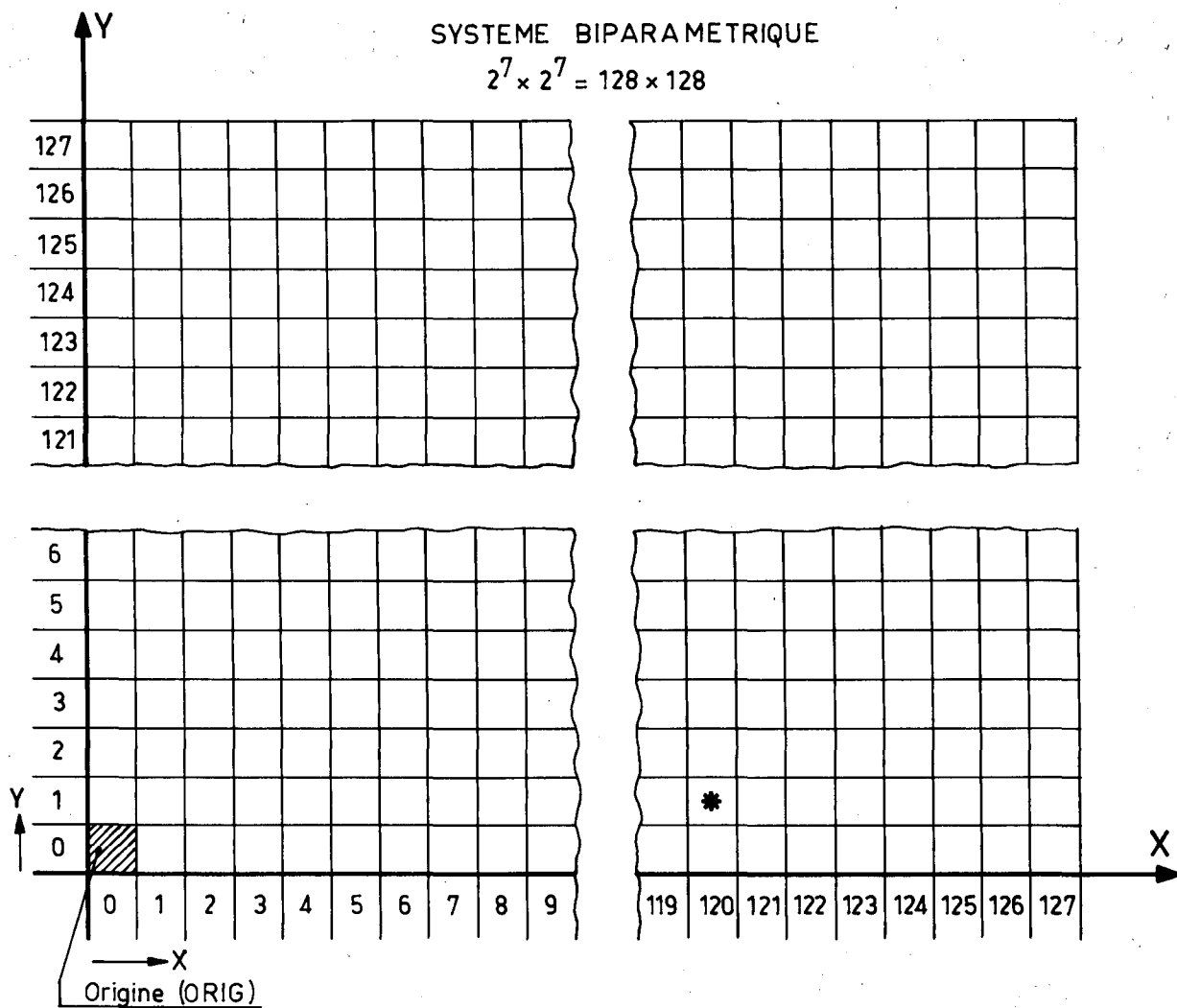
$$\text{Adresse}_2 = Y \cdot 2^7 + X + \text{ORIG} \quad (1)$$

Pour  $Y = 0$  les adresses vont en croissant à partir du début de zone vers la droite (fig. 5.1). Une fois que  $X_{\max}$  est atteint ( $Y = 0$ ,  $X = 127$ , ou le 128ème mot de la zone réservée, le 129ème mot de la zone est le mot  $Y = 1$ ,  $X = 0$ , et ensuite  $Y = 1$ ,  $X = 1$  est le 130ème, etc. Autrement dit, si on veut représenter la zone réservée pour l'acquisition par un axe orienté, les adresses croissant vers la droite, (comme sur la fig. 4.1 pour le système monoparamétrique), il faudrait découper sur la fig. 5.1 la bande de papier  $Y = 1$  et la poser à droite de la bande  $Y = 0$ . A l'extrémité droite de  $Y = 1$  il faudrait poser la bande  $Y = 2$ , puis  $Y = 3$  etc. Il est important de savoir cela lorsque l'on sort le contenu de la zone.

Si par exemple après la conversion on avait  $X = 120$  et  $Y = 1$ , il faudrait ajouter un, d'après la fig. 5.1, au 249ème mot de la zone.

SYSTEME BIPARAMETRIQUE

$$2^7 \times 2^7 = 128 \times 128$$



Y	0	0	0	0	0	0
X	0	1	2	3	4	5
Mot de la zone mémoire réservée	1	2	3	4	5	6

0	0	0	0	0	0	0	0	0
119	120	121	122	123	124	125	126	127
120	121	122	123	124	125	126	127	128

Y	1	1	1	1	1	1
X	0	1	2	3	4	5
Mot de la zone mémoire réservée	129	130	131	132	133	134

1	1	1	1	1	1	1	1	1
119	120	121	122	123	124	125	126	127
248	249	250	251	252	253	254	255	256

Fig. 5-1

On obtient ce résultat en utilisant l'adresse de rangement donnée par la formule (1).

Il faut maintenant voir comment l'adresse sera formée dans le cas du système biparamétrique d'après la formule (1). Supposons que l'information reçue du convertisseur X se trouve dans le mot mémoire ADRES X + 1 et l'information reçue du convertisseur Y dans le mot mémoire ADRES Y. Supposons aussi, par exemple, que les informations X et Y ont été  $X = 127$  et  $Y = 127$  c'est-à-dire maximum. Alors l'adresse à laquelle il faudra ajouter un, sera le dernier mot de la zone mémoire réservée. Les états des registres A et B et du "bit témoin" (F) du calculateur C 90-10 sont présentés sur la fig. 5.2. D'abord le registre A est remis à zéro et l'information Y est transférée dans le registre B. Un décalage de 7 positions binaires de A et B fait la multiplication de Y par  $128 = 2^7$ . Ensuite l'information X est ajoutée. On suppose qu'après le décalage de 7 positions (opération CYD 7) et l'addition d'adresse X ( $X_{max} = 127$ ) le registre B ne peut pas déborder, et qu'il n'y aura jamais de report. Il faut encore ajouter ORIG (sur la fig. 5.2  $0500_{(8)}$  en octal ou  $320_{(10)}$  en décimal. Cette fois on peut arriver à un débordement et le report est transmis dans le bit témoin (F) du calculateur C 90-10. Ensuite le contenu du registre B est transmis dans l'emplacement de la mémoire ADRESX + 1 et le contenu du registre A dans l'emplacement ADRESX. Ajoutons encore l'éventuel report du registre B qui se trouve dans F à l'emplacement ADRESX et l'adresse complètement formée se trouvera dans les emplacements de la mémoire ADRESX et ADRESX + 1. Pour l'exemple considéré on obtient l'adresse  $40477_{(8)} = 16703_{(10)}$  au contenu de laquelle il faut ajouter une unité. Un exemple de sous-programme pour acquisition biparamétrique est décrit page 15 pour le cas où l'adresse de l'interface X est  $025_{(8)}$  et celle de l'interface Y est  $022_{(8)}$ .

On peut faire des programmes semblables à celui présenté pour d'autres combinaisons X, Y (fig. 5.3). Mais le C 90-10 peut avoir au maximum une mémoire de 32 k et il faut garder un certain nombre d'emplacements mémoire pour le programme principal et les sous-programmes. D'autre part les convertisseurs CA 12 et CA 25 peuvent délivrer l'information dans un format de 5 à 10 bits. Dans ces conditions il apparaît que le maximum d'emplacements mémoire utilisables pour l'acquisition de données biparamétriques

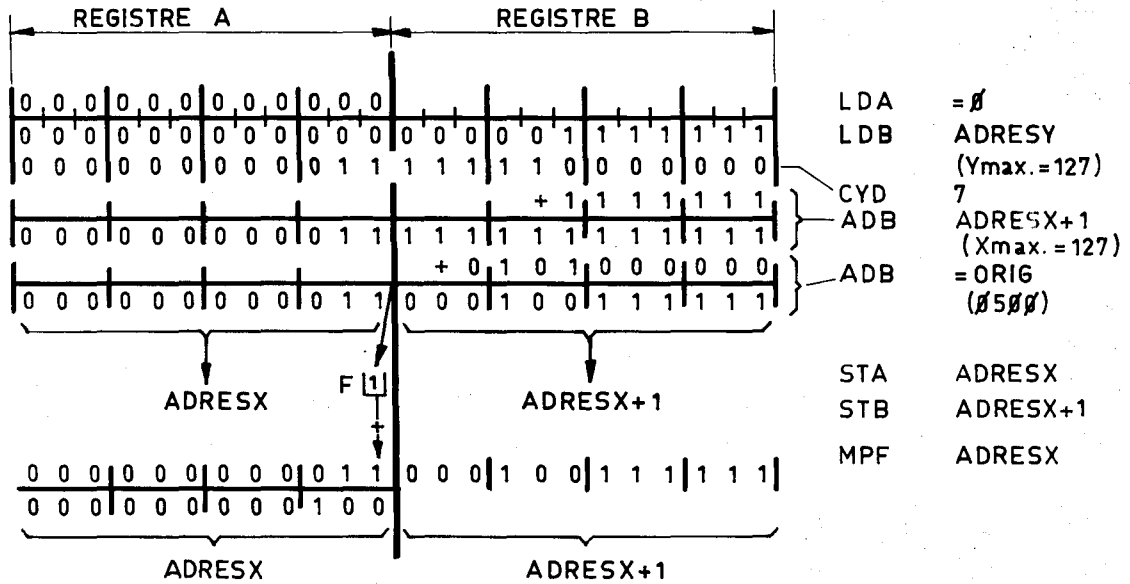
est de 16 k, c'est à dire 14 bits, avec les combinaisons X, Y suivantes :  
 $2^7 \times 2^7$ ,  $2^6 \times 2^8$ ,  $2^5 \times 2^9$ .

Le nombre des évènements mesurés par le système biparamétrique est, à cause des coïncidences nécessaires, assez faible. D'autre part le nombre des emplacements de mémoire nécessaires pour le stockage est très élevé. C'est pourquoi l'éventualité du débordement n'a pas été prise en considération.

Avant de commencer l'acquisition des données dans la zone de la mémoire il faut remettre à zéro une zone plus grande qu'elle n'était dans le système monoparamétrique. Un programme qui remet à zéro la zone de ORIG à ORIG + 040 000<sub>(8)</sub> est présenté page 13.

Quand on utilise les deux convertisseurs du CA-25 ou du [CA-12] dans un système biparamétrique le transfert d'information des convertisseurs aux interfaces se produit simultanément lorsque la conversion la plus longue est terminée. Une seule demande d'interruption est nécessaire et elle peut être fournie par l'une ou l'autre interface. D'autre part chaque interface peut bloquer l'entrée du convertisseur correspondant.

ETATS DES REGISTRES A, B et F DU CALCULATEUR C90-10 PENDANT LA FORMATION D'ADRESSE



SYSTEME BIPARAMETRIQUE  $2^7 \times 2^7 = 128 \times 128$

Fig. 5-2

SYSTEME BIPARAMETRIQUE  $2^6 \times 2^8 = 64 \times 256$

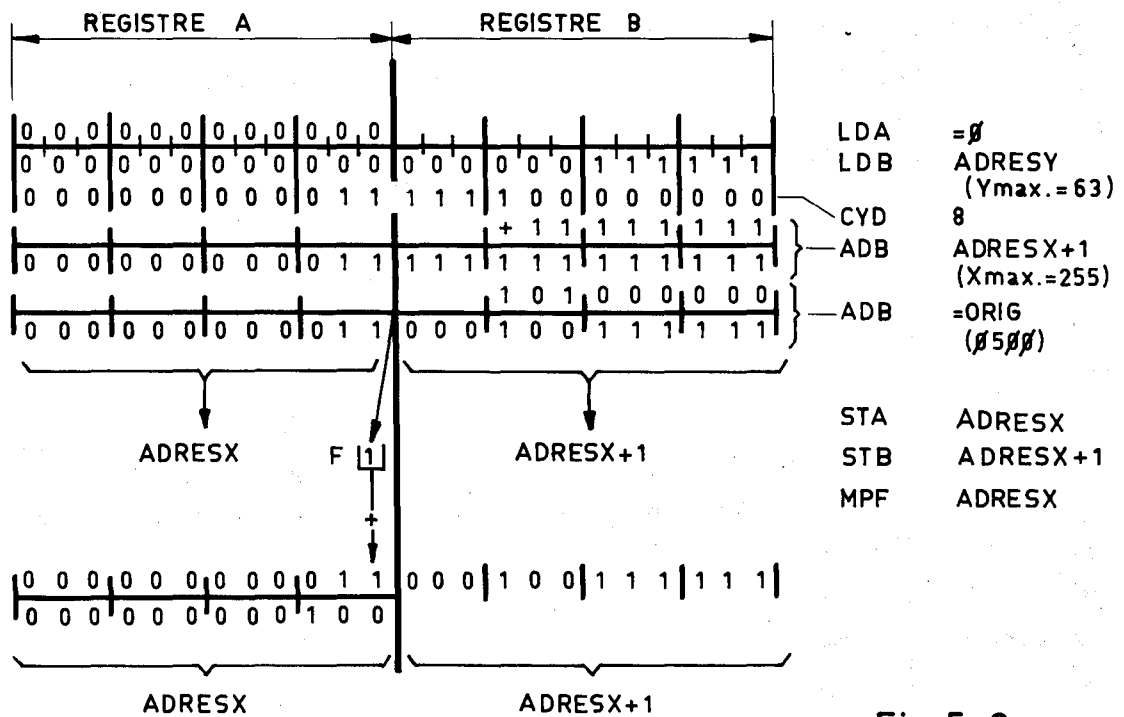


Fig. 5-3

SYSTEME BIPARAMETRIQUE  $X=2^7$ ,  $Y=2^7$

SOUS PROGRAMME

	AORG	Ø 5ØØ
ORIG	RES	Ø 4ØØØØ
ADRESX	DATA	Ø , Ø
ADRESY	DATA	Ø
SPBI	PZE	
	EOM	Ø3ØØ22
	PIN	ADRESY
	EOM	Ø3ØØ25
	PIN	ADRESX+1
	LDA	= Ø
	LDB	ADRESY
	CYD	7
	ADB	ADRESX+1
	ADB	= ORIG
	STA	ADRESX
	STB	ADRESX+1
	MPF	ADRESX
	MPO	* ADRESX
	BRC	* SPBI

SYSTEME BIPARAMETRIQUE  
PROGRAMME REMISE A ZERO, SIMULATION DU PROGRAMME  
PRINCIPAL ET ARMEMENT D'INTERRUPTION

	AORG	Ø25Ø
COMPT	PZE	
	LDA	= ORIG
	STA	COMPT+1
DEB	LDA	= Ø
	STA	* COMPT
	MPO	COMPT+1
	MPF	COMPT
	LDA	= ORIG
	CEA	COMPT+1
	BFT	DEB
	LDA	= 4
	CEA	COMPT
	BFT	DEB
	EIR	
	LDA	= 1
	SUA	= 1
	BRU	§ - 5
	AORG	Ø174
	BMC	SPBI



## VI - SYSTEME TRIPARAMETRIQUE

Le troisième paramètre d'une expérience nucléaire sera le plus souvent le temps. Bien qu'une interface qui rendrait possible le transfert des informations d'un convertisseur temps-numérique vers le calculateur n'ait pas été construite, l'étude du programme du système triparamétrique a été abordée. Elle est une suite logique du système biparamétrique.

Si le troisième paramètre est aussi une amplitude, il faudra ajouter au système déjà présenté un circuit de demande d'interruption (fig. 6.1). De plus, la sortie de "temps mort" du système X-Y doit être connectée au "blocage extérieur" du système Z, et réciproquement .

Toujours dans le cas où le maximum de la mémoire disponible est de  $2^{14}$ , considérons l'exemple d'un système  $X = 2^5$ ,  $Y = 2^5$  et  $Z = 2^4$  (cf. fig. 6.2 où Z est le troisième paramètre). Supposons que les adresses de la zone réservée augmentent à partir du premier mot de la zone (ORIG :  $Z = 0$ ,  $Y = 0$ ,  $X = 0$ ) d'abord dans la surface  $Z = 0$  de la même façon que dans le cas du système biparamétrique. Après le dernier mot de la surface  $Z = 0$  c'est-à-dire  $Y = 31$ ,  $X = 31$ , arrive le premier mot de la surface  $Z = 1$  avec  $Y = 0$ ,  $X = 0$ . Les adresses de la deuxième surface augmentent de la même façon, etc. . .

Autrement dit, on augmente d'abord X jusqu'au maximum possible, ensuite Y et enfin Z. En conséquence l'adresse à laquelle il faut ajouter une unité peut se calculer ainsi :

$$\text{Adresse}_3 = Z \cdot 2^{10} + Y \cdot 2^5 + X + \text{ORIG} \quad (2)$$

L'état des registres A, B, F du calculateur C 90-10 ainsi que le programme nécessaire à la formation de cette adresse sont décrits fig. 6.3 en considérant le cas  $X = X_{\max} = 31$ ,  $Y = Y_{\max} = 31$  et  $Z = Z_{\max} = 15$ .

Un exemple du programme complet est décrit page 27.

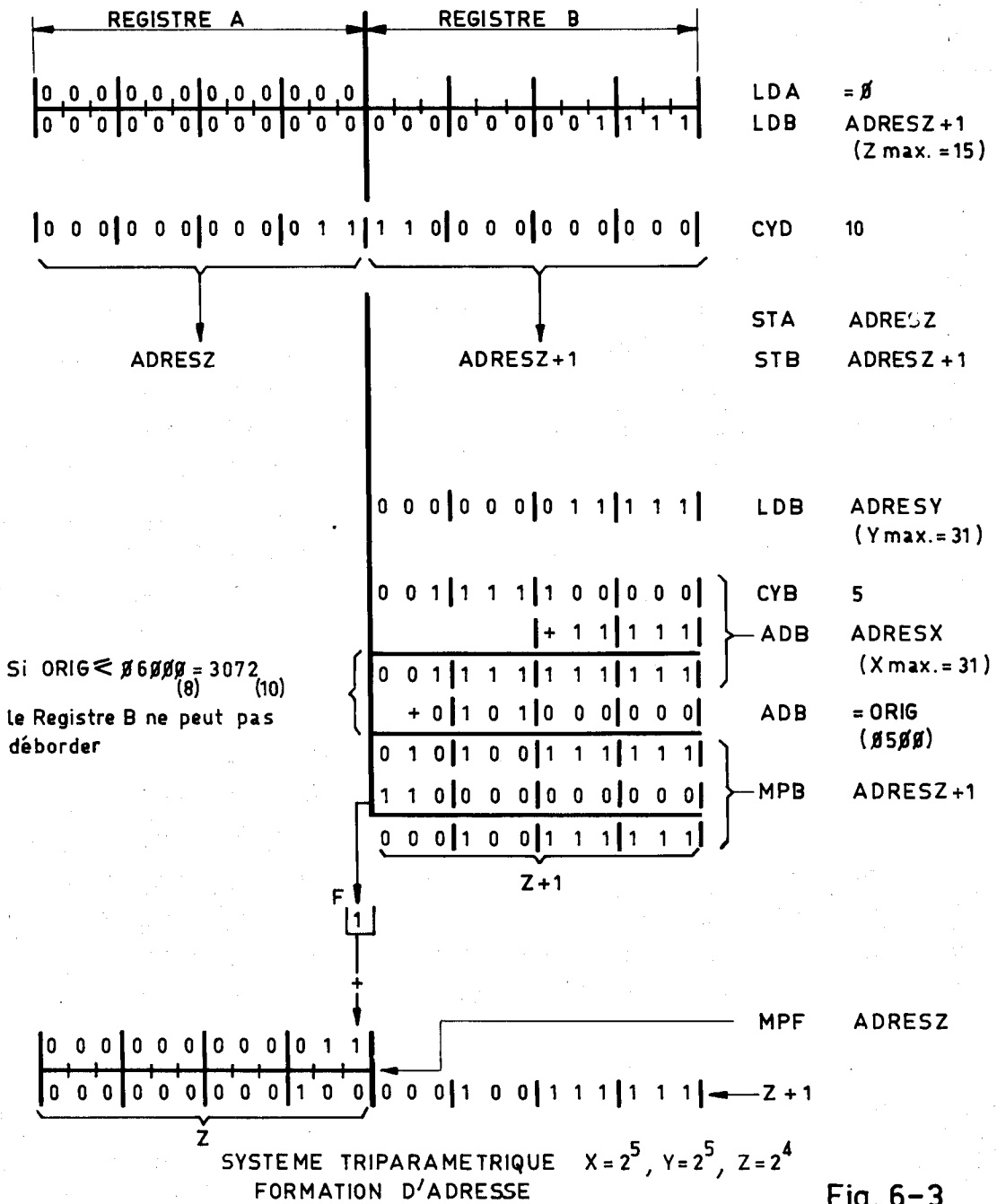


Fig. 6-3

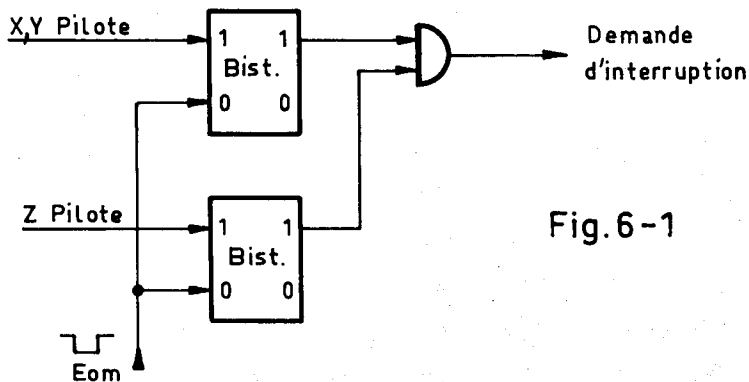


Fig. 6-1

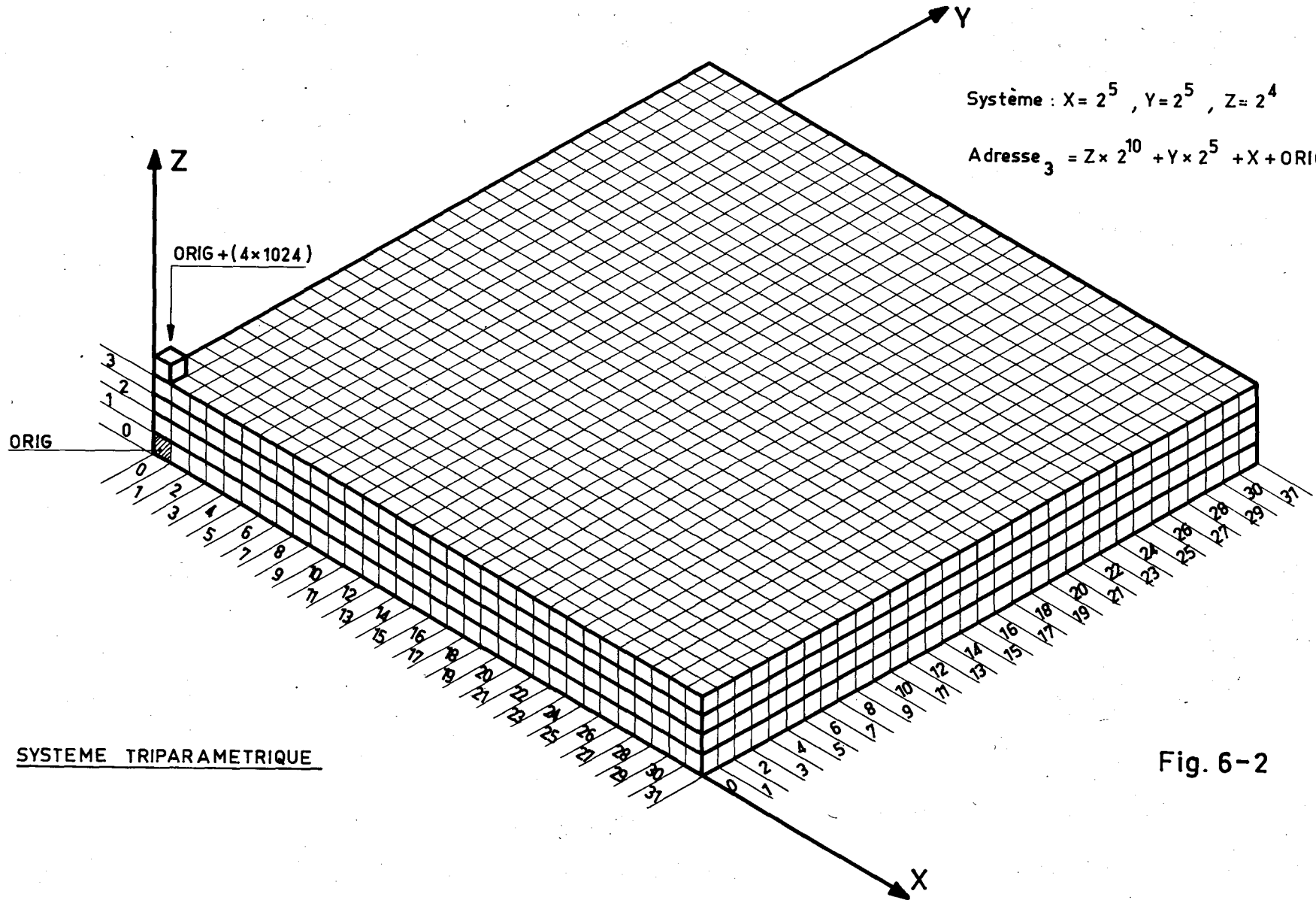


Fig. 6-2

SYSTEME TRIPARAMETRIQUE  $X=2^5$ ,  $Y=2^5$ ,  $Z=2^4$

SOUS PROGRAMME

	AORG	Ø 5ØØ
ORIG	RES	Ø 4ØØØØ
ADRESX	DATA	Ø
ADRESY	DATA	Ø
ADRESZ	DATA	Ø, Ø
SPTRI	PZE	
	EOM	Ø 3ØØ2Ø
	PIN	ADRESZ+1
	EOM	Ø 3ØØ22
	PIN	ADRESY
	EOM	Ø 3ØØ25
	PIN	ADRESX
	LDA	= 0
	LDB	ADRESZ+1
	CYD	10
	STA	ADRESZ
	STB	ADRESZ+1
	LDB	ADRESY
	CYB	5
	ADB	ADRESX
	ADB	= ORIG
	MPB	ADRESZ+1
	MPF	ADRESZ
	MPO	* ADRESZ
	BRC	* SPTRI

## VII - REALISATION DE L'INTERFACE

La réalisation de l'interface est présentée sur la fig. 7.1. Cette interface est connectée au calculateur à travers une console d'adaptation et de standardisation de niveaux en logique négative (impulsions Eom, Pin, Rtf, Rti, lignes d'information). Seule la demande d'interruption est liée directement au calculateur. La présence d'un niveau +12V signifie une demande d'interruption. D'autre part les impulsions sortant des convertisseurs (pilote, digits d'information, ainsi que l'impulsion de blocage du convertisseur) ont une forme correspondante au standard adopté pour la chaîne d'analyse multiparamétrique INTERTECHNIQUE (+7V, 1 $\mu$ s).

Les portes de sortie sont nécessaires pour avoir la possibilité de couper la liaison entre le calculateur et l'interface quand celle-ci n'est pas utile et notamment quand d'autres unités périphériques doivent être connectées au calculateur. Les portes sont réalisées par des circuits intégrés type SN7401N de la série TTL et elles exigent une résistance de rappel extérieure. Elles se trouvent dans le tiroir de connexion au calculateur (console).

Dans son fonctionnement normal, le convertisseur CA 12 ajoute automatiquement après la conversion un temps mort de 50 à 60 $\mu$ s pour en rendre possible l'exploitation. Puisque dans le système considéré le temps mort est défini par l'interface un tel temps mort est inutile et nuisible et il faut le supprimer. Pour ce faire, l'impulsion Pin attaque l'entrée "Fin de programme" du CA 12. Ainsi le temps mort est défini par l'interface comme pour le CA 25.

**BIBLIOGRAPHIE INDISPENSABLE**

1. C 90-10 - Computer Interface Design manual (CII)
2. C 90-10 - Manuel d'utilisation (CII)
3. Le langage SYMBOL CAE 90-10 et le programme d'Assemblage  
(CAE - NT - 1856)
4. CA-25 - Notices (Inter technique)
5. CA-12 - Notices (Inter technique)

*Manuscrit reçu le 11 Avril 1969*

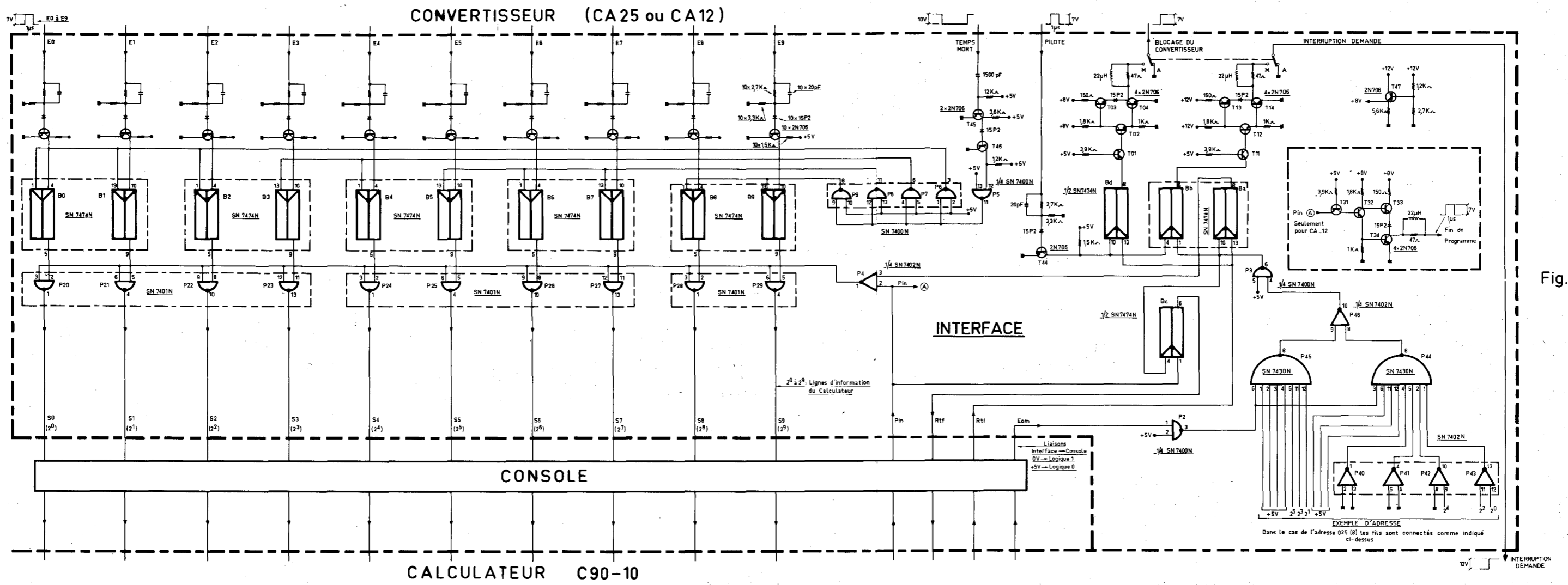


Fig. 7-1

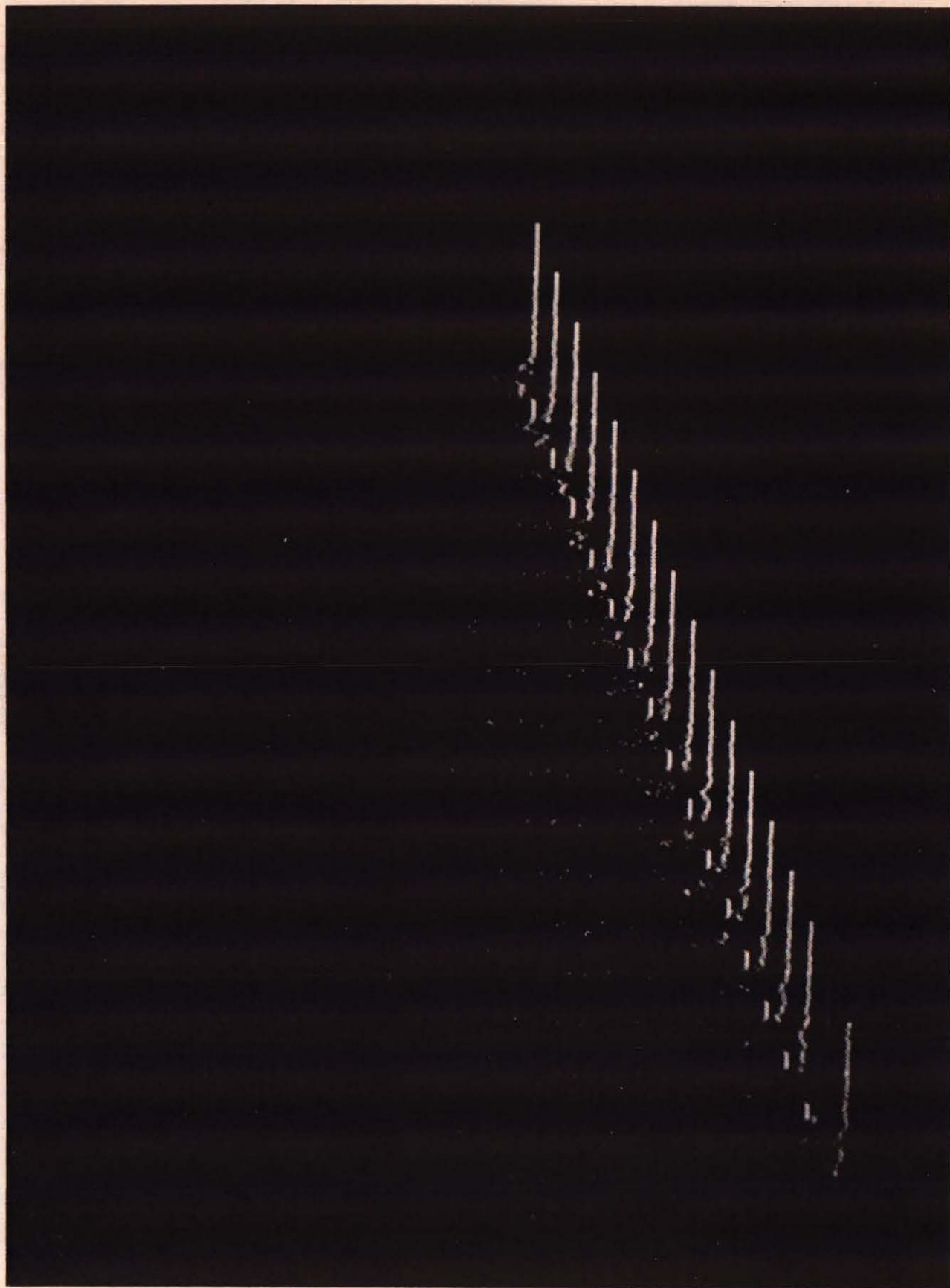


Fig. 7.2 - Spectre biparamétrique du  $\text{Co}^{60}$ , vue isométrique



*Edité par*

*le Service Central de Documentation du C.E.A.*

*Centre d'Etudes Nucléaires de Saclay*

*Boîte Postale n° 2*

*91 - GIF-sur-YVETTE (France)*