

*Tex*

وزارة التعليم العالي  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

## ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : *ELECTRONIQUE*

# PROJET DE FIN D'ETUDES

### SUJET

CONCEPTION et REALISATION

D'UN DEMODULATEUR DPSK à 280

Proposé par :  
M<sup>r</sup> J. GORALSKI

Etudié par :  
I - KACHA

Dirigé par :  
M<sup>r</sup> J. GORALSKI

PROMOTION :

JUIN 1990



الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT :

## PROJET DE FIN D'ETUDES

SUJET

Proposé par :

Etudié par :

Dirigé par :

PROMOTION :

DEDICACES

À ma mère, à ma mère, à ma mère.

À ma mère qui m'a toujours soutenu, qui m' a entouré d'une immense affection et à qui j'espère avoir réalisé ses souhaits.

À mon père qui a tant souffert pour me rendre faciles tant de choses.

À la mémoire de mon oncle "El-hacene", tombé au champ d'honneur.

À Nadja, Taous et Latifa.

À Hassan, Sofiane et Younes.

À Hakim.

À toute la famille et à tous les amis.

Et sans oublier la petite SARA.

Brahim

## REMERCIEMENTS

Je ne saurais présenter cette étude sans mentionner les personnes de près ou de loin, matériellement ou moralement ont contribué à son aboutissement, aussi j'adresse mes remerciements :

À tous les enseignants de l'ENP qui ont contribué à ma formation, et en particulier à M. J GORALSKI qui a dirigé cette étude et dont les précieux conseils m'ont permis d'acquérir beaucoup de connaissances.

À M. A TAGHLIT. de l'ENSI.

Ibrahim KACHA

## SOMMAIRE

Introduction .....	1
A/ Généralités sur la transmission de données .....	2
I) Définitions .....	2
I-1) Signaux numériques .....	2
I-2) Débit binaire; vitesse de modulation .....	3
II) Transmission en bande transposée .....	5
II-1) Modulation .....	5
II-2) Transmission physique de données .....	6
B/ Développement d'un algorithme de démodulation DPSK .....	9
III) Modulation PSK et DPSK .....	9
III-1) Modulation PSK .....	9
III-2) Modulation DPSK .....	10
III-3) Code de modulation standards pour la transmission de données en DPSK .....	12
IV) Démodulation DPSK .....	14
IV-1) Notion de produit scalaire .....	14
IV-2) Développement d'un algorithme de démodulation DPSK	
a) Méthode d'auto-corrélation .....	16
b) Méthode de corrélation .....	18
c) Méthode synchrone .....	21
C/ Réalisation pratique .....	23
V) Description générale .....	23
V-1) Synoptique général d'un ETCD récepteur .....	23
V-2) Fiche technique .....	24
V-3) Le microprocesseur Z80 .....	24
V-4) Synoptique général de la carte démodulateur .....	25

VI) Description détaillée de la réalisation .....	29
VI-1) Unité centrale de traitement .....	29
VI-2) Schéma d'adressage .....	30
VI-3) Interface d'entrée .....	31
VI-4) Interface de sortie .....	32
VII) Programme moniteur .....	33
VII-1) Présentation générale .....	33
VII-2) Sous-programme BITA .....	35
VII-3) Sous-programme BITB .....	37
VII-4) Sous-programme DIBITA .....	37
VII-5) Sous-programme DIBITB .....	37
Conclusion .....	44
Bibliographie .....	46
Annex e .....	47

## Introduction :

A l'aube des années 1990, troisième décennie de la "Révolution informatique", les télécommunications prennent une place de plus en plus importante parmi les services que nous offrent les ordinateurs. Chaque fois qu'il y a communication entre des ordinateurs séparés par des distances importantes, il y a de très fortes chances qu'un modem soit présent quelque part sur le circuit. Les progrès considérables enregistrés dans les domaines de l'informatique et des télécommunications ont donné naissance à une nouvelle discipline : la téléinformatique.

Dans ce même sens, il nous a été demandé de concevoir et de réaliser un démodulateur par saut de phase différentiel (DPSK : Differential Phase-Shift keying), autour d'un microprocesseur de la famille Z80. Pour la présentation de notre travail, nous avons adopté le plan suivant :

La première partie (partie A), donne quelques notions fondamentales et quelques définitions sur la transmission de données. La partie B, traite l'élaboration d'un algorithme de démodulations DPSK, tout en exposant brièvement les modulation PSK et DPSK. Dans la partie C nous avons décrit la réalisation pratique et le programme moniteur. Une conclusion qui est axée sur une discussion du programme et des suggestions pour une éventuelle amélioration du système. Et à la fin de cette étude une bibliographie et une annexe contenant le listing du programme moniteur source, les différents brochages de CI utilisés, ainsi que la planche du schéma électrique général.

## A/ Généralités sur la transmission de données :

### I) Définitions :

#### I-1) Signaux numériques :

La transmission nécessite sur le plan temporel une synchronisation, afin de permettre au récepteur de reconnaître notamment le début et la fin d'un message, on distingue :

##### a) Signaux synchrones :

Un signal est dit synchrone si les intervalles de temps alloués à chaque symbole, sont égaux et coïncident avec une base de temps.

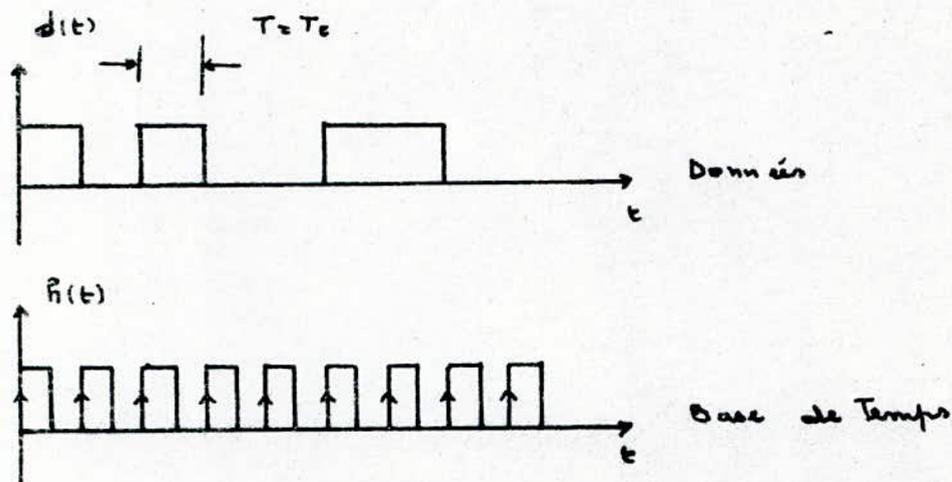


Fig.I-1: Signal synchrone

##### b) signaux asynchrones :

###### - Signaux anisochrones :

Le nombre d'états est fini et la durée des états n'est pas un multiple de l'intervalle élémentaire  $T_e$ . L'intervalle entre deux transitions successives est supérieure ou égale à  $T_e$ .

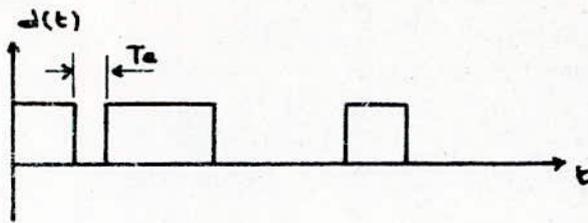


Fig.I-2: Signal anisochrone

**-Signaux arithmiques :**

Chaque caractère se présente sous forme d'un petit bloc d'information synchrone qui débute par un bit de départ (start) et se termine par un bit d'arrêt (stop).

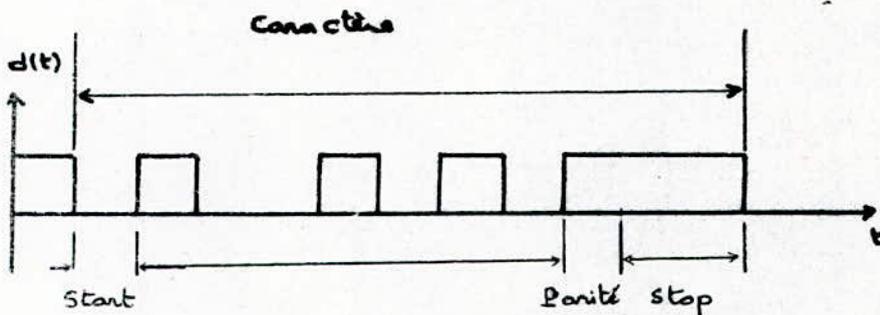


Fig.I-3: signal arithmique

**I-2) Débit binaire; vitesse de modulation :**

Le débit binaire  $D$  est le nombre maximum de bits transmis par seconde :

$$D = 1/T \text{ (bits par seconde)} \quad (I-1)$$

$T$  : durée d'un bit .

Ceci dans le cas d'une transmission en mode synchrone pour un signal bivalent.

On appelle vitesse de modulation ou "Rapidité de transmission", le nombre d'intervalles élémentaires par

seconde, elle s'exprime en bauds :

$$\nu_m = 1/T_e \quad (\text{bauds}) \quad (\text{I-2}).$$

D'une manière générale la notion de débit binaire est différente de la notion de vitesse de modulation, dans le cas d'une transmission synchrone, on a la relation suivante :

$$D = \nu_m \cdot \log_2(w) \quad (\text{I-3})$$

avec :  $w = 2^k \quad (\text{I-4})$

$k$  étant le nombre de bits contenu dans un signal émis (voir l'exemple de la fig. I-4).

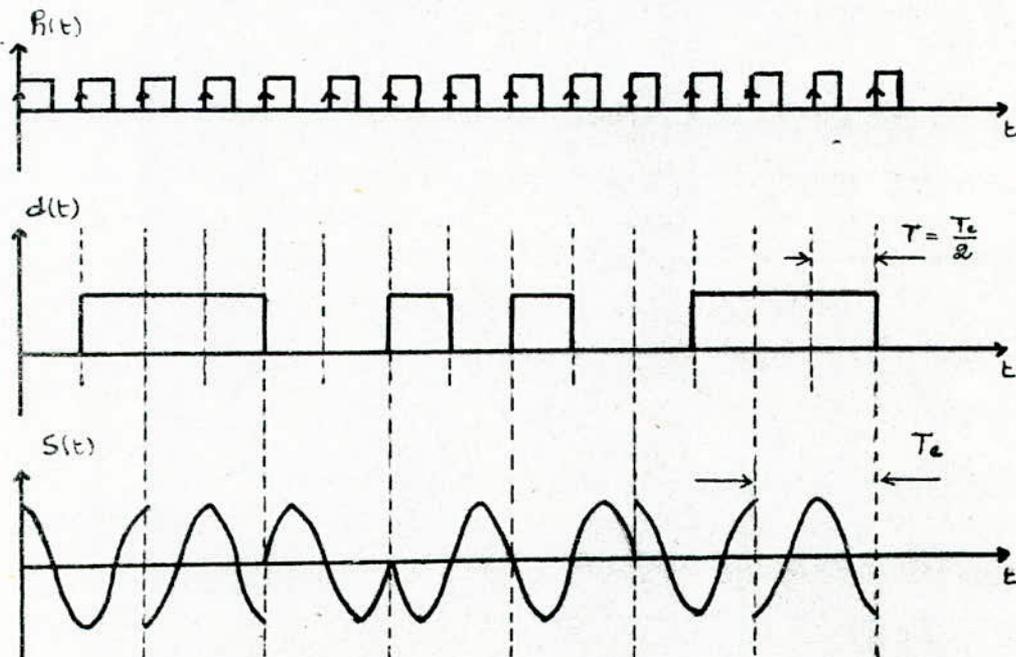


Fig. I-4: Modulation par saut de phase

Dibit	Phase
00	0°
01	90°
10	180°
11	270°

Tab. I-1: Code de modulation.

## II) Transmission en bande transposée :

La plupart des supports de transmission, et en particulier, le canal téléphonique, ne permettent pas la transmission directe d'un signal numérique en bande de base. Cette limitation est due essentiellement au fait, que la bande de fréquence occupée par le signal transmis ne coïncide pas avec la bande passante du support de transmission.

Ce problème peut être résolu en modulant une sinusoïde porteuse de fréquence convenable par le signal à transmettre. L'opération de modulation équivaut en effet, à une translation du spectre du signal dans le domaine des fréquences et permet de centrer son énergie à l'intérieur de la bande passante du support. L'opération de démodulation effectuée dans le récepteur est une translation égale à la précédente et de signe opposée, elle restitue le signal numérique sous sa forme initiale.

### II-1) Modulation :

La modulation est une opération qui consiste à transformer le message en bande de base en un signal  $S_n(t)$ , tel que, la connaissance de  $S_n(t)$  sur un intervalle significatif :  $[nT_m, (n+1)T_m]$  ( $T_m = 1/\nu_m$ ), permet de retrouver le caractère  $A_n$  ( $A_n = d_1 d_2 \dots d_k$ , avec  $d_i \in [0,1]$ ) correspondant à cet intervalle. Le signal  $S_n(t)$  est appelé : "signal émis". Il est généralement obtenu en variant les paramètres d'une onde sinusoïdale.

#### a) Modulation par saut d'amplitude (ASK) :

L'amplitude de la sinusoïde transmise est choisie parmi un ensemble discret d'amplitudes possibles (voir Fig. II-1).

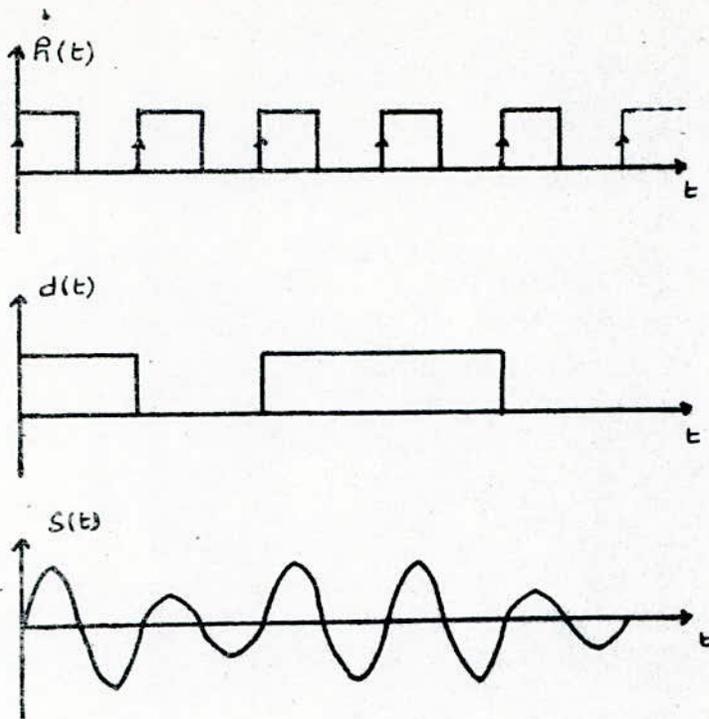


Fig.II-1: Modulation ASK

**b) Modulation par déplacement de fréquence (FSK) :**

La fréquence de la sinusoïde transmise est choisie parmi un ensemble discret de fréquences possibles (voir Fig.II-2).

**c) Modulation par déplacement de phase (PSK) :**

La phase de la sinusoïde transmise est choisie parmi un ensemble discret de phases possibles (voir Fig.I-4 et Tab.I-1).

**II-2) Transmission physique de données :**

L'échange de messages de données entre un terminal A et un terminal B, éloignés géographiquement l'un de l'autre, est réalisé à l'aide d'équipements spéciaux, par l'intermédiaire du réseau de télécommunications.

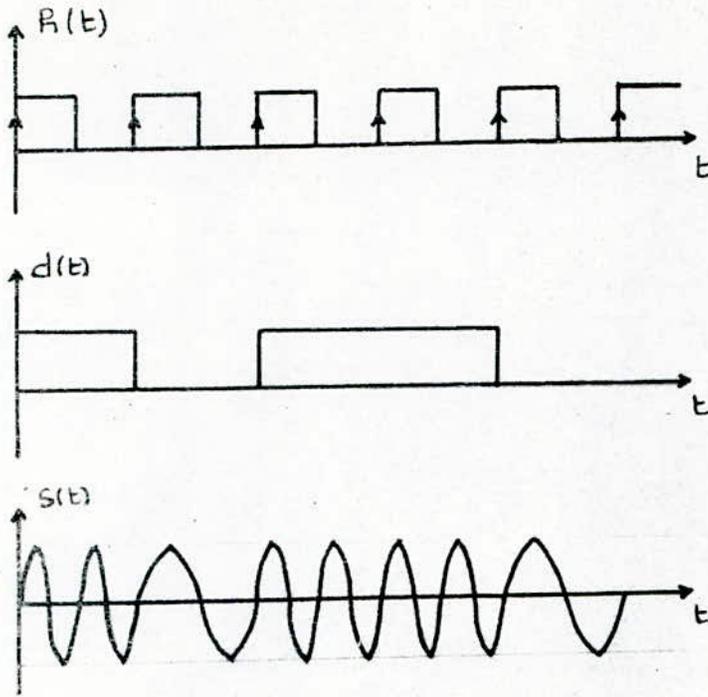


Fig.II-2: Modulation FSK

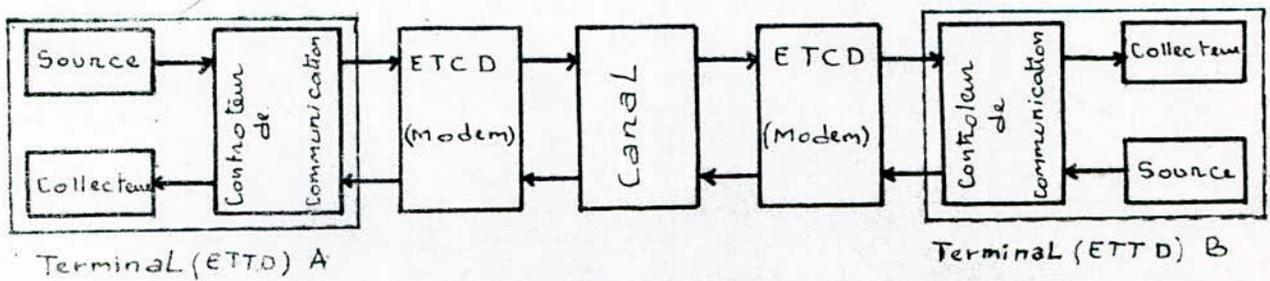


Fig.II-3: Synoptique général d'un système téléinformatique

a) Equipement terminal de traitement de données (ETTD) :

L'information est émise ou reçue par un ETTD, souvent appelé terminal, mais qui peut être un ordinateur ou terminal.

dans un ETTD nous distinguons deux parties : la machine de traitement qui peut être source ou collecteur de données, et le contrôleur de communication qui est chargé en particulier de la protection contre les erreurs et introduit les éléments (caractères) de service permettant le dialogue entre les deux terminaux.

**b) Equipement de terminaison de circuit de données (ETCD) :**

C'est l'organe chargé, d'adapter le signal électrique délivré par le terminal au support de transmission (cas d'un ETCD émetteur), ou faire l'opération inverse (cas d'un ETCD récepteur). Il est usuel de grouper dans un même ensemble un ETCD émetteur et un ETCD récepteur. Cet ensemble réalise en particulier des fonctions de modulation et de démodulation c'est pourquoi il est couramment appelé MODEM. Il faut noter que ce terme est impropre puisque les ETCD réalisent aussi des fonctions de codage, de décodage, la jonction avec le terminal, etc. Ils peuvent même ne réaliser à proprement parler aucune fonction de modulation ou de démodulation, comme le cas des transmissions en bande de base.

B/ Developpement d'un algorithme de démodulation DPSK :

III) Modulation PSK et DPSK :

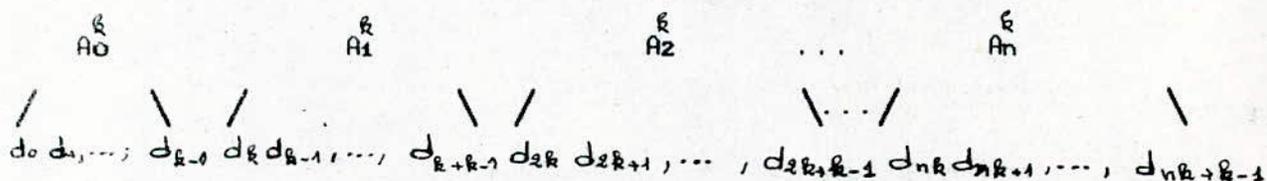
III-1) Modulation PSK :

Considérons une suite de données binaires synchrone  
 serie :  $\{d_n\}$ ,  $d_n \in [0,1]$ . Ayant un débit binaire :  $D = 1/T$ .  
 Cette suite peut être représenter mathématiquement par  
 l'équation suivante :

$$d(t) = \sum_n d_n \cdot G_T(t-nT) \quad \text{(III-1)}$$

$$G_T(t) = \begin{cases} 1 & 0 \leq t \leq T \\ 0 & \text{sinon} \end{cases} \quad \text{(III-2)}$$

considérons maintenant la suite de caractères  $\{A_n^R\}$ , formée à  
 partir de la suite  $\{d_n\}$ , chaque caractère  $A_n^R$  est un groupement  
 de k bits successifs (multibits) de la suite  $\{d_n\}$ .



$$A_n^R \in \{A\}_R = \{A_0, A_1, \dots, A_{w-1}\}, \text{ avec } w = 2^k$$

$A_i$ ,  $i=0,1, \dots, w-1$ , est une combinaison de k éléments  
 binaires. D'après l'équation (I-3), les caractères  $A_n^R$  sont  
 émis à une vitesse :  $\nu_m = D/\log(w) = 1/kT$  (III-3)

Dans le cas où  $k = 1$  on a  $\nu_m = 1/T$ .

La modulation par saut de phase (PSK:Phase-Shift Keying), consiste à associer à chaque caractère  $A_n$  la phase " $\varphi_n$ " d'une onde sinusoïdale :  $S_n(t) = U \cos(\omega t + \varphi^0 + \varphi_n)$  (III-4)

Durant l'intervalle de temps :  $[nkT, (n+1)kT]$ ,  $A_n \xrightarrow{k} \varphi_n$ . La suite de sinusoïdes issues de l'opération de modulation peut être représentée mathématiquement par l'équation suivante :

$$S(t) = U \cos[\omega t + \varphi^0 + \sum_n \varphi_n G_k T(t - nkT)] \quad (III-5)$$

$\varphi_n \in \{\phi\}_w = \{\phi_0, \phi_1, \dots, \phi_{w-1}\}$ ,  $\phi_i \in [0, 2\pi]$ ,  $\phi_i = 2\pi i/w$ ,  
 $i = 0, 1, \dots, w-1$ ;  $\varphi^0$  : phase de référence du signal  $S(t)$ .

$$\text{L'application } M : \begin{matrix} \{A\}_k & \longrightarrow & \{\phi\}_w \\ A_n & \longrightarrow & \varphi_n \end{matrix}$$

definit ce qu'on appelle le code de modulation (voir: Fig.I-4 et Tab.I-1).

### III-2) Modulation DPSK :

La modulation par saut de phase différentielle (DPSK : Differential Phase-Shift keying), consiste à associer à chaque caractère  $A_n$  la différence de phase " $\Delta\varphi_n$ " entre deux ondes sinusoïdales successives,  $S_{n-1}(t) = U \cos(\omega t + \varphi^0 + \varphi_{n-1})$  et  $S_n(t) = U \cos(\omega t + \varphi^0 + \varphi_n)$ .

$$A_n \xrightarrow{k} \Delta\varphi_n = (\varphi^0 + \varphi_n) - (\varphi^0 + \varphi_{n-1}) = \varphi_n - \varphi_{n-1}$$

$$\Delta\varphi_n \in \{\Delta\phi\}_w = \{\Delta\phi_0, \Delta\phi_1, \dots, \Delta\phi_{w-1}\}$$

$$\Delta\phi_i \in [0, 2\pi], \Delta\phi_i = 2\pi i/w = \Delta\theta_i ; i = 0, 1, \dots, w-1.$$

$$\text{L'application } M : \begin{matrix} \{A\}_k & \longrightarrow & \{\Delta\phi\}_w \\ A_n & \longrightarrow & \Delta\varphi_n = \Delta\theta C_n \end{matrix}$$

definit le code de modulation (voir Fig.III-1 et Tab.III-1),

$$C_n \in \{C_n\} = \{0, 1, \dots, w-1\} \quad (C_n \text{ élément de la suite } \{C_n\}).$$

lors de l'affectation de la phase  $\varphi_n$  à la sinusoïde  $S_n(t)$  correspondante durant l'intervalle de temps :  $[nkT, (n+1)kT]$ , il faut tenir compte de la phase de la sinusoïde précédente  $S_{n-1}(t)$ , de façon à ce que,  $\Delta\varphi_n = \varphi_n - \varphi_{n-1} = \Delta\theta C_n$ , correspondra au code de modulation choisit, ceci est possible en définissant une suite  $\{b_n\}$  à partir de la suite  $\{C_n\}$  :

$$\varphi_{n-1} = \Delta\theta b_{n-1}$$

$$\varphi_n = \Delta\theta b_n$$

$$\Delta\varphi_n = \Delta\theta(b_n - b_{n-1}) = \Delta\theta C_n, \text{ or la suite } \{b_n\} \text{ est définit}$$

$$\text{comme suit : } b_n = b_{n-1} + C_n \quad \text{(III-6)}$$

La suite de sinusoïde issues de l'opération de modulation peut être représenté par la relation mathématique suivante :

$$S(t) = U \cos[\omega_0 t + \varphi^0 + \Delta\theta \sum_n b_n G_k T(t - nkT)] \quad \text{(III-7)}$$

$$S(t) = U \sum_n G_k T(t - nkT) \cos(\omega_0 t + \varphi^0 + \Delta\theta b_n) = \sum_n S_n(t).$$

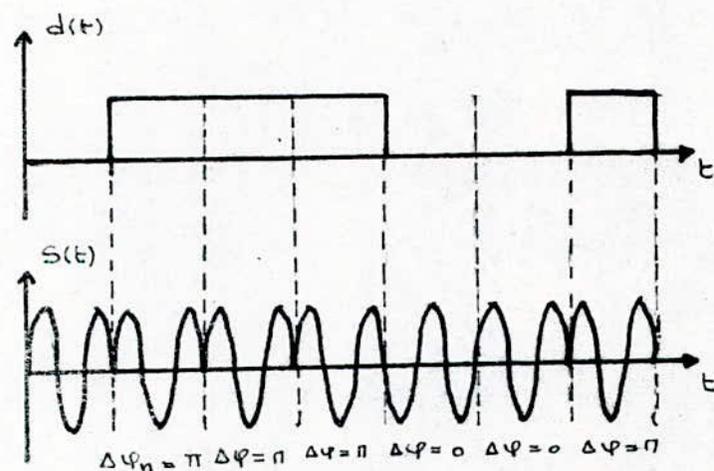


Fig. III-1: Modulation DPSK à deux niveau

$d_n$	$\Delta\varphi_n$
0	$0^\circ$
1	$180^\circ$

Tab.III-1: code de modulation

### III-3) Codes de modulation standards pour la transmission de données en DPSK

L'utilisation de la modulation DPSK pour la transmission de donnée par l'intermédiaire de MODEM, nécessite une normalisation, non seulement des débits transmis par ces modems et des supports de transmission utilisés, mais aussi des codes de modulation utilisés. Les modems conformes aux recommandations des avis :

- V26 (2400 bits/s)
- V26 bis, V26 ter (1200 bits/s, 2400 bits/s)
- V27 (4800 bits/s)
- V27 bis, V27 ter (2400 bits/s, 4800 bits/s),

du CCITT (Comité Consultatif International Télégraphique et Téléphonique), utilisent la modulation DPSK. Les codes sont normalisés suivant le nombre d'états possibles que peut avoir le multibit.

#### a) Modulation à deux états :

deux variantes de codage sont normalisées, conformément aux avis V26 bis (variante B), V26 ter (variante A) du CCITT, pour un débit de 1200 bits/s.

#### b) Modulation à quatre états :

Là aussi deux variantes sont normalisées, avis V26 (variantes A et B), avis V26 bis (variante B), avis V26 ter

(variante A), U27 bis, U27 ter (variante A); pour un débit de 2400 bits/s.

dn	$\Delta\varphi_n$	
	Variante A	Variante B
0	0°	90°
1	180°	270°

Tab.III-2: codage à deux états

$A_n^2$	$\Delta\varphi_n$	
	Variante A	Variante B
0 0	0	45°
0 1	90°	135°
1 1	180°	225°
1 0	270°	315°

Tab.III-2: codage à quatre états

c) Modulation à huit états :

Un seule variante a été normalisée, selon les recommandations des avis U27, U27 bis et U27 ter du CCITT, pour un débit de 4800 bits/s.

Tab.III-4: codage à huit états

$A_n^3$	$\Delta\varphi_n$
0 0 1	0°
0 0 0	45°
0 1 0	90°
0 1 1	135°
1 1 1	180°
1 1 0	225°
1 0 0	270°
1 0 1	315°

#### IV) Démodulation DPSK :

##### IV-1) Notion de produit scalaire :

Soient les deux signaux réels :

$$S_{n-1}(t) = U G_{kT}(t - (n-1)kT) \cos(\omega_0 t + \varphi_{n-1}) \quad (\text{IV-1})$$

$$S_n(t) = U G_{kT}(t - nkT) \cos(\omega_0 t + \varphi_n) \quad (\text{IV-2})$$

Leur produit scalaire est défini par :

$$(S_{n-1}, S_n) = \int_{nT_m}^{(n+1)T_m} S_n(t) S_{n-1}(t - T_m) dt \quad (\text{IV-3})$$

avec :  $T_m = kT = 1/\nu_m$ .

Ce produit scalaire est lié à la norme :

$$\begin{aligned} \| S_n \|^2 &= (S_n, S_n) = \int_{nT_m}^{(n+1)T_m} S_n(t) dt \\ \| S_n \|^2 &= \left[ \int_{nT_m}^{(n+1)T_m} S_n(t) dt \right]^2 \end{aligned} \quad (\text{IV-4})$$

Par analogie au produit scalaire de deux vecteurs, on a :

$$(S_{n-1}, S_n) = \| S_{n-1} \| \| S_n \| \cos(\Delta\varphi_n) \quad (\text{IV-5})$$

$$\text{d'où on a : } \cos(\Delta\varphi_n) = (S_{n-1}, S_n) / \| S_{n-1} \| \| S_n \| \quad (\text{IV-6})$$

$$\text{Si : } \check{S}_n(t) = U G_{kT}(t - nkT) \cos(\omega_0 t + \varphi_n + \beta) \quad (\text{IV-7})$$

$$\text{on aura : } \cos(\Delta\varphi_n + \beta) = (S_{n-1}, \check{S}_n) / \| S_{n-1} \| \| \check{S}_n \| \quad (\text{IV-8})$$

$$\begin{aligned} \text{Rmq(1) : Dans notre cas on a : } & \| S_{n-1} \| = \| S_n \| = \| \check{S}_n \| \\ & = \| S \| \end{aligned}$$

évidemment  $\| S \| > 0$ .

##### IV-2) Développement d'un algorithme de démodulation DPSK :

$d_n$	$\Delta\varphi_n$	$\cos \Delta\varphi_n$
0	0	+1
1	180°	-1

Tab. IV-1 :

$d_n$	$\Delta\varphi_n$	$\cos(\Delta\varphi_n - \frac{\pi}{2})$
0	90°	+1
1	270°	-1

Tab. IV-2 :

$A_n^2$	$\Delta\varphi_n$	$\cos(\Delta\varphi_n - \frac{\pi}{4})$	$\cos(\Delta\varphi_n + \frac{\pi}{4})$
0 0	$0^\circ$	$+\frac{\sqrt{2}}{2}$	$+\frac{\sqrt{2}}{2}$
0 1	$90^\circ$	$+\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
1 1	$180^\circ$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
1 0	$270^\circ$	$-\frac{\sqrt{2}}{2}$	$+\frac{\sqrt{2}}{2}$

Tab. IV-3:

$A_n^2$	$\Delta\varphi_n$	$\cos(\Delta\varphi_n - \frac{\pi}{2})$	$\cos \Delta\varphi_n$
0 0	$45^\circ$	$+\frac{\sqrt{2}}{2}$	$+\frac{\sqrt{2}}{2}$
0 1	$135^\circ$	$+\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
1 1	$225^\circ$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
1 0	$315^\circ$	$-\frac{\sqrt{2}}{2}$	$+\frac{\sqrt{2}}{2}$

Tab. IV-4:

L'opération de démodulation consiste à extraire les données numériques à partir des signaux analogique reçus, c'est à dire, arriver à déterminer la valeur de  $\Delta\varphi_n$ , connaissant les signaux  $S_{n-1}(t)$ ,  $S_n(t)$ .

Mais un examen des tables : IV-1, IV-2, IV-3 et IV-4, montre qu'il n'est pas nécessaire de déterminer  $\Delta\varphi_n$ , mais il suffit de connaître le signe de :  $\cos(\gamma)$ , pour estimer le bit correspondant,  $\gamma = \Delta\varphi_n + \beta$ , la valeur de  $\beta$  dépend du code de

modulation choisit. Il est à noter que d'après les relations: (IV-6), (IV-8) et de la remarque(1) on a :

$$- \text{sign} [\cos(\Delta\varphi_n)] = \text{sign}(S_{n-1}, S_n) \quad (\text{IV-9})$$

$$- \text{sign} [\sin(\Delta\varphi_n)] = \text{sign} [\cos(\Delta\varphi_n - \pi/2)] = \text{sign}(S_{n-1}, \dot{S}_n) \quad (\text{IV-10})$$

$$- \text{sign} [\cos(\Delta\varphi_n + \beta)] = \text{sign} [\cos\Delta\varphi_n \cos\beta - \sin\Delta\varphi_n \sin\beta] \quad (\text{IV-11})$$

$$\text{avec } \dot{S}_n(t) = U \cos(\omega_0 + \Delta\varphi_n - \pi/2) \quad (\text{IV-12})$$

Parmi les méthodes de démodulation les plus connues, on distingue :

a) Méthode d'auto-corrélation :

1) Codage à deux niveaux :

- Variante A (Tab. IV-1) :

$$H_n = \text{sign} [\cos(\Delta\varphi_n)] = \text{sign}(S_{n-1}, S_n)$$

$$H_n = \text{sign} \left[ \int_{nT_m}^{(n+1)T_m} S_n(t) S_{n-1}(t) dt \right] \quad (\text{IV-13})$$

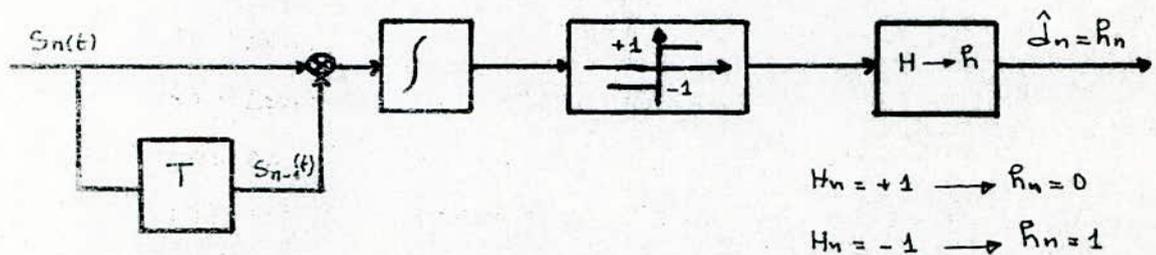


Fig. IV-1: Structure d'un démodulateur DPSK auto-corrélatif à deux états de codage (variante A)

- Variante B (Tab. IV-2) :

$$H_n = \text{sign} [\cos(\Delta\varphi_n - \pi/2)]$$

$$H_n = \text{sign}(S_{n-1}, \dot{S}_n) \quad (\text{IV-14})$$

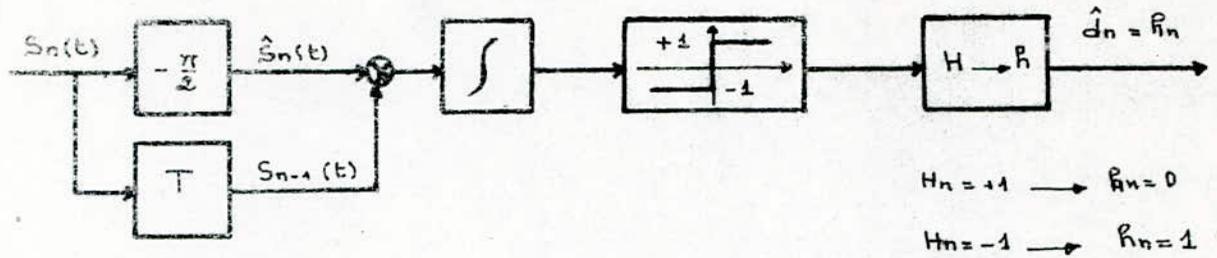


Fig. IV-2: Structure d'un démodulateur DPSK auto-corrélatif à deux états de codage (variante B)

2) codage à quatre niveaux :

- Variante A (Tab. IV-3) :

$$H_n^1 = \text{sign} [\cos(\Delta\varphi_n - \pi/4)]$$

$$H_n^2 = \text{sign} [\sqrt{2}/2 \cos(\Delta\varphi_n) + \sqrt{2}/2 \sin(\Delta\varphi_n)]$$

$$H_n^1 = \text{sign} [(S_n, S_{n-1}) + (\hat{S}_n, \hat{S}_{n-1})]$$

(IV-15)

$$H_n^3 = \text{sign} [\cos(\Delta\varphi_n + \pi/4)]$$

$$H_n^2 = \text{sign} [\sqrt{2}/2 \cos(\Delta\varphi_n) - \sqrt{2}/2 \sin(\Delta\varphi_n)]$$

$$H_n^2 = \text{sign} [(S_n, S_{n-1}) - (\hat{S}_n, \hat{S}_{n-1})]$$

(IV-16)

- Variante B (Tab. IV-3) :

$$H_n^1 = \text{sign} [\cos(\Delta\varphi_n - \pi/2)] = \text{sign} [(\hat{S}_n, S_{n-1})]$$

(IV-17)

$$H_n^2 = \text{sign} [\cos(\Delta\varphi_n)] = \text{sign} [(S_n, S_{n-1})]$$

(IV-18)

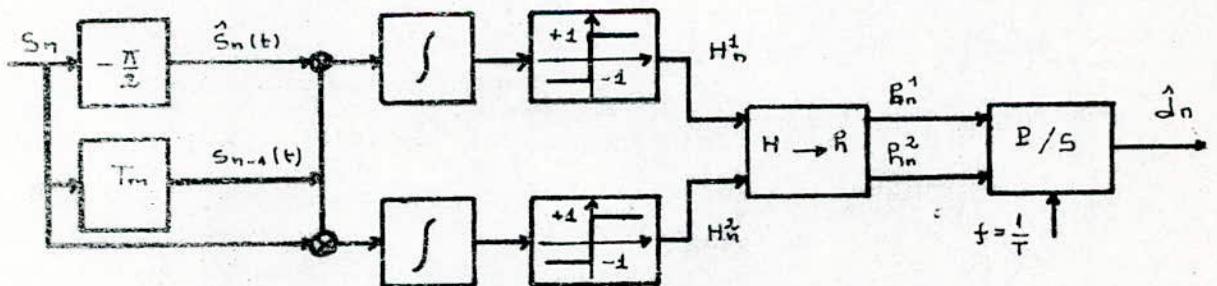


Fig. IV-3: structure d'un démodulateur DPSK auto-corrélatif à quatre états de codage (variante B).

b) Méthode de corrélation :

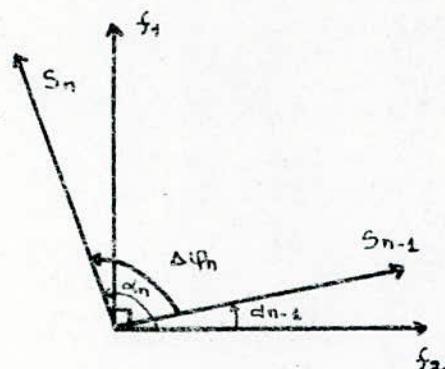


Fig.IV-4:

La méthode précédente (d'auto-corrélation), calcul directement  $\cos \gamma$ , connaissant seulement les signaux  $S_{n-1}$  et  $S_n$ . La méthode de corrélation quand à elle, prend comme référence, deux ondes sinusoïdale en quadrature de fréquence :  $\omega$  (même fréquence que celle de l'onde porteuse) :

$$f_1(t) = \cos(\omega t)$$

$$f_2(t) = \sin(\omega t)$$

$$\text{D'après la figure IV-4, on a : } \Delta \varphi_n = \alpha_n - \alpha_{n-1} \quad (\text{IV-19})$$

$$\cos(\Delta \varphi_n) = \cos(\alpha_n - \alpha_{n-1})$$

$$\cos(\Delta \varphi_n) = \cos \alpha_n \cos \alpha_{n-1} + \sin \alpha_n \sin \alpha_{n-1} \quad (\text{IV-20})$$

$$\sin(\Delta \varphi_n) = \sin(\alpha_n - \alpha_{n-1})$$

$$\sin(\Delta \varphi_n) = \sin \alpha_n \cos \alpha_{n-1} - \cos \alpha_n \sin \alpha_{n-1} \quad (\text{IV-21})$$

Dans notre on plus de la remarque (1) on a :

$$\| \| f_1 \| \| = \| \| f_2 \| \| = \| \| f \| \| .$$

D'après les relations : (IV-6) et (IV-8), on a :

$$- \cos \alpha_n = (S_n, f_2) / (\| \| S \| \| \| f \| \|)$$

$$- \cos \alpha_{n-1} = (S_{n-1}, f_2) / (\| \| S \| \| \| f \| \|)$$

$$\begin{aligned} - \sin \alpha_n &= \cos(\alpha_n - \pi/2) = (S_n, f_2) / (\| \| S \| \| \| f \| \|) \\ &= (S_n, f_2) / (\| \| S \| \| \| f \| \|) = (S_n, f_1) / (\| \| S \| \| \| f \| \|) \end{aligned}$$

$$\begin{aligned} - \sin \alpha_{n-1} &= \cos(\alpha_{n-1} - \pi/2) = (S_{n-1}, f_2) / (\| \| S \| \| \| f \| \|) \\ &= (S_{n-1}, f_2) / (\| \| S \| \| \| f \| \|) = (S_{n-1}, f_1) / (\| \| S \| \| \| f \| \|) \end{aligned}$$

En remplaçant dans les équations (IV-20) et (IV-21) on obtient les expressions suivantes :

$$- \cos \Delta \varphi_n = [(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2)] / (|| S || || f ||) \quad \dots \quad (IV-22)$$

$$- \sin \Delta \varphi_n = [(S_n, f_1)(S_{n-1}, f_2) - (S_n, f_2)(S_{n-1}, f_1)] / (|| S || || f ||) \quad \dots \quad (IV-23)$$

1) Codage à deux états :

- Variante A (Tab. IV-1) :

$$H_n = \text{sign} [\cos(\Delta \varphi_n)]$$

$$H_n = \text{sign} [(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2)] \quad (IV-24)$$

- Variante B (Tab. IV-2) :

$$H_n = \text{sign} [\cos(\Delta \varphi_n - \pi/2)] = \text{sign} [\sin(\Delta \varphi_n)]$$

$$H_n = \text{sign} [(S_n, f_1)(S_{n-1}, f_2) - (S_n, f_2)(S_{n-1}, f_1)] \quad (IV-25)$$

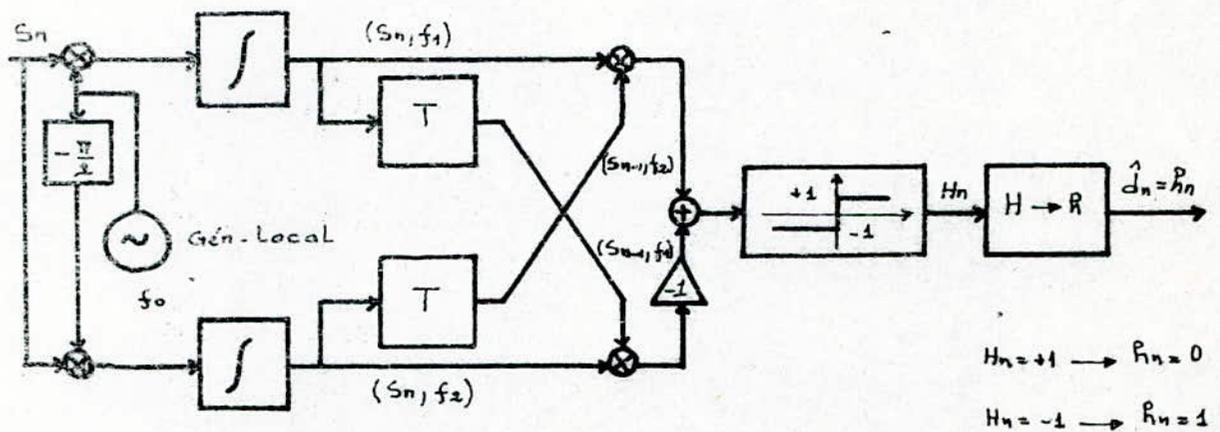


Fig. IV-5: Structure d'un démodulateur corrélatif à deux états de codage (variante B)

2) Codage à quatre états :

- Variante A (Tab. IV-3) :

$$H_n^1 = \text{sign} [\cos(\Delta \varphi_n - \pi/4)] = \text{sign} [\cos(\Delta \varphi_n) + (\sin \Delta \varphi_n)]$$

$$H_n^1 = \text{sign} [(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2) + (S_n, f_1)(S_{n-1}, f_2) - (S_n, f_2)(S_{n-1}, f_1)] \quad (\text{IV-26})$$

$$H_n^2 = \text{sign} [\cos(\Delta\varphi_n + \pi/4)] = \text{sign} [\cos(\Delta\varphi_n) - \sin(\Delta\varphi_n)]$$

$$H_n^3 = \text{sign} [(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2) - (S_n, f_1)(S_{n-1}, f_2) + (S_n, f_2)(S_{n-1}, f_1)] \quad (\text{IV-27})$$

- Variante B (Tab. IV-4) :

$$H_n^4 = \text{sign} [\cos(\Delta\varphi_n - \pi/2)] = \text{sign} [\sin(\Delta\varphi_n)]$$

$$H_n^5 = \text{sign} [(S_n, f_1)(S_{n-1}, f_2) - (S_n, f_2)(S_{n-1}, f_1)] \quad (\text{IV-28})$$

$$H_n^6 = \text{sign} [\cos(\Delta\varphi_n)]$$

$$H_n^7 = \text{sign} [(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2)] \quad (\text{IV-29})$$

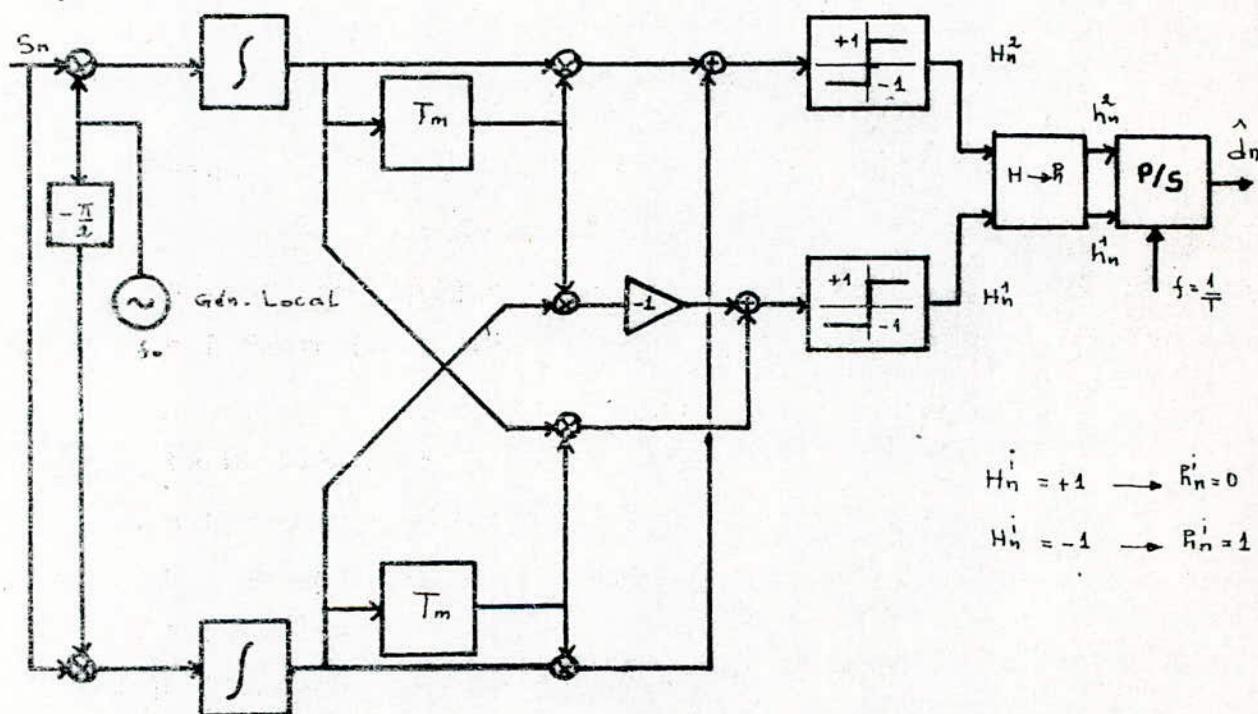


Fig. IV-6: Structure d'un démodulateur corrélatif à quatre états de codage (variante B)

c) Méthode synchrone (cohérente) :

La méthode synchrone consiste à introduire un certain nombre d'ondes de référence, qui varie en fonction du code de modulation utilisé (Fig.IV-7).

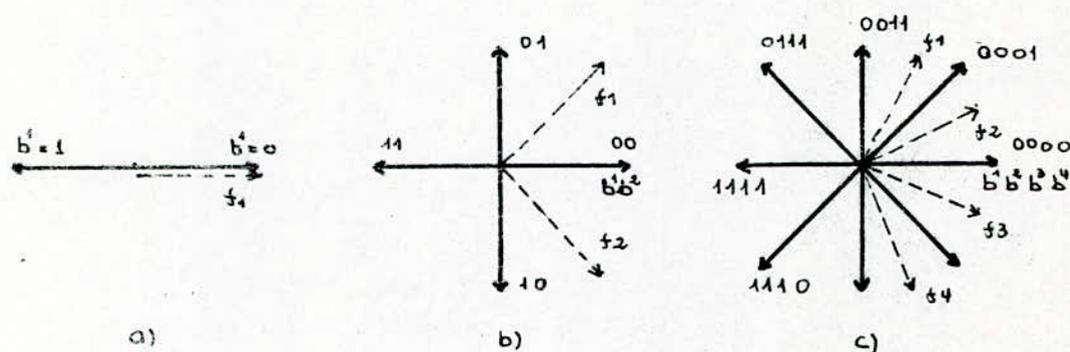


Fig.IV-7: Emplacement des ondes de référence -

- a) Modulation à deux états (variante A)
- b) Modulation à quatre états (variante A)
- c) Modulation à quatre états (variante B)

Cette méthode consiste à déterminer le signe de : " $\cos(\alpha_n, i)$ ",  
 $\alpha_n, i$  : déphasage entre le signal  $S_n$  et l'onde  $f_i$ .

$B_n^i = \text{sign} [\cos(\alpha_n, i)] = \text{sign}(S_n, f_i)$ , si " $b_n^i$ " est le symbole binaire correspondant à  $B_n^i$ , on a :

$$B_n^i = +1 \longrightarrow b_n^i = 0$$

$$B_n^i = -1 \longrightarrow b_n^i = 1$$

Or, au signal : " $S_n$ ", correspondra un ensemble de symboles

binaires :  $\{b_n^1, \dots, b_n^p\}$  et au signal : " $S_{n-1}$ ", l'ensemble :

$\{b_{n-1}^1, \dots, b_{n-1}^p\}$ ,  $p$  étant le nombre d'onde de référence.

Chaque bit du caractère :  $A_n$ , par une opération de décodage des symboles :  $b_n^i$ . A partir des tables : (IV-5) et (IV-6), dans

le cas d'un codage à deux états on a :

$$- d_n = b_{n-1}^1 \oplus b_n^1 \quad (0^\circ, 180^\circ)$$

$$- d_n = b_{n-1}^2 \oplus b_n^1 \quad (90^\circ, 270^\circ).$$

$d_n$	$\Delta\varphi_n$	$b_{n-1}^1$	$b_n^1$
0	$0^\circ$	0 1	0 1
1	$180^\circ$	0 1	1 0

Tab. IV-5:

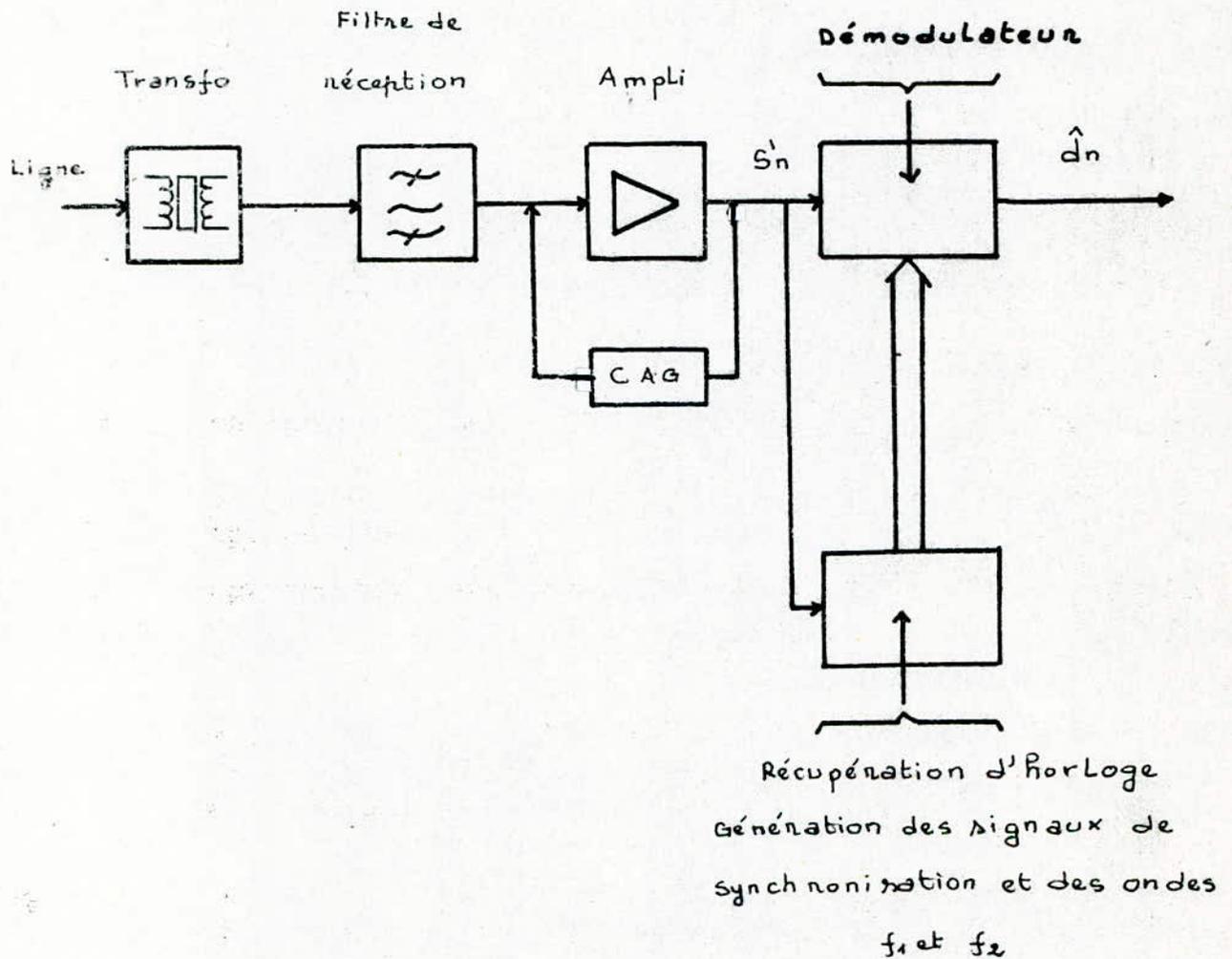
$d_n$	$d_n^1$	$d_n^2$	$\Delta\varphi_n$	$b_{n-1}^1$	$b_{n-1}^2$	$b_n^1$	$b_n^2$
0	0	1	$90^\circ$	0	0	0	1
				0	1	1	1
				1	1	1	0
				1	0	0	0
1	1	0	$270^\circ$	0	0	1	0
				0	1	0	0
				1	1	0	1
				1	0	1	1
	0	0	$0^\circ$	0	0	0	0
				0	1	0	1
				1	1	1	1
				1	0	1	0
	1	1	$180^\circ$	0	0	1	1
				0	1	1	0
				1	1	0	0
				1	0	0	1

Tab. IV-6:

✓ Réalisation pratique:

V) Description générale :

V-1) Synoptique générale d'un ETCD récepteur :



### V-2) Fiche technique :

- Démodulateur DPSK conçu autour d'un microprocesseur (Z80 A), c'est un système qui fonctionne en temps réel et en mode interruptible.
- Démodulation par la méthode de corrélation.
- Utilisation des codes de modulation à deux et à quatre états variantes : A et B.
- Vitesse de modulation : 1200 bauds.
- Débit binaire : 1200 bits/s et 2400bits/s.
- les ondes de références  $f_1$  et  $f_2$  sont des ondes carrée en quadrature ayant une fréquence de : 1800 Hz (même fréquence que celle de l'onde porteuse).

### V-3) Le microprocesseur Z80 :

Le Z80 est un microprocesseur 8bits, conditionné dans un boîtier de 40 broches, ayant une fréquence d'horloge de 2.5 MHz. Mais dans la famille Z80, il existe deux autres produits plus rapides : le Z80 A (4 MHz) et le Z80 B (6 MHz).

Le bus du Z80 comprend 16 lignes d'adresses, il dispose également de 8 lignes de donnée et de 13 lignes de commandes, des signaux d'alimentation électrique et d'une entrée d'horloge.

Du fait que le système fonctionne en mode interruptible, il est important de comprendre comment le Z80 gère les demandes d'interruption. Il ya deux d'entrée

d'interruption sur le Z80 :  $\overline{\text{NMI}}$  (interruption non masquable) et  $\overline{\text{INT}}$  (interruption masquable). Dans notre cas on ne s'intéresse qu'à cette dernière.

Supposant que les interruptions ont été autorisées par l'instruction EI (enable interrupt) et qu'une demande d'interruption est formulée en mettant à l'état bas l'entrée INT. Alors le Z80 peut traiter cette demande suivant 3 modes (mode0, mode1, mode2); pour programmer ces trois modes, il ya trois instructions distinctes : IM0, IM1, et IM2.

En ce qui nous concerne, les demandes d'interruption sont traitées en mode 2. Ce mode est très puissant et simple à utiliser; il a la structure générale suivante :

Quand le Z80 reconnaît la demande d'interruption par l'activation simultanée de  $\overline{\text{MI}}$  et  $\overline{\text{IORQ}}$ , un circuit introduit un octet de données dans le bus de données (voir Fig.V-3). Le Z80 prend l'octet de données et forme un mot de 16 bits avec un registre interne de 8 bits. Ce registre est le registre I. Ce mot de 16 bits est stocké dans le PC. Les données situées à la nouvelle de 16 bits constituent une autre adresse. Cette nouvelle adresse est l'adresse mémoire absolue de la routine de gestion d'interruption(voir Fig.V-4).

#### V-4) Synoptique général de la carte démodulateur :

Le rôle du démodulateur est d'extraire les données numériques(binaires) à partir de signaux analogiques. Sachant qu'il est nécessaire que la réception de données soit synchrone à leurs émission, on remarque que l'opération de démodulation est piloté par le bloc :synchronisation (4). Ce bloc a pour tâches :

- Déblencher l'opération de démodulation, par l'intermédiaire

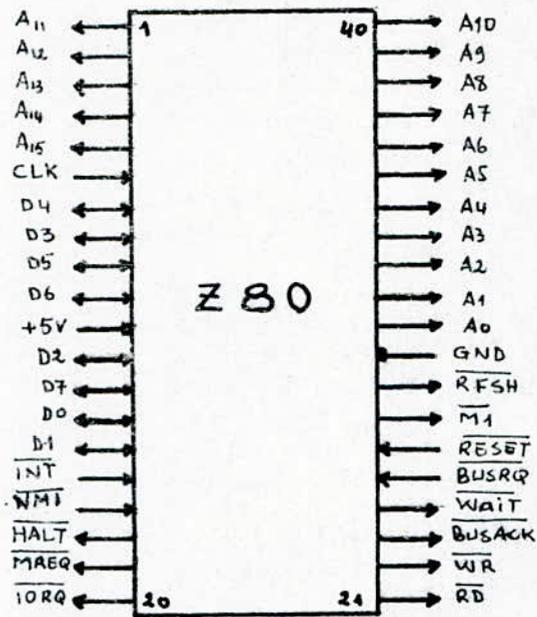


Fig.U-2: Brochage du microprocesseur Z80

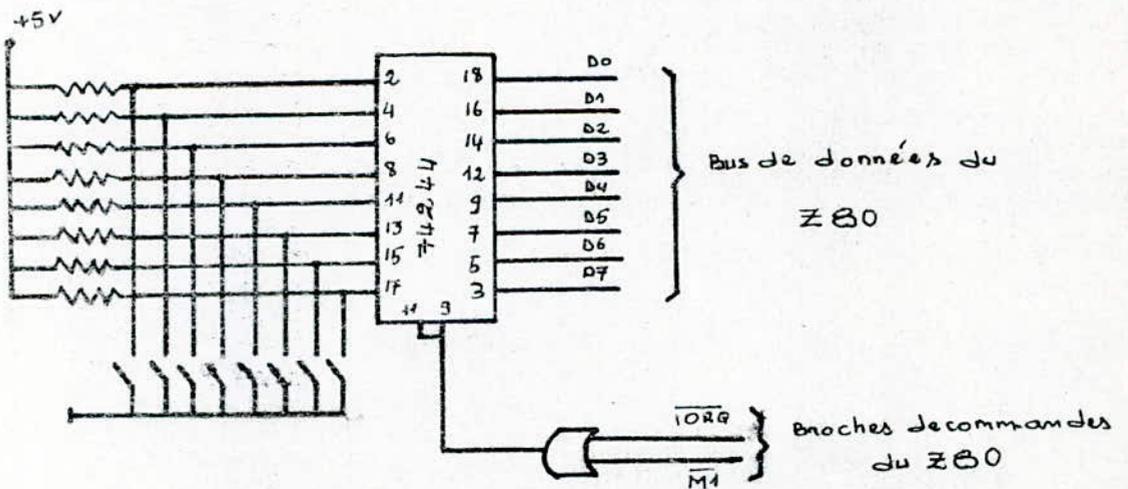
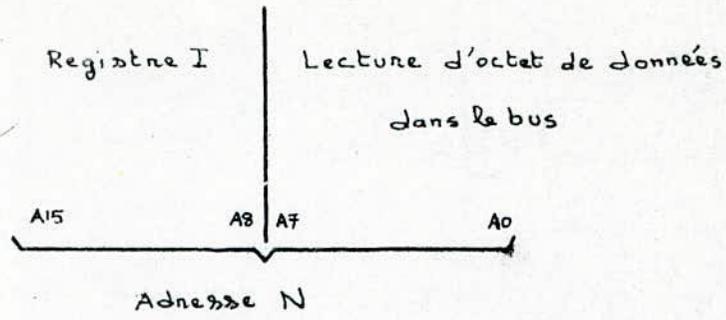


Fig.U-3: circuit d'introduction de l'octet de données



Adresse N → data (A7, .., A0) } les deux octets forment  
 Adresse N+1 → data (A15, .., A8) } une adresse qui est  
 l'adresse absolue de la routine de service d'interruption

Fig.U-4: Schéma montrant comment est formé l'adresse de pointeur à 16 bits dans un plan d'interruption en mode 2

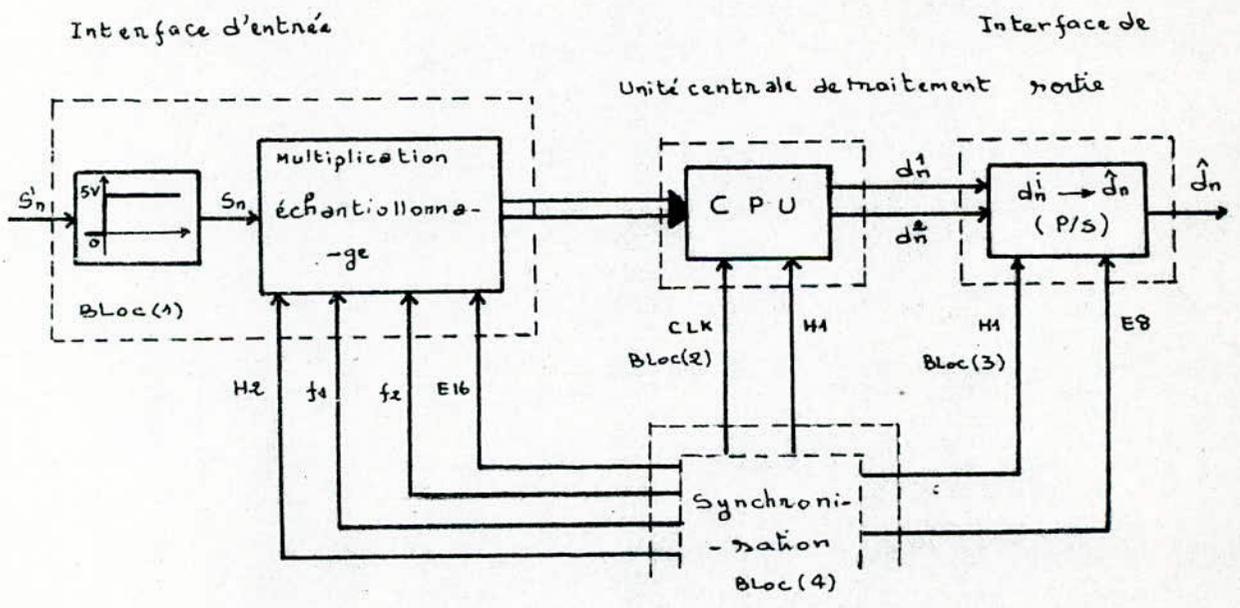


Fig.U-5: Schéma-blocs de la carte démodulateur

du signal H1, chaque fois qu'une séquence significative de l'onde porteuse arrive.

- Fournir les ondes de référence  $f_1$  et  $f_2$ .
- Fournir le signal d'horloge (CLK) pour le Z80 MPU.
- fournir des signaux nécessaires au fonctionnement de registres qui font partis des blocs (1) et (3).

2. L'interface d'entrée (1), a pour rôles :

- Adapter le signal analogique (l'onde porteuse) à l'organe de traitement principal. Cette adaptation se fait grâce à un redresseur-écrêteur (comparateur), qui transforme l'onde sinusoïdale en onde carrée (compatible TTL). Ce qui est tout à fait légitime, du fait que l'algorithme de démodulation reste valable quelque soit le signal modulé du moment qu'il est périodique; en plus l'information que nous cherchons (la différence de phase entre deux signaux successifs) n'est pas affectée par cette opération.

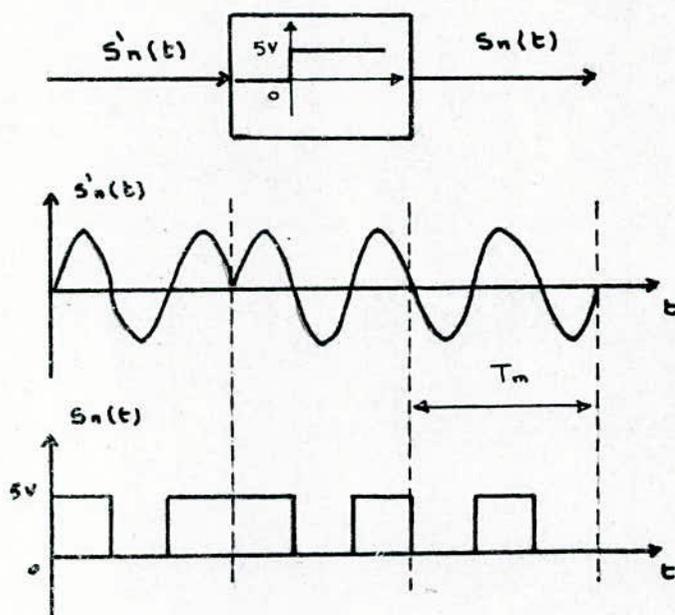


Fig.V-6: Adaptation du signal d'entrée

- En plus le bloc (1), effectue quelques opérations de traitement (multiplication, échantillonnage), qui seront explicitées plus loin.

3. L'organe de traitement (2), qui a pour rôle de choisir la séquence de programme (sous programme) correspondant au codage utilisé et l'exécuter.

4. L'interface de sortie (3), sert à réguler le débit de données à la vitesse correspondante.

## VI) Description détaillée de la réalisation :

### VI-1) Unité centrale de traitement :

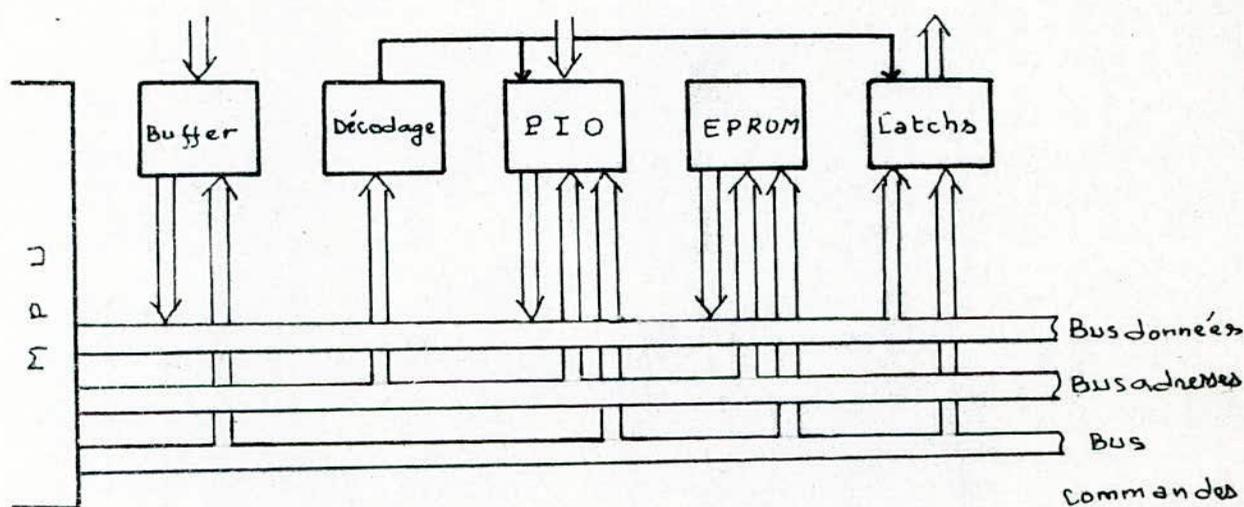


Fig.VI-1: Schéma blocs du CPU

ce système à microprocesseur est constitué d'éléments classiques dans ce genre de réalisation : un microprocesseur (Z80 A de Zilog), une mémoire morte dont la capacité est de 2 Koctets contenant le programme de pilotage de l'application (EPROM 2716), un organe d'entrée (PIO de Zilog), un organe de sortie (latch de 4 bits adressable 2 à 2 : 7475), un décodeur

(74139) pour permettre la sélection des différents boîtiers d'interfaces d'entrée et de sorties, et enfin un buffer 3 états (74244) pour introduire un octet de données sur le bus de données afin de compléter le vecteur d'interruption.

Comme il été introduit ce système fonctionne en mode interruptible. Une interruption hardware est provoquée à des instants significatifs par le signal H1, qui valide l'entrée  $\overline{\text{INT}}$  du Z80 MPU. Ce dernier reconnaît la demande d'interruption et répond en mode 2. Le MPU se positionne sur le sous programme qui correspond au code de modulation utilisé par l'émetteur. Les ports A et B du PIO sont programmés en entrée, ils introduisent les données à traiter par le MPU. chaque deux bits du 7475 sont adressés et considérés comme port de sortie et jouent le rôle de sous canaux du démodulateur.

#### VI-2) Schéma d'adressage :

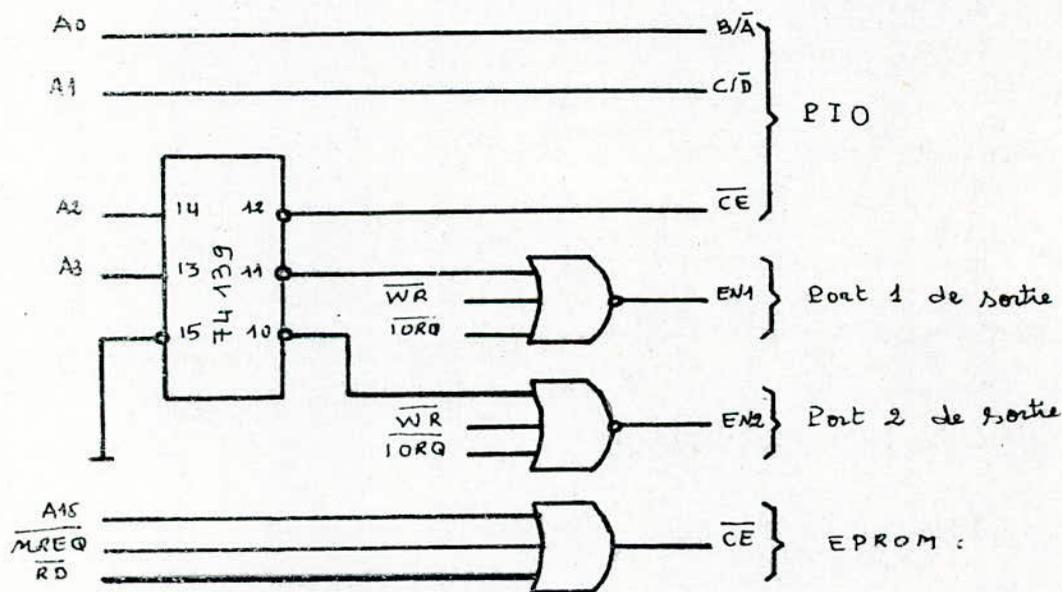


Fig. VI-2: Schéma d'adressage

Le Z80 à une structure d'entrée/sortie par instructions d'entrée/sortie, ce qui étendra le champ d'adressage du MPU. Pour ce qui est du mode d'adressage, on a utilisé un adressage par décodage pour sélectionner les boîtiers d'entrée/sortie et un adressage par sélection linéaire pour valider le boîtier de l'EPROM, les 2048 cases mémoire de l'EPROM sont sélectionnées grâce aux lignes A0, .. ,A10 du bus adresse.

### VI-3) Interface d'entrée :

Comme il a été introduit, ce bloc a pour fonctions d'adapter le signal analogique au système (ce qui a été expliqué succinctement précédemment) et de faire des opérations de traitement. Ces opérations consistent à :

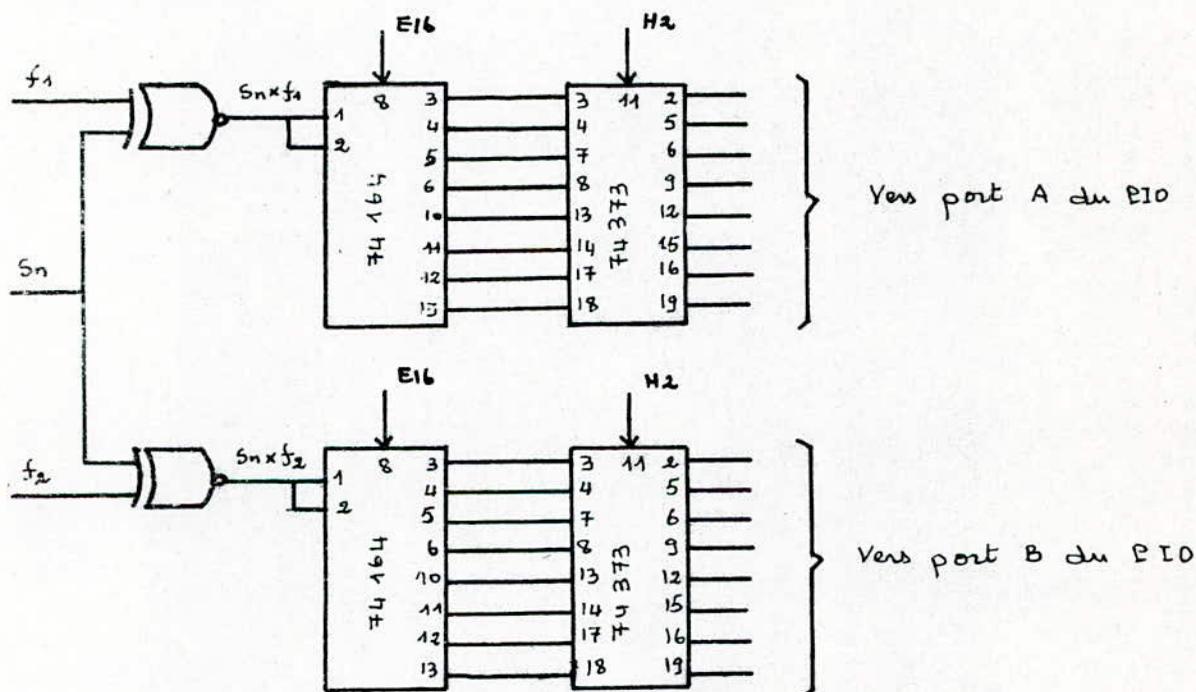


Fig.VI-3: Interface d'entrée

- Faire une multiplication analogique entre "Sn" et "f1" d'une

part, et "S<sub>n</sub>" et "f<sub>2</sub>" d'autre part, par l'intermédiaire de portes :  $\overline{XOR}$ .

- Échantillonner les signaux issus de ces deux produits par l'intermédiaire de registres à décalage (74164 : registre série/parallèle) à une fréquence  $f_0 = 16\mu m$ , cette opération est piloté par le signal E16.

- Stocker les huit échantillons centraux dans un verrou (74373), la sortie de chaque verrou est reliée à un port d'entrée du PIO. Le stockage des échantillons est effectué à des moments bien significatifs. Cette opération est déclenchée par le signal H2 qui est en quadrature avec H1.

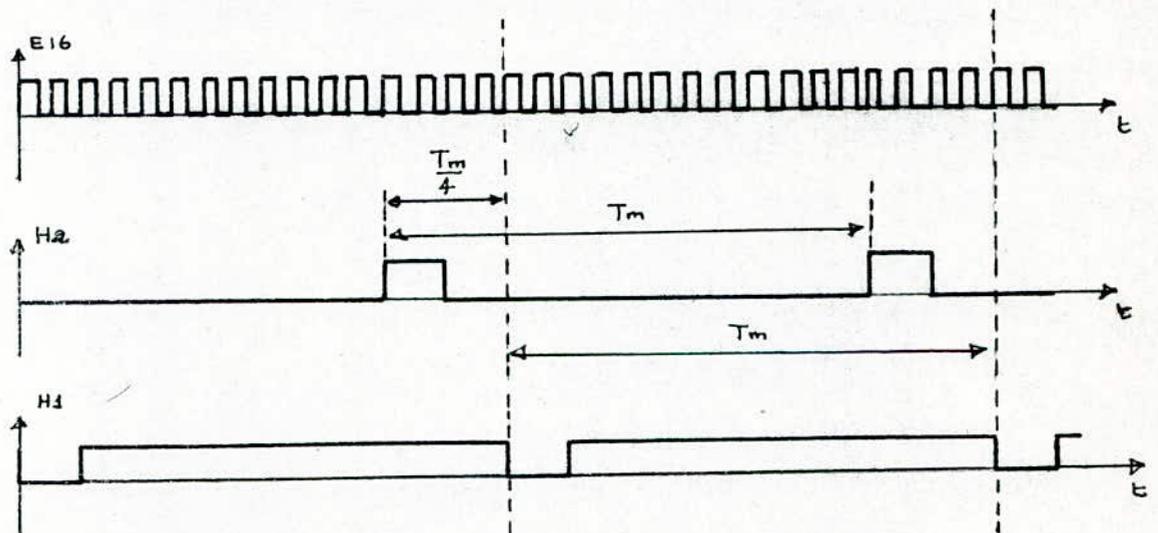


Fig.VI-4: Chronogramme des signaux de commandes de l'interface d'entrée

#### VI-4) Interface de sortie :

L'interface de sortie a pour rôle de serialiser les données qui sont sur les deux sous canaux du démodulateur (sortie du latch1 et du latch2). Ce qui a pour effet de réguler le débit binaire des données à la vitesse

correspondante. Cette opération est réalisée par un registre à décalage (74165: registre parallèle/serie) et elle est pilotée par les signaux H1 et E8.

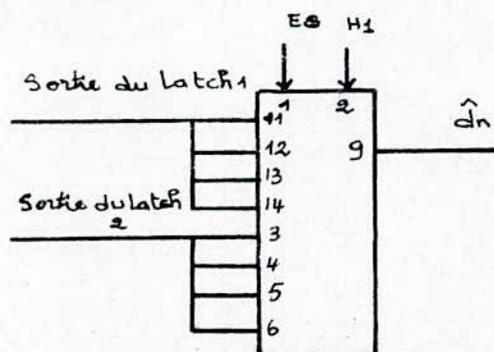


Fig.VI-5: Interface de sortie -

## VII) Programme moniteur :

### VII-1) Présentation générale :

Incontestablement, un des plus importants aspects de l'étude qui nous a été proposée est le programme de pilotage de l'application. Son écriture, d'une part, suit exhaustivement les recommandations inhérentes au comportement général désiré du démodulateur et, d'autre part, tient compte des contraintes qui lui sont imposées.

Comme préalable à tout dessin d'explication du fonctionnement choisi, il est d'usage d'en présenter les principales fonctions sous forme d'un organigramme général. Cet organigramme est présenté sur la figure: VII-1.

Une première analyse du séquencement des actions qui y figurent révèle les éléments d'explication préliminaires suivant :

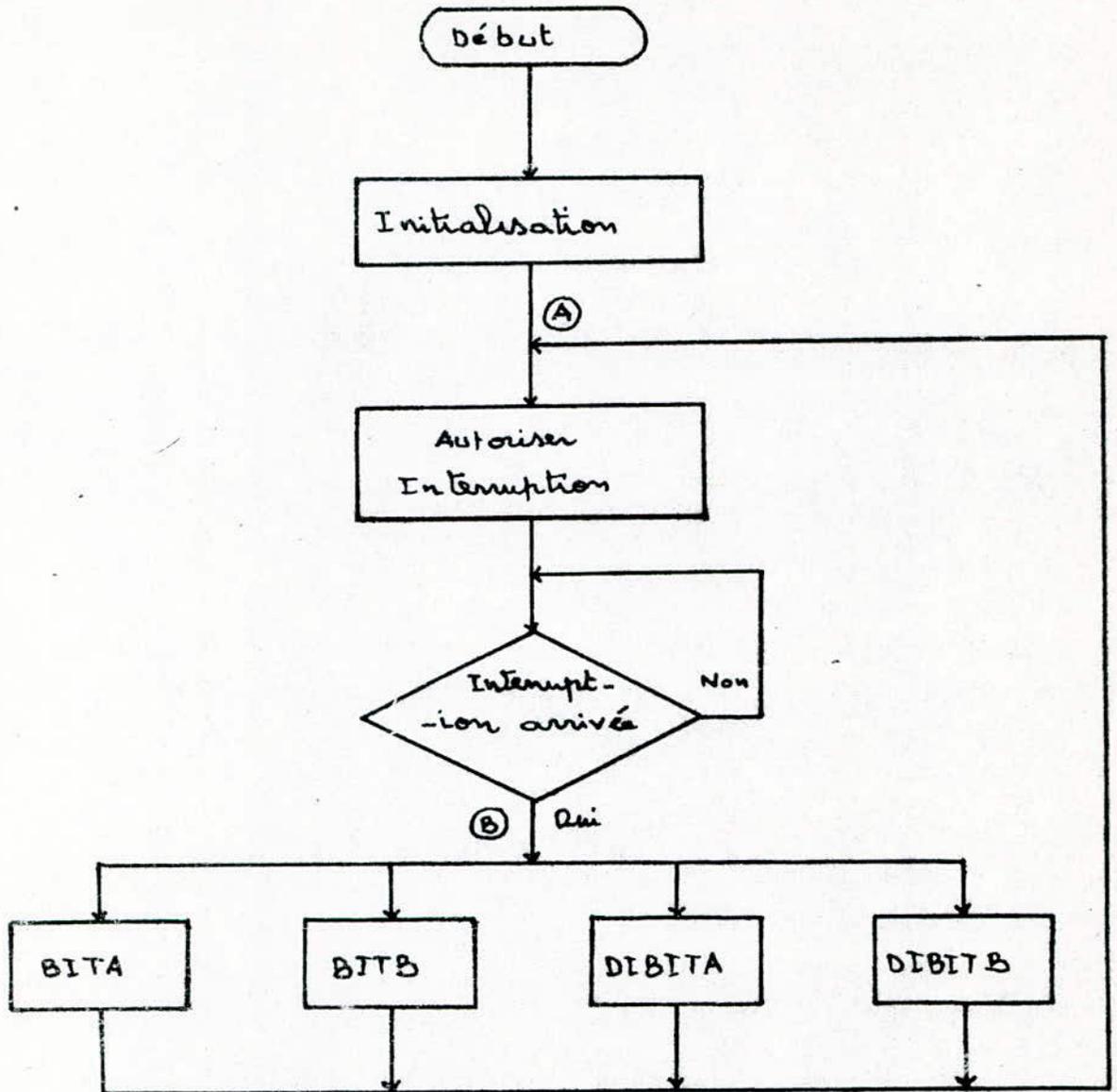


Fig.VII-1: Organigramme général

1. Le programme commence, tel qu'il pourrait être pensé intuitivement par initialiser le système en fixant les paramètres fonctionnels de sorte à offrir des conditions de démarrage identique à chaque lancement de l'application.

2. Ensuite, le moniteur autorise les interruption, et puis et par l'intermédiaire de l'instruction HALT le système est en état d'attente d'une interruption.

3. Dèsqu'une interruption arrive une des quatre routines suivante est active : BITA, BITB, DIBITA OU DIBITB.

4. Habituellement, après l'exécution de la routine d'interruption le programme est repris du point B, mais dans cette application la reprise du programme principal se fait à partir du point A, car la pile est figée dans l'EPROM, évidemment le contenu de la case mémoire adressée par le pointeur de pile est figé et correspond à l'emplacement du point A.

#### VII-2) Sous programme BITA :

Cette routine est activée dans le cas où le code de modulation utilisé, est à deux niveaux variante A (Tab.IV-1).d'après l'équation : (IV-24), cette routine calcule les produits scalaires  $(S_n, f_1)$  et  $(S_n, f_2)$ , et détermine le signe de l'expression :  $(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2)$ , les produits scalaires  $(S_{n-1}, f_1)$  et  $(S_{n-1}, f_2)$  sont issue de l'exécution précédente de la routine. Et selon la nature de ce signe la décision, d'envoyer un "0" logique ou un "1" logique sur les ports de sorties, est prise.

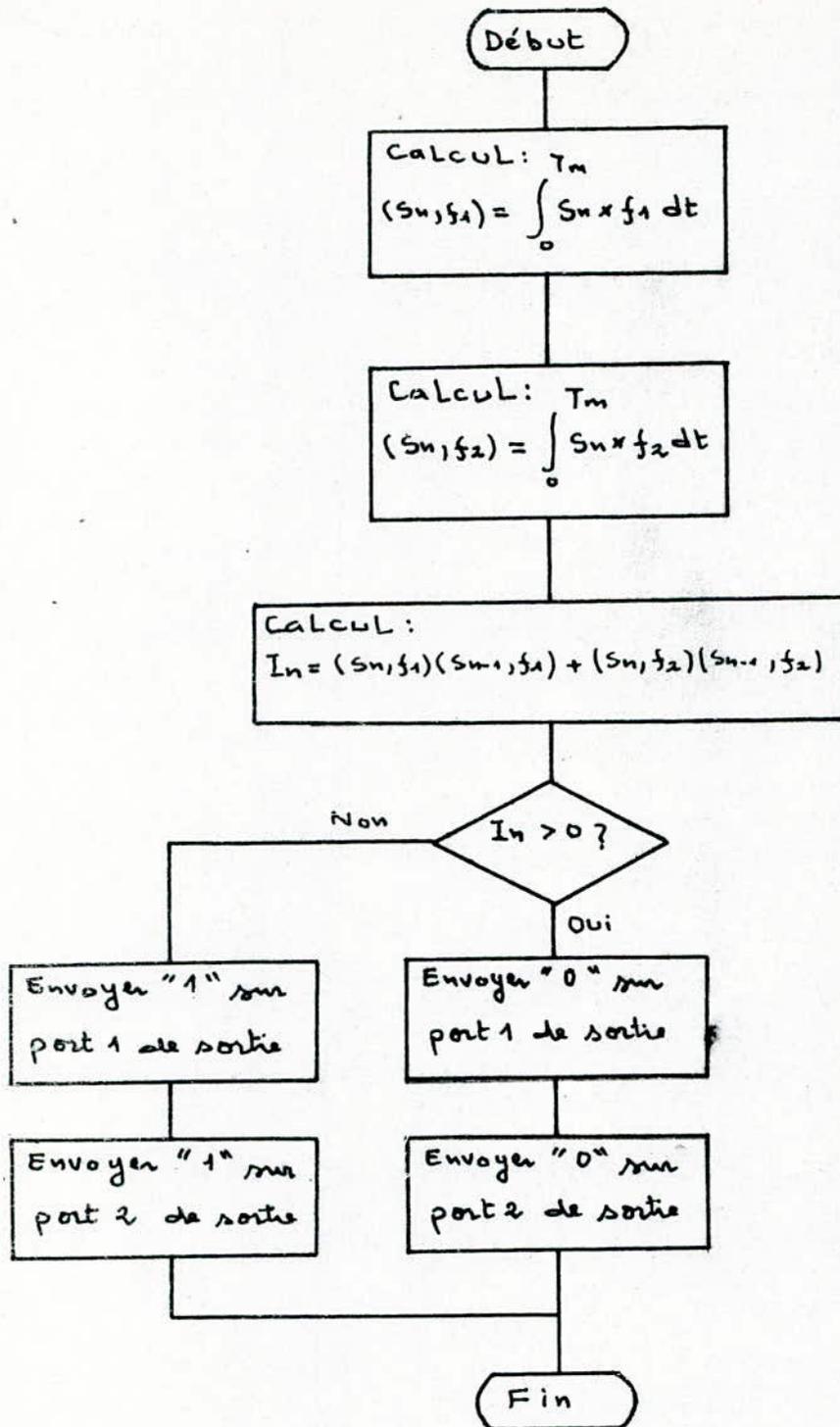


Fig. VII-2 : Organigramme du sous-programme BITA

### VII-3) Sous programme BITB :

Cette routine est activée dans le cas où le code de modulation utilisé est à deux niveau variante B (Tab.IV-2). Et d'après l'équation (IV-25), cette routine doit calculer les produits scalaires  $(S_n, f_1)$  et  $(S_n, f_2)$ , déterminer le signe de l'expression :  $(S_n, f_1)(S_{n-1}, f_2) - (S_n, f_2)(S_{n-1}, f_1)$ , et selon la nature de ce signe envoyer un "0" logique ou un "1" logique sur les ports de sorties.

### VII-4) Sous programme DIBITA :

Cette routine est activée dans le cas où le code de modulation utilisé est à quatre états variante A (Tab.IV-3). Et d'après les équations (IV-26) et (IV-27), cette routine doit calculer les produits scalaires  $(S_n, f_1)$  et  $(S_n, f_2)$ , et déterminer le signe de :  $\cos(\Delta\phi_n - \pi/4)$  et  $\cos(\Delta\phi_n + \pi/4)$ , et suivant la nature de ces signes envoyer des "0" ou des "1" sur les ports de sorties.

### VII-5) Sous programme DIBITB :

Dans le cas où le code de modulation est à quatre états variante B (Tab.IV-4), cette routine est active. Et d'après les équations (IV-28) et (IV-29), cette routine routine fait la même chose que la précédente, sauf qu'elle détermine les signes de :  $\cos(\Delta\phi_n - \pi/2)$  et  $\cos(\Delta\phi_n)$ .

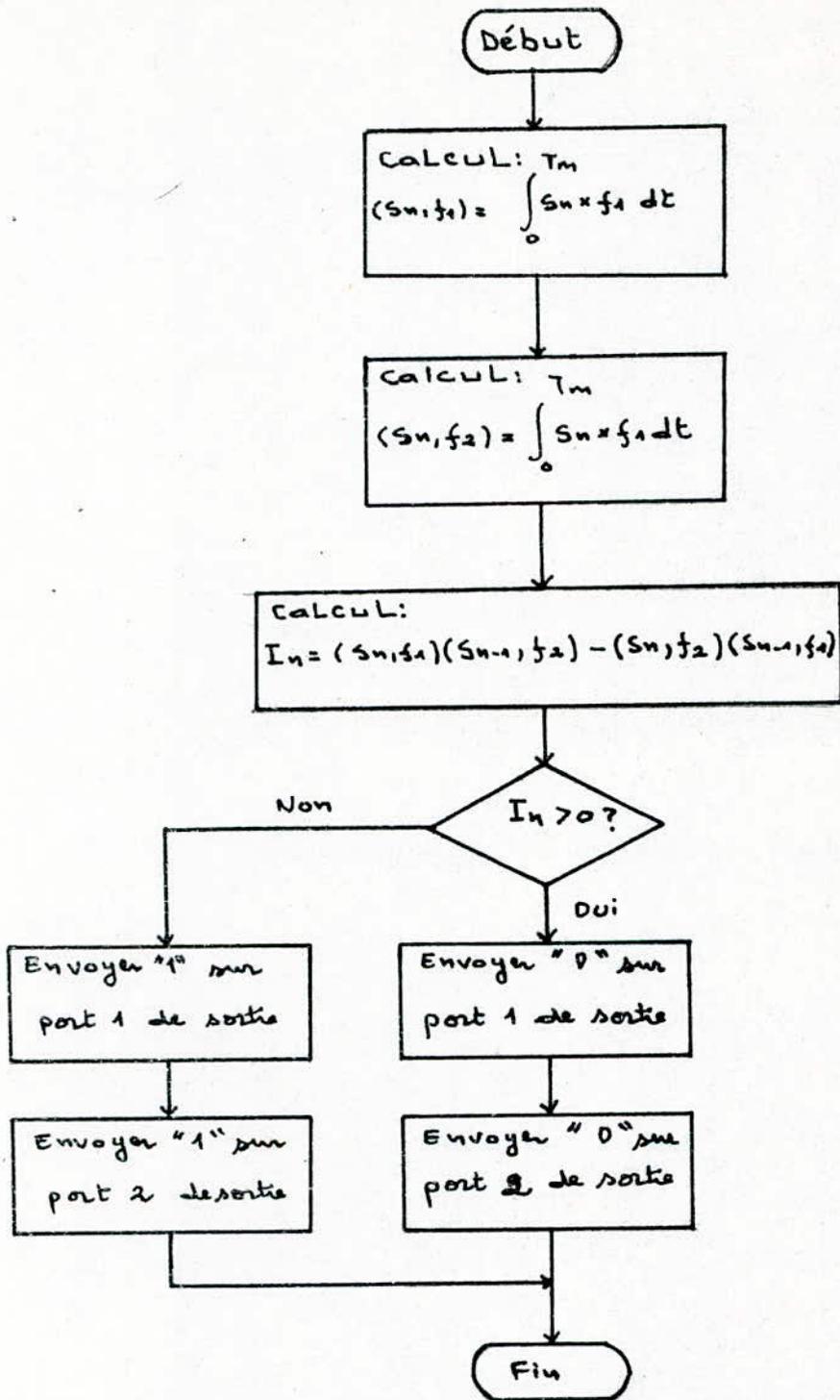


Fig. VII-3: Organigramme du sous-programme BITB

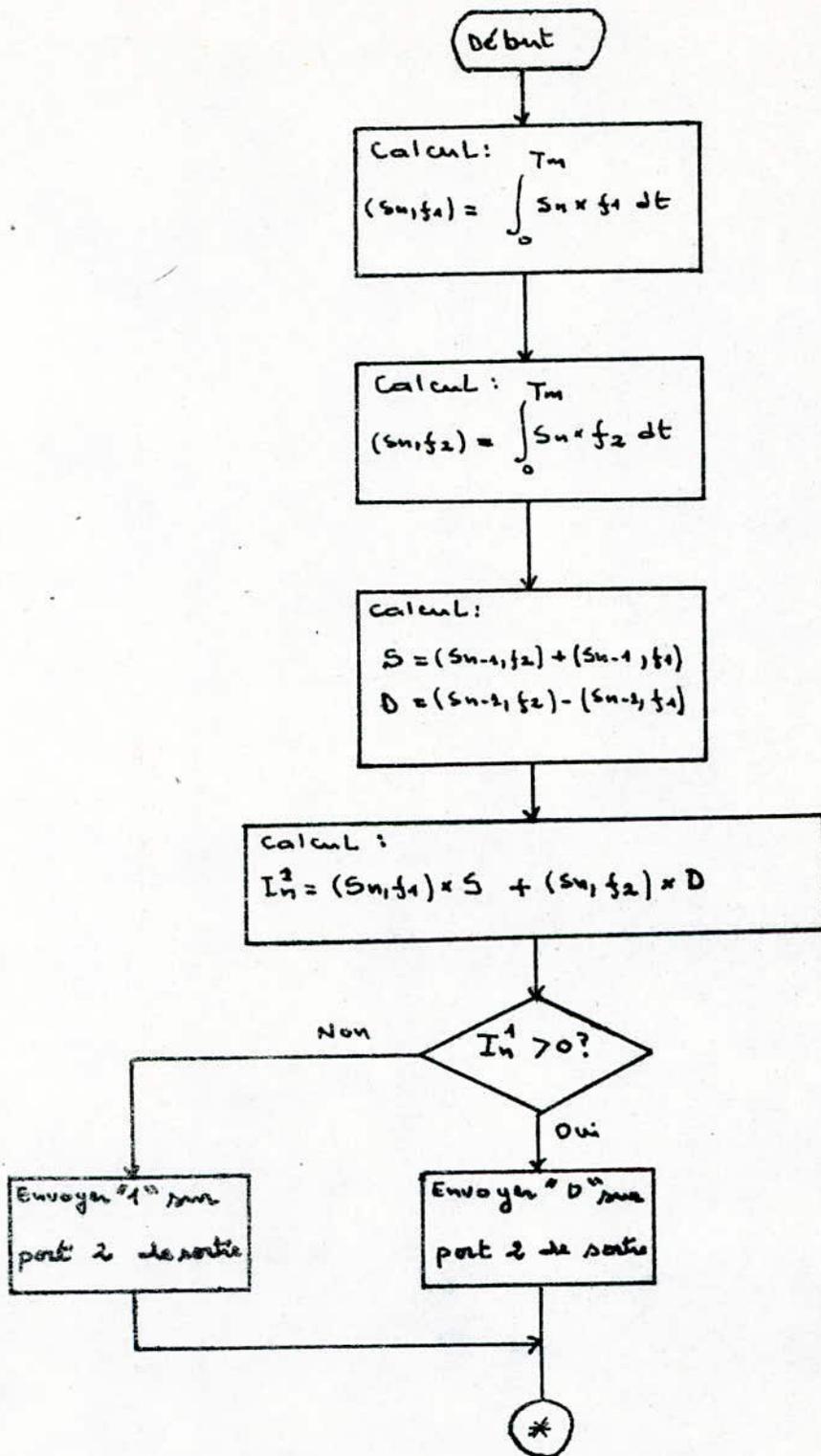
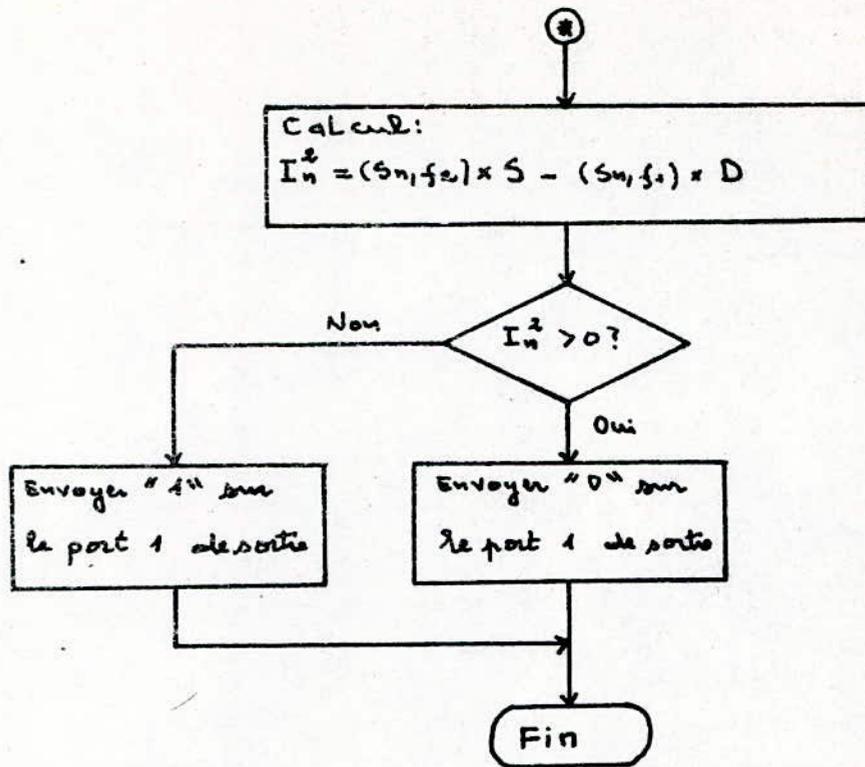


Fig. VII-4 : Organigramme du sous-programme DIBITA



Organigramme du sous-programme DIBITA (suite)

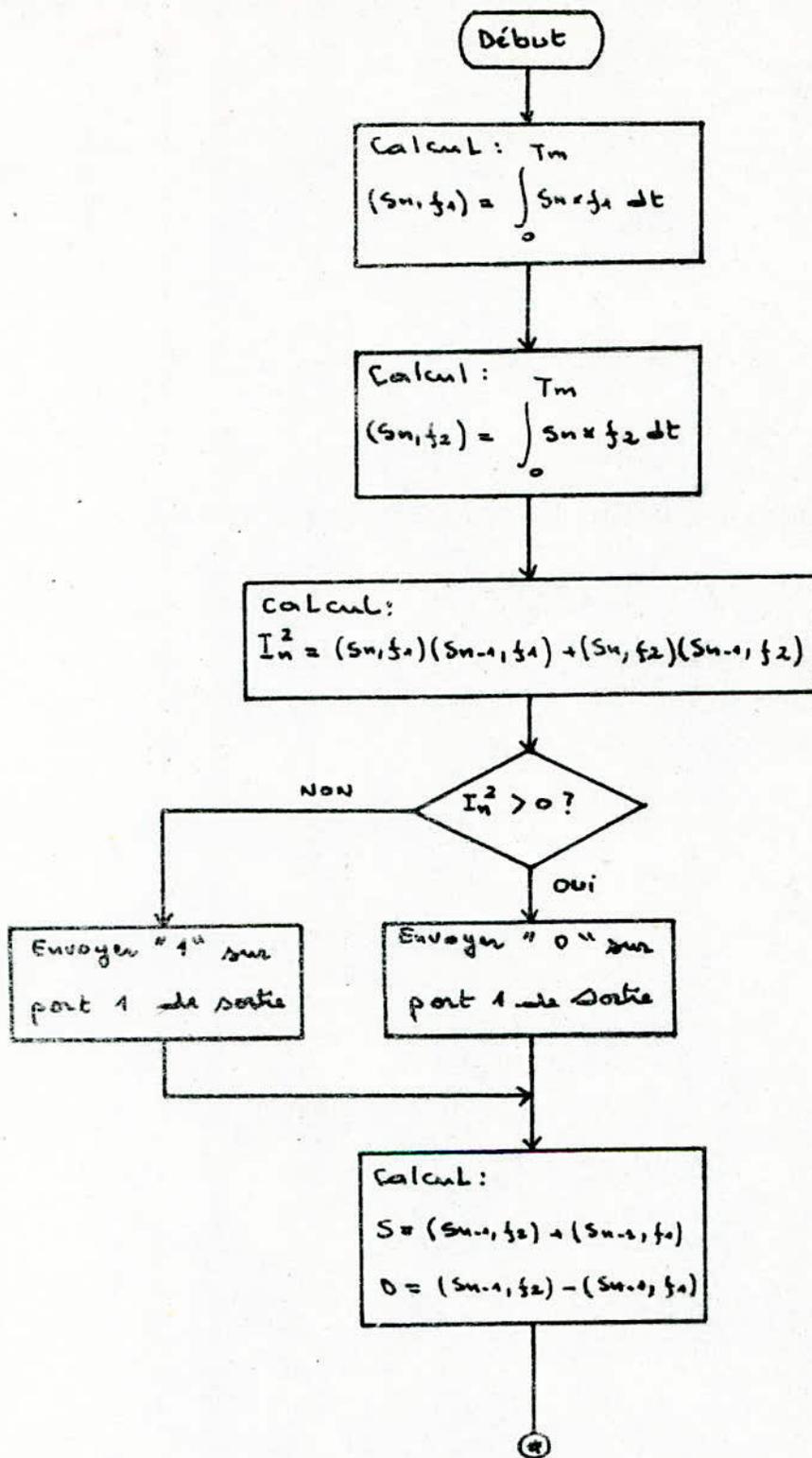
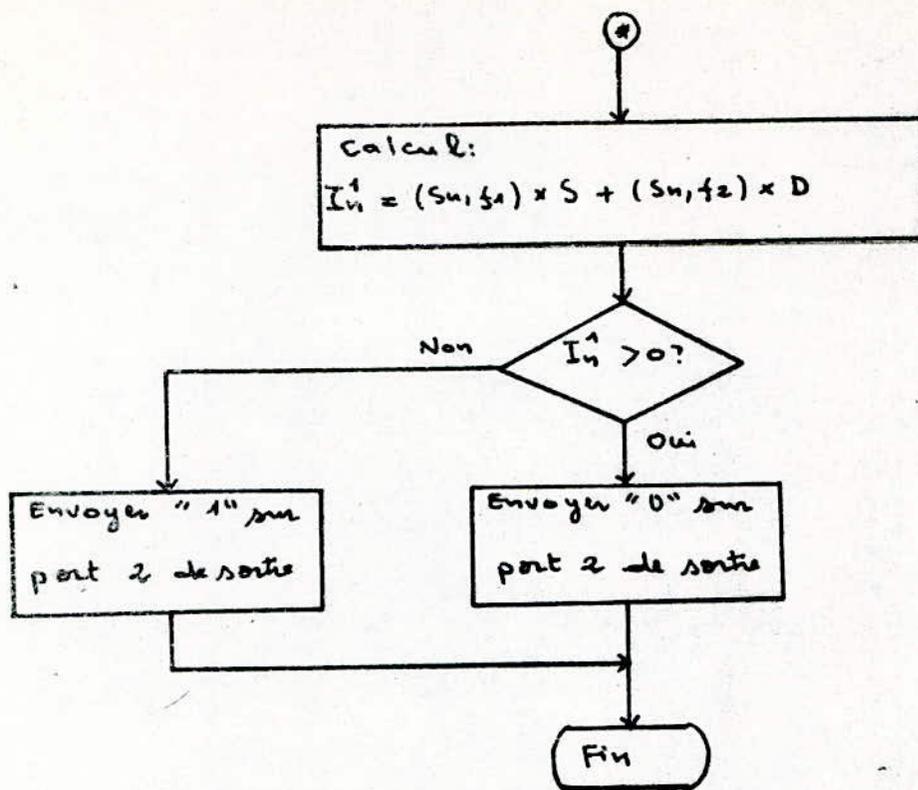


Fig. III-5: Organigramme du sous-programme DEBITB



Organigramme du sous-programme DIBITE (suite).

## Conclusion :

Sachant que la démodulation est une des fonctions essentielles d'un ETCO récepteur, la présente étude a pour but de traiter la conception et la réalisation d'un démodulateur DPSK à base d'un microprocesseur de la famille Z80.

Une brève analyse des sous programmes de démodulation nous a permis de déterminer les temps d'exécution maximaux pour chaque sous programme avec les différents microprocesseurs de la famille Z80. Un résumé est présenté dans le tableau suivant:

Routine	Nbre de cycle d'horloge ( $T_H$ )	Temps d'exécution ( $\mu s$ )			Temps d'attente (libre) ( $\mu s$ )		
		Z80	Z80 A	Z80 B	Z80	Z80 A	Z80 B
BITA	1015	406	253,75	167,5	427	579,25	665,5
BITB	1015	406	253,75	167,5	427	579,25	665,5
DIBITA	1476	590,4	369	243,6	242,6	464	589,4
DIBITB	1476	590,4	369	243,6	242,6	464	589,4

En rappelant que le point essentiel de cette étude est d'arriver à concevoir un système qui fonctionne en temps réel. Le constat qu'on établit de ces résultats est que le système, tel qu'il est conçu fonctionne en temps réel et assume les tâches qui lui sont attribuées en un temps largement inférieur à la période de modulation ( $T_m = 833 \mu s$ ). De même il permet l'interchangabilité entre les différents microprocesseur de la famille Z80 tout en respectant ce

critère.

Il est clair que le système, en plus des charges qui lui sont attribuées, peut réaliser d'autres fonctions d'un ETCD récepteur, surtout s'il est piloté par un Z80B MPU, en exploitant l'état d'attente auquel il est soumis.

Signalons qu'une recherche poussée dans ce sens peut être menée, vu qu'elle ne demande pas de gros moyens de développement, et qu'on peut aboutir à des résultats fort intéressants du point de vue performances et prix.

## Bibliographie

- C MACCI, J-F GUILBERT "Téléinformatique"  
ed DUNOD 1979
- M STEIN "Les modems pour la transmission de données"  
ed MASSON 1986
- J GORALSKI "Pomoce dydaktyczne" Zeszyt 72  
Warszawa 1986
- J-W COFFRON "Applications du Z80"  
ed SYBEX 1984
- Data book Z80 microprocessor family  
ed SGS 1982
- CCITT, red book, Vol.VIII Data Transmission  
Genewa 1985

## ANNEXE

```

;*****
;
; Ecole National Polytechnique
; Projet de fin d'étude
; Conception et réalisation d'un démodulateur DPSK à Z80
;
; Programme moniteur
;
;*****
PAD EQU 00H ;Registre données du port A du PIO
PBD EQU 01H ;Registre données du port B du PIO
PAC EQU 02H ;Rrgistre commande du port A du PIO
PBC EQU 03H ;Registre commande du port B du PIO
P1S EQU 04H ;Port 1 de sortie
P2S EQU 08H ;Port 2 de sortie
;
;*****
;Initialisation
;*****
;
LD A,4F ;Initialisation du PIO
OUT PAC,A ;Ports A et B en sortie
OUT PBC,A ;
LD SP,070F ;Initialisation pointeur pile
IM2 ;Mode 2 d'interruption
LD A,07 ;Initialisation vecteur
LD I,A ;d'interruption
EI ;Autoriser interruption
HALT ;Attendre
;
;*****
;Sous-programme BITA
;*****
;
;Calcul de (Sn,fi) :
;
IN A,PAD ;Introduire échantillons
LD B,08 ;
LD C,00 ;

```

```

L1  RLA          ;Calcul somme analogique
    JRNC L2     ;des échantillons
    INC C       ;
L2  DJNZ L1     ;
    LD A,C      ;
    SUB 04      ;
    LD D,A      ;Stocker (Sn,f1) dans le reg D
;
;Calcul de (Sn,f2) :
;
    IN A,PBD    ;Introduire échantillons
    LD B,08     ;
    LD C,00     ;
L3  RLA          ;Calcul somme analogique
    JRNC L4     ;des échantillons
    LD A,C      ;
    INC C       ;
L4  DJNZ L3     ;
    LD A,C      ;
    SUB 04      ;
    LD E,A      ;Stocker (Sn,f1) dans le reg E
;
;Démodulation :
;Calcul de : (Sn,f1)(Sn-1,f1)+(Sn,f2)(Sn-1,f2)
;(Sn-1,f1) est stocké dans le reg H
;(Sn-1,f2) est stocké dans le reg L
;
    LD A,00     ;
    LD B,H      ;
    CPB         ;
    JRZ L7      ;
    JPM L6      ;
    SUB B       ;
    LD B,A      ;
    LD A,00     ;
L5  ADD A,D     ;
    DJNZ L5     ;
    NEG        ;
    JR L7       ;
L6  ADD A,D     ;
    DJNZ L6     ;
L7  LD C,A      ;
    LD A,00     ;
    LD B,L      ;

```

```

CPB      ;
JRZ      L10 ;
JPM      L9  ;
SUB      B   ;
LD       B,A ;
LD       A,00 ;
L8 ADD    A,E ;
DJNZ    L8  ;
NEG      ;
JR       L10 ;
L9 ADD    A,E ;
DJNZ    L9  ;
L10 ADD   A,C ;

```

```

;
;Sortie des données :
;

```

```

JPP      L11 ;
LD       A,01 ;
OUT      P1S,A ;
OUT      P2S,A ;
JR       L12 ;
L11 LD    A,00 ;
OUT      P1S,A ;
OUT      P2S,A ;
L12 LD    H,D ;
LD       L,E ;
RET      ;

```

```

;
;*****
;Sous-programme BITB
;*****
;

```

```

;Calcul de (Sn, f1) :
;

```

```

IN       A,PAD ;
LD       B,08 ;
LD       C,00 ;
L1 RLA      ;
JRNC    L2  ;
INC     C   ;
L2 DJNZ   L1 ;
LD      A,C ;
SUB     04  ;
LD      D,A ;

```

;  
;Calcul de (Sn,f2) :  
;

```
      IN   PBD   ;  
      LD   B,08  ;  
      LD   C,00  ;  
L3    RLA      ;  
      JRNC L4   ;  
      INC   C    ;  
L4    DJNZ L3   ;  
      LD   A,C   ;  
      SUB  04    ;  
      LD   E,A   ;
```

;  
;Démodulation :  
;Calcul de :  $(S_n, f_1)(S_{n-1}, f_2) - (S_n, f_2)(S_{n-1}, f_1)$   
;(Sn-1,f1) est stocké dans le reg H  
;(Sn-1,f2) est stocké dans le reg L  
;

```
      LD   A,00  ;  
      LD   B,H   ;  
      CPB      ;  
      JRZ   L7   ;  
      JPM  L6   ;  
      SUB  B     ;  
      LD   B,A   ;  
      LD   A,00  ;  
L5    ADD  A,E   ;  
      DJNZ L5   ;  
      NEG      ;  
      JR   L7   ;  
L6    ADD  A,E   ;  
      DJNZ L6   ;  
L7    LD   C,A   ;  
      LD   A,00  ;  
      LD   B,L   ;  
      CPB      ;  
      JRZ   L10  ;  
      JPM  L9   ;  
      SUB  B     ;  
      LD   B,A   ;  
      LD   A,00  ;  
L8    ADD  A,D   ;  
      DJNZ L8   ;
```

```

        NEG          ;
        JR    L10    ;
L9     ADD    A,E    ;
        DJNZ  L9     ;
L10    SUB    C      ;

```

```

;
;Sortie des données :
;

```

```

        JPP    L11    ;
        LD     A,01    ;
        OUT   P1S,A   ;
        OUT   P2S,A   ;
        JR    L12    ;
L11    LD     A,00    ;
        OUT   P1S,A   ;
        OUT   P2S,A   ;
L12    LD     H,D     ;
        LD     L,E     ;
        RET                    ;

```

```

;
;*****

```

```

;Sous-programme DIBITA

```

```

;*****
;

```

```

;Calcul de (Sn,f1) :
;

```

```

        IN     A,PAD  ;
        LD     B,08   ;
        LD     C,00   ;
L1     RLA                    ;
        JRNC  L2     ;
        INC   C         ;
L2     DJNZ  L1     ;
        LD     A,C     ;
        SUB   04      ;
        LD     D,A     ;

```

```

;Calcul de (Sn,f2) :
;

```

```

        IN     A,PBD  ;
        LD     B,08   ;
        LD     C,00   ;
L3     RLA                    ;

```

```

        JRNC L4      ;
        INC C        ;
L4     DJNZ L3      ;
        LD A,C      ;
        SUB 04      ;
        LD E,A      ;
;
;Démodulation"extraction du 1er bit" :
;
;Calcul de :
;(Sn-1,f2)+(Sn-1,f2) et (Sn-1,f2)-(Sn-1,f1)
;(Sn-1,f1) est stocké dans le reg H
;(Sn-1,f2) est stocké dans le reg L
;
        LD A,H      ;
        ADD A,L     ;
        LD C,A      ;
        LD A,L      ;
        SUB H       ;
        LD L,A      ;Stocker (Sn-1,f2)-(Sn-1,f1) et
        LD H,C      ;(Sn-1,f2)+(Sn-1,f1) dans les reg
                    ;L et H
;
;Calcul de :
;(Sn,f1)[(Sn-1,f2)+(Sn-1,f1)]+(Sn,f2)[(Sn-1,f2)-(Sn-1,f1)]
;
        LD A,00     ;
        LD B,H      ;
        CPB        ;
        JRZ L7      ;
        JPM L6      ;
        SUB B       ;
        LD B,A      ;
        LD A,00     ;
L5     ADD A,D      ;
        DJNZ L5     ;
        NEG        ;
        JR L7      ;
L6     ADD A,D      ;
        DJNZ L6     ;
L7     LD C,A      ;
        LD A,00     ;
        LD B,L      ;
        CPB        ;

```

```

        JRZ  L10    ;
        JPM  L9     ;
        SUB  B      ;
        LD   B,A    ;
        LD   A,00   ;
L8      ADD  A,E    ;
        DJNZ L8    ;
        NEG                ;
        JR   L10   ;
L9      ADD  A,E    ;
        DJNZ L9    ;
L10     ADD  A,C    ;
;
;Sortie des données :
;
        JPP  L11    ;
        LD   A,01   ;
        OUT  P2S,A  ;
        JR   L12   ;
L11     LD   A,00   ;
        OUT  P2S,A  ;
;
;Démodulation"extraction du 2éme bit" :
;
;Calcul de :
;(Sn, f2)[(Sn-1, f2)+(Sn-1, f1)]-(Sn, f1)[(Sn-1, f2)-(Sn-1, f1)]
;
L12     LD   A,00   ;
        LD   B,L    ;
        CPB                ;
        JRZ  L15    ;
        JPM  L14    ;
        SUB  B      ;
        LD   B,A    ;
        LD   A,00   ;
L13     ADD  A,D    ;
        DJNZ L13    ;
        NEG                ;
        JR   L15    ;
L14     ADD  A,D    ;
        DJNZ L14    ;
L15     LD   C,A    ;
        LD   A,00   ;
        LD   B,H    ;

```

```

CPB ;
JRZ L18 ;
JPM L17 ;
SUB B ;
LD B,A ;
LD A,00 ;
L16 ADD A,E ;
DJNZ L16 ;
NEG ;
JR L18 ;
L17 ADD A,E ;
DJNZ L17 ;
L18 SUB C ;

```

;  
;Sortie des données :

```

JPP L19 ;
LD A,01 ;
OUT P15,A ;
JR L20 ;
L19 LD A,00 ;
OUT P15,A ;
L20 LD H,D ;
LD L,E ;
RET ;

```

;  
;\*\*\*\*\*  
;Sous-programme DIBITB  
;\*\*\*\*\*

;  
;Calcul de (Sn, f1)

```

IN A,PAD ;
LD B,08 ;
LD C,00 ;
L1 RLA ;
JRNC L2 ;
INC C ;
L2 DJNZ L1 ;
LD A,C ;
SUB 04 ;
LD D,A ;

```

;  
;Calcul de (Sn, f2) :

```

      IN  A,PBD ;
      LD  B,08  ;
      LD  C,00  ;
L3   RLA      ;
      JRNC L4  ;
      INC  C    ;
L4   DJNZ L3  ;
      LD  A,C  ;
      SUB  04  ;
      LD  E,A  ;

```

```

;
;Démodulation"extraction du 2éme bit" :
;

```

```

;Calcul de  $:(S_n, f_1)(S_{n-1}, f_1) + (S_n, f_2)(S_{n-1}, f_2)$ 
;(Sn-1, f1) est stocké dans le reg H
;(Sn-1, f2) est stocké dans le reg L
;

```

```

      LD  A,00  ;
      LD  B,H   ;
      CPB      ;
      JRZ  L7   ;
      JPM  L6   ;
      SUB  B    ;
      LD  B,A   ;
      LD  A,00  ;
L5   ADD  A,D   ;
      DJNZ L5  ;
      NEG      ;
      JR   L7   ;
L6   ADD  A,D   ;
      DJNZ L6  ;
L7   LD   C,A   ;
      LD  A,00  ;
      LD  B,L   ;
      CPB      ;
      JRZ  L10  ;
      JPM  L9   ;
      SUB  B    ;
      LD  B,A   ;
      LD  A,00  ;
L8   ADD  A,E   ;
      DJNZ L8  ;
      JR   L10  ;

```

```

        L9  ADD  A,E    ;
           DJNZ L9     ;
        L10 ADD  A,C   ;
;
;
;Sortie des données :
;
           JPP  L11    ;
           LD   A,01   ;
           OUT  P1S,A  ;
           JR   L12    ;
L11  LD   A,00       ;
           OUT  P1S,A  ;
;
;Démodulation "extraction du 1er bit" :
;
;Calcul de :
;(Sn-1,f2)+(Sn-1,f1) et (Sn-1,f2)-(Sn-1,f1)
;
        L12 LD   A,H   ;
           ADD  A,L   ;
           LD   C,A   ;
           LD   A,L   ;
           SUB  H     ;
           LD   L,A   ;
           LD   H,A   ;
;
;Calcul de :
;(Sn,f1)[(Sn-1,f2)+(Sn-1,f1)]+(Sn,f2)[(Sn-1,f2)-(Sn-1,f1)]
;
           LD   A,00   ;
           LD   B,H   ;
           CPB           ;
           JRZ  L15    ;
           JPM  L14    ;
           SUB  B     ;
           LD   B,A   ;
           LD   A,00   ;
L13  ADD  A,D   ;
           DJNZ L13   ;
           NEG           ;
           JR   L15    ;
L14  ADD  A,D   ;
           DJNZ L14   ;

```

```

L15 LD C,A ;
LD A,00 ;
LD B,L ;
CPB ;
JRZ L18 ;
JPM L17 ;
SUB B ;
LD B,A ;
LD A,00 ;
L16 ADD A,E ;
DJNZ L16 ;
NEG ;
JR L18 ;
L17 ADD A,E ;
DJNZ L17 ;
L18 ADD A,C ;

```

```

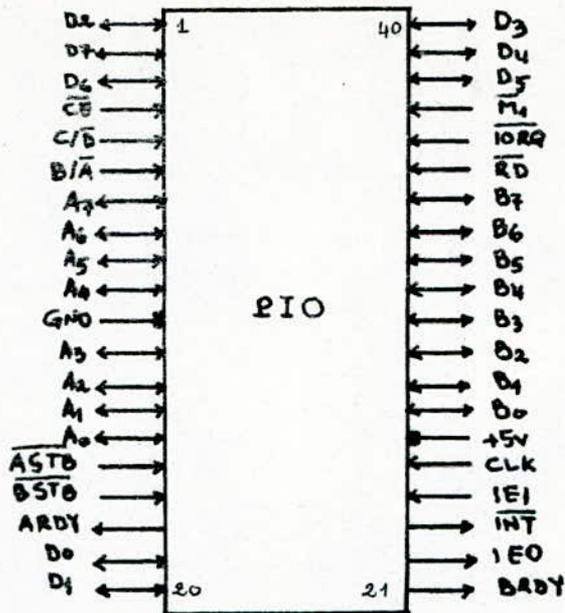
;
;Sortie des données :
;

```

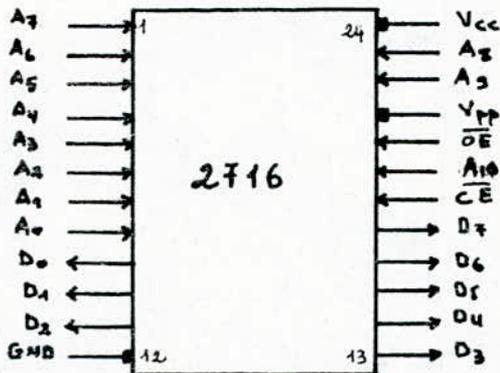
```

JPP L19 ;
LD A,01 ;
OUT P2S,A ;
JR L20 ;
L19 LD A,00 ;
OUT P2S,A ;
L20 LD H,D ;
LD L,E ;
RET ;

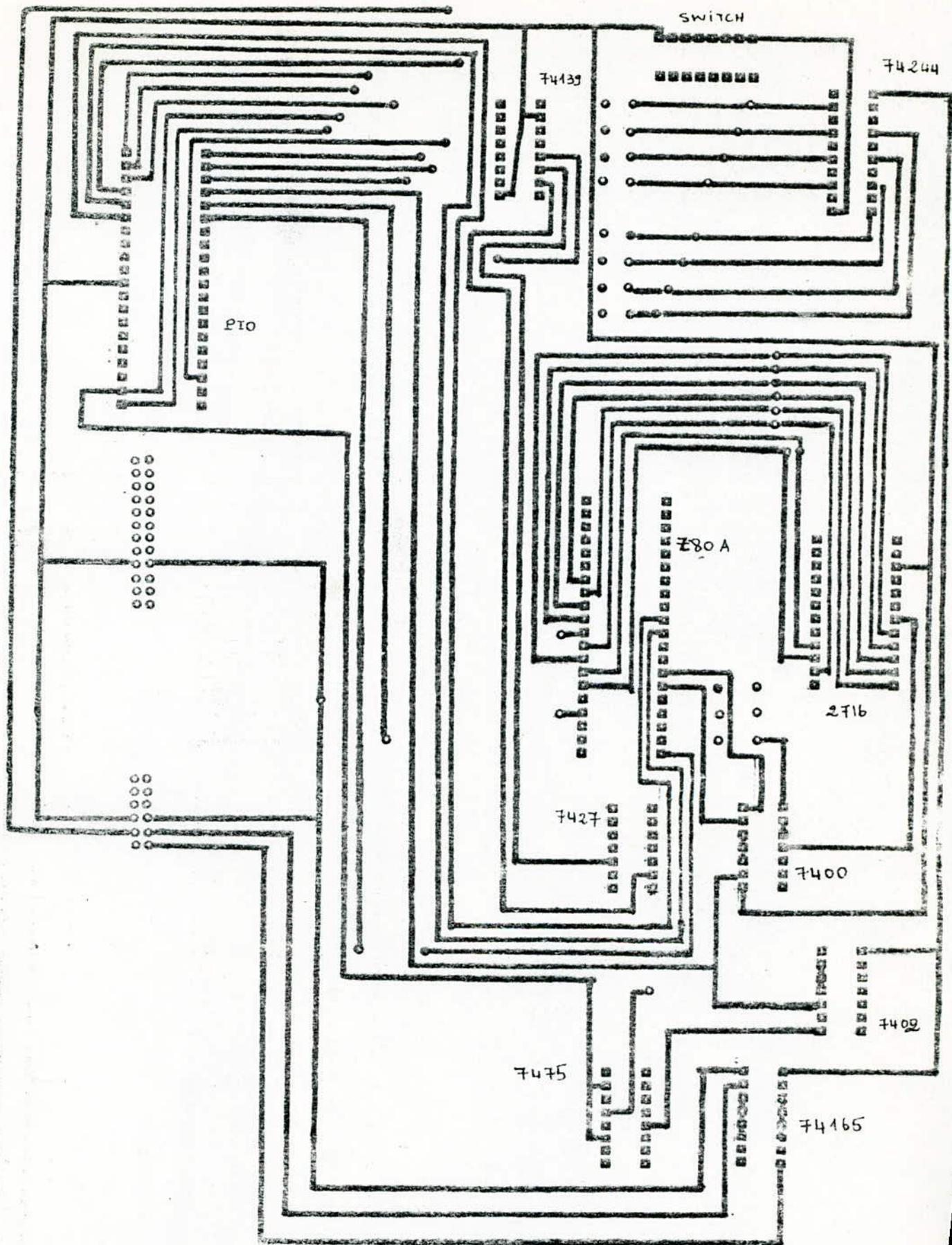
```



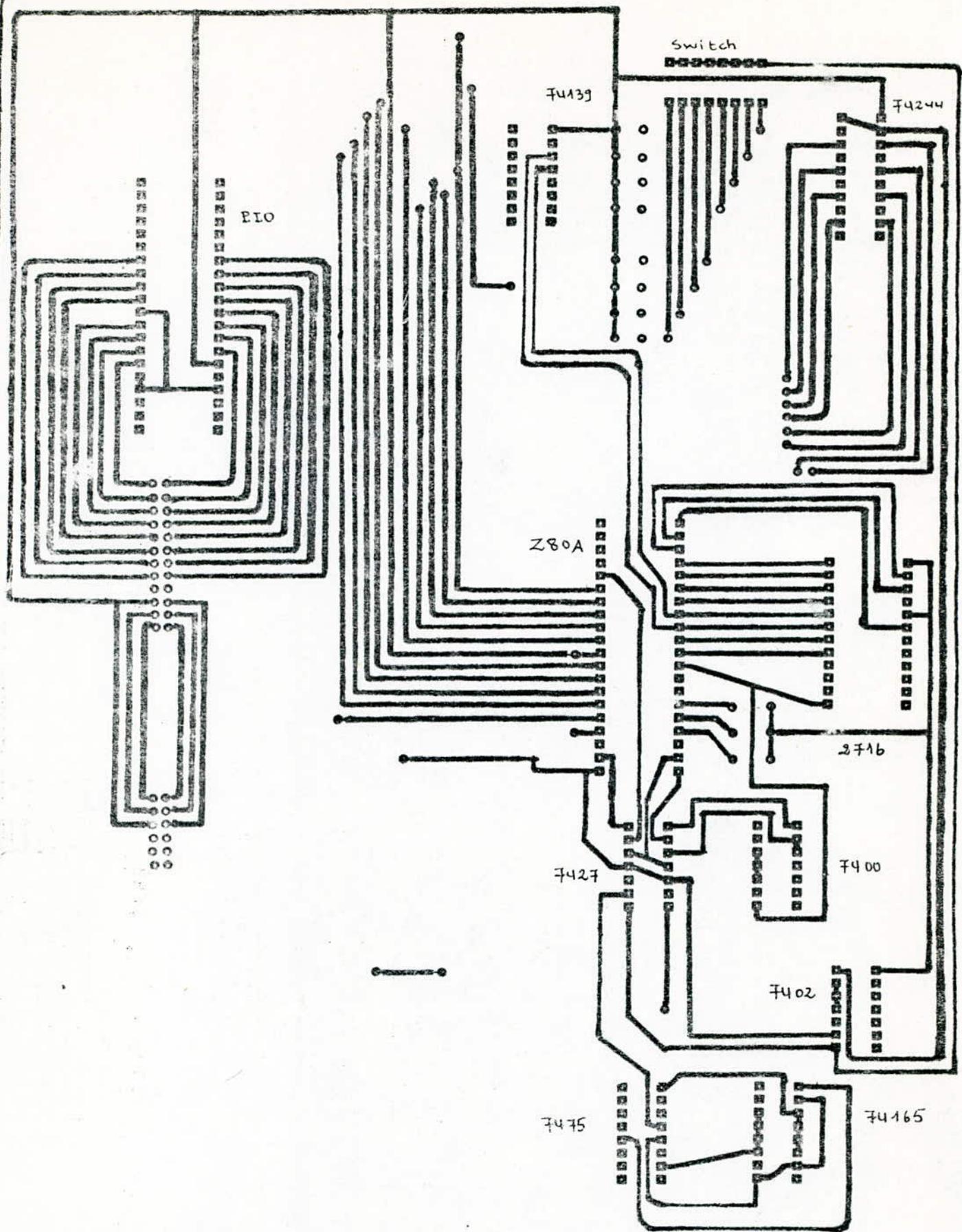
Brochage du PIO



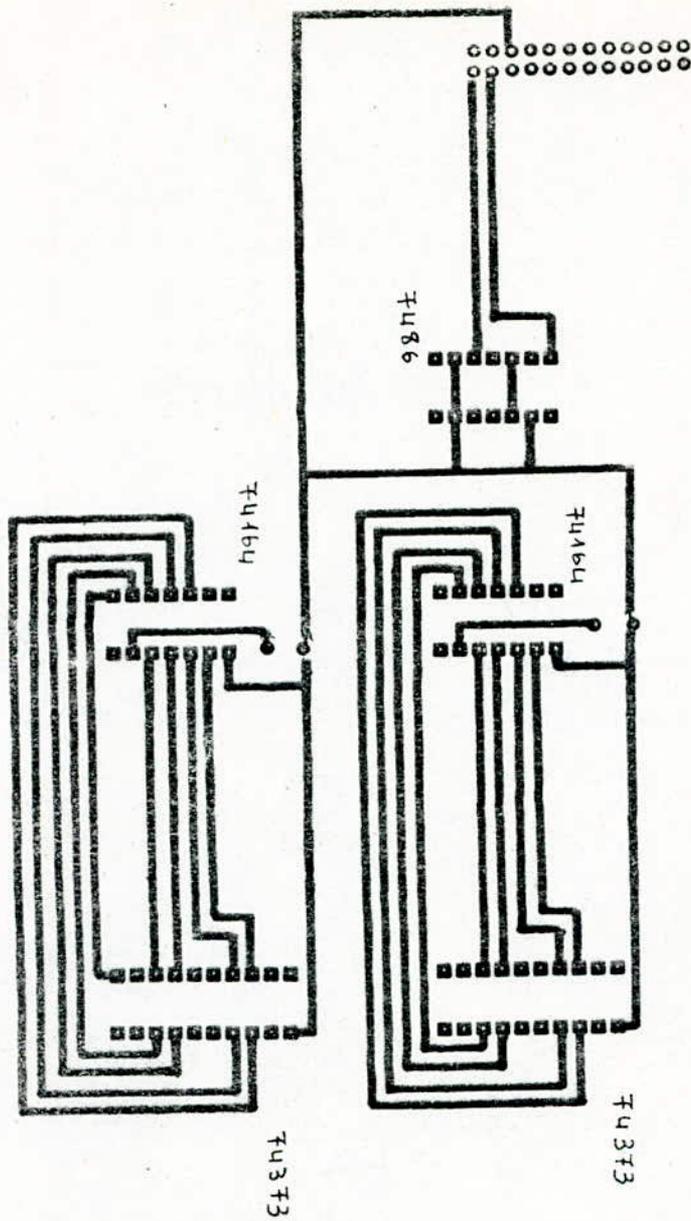
Brochage de l'EPROM 2716



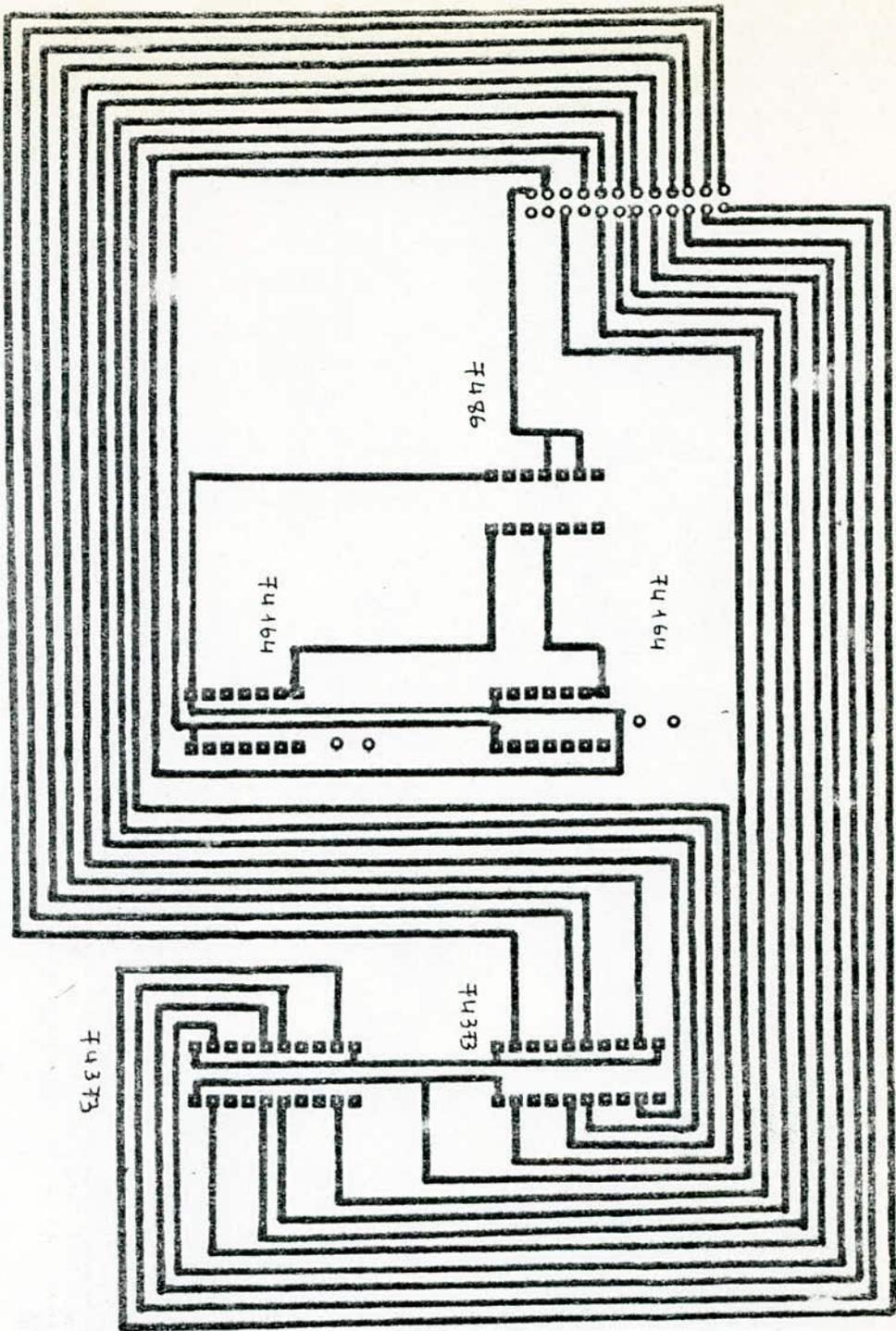
Carte CPU et interface de sortie  
 "Face composants"



Carte CPU et interface de sortie  
 " face soudure "



Carte interface d'entrée  
 "face composants"



Carte interface d'entrée  
 "face soudure"

