الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

14/86

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

Tesc

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

CODES DETECTEURS ET CORRECTEURS D'ERREURS APPLIQUES A LA TRANSMISSION DE DONNEES ET AUX TESTS DE SYSTEMES A MICROPROCESSEURS

Proposé par : Mme B. KAMINSKA

Etudié par : DOUMI Nacer-eddine
TAIBI Salah

Dirigé par : Mme B.KAMINSKA

PROMOTION :
JANVIER 86

E.N.P. 10, Avenue Hacen Badi – EL-HARRACH – ALGER

D E D I C A C E S

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

- A mes parents,

-Mes frères,

-Mes soeurs,

-Toute ma famille,

- et tous les autres, proches ou lointains, de passage ou de rencontre, aux anciens et aux nouveaux, aux petits et aux grands, bref, à tout mes amis.....

DOUMI Nacer

-A la mémoire de mon père : qu'il repose en paix,

-A celle qui a sué pour faire de moi ce que je suis : ma mère

-A celui qui m'a tant aidé : mon frère,

-A ma belle soeur, ma soeur et ma petite nièce,

-Aux familles : YAHIA-CHERIF HOUACINE HACEINI TAIBI

TAIBI Salah

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

REMERCIEMENTS

-Que Mme B. KAMINSKA trouve ici nos plus vifs remerciements pour les conseils qu'elle nous a prodiguée dans l'élaboration de ce modeste travail.

--Nous remercions tout le personnel du Département enseignants et travailleurs pour l'aide qu'ils nous ont donnée durant notre formation

-Nous ne terminerons pas sans remercier vivement Djamila S. pour le formidable travail de mise en page, ainsi que Mohamed DOUMI pour la documentation qu'il a mis à notre disposition

TAIBI -DOUMI

S O M M A I R E

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PREMIERE PARTIE : ETUDE THEORIQUE DES CODES

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

I. INTRODUCTION :

La transmission et le traitement de données sont des techniques incorporées dans le domaine informatique. La transmission de données à longue ou courte distance à travers une ligne de transmission est exposée à des perturbations extérieures qui altèrent l'information acheminée par celle-ci. De même les systèmes informatiques qui ont à transmettre ou à traiter ces données sont sujet à des défauts technologiques et électriques dont les conséquences sont des erreurs logiques qui faussent ces données. Pour palier ce manque d'exactitude, on a été amenés à faire une recherche dans le domaine des codes détecteurs et correcteurs d'erreurs, notamment les codes cycliques qui répondent aux critères de la détection et éventuellement correction des erreurs indépendantes et sous forme de paquets les plus fréquents et dont les algorithmes de codage et de décodage sont simples.On utilisera ces codes pour détecter et corriger les erreurs qui altèrent L'information lors de sa transmission. Le principe consiste à introduire avec l'information utile une redondance dont le rôle est la protection de l'information. A la réception le traitement de la redondance nous renseignera sur l'exactitude ou non de l'information transmise. Ceci n'aura lieu qui si le système qui a àtraiter ces données fonctionne bien, c'est à dire il n'est pas sujet à des défauts. Donc ce système demande un test de vérification de son bon fonctionnement qui sera fait grace à un algorithme de détection d'erreurs réalisé sous forme programmée. Cet algorithme de détection testera les modules composant ce système et nous renseignra sur leur fiabilité.

L'objet principal de ce travail consiste à mettre en évidence à l'aide des codes, des méthodes de protection de données et de test d'un système à microprocesseur.

Dans la première partie, on a fait l'étude théorique de quelques codes détecteurs et correcteurs des erreurs les plus fréquentes ainsi que les causes de défaillance d'un système à microprocesseur.

Dans la deuxième partie, on a donné un exemple d'application des codes dans les transmissions de données.

Un chapitre est réservé à une étude brève de la structure interne d'un système à microprocesseur ainsi que les codes utilisés pour le test des modules le composant tel que le microprocesseur et les mémoires.

II. CARACTERISTIQUES DES CODES :

2.1. EFFICACITE D'UN CODE : e

On appelle efficacité d'un code, le rapport moyen du nombre de messages faux et reconnus faux au nombre total de messages faux. La proportion (1-e) est donc fausse et reconnue comme juste.

2.2. REDONDANCE : K

Ce sont des bits supplémentaires transmis avec l'information utile dans le but de contrôler le message et éventuellement le corriger en cas d'erreurs.

2.3. TAUX D'ERREUR BRUT : T_b :

C'est la proportion moyenne de message faux reçus, détectés ou non. Il mesure la qualité intrinsèque de la voie de transmission et il dépend de la loi de probabilités de répartition des erreurs sur le canal. Si ces erreurs sont indépendantes et ont pour probabilités P, sur un message de longueur n bits, on a :

$$T_b = n.P$$

2.4. Taux d'erreurs risiduel : T_r :

C'est la proportion de messages finalement faux après épuisement de la procédure qui peut consister à demander répétition des messages faux, correction de certaines erreurs etc.

Il mesure la qualité finale de la transmission, celle qui interesse l'utilisateur. La relation qui lie T_b , T_r , e est :

$$T_r = T_b\ (1-e)$$

2.5. RENDEMENT : R

C'est le rapport entre le nombre de bits utiles et le nombre de bits émis. SOit un message de m bits utiles, c'est à dire informatifs et K bits de redondance; le rendement du code est :

$$R = \frac{m}{m + K}$$

$$n = m + K$$

2.6. POIDS D'UN MOT :

On appelle poids d'un message, le nombre de"1" qu'il contient exp : 10 100 110, c'est un mot de 8 bits de poids égale à 4 .

2.7. NOTION DE DISTANCE : dm

La distance entre 2 messages . L'un transmis , l'autre reçu est le nombre de bits différents entre les 2 messages.

Exp :

$$M = \boxed{1}\ 0\ \boxed{0}\ 110\ \boxed{1}$$
$$M' = \boxed{0}\ 0\ \boxed{1}\ 110\ \boxed{0} \qquad dm = 3$$

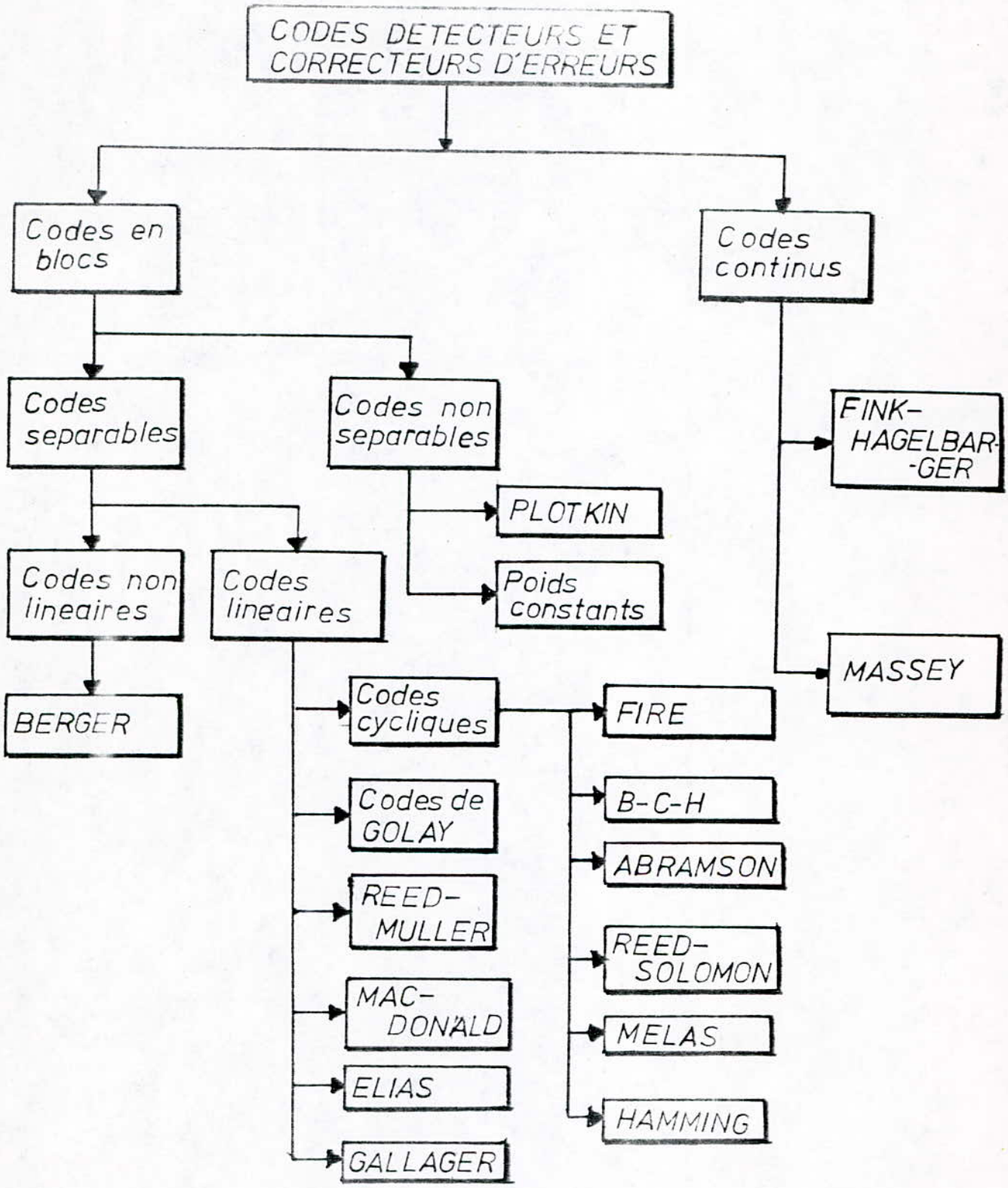
Remarques :

1. La distance dm représente le nombre d'erreurs contenues dans le message reçu;
2. Un code formé de message à distance $\geqslant 2d + 1$ les uns des autres, detecte toutes les erreurs d'ordre $\leqslant 2d$ et corrige toutes les erreurs d'ordre $\leqslant d$.

III. INTERET DU CODAGE :

1. le codage simplifie notablement la traduction de l'informati numérique par des signaux électriques;
2. On peut jouer sur le codage pour détecter et éventuellement corriger les erreurs dans les transmissions de données et les pannes dans les systèmes informatiques.

IV. CLASSIFICATION DES CODES :



V. CODES EN BLOCS :

5.1. DEFINITION : L'information après codage est divisible en blocs indépendants de longueurs :

- fixes : codes en blocs homogènes
- variables : codes en blocs hétérogènes.

La longueur des blocs homogènes est n bits . Le message utile M (x) est de longueur m bits, tel que n > m . Dans ce cas on a un code (n,m) . Chaque message expédié aura donc :

$$\boxed{n - m = K}$$ bits de contrôle

Ces K bits peuvent -être classés de 2 manières dans les codes :

- séparés,
- non-séparés.

On a donc des codes séparables et non-séparables. Parmi les codes séparables on distingue les codes linéaires et, non-linéaires et, parmi les codes linéaires on trouve les codes très importants que sont les codes cycliques [VOIR CLASSIFICATION DES CODES]

5.2. CODES LINEAIRES :

5.2.1. DEFINITION : Soit un ensemble $\mathcal{M}$ des $M = 2^m$ combinaisons de m bits; la somme de 2 éléments quelconques de M appartient à $\mathcal{M}$, donc $\mathcal{M}$ constitue un groupe. Si on choisit 2 éléments M_1 , M_2 de $\mathcal{M}$ et qui ont pour correspondants dans L, N_1 et N_2; la somme de M_1 et M_2 a pour correspondants la somme de N_1 et N_2 :

$$M_1 \in \mathcal{M} \text{ --------- } N_1 \in L$$
$$M_2 \in \mathcal{M} \text{ --------- } N_2 \in L$$
$$(M_1 + M_2) \in \mathcal{M} \text{ --------- } (N_1 + N_2) \in L$$

Un code L est linéaire s'il existe un opérateur linéaire qui transforme les éléments de $\mathcal{M}$ en les éléments de L, de ce fait nous pouvons considérer un code linéaire L (n,m) comme formé de 2^m vecteurs dans un espace vectoriel à m dimensions, la base de cet espace étant les n vecteurs générateurs à partir desquels le code a été construit [REF 1].

5.2.2. CODES LINEAIRES PARTICULIERS :

Il existe plusieurs codes linéaires, on peut citer parmi ceux-ci : codes de GOLAY, de REED-MULLER et MAC -DONALD [REF 1]. La recherche dans le domaine des codes linéaires a eu pour objet de trouver des familles de codes possèdant les propriétés suivantes :

-la capacité de détecter et de corriger des erreurs indépendantes ou se présentant sous formes de paquets .

-la simplicité de réalisation du codeur et du décodeur.

Les codes cycliques sont une famille de codes linéaires particulièrement interessantes répondant notament aux critères précédents [REF 3];

5.3. CODES CYCLIQUES :

5.3.1. DEFINITION : On appelle code cyclique, un code linéaire tel que tout message déduit par permutation circulaire d'un message quelconque du code appartient au code .

Exemple : soit un code :

$C_i = (Co, C_1, C_2 \ldots\ldots C_{n-1}) \in L(n, m)$.

alors si : $C_i = (C_{n-1}, Co, C_1 \ldots c_{n-2}) \in L(n-m)$

Le code L (n-m) est un code cyclique noté $C(nim)$

5.3.2. INTERET DES CODES CYCLIQUES:

1. ils sont de réalisation technologique simple, au moyen de registres à décalages .

2. ils sont bien adaptés à la détection et à la correction des erreurs groupés en paquets.

3. la théorie des polynômes est particulièrement appropriée à l'étude de ces codes. Elle donne les moyens de la construction des codes et de la discussion des erreurs.

5.3.3. REPRESENTATION MATHEMATIQUE DES CODES CYCLIQUES :

1. Un mot-code $C = (c_o, c_1, \ldots C_{n-1})$ est représenté par le polynôme :

$$F(X) = Co\, X^0 + C_1 X^1 + \ldots C_{n-1} X^{n-1}$$

La multiplication du polynôme $F(X)$ par X^i correspond a i permutations successives vers la droite.

$$F(X) \in C(n.m) \iff X^i F(x) \bmod (x^n + 1) \in C(n.m).$$

5.3.4. POLYNOME GENERATEUR :

a. tous les éléments d'un code cyclique C (n. m) sont multiples d'un même élément fondamental appelé polynôme générateur g (x) de degré K = n -m.

b. $g(x)$; $X.g(x)$;.....;$X^{n-K-1} g(x)$ constituent une base du code C (n.m);

c. le polynôme générateur du code C (n . m)est un polynôme qui divise $(X^n + 1)$ [VOIR REF 1]

5.3.5. RECHERCHE DES POLYNOMES GENERATEURS :

D'après le théorème de d'ALEMBERT, si X est racine de p (x), alors x^2 , x^4 ,..... $x^{2\,n-1}$ le sont aussi; on démontre aussi que :

$$[P(\alpha)]^2 = P[\alpha^2] = 0 \quad \text{[REF 1]}$$

Remarques :

1. Si p (x) est décomposable en facteurs ayant leurs coeficients dans (0,1), il est réductible. Il est irréductible dans le cas contraire.

2. Un polynôme de degré n est dit primitif si les puissances successives de α jusqu'à α^{2^n-1} engendrent les éléments non nuls de C G (2^n). Il est dit non -primitif dans le cas contraire.

3. Définition de C G (p) : l'anneau des entiers modulo p est un corps si et seulement si p est un nombre premier. Un tel corps s'appelle un corps de GALOIS et on le note C G (p) [REF 1];

4. Afin d'assurer une protection efficace de l'information, le polynôme générateur g (x) doit-être choisi de telle sorte qu'il ne divise aucun polynôme E (X) correspondant aux erreurs que l'on désire déceler [VOIR VI].

5.3.6. CONSTRUCTION DES CODES CYCLIQUES :

Construire un code, c'est déterminer les bits de contrôle en fonction des bits informatifs. Dans chaque code cyclique G (x)

on considère la partie informative S (x) de degré (M-1) séparé de la partie test t (x).

$$C\ (x) = S\ (x) + t\ (x)$$

$$S\ (x) = X^k\ M\ (x)$$

d'après l'étude des polynômes générateurs on doit avoir :

$$C\ (x) = g\ (x)\ Q\ (x)$$

or d'après le corps de Galois, on a : $t = -t$.

donc : $S\ (x) = g\ (x)\ Q\ (x) + t\ (x)$

avec degré t (x) < degré g (x) d'où la méthode suivante :
La partie test t (x) est le reste de la division de la partie significative S (x) par le polynôme générateur g (x).

Exemple : code (n,m) = (7,4)
n = 7 largeur du mot-code
m = 4 longueur du message utile
k = n-m = 3 nombre de bits de contrôle.

d'après la définition des polynômes générateurs, g(x) doit être de degré K = 3.
La décomposition du polynôme ($X^7 + 1$) en polynôme irréductibles donne :

$$X^7 + 1 = (X+1)\ (X^3 + X^2 + 1)\ (X^2 + X + 1)$$ [VOIR A1

on choisit alors le polynôme générateur suivant :

$$g\ (x) = X^3 + X\ 1$$

soit maintenant le mot utile qu'on veut coder :

M = 1010. le polynôme S(x) correspondant est :

$S\ (x) = X^k\ M\ (x) = X^6 + X^4$ degré S (x) < n-1

d'où : S = 10 10 000
cherchons t (x) : pour celà la division de S (x) par g (x) donn

t (x), soit : t (x) = X + 1 degré t (x) < K-1
d'où $t\ (x) = 0\ X^2 + 1\ X + 1\ X^0$
t = 011 les bits de contrôle sont donc 011
Le mot code (n.m) est donc :

$C\ (x) = X^6 + X^4 + X + 1$
C = 10 100 11

5.3.7. MISE EN OEUVRE DES CODES CYCLIQUES : CODAGE ET DECODAG

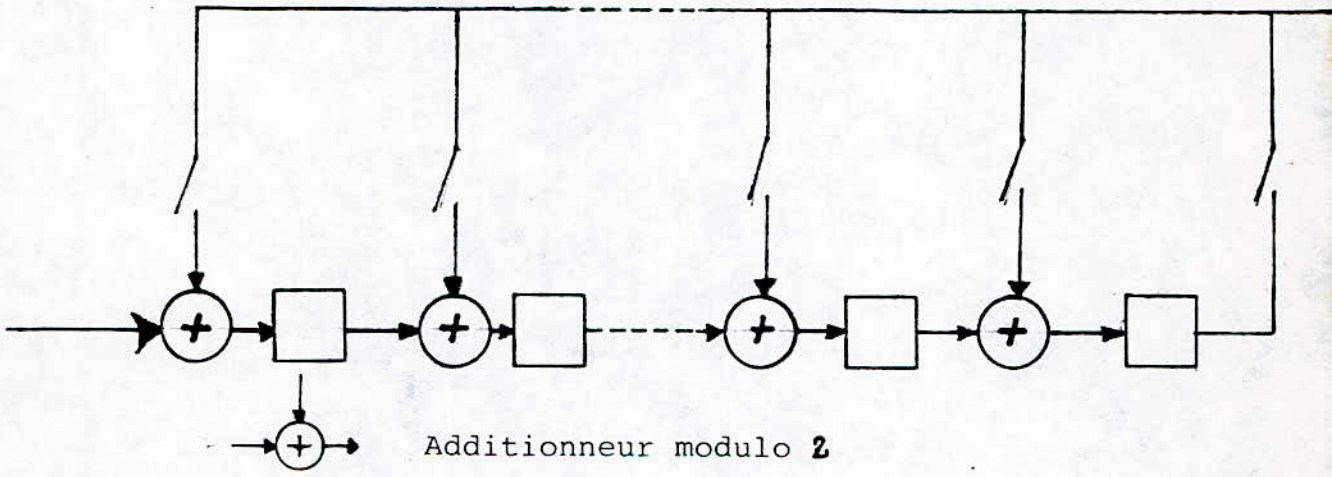
Soit un message utile de m bits. On ajoute K zéros à la fin de M (on forme ainsi X^k M (x). on réalise la division modulo 2 de X^k M (x) par g (x) :

$$X^k \quad M(x) = g(x) \quad q(x) + r(x)$$

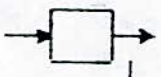
T (x) étant le reste de la division, on le récupère et on le somme à X^k M (x) on obtient le mot-code :

$$X^k \; M \; (x) + r(x) = C(x)$$

qui appartient au code puisque divisible par g(x). Les K premières cases du message sont occupées par les K bits de redondance, les (n-k) autres par l'information utile S(x). Le mécanisme de la division est à la base de l'opération de codage. Le synoptique est le suivant : FIG 1 :



Additionneur modulo 2



Registre à décalage

Interrupteur

Le message à diviser X^k M (x) s'écrit :

$$X^k \; M(x) = a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_0.$$

Les termes inférieurs ou égaux à (k-1) sont tous nuls. le polynôme générateur diviseur s'écrit :

$$g(x) = b_k \; x^k + b_{k-1} \; x^{k-1} + \ldots\ldots + b_o$$

$b_k = 1$ et $b_o = 1$

A chaque interrupteur bi (i = 0,1 ..., k) est affecté un coefficie bi de g(x). Si bi = 1, l'interrupteur est fermé. Le polynôme X^k (M (x) est présenté à l'entrée poids fort en tête. A chaque impul sion d'horloge, le contenu d'une cellule est remplacé par la somme modulo 2 du contenu de la cellule précédente et de la dernière cellule.

Au départ, les cellules sont vides. Au bout de K impulsions, les K premièrs termes du devidende X^k M (x) sont dans les cellules :

-la cellule 1 contient a_{n-k}

-la cellule k contient a_{n-1}

A l'impulsion suivante, la division commence, le contenu des cellules est alors :

$a_{n-1} \oplus a_{n-k-3}$; $a_{n-k} \oplus b_1 \cdot a_{n-1}$ . ..... ; $a_{n-3} \oplus b_{k-2}$ a_{n-1} ; $a_{n-2} \oplus b_{k-1} \cdot a_{n-1}$.

Ce contenu est égale précisement au premier reste partiel de la division de X^k M (x) par g(x).

Au bout du (n-k) ième coup, le reste final r (x) de la division est contenu dans le registre.

VI. DETECTION ET CORRECTION DES ERREURS :

6.1. DEFINITION :

L'erreur si elle se produit, affecte la donnée à traiter que ce soit lors de sa transmission ou de son traitement par un système à microprocesseur. Elle est d'ordre logique, c'est à dire des "0" qu se transforment en "1" et vis-versa. L'origine de l'erreur est dûe à des défauts technologiques [c'est à dire les matériaux choisis les techniques de conception etc...] à des défauts électriques dont notament les courts-circuits, claquage de jonction, circuits ouverts etc...; enfin a du bruit tel que les perturbations extérieures. La conséquence de ces défauts est une erreur logique.

6.2. ERREURS DANS LES CIRCUITS INTEGRES :

Les circuits intégrés entrant dans la réalisation d'un système à microprocesseur sont de différentes sortes, dont notament les microprocesseurs, les mémoires et les interfaces d'E/S. Les microprocesseurs sont des circuits intégrés complexes [VLSI, LSI]. Les défauts qui peuvent se présenter dans ceux-ci sont dûs à des défaillances de registres, à des bus d'adresses et de données dont résultent 2^{16} possibilités d'adressage et 2^8 possibilités de changement d'une donnée.

Dans les mémoires, les défauts sont relatifs à la défaillance du décodeur d'où résulte un mauvais adressage des mémoires ; à des écritures multiples c'est à dire une donnée adressée à une seule cellule se trouvant mémorisée dans d'autres, etc...

Ces défauts relatifs aux micropresseurset aux mémoires engendrent des erreurs logiques qui affectent le traitement de la donnée.

6.3. ERREURS DANS LES TRANSMISSIONS DE DONNEES :

Les erreurs se produisant dans les transmissions de données sont relatives aux perturbations extérieures et à l'équipement interne d'une ligne de transmission. Elles affectent de ce fait l'information apportée par celles-ci. Ces erreurs sont de nature logiques et elles se produisent le long de la ligne de transmission.

Pour detecter des erreurs se produisant dans les microprocesseurs et les mémoires, on utilise des codes détecteurs d'erreurs, ceux-ci ont pour rôle de vérifier -par exemple pour un microprocesseur si la fonction qui lui est demandée à bien lieu et ne s'est pas transformée en une autre; et-pour une mémoire-de vérifier que l'écriture d'un mot dans une position-mémoire ne s'est pas transformé en un autre. POur les transmissions de données, on vérifie à la réception que l'information utile est bien celle émise et ceci en utilisant des codes détecteurs et correcteurs d'erreurs qui ont pour principe l'utilisation d'une redondance avec l'information utile. A la réception, il s'agit d'analyser le mot-code reçu $C'(x) = C(x) + E(x)$; avec $C(x)$ mot-code transmis et $E(x)$ un polynôme d'erreurs qui peut se présenter sous forme d'une erreur simple, double ou même sous forme de paquets. Avant d'aborder la méthode de détection des erreurs donnons quelques propriétés importantes.

PROPRIETES :

1. En tant que codes linéaires, les codes cycliques détectent toutes les erreurs extérieures à eux et ne détectent jamais les erreurs qui transforment un mot-code en un autre.

2. Les codes cycliques détectent toutes les erreurs simples. Dans ce cas le polynôme d'erreur s'écrit $E(x) = X^i$, erreur dans la $(i + 1)$ ième position.

3. Les codes cycliques formés à partir de polynômes générateurs contenant $(1 + X^h)$ en facteur détectent tous les cas d'erreurs en nombre impair :

$$\boxed{g(x) = (1 + X^h) \cdot F(x)}$$

Si f (x) possède t termes, g (x) en possède 2 t moins les contractions de termes 2 à 2 dûes aux additions modulo 2. g(x) possède en tous cas un nombre pair de termes, E (x) en possède un nombre impair, donc il ne peut être divisé par g (x).

4. Si le polynôme générateur est primitif [VOIR 5-3-5], le code C (n,m) detecte tous les cas d'erreurs simples et d'erreurs doubles. Dans ce cas E (x) est de la forme

$$\boxed{E(x) = X^i + X^j = X^i (1 + X^{j-i})}$$

puisque g(x) est primitif donc il divise $(x^n + 1)$, mais il ne peut diviser $(x^{j-i} + 1)$ car $(j-i) < n$.

5.Tout code C (n,m) généré par un polynôme générateur de degré K detecte tout paquet comprenant jusqu'à K erreurs. Dans ce cas E (x) s'écrit :

$$\boxed{E(x) = X^a \cdot F(x)}$$

degré F (x) $\leqslant$ K-1, or degré g (x) = K, donc F (x) n'est pas divisible par g (x). La proportion de paquets d'erreurs de longueurs $l > k$ non detectée est :

$$\begin{cases} 2^{-(k+1)} & \text{si } l = k + 1 \\ 2^{-k} & \text{si } l > k + 1 \end{cases}$$

Cette expression exprime la probalité d'erreurs P_e de ne pas déte
ter un de ces paquets d'erreurs de longueur 1.

6.4. DETECTION DES ERREURS :

Soit C' (x) le mot-code reçu :

$$C'(x) = C(x) + E(x).$$

trois possibilités se présentent :

1. E = (00...0) dans ce cas C' (x) = C (x). On dit que la transmission a été correcte, donc pas d'erreur.

2. $E \neq (00....0)$ mais E appartient aux 2^n mots-codes, donc $C'(x) \neq C(x)$ on dit qu'il y à une erreur mais non décelable.

3. $E \neq (00...0)$ et E n'appartient pas aux 2^n mot-codes, donc $C'(x) \neq C(x)$, il y à une erreur qui est décelable.

REMARQUES :

1. La non-appartenance d'un mot-erreur aux 2^n mots codes est une condition nécessaire et suffisante de détection.

2. Soit E (x) le polynôme erreur. La condition nécessaire et suffisante qu'une erreur soit décelable est que son polynôme E (x) ne soit pas divisible par le polynôme générateur g (x) du code. D'après cette deuxième remarque on peut déduire :

A la réception on divise le polynôme reçu C' (x) par g (x). Si le reste est différent de zéro, il y a erreur dans la transmission de C (x). Les erreurs sont de différentes sortes : erreurs simples doubles où par paquets :

6.4.1. ERREURS SIMPLES :

Tout mot erreur est un permuté du mot : E = (00...01). Le polynôme correspondant est

$$E(x) = X^i \qquad 0 \leqslant i \leqslant n-1$$

erreur dans la (i + 1) ième position. Ce polynôme n'est pas divisible par g (x) car : g (x) = 1 + ...

6.4.2. ERREURS DOUBLES :

soit E = (00...100...010...0) . Les mots-erreurs sont les permutés de E. Le polynôme erreur associé est de la forme :

$$E(x) = X^{i} + X^{j} \quad 0 \leqslant I \leqslant n-1 \; ; \; o \leqslant j \leqslant n-1 \; ; \; i \neq j$$

pour que ce polynôme d'erreur soit détecté il faut et il suffit que g (x) ne divise aucun des polynômes E (x).
La condition nécessaire et suffisante est que g (x) soit un polynôme primitif. [VOIR 5.3.4.].

6.4.3. PAQUETS D'ERREURS :

On appelle paquets d'erreurs de longueur au plus égale à 1, tout permuté circulaire du mot : E = (00... OO C_1 C_2 .... C_1); où les C_i ne sont pas tous nuls. Il existe (2^{l}-1) mots, dont les polynômes associés sont de degré inférieur à 1. Pour que ces polynômes ne soient pas divisibles par g (x) ;il faut et il suffit que degré g (x) > 1.
CONCLUSION :

Les codes cycliques détectent tous les paquets d'erreurs de longueurs :

$$l \leqslant k \leqslant n-m$$

6.5. CORRECTION DES ERREURS :

Dans un système à micro processeur, on a pas à corriger les erreurs qui affectent celui-ci, mais il s'agit uniquement de détecter les modules défaillants, ceci grâce à l'analyse des résultats de test.

Dans les transmissions de données, nous avons par contre à corriger des erreurs qui se produisent le long de la ligne de transmission et qui sont comme on la spécifié, simples; doubles oupar paquets. La méthode utilisée pour détecter et corriger ces erreurs est le calcul du syndrome du mot-code reçu C' (x).

6.5.1. DEFINITION DU SYNDROME :

On appelle syndrome, le reste de la division du message reçu par le polynôme générateur .

$$S(x) = \text{reste de} \left[\frac{C'(x)}{g(x)} \right] \qquad \text{degré } S(x) \leqslant k-1$$

On sait que seul C (x) = C' (x) + E (x) est divisible par g (x), donc on peut aisément corriger l'erreur; pour celà calculons le syndrome de c' (x) :

$$S\ [\ C'\ (x)\ ] = S\ [\ C\ (x)\ ] + S\ [\ E\ (x)\ ]$$

or $S\ [\ C(x)\ ]$ reste $[\frac{C\ (x)}{g\ (x)}\ ] = 0$

d où : $S\ [\ C'(x)\ ] = S\ [\ E\ (x)\ ]$

A la reception on teste le syndrome d'erreur : :

- si $S\ [\ C'\ (x)\ ] \neq 0$, il y a erreur . $E\ (x) \neq 0$

-si $S\ [\ C'\ (x)\ ] = 0$, $C'\ (x) = C\ (x)$ et $E\ (x) = 0$, mais le message peut être entaché d'un type d'erreurs non-décelables, dans ce cas $E\ (x) \neq 0$.

CONCLUSION :

Les opérations de codage et de décodage consistent à diviser un polynôme par g (x) et, dans les 2 cas, on ne s'interressera qu'au reste de la division.

6.5.2. CORRECTION D'UNE ERREUR SIMPLE :

Le polynôme d'une erreur simple s'écrit : $E\ (x) = X^i$, $o \leqslant i \leqslant n-1$
Le syndrome d'erreur est : $S\ [\ E\ (x)\ ]$ = reste $[\frac{xi}{g\ (x)}] = R\ (x)$ pour localiser l'erreur x^i, il suffit de résoudre l'équation :

$$\boxed{X^i = R\ (x)\ \text{modulo}\ (g\ (x))}$$

la valeur (i + 1) désigne la position du bit erroné dans le message reçu C (x)

6.5.3. CORRECTION DE t erreurs :

On montre qu'un code formé de message à distance minimale :
$dm \geqslant 2t + 1$ détecte toutes les erreurs d'ordre $\leqslant 2t$ et corrige les erreurs d'ordre $\leqslant t$ [VOIR REF 9].

On va aborder quelques codes cycliques qu'on a jugé interessant du point de vue compatibilité avec les différentes erreurs citées :

VII . CODES CYCLIQUES PARTICULIERS :

7-1 CODES BOSE CHAUDHURI HOCQUENGHEM B C H

7-1-2 DEFINITION

Les codes B-C-H sont remarquables par leurs grande capacité de détection des erreurs usuelles et leurs faibles redondance. Ils sont faciles à construire. Le polynôme générateur g(x) est un diviseur de $(X^n + 1)$; si α est une racine n ième est primitive de l'unité et si g (x) a entre autre parmis ces racines les puissances successives de x jusqu'à l'ordre S, soit :

α, α^2,...., α^S; il donne naissance à un code B-C-H de longueur n et de distance minimale dm = s + 1.

7.1.2. RECHERCHE DU POLYNOME GENERATEUR :

partons de la longueur n supposée de la forme :

$$\boxed{n = 2^r - 1}$$

et d'une racine n ième primitive α de l'unité. On impose au code de corriger t erreurs. D'après [6-4.3], S doit être tel que :

$$\boxed{S = 2^t}$$

par conséquent la suite des 2 t racines est : $x, x^2, \ldots x^{2t}$

Le polynôme générateur g (x) s'obtient à partir de polynôme irreductible Pi (x), ces polynômes sont tels que :

- ils sont au plus de degré r,
- ils admettent avec leurs racines les carrés de celles -ci,
- ils sont calculés à partir de leurs racines connues [VOIR A2]

il suffit donc de t racines : $\alpha, \alpha^3, \ldots\ldots \alpha^{2t-1}$

d'où : $$\boxed{g(x) = \text{PPCM} \quad P_1(x), P_3(x) \ldots, P_{2t-1}(x)}$$

Le PPCM a ici pour seul effet de n'utiliser que les facteurs irréductibles distincts. g (x) est donc de degré k au plus égal à (t.r).

k désigne le nombre de bits de contrôle et la longueur des paquets d'erreurs décelables.

7.2. CODES DE REED-SOLOMON R-S :

7.2.1. DEFINITION :

Dans le cas des codes B-C-H, l'élément primitif appartient à un corps de GALOIS C G (2^P) et, en base 2. Dans le cas des codes R-S, la base n'est plus 2, mais un entier quelconque q, donnant lieu à un corps de GALOIS C G (q^n). Les coefficients des polynômes sont des entiers : 0,1,2, ..., q-1 (base q) et non seulement (0,1) base 2 comme dans le cas B-C-H. Si q est de la forme 2^P, avec P un entier quelconque; chaque symbôle est représenté par P bits.On a dans ce cas un code R-S à structure binaire. La largeur du mot-code est :

$$\boxed{n = 2^P - 1}$$

Si sa distance minimale est dm, le nombre de symbôles de redondance K = dm -1, si t est le nombre d'erreurs à corriger,

on a : $\boxed{dm = 2t + 1}$

d'où $\boxed{k = 2t}$

le nombre de symboles d'information est :

$$\boxed{m = n - k = 2^P - 2t - 1}$$

on a donc un code R-S , C ($2^P - 1$, $2^P - 1 - 2t$) exprimé en symbôles de p bits, ce qui donne :

$$\boxed{n = P\,(2^P - 1)\ \text{bits}}$$

$$\boxed{m = P\,(2^P - 1 - 2t)\ \text{bits}}$$

7.2.2. RECHERCHE DU POLYNOME GENERATEUR :

Si α est un élément primitif quelconque de C G(2^P), le polynôme irreductible associé à α^i étant ($x-\alpha^i$), le polynôme générateur g (x) s'écrit

$$\boxed{g(x) = (x - \alpha)\ (x - \alpha^2)\ \ldots\ (x - \alpha^{2t})}$$

t : nombre d'erreurs à corriger; degré g (x) = 2 t = k.

α : racine d'un polynôme primitif p (x) de degré P. g (x) se met sous la forme :

$$g(x) = x^{2t} + g_{2t-1} x^{2t-1} + g_{2t-2} x^{2t-2} + \ldots + g_i x^i + \ldots + g_0$$

Par identification des 2 relations de g (x), on détermine les g_i d'où le polynôme générateur g (x).

7.3. CODE FIRE :

7.3.1. DEFINITION

Ils sont adaptés à la correction de paquets d'erreurs de longueur importante ,tout en ayant un rendement assez élevé et, ils sont de réalisation simple.

7.3.2. RECHERCHE DU POLYNOME GENERATEUR :

Les codes FIRE sont définis par le polynôme générateur de la forme.

$$g(x) = P(x) \cdot [X^q + 1]$$

$$\text{degré } g(x) = k = p * q$$

P : degré du polynôme irréductible P (x) .La longueur n du code est définie par :

$$n = PPCM[e, q]$$

e : ordre des racines de p (x).

$$e = 2^P - 1$$

l'ordre e ne divise par q .

7.4. COMPARAISON DES CODES FIRE B-C-H et R-S

CODE FIRE :

Ils sont adaptés à la protection de l'information contre les paquets d'erreurs et jouissent d'un simplicité de réalisation technologique remarquable, surtout par les codes de FIRE raccourcis. Néanmoins ils ne sont pas efficaces pour les erreurs isolées répartis sur tous le bloc .

CODES B-C-H :

Ils sont adaptés à la correction des erreurs isolées multiples. Pour une configuration d'erreurs sous forme de paquets, de longueur supérieure au nombre d'erreurs que peut corriget le code, la capacité de correction ne suffit plus.

CODE R-S :

Ils sont en théorie puissants, puisqu'ils peuvent corriger des erreurs par paquets, à condition que ces paquets soient bien répartis. pour les faibles largeurs de paquets d'erreurs, ils ne sont plus efficaces. Il est nécessaire d'augmenter la puissance de correction de plusieurs paquets par blocs, ce qui se paye bien sur par une compléxité technologique et une perte en rapidité d'éxécution.

XIII. CONCLUSION :

Dans cette première partie, nous avons fait une étude mathématique restreinte de quelques codes dont notament cycliques qui sont les mieux adaptés à la détection et éventuellement correction des erreurs les plus fréquentes quon rencontre lors de la transmission de données ou dans des systèmes informatiques déffectueux.

APPLICATION DES CODES

IX. APPPLICATION DES CODES A LA TRANSMISSION DES DONNEES :

Le traitement des informations à distance à travers les lignes de télécommunication à partir de périphériques dits terminaux est sujet à des perturbations pouvant entrainer une déformation de l'information et fausser les résultats des traitements.

Pour celà la théorie des codes est bien adaptée pour la protection.

Dans l'application suivante, on verra un cas pratique de ces codes. Le choix du code de FIRE est dû au fait que celui-ci est très facile à mettre en eouvre que ce soit par une logique cablée ou programmée.

9.1. CODE DE FIRE : [VOIR 7-3]

9.1.1. CAPACITE DE DETECTION ET DE CORRECTION :

soient 2 paquets d'erreurs, l'un de longueur b, son polynôme associé est :

$$E(x) = x^{j} B(x) \qquad \text{degré } B(x) = b-i$$

l'autre de longueur d, son polynôme associé est :

$$E(x) = x^{j} D(x) \qquad \text{degré } D(x) = d-j$$

Ces codes peuvent corriget tout paquet d'erreurs de longueur au plus égale à b et déceler simultanément tout paquet de longueur au plus égale à d, si et seulement si :

$$\begin{cases} d > b \\ q \geqslant d+b-1 \\ p \geqslant b \end{cases}$$

[REF 4]

Ils peuvent déceler toutes combinaison de 2 paquets, si la longueur du paquet le plus court est inférieure ou égale à p et la somme des longueurs des paquets n'est pas supérieure à (q+1); ou encore détecter tout paquet d'erreur dont la longueur n'est pas supérieure a p+q.

Le polynôme d'erreur s'écrit :

$$E(x) = X^{i} B(x) + X^{j} D(X)$$

9.1.2. CODAGE DES CODES FIRES

Pour coder l'information suivant un code de FIRE, on procède de la même façon qu'en(5-3.7) :

$$\boxed{\begin{array}{l} C(x) = X^k M(x) + R(x) \\ R(x) = \text{reste de } (X^k M(x)/g(x)) \end{array}}$$

9.1.3. DECODAGE DES CODES FIRE :

Si le code FIRE a à corriger une erreur de la forme :

$$E(x) = X^i B(x) \text{ modulo } (X^n + 1)$$
$$i < (n-1) \text{ et degré } B(x) = b-i$$

Le mot reçu s'écrit :

$$C'(x) = C(x) + X^i B(x) = C(x) + E(x)$$

Le syndrôme d'erreur est :

$S(C'(x)) =$ reste de $(E(x)/g(x)) =$ reste $(x^i B(x)/g(x))$.

Si $i = 0$, alors $S(C'(x)) = E(x)$, car degré $E(x) <$ degré $g(x)$

si les $(k-b)$ termes de poids forts du syndrôme sont nuls, les b termes de poids faibles de ce syndrôme désigent la valeur du paquet d'erreur.

Dans ce cas le polynôme d'erreur est localisé.

Si les $(k-b)$ termes ne sont pas nuls, alors $B(x)$ ne sera pas localisé. Pour le localiser il faut s'appuyer sur la définition des codes cycliques:

"la propriété d'appartenance ou de non appartenance au code est invariante par permutation cyclique ".

soit $X\,C'(x)$ un permuté de $C'(x)$.

Si les $(k-b)$ termes de poids forts de $S_1(x) =$ reste $(X\,C'(x)/g(x))$ sont nuls, donc les b termes de plus bas degré de $X\,C'(x) = C_1(x)$ sont erronés. La valeur de ces erreurs est les S_b termes de plus bas degré de :

$$S_1(x) = XS(x) \text{ modulo } g(x)$$

Soit un buffer qui sauvegarde les n bits de $C'(x)$. Le décodeur calcule $S(C'(x))$. $C_1(x)$ est le contenu du buffer après permu-circulaire de $C'(x)$.

$$C_1(x) = X\,C'(x)$$

$$S[C_1(x)] = S[X\,C'(x)] = \text{reste}\,[X\,C'(x)/g(x)]$$

$$= \text{reste}\,[X\,C(x) + XE(x)]/g(x) = \text{reste}$$

$$[X\,E(x)/g(x) = X\,\text{reste}\,[E(x)/g(x)] = X \cdot S[C'(x)]$$

$$= X \cdot S(x) \text{ modulo } g(x).$$

REMARQUES :

1. Si les S_{k-b} bits de $S_1(x)$ sont différents de zéro, on itère l'opération jusqu'à obtenir les S K - b = 0

$$\boxed{s(x) = X^i\ S(x) \text{ modulo } g(x)}$$

dans ce cas la correction se fait en ajoutant les S_b bits de $S_i(x)$ aux b termes de plus faibles poids de $C_i(x)$.

2. Si parès n itérations on arrive pas à obtenir les $S_{k-b} = 0$, donc on a dépassement de capacité et le code détecte uniquement des erreurs mais ne les corrige pas.

3. Pour obtenir le mot-code initial C (x), on décale le mot-code corrigé $C_i(x)$ de (n-i) position vers la droite : on obtient :

$$\boxed{C(x) = C_i(x) \quad \text{corrigé}}$$

9.1.4. EXEMPLE : CODE FIRE C (15,8)

On va traiter un exemple pratique des codes FIRE qui corrigent les paquets d'erreurs de longueurs $1 \leqslant b$. Pour celà fixons d'abord les longueurs de 2 paquets que pourra déceler ce code. Soit b = 2 la longueur du paquet le plus court et d = 4 celle du plus grand. Le polynôme irréductible p (x) doit être de degré $P > b$ on prend P = 2;

Le polynôme générateur g(x) est de degré k = p+q; avec $q \geqslant b+d-1$ d'où q=5 et k=7.

D'après la table donnant les polynômes irréductibles [VOIR A 1] :

$$p(x) = X^2 + X + 1$$

$$g(x) = p(x)\ [X^9 + 1] = X^7 + X^6 + X^5 + X^2 + X + 1$$

L'ordre des racines de p (x) est :

$$e = 2^P - 1 = 3$$

La longueur du mot-code est :

$$n = \text{ppcm} \left[\quad ; g \right] = 15$$

en résumé :

$$g(x) = X^7 + X^6 + X^5 + X^2 + X + 1$$

$$n = 15$$

$$k = 7$$

$$m = 8$$

Le rendement $R = M/n = 53,3\%$

Le code est C (15,8)

CODAGE :

$C(x) = X^k M(x) + R(X)$

M (x) =message utile de degré (M-1) = 7

R (x) =reste de (X^k M (x)/g (x))

choisissonsle message utile suivant :

$$M(x) = X^7 + X^5 + X^4 + X^2 + X \quad ; \quad M = 10110110$$

$$M^7 \quad M(x) = X^{14} + X^{12} + X^{11} + X^9 + X^8$$

faisons la division euclidienne (modulo 2) de $X^7 M(x)$ par g (x)

Le reste est $R(x) = X^6 + X^3 + X^2$ degré R (x) = k-1 = 6

Le mot code sera donc :

$$C(x) = X^{14} + X^{12} + X^{11} + X^9 + X^8 + X^6 + X^3 + X^2$$

ou : C = 101101101001100

DECODAGE :

Le mot recu sera : $C'(x) = C(x) + E(x)$

introduisons des erreurs par paquets de longueur variables, puis voyons comment le code arrive à les détecter et les corriger.

1. Soit le bit X^{14} erroné. Donc l=1 eereur simple

Le mot reçu sera :

$$C'(x) = X^{12} + X^{11} + X^9 + X^8 + X^6 + X^3 + X^2$$

C = 001101101001100

La division de C' (x) par g(x) donne un reste qui est un syndrome d'erreur . Ce reste nous renseigne sur la position des erreurs :

$$S(x) = X^6 + X^5 + X^4 + X + 1 \qquad S = 1110011$$

degré de S (x) = k-1 = 6

Les S_{k-b} bits de poids forts de S (x) ne sont pas nuls. Utilisons la clé de FIRE, c'est à dire décalons C' (x) d'une position vers la droite et calculons son syndrome S_1 (x) correspondant au mot-code

C_1 (x) = X C' (x)

La division euclidienne de C_1 (x) par g (x) donne

$$S_1 (x) = X\ S (x) \text{ modulo } g (x) = X^\circ = 1$$

$$S_1 = 0000001$$

Les S_{k-1} bits de poids forts de S_1 (x) sont nuls. On peut donc corriger le bit erroné en ajoutant les S_1 bits de poids faibles de S_1 (x) aux 1 bits de poids faibles de C_1 (x).

X°														X^{14}	
0	0	1	1	0	0	1	0	1	1	0	1	1	0	0	C' (x) erroné
0	0	0	1	1	0	0	1	0	1	1	0	1	1	0	C_1 (x)
1	0	0	0	0	0	0									S_1 (x)
1	0	0	1	1	0	0	1	0	1	1	0	1	1	0	C_1 (x) corrigé

pour retrouver le mot-code C (x) on continue à décaler C_1 (x) de (n-i) positions - (ici i = 1)-

2- a/ soit les 2 bits de poids forts de C_1 (x) faux donc l =2

$$C' (x) = X^{13} + X^{12} + X^{11} + X^9 + X^8 + X^6 + X^3 + X^2$$

$$S (x) = X^5 + X^4 + X^3 + X^\circ \qquad S = 0111001$$

Les S_{k-1} bits de poids forts de S (x) sont différents de zéro. Calculons S_1 (x) = X S (x) modulo g (x) relatif au mot-code :

$$C_1 (x) = X\ C' (x)$$

$$S_1 (x) = X^6 + X^5 + X^4 + X \quad , \quad S_1 = 1110010$$

de même on calcule S_2 (x):

$$S_2 (x) = X_1\ S (x) \text{ modulo } g(x)$$

$$S_2 = 0000011$$

Les S_{k-1} bits de poids forts de S_2 (x) sont nuls. Pour corriger C_2 (x) on doit ajouter à ses 1 bits de poids faibles les S_I bits de poids faibles de S_2 (x).

x^0														x^{14}	
0	0	1	1	0	0	1	0	1	1	0	1	1	1	0	C' (x) erroné
0	0	0	1	1	0	0	1	0	1	1	0	1	1	1	C_1 (x)
1	0	0	0	1	1	0	0	1	0	1	1	0	1	1	C_2 (x)
1	1	0	0	0	0										S_2 (x)
0	1	0	0	1	1	0	0	1	0	1	1	0	1	1	C_2 (x) corrigé

pour retrouver le mo-code C(x), on continue le décalage de C_2(x) corrigé jusqu'à la position n = 15, c'est à dire on le décale encore de n-2) positions

b. introduisons les erreurs aux 2 bits de poids faibles de C (x), on a toujours l = 2

$$C' (x) = X^{14} + X^{12} + X^{11} + X^9 + X^8 + X^6 + X^3 + X^2 + X + X^0$$

C' = 101101101001111

$$S (x) = x + x^0 \qquad S = 0000011$$

les S_{k-1} bits de poids forts de S (x) sont nuls. On peut corriger C'(x). On ajoute les S_1 bits de poids faibles de S (x) aux 1 bits de poids faibles de C' (x).

x													x14		
1	1	1	1	0	0	1	0	1	1	0	1	1	0	1	c' (x) erroné
1	1	0	0	0	0	0									S (x)
0	0	1	1	0	0	1	0	1	1	0	1	1	0	1	C' (x) corrigé

de même on décale C' (x) corrigé jusqu'à la position n = 15 pour retrouver le mot-code initial.

c. introduisons les erreurs au milieu du mot-code C (x) : X^{10} X^9 faux;

l = 2

$$C' (x) = X^{14} + X^{12} + X^{11} + X^{10} + X^8 + X^6 + X^3 + X^2$$

C' = 101110101001100

$S(x) = X^4 + X^\circ$ — $S = 0010001$

$S_1(x) = X^5 + X$ — $S_1 = 0100010$

$S_2(x) = X^6 + X^2$ — $S_2 = 1000100$

$S_3(x) = X^6 + X^5 + X^3 + X^2 + X$ — $S_3 = 1101110$

$S_4(x) = X^5 + X^4 + X^3 + X^\circ$ — $S_4 = 0111001$

$S_5(x) = X^6 + X^5 + X^4 + X$ — $S_5 = 1110010$

$S_6(x) = X + X^\circ$ — $S_6 = 0000011$

Les S_{k-1} bits de poids forts de S_6 (x) sont nuls. Donc on peut corriger C_6 (x) en ajoutant à ses 1 bits de poids faibles les S_1 bits de poids faibles de S_6 (x).

X°														X 14	
0	0	1	1	0	0	1	0	1	**0**	**1**	1	1	0	1	C' (x) erroné
1	0	0	1	1	0	0	1	0	1	**0**	**1**	1	1	0	C1 (x) "
0	1	0	0	1	1	0	0	1	0	1	**0**	**1**	1	1	C2 (x) "
1	0	1	0	0	1	1	0	0	1	0	1	**0**	**1**	1	C3 (x) "
1	1	0	1	0	0	1	1	0	0	1	0	1	**0**	**1**	C4 (x) "
1	1	1	0	1	0	0	1	1	0	0	1	0	1	**0**	C5 (x) "
0	**1**	1	1	0	1	0	0	1	1	0	0	1	0	1	C6 (x) "
1	1	0	0	0	0	0									S6 (x) "
1	0	1	1	0	1	0	0	1	1	0	0	1	0	A	C6 (x) corrigé

On décale C_6(x) corrigé de (n-6) positions, on retrouve ainsi le mot-code initial C_6 (x).

9.2. ALGORITHME CODAGE :

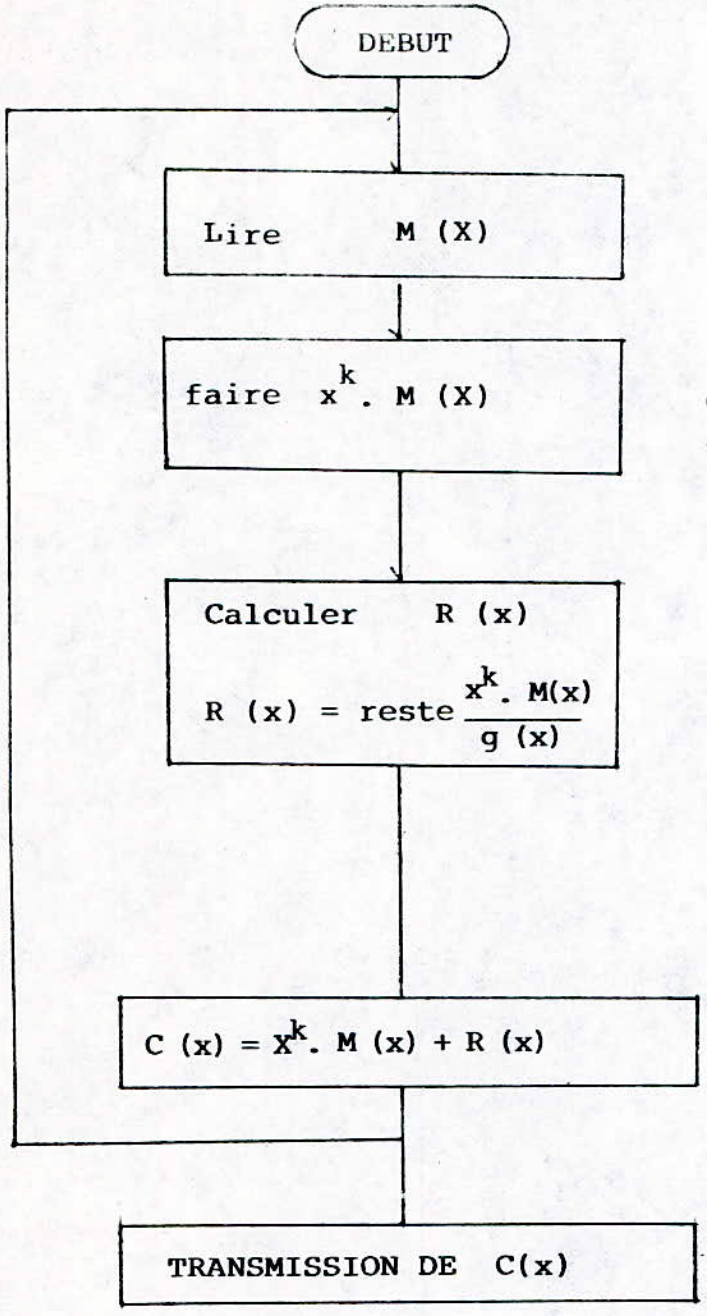
Soit le message utile M (x) , t q :

$$m(x) = a_{m-1} X^{m-1} + a_{m-2} + X^{m-2} + \dots + ao.$$

On multiplie M (x) par X^k, ce qui revient à le décaler de K positions vers la gauche, ou bien ajouter K zéros à la fin du mot M (x). On obtient ainsi : X^k M(x).

On divise X^k M (x) par le polynôme générateur g (x). Le reste de la division noté R (x) sera ajouté à X^k (x). On obtient ainsi le mot-code C (x) = X^k (M (x) + R (x).

9.3. Organigramme codage :

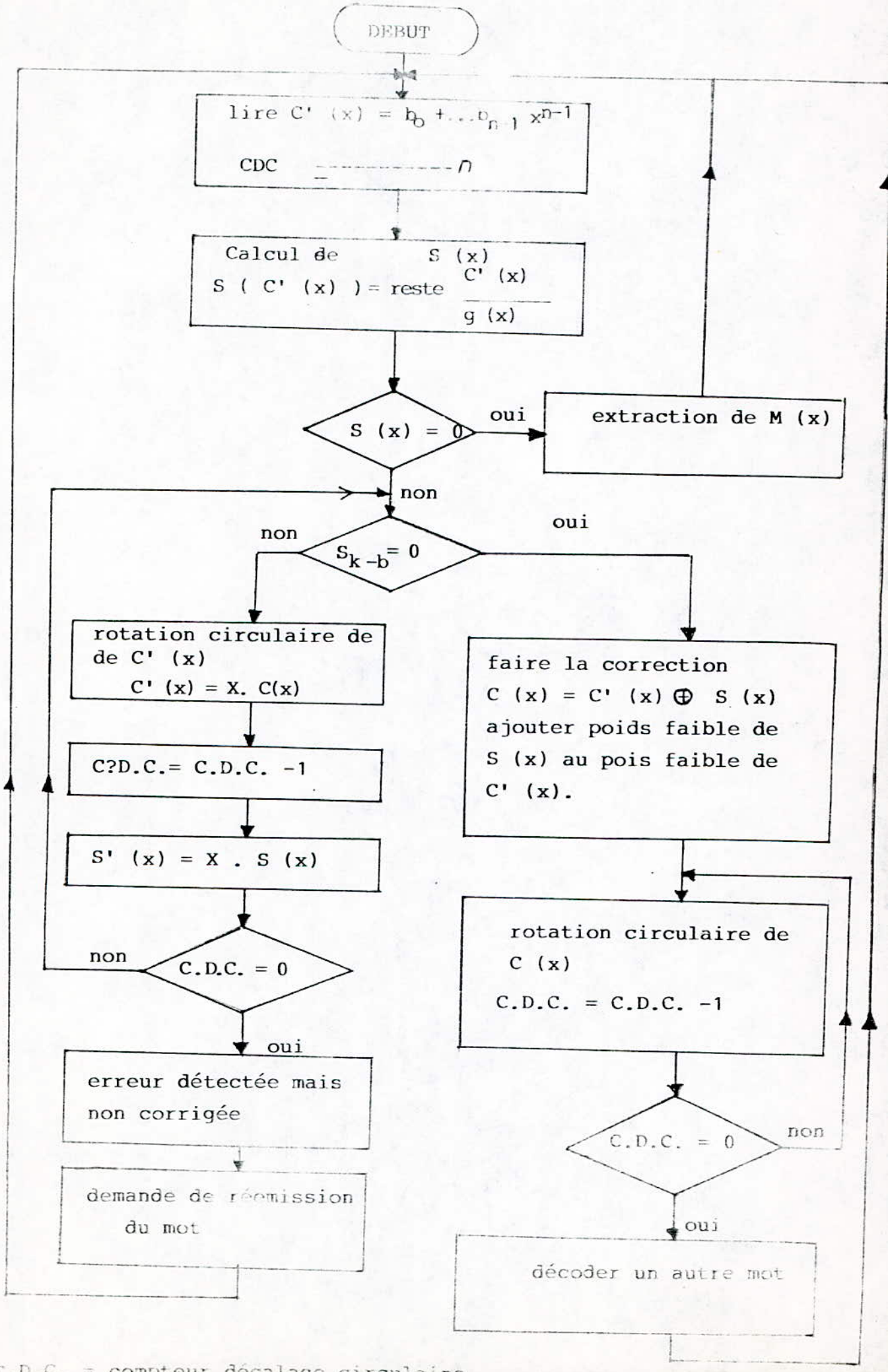


décalage de M (x) de k; position

calcul de la redondance

formation du mot -code

9.4. Organigramme décodage :



C.D.C. = compteur décalage circulaire

Les données sont transmises en parallèle suivant le schéma de la fig 2), constitué de 2 KITS 6802 05 de motorola, dont le premier KIT joue le rôle d'un émetteur et le second celui du récepteur.

Avant la transmission, les données sont stockées dans le port B du PIA du KIT " EMMETEUR". Dans ce KIT s'effectue tout le codage des données à transmettre.

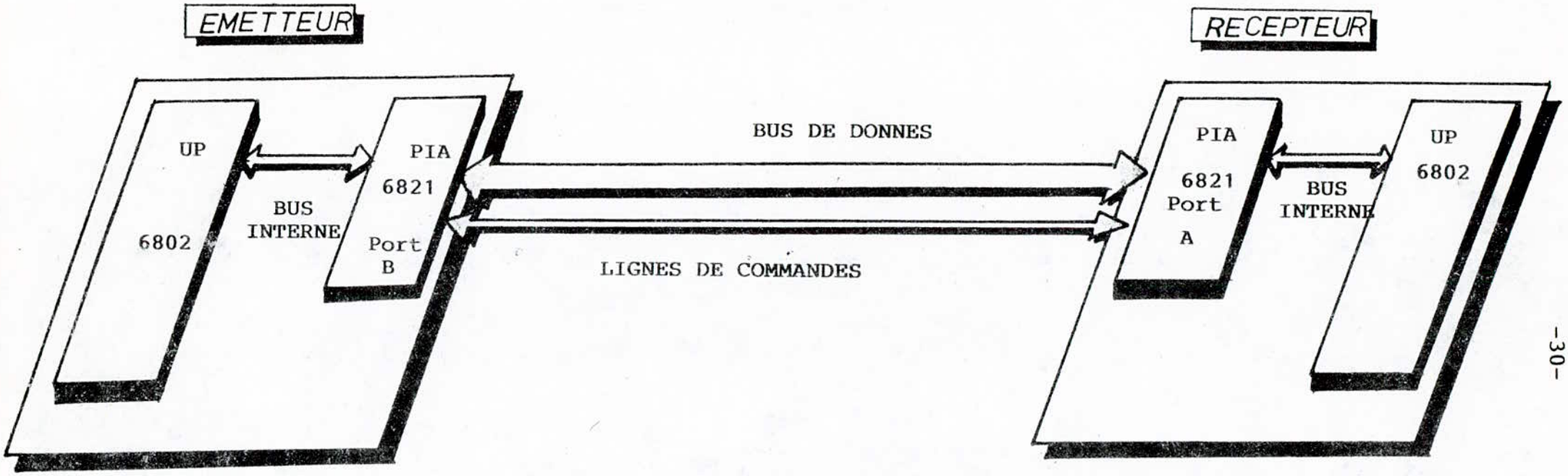
Après transmission, les données seront reçues dans le port A du PIA du KIT "RECEPTEUR".

Elles sont décodées et corrigées des erreurs dont elles sont affectées par un programme de décodage.

Ce même programme après décodage et correction d'eventuelles erreurs aura à extraire l'information utile du mot-code, c'est à dire éliminer la partie redondante.

REMARQUE :

L'introduction des erreurs se fait par silulation à l'aide d'un programme introduit dans l'émetteur.



Application du code de FIRE C(15,8)

fig. 2

LIAISONS : U 14

	émetteur		recepteur
PB0	(3)	----------	PA0 (20)
PB1	(4)	----------	PA1 (19)
PB2	(5)	----------	PA2 (18)
PB3	(6)	----------	PA3 (17)
PB4	(7)	----------	PA4 (16)
PB5	(8)	----------	PA5 (15)
PB6	(9)	----------	PA6 (1)
PB7	(10)	----------	PA7 (2)
CB1	(11)	----------	CA2 (21)
CB2	(12)	----------	CA1 (22)
GND	(13)	----------	GND (13)

10.1. ARCHITECTURE D'UN SYSTEME A MICROPROCESSEUR :

Un système à microprocesseur [FIG. 3] est composé d'un certain nombre de circuits intégrés : L SI , se sont :

- le microprocesseur,
- les mémoires vives : RAM,
- les mémoires mortes : ROM, PROM ou REPROM
- les interfaces d'entrée/sortie : PIA, ACIA, PTM, ...
- les périphériques : clavier, impimante, ...

10.1.1. STRUCTURE ET FONCTIONNEMENT D'UN MICROPROCESSEUR :

Schéma [VOIR FIG 4]

1. REGISTRES : stockent provisoirement les informations, ils sont constitués de bascules.

2. ACCUMULATEUR : c'est un registre à 8 bits. Il contient un opérande au début de l'opération, puis le résultat à la fin de l'opération .Il est à chargement parallèle à décalage à gauche où à droite. Il recoit des données de l'éxtérieur, il les transmet vers l'extérieur après traitement par l'UAL

3.COMPTEUR ORDINAL : Il contient l'adresse de la prochaine instruction à exécuter, son contenu est envoyé au registre d'adresse, L'éxécution d'une instruction l'incrémente automatiquement. Toutefois, il est possible, dans certains cas , d'aller se brancher à la première adresse d'un sous-programme, ou à une adresse bien définie.

4. REGISTRE D'INSTRUCTION : il sert au stockage provisoire de l'instruction provenant de la RAM ou ROM.

5. REGISTRE D'ETAT (CONDITION) :

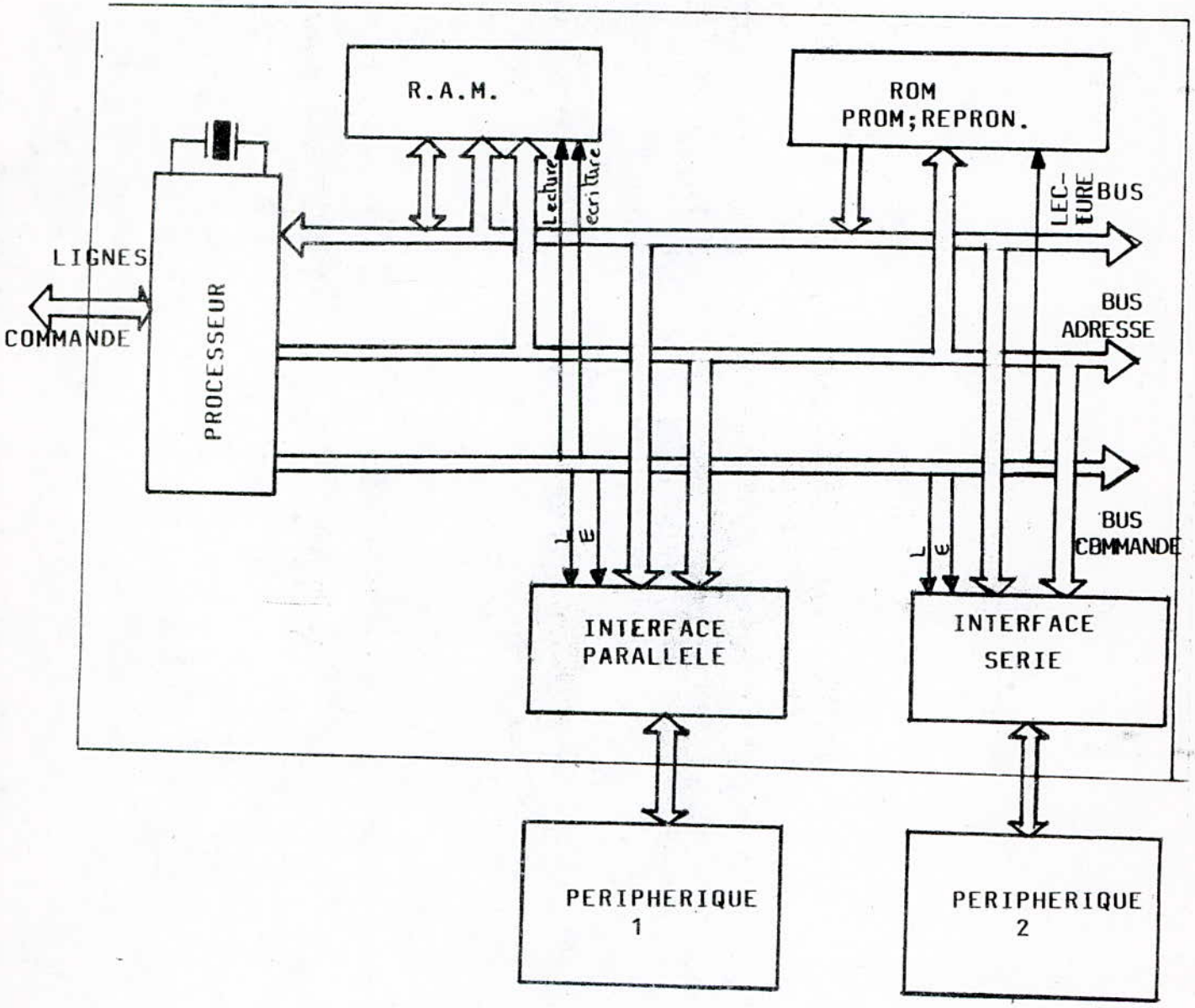
1	1	H	I	N	Z	V	C

Les indications C,V,Z,N,I et H donnent des informations précieuses sur les résultats des opérations mettant en jeu les accumulateurs et l'UAL.

REMARQUE : seuls les indicateurs C,Z et N sont utilisées fréquements

6. POINTEUR DE PILE :SP : le pointeur de pile est un registre 16 bits qui contient l'adresse de la position mémoire disponible dans la pile, zone réservée de la RAM pour sauvegarder les contenus des registres lorsque cela est nécessaire.

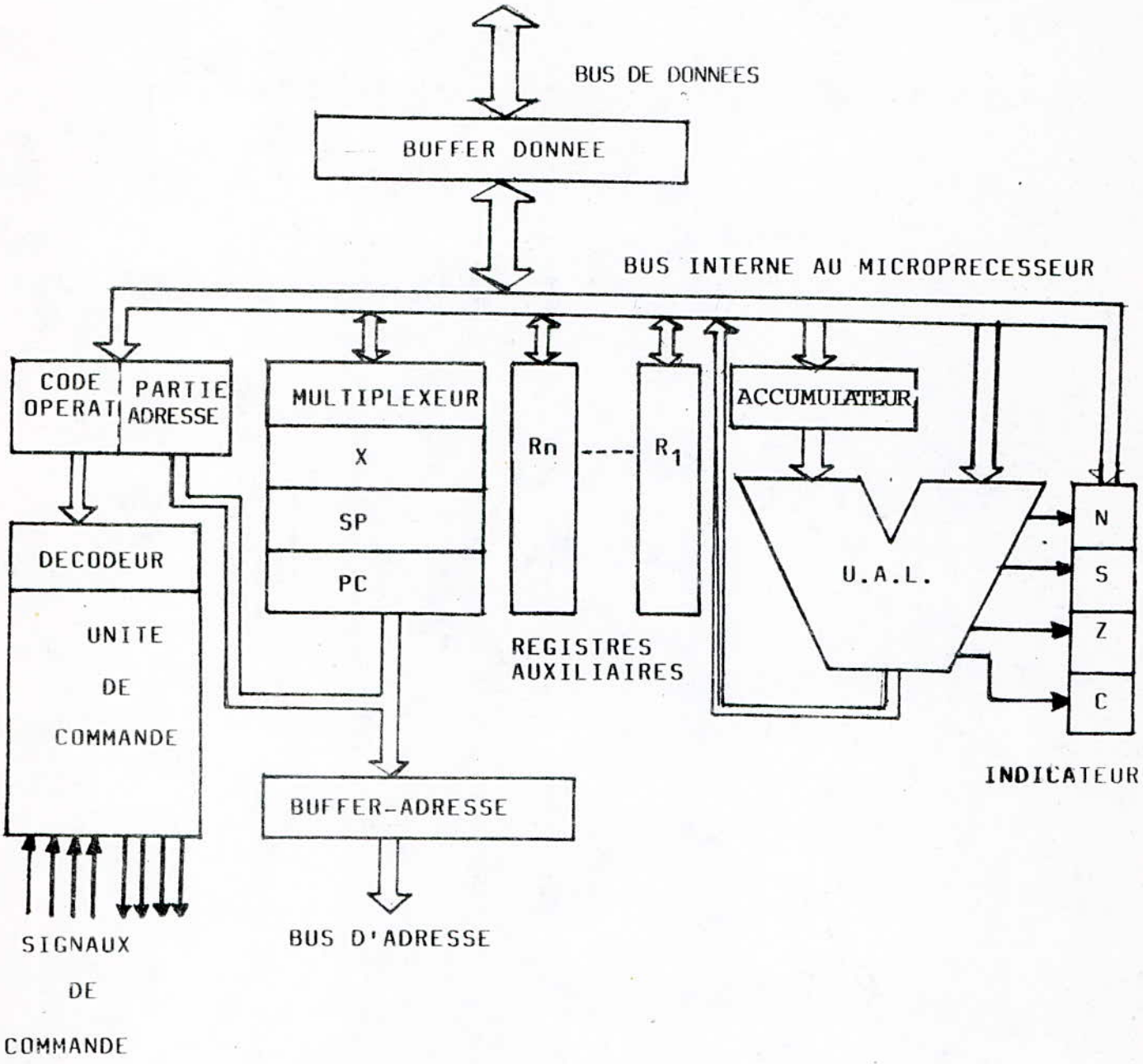
ARCHITECTURE D'UN SYSTEME A MICROPROCESSEUR :



SYSTEME A MICROPROCESSEUR

fig.3

1 STRUCTURE INTERNE DE LA CPU:



ARCHITECTURE STANDARD D'UN MICROPROCESSEUR

fig.4

7. REGISTRE D'INDEX : IX : c'est un registre très utile dans l'adressage indexé, il est de 16 bits. Dans ce mode d'adressage on ajoute la valeur de l'index au contenu du registre d'index pour obtenir l'adresse effective de la position-mémoire à laquelle doit accéder le microprocesseur.

8. U.A.L: ce module intégré dans le microprocesseur éxécute les opération arithmétiques et logiques, c'est une unité construite de logique cablée très complexe (VOIR REF 6).

9. UNITE DE COMMANDE : elle est capable de commander L'enchainement des différents cycles éxigés par l'instruction à traiter.

10. BUS DE DONNEES : il est bidirectionnel, par ce bus transitent les opérandes, les résultats de calculs, etc...

11. BUS D'ADRESSES : il est unidirectionnel, les adresses des mémoires sont transportées par ce bus .

12. BUS DE COMMANDE : il est destiné à commander les opérations d'E/S, telles que lecture-écriture, ...

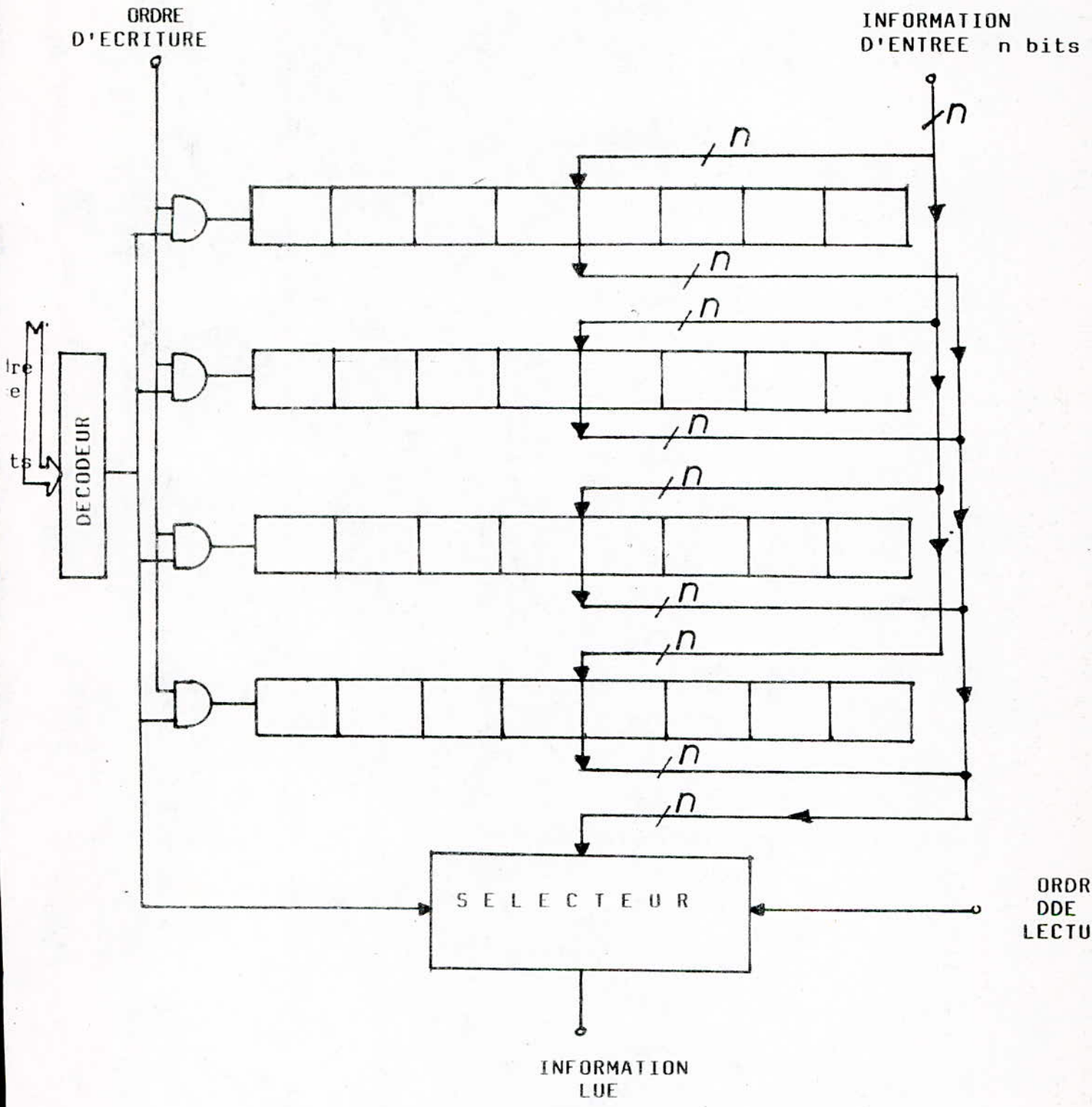
13. SIGNAUX DE COMMANDE : quelques signaux particuliers permettent de faire certaines commandes particulières du microprocesseurs (intialisation, commande d'attente...).

14. ALIMENTATIONS : selon les microprocesseurs, il existe une, deux ou trois alimentations plus la masse. Certains ne nécessitent que (+ 5V), donc compatible avec la TTL.

10.1.2. STRUCTURE DES MEMOIRES : (FIG. 5)

Une mémoire est une juxtaposition de positions-mémoire dont celles-ci sont réalisées à partir de bascules mémorisant des états logiques. Les paramètres caractéristiques d'une mémoire sont :

-sa capacité totale : nombre d'éléments binaires,
-son organisation : nombre de mots, nombre de bits par mot,
-son mode d'accès : oléatoire et séquentiel,
-sa vitesse : temps d'accès, durée d'obtention d'une information après une demande de lecture,
-temps de cycle : durée minimum séparant 2 appels successifs à la mémoire pour une lecture ou une écriture,
-son mode d'adressage : adressables, non-adressables.



STRUCTURE INTERNE D'UNE RAM

fig·5

Les mémoires se subdivisent en 3 groupes : mémoires vives, mémoires mortes, mémoires de masses.

Les mémoires vives sont appelées les RAM, elles permettent la lecture et l'écriture, elles peuvent êtres de technologie différentes bipolaires, MOS statiques, CMOS statiques, MOS dynamiques..., Les mémoires mortes ROM, PROM, REPROM, permettent uniquement la lecture elles peuvent être bipolaires ou MOS.

Les mémoires de masses sont le disque, bande magnétique etc... les mémoires sont adressables par positions-mémoires (4, 8 et 16 bits Une mémoire est généralement structurée en matrice de N lignes à n cellules chacune.

Chaque cellule correspond à un bit (1 ou 0 logique). Chaque ligne correspond à un mot. Dans une mémoire morte, l'ordre d'écriture est suprimé.

10.1.3. INTERFACES D'ENTREE-SORTIE

10.1.3.a. INTERFACE PARALLELE PROGRAMMABLE : PIA

1. DEFINITION :

Le PIA est un circuit d'interface entre le microprocesseur et une unité périphérique. La connexion avec les périphériques se fait avec 2 bus de données bidirectionnels et 4 lignes de commande PA0-7, PB0-7, CA1, CA2, CB1, CB2.

2. STRUCTURE INTERNE :

Le PIA est constitué de 2 registre tampons appelés port A et port B; de 2 registre de sens de transfert des données DDRA et DDRB; de 2 registres de commande CRA et CRB, enfin de 2 registres de sorite ORA et ORB (VOIR REF 6).. Les ports A et B peuvent être programmés en entrées ou en sorite.

3. SCHEMA DU PIA 6820 : (FIG 6).

Les lignes d'interface avec le microprocesseur sont :

-RS0, RS1 : lignes de selection du registre interne du PIA avec lequel le microprocesseur désire dialoguer, raccordées à 2 lignes du bus adresse.

-CS0, CS1, CS2 : lignes de selection du circuit, généralement raccoirdées aux lignes du bus adresse, le microprocesseur 6800 considérant le PIA comme une mémoire vive (RAM) qu'il peut adresser écririe ou lire.

-R/W : cette ligne directement connectée à la sortie de même nom du microprocesseur, va déterminer le sens du transfert des données sur le bus donnée.

-E : ligne d'activation du PIA, reliée à la phase $\emptyset_2$ de l'horloge (E constitue un signal de synchronisation).

-RESET : ligne de mise à l'état initial du PIA, reliée à la ligne de même nom du microprocesseur.

-IRQA, IRQB : lignes de commande d'interruption, raccordées directement à la ligne IRQ du microprocesseur (ou éventuellement NMI) ; elles permettent aux unités périphériques d'interrompre le programme en cours d'éxécution pour leur permettre dedialoguer avec le microprocesseur.

-CA1, CA2, CB1, CB2 : lignes de commande de la partie A et B du PIA.

10.1.3.b. INTERFACE SERIE -PROGRAMMABLE : ACIA

1. DEFINITION :

L'interface série ACIA est un circuit permettant de transformer des données parallèles en données série ou le contraire.

2. STRUCTURE INTERNE : (FIG 7).

L'ACIA est constitué de 4 registres, un registre de réception, un registre de transmission, un registre d'état et un autre de commande (REF 6).

Les lignes d'interface avec le microprocesseur sont :

-CSO, CS1, CS2 : lignes de selection boitier, raccordées au bus d'adresse et à VMA du microprocesseur,

-Do-D7 :lignes du bus de données,

-RS : ligne de selection du registre interne (avec l'aide de R/W)

-IRQ : ligne de commande d'interruption reliée à la ligne de même nom du microprocesseur.

-E: activation de l'ACIA

-T_x C : horloge transmission

-R x C : horloge reception

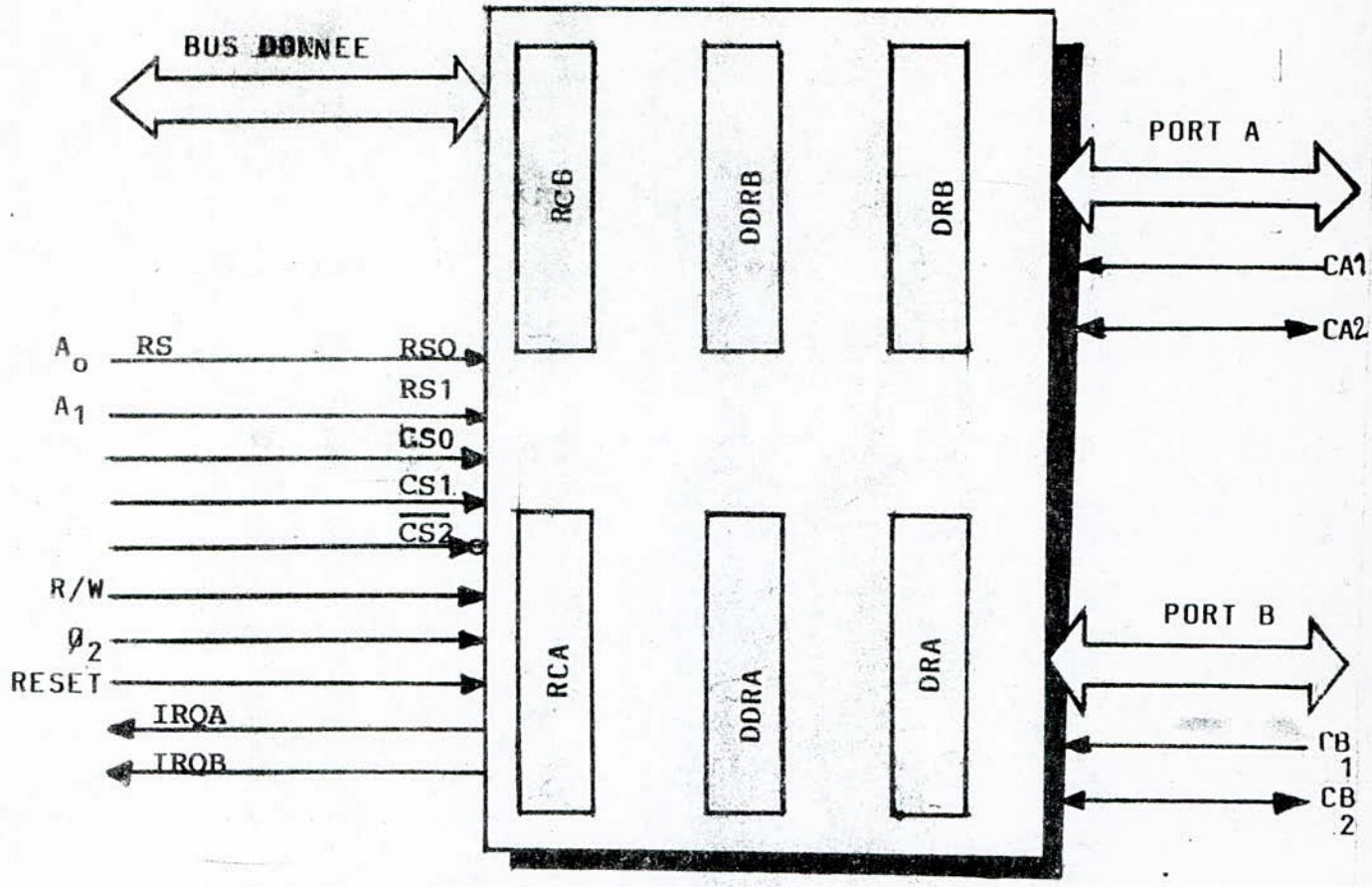
-T x D : ligne de transmission des données,

-R x D : ligne de réception des données,

-CTS : inhibition de l'éme heur,

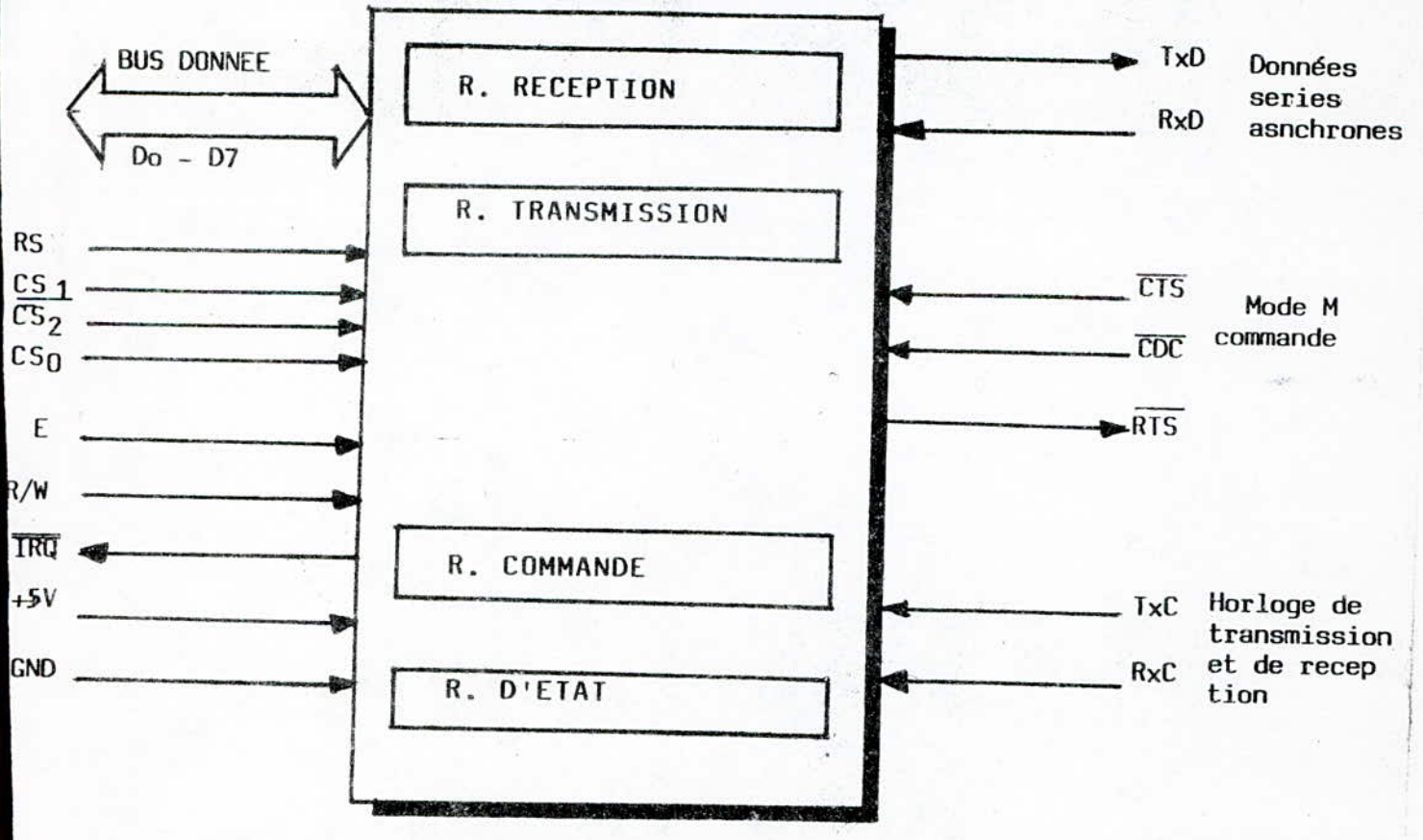
-RTS : demande d'emission

-DCD : perte de la porteuse de donnée.



INTERFACE P.I.A

FIG 6



INTERFACE A.C.I.A.

fig.7

CODE DANS LE TEST DES MEMOIRES :

METHODE DE TEST AVEC LE CODE FIRE :

La recherche dans le domaine des codes appliqués aux tests des systèmes à microprocesseur ne cesse de se développer.

En effet, un code détecteur d'erreur bien choisi permettre une plus grande rapidité de test ainsi qu'une grande rigueur; dans notre cas nous avons appliqué un code FIRE C (15,8) qui permet de détecter trois erreurs.

Ce test consiste à choisir un mot, de le coder suivant FIRE (voir code de FIRE appliqué à la transmission de données).
Le mot-code occupe 15 bits, il est stocké dans 2 positions -mémoires adjacentes. On relie le mot code ainsi stocké et on calcule son syndrome d'erreur.

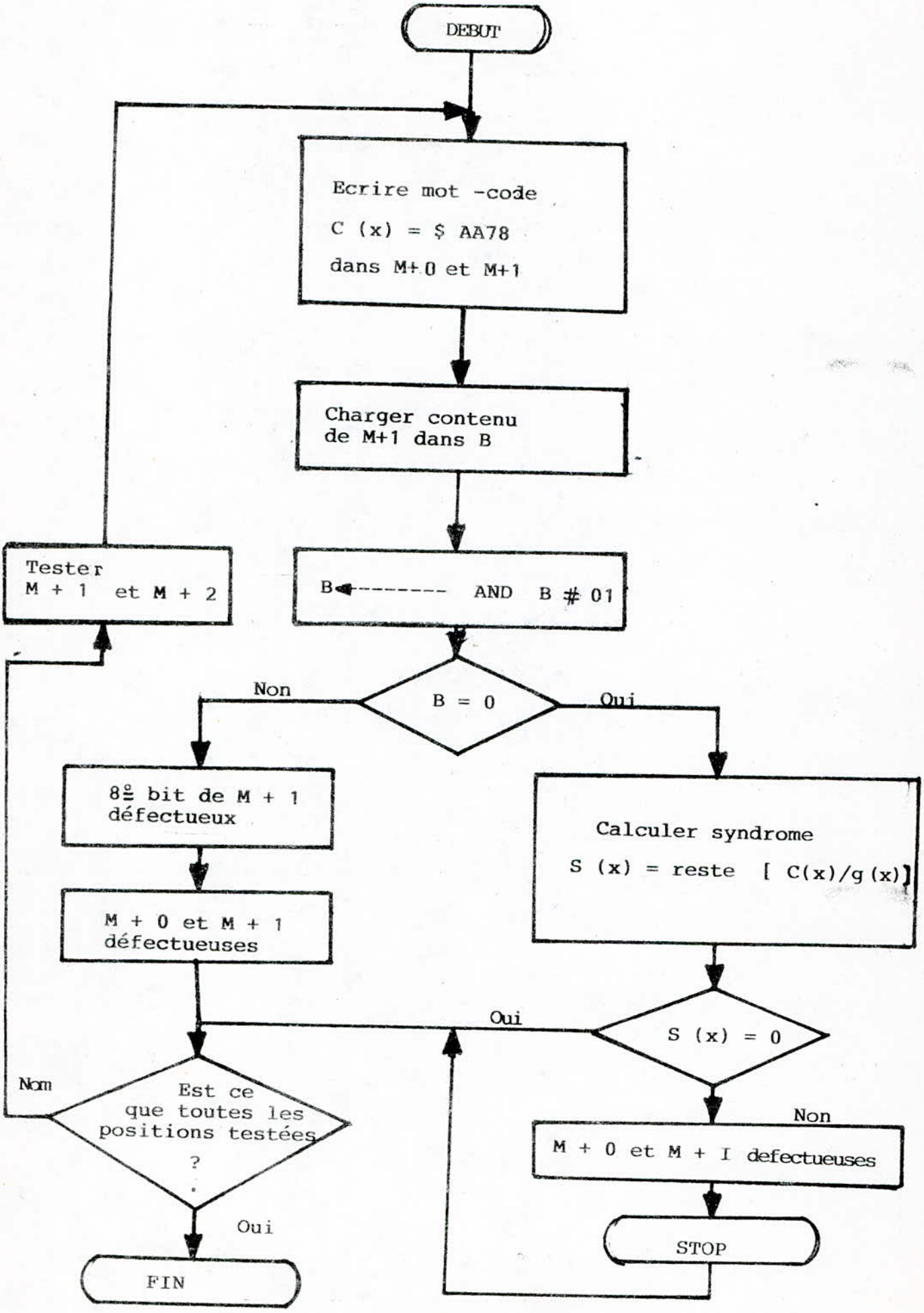
Si ce syndrome est nul, on déclarera les 2 positions comme étant correctes, sinon l'une des 2 positions ou les deux sontdeffectueuses.

De même on testera le 16 ième bit :

-Si les 2 positions-mémoires testées sont bonnes, on passe au test de la 2éme et la 3éme position-mémoire ainsi de suite.

-A chaque fois que le code détecte une erreur le programme s'arrête, on note la valeur du registre d'index [RX] et [RX + 1] et on relance le programme.

ORGANIGRAMME TEST RAM AVEC CODE DE FIRE C (15,8)



CODES DANS LE TEST DES MICROPROCESSEURS :

CODE REED-MULLER

Les codes de REED-MULLER sont des codes de groupe :

-la longueur M du code est une puissance de 2 , $n = 2^k$

-pour un entier positif r, le nombre des positions d'information es

$$m = \sum_{i=o}^{r} C_n^i \quad , \; r \leqslant m$$

-celui des positions de contrôle (n-m)

Le problème revient à adresser une matrice de dimension $K.2^k$ permettant par certains contrôles de parité de localiser les erreurs. Les conditions auxquelles doivent satisfaire les codes de REED-MULLER son

$$-n = 2^K$$

$$-m = \sum_{i=o}^{r} C_k^i$$

$$-n-m = \sum_{i=o}^{k-(r+1)} C_k^i$$

Pour former la matrice du code de REED-MULLER, on procède de la façon suivante :

-le premier vecteur ligne Xo a toutes ses composantes égales à 1, le deuxième X_Ka ses 2^{k-1} premières composantes égales à 0, les 2^{k-1} suivants égales à 1. Le troisième vecteur ayant les 2^{k-2} première com posantes égales à 0, les 2^{k-2} suivantes égales à 1 puis à nouveau 2^{k-2} égales à 0, enfin les 2^{k-2} derniers égales à 1.

On poursuit le classement des vecteurs lignes jusqu'au dernier vecteur X_1 dans lequel les coefficients 0 et 1 alternent régulièrement les (k+1) vecteurs, X_o à X_k , ainsi formés sont les vecteurs de base du code.

On poursuit alors la formation de la matrice ligne par ligne en effectuant d'abord les produits de tous les vecteurs de base 2 à 2 puis 3 à 3 et ainsi de suite jusqu'au produit des k vecteurs de base différents de Xo. On a ainsi constitué une matrice carrée d'ordre 2^k. tous les vecteurs lignes de cette matrice sont indépendantes.

On appelle code du r-ième ordre un code dans lequel on utilise les premiers vecteurs lignes X_o , X_k .... , X_1 les combinaisons des vecteurs de base 2 à 2 , 3 à 3 jusqu'à r x r.

L'intérêt de ces codes réside dans la simplicité du procédé permettant le décodage d'un mot code, c'est à dire la localisation des erreurs.

Pour un code donné du r-ième ordre on a fixé k et r, on a alors la longueur du mot code $n = 2^k$, le nombre de positions d'information :

$$m = \sum_{i=0}^{r} C_k^i$$

correspondant au nombre de lignes à retenir dans la matrice réduite de ce code. C'est un code $C(n,m)$

Exemple : formation de la matrice du code de REED-MULLER C (8,4).

$$\begin{array}{c} x_o \\ x_3 \\ x_2 \\ x_1 \\ x_3\ x_2 \\ x_3\ x_1 \\ x_2\ x_1 \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ici : k = 3, donc $n = 2^3 = 8$

si on fixe r = 1 on trouve m = 4

si $k = 4$, $n = 2^4 = 16$

La matrice correspondante est :

	x_o	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_4	x_3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
x_4	x_2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
x_4	x_1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
x_3	x_2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
x_3	x_1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
x_2	x_1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
x_4 x_3	x_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
x_4 x_3	x_1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
x_4 x_2	x_1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
x_3 x_2	x_1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}

-test de l'unité de commande,

-test de l'ensemble.

La connaissance de la structure interne (VOIR 10-1-1) nous oriente vers le choix du test des modules internes du microprocesseur. Dans ce test seuls les accumulateurs, le registre d'index, le registre d'état et le pointeur de pile seront testés.

Le programme counter (pc), l'unité de commande, l'unité arithmétique et logique ainsi que le registre d'instruction seront testés indirectement c'est à dire que le test porté sur les registres cités influe indirectement sur ces derniers en utilisant des opérations spéciales réalisées à partir des instructions .

Le registre d'instruction reçoitet conserve le premier octet de l'instruction définissant l'opération que va devoir éxécuter le microprocesseur. Si l'instruction est bien éxécutée on déduit que le registre d'instruction est en bon état.

Le programme counter contient l'adresse de l'octet présent sur le bus des données. Si l'adresse de cet octet est fausse alors le programme à exécuter sera erronné, on conclut alors que le programme counter est deffectueux.

L'unité de commande et l'UAL commandent et éxécutent toutes les opérations que doit effectuer le microprocesseur. Si ces opérations sont faussées, alors on peut conclure que l'unité de commande et/ou l'UAL sont deffectueuses.

On va exposer la méthode de test des registres déjà cités, en commencant par l'accumulateur A et le registre d'état puis l'accumulateur B et le registre d'état enfin par ordre : le registre d'index, le pointeur de pile. On termine par une opération arithmétique qui agit sur les accumulateurs A et B pour faire intervenir les modules non testés.

TEST INDIVIDUEL DES MODULES INTERNES DU MICROPROCESSEUR :

Dans ce test le microprocesseur est partagé en modules indépendants ou mutuellement dépendants. Chaque module est testé individuellement par introduction d'une séquence d'instructions ou parfois d'une partie des instructions, sans considérer les cycles élémentaires mis en oeuvre, ni les conditions dans lesquelles chaque cycle se déroule.

A chaque fois, on compare le résultat final du registre à une valeur déjà connue. Si le résultat est identique, on continue alors le test des autres registres. Sinon le registre en question est considéré comme défaillant et de ce fait le microprocesseur l'est aussi.

INCONVENIENTS:

1- pour un microprocesseur ayant un grand nombre de registres internes, la méthode devient difficilement utilisable et longue.

2- si le test final révèle que le microprocesseur est bon, on ne peut conclure que celui-çi l'est réellement car des erreurs peuvent être compensées et ne pas être detectées. Néanmoins il y a une grande probalité pour qu'il soit bon.

AVANTAGES:

1- c'est une méthode dont le coût de revient est faible,

2- elle ne demande pas de matériel supplémentaire.

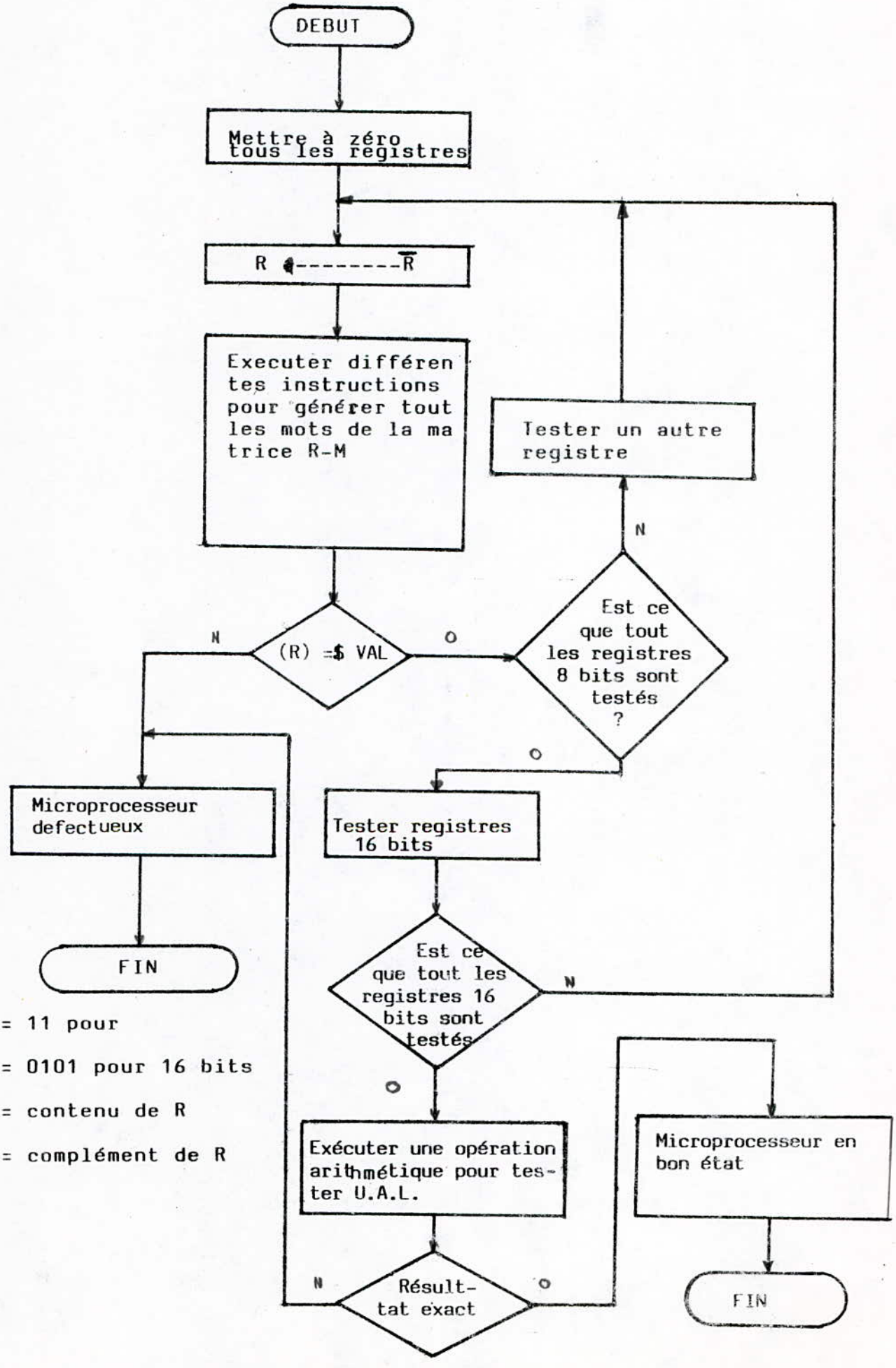
METHODE DE TEST AVEC LE CODE REED-MULLER :

La méthode de test du microprocesseur avec le code REED-MULLER consiste à générer la matrice du code. Pour celà, on doit introduire dans le registre à tester le premier mot de cette matrice, et à partir de celui-ci essayer de trouver tous les autres mots de la matrice.

Pour vérifier que la génération a bien eu lieu, on doit lire le dernier mot de la matrice qui doit être [$ 11] pour les registres 8 bits et [$ 0101] pour les registres 16 bits.

Si le test est positif, on décalera le registre, en bon état et qu'il suit bien le rythme du chargement qui lui est imposé si le test est négatif (dernier mot différent de [$ 11] ou [$ 0101], on déclarera alors le registre défectueux et partant de là le microprocesseur l'est aussi.

ORGANIGRAMME TEST MICROPROCESSEUR AVEC CODE -REED-MULLER :



$ VAL = 11 pour

$ VAL = 0101 pour 16 bits

(R) = contenu de R

$\overline{R}$ = complément de R

CONCLUSION :

L'objectif principal de notre travail était de montrer que la detection et la correction des erreurs dans la transmission de données entre système à microprocesseurs et la detection des pannes dans ces derniers était possible grâce à des codes détecteur et correcteur.

L'étude théorique, nous a montré que les codes cycliques sont bien adaptés à toutes les configurations d'erreurs, de même ils sont de réalisation simple.

Pour la transmission de données entre 2 kits 6802 D5E de MOTOROLA, nous avons fait un programme en assembleur. Ce programme détecte et corrige jusqu'à deux erreurs et detecte trois erreurs.

En ce qui concerne les tests de la RAM et de la CPU nous avons utilisé deux codes détecteurs.

Une bonne connaissance de l'assembleur du 6802 améliorera la rapiditié du codage et du décolage.

Une étude poussée de la distribution statistique des erreurs sur le canal de transmission permettrait de choisir un code efficace qui éliminerait les erreurs les plus fréquentes. En combinant plusieurs codes, l'efficacité de la protection augmentera.

Pour le test des système à microprocesseurs, des techniques avancées ont été crées dans le but d'exercerles circuits dans les plus mauvaises conditions, ces techniques reposent sur un matériel assez important et couteux.

ANNEXES

A1. TABLE DES POLYNOMES IRREDUCTIBLES :

La table suivante donne les polynomes irreductibles de degré ≤ 7, dans le corps de GALOIS C.G. (2). Ces polynômes peuvent être primitifs où non-primitifs, leurs racines sont liéairement dépendantes ou indépendantes.

DEGRE 1	$P_1(x) = X+1$	
DEGRE 2	$P_1(x) = X^2+X+1+$	
DEGRE 3	$P_1(x) = X^3+X+1$	
DEGRE 4	$P_1(x) = X^4+X+1$	$P_3(x) = X^4+X^3+X^2+X+1$
DEGRE 5	$P_1(x) = X^5+X^2+1$ $P_5(x) = X^5+X^4+X^2+X+1$	$P_3(x) = X^5+X^4+X^3+X^2+1$
DEGRE 6	$P_1(x) = X^6+X+1$ $P_5(x) = X^6+X^5+X^2+X+1$ $P_{11}(x) = X^6+X^5+X^3+X^2+1$	$P_3(x) = X^6+X^4+X^2+X+1$ $P_7(x) = X^6+X^3+1$
DEGRE 7	$P_1(x) = X^7+X^3+1$ $P_5(x) = X^7+X^4+X^3+X^2+1$ $P_9(x) = X^7+X^5+X^4+X^3+X^2+X+1$ $P_{11}(x) = X^7+X^6+X^4+X^2+1$ $P_{13}(x) = X^7+X+1$ $P_{19}(x) = X^7+X^6+X^3+X+1$ $P_{21}(x) = X^7+X^6+X^5+X^2+1$	$P_3(x) = X^7+X^3+X^2+X+1$ $P_7(x) = X^7+X^6+X^5+X^4+X^2+X+1$

A2. PRODUCTION DES $x^n + 1$ EN FACTEURS PREMIERS :

La construction des codes cycliques repose sur la décomposition des polynômes $(x^n + 1)$ en facteurs irréductibles sur CG (2) (VOIR REF. 2)

a-n pair : $n = 2^S \cdot P$

$x^n + 1 = x^{2^S \cdot P} + 1 = (x^P + 1)^{2^S} \bmod 2.$

on continuera la décomposition jusqu'à obtenir un exposant impair. On ne l'obtiendra jamais si $n = 2^S$, car $x^{2S} + 1 = (x+1)^{2S}$ donc l'étude se ramène au cas n impair.

a-n impair :

les n racines de $(x^n + 1)$ sont toutes distinctes, soit :

$$\beta, \beta^2, \beta^4, \ldots \beta^{2S}, \ldots, \beta^{2K-1} \quad \text{avec } \beta^{2^k} = \beta$$

pour que P (x) divise $(x^n + 1)$; il faut et il suffit que ses racines β soient identifiables à K racines α de $(x^n + 1)$

identifions β à α^P; $\quad 1 \leqslant P \leqslant (n-1)$

on doit avoir : $\beta = \alpha^P, \beta^2 = \alpha^{2P}, \ldots, \beta^{2^S} = \alpha^{2^S \cdot P}$

or $\alpha^{2^S \cdot P} = \alpha^r$, si r est le reste de la division de $2^S.P$ par n.

$$\boxed{2^S . P = n\,q + r \quad o < r < n-1}$$

La procédure à suivre est la suivante :

On prend d'abord P=1, on divise par n les puissances successives de 2. On s'arrête lorsqu'on obtient un reste déjà trouvé : les racines $\alpha^{r_1}, \alpha^{r_2}, \ldots,$ obtenues sont celles d'un polynôme cherché. On recommence l'opération pour une nouvelle valeur de P différente des restes (racines) déjà trouvés, etc...

Exemple :

n= 15 on a donc $(X^{15} + 1)$

p= 1 d'où : $2^S . 1 = 15\,q + r$

2^0 : 15---------- $r_1 = 1$

2^1 : 15---------- $r_2 = 2$

2^2 : 15---------- $r_3 = 4$

2^3 : 15---------- $r_4 = 8$

2^4 : 15---------- $r_5 = 1$

On s'arrête car le reste $r_5 = 1$ a été déjà trouvé

on a donc le polynôme $P_1(x)$ suivant :

$$P_1(x) = (x+\alpha^{r_1})(x+\alpha^{r_2})(x+\alpha^{r_3})(x+\alpha^{r_4})$$

$$P_1(x) = (x+\alpha)(x+\alpha^2)(x+\alpha^4)(x+\alpha^8)$$

Ce polynôme est de degré 4.

on ne prend pas P = 2 ,car on a déjà trouvé un reste égal à 2.

P= 3 $\qquad 2^S .3 = 15\ q + r$

$2^0.3$:15--------- $r_1 = 3$

$2^1.3$:15--------- $r_2 = 6$

$2^2.3$:15--------- $r_3 = 12$

$2^3.3$:15--------- $r_4 = 9$

$2^4.3$:15--------- $r_5 = 3$ déjà trouvé

Le polynôme $P_3(x)$ est le suivant, il est de degré 4 :

$$P_3(x) = (x+\alpha^3)\ (x+\alpha^6)\ (x+\alpha^9)\ (x+\alpha^{12})$$

P =5 $\qquad 2^S .5 = 15\ q+ r$

$2^0 .5$:15-------- $r_1 = 5$

$2^1 .5$:15 $\qquad r_2 = 10$

$2^2 .5$:15 $\qquad r_3 = 5$ déjà trouvé

$$P_5(x) = (x+\alpha^5)\ (x+\alpha^{10})$$

P =7 $\qquad 2^S .7 = 15\ q+ r$

$2^0 .7$:15------- $r_1 = 7$

$2 .7$:15------- $r_2 = 14$

$2 .7$:15------- $r_3 = 13$

$2 .7$:15------- $r_4 = 11$

$2 .7$:15------- $r_5 = 7$ déjà trouvé

$$P_7(x) = (x+\alpha^7)\ (x+\alpha^{11})\ (x+\alpha^{13})\ (x+\alpha^{14})$$

On a ainsi obtenu toutes les racines $\alpha^1, \alpha^2, \ldots \alpha^{14}$; avec 4 polynômes $P_1(x)$, $P_3(x)$, $P_5(x)$, $P_7(x)$

REMARQUE : soit P (x) de degré n décomposable sous la forme suivante :

$$P(x) = P_{q_1}(x)\ .\ P_{q_2}(x) \ldots\ P_{q_i}(x)$$

$P_{q_i}(x)$ est primitif si q_i et n sont premiers entre eux. Dans l'exemple précédent :

$$P_3(x) = (x+\alpha^3)\ (x+\alpha^6)\ (x+\alpha^9)\ (x+\alpha^{12})$$

$q_i = 3$ et $n = 15$

q_i et n ne sont pas premiers entre eux donc P_3 (x) est non primitif

Soit $\frac{n}{q_i} = 5$ donc P_3 (x) divise un polynôme $(X^5 + 1)$

$(X^5 + 1) / (X + 1) = X^4 + X^3 + X^2 + X + 1$

d'où : $\boxed{P_3 (x) = X^4 + X^3 + X^2 + X + 1}$

P_3 (x) est irréductible et non- primitif

Avec le même procédé que pour P_3 (x), on retrouve :

$$\boxed{P_5 (x) = X^2 + X + 1}$$

P_5 (x) est irréductible et non- primitif

Pour P_7(x) et P_1 (x) ; ils doivent être irréductibles :

$q_i (P_7 (x)) = 7$ et $q_i (P_1 (x)) = 1$ avec n = 15

Ils sont premiers entre eux . Donc P_7 (x) et P_1 (x) sont primitifs :

Soit : $P_1 (x) = X^4 + a'X^3 + bX^2 + CX + 1$

$P_7 (x) = X^4 + a'X^3 + b'X^2 + C'X + 1$

ont démontre que :

$$P_7 (X) = X^4 \; P_1 (-\frac{1}{x})$$

d'où $C' = a$ $C = a'$ et $b = b'$

$P_1 (x) = X^4 + aX^3 + bX^2 + C + 1$

$P_7 (x) = X^4 + CX^3 + bX^2 + AX + 1$

pour déterminer (a,b,c) on procède de la façon suivante :

a b c	P_1 (x)	P_7 (x)	Observations
0 0 0	$X^4 + 1$	$X^4 + 1$	EXCLUS (VOIR Rg 1)
0 0 1	$X^4 + X + 1$	$X^4 + X^3 + 1$	
0 1 0	$X^4 + X^2 + 1$	$X^4 + X^2 + 1$	EXCLUS (VOIR Rg 2)
0 1 1	$X^4 + X^2 + X + 1$	$X^4 + X^3 + X^2 + 1$	EXCLUS (VOIR Rg 1)
1 0 0	$X^4 + X^3 + 1$	$X^4 + X + 1$	
1 0 1	$X^4 + X^3 + X + 1$	$X^4 + X^3 + X + 1$	EXCLUS (VOIR Rg 1)
1 1 0	$X^4 + X^3 + X^2 + 1$	$X^4 + X^2 + X + 1$	(VOIR Rg 1)
1 1 1	$X^4 + X^3 + X^2 + X + 1$	$X^4 + X^3 + X^2 + X + 1$	EXCLUS (voir rg 3)

donc finalement p_1 (x) et p_7 (x) seront choisis comme suit :

$P_1 (x) = X^4 + X + 1$
$P_7 (x) = X^4 + X^3 + 1$

où

$P_1 (x) = X^4 + X^3 + 1$
$P_7 (x) = X^4 + X + 1$

.

Ils sont irréductibles et de degré 4 et primitifs :

REMARQUE 1 :
On exclut les polynomes à nombres de termes pairs, car la somme de leurs coefficients est nulle ; ce qui signifie qu'ils sont divisibles par (x+1) et par conséquent non irreductibles

REMARQUE 2 : on exclut les polynômes non distincts

REMARQUE 3 : on exclut les polynômes déjà trouvés.

REMARQUE 4 : chaque polynôme P_{qi} (x) génère un code particulier
le polynôme $(X^n + 1) = X^{15} + 1$ s'écrit comme suit :

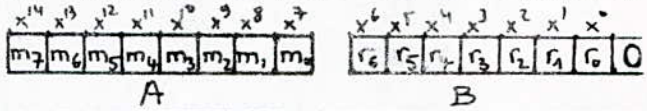
$X^{15} + 1 = (X+1)\ (X^4 + X + 1)\ (X^4 + X^3 + X^2 + X + 1)\ (X^2 + X + 1)\ (X^4 + X^3 + 1)$

$X^{15} + 1 = P_o (x) \cdot P_1 (x) \cdot P_3 (x) \cdot P_5 (x) \cdot P_7 (x)$

Tous les produits des polynômes trouvés peuvent engendrer des codes cycliques C (n.m) et ceci suivant certaines propriétés caractéristiques de chaque code cyclique qu'on veut construire, autrement dit la structure du polynôme générateur caractèrise le code cyclique qu'il détermine.

Programme CODAGE-EMISSION

E000	7F	E483		CLR		E483	Programmation du Port B du PIA en sortie
E003	86	FF		LDA	A	# FF	
E005	B7	E482		STA	A	E482	
E008	86	36		LDA	A	# 36	Programmation du registre de commande du PIA Port B
E00A	B7	E483		STA	A	E483	
E00D	CE	0000		LDX		#0000	Chargement de la 1ère adress de la table à transmettre
E010	7D	E483	1	TST		E483	Est-ce que le recepteur est Prêt à recevoir un mot-code?
E013	2A	FB		BPL	1		
E015	86	07		LDA	A	# 07	initialisation du compteur de décalage pour former $x^7 . M(x)$
E017	B7	E100		STA	A	E100	
E01A	E6	00		LDA	B	00,X	Formation de $x^7 . M(x)$
E01C	4F			CLR	A		on charge le mot-à-coder dans l'accumulateur B, on decale B à gauche, d'où le bit de poids fort de B passe dans Le carry une rotation de A, fait passer C dans Le bit de poids faible de A
E01D	58		2	ASL	B		
E01E	49			ROL	A		
E01F	7A	E100		DEC		E100	
E022	26	F9		BNE	2		
E024	36			PSH	A		
E025	86	08		LDA	A	# 08	initialisation du compteur de bits Pour la division avec addition modulo 2.
E027	B7	E100		STA	A	E100	
E02A	32			PUL	A		
E02B	4D		3	TST	A		Division.
E02C	2A	02		BPL	4		si le bit de poids fort de f est egal à 1, on fait une addition modulo 2 entre A et le polynome générateur $g(x) = E7\ (x^7 x^6 x^5 \ldots x^2 x\ 1)$ sinon on decale $x^7 . M(x)$ à la fin de la division R(x) est dans A.
E02E	88	E7		EOR	A	# E7	
E030	58		4	ASL	B		
E031	49			ROL	A		
E032	7A	E100		DEC		E100	
E035	26	F4		BNE	3		
E037	4D			TST	A		
E038	2A	02		BPL	5		
E03A	88	E7		EOR	A	# E7	
E03C	16		5	TAB			On forme le mot-code dans Ac
E03D	58			ASL	B		
E03E	A6	00		LDA	A	00,X	
E040	01			NOP			simulation de l'erreur
E041	01			NOP			AND A # [erreu
E042	01			NOP			AND B # [erre
E043	01			NOP			



E044	B7	E482		STA	A	E482	on charge le registre de donnée du PIA B, avec le contenu de A, On avertit le recepteur qu'il ya une donnée et ceci en portant CB2 à "1", ce qui entraine que le bit b7 du RCPIA A du recepteur se met à 1, on remet CB2 bas, une lecture du RDPIAB remet automatiquement b7 à 0
E047	B6	E482		LDA	A	E482	
E04A	B6	E483		LDA	A	E483	
E04D	8A	08		ORA	A	# 08	
E04F	B7	E483		STA	A	E483	
E052	84	F7		AND	A	# F7	
E054	B7	E483		STA	A	E483	
E057	7D	E483	6	TST		E483	est-ce-que le recepteur est prêt à recevoir le 2° octet du mot-code?
E05A	2A	FB		BPL	6		
E05C	F7	E482		STA	B	E482	on refait le même procédé que précédement.
E05F	F6	E482		LDA	B	E482	
E062	B6	E483		LDA	A	E483	
E065	8A	08		ORA	A	# 08	
E067	B7	E483		STA	A	E483	
E06A	84	F7		AND	A	# F7	
E06C	B7	E483		STA	A	E483	
E06F	08			INX			on passe au un autre mot à coder
E070	8C	0080		CPX		#0080	
E073	26	9B		BNE	1		
E075	3F			SWI			

Programme RECEPTION-DECODAGE

E000	7F	E481		CLR		E481	Programmation du Port A du PIA en entrée.
E003	86	00		LDA	A	#00	
E005	B7	E480		STA	A	E480	
E008	86	36		LDA	A	#36	Programmation du RC du PIA Port A
E00A	B7	E481		STA	A	E481	
E00D	CE	0000		LDX		#0000	chargement de la 1° adresse de la table de rangement.
E010	B6	E481	1	LDA	A	E481	avertir emetteur : Recepteur Prêt à recevoir Une donnée.
E013	8A	08		ORA	A	#08	
E015	B7	E481		STA	A	E481	
E018	84	F7		AND	A	#F7	
E01A	B7	E481		STA	A	E481	
E01D	7D	E481	2	TST		E481	est-ce-qu'il y a une donnée presente sur le RD du PIA A ?
E020	2A	FB		BPL	2		
E022	B6	E480		LDA	A	E480	rangement du 1° octet du mot-code reçu.
E025	A7	00		STA	A	00,X	
E027	B6	E481		LDA	A	E481	Prêt à recevoir 2° octet.
E02A	8A	08		ORA	A	#08	
E02C	B7	E481		STA	A	E481	
E02F	84	F7		AND	A	#F7	
E031	B7	E481		STA	A	E481	
E034	7D	E481	3	TST		E481	est-ce-que le 2° octet est present sur le RD du PIA A ?
E037	2A	FB		BPL	3		
E039	B6	E480		LDA	A	E480	rangement du 2° octet du mot-code reçu
E03C	A7	01		STA	A	01,X	
E03E	86	08		LDA	A	#08	initialisation du compteur de bit pour la division.
E040	B7	E100		STA	A	E100	
E043	A6	00		LDA	A	00,X	Positionnement du mot-code reçu pour le decodage.
E045	E6	01		LDA	B	01,X	
E047	44			LSR	A		
E048	56			ROR	B		
E049	4D		4	TST	A		Division avec Addition mod(2)
E04A	2A	02		BPL	5		on calcul le syndrome
E04C	88	E7		EOR	A	#E7	$S(x) = [c'(x) / g(x)]$
E04E	59		5	ROL	B		
E04F	49			ROL	A		
E050	7A	E100		DEC		E100	$c'(x)$: mot-code reçu
E053	26	F4		BNE	4		$g(x)$: polynome générateur.
E055	4D			TST	A		
E056	2A	02		BPL	6		
E058	88	E7		EOR	A	#E7	

E05A	27	36	6	BEQ	10		si le syndrome est nul, l'erreur est: - soit inexistante ($E(x)=0$) - soit non décelable
E05C	C6	11		LDA	B	# 11	initialisation du compteur
E05E	F7	E100		STA	A	E100	de décalage.
E061	34			DES			
E062	16		7	TAB			
E063	84	FC		AND	A	# FC	Test des S_{k-b} bits de poids fort
E065	27	15		BEQ	8		de $S(x)$ $[S_{k-b}=0]$
E067	32			PUL	A		Rappel du Registre Condition.
E068	06			TAP			
E069	69	01		ROL		01, X	Rotation circulaire du mot-code
E06B	69	00		ROL		00, X	reçu. Si on fait 17 rotations et
E06D	7A	E100		DEC		E100	que le test précédent n'est pas
E070	27	20		BEQ	10		vérifié, le mot ne sera pas corriger
E072	07			TPA			sauvegarde du Registre
E073	36			PSH	A		Condition.
E074	17			TBA			calcul $S'(x) = X\,S(x)$
E075	48			ASL	A		modulo $g(x)$
E076	2A	EA		BPL	7		$g(x) = E7$
E078	88	E7		EOR	A	# E7	
E07A	20	E6		BRA	7		
E07C	69	01	8	ROL		01, X	Rotation circulaire de C(x) pour
E07E	69	00		ROL		00, X	le corriger.
E080	E8	01		EOR	B	01, X	correction du mot code reçu, on
E083	E7	01		STA	B	01, X	ajoute S_{k-b} aux 2e octet du mot-code.
E085	7A	E100		DEC		E100	
E088	27	09		BEQ	10		
E08B	69	01	9	ROL		01, X	
E08B	69	00		ROL		00, X	Rotation circulaire du mot corrigé
E08D	7A	E100		DEC		E100	pour retrouver le message
E090	26	F7		BNE	9		initiale.
E092	08			INX			
E093	8C	0080		CPX		# 0080	On passe a un autre mot.
E096	27	03		BEQ		FIN	
E098	7E	E010		JMP		E010	
E09B	3F			SWI			

TEST RAM AVEC FIRE C(15,8)

0000	CE	E000		LDX		#E000	adresse de la 1ère Position à tester
0003	86	AA		LDA	A	M(X)	chargement du mot code
0005	C6	78		LDA	B	R(X)	C(X)= M(X)+R(X) = AA 78
0007	A7	00	DEBUT	STA	A	00,X	chargement de mot code dans
0009	E7	01		STA	B	01,X	(M+0) et (M+1)
000B	E6	01		LDA	B	01,X	test du bit b0 de [M+1]
000D	C4	01		AND	B	#01	s'il est different de zero, la
000F	26	1D		BNE	FIN		position est defectueuse.
0011	A6	00		LDA	A	00,X	lecture du mot se trouvant
0013	E6	01		LDA	B	01,X	dans M+0 et M+1.
0015	44			LSR	A		
0016	56			ROR	B		
0017	FF	E417		STX		E417	sauvegarde du RX.
001A	CE	0008		LDX		#0008	Le RX est utilisé comme compteur de bits pour la division.
001D	4D		1	TST	A		Division avec addition
001E	2A	02		BPL	2		modulo 2.
0020	88	E7		EOR	A	#E7	(Voir Transmission de donné
0022	59		2	ROL	B		
0023	49			ROL	A		
0024	09			DEX			
0025	26	F6		BNE	1		
0027	4D			TST	A		
0028	2A	02		BPL	3		
002A	88	E7		EOR	A	#E7	
002C	27	01	3	BEQ	4		
002E	3F		FIN	SWI			si le Programme s'arrête la il y a une Position defectueuse. on déclare alors M+0 et M+1 defec
002F	FE	E417	4	LDX		#E417	on passe au test de M+1 et M+
0032	08			INX			
0033	8C	E3FF		CPX		#E3FF	
0036	26	CB		BNE		DEBUT	
0038	3F			SWI			

TEST MICROPROCEUR AVEC CODE R-M

```
E000  4F          CLR  A
E001  26 3E       BNE  DEFAUT
E003  43          COM  A
E004  27 3B       BEQ  DEFAUT
E006  0C          CLC
E007  46          ROR  A
E008  47          ASR  A
E009  47          ASR  A
E00A  47          ASR  A
E00B  81 0F       CMP  A  #0F
E00D  26 32       BNE  DEFAUT
E00F  0D          SEC
E010  46          ROR  A
E011  46          ROR  A
E012  0C          CLC
E013  46          ROR  A
E014  47          ASR  A
E015  46          ROR  A
E016  47          ASR  A
E017  46          ROR  A
E018  46          ROR  A
E019  81 33       CMP  A  #33
E01B  26 24       BNE  DEFAUT
E01D  0C          CLC
E01E  88 66       EOR  A  #66
E020  81 55       CMP  A  #55
E022  26 1E       BNE  DEFAUT
E024  0C          CLC
E025  88 66       EOR  A  #66
E027  84 03       AND  A  #03
E029  81 03       CMP  A  #03
E02B  26 15       BNE  DEFAUT
E02D  0C 02       CLC
E02F  8B 05       ADD  A  #02
E031  81 0D       CMP  A  #05
E033  26          BNE  DEFAUT
E034  0D          SEC
E035  46          ROR  A
E036  0C          CLC
E037  46          ROR  A
E038  46          ROR  A
E039  47          ASR  A
E03A  47          ASR  A
E03B  0D          SEC
E03C  49          ROL  A
E03D  81 11       CMP  A  #11
E03F  27 01       BEQ  suite 1
E041  3F    DEFAUT SWI
```

Le test du microprocesseur consiste à faire fonctionner celui-ci dans les condition les plus difficiles.

La matrice génératrice du code Reed-Muller, fait travailler le microprocesseur dans des conditions defavorable. En effet, pour la créer, partant du vecteur 00, et en utilisant diverses instructions du microprocesseurs.

Celui-ci doit supporter de changement de valeur très fréquent

- Complémentation.
- Plusieurs decalages à gauche, puis un à droit
- des décrémentation suivi d'incrémentation.
- des sauts frequents
- ....

Le test se fait registre aprés registre, certains modules du microprocesseurs ne peuvent être actionnés directement par des instructions, (Registre conditions, SP, U.A.L., U.C.) Ces parties seront testés indirectements, c'est à dire en faisant fonction certains registre (ACCU, RX,.. l'etat physique de ces partie influerait sur le contenu de ces registres.

Dans cette 1ère Partie, on teste l'accumulateur A, en lui faisant subir plusieurs opérations dans le but de générer la matrice R-M

Si un des vecteurs n'est pas trouvé, il y a donc un défaut dans l'accumulateur A le registre condition est testé parallelement à A Si l'accumulateur A est en bon etat, on passe au test de l'accumulateur B, sinon le Prog. se

E042	5F		suite1	CLR	B			
E043	26	3E		BNE		DEFAUT		Test de l'accumulat
E045	53			COM	B			1 1 1 1 1 1 1 1
E046	27	3B		BEQ		DEFAUT		
E048	OC			CLC				
E049	56			ROR	B			
E04A	57			ASR	B			
E04B	57			ASR	B			
E04C	57			ASR				
E04D	C1	OF		CMP	B	# OF		0 0 0 0 1 1 1 1
E04F	26	32		BNE		DEFAUT		
E051	OD			SEC				
E052	56			ROR	B			
E053	56			ROR	B			
E054	OC			CLC				
E055	56			ROR	B			
E056	57			ASR	B			
E057	56			ROR	B			
E058	57			ASR	B			
E059	56			ROR	B			
E05A	56			ROR	B			
E05B	C1	33		CMP	B	# 33		0 0 1 1 0 0 1 1
E05D	26	24		BNE		DEFAUT		
E05F	OC			CLC				
E060	C8	66		EOR	B	# 66		
E062	C1	55		CMP	B	# 55		0 1 0 1 0 1 0 1
E064	26	1E		BNE		DEFAUT		
E066	OC			CLC				
E067	C8	66		EOR	B	# 66		
E069	C4	03		AND	B	# 03		
E06B	C1	03		CMP	B	# 03		0 0 0 0 0 0 1 1
E06D	26	15		BNE		DEFAUT		
E06F	OC			CLC				
E070	CB	02		ADD	B	# 02		
E072	C1	05		CMP	B	# 05		0 0 0 0 0 1 0 1
E074	26	OD		BNE		DEFAUT		
E076	OD			SEC				
E077	56			ROR	B			
E078	OC			CLC				
E079	56			ROR	B			
E07A	56			ROR	B			
E07B	57			ASR	B			
E07C	57			ASR	B			
E07D	OD			SEC				
E07E	59			ROL	B			
E07F	C1	11		CMP	B	#11		0 0 0 1 0 0 0 1
E081	27	01		BEQ		suite 2		
E083	3F		DEFAUT	SWI				

```
E084  17          Suite2  TBA
E085  29 05               BVS     DEFAUT   Test de l'indicateur V (over
E087  5F                  CLB     B
E088  2B 02               BMI     DEFAUT   Test de l'indicateur N (Sig
E08A  20 01               BRA     suite3
E08C  3F          DEFAUT  SWI
E08D  5F          Suite3  CLR     B              TEST du RX.
E08E  4F                  CLR     A        génération de la matrice R-
E08F  CE 00 00            LDX     #0000
E092  CE FF FF            LDX     #FFFF    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
E095  8B 01         1     ADD     A  #01
E097  C9 00               ADC     B  #00
E099  09                  DEX
E09A  8C 00 FF            CPX     #00FF    0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
E09D  26 F6               BNE     1
E09F  81 00               CMP     A  #00
E0A1  26 5E               BNE     DEFAUT
E0A3  C1 FF               CMP     B  #FF
E0A5  26 5A               BNE     DEFAUT
E0A7  8B 01         2     ADD     A  #01
E0A9  C9 00               ADC     B  #00
E0AB  08                  INX
E0AC  8C 0F 0F            CPX     #0F0F    0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
E0AF  26 F6               BNE     2
E0B1  81 10               CMP     A  #10
E0B3  26 4C               BNE     DEFAUT
E0B5  C1 0D               CMP     B  #0D
E0B7  26 48               BNE     DEFAUT
E0B9  8B 01         3     ADD     A  #01
E0BB  C9 00               ADC     B  #00
E0BD  08                  INX
E0BE  8C 33 33            CPX     #3333    0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
E0C1  26 FC               BNE     3
E0C3  81 34               CMP     A  #34
E0C5  26 3A               BNE     DEFAUT
E0C7  C1 31               CMP     B  #31
E0C9  26 36               BNE     DEFAUT
E0CB  8B 01         4     ADD     A  #01
E0CD  C9 00               ADC     B  #00
E0CF  08                  INX
E0D0  8C 55 55            CPX     #5555    0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
E0D3  26 F6               BNE     4
E0D5  81 56               CMP     A  #56
E0D7  26 28               BNE     DEFAUT
E0D9  C1 53               CMP     B  #53
E0DB  26 25               BNE     DEFAUT
E0DD  4F                  CLR     A
E0DE  5F                  CLR     B
E0DF  8B 01         5     ADD     A  #01
E0E1  C9 00               ADC     B  #00
E0E3  09                  DEX
E0E4  8C 00 0F            CPX     #000F    0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
E0E7  26 F6               BNE     5
```

E0E7	81	46		CMP	A	#46	
E0E9	26	14		BNE		DEFAUT	
E0EB	C1	55		CMP	B	#55	
E0ED	8B	01		ADD	A	#01	
E0EF	C9	00		ADC	B	#00	
E0F1	08			INX			
E0F2	8C	0033		CPX		#0033	0000000000110011
E0F5	26	F6		BNE	6		
E0F7	81	6A		CMP	A	#6A	
E0F9	26	04		BNE		DEFAUT	
E0FB	C1	55		CMP	B	#55	
E0FD	27	02		BEQ		Suite7	
E0FF	20	24	DEFAUT	BRA		DEFAUT	
E101	8B	01	Suite7	ADD	A	#01	
E103	C9	00		ADC	B	#00	
E105	08			INX			
E106	8C	0055		CPX		#0055	0000000001010101
E109	26	F6		BNE	7		
E10B	81	8C		CMP	A	#8C	
E10D	26	16		BNE		DEFAUT	
E10F	C1	55		CMP	B	#55	
E111	26	12		BNE		DEFAUT	
E113	8B	01	8	ADD	A	#01	
E115	C9	00		ADC	B	#00	
E117	08			INX			
E118	8C	0303		CPX		#0303	0000001100000011
E11B	26	F6		BNE	8		
E11D	81	3A		CMP	A	#3A	
E11F	26	04		BNE		DEFAUT	
E121	C1	58		CMP	B	#58	
E123	27	02		BEQ		Suite9	
E125	20	5F	DEFAUT	BRA		DEFAUT	
E127	8B	01	Suite 9	ADD	A	#01	
E129	C9	00		ADC	B	#00	
E12B	08			INX			
E12C	8C	0505		CPX		#0505	0000010100000101
E12F	26	F6		BNE	9		
E131	81	3C		CMP	A	#3C	
E133	26	51		BNE		DEFAUT	
E135	C1	5A		CMP	B	#5A	
E137	26	4D		BNE		DEFAUT	
E139	CE	1111		LDX		#1111	0001000100010001
E13C	8C	1111		CPX		#1111	
E13F	26	45		BNE		DEFAUT	
E141	5F			CLR	B		
E142	4F			CLR	A		
E143	8B	01	10	ADD	A	#01	
E145	C9	00		ADC	B	#00	
E147	09			DEX			
E148	8C	0003		CPX		#0003	0000000000000011
E14B	26	F6		BNE	10		
E14D	81	0E		CMP	A	#0E	
E14F	26	35		BNE		DEFAUT	

E152	26	31		BNE		DEFAUT	
E154	CE	0005		LDX		# 0005	0000000000000101
E157	8C	0005		CPX		# 0005	
E15A	26	29		BNE		DEFAUT	
E15C	5F			CLR	B		
E15D	4F			CLR	A		
E15E	8B	01	11	ADD	A	# 01	
E160	C9	00		ADC	B	# 00	
E162	08			INX			
E163	8C	00 09		CPX		# 0009	0000000000001001
E166	26	F6		BNE	11		
E168	81	04		CMP	A	# 04	
E16A	26	19		BNE		DEFAUT	
E16C	CE	01 02		LDX		# 0102	
E16F	8C	01 02		CPX		# 0102	0000 0001 0000 0010
E172	26	11		BNE		DEFAUT	
E174	5F			CLR	B		
E175	4F			CLR	A		
E176	8B	01		ADD	A	# 01	
E178	C9	00		ADC	B	# 00	
E17A	09			DEX			
E17B	8C	01 01		CPX		# 0101	0000000100000001
E17E	26	F6		BNE	12		
E180	81	01		CMP	A	# 01	
E182	27	01		BEQ		~~suit~~AFF	
E184	3F		DEFAUT	SWI			Si Le Test de la RAM, de
E185	86	7D	AFF	LDA	A	# 7D	A, B, RX, CC est positif.
E187	B7	E41D		STA	A	E41D	(pas de defaut)
E18A	86	3F		LDA	A	# 3F	
E18C	B7	E41E		STA	A	E41E	Le mot GOOd r5 sera
E18F	86	3F		LDA	A	#3F	affiché.
E191	B7	E41F		STA	A	E41~~5~~F	il suffit d'appuyer
E194	86	5E		LDA	A	#5E	sur R/S et introduire son
E196	B7	E420		STA	A	E420	programme.
E199	86	50		LDA	A	# 50	
E19B	B7	E421		STA	A	E421	
E19E	86	6D		LDA	A	# 6D	
E1A0	B7	E422		STA	A	E422	
E1A3	CE	F0A2		LDX		#IDDLE	
E1A6	FF	E419		STX		MNPTR	
E1A9	7E	F0BB		JMP		PUT	
E1AC	3F			SWI			

Remarques: - Il est difficile de générer la matrice R-M avec le "pointeur de pile" (SP) car celui est géré par le moniteur du kit, donc son contenu varie suivant le deroulement du programme.

- l'indicateur H est testé indirectement en utilisant les instructions qui l'affecte (ADD, ADC, ...)

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

[1] THEORIE ET TECHNIQUE DE LA TRANSMISSION DES DONNEES
J-CLAVIER
G-COFFINET TOME -I-

[2] INTRODUCTION A LA THEORIE DE LA COMMUNICATION
"THEORIE DE L'INFORMATION"
E-ROUBINE TOME - III

[3] CODE AUTO-CORRECTEURS D'ERREURS ADAPTES A LA PROTECTION DE L'INFORMATION NUMERIQUE SUR VIDEO DISQUE ANALOGIQUE
ABDELLAH MAHDI THESE 3EME CYCLE

[4] CODES DETECTEURS ET CORRECTEURS D'ERREURS
G CULLMANN

[5] METHODES DE TEST D'UNE FAMILLE DE MICROODINATEURS
M-BOUTRIA
M-SOUFI THESE D'INGENIORAT

[6] INITIATION A LA LOGIQUE PROGRAMMEE ET AU MICROPROCESSEUR
J-COUDERC

[7] L'EMPLOI DES MICROPROCESSEURS
M- AUMIAUX

[8] DE LA LOGIQUE CABLEE AU MICROPROCESSEUR
J-M-BERNARD
J- HUGON
R-LE CORVEC TOME -I-

[9] INTRODUCTION AUX MATHEMATIQUES DISCRETES
S. YABLONSKI