UNIVERSITE SIDI MOHAMMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES FES
DEPARTEMENT DE GENIE ELECTRIQUE

Diplôme de Licence

Electronique Télécommunication et Informatique
(ETI)

RAPPORT DE FIN D'ETUDES

Intitulé :

Code Reed-Solomon.

Réalisé Par :

Lamrani Imane

Encadré par :

M^R A. Lichioui (Société Nationale de Radiodiffusion et Télévision)

P^r A.ahaitouf et P^r F.Abdi (FST)

Soutenu le 18 Juin 2011 devant le jury

Pr	**A.Ahaitouf**	**(FST FES)**
Pr	**F.Abdi**	**(FST FES)**
Pr	**F.Errahimi**	**(FST FES)**
Pr	**Ait Madi**	**(FST FES)**

Année Universitaire 2010/2011

TABLE DES MATIERES

Remerciements

Ce travail rentre dans le cadre de mon stage de fin d'études pour le diplôme de Licence Science et Techniques en Electronique Télécommunications et Informatique, dont le sujet porte sur le codage de REED SOLOMON. Il a été effectué à la SNRT (Société Nationale de Radiodiffusion et de Télévision). Ce thème a été choisi vu son adéquation avec ma formation.

A son terme, je tiens à exprimer ma profonde gratitude à mon encadrant Dr. Ingénieur A. LICHIOUI et à l'ensemble des techniciens de la SNRT pour leur aide précieuse, leurs conseils et leurs suggestions avisées qui m'a aidé à mener à bien ce travail.
Le mérite revient également à Mr A .AHAITOUF et Mr F.Abdi pour leur soutien durant toute la période de mon stage. Enfin, je remercie AIT MADI ainsi que tous mes enseignants pour la qualité de l'enseignement qu'ils ont prodigué à toute la promotion durant nos études.

INTRODUCTION GENERALE

Il est clair que, les communications jouent un rôle de plus en plus primordial vu la place qu'elles occupent dans le quotidien de chacun. De plus les besoins constants en ergonomie et en qualité de service ne peuvent être atteints qu'avec une meilleure qualité du signal transmis. Pour une communication à distance, il est nécessaire de disposer d'un canal de transmission qui achemine l'information sans la modifier. Ce canal est assujetti à plusieurs perturbations qui peuvent dégrader la qualité du signal. Pour remédier à ce problème on utilise le codage de l'information. Il faut alors détecter et corriger les éventuelles erreurs et pour cela on dispose des codes détecteurs et correcteurs d'erreurs dont le code Reed Solomon qui fait l'objet de ce rapport.

Le codage Reed Solomon qui fait l'objet de ce projet reçoit des applications multiples dans :

*La sauvegarde des données, c'est ainsi que par exemple pour les CD (Compact Disc), il permet de résister aux trains d'erreurs que peut provoquer une rayure qui détruit beaucoup d'octets localement. Pour les DVD (Digital Versatile Disc) le principe est le même que pour les CD.

*La communication mobile, les réseaux sans fils (wireless…), les communications satellitaires, la télévision et radio numériques ainsi que les modems ADSL.

Le codage Reed Solomon peut aussi être utilisé pour des sondes spatiales, d'exploration lointaine, comme les sondes Voyager par exemple. Ils sont relativement compliqués `à décoder, mais les techniques de décodage évoluent.

Il constitue ainsi un outil incontournable dans le traitement et la transmission des données actuellement.

Ce rapport est le fruit d'un stage de fin d'études effectué a la SNRT, Société Nationale de Radiodiffusion et Télévision et dont le thème est : LE CODAGE REED SOLOMON.

Il m'a été fixé comme cahier de charge :

1-comprendre et maitriser le codage et décodage de Reed Solomon.

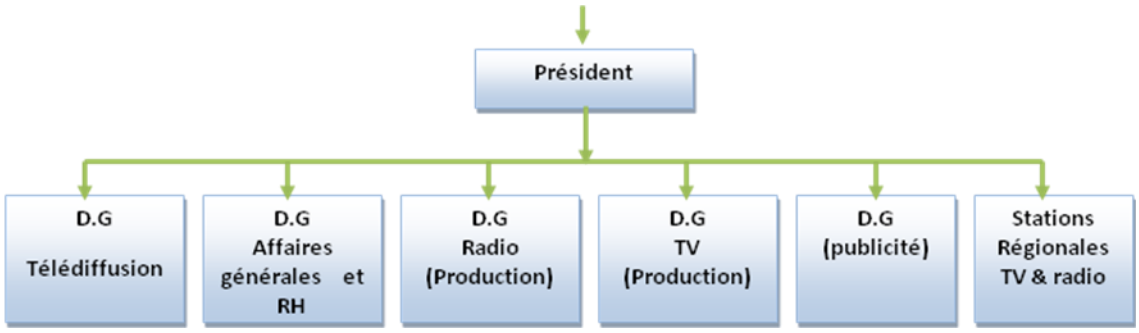
2-Faire une simulation sur Matlab.

Concernant la structuration du rapport :

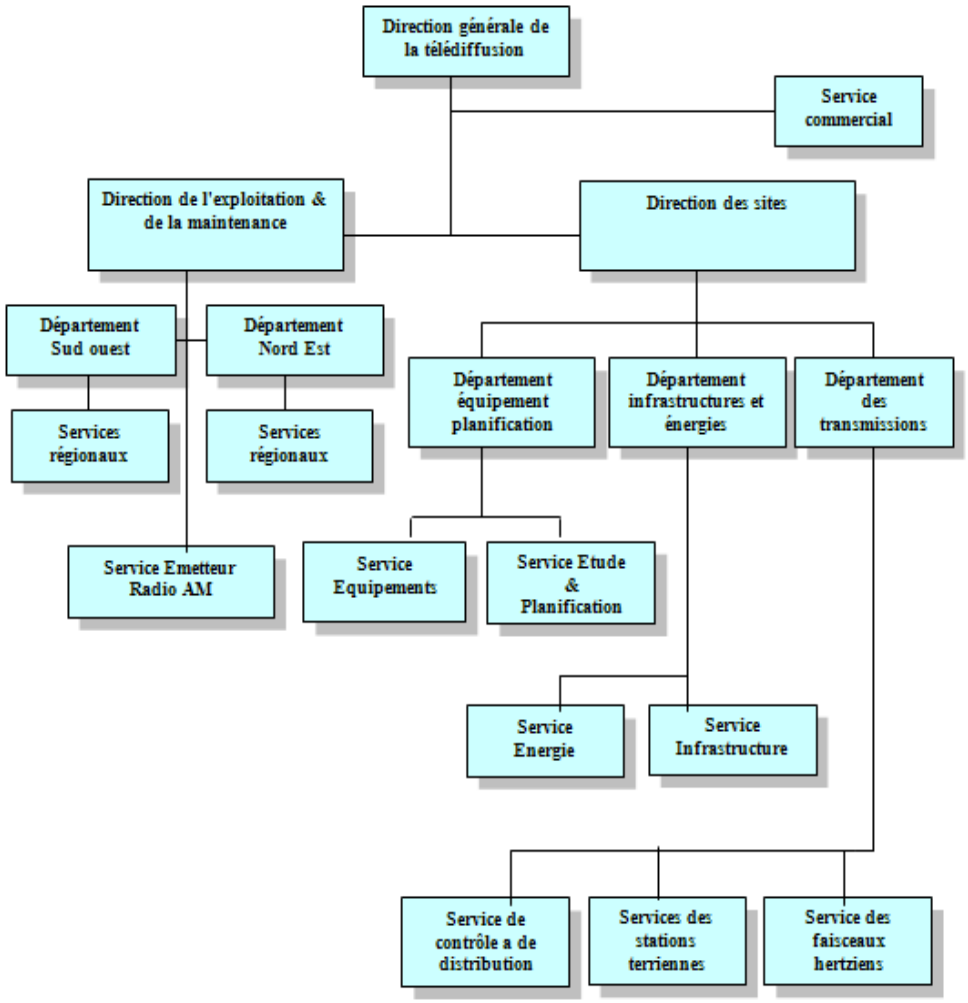
Après une présentation de l'organisme d'accueil, le deuxième chapitre concernera le codage de canal d'une façon générale. Le troisième chapitre traitera du codage REED SOLOMON et des simulations sur Matlab qui constituent le noyau dur de ce travail ;

Chapitre 1 : Présentation de l'environnement de travail

La Société Nationale de Radiodiffusion et Télévision, représentées dans l'organigramme suivant :



La direction de la Télédiffusion : c'est le cœur de la SNRT. Cette direction s'occupe de la transmission par faisceaux hertziens ou par satellites des signaux TV et sons. Elle s'occupe aussi de la diffusion de la Radio AM et FM



C'est dans cette direction de l'exploitation et de la maintenance ou j'ai effectué mon stage ;

La direction de la production télévisée: Elle s'occupe de tout ce qui est programmes de Télévision, elle gère les programmes TV. Cette direction se compose de sous directions gérant chacune une chaîne de télévision(Al oula , Al Assadissa..)

La Direction de la production Radio : Elle se charge de la conception des émissions Radio aussi bien au niveau des stations nationales que des stations régionales comme celles de Tanger, Fès, Oujda, Marrakech, Casablanca, Agadir, Meknès et Laayoune. Dans ces stations régionales des reportages TV et Radio des activités locales sont réalisés et transmis aux studios de Rabat.

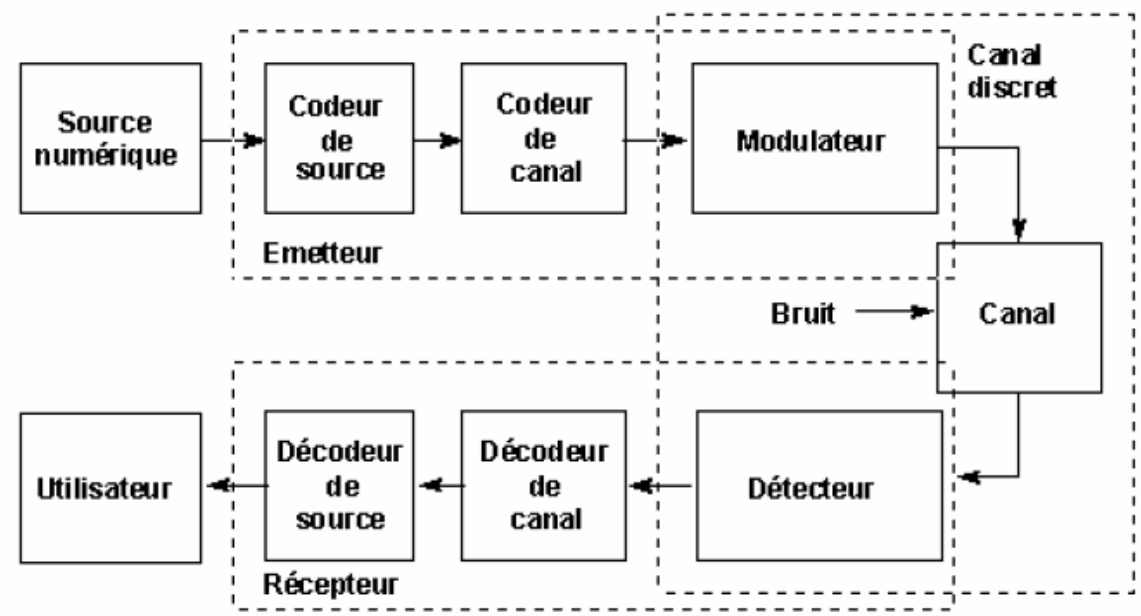
La Direction de la publicité : C'est l'organe qui s'occupe de tout ce qui est publicité. Elle gère les spots publicitaires et assure la gestion financière des ressources de la publicité.

La Direction des affaires administratives et financières : Cette direction s'occupe de tout ce qui est relations internes, relations externes, personnel, salaires, marchés, et tout ce qui concerne les ressources humaines de la Société Nationale de la Radiodiffusion et de Télévision.

Chapitre 2 : LE CODAGE DE CANAL

I- Système de communication numérique.

Un système de communication permet le transfert des informations d'une source vers une destination via un canal de transmission. Ce dernier possède un certain nombre de caractéristiques : capacité (bande passante…), nature physique et le bruit qui va entacher l'information d'erreurs. La figure suivante présente une chaîne de communication générale.



Synoptique d'un système de transmission numérique

L'émetteur est constitué d'un codeur de source d'un codeur de canal et d'un modulateur.

Le bloc source numérique : Qui contient la source initiale du message c.à.d. l'information

Le codage de source : Qui vise à minimiser les ressources nécessaires à la transmission (temps, puissance, bande passante, surface de stockage, etc.).

Le modulateur : Il permet d'adapter les caractéristiques du signal à celles d'un canal.

II- Le codage de canal :

Sur le canal de transmission plusieurs erreurs peuvent perturber le signal utile qui peuvent être dues aux interférences aux bruits.. Ces erreurs dégradent les signaux de communication transmis et provoquent des erreurs de détection à la réception.

Pour un certain niveau de bruit et d'interférences, on peut réduire la probabilité d'erreur en augmentant la puissance d'émission. Mais, cette augmentation de puissance n'est pas toujours souhaitable car d'une part elle se traduit par une grande consommation électrique du terminal, et d'autre part, dans le cas d'une transmission en espace libre, elle augmente les interférences inter-utilisateurs, ce qui accroît la probabilité d'erreur.

Une autre solution est **le codage de canal**, qui permet de corriger une ou plusieurs erreurs dans un mot code en ajoutant à l'information des symboles de redondance ou symboles de contrôle, de telle sorte que le message codé ait une structure particulière.
A la réception, le décodeur de canal vérifie si cette structure est bien respectée. Dans le cas contraire, si une erreur est détectée elle sera éventuellement corrigée. Cette opération s'appelle le décodage de canal.

L'information de la source est mise en trames de longueurs fixes que nous devons transmettre : c'est le message. Le codage de canal prend ce message pour en faire un mot de code :

Message (K=11) ⇨ **codage de canal** ⇨ **mot de code(n=15)**

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11) (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ***11, 10, 14, 6***)

Le message est constitué de **k** caractères. Le mot de code utilisé sera lui aussi de longueur fixe de **n** caractères. Avec **n > k** il y aura donc **n-k** caractères du mot de code qui sont redondants et serviront à traiter les erreurs éventuelles.

On caractérise les codes par leur capacité de correction d'erreurs. En général il y'a deux types de codages :

- Ceux qui sont bien adaptés aux coupures (paquet d'erreurs) comme les codes de FIRE, les codes **de Reed Solomon** qui fera l'objet d'un chapitre à part et qui constitue l'essentiel de ce rapport.

- Ceux qui luttent bien contre les erreurs isolées tels que les codes de BCH le code de GOLAY de Reed Muller et le codage de Hamming dont je rappelle l'essentiel ci-après.

[1]

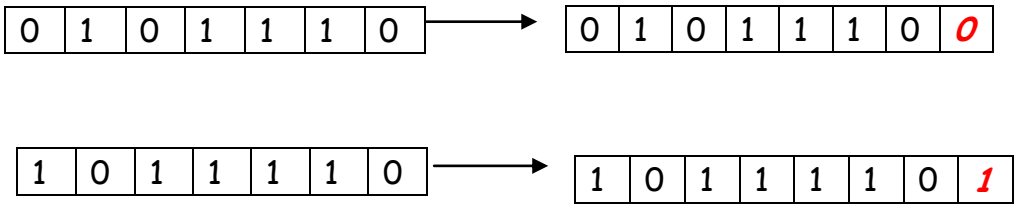
III- Le Codage de Hamming :

C'est le premier code correcteur véritablement efficace, basé sur le test de parité.

a-le bit de parité :

On sectionne l'information à transmettre en paquet de n bits, correspondant au message. Le message contrôlé est formé en ajoutant **un bit de parité** à l'information initiale de sorte qu'il y'ait en tout un nombre pair de 1.

Ex :



L'émetteur transmet alors le message contrôlé et le récepteur vérifie la parité du message reçu.
Si lors de la transmission, il y'a erreur, la parité du message reçu n'est pas correcte et le récepteur détecte l'erreur et demande de transmettre une deuxième fois le message.
Mais si lors d'une transmission deux erreurs surviennent, la parité du message reçu est correcte et donc les erreurs ne sont pas détectées.

b- théorie du codage de Hamming :

Le message contrôlé est de 2^n bits. Pour illustrer nous allons prendre n=3, i.e. 1 octet à transmettre dont les bits sont numérotés de 0 à 7 allant de droite a gauche.

7	6	5	4	3	2	1	0

Le bit 0 : contrôle la parité de l'octet.

Les bits (1, 2 ,4) qui correspondent aux puissances de 2 : contrôlent le message.

Les bits 3, 5, 6, 7 : Ces bits sont ceux qui contiennent l'information transmise.

On décompose ces numéros de bits en puissance de 2

3=2+1 5=4+1 6=4+2 7=4+2+1

Le bit 1 apparait dans la décomposition de 3 5 et 7 et donc il va intervenir dans le contrôle de la parité de ces trois bits.

Le bit 2 apparait dans la décomposition de 3 6 et 7 et donc il va intervenir dans le contrôle de la parité de ces trois bits.

Le bit 4 apparait dans la décomposition de 5 6 et 7 et donc il va intervenir dans le contrôle de la parité de ces trois bits.

Exemple :

L'information à transmettre étant : 1101

On voit bien que les bits 3 5 6 et 7 sont ceux qui portent l'information et dont la parité sera contrôlée par les autres bits.

7	6	5	4	3	2	1	0
1	**1**	**0**		**1**			

Le bit 1 contrôle la parité des bits 3,5 et 7. La parité est respectée et donc le bit 1 sera nul.

7	6	5	4	3	2	1	0
1	**1**	**0**		**1**		**0**	

Le bit 2 contrôle la parité des bits **3 ,6** et **7**. La parité n'étant pas respectée et donc le bit 2 sera égal à 1 pour rétablir la parité.

7	6	5	4	3	2	1	0
1	**1**	**0**		**1**	**1**	**0**	

Le bit 4 contrôle la parité des bits **5 ,6** et **7**. Ici également la parité est respectée.

7	6	5	4	3	2	1	0
1	**1**	**0**	**0**	**1**	**1**	**0**	

Le bit 0 contrôle la parité de l'ensemble de l'octet. Ici et suite à l'intervention des bits 1 2 et 4 on voit bien que la parité de l'octet est respectée (nombre pair de 1). Et donc :

7	6	5	4	3	2	1	0
1	**1**	**0**	**0**	**1**	**1**	**0**	**0**

Si lors de la transmission, une erreur est commise on en détecte l'existence par l'imparité du message transmis, on contrôle ensuite la parité des bits 1,2 et 4.

Comme dans la réalité on peut avoir plus d'une erreur par mot, le code de Hamming se révèle insuffisant.

Chapitre 3 : LE CODAGE DE REED SOLOMON

La Réalisation pratique du code correcteur d'erreurs Reed Solomon(RS) nécessite la connaissance mathématique des champs de Galois sur lesquels il est basé.

C'est pour cela qu'avant de présenter le code de Reed-Solomon, je me propose de faire une présentation simplifiée de l'arithmétique du champ de Galois.

I- Notions mathématiques appliquées dans le code de Reed Solomon :

[2]

1. champs de Galois :

Les « champs de Galois » font partie d'une branche particulière des mathématiques qui modélise les fonctions du monde numérique. Ils sont très utilisés dans la cryptographie ainsi que pour la reconstruction des données comme on le verra dans le chapitre 3.
La dénomination « champ de Galois » provient du mathématicien français Galois qui en a découvert les propriétés fondamentales.

2. Polynôme primitif :

Ce polynôme permet de construire le « champ de Galois » souhaité. Tous les éléments non nuls du champ peuvent être construits en utilisant l'élément a comme racine du polynôme primitif. Chaque m peut avoir plusieurs polynômes primitifs p(x), mais dans le tableau ci-dessous, on mentionne seulement les polynômes ayant le moins d'éléments. Les polynômes primitifs pour les principaux « champs de Galois » sont les suivants:

M	P(x)	M	P(x)
3	$1+X+X^3$	14	$1+X+X^6+X^{10}+X^{14}$
4	$1+X+X^4$ ([1])	15	$1+X+X^{15}$
5	$1+X^2+X^5$	16	$1+X+X^3+X^{12}+X^6$
6	$1+X+X^6$	17	$1+X^3+X^{17}$
7	$1+X^3+X^7$	18	$1+X^7+X^{18}$
8	$1+X+X^3+X^4+X^8$ ([1])	19	$1+X+X^2+X^5+X^{19}$

9	$1+X^4+X^9$	20	$1+X^3+X^{20}$
10	$1+X^3+X^{10}$	21	$1+X^2+X^{21}$
11	$1+X^2+X^{11}$	22	$1+X+X^{22}$
12	$1+X+X^4+X^6+X^{12}$	23	$1+X^5+X^{23}$
13	$1+X+X^3+X^4+X^{13}$	24	$1+X+X^2+X^7+X^{24}$

Tableau : polynômes primitifs dans GF (2^m)

3. Eléments des champs de Galois :

Un « champ de Galois » se compose d'un ensemble d'éléments. Ces éléments sont basés sur un élément primitif noté α et prennent les valeurs :

$$0,1,\alpha,\alpha^2,\alpha^3,\ldots.\ ,\alpha^{n-1}$$

En prenant $n=2^m-1$, on forme un ensemble de 2^m éléments. Le champ de Galois est alors noté GF (2^m).

*Exemple de construction d'un Champ de Galois pour m =4 (GF(2^4) :

Dans notre exemple m=4 et donc le polynôme primitif sera : $P(X)=1+X+X^4$

L'élément α est racine du polynôme primitif,

Et donc $p(\alpha)=1+\alpha+\alpha^4$

Voici un tableau illustrant notre propos pour les éléments de GF (**2^4**) :

Eléments	Formes polynomiales	Formes binaires	Formes décimales
0	0	0000	0
1	1	0001	1
α	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha+1$	0011	3
α^5	$\alpha^2+\alpha$	0110	6
α^6	$\alpha^3+\alpha^2$	1100	12
α^7	$\alpha^3+\alpha+1$	1011	11
α^8	α^2+1	0101	5
α^9	$\alpha^3+\alpha$	1010	10

α^{10}	$\alpha^2+\alpha+1$	0111	7
α^{11}	$\alpha^3+\alpha^2+\alpha$	1110	14
α^{12}	$\alpha^3+\alpha^2+\alpha+1$	1111	15
α^{13}	$\alpha^3+\alpha^2+1$	1101	13
α^{14}	α^3+1	1001	9

Procédure pour obtenir ces résultats :

Les éléments 0, 1, α, α^2, α^3 sont simplement représentés en binaire par (0000, 0001, 0010, 0100, 1000) et pour les autres éléments on fera une division par le polynôme primitif P(a), le résultat étant le reste de la division.

II- Les opérations arithmétiques dans le Champ de Galois :

1- L'adition dans le champ de Galois (GF):

L addition dans le champ de Galois est tout simplement comme l'addition de deux polynômes, soient les deux polynômes suivant.

$$(a_{m-1}x^{m-1}+\dots\dots a_1x^1+a_0) + (b_{m-1}x^{m-1}+\dots\dots b_1x^1+b_0) =$$

$$c_{m-1}x^{m-1}+\dots c_1x^1+c_0$$

Avec $c_i=a_i+b_i$ pour $0\leq i \leq m-1$ comme les coefficients ne peut prendre que les valeurs 0 et 1 donc :

$$C_i=0 \quad \text{pour} \quad a_i=b_i$$

$$C_i=1 \quad \text{pour} \quad a_i\neq b_i$$

Pour additionner deux éléments dans le champ du Galois on utilise une porte **<<XOR>>.**

Exemple

x^3+x et x^3+x^2+1 on obtient x^2+x+1

En binaire 1010 + 1101 = 0111 ou en décimale 10+13= 7

Dans le champ du Galois, l'addition de deux éléments identique est nulle.

Par exemple 2+2=0 en binaire 10+10 = 00.

2- La Soustraction dans la champ de Galois:

La soustraction de deux éléments dans le champ de Galois revient à faire l'addition de ces deux éléments. Et on constate que :

$C_i = a_i - b_i$ avec :

$C_i = 0$ pour $a_i = b_i$

$C_i = 1$ pour $a_i \neq b_i$

On constate que la soustraction dans GF(2) effectue la même opération que l'addition dans le même champ, c'est-à-dire une opération logique « XOR ».

3- La Multiplication dans GF :

On fait les calculs tout en remplaçant chaque valeur par son équivalent dans le tableau des éléments de GF (2^m)

Exemple :

Pour GF(2^4) : considérons la multiplication des deux polynômes suivants :

$(x^3 + 7x^2 + 14x + 8)(x + 8) = (x^3 + \alpha^{10}x^2 + \alpha^{11}x + \alpha^3)(x + \alpha^3)$

$= (x^4 + \alpha^3 x^3 + \alpha^{10}x^3 + \alpha^{13}x^2 + \alpha^{11}x^2 + \alpha^{14}x + \alpha^3 x + \alpha^6$

$= x^4 + (\alpha^{10} + \alpha^3)x^3 + (\alpha^{13} + \alpha^{11})x^2 + (\alpha^{14} + \alpha^3)x + \alpha^6$

$= x^4 + (\alpha^3 + \alpha^2 + \alpha + 1)x^3 + (\alpha^3 + \alpha^2 + 1 + \alpha^3 + \alpha^2 + \alpha)x^2 + (\alpha^3 + 1 + \alpha^3) + \alpha^6$

$= \alpha^0 x^4 + \alpha^{12} x^3 + \alpha^4 x^2 + \alpha^0 x + \alpha^6$

4- La division dans le champ de Galois:

La division peut être faite en faisant une multiplication par l'élément inverse du dénominateur:

$$\frac{\alpha^i}{\alpha^j} = \alpha^i \times \alpha^{-j}$$

Le calcul de l'inverse dans un « champ de Galois » est défini comme suit :

$$\alpha^{-j} = \alpha^{element\_max - j}$$

On va utiliser cette relation pour déterminer l'inverse de tous les de champs de Galois.

Entrée en décimal	Forme polynomiale	Inverse	Inverse en Décimale
0	0	0	0
1	α^0	α^0	1
2	α^1	$\alpha^{-1} = \alpha^{14}$	9
3	α^4	$\alpha^{-4} = \alpha^{11}$	14
4	α^2	$\alpha^{-2} = \alpha^{13}$	13
5	α^8	$\alpha^{-8} = \alpha^{7}$	11
6	α^5	$\alpha^{-5} = \alpha^{10}$	7
7	α^{10}	$\alpha^{-10} = \alpha^{5}$	6
8	α^3	$\alpha^{-3} = \alpha^{12}$	15
9	α^{14}	$\alpha^{-14} = \alpha^{1}$	2
10	α^9	$\alpha^{-9} = \alpha^{6}$	12
11	α^7	$\alpha^{-7} = \alpha^{8}$	5
12	α^6	$\alpha^{-6} = \alpha^{9}$	10
13	α^{13}	$\alpha^{-13} = \alpha^{2}$	4
14	α^{11}	$\alpha^{-11} = \alpha^{4}$	3
15	α^{12}	$\alpha^{-12} = \alpha^{3}$	8

Inverse des éléments dans le **GF (2^m)**

Exemple:

$$\frac{10}{13} = \frac{\alpha^9}{\alpha^{13}} = \alpha^9 \times \alpha^{15-13}$$

$$\alpha^9 \times \alpha^2 = \alpha^{11} = 14$$

5- Construction d'un champ de GF (2^4) sous Matlab :

Les éléments d'un « champ de Galois » peuvent être aussi calculés avec **Matlab** selon les instructions suivantes :

```
p=2;   % Nombre base du champ
m=4;   % Eléments

champ = gftuple([-1:p^m-2]',m,p); %Calcul du champ en binaire
```

Code : éléments dans GF (2^4) sous Matlab

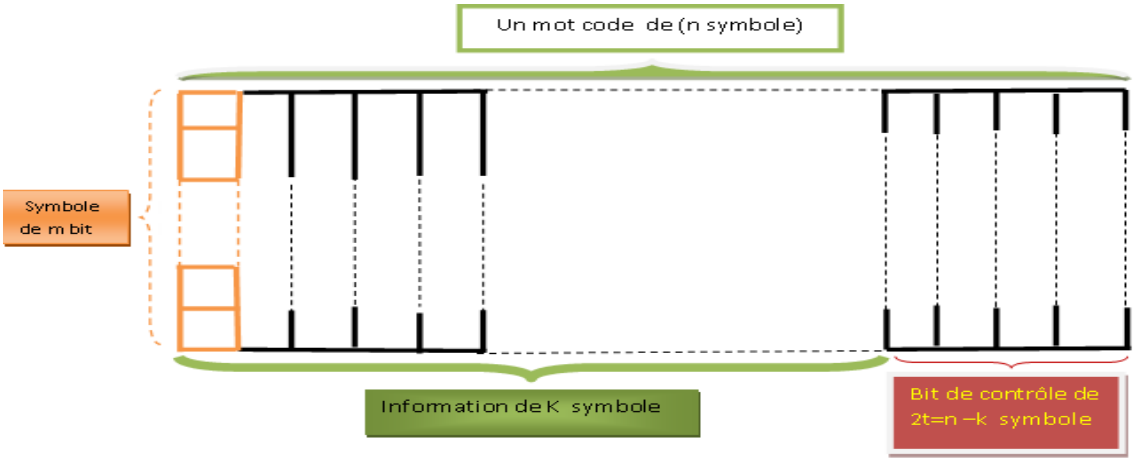
III- Théorie du codage Reed Solomon.

Le code Reed Solomon est un code en bloc ou les messages sont divisés en blocs auxquels on a ajouté des informations redondantes. La longueur des blocs dépend de la capacité des codeurs.

Pour chaque bloc on ajoute des bits de protection ou bien de la parité supplémentaire pour former un mot code de n symboles.

C'est également un code systématique, c.à.d. les symboles de protection sont ajoutés à la fin de l'information.

Le codeur prend K symboles de donnée et calcule les informations de contrôle pour construire n symboles, ce qui donne n-k symboles de contrôle. Le décodeur peut corriger au maximum t symboles, ou 2t= n-k.



RS (n, k) :

n : nombre de symboles d'un mot code.

k : nombre de symboles de l'information.

La longueur maximale d'un code de Reed-Solomon est définie comme :

$$\mathbf{n = 2^m - 1 = k + 2t}$$

Avec t : La capacité de correction des erreurs du système (avec 2t=n-k).

m: le nombre de bits dans un symbole

La distance minimale d'un code Reed – Solomon est :

$$\mathbf{d_{min} = 2t + 1}$$

Ainsi le code Reed-Solomon peut détecter **2t=n-k** erreurs, et capable de corriger

t= (n-k)/2 erreurs.

Le nombre de bits **m** par symbole :

$n = 2^m - 1$ Donc **m=ln (n+1) /ln(2)**

1-Construction du mot code :

L'équation clé définissant le codage systématique de REED SOLOMON (n,k) est :

$$T(x) = M(x) \times x^{n-k} + [M(x) \times x^{n-k}] \bmod (g(x))$$

Avec

T(x) : polynôme du mot code de degré n-1.

M(x) : polynôme d'information de degré k-1.

[M(x)*x^{n-k}]mod(g(x)) : Polynôme de contrôle de degré n-k-1.

g(x) : Polynôme générateur de degré n-k

mod : reste de la division

1.1 génération des symboles de contrôles :

Les symboles de contrôles sont générés a l'aide de polynômes particuliers appelés polynômes générateurs..

Le polynôme générateur se présente sous la forme :

$$g(X) = \prod_{i=k}^{n-1} (X - \alpha^{-i}) = \prod_{i=k}^{n-1} (X - \alpha^{-i} \times \alpha^{n}) = \prod_{i=1}^{n-k} (X - \alpha^{i})$$

$g(x)=(x-\alpha^1)(x-\alpha^2)\ldots..(x-\alpha^{2t})$

Exposons maintenant deux exemples de calculs des coefficients du polynôme générateur l'une avec RS(15,9) l'autre RS(15,11).

Pour chaque exemple le calcul sera effectué successivement de façon manuelle et puis sous MATLAB.

Exemple1 : calcul manuel des coefficients du polynôme générateur pour RS(15,9) :

n=15 ; k=9
Donc **2t=n-k=15-9=6**
D'où le code Reed Solomon détectera 6 erreurs et corrigera **t=6 /2=3 erreurs**.
m=ln(n+1)/ln(2)=ln(16) /ln(2)=4 donc on aura **m= 4 bits par symbole**.

En développant la formule générale du polynôme générateur dans notre cas on obtient :

$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5)(x-\alpha^4)$

$=(x^2+\alpha^5x+\alpha^3)(x^2+\alpha^7x+\alpha^7)(x^2+\alpha^9x+\alpha^{11})$

$=(x^4+\alpha^{13}x^3+\alpha^6x^2+\alpha^3x+\alpha^{10})(x^2+\alpha^9x+\alpha^{11})$

$=x^6+x^5(\alpha^{13}+\alpha^9)+x^4(\alpha^6+\alpha^7+\alpha^{11})+x^3(\alpha^3+1+\alpha^9)+x^2(\alpha^{10}+\alpha^{12}+\alpha^2)+x(\alpha^4+\alpha^{14})+\alpha^6$

$=x^6+\alpha^{10}x^5+\alpha^{14}x^4+\alpha^4x^3+\alpha^6x^2+\alpha^9x+\alpha^6$

En prenant l'équivalence en décimal on obtient :

$g(x)=x^6+7x^5+9x^4+3x^3+12x^2+10x+12$

Exemple1 : calcul sous MATLAB des coefficients du polynôme générateur pour RS(15,9) :

```
n=15; k=9;
m=4;
 genpoly = rsgenpoly(n,k)
```

```
genpoly = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     7     9     3    12    10    12

>>
```

On voit bien que les calculs manuels sont identiques à ceux calculés sous Matlab.

Exemple 2 :Calcul manuel des coefficients du polynôme générateur pour RS(15,11) :

n=15 ; k=11
Donc **2t=n-k=15-11=4**
D'où le code Reed Solomon détectera 4 erreurs et corrigera **t=4 /2=2 erreurs**.
m=ln (n+1)/ln(2)=ln(16) /ln(2)=4 donc on aura **m= 4 bits par symbole**.

En développant la formule générale du polynôme générateur dans notre cas on obtient :

$$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)$$
$$=(x^2-\alpha^2x-\alpha^1 x+\alpha^3)(x-\alpha^3)(x-\alpha^4)$$
$$=(x2-(\alpha^2+\alpha)x+\alpha^3)(x-\alpha^3)(x-\alpha^4)$$
$$=(x^2-\alpha^5x+\alpha^3)(x-\alpha^3)(x-\alpha^4)$$
$$=(x^3-\alpha^3x^2-\alpha^5x^2+\alpha^8x+\alpha^3x-\alpha^6)(x-\alpha^4)$$
$$=(x^3-\alpha^{11}x^2+\alpha^{13}x-\alpha^6)(x-\alpha^4)$$
$$=x^4-\alpha^4x^3-\alpha^{11}x^3+\alpha^{15}x^2+\alpha^{13}x^2-\alpha^{17}x-\alpha^6x+\alpha^{10}$$
$$=x^4-(\alpha^4+\alpha^{11})x^3+(\alpha^{15}+\alpha^{13})x^2-(\alpha^{17}+\alpha^6)x+\alpha^{10}$$
$$=x^4-13x^3+12x^2-8x+7$$

Exemple 2 : Calcul des coefficients du polynôme générateur de RS(15,11) en utilisant **Matlab** :

```
n=15; k=11;
m=4;
 genpoly = rsgenpoly(n,k)
```

```
genpoly = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     13     12      8      7
```

Pour les deux exemples les résultats manuels correspondent bien aux résultats sous MATLAB.

1.2 Le codage du message :

Pour coder le message M(x), il suffit d'abord de multiplier le message M(x) par X^{n-k} et puis de le diviser par le polynôme générateur g(x). Le reste **R(x)** de la division représente les symboles de contrôles.

$$\mathbf{T(x) = M(x) \times x^{n-k} + \left[M(x) \times x^{n-k}\right] mod(g(x))}$$

Avec

T(x) : polynôme du **mot code** de degré n-1.

M(x) : polynôme du message

Et les coefficients m_{k-1} ...., m_1, m_0 sont des symboles ; m_{k-1} est le premier symbole du message.

Le mot de code transmis T (x) peut alors être formé en combinant le M (x) et R (x) comme suite : $T(x) = (M(x) \times x^{n-k}) + r(x)$

Reprenons maintenant les exemples précédents et construisons sous MATLAB un mot code pour chacun d'eux.

Exemple 1 : Construction d'un mot code du RS(15,9) sous **Matlab** :

```
n=15; k=9;
m=4;
genpoly = rsgenpoly(n,k)
```

```
msg  = gf([1 2 3 4 5 6 7 8 9],m)
code = rsenc(msg,n,k)
```

```
genpoly = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     7     9     3    12    10    12


msg = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     2     3     4     5     6     7     8     9


code = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

  Columns 1 through 14

     1     2     3     4     5     6     7     8     9     2     1     3    12    15

  Column 15

    11

>>
```

Et donc le mot code est : T(x) = [1 2 3 4 5 6 7 8 9 2 1 3 12 15]

Exemple 2 :Construction d'un mot code du RS(15,11) sur **Matlab** :

```
n=15; k=11;
m=4;
genpoly = rsgenpoly(n,k)
msg  = gf([1 2 3 4 5 6 7 8 9 10 11],m)
code = rsenc(msg,n,k)
```

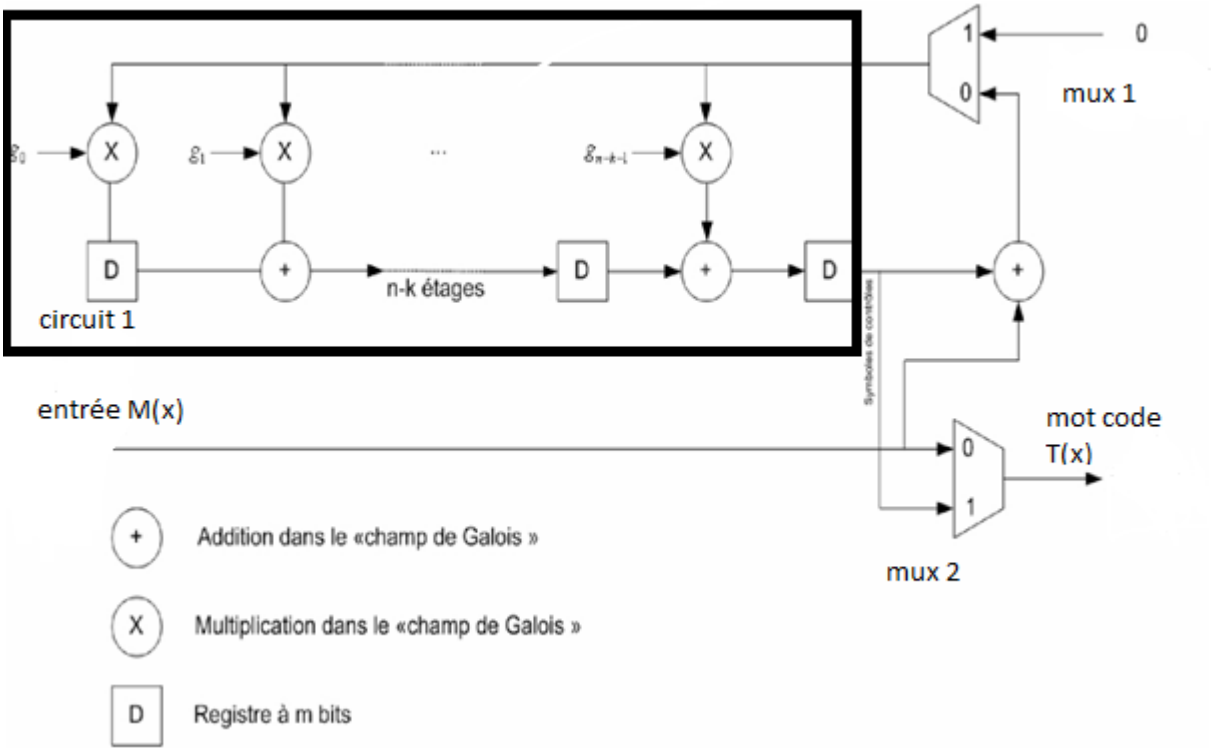
```
genpoly = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1    13    12     8     7


msg = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     2     3     4     5     6     7     8     9    10    11


code = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     2     3     4     5     6     7     8     9    10    11    11    10    14     6

>>
```

Schéma général du codage Reed Solomon :

Mathématiquement le codage est effectué selon la fonction :

$$T(x) = \left(M(x) \times x^{n-k}\right) + r(x)$$

Avec r(x) : le reste de la division de l'information M(x) et le polynôme générateur. Cette division nous donne les symboles de contrôle.
Donc on constate que la réalisation d'un codeur nécessite deux opérateurs : Le décalage et la division. Ces deux opérations peuvent être effectuées grâce a des registres de décalages et à des multiplexeurs.



Principe du schéma :

*initialement on remet tous les registres a 0 .

*Pendant k coups d'horloge le multiplexeur 1 laisse passer l'information directement à la sortie.

En même temps **le circuit 1** permet de calculer les symboles de contrôle.

Démonstration : [3]

$$\left[\left(g0.c(x)D + g1.c(x)\right)D + g2.c(x)\right]D + i = g3.c(x) mod 2$$

$$\left[\left(g0.c(x)D + g1.c(x)\right).D + g2.c(x)\right].D + i = g3.c(x)$$

$$\left[\left(\boldsymbol{g0}.\boldsymbol{c}(\boldsymbol{x})\boldsymbol{D^2}+\boldsymbol{g1}.\boldsymbol{c}(\boldsymbol{x})\right).\boldsymbol{D}+\boldsymbol{g2}.\boldsymbol{c}(\boldsymbol{x})\right].\boldsymbol{D}+\boldsymbol{i}=\boldsymbol{g3}.\boldsymbol{c}(\boldsymbol{x})$$

$$[\boldsymbol{g0}.\boldsymbol{c}(\boldsymbol{x}).\boldsymbol{D^3}+\boldsymbol{g1}.\boldsymbol{c}(\boldsymbol{x}).\boldsymbol{D^2}+\boldsymbol{g2}.\boldsymbol{c}(\boldsymbol{x}).\boldsymbol{D}]+\boldsymbol{i}=\boldsymbol{g3}.\boldsymbol{c}(\boldsymbol{x})$$

$$\boldsymbol{i}=\boldsymbol{c}(\boldsymbol{x})[\boldsymbol{g3}+\boldsymbol{g0}.\boldsymbol{D^3}+\boldsymbol{g1}.\boldsymbol{D^2}+\boldsymbol{g2}.\boldsymbol{D}]$$
$$\boldsymbol{i}=\boldsymbol{c}(\boldsymbol{x}).\boldsymbol{D^3}\ (\boldsymbol{g0}+\boldsymbol{g1}\boldsymbol{D^{-1}}+\boldsymbol{g2}.\boldsymbol{D^{-2}}+\boldsymbol{g3}\boldsymbol{D^{-3}})$$

$$\boldsymbol{i}.\boldsymbol{D^{-3}}=\boldsymbol{c}(\boldsymbol{x})(\boldsymbol{g0}+\boldsymbol{g1}\boldsymbol{D^{-1}}+\boldsymbol{g2}.\boldsymbol{D^{-2}}+\boldsymbol{g3}\boldsymbol{D^{-3}})$$

On prend $D^{-1}=x$

$$i(x)x^3=c(x)\times(g0+g1x^1+g2x^2+g3x^3\ )$$

Et par conséquent $$\boldsymbol{C(x)}=\frac{\boldsymbol{x^{n-k}}\times\boldsymbol{i(x)}}{\boldsymbol{g(x)}}$$

A la fin de K coups d'horloge, le reste de la division sera stocké dans les registres.
Le multiplexeur 2 laisse passer les zéros pour pouvoir faire sortir le contenu de ces registres. Ce contenu représente <u>la redondance</u>.

IV- LE DECODAGE DE REED SOLOMON:

Plusieurs étapes peuvent être nécessaires pour le décodage de ces codes :

- Calcul du syndrome.
- Calcul des polynômes de localisation des erreurs et de d'amplitude.
- Calcul des racines et évaluation des deux polynômes.
- Sommation du polynôme constitué et du polynôme reçu pour reconstituer l'information de départ sans erreur.

1 - Calcul du syndrome :

Le calcul du syndrome peut être effectué par un processus itératif. Avant de pouvoir calculer le polynôme du syndrome, on doit attendre que l'on ait reçu tous les éléments du polynôme *r*(x).
Avec r(x) : le code que l'on reçoit. Et T(x) : le code transmis.

$$Si=r(d^{i})$$

Donc la forme polynomiale du syndrome est la suivante :

$$S(x) = \ldots + S_{2t+1}x^{2t} + S_{2t}x^{2t-1} + \ldots + S_2x + S_1$$

[2]

Remarque :

Si le code reçu n'est pas affecté par des erreurs alors tous les coefficients du syndrome seront nuls.

Exemple :

On a un code RS (15,9) donc 15-9=6 symboles de contrôle (2t=6).
Si on reçoit le polynôme suivant :

$$r(x) = \alpha x^{14} + \alpha^2 x^{12} + \alpha^{13} x^4$$

Le calcul du syndrome :

$$S_1 = r(x=\alpha) = \alpha\left(\alpha^{14}\right) + \alpha^2\left(\alpha^{12}\right) + \alpha^{13}\left(\alpha^4\right)$$
$$= \alpha^{1+14} + \alpha^{2+12} + \alpha^{13+4}$$
$$= 1 + \alpha^3 + 1 + \alpha^2$$
$$= \alpha^3 + \alpha^2 = \alpha^6$$

$$S_2 = r\left(x=\alpha^2\right) = \alpha\left(\left(\alpha^2\right)^{14}\right) + \alpha^2\left(\left(\alpha^2\right)^{12}\right) + \alpha^7\left(\left(\alpha^2\right)^4\right) = \alpha^7$$

$$S_3 = r\left(x=\alpha^3\right) = \alpha\left(\left(\alpha^3\right)^{14}\right) + \alpha^2\left(\left(\alpha^3\right)^{12}\right) + \alpha^7\left(\left(\alpha^3\right)^4\right) = \alpha^{12}$$

$$S_4 = r\left(x=\alpha^4\right) = \alpha\left(\left(\alpha^4\right)^{14}\right) + \alpha^2\left(\left(\alpha^4\right)^{12}\right) + \alpha^7\left(\left(\alpha^4\right)^4\right) = 0$$

$$S_5 = r\left(x=\alpha^5\right) = \alpha\left(\left(\alpha^5\right)^{14}\right) + \alpha^2\left(\left(\alpha^5\right)^{12}\right) + \alpha^7\left(\left(\alpha^5\right)^4\right) = \alpha$$

$$S_6 = r\left(x=\alpha^6\right) = \alpha\left(\left(\alpha^6\right)^{14}\right) + \alpha^2\left(\left(\alpha^6\right)^{12}\right) + \alpha^7\left(\left(\alpha^6\right)^4\right) = \alpha^8$$

2- Calcul des polynômes de localisation des erreurs et de d'amplitude:

Si on reçoit le polynôme suivant :

$$r(x) = \alpha x^{14} + \alpha^2 x^{12} + \alpha^{13} x^4$$

Initialisation : $r_0=x^{2t}$; $r_1=S(x)$

On a 2t=15-9=6

Donc $r_0=x^6$ Et

$$r_1(x) = \alpha^8 x^5 + \alpha x^4 + \alpha^{12} x^2 + \alpha^7 x + \alpha^6$$

On commence par diviser r_0 par r_1 :

$$x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 0x + 0 \quad \Big| \quad r_1(x) = \alpha^8 x^5 + \alpha x^4 + \alpha^{12} x^2 + \alpha^7 x + \alpha^6$$

$$x^6 + \alpha^8 x^5 + 0x^4 + \alpha^4 x^3 + \alpha^{14} x^2 + \alpha^{13} x \quad \Big| \quad \alpha^7 x + 1$$

$$\alpha^8 x^5 + 0x^4 + \alpha^4 x^3 + \alpha^{14} x^2 + \alpha^{13} x + 0$$

$$\alpha^8 x^5 + \alpha x^4 + 0x^3 + \alpha^{12} x^2 + \alpha^7 x + \alpha^6$$

$$\alpha x^4 + \alpha^4 x^3 + \alpha^5 x^2 + \alpha^5 x + \alpha^6$$

Donc

$r_2 = \alpha x^4 + \alpha^4 x^3 + \alpha^5 x^2 + \alpha^5 x + \alpha^6$

et

$$Q(x) = \frac{r_{i-2}(x)}{r_{i-1}(x)}$$

Donc $Q_2 = \alpha^7 x + 1$

***<u>Le polynôme de localisation des erreurs</u>** est défini par :

$$\sigma_i(x) = \sigma_{i-2}(x) + \sigma_{i-1}(x) Q_i(x)$$

Le polynôme de <u>localisation des erreurs intermédiaire</u> est :

$$\sigma_2 = \sigma_0(x) + \sigma_1(x) Q_2(x) = 0 + 1 * \alpha^7 x + 1 = \alpha^7 x + 1$$

Avec

$$\sigma_i(x) = B_i(x)$$

Initialisation de l'algorithme avec :

i=2

$B_{i-2}(x) = 0$

$B_{i-1}(x) = 1$

Et donc $B_0 = 0$; $B_1 = 1$

Puisque le degré de $r_2 = 4$ est supérieur à t=3,

- on continue l'algorithme

Division de r_1 par r_2 :

$$\begin{array}{l|l}
\alpha^8x^5+\alpha x^4+0x^3+\alpha^{12}x^2+\alpha^7x+\alpha^6 & \alpha x^4+\alpha^4x^3+\alpha^5x^2+\alpha^5x+\alpha^6 \\
\alpha^8x^5+\alpha^{11}x^4+\alpha^{12}x^3+\alpha^{12}x^2+\alpha^{13}x & \alpha^7x+\alpha^5 \\
\hline
\alpha^6x^4+\alpha^{12}x^3+0x^2+\alpha^5x+\alpha^6 & \\
\alpha^6x^4+\alpha^9x^3+\alpha^{10}x^2+\alpha^{10}x+\alpha^{11} & \\
\hline
\alpha^8x^3+\alpha^{10}x^2+x+\alpha &
\end{array}$$

Donc $r_3 = \alpha^8x^3+\alpha^{10}x^2+x+\alpha$
Et $Q_3 = \alpha^7x+\alpha^5$

On a degré $r_3=3$:

$$\sigma_3 = \sigma_1(x)+\sigma_2(x)Q_3(x) = 1+(\alpha^7x+1)(\alpha^7x+\alpha^5) = \alpha^{14}x^2+\alpha^2x+\alpha^{10}$$

- on continue l'algorithme :

Division de r_2 par r_3 :

$$\begin{array}{l|l}
\alpha x^4+\alpha^4x^3+\alpha^5x^2+\alpha^5x+\alpha^6 & \alpha^8x^3+\alpha^{10}x^2+x+\alpha \\
\alpha x^4+\alpha^3x^3+\alpha^8x^2+\alpha^9x & \alpha^8x+\alpha^{14} \\
\hline
\alpha^7x^3+\alpha^4x^2+\alpha^6x+\alpha^6 & \\
\alpha^7x^3+\alpha^9x^2+\alpha^{14}x+1 & \\
\hline
\alpha^{14}x^2+\alpha^8x+\alpha^{13} &
\end{array}$$

Deg $(r_4) = 2 < 3$ donc on arrête l'algorithme

Le dernier reste de la division est le polynôme d'amplitude cherché :

$$\omega(x) = \alpha^{14}x^2+\alpha^8x+\alpha^{13}$$

Le polynôme de localisation des erreurs cherché est :

$$\sigma_4 = \sigma_2(x)+\sigma_3(x)Q_4(x) = \alpha^7x+1+(\alpha^{14}x^2+\alpha^2x+\alpha^{10})(\alpha^8x+\alpha^{14}) = \alpha^7x^3+\alpha^9x^2+x+\alpha^7$$

Une fois le polynôme de localisation calculé, on évalue ses racines et ses dérivées qu'on effectue grâce à l'algorithme CHIEN SEARCH. [3]

On a le polynôme de localisation :

$$= \alpha^7 x^3 + \alpha^9 x^2 + x + \alpha^7$$

On doit essayer tous les éléments possibles de GF(2^4) :

Le calcul des racines :

$$\sigma\left(\alpha^i\right) = \alpha^7\left(\alpha^i\right)^3 + \alpha^9\left(\alpha^i\right)^2 + \left(\alpha^i\right) + \alpha^7$$

On obtient le tableau suivant :

Elément	Résultat
α	0
α^2	α^{12}
α^3	0
α^4	α^{12}
α^5	α^8
α^6	α^6
α^7	α^3
α^8	α^7
α^9	α^{13}
α^{10}	α^{11}
α^{11}	0
α^{12}	1
α^{13}	α
α^{14}	α^9
1	α^7

3- Calcul du polynôme d'erreur e(x) :

Une fois les différentes valeurs de ɯ(α^i) calculées, on applique un algorithme dit de Forney et qui est défini comme suit :

$$e_i = \frac{\omega\left(\alpha^i\right)}{\sigma'\left(\alpha^i\right)}$$

Avec :

w(d^i) : polynôme d'amplitude évalué pour les valeurs de $GF\left(2^4\right)$

σ'(α^i) : dérivées du polynôme de localisation des erreurs pour les valeurs de $GF\left(2^4\right)$

Et donc pour corriger les erreurs on effectue l'opération suivante: **T(x)=r(x)-e(x)**

Programme qui permet de réaliser le Code Reed Solomon :

```
k=11;  %
m=4;   % Nombre de bits dans un symbole
n=2^m-1;
%section codage REED_SOLOMON
msg=gf([1 2 3 4 5 6 7 8 9 10 11],m);
PRIM_POLY=19; %[1 0 0 1 1];représentation des coefficients du polynôme primtif par un nombre entier.
GENPOLY = RSGENPOLY(n,k,PRIM_POLY,0);%definit le polynôme générateur g(x) du code
mot_code = rsenc(msg,n,k,GENPOLY)
%Section decodage
errors = gf([3 2 0 0 0 0 0 0 0 0 0 0 0 0 0],m);%ajout de 2 erreurs
codeNoi = mot_code + errors
[dec,cnumerr] = rsdec(codeNoi,n,k,GENPOLY)
```

Simulation :

```
Array elements =

  Columns 1 through 11

     1     2     3     4     5     6     7     8     9    10    11

  Columns 12 through 15

     3     3    12    12


codeNoi = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

  Columns 1 through 11

     2     0     3     4     5     6     7     8     9    10    11

  Columns 12 through 15

     3     3    12    12


dec = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)

Array elements =

     1     2     3     4     5     6     7     8     9    10    11


cnumerr =

     2
```

Conclusion

Durant la période que j'ai passé a la Société Nationale de Radiodiffusion et Télévision, je me suis intéressée au Codage Reed Solomon qui permet de détecter et corriger les éventuelles erreurs qui interviennent dans un message.

J'ai constaté que le code détecteur et correcteur d'erreur Reed Solomon constitue un outil incontournable dans la transmission des données .

Et pour cela il fallait tout d'abord que je maitrise les notions fondamentales mathématiques et théoriques sur lesquels se base le code Reed Solomon.

Je voulais aussi faire un programme détaillé du codage Reed Solomon sous Matlab d'autant plus que les instructions Matlab Permettent de faciliter cette tache.

REFERENCES WEBOGRAPHIQUES:

[1] http://www.webreview.dz/IMG/pdf/ALI-PACHA.pdf

[2] http://www.sweegy.ch/documents/reports/Projet_Diplome_2005%20v1.1-dietler%20.pdf

http://documents.irevues.inist.fr/bitstream/handle/2042/2118/005.PDF%20TEXTE.pdf?sequence=1

http://documents.irevues.inist.fr/bitstream/handle/2042/2118/005.PDF%20TEXTE.pdf?sequence=1

REFERENCES BIBLIOGRAPHIQUES

Télévision numérique terrestre de la production à la diffusion par Ahmed Lichioui et Hamid Seriegh.

DVB-T/H Technologie et Impact Economique Nouveau mode de télédiffusion au maroc par Ahmed lichioui et Ahmed Mnijel.