DIXIEME COLLOQUE SUR LE TRAITEMENT DU
SIGNAL ET SES APPLICATIONS

NICE du 20 au 24 MAI 1985

LE CODAGE DU CODE DE REED SOLOMON (255,223) PAR UN MONOPROCESSEUR

S. Harari et P. Rabbizzoni

Universite de Toulon et du Var Avenue de l'Université 83130 La Garde

RESUME

Dans ce travail nous exposons de manière détaillée trois méthodes de codage du code de Reed Solomon (255,223):l'algorithme de division, l'algorithme de Berlekamp et un nouvel algorithme utilisant des précalculs.Nous donnons le nombre précis d'opérations nécéssaires pour calculer les symboles de parité d'un mot de code. Il apparaît en conclusion que la méthode de Berlekamp n'est pas optimale dans le cadre d'un monoprocesseur.

SUMMARY

In this work we give in a detailed fashion three methods for coding the Reed Solomon (255,223) code over F256. The usual division algorithm, the Berlekamp algorithm and a new method using precomputations. We give a precise evaluation of the necessary operations needed to obtain the parity check symbols of a codeword. It appears that the Berlekamp method is not optimal in the case of asingle processor.

1 1 INTRODUCTION

Dans un recent article (1) Berlekamp a propose un algorithme de codage pour les codes de Reed Solomon (255 223) sur le corps F256 optimise pour un e realisation en circuit integre.

Dans ce travail nous comparons cet algorithme avec l'algorithme classique du point de vue du nombre d'operations. Nous proposons un nouvel algorithme et l'etudions de ce point de vue

Dans cette meme optique nous examinons les differentes representations de F256 sur F2 et leur impact sur les operations elementaires.

2.1 L'ALGORITHME CLASSIQUE

L'Algorithme de division usuel necessite 32 multiplications et 32 additions par coefficient du quotient.

Au total il faut effectuer 7136 multiplications et 7136 additions pour obtenir les symboles de controle. Ces operations sont effectuees sur F256.

Trois classes de representations de F256 sur F2 sont possibles.

2.2 LES REPRESENTATIONS DE F256 sur F2

F256 est un espace vectoriel de dimension 8 sur F2. Ses elements sont donc des octets binaires. Le groupe multiplicatif du corps F256 n'est autre que Z256, ce qui conduit a la deuxieme representation. Enfin On peut considerer F256 comme l'espace des formes lineaires sur F256. et on obtient la troisieme representation.

2.2.1 Les representations Additives.

Dans cette representation les elements sont representes par des octets binaires. L'addition consiste alors en 7 additions suivies de 7 reduction modulo 2. Une reduction modulo 2 est constituee de une division, une multiplication et une soustraction. Ces operations sont les operations machines. Pour assurer la representation de la multiplication , on assimile chaque octet binaire a un polynome de degre 7 dans l'algebre des polynomes a coefficients binaires reduits modulo un polynome irreductible de degre 7. La multiplication de deux octets est constituee de 49 multiplications machines 49 additions machines suivies dela reduction modulo qui, en moyenne comporte 3 additions binaires suivies de 3 reductions modulo binaires.

2.2 La representation multiplicative.

Soit a un element primitif de F256: c'est a dire un generateur du groupe multiplicatif de F256.

Tout element non nul de F256 s'ecrit de maniere unique sous la forme y=a**x. On dit alors que x est le logarithme de y de base Ba.Le logarithme de Zech de y est par definition le logarithme de 1+y de base a.On a donc

$$a^{z(y)}=1+a^{x}.$$

L'element 0 est represente par un grand entier negatif.

Dans cette representation la multiplication s'effectue come suit: Soient y1 et y2 deux elements non nuls de F256.

y1= a**x1 et y2=a**x2.

Le produit y1.y2 a pour logarithme

y1.y2=a**(x1+x2) mod255.

La somme de deux elements y1 et y2 se calcule comme suit:

y1+y2=a**x1+a**x2
=a**x1(1+a**(x2-x1))
=a**(x1+Z(x2-x1)).

Les exposants sont pris modulo 255.
Si une table des elements Z(i) i=0...255 est preetablie, l'addition fait appel a une lecture de table, la multiplication se reduit a une addition modulaire.

Des tests sont necessaires pour traiter le cas ou y1 est egal a y2. La somme vaut alors l'entier negatif representant 0, car la caracteristique du corps est 2. De meme il faut tester la nullite de l'un des facteurs du produit de deux elements auquel cas le produit est nul.
Une multiplication dans cette representation est constituee d'une addition, suivie d'une reduction modulo d'entiers et d'un test.
Une addition dans F256 consiste en une addition, un appel en table , un test suivi d'une reduction modulaire d'entiers.

2.3 La Representation par Bases Traces Duales.

Soient (ei) et (ti) deux bases de F256. Elles sont traces duales si tr(ei.tj)= dij ou dij est le symbole de Kronecker, qui vaut 1 si i=j et 0 sinon.La trace est definie par

tr(x)=x + x**2+ x**4 +. . + x**128.

la trace d'un element est binaire.

Elle a la propriete suivante: soient a0,....a7 les composantes de a sur la base (ti). on a alors ai = tr(a.ei) pour tout i.
F256 est construit a l'aide du polynome

x**8 + x**7 + x**2 + x + 1.

et en prenant pour premiere base la base des puissnces d'une racine primitive 256eme de l'unite on obtient pour base duale la base suivante
b0 = a**99 , b1 = a**197 ,
b2 = a**203 , b3 = a**202 ,
b4 = a**201 , b5 = a**200 ,
b6 = a**199 , b7 = a**100 .

2.2.2 Applications aux produits dans F256

Soient x et y deux elements de F256. Soient (xi) les composantes de x dans la base (ai) et (yi) les composantes de y dans la base (bi). Les composantes de x.y dans la base (bi) sont les tr(x.y.a**i). Les calculs de ces traces s'effectue aisement.

tr(x.y) = xj.tr(y.a**j)
= xj.yj

Cette somme est un simple controle de parite.
Pour i non nul

tr(x.y.a**i) = xj.tr(y.a**i+j).

Le calcul de cete dernier trace est rapide...

3.1 L'algorithme de Berlekamp

Pour trouver le reste de la division euclidienne de a(x) par le polynome generateur g(x) il suffit de calculer les produits du quotient par g(x) : le coefficient du terme de plus haut degre de g(x) vaut 1.
Soit q un coefficient du quotient, l'algorithme calcule simultanement les produits

g0.q, g1.q, ... , g31.q .

Un registre Z est initialement charge par les coordonnees de q sur la base (bj).

Z(i) = tr(q.a**i)
pour i=0,....7.

La premiere coordonnee sur la base bj de chaque coefficient q.gl vaut gl.Z(i).
En effectuant les transferts suivants

Z(7)+Z(2)+Z(1)+Z(0) dans Z(7)
puis Z(i+1) dans Z(i)
pour i=0,...,6 on obtient le resultat cherche.

7 iterations de ce processsus achevent le calcul des produits.

3.2 Remarques.

La methode est amelioree en choisisant un polynome generateur symetrique afin de minimiser le nombre de coefficients distincts et donc le nombre de produits a calculer.

4 Le Nouvel Algorithme

Soit m(x) = mi.x**i un polynome d'information a coder. Notons par °m(x)/g(x)§ le reste de la division de m(x) par g(x). Le polynome m(x) ext la somme de ses monomes x**i, et ce sont ces monome qui interviennent dans le calcul du reste. Si on effectue un precalcul des °x**i/g(x)§ pour i=32....254 , on obtiendra des polynomes ri(x) de degre inferieur a 32

°m(x)/g(x)§ = mi*ri(x).

Aucune division polynomiale n'est necessaire au moment du codage car le degre des polynomes qui interviennent dans cette somme ne depasse 31, et donc ils representent leur propre reste dans une division par g(x) qui est de degre 32. Les principales etapes de l'algorithme sont les suivantes.

a) Precalcul

Etablissement d'une matrice R(223,32) a 223 lignes et 32 colonnes.
La iemme ligne de cette matrice est constituee des coefficients du polynome ri(x).

b) Operation de codage

soit m(x)= mi.x**i a coder.
°m(x)/g(x)§ = mi*ri(x).

Le ieme coefficient du reste final s'obtient comme une somme de 223 produits de 2 elements de F256. Au total pour coder un mot il faut effectuer 223.31 produits et 223.31 sommes dans F256.
Remarques.
La matrice R(i.j) a une structure de "registre a decalage" . En effet la ligne i+1 s'obtient a partir de la ligne par une permutation circulaire suivie d'une retroaction si le coefficient du terme de plus haut degre est non nul. Cette structure permet une eventuelle economie de memoire: il suffit de garder en memoire la premiere ligne de la matrice et de n'engendrer les suivantes. qu'au fur et a mesure que les coefficients parviennent a l'unite de codage.

5 Conclusion

Une comparaison a eté faite entre l'algorithme classique de codage et le nouvel algorithme propose par Berlekamp.Dans le cadre d'une realisation monoprocesseur le premier se revele plus performant . Nous proposons un nouvel algorithme plus performant que les deux precedents .

Bibliographie

°1§ E.R. Berlekamp Bit serial encoder for Reed Solomon Codes I.E.E.E. IT December 1983.

°2§ MacWilliams et Sloane The theory of Error Correcting Codes North Holland 1977.

Tableau Comparatif du nombre d'opérations sur F2.

Algorithme	Multiplications	Additions	app. Mém.	Division
Div. Pol.				
Berlekamp	631 536	751064.	299712	115 960
Précalculs	21408	49952	21408	42816

Tableau Comparatif du nombre d'opérations sur F256

Algorithme	Multiplications	Additions	Appels mémoire
Div. Polyn.	7136	7136	6913
Berlekamp	/	/	—
Précalculs	32	7136	7136.