Conception d'un turbo décodeur de code produit

Patrick Adde, Ramesh Pyndiah, Jean-René Inisan et Yves Sichez

Ecole Nationale Supérieure des Télécommunications de Bretagne
BP 832
29285 BREST CEDEX

Mail: Patrick.Adde@enst-bretagne.fr Ramesh.Pyndiah@enst-bretagne.fr
JR.Inisan@enst-bretagne.fr Yves.Sichez@enst-bretagne.fr

RÉSUMÉ

Cet article présente les derniers résultats concernant les turbo codes en blocs, obtenus à partir de codes BCH étendus. Après avoir rappelé l'algorithme itératif de décodage utilisé, nous évaluons la complexité des turbo décodeurs associés. Nous proposons des simplifications dans le calcul de la pondération et dans la mise en œuvre de l'algorithme de Chase. Ces simplifications nous ont conduits à concevoir une maquette qui nous permettra à terme de valider complètement les nouveaux concepts liés au turbo décodage des codes produits. Cette réalisation à partir de circuits prédiffusés programmables ouvre la porte à de futures intégrations sur le silicium, plus denses et plus rapides.

ABSTRACT

This paper presents the latest results on block turbo codes obtained from extended BCH codes. After a review of the iterative decoding algorithm used, we evaluate the complexity of the associated decoders. We propose simplifications in the weighting method and in implementing the Chase algorithm. With these simplifications, a prototype is being developed at E.N.S.T. de Bretagne. It will validate the new concepts introduced by block turbo decoders. This realization done from programmable gate arrays allow us to think that future silicon implementations are possible : they will be more compact and more rapid.

1 Introduction

Les turbo codes, inventés à l'E.N.S.T de Bretagne (département Electronique et département Signal et Communications), sont des codes correcteurs d'erreurs dont le pouvoir de correction, dans des conditions idéales de codage, avoisine la limite théorique prédite par C. E. Shannon. Depuis leur invention, en 1990, les turbo codes ont fait l'objet de brevets [1] [2] [3] [4] et de nombreuses publications [5]. Ils ont suscité l'intérêt de la communauté scientifique, notamment pour les applications de diffusion numérique de télévision, par voie hertzienne et satellite, et plus généralement pour les futures autoroutes de l'information.

Il a semblé très tôt, lors des premières études sur cette nouvelle famille de codes correcteurs d'erreurs mettant en œuvre des concepts très novateurs, qu'il était nécessaire d'en prouver rapidement la validité par une implantation sur silicium. Une première réalisation a été menée à son terme et un circuit VLSI a été inscrit au catalogue de la société COMATLAS sous le nom de CAS5093. Ce circuit, constitué à base de code convolutif, est actuellement en test dans de nombreux laboratoires et a rempli ses objectifs puisqu'il a permis de valider le concept des turbo codes. Il contient 2,5 modules pour deux codes de mémoire ν=3 concaténés. Un second circuit, TURBO4, a été conçu : il contient un module cascadable constitué de deux codes convolutifs à ν=4 et décode une itération.

Comme il a été montré dans [4] [6], un nouvel algorithme, reprenant les concepts de turbo décodage, permet le décodage itératif de code produit avec des résultats très intéressants en terme de taux d'erreurs. En particulier, ils semblent meilleurs que ceux obtenus par les turbo codes utilisant des codes convolutifs pour des rendements supérieurs à 0,6. Cet article permet d'évaluer en termes d'architecture et de complexité l'intégration sur le silicium de cet algorithme de turbo décodage et d'en présenter une réalisation à base de circuits prédiffusés programmables.

2 Algorithme de turbo décodage de code produit

Le code produit permet d'obtenir à partir de deux codes en blocs simples (faible distance de Hamming minimale δ) un code dont la distance de Hamming minimale est égale au produit des distances de Hamming des codes élémentaires utilisés et le rendement au produit des rendements élémentaires. Si on considère deux codes en blocs élémentaires $C_1(n_1, k_1, \delta_1)$ et $C_2(n_2, k_2, \delta_2)$, le code produit se présente sous forme de matrice C à n_1 lignes et n_2 colonnes où :

-les symboles binaires d'information sont représentés par une sous-matrice M à k_1 lignes et k_2 colonnes,

-chacune des k_1 lignes de la matrice M est codée par le code C_2,

-chacune des n_2 colonnes de la matrice C est codée par le code C_1.

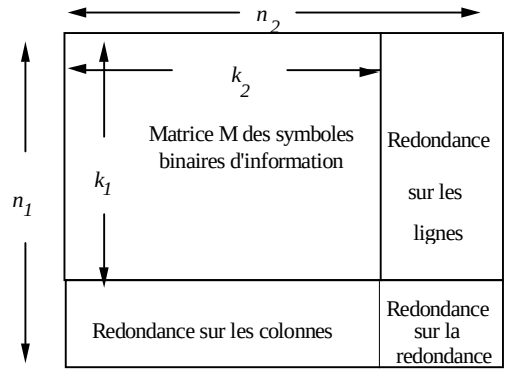


Figure 1 : Principe du codage d'un code produit.

L'algorithme de décodage proposé [4] [6] est un algorithme itératif. Son principe consiste à cascader des cellules élémentaires dont la mise en œuvre est illustrée par le schéma suivant :

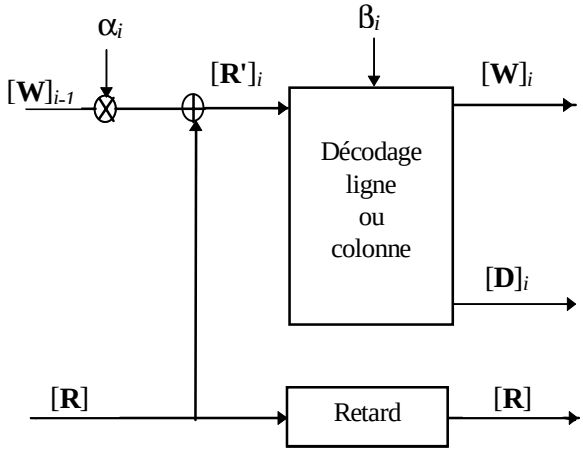


Figure 2 : Schéma de principe d'un décodeur élémentaire correspondant à une demi-itération.

où

un vecteur représente soit une ligne, soit une colonne du code produit,

[**R**] est le vecteur reçu,

$[\mathbf{W}]_{i-1}$ est le vecteur qui caractérise l'apport du décodage précédent (ou information extrinsèque),

$[\mathbf{R'}]_i = [\mathbf{R}] + \alpha_i [\mathbf{W}]_{i-1}$,

$[\mathbf{D}]_i$ est le vecteur "décision" (résultat binaire du décodage courant),

α_i et β_i sont des constantes, fonctions de la demi-itération courante.

Dans ce qui suit, nous allons considérer que C_1 et C_2 sont identiques et appartiennent aux codes BCH étendus.

Les étapes successives de décodage sont les suivantes :

1-Repérage des positions des composantes de [**R'**] les moins fiables, notées I_1, I_2 ...I_m, tel que :

$|R'_{I1}| < R'_j \quad \forall j \neq I_1.$

$|R'_{I2}| < R'_j \quad \forall j \neq I_1, I_2.$

....

2-Construction des séquences de test $[\mathbf{T}_k]$ qui sont des combinaisons de vecteurs de test élémentaires $[\mathbf{T}_k]^e$ ayant un 1 dans la position I_j et 0 ailleurs.

3-Pour chaque mot de test $[\mathbf{T}_k]$, calcul de $[\mathbf{Z}_k]$ tel que :

$[\mathbf{Z}_k] = [\mathbf{T}_k] \oplus$ signe [**R'**]

4-Décodage de $[\mathbf{Z}_k]$ par l'algorithme de Berlekamp et vérification que le résultat $[\mathbf{C}_k]$ est un mot de code, dont les composantes sont prises entre -1 et + 1.

5-Pour chaque vecteur trouvé, calcul de la distance euclidienne entre [**R'**] et $[\mathbf{C}_k]$:

$$M_k = \left\| [\mathbf{R'}] - [\mathbf{C}_k] \right\|^2$$

6-Sélection du mot de code $[\mathbf{C}^d]$ à distance minimale de [**R'**]. On a alors :

$[\mathbf{D}] = [\mathbf{C}^d]$ résultat binaire du décodage.

7-Calcul des fiabilités F_j pour chaque élément D_j de [**D**]:

pour celà, recherche d'un mot de code $[\mathbf{C}^c]$ à distance euclidienne minimale de $[\mathbf{C}^d]$ (appelé mot de code concurrent) et tel que : $C_j^d \neq C_j^c$

* si ce mot de code concurrent existe, alors

$$F_j = \left[\frac{M^c - M^d}{4} \right]$$

* sinon $F_j = \beta_i$.

8-Calcul des informations extrinsèques:

$$W_j = \left[F_j - C_j^d . R'_j \right] . C_j^d .$$

Les performances de cette méthode de turbo décodage sont reportées sur les courbes suivantes. Elles caractérisent le taux d'erreurs binaires (TEB) dans le cas d'une transmission sur un canal Gaussien utilisant un code produit et une modulation de phase à 4 états. Les symboles sont quantifiés sur 4 bits, dont un bit de signe.

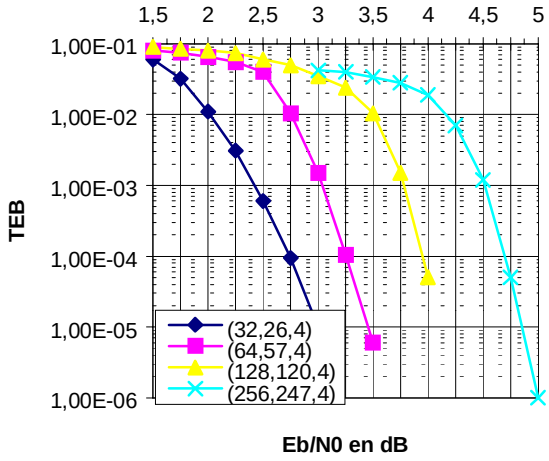


Figure 3 : Performances des codes produits après 4 itérations

3 Architectures et complexité des turbo décodeurs de code produit

Une étude demandée par le Centre Commun d'Etudes de la Télédiffusion et des Télécommunications (CCETT) a permis d'évaluer en terme de complexité l'intégration sur le silicium de cet algorithme pour un code produit obtenu à partir du code BCH(64,57,4) [7]. Elle a conduit à considérer deux types d'architectures [8][9] : à traitement séquentiel et à traitement par bloc, pour lesquelles ont été estimées les surfaces nécessaires en mémoires RAM (mémorisations de [**R**] et [**W**]) ainsi que celles requises pour l'unité de traitement. Le choix de l'architecture dépend essentiellement de l'application. Le débit prévu, la latence admise, le nombre d'itérations envisagé sont les facteurs à prendre en compte pour optimiser ce choix.

Afin d'évaluer de manière plus précise ces complexités, nous avons considéré comme bibliothèque de référence une bibliothèque CMOS précaractérisé 0,8µm ; les données [**R**] et [**W**] sont codées sur q=4 bits (un bit de signe et trois pour la fiabilité) et 16 vecteurs de test sont utilisés. Nous pouvons décomposer fonctionnellement l'algorithme de la manière suivante :

-lecture de [**R**] et de [**W**], calculs du syndrome et de la parité initiale (a)
-recherche des cinq bits les moins fiables (b)
-calculs des syndromes des vecteurs de test (c)
-décodage algébrique des vecteurs de test (d)
-calculs des distances euclidiennes (e)
-recherche de la distances euclidienne minimale (f)
-calcul de la fiabilité pour chaque symbole (g)

Dans le tableau ci-dessous sont reportées les surfaces nécessaires pour les deux types d'architectures.

Item Nombre de portes	(a)	(b)	(c)	(d)	(e)	(f)	(g)	Total
Structure à traitement séquentiel	8383	1140	1118	1420	4603	690	25423	42777
Structure à traitement par bloc	9385	19260	1118	12480	3008	690	56192	102133

Tableau 1 : Bilan du nombre de portes pour les deux architectures.

Avec la technologie cible envisagée, et si la pondération ne pose pas de problèmes au niveau routage, ces nombres de portes demandent une surface de 22 mm^2 (40 mm^2, y compris les mémoires RAM) pour la solution à traitement séquentiel et 52 mm^2 (70 mm^2 avec les RAM) pour la solution à traitement par bloc.

Ces résultats montrent que la méthode de pondération (calcul des fiabilités F_j) nécessite pour 4 itérations :

-59% de la surface totale (25000 portes sur 43000) dans le cas d'une structure à traitement séquentiel,
-55% (56000 portes sur 102000) dans le cas d'une structure à traitement par bloc.

Nous proposons une simplification de la mise en œuvre de ce calcul de la fiabilité, qui se concrétise par un gain de 22500 portes (90% de la surface initiale) pour la première structure et de 45000 portes (80% de la surface initiale) pour la seconde. Elle consiste à considérer, parmi les [$\mathbf{Z}_k$] décodés, le mot de code [$\mathbf{C}^c$] à distance euclidienne minimale de [$\mathbf{C}^d$]. Si on appelle M^c, la distance euclidienne entre [$\mathbf{C}^c$] et [$\mathbf{R}'$], le calcul de la fiabilité est le suivant:

*si le syndrome de [R'] = 0, alors $F_j = \gamma_i$,

*sinon $$F_j = \left[\frac{M^c - M^d}{4} \right]$$

si $C_j^d \neq C_j^c$

$$F_j = \beta_i \qquad \text{si } C_j^d = C_j^c$$

γ_i et β_i sont des constantes qui dépendent de i, demi-itération considérée ($\gamma_i > \beta_i$).

Nous donnons ci-contre les résultats de simulations pour un code produit BCH(64,57,4)⊗BCH(64,57,4) lors de la transmission sur un canal Gaussien d'une modulation à quatre états de phase (MDP4). Le décodage ligne puis colonne est effectué quatre fois (8 demi-itérations). Les symboles sont quantifiés sur 4 bits, y compris un bit de signe.

La courbe 1 correspond à l'algorithme initial.

La courbe 2 donne le TEB avec la simplification précédente.

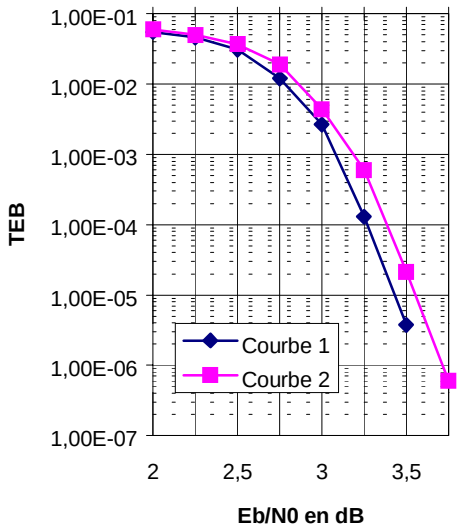


Figure 4 : Comparaison entre l'algorithme initial et l'algorithme simplifié.

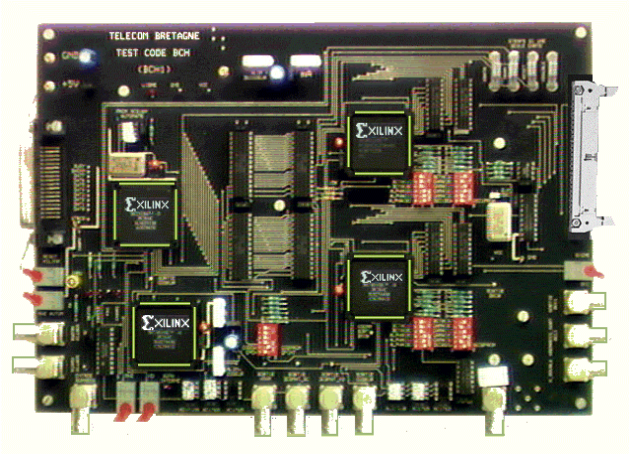
L'écart entre les 2 courbes varie entre 0.1 et 0.15 dB.

Toujours dans un souci de simplification, il est possible de réduire le nombre de vecteurs de tests nécessaires pour la mise en place de l'algorithme de Chase [10] : 8 vecteurs (au lieu de 16) apportent une dégradation de 0,05 dB et 4 vecteurs 0,3dB, par rapport à la courbe 2 et à un TEB = 10^{-5}.

Le gain en nombre de portes est cependant conséquent : pour chacun des items (sauf b), ce gain est à peu près de 2 (8 vecteurs) ou 4 (4 vecteurs).

4 Conception d'un turbo décodeur de code produit

Actuellement, un prototype est développé à l'E.N.S.T de Bretagne. Il valide les nouveaux concepts introduits par les turbo décodeurs et les simplifications précédemment décrites. Les caractéristiques de ce prototype sont :
-un code produit avec comme code élémentaire le code BCH(32,26,4),
-l'utilisation d'une structure à traitement séquentiel regroupant 4 itérations,
-l'algorithme de pondération simplifié utilisant 8 vecteurs de test,
-la programmation des valeurs α_i, β_i et γ_i.
Le décodeur élémentaire est implanté dans un circuit programmable FPGA Xilinx de 10000 portes (4010). Son fonctionnement a été validé par des simulations fonctionnelles VHDL, qui ont été comparées aux résultats donnés par les modèles de base, écrits en langage C. Cette modélisation VHDL a permis une synthèse logique entièrement automatique ainsi qu'une transcription immédiate en circuit programmable. Le nombre de portes effectif pour une demi-itération est environ 6100.
La maquette comprend un circuit réalisant la fonction de codage (FPGA Xilinx 4010), obtenu à partir d'une synthèse manuelle.
Les symboles bruités sont générés à partir d'un programme en C, chargés dans des mémoires de grandes tailles (256kbitsx4) et lus de manière aléatoire [11]. Une comparaison entre les résultats théoriques et pratiques devrait valider en terme de taux d'erreurs binaires ce type de décodeur.
Deux maquettes ont été réalisées :
-l'une comportant le codeur, le simulateur de canal gaussien et les deux décodeurs de la première itération : elle est représentée sur la photographie ci-dessous,



-l'autre regroupant les six décodeurs nécessaires aux trois itérations suivantes ; ce second circuit peut être dupliqué pour permettre d'augmenter le nombre d'itérations, si besoin est.

6 Conclusion

Cette étude a permis de montrer la faisabilité d'intégration sur le silicium de l'algorithme de décodage des codes produits présenté dans [4] et [6]. Les premiers résultats donnaient une complexité relativement importante pour les deux types d'architectures envisagées reprenant l'algorithme de décodage in extenso. Il est possible de réduire la surface nécessaire à cette intégration, sans pertes significatives de performances, en simplifiant la méthode de pondération et en utilisant deux fois moins de vecteurs de test.
Une intégration future peut donc être envisagée avec des circuits de dimension acceptable (pour quatre itérations, 50000 portes et 16 mémoires RAM de 4x1kbits sont nécessaires, ce qui donne 14mm^2 pour les unités de traitement et 12,5mm^2 pour les RAM en technologie CMOS 0,6µm, réseaux précaractérisés VLSI). Ces chiffres sont à diviser par 3, si la nouvelle technologie CMOS 0,35µm, est retenue.

Bibliographie

[1] C. Berrou, "Procédé de codage convolutif correcteur d'erreurs pseudo-systématique, procédé de décodage et dispositifs correspondants", brevet France n° 9105278, France Télécom, TDF, avril 1991.
[2] C. Berrou, "Procédé de codage correcteur d'erreurs à au moins deux codages convolutifs systématiques en parallèle, procédé de décodage itératif, module de décodage et décodeur correspondants", brevet France n° 9105280, Europe n° 92460013.3, USA n° 07/870,814, France Télécom, TDF, avril 1992.
[3] C. Berrou et P. Adde, "Procédé de décodage d'un code convolutif à maximun de vraisemblance et pondération des décisions, et décodeur correspondant", brevet France n° 9105279, Europe n° 92460011.7, USA n° 07/870,483, France Télécom, TDF, avril 1992.
[4] R. Pyndiah, A. Glavieux et C. Berrou, "Décodage itératif de codes produits", brevet France N°93 13858, France Télécom, novembre 1993.
[5] 'European Transactions on Telecommunications', Vol. 6, N°5, special issue on turbo decoding, Sept.-Oct. 1995.
[6] R. Pyndiah, A. Glavieux, A. Picard et S. Jacq, "Near optimum decoding of product codes", IEEE proc. of *GLOBECOM*'94, San Francisco, Nov. 1994.
[7] P. Adde, R. Pyndiah, O. Raoul, "Étude de la mise en oeuvre des turbo-codes en blocs. Évaluation de complexité matérielle", Rapport final, contrat CCETT n° 94 ME 19, marché d'étude de faisabilité sur les applications potentielles d'une nouvelle famille d'outils de codage de canal, décembre 1995.
[8] O. Raoul, P. Adde and R. Pyndiah, "Architecture et conception d'un circuit turbo-décodeur de codes produits," *GRETSI'95*, Juan-Les-Pins, Sept. 1995, pp 981-984.
[9] P. Adde, R. Pyndiah, O. Raoul, "Performance and complexity of block turbo decoder circuits," *Third International Conference on Electronics, Circuits and System ECECS'* 96, pp.172-175, 13-16 October 1996 - Rodos, Greece.
[10] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol IT-18, Jan. 1972, pp. 170-182.
[11] M. Jézéquel, C. Berrou, J. R. Inisan and Y. Sichez, "Test of a Turbo-Encoder/Decoder", TURBO CODING Seminar, Lund, Sweden, pp. 35-41, 28-29 August 1996.