

# Turbo décodage de code produit haut débit utilisant un code BCH étendu (128,106,8) $t=3$

J. Cuevas, P. Adde, S. Kerouédan, M. Arzel et J. Le Masson

ENST Bretagne, Technopôle Brest Iroise, CS 83818, 29238 BREST Cedex, France

[javier.cuevas@enst-bretagne.fr](mailto:javier.cuevas@enst-bretagne.fr), [patrick.adde@enst-bretagne.fr](mailto:patrick.adde@enst-bretagne.fr), [sylvie.kerouedan@enst-bretagne.fr](mailto:sylvie.kerouedan@enst-bretagne.fr), [matthieu.arzel@enst-bretagne.fr](mailto:matthieu.arzel@enst-bretagne.fr), [jerome.lemasson@enst-bretagne.fr](mailto:jerome.lemasson@enst-bretagne.fr)

**Résumé** – Cet article propose une nouvelle architecture de turbo décodage de codes BCH (128,106,8) à entrées et sorties pondérées corrigeant 3 erreurs. En utilisant le concept de parallélisme et les propriétés de la matrice générée par un code produit, cet article présente la conception et la complexité de trois décodeurs capables de traiter 2, 4 et 8 données en même temps pour obtenir le haut débit dans le décodage. Pour comparer les performances et la complexité entre les différents décodeurs, le langage C a été utilisé pour les simulations, le langage VHDL pour les simulations fonctionnelles et Synopsys Design Compiler pour la synthèse. Les résultats ainsi obtenus ouvrent la possibilité d'intégration sur le silicium de turbo décodeurs à fort pouvoir de correction (distance de 64, rendement de 0,8) et à haut débit (6,4Gbits/s).

**Abstract** – This paper presents a new turbo decoding architecture for BCH(128,106,8)  $t=3$  codes based on soft inputs and soft outputs. The design of each decoder is based on the concept of parallel decoding architecture and from the properties of product codes, this paper presents the design and complexity of three decoders able to process 2, 4 and 8 data at the same time to achieve very high data rates in decoding. To compare the performances and complexity of the decoders we have used language C for simulations, VHDL for functional simulations and Synopsys Design Compiler for the synthesis. The results obtained open the possibility future of integration on silicon of turbo decoders for high data rates (6.4 Gbits/s) with strong error correction power (minimum distance of 64, code rate of 0.8).

## I. Introduction

En 1994, R. Pyndiah [1][2] a proposé les turbo codes en blocs (TCB) comme une solution alternative aux turbo codes convolutifs (TCC) [3]. Le schéma de codage est basé sur la concaténation série de codes en blocs (code produit, introduit en 1954 par P. Elias [4]) et le décodage reprend le processus itératif obtenu à partir de décodeurs élémentaires à entrées et sorties pondérées.

Cet article présente, dans la partie II, les aspects fondamentaux de la théorie des TCB : leur construction, leur principe de décodage, le principe de décodage itératif et l'algorithme de décodage utilisé.

Dans la partie III, nous décrivons l'architecture d'un décodeur élémentaire BCH étendu (128,106,8) à entrées et sorties pondérées corrigeant 3 erreurs. Nous y présentons les différents blocs fonctionnels de ce décodeur élémentaire groupés en réception, traitement et émission des données. La complexité du décodeur et les performances de l'algorithme y sont également détaillées.

De cette architecture, nous faisons dans la partie IV usage du concept d'architecture parallèle pour la conception de trois décodeurs élémentaires qui traitent 2, 4 et 8 données en même temps, permettant d'obtenir des vitesses élevées pour le décodage.

Dans la partie V, nous présentons et comparons quelques résultats obtenus par simulation et synthèse des circuits pour donner finalement une conclusion du travail présenté dans cet article.

## II. Les codes produits

### II.1 Aspects fondamentaux des codes produits

Les codes produits sont construits par concaténation de deux ou plusieurs codes en blocs linéaires. Nous considérons deux codes en blocs élémentaires  $C_1$  et  $C_2$  caractérisés par les paramètres  $(n_1, k_1, d_1)$  et  $(n_2, k_2, d_2)$  où  $n_i$  représente la longueur du code,  $k_i$  la longueur du message et  $d_i$  la distance minimale de Hamming.

Le code produit  $C = C_1 \otimes C_2$  est représenté sous forme de matrice  $[C]$  avec  $n_1$  lignes et  $n_2$  colonnes où l'information binaire forme une sous-matrice  $[M]$  de  $k_1$  lignes et  $k_2$  colonnes; chacune des  $k_1$  lignes de  $[M]$  est codée par le code  $C_2$ ; chacune des  $n_2$  colonnes de  $[C]$  est codée par le code  $C_1$ .

Les paramètres du code produit résultant sont  $(n, k, d)$  où  $n=n_1 n_2$ ,  $k=k_1 k_2$  et  $d=d_1 d_2$ . Le rendement du code produit est aussi égal au produit du rendement des codes  $C_1$  et  $C_2$ . Une propriété importante de ces codes obtenus à partir de codes en blocs linéaires, est que si les  $n_2$  colonnes de  $[C]$  sont par construction des mots du code  $C_1$  et les  $k_1$  lignes de  $[M]$  des mots de code de  $C_2$ , les  $n_1 - k_1$  lignes restantes de  $[C]$  sont des mots du code  $C_2$ .

Par la suite, nous considérons que les codes  $C_1$  et  $C_2$  sont identiques. Nous utilisons spécifiquement le code  $C_1 = C_2 =$  BCH étendu (128,106,8). Le code produit  $C = C_1 \otimes C_2$  atteint une distance minimale  $d=d_1 d_2 = 64$ .

### II.2 Décodage itératif des codes produits

Le turbo décodage des codes produits est un algorithme itératif. Le principe consiste à placer des décodeurs élémentaires en cascade associés à un décodage à entrées et sorties pondérées. Le décodage de l'ensemble des lignes ou des colonnes correspond à une demi-itération effectuée par le décodeur élémentaire de la figure 1 :

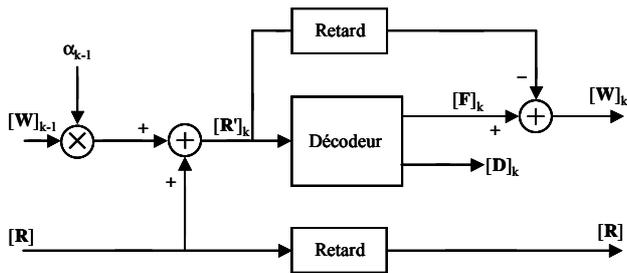


FIG. 1 : Schéma bloc d'une demi-itération

où  $[R]$  est le vecteur reçu du canal,  $[D]_k$  est le vecteur de décision binaire (résultat binaire du décodage courant),  $[W]_{k-1}$  est le vecteur d'information extrinsèque (apport du décodeur de la demi-itération précédente),  $[R]'_k = [R] + \alpha_{k-1} \times [W]_{k-1}$  et  $\alpha_{k-1}$  est le coefficient de contre-réaction. Ce coefficient pondère la contribution de l'information extrinsèque et sa valeur est moindre lors des premières itérations.

### II.3 L'algorithme de décodage

Nous décrivons les différentes étapes de l'algorithme de décodage Chase-Pyndiah [5][1], en détaillant plus particulièrement le décodage binaire :

- 1- Repérage de la position des 5 symboles binaires les moins fiables du mot reçu,
- 2- Construction de 16 vecteurs de test obtenus par inversion d'un certain nombre de symboles binaires parmi les 5 moins fiables,
- 3- Décodage binaire de chacun de 16 vecteurs de test obtenu à partir du signe du mot reçu. Celui que nous utilisons dans cette implémentation se décompose de la manière suivante :
  - calcul des composantes des syndromes  $S_1, S_3, S_5$ ,
  - calcul des coefficients  $\sigma_1, \sigma_2, \sigma_3$  du polynôme localisateur d'erreurs  $\sigma(X)$  par la méthode de Peterson [6],
  - recherche des racines du polynôme  $\sigma(X)$  par l'algorithme de Chien [7] qui permet de repérer la position des erreurs.
- 5- Calcul des distances euclidiennes de chaque vecteur de test et choix des vecteurs décidé et concurrents,
- 6- Calcul de la fiabilité de chacun des symboles du mot décidé,
- 7- Calcul de l'information extrinsèque  $[W]_k = [F]_k - [R]_k$ ,
- 8- Addition de l'information extrinsèque avec le mot reçu du canal,  $[R]'_k = [R] + \alpha_{k-1} \times [W]_{k-1}$ .

### II.4 Les performances de l'algorithme

Pour déterminer les performances de l'algorithme de turbo décodage, nous utilisons des simulations de type Monte-Carlo. Il s'agit de la transmission sur canal Gaussien d'une modulation de phase à 4 états, utilisant un code produit construit à partir des codes de longueur  $n=128$  et corrigeant 1 erreur (BCH 128,120), 2 erreurs (BCH 128,113) et 3 erreurs (BCH 128,106). Les résultats des simulations sont reportés sur les courbes de la figure 2. Les mots traités sont codés sur 5 bits : 1 bit de signe qui représente la valeur binaire du symbole et 4 bits de fiabilité qui représentent le niveau de confiance avec lequel ce bit est reçu.

On peut noter le pouvoir de correction important ainsi que la pente des courbes (23 décades/dB pour le turbo code construit à partir du code étendu BCH 128,106).

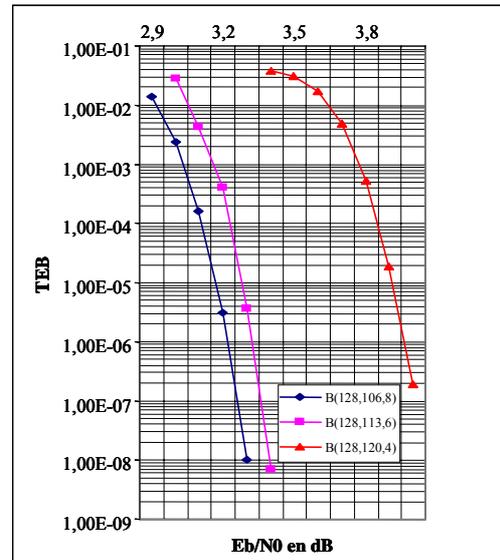


FIG. 2 : Taux d'erreur binaire TEB fonction du rapport signal/ bruit  $E_b/N_0$  des codes produits construits à partir des codes de longueur  $n=128$

## III. Nouvelle architecture turbo décodage de code produit utilisant un code BCH étendu (128, 106, 8) $t=3$

Depuis 1994, date à laquelle R. Pyndiah [1] a proposé l'algorithme de turbo décodage, les chercheurs de l'École Nationale Supérieure des Télécommunications de Bretagne travaillent sur l'étude de l'intégration sur silicium de l'algorithme de décodage itératif des codes BCH étendus de pouvoir de correction  $t=1$  (codes de Hamming) et  $t=2$  [8][9][10].

Dans la continuité de cette recherche, nous présentons dans cette partie une nouvelle architecture d'un décodeur élémentaire utilisant un code BCH étendu (128,106,8) de pouvoir de correction  $t=3$  (distance minimale du code produit  $d=64$ ). Ce décodeur élémentaire est constitué de trois grands blocs fonctionnels [9][11][12] : réception, traitement et émission.

**Les blocs de réception** : ces blocs travaillent au fur et à mesure de l'arrivée des symboles binaires du mot reçu. Les blocs de réception fonctionnent de manière indépendante les uns des autres et comprennent le calcul des syndromes  $S_1, S_3, S_5$ , les positions des symboles les moins fiables et le bloc mémoire qui stocke les données d'entrée.

**Les blocs de traitement** : les opérations séquentielles effectuées par ces blocs comprennent le calcul des coefficients du polynôme localisateur d'erreurs, les composantes des syndromes des vecteurs de test, le décodage binaire, le calcul des distances euclidiennes et le choix des vecteurs décidé et concurrents.

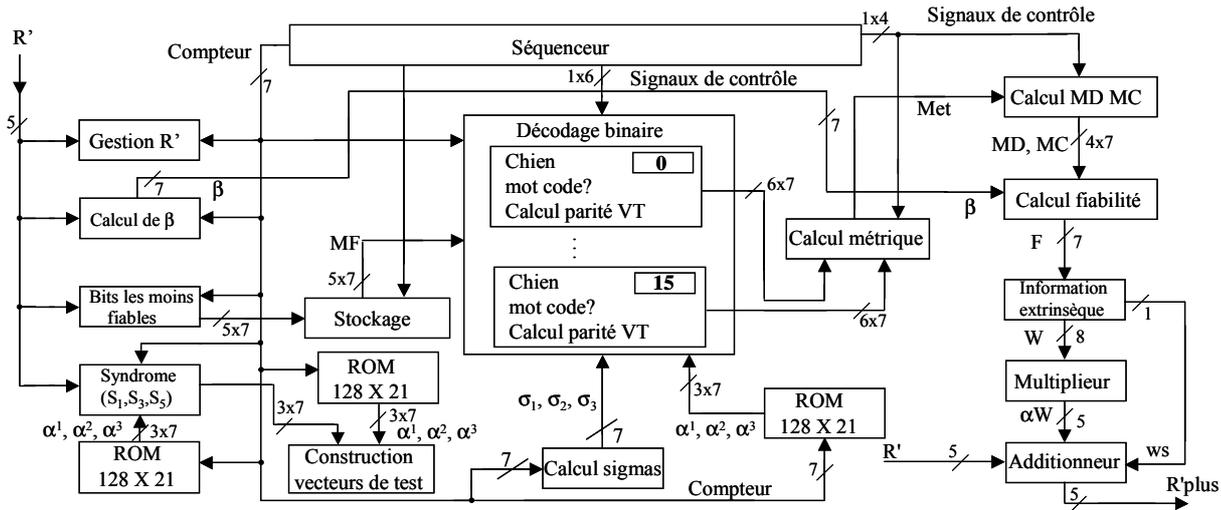


FIG. 3 : Schéma bloc du décodeur élémentaire

**Les blocs d'émission :** ces blocs travaillent au fur et à mesure de l'émission des symboles. Les blocs qui composent cette partie sont : le calcul des fiabilités, le calcul de l'information extrinsèque, la multiplication par le coefficient de contre-réaction  $\alpha$  et le calcul des données concernant l'itération suivante.

La figure 3 montre le schéma bloc du décodeur élémentaire. Six mémoires RAMs stockent et restituent les données. Les mémoires ROMs établissent la correspondance entre positions dans le mot de code et éléments du corps de Galois  $CG[2^7]$  (dans les deux sens). Le bloc « séquenceur » contrôle l'enchaînement des opérations dans le décodeur. Il fournit les signaux de contrôle et permet aux différents blocs de se repérer dans le temps. Le bloc décodage binaire est formé par trois sous-blocs : «Chien», qui repère la position des erreurs, «mot code», qui permet de vérifier si le mot du vecteur de test est un mot du code, et «calcul parité VT» qui calcule la parité du vecteur de test en traitement.

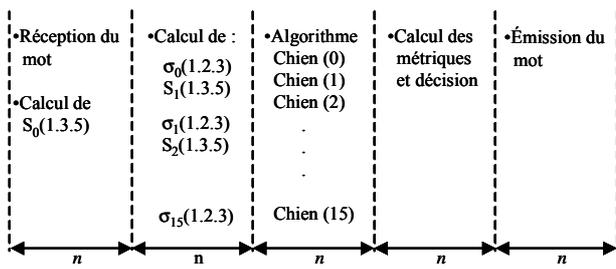


FIG. 4 : Décomposition temporelle du décodage

Les blocs, «calcul des syndromes  $S_1, S_3, S_5$ » et «construction des vecteurs de test», utilisent chacun une mémoire ROM de  $128 \times 21$  dans laquelle on stocke les éléments  $\alpha^1, \alpha^2$  et  $\alpha^3$  du corps de Galois utilisés par ces blocs. L'architecture du séquenceur repose sur un compteur par 128. Le temps de latence du décodeur est 512 périodes d'horloge ce qui correspond à 4 mots. La figure 4 montre la décomposition temporelle du codage.

## IV. Nouvelle architecture pour hauts débits

Dans cette partie, nous présentons la façon d'organiser les données dans la mémoire pour accélérer le processus de décodage [13]. Les nouvelles conceptions pour hauts débits que nous y présentons sont une extension de la figure 3, et traitent maintenant  $m=2, 4$  et  $8$  données à la fois [14].

### IV.1 Mémorisation des données

En augmentant le nombre de bits traités pendant une période de la fréquence débit, le débit global est alors augmenté. Nous considérons que 4 échantillons ( $m=2$ ) de la matrice initiale  $(i,j), (i,j+1), (i+1,j)$  et  $(i+1,j+1)$  constituent un mot de la nouvelle matrice qui a 4 fois moins d'adresses  $(I,J)$  mais des mots 4 fois plus grands.

Pour le décodage par ligne, les données  $(i,j), (i,j+1)$  sont traitées par un premier décodeur élémentaire (DE1), et les données  $(i+1,j)$  et  $(i+1,j+1)$  sont traitées par un second décodeur élémentaire (DE2). Pour le décodage par colonne, il faut prendre  $(i,j)$  et  $(i+1,j)$  pour DE1 et  $(i,j+1)$  et  $(i+1,j+1)$  pour DE2.

Le traitement de la matrice est 4 fois plus rapide que celui de la matrice initiale. En conclusion, si un mot de la nouvelle matrice contient  $m$  échantillons d'une ligne ou d'une colonne avec  $m$  décodeurs élémentaires, le traitement de la matrice est  $m^2$  fois plus rapide que celui d'une architecture traditionnelle [14].

### IV.2 Nouvelle architecture pour hauts débits

Le décodeur élémentaire «1-donnée» traitant une donnée est utilisé comme référence pour la conception de trois décodeurs élémentaires capables de traiter 2, 4 et 8 données à la fois. Le code utilisé est le code BCH étendu  $(128,106,8)$  à entrées et sorties pondérées corrigeant 3 erreurs. L'entrée des mots traités est codée sur 5 bits (1 bit de signe et 4 bits de fiabilité), 16 vecteurs de test et 3 concurrents.

Le décodeur élémentaire, qui traite 2 données à la fois, augmente la vitesse de décodage en réduisant le temps de traitement de chacune des 3 parties du décodeur analysées dans la partie III. Par exemple, la réception du mot exécuté par le

décodeur élémentaire de la figure 3 est effectuée en  $n=128$  périodes d'horloge. Ce même processus est exécuté par le décodeur élémentaire qui traite 2 données à la fois mais dans un temps moindre, c'est-à-dire en  $n=128/2$  périodes d'horloge. L'augmentation de la vitesse de décodage induit l'augmentation de la surface du décodeur élémentaire en faisant usage du concept d'architecture en parallèle [12]. En suivant ce même principe nous avons conçu les décodeurs qui traitent 4 et 8 données à la fois. Nous présentons les résultats obtenus dans la partie IV.3.

### IV.3 Complexité des décodeurs

Pour comparer les performances et la complexité entre les différents décodeurs, on a utilisé le langage C pour les modélisations et simulations, VHDL pour les simulations fonctionnelles et Synopsys Design Compiler pour la synthèse logique. Le tableau 1 montre la complexité, en fonction du nombre de portes ne comprenant pas la mémoire, de quatre décodeurs élémentaires où 1, 2, 4, et 8 représentent les nombres de données traitées dans le même temps. Nous prenons comme référence le décodeur de 1-donnée et pouvons ainsi donner nos conclusions des conceptions décodeurs de 2-, 4- et 8-données.

TAB. 1 : Bilan du nombre de portes pour quatre décodeurs élémentaires traitant  $m$  données à la fois

$m$	1	2	4	8
Nombre de portes	40 800	56 500	81 800	120 300

La synthèse pour ces 4 décodeurs donne à peu près le même chemin critique de 100 ns avec la technologie ST Microelectronics CMOS 0,18 $\mu$ m, ce qui correspond à une fréquence de fonctionnement de 100MHz. Le décodeur fonctionnant par octets utilise 3 fois plus de surface, avec un débit 8 fois plus grand. En parallélisant 8 décodeurs (qui prennent en compte les 64 données d'une adresse mémoire), on obtient pour une surface silicium 24 fois plus grande, un débit de 6.4Gbits/s (64 fois celui du décodeur initial «1-donnée»).

## V. Résultats et Conclusions

Nous avons proposé une nouvelle architecture de décodage des codes produits utilisant des codes BCH (128,106,8) corrigeant 3 erreurs, à haut pouvoir de correction et fort rendement ( $R$  proche de 0,8). Cette architecture présente l'avantage de n'utiliser **qu'un seul plan mémoire** ( $n^2$  échantillons) en entrée.

En utilisant le concept d'architecture parallèle, nous avons présenté la conception et la complexité de trois décodeurs capables de traiter 2, 4 et 8 données à la fois pour obtenir un décodage haut débit. Des résultats de la synthèse, nous concluons que si la surface du décodeur « 1-donnée » est  $S_1=40\ 800$  portes, alors la surface des décodeurs 2-données, 4-données et 8-données est  $S_2=1,4.S_1$ ,  $S_4=2.S_1$  et  $S_8=3.S_1$ . Ces résultats ouvrent des possibilités concernant l'intégration

future sur silicium de décodeur de code produit à haut pouvoir de correction, haut rendement et haut débit.

## Références

- [1] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, « *Near optimum decoding of product codes* » in proc. of IEEE GLOBECOM '94 Conference, vol. 1/3, Nov.- Dec. 1994, San Francisco, pp. 339-343 .
- [2] R. Pyndiah, « *Near optimum decoding of product codes : Block Turbo Codes* », IEEE Trans. on Comm., vol 46, n° 8, August 1998, pp. 1003-1010.
- [3] C. Berrou, A. Glavieux and P. Thitimajshima, «Near Shannon limit error-correcting coding and decoding : Turbo-codes (1), » *IEEE Int. Conf. on Comm. ICC' 93*, vol 2/3, May 1993, pp. 1064-1071.
- [4] P. Elias, « *Error-free coding* », IRE Trans. on Inf. Theory, vol. IT-4, pp. 29-37, Sept. 1954.
- [5] 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.
- [6] W. W. Peterson, « *Encoding and error correcting procedures for the Bose-Chaudhuri codes* », IRE Transf. Theory, vol IT-6 pp 459-470, Sept. 1960.
- [7] R.T. Chien, « *Cyclic decoding procedures for the Bose-Chaudhuri Hocquenghen codes* », IEEE Tranf. Theory, vol. IT 10, pp 357-363, Oct. 1964.
- [8] O. Raoul, P. Adde, R. Pyndiah, « *Architecture et conception d'un circuit turbo-décodeur de codes produits* », Quinzième Colloque sur le Traitement du Signal et des Images (GRETSI'95), Tome 2, pp. 981-984, 18-21 septembre 1995.
- [9] P. Adde, S. Kerouédan, JR Inisan, « *Conception d'un décodeur BCH (30,19,6) à entrées et sorties pondérées : application au turbo décodage* », Dix-septième Colloque GRETSI, pp. 103-106, 13-17 septembre 1999, Vannes, France.
- [10] S. Kerouédan and P. Adde, « *Implementation of a Block Turbo Coder on a single chip* », 2<sup>nd</sup> International Symposium on Turbo Codes and Related Topics, Brest, France, 2000, pp. 243-246.
- [11] P. Adde, R. Pyndiah, O. Raoul, « *Performance and complexity of block turbo decoder circuits* », Third International Conference on Electronics, Circuits and System ICECS'96, pp. 172-175, 13-16 October 1996 - Rodos, Greece.
- [12] P. Adde, R. Pyndiah, O. Raoul and J.R. Inisan, « *Block turbo decoder design* », Int. Symposium on turbo codes and related topics, Brest, Sept. 1997, pp. 166-169.
- [13] P. Adde et R. Pyndiah, « *Architecture de décodeur de code produit haut débit* », Grets'i'01, Toulouse, France, 10-13 Sept. 2001.
- [14] J. Cuevas, P. Adde, S. Kerouédan, R. Pyndiah, « *New architecture for high data rate turbo decoding of product codes* », GLOBECOM 2002, Taipei, Taiwan, pp. 139-143, November 17-21.