HAL
open science

Implantation d'un décodeur à entrées pondérées du code à résidus quadratiques étendu (48,24,12)

Patrick Adde, Raphaël Le Bidan, Magali Le Gall

To cite this version:

Patrick Adde, Raphaël Le Bidan, Magali Le Gall. Implantation d'un décodeur à entrées pondérées du code à résidus quadratiques étendu (48,24,12). GRETSI 2013 : 24 ème colloque 24ème colloque du Groupement de Recherche en Traitement du Signal et des Images, Sep 2013, Brest, France. hal-00869244

HAL Id: hal-00869244

https://hal.science/hal-00869244v1

Submitted on 2 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IMPLANTATION D'UN DECODEUR A ENTREES PONDEREES DU CODE A RESIDUS QUADRATIQUES ETENDU (48, 24,12)

Patrick ADDE, Raphael LE BIDAN et Magali LE GALL
Institut Mines-Télécom; TELECOM Bretagne; CNRS Lab-STICC UMR 3192 Technopôle Brest Iroise, CS 83818
29238 Brest Cedex 3
Université Européenne de Bretagne, France email: prénom.nom@telecom-bretagne.eu

Résumé --- Les codes à résidus quadratiques RQ sont une famille de codes en blocs linéaires cycliques connus pour avoir de bonnes propriétés de distance. Ce papier explore une solution de conception d'un décodeur pondéré pour le code RQ étendu de paramètres (48,24,12). Nous décrivons un algorithme qui exploite la propriété d'auto-dualité pour simplifier le décodage. Les résultats de simulation montrent que l'algorithme proposé réalise un très bon compromis entre performance et complexité, et que ce code constitue ainsi une alternative attractive au code convolutif à 64 états ainsi qu'au code BCH (127,64,21). Nous présentons une architecture de décodage à entrée pondérée pour le code (48,24,12), conçue et implantée avec succès sur une cible FPGA.

Abstract --- Quadratic residue (QR) codes are cyclic block codes with good distance properties. This paper considers the design of an efficient soft-decision decoder architecture for the (48,24,12) extended QR code. We first describe a soft-decision decoding algorithm that takes advantage of the self-dual property of this code to reduce decoding complexity. Simulation results show that this code compare favorably to the 64-state convolutional code and (127,64,7) BCH code in terms of performance and decoder complexity. A hardware soft-decision decoder architecture is then introduced and validated on FPGA.

I. Introduction

Les codes correcteurs d'erreur (CCE) permettent d'améliorer significativement les performances des systèmes de communications numériques. De nos jours, des CCE très puissants tels que les turbocodes [1] ou bien les codes LDPC (Low-Density Parity-Check) [2] approchent de très près les limites théoriques promises par la théorie de l'information pour des longueurs de codes suffisamment grandes, sur de nombreux modèles de canaux. Mais pour des blocs de données relativement courts (quelques centaines de bits ou moins), un code en bloc classique couplé à un décodeur à entrées pondérées peut constituer une alternative intéressante aux techniques de codage avancées de type Turbo ou LDPC, ainsi qu'à des solutions plus conventionnelles (code convolutif couplé à un décodeur de Viterbi [3] par exemple).

Cela nécessite à la fois des codes en blocs avec de grandes distances minimales, pour garantir de bonnes performances asymptotiques, ainsi que des architectures de décodage de faible complexité. Le décodage à entrées pondérées optimal au sens du maximun de vraisemblance (Maximum-likelihood, ML) offre les meilleures performances de décodage mais, très souvent, ne peut être mis en œuvre pour des raisons de complexité. Des algorithmes de décodage sous-optimaux lui sont alors préférés. L'une de ces approches exploite ainsi la fiabilité des symboles reçus pour limiter la recherche du mot de code le plus probable à une liste réduite de candidats. Cette catégorie inclut notamment l'algorithme proposé par Chase [4]. Récemment, nous avons proposé dans [5] une solution apparentée (on pourrait l'appeler algorithme de Chase bidirectionnel) permettant de décoder des codes en blocs linéaires systématiques de rendement ½ possédant une matrice de parité **H,** composée d'une sous-matrice **P** inversible. Cet algorithme s'applique notamment aux codes auto-duaux. Il s'avère par ailleurs particulièrement intéressant pour les codes courts en offrant la possibilité d'atteindre des performances quasi-ML avec une faible complexité de décodage.

La suite de ce papier est organisée comme suit. La section II introduit l'algorithme de décodage proposé et ses performances. L'architecture du décodeur (48,24,12) est présentée dans la section III. La réalisation d'un prototype FPGA de ce décodeur est ensuite décrite en section IV, accompagnée d'une réflexion sur les architectures futures.

II. Décodage pondéré des codes auto-duaux

L'algorithme de décodage pondéré introduit en [5] applique une série de N_{eps} vecteurs de test pour tenter d'éliminer les erreurs localisées dans les positions les moins fiables parmi les k bits d'information. Chaque séquence d'information candidate ainsi créée est alors ré-encodée pour produire un mot de code candidat. La même procédure est appliquée en parallèle sur les k bits de parité, en utilisant les équations de codage inverse servant à calculer les k bits d'information à partir des k bits de parité. Une seconde liste N_{epr} mots de code candidats est ainsi obtenue. Le décodeur sélectionne finalement le mot à distance euclidienne minimale du mot reçu dans la liste des mots candidats. L'algorithme conduit à une implantation matérielle particulièrement simple pour des codes de petite dimension (k<32); il ne nécessite par

ailleurs ni pivot de Gauss, ni décodeur algébrique. La stratégie de génération des vecteurs de test est similaire à celle présentée dans [6].

Sur la Figure 1, on voit que les performances en terme de taux d'erreur binaire (TEB) du code (48,24,12) pour une transmission QPSK sur canal à bruit additif blanc gaussien, avec N_{ep} =1536, sont très proches des performances ML (0,1 à 0,15 dB). Rappelons ici qu'en comparaison, une implantation « naïve » du décodage ML nécessite l'examen de 2^{24}≈16,8 millions de vecteurs de tests. Le nombre de bits de quantification à l'entrée du décodeur est q=8 bits, bit de signe compris.

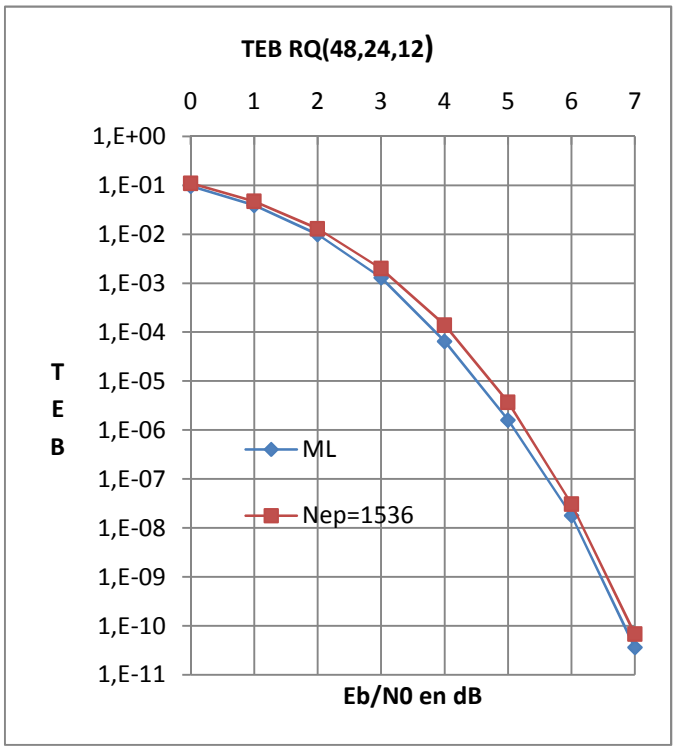

Figure 1- *Performances du décodage pondéré du code RQ(48,24,12) avec Nep =1536*

Les performances quasi-ML sont obtenues en choisissant de manière optimale les N_{ep} =1536 vecteurs de test car la complexité de décodage est directement reliée au nombre ainsi qu'à la construction de ces vecteurs. Ces inversions sont réalisées à l'aide de masques stockés en mémoire, chaque masque présentant un '1' à la position d'une inversion et des '0' ailleurs. Les masques d'inversion choisis pour le décodeur (48,24) en repérant les 9 bits les moins fiables dans la partie données et la partie redondance sont les suivants :

- Pas d'inversion
- 1 inversion parmi les 9 bits les moins fiables
- 2 inversions parmi les 9 bits les moins fiables
- 3 inversions parmi les 9 bits les moins fiables
- 1 inversion parmi les bits les plus fiables (bits non triés de la position 10 à la position 24)
- 4 inversions parmi les 9 bits les moins fiables
- 5 inversions parmi les 9 bits les moins fiables
- 1 inversion parmi les 9 bits les moins fiables combinée à 1 inversion parmi les bits les plus fiables
- 2 inversions parmi les 6 premiers bits les moins fiables, combinées avec 1 inversion parmi les bits les plus fiables
- On complète ensuite les vecteurs jusqu'à en obtenir 768 (soit 1536, 768 pour les données et 768 pour les parités).

Ces 768 masques sont subdivisés en 16 matrices pour réaliser le traitement en parallèle.
A un instant donné, on traite donc simultanément 16 vecteurs sur la partie « données » et 16 vecteurs sur la partie « redondances ». Le traitement d'un mot se fait donc en n=48 périodes d'horloge (32xn = 1536).

III. Architecture d'un décodeur pondéré pour codes à résidus quadratiques

Nous décrivons l'implantation numérique d'un décodeur appliquant l'algorithme de Chase bidirectionnel proposé au cas particulier du code RQ étendu (48,24) (Figure 2). Le décodeur utilise 8 bits de quantification (bit de signe compris), opère sur les L_{rs} = L_{rp} = 9 bits les moins fiables, et utilise N_{ep} = N_{eps}+N_{epr} = 1536 vecteurs de test (N_{eps} = N_{epr} = 16*n : 16 blocs d'inversion pour les données et 16 blocs d'inversion pour les parités).
L'architecture du décodeur est constituée de quatre blocs : réception, traitement, transmission et contrôle. Dans la partie réception, les n = 48 symboles binaires du mot reçu sont traités séquentiellement. Ainsi, ce bloc repère successivement les L_{rs} = 9 et L_{rp} = 9 symboles les moins fiables dans la partie systématique et la partie redondance du mot reçu. En parallèle, un registre à décalage entrée série/sorties parallèles mémorise séquentiellement les 48 symboles. La partie traitement comprend trois tâches principales concernant la partie « données » et la partie « redondance »:
- génération des vecteurs de tests en inversant les bits de signe du mot reçu aux positions les moins fiables,
- ré-encodage de ces vecteurs,
- sélection du mot à distance euclidienne minimale par calcul des métriques de chacun des vecteurs de test.

Il faut noter que le traitement des N_{eps} vecteurs de test sur la partie information et le traitement des N_{epr} vecteurs de test sur la partie parité se font en parallèle. Comme 48 symboles binaires sont nécessaires au fonctionnement du registre à décalage entrée série/sorties parallèles, la métrique de chaque vecteur de test ré-encodé peut être directement calculée. L'unité de transmission se compose uniquement d'un registre à entrées parallèles/sortie série : elle retransmet séquentiellement la partie systématique du mot binaire décodé. Les trois blocs précédents sont supervisés par un bloc de contrôle. Dans notre conception, un compteur modulo 48 est suffisant pour générer les signaux nécessaires à ce contrôle. Le premier étage traite séquentiellement n = 48 symboles binaires en 48 périodes d'horloge. Le second étage est composé de deux blocs principaux. N_{eps}

et N_{epr} vecteurs de test sont générés, codés et comparés en 48 périodes d'horloge. Finalement, les 24 bits décodés sont transmis séquentiellement en 24 (ou 48) périodes d'horloge. La latence du décodeur, L, peut être définie comme le nombre de symboles traités par le décodeur durant le temps nécessaire au décodage d'un symbole. Elle dépend du nombre de niveaux de pipeline et de la longueur n du mot. Dans notre exemple, la latence résultante est de $L = 2n = 96$ périodes d'horloge. Cette architecture optimise la latence du décodeur, permet un débit de décodage optimal et donne des performances quasi-ML.

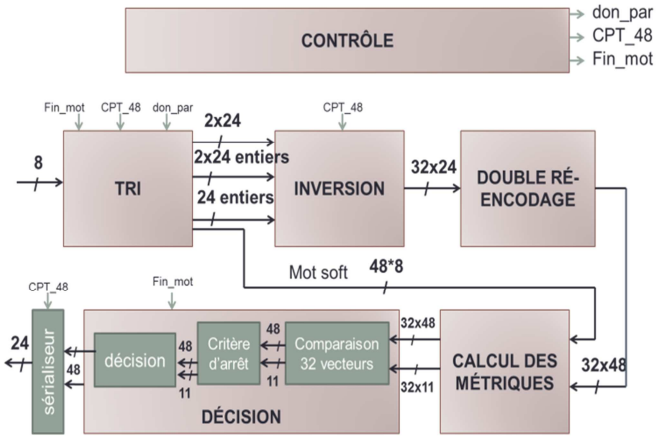

Figure 2- *Schéma bloc du décodeur considéré*

Au moment du calcul des métriques, un bloc « Critère d'arrêt » est mis en œuvre : si une métrique est inférieure à une certaine valeur MIN on considère qu'elle correspond à la solution optimale (mot à distance minimale) et le processus de traitement est stoppé (MIN= fiabilité maximale des symboles entrants/2 = 63 si q=8).

Le bloc de contrôle est complété par deux machines d'état, l'une gérant la réception du mot à décoder et l'autre le traitement et la sérialisation du mot décodé en sortie. La machine de réception contrôle le traitement en envoyant un signal 'début_calcul' à la machine de traitement dès la réception complète du mot à décoder. Ce signal indique que l'ensemble du mot est disponible (soit une période d'horloge après réception complète de l'observation). Les deux machines sont couplées à des compteurs par 48 permettant de générer les différents signaux de contrôle.

La synthèse logique pour l'estimation de la complexité circuit a été réalisée avec l'outil de Synopsys, pour une technologie cible ASIC 90 nm de STMicrolectronics. Pour le décodeur considéré, un peu moins de 120 000 portes sont nécessaires lorsque l'on prend q = 8 bits de quantification, en considérant la fréquence du débit égale à la fréquence de traitement f_0 (Table 1).

On peut constater que les blocs « construction des vecteurs de tests » et « Calcul des métriques et sélection » sont les plus complexes. On peut réduire cette complexité en diminuant le débit sans changer f_0 , et/ou en diminuant q, le nombre de bits de quantification. Par exemple, prendre la fréquence débit = $f_0/16$ permet d'utiliser 16 fois moins de blocs d'inversion (donc gain important), mais, si le critère d'arrêt n'est pas mis en marche, la latence est (16+1)n.

Table 1*: Complexité de l'architecture de décodage pondéré proposée*

ST Microlectronics 90 nm CMOS	Fonctions du décodeur soft	Nombre de portes équivalent
Bloc de réception	Bits les moins fiables	2900
	Regitre série/parallèle 48x8	7066
Bloc de traitement	Construction des vecteurs de test	1730
	Ré-encodages direct et inverse	9405
	Calcul des métriques et sélection	93660
Bloc d'émission	Registre parallèle/série 24x1	130
Bloc de contrôle	Compteur 6 bits	475
Complexité	**115366**	

De même, prendre q=4 permet de diviser à peu près par 2 la complexité globale.
La Figure 3 permet de vérifier que, dans ce cas, les performances du code sont faiblement dégradées pour q=4 (bit de signe compris).

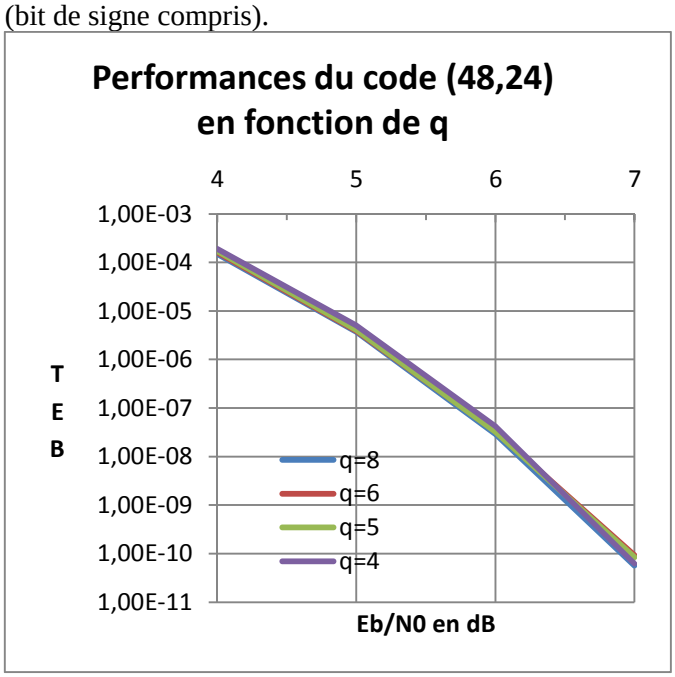

Figure 3- *Performances du décodage en fonction du nombre q de bits de quantification*

IV. Prototype FPGA du décodeur pondéré

Afin de valider le décodeur pondéré considéré, des mesures de TEB ont été réalisées sur une maquette expérimentale construite à partir d'un produit de Dinigroup DN9000K10PCI qui contient 6 Virtex5 LX330 de Xilinx. Il occupe 30510 LUTs, 6864 Blocs Flips-Flops. La fréquence maximale de fonctionnement avant placement /routage est 66MHz, permettant un débit d'entrée de 66 Mb/s. Le chemin critique se trouve dans le bloc « sélection de la métrique la plus faible (parmi 32) ». En

diminuant le nombre de blocs d'inversion(2*16), ce débit sera plus grand (ainsi que la latence). Si on prend en compte le rendement R=1/2 du code, le débit des données est donc de 33 Mb/s. Un seul circuit a été nécessaire pour le décodeur.

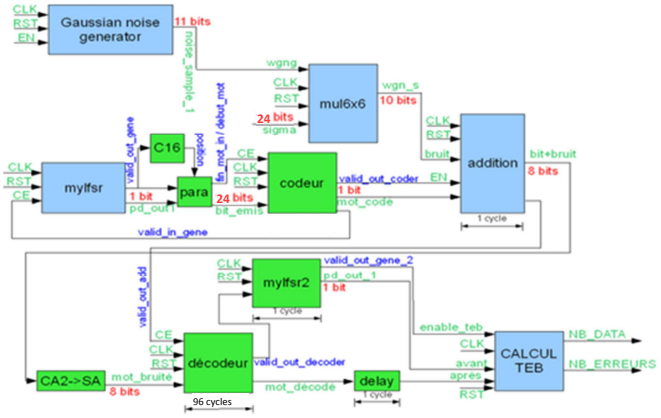

Figure 4: Schéma bloc de la chaîne de transmission mise en œuvre

Un générateur pseudo aléatoire envoie une donnée à chaque période de l'horloge f_0. Un codeur calcule la parité associée à chaque paquet de 24 données. Données et parités sont modulées en QPSK (opération réalisée par les blocs addition et CA2→SA) puis bruitées par un bruit additif blanc gaussien (canal AWGN émulé en utilisant la méthode de Wallace). La synthèse de l'ensemble de la chaîne n'utilise qu'un seul circuit VIRTEX5 et la fréquence de fonctionnement est de 64Mhz

La mesure des performances du décodage pondéré du code RQ étendu (48,24,12) a permis de valider l'architecture proposée. Ces performances ont été comparées avec des simulations de type Monte-Carlo produites par un modèle en C (Figure 1). Elles correspondent aux résultats donnés en [7]. Dans ce même ouvrage, une comparaison de trois codes de rendements proches de ½ montre que le décodage pondéré du code (48,24) présente des performances tout à fait comparables à celles du décodage de Viterbi du code convolutif de rendement ½ à 64 états, et supérieures à celles du décodage algébrique du code BCH (127,64). Ces différentes performances sont reproduites sur la Figure 5.

V. Conclusion

Nous avons décrit un algorithme de décodage à entrée souple pour le code RQ étendu de paramètres (48,24,12), ainsi que sa mise en œuvre matérielle sur cibles ASIC et FPGA. Les résultats de simulation, proches de l'optimal (ML), et la complexité du circuit mis en œuvre dans un prototype démontrent la faisabilité et l'intérêt de la solution de codage proposée pour les transmissions utilisant des blocs courts, comparativement à d'autres solutions concurrentes telle que le code convolutif de rendement ½ à 64 états par exemple.

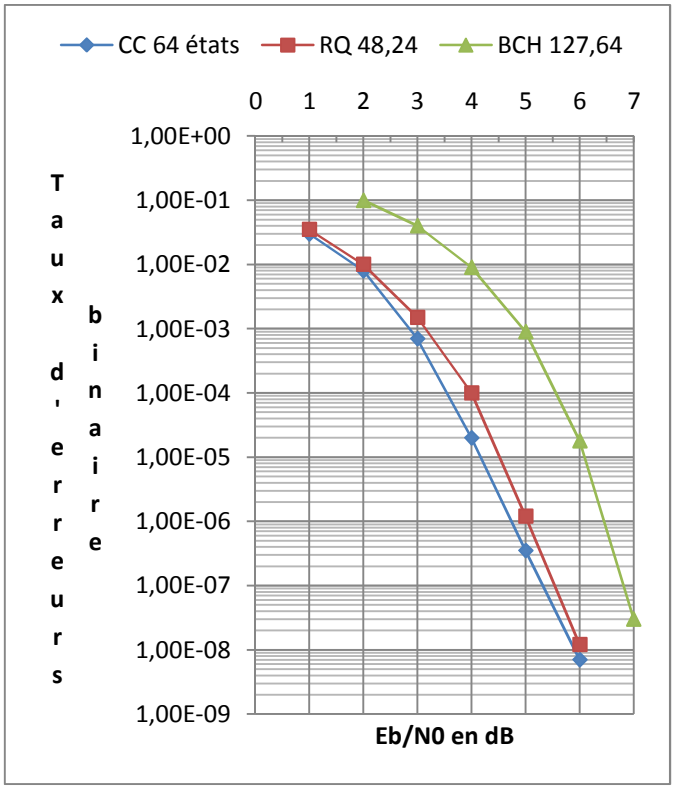

Figure 5-*Performances de différents codes de rendement ½ : convolutif 64 états, RQ (48,24) et BCH 127,64* [7]

Remerciements

Les auteurs tiennent à remercier Xavier Wiklund, Zakari Affoh, et Valentin Mena Morales, étudiants de Télécoms Bretagne, pour leur travail en projet portant sur l'architecture du décodeur présenté.

Références

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo Codes", IEEE International, Conference on Communication ICC93, vol. 2/3, May 1993, pp. 1064-1071.

[2] R. G. Gallager, "Low Density Parity Check Codes", IRE Trans. Inform. Theory, Jan. 1962, pp. 21-28.

[3] A.J. Viterbi, "Errors bounds for convutional codes and an asymtotically optimun decoding algorithm", IEEE Trans. Inform. Theory, Vol. IT-13, N° 2, pp 260-269, April 1967.

[4] 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.

[5] P. Adde, C. Jégo and R. Le Bidan,**"** Near maximum likelihood soft-decision decoding of a particular class of rate-1/2 systematic linear block codes**"**, Electronics Letters, February 2011, vol. 47, n° 4, pp. 259-260

[6] P. Adde, D. Gomez Toro and C. Jégo, "Maximun likelihood soft decoding of systematic block codes: from algorithm to architecture," *IEEE Trans. Signal Proc.*, July 2012, vol. 60, n°7, pp. 3914-3919.

[7] Robert J McEliece, "The theory of information and coding", Cambridge University Press, 2002.