



HAL
open science

Parallélisme et turbocodes convolutifs

Olivier Muller, Amer Baghdadi, Michel Jezequel

► **To cite this version:**

Olivier Muller, Amer Baghdadi, Michel Jezequel. Parallélisme et turbocodes convolutifs. MajecSTIC 2006 : MANifestation des JEunes Chercheurs STIC, Nov 2006, Lorient, France. hal-02280103

HAL Id: hal-02280103

<https://hal.science/hal-02280103>

Submitted on 6 Sep 2019

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.

Parallélisme et turbo codes convolutifs

Olivier Muller, Amer Baghdadi, Michel Jézéquel

Département Electronique, GET/ENST Bretagne,

Technopôle Brest Iroise, 29238 Brest, France

prenom.nom@enst-bretagne.fr

Résumé : La complexité algorithmique, dégagée par les décodeurs itératifs comme les turbo codes, impose une gestion optimale du parallélisme pour atteindre de hauts débits. Ce papier explore le parallélisme dans le turbo décodage de codes convolutifs avec l'algorithme BCJR et propose une classification à plusieurs niveaux des techniques de parallélisme. Il montre également pour le parallélisme de sous-bloc que d'une part l'initialisation par passage de message surpasse l'initialisation par acquisition, et que d'autre part ce parallélisme est inefficace à fort degré de parallélisme. A l'inverse, le parallélisme de décodeur composant s'y révèle très efficace.

Mots Clés : Parallélisme, turbo codes, architecture matérielle, microélectronique.

1 INTRODUCTION

Les systèmes de communication numériques, telles que les communications sans fils, sur fibre optique ou pour du stockage numérique, nécessitent de très haut débit et également d'excellentes capacités de correction d'erreurs. Pour approcher la limite de Shannon, on a recours aux algorithmes de décodage itératif tels que les turbo codes [1] ou les LDPC. Cependant de part leur complexité, ces algorithmes imposent une exploitation optimale du parallélisme pour atteindre de très hauts débits.

Le parallélisme des turbo codes a déjà été étudié à plusieurs niveaux de granularité : parallélisme de grain fin, sur les calculs élémentaires de l'algorithme de décodage [3][4]; parallélisme de grain moyen avec des schémas de décodage de trame [5][6] ou plus récemment avec un parallélisme de décodeur : le décodage combiné (*shuffled decoding*) [7].

Ce papier présente une classification du parallélisme existant pour les turbo codes convolutif et une analyse de performance du parallélisme de grain moyen.

La section suivante présente brièvement le turbo décodage des codes convolutifs. La section 3 propose la classification à plusieurs niveaux du parallélisme de turbo codage. Les sections 4 et 5 analysent respectivement le parallélisme de sous-bloc et le parallélisme de décodeur composant sur la base d'un critère d'efficacité de parallélisme que nous introduisons.

2 TURBO CODES CONVOLUTIFS

Découvert en 1993 [1], le principe turbo repose sur un échange itératif d'information extrinsèque entre des blocs élémentaires. Pour les turbo codes convolutifs (figure 1.a),

les blocs élémentaires sont des décodeurs composants utilisant l'algorithme BCJR [2] qui permet un décodage optimal des codes convolutifs selon le critère Maximum a Posteriori (MAP). Les décodeurs, appelé SISO, sont à entrée et sortie pondérée pour permettre l'échange d'information extrinsèque. L'information sortant d'un SISO est ensuite entrelacée (Π) ou desentrelacée (Π^{-1}) pour servir d'information entrante à un autre décodeur.

La figure 1.b illustre les principales étapes de l'algorithme BCJR. Premièrement, on calcule les métriques de branches du treillis (γ) qui représentent la probabilité d'une transition entre deux états. Ensuite la récursion aller (α) calcule récursivement les probabilités d'une section de treillis (i.e. pour tous les états) à partir des observations passées. De manière similaire, la récursion retour (β) calcule les probabilités d'une section de treillis à partir des observations futures. Finalement, l'information extrinsèque sur un symbole d_k est obtenue à partir de la récursion aller de la section $k-1$, de la partie extrinsèque de la métrique de branche et de la récursion retour de la section k .

3 LES NIVEAUX DE PARALLELISME

Le parallélisme peut intervenir à trois niveaux lors d'un turbo décodage avec l'algorithme BCJR : au niveau des métriques BCJR, au niveau des décodeurs BCJR SISO, ou/et au niveau du turbo décodeur.

3.1 Parallélisme des métriques BCJR

Le niveau de parallélisme des métriques BCJR concerne les calculs de toutes les métriques impliquées dans le décodage. Il exploite à la fois le parallélisme inhérent à la structure du treillis, mais aussi le parallélisme des calculs du BCJR.

3.1.1 Parallélisme de transition du treillis

Dans un treillis, les mêmes opérations sont répétées par paires pour toutes les transitions. Ainsi on peut extraire de cette répétition le parallélisme des transitions de treillis. Dans le domaine logarithmique, ces opérations sont soit des opérateurs ACS (Addition Comparaison Sélection) pour l'algorithme max-log-MAP, soit des opérateurs ACSO (ACS avec un offset) pour l'algorithme log-MAP [4].

3.1.2 Parallélisme de transition du treillis

Dans un treillis, les mêmes opérations sont répétées par paires pour toutes les transitions. Ainsi on peut extraire de cette répétition le parallélisme des transitions de treillis.

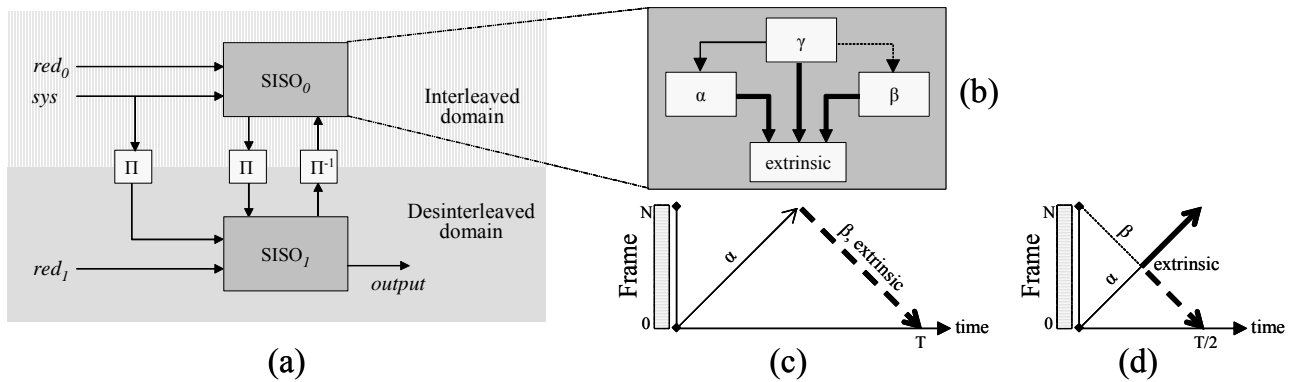


Figure 1. Turbo décodage: (a) turbo décodeur, (b) BCJR SISO, (c) schéma aller-retour, (d) schéma papillon

Dans le domaine logarithmique, ces opérations sont soit des opérateurs ACS (Addition Comparaison Sélection) pour l'algorithme max-log-MAP, soit des opérateurs ACSO (ACS avec un offset) pour l'algorithme log-MAP[4].

Les trois calculs de l'algorithme BCJR nécessitent chacun un nombre d'opérateurs ACS égal à la moitié du nombre de transition dans une section. Ainsi ce nombre borne naturellement le degré de parallélisme de transition du treillis. Néanmoins ce parallélisme est peu coûteux en ressources matérielles, car il ne duplique que les opérateurs ACS. Les mémoires, très coûteuses en surface, sont épargnées puisque toutes les opérations sont exécutées sur une même section de treillis.

3.1.3 Parallélisme des calculs du BCJR

Orthogonalement, on peut extraire du parallélisme en réalisant en parallèle les trois calculs du BCJR. Par exemple, une exécution parallèle du calcul de la récursion retour et de l'information extrinsèque est utilisée dans le schéma original de décodage appelé aller-retour (figure 1.c). On peut donc y observer un degré de parallélisme des calculs BCJR de un dans la partie aller et de deux dans la partie retour.

La figure 1.d représente le schéma papillon qui permet de doubler le degré de parallélisme du schéma original en parallélisant les récursions aller et retour.

Comme le parallélisme de transition de treillis, ce parallélisme duplique uniquement les unités de calculs BCJR. Donc, il est très efficace en surface même s'il est limité par un degré trois de parallélisme.

Pour conclure, le niveau de parallélisme des métriques BCJR est très économique en surface mais il souffre d'une limitation inhérente à l'algorithme BCJR. Ainsi un concepteur désireux d'augmenter le parallélisme devra chercher à un niveau de granularité supérieur.

3.2 Parallélisme de décodeur BCJR-SISO

Le parallélisme de décodeur BCJR-SISO consiste à utiliser plusieurs décodeurs SISO exécutant chacun l'algorithme BCJR sur un sous-bloc de la trame décodée dans un des ordres d'entrelacement. Le parallélisme peut donc être appliqué soit sur les sous-blocs soit sur les décodeurs composants.

3.2.1 Parallélisme de sous-blocs

Avec le parallélisme de sous-bloc, chaque trame est divisée en M sous-blocs, qui sont ensuite traités simultanément chacun par un décodeur SISO initialisé de manière adéquate. En plus de la duplication des décodeurs SISO et des mémoires attachées, deux autres contraintes s'imposent. D'une part, il faut aussi paralléliser l'entrelacement pour offrir une bande passante permettant l'échange des informations. D'autre part, les décodeurs SISO doivent être initialisés correctement soit par acquisition [5] soit par passage de message [6].

3.2.2 Parallélisme de décodeur composant

Le parallélisme de décodage composant n'a vraiment pris de sens qu'avec l'apparition du décodage combiné [7] Il s'agit d'exécuter simultanément tous les décodeurs composants et d'échanger les informations extrinsèques au plus vite. Cette méthode permet donc de diviser par deux la durée d'une itération.

Le parallélisme de décodeur BCJR-SISO permet d'atteindre un degré de parallélisme raisonnable (bien que limité par la taille de la trame) en ne dupliquant qu'une partie des mémoires.

3.3 Parallélisme de turbo décodeur

On peut également dupliquer l'ensemble du turbo décodeur afin de traiter en parallèle les itérations et/ou les trames. Le parallélisme d'itérations s'opère de manière pipeliné sur une profondeur maximum égale au nombre d'itération. En revanche, le parallélisme de trame ne présente aucune limitation de parallélisme.

Néanmoins, le parallélisme de décodage n'est pas envisageable de part son coût en surface (toutes les ressources sont dupliquées), mais aussi par l'absence de gain dans la latence de décodage.

4 INITIALISATION ET PARALLELISME DE SOUS-BLOCS

Le parallélisme de sous-bloc nécessite l'initialisation des métriques de récursion, car seules les extrémités de trame disposent d'information sur les métriques de récursions et non les extrémités de sous-blocs. Une estimation des informations indéterminées peut être obtenue soit par une acquisition, soit par un passage de message entre des sous-blocs voisins.

4.1 Initialisation par acquisition

La technique d'initialisation par acquisition permet d'estimer les métriques de récursion grâce à une fenêtre d'acquisition (ou prologue) chevauchant le sous-bloc voisin. La longueur de la fenêtre d'acquisition (AL) permet à partir d'une initialisation équiprobable des états de fiabiliser la métrique de récursion manquante. AL est déterminé à la conception pour minimiser les dégradations. Empiriquement elle est fixée entre 3 et 5 fois la longueur de contrainte du code [5].

Quand tous les sous-blocs sont initialisés avec cette méthode, le temps de décodage (t_d), le gain en vitesse (S_g) et le surplus de calcul (R_c) peuvent être exprimé ainsi :

$$t_d \propto \left(\frac{N}{d} + AL\right) \cdot it \quad (1)$$

$$S_g = d / \left(1 + \frac{AL \cdot d}{N}\right) \quad (2)$$

$$R_c = \frac{AL \cdot (d-1)}{N} \quad (3)$$

avec N la longueur de trame, d le degré de parallélisme de sous-bloc et it le nombre d'itérations.

L'équation (1) montre que le temps de décodage tend vers une constante quand N croit. Donc le parallélisme de sous-blocs avec l'initialisation par acquisition est plafonné en débit avec un gain maximum de $N/(AL+1)$. En définissant l'efficacité comme le rapport 'gain en vitesse sur degré de parallélisme', l'efficacité correspondante tendrait vers $1/(AL+1)$. En revanche, le surplus de calcul, qui affecte le calcul des récursions et les accès aux mémoires de données entrantes, s'accroît de manière linéaire avec le degré de parallélisme.

4.2 Initialisation par passage de message

Une seconde méthode consiste à initialiser un décodeur SISO dynamiquement avec les métriques de récursions calculées par ces voisins durant la dernière itération [6]. De fait, cette technique ne nécessite presque aucun ajout matériel. Pour évaluer cette technique, la dégradation des taux d'erreur a été étudiée et compensée par l'ajout d'itérations.

La figure 2 représente pour cette méthode la convergence du taux d'erreur trame (FER) au fil des itérations pour différents degrés de parallélisme. La limite asymptotique inchangée par le parallélisme assure que l'utilisation d'itérations supplémentaires peut corriger une dégradation éventuelle.

Le temps de décodage et le gain en vitesse s'exprime en fonction du nombre d'itérations nécessaire (it_{MP}) avec cette technique :

$$t_d \propto \frac{N}{d} \cdot it_{MP} \quad (4)$$

$$S_g = d \cdot \frac{it}{it_{MP}} \quad (5)$$

Une estimation de it_{MP} peut être obtenue par simulation à taux d'erreur trame fixé. La figure 3 montre l'évolution de cette estimation, du gain en vitesse et de l'efficacité avec le degré de parallélisme. On remarque que le nombre d'itérations est constant pour un faible degré de parallélisme puis croît linéairement passé un seuil (8 dans ce cas).

On peut donc écrire:

$$it_{MP} = it \quad \text{si } d \text{ inférieur au seuil.}$$

$$it_{MP} = it + C \cdot d \quad \text{si } d \text{ supérieur au seuil, avec } C \text{ constant}$$

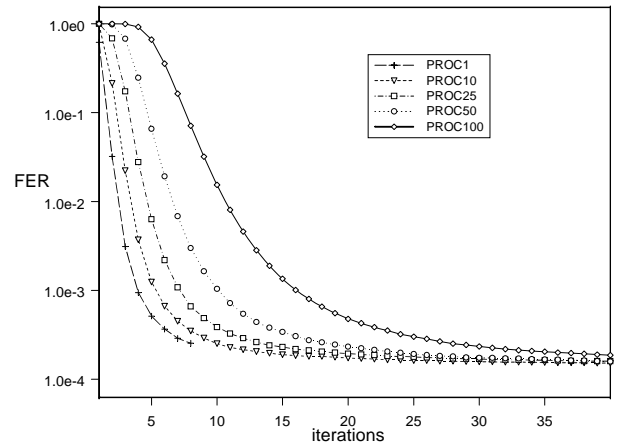


Figure 2. Convergence de la technique par passage de message, DVB-RCS, R=6/7, trame de 188 octets, SNR=4.2 dB, 5 bits de quantification, algorithme Log-MAP

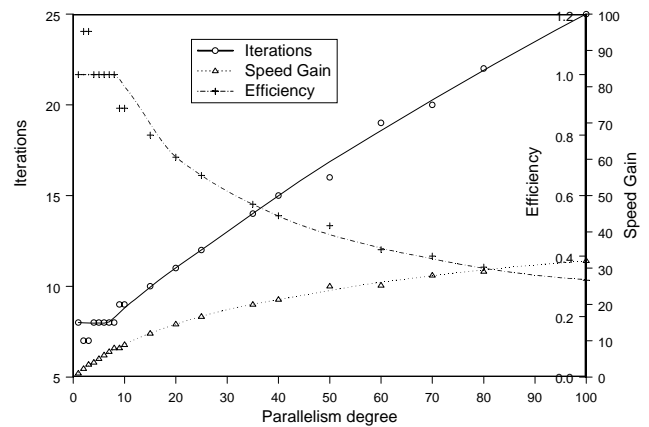


Figure 3. Itérations, gain en vitesse et efficacité avec la technique par passage de message, DVB-RCS, R=6/7, trame de 188 octets, SNR=4.2 dB, 5 bits de quantification, algorithme Log-MAP

Le gain de vitesse plafonne à la valeur it/C correspondant à une efficacité de $it/(C \cdot N)$.

Le seuil dépend fortement de la taille des sous-blocs. Il peut être interprété comme la taille de bloc minimale fournissant des métriques de récursions fiables à la fin de la première itération. Sous cette taille, les valeurs des récursions doivent être compensées par de nouvelles itérations. Le seuil dépend également du rendement du code, puisque ce dernier influence la fiabilité des récursions.

4.3 Efficacité et comparaison

La figure 4 compare l'efficacité du parallélisme de sous-blocs entre les deux méthodes d'initialisation. Dans les conditions présentées, les techniques par acquisition de 16 symboles et par passage de message sont aussi efficaces à fort degré de parallélisme. Mais, à faible degré, seule l'efficacité du passage par message est optimale. De plus, l'initialisation par acquisition dégrade les performances.

La comparaison tend donc clairement en faveur de l'initialisation par passage de message, qui se révèle au moins aussi efficace mais non dégradante.

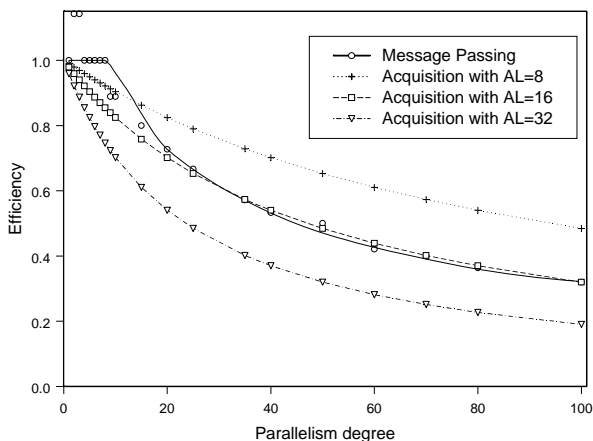


Figure 4. Efficacité et méthodes d'initialisation DVB-RCS, R=6/7, trame de 188 octets, SNR=4.2 dB, 5 bits de quantification, algorithme Log-MAP

Par contre, on s'aperçoit que le parallélisme de sous-bloc est très peu efficace à fort degré de parallélisme.

5 DECODAGE COMBINÉ

5.1 Efficacité du décodage combiné

L'efficacité du parallélisme de décodeur composant est définie comme le rapport 'gain en vitesse sur degré de parallélisme' à des taux d'erreurs identiques. Ainsi l'efficacité du décodage combiné peut être simplifiée par le rapport entre le nombre d'itération avec un décodage simple (pour atteindre un taux d'erreur fixé) et avec un décodage combiné (pour atteindre ce même taux d'erreur). Par simulation, on a observé une efficacité variant de 0,6 à 0,95. Les variations les plus significatives ont été obtenues en changeant le degré de parallélisme de sous-bloc et/ou l'entrelacement.

5.2 Décodage combiné et parallélisme de sous-blocs

En cas d'utilisation conjointe des parallélismes de sous-blocs et de décodeur composant, on s'aperçoit (voir table 1 et 2) que l'efficacité du décodage combiné augmente avec le degré de parallélisme de sous-blocs. Comme l'efficacité de parallélisme de sous-bloc diminue avec le degré de parallélisme de sous-bloc, le parallélisme de décodeur composant devient plus intéressant passé un seuil. Ce seuil est franchi lorsque l'efficacité d'un décodage simple avec un degré $2d$ de parallélisme de sous-bloc devient inférieur à l'efficacité d'un décodage combiné ayant un degré d de parallélisme de sous-bloc.

Table 1. Efficacité et degré de parallélisme de sous-bloc avec l'entrelacement DVB-RCS de 53 octets (R=6/7; max log MAP; SNR=4,0 dB; FER =1,6e-03)

Degré de parallélisme de sous-bloc	# itérations sans décodage combiné	# itérations avec décodage combiné	Efficacité
1	8	12	0.66
4	11	15	0.73
8	16	20	0.8
53	47	51	0.92

Dans les exemples du tableau 1 et 2, le décodage combiné devient plus intéressant pour d valant 4 et 16.

Table 2. Efficacité et degré de parallélisme de sous-bloc avec l'entrelacement DVB-RCS de 188 octets (R=6/7; max log MAP; SNR=4,0 dB; FER =1,6e-03)

Degré de parallélisme de sous-bloc	# itérations sans décodage combiné	# itérations avec décodage combiné	Efficacité
1	8	11	0.72
2	9	11	0.82
4	9	12	0.75
16	13	15	0.86
64	19	23	0.83
128	34	37	0.92

6 CONCLUSION

Dans ce papier, nous analysons et classifions les diverses techniques de parallélisme existantes pour le turbo décodage avec l'algorithme BCJR. La classification propose trois niveaux de parallélisme : le niveau des métriques BCJR, le niveau des décodeurs BCJR SISO et le niveau des turbo décodeurs. Pour le parallélisme de sous-bloc, nous avons montré d'une part que l'initialisation des sous-blocs est plus efficace par passage de message que par acquisition, et d'autre part que ce parallélisme est inefficace à un fort degré de parallélisme de sous-bloc. Au contraire, le décodage combiné devient de plus en plus efficace avec ce degré de parallélisme. C'est pourquoi un critère est proposé pour choisir au mieux le parallélisme parmi ces deux techniques.

BIBLIOGRAPHIE

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," in Proc. 1993 International Conference on Communications (ICC'93), Geneva, Switzerland, 1993.
- [2] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inf. Theory, vol. IT-20, pp. 284-287, Mar. 1974.
- [3] G. Masera, G. Piccinini, M. R. Roch, and M. Zamboni, "VLSI architectures for turbo codes," IEEE Trans. VLSI Syst., vol. 7, pp. 369-379, Sept. 1999.
- [4] E. Boutillon, W. J. Gross, and P. G. Gulak, "VLSI architectures for the MAP algorithm," IEEE Trans. Commun., vol. 51, pp. 175-185, Feb. 2003.
- [5] T. Wolf, "Initialization of Sliding Windows in Turbo decoders", 3rd International Symposium on Turbo Codes and Related Topics, Brest, France, pp. 219-222, Sept. 2003.
- [6] A. Abbasfar, and K. Yao, "An Efficient Architecture for High Speed Turbo Decoders," Proc. of ICASSP 2003, pp. IV-521-IV-524, April 2003.
- [7] J. Zhang, and M.P.C. Fossorier, "Shuffled iterative decoding", IEEE Transactions on Communications Volume 53, Issue 2, Feb. 2005 Page(s):209 - 213.