

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

DÉPARTEMENT D'ÉLECTRONIQUE

Mémoire de master en électronique

---

# Le rôle des réseaux de Benes dans l'implémentation d'un décodeur LDPC flexible

---

**Oualid MOUHOUBI**

Sous la direction de :  
**M. Mohamed TAGHI**

Présenté et soutenu publiquement le 23/06/2016

**Composition du Jury :**

Président	M Zidane TERRA	Dr	ENP
Rapporteur	M. Mohamed TAGHI	Pr	ENP
Examineur	Mme. Aicha MOUSSAOUI	Dr	ENP

**Promotion : Juin 2016**

## *Dédicace*

" Je dédie ce travail à toute ma famille. À mes chers parents pour tous leurs sacrifices consentis pour mon éducation et ma formation, sans eux je ne serais jamais arrivé à ce niveau. À mon frère et ma sœur pour leurs précieux soutiens et encouragements. À tous mes amis et camarades, à toutes les personnes qui m'ont enseigné tout au long de mon parcours. À tous ceux que je n'ai pas cités et que je n'oublierai jamais leurs soutiens et leurs aides. "

# *Remerciement*

Je remercie M. M.TAGHI pour son soutien continu, sa disponibilité, son aide et ses précieux conseils.

je tien à remercier chaleureusement monsieur M. Z.TERRA pour m'avoir fait l'honneur de présider le jury de ce mémoire, ainsi que Mme. A.MOUSSAOUI pour voir accepté d'examiner mon travail.

je remercie toute personne qui, d'une manière ou d'une autre a contribué à l'élaboration de ce travail. Enfin, je tien à remercier tous mes amis et camarades ainsi que tous les enseignants du département d'Électronique de l'ENP.

## ملخص

تمثل رموز اختيار التكافؤ منخفض الكثافة (LDPC) رموز مصححات الخطأ الأكثر فعالية لأنها تسمح بالحصول على نتائج قريبة من تلك المعرفة ب نهاية شانون (SHANON) للرموز ذات المجموعات الواسعة جدا فيما يخص جودة تصحيح الخطأ. هذه الدراسة خصصناها لتصميم تخطيط نصف متوازي مرن لجهاز فك الرموز بالاعتماد على نظام الحلول الحسابية (خوارزمية) لفك الترميز Min-Sum. تم تأكيد كفاءة هذه الخوارزمية عن طريق المحاكاة على جهاز الكمبيوتر. تخطيط جهاز فك الرموز تم دمجها في بطاقة FPGA بعد تقليص حجم رموز ال (LDPC) الخاص بهذا العمل بسبب متطلبات الاختبار. في الأخير تم ربط نتائج التركيب مع الأنماط البيانية المولدة من لغة وصف الأجهزة HDL الخاصة بهذا التخطيط.

**كلمات مفتاحية :** اختيار التكافؤ منخفض الكثافة (LDPC) , جهاز فك الرموز HDL , بطاقة FPGA , التخطيط , لفك الترميز , Min-Sum.

## Abstract

Low-density parity-check (LDPC) are among the most powerful forward error correcting codes since they achieve error correction performance very close to the Shannon limit for large block lengths. LDPC block. In this thesis, we investigate into the design architecture of an array type LDPC code based on min-sum algorithm. The performance of the decoding algorithm was first validated via simulations. The detailed design of the decoding architecture was implemented on a field-programmable gate array (FPGA) kit with a short block length as an example. The schematics generated have been documented along with the synthesis results

**Key words :** LDPC, FPGA, HDL, Min-Sum, Decoder, Architecture, Implementation.

## Résumé

Low-density parity-check (LDPC) codes font partie des codes correcteurs d'erreur les plus performant, puisque ils permettent d'atteindre une performance de correction d'erreur très proche de la limite de Shanon pour des codes en block très larges. Nous avons consacré notre travail à la conception d'une architecture semi parallèle, flexible d'un décodeur LDPC basée sur l' de décodage Min-sum. Les performances de cet algorithme de décodage ont été validé dans un premier temps par le biais d'une simulation. La conception de l'architecture du décodeur a été ensuite implémentée sur la carte FPGA après réduction de la taille du code LDPC considéré pour ce travail, à cause des exigences de test. Les schémas générés par la description HDL de cette architecture ont été associé aux résultats de synthèse.

**Mots Clés :** LDPC, FPGA, HDL, Min-Sum, Décodeur, Architecture, Implémentation.

# Table des matières

Table des figures

- Introduction générale** **7**
  
- 1 Ordonnancement** **8**
  
- 2 Implémentation flexible d'un décodeur LDPC** **11**
  - 2.1 Généralités . . . . . 11
  - 2.2 Message Permutation Block . . . . . 13
  - 2.3 Configuration des switches du réseau de Benes . . . . . 14
    - 2.3.1 Vue d'ensemble . . . . . 14
    - 2.3.2 Exemple . . . . . 15
  
- Conclusion générale** **15**
  
- Bibliography** **19**

# Table des figures

- 1.1 Comparaison des performances entre les différents algorithmes de décodage pour N=1024. . . . . 9
- 1.2 Les première étape de décodage dans le cas de l'ordonnancement horizontale lors 'une itération  $i$  . . . . . 10
  
- 2.1 Diagramme du décodeur . . . . . 12
- 2.2 Machine d'état du décodeur . . . . . 13
- 2.3 Bloc de permutation des messages . . . . . 14
- 2.4 Bloc de permutation des messages . . . . . 14
- 2.5 Bloc de permutation des messages . . . . . 15
- 2.6 Réseau de Benes à N entrée . . . . . 16
- 2.7 Matrice de permutation . . . . . 16
- 2.8 Bloc de permutation  $P_A$  et  $P_B$  . . . . . 16

# Introduction

Le monde vit ces 2 dernières décennies l'ère du tout numérique où la technologie des communications sans fils se taille la part du lion. Cet extraordinaire essor que connaît cette technologie est principalement dû aux avancées considérables dans domaine de la micro(nano) électronique ainsi qu'à la demande de plus en plus pressante de moyens de communications souples, plus rapides et plus efficaces.

La découverte dans les années 90 des Turbo-codes et, plus généralement du principe itératif appliqué au traitement du signal, a révolutionné la manière d'appréhender un système de communications numériques. Cette avancée notable a permis la redécouverte des codes correcteurs d'erreurs inventés par R. Gallager en 1963, appelés codes Low Density Parity Check (LDPC).

Dans le contexte des systèmes de communication souple, ce projet a pour objectif de présenter une méthode pour l'implémentation d'un décodeur LDPC flexible sur la carte FPGA (Field Programmable Gate Array), qui sera capable de supporter plusieurs types de codes, et d'offrir une possibilité à une variété de code d'être testée sans devoir implémenter une nouvelle architecture au décodeur.

# Chapitre 1

## Ordonnancement

L'ordonnancement d'un algorithme de décodage est l'ordre dans lequel les messages devraient se propager dans le graphe de Tanner. deux type d'ordonnancement sont distingués dans cette section [1].

- Ordonnancement bi-directionnel : c'est un ordonnancement orienté en série, où seulement les messages pertinent sont traités et envoyés aux noeuds avec lesquels ils sont reliés.
- Ordonnancement [2]à flux : est un ordonnancement orienté en parallèle, où tout les noeuds sont traités en même temps, dans ce cas ce cas les messages arrivés se comporte comme des déclencheur pour les processeur des noeuds.

Pour une exploitation pratique des codes LDPC, l'ordonnancement à flux est souvent utilisé, le comportement de chaque processeur des noeuds devient plus simple. Mais la question qui se pose déjà est dans quel ordre les processeurs des noeuds calculent les messages de sortie? ce choix peut engendrer des cycles inutiles dans le graphe de Tanner et provoquer une divergence au niveau du décodage, cependant trois familles d'ordonnancement existent dans la littérature que nous allons décrire :

### Ordonnancement à écoulement

Ce type désigne l'au-delà de la façon classique d'ordonnancement des algorithmes BP. le sens d'écoulement, peut avoir été changé depuis son original sens [3]. dans ce type d'ordonnancement, les noeuds du même genre sont mise à jours dans un premier temps ensuite l'autre types de noeuds. (Voir figure 1.1 a). La mise à jour d'un type de noeud peut être faite noeud après noeud (en série) ou tous à la fois : cela n'affectera pas les valeurs de sorties

### Ordonnancement probabiliste

L'auteur de [4] décrit l'ordonnancement probabiliste. La première idée est de se débarrasser de l'auto confirmation des messages causée par les cycles du graphe de Tanner : parfois cela est évité par la non activation de quelques noeuds, qui, sont censé l'être dans le cas de l'ordonnancement à écoulement. Voir figure 1.1 b).



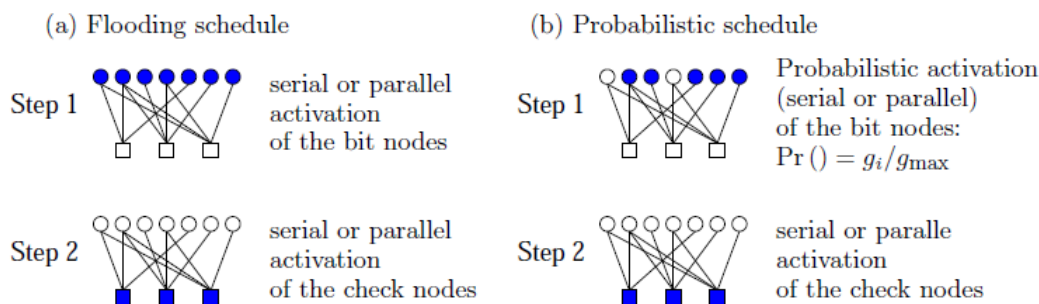


FIGURE 1.1 – Comparaison des performances entre les différents algorithmes de décodage pour  $N=1024$ .

## Ordonnancement verticale

L’auteur de [5] a proposé " shuffle " algorithme BP qui converge plus rapidement qu’un algorithme BP classique. L’idée est de mettre à jour les informations dès que les messages soient calculés, pour permettre au prochain processeur de noeud d’utiliser plus d’informations avant qu’il ne se mette à jour. Cet ordonnancement opère le longs des noeuds de données, ce qui signifie que tout les noeuds de données sont traités l’un après l’autre.

Figure 1.2 montre un exemple des premières étapes prises pour une itération  $i$ . l’ordonnancement d’une itération  $i$  est en série : les noeuds de contrôle impliqués dans le premier noeud de données sont traités (étape 1) ce noeud de données est ensuite mis à jour (étape 2). Ensuite les noeuds de contrôle impliquant le deuxièmes noeud de donnée sont traités (étape 3), ce dernier est mis à jour (étape 4), ainsi de suite, jusqu’à mettre à jour tout les noeuds de données.[6]

## Ordonnancement horizontale

L’auteur de [6] a proposer une architecture en série s’appuyant sur un ordonnancement qui consiste à traiter les processeurs des équations de contrôle de parité en série. l’information envoyée au noeuds de contrôle désigné par le traitement, disons  $CN_m$  prend en compte l’information de l’itération précédente et celle de l’itération actuelle qui a été mise à jour par tout les autre noeuds de contrôle précédent  $CN_{m0}$ ,  $m0 < m$ . cet ordonnancement est souvent associé par l’algorithme APP, car les résultats obtenus avec l’algorithme BP sont plus ou moins médiocre.

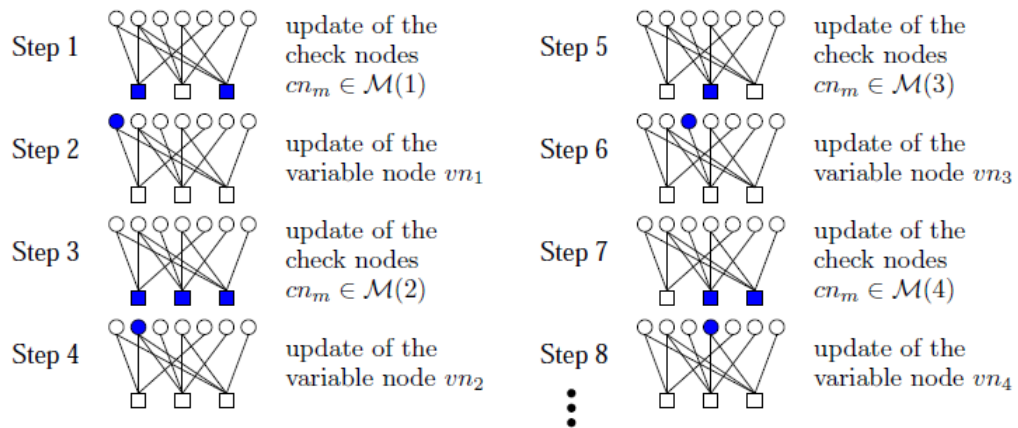


FIGURE 1.2 – Les première étape de décodage dans le cas de l’ordonnancement horizontale lors ’une itération  $i$

# Chapitre 2

## Implémentation flexible d'un décodeur LDPC

Après avoir introduit une variété des types de de conception des décodeurs LDPC dans le chapitre précédent, nous avons opté pour l'implémentation d'un décodeur LDPC semi-parallèle pour pouvoir bénéficier à la fois de l'avantage de sérialisation de traitement de donnée, cependant offrir une flexibilité pour notre architecture, et en même temps de l'avantage de rapidité offert par le parallélisme. Un compromis entre la vitesse de traitement et la complexité peut être appliquée en faisant varier le nombre de noeuds fonctionnel parallèle dans l'architecture.

Le but principale de la flexibilité, est que le décodeur soit capable de supporter une grande variété de codes LDPC sans avoir à changer son architecture.

### 2.1 Généralités

L'architecture comporte  $P$  processeurs, un bloc de permutation et un bloc de contrôle logique, comme le montre la figure 2.1. Il existe un nombre très petit de processeur des noeuds de contrôle que de processeur des noeuds de données. Les processeurs ont pour fonction de recevoir et stocker les messages entrant, effectuer les opérations respectives et calculer les sorties, où l'affectation des noeuds aux processeur est dirigée par le bloc de contrôle. Le processus de décodage est illustré dans la figure 2.5. les demi-itérations (bit to check) et (check to bit) sont répétées un certain nombre de fois avant de produire le mot de code. le nombre d'itération est choisi très petit (alentour de 10) pour garder le temps de décodage petit. Aucune vérification pour un mot de code valide est effectuée, le temps d'exécuter une itération est aussi long que la demi-itération l'est. Doncn en exécutant une seule chaque itération permet d'augmenter le temps d'exécution d'une itération par le tier  $1/3$ . Avec un nombre d'itération réduit, le profit de termier le décodage à temps est relatif au nombre de cycle éféctués.[5]

- Initialisation :

Durant l'initialisation, les sorties du canal sont chargés dans les processeurs des noeuds de données

- Bit to check

durant cette demi-itération, le processeur des noeuds de données exécute la fonction sa fonction principale qui est donnée par l'équation 1.2. Dans la première itération

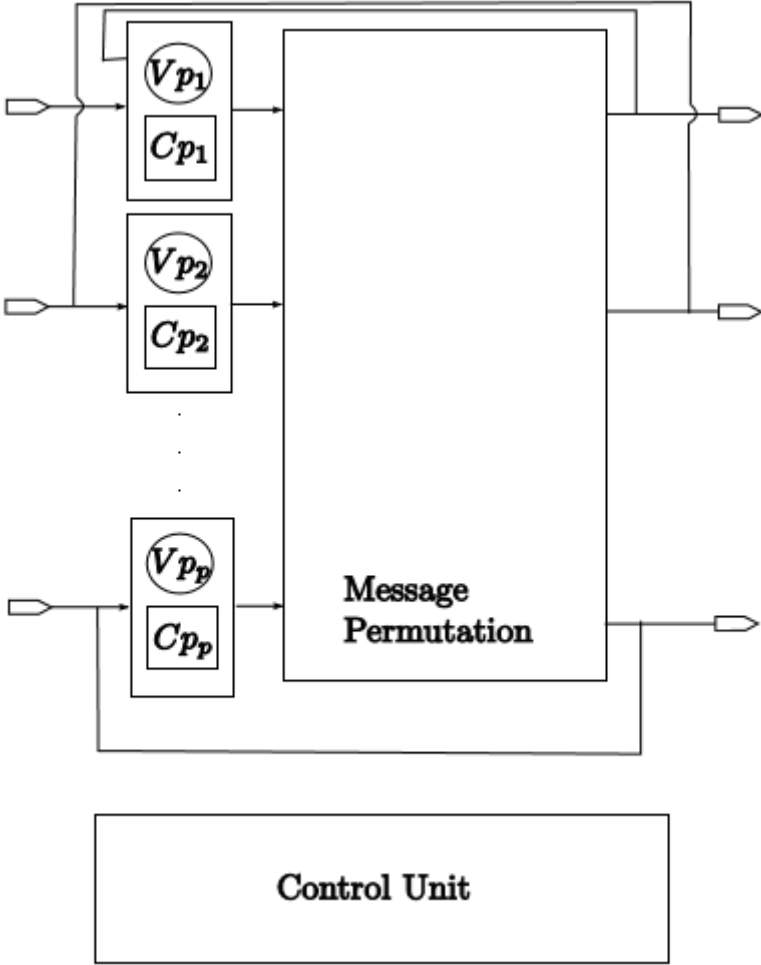


FIGURE 2.1 – Diagramme du décodeur

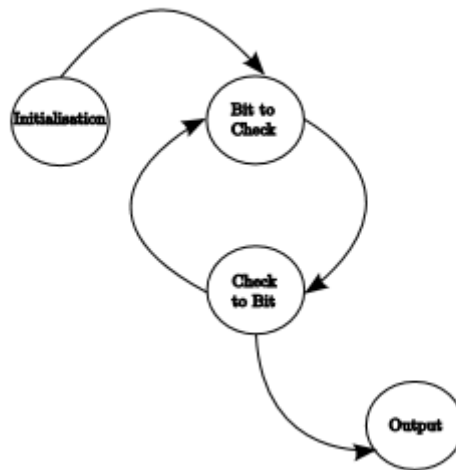


FIGURE 2.2 – Machine d'état du décodeur

il n'y a pas de messages qui arrive à du bloc opposé, le processeur envoie donc les messages qui a reçu directement du canal de transmission. Les messages se dirigent des processeurs des noeuds de données en traversant vers les processeurs des noeuds de contrôle respectifs.

- Check to bit.[4]

durant cette itération, les processeurs des noeuds de controle exécutent la fonction décrite dans l'équation 1.8. les messages en sortie passent par le bloc de permutation, qui sert à les acheminer vers les processeurs des noeuds de données respectifs.

- état final

Après un certain nombre d'itérations du cycle " bit to check " à " check to bit ", le décodeur passe à l'état final et produit une sortie des messages, ou du mot de code corrigé. [3]

## 2.2 Message Permutation Block

Le bloc de permutation de messages est l'origine de la flexibilité du décodeur. Son objectif est de connecter les processeurs des noeuds de données aux processeurs des noeuds de contrôle (et vice versa), pour que le processeur des noeuds de contrôle reçoit les messages en entrée dans l'ordre du graphe de Tanner.

Le bloc de permutation des messages comporte deux switches croisés avec un banc d'entrelaceur au milieu comme le montre la figure ???. Les messages qui arrivent sont permutés d'abord en espace par le réseau de Benes qui se trouve à l'extrême gauche, et stockés ensuite dans les band d'entrelacement. Ce dernier permute les messages sortants dans le temps avant qu'ils ne soient permutés encore une fois en espace par le réseau de Benes qui se situe à l'extrême droite. Ces permutations spatiales et temporelles offre une flexibilité extrême au décodeur, en permettant aux messages entrants d'arriver à destination dans l'ordre adéquat. La configuration des réseaux de Benes et des banc d'entrelacement est rétabli à chaque cycle d'horloge selon l'ordonnancement stocké dans l'unité de contrôle. L'algorithme qui détermine l'ordonnancement est décrit en détail dans

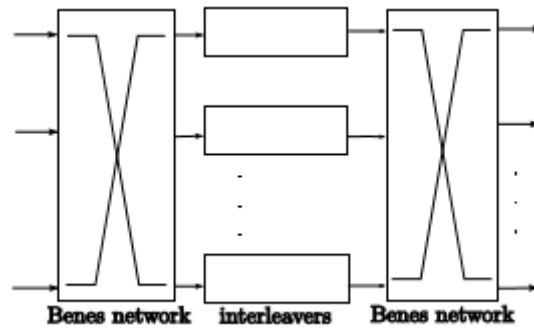


FIGURE 2.3 – Bloc de permutation des messages

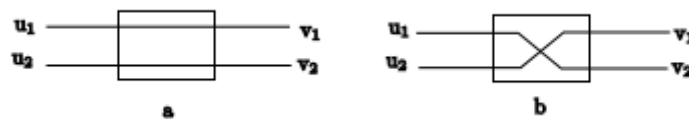


FIGURE 2.4 – Bloc de permutation des messages

## 2.3 Configuration des switches du réseau de Benes

Un réseau de Benes est une commutation multi-étages formalisé par Benes pour les circuits de commutation téléphonique. Elle élimine le besoin des commutateurs "crossbar" à large capacité en les remplaçant par un ensemble de deux élément de commutation avec simplement 2 entrée, 2 sorties comme illustré dans la figure ??  $N$  réseaux d'entrée ( $N = 2^r$ ) sont construits récursivement en utilisant des cellules élémentaires ( $2N/2$ ) comme le montre la figure ??;  $I_i$  sont des switches (commutateurs) élémentaire connectés aux entrées, chaque commutateur relie à la partie supérieur ( $P_A$ ) et à la partie inférieur ( $P_B$ )  $N/2$  réseaux de Benes;  $O_i$  sont des switches élémentaire connectés à la sortie des réseaux de Benes, chaque switch est connectés à la prtie supérieur  $P_A$  et à la partie inférieur  $P_B$  du sous bloc de permutation. ce processus de décomposition est répété jusqu'à ce qu'il reste seulement des cellules élémentaire. Pour un réseau de Benes à  $N$  entrée on aura  $N \log_2(N - 1)$  étages, et  $N/2$  switch par étage.

### 2.3.1 Vue d'ensemble

Le processus qui permet de déterminer la configuration des switch pour une configuration donnée des entrée/sorties se fait en trois étapes et est basé sur les travaux de Waksman

- Trouver la matrice de permutation  
La matrice de permutation est obtenu, en commençant avec une configuration du switch supérieur droit, et détailler quels entrée et sorties devraient être connectées au sous-bloc supérieur et/ou sous bloc inférieur.
- Trouver la configuration des switches  
La matrice de permutation est utilisée pour trouver la configuration des switches élémentaire.

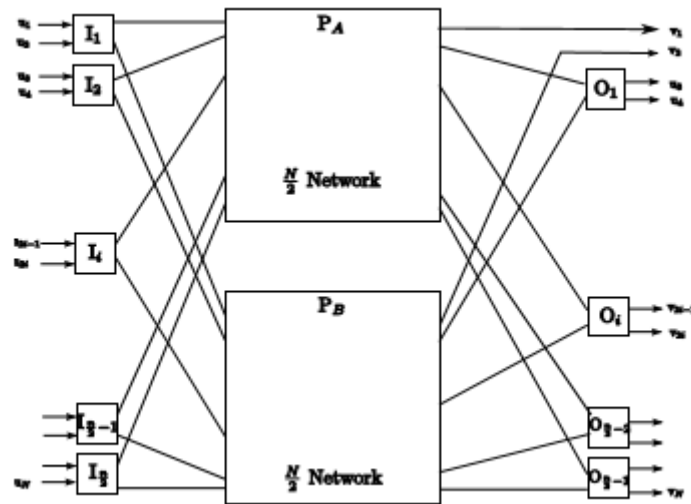


FIGURE 2.5 – Bloc de permutation des messages

- Trouver la cartographie des permutation pour les sous-bloc inférieurs et supérieurs. Connaissant quels entrées et sorties sont connectés aux permutations sous-bloc supérieur et sous bloc inférieurs, le processus peut être décomposé récursivement. Nous commençons le processus encore sur les blocs supérieurs et blocs inférieurs. Le processus est répété jusqu'à atteindre le switch élémentaire du sous bloc.

### 2.3.2 Exemple

La figure 2.6 montre un réseau de Benes pour  $N$  entrée. Le tableau 2.7 montre la matrice de permutation de 8 entrée et 8 sortie, tandis que le tableau 2.8 décrit la permutation des deux blocs supérieur et inférieur  $P_A$  et  $P_B$

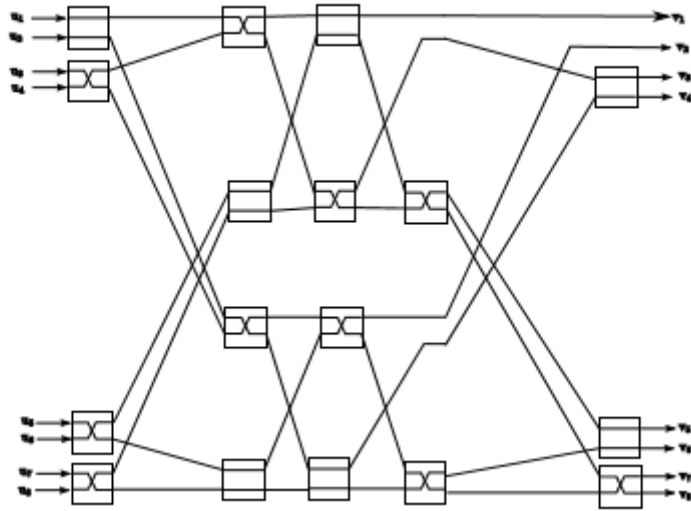


FIGURE 2.6 – Réseau de Benes à N entrée

u\v	1	2	3	4	5	6	7	8
1					A			
2				B				
3							B	
4	A							
5		B						
6								A
7						B		
8			A					

FIGURE 2.7 – Matrice de permutation

u\v	1	2	3	4	u\v	1	2	3	4
1			1		1		1		
2					2				1
3	1			1	3	1			
4		1			4			1	

Table 3.2: Permutation Mappings for  $P_A$  and  $P_B$

FIGURE 2.8 – Bloc de permutation  $P_A$  et  $P_B$



# Conclusion

Grâce à l'utilisation des bloc de permutations , l'architecture du décodeur est devenu très flexible, ce qui permet d'implémenter plusieurs types de codes LDPC sans avoir besoin de concevoir à nouveau une architecture du décodeur. La configuration est généré hors ligne puis chargé dans un bloc de mémoire ROM à l'intérieur l'unité de contrôle sans appliquer de changement à l'architecture du décodeur. Le bloc de mémoire ROM contient la configuration spatiale et temporelle des réseaux de Benes, des bancs d'entrelaceurs et l'ordonnancement des processeurs des noeuds de données et des noeuds de contrôle . cette architecture supporte aussi des codes irréguliers avec certaines contraintes.

# Bibliographie

- [1] CCSDS STANDARD. “Low Density Parity Check for use in the Near-Earth and Deep Space Applications”. In : *Signals, Systems and Computers*. Sous la dir. d’Orange BOOK. 34th European, 2007.
- [2] F. DEMONGEL, N. FAU et N. DRABIK. “A Generic Architecture of CCDS Low Density Parity Check Decoder for Near-Earth Applications”. In : (2009).
- [3] Kschischang F.R. et B.J. FREY. “Iterative Decoding of Compound Codes by Probability Propagation in Graphical Models”. In : *Journal on Selected Areas in Communications* 16 (1998), p. 219–230.
- [4] Y MAO et A. H. BANIHASHEMI. “A heuristic search for good low-density parity-check codes at short block lengths.” In : sous la dir. d’IEEE International Conference on COMMUNICATIONS. 34th European, 2001.
- [5] ZHANG, JUNTAN et M. FOSSORIER. “Shuffled belief propagation decoding.” In : *Signals, Systems and Computers*. Sous la dir. de Conference Record of the THIRTY-SIXTH ASILOMAR CONFERENCE ON. 34th European, 2002.
- [6] E. YEO, B. NIKOLIĆ et V. ANANTHARAM. “High Throughput Low- Density Parity-Check Decoder Architectures.” In : *EEE Globecom, San Antonio*. (Nov. 2001).
- [7] B. VASIC. “Structured iteratively decodable codes based on Steiner systems and their application in magnetic recording”. In : sous la dir. de TX SAN ANTONIO. IEEE Globecom Conf., San Antonio, TX, November 2001, p. 2954–2960.
- [8] R. TANNER. “A recursive approach to low complexity codes”. In : *IEEE Transactions on Information Theory* 27 (sept. 1981).
- [9] B. SAKKOUR. “Etude du décodage des codes Reed-Muller et application à la cryptographie”. Thèse de doct. UMA-ENSTA, 1996.
- [10] N. WIBERG. “Coding and decoding in general graphs”. In : Department of Electrical Engineering, Linköping, Sweden, 2009.
- [11] F. CLERMIDY, C. BERNARD, R. LEMAIRE, J. MARTIN, I. MIRO-PANADES, Y. THONNART, P. VIVET et N. WEHN. “A 477 mW NoC-based digital baseband for MIMO 4G SDR”. In : *In Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. 2010 IEEE International, February 2010, p. 278–279.
- [12] T. LIMBERG, M. WINTER, M. BIMBERG et P. ROBELLY. “A fully programmable 40 gops sdr single chip baseband for lte/wimax terminals”. In : *In Solid-State Circuits Conference, 2008*. Sous la dir. d’ESSCIRC 2008. 34th European, september 2008, p. 466–469.
- [13] IEEE P802.3AN. *10GBASE-T task force*. URL : <http://www.ieee802.org/3/an..>
- [14] G.HN/G.9960. *next generation standard for wired home network*. URL : <http://www.itu.int/ITU-T..>

- [15] T.T.S.I. *digital video broadcasting (DVB) second generation framing structure for broadband satellite applications*. URL : <http://www.dvb.org>..
- [16] IEEE 802.16E. “air interface for fixed and mobile broadband wireless access systems”. In : *ieee p802.16e/d12 draft* (oct. 2005).
- [17] COCCO.M, DIELISSSEN.J, HEIJLLIGERS.M, HEKSTA.A et HUISKEN.J. “A scalable architecture for LDPC decoding”. In : *In : proceeding of 2004 Design, Automation and test in Europe* 42 (mar. 2004), p. 429–455.
- [18] T. MOHSENIN. “Algorithms and Architectures for Efficient Low Density Parity Check (LDPC) Decoder Hardware”. Thèse de doct. UNIVERSITY OF CALIFORNIA, ELECTRICAL et COMPUTER ENGINEERING, 2010.
- [19] BLANKSBY.A, HOWLAN et C.J. “A 690-mW 1-Gb/s, Rate-1/2 low-density parity-check code decoder”. In : *IEEE J.Solid-State Circ.*37(3) 42 (2002), p. 404–412.
- [20] Alles. M. “Implementation Aspects of Advanced Channel Decoding”. Thèse de doct. University of Kaiserslautern, 2010.
- [21] TEMMER ELIAS. “Etude des codes LDPC et application dans un système MIMO”. Thèse de doct. Ecole Nationale Polytechnique d’Alger, 2009.
- [22] S.KAISER et K.FAZEL. *Multi-carrier and spread spectrum systems : From OFDM and MCCDMA to LTE and WiMAX*. deuxième édition. Wiley, G Bretagne, 2008.