Architectures numériques configurables pour le traitement rapide sur FPGA de codes correcteurs d'erreurs de la famille QC-LDPC

Alaa Aldin Al Hariri

UNIVERSITÉ DE LORRAINE

École Doctorale IAEM - Lorraine

Architectures numériques configurables pour le traitement rapide sur FPGA de codes correcteurs d'erreurs de la famille QC-LDPC

THÈSE

Soutenue le 10 décembre 2015
pour obtenir du grade de

Docteur de l'Université de Lorraine
en Systèmes Électroniques

par

Alaa-Aldin AL HARIRI

Composition du jury :

AMER BAGHDADI	Professeur, Télécom Bretagne	Rapporteur
VINCENT FRICK	MCF HDR, Universté de Strasbourg	Rapporteur
ABBAS DANDACHE	Professeur, Université de Lorraine	Président du Jury
FABRICE MONTEIRO	Professeur, Université de Lorraine	Directeur de thèse

Laboratoire de Conception, Optimisation et Modélisation des Systèmes, 7 Rue Marconi, 57070 Metz, France
Tel : +33 (0)3 87 31 56 57 - Fax : +33 (0)3 87 54 73 07

Remerciements

Mes remerciements vont en tout premier lieu à mon Directeur de thèse, le Professeur Fabrice MONTEIRO, pour avoir encadré mes travaux de thèse, ainsi que pour ses conseils et discussions enrichisssantes. Je le remercie aussi pour son soutien et l'aide personnelle qu'il m'a apporté.

Je remercie également Monsieur Abbas DANDACHE, Professeur à l'Université de Lorraine, pour avoir accepté d'être le Président de ce Jury, ainsi qu'à Monsieur Amer BAGHDADI, Professeur à Télécom Bretagne, Brest, et à Monsieur Vincent FRICK, MCF-HDR à l'Université de Strasbourg, pour avoir accepté d'être les rapporteurs de ce mémoire de thèse.

Mes remerciements vont également au Pr. Camel TANOUGAST, au Dr. Camille DIOU et au Dr. Loïc Sieler pour leur aimable soutien pendant ces cinq années.

Merci aussi à tous mes collègues et amis de longue date du laboratoire. Je leur exprime ma profonde sympathie et leur souhaite beaucoup de bien. En particulier, je cite : Lucas, Mohsin, Mikaël, Mohamad, Rami, Osman, Sarah...

Enfin, je tiens à remercier ma famille pour m'avoir fourni un soutien affectif pendant toute la durée de ma thèse : un grand merci à mes parents et àmon frère, à ma sœur et à Ekbal, mon épouse, qui ont tous beaucoup contribué à m'aider pendant cette période un peu difficile. Je souhaite également remercier mon fils Abdulrahman d'avoir été suffisamment sage pour que je puisse me concentrer sur mon travail.

Table des matières

Table des figures

Liste des tableaux

INTRODUCTION GÉNÉRALE

Introduction Générale

La connectivité sans fil est une fonction devenue essentielle dans un nombre sans cesse croissant (plusieurs dizaines de milliards à l'heure actuelle) et diversifié d'appareils et dispositifs, ceci dans quasiment tous les domaines d'activité et usages, professionnels ou privés : ordinateurs (de bureau et portables), tablettes, téléphones cellulaires, appareils photo...

Les applications et les types de protocoles utilisés pour le trafic des données sont de plus en plus nombreux aujourd'hui, et imposent sans cesse de nouvelles contraintes et exigences. Cette situation impose que des innovations soient en permanence réalisées dans les domaines de la transmission, des architectures matérielles/logicielles et des méthodologies de conception, afin d'être en mesure d'accroître encore les débits, l'autonomie et l'intégration tout en maîtrisant, voire réduisant, les coûts.

Le codage de canal (ou codage canal) est l'une des techniques clef permettant le transfert fiable à haut débit de données via des canaux de transmission sans fil non fiables. La très grande diversité de types et paramétrages du codage canal spécifiés dans les normes de communication, actuelles et futures, rend de plus en plus indispensable de développer des architectures de codeur/décodeur flexibles.

De nombreuses techniques puissantes de correction d'erreurs existent à ce jour, chacune plus spécifiquement adaptée à une catégorie d'application, caractérisées notamment par la nature des données (voix, images, video, données bancaires, etc) et du canal de transmission (rapport signal/bruit, bande passante, etc). Les codes correcteurs d'erreurs les plus populaires dans les normes de transmission actuelles sont les codes convolutionnels (*CC*), les turbo-codes et les codes LDPC (*Low Density Parity Codes*). Ces derniers constituent actuellement la famille de techniques de codage la plus performante du point de vue de leur potentiel à se rapprocher de la capacité limite de Shannon. En effet, certains

des codes LDPC développés à ce jour permettent de se rapprocher à tout juste 0.04dB de cette limite.

Les codes LDPC sont utilisés notamment dans les normes de télévision numérique de deuxième génération (*DVB-S2 — Digital Video Broadcasting - Satellite, second generation*) pour transmettre son et image entre les stations de télévision d'une part, et les unités de réception distantes (fixes et/ou mobiles) situés à domicile par exemple. Les codes LDPC sont également utilisés dans de nombreuses normes de transmission sans fil mobiles telles que IEEE 802.11n (WiFi) et IEEE 802.16e (WiMAX).

Objectifs et plan de la thèse

Ce travail de thèse ici présent propose de nouvelles architectures, performantes et paramétrables, pour le codage et le décodage d'une famille de codes LDPC, les codes QC-LDPC. Les architectures proposées pour les codeurs et décodeurs ciblent une réalisation sur FPGA. Elles offrent des vitesses de traitement qui peuvent être très élevées grâce à des structures matérielles fortement parallèles. Les paramètres configurables concernent les caractéristiques du code ciblé et le degré de traitement parallèle souhaité, ce qui permette de choisir, lors de l'implémentation, le degré de traitement parallèle souhaité, qui conditionne la vitesse de codage ou décodage (débit) et le coût matériel (quantité de ressources matérielles requise, telles que les ressources logigues et de mémorisation).

Le manuscrit de thèse est composé de quatre chapitres principaux et d'un court chapitre de conclusion et perspectives.

- Le premier chapitre introduit les techniques avancées de codage de canal et les codes correcteurs d'erreur qui sont largement utilisés pour accroître, dans les systèmes de communication sans fils, la capacité utile du canal de transmission (bruité, donc perturbé), c'est-à-dire, la quantité de données pouvant être transmise sans perte d'information par unité de temps.

- Le deuxième chapitre présente plus en profondeur, les codes correcteurs linéaires en blocs. L'accent est mis sur les codes LDPC et en particulier la sous-famille QC-LDPC. Les algorithmes le plus couramment utilisés, dérivés de l'algorithme de pro-

pagation de croyance BP (*Belief Propagation*, sont décrits, et particulièrement, les algorithmes Min-Sum et Min-Sum normalisé qui servent de base aux architectures de décodage proposée dans ce travail de thèse.

- Le première partie du chapitre trois débute par un rapide état de l'art sur les méthodes utilisées pour réduire la complexité de codage liée à la structure de la matrice de contrôle de parité du code. Dans la deuxième partie, une nouvelle architecture de codeur paramétrable et proposées pour les codes QC-LDPC (*Quasi-Cyclic Low-Density Parity-Codes*). Cette architecture permet, par paramétrage le choix des caractéristiques du codes (taille, rendement) au sein d'une classe de codes QC-LDPC), et de sélectionner le niveau de traitement parallèle dans l'architecture, qui conditionne les performances (vitesse) et coût (ressources matérielles consommées) de l'architecture. Cette architecture est détaillée, en particulier la fenêtre de calcul sur laquelle repose le principe de fonctionnement de l'architecture. La troisième partie de l'architecture présente les résultats d'implémentation pour différents codes issus de normes actuelles de communications sans fils.

- Le quatrième chapitre concerne la conception de décodeurs pour des mêmes types de codes QC-LDPC qu'au chapitre 3. Comme pour le codage, l'architecture proposée est paramétrable, permettant de choisir lors le l'implémentation, le code (taille, rendement) ainsi que le niveau de traitement parallèle souhaité. Les deux phases importantes de l'algorithme de décodage (de type *message passing* sont présentées plus en détail afin de permettre une meilleure compréhension des éléments constituant l'architecture et décrits dans la suite. Le chapitre se termine par une présentation de résultats de validation de l'architecture par implantation sur FPGA pour des codes issus de normes de communications sans fils actuelles.

INTRODUCTION AUX TECHNIQUES DE CODAGE DE CANAL AVANCÉES

Chapitre 1

Introduction aux techniques de codage de canal avancées

Ce premier chapitre introduit les techniques de codage de canal avancées. Tout d'abord, la notion de codage de canal (ou codage canal) ainsi que quelques notations seront introduites. Puis, dans une deuxième partie du chapitre, nous présenterons les codes correcteurs d'erreur, qui sont largement utilisés pour augmenter la capacité de transmission, voire l'efficacité énergétique, des systèmes de communication sans fil. Les codes convolutifs, les turbo-codes et les codes LDPC (*Low Density Parity Check codes*) seront plus particulièrement ciblés.

1.1 Système de communication numérique

1.1.1 Généralités

Le schéma de principe d'une chaîne de communication numérique est présenté dans la figure 1.1, mettant en relief le rôle important du codage canal. Celui-ci repose sur les techniques de codage détecteur d'erreur et/ou de codage correcteur d'erreurs. Le codage détecteur d'erreur repose sur l'emploi de codes détecteurs d'erreurs (EDC – *Error Detecting Codes*) et de procédures de retransmission en cas de détection d'erreurs. Le codage correcteur d'erreur repose lui sur l'emploi de codes correcteurs d'erreurs (ECC – *Error Correcting Codes*) permettant la correction des erreurs sans procédures de retransmission.

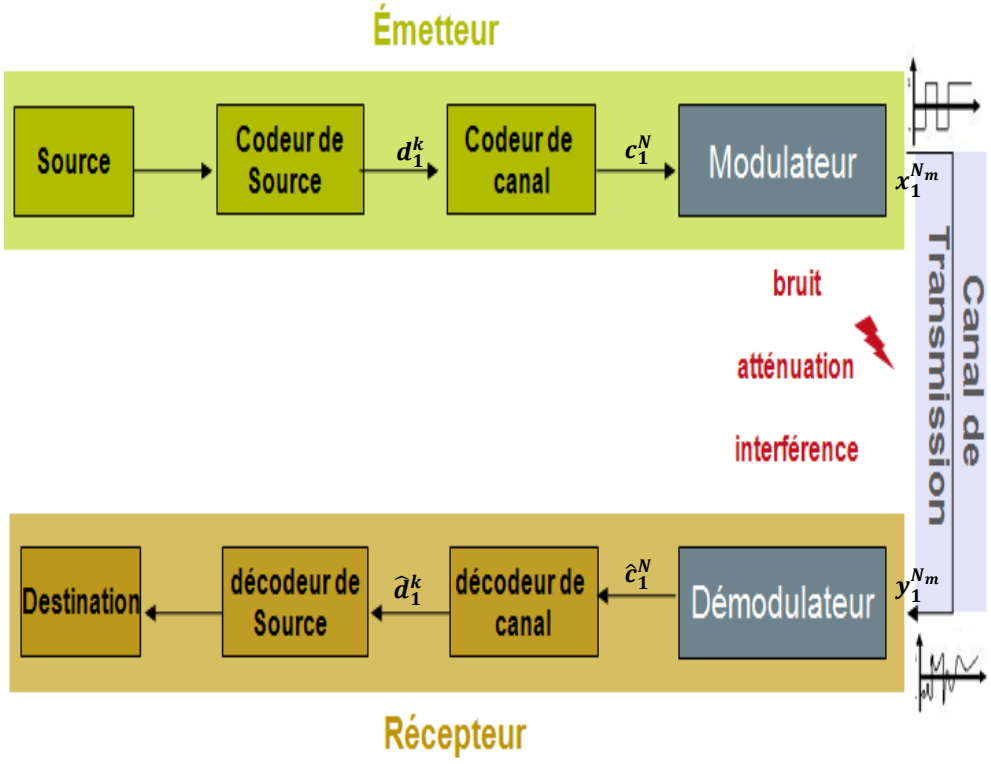


FIGURE 1.1 : Schéma de principe d'une communication numérique.

A l'émission, on trouve la source qui met en forme et prépare les données à transmettre (par exemple des données audio, vidéo, traitement de texte, etc.), représentées sous forme binaire [PJ87]. Dans le cas d'une source analogique, une première étape de conversion doit être réalisée afin de représenter numériquement le signal à transmettre. La deuxième étape consiste à procéder au codage source, dont le rôle est de supprimer la redondance dans les bits d'informations et d'ainsi réduire le volume de données à transmettre, par exemple, en compressant une image au format JPEG. Le message binaire issu de cette phase en composé de k éléments binaires, et est noté $d_1^k = \{d_1, \cdots, d_k\}$ dans la suite. Ce message binaire, composé des information utiles à transmettre, passe ensuite par le bloc de codage canal. Cette opération vise à transformer le message pour le protéger contre les diverses altérations qu'il est succeptible de subir pendant la transmission, en conséquence des perturbations affectant le canal de communication (bruit, distorsions, etc.). Suite à cette opération, un mot binaire de n éléments binaires est obtenu. Ce message codé est dans la suite noté $c_1^n = \{c_1, \cdots, c_n\}$. Ce codage canal est également appelée codage détecteur/correcteur d'erreurs. Il consiste à introduire dans le message à transmettre, une redondance en complément de l'information utile. On définit le rendement R du co-

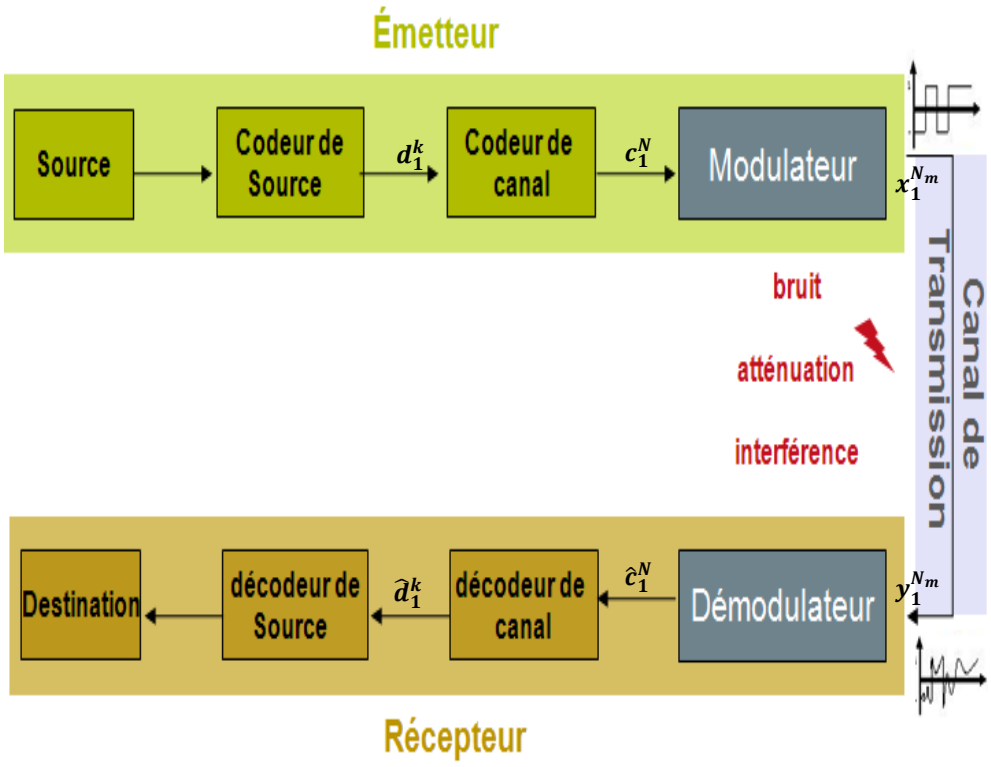


FIGURE 1.2 : Schéma de principe d'une communication numérique.

dage canal par l'équation 1.1. Cette opération de codage est détaillée dans la suite de ce chapitre.

$$R = k.n \tag{1.1}$$

Une fois la redondance ajoutée à information utile, le message passe par le modulateur dont le rôle est de transformer les séquences numériques codées c_1^n en un signal analogique aux formes d'onde appropriées à sa propagation $x_1^{n_m} = \{x_1, \cdots, x_{n_m}\}$. La modulation peut être réalisée en faisant varier une combinaison quelconque des paramètres amplitude, phase ou fréquence d'un signal sinusoïdal appelé porteuse. Le signal analogique, correspondant au train binaire composant le message codé, est alors transmis sur sur le canal de transmission bruité.

Le canal de transmission est le milieu de communication sur lequel les données sont transmises (par exemple : air, câble en cuivre, fibre optique, etc.). Les principales caractéristiques limitant la capacité de transmission des canaux réels sont le bruit (notamment

thermique) et la largeur de bande utile limitée.

À la réception, la séquence analogique reçue, notée $y_1^{n_m} = \{y_1, \cdots, y_{n_m}\}$, contient n_m éléments. À partir de cette séquence $y_1^{n_m}$, le démodulateur reconstitue une séquence numérique correspondante de n bits, présentée en entrée du module de décodage canal. Celui-ci cherche à reconstituer une estimation $\hat{c_1^n} = \{\hat{c_1}, \cdots, \hat{c_n}\}$ du message émis, et en extraire une estimation $\hat{d_1^k} = \{\hat{d_1}, \cdots, \hat{d_k}\}$ de l'information utile codée dans le message émis. Pour ce faire, le décodeur canal exploite la redondance introduite par le codeur canal dans le but de détecter, puis de corriger les erreurs ayant pu survenir pendant de la transmission. L'ajout de bits de redondance dans le message à transmettre entraîne une perte d'efficacité du système. En effet, ces bits de redondance ne véhiculent pas d'information utile additionnelle et réduisent d'autant plus le rendement que leur nombre est élevé. Cependant, cette perte est équilibrée par le gain en qualité de la transmission obtenu grâce au codage canal. Le calcul de ce gain sera exposé dans les paragraphes suivants.

On distingue deux familles de décodeurs travaillant à partir données (séquences de bits) issues du démodulateur. Le premier type de décodeur est appelé *décodeur à décisions dures* et fonctionne à partir des seules données *fermes* (séquences de 0 et 1) issues du démodulateur. Le second type de décodeur est dit *décodeur à décisions pondérées*. Dans ce dernier cas, le démodulateur fournit les données fermes en les accompagnant de mesures de fiabilité qui sont exploitées par le décodeur [Skl01].

Les différentes transformations réalisées par la chaîne de transmission sont détaillées tout au long de cette section.

1.1.2 Canal de transmission

Le principe d'une transmission étant de se faire à distance, il faut disposer d'un support physique assurant le lien entre la source et la destination, et d'un signal approprié à ce support physique, c'est-à-dire, capable de se propager correctement sur ce support (par exemple, un signal électrique sur un fil conducteur électrique ou un signal électromagnétique dans l'atmosphère). Cependant, quel que soit le type de support, les propriétés physiques du canal de transmission vont inévitablement engendrer une dégradation du signal transmis [Bla90] .

Les différentes perturbations ont été modélisées et de nombreuses représentations de

ces dernières ont été réalisées. Le canal à bruit blanc additif gaussien (AWGN – *Additive White Gaussian Noise*) est le canal de référence pour étudier les codes correcteurs d'erreurs et les algorithmes de décodage qui leur sont associés. Le canal AWGN suppose qu'un élément de bruit b_i est ajouté à l'élément émis x_i. L'équation 1.2 représente l'impact du canal AWGN sur la séquence de bits transmise.

$$y_i = x_i + b_i \tag{1.2}$$

En conditions parfaites de modulation, synchronisation et démodulation, il est possible de représenter le canal à l'aide d'un modèle discrétisé. Dans le travail présent, la modélisation du canal est réalisée en additionnant au message incidant (message présenté en entrée du canal), un bruit blanc gaussien de moyenne μ nulle et de variance σ^2. La densité de probabilité de du bruit suit la loi établie dans l'équation 1.4.

$$p(Y_i|X_i) = \prod_{k=1}^{n_m} \left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(Y_{i,k} - X_{i,k})^2}{\sigma^2} \right) \right) \tag{1.3}$$

$$p(Y_i|X_i) = K \cdot \exp\left(\frac{\sum_{k=1}^{n_m} Y_{i,k} \cdot X_{i,k}}{\sigma^2} \right) \tag{1.4}$$

Les taux d'erreurs binaires (BER – *Bit Error Rate*) et d'erreurs de trame (FER – *Frame Error Rate*) propres au codage canal dépendent du rapport signal/bruit E_b/N_0, où E_b est l'énergie par bit d'information et N_0 est la densité réelle du spectre de puissance du bruit. La variance du bruit gaussien σ du canal est fonction du rapport signal/bruit et du rendement R du codage canal.

$$\sigma = \sqrt{\frac{1}{2R \times \frac{E_b}{N_0}}} \tag{1.5}$$

1.1.3 Modulations numériques

La modulation consiste à adapter les données à transmettre, qui appartiennent au domaine numérique, en signaux analogiques adaptés au canal physique sur lequel la transmission doit être effectuée. Dans les modulations habituelles, le procédé consiste à faire varier l'un ou plusieurs paramètres d'une onde sinusoïdale que sont l'amplitude, la fréquence et la phase, en fonction des bits du message à transmettre. Plus précisément, le

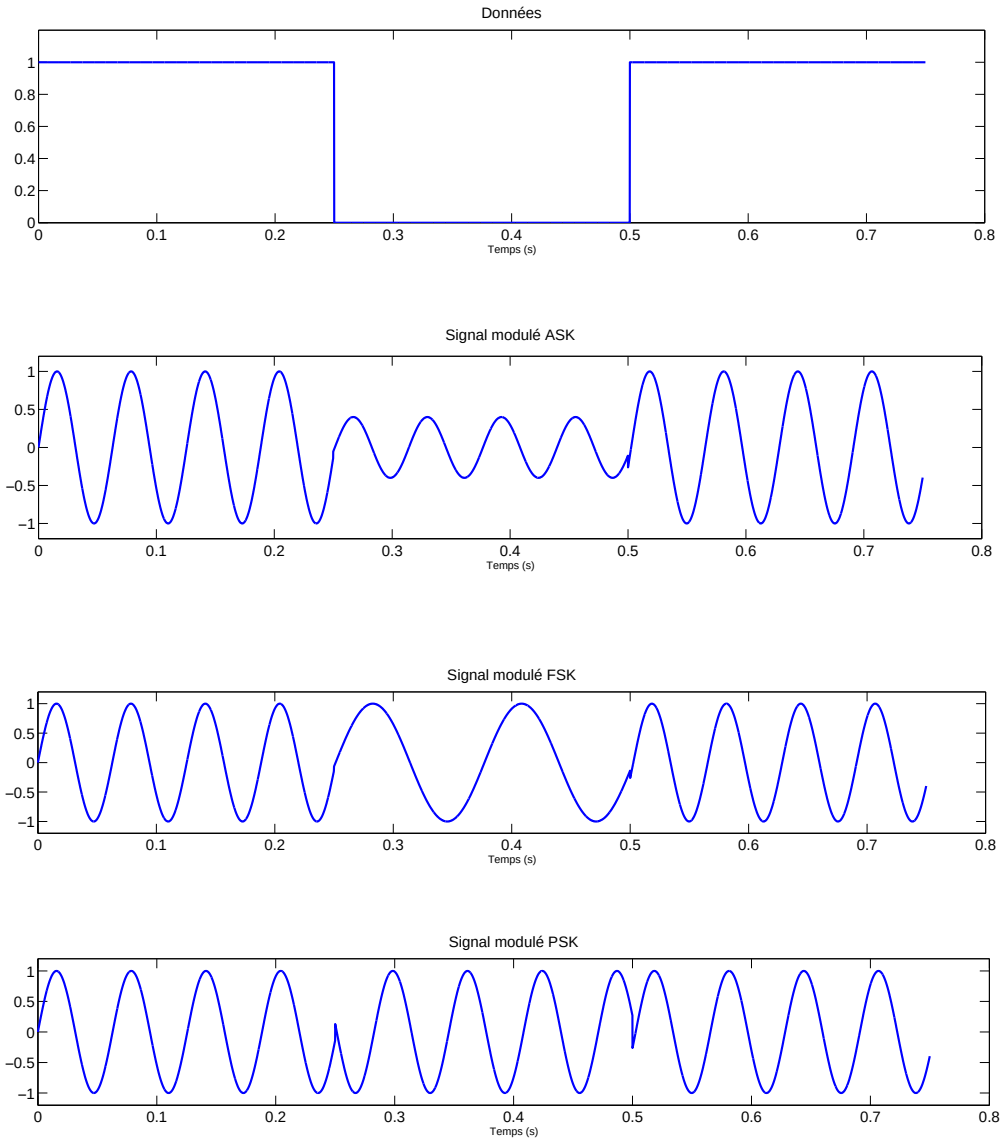


FIGURE 1.3 : Chronogrammes des différents types de modulation numérique

message est préalablement découpé en petit blocs de bits, ou symboles(représentant un nombre entier de bits), et ce sont ces symboles que servent à paramétrer l'onde sinusoïdale. La valence (ou valuation) d'une modulation est définie par le nombre différents d'états (paramétrages) que peut prendre l'onde sinusoïdale et dépend donc du nombre de bits d'un symbole.

Les modulations numériques en amplitude (ASK – *Amplitude Shift Keying*) ont pour principe d'ajuster l'amplitude du signal selon le symbole à transmettre [PZB95]. Bien que la mise en œuvre de cette modulation nécessite peu de moyens, elle est très peu utilisée de par sa grande sensibilité aux perturbations du milieu de transmission (bruit, distorsion,

propagation, etc.). La figure 1.3 présente un exemple de modulation ASK bi-valuée. La modulation d'amplitude la plus simple est la modulation bi-valuée OOK (*On-Off Keying*), pour laquelle à l'un des deux états correspond l'amplitude nulle, soit donc, à l'absence de signal.

Avec les modulations en fréquence (FSK – *Frequency Shift Keying*), c'est la fréquence du signal émis qui varie en fonction du symbole à transmettre. La figure 1.3 présente un exemple de modulation FSK bi-valuée.

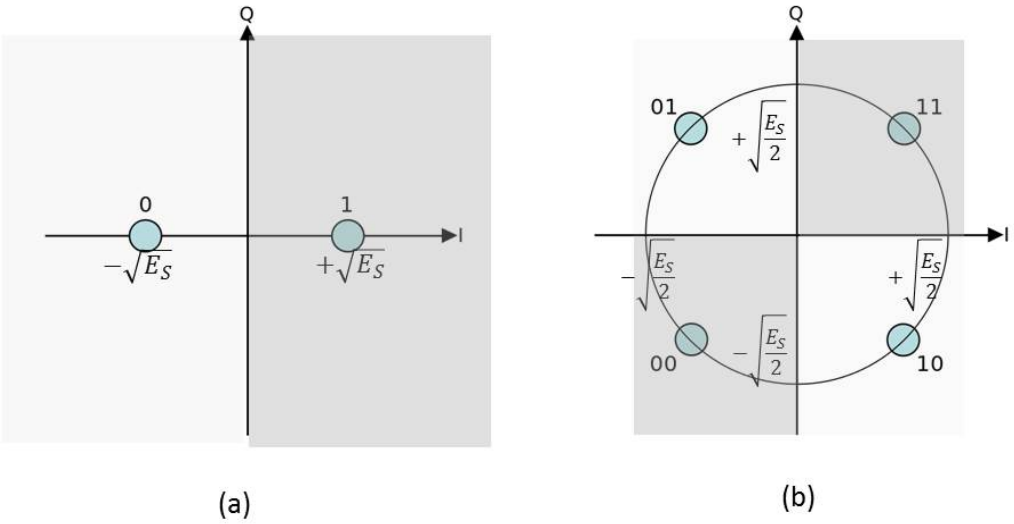


FIGURE 1.4 : Diagramme de constellation pour une modulation BPSK (a) et QPSK (b)

Pour les modulations par changement de phase (PSK – *Phase Shift Keying*), c'est la phase du signal qui est modifiée selon les symboles à transmettre [Skl01]. Cette transformation est habituellement représentée sous forme complexe, un diagramme de constellation (également nommé diagramme IQ pour *Image Quadrature*) représente le codage des symboles comme éléments complexes sur un cercle d'amplitude fixe et répartis selon les racines complexes de l'unité (m racines pour les m symboles d'une modulation m-valuée). On distingue plusieurs types de modulations selon la valuation. La plus simple, de valence 2, est la modulation BPSK (*Binary Phase-Shift Keying*) représentée dans la figure 1.4 et qui utilise deux phases opposées (déphasage de π). La modulation QPSK (*Quadratic Phase-Shift Keying*), représentée dans la figure 1.4b, a une valence de 4 (symboles de 2 bits). Des modulations de phase de valences plus élevées sont possibles (par exemple 8-PSK). Dans la suite de ces travaux, la modulation BPSK est privilégiée comme modulation de référence.

La propagation signal sur le canal de transmission entraîne une déformation de la constellation entre l'émission et la réception. Le démodulateur établit une première décision à la réception, notée $\hat{c_i}$, en fonction du diagramme *IQ*. Pour ce faire, le diagramme de constellation est décomposé en zones de décisions. La figure 1.4 montre les secteurs de décision pour les modulations BPSK et QPSK [NPR05]. Cette décision prise au niveau binaire, appelée décision dure, peut entraîner une forte perte d'information des éléments reçus. En effet, la notion de distance entre un élément reçu y_i et sa décision dure $\hat{c}_i$ est perdue.

On peut donc améliorer cette décision en intégrant la relation de distance entre les éléments et en réalisant ainsi une décision souple. Pour d'intégrer la notion de distance, on définit le *Log-Rapport* de vraisemblance, noté $L^c(c_k)$, comme étant le rapport de probabilité entre les différentes décisions (dures et souples) sur les éléments binaires en fonction des informations du canal. Ce *Log-Rapport* de vraisemblance est défini par l'équation 1.6 dans le cas où $c_k \in \{0, 1\}$, sinon il est défini par l'équation 1.7.

$$L^c(c_k) = \log\left(\frac{Pr\{c_k = 0|y_k\}}{Pr\{c_k = 1|y_k\}}\right) \tag{1.6}$$

$$L^{c,i}(c_k) = \log\left(\frac{Pr\{c_k = i|y_k\}}{Pr\{c_k = 1|y_k\}}\right) \tag{1.7}$$

La transmission sur un canal dispersif induit des interférences entre les symboles. En réception une étape d'égalisation devient indispensable afin de réduire l'impact de ces interférences [TKS02]. L'utilisation d'une modulation multi-porteuses permet de limiter les interférences entre symboles. Mais une étape d'égalisation reste toutefois nécessaire pour supprimer les résidus d'interférences, surtout dans le cas où le canal est très sélectif et où le nombre de sous-porteuses n'est pas suffisant pour que l'on puisse considérer que le canal correspondant à chaque sous-porteuse est plat.

1.1.4 Démodulation et égalisation

Démodulation utilisée pour récupérer le contenu de l'information de l'onde porteuse modulée. Il existe plusieurs moyens de démodulation en fonction de la façon dont paramètres du signal de bande de base, tels que l'amplitude, la fréquence ou de phase sont

transmis dans le signal de porteuse comme nous l'avons expliqué dans modulations .

1.1.5 Codage de Canal

Le but de tout système de communication numérique est de transmettre des données à un débit assez élevé, avec peu ou pas d'erreurs de transmission, en utilisant un minimum de puissance. La largeur de bande de fréquences nécessaire pour le signal émis augmente avec le débit (vitesse) de transmission. En effet, plus la vitesse est élevée, plus la forme du signal nécessite de changements rapides dans le temps, ce qui fait croître la bande passante nécessaire. Cependant, l'inertie du canal (plus ou moins importante, en fonction de la bande de fréquences et du média de transmission) entrave les changements brusques du signal. Différents modèles probabilistes ont été proposés pour simplifier l'analyse mathématique, comme par exemple le canal *AWGN* (de bruit blanc gaussien additif) présenté précédemment ou encore le canal de Rayleigh.

Pour un canal de largeur de bande B, il existe une limite supérieure pour le débit de données (vitesse V_{max}) liée à l'énergie $\frac{E_b}{N_0}$ par bit du bruit du canal. Cette limite maximale est appelée capacité du canal et est définie par la limite de Shannon [Sha48]. Cette limite pour un canal *AWGN* est donnée par l'équation 1.8.

$$V_{max} = B \cdot log_2(\frac{E_b}{N_0}) \tag{1.8}$$

Le perturbation induites par le canal de transmission ne permettent pas de transmettre le flot de données (plus précisément, le flot de symboles) sans altération. De ce fait, la capacité de canal n'est qu'une capacité théorique qu'il n'est possible d'approcher que par l'utilisation de techniques de codage canal. Comme indiqué précédemment, parmi les techniques de codage canal généralement utilisées figurent les codes correcteurs d'erreurs. Plusieurs types de codes correcteurs d'erreurs ont été proposés jusqu'à présent, l'objectif étant d'approcher autant que possible la capacité de canal. On peut citer les codes en blocs, tels que Hamming, Reed-Solomon ou LDPC [DM98], les codes convolutifs et les turbo-codes [BG03].

La Figure 1.5 illustre les taux d'erreurs résiduels pour divers codes correcteurs d'erreurs de rendement semblable par rapport à la capacité du canal, en fonction du rapport

signal/bruit [Vel12].

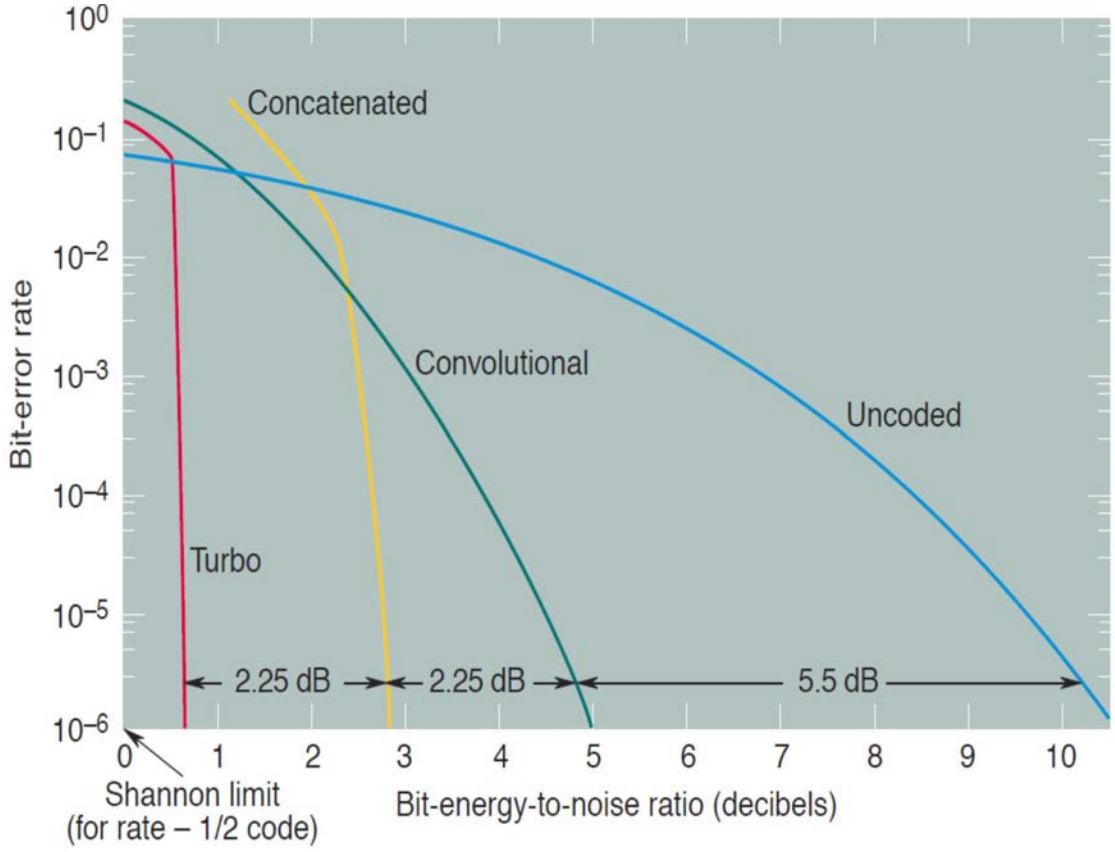


FIGURE 1.5 : Taux d'erreurs résiduels pour divers codes Turbo, LDPC et convolutifs

Bien que de familles différentes, les turbo-codes et les codes LDPC s'avèrent posséder des capacités proches [GE09]. On peut voir dans la figure 1.5) que pour un taux d'erreur binaire cible de 10^{-6}, l'utilisation de codes convolutifs ou de codes concaténé permettent d'obtenir une amélioration de capacité équivalent à une augmentation de (5.5 dB , 7.75 dB) respectivement du rapport d'énergie signal/bruit, par rapport au système non codé. L'utilisation d'un turbo-code ou d'un code LDPC de taux code 1/2 montre une amélioration de 2.25 dB supplémentaires par rapport à un code concaténé, c'est-à-dire, gain de codage total de 10 dB par rapport au système non codé.

Avec l'utilisation d'un Turbo-code de taux 1/2, la performance d'erreur du système se rapproche de la limite théorique de Shannon. Des courbes semblables peuvent être obtenues avec des codes LDPC. Certains des codes LDPC développés jusqu'à présent, permettent même d'approcher d'extrêmement prêt la limite de Shannon [CFRU01] (0.04 dB avec des blocs de longueur 10^7 bits).

1.2 Codes correcteurs d'erreurs

Les systèmes de communication modernes répondent à des exigences très fortes au niveau des performances, concernant notamment les volumes de données à transmettre et la vitesse de transmission, ainsi que l'énergie émise pour la transmission, notamment pour les transmission sans fils. Pour répondre à ces contraintes, les codes de correction d'erreur sont largement mis en œuvre.

TABLE 1.1 : les codes les plus utilisés réseaux de téléphonie mobile

Génération	Standard	Codes Correcteurs d'erreurs
2G	GSM, CDMA, TDMA	codeurs convolutés
3G	W-CDMA, LTE, WiMAX (802.16e), CDMA2000 1xEV-DO	Turbo-codes
4G	LTE-Advanced, WiMAX (802.16n), High Speed Packet Access (HSDPA / HSUPA)	LDPC codes, Turbo-codes

Le Tableau 1.1 résume l'usage qui est fait dans les protocoles de réseaux sans fils (notamment de téléphonie mobile) à haut débit, des codes correcteurs d'erreurs (protocoles de type FEC, *Forward Error Correction*) les plus couramment utilisés, à savoir, les codes convolutifs (CC), les turbo-codes (TC) et les codes LDPC.

Les turbo-codes ont développé dans les années 1993 par Berrou et al [BG96]. Lors de leur création, les turbos codes ont constitué une percée majeure dans le domaine des communications numériques. Ils sont utilisés dans de nombreux standard de téléphonie mobile 3G (UMTS, LTE), de communications par satellites (DVB-S2, DVB-RCS) ou de télévision numérique terrestre (DVB-T2). À ce jour, ils sont également utilisés dans plusieurs réseaux de téléphonie mobile 4G.

Les codes LDPC ont été inventés par Gallager en 1962 [Gal62]. Ces codes permettent à des transmissions sur des canaux bruités de se rapprocher au plus près de la limite de Shannon.Les codes LDPC a été pour la première fois normalisé dans un contexte de diffusion par satellite dans le standard DVB-S2 [Rei06]. Plus récemment, des codes LDPC ont été introduits dans les standards IEEE 802.16e [Ete08a] (Wimax mobile), IEEE 802.11n [iee12] (Wifi) et il est également dans de nombreux réseaux de télépho-

niques de quatrième génération tels que LTE-Advanced et HSDPA/HSUPA). La nécessaire augmentation des débits, la réduction des puissances émises, la rareté de bande passante hertzienne disponible, ainsi que d'autres facteur, ont contribué à fortement accroître l'intérêt porté aux codes LDPC, tant dans le domaine des applications industrielles que de la recherche scientifique.

1.2.1 Codage convolutif

Les codes convolutifs introduits en 1955 par Elias, constituent une classe extrêmement souple et efficace de codes correcteurs d'erreurs. Ces codes sont largement utilisés dans les systèmes de télécommunications fixes et mobiles. Leur popularité est due à leur structure simple et la facilité à réaliser des codeurs ainsi que des décodeurs à maximum de vraisemblance calculée par décision douce. Les codes convolutifs s'appliquent à des séquences de symboles d'information et génèrent des séquences de symboles codées. Un code convolutif diffère d'un code bloc par le fait que chaque bloc de n éléments en sortie ne dépend pas seulement des k entrées à un instant donné mais également des m symboles précédents [Jab09] comme cela est montré dans la figure 1.6. Le codeur peut être vu comme possédant une variable d'état interne représentant son passé, lié à la séquence de blocs déjà traitées.

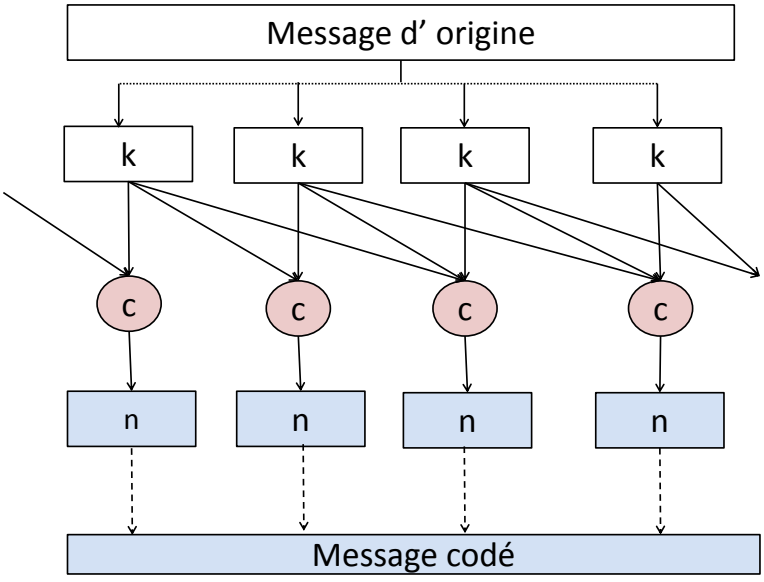


FIGURE 1.6 : Codage convolutif

La capacité potentielle de décodage des codes convolutifs est d'autant plus grande que le nombre d'états pouvant être pris par la variable d'état interne est élevé, pouvant approcher de quelques décibels la limite de Shannon. Cependant, la complexité de décodage s'avérerait très élevé sans l'introduction du principe de décodage. Ce principe est à l'œuvre dans l'algorithme de Viterbi, qui constitue une méthode efficace et relativement simple de décodage pour les codes convolutifs. L'algorithme est employé dans plusieurs systèmes de communication sans fil [JT06,SC06,CL07] : dans les modems séries V.3x, les systèmes téléphoniques GSM, les systèmes de transmissions satellite DVB-S, les normes de télévision mobile telles que DVB-H (Digital Video Broadcasting - Handhelds).

Représentations numériques

Codes de convolution ont été largement utilisés dans des applications telles que l'espace et les communications par satellite, mobile cellulaire, la télévision numérique, etc. Un code de convolution de taux $r = k/n$ est une fonction linéaire qui, à chaque instant i, transforme un symbole entrée d_i de k bits en une sortie codée symbole c_i de n bits ou $(k > n)$. Un code est appelé systématique si une partie de la production est constituée de bits systématique $s_i = d_i$ et le reste des bits $(n - k)$ sont constitués de bits de parité. Les bits de sortie sont la combinaison linéaire du courant et de bits d'entrée précédents. La fonction linéaire est habituellement donnée par polynômes générateurs.

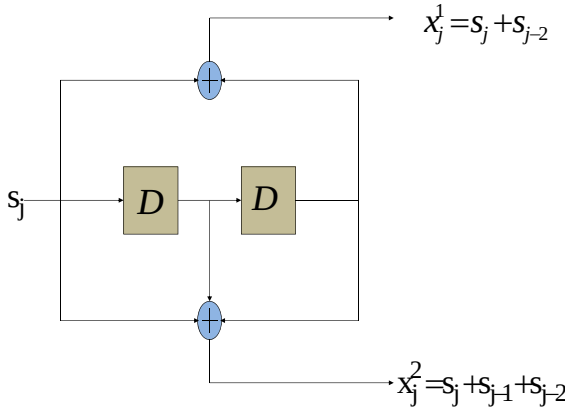


FIGURE 1.7 : Exemple de représentation schématique d'un codeur convolutif

L'exemple de la figure 1.7 représente un codeur convolutif non systématique de rendement $R = 1/2$ et de profondeur de mémoire $m = 2$. À chaque instant j, on combine

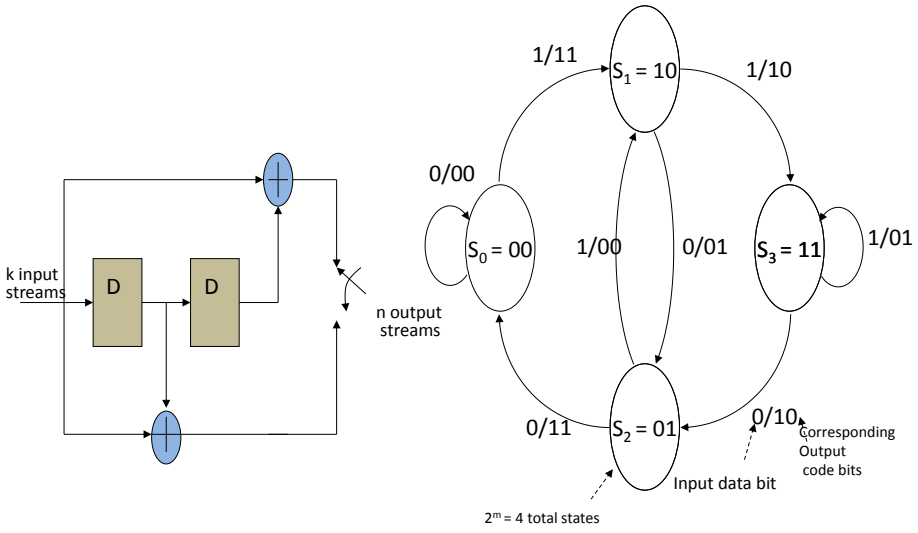
les valeurs d'entrée et de la mémoire (constituée d'un registres à décalage de longueur m) pour générer les sorties, la mémoire (registre à décalage) étant mise à jour en parallèle.

Représentations graphiques

Dans de nombreuses situations, les graphes de représentation s'avèrent plus faciles à manipuler que les représentations numériques et permettent de plus facilement en déduire des résultats exploitables.

Tout code convolutif peut être représenté par trois graphes équivalents mais différents : le diagramme d'état, l'arbre du code et le treillis du code.

Diagramme l'état



(a) convolutional encoder non recursive

(b) diagramme d'état

FIGURE 1.8 : Exemple de diagramme d'état pour un code de profondeur de mémoire $m = 2$

Le diagramme d'état s'avère utile pour le calcul de la fonction de transfert du code [Ben99], sans faire apparaître explicitement le temps. Il représente les transitions possibles entre états. Les valeurs de sortie du codeur sont indiquées sur chacune des transitions. Tous les états internes possibles du codeur sont représentés par des noeuds S_j. La

figure 1.8 représente un tel diagramme d'état. Voici les caractéristique du code correspondant :

- code convolutif à $2^m = 2^2 = 4$ états ;
- rendement $R = 1/2$;
- 8 transitions différentes possible états.

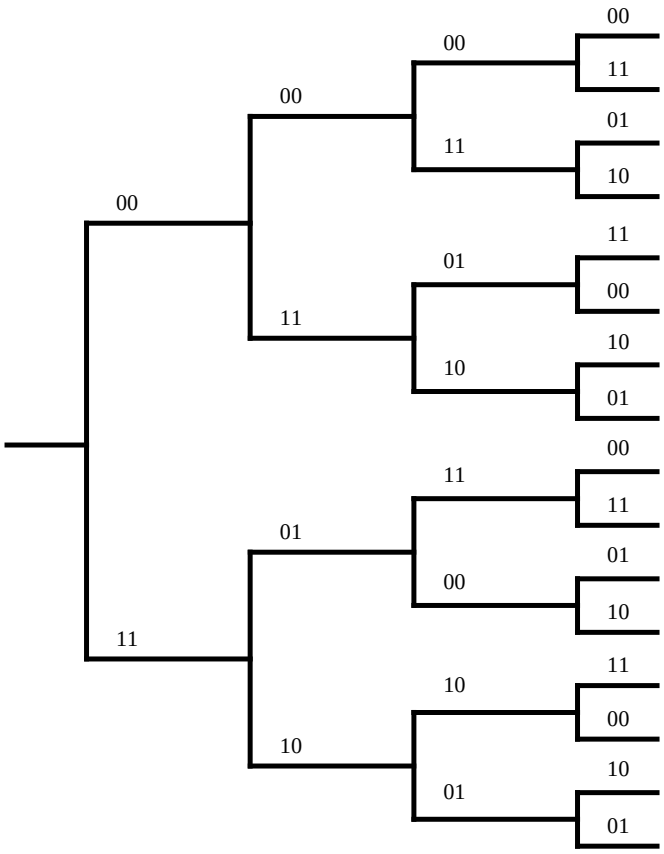


FIGURE 1.9 : Représentation en arbre du codeur de la figure 1.8

La figure 1.9 propose une représentation en arbre du codeur de la figure 1.8. Chaque arrête symbolise une transition possible entre deux états consécutifs. Classiquement l'arbre commence à son sommet par l'état 0 (le registre à décalage est initialisée à 0). Chaque parcours dans l'arbre du code correspond à une séquence possible (un mot de code) en sortie du codeur convolutif. Si par exemple, la séquence d'information présentée en entrée de ce codeur "1001", alors le mot code associé en sortie est "11011111".

Représentation en treillis

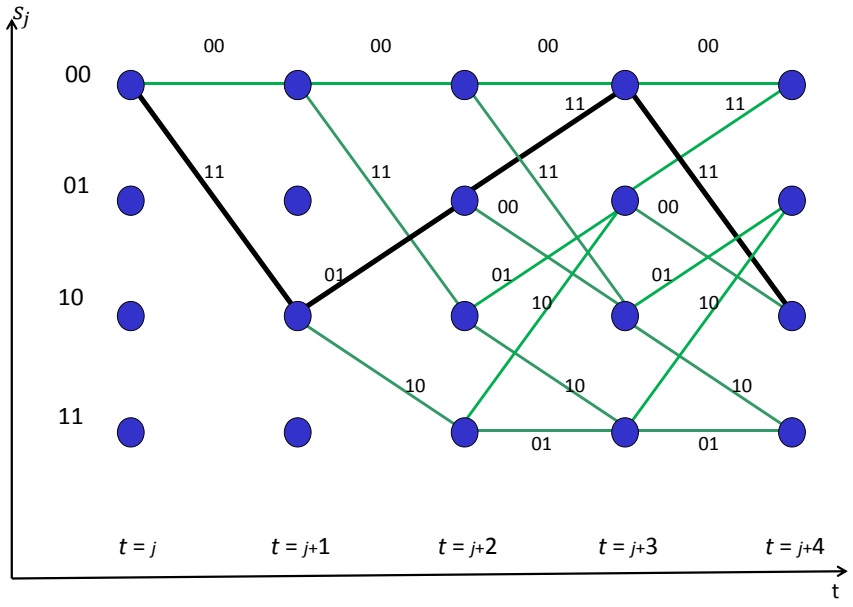


FIGURE 1.10 : Représentation en treillis du codeur de la figure 1.8

Le treillis est une expansion du diagramme d'état en fonction du temps. Le treillis est obtenu en repliant l'arbre sur sa largeur, par fusion des sommets représentant le même état au même instant. Le diagramme en treillis correspondant au codeur convolutif de la figure 1.8 est représenté dans la figure 1.10. L'axe des abscisses représente l'instant t (discrétisé) et l'axe des S_j, les 2^m états du codeur convolutif. Les points correspondent aux états successifs possibles du codeur. Les transitions entre les états sont représentées par des arcs (branches) orientées de gauche à droite. Le diagramme en treillis est ainsi construit en reliant via des arcs toutes les évolutions possibles d'un état vers autre. L'évolution temporelle du codeur, pour une séquence d'information donnée, est déduit du diagramme en treillis, par la succession d'états connectés par les arcs. Une telle chaîne d'états est appelée un chemin dans le treillis. À chaque chemin est associée une séquence codés de bits. Le décodeur utilise des observations de la séquence codée pour tenter de récupérer le chemin associé dans le treillis, pour ainsi en déduire, et la séquence d'information originelle (non codée) qui a conduit à cette séquence codée.

Décodage

Pour les transmissions sur des canaux de communication sans effet de mémoire, les systèmes utilisant le codage convolutif sont parmi les plus intéressants, tant pour leurs performance en termes de débit (approchant de près la capacité théorique de Shannon) que pour leur réalisation et implantation matérielle.

Les deux principales techniques de décodage des codes convolutifs sont le décodage de Viterbi [FJ73] et le décodage séquentiel [For74]. Chacune de ces deux techniques cherche à retrouver un chemin particulier (correspondant au message transmis), dans un graphe orienté où l'on assigne aux branches des métriques ou valeurs de vraisemblance entre les données reçues et les données qui auraient pu être transmises.

L'algorithme de Viterbi peut être modifié afin de fournir à sa sortie une valeur de confiance ou de fiabilité (approximant une probabilité a posteriori) associée à chaque bit décodé. Les valeurs des bits décodés sont toujours données par le chemin ayant la métrique cumulée minimale selon l'algorithme de décodage.

1.2.2 Turbo-codes

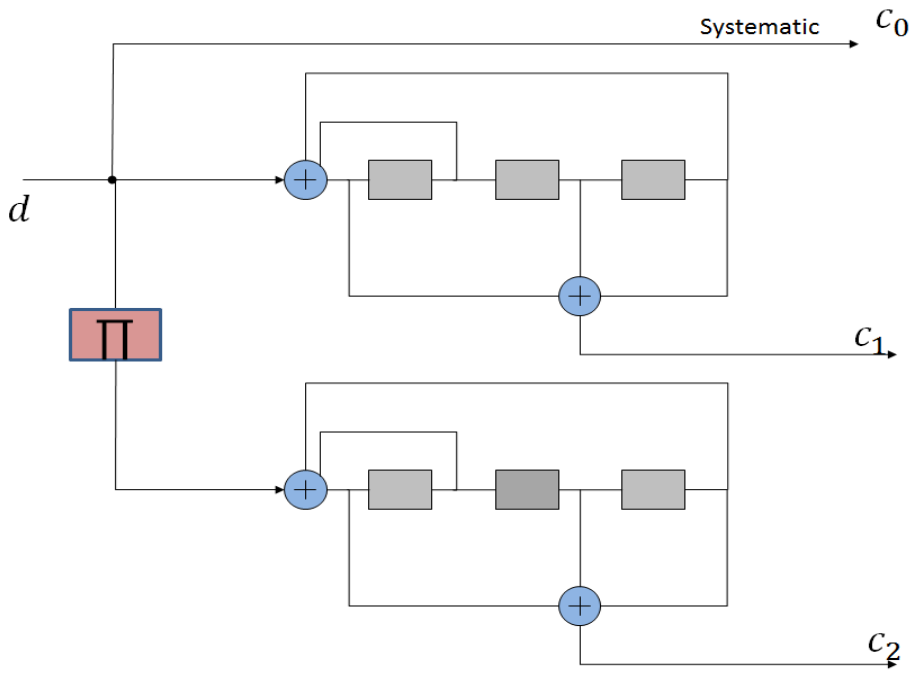


FIGURE 1.11 : Schéma de principe d'un codeur pour turbo-code

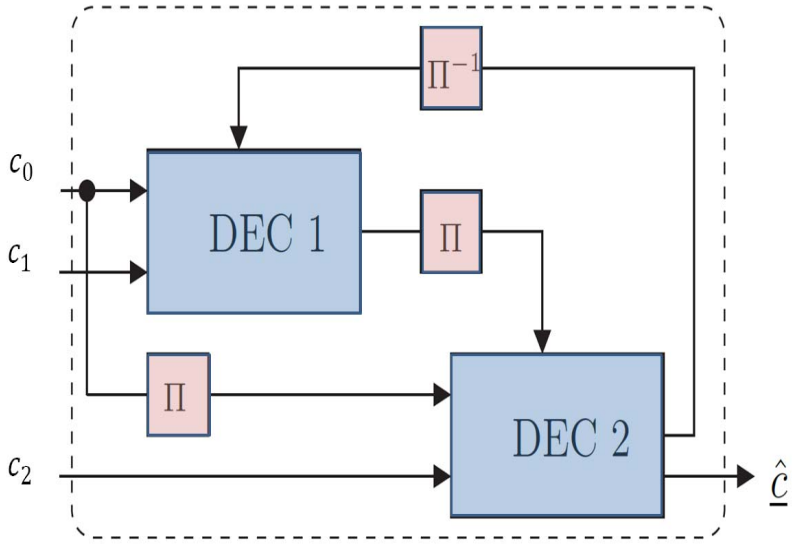


FIGURE 1.12 : Schéma de principe d'un décodeur pour turbo-code

Les turbo-codes a l'origine sont des codes construits sur la base de codes convolutifs et sont les plus utilisés parmi ceux-ci. Leur nom générique a été proposé par [BG96]. Au turbo-codage correspond un nouveau schéma de codage/décodage construit à partir d'une concaténation parallèle de deux codeurs convolutifs dont les entrées sont séparées par un entrelaceur. Le but est d'assurer une décorrélation entre les entrées des deux codeurs et assurer ainsi une meilleure diversité temporelle. Le principal intérêt des turbo-codes n'est pas dans le schéma de codage, mais plutôt dans le schéma de décodage lequel introduit un échange itératif d'information permettant d'exploiter au mieux la diversité temporelle. La figure 1.11 montre le schéma de principe d'un codeur turbo-code, construit à partir de deux codeur convolutifs récursifs et d'un entrelaceur $\prod$. Le schéma de principe d'un décodeur turbo est fournit dans la figure 1.12. La notation $\prod$ représente la fonction d'entrelacement et $\prod^{-1}$ celle entrelacé de dés entrelacement. À partir des symboles reçus c_0 et c_1, le premier décodeur convolutif (DEC 1) calcule une information extrinsèque qui est échangée avec le second décodeur (DEC 2). De même, le décodeur DEC 2 calcule à partir des observations du canal entrelacées et c_2 une nouvelle information extrinsèque, accompagnée d'une estimation des bits d'information. Ce processus peut être itéré autant de fois que nécessaire. Cette opération permet à l'ensemble de fournir au final des performances très proches de la limite de Shannon.

1.2.3 Codes correcteurs en blocs

Historiquement, les codes correcteurs en blocs ont été les premiers codes correcteurs d'erreur crées, notamment le code de Hamming, dont la redondance d'information est basée sur le calcul de parités.

Avec le codage en bloc, le flot de données à transmettre est découpé en blocs (ou mots) de données de taille fixe, chaque bloc étant codé indépendamment des autres. Plusieurs types ou familles de codes en blocs existent, mais une caractéristique commune de tous les code en bloc est d'être définis notamment par la paire de paramètres (n, k) où k est la taille des blocs de données à coder et n celle des blocs codés. Autrement dit, à chaque séquence de k bits présentée en entrée du codeur correspond une séquence codée de n bits en sortie. Le rendement d'un code bloc est par conséquent définit par l'équation suivante :

$$R = \frac{k}{n} \tag{1.9}$$

La capacité de détection et de correction d'erreurs d'un code en blocs dépend à la fois du type de code et du rendement du code. Ainsi, par exemple, à rendement égal, des données codées selon un codage Reed-Solomon pourront mieux résister aux altérations induites par le canal de transmission, qu'avec un codage de Hamming. À type de codage identique, la protection sera d'autant plus élevé que la redondance sera importante et donc, le rendement faible.

1.2.4 Codes LDPC

Les codes LDPC (*Low-Density Parity-Check*) appartient à la classe des codes en blocs. Bien qu'inventés en 1963 par Gallager, ils sont restés quasiment ignorés pendant de nombreuses années, en bonne partie, à cause de la très grande complexité pratique des algorithmes de codage et décodage. Ce n'est que depuis 1997 qu'ils ont été rédécouverts, de manière indépendante, par [MN97] et [RU01a]. Ils ont depuis acquis un forte popularité, en tant qu'alternative aux turbo-codes, en raison de leur excellente capacité de correction d'erreur, à rendement égal, par rapport aux autres codes.

Les codes LDPC sont, comme pour tout code en blocs, défini par les dimensions (n, k). La principale difficulté avec ces codes LDPC provient des valeurs très élevés que

doivent prendre n et k pour pouvoir assurer à la fois des gains de codage significatifs et des rendements très élevés [Eli54,CFRU01]. Des valeurs aussi élevées, ou plus, que celles employés pour le WiMAX [Ete08b] ($576 \leq N \leq 2304$), ou ($16200 \leq N \leq 64800$) le standard VB-S2 [Rei06] (*Digital Video Broadcasting – Satellite 2*).

Plus la taille n d'un code LDPC est élevée, plus il est possible avec une redondance $n - k$ données (et donc un rendement k/n croissant) d'approcher la capacité limite de Shannon du canal. Des codes approchant à moins de $0.04dB$ de la limite de Shannon ont pu être crée à ce jour pour des canaux de type AWGN et un BER inférieur à 10^{-6}) avec un décodage itératif et de grandes tailles de données [CFJRU01] ($\approx 10^7$ bits).

Les codes en blocs LDPC sont présentés de manière plus approfondie dans le chapitre suivant.

1.3 Conclusion

Au travers de ce chapitre, nous avons pu identifier et caractériser les éléments d'une chaîne de transmission, tant à l'émission qu'à la réception. Le but principal de la chaîne de transmission est d'assurer la meilleure qualité de transmission possible, c'est-à-dire, la plus faible dégradation possible de l'information transmise (aucune, si possible) avec le débit le plus élevé.

Nous nous sommes tout d'abord focalisés sur le canal de transmission, et plus particulièrement sur les conditions de bruit et de modulation choisies pour servir de référence dans la suite de ces travaux.

Puis, nous avons ensuite abordé l'aspect codage de canal de la transmission. Ce dernier point est central dans le présent travail de thèse. Dans cette partie ont été introduite les deux grandes familles de codes correcteurs d'erreurs que sont le codage convolutif et le codage en bloc. Pour chacune de ces techniques de codage, nous avons identifiés les codes les plus performants et les plus utilisés à ce jours, à savoir, les turbo codes (pour les codes liés à la famille des codes convolutifs) et les codes LDPC (pour les codes en blocs), ces derniers étant ceux approchant le plus de la capacité limite théorique fixée par Shannon.

Les turbo codes, inventés par [BG96], sont utilisés depuis longtemps dans le domaine des communications, notamment en téléphonie mobile de 3ème (UMTS) et 4ème génération

(LTE). L'intérêt porté à ces codes correcteurs provient notamment de la grande flexibilité de ces derniers.

Les codes LDPC ont été récemment introduits dans une variété de standards de communication, tels que les normes IEEE 802.11n (WiFi), IEEE 802.16e (WiMAX), DVB-S2, etc. Cependant, les capacités de ces codes correcteurs d'erreur ne sont actuellement pas pleinement exploitées et ce domaine est un sujet de recherche actif. Le chapitre suivant est consacré aux codes LDPC, et en particulier à la sous-famille des codes QC-LDPC (*Quasi-Cyclic Low-Density Parity Check codes*) adoptée dans les normes récentes de communications sans fils. Une présentation des techniques de codage et décodage sera proposée en partant de la description de ces codes en tant que codes en blocs linéaires.

LOW-DENSITY PARITY-CHECK CODES

Chapitre 2

Low-Density Parity-Check Codes

Les codes LDPC constituent une classe de codes en blocs linéaires. Les codes en blocs linéaires ont la propriété de linéarité et s'appliquent à des blocs de bits du message à coder, d'où leur nom. Le nom *Low-Densisty Parity Check codes* provient des caractéristiques de la matrice de contrôle de parité que l'on peut qualifier de matrice creuse. Elle ne contient qu'une très faible quantité de valeur non nulles (égales à 1) comparativement à la quantité de valeurs nulles (égales à 0). L'avantage principal des codes LDPC est d'être capables d'atteindre des performances très proches de la capacité limite de Shannon, pour une grande diversité de canaux, avec des algorithmes à temps de décodage linéaire (mais complexes toutefois). Leur structure est adaptée aux traitements parallèles intensifs que les utilisations temps-réel imposent.

2.1 Codes blocs linéaires

2.1.1 Propriétés

Un code correcteur d'erreur est dit code en blocs, si pour chaque message d de longueur k composé par de symboles appartenant à l'alphabet X, le codeur de canal produit, au moyen d'une transformation G, un mot code correspondant $c \in C$ de longueur n, avec $n > k$ et où C est le code en blocs considéré. La différence de $n - k$ symboles correspond à la redondance introduite et est utilisée par le décodeur pour détecter les erreurs de transmission et les corriger.

La quantité de redondance nécessaire dépend généralement des caractéristiques du canal de transmission, et en particulier du rapport signal sur bruit SNR (*Signal-to-Noise Ratio*). La redondance est directement liée aux dimensions, et donc, au rendement ou taux du code, défini dans l'équation suivante.

$$R = k/n \tag{2.1}$$

Un code en bloc, tel que défini dans [ACC+93], est linéaire si toute somme d'un nombre quelconque de mots codes est encore un mot code (cf. equation 2.2).

$$\forall c_1, c_2, \cdots, c_q \in C, c = c_1 + c_2 + \cdots + c_q \implies c \in C \tag{2.2}$$

Dans le cadre de ce travail, nous utilisons uniquement des codes binaires avec pour alphabet correspondant X = {0, 1}. Ceci permet l'utilisation de l'arithmétique "modulo 2", avec l'opération "somme modulo 2" représentés par $\oplus$. L'équation 2.2 devient écrit alors 2.3.

$$\forall c_1, c_2, \cdots, c_q \in C, c = c_1 \oplus c_2 \oplus \cdots \oplus c_q \implies c \in C \tag{2.3}$$

Compte tenu des propriétés de l'opération $\oplus$, l'on peut voir que somme "modulo 2" de tout mot de code c avec lui-même produit le vecteur nul. Cela révèle une autre propriété de linéarité des codes binaires : le vecteur nul doit être un des mots de code $c \in C$.

Les propriétés algébriques de certains codes linéaires ont permis le développement d'algorithmes de codage et de décodage efficaces [Moo05], ce qui représente un avantage important.

Nous allons dès à présent détailler les différentes éléments intervenant dans le codage/décodage d'un code bloc linéaire.

2.1.2 Matrice G de génération du code

Considérons le message d de longueur k défini par la relation suivante :

$$d = [d_1, \cdots, d_k] \tag{2.4}$$

Une fois ce dernier codé, nous obtenons le vecteur c de longueur n.

$$c = [c_1, \cdots, c_n] \tag{2.5}$$

Pour générer le mot codé c, on applique l'opération binaire suivante à d :

$$c = d \cdot G \tag{2.6}$$

où la matrice G est appelée matrice de codage et a pour dimensions $k \times n$.

Afin d'être en mesure de décoder correctement le message initial au niveau du du décodeur, deux messages d^i et d^j différents doivents engendrer des mots codes c^i et c^j in C différents. L'analysede l'équation 2.6 montre qu'un mot de code c est une combinaison linéaire des lignes de G. Pour cette raison, et pour garantir que chaque mot de code c est unique, toutes les lignes de G doivent être linéairement indépendantes (c'est-à-dire, G est de plein rang).

Certains codes en blocs sont dits systématiques, ce qui signifie que dans la représentation d'un mot de code c, les bits constitutifs du message d peuvent être directement isolés des bits de contrôle de parité. La représentation de c peut ainsi être décomposée comme indiqué dans l'équation 2.7.

$$c = [c_1, \cdots, c_n] = [p_1, \cdots, p_{n-k}, d_1, \cdots, d_k] = [p|d] \tag{2.7}$$

où $p = (p_1, \cdots, p_{n-k})$ est le vecteur des bits de contrôle de parité et $d = (d_1, \cdots, d_k)$ le vecteur des bits de données. La matrice G peut alors être vue comme étant la juxtaposition d'un matrice de parité P de dimensions $k \times (n-k)$ et d'un matrice identité I_k de dimensions $k \times k$.

$$G = [P \mid I_k] \tag{2.8}$$

2.1.3 Matrice H de contrôle de parité

Si les 2^k mots de code d'un code binaire satisfont $(n-k)$ équations linéaires homogènes et indépendantes en bits c_i avec $iin\{0, 1, \cdots, n-1\}$, le code binaire est linéaire [Moo05]. Ceci revient à disposer d'un système linéaire d'équations homogènes,

que l'on appelle "équation de contrôle de parité", à n variables binaires et $(n-k)$ équations. Ce système définit alors un code linéaire, qui peut être entièrement décrit par sa matrice de contrôle de parité H, de taille $(n-k)k \times n$, définie comme suit :

$$H \cdot c^T = [0] \tag{2.9}$$

où [0] est un vecteur de taille $(n-k)$ et ou les lignes de H doivent être toutes linéairement indépendantes pour que le code puisse être considéré comme un code (n, k). Un code linéaire binaire défini par une matrice de plein rang H peut être également systématique. À partir des équations 2.6 et 2.9, on peut montrer que H et G sont orthogonaux.

$$\begin{aligned} H \cdot c^T = [0] &\Leftrightarrow c \cdot H^T = [0] \\ &\Leftrightarrow m \cdot G \cdot H^T = [0] \\ &\Rightarrow G \cdot H^T = Z \end{aligned} \tag{2.10}$$

où Z est un matrice de dimensions $k \times (n-k)$ ne contenant que des 0.

En appliquant des opérations algébriques élémentaires à partir des équations 2.8 et 2.10, H peut être représenté sous forme systématique suivante :

$$H = [I_(n-k) \mid P^T] \tag{2.11}$$

ou $I_{(n-k)}$ est une matrice identité de taille $(n-k) \times (n-k)$.

Prenons pour exemple la matrice de contrôle de parité suivant d'un code de rendement $1/2$ produisant 4 bits de redondance :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Les équations de parité associées à cette matrice et à un tout mot de code $c =$

$[c_1, c_2, \cdots, c_8]$ sont :

$$\begin{aligned} c_1 \oplus c_5 &= 0 \\ c_2 \oplus c_5 \oplus c_6 &= 0 \\ c_3 \oplus c_6 \oplus c_7 &= 0 \\ c_4 \oplus c_7 \oplus c_8 &= 0 \end{aligned}$$

Outre la représentation selon sa matrice de contrôle H, un code LDPC peut être représentée sous une forme graphique appelée graphe de Tanner [Tan81] ou, plus généralement, graphe factoriel. Un graphe factoriel contient deux types de nœuds : les nœuds de variable et les nœuds de parité. Deux nœuds de chacun des types peuvent être reliés entre eux par une branche (ou arc). Dans le cas des codes LDPC, les nœuds de variables représentent le mot de code et les nœuds contrôle correspondent aux contraintes de parité. Nous appellerons donc les nœuds contrôle, nœud de contrôle de parité. Un nœud de données i est relié à un nœud de contrôle j par un arc, si et seulement si l'élément correspondant à la $j^{\text{ème}}$ ligne et la $i^{\text{ème}}$ colonne de la matrice de contrôle de parité est non nul.

Par convention, les nœuds de données sont représentés par des cercles et les nœuds de parité par des carrés. Un nœud de variable correspondant à un bit du mot de code transmis est représenté par un cercle blanc. Si un bit du mot de code n'est pas transmis (on parle de bit poinçonné), le nœud est représenté par un cercle plein noir et appelé nœud poinçonné ou nœud caché.

La figure 2.1 représente le graphe de la matrice de contrôle de parité définie dans l'exemple précédent. Le mot de code transmis est le vecteur d où le bit c_8 a été poinçonné.

Les codes LDPC les plus performants sont ceux dont la matrice H suit une loi de construction aléatoire. Cependant, les matrice H de construction aléatoire ne sont pas faciles à mettre en œuvre sous forme de circuit numérique dédié de part la grande complexité induite par la structure de connexion irrégulière entre nœuds de contrôle et nœuds de variable dans le graphe de Tanner.

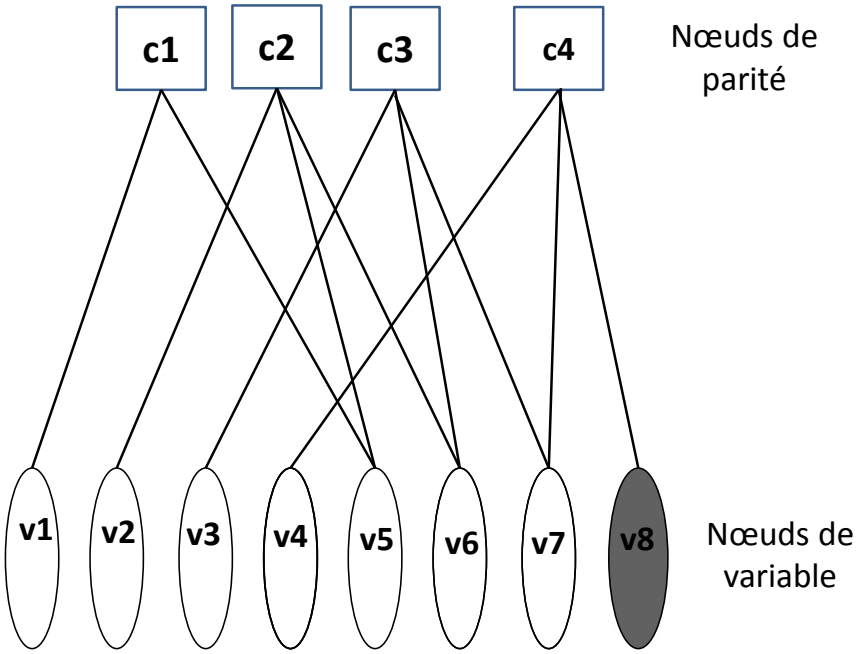


FIGURE 2.1 : Graphe de Tanner d'un code LDPC à 4 nœuds de parité et 8 nœuds de variables

2.1.4 Codes QC-LDPC

Les codes QC-LDPC (*Quasi-Cyclic Low-Density Parity Check codes* constituent une sous-classe de code LDPC. Ils sont très présent dans une grande variété de standards de communications, tels que IEEE 802.11 [BJH+03, PS13] (WiFi), IEEE 802.16e [Ete08a] (WiMAX), DMB-T et DVB-S2 [MM06]. Les matrices de contrôle de parité de ces codes peuvent être décomposées en deux sous-matrices H_d et H_p comme indiqué dans l'équation 2.12. La sous-matrice H_d est de taille $(n-k) \times k$ et est constituée de matrices carrées circulantes que l'on notera $I_{(i,j)}$ dans la suite.

$$H = \overbrace{\left[\begin{array}{cccc} A_{0,0} & A_{0,1} & \cdots & A_{0,L-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,L-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{(T-1),0} & A_{(T-1),1} & \cdots & A_{(T-1),L-1} \end{array}\right.}^{H_d} \overbrace{\left|\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 1 \end{array}\right]}^{H_p} \tag{2.12}$$

Le nombre de sous-matrices dans la matrice H_d est donné par $T * L$ où T est le nombre sous-matrices selon les ligne et L est le nombre sous-matrices selon dans les colonne. Les

matrices $A_{(i,j)]}$, de taille $w \times w$, sont obtenues par décalage cyclique de $(i + j)$ positions d'une matrice décalage, de base. En pratique, cela signifie $A_{(i,j)} = (A_{(0,1)})^{(i+j-)}$ avec $0 \leq i \leq L - 1$ et $0 \leq j \leq T - 1$.

$$A_{(0,1)} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{2.13}$$

H_p est une matrice carrée bi-diagonale de taille $(n - k) \times (n - k)$. Cette structure de matrice rend les processus de codage et de décodage plus simple. Le poids W_c d'une colonne est le nombre de '1' par colonne et le poids W_r d'une ligne est le nombre de '1' par ligne.

Standard	IEEE 802.11n	IEEE 802.16e	T-DMB
longueur de bloc	648-1944	576-2304	7493
Tailles sous-matrice (w)	27-81	24-96	127
taux de codage	1/2, 2/3, 3/4, 5/6,	1/2, 2/3, 3/4, 5/6	2/5, 3/5, 4/5 4/5
nombre sous-matrice dans la colonne (L)	24	24	60
nombre sous-matrice dans la ligne (T)	4-12	, 4-12	24-48

TABLE 2.1 : Paramètres de conception de la matrice H dans diverses normes

Les codes QC-LDPC suivent des organisations très précisément déterminés par les différents paramètres de codage tels que (n, k, w, L, T). Des normes comme le WiFi, Wi-MAX, T-DMB [CLB+07] (Terrestrial Digital Multimedia Broadcasting) et DVB-S jouent des valeurs de ces différents paramètres pour créer des codes QC-LDPC divers. Par conséquent, il est souhaitable de pouvoir construire, pour les systèmes mettant en œuvre de telles normes, des décodeurs possédant une grande souplesse de fonctionnement et de reconfigurabilité. Le tableau 2.1 présente les paramètres de codage LDPC de trois des normes citées précédemment.

Une matrice QC-LDPC irrégulière est définie comme une matrice bloc composée de matrices $I_{(i,j)]}$ ou de matrices carrées nulles de même taille (représentées par un '-' dans la figure 2.2). Celle-ci illustre une matrice QC-LDPC provenant du standard IEEE 802.11n,

de taille m = 972, n = 1944 et de sous-matrice de taille 81 avec un rendement ou taux de code $1/2$. En réutilisant les notations précédemment établies, les matrices QC-LDPC sont aisément représentées.

I_0	-	-	-	I_0	I_0	-	-	I_0	-	-	I_0	I_1	I_0	-	-	-	-	-	-	-	-	-
I_{22}	I_{17}	-	-	I_0	-	I_0	I_0	I_{12}	-	-	-	-	I_0	I_0	-	-	-	-	-	-	-	-
I_6	-	I_0	-	I_{10}	-	-	-	I_{24}	-	I_0	-	-	-	I_0	I_0	-	-	-	-	-	-	-
I_2	-	-	I_0	I_{20}	-	-	-	I_{25}	I_0	-	-	-	-	-	I_0	I_0	-	-	-	-	-	-
I_{23}	-	-	-	I_3	-	-	-	I_0	-	I_9	I_{11}	-	-	-	-	I_0	I_0	-	-	-	-	-
I_{24}	-	I_{23}	I_1	I_{17}	-	I_3	-	I_{10}	-	-	-	-	-	-	-	-	I_0	I_0	-	-	-	-
I_{25}	-	-	-	I_8	-	-	-	I_7	I_{18}	-	-	I_0	-	-	-	-	-	I_0	I_0	-	-	-
I_{13}	I_{24}	-	-	I_0	-	I_8	-	I_6	-	-	-	-	-	-	-	-	-	-	I_0	I_0	-	-
I_7	I_{20}	-	I_{16}	I_{22}	I_{10}	-	-	I_{23}	-	-	-	-	-	-	-	-	-	-	-	I_0	I_0	-
I_{11}	-	-	-	I_{19}	-	-	-	I_{13}	-	I_3	I_{17}	-	-	-	-	-	-	-	-	-	I_0	I_0	-
I_{25}	-	I_8	-	I_{23}	I_{18}	-	I_{14}	I_9	-	-	-	-	-	-	-	-	-	-	-	-	-	I_0	I_0
I_3	-	-	-	I_{16}	-	-	I_2	I_{25}	I_5	-	-	I_1	-	-	-	-	-	-	-	-	-	-	I_0

FIGURE 2.2 : Matrice de parité d'un code QC-LDPC du standard IEEE 802.11n

2.2 Techniques de codage LDPC

Le codage des codes LDPC peut être réalisée selon deux approches.

La première méthode consiste à utiliser la matrice génératrice G. Elle consiste à multiplier la séquence de données par la matrice génératrice pour obtenir la séquence correspondant au mot de code. Cependant, cette approche présente deux inconvénients :

- nécessité de construire la matrice génératrice (systématique ou non) : les codes LDPC étant généralement définis par leur matrice de contrôle de parité, le calcul de la matrice de génératrice doit être réalisé à partir de la matrice de contrôle (à l'aide d'une méthode d'élimination gaussienne, par exemple), préalablement à toute opération de codage. Si le code est connu à l'avance, ce calcul peut être fait *à priori* et n'a aucune incidence sur la complexité de l'étape de codage. Mais si le code n'est pas connu à l'avance et est de plus généré de manière aléatoire, ce calcul peut s'avérer extrêmement coûteux, matériellement et en temps de calcul.

- complexité du codage proprement dit : la multiplication de la matrice génératrice (qui est dense en général) avec les données est très coûteuse en ressources matérielles (elle requiert de l'ordre de $((n-k) \times N)$ opérations).

En pratique, la matrice génératrice n'est pas utilisée pour le codage des codes LDPC en raison de la longueur importante des mots de code. Oenning et al [OM01] propose toutefois de construire directement une matrice génératrice systématique et creuse, de telle sorte que la matrice de contrôle de parité reste ???. Ces codes sont appelés LDGM (*Low Density Generator Matrix*). Les performances de ces codes sont cependant médiocres [Mac99], même s'il est possible d'optimiser leur construction [GFZ03] pour réduire le seuil d'erreur.

La deuxième méthode consiste à utiliser directement la matrice de contrôle de parité. Richardson et al [RU01b] propose une approche pour réduire la complexité de l'encodage. Cette méthode transforme la matrice de contrôle de parité H définissant le code, en une matrice de contrôle de parité à structure triangulaire sur la partie inférieure droite de la matrice, en procédant uniquement à des permutation de rangées et de colonnes. Ensuite, la résolution ainsi simplifiée du système linéaire représentée dans l'équation matricielle 2.9 permet d'obtenir le mot de code.

La complexité de codage des codes LDPC demeure cependant trop élevé, ce qui constitue encore à ce jour l'un problème majeur. Plusieurs travaux ont été réalisées afin de tenter de réduire la complexité de codage, notamment en utilisant des matrices de structures particulières, telles que des matrices triangulaires inférieures [RU01b] ou des matrices semi-aléatoires [YHL03, Lee05]. Les méthodes de codage LDPC seront discutées dans le chapitre suivant avec les différents formes de matrice de contrôle de parité H ainsi qu'une proposition d'architecture matérielle pour le codage LDPC, ciblant une réalisation sur FPGA.

2.3 Algorithmes de décodage LDPC

Deux catégories d'algorithmes itératifs sous-optimaux existent : une catégorie dite "dure" et une autre dite "douce".

Parmi les algorithmes proposées pour le décodage des codes LDPC, ceux du type dur

dérivent généralement de l'algorithme de Gallager [Gal62]. Mais la majorité des algorithmes proposés sont basés sur l'algorithme de propagation de croyance [Gal62] (BP – *Belief Propagation*).

2.3.1 Algorithme de propagation de croyance

Le principe de base de cette technique consiste à échanger de manière itérative des messages (probabilités ou convictions) le long des branches du graphe de Tanner.

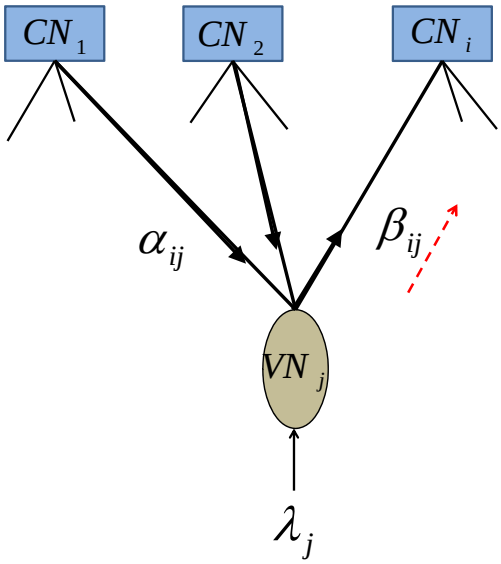


FIGURE 2.3 : Messages β_{ij} se propageant des nœud de données vers les nœuds de contrôle

Deux étapes fondamentales alternent dans l'algorithme de propagation de croyance : la première étape est une phase où sont calculées les messages émis par les nœuds de données à destination des nœuds de contrôle, comme le montre la figure 2.3. Dans l'étape qui succède, les nœuds de contrôle transmettent leur résultat aux nœuds de données (figure 2.4). Cette alternance est répétée un nombre de fois déterminé par l'erreur maximale admissible par l'application. Une prise de décision est nécessaire une fois les itérations effectuées.

La mise à jour des messages α_{ij} émis des nœuds de contrôle vers les nœuds de données est calculée de la façon suivante :

$$sign(\alpha_{ij}) = \prod_{j' \in V(i) \setminus j} sign(\beta_{ij'}) \tag{2.14}$$

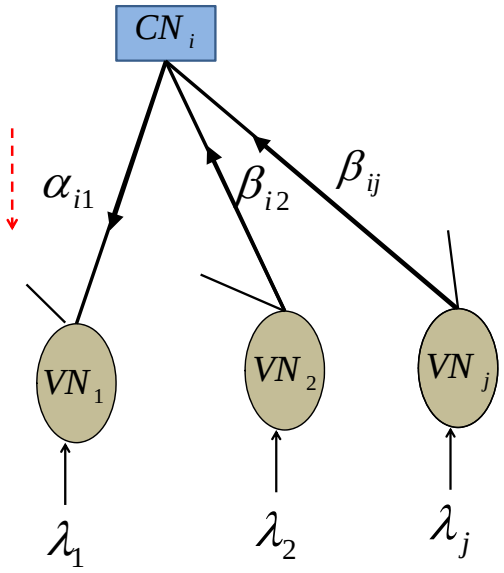


FIGURE 2.4 : Messages α_{ij} se propageant des nœud de contrôle vers les nœuds de données

$$|\alpha_{ij}| = \psi\left(\sum_{j' \in VN(i)\setminus j} \psi\left(|\beta_{ij'}|\right)\right) \tag{2.15}$$

où $j' \in V(i) \setminus j$ représente au nœud de contrôle CN_i l'ensemble des messages qui lui parviennent provenant des nœuds de variables, à l'exclusion du message β_{ij} généré par VN_j, comme illustré dans la figure 2.4.

La fonction non-linéaire $\psi(x)$, définie par l'équation 2.16, est représentée dans la figure 2.5.

$$\forall x \in R_+^\star \Rightarrow \psi(x) = -\psi(x) = -\log\left(\tan\left(\frac{|x|}{2}\right)\right) \tag{2.16}$$

La mise à jour des messages β_{ij} issus du nœud de données V est calculée de la façon suivante :

$$\beta_{ij} = \lambda_j + \sum_{i' \in C(j)\setminus i} \alpha_{i'j} \tag{2.17}$$

où λ est le log-rapport de vraisemblance pour chaque bit n_j en sortie de canal. Le log-rapport est défini par l'équation 2.18, où $\alpha_{i'j}$ représente au niveau du nœud variable VN_j, l'ensemble des messages lui provenant des nœuds de contrôle, calculés lors de la dernière itération, à l'exclusion du message α_{ij} générée par CN_i. La figure 2.3 fourni une représentation graphique de cette étape.

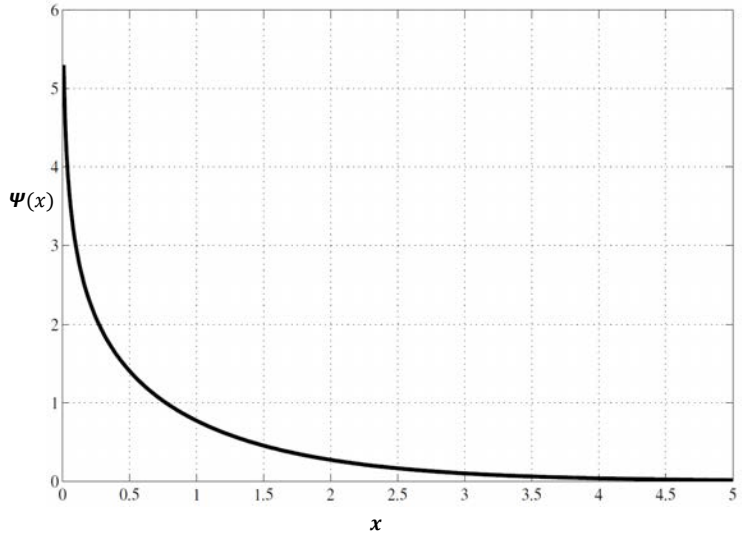


FIGURE 2.5 : Représentation de la fonction ψ

$$\lambda_{n_j} = L_{ch}(n_j) = \log\left(\frac{Pr(n_j = 0)}{Pr(n_j = 1)}\right) \tag{2.18}$$

2.3.2 Algorithmes dérivés de la propagation de croyance

En pratique, des algorithmes dérivés de l'algorithme BP peuvent être considérés. Ces algorithmes doivent être vus comme des simplifications de l'algorithme BP et donc considérés comme moins efficaces. La diminution de performance engendrée par ces simplifications sont à pondérer par les réductions de complexité résultantes pour le décodeur. La majorité des algorithmes dérivés du BP reposent sur la simplification des opérations de mise à jour des fiabilités en sortie des nœuds de contrôle.

Certains algorithmes, dérivés de l'algorithme BP et communément rencontrés dans la littérature, sont présentés dans le tableau 2.2. Les différentes simplifications proposées peuvent être combinées entre elles pour produire construire d'autresx algorithmes de décodage en vue d'atteindre de nouveaux objectif de compromis performance-complexité.

Afin de réduire la complexité de mise en œuvre de l'algorithme BP, on utilise couramment des approximations de la fonction non-linéaire $\psi(x)$. Les deux algorithmes *Normalized Min-Sum* et *offset Min-Sum* sont les plus utilisés [CF02a] et présentés plus en détail ci-dessous. L'algorithme *A-min*, défini par [JVSV03], permet de réduire le nombre de messages à stocker en sortie d'un nœud de contrôle. L'objectif de *APP-Check* est

TABLE 2.2 : Approximations du décodage pour une équation de parité

Nom de l'algorithme	Expressions analytiques approximées de la mise à jour des nœuds de contrôle
Belief propagation (BP) [Gal62]	$\|\alpha_{ij}\| = \psi\left(\sum_{j' \in VN(i)\backslash j} \psi\left(\|\beta_{ij'}\|\right)\right)$
BP-Based/Min-Sum (MS) [FMI99]	$\|\alpha_{ij}\| = \min_{j' \in VN(i)\backslash j}\left(\|\beta_{ij'}\|\right)$
Offset Min-Sum (OMS) [CF02a]	$\|\alpha_{ij}\| = \min\limits_{j' \in VN(i)\backslash j}\left(\|\beta_{ij'}\|\right) + Offset$
Normalized Min-Sum [CF02a]	$\|\alpha_{ij}\| = S \times \min\limits_{j' \in VN(i)\backslash j}\left(\|\beta_{ij'}\|\right)$ S est le facteur de normalisation 0 < S-facteur< 1
λ-min [Gui04]	$\|\alpha_{ij}\| = \psi\left(\sum\limits_{j' \in VN(i)^{\lambda}\backslash j} \psi\left(\|\beta_{ij'}\|\right)\right)$
A-min* [JVSV03]	Si $\lambda = arg \min\limits_{j' \in VN(i)}\left(\|\beta_{ij'}\|\right)$ $\|\alpha_{ij}\| = \psi\left(\sum\limits_{j' \in VN(i)\backslash j} \psi\left(\|\beta_{ij'}\|\right)\right)$ Sinon $\|\alpha_{ij}\| = \psi\left(\sum\limits_{j' \in VN(i)} \psi\left(\|\beta_{ij'}\|\right)\right)$
APP - check [FMI99]	$\|\alpha_{ij}\| = \psi\left(\sum\limits_{j' \in VN(i)} \psi\left(\|\beta_{ij'}\|\right)\right)$
APP-variable [FMI99]	$\|\alpha_{ij}\| = \psi\left(\sum\limits_{j' \in VN(i)\backslash j} \psi\left(\|\beta_{ij'}\|\right)\right)$

de réduire le nombre de messages en sortie d'un nœud de contrôle [FMI99] alors que *APP-variable* vise, quant à lui, à réduire le nombre de messages en sortie d'un nœud

variable [FMI99].

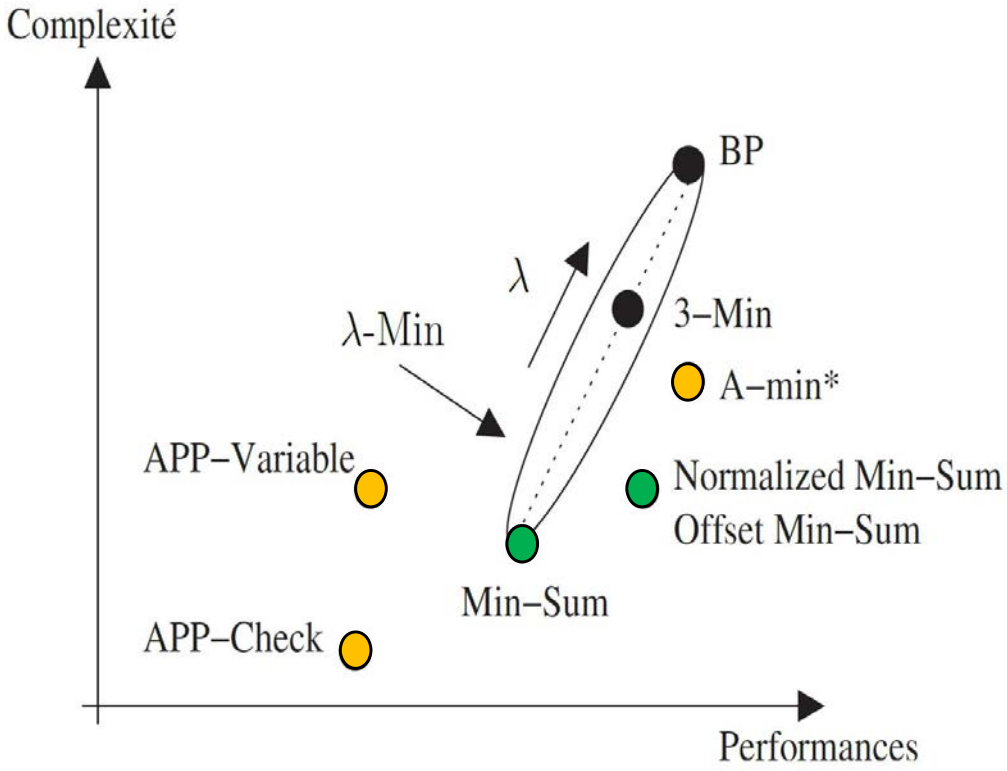


FIGURE 2.6 : Algorithmes de décodage comparés selon le rapport performance/complexité [Dor07]

Algorithmes Min-Sum

L'algorithme Min-Sum réduit la complexité de décodage et est indépendante de la variance du bruit. La plupart des efforts sont faits au niveau du décodeur LDPC pour parvenir à un compromis optimal (voir figure 2.6) entre la complexité et la performance obtenue (erreur résiduelles). Plusieurs approches ont été réalisées afin de maintenir les performances les plus proches possible de la limite de Shannon, tout en réduisant la complexité matérielle. Différentes approches sont utilisées pour aboutir à une forme simplifiée de l'algorithme BP tout en approchant au plus près de ses performances. Les approches les plus populaires sont le *Normalized Min-Sum* et le *offset Min-Sum* [CDE+05].

Pour réduire l'amplitude d'une surestimation, le message de contrôle est modifié pendant le processus d'itération ce qui pour conséquence de permettre aux algorithmes Min-Sum, d'atteindre une limite proche de la performance de la norme SPA [CF02a, CF02b]

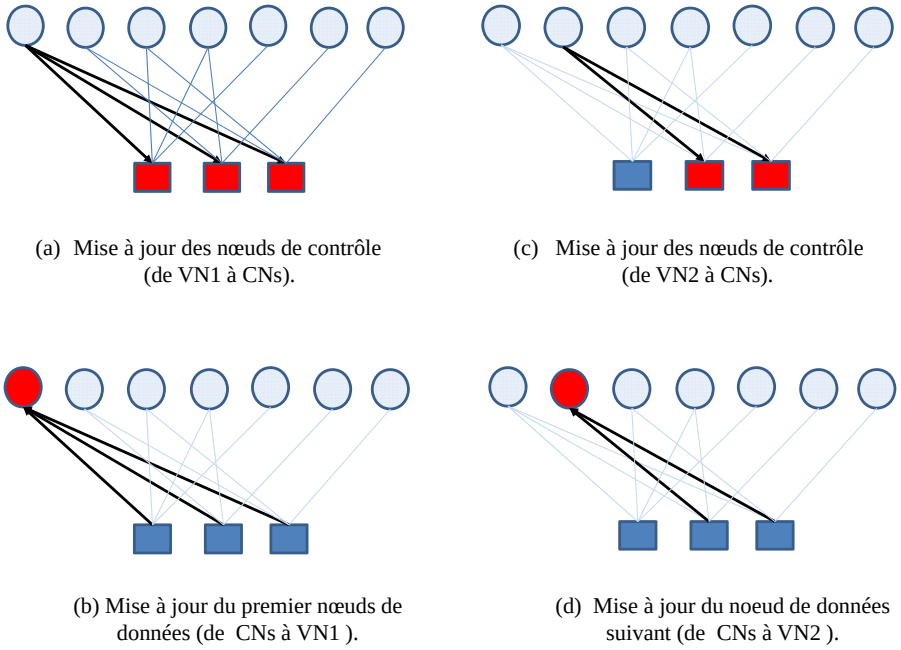
et les rend appropriés pour des applications pratiques [ZZB05]. De longueur modérée, l'algorithme de décodage LDPC min-Min-Sum a gagné large popularité dans le domaine des communication sans fil, en raison notamment de sa complexité réduite.

2.3.3 Algorithme de décodage par ordonnancement

Qu'est ce que l'ordonnancement (*scheduling*) ?

Les algorithmes précédents utilisent l'ordonnancement par inondation ou *flooding scheduling*, ce qui est l'approche standard pour l'algorithme BP. Plusieurs autres techniques d'ordonnancement ont été étudiés dans la littérature et un résumé est présenté ci-dessous.

Vertical Shuffle Scheduling



(a) Mise à jour des nœuds de contrôle (de VN1 à CNs).

(c) Mise à jour des nœuds de contrôle (de VN2 à CNs).

(b) Mise à jour du premier nœuds de données (de CNs à VN1).

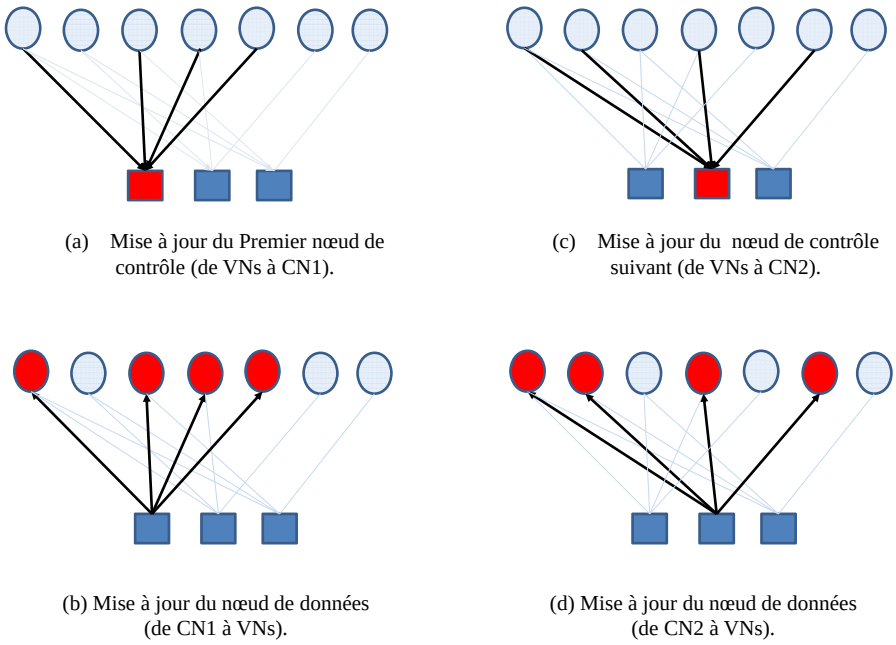
(d) Mise à jour du noeud de données suivant (de CNs à VN2).

FIGURE 2.7 : *Vertical Shuffle Scheduling*

[ZWFY07] a proposé un algorithme BP hybride qui converge plus rapidement que l'algorithme BP standard. La phase d'ordonnancement est réalisée en mettant à jour les informations dès qu'elles aient été calculées afin que le prochain nœud de contrôle (ou nœud de variable) puisse les utiliser. Le processus de décodage développé par [ZWFY07]

est le suivant :

- à chaque étape, tous les nœuds de contrôle connectés aux nœuds de variables sont mis à jour selon l'équation 2.17. Cette étape est est suivie par la mise à jour des nœuds de variables selon l'équation 2.15. Par conséquent, tous les nœuds de variable sont traités les uns après les autres. Le figure 2.7 présente le mécanisme d'action de cette méthode.



(a) Mise à jour du Premier nœud de contrôle (de VNs à CN1).

(c) Mise à jour du nœud de contrôle suivant (de VNs à CN2).

(b) Mise à jour du nœud de données (de CN1 à VNs).

(d) Mise à jour du nœud de données (de CN2 à VNs).

FIGURE 2.8 : *Horizontal Shuffle Scheduling*

Horizontal Shuffle Scheduling

Cet ordonnancement est une alternative de la technique précédente. Dans cette variante, un nœud de contrôle reçoit les mises à jour à partir des nœuds de variable qui y sont connectés. La figure 2.8 décrit le fonctionnement d'un ordonnancement horizontal.

2.4 Conclusion

Dans ce chapitre, nous avons présenté la théorie associée aux codes correcteurs d'erreurs LDPC et les principales techniques de codage et de décodage.

Des mises en œuvres des techniques de codage présentées, ont été réalisées à diverses reprises, selon deux approches :

- l'approche logicielle qui présente de faibles performances, découlant des temps de calculs importants, mais qui est flexible ;
- l'approche matérielle qui, bien que plus performante, n'est pas un sujet actif de la recherche actuelle.

La partie décodage a également été abordée avec un état de l'art des principales techniques utilisées actuellement. Parmi les différents algorithmes présentés, nous avons pu constater la forte prédominance des algorithmes *Min-Sum*. En effet, ces derniers permettent d'atteindre de bonnes performances tout en maintenant un faible niveau de complexité.

Dans le but d'atteindre un bon compromis entre performances élevées et complexité raisonnable, nous proposerons dans le prochain chapitre, une architecture matérielle de codeur LDPC présentant des performances élevées tout en préservant un important degré de flexibilité.

Nous poursuivrons dans le chapitre suivant, avec la proposition et mise en œuvre matérielle d'une architecture de décodeur LDPC, basée sur l'algorithme *N-Min-Sum*, permettant d'atteindre des performances plus élevées que celles rencontrées dans la littérature.

CODEUR QC-LDPC

Chapitre 3

Codeur QC-LDPC

Comme indiqué dans les chapitres précédents, les codes LDPC constituent à ce jour la classe de codes correcteurs d'erreurs la plus performante au sens de leur capacité à s'approcher le plus de la limite de Shannon. Ceci les rends de plus en plus populaires dans les applications de télécommunications. Cette évolution motive les effors de recherche vers la conception de nouveaux codes LDPC et des codeurs/décodeurs correspondants, présentant des caractéristiques de performance, coût et flexibilité toujours meilleures.

Dans le contexte des applications à forte contraintes de traitement temps réel (débit de données très élevés), la demande est forte pour des architectures matérielles rapides, efficaces et à faible coût. Les FPGA constituent une cible technologique favorable, avec une offre devenue au fils des années largement accessibles (disponibilité et coût des composants) et performante (tailles importantes permettant de réaliser des architectures très massivement parallèles. Cette évolution permet désormais d'envisager le développement d'architectures de codeurs/décodeurs offrant une grande flexibilité et l'accès à des débits de transmission toujours plus élevés.

À ce jour, la plupart des travaux consacrés aux codeurs visent principalement à minimiser le coût matériel, offrant des débit relativement modestes comparativement à ce qu'il serait en théorie possible de faire. La raison habituellement avancée est que, dans une chaîne de transmission, les décodeurs étant normalement beaucoup plus complexes et lents, il ne semble pas nécessaire de disposer de codeurs ultra rapide la plupart du temps.

Dans ce chapitre, nous présentons cependant des architectures de codeurs capables

d'offrir des débits plus élevés que ceux habituellement proposés sans pour autant entraîner un surcoût matériel important pour leur mise en œuvre.

3.1 État de l'art sur la conception de codeurs LDPC

Plusieurs types d'architectures ont été présentées dans le passé pour le codeur. Zhang [ZP01b] a proposé un schéma de codage systématique efficace, mais limité à un cas particulier de code LDPC régulier et de paramètres de code fixes. Le processus de codage s'avère plus complexe que pour les turbo codes, requérant par exemple beaucoup d'opérations de multiplication matricielles. Dans [MWD99] et [RU01b], il est suggéré que la complexité de codage peut être considérablement réduite en choisissant une matrice de contrôle de parité triangulaire pour la définition du code LDPC. Sur cette base, Zhang a présenté dans [ZP01a] un schéma de codage efficace pour des codes LDPC généralisés. Dans [LLWJ04, KG06, KAM06], la conception matérielle d'un codeur LDPC efficace est présentée, basée sur Richardson et Urbanke [MWD99]. [SKC06] Récemment, une architecture de codeur/décodeur LDPC à matrice structurée a été présenté, prenant en charge la norme IEEE 802.11n. Deux structures sont couramment utilisées pour la matrice de contrôle de parité afin de faciliter le codage LDPC : la structure triangulaire inférieure approximative et la structure bi-diagonale.

3.2 Matrice de contrôle structurée

Le codes LDPC ont généralement une taille de bloc importante afin d'atteindre avec un fort rendement R, des performances proche de la limite de Shannon. La faible densité de la matrice H de contrôle de parité permet de réaliser avec une faible complexité le décodage par une approche de type graphique. Cependant, l'encodage correspondant n'est ni direct ni de faible complexité, la matrice génératrice G étant généralement inconnue. Une approche usuelle pour les codes en blocs classiques, qui consisterait à coder à l'aide de la matrice génératrice G (déduite de la matrice de contrôle H par une technique d'élimination de Gauss) est difficilement applicable. En effet, bien que la méthode puisse s'appliquer à tout code en bloc, il est peu envisageable de l'employer avec les

codes LDPC. La complexité du codage serait bien trop importante de par la taille des matrices et blocs de données (pré-traitement pour construire G et codage proprement par multiplication matricielles, au minimum d'ordre $O(n^3)$, où n est la taille d'un mot de code).

3.2.1 Structure triangulaire inférieure approximative

Le codage LDPC avec matrice de contrôle triangulaire inférieure est à rapprocher de l'encodage direct de codes en blocs à l'aide de la matrice de parité H. Il existe deux types de méthodes de codage à partir d'une matrice de contrôle H triangulaire inférieure.

La première consiste à utiliser l'élimination de Gauss pour convertir, avant codage, la matrice de contrôle H en une matrice de structure triangulaire inférieure (comme montré dans la figure 3.1). La complexité du codage est alors en $O(n + g^2)$, comme il est facile de le déduire des équations 3.1 et 3.2 (cf. figure 3.1 pour g).

$$p_0^t = -(-EB^-1 + C^-1(-ET^-1A + C)d^t \quad (3.1)$$

$$p_1^t = -T^-1(Ad^t + Bp_0^t) \quad (3.2)$$

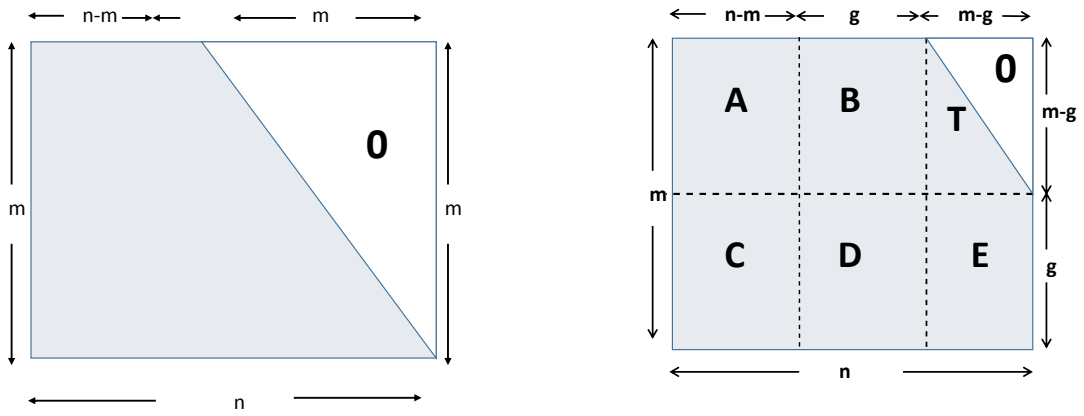


FIGURE 3.1 : Matrice de contrôle : (1) quasi-triangulaire ; (2) triangulaire inférieure approximative

La seconde approche est d'utiliser directement une matrice de contrôle triangulaire

inférieure clairsemée (creuse) pour l'encodage, ce qui peut entraîner une perte de performance de codage. La complexité de codage avec une telle matrice est d'ordre $O(n)$.

3.2.2 Structure bi-diagonale

La structure bi-diagonale pour la matrice de contrôle H s'avère plus efficace pour le codage et le décodage de codes LDPC Au niveau de la consommation surface. notamment ceux adoptés dans lesnormes de réseaux sans fils IEEE 802.11n [TVLZ15] et IEEE 802.16e [ZXZ+15]. La matrice H est générée en utilisant une technique semi-aléatoire. Cette matrice est composée de deux parties [PLP99, JYL10, EC01], H_d et H_p, dont les dimensions sont respectivement $(n-m)\cdot m$ et $(n-m)\cdot n-m)$. Le mot de code systématique $c = [d \mid p]$ est la concaténation de d et p, respectivement les bits d'information (correspondant au bloc de données à coder) et les bits de parité. H_d est constituée de matrices carrées circulantes $I_{(i,j)}$. Ces matrices $A_{(i,j)]}$ sont des matrices de décalage cyclique obtenues par décalage cyclique de $(i+j) \mod w$ positions, où $w \times w$ est la taille de la matrice carrée $A_{(i,j)}$.

$$Hc^t = [H_d \mid H_p][d \mid p]^t = 0 \tag{3.3}$$

$$0 = \overbrace{\left[\begin{array}{cccc|cccc} A_{0,0} & A_{0,1} & \cdots & A_{0,L-1} & 1 & 0 & \cdots & 0 \\ A_{1,0} & A_{1,1} & \cdots & A_{1,L-1} & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\ A_{(T-1),0} & A_{(T-1),1} & \cdots & A_{(T-1),L-1} & 0 & \cdots & 1 & 1 \end{array}\right]}^{H_d \qquad\qquad\qquad\qquad\qquad H_P} \cdot \overbrace{\left[\begin{array}{c} d_0^t \\ d_1^t \\ \vdots \\ d_{n-m-1}^t \\ p_0^t \\ p_1^t \\ \vdots \\ p_{m-1}^t \end{array}\right]}^{\textit{informations et parités}} \tag{3.4}$$

Comme indiqué dans le chapitre 2, le nombre de sous-matrices dans la matrice H_d est $T \times L$, où T est le nombre sous-matrice selon la ligne et L selon la colonne. $A_{(i,j)} =$

$(A_{(0,1)})^{(i+j)}$, avec $0 \leq i \leq L-1$ et $0 \leq j \leq T-1$.

$$P_0 = \sum_j h_{1,j} d_j \tag{3.5}$$

$$P_i = p_{i-1} + \sum_j h_{i,j} d_j \tag{3.6}$$

3.3 Conception du codeur

Le codage LDPC est bien plus complexe que pour les codes en bloc de petite taille ou pour les codes convolutifs. Il est défini à partir d'opérations matricielles, comme cela transparaît de l'équation 3.3, la taille de la matrice H pouvant être très importante, en fonction du code sélectionné dans l'application cible.

Différentes stratégies architecturales de réalisation des codeurs existent, selon les applications cilbes, permettant de s'adapter aux différentes contraintes de débit et taille des codes.

Lorsque les contraintes de débit sont très faible, l'approche logicielle est celle qui convient en général le mieux, combinant faible coût et souplesse dans le développement et l'utilisation. Les logiciels peuvent être conçus pour être exploités sur des ordinateurs de bureaux (ou portables), ou bien sur des systèmes embarqués à base de microcontrôleurs ou processeurs de traitement numérique du signal [KAM06] dits DSP de l'anglais *Digital Signal Processor*). Une telle approche pour la réalisation du codeur est spécifié par exemple dans la norme IEEE P802.16E/D7, ne permettant d'atteindre au mieux qu'un débit d'environ 2.6 Mbs.

Avec l'accroissement des contraintes de débit, il devient nécessaire de passer à une approche matérielle pour la réalisation des codeurs. Plusieurs aspects différencient les architectures matérielles des codeurs, l'un d'entre eux étant le niveau de traitement parallèle supporté.

Aux plus faibles débits requis correspondent normalement les capacités de traitement parallèle les plus réduites. Il est courant de donner le nom d'architecture série (en termes de point de référence pour la capacité de traitement), à une architecture présentant une faible capacité de traitement parallèle et ne permettant d'accéder qu'à de faibles débits. C'est souvent une architecture dont l'interface d'entrée (et/ou de sortie) ne permet de

transférer qu'un bit de données à la fois,ou un nombre très limité tel que 8, 16 ou 32, correspondant à la largeur du bus d'interconnexion du codeur avec la mémoire où se trouvent les données à traiter. La figure 3.2 illustre un tel type de codeur, capable de traiter uniquement un bit à la fois.

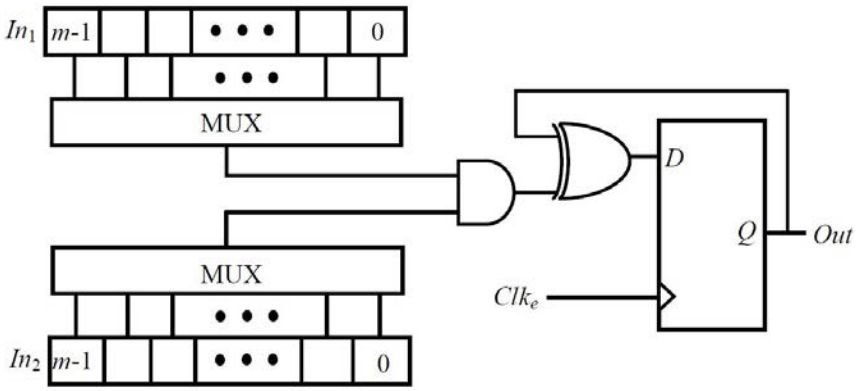


FIGURE 3.2 : Exemple de codeur série pour un code LDPC

À l'opposé, de fortes contraintes sur les débits imposent des capacités de traitement parallèle très élevés, auxquelles vont correspondre des architectures massivement parallèles. Un très fort niveau de traitement parallèle serait nécessaire en général avec un code LDPV, pour être capable d'encoder en un seul cycle d'horloge, un bloc de données en mot de code.

Des niveaux intermédiaires de traitement parallèles permettront de coder un bloc de données en un mot code en un nombre de cycles intermédiaire entre les architectures série et massivement parallèle invoquées ci-dessus. On désignera dans la suite de ce document, une telle architecture par "partiellement parallèle".

Une architecture flexible et extensible du point de vue des contraintes de débit, doit permettre de sélectionner le niveau de traitement parallèle souhaité lors de la réalisation ou implantation matérielle. C'est cet objectif que nous nous proposons d'atteindre.

3.4 Codeur QC-LDPC partiellement parallèle

Les codes LDPC sont utilisés dans de nombreuses applications : il existe plus de 200 normalisés de codes LDPC avec rendement du code et tailles de codes différent. Comme l'on peut aisément le concevoir en observant l'équation 3.3 et la représentation

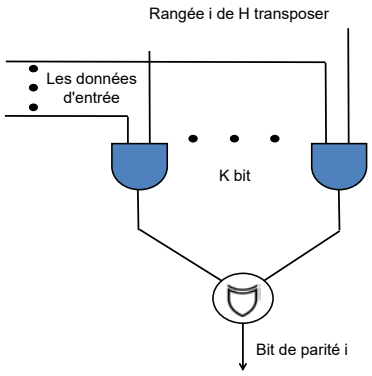


FIGURE 3.3 : Représentation simplifiée d'un codeur LDPC "standard"

en figure 3.3 de la génération d'un bits de mot code, le codage par l'intermédiaire de la matrice de contrôle H peut être effectué de manière parallèle. Ceci découle directement des propriétés H, chaque rangée de sous-matrices dans H étant traitée indépendamment des autres.

Le parallélisme au sein du codeur LDPC peut ne concerner que le traitement d'une seule sous-matrice de H la fois, le processus étant répété jusqu'à ce que la totalité des sous-matrices de H aient été prises en charge (par balayage horizontal et vertical de H). Ce parallélisme peut être renforcé en étendant la zone de traitement parallèle horizontalement et/ou verticalement, ceci en recouvrant des zones de H dont la taille s'étendrait sur l'équivalent de $s \times w$ la taille d'une sous-matrice de H ci-dessus, comme le montre la figure 3.4. Nous discuterons plus loin des avantages et inconvénients qui en découlent.

La durée du codage dépend bien sûr, de la taille n du mot code, du rendement R du code, de la fréquence d'horloge et de la taille de fenêtre de traitement. À chaque cycle, les mêmes traitements (calculs) sont effectués en utilisant les mêmes ressources matérielles, et ceci jusqu'à ce que l'encodage soit globalement terminé.

3.4.1 Architecture du codeur QC-LDPC

Nous proposons ici une architecture de codeur QC-LDPC configurable, plusieurs paramètres architecturaux pouvant être fixés lors de la réalisation matérielle selon les besoins de l'application cible : taille et taux/rendement du code, taux de parallélisme (selon l'approche dont le principe a été ébauché dans la section ci-dessus). Les objectifs visés

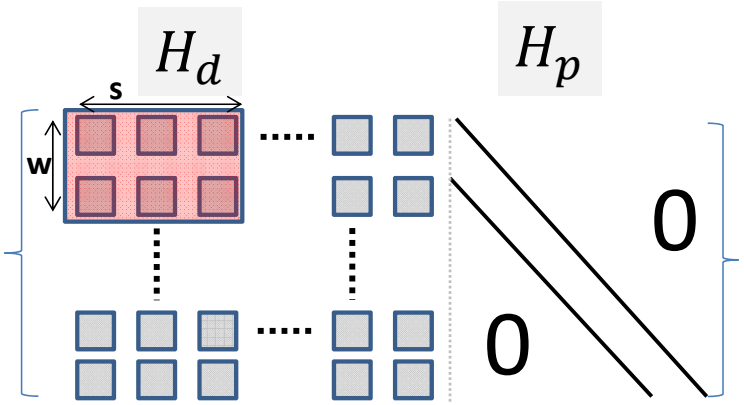


FIGURE 3.4 : Fenêtre de traitement de taille $s \times w$ pour un codeur partiellement parallèle

sont le support d'un large choix de codes QC-LDPC d'une part, et d'autre part, la capacité de choisir le niveau de compromis entre performances (au sens vitesse de traitement, c'est-à-dire, débit de transmission) et le coût matériel (ressources matérielles nécessaires).

L'architecture proposée est composée de quatre parties principales, comme le montre la figure 3.5 :

- une fenêtre de calcul (élément central du codeur) ;
- une unité de contrôle ;
- une mémoire temporaire de travail (de type registre à décalage FIFO) ;
- une mémoire générale (adressable).

Chacune de ces parties est détaillée dans les sections qui suivent.

Fenêtre de calcul

Le choix, par paramétrage, du niveau de parallélisme de l'architecture, permet de contrôler compromis entre performances et coût matériel.

Des quatre parties constituant le codeur, la fenêtre de calcul est celle où sont effectués en pratique l'essentiel des calculs et celle qui conditionne la quantité de calculs pouvant être réalisés simultanément, c'est-à-dire, qui détermine le degré de parallélisme de traitement. À ce titre, elle constitue l'élément central de l'architecture.

Le rôle principal de la fenêtre de calcul est de réaliser, dans une structure de traitement parallèle, les calculs relatifs à une sous matrice de H_d définie selon l'approche ébauchée

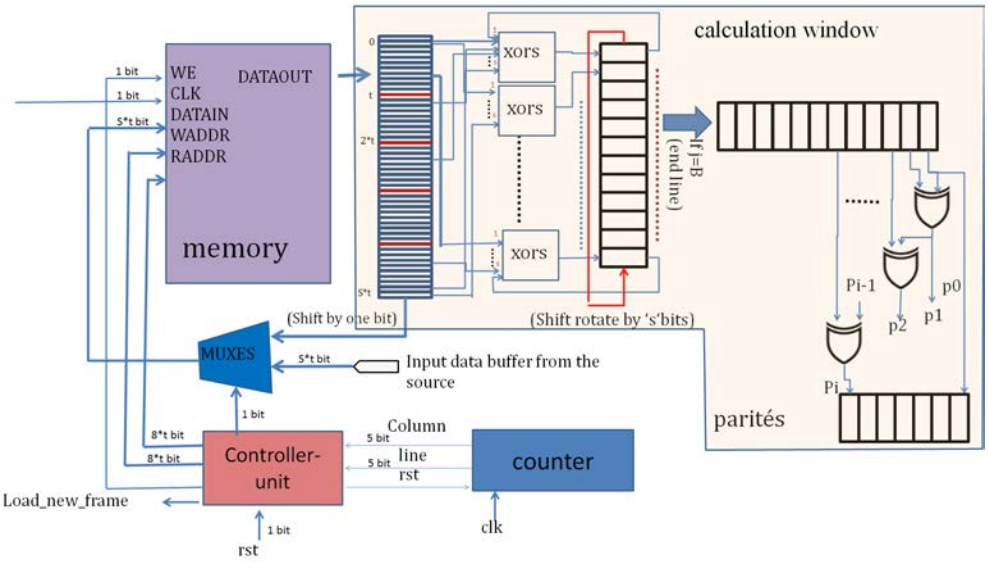


FIGURE 3.5 : Architecture du codeur QC-LDPC

dans le paragraphe 3.4 et la figure 3.4. Les dimensions de cette sous-matrice déterminent le niveau de parallélisme disponible. Comme indiqué précédemment, les dimensions $s \times w$ sont sélectionnées de manière à ce que la sous-matrice puisse recouvrir strictement un nombre entier de sous-matrices $A_{i,j}$, respectivement s à l'horisontale et w à la verticale.

Les deux dimensions ne sont pas équivalentes dans la recherche d'un compromis efficace entre performances et coût. En effet, l'augmentation de la dimension horizontale s a pour conséquence directe l'allongement du chemin critique des opérations réalisées, et donc une limitation potentielle importante de la fréquence maximale de fonctionnement du codeur. Dans ces conditions, le bénéfice potentiel du parallélisme de traitement est fortement compromis, voire même totalement anéanti, par la chute le la fréquence de fonctionnement. Ceci peut être facilement constaté à partir des équations 3.5 et 3.6. Une autre conséquence de l'augmentation de s est l'accroissement inévitable de la largeur du bus d'accès à la mémoire générale (cf. figure 3.5, ce qui constitue vite un handicap important.

L'accroissement du parallélisme de traitement par augmentation de la dimension verticale w s'avère bien plus favorable. En effet, dans la multiplication entre la matrice H_d et la matrice d, les lignes de H_d peuvent être traitées de manière totalement indépendante, les chemins de données étant naturellement disjoints (voir zones colorées en rouge dans l'équation 3.3). Par conséquent, le chemin critique ne dépendant pas de w, l'aug-

mentation du parallélisme de traitement conduit effectivement à une augmentation des performances, c'est-à-dire, à une augmentation du débit de données traversant le codeur.

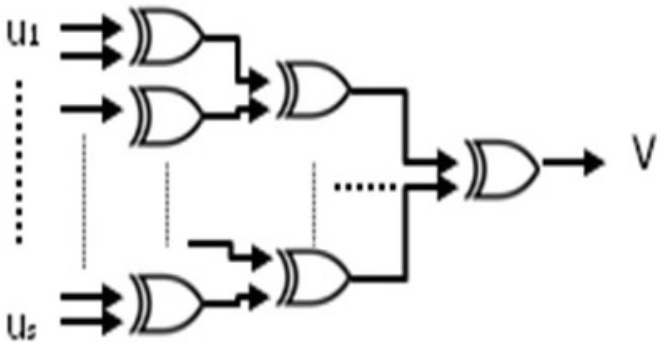


FIGURE 3.6 : Structure en blocs de "ou" exclusifs (XOR)

La fenêtre de calcul est un bloc combinatoire, structuré sous forme de réseaux arborescents (arbres) de portes "XOR", connectés à la mémoire temporaire de travail (de type registre à décalage). Le rôle de cette fenêtre est de calculer, conformément aux équations 3.5 et 3.6, les parités correspondant aux données présentes dans la mémoire temporaire de travail (il s'agit donc de calculs partiels des parités, limité par les dimensions de la fenêtre). La structure arborescente du bloc induit, pour chaque arbre, une croissance logarithmique du chemin critique avec le nombre d'entrées respectifs (nombre de couches de portes "XOR" traversées). Ce dernier nombre croît proportionnellement avec la dimension s de la fenêtre.

Unité de contrôle

Cette unité, qui constitue le seul organe de contrôle (au sens commande) de l'architecture, pilote l'enchaînement des opérations de l'algorithme de codage. Elle est définie en fonction de la sous-matrice $A_{0,0}$, du rendement du code et des dimensions $w \times s$ de la fenêtre de calcul, elle est construite sous forme de machine d'états finie à cinq états (cf. figure e 3.7). Dans la pratique, elle contrôle l'enchaînement des calculs de l'algorithme de codage en pilotant les mises à jours de la mémoire temporaire de travail et les accès à la mémoire générale.

Les états du graphe de contrôle sont :

- un état S_0 de réinitialisation ;

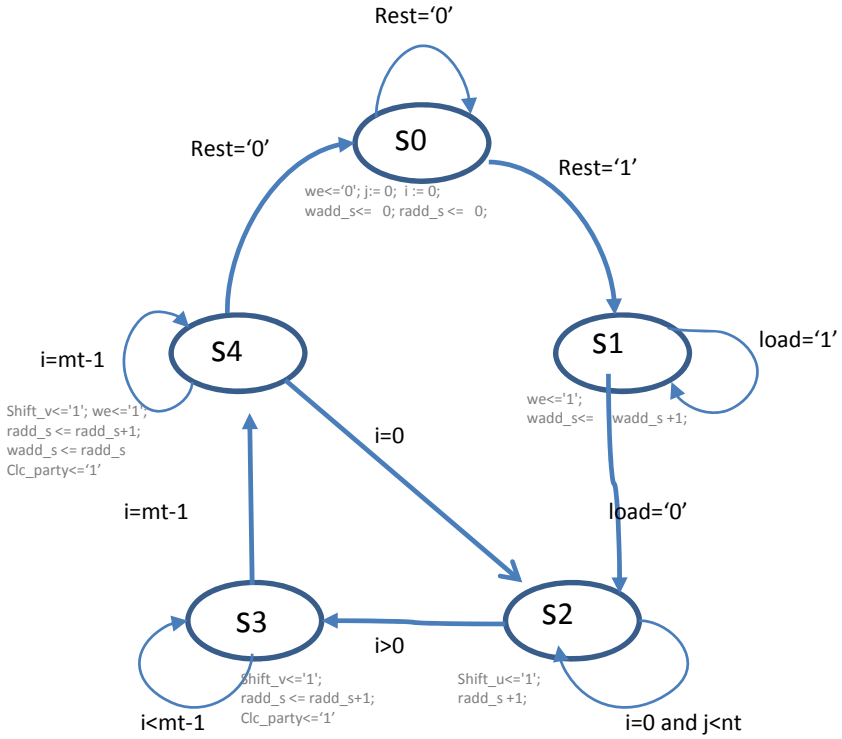


FIGURE 3.7 : Graphe d'états de l'unité de contrôle

- un état S_1 de chargement en mémoire de travail d'une nouvelle tranche des données constituant le message à coder ;

- des états S_2 à S_4 chargés de piloter les calculs du codage proprement dit ; ces états sont différenciés afin de pouvoir traiter de manière différenciées les calculs en fonction de leur positionnement dans la matrice H_d.

Mémoires

Les différentes données, à savoir, les message (blocs) de données à coder, les messages codés (mots codes) et les données temporaires intermédiaires des calculs en cours sont conservées dans des zones de mémorisation de deux types : la mémoire temporaire de travail et la mémoire générale.

La principale raison du découpage en deux type de mémoire tient à la nature des opérations réalisées dans la fenêtre de calcul. À chaque cycle d'horloge, un nombre important de données doit être traité par les structures arborescentes de portes "XOR" de la fenêtre de traitement. Ces données doivent donc être fournies en parallèle en entrée de la fenêtre

de traitement. Ceci est difficilement conciliable avec une organisation de type RAM, forcément limitée en débit parallèle par la largeur de son bus de données, qui ne peut grandir démesurément. Un autre type de structure est nécessaire. Heureusement, la principale raison d'être d'une RAM (*Random Access Memory*), qui est de pouvoir accéder aux données quelle contient dans un ordre quelconque n'est nullement requise ici. En effet, l'accès aux données (tranches successives du message d'entrée) se fait selon un ordre prédéterminé séquentiel. Au sein de tranche, de simple décalages cycliques sont requis pour le calculs partiels de parités. De ce fait, une structure de type registre à décalage (avec contrôle de rebouclage) est la plus adaptée.

En pratique, la mémoire de travail temporaire est composée de deux blocs de mémoire de même type (registre à décalage) et même taille. L'un des bloc contient les données en cours de traitement et est directement connecté à l'unité "fenêtre de calcul". Pendant que ce bloc est en cours d'utilisation par la fenêtre de calcul actuelle, l'autre bloc est progressivement chargé, à partir de la mémoire générale, avec les données de la prochaine fenêtre de calcul. Ce deuxième bloc est donc une sorte de cache de préchargement pour l'unité "fenêtre de calcul".

Grâce à ce dispositif, il est possible de limiter la largeur du bus de données de la mémoire générale, de type RAM, d'où proviennent les données du message à coder et où sont renvoyées au final celles du message codé. Afin de permettre ces deux type de transferts d'avoir lieu simultanément, la mémoire générale est une mémoire RAM à double-port (un port lecture et un port écriture).

La mémoire de type de RAM est indispensable pour les codes de grandes taille telles que ceux employés dans les normes DVB-S2 et DVB-T2. Avec des codes plus petits, comme dans IEEE 802.15.3c [BSL$^+$11], IEEE 802.11ad [PCPY10] ou IEEE 802.11ac [PS13], il éventuellement possible de s'en passer, en conservant l'intégralité du message à coder dans un très grand registre à décalage. Cette approche est limitée par la taille pratique admissible pour un tel registre de par son coût important en ressources FPGA consommées.

3.5 Implémentation sur FPGA

Afin de valider l'architecture proposée, une description VHDL paramétrable a été réalisée au niveau RTL (*Register Transfer Level*) et synthétisée pour différents codes issus des normes DVB-S2, DVB-T2, WiMAX et WiFi. La synthèse a été réalisée avec Altera Quartus II et en ciblant un FPGA de type Altera Statix III.

Les codes ciblés proviennent de normes DVB (*Digital Video Broacasting*) de seconde génération en diffusion satellite ou terrestre. Il s'agit d'une part, d'un code de longueur de bloc 64000 bits et de rendement $5/6$ (DVB-S2), et d'autre part, d'un code de longueur de bloc 16200 bits et de rendement $7/9$ (DVB-T2). Différentes tailles w ont été utilisées pour la fenêtre de calcul, conduisant à des performances et coûts matériels différents. Les résultats sont répertoriés dans les tableaux 3.1 et 3.2, ainsi que dans la figure 3.8.

TABLE 3.1 : Résultats de synthèse pour le codeur DVB-S2

Fréquence maximale (MHz)	Surface totale	Registres	Logique combinatoire	Débit (Mbits/s)	Débit/Surface (Mbits/elem.log.)
328.19	2128	1001	1127	984.57	0.46
289.7	4685	2525	2160	2897	0.61
251.45	8188	4427	3761	6034.8	0.73
250.8	18195	8329	9866	19347	1.06
160.75	32734	14697	18037	28935	0.88

TABLE 3.2 : Résultats de synthèse pour le DVB-T2

Fréquence maximale (Mhz)	Surface totale	Registres	Logique combinatoire	Débit (Mbits/s)
376	238	162	76	105.5
375.9	618	476	142	1057
325	2117	977	1120	3140
317.7	3715	1714	9866	2001.35
289.4	8189	3761	4428	19534

Le paramètre w affecte directement le nombre total d'éléments logiques utilisés. Le coût matériel ainsi que les performances sont d'autant plus grands que w est grand. Par exemple, avec le code issu de DVB-S2, la consommation matérielle en éléments logiques n'est que de 2128 pour $w = 30$, avec un débit atteint de 985 Mbits/s (pour un fréquence d'horloge de 328 Mhz). Un débit très important de 28.9 Gbits/s (horloge de 160.75 MHz)

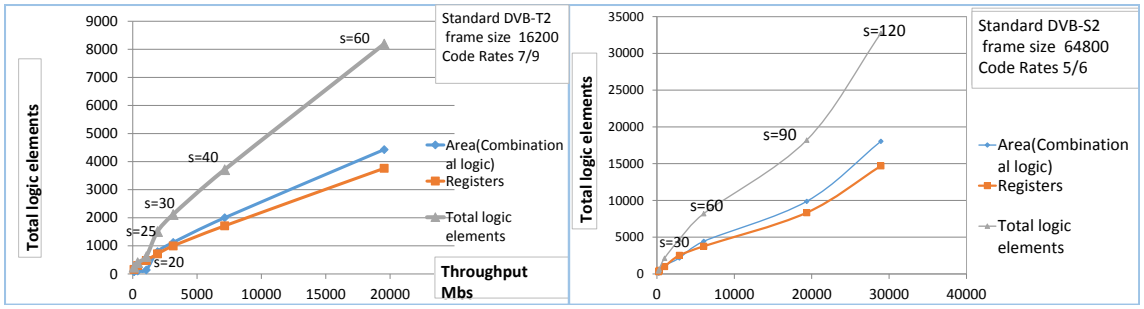


FIGURE 3.8 : Présentation graphique des résultats pour DVB-S2 et DVB-T2

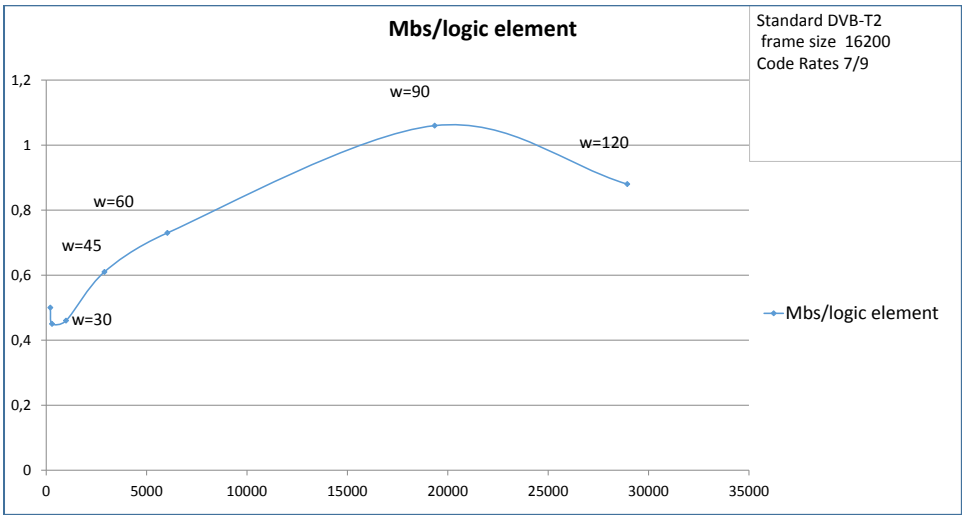


FIGURE 3.9 : Compromis débit/surface en fonction de w

est atteint pour $w = 120$, mais avec un coût matériel très élevé de 18037 éléments logiques.

Le graphe de la figure 3.9 permet d'observer le profil d'efficacité, mesurée en termes de rapport de performance/coût, en fonction de la valeur de $w(\in \{30, 45, 60, 90, 120\})$, le paramètre de dimensionnement de la fenêtre de calcul.

3.5.1 Comparaison avec d'autres codeurs

Le tableau 3.3 présente quelques résultats comparatifs pour des codeurs issus de l'état de l'art. Les codes sources de ces codeurs n'étant pas disponibles, ni les détails précis de ces architectures, il n'a pas été possible de les synthétiser pour les mêmes cibles technologiques (même familles de FPGAs) avec les mêmes outils de synthèse. De ce fait, la comparaison n'est qu'indicative.

TABLE 3.3 : Comparaison de quelques implémentations FPGA de codeurs

	[LLWJ04]	[EC01]	[YC09]	Codeur proposé		
Taille bloc (en bits), Rendement	Variable	2304, 1/2 2304, 3/4 576, 5/6	Variable	Variable	2304, 1/2	1152, 1/2
Coût (nb éléments logiques)	-	10 339 12 727 4 295	11 430	8 189 32 734	1724	912
Taille Memoire	RAM 19 blocs	-	3, 911, 680	Variable (2 blocs)	4608	2304
Fréquence (MHz)	(de 80 à 170)	(de 140 à 195)	60	(de 160 à 320)	291	320
Technologie FPGA	Virtex-II	Stratix	Stratix	Stratix-II	Stratix	Stratix
	44M @2000 1/2 33M@2000, 2/3	3 350 5 170 6 280	239M @ 2304, 3/4 357M @ 2304, 5/6	19,534M@ 162000, 7/9 28,935M @ 64800, 5/6	32,832M	17,786M

3.6 Bilan

Dans ce chapitre, nous avons proposé une architecture modulaire paramétrable pour une classe de codeur LDPC particulière, largement utilisée dans le domaine des communications : les codeurs QC-LDPC à matrice H de structure bi-diagonale.

L'architecture proposée, est paramétrable permettant de sélectionner lors de l'implémentation, la taille et le rendement du code, et son taux de traitement parallèle. Ainsi, différents codes QC-LDPC peuvent être supportés avec la capacité de choisir le niveau de compromis entre performances (débit de transmission) et le coût matériel (ressources matérielles nécessaires). L'architecture repose sur une structure appelée "fenêtre de traitement". Permettez-nous de contrôler le niveau de parallélisme par les paramètres configurables de cette architecture qui concernent les caractéristiques du code ciblé et le degré de traitement parallèle souhaité.

le traitement parallèle partie plus ou moins large (paramétrage) de la matrice de contrôle H.

L'architecture a été validée, en synthétisant à l'aide de l'outil Altera Quartus II, des codeurs sur FPGA de la famille Altera Stratix III, pour des codes provenant des normes DVB-S2 et DVB-T2. Des débits allant d'environ jusqu'à près de 29 Gbits/s et 19.5 Gbits/s ont pu être atteints sur cette architecture flexible pour respectivement DVB-S2 et DVB-T2.

DÉCODEUR QC-LDPC

Chapitre 4

Décodeur QC-LDPC

Les algorithmes du type “propagation de message” (*message passing*), tels que ceux présentés dans la section 2.3, sont intrinsèquement parallèles, car les opérations de traitement de lignes sont totalement indépendantes d’une ligne à l’autre. La réciproque est également vraie pour les opérations de traitement de colonne. Ces caractéristiques sont importantes car elle vont permettre de définir des architectures dotés de fortes capacités de traitement parallèle pour exploiter ce parallélisme intrinsèque, et d’ainsi offrir des performances élevées (débits élevés).

Des décodeurs très fortement parallèles, capables de traiter des mots de codes en peu de cycles ont été proposés pour des codes spécifiques fixes, donc de taille et rendement prédéfinis [MB10, CYL+14]. Cependant, ces décodeurs affichent des coût matériels importants (quantités de ressources matérielles utilisées élevées) afin d’offrir les taux de traitement parallèle requis pour ces forts débits. Afin de limiter ces coûts quand les performances requises sont moindres, des décodeurs parallèles partiels ont étés proposés [WC07, UPH+08, FFF07].

L’architecture que nous proposons dans ce chapitre offre un décodeur paramétrable partiellement parallèle, permettant par paramétrage lors de l’implémantation, le choix du code QC-LDPC (taille du code et rendement de codage).

4.1 Algorithme de décodage pour LDPC et le flux de données

Plusieurs algorithmes ont été proposés pour le décodage des codes LDPC. La plupart d'entre eux dérivant de l'algorithme de propagation de croyance [Gal62] (BP – *belief propagation*), un algorithme de type itératif basé sur le principe de propage de message (*message passing*). Ces algorithmes appliquent de manière itérative et séquentielle les deux opérations que sont le traitement de ligne et le traitement de colonne. Pendant le traitement de ligne, tous les nœuds de contrôle reçoivent des messages des nœuds de variables adjacents, avant d'effectuer les opérations de contrôle de parité, puis de renvoyer les résultats aux nœuds variables adjacents. Pendant le traitement de colonne, les nœuds de variable mettent à jour les informations souples associées aux bits décodés en utilisant les informations reçues des nœuds de contrôle, puis renvoyent les mises à jour aux nœuds de contrôle. Ce processus se répète itérativement jusqu'à ce que la condition d'arrêt soit

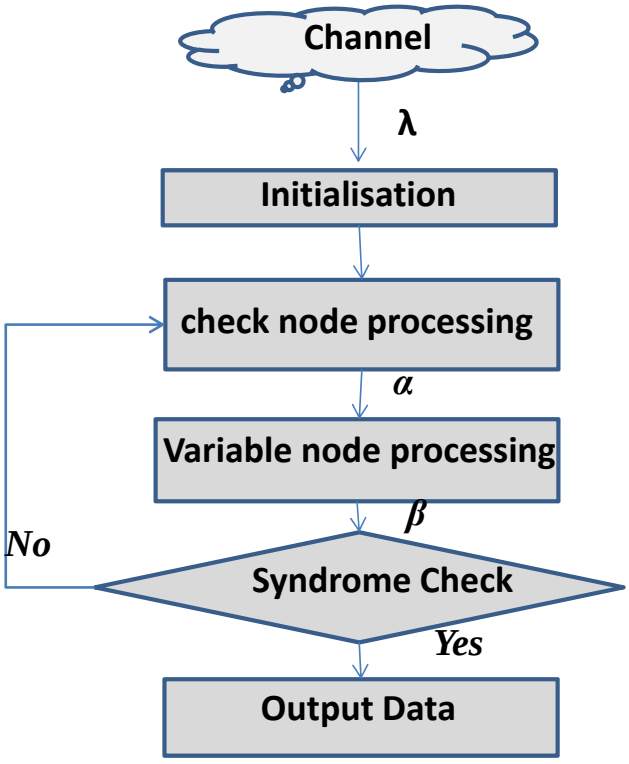


FIGURE 4.1 : Organigramme type des algorithmes de décodage par passage de messages

Le principe de base de l'algorithme TPMP est illustré dans la figure 4.1. Après que les informations éventuellement corrompues aient été reçues depuis un canal AWGNλ, le décodage peut commencer. Pour commencer, toutes les entrées de contrôle nœuds sont initialisés à 0. Ensuite, le processus itératif commence où alternent les deux phases :

- une phase de mise à jour des nœud de contrôle (traitement de ligne) pendant laquelle les messages α sont générés et envoyés vers les nœeud de variable ;
- un phase de mise à jour des nœud de variable (traitement de colonne), suite à la réception des message α, pendant laquelle sont générés les messages β, renvoyées à leur tour vers les nœuds de contrôle.

Ce processus itératif en deux étapes se répète jusqu'à ce que toutes les erreurs sont corrigées ou qu'un nombre limite d'itérations soit atteint.

Les algorithmes Sum-Produit [Mac99] et Min-Sum [FMI99] sont largement utilisés pour le décodage. Dans notre architecture, nous utilisons l'algorithme MinSum (MS), décrit en détail qui est décrit dans la section 2.3.

4.2 Architecture du décodeur QC-LDPC

Le domaine des communications numériques évolue rapidement et les exigences augmente Nous devons donc travailler en parallèle dans ce domaine, il existe de nombreuses façons construite une architecture en parallèle, et chacun des aspects positifs et négatifs.

4.2.1 Décodeurs totalement parallèles

Dans un décodeur qualifié de "totalement" parallèle, chaque ligne et chaque colonne de la matrice de contrôle de parité H est connectée à une unité de traitement différents, tous les unités fonctionnant en parallèle [BH02, DCK08, MB07, MTB09]. Tous les nœuds de contrôle CN et les nœuds de variables VN ainsi que leurs ($\times W_r + N \times W_c$) interconnections sont mis en œuvre, où W_c est le poids de la colonne et W_r est le poids de la ligne (nombres de 1 dans une ligne ou colonne, respectivement). Pour ce décodeur totalement parallèle, $M + N$ éléments de traitement seront nécessaires pour l'ensemble des nœuds de contrôle et des nœuds de variable, ainsi que $M \times W_r + N \times W_c$ interconnexions.

Pour un décodeur LDPC 10GBASE-T construit selon ce schéma totalement parallèle, 2432 éléments de traitement seront nécessaires inter-connectés par $24576 \times b$ connexions (et leurs répéteurs, le cas échéant), où b est le nombre de bits sur chaque sur chaque connexion (données échangées en format virgule-fixe). La précision choisie pour le format virgule-fixe est un un paramètre déterminant dans le coût matériel du codeur, qu'il convient d'optimiser.

En général, les décodeurs totalement parallèles occupent plus de surface et affichent des fréquences de fonctionnement plus faibles , Parce que de nombreuses opérations doivent être effectuées en parallèle dans le même temps. Ils ne requièrent en général qu'un seul cycle par itération de message-passing et ils sont par ailleurs, intrinsèquement plus économes en énergie [DCK08].

4.2.2 Décodeur partiellement parallèle

Comme indiqué précédemment, les algorithmes par passage de message sont essentiellement parallèles car les opérations de traitement de ligne sont totalement indépendantes les unes par rapport aux autres, et réciproquement pour les opérations de traitement de colonne.

L'inconvénient majeur des architectures totalement parallèles est leur très fort besoins en ressources matérielles. Elles offrent en contrepartie des performances très élevées. D'autre part, la plupart des architectures totalement parallèles proposées dans la littérature sont conçues pour un code LDPC en particulier (donc, pour une longueur et un rendement précis et une matrice H en particulier).

Pour limiter les besoins en ressources matérielles, plusieurs architectures de décodeurs partiels parallèles ont été proposées dans la littérature [WC07] [UPH$^+$08] [FFF07].

Nous proposons dans la suite de ce chapitre une architecture partiellement parallèle de décodage, basée, pour le traitement de la matrice de contrôle M, sur l'utilisation du principe d'une fenêtre de calcul de dimension $w \times s$, comme illustré dans la figure 4.2.

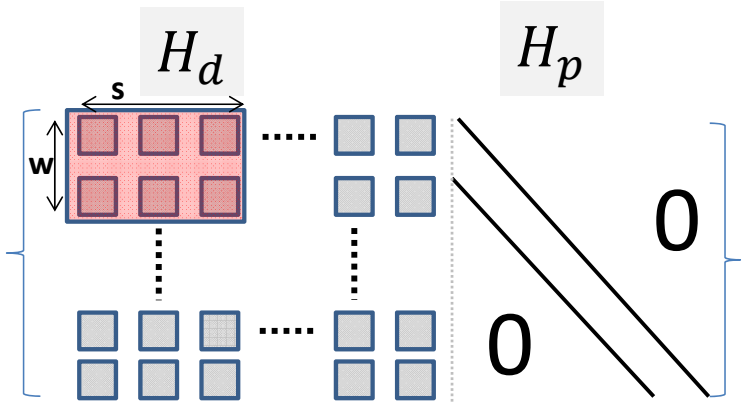


FIGURE 4.2 : Principe de fenêtre de calcul pour le décodage partiellement parallèle

4.3 Architecture d'un décodeur LDPC flexible

4.3.1 Vue générale sur l'architecture du décodeur LDPC proposée

Nous proposons ici une architecture paramétrable de décodeur QC-LDPC, les paramètres étant : la taille de bloc de code et le rendement de code R du code, la taille d'une fenêtre de calculs, w.

L'architecture du décodeur est constituée des deux parties principales dédiées respectivement au traitement des deux étapes de base de l'algorithme itératif : le traitement des nœuds de contrôle et le traitement des nœuds de variables (cf. figure 4.3).

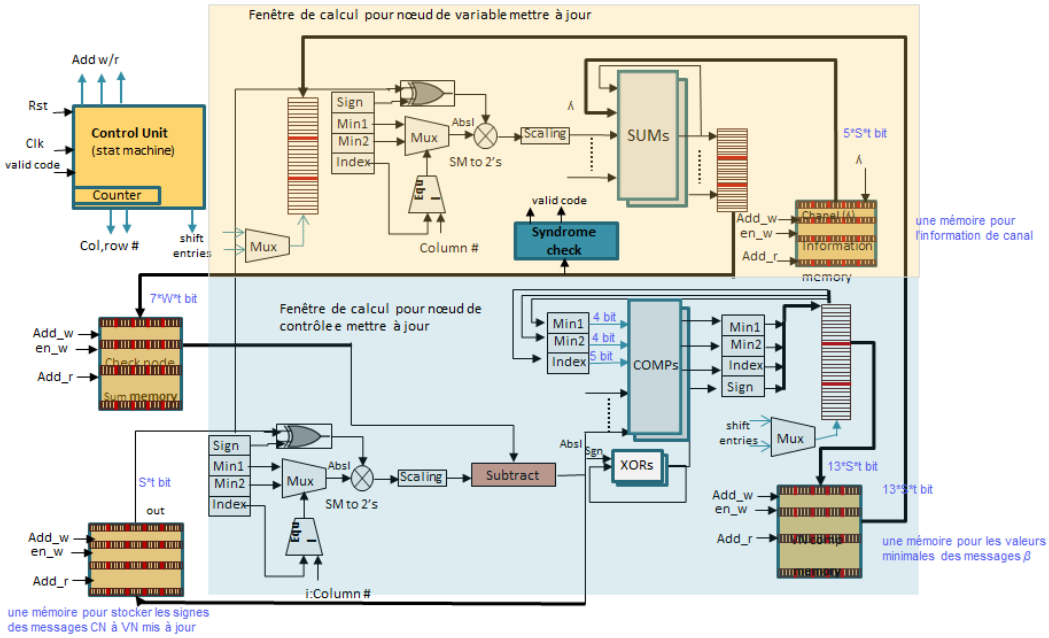


FIGURE 4.3 : Synoptique général du décodeur QC-LDPC

Pour la réalisation des deux parties, différents blocs sont nécessaires :

- deux fenêtres de calcul, chaque partie du décodeur possédant sa fenêtre de calcul ;
- une unité de contrôle ;
- des mémoires ;
- une unité de vérification.

Fenêtre de calcul de la partie traitement des nœuds de variable

Cette fenêtre de calcul est dédiée aux traitements liés aux nœuds de variable. Les dimensions de la fenêtre sont s et w respectivement à l'horizontale et à la verticale, correspondant aux nombre $s \times w$ de sous-matrices $i_{i,j}$ couvertes par la fenêtre et traitées simultanément.

Le nombre de fonctions de comparaison (*Comp*) dans la fenêtre est égal à $(T \times w)$, chaque Comp possédant s entrées. La structure d'une fonction *Comp* est représentée dans la figure 4.5). Elles effectuent la comparaison entre s valeurs absolues de messages β_{ij} de nœuds de variables à et les résultats provisoires de la dernière fenêtre traitée. Il faut également calculer les β_{ij} comme indiqué dans l'équation 4.1.

$$\beta_{ij} = Z_j - \alpha_{ij} \tag{4.1}$$

.

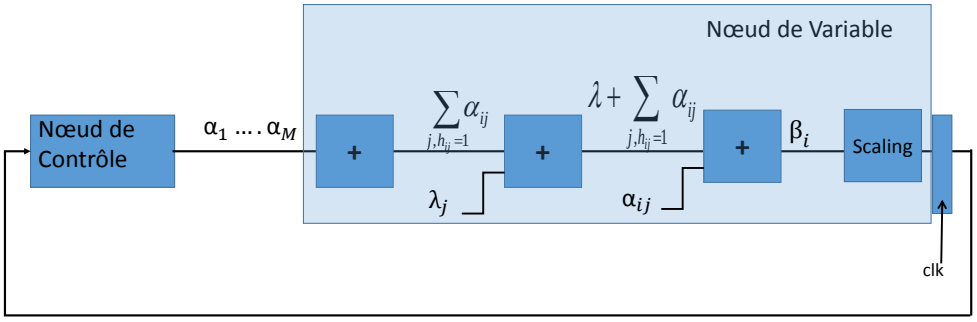


FIGURE 4.4 : Chemin de données pour le calcul de β_{ij}

Fenêtre de calcul de la partie traitement des nœuds de contrôle

Cette fenêtre de calcul est dédiée aux traitements liés aux noeud de contrôle comme représenté dans la figure 4.4. Elle est de même dimensions $(s \times w)$ que l'autre fenêtre de calcul mais utilise, à la place des fonctions *Comp*, des fonctions *SUM* pour le cumul de tous les messages α des nœuds de contrôle vers les nœuds de variables avec des informations sur le canal *AWGN* λ comme indiqué dans l'équation 4.2 et représenté dans la figure 4.6.

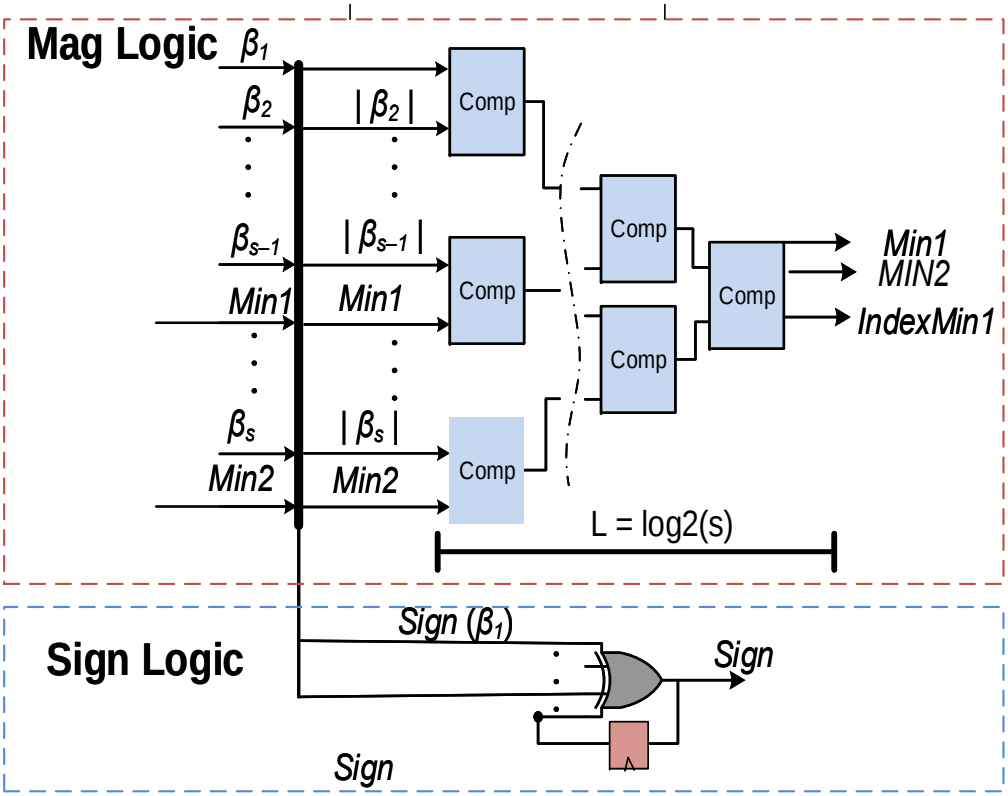


FIGURE 4.5 : Structure d'une fonction *comp*

$$Z_j = \lambda_j + \sum_{j', h_i j = 1} \alpha_{ij'} \tag{4.2}$$

4.3.2 Unité de contrôle

L'unité de contrôle est constituée d'une machine à états finie, et a pour rôle d'assurer l'enchaînement des opérations de l'algorithme de décodage. Elle est définie en fonction de la sous-matrice $I_{0,0}$, du rendement du code et des dimensions $w \times s$ des fenêtres de calcul.

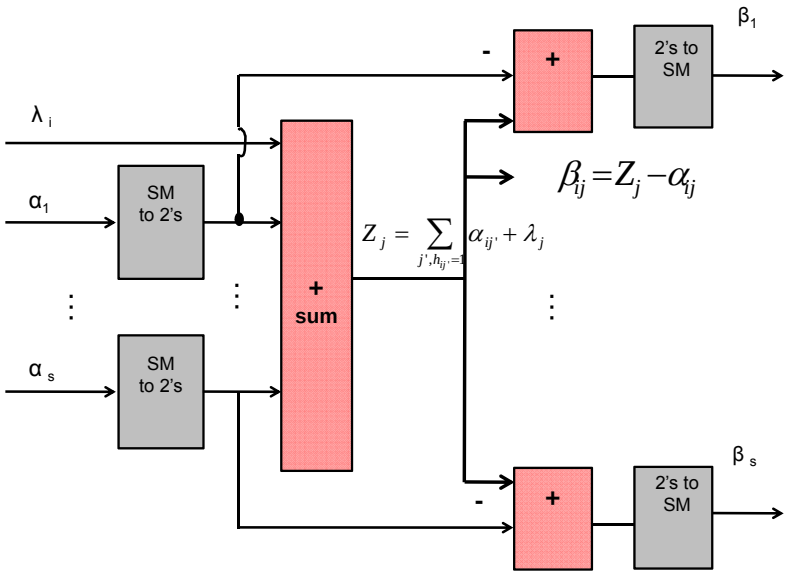


FIGURE 4.6 : Schéma de nœud variable Processeur unité

Elle assure la synchronisation des transferts entre les différents blocs de l'architecture, et la gestion des mémoires. Elle contrôle en outre condition d'arrêt de l'algorithme (fin du décodage).

Mémoires

Chaque fenêtre de calcul est connectée à des registres à décalage selon le même principe présenté pour le codeur. La différence principale est la présence de registres à décalage longs utilisées pour le stockage des résultats temporaires obtenus avec chaque fenêtre et réarrangées avant d'être réutilisés dans le cycle de traitement suivant.

Quatre mémoires de type RAM simple-port, sont utilisées pour le stockage des données restantes

- une mémoire pour l'information de canal ;
- une mémoire pour retenir les derniers messages de sommes générés par l'addition des informations de canal λ_j et les messages α des nœuds de contrôle ;
- une mémoire pour les valeurs minimales des messages β x min1, le deuxième minimum global min2, l'indice du noeud de variable connectés fournir min1 (indice) et le signe globale (sign) pour chaque message des noeud de contrôle C_i en conséquence comme représenté dans la figure4.7 ;

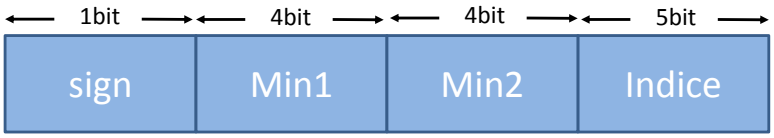


FIGURE 4.7 : Message de sortie de fonction de comparaison

Unité de vérification (Check syndrome unit)

Le décodage du code QC-LDPC est éffectué en utilisant un algorithme itératif. Comme nous l'expliquions dans chapitre 2, le nombre d'itérations peut être soit fixe si l'on considère les caractéristiques du canal connues et fixes pendant toutes la durée de la transmission (avec par conséquent un risque de dégradation de la qualité du décodage, si les caractéristiques du canal ne sont pas stables). À contrario, le nombre d'itérations peut être évalué de façon dynamique afin de s'adapter à des caractéristiques de canal variables dans le temps. Pour cela, une unité de vérification a été introduite pour tester le résultat généré après chaque itération, en multipliant le mot de code, dans sont état actuel de correction, par la matrice H comme indiqué dans l'équation 4.3 , afin de vérifier si toutes les erreurs ont été corrigées.

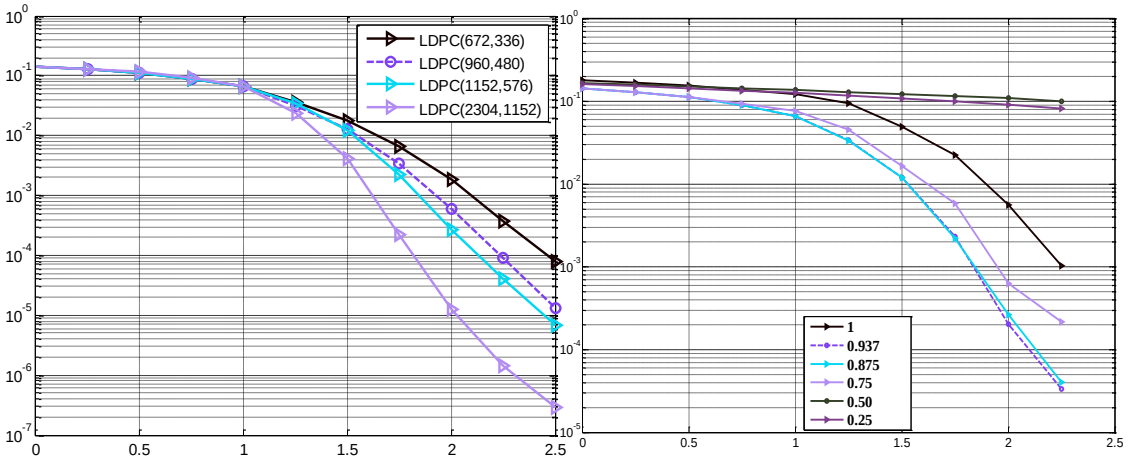
$$H \cdot \hat{c}^T = [0] \tag{4.3}$$

4.4 Simulation et analyse des algorithmes de décodage

Afin d'atteindre un décodage de canal optimal préservant les performances de communication en termes de taux d'erreur, une analyse par simulation précise de C-modèles quantifiées des algorithmes de décodage LDPC est indispensable. L'analyse présentée dans cette section considère les niveaux de LLR d'entrée et de quantification extrinsèque avec facteurs extrinsèques d'échelle ψ afin de d'atteindre la convergence à des niveaux de dégradation de performances acceptables en termes de taux d'erreurs binaire.

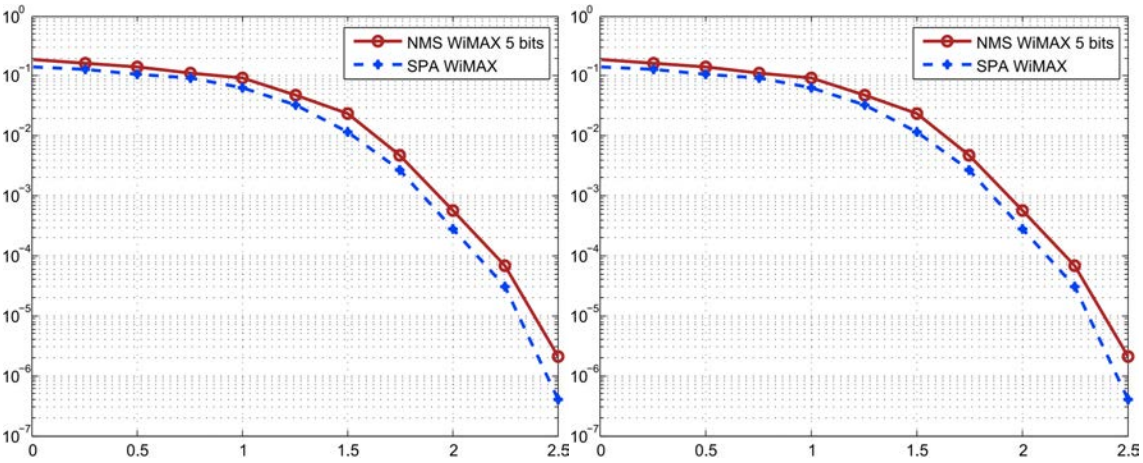
Pour le décodage LDPC, l'algorithme considéré est l'algorithme Min-Sum normalisé présenté dans le chapitre 2. Pour faciliter la présentation, on notera les valeurs de quantification des LLR d'entrée *in* comme par la canal, les LLR extrinsèques *ex* à la sortie de

nœud de variable, et les facteurs d'échelle extrinsèques ψ regroupés (in, ex, ψ).



A-performances de décodage signalé au nombre de représentation bits (5,7) et taille de bloc pour LDPC WiMAX

B-l'analyse du facteur d'échelle pour LDPC WiMAX avec une longueur de bloc 1152 bits ,taux=1/2



C-taux d'erreurs binaires(BER) pour WiMAX (IEEE 802.16e) avec une longueur de bloc 1152 bits ,taux=1/2

D-taux d'erreur de bloc(FER) pour WiMAX (IEEE 802.16e) avec une longueur de bloc 1152 bits ,taux=1/2

FIGURE 4.8 : Performances de décodage sur C-modèles pour qualifier les architectures de décodage

Les figures 4.8.C et 4.8.D montrent les courbes de référence obtenues pour des codes LDPC WiMAX de rendements $1/2$, de taille de blocs de 1152 bits et de niveaux de quantification d'entrée et extrinsèque de $(5, 7)$ respectivement. Le modèle quantifié avec $(7, 5, 0.875)$ présente une dégradation de moins de $0,2dB$ par rapport à la référence en utilisant l'algorithme Sum-produit (SPA) comme le montre la figure 4.8. L'analyse du facteur d'échelle ψ montre que pour le même code LDPC (WiMAX), on peut grandement améliorer le BER. Un facteur d'échelle de $0,875$ est considéré comme apte à réaliser un

compromis satisfaisant entre performances de communication et de mise en œuvre matérielle.

4.5 Implémentation sur FPGA

Afin de valider l'architecture de décodeur proposée,une description VHDL paramétrable (concernent les caractéristiques du code ciblé et le degré de traitement parallèle souhaité) a été réalisée au niveau RTL (*Register Transfer Level*) et synthétisée pour différents codes issus de différentes normes, la synthèse ayant été réalisée avec Altera Quartus II en ciblant un FPGA de type Altera Statix III.

TABLE 4.1 : résultats de synthèse sur Stratix III pour decoder LDPC de l'Architecture decoduer (WiMAX (IEEE 802.16e) avec une longueur de bloc 1152 bits ,taux=1/2)

Fréquence maximale (Mhz)	Surface total (élément logique)	Registres (bit)	logique combinatoire	Taille de fenêtre (sous-matrice)	Débit (Mbs)
168.1	5617	1388	4229	1	72
147.8	9673	2107	7566	2	128.2
137.1	18463	3546	14917	4	235
108	24339	4887	19506	6	277.7

TABLE 4.2 : résultats de synthèse sur Stratix III pour decoder LDPC de l'Architecture(WiMAX (IEEE 802.16e) avec une longueur de bloc 2304 bits ,taux=1/2)

Fréquence maximale (Mhz)	Surface total élément logique	Registres (bit)	logique combinatoire	taille de fenêtre (sous-matrice)	Débit (Mbs)
146	11246	2733	8513	1	130.5
144.2	18743	4172	14542	2	247.2
131.2	37622	7051	30571	4	449.7

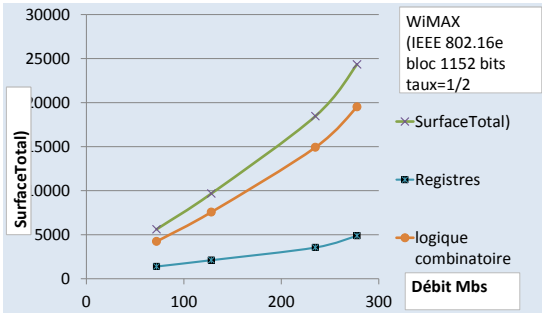
Des décodeurs de degré de parallélisme divers ont été générés pour deux codes de la norme WiMAX (IEEE 802.16e), de longueurs de bloc 1152 et 2304 bits respectivement, et de rendement de codage de $1/2$ (cf. tables 4.1 et 4.2. Des décodeurs additionnels ont éte générés pour une code de la norme WiFi (IEEE 802.11ac), de longueur de bloc 1944 bits et rendement de codage $1/2$. Ces derniers décodeurs ont été comparés (cf. table 4.3) à des implémentations issues de l'état de l'art [DYLD09], de configurations proches et utilisant le même algorithme (Min-Sum). Les performances affichées par notre décodeur sont meilleures : débit deux fois plus élevé, pour une consommation d'éléments logiques

TABLE 4.3 : Comparaison des résultats de synthèse sur FPGA pour décodeur LDPC architecture

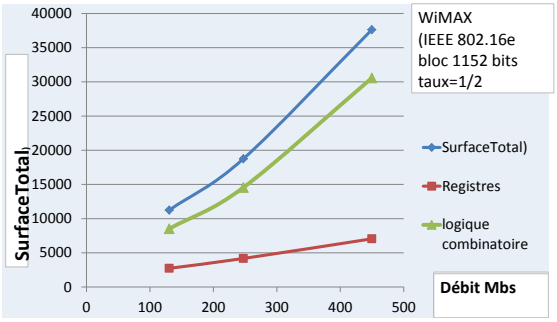
	Proposed encoder		decoder [CA12]	decoder [DYLD09]	Proposed decoder		
Block Length (bits)	2304	1152	1944	2304	1944	1152	2304
Registers	906	473	1604	6598	6106	3546	7051
Area(Slices /ALUTs)	818	394	5908	17,259	21494	14955	30571
4-input LUTs	NA	NA	10816	NA	39797	NA	
Memory bits	4608	2304	56376	271104	36936	21888	43776
FPGA	Stratix-III		Virtex-II	Stratix-II	Virtex-IV	Stratix-III	
Throughput (Mbps)	32832	18528	32	232.5	291	235	449.7

et une quantité deux fois moindre de registres . Le décodeur de [CA12] est mise en œuvre sur une technologie Xilinx Virtex, pour un code WiFi (IEEE 802.11n) de taille de bloc 1944 bits et rendement de codage $1/2$. Nous avons réalisé synthétisé une instance de notre codeur pour le même code sur technologie Xilinx Virtex également. Bien que notre architecture soit plus gourmande en ressource matérielles, elle offre un débit bien plus élevé (que n'explique pas la différence de génération technologique de FPGA).

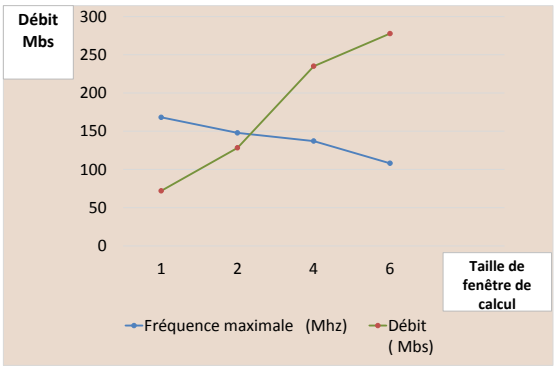
Comme pour le chapitre consacré au codeur, les résultats obtenus avec notre architecture sont également présentés sous forme graphique pour une meilleure interprétation (cf. figure 4.9). L'efficacité relative à la consommation de mémoire provient de la méthode de calcul de la variable mise à jour du nœud et vérifier la mise à jour des massages de noeuds lorsque tous les massages sont stockées dans la mémoire que le bruissement de la fonction COMP (min1, min2, index, signal) pour les très rangée et les sommations globaux pour chaque colonne. Le débit du décodeur proposé est facilement évolutif en



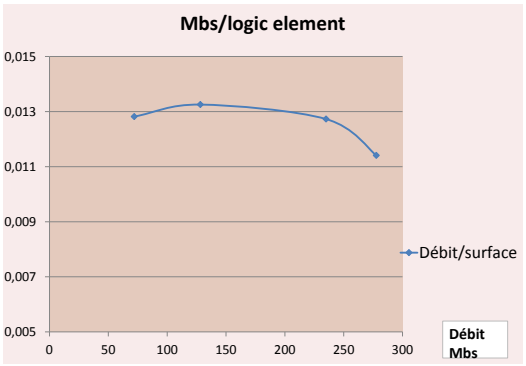
A. Surface de l'Architecture(WiMAX (IEEE 802.16e) avec une longueur de bloc 2304 bits ,taux=1/2.



A. Surface de l'Architecture(WiMAX (IEEE 802.16e) avec une longueur de bloc 1152 bits ,taux=1/2.



C. Fréquence maximale et le Débit pour WiMAX (IEEE 802.16e) avec une longueur de bloc 1152 bits ,taux=1/2.



D. compromis Débit/surface pour decoder pour WiMAX (IEEE 802.16e) avec une longueur de bloc 1152 bits ,taux=1/2.

FIGURE 4.9 : résultats pour un code WiMAX de longueur 115 et 22304)bits rendement de codage taux=1/2

augmentant le facteur de parallélisme ou de modifier le taux de code par exemple le premier décodeur LDPC WiMAX est capable d'atteindre 732 Mbits/s (à une fréquence de fonctionnement de 122 MHz) pour un rendement de codage de $5/6$.

4.6 Conclusion

Dans ce chapitre, nous avons proposé une architecture de décodeur paramétrable partiellement parallèle, permettant par paramétrage lors de l'implémentation. Les architectures proposées ont été validées sur plusieurs codes QC-LDPC. À chaque fois, différents

degrés de traitement parallèle ont été sélectionnées, permettant de varier le compromis entre performances (vitesse de traitement, donc débit) et coût (surface matérielle requise) le choix du code QC-LDPC (taille du code et rendement de codage), ainsi que le degré de traitement parallèle qui conditionne les performances (débit) et le coût matériel (ressources consommées) du décodeur. L'architecture repose sur des une structures appelées "fenêtres de traitement", permettant le traitement parallèle sur un partie plus ou moins large (paramétrage) de la matrice de contrôle H.

Une rapide analyse de simulation des algorithmes fonctionnels de décodage a permis de sélectionner les quantifications à appliquer pour les implémentations, puis, comme pour le codeur, l'architecture du décodeur a été validée, en ciblant l'implémentation sur des FPGA des familles Altera Stratix III et Xilinx Virtex IV.

CONCLUSION GÉNÉRALE

Conclusion générale

Ce travail présenté dans cette thèse concerne le développement de nouvelles architectures de codage et de décodage pour des codes correcteurs d'erreur linéaires en blocs de la famille QC-LDPC (*Quais-Cyclic Low-Density Parity-Codes*). L'objectif visé pour ces architectures est d'offrir des niveaux élevés de flexibilité et de performances. La flexibilité est obtenue par l'intermédiaire d'architectures paramétrables, permettant de choisir parmi une grande diversité de codes au sein d'une classe de codes QC-LDPC (taille et rendement du code) et le niveau de traitement parallèle qui conditionne les performances (vitesse de codage/décodage) et le coût (ressources matérielles requises). Il est ainsi plus facile de s'adapter aux nombreux codes des normes de communication actuelles et futures.

La flexibilité proposée ne doit pas l'être au détriment des performances. Pour cela, les architectures de codage et de décodage proposées reposent sur une exploitant au mieux les possibilité intrinsèques de parallélisme des algorithmes de codage et décodage. Les solutions proposées sont basées sur l'emploi de "fenêtres de calcul" permettant de réaliser par traitement parallèle, les calculs au niveau d'une sous-matrice de la matrice de contrôle de parité du code.

Après une premier chapitre introduisant, d'une part la structure d'une chaîne de transmission et d'autre part, les principes du codage canal et du codage correcteur d'erreur qui sert à le réaliser, le chapitre deux s'est penché de manière plus approfondie sur les codes LDPC et la sous-famille QC-LDPC ciblée par les architectures proposées dans les deux chapitres suivants.

Dans le chapitre trois, consacré au codage, a été proposée une architecture paramétrable performante, pour une classe de codes QC-LDPC, ayant pour caractéristique une matrice H bi-diagonale de forme particulière et sur une structure de traitement basée sur le principe de la "fenêtre de calcul". Suite à la présentation des principes mis en œuvre

dans l'architecture de codage, des implémentations sur FPGA on été proposées pour des codes issus des normes DVB-S2 et DVB-T2. L'architecture, dont le niveau de flexibilité est avéré, permet d'obtenir des niveau de performance élevé, des débit jusqu'à 29 Gbits/s ayant été atteint pour DVB-S2 avec un rendement de codage de $5/6$.

Dans le chapitre quatre, c'est une architecture paramétrable performante de décodeur que a été proposée, pour la même classe de codes QC-LDPC qu'au chapitre trois. La encore, le principe de l'architecture repose sur une explotation du principe de "fenêtre de calcul" appliqués aux algorithmes de décodage Min-Sum et Min-Sum normalisés, qui sont des algorithmes de type *message passing*. Des implémentations sur FPGA ont également été proposée pour des codes issus des normes de communications sans fils (WiMAX et WiFi), permettant à nouveau d'obtenir de bon compromis performance/coût sous les contraintes de flexibilité qui permettent de choisir parmi une grande diversité de codes (taille, rendement) et de niveaux de traitement parallèle, des débits élevés, jusqu'à 732 Mbits/s, ayant pu être atteints.

Les perspectives de développement des architectures de codage et décodage, peuvent être considérée selon plusieurs axes.

- Un premier axe consisterait à renforcer la flexibilité des architectures proposées. Le type de paramétrage disponible avec les architectures présentées dans ce travail est un paramétrage statique, accessible uniquement avant synthèse. Ce n'est pas un paramétrage dynamique, pouvant être effectué en cours de fonctionnement. Les obstacles à franchir sur cette voie concernent principalement les très pénalisantes augmentation du coût matériel et diminution des performances.

- Un autre axe consisterait à étendre les architectures à d'autres familles de codes QC-LDPC. Cette voie conduit très certainement vers une complexité accrue des architectures.

- Un autre axe à plus long terme, consisterai à favoriser un meilleur compromis vitesse/coût en cherchant une approche plus modulaire du principe de "fenêtre de calcul" : des fenêtres plus nombreuses mais plus petites, et potentiellement plus rapides (grâce à des chemins critiques plus courts). Les obstacles sont nombreux sur

cette voie, à commencer par la complexité très fortement accrue de l'unité (ou des unités) de contrôle pilotant les calculs dans ces architecture

section*Perspectives Cette conception peut avoir besoin de plus du développement, concernant cet objectif plusieurs idées peuvent être étudie :

- augmentation de la flexibilité pour supporter reconfiguration dynamique, cela nous donnera plus de la facilité de changement de norme à la autres norme ou changer la paramètre de code sans resynthèse, cette chose peut être appliquée pour décoder et le codeur.

- dans l'architecture décoder présentée, on peut utilise une seule fenêtre de calcule pour α et β et les deux fonction (COMP,SUM),ils seront devenir un seule fonction,pour réduire les coûts matériels et l'accès des mémoires.

- le méthode de partitions les lignes de matrice(Multi-Split)en plusieurs blocs peut permettre de réduire chemin critique de la décodeur.

- pour les systèmes de télécommunication qui exige un débit élevée, la implémentation sur Virtex-7 FPGA capable de générer performances suffisamment élevée pour des applications de future.

- Explorer d'autres technologies comme CMOS qui peut permettre de nouvelles améliorations en vitesse et d'efficacité architecture.

Bibliographie

[ACC+93] J. Arlat, A. Costes, Y. Crouzet, J. C Laprie, and D. Powell. Fault injection and dependability evaluation of fault-tolerant systems. *IEEE Transactions on Computers*, page 913–923, 1993.

[Ben99] Sergio Benedetto. *Principles of digital transmission : with wireless applications*. Springer, 1999.

[BG96] Claude Berrou and Alain Glavieux. Near optimum error correcting coding and decoding : Turbo-codes. *Communications, IEEE Transactions on*, 44(10) :1261–1271, 1996.

[BG03] Claude Berrou and Alain Glavieux. Turbo codes. *Encyclopedia of Telecommunications*, 2003.

[BH02] Andrew J Blanksby and Chris J Howland. A 690-mw 1-gb/s 1024-b, rate-1/2 low-density parity-check code decoder. *Solid-State Circuits, IEEE Journal of*, 37(3) :404–412, 2002.

[BJH+03] Boyd Bangerter, Eric Jacobsen, Minnie Ho, Adrian Stephens, Alexander Maltsev, Alexey Rubtsov, and Ali Sadri. High-throughput wireless lan air interface. *Intel Technology Journal*, 7(3) :47–57, 2003.

[Bla90] Richard E Blahut. *Digital transmission of information*. Addison-Wesley, 1990.

[BSL+11] Tuncer Baykas, Chin-Sean Sum, Zhou Lan, Junyi Wang, M Azizur Rahman, Hiroshi Harada, and Shuzo Kato. Ieee 802.15. 3c : the first ieee wire-

less standard for data rates over 1 gb/s. *Communications Magazine, IEEE*, 49(7) :114–121, 2011.

[CA12] Vikram Arkalgud Chandrasetty and Syed Mahfuzul Aziz. A highly flexible ldpc decoder using hierarchical quasi-cyclic matrix with layered permutation. *Journal of Networks*, 7(3) :441–449, 2012.

[CDE+05] Jinghu Chen, Ajay Dholakia, Evangelos Eleftheriou, Marc PC Fossorier, and Xiao-Yu Hu. Reduced-complexity decoding of ldpc codes. *Communications, IEEE Transactions on*, 53(8) :1288–1299, 2005.

[CF02a] Jinghu Chen and Marc PC Fossorier. Density evolution for two improved bp-based decoding algorithms of ldpc codes. *Communications Letters, IEEE*, 6(5) :208–210, 2002.

[CF02b] Jinghu Chen and Marc PC Fossorier. Near optimum universal belief propagation based decoding of low-density parity check codes. *Communications, IEEE Transactions on*, 50(3) :406–414, 2002.

[CFJRU01] Sae-Young Chung, G David Forney Jr, Thomas J Richardson, and Rüdiger Urbanke. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *Communications Letters, IEEE*, 5(2) :58–60, 2001.

[CFRU01] Sae-Young Chung, G.D. Forney, T.J. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 dB of the shannon limit. *IEEE Communications Letters*, 5(2) :58–60, 2001.

[CL07] Lei-Fone Chen and Chen-Yi Lee. Design of a dvb-t/h cofdm receiver for portable video applications [topics in circuits for communications]. *Communications Magazine, IEEE*, 45(8) :112–120, 2007.

[CLB+07] Sammo Cho, GwangSoon Lee, Byungjun Bae, KyuTae Yang, Chung-Hyun Ahn, Soo-In Lee, and Chiteuk Ahn. System and services of terrestrial digital multimedia broadcasting (t-dmb). *IEEE Transactions on Broadcasting*, 53(1) :171–178, 2007.

[CYL+14] Chung-Chao Cheng, Jeng-Da Yang, Huang-Chang Lee, Chia-Hsiang Yang, and Yeong-Luh Ueng. A fully parallel ldpc decoder architecture using probabilistic min-sum algorithm for high-throughput applications. *Circuits and Systems I : Regular Papers, IEEE Transactions on*, 61(9) :2738–2746, 2014.

[DCK08] Ahmad Darabiha, A Chan Carusone, and Frank R Kschischang. Power reduction techniques for ldpc decoders. *Solid-State Circuits, IEEE Journal of*, 43(8) :1835–1845, 2008.

[DM98] Hllatthew C Davey and David JC MacKay. Low density parity check codes over gf (q). In *Information Theory Workshop, 1998*, pages 70–71. IEEE, 1998.

[Dor07] Jean-Baptiste Doré. *Optimisation conjointe de codes LDPC et de leurs architectures de décodage et mise en œuvre sur FPGA*. PhD thesis, INSA de Rennes, 2007.

[DYLD09] Hong Ding, Shuai Yang, Wu Luo, and Mingke Dong. Design and implementation for high speed ldpc decoder with layered decoding. In *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*, volume 1, pages 156–160. IEEE, 2009.

[EC01] Rich Echard and Shih-Chun Chang. The π-rotation low-density parity check codes. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 2, pages 980–984. IEEE, 2001.

[Eli54] Peter Elias. *ERROR.—FREE CODlNG*. 1954.

[Ete08a] Kamran Etemad. Overview of mobile wimax technology and evolution. *Communications magazine, IEEE*, 46(10) :31–40, 2008.

[Ete08b] Kamran Etemad. Overview of mobile wimax technology and evolution. *Communications magazine, IEEE*, 46(10) :31–40, 2008.

[FFF07] Colm P Fewer, Mark F Flanagan, and Anthony D Fagan. A versatile variable rate ldpc codec architecture. *Circuits and Systems I : Regular Papers, IEEE Transactions on*, 54(10) :2240–2251, 2007.

[FJ73] G David Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3) :268–278, 1973.

[FMI99] Marc PC Fossorier, Miodrag Mihaljevic, and Hideki Imai. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *Communications, IEEE Transactions on*, 47(5) :673–680, 1999.

[For74] G David Forney. Convolutional codes iii. sequential decoding. *Information and Control*, 25(3) :267–297, 1974.

[Gal62] Robert G Gallager. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1) :21–28, 1962.

[GE09] H. Griesser and J.P. Elbers. *Forward error correction coding*. Google Patents, January 2009. US Patent 7,484,165.

[GFZ03] Javier Garcia-Frias and Wei Zhong. Approaching shannon performance by iterative decoding of linear codes with low-density generator matrix. *Communications Letters, IEEE*, 7(6) :266–268, 2003.

[Gui04] Frederic Guilloud. Generic architecture for ldpc codes decoding. *ENST Paris, Ph. D Thesis*, 2004.

[iee12] Ieee 802.11 standard-2012. Technical report, March 2012.

[Jab09] Houssein Jaber. *Conception architecturale haut débit et sûre de fonctionnement pour les codes correcteurs d'erreurs*. PhD thesis, Metz, 2009.

[JT06] Jie Jin and Chi-Ying Tsui. A low power viterbi decoder implementation using scarce state transition and path pruning scheme for high throughput wireless applications. In *Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on*, pages 406–411. IEEE, 2006.

[JVSV03] Christopher Jones, Esteban Vallés, Michael Smith, and John Villasenor. Approximate-min constraint node updating for ldpc code decoding. In *Mi-*

litary Communications Conference, 2003. MILCOM'03. 2003 IEEE, volume 1, pages 157–162. IEEE, 2003.

[JYL10] Xueqin Jiang, Yier Yan, and Moon Ho Lee. Semi-random and quasi-cyclic ldpc codes based on multiple parity-check codes. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[KAM06] Zahid Khan, Tughrul Arslan, and Scott MacDougall. A real time programmable encoder for low density parity check code as specified in the ieee p802. 16e/d7 standard and its efficient implementation on a dsp processor. In *SoC Conference, 2006 IEEE International*, pages 17–20. IEEE, 2006.

[KG06] S Kopparthi and DM Gruenbacher. A high speed flexible encoder for low-density parity-check codes. In *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, volume 1, pages 347–351. IEEE, 2006.

[Lee05] Chanho Lee. Design of encoder and decoder for ldpc codes using hybrid h-matrix. *ETRI journal*, 27(5) :557–562, 2005.

[LLWJ04] D-U Lee, Wayne Luk, Connie Wang, and Christopher Jones. A flexible hardware encoder for low-density parity-check codes. In *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*, pages 101–111. IEEE, 2004.

[Mac99] David JC MacKay. Good error-correcting codes based on very sparse matrices. *Information Theory, IEEE Transactions on*, 45(2) :399–431, 1999.

[MB07] Tinoosh Mohsenin and Bevan M Baas. High-throughput ldpc decoders using a multiple split-row method. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–13. IEEE, 2007.

[MB10] Tinoosh Mohsenin and Bevan M Baas. A split-decoding message passing algorithm for low density parity check decoders. *Journal of Signal Processing Systems*, 61(3) :329–345, 2010.

[MM06] Alberto Morello and Vittoria Mignone. Dvb-s2 : the second generation standard for satellite broad-band services. *Proceedings of the IEEE*, 94(1) :210–227, 2006.

[MN97] David JC MacKay and Radford M Neal. Good codes based on very sparse matrices. In *Cryptography and Coding*, pages 100–111. Springer, 1997.

[Moo05] Todd K Moon. Error correction coding. *Mathematical Methods and Algorithms. Jhon Wiley and Son*, 2005.

[MTB09] Tinoosh Mohsenin, Dean Truong, and Bevan Baas. Multi-split-row threshold decoding implementations for ldpc codes. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2449–2452. IEEE, 2009.

[MWD99] David JC MacKay, Simon T Wilson, and Matthew C Davey. Comparison of constructions of irregular gallager codes. *Communications, IEEE Transactions on*, 47(10) :1449–1454, 1999.

[NPR05] Tom Nelson, Erik Perrins, and Michael Rice. Common detectors for shaped offset qpsk (soqpsk) and feher-patented qpsk (fqpsk). In *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, volume 6, pages 5–pp. IEEE, 2005.

[OM01] Travis R Oenning and Jaekyun Moon. A low-density generator matrix interpretation of parallel concatenated single bit parity codes. *Magnetics, IEEE Transactions on*, 37(2) :737–741, 2001.

[PCPY10] Eldad Perahia, Carlos Cordeiro, Minyoung Park, and L Lily Yang. Ieee 802.11 ad : Defining the next generation multi-gbps wi-fi. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.

[PJ87] Peyton Z Peebles Jr. Digital communication systems. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1987, 445 p.*, 1, 1987.

[PLP99] Li Ping, WK Leung, and Nam Phamdo. Low density parity check codes with semi-random parity check matrix. *Electronics Letters*, 35(1) :38–39, 1999.

[PS13] Eldad Perahia and Robert Stacey. *Next Generation Wireless LANs : 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[PZB95] Roger L Peterson, Rodger E Ziemer, and David E Borth. *Introduction to spread-spectrum communications*, volume 995. Prentice Hall New Jersey, 1995.

[Rei06] Ulrich H Reimers. Dvb-the family of international standards for digital video broadcasting. *Proceedings of the IEEE*, 94(1) :173–182, 2006.

[RU01a] Thomas J Richardson and Rüdiger L Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *Information Theory, IEEE Transactions on*, 47(2) :599–618, 2001.

[RU01b] Thomas J Richardson and Rüdiger L Urbanke. Efficient encoding of low-density parity-check codes. *Information Theory, IEEE Transactions on*, 47(2) :638–656, 2001.

[SC06] Manhung Siu and Arthur Chan. A robust viterbi algorithm against impulsive noise with application to speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(6) :2122–2133, 2006.

[Sha48] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423, 623–656, July, October 1948.

[SKC06] Yang Sun, Marjan Karkooti, and Joseph R Cavallaro. High throughput, parallel, scalable ldpc encoder/decoder architecture for ofdm systems. In *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, pages 39–42. IEEE, 2006.

[Skl01] Bernard Sklar. *Digital communications*, volume 2. Prentice Hall NJ, 2001.

[Tan81] Robert Michael Tanner. A recursive approach to low complexity codes. *Information Theory, IEEE Transactions on*, 27(5) :533–547, 1981.

[TKS02] Michael Tüchler, Ralf Koetter, and Andrew C Singer. Turbo equalization : principles and new results. *Communications, IEEE Transactions on*, 50(5) :754–767, 2002.

[TVLZ15] Federico Tramarin, Stefano Vitturi, Michele Luvisotto, and Andrea Zanella. The ieee 802.11 n wireless lan for real-time industrial communication. In *Factory Communication Systems (WFCS), 2015 IEEE World Conference on*, pages 1–4. IEEE, 2015.

[UPH+08] Pascal Urard, L Paumier, V Heinrich, N Raina, and Nitin Chawla. A 360mw 105mb/s dvb-s2 compliant codec based on 64800b ldpc and bch codes enabling satellite-transmission portable devices. In *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pages 310–311. IEEE, 2008.

[Vel12] Purushotham Murugappa Velayuthan. *Towards Optimized Flexible Multi-ASIP Architectures for LDPC/Turbo Decoding*. PhD thesis, Télécom Bretagne, Université de Bretagne-Sud, 2012.

[WC07] Zhongfeng Wang and Zhiqiang Cui. Low-complexity high-speed decoder design for quasi-cyclic ldpc codes. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 15(1) :104–114, 2007.

[YC09] Hemesh Yasotharan and Anthony Chan Carusone. A flexible hardware encoder for systematic low-density parity-check codes. In *Circuits and Systems, 2009. MWSCAS'09. 52nd IEEE International Midwest Symposium on*, pages 54–57. IEEE, 2009.

[YHL03] Yu Yi, Jia Hou, and Moon Ho Lee. Design of semi-algebraic low-density parity-check (sa-ldpc) codes for multilevel coded modulation. In *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 931–934. IEEE, 2003.

[ZP01a] Tong Zhang and Keshab K Parhi. A class of efficient-encoding generalized low-density parity-check codes. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 4, pages 2477–2480. IEEE, 2001.

[ZP01b] Tong Zhang and Keshab K Parhi. Vlsi implementation-oriented (3, k)-regular low-density parity-check codes. In *Signal Processing Systems, 2001 IEEE Workshop on*, pages 25–36. IEEE, 2001.

[ZWFY07] Juntan Zhang, Yige Wang, Marc PC Fossorier, and Jonathan S Yedidia. Iterative decoding with replicas. *Information Theory, IEEE Transactions on*, 53(5) :1644–1663, 2007.

[ZXZ+15] Yijin Zhao, Yin Xu, Kang Zhao, Dazhi He, Wenjun Zhang, Hongbin Li, and Jingfei Cui. High throughput ldpc code and decoder design for hinoc 2.0 systems. In *Broadband Multimedia Systems and Broadcasting (BMSB), 2015 IEEE International Symposium on*, pages 1–5. IEEE, 2015.

[ZZB05] Jianguang Zhao, Farhad Zarkeshvari, and Amir H Banihashemi. On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (ldpc) codes. *Communications, IEEE Transactions on*, 53(4) :549–554, 2005.

RÉSUMÉ

La connectivité sans fils est devenue essentielle dans un nombre sans cesse croissant de systèmes dans quasiment tous les domaines d'activité, tant professionnels que privés. Les applications et appareils concernés, très variés, doivent faire usage de protocoles toujours plus nombreux et diversifiés. Ceci induit en permanence de nouvelles contraintes et exigences auxquelles il n'est possible de faire face que par un effort permanent d'innovation dans les domaines des techniques transmission, des architectures matérielles/logicielles et des méthodologies de conception. Il s'agit notamment d'accroître encore les débits, l'autonomie et l'intégration, tout en maîtrisant, voire réduisant, les coûts de ces systèmes.
La transfert fiable à haut débit de données via des canaux de transmission sans fils bruités, donc non fiable, nécessite l'emploi d'un codage canal réalisé le plus souvent à l'aide de techniques de codage correcteur d'erreur. Parmi ces dernières, très nombreuses, les codes LDPC constituent l'une des famille les plus efficaces concernant la capacité à approcher de très près la limite de Shannon, ce qui les rends très attractifs dans de nombreux domaines et normes (ex : dans DVB-S2 en télévision numérique, IEEE 802.11n [WiFi] et IEEE 802.16e [WiMAX] en transmission sans fils mobile).
Ce travail de thèse propose de nouvelles architectures rapides et facilement configurables pour le codage et le décodage d'une famille de codes LDPC, les codes QC-LDPC. Les architectures proposées pour les codeurs et décodeurs ciblent une réalisation sur FPGA. Elles offrent des vitesses de traitement qui peuvent être très élevées grâce à des structures matérielles fortement parallèles. Les paramètres configurables concernent les caractéristiques du code ciblé et le degré de traitement parallèle souhaité.
Les architectures proposées ont été validées sur plusieurs codes QC-LDPC. À chaque fois, différents degrés de traitement parallèle ont été sélectionnées, permettant de varier le compromis entre performances (vitesse de traitement, donc débit) et coût (surface matérielle requise). Malgré les contraintes induites par le choix de la configurabilité, des débits relativement élevés peuvent être atteint avec des niveau de parallélisme raisonnables. En limitant le niveau de parallélisme, il est possible de fortement restreindre les coûts matériels sans trop limiter les débits atteints.

Mots clés :

ABSTRACT

Wireless connectivity has turned essential in a growing number of systems in nearly all the domains of activity, either professional or personal. The concerned applications and devices, very varied, must make use of ever increasing and changing protocols. This constantly incurs new constraints and requirements that can only be faced with permanent innovation effort in the fields of transmission techniques, hardware/software architectures and design methodologies.
The reliable transmission of data at high data rates over noisy and hence unreliable channels, requires channel coding to be used most of the time employing error correcting codes. Of these, LDPC codes are among the most effective concerning their ability to achieve near Shannon limit performance. This makes them very attractive in many areas and standars (eg. DVB-S2 in digital television, IEEE 802.11n [WiFi] and IEEE 802.16e [WiMAX] in wireless networks).
The work in this thesis proposes a new architectures that is fast and easily configurable for the coding and decoding of codes from the QC-LDPC sub-family of LDPC codes. The proposed encoder and decoder architectures target implementation on FPGA. These architectures provide processing speeds that can be very high thanks to highly parallel hardware structures. Configurable parameters relate to the target code characteristics and the desired degree of parallel processing.
The proposed architectures have been validated for many QC-LDPC codes. Each time, different levels of parallel processing were selected to vary the trade-off between performance (processing speed, id. flow) and cost (hardware area requirements). Despite the constraints inherent to the choice of configurability, rather high flow rates can be achieved with reasonable level of parallelism. By limiting the level of parallelism, it is possible to greatly restrict the hardware costs without yet too much limiting the achieved throughputs.

Keywords :