**Article court**

Short paper

# Conception VHDL[1] et implémentation sur FPGA[2] du Code Reed Solomon (15,k,d)

VHDL Conception and implementation on FPGA of (15,k,d) Reed-Solomon code

**Said Najah, Mostafa Mrabti**

Faculté des sciences Dhar el Mhraz, Département de physique, LESSI
B.P. 1796 Fès, Maroc
E-mail : {mrabti_lessi@yahoo.fr} {najah.lessi@caramail.com}

Manuscrit reçu le 28 juillet 2004

Résumé et mots clés

**Le code Reed Solomon est un code détecteur et correcteur d'erreurs qui joue un rôle très important pour la transmission numérique. Nous proposons dans ce papier une implémentation matérielle à partir d'une description VHDL de ce code. L'implémentation est réalisée sur un FPGA de Xilinx. L'architecture proposée a un débit de 80 Mbps avec une fréquence de 20 MHZ, et une surface de 1308 CLBs.**

**Code détecteur et correcteur d'erreurs, code Reed Solomon, VHDL, FPGA.**

Abstract and key words

The Reed Solomon code is a detecting corrective code, which play a very important role for the digital transmission. We propose in this paper a design and implementation with VHDL langage description. The implementation is realized on a FPGA of Xilinx. The proposed architecture has throughput of 80 Mbps with a frequency of 20 MHZ, and a surface of 1308 CLBs.

Detecting correcting code, Reed Solomon code, VHDL, FPGA.

## 1. Introduction

Dans tout système qui manipule de l'information, il est impossible sans technique de codage d'éviter qu'une information ne soit modifiée par des phénomènes d'origines diverses, (mécanique, électrique, électromagnétique etc…).
L'imprévisibilité du message émis par la source impose alors au récepteur l'utilisation de techniques lui permettant de vérifier à la fois l'exactitude et la certitude de l'information reçue.

Le mécanisme consiste à coder l'information en rajoutant des symboles au message suivant une loi connue à la fois de l'émetteur et du récepteur. Deux types de codes existent, à savoir le code détecteur et le code détecteur et correcteur d'erreurs.
On parle de code détecteur d'erreurs lorsqu'il détermine uniquement si le message reçu est entaché d'erreurs. Comme il n'y a que la détection des erreurs, cela entraîne une retransmission du code reçu faux détecté comme tel. Cette stratégie nécessite cependant un canal de retour du type half ou full duplex et la répétition se fait à la requête du récepteur.

1. VHDL : VHSIC Hardware Description Language
VHSIC : Very High Speed Integrated Circuits
2. FPGA : Field Programmable Gate Arrays

Il s'agit d'un code détecteur et correcteur d'erreurs si celui-ci permet de détecter et de corriger les erreurs présentes dans le message. La transmission du signal s'effectue toujours dans un milieu physique, où des dégradations interviennent. Le signal émis, subit physiquement des altérations qui entraînent des erreurs sur l'information reçue, d'où la nécessite de la détection et de la correction des erreurs. La conception d'un code détecteur et correcteur d'erreurs doit tenir compte donc de celui qui arrête le mieux les configurations d'erreurs fréquemment rencontrées sur la ligne de transmission utilisée.
Les implémentations logicielles des algorithmes de codage ne répondent pas cependant aux besoins en performances et en vitesse pour les systèmes temps réel. La solution est donc de concevoir des architectures qui peuvent être implantées matériellement sur des circuits VLSI. Dans ce travail on donne une simulation du code Reed Solomon (15,k,d) à l'aide du langage VHDL, et on propose d'implémenter ce code sur une carte spécialisée fonctionnant autour d'un composant programmable FPGA de Xilinx. Dans un premier temps, les différents blocs ont été testés séparément sur le FPGA XC4013[3]. L'implémentation du schéma complet du circuit a nécessité l'utilisation du FPGA XC4062[3] de taille plus grande. La surface occupée pour le code RS (15,9,7) implémenté sur le FPGA XC4062 est de 1308 CLBs avec un débit de 80 Mbps.

## 2. Code Reed Solomon

Les codes de Reed Solomon constituent une famille de codes d'une importance exceptionnelle, tant du point de vue de la théorie que des applications. Le code raccourci de Reed Solomon (15,9,7), est très utilisé en communication radios et par satellite. Le RS(15,9,7) fera l'objet d'un travail ultérieur ou il est concaténé avec un code convolutif pour une meilleure correction. Les codes Reed Solomon sont des codes non binaires construits sur un alphabet de taille $q$ muni des propriétés d'un corps fini. L'alphabet de taille $q$ est telle que $q = p^m$, où $p$ est premier et $m$ entier, il s'agit d'une extension du corps binaire, donc $p = 2$ dans la plupart des applications [1], [2].
Les codes Reed Solomon peuvent être considérés comme un cas particulier des codes BCH, bien adaptés à la correction des paquets d'erreurs et atteignant la borne de Singleton [1].
Le polynôme irréductible et primitif $P(X)$ de degré $m$ sert à générer les éléments du corps de Galois utilisés dans tous les calculs.

### 2.1. Code Reed Solomon (15,k,d)

La distance minimum $d = 2^*t + 1$ où $t$ représente le nombre d'erreurs corrigibles, renseigne sur la capacité de correction du code [1], [2].

Le code Reed Solomon (15,k,d) est entièrement défini par le polynôme générateur $g(X)$. Les racines $\alpha$ de $P(X)$ sont les éléments du corps de Galois à $2^4 = 16$ éléments.
Le polynôme générateur $g(X)$ caractérise entièrement les propriétés de ce code en matière de détection et de correction. La taille des symboles est de 4 bits ($m = 4$).

### 2.2. Codage

La distance de Hamming $d$ permet de déterminer la capacité de correction du code détecteur correcteur d'erreurs.
Les paramètres $n$, $k$ et $d$ sont définis par :
- Longueur du code : $n = 2^m - 1$
- Taille du message : $k = 2^m - 1 - 2^*t$,
  $t$ : représente le nombre d'erreurs corrigeables.
- La distance de Hamming : $d = 2^*t + 1$.

Le polynôme générateur $g(X)$ est donné par :

$$g(X) = \prod_{i=0}^{d-2}(X - \alpha^i) = (X - \alpha^0)(X - \alpha^1)\ldots(X - \alpha^{d-2}). \quad (1)$$

Les éléments d'informations peuvent se mettre sous la forme suivante :

$$M(X) = \sum_{i=k-1}^{0} a_i X^i = a_{k-1}X^{k-1} + \ldots + a_1 X + a_0 \quad (2)$$

avec $a_i \in GF$ (16).
La redondance est le reste de la division de $X^{n-k*}M(X)$ par le polynôme $g(X)$. L'addition des coefficients est une arithmétique modulo deux.
Le reste peut s'écrire sous la forme de sommes :

$$R(X) = \sum_{j=n-k-1}^{0} r_j X^j = r_{n-k-1}X^{n-k-1} + \ldots + r_1 X + r_0 \quad (3)$$

avec $r_j \in GF$ (16).
Le reste $R(X)$ ainsi obtenu complète le message pour former le mot de code $C(X)$, ainsi l'expression littérale de $C(X)$ est donnée par :

$$C(X) = X^{n-k}\sum_{i=k-1}^{0} a_i X^i + \sum_{j=n-k-1}^{0} r_j X^j. \quad (4)$$

On signale que le codage est systématique. Les coefficients des polynômes $M(X)$, $R(X)$ et $C(X)$ peuvent être représentés soit sous forme de valeurs discrètes comprises entre 0 et 15, soit sous forme de puissance de $\alpha$.

3. XC4013 et XC4062 deux composantes de la société xilinx.

### 2.3. Décodage

Le mot de code $C(X)$ diffusé ou transmis peut subir des altérations dues à l'environnement. Le mot reçu $r(X)$ est égal à :

$$r(X) = [C(X) + E(X)] \bmod 2 \tag{5}$$

$E(X)$ représente l'expression polynomiale des erreurs.

$$E(X) = \sum_{j=n-1}^{0} b_j X^j = b_{n-1}X^{n-1} + b_1 X + \ldots + b_0 \tag{6}$$

avec $b_j \in GF$ (16).

#### 2.3.1. Syndrome $S(x)$

Un mot de code de Reed Solomon a $2t$ syndromes qui dépendent seulement des erreurs et sont calculés en substituant les $2t$ racines du polynôme générateur $g(X)$ dans $r(X)$

$$\begin{aligned} S(X) &= \sum_{i=1}^{d-1} S_i X^{i-1} = S_1 + S_2 X + \ldots\ldots + S_{d-1}X^{d-2} \\ S_i &= r(\alpha^{i-1}) \text{ et } i \in \{1, d-1\} \end{aligned} \tag{7}$$

#### 2.3.2. Calcul des polynômes localisateurs et évaluateurs

La méthode de décodage des codes Reed Solomon est basée sur l'équation clé de décodage :

$$\beta(X)S(X) = \gamma(X) \bmod [X^{2t}]$$

où $S(X)$ est le syndrome du mot reçu, $\beta(X)$ le polynôme localisateur d'erreurs correspondant et $\gamma(X)$ le polynôme évaluateur d'erreurs. L'algorithme de Berlekamp-Massey permet de calculer les polynômes $\beta(X)$ et $\gamma(X)$. Les racines de $\beta(X)$ obtenues sous forme de puissances de $\alpha$ permettent de localiser les positions des erreurs à l'intérieur du mot reçu. Nous utilisons le polynôme $\gamma(X)$ évaluateur d'erreurs pour calculer la valeur corrective des erreurs. Les coefficients de $\beta(X)$ et $\gamma(X)$ sont donnés par l'algorithme Berlekamp-Massey :

Initialisation :
$\beta_0 \longleftarrow 1$, $\gamma_0 \longleftarrow 0$, $\beta'_0 \longleftarrow 0$, $\gamma'_0 \longleftarrow -1$, $d_0 \longleftarrow 0$
Faire pour $j = 0,1,\ldots,2^*t - 1$ :
Calculer l'anomalie de rang $j$ définie par
$\Delta_j =$ le coefficient en $X^j$ du produit $\beta_j(X)S(X)$
Calculer
$\beta_{j+1} \longleftarrow \beta_j - \Delta_j \beta'_j$, $\gamma_{j+1} \longleftarrow \gamma_j - \Delta_j \gamma'_j$
Si ($\Delta_j = 0$ ou $2^*d_j > j$) alors
$\beta'_{j+1} \longleftarrow X\beta'_j$, $\gamma'_{j+1} \longleftarrow X\gamma'_j$, $d_{j-1} \longleftarrow d_j$
sinon
$\beta'_{j+1} \longleftarrow X\Delta_j^{-1}\beta_j$, $\gamma'_{j+1} \longleftarrow X\Delta_j^{-1}\gamma_j$, $d_{j+1} \longleftarrow j + 1 - d_j$

# 3. Écriture VHDL du circuit

Les FPGA ont connu une grande amélioration en taille et en vitesse. Aussi, les FPGA constituent des plates formes plus adéquates pour l'implémentation des applications des codes détecteurs correcteurs d'erreurs où l'application exacte subit de nombreux changements. Plusieurs études sur des « codeurs/décodeurs » de Reed Solomon ont déjà été réalisées tant dans le domaine universitaire [3], [4], [5] qu'industriel [6], [7]. Nous avons donc étudié l'implémentation sur FPGA d'une fonction « codeur/décodeur » de Reed Solomon qui peut être réutilisée comme composant pour la synthèse d'autres applications traitant des flots de données en continu. L'ensemble des blocs fonctionnels a été étudié avec une architecture pipeline.
La description VHDL du code Reed Solomon est faite de telle sorte que chaque bloc de l'architecture proposée est décrit dans une entité indépendante. L'architecture correspondante à chaque entité détermine son rôle dans le circuit global. Pour assurer toutes les fonctions du système, une entité globale et son architecture sont décrites en utilisant les entités précédentes comme des composants.

### 3.1. Multiplieur dans un corps de Galois

Le multiplieur dans un corps de Galois est un des composants principaux utilisés, surtout dans l'algorithme de Berlekamp-Massey. Nous avons utilisé un multiplieur de Mastrovito qui utilise une matrice pour le calcul du produit de deux vecteurs [9]-[10]. La présentation des éléments du corps de Galois est obtenue en choisissant comme polynôme primitif :

$$P(X) = X^4 + X + 1$$

### 3.2. Codage

Dans cette étude, nous avons choisi les paramètres suivants : (n,k,d) = (15,9,7). Un circuit qui calcule le reste de la division de $X^{6*}M(X)$ par le polynôme $g(X)$ :

$$g(X) = X^6 + \alpha^9 X^5 + \alpha^{12} X^4 + \alpha X^3 + \alpha^2 X^2 + \alpha^4 X + 1$$

Le schéma du codeur est donné par la figure suivante.
CLK : Signal d'horloge du circuit actif à l'état montant.
RESET : Remise à zéro du circuit
ENABLE : Sert à contrôler toutes les entrées synchrones sauf RESET (lorsque ENABLE est dans un état inactif l'entrée RESET est dans son état courant).
D_IN : Entrée de données (elle est validée sur un front montant du signal d'horloge CLK).
OUT_ENB : Signal actif lors du fonctionnement du circuit.
D_OUT : Sortie codée.

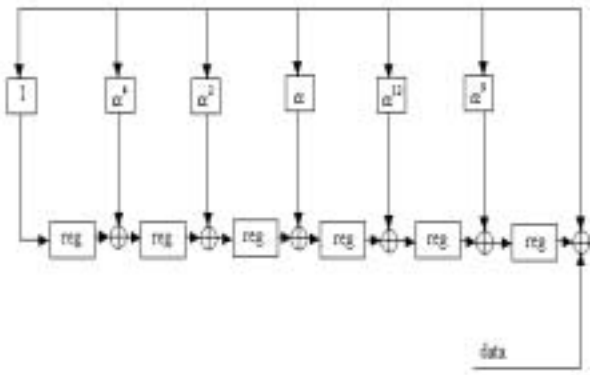


*Figure 1. Circuit de calcul du reste.*

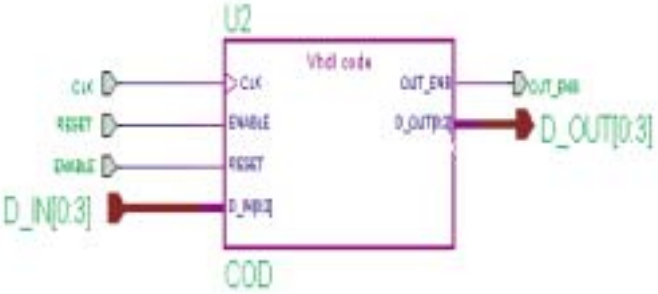


*Figure 2. Broches d'entrées/sorties du codeur.*

### 3.3. Le décodage

Pour le décodage nous avons utilisé l'architecture suivante :

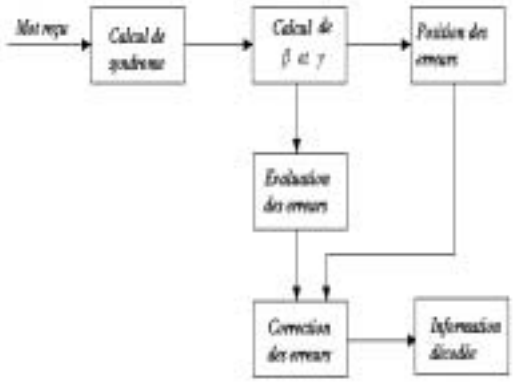


*Figure 3. Architecture utilisée pour le décodeur du Reed Solomon (15, k, d).*

### 3.4. Polynômes localisateurs et évaluateurs

On utilise un circuit qui calcule les coefficients des deux polynômes $\beta(X)$ et $\gamma(X)$. Ce circuit est constitué par des circuits élémentaires. Les circuits des figures 4 et 5 traduisent l'algorithme de Berlekamp-Massey (paragraphe 2.3.2) qui calcule les coefficients des polynômes localisateurs et évaluateurs.
Chaque étape de l'algorithme de Berlekamp-Massey est manifestée sur les figures 4 et 5 par un bloc traduisant tous les calculs et les comparaisons nécessaires. Le calcul de l'anomalie $\Delta_j$

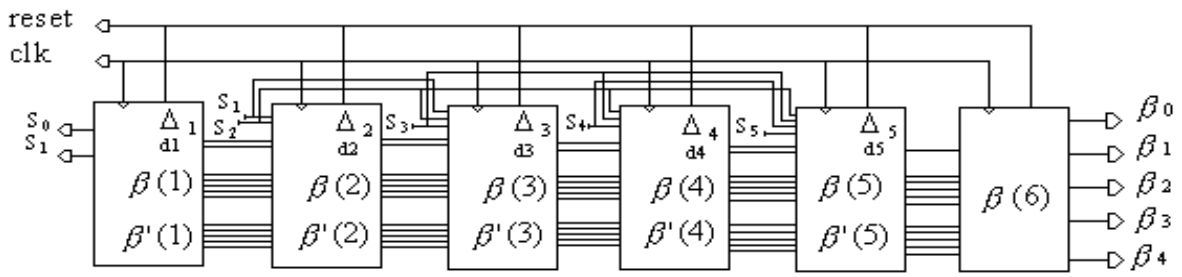


*Figure 4. Schéma d'implantation du polynôme $\beta(X)$.*

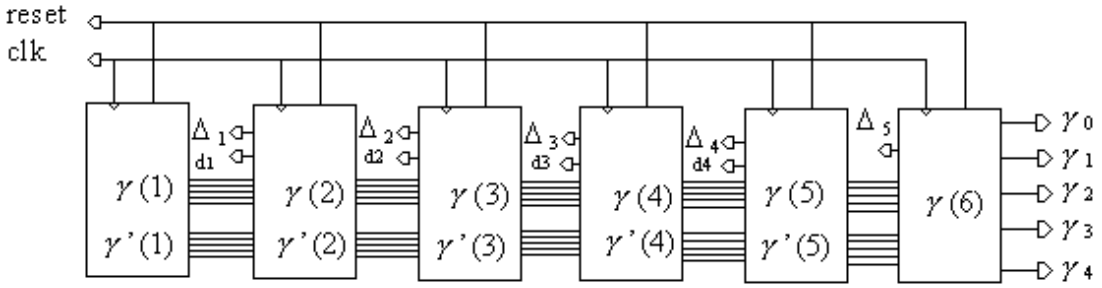


*Figure 5. Schéma d'implantation du polynôme $\gamma(X)$.*

de rang $j$ qui représente le coefficient en $X^j$ du produit $\beta_j(X)S(X)$ nous permet de déduire d'une manière récursive les valeurs des coefficients $\beta_j$ (resp. $\gamma_j$) des polynômes localisateur (resp. évaluateur).
Le polynôme $\beta(X)$ étant connu, un calcul de $\beta(\alpha^i)$ avec $i$ variant de 0 à 14 (taille de la donnée) permet grâce à un test sur zéro de connaître les racines du polynôme $\beta(X)$ et donc les positions des erreurs. Quand ces positions sont connues, on peut calculer les valeurs correctives grâce au polynôme $\gamma(X)$.
Le circuit final qui donne les coefficients du polynôme $\beta(X)$ (voir figure 4).
Le circuit final qui donne les coefficients du polynôme $\gamma(X)$ (voir figure 5).

# 4. Résultats

La méthodologie de simulation adoptée dans ce travail consiste à former le mot de code $C(X)$ en appliquant l'algorithme de codage sur la trame de l'information. L'injection d'erreurs consiste à faire une addition modulo 2 entre les mots $C(X)$ et ceux d'erreurs représentant un modèle d'environnement.
L'exemple de simulation, traite le cas d'un message affecté de deux erreurs. Le message à transmettre est le suivant :

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\alpha^{14}$ | $\alpha^{14}$ | $\alpha^{12}$ | $\alpha^2$ | $\alpha$ | 1 | $\alpha^3$ | $\alpha^4$ | $\alpha^3$ |

Le message codé est :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\alpha^{14}$ | $\alpha^{14}$ | $\alpha^{12}$ | $\alpha^2$ | $\alpha$ | 1 | $\alpha^3$ | $\alpha^4$ |
| $\alpha^3$ | $\alpha^{14}$ | $\alpha^3$ | $\alpha^9$ | $\alpha^3$ | $\alpha^{12}$ | $\alpha^{10}$ | |

Le message reçu est affecté par deux erreurs en position 9 et 13 avec respectivement les amplitudes $\alpha^8$ et $\alpha^7$.

Le message reçu bruité est :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\alpha^{14}$ | $\alpha$ | $\alpha^{12}$ | $\alpha^2$ | $\alpha$ | $\alpha^2$ | $\alpha^3$ | $\alpha^4$ |
| $\alpha^3$ | $\alpha^{14}$ | $\alpha^3$ | $\alpha^9$ | $\alpha^3$ | $\alpha^{12}$ | $\alpha^{10}$ | |

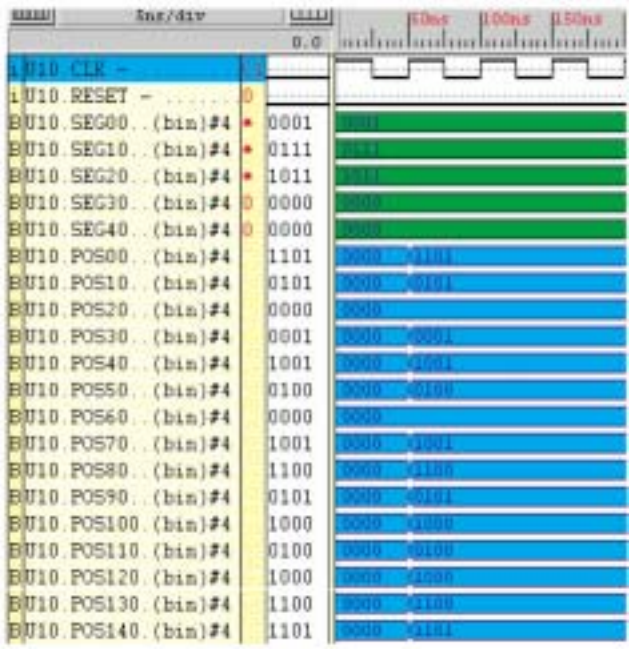


*Figure 6. Détection des positions d'erreurs.*

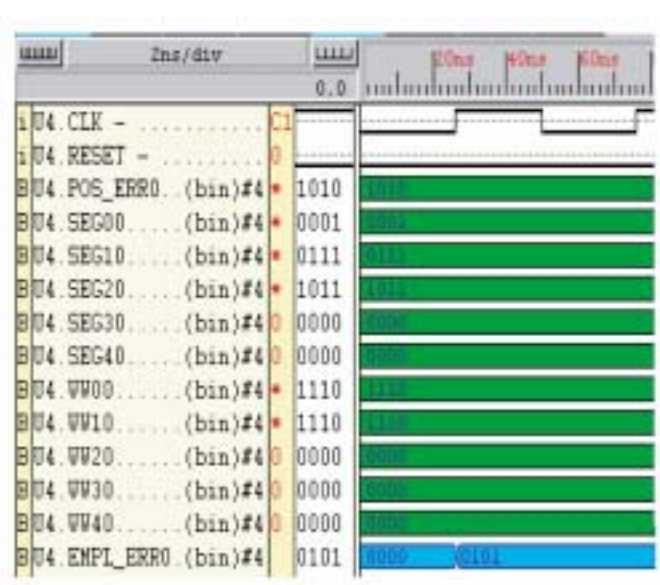


*Figure 7. Amplitude de première erreur détectée.*

Après le calcul des coefficients du syndrome et les coefficients des deux polynômes $\beta(X)$ et $\gamma(X)$, un calcul par une architecture pipeline de $\beta(\alpha^i)$ avec $i$ variant de 0 à 14 (taille de la donnée) permet de localiser les positions des erreurs qui correspondent à $\beta(\alpha^i) = 0$ (figure 6).
Les amplitudes des erreurs détectées et qui sont calculées à partir du polynôme évaluateur $\gamma(\alpha^i)$ sont données par les figures 7 et 8.

Le FPGA XC4013 de Xilinx, sur lequel on a testé au départ les différents blocs séparément contient 576 CLBs et 192 broches entrées/sorties. Le XC4062 est caractérisé par 5376 CLBs et 384 broches entrées/sorties.
L'architecture du codeur/décodeur a été décrite en VHDL et implantée sur FPGA XC4062 en utilisant Xact de la société Xilinx.

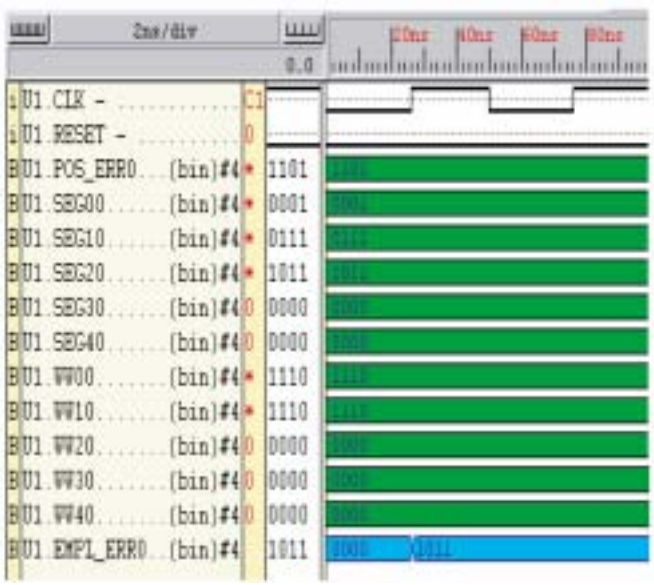


*Figure 8. Amplitude de deuxième erreur détectée.*

La fréquence de fonctionnement est de 20 MHZ, le débit est de 80 Mbps et la surface occupée par chaque circuit est donnée par le tableau suivant :

*Tableau 1. Résultat des simulations.*

| | Nombre de CLBs |
|---|---|
| Multiplieur dans le corps de Galois | **6** |
| Inverseur dans le corps de Galois | **2** |
| Encodeur Reed Solomon | **14** |
| Syndrome | **459** |
| Algorithme De Berlekamp | **129** |
| Détecteur de la position d'erreurs | **53** |
| Calculateur de valeurs d'erreurs | **645** |

L'architecture choisie, pour cette implantation permet de diminuer le nombre de cycles $N$ nécessaires pour avoir une donnée décodée :

$$N = N_1 + N_2 + N_3 + N_4$$

$N_1$ : nombre de cycles d'horloge nécessaire au calcul du syndrome (2 dans notre cas).
$N_2$ : nombre de cycles nécessaires pour calculer les deux polynômes (4 cycles)
$N_3$ : nombre de cycles d'horloge nécessaires pour déterminer la position des erreurs (un cycle).
$N_4$ : nombre de cycles d'horloge nécessaires à la correction des erreurs (un cycle).
Pour le cas du code Reed Solomon (15,9,7), les opérations nécessitent un temps de latence de 8 cycles d'horloge.
Pour l'architecture adoptée en [5] il faudrait 15 cycles pour calculer uniquement les coefficients du syndrome.
La surface occupée pour le code RS (15,9,7) implémentée sur XC4062 de Xilinx est de 2800 CLBs [8], soit 52,08 % de la surface du FPGA. Pour notre étude on a obtenu pour le FPGA XC4062 de Xilinx 1308 CLBs, soit 24,33 % de la surface du FPGA. C'est-à-dire on a diminué la surface de plus de 50 % par rapport aux résultats [8].

# 5. Conclusion

Nous avons réalisé la conception d'un codeur/décodeur Reed Solomon en utilisant le langage VHDL. Chaque bloc est créé indépendamment des autres. La mise en œuvre de chaque bloc facilite la mise au point du programme global.
Les résultats obtenus, surface occupée et le temps de latence sont très convaincants puisqu'on a diminué le temps pour qu'un mot de code soit décodé et la surface occupée en adaptant une architecture dans laquelle chaque bloc soit pipeline et/ou parallélisé.
La prochaine étape consiste à traiter le code Reed Solomon avec effacement pour donner une efficacité à la correction.

# Références


[1] G. BATTAIL, Théorie de l'information : « Application aux techniques de communication », *Masson*, 1997.

[2] G. COHEN, J.-L. DORNSTELLER, P. GODLEWSKI, « Codes correcteurs d'erreurs : une introduction au codage algébrique », *Masson*, Paris, 1992.

[3] B. HEATHER, HUI ZHANG, « Comparison of Reed Solomon code implementations », CS252 Project, Université de Berkeley, 1996.

[4] A. DABBAGH, « Étude et conception d'un circuit de détection et correction d'erreurs en transmission d'informations numériques », *Thèse présentée à l'université de Rennes I*, 1995.

[5] A. DANDACHE, T. VALLINO, F. MONTEIRO, J.-P. DELAHAYE, « code Reed Solomon (127,k,d) avec effacement : simulation et conception sur réseaux de circuits programmables (FPGA) » *Traitement de signal*, volume 16, n°4, pp. 331-341, 1999.

[6] « Reed Solomon decoders with erasures », Société *Hammer cores*, mars 1999.

[7] AHA 4011 : « 10 Mbytes/sec Reed Solomon Error correction device, Product specification, Advanced Hardware Architectures ».

[8] C. GREGORY, C. AHLQUIST, M. RICE, B. NELSON, « Error control coding in software radios: an FPGA Approch », *IEEE communication*, August 1999.

[9] E. MOSTROVITO, « VLSI Design For Multiplication over Finite Fields GF($2^n$) », Lecture Notes in Computer science 357, pp. 297-309, Berlin: Springer-Verlag, Mar. 1989.

[10] E. MASTROVITO, « VLSI Architectures For Computation in Galois Fields », PhD Thesis, Linköping Univ., Dept. of Electrical Eng., LinKöping, Sweden, 1991.



Mostafa **Mrabti**

Docteur d'État en Automatique et Traitement du Signal de la Faculté des Sciences de Fès (1996). Professeur à l'Université Sidi Mohammed Ben Abdellah de Fès et responsable du groupe de recherche SCAM (Signaux, Communications, Acoustique et Multimédia) au sein du laboratoire LESSI de la Faculté des Sciences de Fès. Membre du pôle de compétence STIC (Sciences et Techniques de l'Information et de la Communication). Ses activités de recherche concernent le codage et la conception des circuits pour des applications Télécom



Said **Najah**

Said Najah est titulaire en 2000 d'un Diplôme des Études Supérieures Approfondies D.E.S.A en Automatique et Analyse des Systèmes de la Faculté des Sciences Dhar Mehraz Fès, Maroc. Doctorant au laboratoire d'Électronique, Signaux, Systèmes et Informatique (LESSI) sous la direction de Professeur Mostafa Mrabti. Membre du groupe de recherche SCAM (Signaux, Communication, Acoustique et Multimédia). Ses thèmes de recherche concernent l'étude des codes détecteurs et correcteurs d'erreurs et l'implantation sur des circuits spécialisés de type FPGA.