

Co-simulation VHDL/Matlab d'une chaîne de communication numérique OFDM : application à la transmission d'images

Guillaume Ferré & Camille Leroux
Projet EN321

1 Objectifs du projet

L'objectif du projet est de réaliser la co-simulation VHDL/Matlab des couches physique et MAC d'une chaîne de communications numériques à haut débit binaire du type OFDM (Orthogonal Frequency Division Multiplexing). La validation du système proposé se fera par des calculs classiques de taux d'erreur binaire à différents endroits de la chaîne de communication, mais aussi par la transmission d'une image. Il s'agira dans ce cas d'une évaluation qualitative de l'image reçue. L'originalité de ce projet se situe à plusieurs niveaux :

- La couche physique que vous allez simuler est, d'un point de vue du détail des blocs à implémenter, similaire aux couches physiques des standards DVB-T (TNT) ou IEEE802.11(Wifi),
- L'interaction entre le VHDL et matlab. Qui va permettre de "décharger" des traitements Matlab "lent" et ainsi accélérer la vitesse d'exécution du simulateur.

Les objectifs du projet sont les suivants :

- Simuler une partie d'une couche physique de communication numérique sans fil haut débit : depuis le modulateur numérique en passant par les modulateur/démodulateur OFDM jusqu'au démodulateur numérique,
- Faire migrer un modèle de simulation logiciel d'une chaîne de communication (écrit en Matlab) vers un modèle intégrant à la fois des éléments logiciels et matériels (description VHDL),
- Comprendre les contraintes de conceptions matérielles liées aux systèmes de communications numériques (cadence bit/symbole, séquenceur, synchronisations de blocs, latence, etc.),
- Comprendre les contraintes de conceptions liées à la mise en œuvre de modèle de co-simulation : cohabitation d'éléments logiciels et matériels,
- Être capable de retranscrire son travail à l'oral.

2 Déroulement du projet

Le projet comporte deux parties. Une partie sera encadrée par M. Leroux et elle consistera à implémenter la partie codage de canal (scrambleur, codeur de Hamming, entrelaceur et codeur

convolutif) sur un circuit FPGA. L'autre partie sera encadrée par M. Ferré. Dans cette partie vous simulerez en Matlab l'émetteur et le récepteur OFDM depuis les symboles en sortie du modulateur numérique jusqu'au symboles détectés en sortie du démodulateur/égaliseur OFDM.

3 Évaluation

L'évaluation se fait à travers une présentation orale lors de laquelle vous présenterez votre travail et les résultats que vous avez obtenu. Une démonstration doit être effectuée. A l'issue de cette présentation/démonstration, des questions vous seront posées à propos de votre travail mais aussi sur des aspects attendant au cours. Le comportement (assiduité, ponctualité, etc.) durant les séances de TP rentre en compte dans la note finale. Noter que la présentation durera 15 minutes et que vos slides de présentation devront être déposés la veille de votre passage au format pdf dans un dossier google drive que vos encadrants partageront avec vous. L'ensemble de vos codes devront également être mis à disposition dans le drive.

4 Travail à réaliser

Les deux parties du projet sont à réaliser à partir du code Matlab fourni :

`com_chain_with_coding.m`

4.1 Partie préliminaire :

Dans cette partie il vous est demandé d'analyser le code Matlab fournit pour cela :

1. Décrire sur papier la chaîne de communications numériques à mettre en œuvre, puis présentez la à vos encadrants.
2. Quelles sont les modulations numériques utilisables ? représenter leur constellation.
3. Pourquoi il est nécessaire d'introduire un formalisme de couche MAC ?
4. Que contient comme information la couche MAC mise en œuvre ?

4.2 Partie 1 : Co-simulation Matlab/VHDL du codage de canal

L'objectif de cette partie est d'implémenter sur circuit FPGA (Xilinx Artix-7 100T-CSG324 intégré sur une carte de Digilent, la Nexys 4) la partie de la chaîne d'émission qui constitue le codage canal. Cette partie de l'émetteur comprend les blocs suivants :

- Un scrambleur,
- un codeur BCH (ou Hamming),
- un entrelaceur convolutif,
- un code convolutif.

4.2.1 Prise en main de la plateforme de co-simulation

Avant de commencer à décrire les blocs matériels, il est nécessaire de bien comprendre le flot de conception que nous allons utiliser. La Figure 1 représente la chaîne (sans OFDM). On remarque sur ce schéma que le scrambleur est décrit dans la chaîne Matlab mais qu'il est également implémenté sur le FPGA. La communication entre Matlab et la carte FPGA se fait via un module UART. Ce principe de co-simulation vous permettra de vérifier le fonctionnement

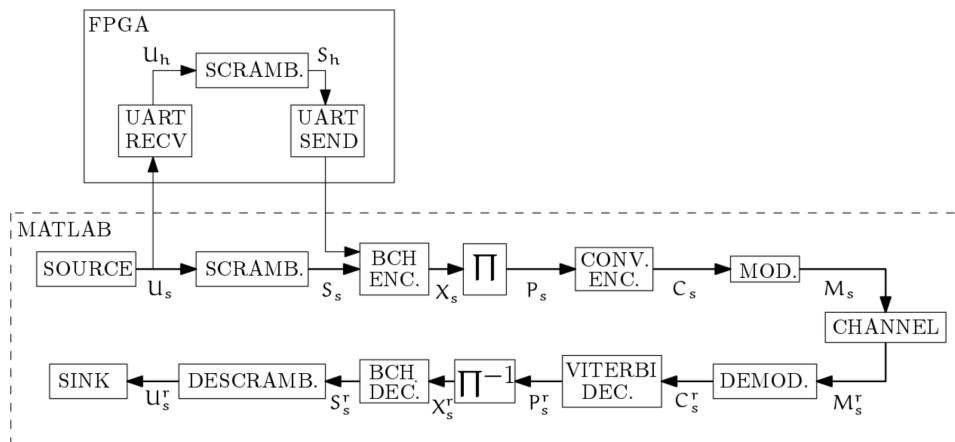


FIGURE 1 – Chaîne de communication sans le modulateur/démodulateur OFDM

de chaque bloc. L'objectif final étant d'avoir une chaîne fonctionnelle avec les 4 blocs du codage de canal sur le FPGA.

4.2.2 Mise en place d'un simple echo sur le FPGA

1. Ouvrez l'outil Vivado
2. Créez un nouveau projet
3. Ajoutez les fichiers sources VHDL fournis
4. Dessinez sur papier les blocs VHDL inclus dans le fichier tb_transmitter.vhd
5. Même chose pour le bloc UART_loop.vhd
6. A partir de là, déterminez ou devront être décrit les blocs de l'émetteur
7. Décrivez en VHDL un simple registre entre l'UART RECV en l'UART SEND
8. Simulez le comportement de votre système
9. Faites la synthèse et le placement routage de UART_loop.vhd
10. Tester l'écho sur la carte en envoyant les données à partir du fichier Matlab fourni.

4.2.3 Implémentation du scrambleur

1. En vous aidant du code Matlab, dessinez sur papier la structure du scrambleur
2. Dessinez le chronogramme du scrambleur
3. Déterminez les signaux de contrôle dont vous allez avoir besoin
4. Décrivez le scrambleur en VHDL
5. Simulez son comportement au sein du transmitter
6. Testez sur carte

4.2.4 Implémentation du codeur BCH

1. En vous aidant du code Matlab, dessinez sur papier la structure du codeur BCH et déterminez la structure des éventuels blocs de sérialisation/désérialisation.
2. Dessinez le chronogramme du codeur BCH
3. Déterminez les signaux de contrôle dont vous allez avoir besoin
4. Décrivez le codeur BCH en VHDL
5. Simulez son comportement au sein du transmetteur
6. Testez sur carte

4.2.5 Implémentation de l'entrelaceur

La description VHDL de l'entrelaceur vous est fournie. Pour ce bloc, votre travail se résume donc à une éventuelle adaptation de son architecture pour coller aux contraintes temporelles (synchronisation) de l'émetteur.

4.2.6 Implémentation du codeur convolutif

1. A partir du polynôme générateur du code Matlab, dessinez l'architecture du codeur convolutif
2. Dessinez le chronogramme du codeur convolutif
3. Déterminez les signaux de contrôle dont vous allez avoir besoin
4. Décrivez le codeur en VHDL
5. Simulez son comportement au sein du transmetteur
6. Testez sur carte

4.3 Partie 2 : simulation Matlab du modulateur / démodulateur OFDM

Dans cette partie vous devez implémenter le modulateur et le démodulateur OFDM. La présence de l'égaliseur de canal en sortie du démodulateur OFDM devra également être intégrée mais dans un second temps. De plus on considérera que le modulateur OFDM utilise $NFFT=64$ sous-porteuses.

Les sections à compléter dans le code `main_EN321_sans_ofdm_double_codeur.m`, sont celles qui encadrent la partie CANAL et l'ajout du bruit au récepteur (voir le code Matlab).

4.3.1 Communications numériques OFDM sur canal AWGN

Afin d'aborder l'implémentation du système par étape, vous allez tout d'abord implémenter le système modulateur / démodulateur OFDM en considérant que la canal de propagation n'introduit pas de distorsion. Autrement dit si h désigne la version discrète du canal vue par le récepteur, alors : $h=1$. Il n'est donc pas nécessaire d'introduire de préfixe cyclique à l'émission (et donc de l'extraire à la réception) ni d'implémenter un égaliseur de canal.

1. A partir de la connaissance a priori :
 - du nombre de bits codés à transmettre (c'est à dire ceux en entrée du modulateur numérique à l'émission),
 - du nombre de bits/symbole,

— du nombre de symbole par symbole OFDM,

Adaptez le code Matlab fourni pour que le nombre de bits en entrée du modulateur numérique soit tel que le nombre de symbole OFDM généré soit un nombre entier. Les bits supplémentaires à rajouter doivent être positionnés à la fin du vecteur de bits all bits obtenu en sortie du codeur convolutif. Les bits ajoutés doivent être uniformément distribués. Attention le récepteur n'aura pas de connaissance a priori sur le nombre de bits de padding ajouté, il faut donc adapter la couche MAC pour faire passer cette information au récepteur. Il faudra de plus justifier le nombre de bits utilisé pour coder cette information.

2. Implémenter le modulateur OFDM à base d'opérateur IFFT,
3. Visualiser l'allure des parties réelles et imaginaires de la trame OFDM en sortie du modulateur OFDM,
4. Visualiser l'histogramme de la partie réelle de la trame OFDM,
5. Implémenter le démodulateur OFDM à base d'opérateur FFT,
6. Visualiser l'allures des symboles détectés en sortie du démodulateur OFDM. Pour cela tracer la partie réelle du signal en fonction de la partie imaginaire (Montrer le résultat obtenu à vos encadrants).
7. Faire vérifier à vos encadrants qu'en l'absence de bruit l'image reçue est parfaitement décodée.
8. Faire varier la puissance de transmission P_{tx} afin de constater son effet sur la qualité de l'image décodée.

4.3.2 Communications numériques OFDM sur canal multi-trajets en présence de bruit

L'utilisation de l'OFDM sur un canal non sélectif en fréquence n'as pas de sens en pratique. Dans cette section vous allez prendre en compte le fait que le canal est cette fois-ci multi-trajets. Ainsi la version discrète du canal de propagation vue par le récepteur sera désormais modéliser par :

$$h = \sqrt{1/(2*L)} * (\text{randn}(1,L) + 1i * \text{randn}(1,L));$$

où L représente le nombre de trajets discrets équivalents dans la réponse impulsionnelle du canal. Afin de conserver l'orthogonalité temporelle entres symboles OFDM successifs il est nécessaire d'introduire un préfixe cyclique devant chaque symbole OFDM en sortie de IFFT (c'est à dire du modulateur OFDM).

1. Modifier votre code Matlab précédent afin d'introduire un préfixe cyclique devant chaque symbole OFDM. Vous noterez CP la variable utilisée dans votre code Matlab. La taille du préfixe cyclique doit être choisie de manière cohérente vis-à-vis du nombre de multi-trajets,
2. Calculer la perte de débit introduit par l'ajout du préfixe cyclique,
3. Modifier le code Matlab du récepteur afin d'extraire le préfixe cyclique avant la démodulation OFDM,
4. Afin de valider l'ajout et l'extraction du préfixe cyclique, changer le temps de cette vérification le canal, en prenant $h=1$, et vérifier que sans bruit l'image reçue est correctement décodée. Faites vérifier le résultat à vos encadrants.
5. Remettre la version multi-trajet du canal et affichez de nouveau l'image reçue. Que constatez-vous ?

6. Afin de limiter les dégradations introduites par le canal de propagation sur le décodage de l'information binaire, il est nécessaire de mettre en place un égaliseur de canal. Implémentez ce dernier en sortie du démodulateur OFDM en supposant que vous connaissez a priori h .
7. Sans bruit, vérifiez que votre égaliseur fonctionne correctement en observant la qualité de l'image décodée. Faites vérifier le résultat à votre encadrant.
8. En considérant le bruit, faites varier la puissance de transmission et observez les dégradations sur l'image reçue. Comparer ces observations à celles faites lorsque le canal était non sélectif en fréquence $h=1$. Conclusion.

4.4 Partie Finale

Fusionner les résultats de la partie 1 avec ceux de la partie 2 et évaluer le gain de traitement obtenu en passant une partie des traitements sur circuit FPGA.

5 Evaluation

L'évaluation se fait à travers une démonstration lors de laquelle vous présenterez votre travail et les résultats que vous avez obtenu. Une démonstration de la fonctionnalité de votre système doit être effectuée. L'objectif est de présenter ce que vous avez fait, vos choix de conceptions, vos hypothèses de travail, etc... Inutile de présenter les notions vues en cours (codage, OFDM, ...), supposez que vos interlocuteurs maîtrisent ces aspects.

Durant cette présentation/démonstration, des questions vous seront posées à propos de votre travail mais aussi éventuellement sur des aspects attendant au cours. Le comportement (assiduité, ponctualité, etc.) durant les séances de TP rentre en compte dans la note finale.

Le temps attribué à chaque binôme est de 15 minutes (présentation, démo et questions). Vos slides de présentation devront être déposés la veille de votre passage AU FORMAT PDF dans un dossier google drive que vos encadrants partageront avec vous. L'ensemble de vos codes devront également être mis à disposition dans le drive.

Autres détails :

- La présentation se déroulera dans une salle TD, un video-projecteur vous sera mis à disposition.
- Ceux qui ont empruntés des cartes Nexys4 sont priés de venir avec pour la rendre à la fin de la présentation/démonstration.
- L'ordre de passage suit celui indiqué dans le fichier excel du Drive.

Contacts

- Guillaume Ferré - guillaume.ferre@ims-bordeaux.fr
- Camille Leroux - camille.leroux@ims-bordeaux.fr