trx_ofdm

1) Fonctionnalités du bloc

Le bloc trx_ofdm est un bloc spécialisé dans le calcul de transformée de fourrier rapide directe (FFT) et inverse (IFFT). Il intègre par ailleurs des fonctionnalités permettant de réaliser une mise en forme des symboles OFDM entrant (framing - insertion des pilotes et des zéros) ainsi qu?une séparation des pilotes et data des symboles sortant (deframing), ce qui lui permet d?être utilisé soit en émission OFDM (tx : framing + IFFT + insertion d?intervalle de garde) soit en réception OFDM (rx : FFT + deframing)

2) Architecture du bloc

L'architecture du bloc trx_ofdm est composée de deux modules, le c?ur et le wrapper. Le c?ur effectue tous les traitements FFT et IFFT. Le wrapper sert à fournir une interface compatible SoClib CABA avec le module MWMR_controller, disponible dans la librairie SoClib.

2.1) Interface du bloc TRX_OFDM

L'interface proposé est compatible avec le module MWMR_controller sans ajouter des modifications ni adaptations. Un rappel du bloc MWMR ainsi que du protocole FIFO est détaillé dans la Section 7.

Le module OFDM dispose des interfaces suivantes:

- Deux FIFOs d'entrée de données.
- Deux FIFOs de sortie de données.
- Une FIFO d'entrée de configuration.
- Six registres de configuration (seulement en écriture).
- Cinq registres de status (seulement en lecture).

Les deux FIFO de données et celle de configuration sont du même type. Vis-à-vis du MWMR_controller, on peut considérer que le bloc TRX_OFDM dispose de trois FIFO d'entrée et deux de sortie.

2.2) Le c?ur TRX_OFDM

Le c?ur TRX_OFDM calcule des symboles OFDM, et nécessite en entrée du calcul que les blocs complets (de 32 à 2048 mots) soient présents. Pour des raisons de performance du c?ur, et afin de ne pas pénaliser les temps de calcul par les temps de communication, une solution d?implémentation en double buffering apparaît donc nécessaire. Par ailleurs, les chargements et déchargements des buffers sont configurables.

Le choix fait pour le modèle du c?ur OFDM est donc de séparer clairement les 3 phases de chargement/calcul/déchargement par un pipeline d?exécution. Le traitement d?un symbole OFDM aura donc ces trois phases configurables. Cela signifie également que 3 symboles OFDM pourront coexister dans le c?ur répartis dans chacune des 3 phases. La Figure 2 illustre ce fonctionnement.

Le c?ur trx_ofdm peut réaliser des opérations de Framing et de Deframing à des flux de données pour séparer les données et les pilotes. Pour cela, le c?ur dispose de deux FIFOs en entrée (FIFO_in 0 et 1) et deux FIFOs en sortie (FIFO_out 0 et 1).

Les opérations possibles sont:

- A partir d'un flux de données (FIFO_in 0) on peut faire une FFT et une opération de Deframing pour obtenir des données (FIFO_out 0) et des pilotes (FIFO_out 1).
- A partir d'un flux de données (FIFO_in 0) et d'un flux de pilotes (FIFO_in 1), on peut faire un Framing et ensuite une IFFT pour obtenir un flux de données (FIFO_out 0).
- A partir d'un flux (FIFO_in 0) qui contient des données et puis des pilotes, on peut faire une opération de Framing et ensuite une IFFT pour obtenir un flux de données (FIFO_out 0)

L'utilisation des opérations de Framing et Deframing nécessite la configuration des masques de données et pilotes (voir Paramètres de configuration).

2.3) Anoc_copro_wrapper OFDM

Les interfaces de communication entre les coprocesseurs développés au CEA-Leti ne sont pas directement compatibles avec les interfaces CABA proposés par SoClib. Pour permettre de faire une conversion de protocole en gardant une fonctionnalité équivalente, un wrapper à été développé.

L'interface fournie par le module MWMR controller a certaines limitations. Par exemple, le contenu des registres configuration ne peut pas être modifié par le coprocesseur. En autre, si le coprocesseur fait une copie locale de ce registre, il ne peut pas savoir quand le contenu de ces registres été mise à jour.

Pour résoudre ce problème, nous avons conçu un wrapper qui détecte le changement du contenu des registres de configuration pour détecter un changement dans ces derniers.

Pour écrire dans un registre de configuration et s'assurer que la valeur à été correctement écrite, on doit faire deux écritures avec des valeurs différentes. Une première avec une valeur quelconque et puis une deuxième avec la bonne valeur. De ce fait, le wrapper détecte un changement de la valeur et met à jour la valeur du registre interne.

2.3.1) Registres de configuration

Les registres ici détaillés correspondent aux registres de configuration accessible à travers du bloc MWMR (voir Section 7 pou plus de détail).

Le wrapper développé à été conçu pour pouvoir supporter plusieurs coprocesseurs avec plusieurs c?urs de calcul. Par contre, le bloc TRX_OFDM contient uniquement un c?ur de calcul, donc tous les registres pour les c?urs 1 à 3 ne sont pas utilisés.