

## **OBJECTIF**

Synthétiser un filtre FIR-LMS sur Vivado HLS et valider sa fonctionnalité à l'aide d'un test Bench en C.

## **FILTRE FIR-LMS**

À l'aide du code Matlab « FIR\_LMS\_EQ.m » fourni, prenez le temps de bien comprendre le fonctionnement du filtre FIR-LMS. Vous pouvez visualiser la vidéo « Presentation du code FIR LMS.mp4 » que j'ai déposé dans le répertoire Lab A1.

Le filtre à synthétiser sur Vivado HLS devra avoir les entrées suivantes :

- ✓ Valeur d'entrée (correspond à la variable yn dans le code Matlab)
- ✓ Valeur du pas d'adaptation (correspond à la variable mu dans le code Matlab)
- ✓ Valeur de référence pour l'ajustement des coefficients (correspond à la variable inp dans le code Matlab)
- ✓ Valeur du nombre de données utilisées pour l'entraînement du filtre (correspond à la variable N\_TRAIN dans le code Matlab)

Le filtre à synthétiser sur Vivado HLS devra avoir en sortie la valeur réelle estimée des données corrompues par le canal de transmission (correspond à la variable inpest dans le code Matlab).

Votre filtre devra avoir exactement le même comportement que celui dans le code Matlab (phase d'entraînement, phase d'égilisation, nombre de coefficients, etc.).

Une fois le code terminé, synthétisez votre filtre pour vérifier que tout fonctionne.

Pour ce laboratoire, votre filtre FIR-LMS n'a pas besoin d'être optimisé. Le but est simplement de vérifier si votre implémentation du filtre en C est valide. Vous pouvez fortement vous inspirer du filtre FIR fourni dans le tutoriel d'introduction pour y ajouter la partie LMS manquante.

## **VALIDATION DU FILTRE**

À l'aide d'un test Bench, validez le fonctionnement de votre filtre. Pour l'instant, vos variables dans

Vivado HLS (code de la fonction FIR-LMS et du test Bench) doivent être de type float.

Votre test Bench devra comporter les aspects suivants :

- ✓ Le code Matlab fourni qui peut écrire dans des fichiers .txt les valeurs des variables yn, inp et inpest
- ✓ Une section du code C (Test Bench dans HLS) qui pourra récupérer les valeurs de yn, inp et inpest
- ✓ Une section du code C (Test Bench dans HLS) qui appliquera les valeurs récupérées à la fonction du filtre FIR-LMS
- ✓ Une section du code C (Test Bench dans HLS) qui comparera les résultats de sortie obtenus avec Matlab et avec votre fonction afin de valider le bon fonctionnement de votre filtre FIR-LMS

La section Annexe montre comment lire un fichier .txt en C.

## QUESTIONS GÉNÉRALES

- ✓ Quelle est la différence entre la synthèse et la simulation ?
- ✓ Quelle peut être l'utilité d'avoir plusieurs solutions dans un même projet ?
- ✓ Expliquez et discutez les résultats de la synthèse.
- ✓ Une fois la synthèse effectuée, quelle est la différence entre «Latency» et «Interval» ?
- ✓ Quelle est la différence en termes de ressources utilisées et de fréquence maximale d'horloge lorsque vous synthétisez votre filtre avec des variables de type float et de type int?

## ANNEXE

Pour lire un fichier .txt en C (exemple avec la variable yn) :

```
FILE *fin;
float input[N];

fin=fopen("yn.txt", "r");

for(int i=0; i<N; ++i){
    fscanf(fin, "%f", &input[i]);
}

fclose(fin);
```