UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

RAPPORT LABO A1

PRÉSENTÉ À Dr. Ouameur

COMME EXIGENCE PARTIELLE DU COURS

CONCEPTION EN VLSI (GEI1064)

PAR

PIETRO LACOMMANDE (LACP71050105)

RÉPONSES AUX QUESTIONS GÉNÉRALES

06 OCTOBRE 2025

**TABLE DES MATIÈRES**

[**TABLE DES MATIÈRES** i](#_Toc210767072)

[**LISTE DES FIGURES** ii](#_Toc210767073)

[**QUESTIONS GÉNÉRALES** 1](#_Toc210767074)

[**Quelle est la différence entre la synthèse et la simulation ?** 1](#_Toc210767075)

[**Quelle peut être l’utilité d’avoir plusieurs solutions dans un même projet ?** 1](#_Toc210767076)

[**Expliquez et discutez les résultats de la synthèse.** 1](#_Toc210767077)

[**Quelle est la différence entre «Latency» et «Interval» ?** 2](#_Toc210767078)

[**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?** 3](#_Toc210767079)

[**Résultat** 4](#_Toc210767080)

[**Code** 5](#_Toc210767081)

**LISTE DES FIGURES**

[Figure 1- Sommaire du matériel 1](#_Toc210767082)

[Figure 2- Latence 2](#_Toc210767083)

[Figure 3- "Timing" 2](#_Toc210767084)

[Figure 4- filtre fir avec integer 3](#_Toc210767085)

[Figure 5- Code du header file 5](#_Toc210767086)

[Figure 6- fir partie 1 6](#_Toc210767087)

[Figure 7- fir partie 2 7](#_Toc210767088)

[Figure 8- main 9](#_Toc210767089)

**QUESTIONS GÉNÉRALES**

**Quelle est la différence entre la synthèse et la simulation ?**

La simulation sert à tester la validité de l’algorithme développer en comparant celui-ci avec une implémentation équivalente dans, par exemple, Matlab. La synthèse par contre s’agit du matériel qui sera consommé sur le FPGA sélectionné comme la BRAM, LUT, et la rapidité et « timing ». Lorsque la simulation fonctionne et qu’on confirme la validité de l’algorithme, la synthèse confirme pour sa part si cet algorithme est faisable sur le matériel en soit.

**Quelle peut être l’utilité d’avoir plusieurs solutions dans un même projet ?**

L’utilité d’avoir plusieurs solutions dans un même projet est de pouvoir tester différentes optimisations sans avoir à créer de nouveaux projets à chaque fois. Comme énoncé dans le tutoriel, les optimisations spécifiées dans le fichier de directive ne sont pas reportées aux autres solutions.

**Expliquez et discutez les résultats de la synthèse.**

A black and white table with numbers and numbers

AI-generated content may be incorrect.

Figure 1- Sommaire du matériel

On peut voir sur le graphique ci-dessus qu’on utilise très peu le matériel disponible sur le FPGA tel que les look-up tables, la BRAM et plus encore. Il serait bien de se servir plus de ceci pour accélérer les calculs et profiter pleinement des forces du calcul sur un FPGA.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2- Latence

Ça prend 13445 cycles d’horloge pour compléter un calcul de fir avec 2000 échantillons avec cette implémentation qui n’optimise rien. On peut observer que la latence associé au calcul du LMS est plus longue (8005 cycles) contrairement au calcul d’égalisation (5419 cycles). Cela est normal considérant que le LMS nécessite 3 sous for-loops et des calculs plus complexes que l’égalisation même si l’égalisation passe 1800 données en traitement.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3- "Timing"

**Quelle est la différence entre «Latency» et «Interval» ?**

La latence correspond au nombre de cycle d’horloge nécessaire pour produire un résultat du début jusqu’à la fin. Plus précisément, la latence globale dans ce contexte-ci correspond au nombre de cycle d’horloge que ça prend entre la première donnée entrée et la dernière donnée sortie. Lorsqu’on connait la fréquence de l’horloge il est possible de déduire le temps total.

L’interval correspond au nombre de coups d’horloge nécessaire avant de pouvoir commencer un nouveau traitement de donnée.

**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?**

L’utilisation de valeur de type entier n’est pas demandée dans le cadre de ce laboratoire, cependant il est clair que les variables de type entier sont moins complexes à calculer et sont donc plus rapide à calculer et prennent moins de ressources. Par contre, les variables de type float sont parfois nécessaire pour obtenir une bonne précision de calcul.

Rapidement, on peut observer que ça prend de ressource avec le int et environ 4000 moins de cycle de calcul.

A screenshot of a computer

AI-generated content may be incorrect.

Figure - filtre fir avec integer

**Résultat**

J’ai créé une fonction qui compare deux tableaux avec une tolérance de 0.01 entre les valeurs, et cette fonction retourne le nombre de fois qu’une valeur du test ne respectait pas cette tolérance. Les tests ont tout passé :  
A screenshot of a computer program

AI-generated content may be incorrect.

**Code**

fir.h :

A black screen with colorful text

AI-generated content may be incorrect.

Figure 5- Code du header file

Ce fichier sert à déclarer la fonction fir. Elle prend un tableau de 2000 valeurs en entrée, les deux valeurs de mu comme dans le code Matlab, un autre tableau de 200 valeurs pour l’apprentissage des valeurs, et un dernier tableau qui servira comme sortie. Les tableaux y\_in et ref sont const car leur valeurs ne doivent pas être altérer comme qu’ils sont des entrées. J’ai aussi ajouté un define pour valider les pointers nulls.

Fir.c :

A screen shot of a computer program

AI-generated content may be incorrect.

Figure - fir partie 1

A screen shot of a computer program

AI-generated content may be incorrect.

Figure - fir partie 2

Dans la fonction fir, on commence par valider les pointeurs et on set la valeur qui sera retourné comme -1 comme mécanisme de défense robuste. Par la suite, la même logique a été suivi dans le code Matlab vers le code en C. Ça prend plus de ligne de calcul et plus de boucle notamment pour le calcul matriciel, chose qui prend seulement une ligne en Matlab. Après validation avec Mr. Poupart, cette implémentation est moins bonne sur une architecture de FPGA car elle utilise plus de ressources et elle nécessite 2000 données à chaque fois que cette fonction est appelée.

Fir\_test.c :  
A screen shot of a computer program

AI-generated content may be incorrect.

Ces trois fonctions ont été créé pour rendre le code plus lisible. Tel que mentionné plus haut, la fonction compareArrays est celle qui teste les données expérimentaux versus leur valeurs théoriques attendus et si un écart est dépassé la variable retournée sera incrémenté à chaque fois.

A computer screen shot of text

AI-generated content may be incorrect.

Figure 8- main

Le code est très lisible dans le main grâce aux fonctions créées. La fonction writeOutputFile est pas nécessaire dans ce contexte comme qu’on compare deux arrays et non deux fichiers text, mais je l’ai quand même laissé là pour consultation.