TP n° 5 : Utilisation du processeur NIOS II avec instructions «spécialisées»

0. Introduction

L'objectif de ce TP est d'utiliser le processeur NIOS II (cœur logiciel) sur une carte FPGA d'Altera pour exécuter des programmes C et mesurer leurs temps d'exécution.
On examinera comment

- compiler une version du processeur à l'aide du logiciel Quartus II
- charger le processeur et l'environnement associé dans le FPGA (charger le fichier de configuration du processeur sur la carte à l'aide de la connexion USB – Blaster)
- charger un programme C dans la mémoire du processeur (charger le fichier correspondant au programme dans le FPGA) et l'exécuter
- écrire des programmes C sous l'environnement IDE, les charger dans le FPGA et les exécuter
- ajouter des instructions spécialisées au processeur NIOS
- mesurer les temps d'exécution des programmes.

1. Développement d'un programme sous l'environnement IDE

Ouvrir le logiciel NIOS IDE
Ouvrir une nouvelle application C : *new* | *CC++ Application* et choisir *Hello_world*. Choisir pour SOC Builder System
C:\altera\kits\nios2\tutorials\niosII_custom_instr_tutorial\niosII_cyclone_1c20\quartus_project\system.ptf
Activer le projet en cliquant dessus.
Dans le menu *Projet* | *Properties* | *C/C++ Build* , sélectionner la configuration *Release* et un niveau d'optimisation (-O2 ou –O3)
Puis faire *Build Projet*.
Puis exécuter le projet avec *Run* | *Run as* | *Nios Hardware*

2. Test des instructions spécialisées ajoutées au processeur NIOS

Dans le TP 4, on a réalisé des instructions spécialisées :

- DECG (a, b) Décalages à gauche d'un octet de mots de 32 bits
- DECD (a, b) Décalage à droite d'un octet de mots de 32 bits
- B2HL (a) Décompactage de deux octets bas en deux demi-mots de 16 bits (extension de zéros)
- B2HH (a) Décompactage de deux octets haut en deux demi-mots de 16 bits (extension de zéros)
- SHL2 : Décalage gauche de deux positions des demi-mots d'un mot de 32 bits
- MIN8 : Minimum SIMD d'octets non signés de deux mots de 32 bits
- MAX8 : Maximum SIMD d'octets non signés de deux mots de 32 bits
- MIN : Minimum de deux mots non signés de 32 bits.
- ADDH : Addition SIMD de deux demi-mots non signés.
- H2B (a, b) et B2HS (a, b) : Compactage de deux mots a et b contenant des demi-mots en un mot contenant 4 octets (version sans et avec saturation).

Ecrire et exécuter un programme pour tester le fonctionnement correct de ces instructions.

3. Filtre min sur une image

Sur un tableau N x N pixels (niveau de gris), on applique un filtre 3 x 3, tel que le point milieu résultat est le minimum des neuf points considérés.
Pour ce TP, on n'utilisera pas des images réelles, mais on considérera des tableaux X[N][N] avec des initialisations quelconques.

Programme C initial

- Ecrire le programme C qui fait correspondre au tableau X[N][N] initial le tableau Y[N][N] correspondant à l'image filtrée. Pour calculer le min, on utilisera les deux versions

a) fonction min qui renvoie la valeur minimale
b) #define min(a,b) ((a<b) ? a : b)

- Insérer dans le programme les mesures de temps. Exécuter chaque version du programme et mesurer le temps d'exécution (CPP) pour les valeurs de N = 32, 64, 128 et 256 .

Programme C avec instruction spécialisée MIN32

Remplacer la fonction min par une instruction spéciale MIN ajoutée au processeur NIOS. Exécuter et mesurer les temps d'exécution (CPP) pour la nouvelle version pour les mêmes valeurs de N = 32, 64, 128 et 256 .
Calculer l'accélération

Programme C avec instructions SIMD incluant MIN8

Ecrire la version SIMD permettant d'effectuer le min sur les 4 octets d'une instruction 32 bits. Exécuter et mesurer les temps d'exécution (CPP) pour la nouvelle version pour les mêmes valeurs de N = 32, 64, 128 et 256 .
Calculer l'accélération

4. Filtre de Gauss sur une image

Sur un tableau N x N pixels (niveau de gris), on applique le filtre Laplacien 3 x 3 ci-dessous

$$1/16 * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Ecrire le programme C qui applique le filtre de Gauss sur une image

- programme C
- programme C utilisant des instructions spécialisées rajoutées au processeur.

Exécuter et mesurer les temps d'exécution (CPP) pour les deux versions pour les mêmes valeurs de N = 32, 64, 128 et 256 .
Calculer l'accélération

5. Annexes (à rajouter)

Mesure des temps d'exécution

Chargement d'une configuration sur le FPGA (processeurs + opérateurs spécialisés)

Modification du processeur et ajout de nouvelles instructions.