TP A4 : Programmation des architectures SoC hétérogènes CPU-FPGA

Contexte :
Nous constatons l'augmentation du niveau de performances des circuits programmables FPGA qui a favorisé le choix de ce type de composants pour une large gamme d'applications. Cette augmentation de performance est le fruit de la croissance continue de la densité d'intégration de ces circuits. Mais elle a engendré un besoin en termes d'outils de conception et de développement. De nos jours, les FPGA sont composés d'unités optimisées et pré-câblées pour l'accélération matérielle de certaines fonctionnalités. Ils intègrent des accélérateurs matériels optimisés tels que des blocs DSP, des contrôleurs mémoires ou encore des processeurs de type Soft-Core. Désormais, ils sont associés, dans la même puce, à des processeurs CPU multicœurs pour constituer des SoC à architecture hétérogène.

Objectif du TP :
Nous choisirons pour ce TP des algorithmes nécessitant la parallélisation de calculs sur une architecture CPU-FPGA afin d'augmenter les performances de traitements. L'évaluation portera sur les mécanismes d'échanges de données entre le CPU et le FPGA. Cette évaluation sera réalisée avec une carte de développement DE1 SoC d'ALTERA en implémentant des opérateurs de traitement répartis entre le FPGA et le CPU. L'évaluation couvrira les aspects fonctionnels, les temps de transfert, les temps de traitement et la précision de calculs.

Pour la mise en œuvre de systèmes à base de ce type d'architecture, nous utiliserons la programmation OpenCL. Les drivers OpenCL et l'environnement runtime d'OpenCL sont installés sur l'ordinateur. Nous utiliserons un outil de génération de code pour des architectures hétérogènes (AOC pour l'architecture Intel FPGA) remplaçant ainsi la conception avec un langage de description de haut niveau tel que VHDL ou Verilog. L'objectif est d'étudier l'interface entre le CPU et le FPGA intégrés sur la même puce dans le système SoC.

Prise en main de la carte et des outils :
Un projet destiné à une évaluation d'un traitement d'images vous sera donné.

1. Analyse du code de la machine "host" (CPU ARM) :

- Analysez le fichier **Makefile**.
- Quelle est la fonctionnalité du programme principal ?
- Pensez-vous que la fonction exécutable sur le processeur ARM est optimisée ?
- Comment se fait la mesure du temps d'exécution ?
- Comment sont transférées les données au FPGA ?

2. Analyse du code de la machine "Device" (FPGA) :

- Pensez-vous que la fonction implémentée sur le FPGA est parallélisée ? si oui, quelle serait l'architecture synthétisée sur l'accélérateur FPGA ?
- Comment peut-on faire la mesure du temps de calcul sur FPAG ?

Utilisation du compilateur pour le processeur ARM :
- Pointez vers le chemin : C:\intelFPGA\18.1\embedded
- Lancez Embedded_Command_Shell.bat

- Allez dans le répertoire contenant le Makefile du projet
- Lancez la compilation du programme avec la commande **make**. L'exécutable (host) est dans le dossier (bin).

Génération du fichier de configuration de l'architecture matérielle sur FPGA :
- Lancez une console avec la commande cmd.
- Il faut s'assurer que quartus et Opencl SDK sont bien installés en lançant la commande suivante : **aoc -version**

En sortie, vous devez avoir la confirmation suivante :

```
-------------------------
Intel(R) FPGA SDK for OpenCL(TM), 64-Bit Offline Compiler
Version 18.1.0 Build 625 Standard Edition
Copyright (C) 2018 Intel Corporation
-------------------------
```

- Il faut pointer vers le dossier (device) contenant le fichier (.cl) et lancer la commande suivante : **aoc device/fichier.cl --sw-dimm-partition -o bin/fichier.aocx --report**

- La compilation et la génération de l'accélérateur FPGA prend une dizaine de minutes ....

Après avoir vérifié que la carte dispose d'une adresse IP valide, des explications seront données pour se connecter au système d'exploitation de la carte, télécharger des fichiers exécutables et évaluer les performances.

- Transférez les fichier (.host et .aocx) sur la carte microSD avec l'outil Filezilla (port 22 avec adresse IP fournie, login: root, pas de mot de passe).
- Avec l'outil Putty : il faut initialiser la configuration OpenCl en lançant le fichier
  init_opencl.sh

En cas d'erreur, il faut lancer successivement les commandes suivantes :

```
export ALTERAOCLSDKROOT=/home/root/opencl_arm32_rte
export AOCL_BOARD_PACKAGE_ROOT=$ALTERAOCLSDKROOT/board/c5soc
export PATH=$ALTERAOCLSDKROOT/bin:$PATH
export LD_LIBRARY_PATH=$ALTERAOCLSDKROOT/host/arm32/lib:$LD_LIBRARY_PATH
insmod $AOCL_BOARD_PACKAGE_ROOT/driver/aclsoc_drv.ko
```

- Lancer le fichier exécutable host (./host)

Evaluation de performances :

3. Analysez les résultats en termes de temps de traitement, ressources utilisées et interface ARM/FPGA.
4. Évaluer les performances du temps de traitement en fonction de la résolution de l'image. Pour cette étude, on peut choisir des résolution adéquates et utiliser la métrique CPP (nombre de cycle par pixel).
5. Décrivez une version séquentielle du programme, procédez à la synthèse et comparez les temps de traitement avec la version parallélisée.

Remarque : on peut explorer l'architecture générée grâce aux outils Quartus et Qsys pour analyser l'architecture RTL du système.