Cours ROB 306

Projet 1 - Accélérateur Matériel avec HLS

Hardware-Software codesign

(Hw-Sw codesign)

Hervé Le Provost

herve.le-provost@cea.fr

Description

Nous souhaitons dans ce projet évaluer les performances en temps d'exécution et en ressources d'une fonction de multiplication de matrices largement utilisée en IA embarquée et traitement d'images/vision en robotique dans 3 configurations possibles : 1. sur processeur généraliste disponible sur PC (Intel, AMD) 2. sur processeur embarqué ARM9 3. en accélérateur matériel sur circuit reconfigurable FPGA XC7Z020 disponible sur carte zedboard.

L'objectif étant de sélectionner la configuration présentant le meilleur compromis performance-cout pour le contexte applicatif.

Nous utiliserons pour ce projet une carte zedboard dans la configuration ci-dessous ou équivalente

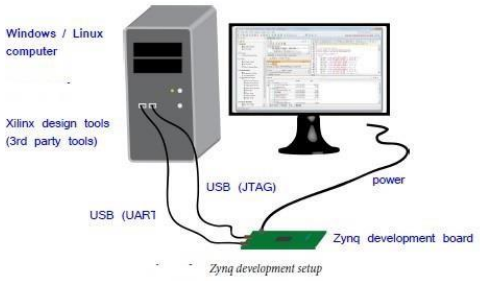


Fig1 –environnement de conception et d'évaluation de performances

La fonction F1 sera donc évaluée selon les méthodes usuelles sur PC et dans les 2 configurations suivantes :

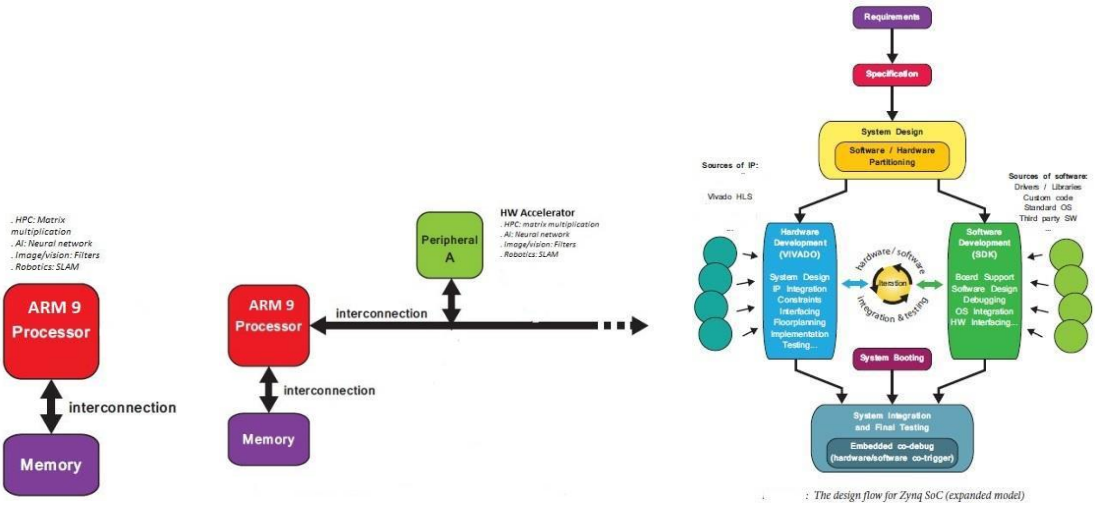


Fig2 – (a) fonction F1 en logiciel embarqué sur processeur ARM9 (b) fonction F1 comme accélérateur matériel sur circuit FPGA Xilinx Zynq (c) flot hw/sw codesign

La fonction choisie sera codée en C/C++ et vous définirez un jeu de tests (ensemble de données de tailles variables) pour la validation en considérant les contraintes mémoires de la zedboard *(512 MB DDR3 –4 GB SDcard).*

Questions

Q1. Evaluation de performances sur PC

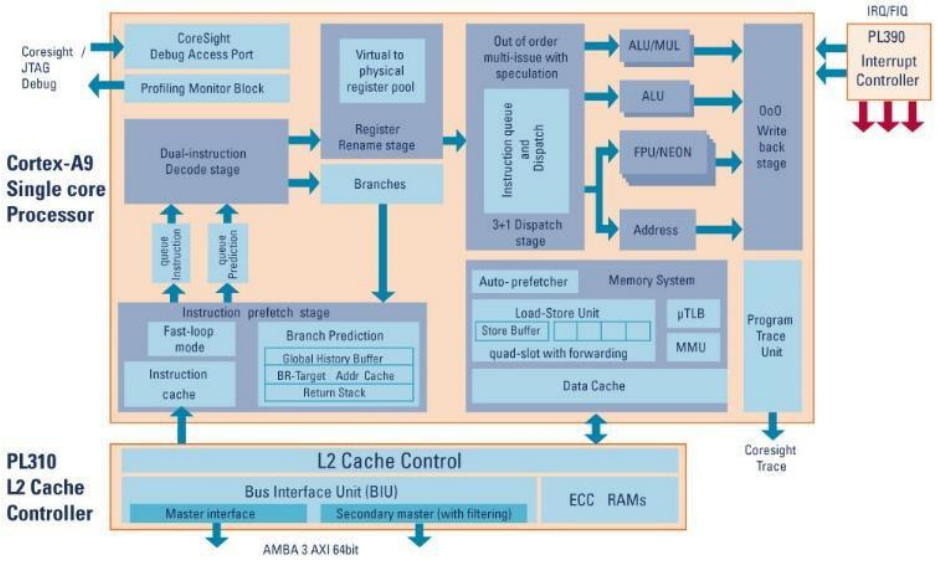
Sur un PC de votre choix vous évaluerez votre code en utilisant le compilateur gcc et en faisant varier les options d'optimisation en compilation (ex. ES 201). Vous pouvez aussi tester plusieurs algorithmes pour la même fonction et voir ainsi l'impact algorithmique. Les processeurs de vos PCs étant des multi-cœurs et si vous êtes familiers avec la programmation parallèle *(multi-threads, OpenMP, etc…)* vous pourrez aussi si vous le souhaitez explorer cette dimension.

Vous tracerez une courbe de performance en temps pour l'ensemble des configurations sélectionnées (algorithme, code, options de compilations).

Vous fournirez les détails de votre configuration d'exécution matérielle (référence processeur (fréquence, énergie), RAM) et logicielle (OS, compilateur)).

Q2. Evaluation de performances sur processeur embarqué ARM9

Vous porterez le code C de votre application sur processeur ARM9 en exploitant l'environnement Vivado 2019 SDK et appliquerez toutes les optimisations à votre disposition dans cet environnement au vu de la microarchitecture du processeur *(caches, prédiction de branchement, superscalaire, unités fonctionnelles, dual-core).*



Vous mesurerez les temps d'exécution de votre application par exécution du code sur le processeur dans la carte zedboard.

Vous tracerez une courbe de performance en temps pour l'ensemble des configurations sélectionnées (algorithme, code, options de compilations).

Vous fournirez les détails de votre configuration d'exécution matérielle (référence processeur (fréquence, énergie), RAM) et logicielle (compilateur)).

Références

- Embedded System Tools Reference Manual
- Generating Basic SoftwarePlatformsReference GuideUG1138 (v2018.2) July 16, 2018UG1138 (v2019.1) May 22, 2019
- Xilinx Software Development Kit (SDK) User Guide - System Performance Analysis

Q3. *Estimation* de performances et de ressources par accélérateur matériel sur circuit FPGA

Vous concevrez un accélérateur de l'application précédente depuis C/C++ en utilisant l'environnement Vivado 2019 HLS et générerez une synthèse de ce circuit.

Vous explorerez différentes directives d'optimisations pour la synthèse HLS et noterez les temps de latence, les débits et les ressources matérielles utilisées du circuit XC7Z020 (logic cells **(% de 85K)**, LUTs **(% de 53200)**, Block RAM **(% de 4,9 Mb)**, DSP slices **(% de 220)**.

Vous tracerez des courbes d'estimation de performance (temps)/ressources du circuit seul pour l'ensemble des configurations sélectionnées (algorithme, code, options de compilations). Confirmez que la courbe a la forme d'une courbe de Pareto au sens de l'optimisation mathématique multiobjective.

Q4. *Mesures* de performances par accélérateur matériel sur circuit FPGA

Vous implémenterez en utilisant Vivado 2019 votre accélérateur sur circuit Zynq et le connecterez au processeur ARM9 en utilisant le réseau AXI.

Vous explorerez différentes directives d'optimisations pour la synthèse HLS et noterez les temps de latence, les débits et les ressources matérielles utilisées du circuit XC7Z020 (logic cells **(% de 85K)**, LUTs **(% de 53200)**, Block RAM **(% de 4,9 Mb)**, DSP slices **(% de 220)**).

Vous tracerez des courbes de performance ***par exécution*** (temps)/ressources du circuit complet pour l'ensemble des configurations sélectionnées (algorithme, code, directives de synthèse et d'optimisations).

Confirmez que la courbe a la forme d'une courbe de Pareto au sens de l'optimisation mathématique multiobjective.

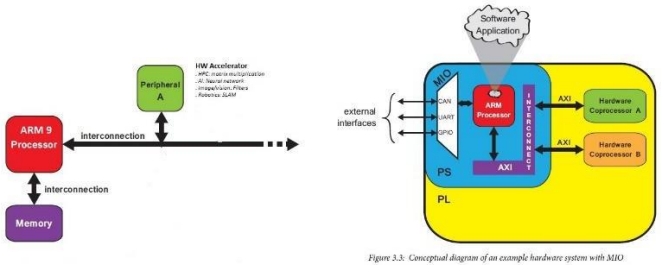


Figure 3.3: Conceptual diagram of an example hardware system with MIO

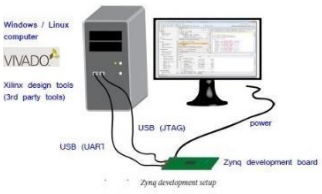
Références

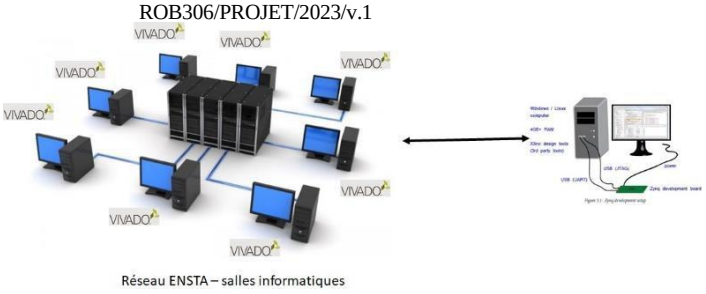
- Vivado Design Suite User Guide High-Level Synthesis - UG902 (v2019.1) July 12, 2019
- Vivado Design Suite Tutorial High-Level Synthesis
- Vivado HLS Optimization Methodology Guide UG1270 (v2017.4) December 20, 2017

Q5. <u>Optionnel :</u> Automatisation (TCL) et parallélisation *(+5 pts)*

Le travail précédent peut être automatisé et permettre ainsi une exploration plus large de l'espace de conception (**DSE** - *Design Space Exploration*). Cette automatisation permet d'améliorer la productivité de conception de circuits complexes (SOC – System on Chip) de manière considérable.

La base est d'utiliser des scripts TCL pour la synthèse et l'implémentation. Le logiciel Vivado 2019 est installé sur l'ensemble du réseau de l'ENSTA et vous pouvez lancer en parallèle N copies de Vivado permettant d'explorer un nombre très important de configurations de SOC.





Réseau ENSTA – salles informatiques

Reference

- Vivado Design Suite User Guide Using Tcl Scripting - UG894 (v2019.1) May 22, 2019
- Vivado Design Suite Tcl Command Reference Guide UG835 (v2019.1) May 22, 2019
- https://www.xilinx.com/video/hardware/using-the-vivado-hls-tcl-interface.html

Q6. Validation Projet et Rapport

Q 6.1 – présentation

Vous soumettrez une présentation par groupe de 4 élèves qui répond aux questions précédentes pour le **16/10/2023** (23H59) dans le répertoire eCampus du cours.

Une validation sur carte sera effectuée le **17/10/2023** de 10H00 a 12H00 après votre soutenance.

Q 6.2 – rapport + fichiers projet

Vous soumettrez un rapport par groupe de 4 élèves qui répond aux questions précédentes et fait la synthèse des résultats obtenus en utilisant le template suivant :

https://www.ieee.org/publications_standards/publications/authors/author_templates.html

pour le **16/10/2023** (23H59) dans le répertoire eCampus dédié à votre groupe.

L'ensemble des fichiers de votre projet devront etre soumis pour le **16/10/2023** (23H59) dans un drive que vous communiquerez directement aux enseignants.

Toute soumission après cette date entraine une pénalité de 3pts par jour de retard et toute soumission avant cette date entraine un bonus de 3 pts par semaine complète d'avance pour un projet ayant une valeur de 14/20 au moins. Pour rappel les modalités d'évaluation sont comme suit : 70% projet 30% examen. La note du projet étant décomposée en : 40 % groupe et 60% individuel.

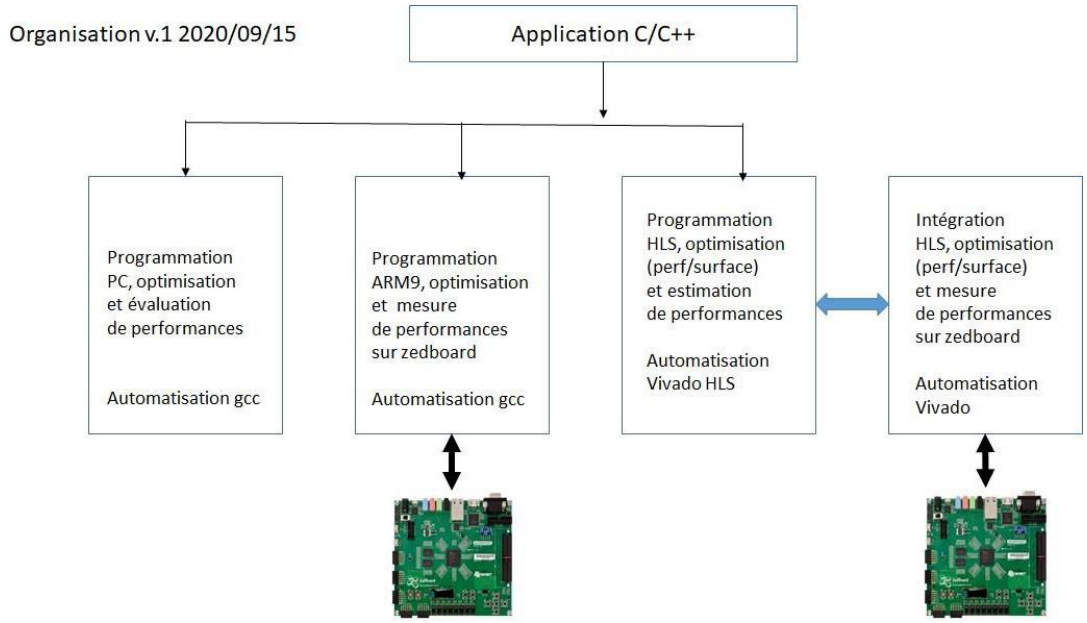
Références

Applications Xilinx

- « A Zynq Accelerator for Floating Point Matrix Multiplication Designed with Vivado HLS » - XAPP1170 (v2.0) January 21, 2016 - Author: Daniele Bagni, A. Di Fresco, J. Noguera, F. M. Vallina
- Zynq All Programmable SoC Sobel Filter Implementation Using the Vivado HLS ToolAuthor: Fernando Martinez Vallina, Christian Kohn, and Pallav Joshi - XAPP890 (v1.0) September 25, 2012

Articles de recherche *(exemples)*

1. HW/SW Co-Design of the HOG algorithm on a Xilinx Zynq SoC JensRettkowski AndrewBoutros DianaGöhringe, Journal of Parallel and Distributed Computing Volume 109, November 2017, Pages 50-62
2. CNN-Grinder: From Algorithmic to High-Level Synthesis descriptions of CNNs for Low-end-low-cost FPGA SoCs, Panagiotis G.MousouliotisLoukas P.Petrou , Microprocessors and Microsystems Volume 73, March 2020, 102990
3. A fast and scalable architecture to run convolutional neural networks in low density FPGAs, Mário P.Véstias[a]Rui P.Duarte[b]José T.de Sousa[b]Horácio C.Neto[b] , Microprocessors and Microsystems Volume 77, September 2020, 103136
4. Design of hand skeleton extraction accelerator for a real-time hand gesture recognition, Seonyoung Lee ; Haengson Son ; Yunjeong Kim ; Kyoungwon Min 2019 International SoC Design Conference (ISOCC)
5. The AXIOM Project: IoT on Heterogeneous Embedded Platforms, IEEE Design & Test, 2019.
6. MulMapper: Towards an Automated FPGA-Based CNN Processor Generator Based on a Dynamic Design Space Exploration, Muluken Hailesellasie ; Syed Rafay Hasan ; Otmane Ait Mohamed 2019 IEEE International Symposium on Circuits and Systems (ISCAS)
7. Automatic Energy-Minimized HW/SW Partitioning for FPGA-Accelerated MPSoCs, Gereon Führ ; Seyit Halil Hamurcu ; Diego Pala ; Thomas Grass ; Rainer Leupers ; Gerd Ascheid ; Juan Fernando Eusse IEEE Embedded Systems Letters 2019
8. Acceleration and implementation of convolutional neural networks based on FPGA, Digital Signal Processing Volume 141, September 2023,
9. MVSym: Efficient symbiotic exploitation of HLS-kernel multi-versioning for collaborative CPU-FPGA cloud systems, Integration Volume 93, November 2023
10. SoC-based real-time SVM classification with integrated training using HLS and PYNQ, Microprocessors and Microsystems Volume 101, September 2023


Organisation v.1 2020/09/15
Application C/C++
Programmation PC, optimisation et évaluation de performances
Automatisation gcc
Programmation ARM9, optimisation et mesure de performances sur zedboard
Automatisation gcc
Programmation HLS, optimisation (perf/surface) et estimation de performances
Automatisation Vivado HLS
Intégration HLS, optimisation (perf/surface) et mesure de performances sur zedboard
Automatisation Vivado