FPGA Acceleration of Domain-specific Kernels via High-Level Synthesis

Abstract

Compute-demanding algorithms in today's applications need to achieve high performance, which is becoming more difficult in general-purpose processors due to the decline of the Moore's law. Domain-specific hardware accelerators can assist general-purpose processors in improving the performance and efficiency while preserving the flexibility. They can accelerate a domain of applications rather than a single application making it possible to use the efficient specialized hardware acceleration techniques in a broad range of applications.

In this thesis, we focus on the development of Domain-Specific Accelerators (DSAs) for a broad domain of applications consisting of biomedical microwave algorithms and Machine Learning (ML) techniques. Although the initial purpose of this research was the development of a biomedical Microwave Imaging (MI) system, The hardware acceleration methods introduced in this thesis are not limited to MI only. We analyzed the recurrent algorithms that are used in these applications to extract their compute-intensive parts that are termed kernels. Then we proposed efficient accelerators for these domain-specific kernels to achieve high performance.

The main computational kernels that are considered in this work are Finite Difference Time Domain (FDTD), Principal Component Analysis (PCA), Support Vector Machine (SVM), and Artificial Neural Networks (ANNs) including Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNNs). For each kernel, we proposed highly efficient hardware accelerators to obtain an optimal performance by considering several factors such as processing time, resource usage, and power consumption. The target hardware platform is Field Programmable Gate Arrays (FPGAs) and the hardware design approach is High Level Synthesis (HLS) which is used to convert a software code written in C or C++ into its corresponding hardware description language. Although several FPGA accelerators have already been presented for the above-mentioned kernels, they have some drawbacks and limitations. Our proposed design methodologies try to address and overcome these limitations.

The proposed hardware accelerator for 3D FDTD considers the impact of polarization currents in dispersive materials, and models the absorbing boundary conditions as Convolutional Perfectly Matched Layers (CPML) in all directions, as opposed to the conventional FDTD accelerators. We use spatial blocking to store a partial block of data while processing the previous block. Local storage of FDTD coefficients and boundary elements, function inlining, and merging the parallel loops are among the other optimization techniques.

The PCA hardware accelerator considered in this work is implemented in FPGA and is designed entirely in HLS. A new block-streaming method is introduced to make the internal PCA computations more efficient. The flexibility of our design allows us to use it for different FPGA targets, with flexible input data dimensions, and it also lets us easily switch from a more accurate floating-point implementation to a higher speed fixed-point solution.%less resource demanding fixed-point solution.

To implement a fast and accurate Support Vector Machine (SVM) classifiers in embedded systems, we propose a flexible FPGA-based SVM accelerator highly optimized through a dataflow architecture. Thanks to HLS and the dataflow method, our design is scalable and can be used for large data dimensions when there is limited on-chip memory. The hardware parallelism is adjustable and can be specified according to the available FPGA resources. The

performance of different SVM kernels is evaluated in hardware. In addition, an efficient fixed-point implementation is proposed to improve the speed.

The last computational kernel considered in this thesis is related to the Neural Networks. Although there are some tools available to generate a hardware design from a high level description of the network (like hls4ml), the selection of network parameters and hardware configurations at the same time is not a trivial task. Although several works have recently addressed the problem of performance co-optimization for hardware and network training, most of them considered either a fixed network or a given hardware architecture. In this work, we propose a new framework for joint optimization of network architecture and hardware configurations, which is based on Bayesian Optimization (BO) on top of HLS. We evaluate our methodology on a network optimized for an FPGA target and show the efficiency of the Pareto set obtained by the proposed joint-optimization approach.