Microprocessors and Microsystems 63 (2018) 199–208

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Design space exploration of multi-core RTL via high level synthesis from OpenCL models

Mehdi Roozmeh*, Luciano Lavagno

Politecnico di Torino Italy

ARTICLE INFO

Article history:
Received 16 February 2018
Revised 20 August 2018
Accepted 21 September 2018
Available online 24 September 2018

Keywords:
Design space exploration
Data center
FPGA
GPU
OpenCL
High-level synthesis
Low-power low-energy computations
Parallel computing

ABSTRACT

As more and more powerful integrated circuits are appearing on the market, more and more applications, with very different requirements and workloads, are making use of available computing power. Designing optimized accelerators that can meet particular requirements has always presented a tremendous challenge to hardware engineers. To do so, designers have to trade off performance for power consumption in a manner such that the final RTL consumes minimum energy to meet the required performance (e.g. FLOPS) target. Moreover, the growing trend towards heterogeneous platforms is crucial to meet time and power consumption constraints of high-performance computing (HPC) applications. The OpenCL parallel programming language and framework enables programming CPU, GPU and recently FPGAs using the high-level synthesis (HLS) methodology. This work presents a design space exploration flow based on execution time, resource utilization and power consumption of OpenCL kernels mapped on FPGAs using the Xilinx high-level synthesis tool chain. Our experiments suggest that the quality of generated solutions, in terms of performance-per-watt, can be determined using analytical formulas prior to implementation, thus enabling fast and accurate DSE by considering on-chip and off-chip sources of parallelism. Moreover, the automated flow suggests design hints to meet a given time constraint within available resources. The proposed technique is demonstrated by optimizing the well known bitonic sorting network from NVIDIA's OpenCL benchmark. Our results report that FPGAs have at least 20% higher performance-per-watt with respect to two high-end GPUs manufactured in the same technology (28 nm). Additionally, FPGAs with more available resources and using a more modern process (20 nm) can outperform the tested GPUs while consuming at least 55% less power at the cost of more expensive devices.

1. Introduction

Energy consumption and power dissipation are significant energy costs for modern datacenters. Intel's acquisition of Altera and FPGA deployment in datacenters and cloud infrastructure by Microsoft, Baidu and Amazon indicate the ever-growing interest of industry leaders to implement a wide range of workloads on FPGA. This option which previously was not interesting due to the very high design costs implied by HDL-based FPGA design, is now becoming interesting thanks to both FPGA architectural improvements (e.g. in terms of on-chip CPU cores and of external DRAM bandwidth) and design flow improvements, namely the broad adoption of High-Level Synthesis (HLS) tools [1].

Intel, for example, is supplying Systems-In-Package including both Xeons and Altera FPGAs. It may also, in the future, integrate them on the same chip. It is also providing its customers with a Software Development Kit (SDK) which supports these heterogeneous SIPs. The SDK allows the programmer to write kernels in OpenCL and then map them uniformly to FPGA resources, in addition to CPUs and GPUs [2].

The OpenCL Programming model has been developed by the Khronos group to overcome the hurdles of programming multi-core and heterogeneous compute platforms [3]. OpenCL enables programmers to develop both close-to-the-metal and portable software. Although OpenCL is a high-level programming language, it provides a low-level abstraction layer that can expose significant architectural aspects of the target hardware, such as massive parallelism and the memory hierarchy. The CPU/GPU based platforms generally have a fixed architecture. While this makes programming easier and compilation times much faster, it is also a limitation because it reduces both the energy efficiency and the on-chip ("local" in OpenCL terms) memory access bandwidth with respect to an FPGA [4].

* Corresponding author.
E-mail addresses: mehdi.roozmeh@polito.it (M. Roozmeh), luciano.lavagno@polito.it (L. Lavagno).

https://doi.org/10.1016/j.micpro.2018.09.009
0141-9331/

SDAccel is a sophisticated tool-chain from Xilinx that supports C/C++ and OpenCL for high-level synthesis targeting Xilinx FPGAs. It starts from software simulation, which only verifies OpenCL functionality, proceeds through the generation of high-quality RTL, whose functionality can be verified through RTL simulation, all the way to placement, routing and bitstream generation.

OpenCL defines a hierarchical memory model that is common between all vendors and can be applied to all OpenCL applications [5]. Global, local and private memories are the main layers of this hierarchy. SDAccel maps them to the FPGA platform as external DRAMs, BRAMs, and registers. SDAccel allows even finer-grained exploitation of the on-chip memory architecture of FPGAs by using directives such as on-chip global memory, multiple AXI buses for kernel global arrays, and partitioned local arrays, which enable a designer to fine-tune the memory architecture and *adapt the RTL architecture to the application, rather than the application to the GPU architecture*.

It has been a long endeavor and evolution in Electronic-System-Level design to generate an optimized RTL from a C-based description. Nowadays, generation of high quality RTL for industrial purposes is possible thanks to modern HLS tools [6–8]. More importantly, available directives such as loop unrolling, memory partitioning and multiple instantiation of compute kernels (multi-core) enable designers to perform broad and in-depth design space exploration (DSE) to realize application driven hardware. In this work we call processing cores the computational units that can be instantiated multiple times on an FPGA, subjected to resource and off-chip memory bandwidth constraints. It is similar to a CUDA core on a GPU and it can be synthesized from an OpenCL work-group. In fact, fine and coarse-grained parallelism of generated RTL are feasible by directing the compilation flow of C-based model using customizable optimization options provided by modern HLS tools to drive optimum solution with highest performance per watt.

2. Background and related work

This section covers design space exploration techniques that were proposed in the past. Afterwards, we will propose an automated DSE flow for multi core RTL generation based on high level synthesis form OpenCL using the SDAccel development environment.

Various DSE approaches have been studied and proposed in the literature. A Pre-RTL power-performance simulator, called Aladdin, is proposed in [9] which enables rapid design space exploration of accelerators with high accuracy in the early stages of design. The Authors of [10] suggest that careful exploration of all solutions can result in area efficiency by proper partitioning of multidimensional arrays and unrolling nested-loops to reduce the area overhead caused by bank switching. An automated DSE flow is proposed in [8] to obtain a Pareto-optimal curve (performance versus area) of an application mapped on FPGAs using HLS methodology. Similarly, an HLS-based DSE approach is discussed in [11] based on user defined area and time constraints which suggests the best RTL solution based on the design requirements. The HLS enables designers to select the bit-width of variables in the behavioral description, hence the author of [12] describes a method to perform DSE for FPGA by controlling the amount of resource sharing using automatic bit-width controller of HLS tool. A Parallel and multi-threaded method for finding the optimum micro-architecture for a given SystemC model is presented in [13]. The author suggests a DSE flow to minimize the size of the design for a given target latency.

As HLS tools are becoming more mature, the demand of C-based intellectual property (IP) blocks is increasing. Synthesizable C-based IPs require DSE at micro-architectural level. The authors of [14,15] presents an automated flow to perform design space exploration for C models generated from Simulink IP block sets, which shields a designer from the need to understand legacy code and obtains a set of Pareto-optimal solutions based on defined constraints. Additionally, macro-architectural trade-offs are considered by wrapping different parts of the C-model into SystemC to derive parallel RTL from behavioral descriptions in Simulink model via HLS. The study in [16] presents a learning-based method for DSE that eases and accelerates micro-architectural modifications using a Random-Forest model which considers all different knobs (micro-architectural choices) such as unrolling and partitioning to select Pareto-optimal solutions for final RTL realization.

Even though important aspects of DSE for the on-chip (FPGA or ASIC) portion of the design are discussed at above-mentioned works, efficient data transfer from off-chip to on-chip memory is a design necessity to drive high performance RTL. Interestingly, an architectural template is proposed in [17] that is capable to consider off-chip memory access parallelism in deep convolutional neural networks. The authors claim that the generated RTL has better performance in comparison to previous works using identical neural networks targeting the same FPGA devices.

In [18–20] the authors discuss the usage of an OpenCL-based synthesis framework targeting FPGAs that encourages many software developers to use them as acceleration platforms. Although using OpenCL as a high-level synthesis input language is not yet mature and significant hurdles should be addressed to achieve high-quality RTL generation, the design speed offered by the new flow more than overcomes any limitations [21].

FPGAs are hence considered as a viable option as an accelerator instead of GPUs especially when energy-per-operation is the main concern. As mentioned above, researchers at Baidu are thus considering FPGAs for accelerating their deep learning models for image search [22]. Microsoft's Bing search engine also uses Altera FPGAs as accelerators in combination with traditional microprocessors from Intel. Keeping in view this market trend and the general perception of the complexity of FPGA programming, two of the major FPGA manufacturers, Intel/Altera and Xilinx, have recently introduced tools to enable the designers to program their respective FPGAs directly using C, C++, SystemC and OpenCL code [23]. There is hence a considerable interest on this topic in the design community. This provided us with a motivation to perform this study.

3. Motivation

Hardware generation starting from OpenCL model is a novel approach that promises designers to map multi core hardware on FPGAs using a C-based model. On-chip memory architecture, number of off-chip memory access ports for each kernel (core), inter and intra kernel optimization (e.g. on-chip buffers, loop-unrolling and function inlining) are important design decisions that can determine the quality of the final RTL in terms of performance-per-watt. Even though each technique could in some cases improve the synthesis result, the careless use of all these techniques to obtain optimum hardware is a vain attempt. On the other hand, choosing the best optimization strategy depends on application, design requirements and synthesis tool. This motivates us to estimate the relative quality of selected solutions using an automated DSE approach to obtain application-dependent hardware and exploit the inherent parallel architecture of FPGAs in a thoughtful manner. This framework compares candidate solutions based on execution time, resource utilization and number of off-chip memory transfers to choose a Pareto-optimal set of golden implementations in terms of energy efficiency and performance. In brief, the novelty of this work relies on the scope of proposed DSE flow which elevates design space exploration complexity by taking to account both on-