

An Automated Tool for Design Space Exploration of Matrix Vector Multiplication (MVM) Kernels Using OpenCL Based Implementation on FPGAs

Jannatun Naher
Electrical and Computer
Engineering Department
North Carolina A & T State
University
Greensboro, NC, USA
jnaher@aggies.ncat.edu

Clay Gloster
Computer Systems &
Technology Department
North Carolina A & T State
University
Greensboro, NC, USA
cgloster@ncat.edu

Christopher C. Doss
Electrical and Computer
Engineering Department
North Carolina A & T State
University
Greensboro, NC, USA
cdoss@ncat.edu

Shrikanth S. Jadhav
Computer Systems &
Technology Department
North Carolina A & T State
University
Greensboro, NC, USA
ssjadhav@ncat.edu

Abstract—OpenCL is a High-Level Synthesis (HLS) based framework that relaxes the programmers considering design time and design complexity. A single OpenCL kernel can be designed in numerous ways by tuning the design knobs. For a given setting of design knobs, logic synthesis of each OpenCL-based design can take several hours. Therefore, the Design Space Exploration (DSE) to search for an optimized design is prohibitive. This paper presents an automated CODE tool to implement an OpenCL-based Matrix-Vector Multiplication (MVM) kernel optimized for throughput and area.

I. INTRODUCTION

Machine learning is used to perform the DSE for an OpenCL based design [1]. In our knowledge, an automated tool was developed for an HLS based design on an FPGAs, but this implementation used Vivado and didn't provide an automated host and kernel code [2].

II. METHODOLOGY

The CODE tool takes function name, matrix dimension, number of random design settings [3], and optimization type from the user as input from a JAVA GUI. A Random Forest python script estimates the resources and throughput. An optimized design setting search was based on the minimum cost for area, throughput, and area-throughput optimization. Finally, this tool generates the host and OpenCL kernel code using this optimized design setting.

III. RESULTS

This research ran this CODE tool to find an optimized design setting using different optimization types for MVM kernel. Fig. 1 shows the percentage of resource utilization estimation using this tool. The JAVA run time of this tool was 2.43 seconds, and the time to run the machine learning algorithm for 1000 design settings was 1.52 seconds, whereas, for actual synthesis, this 1000 designs could take 5.5 months.

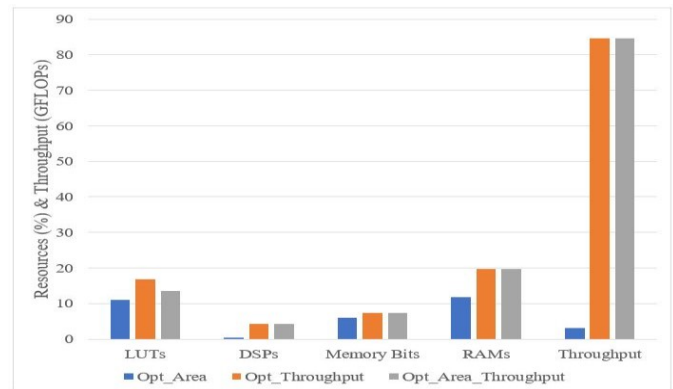


Fig 1. Resource Utilizations (%) and throughput (GFLOPs) in various optimization way for MVM kernel

IV. CONCLUSION

This research developed an automated tool that found the optimized design setting using different types of optimization techniques after performing a DSE using the random-search methodology. For an OpenCL based design using the optimized design setting, this tool generated a host and the kernel code automatically.

REFERENCES

- [1] Q. Gautier, A. Althoff, P. Meng and R. Kastner, "Spector: An OpenCL FPGA benchmark suite," 2016 International Conference on Field-Programmable Technology (FPT), Xi'an, 2016, pp. 141-148.
- [2] G. Zhong, A. Prakash, S. Wang, Y. Liang, T. Mitra and S. Niar, "Design Space exploration of FPGA-based accelerators with multi-level parallelism," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, 2017, pp. 1141-1146.
- [3] J. Naher, C. Gloster, C. C. Doss and S. S. Jadhav, "Using Machine Learning to Estimate Utilization and Throughput for OpenCL-Based Matrix-Vector Multiplication (MVM)," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0365-0372.