

Verification of the Prefix Sum Program in an OpenCL Environment

Thijs Wiefferink
thijs@wiefferink.me, s1366564

University of Twente

Abstract

todo: write

Keywords

todo: write

Preface

This project is the continuation of my bachelor thesis project, in which the verification of the prefix sum algorithm has been started. Since only the 'data race free' part has been proven for the first part of the algorithm (the upsweep), the goal of this project is to prove the complete algorithm data race free, and additionally prove the functionality of the algorithm.

The necessary background information will be included in this report to understand the goal and results of the project, but full details of the Bachelor project can be read in the paper of that project [1].

Contents

1	Introduction	5
1.1	Research domain	5
1.2	Problem statement	5
1.3	Research question	5
1.4	Approach	5
1.5	Report structure	5
2	Prefix sum algorithm	6
3	Verifying permissions	7
4	Verifying functionality	8
5	Discussion	9
5.1	Results	9
5.2	Related work	9
6	Conclusion	10
6.1	Research question answer	10
6.2	Limitations and problems	10
6.3	Research value	10
6.4	Future work	10
7	References	11

1 Introduction

This chapter describes the research domain, shows the problem that is solved, introduces the research questions and explains the approach.

1.1 Research domain

A graphics processing unit (GPU) is a device designed to rapidly manipulate and alter memory to accelerate the creation of images, for example while watching a video or playing a game. However, GPUs are also used more for general purpose computing, which is traditionally handled by the central processing unit (CPU). GPUs are better than CPUs doing parallel execution on large data sets. For example increasing the brightness of an image is easily done by a GPU, since this operation can be done in parallel on all pixels of the image. GPUs are however also used for physics calculations or mining crypto currencies.

todo: verification language, VerCors

1.2 Problem statement

todo: use of verification

1.3 Research question

todo: write

1.4 Approach

todo: verification process and problem solving

1.5 Report structure

todo: introduce chapters

2 Prefix sum algorithm

todo: explain parallel prefix sum, based on bachelorreferaat paper

3 Verifying permissions

todo: verification process of the read/write permissions of the array (initial based on bachelorreferaat, extended to allow functional verification)

4 Verifying functionality

todo: verification process of the functionality of the program

5 Discussion

todo: summary

5.1 Results

todo: result description: verified

5.2 Related work

todo: similar verification projects

6 Conclusion

todo: summary

6.1 Research question answer

todo: works, not for huge programs though

6.2 Limitations and problems

todo: tool limitations, change code for verification process

6.3 Research value

todo: do larger programs, improve tool, concurrency

6.4 Future work

todo: larger programs, improve tool

7 References

- [1] T. Wiefferink. Optimization, specification and verification of the prefix sum program in an opencl environment. 2015.