

DEPARTMENT OF INFORMATION TECHNOLOGY AND  
ELECTRICAL ENGINEERING

Spring Semester 2020

# L<sup>A</sup>T<sub>E</sub>X Report Template

Semester Project / Master Project



Titlepage  
Logo  
Placeholder

Pierre-Hugues BLELLY  
pblelly@student.ethz.ch

May 2020

Supervisors: Matheus Cavalcante, matheusd@iis.ee.ethz.ch  
Samuel Riedel, sriedel@student.ethz.ch

Professor: Prof. Luca Benini, lbenini@ethz.ch

# Acknowledgements

Ceci est un test de mon script

# Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

# Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor. For a detailed version of the declaration of originality, please refer to Appendix ??

Pierre-Hugues BLELLY,  
Zurich, May 2020

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Heterogeneous systems . . . . .	1
1.2	Design Issue with heterogeneous systems . . . . .	1
1.3	Currently Available Workflow for Halide . . . . .	1
<b>2</b>	<b>Preliminaries / Background</b>	<b>3</b>
2.1	Hero . . . . .	3
2.2	Halide Language . . . . .	3
2.2.1	Programing model . . . . .	3
2.2.2	Available Functionnalities . . . . .	3
2.2.3	Porting Halide to new Platforms . . . . .	3
2.3	Compilation Workflow . . . . .	3
<b>3</b>	<b>Design Implementation</b>	<b>4</b>
3.1	Matrix Multiplication . . . . .	4
3.2	Schedule Implementation . . . . .	4
<b>4</b>	<b>Results</b>	<b>5</b>
4.1	Test Setup . . . . .	5
4.2	Comparaison between OpenMp and Halide on the different platforms . . .	5
<b>5</b>	<b>Conclusion and Future Work</b>	<b>6</b>
5.1	First Section . . . . .	6
5.2	Second Section . . . . .	6
	<b>Glossary</b>	<b>7</b>

## List of Figures

# List of Tables

# Chapter 1

## Introduction

### 1.1 Heterogeneous systems

Heterogeneous Systems, relies on different accelerators to achieve the best possible efficiency. Most of them are composed of one General Purpose Core and one or multiple Programmable Many Core Accelerator (PCMA). These systems are interesting because of their energy efficiency, the accelerators are designed to compute a lot of data in parallel, and the host allow for more flexibility and handle the program flow.

Hero is an heterogeneous system developed by the IIS (ETHZ) and the EEES (University of Bologna). This platform is composed of a hard multicore ARM 64 Juno SOC and one pulp cluster (eight RI5CY cores), running on an FPGA. During my project, I used this target to port Halide.

### 1.2 Design Issue with heterogeneous systems

Due to their heterogeneous nature, those systems are difficult to program. The compilation process is complex, requiring multiple passes on different toolchains to distribute the work to the host and the accelerator, and to allocate the memory. And this complexity is also forced on the end user, as he needs to know the target perfectly to make use of all the available resources.

### 1.3 Currently Available Workflow for Halide

Currently Hero support OpenMp, which is an API which "defines a portable, scalable model with a simple and flexible interface for developing parallel applications on plat-



## 1 Introduction

forms from the desktop to the supercomputer" (Ref: OpenMP website). This API has been implemented on hero, and is currently used to develop some test application. But exploring the design space using OpenMp's directive, isn't perfect, and require the developer to adapt it's code to run with a specific schedule. This approach leads to an important developpement time but also an extensive texting process whenever the schedule is changed to ensure that the resulting code works as intended.

Halide is a programming language that was designed to allow the developer to explore multiple design choices quickly by separating the algorithm from the execution schedule. This language was designed to be used in image or array processing applications. Every processing pipeline designed with Halide have two parts. Teh first part consist of the fonctionnal description of the processing kernel, this is the algorithm that will be executed on the arry. And the second part is the schedule of the pipeline. This schedule describe how the algorithm will be executed on the system. This programming model is interesting because the developer can in the first time implement the algorithm without having to take into account the boundaries of the functions or the border effects. Then he can quickly bound the different variables of the pipeline and design it's schedule afterwards. All the constraints will be asserted during the compilation without any intervention from the developer.

# Chapter 2

## Preliminaries / Background

### 2.1 Hero

### 2.2 Halide Language

#### 2.2.1 Programing model

#### 2.2.2 Available Functionnalities

#### 2.2.3 Porting Halide to new Platforms

### 2.3 Compilation Workflow

# Chapter 3

## Design Implementation

### 3.1 Matrix Multiplication

### 3.2 Schedule Implementation

# Chapter 4

## Results

### 4.1 Test Setup

### 4.2 Comparaison between OpenMp and Halide on the different platforms

# Chapter 5

## Conclusion and Future Work

Draw your conclusions from the results you achieved and summarize your contributions. Comparisons (e.g., of hardware figures) with related work are also appropriate here. Point out things that could or need to be investigated further.

### 5.1 First Section

### 5.2 Second Section

# Glossary

**Apple** Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

**Candle** Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

**Guitar** Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

**Monkey** Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

**Snake** Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

# Bibliography

- [1] D. E. Knuth, *The TeXbook*. Addison-Wesley, 1984.
- [2] H. Kaeslin, *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*. Cambridge University Press, Apr. 2008.
- [3] D. Hankerson, S. Vanstone, and A. Menezes, *Guide to Elliptic Curve Cryptography*, ser. Springer Professional Computing. Springer, 2004.
- [4] NIST, *Advanced Encryption Standard (AES) (FIPS PUB 197)*, National Institute of Standards and Technology, Nov. 2001.
- [5] P. Rogaway, M. Bellare, J. Black, and T. Krovetz, “OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption,” in *ACM Conference on Computer and Communications Security*, 2001, pp. 196–205.
- [6] Xilinx. (2011, Nov.) Virtex-6 FPGA Configuration - User Guide. UG360 (v3.4) November 18, 2011. [Online]. Available: [http://www.xilinx.com/support/documentation/user\\_guides/ug360.pdf](http://www.xilinx.com/support/documentation/user_guides/ug360.pdf)
- [7] ——. (2011, Oct.) 7 Series FPGAs Configuration - User Guide. UG470 (v1.2) October 26, 2011. [Online]. Available: [http://www.xilinx.com/support/documentation/user\\_guides/ug470\\_7Series\\_Config.pdf](http://www.xilinx.com/support/documentation/user_guides/ug470_7Series_Config.pdf)
- [8] Wikipedia, “Isaac newton — wikipedia, the free encyclopedia,” 2012, [Online; accessed 1-October-2012]. [Online]. Available: [http://en.wikipedia.org/w/index.php?title=Isaac\\_Newton&oldid=514997436](http://en.wikipedia.org/w/index.php?title=Isaac_Newton&oldid=514997436)