Bachelor Informatica

Universiteit van Amsterdam

Applying FPGAs in computer architecture and organization education

Matthijs Bos

May 14, 2015

Supervisor(s): Toto van Inge (UvA), Taco Walstra (UvA)

Signed: n.a.

Abstract

TODO Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

CHAPTER 1

Introduction

Computer architecture and organization is considered an important subject in undergraduate information science curricula, such as computer engineering, computer science and software engineering. It is the field of study where one acquires understanding of a computer's central processing unit (CPU). The subject's body of knowledge ranges from a low level understanding of a CPU's internal workings, design and surrounding systems to a more abstract view that considers a CPU from a software point of view [1, p.60]. Depending on the curriculum, computer architecture and organization courses are often succeeded by more advanced courses that focus on digital design and hardware description languages (HDL). These more advanced subjects are often taught in a later stage of the curriculum.

One often attempts to reinforce the theoretical understanding of computer architecture through a practical experience. Computer architecture simulators are a popular tool in facilitating this practical experience and many implementations exist, such as [2]. In this virtual approach however, students lack a physical view of computers [3, p.1]. The need for a more practical, hands-on experience is supported by [4], which claims that traditional teaching methods are inappropriate for modern, 'digital native' students.

Several computer architectures have been developed for the purpose of computer architecture education [3], [5]–[8]. Although not especially designed for the purpose of education, other simple architectures could be leveraged for this purpose as well [9].

Field programmable gate arrays (FPGA) are a class of integrated circuits that are user configurable and can be used to implement a digital system's design in 'real' hardware. Designs are usually defined using a HDL such as Verilog HDL or VHDL and programmed on the device. For some time now, the costs and capabilities of these devices have reached a point such that they have become useful for educational purposes and have proven to be a successful tool in courses that teach students the subjects of digital design and HDLs.

Learning digital hardware design using FPGAs is highly dependent on HDLs. Learning HDLs however, is generally considered a challenging task. Students that already have experience with other programming languages such as C or Java have trouble adopting HDLs' parallel behaviour, since it differs a great deal from the behaviour in these imperative languages. Furthermore, the application of HDLs requires a certain level of knowledge on the subjects of digital systems and physical fundamentals of computers [3, p.2].

Many solutions exist that translates 'traditional' code such as C (dialects) into a HDL format. It is unknown whether these technologies have been applied for the purpose of education. The level of abstraction that these technologies add on top of HDLs probably makes it unusable for computer architecture education.

A number of educative methods have been developed that incorporate FPGAs as a tool to teach the concepts of computer architecture and organization [3], [5], [7], [10]. These methods however, all teach or assume some knowlegde on the subject of HDLs. These methods have a common approach in which students implement some (part of a) microarchitecture using HDLs and verify their work using an FPGA. The approach presented in [8] presents a method that utilizes FPGAs in an interactive way without using HDLs.

CHAPTER 2

System Architecture

The system will consist of a component running on the FPGA and PC software controlling the device. Both

2.1 Design considerations

CHAPTER 3

Implementation

3.1 FPGA VHDL implementation

3.2 PC Software implementation

CHAPTER 4

Experiments

4.1 Experiment setup

4.2 Experiment results

CHAPTER 5

Discussion

5.1 Future Work

CHAPTER 6

Conclusions