A PROCESSOR DESIGN TOOLKIT FOR AN OPEN-SOURCE FPGA TOOLCHAIN

April 18, 2017

By GROUP 101.

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A concept Paper submitted to the school of computing and informatics technology
For the Study of a processor design toolkit for an open-source FPGA toolchain

Supervisor
Enerst Mwebaze
Department of Computer Science
School of Computing and Informatics Technology, Makerere University

GROUP MEMBERSHIP:

Name	Reg Number	Student Number
Kiwala Martin	15/U/485	215000474
Kyagera Sulaiman	15/U/6849/PS	215008420
Tugumisirize Enoth	15/U/13228/PS	215007998
Akora Brian Marco	15/U/79	215001425

1 Introdution

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing.
For this reason, it is referred to as Field Programmable.
Programming frameworks for reconfigurable architectures are highly dependent on the structure, the hardware granularity and the language used.
Although far from being an ideal hardware description language, Verilog which is very similar to C was selected as an appealing entry-point for the configuration of reconfigurable processors since the first architectures.

2 Backgroung to the problem

The use of FPGAs outside the industry has previously been limited due to the high cost and complexity of their associated development software.
And also there has been a very big problem of people learning the hardware description languages like VERILOG and VHDL because of the lack of tutorials to teach such languages which are the only languages that an be used to configure the FPGAs.
However this is changing, with an open-source toolchain for popular FPGA devices becoming available in the previous years(the equivalent of Linux in the OS world).

3 Problem Statement

This project will build on the existing toolchain to develop additional software necessary to allow students to design and explore simple processor designs on a custom FPGA development board.
The current toolchainn will require the use of VERILOG which is a hardware description language as mentioned in the introduction above.
VERILOG is very powerful but also very general therefore this project will develop a higher level language(possibly a graphical language) focused on the development of simple processor designs from a small number of standard components. (such as RAM, multiplexers and registers).

4 Objectives

4.1 Main Objective

We seek to develop additional software to the existing toolchain inorder to enable students to explore/manipulate and develop simple processor designs on a custom FPGA development board.

4.2 Other Objective

Alongside the main objective we also seek to develop an FPGA from scratch and have a newly constructed circuit board.
we also seek to use the available two hardware description languages i.e VERILOG and VHDL to configure the different custom FPGAs.

5 Methodology

To begin with, we shall use Solaris a Unix Operating System backed by a SPARC workstation from SUN. We considered the use of a Virtual Machine and opted to explore the installation of Linux in a Virtual machine.
The Preliminary investigations into doing this project revealed that we had three ways to do this: Parallels Desktop, VMware Fusion and Virtual Box and chose to go for VMware Fusion. With the two major FPGA vendors Altera and Xilinx, we opted for Xilinx; largely due to the Micro Blaze soft processor. The development board of choice is to be between the ML403 board that has a Virtex-4 FPGA with a PowerPC 405 core and the Spartan 3 based board. The embedded processor to be used will be the Micro Blaze soft processor core: a 32-bit Harvard RISC architecture optimized for vendor-of-choices (Xilinxs) FPGAs. The basic architecture consists of 32 general-purpose registers, an Arithmetic Logic Unit, a shift unit and two levels of interrupt.
The FPGA design software packages we will use are bundled on the ML403 board and are called Integrated Software Environment (ISE) and Embedded Design Kit (EDK) and will be used to design and implement the additional software intended for exploring simple processor designs.
The logic simulators required saw us explore the one provided by Xilinx software which is Verilog and VHDL Simulator but this runs under only Windows XP. So we opted to use Incisive Unified Simulator but only after obtaining an evaluation license from Cadence The Synthesis tool to be utilized will be from Xilinx but XST Synthesis tool seems like a sufficient one to use. We will also use C programming Language substantially as most of the Xilinx software device drivers are written in C.

6 Outcomes

- At the end of this project students will be able to develop simple processor designs using the higher level language developed in this project.
- Students will now be able to manipulate and change any processor design and be able to customize it to whatever logic design one feels best.
- Students will be able to develop an FPGA from scratch and be able to configure it in any logic way that best suits them.

References

[1] a stack-based virtual machine for physical devices 21st October 2016 by Prof. Alex Rogers
http://www.cs.ox.ac.uk/people/alex.rogers/stack/Stack.pdf

[2] Desarrollo de un enlace inalmbrico para telefona fija empleando una FPGA. Final Project at the ETSII, University of Valladolid, Spain, 2006.

[3] FPGA CPU Links.
http://www. fpgacpu.org/links.html, September, 2002.

[4] https://www.quora.com/advantages-and-disadvantages-of-Verilog-or-VHDL

[5] https://www.sparkfun.com/news/1203