University of Toronto
Department of Electrical and Computer Engineering
ECE496 Design Project

Virtual FPGA fabrics
Implementation of a Virtual FPGA Architecture

Group Progress Report

January 17, 2012

Neil Isaac
n.isaac@utoronto.ca

Keyi Shi
keyi.shi@utoronto.ca

Project ID:	2011017
Supervisor:	Jason Anderson
Administrator:	Ross Gillett
Section:	#7

Contents

1 Project Overview

The goal of this project is to provide a physical platform for researchers to carry out FPGA architecture studies. We are developing an *Overlay FPGA*[1], an FPGA circuit working on top of a commercial FPGA product. Our project will consist of the Overlay FPGA circuit, and software that enables the user to translate their Verilog code into a *bitstream*[2] that can be programmed onto the Overlay FPGA. The goals of our project can be reviewed in Appendix C.

The validation of the design will involve testing a set of benchmark circuits by placing and routing them with VPR, then transferring them to the FPGA overlay. The circuits can then be tested for correct behavior, confirming that the overlay design can be correctly programmed using VPR output, and that the inputs and outputs to the design are functioning properly.

[1] We also refer to the Overlay FPGA design as a *Virtual FPGA*.

[2] A bitstream is a file containing the encoded data required to program a circuit an FPGA.

2 Group Progress Summary

Our project is progressing well and we are confident that we can meet our deadlines. Our initial schedule was very optimistic, leaving a large amount of slack time at the end of the term. We have adjusted the gantt chart to reflect more realistic expectations as we progress through the project and experience the true complexity of the individual tasks. The updated gantt chart is included in Appendix A, and the old version is in Appendix B. Aside from the schedule change, our goals, work plan, and work allocation has not changed.

In order to build a working *Overlay FPGA*, we need to produce the following:

- Verilog for the individual FPGA building-blocks: the logic cell, logic block, connection block, and the switch block. (Keyi)
- Verilog to connect the building block components together in a grid and feed the programming signals through them. (Keyi)
- Verilog circuit to interface with the UART to enable serial programming of the Overlay FPGA. (Neil)
- Scripts to run the third-party tools which will process an input circuit (from the user) and produce a valid placment and routing. Some adjustments need to be made so the tools can read each others' outputs. (Neil)
- Software to convert the placement and routing into a programmable bitstream. (Neil)

The basic components of the overlay have been completed and are individually functional, but extensive debugging is required to get it all working together.

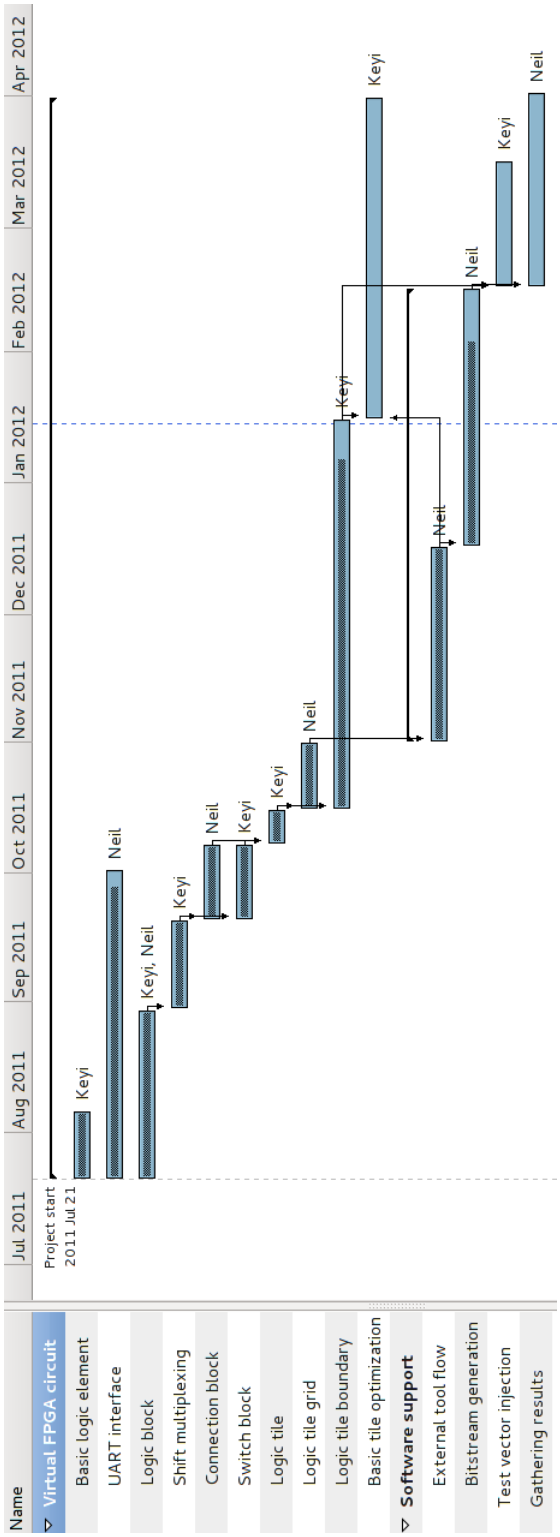
The software is also complete, but can't be fully tested until the hardware is fully functional. Partial validation was done on the software by manually reading the bitstream file it produces.

Once we complete this debugging, we have a number of improvements planned to improve the quality of the virtual FPGA (in terms of usability to researchers, as well as its efficiency.) These improvements are not necessary for a proof-of-concept, or for a fully functional demo.

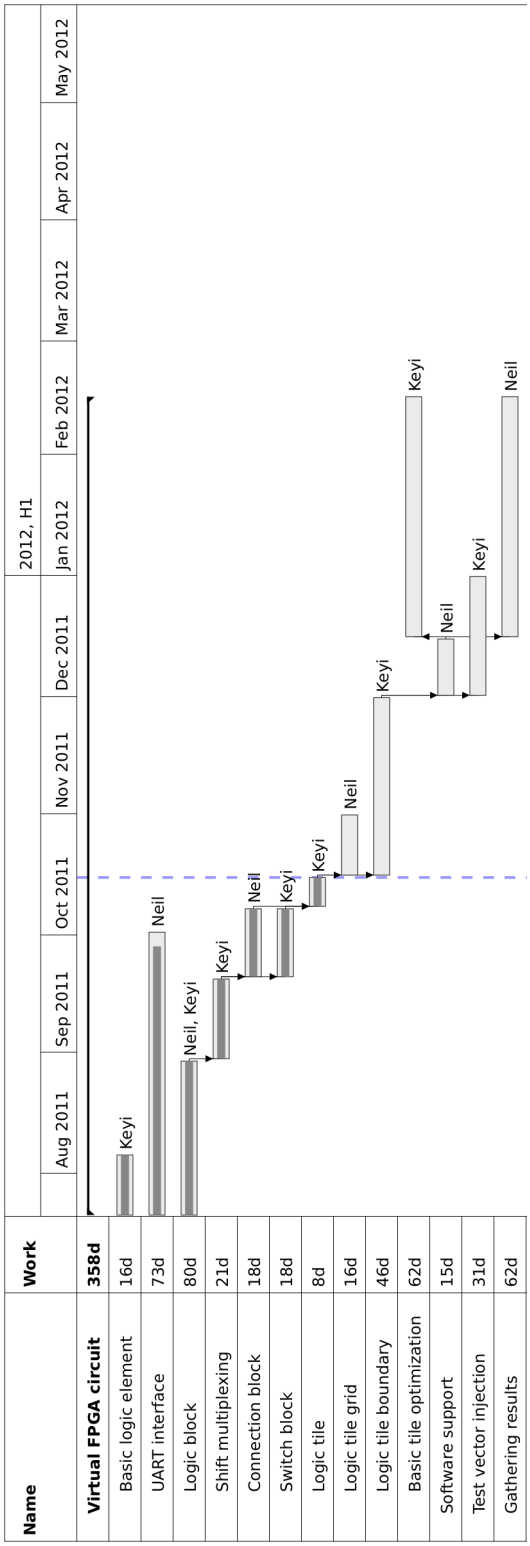
We are confident that we can complete these remaining tasks before the design fair.

Appendices

A Updated Gantt Chart



B Gantt Chart from Project Proposal



C Project Goals and Requirements

C.1 Functional Requirements

An Overlay FGPA circuit that will:

- Work on a commercially available FPGA chip.
- Be re-programmable over a serial interface after being flashed to the FPGA.
- Have a tunable number and arrangement of logic cells, and have tunable connectivity parameters.
- Support inputs to and outputs from test circuits programmed onto the Overlay FGPA.

A software program that can:

- Translate VPR placement and routing data for a test circuit into a bitstream for the Overlay FGPA.
- Program the bitstream onto the Overlay FGPA to implement the circuit.

C.2 Constraints

- The overlay circuit must support at least 3000 logic cells[3] in order to accommodate the *"Golden 20"* MCNC benchmark circuits[4] commonly used in FPGA research.

C.3 Objectives

- Be compatible with a family of commercial FPGAs that are available to researchers.
- Use the native logic cells in the physical FPGA directly in the overlay FPGA design to reduce area and latency.
- Be fast enough that it outperforms software emulation of most test circuits.

---

[3] 3000 logic cells was chosen as the minimum target because the largest of the "Golden 20" circuits, *"s38417"* requires 2567 6-input logic cells.

[4] The "Golden 20" MCNC circuits are available in BLIF format at http://www.ece.ubc.ca/~julienl/benchmarks.htm.