ECE496Y Project Proposal Grading Form - Draft A

Project ID	2011017	Supervisor(s):	Jason Anderson	Section:7
Project Title:	Virtual FPGA fabrics: Implementation of a virtual FPGA architecture			Administrator: Gillett

Administrator's Evaluation

Required Sections for Draft A	Excellent	Good	Satisfactory	Marginal	Poor	Missing/ Unacceptable
Background and Motivation: A clear introduction and description of the design problem and its context including the state of the art, and existing work and technology. A clear motivation for the work.	□	□	□	□	□	□
Project Goal and Requirements: Clear project goal and requirements, links to the original design problem. Problems (circle all that apply): Unverifiable/unmeasurable requirements, unnecessary implementation details	□	□	□	□	□	□
Validation and Acceptance Tests: Clear description of tests, explanations of how tests demonstrate that the project goal and requirements have been achieved.	□	□	□	□	□	□
Possible Solutions and Design Alternatives: Clear description of possible approaches and discussion of design trade-offs.	□	□	□	□	□	□
Feasibility Assessment: Clear description of required skills and resources and key risks.	□	□	□	□	□	□

Optional Sections (if included)

System-level overview: A clear description of the top level design.	□	□	□	□	□	□
Module-level descriptions: A clear description of each key component of the design.	□	□	□	□	□	□
Assessment of Proposed Design: A clear description of how the solution addresses the technical problem. A clear discussion of the strengths, limitations, and trade-offs of the design.	□	□	□	□	□	□
Work breakdown structure (circle specific problems) persons not assigned to tasks, tasks not numbered	□	□	□	□	□	□
Gantt chart (circle specific problems) missing task numbers or descriptions, missing persons assigned to each task	□	□	□	□	□	□
Financial plan	□	□	□	□	□	□

Overall Quality of Document

Presentation (circle specific problem areas) Grammar, spelling, clarity, style, other (specify):	□	□	□	□	□	□
Content (circle specific problem areas) Organization, logical coherence, substance, references, other (specify):	□	□	□	□	□	□

Other Problems: Report late, other (specify):

Approximate grade out of 10 (based on required sections) 5 6 7 8 9 10	**Administrator's Signature:**	**Note to Supervisors:** This draft is <u>not</u> graded. Comments and approximate grade are provided only to give feedback. Actual grading done for the final draft only.

Administrator's Comments (also see comments in report)

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering

University of Toronto

ECE496Y Design Project Course

Group Project Proposal (Draft A)

Title: Virtual FPGA fabrics: Implementation of a virtual FPGA architecture

Project I.D.#:	2011017	
Team members: (Select one member to be the main contact. Mark with ‘*’)	Name: Neil Isaac *	Email: n.isaac@utoronto.ca
	Keyi Shi	keyi.shi@utoronto.ca
Supervisor:	Jason Anderson	
Section #:	7	
Course Administrator:	Gillett	
Date:	September 20, 2011	

Executive Summary

Academic studies of Field Programmable Gate Array (FPGA) chip architecture rely on simulations, as commercial FPGA chips contain proprietary designs that make their underlying architecture inaccessible to academic researchers. The goal of this project is to provide a physical platform for researchers to carry out FPGA architecture studies. The finished design should be financially accessible and capable of running common benchmark circuits.

The proposed design will use an overlay circuit to implement a virtual FPGA on an existing, commercially available FPGA chip. Using a commercial FPGA as the physical medium for this project makes the design cheaper and more accessible to researchers, as they may have an appropriate FPGA chip already.

The overlay design may use architectural features that may specific to a particular FPGA family. Although this will limit the models FPGA chips the overlay circuit can be implemented on, it should reduce the design's area overhead and improve its timing characteristics.

We have selected the Xilinx Virtex 5 FPGA as our development platform. Any architectural features we use on the Virtex 5 will be forward-compatible with all current-generation Xilinx FPGA products, allowing the researcher to use a variety of FPGAs.

The overlay FPGA is intended to be used in conjunction with VPR, a free, open-source placement and routing tool that is used in FPGA architecture research. As such, 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, and that the inputs and outputs to the design are functioning properly.

Contents

1 Project Description

1.1 Background and Motivation

Academic researchers who study Field Programmable Gate Array (FPGA) design commonly use variations of an FPGA design architecture, described by Kuon et al[1], which we will refer to as the *Academic FPGA Model*. While FPGA chips are available from a variety of commercial vendors, their design is largely proprietary, making their architectures difficult to study.Furthermore, there is no existing physical implementation[1] of an Academic FPGA Model.

One of the tools used in FPGA architecture research is VPR[2]. VPR is a free, open-source placement and routing tool that accepts a wide variety of architecture parameters. It is not currently possible to realize VPR output on a commercial FPGA.[2]

As such, Computer Aided Design (CAD) researchers who work on placement and routing algorithms for FPGA designs are presently limited to using simulations to evaluate or verify their work. They may be interested in testing circuit realizations on a physical medium.

1.2 Project Goal

The goal of this project is to produce a circuit design based on the Academic FPGA model. Researchers will be able to use the circuit to study FPGA architecture and CAD algorithms with circuits produced by VPR.

[1] Alex Brant is also developing a comparable FPGA overlay platform with Prof. Guy Lemieux at University of British Columbia.

[2] A technology-mapped input netlist for VPR can be converted to an Altera Quartus VQM netlist file using *nettovqm*[3], but the placement and routing can not be converted.

1.3 Project Requirements

1.3.1 Functional Requirements

Researchers must be able to:

- implement the overlay FPGA circuit on commercially available FPGA chips,
- tune the number, arrangement, and logic cell connectivity of the overlay FPGA,
- program the overlay FPGA using an output circuit from VPR,
- modify the inputs and outputs of the overlay FPGA.

1.3.2 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.

1.3.3 Objectives

- Be compatible with accessibly priced commercial FPGAs.
- Take advantage of the underlying FPGA architectural features in the overlay FPGA design to reduce area and latency.

1.4 Validation and Acceptance Tests

1.4.1 Functional validation

To ensure that the overlay FPGA circuit design is functional, we will:

1. select and use a benchmark circuit commonly used to test VPR,

[3] 3000 logic cells was choosen as the minimum target because the largest of the "Golden 20" circuits, *"s38417"* requires 2567 6-input logic cells[4].

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

2. configure VPR to match our architecture and dimensions,
3. place and route the benchmark circuit with VPR,
4. convert the VPR output into a bitstream for the overlay FPGA,
5. load the bitstream onto the overlay FPGA, then
6. test the functionality of the benchmark circuit running on the overlay FPGA.

This test procedure ensures that:

- circuits can be implemented using VPR output,
- circuits can be transferred correctly to the overlay FPGA, and
- inputs can be set and outputs can be read.

The exact verification process for inputs and outputs will depend on the benchmark circuit's intended function. We will need to develop an appropriate testing mechanism for the benchmark circuit.

1.4.2 Size and overhead validation

To ensure that the overhead is low enough that the overlay FPGA can fit useful circuits, we will test it using the *"Golden 20"* MCNC benchmark circuits. For each circuit, we will:

1. run synthesis and technology mapping using ABC,
2. run placement and routing using VPR configured, and
3. confirm that VPR can place and route the benchmark circuit using the number and arrangement of logic blocks that we can fit.

We discuss risk mitigation for high overhead in section 3.1.2.

1.4.3 Validation of improvements from architectural optimizations

To evaluate the benefits of utilizing architectural FPGA features of the host FPGA board, an alternate design can be created that implements the same functionality using only standard verilog. The size and timing of the two designs can then be compared to measure any efficiency gained by the design that uses special FPGA features.

For example, in select Xilinx boards, a lookup table can be used as a 32-bit shift register; the same function could be implemented in plain Verilog using multiplexers and flip-flops, but is expected to be slower and consume more area. The two equivalent circuits can be compiled separately in order to compare their resource use and limiting timing path. Because the utilization of architectural features limit the design to specific board families, they must be shown to enhance the circuit efficiency of this project in order to justify their use.

2 Technical Design

2.1 Design Alternatives

2.1.1 Implementation medium

The implementation medium for our circuit is a major decision impacting how accessible our circuit will be to researchers. The main criteria are cost, size, and ease of use. The lower the cost of the finished design to the researcher, the better. We must also ensure that the design is large enough to handle circuits the researchers wish to test. Finally, we want to make interfacing with the design's inputs and outputs as hassle-free as possible. The alternatives are as follows:

1. Custom integrated circuit
   - Faster, smaller and more power efficient.
   - High design and manufacturing costs.
   - Lengthy design and manufacturing time-line.
   - Once built, the parameters can't be modified without manufacturing a new chip.
   - Inputs and outputs will require extra circuitry to interface with the circuit.
2. Overlay FPGA implemented on commercial FPGA
   - Researchers may already own a compatible FPGA so they won't need to purchase new hardware.

- Using an FPGA allows the researcher to implement a virtual circuit to interface with the overlay FPGA.
- Need to pick a FPGA platform to target:
  (a) Basic FPGA without using architecture-specific features
    - Circuit will work on most FPGAs from most vendors, so it is the most widely accessible.
    - Can't use architecture-specific features to save area and gain performance.

  (b) Xilinx Virtex 5 or newer
    - Lookup tables can be programmed directly as 32-bit shift registers.
    - Large FPGAs with 330,000 logic cells for Virtex 5[5] will fit a larger overlay circuit. Virtex 6 and 7 feature up to 760,000 and 2,000,000 logic cells respectively[6].
    - Higher cost for researchers.

  (c) Xilinx Spartan 6
    - Lookup tables can be programmed directly as 32-bit shift registers.
    - Smaller FPGA with 150,000 logic cells[6], allowing smaller overlay circuit.
    - Lower cost than Virtex 5.

  (d) Altera Stratix IV or newer
    - Higher cost than Xilinx Spartan FPGAs.
    - Large FPGAs with up to 820,000 logic cells for Stratix IV[7] and up to 952,000 for Stratix V[8].

Developing a custom integrated circuit is far too costly and time consuming for the scope of this project. It was explored as an alternative to illustrate by contrast the necessity of targeting an existing FPGA. We have tentatively selected the Virtex 5 FPGA because our supervisor has numerous development boards and software licenses readily available. We also intend to use the 32-bit shift register functionality that is available in logic blocks in Virtex 5 and newer FPGAs. This feature will allow us to reduce the overhead of the overlay FPGA circuit by directly using the native FPGA's features. This selection limits the use of our circuit to modern Xilinx FPGAs including Spartan 6, Artix 7, Kintex 7, and Virtex 5, 6 and 7.

2.1.2 Configuration mechanism

Various parameters of our circuit, including the number, arrangement, and connectivity of the logic cells will be tunable. There are two alternatives for the implementation of the configuration mechanism:

1. Parameterized Verilog
    - Requires the user to modify values within the Verilog source.
    - Involves more complex Verilog code to accommodate flexible parameters.
2. Software front end to generate Verilog code
    - The front end interface could be easier to use than modifying Verilog code.
    - Front end code will be easier to write than parameterized Verilog.
    - User may need to install a compiler or interpreter to run the software.

We have tentatively decided to use parameterized Verilog because we deemed that the complexity of the configuration in our present design concept does not warrant a front end code generator. If added features or a reevaluation of the design add configuration complexity, we may reconsider this, as this decision could be changed without revising a great deal of work.

3 Work plan

3.1 Feasibility Assessment

3.1.1 Skills and Resources

- Required Software
    - ABC synthesis system to product input for VPR - available online[5]
    - VPR placement and routing to produce input our input data - available online[6]

[5] ABC can be downloaded from: http://www.eecs.berkeley.edu/~alanmi/abc/

[6] VPR 5.0.2 can be downloaded from: http://www.eecg.utoronto.ca/vpr/

  - FPGA vendor tools to implement our circuit: Xilinx ISE - licenses through supervisor

- Required Hardware
  - FPGA development board - currently using Xilinx Virtex 5 from supervisor; Spartan 6 boards are also available from supervisor.
  - Host computer to program the physical and overlay FPGAs - personal laptops
  - Serial cable and USB adapter to interface with overlay FPGA - available at electronics stores

- Required Skills and Knowledge
  - Verilog circuit design skills - gained from previous coursework and projects
  - Knowledge of FPGA architecture - consulting with supervisor, studying recommended papers, software tools, and online resources
  - VPR interfacing - consulting with graduate students who are working on VPR

3.1.2 Risk Assessment

Overlay FPGA implementation overhead

- Implementing the overlay's logic cells and interconnect on a physical FPGA will consume more area than the area consumed by an equivalent amount of logic cells and interconnect on the physical FPGA.
- If the overhead is too high, then we may not be able to fit enough logic cells for our benchmark circuits.
- For example, if a benchmark circuit consumes 5% of a normal FPGA and the overlay has a 20x overhead, then the benchmark circuit may not fit on the overlay FPGA.
- Risk Mitigation strategies:
  - Use a larger, more costly FPGA.
  - Look for more architectural features in the physical FPGA that can be exploited to reduce the overhead of the overlay FPGA circuit.
  - Use a smaller benchmark circuit for the proof of concept.

Unbalanced timing

- Logic cells in the overlay FPGA would ideally be arranged in a perfect grid, but we expect that the placement algorithm will not produce this arrangement on the physical FPGA.
- This means that the timing delay of two cells to their respective "right" neighbours may differ.
- If this imbalance is too big, then the performance of the benchmark circuits may not be satisfactory.
- Risk Mitigation: Constrain a "tile" containing only one logic block to a rectangular shape, then tessellate it to form a grid, producing a more balanced design.

References

[1] I. Kuon, R. Tessier, and J. Rose, "FPGA architecture: Survey and challenges," *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 2, pp. 135–253, 2007.

[2] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for FPGA research," in *International Workshop on Field Programmable Logic and Applications*, 1997.

[3] V. Betz. (2011, Sep.) "A Utility to Convert a VPR netlist to Quartus format". [Accessed: September 17, 2011]. [Online]. Available: http://www.eecg.toronto.edu/~vaughn/vpr/download.html

[4] J. H. Anderson, Q. Wang, and C. Ravishankar, "Raising FPGA logic density through synthesis-inspired architecture," 2010.

[5] Xilinx Inc. (2011, Sep.) "Virtex-5 FPGA Family". [Accessed: September 17, 2011]. [Online]. Available: http://www.xilinx.com/products/virtex5/

[6] ——. (2011, Sep.) "Xilinx FPGAs Offer High-Performance, Low-Power, Low-Cost Silicon Devices". [Accessed: September 17, 2011]. [Online]. Available: http://www.xilinx.com/products/silicon-devices/fpga/index.htm

[7] Altera Corporation. (2011, Sep.) "Stratix IV FPGA: High Density, High Performance AND Low Power". [Accessed: September 17, 2011]. [Online]. Available: http://www.altera.com/products/devices/stratix-fpgas/stratix-iv/stxiv-index.jsp

[8] ——. (2011, Sep.) "Stratix V FPGAs: Built for Bandwidth". [Accessed: September 17, 2011]. [Online]. Available: http://www.altera.com/products/devices/stratix-fpgas/stratix-v/stxv-index.jsp