Executive Summary

[This is an executive summary. It's light and fluffy and has no page number!]

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 model, 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.[citation needed]Furthermore, there is no existing physical implementation[1] of an Academic FPGA Model.

One of the tools used in FPGA architecture research is VPR. VPR is a free, open-source placement and routing tool that accepts a wide variety of architecture parameters. [citation needed]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 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 that researchers can use to evaluate FPGA architecture, placement, and routing using 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*[2], 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 FPGA circuit must:

- have enough [we will quantify this by checking size of VPR benchmark circuits] logic cells to accommodate reasonable test circuits,
- be compatible with accessibly priced commercial FPGAs.

1.3.3 Objectives

- Take advantage of the underlying FPGA architectural features in the overlay FPGA design to reduce area and latency.

1.4 Validation and Acceptance Tests

To ensure that the design will be functional and useful its intended user base, we will test it using selected benchmark circuits commonly used for research purposes. The benchmark circuits will be synthesized using ABC and placed and routed using VPR. The circuits will then be loaded onto the design, and tested for proper functionality. This test procedure is intended to ensure that:

- circuits can be implemented using VPR output,

- circuits can be transferred correctly to the overlay FPGA,
- the overlay FPGA can fit reasonably sized circuits, and
- inputs can be set and outputs can be read.

The exact verification process for the benchmark circuits will depend on that circuit's intended function. We will need to develop an appropriate testing mechanism for each benchmark circuit used.

To evaluate the benefits of utilizing architectural FPGA features of the host FPGA board, a 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 criterias are cost, size, and ease of use. The lower the cost of the finished design to the reasercher, the better. We must also ensure that the design is large enough to handle circuits the reaserchers 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 timeline.
   - 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[3] will fit a larger overlay circuit. Virtex 6 and 7 feature up to 760,000 and 2,000,000 logic cells respectively[4].

    - 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[4], 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[5] and up to 952,000 for Stratix V[6].
    - [check for something similar to SRL32]

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. Verilog parameters
    - 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 - available online[3]
  - VPR placement and routing - available online[4]
  - Vendor FPGA tools: Xilinx ISE - licenses from supervisor
- Required Hardware
  - FPGA development board - currently using Xilinx Virtex 5 from supervisor. We can discuss with our supervisor the possibility of getting alternate boards should the need arise.
  - Host computer - personal laptops
  - Serial cable and USB adapter - widely available at electronics stores at low prices
- Required Skills and Knowledge
  - Verilog circuit design skill - gained from previous coursework and PEY experiences
  - Knowledge of FPGA architecture - consulting with supervisor and studying recommended resources
  - VPR interfacing - consulting with graduate students

3.1.2 Risk Assessment

- FPGA Size
  - The circuit design may have too much overhead, making it impossible to fit on the selected FPGA board.
  - Risk Mitigation: calculate circuitry resources needed for the overall design, and compare to resources available. If it's insufficient, either a larger FPGA board must be chosen, or the circuit design needs to be re-done.

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

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

- Timing
  - Timing on signals travelling between logic blocks may be unbalanced, as in the overlay logically adjacent cells may not be physically adjacent.
  - Risk Mitigation: Design a rectangular "tile" containing only one logic block, and tesselate it to create the overall layout to keep timing consistent.

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. (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

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

[4] Xilinx. (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

[5] Altera. (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

[6] ——. (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