Chapter 7

Conclusions and Future Work

7.1 Thesis Summary.

In this thesis we make new inroads into the understanding of digital circuits as graphs. We introduce a new method for dealing with the shortage of quality benchmark circuits for computer-aided design and for answering questions about FPGA architectures. The use of benchmarks is crucial for these applications because of their inherent heuristic or approximate nature.

Our approach to this problem involves first determining a combinatorial *characterization* of combinational and sequential circuits. We apply the new characteristics developed in this dissertation to form a statistical *profile* of circuits. Based on our abstract model of combinational and sequential circuits, we define the problem of *parameterized circuit generation* and give an algorithm to solve the problem. To bind this work together, we provide a method for the *validation* of benchmark circuit quality. In this validation process, we show both strong empirical evidence that the circuits produced by our software are good proxies for existing real benchmark circuits and that random graphs are not.

Using the methods developed here, we are able to generate large numbers of sequential benchmark circuits of up to 200,000 logic elements. The software implementation of the algorithm is fast, and can generate a circuit with 30,000 nodes, beyond the size of current FPGAs, in less than one minute of Sparc4 CPU time. The tools are practical and can output circuit netlists in the Berkeley BLIF format, or in other commercial formats such as Xilinx XNF [73], Altera AHDL/TDF [4], Actel ADL [1] and a subset of Verilog. The tools are of interest to industry as well as academia, and have already been used at a number of

companies.

7.2 Specific Contributions.

This thesis contributes to the state of knowledge in the following ways:

- We define a set of new statistical characteristics of combinational circuits: *shape*, *edge length* and *output distribution* and formalize a model and description of combinational circuits in terms of these and other parameters.
- We define a new theoretical combinatorial characterization of reconvergent fanout in both combinational and sequential circuits, and give an algorithm to extract the reconvergence parameter from a circuit.
- We define a characterization of "locality" in combinational circuits, and give an algorithm to efficiently extract locality information from a circuit.
- We define a new abstract model of sequential circuits, and a set of new characteristic parameters of sequential circuit graphs.
- Using existing benchmarks we form a *profile* of circuits in terms of the above characteristics. This profile is given in Appendix A, in the GEN specification language SYMPLE.
- We identify and formally define the problem of "parameterized random circuit generation" and set the ground rules for what type of generation tool is acceptable. We give a detailed algorithm for generating circuits using the combinational and sequential characteristics above, and incorporating the default profile just mentioned.
- We give a method of validating the quality of our benchmark circuits, and provide conclusive evidence both that this algorithm generates high quality circuits, and that random graphs produced by other means are not good for use as benchmarks.
- With the approach of this thesis, we provide a new methodological framework for approaching the design and analysis of heuristic algorithms, and the validation process for these algorithms. This paradigm is increasingly important for the algorithms

community as data sizes and execution time for hard problems continue to increase faster than algorithmic and machine speedups.

In addition, this thesis makes specific practical contributions to the community by providing two new freely-available software tools CIRC and GEN, together comprising about 50,000 lines of C source-code. CIRC is a tool for the analysis and extraction of all the circuit characteristics mentioned above. GEN implements the complete algorithm of Chapter 5, taking an input parameterization, in combination with the default profile, and producing a usable benchmark circuit which meets that parameterization. The code for CIRC and GEN can be obtained from the project web-site [40]. Copies of the source code have been downloaded under an academic license by more than 30 persons representing more than 20 companies and academic institutions, and have been installed by the author for use at Xilinx, Altera, Actel, and Hewlett Packard Corporations. Circuits produced by GEN have also been used in an academic partitioning competition held at the 1996 ACM/SIGDA Design Automation Conference.

7.3 Future Work.

The concept of circuit characterization and parameterized benchmark generation has not been studied before, and there are numerous ways in which it can be extended. We divide these into two areas: research into new understanding of circuits and better methods of generation, and suggestions for the practical improvement of the current CIRC and GEN implementations.

7.3.1 Further Research

The most interesting area of future research is to improve on the combinatorial characterization of locality in circuits. Rent's Rule provides us with a rough guide that we can apply on average, and our characterization of locality from Section 3.5 provides empirical data that we can apply directly to generation. A better understanding of local structure and hierarchy would provide new insights into all phases of computer-aided design, especially partitioning and placement. This knowledge would greatly help us with questions about how to properly scale the architectural features of FPGA architectures and whether to use flat or hierarchical architectures. Along with a study of locality, any other new

characterizations which provide us with knowledge about circuits would be useful.

We need to know more about how the structure of circuits changes with size. The circuits examined in this thesis range to 4,500 LUTs, about the boundary where circuits change from single-purpose computations to "system on a chip" designs. The next generation FPGAs will begin to implement these system level functions, and we must be prepared for them. A closely related issue would be how to validate circuits when we don't have any real circuits available to clone. It is difficult, however, to think of validating the routability and structure of a 200,000 LUT circuit when we have never seen one.

As circuits move towards system level designs, we will want to generate benchmarks which have multiple different types of logic in the same design. This means adding memory and datapath elements to the control logic that we currently generate. Our model of sequential circuits abstractly extends to an arbitrary form of hierarchy, but further investigations into how to "stitch" datapath, memory, or existing real circuits into GEN-circuits would be very interesting. Wilton [69, 70] has done investigations into the structure of reconfigurable memory for FPGAs; the marrying of this work with GEN would be particularly interesting now that FPGAs are starting to include large configurable memory blocks on-chip. Implementation of a more generalized hierarchy would also allow us to incorporate aspects aspects of Darnauer and Dai's Rent-based algorithm to the generation of very large control circuits with controllable partition trees.

7.3.2 Improvements for GEN.

An obvious effort for future work on GEN would be to fine-tune the parameters, the default profile, and the algorithm. The current profile is based on the MCNC circuits, and it would be beneficial to extend this to an empirical analysis of more, and more varied circuits. Such an exercise is as much political as academic, because it involves convincing competing vendors that there is mutual benefit to pooling their information.

Were more circuits available, it would be very useful to analyze different *types* of circuits separately. We roughly classified circuits as datapath or control logic in Chapter 2, but there are a number of distinct circuit classes within each of these: arithmetic, digital signal processing, state-machines and encryption, for example. Separate default scripts for each type of circuit would be useful.

The current version of GEN outputs a structural netlist, with all lookup-tables simply

programmed as nand gates. This means that we cannot guarantee the usefulness of circuits for the evaluation of synthesis and optimization tools. Trevillyan [67] has done some preliminary work to analyze the statistical contents of lookup-tables in technology mapped circuits, and it would be a good practical improvement to GEN to study this problem further and incorporate functionality into the output netlist.