

AN EXPERIMENTAL EVALUATION OF ALGORITHMS FOR VLSI PARTITIONING

M. Rabinovich, K. Robarge, J. Szmyd, and J. Carletta
Department of Electrical Engineering and Computer Science
Case Western Reserve University

COMPUTING RESEARCH ASSOCIATION COMPUTING RESEARCH EXPERIENCE FOR WOMEN PROJECT FINAL REPORT

Abstract

This report describes work in VLSI partitioning done under the Computing Research Association's Computing Research Experience for Women program. Three undergraduate women participated throughout the 1998-1999 academic year. In the process, the team wrote software to implement two existing "classic" methods for VLSI partitioning, and studied modern algorithms by reading recent journal and conference papers. The team then developed, coded and experimented with new algorithms, including a genetic algorithm and algorithm that uses clustering to speed up execution time. A parser for structural circuit descriptions written in the VHDL language was also developed, so that the partitioning algorithms could be evaluated using a wide variety of example circuits.

Introduction

Partitioning is a problem that runs central to VLSI design automation, and one that has attracted a great deal of interest; a recent survey [1] lists almost 200 papers on the subject. Given a circuit, partitioning divides the circuit into two or more smaller parts. The most common variation of partitioning, min-cut bipartitioning, breaks a circuit into two parts so as to minimize the number of interconnections between the two parts. Partitioning has many applications in VLSI design. The most obvious is that of dividing circuitry that will not fit on a single chip among a number of chips. Partitioning has also become increasingly important as systems to be designed have outgrown existing computer-aided design (CAD) tools for VLSI. Computationally, it is no longer feasible to design VLSI systems in one piece; system level partitioning is used to break the design into smaller pieces that the software tools can handle.

Project Goals and Research Process

The goals of this project were to investigate current algorithms for VLSI partitioning, to experimentally evaluate some of the algorithms, and to suggest modifications to improve the algorithms. A major goal was also to introduce undergraduate women to research. The experience was organized as a special topics course within the Department of Electrical Engineering and Computer Science. Because the students were able to receive academic credit for the project as a technical elective, participation in the project did not slow down their progress towards their degrees. The student team attended weekly coordination meetings with the team mentor, and often worked together outside of that weekly meeting. Each student reported on her progress for the week at the coordination meetings, and set goals for the coming week. All work was documented in engineering design notebooks.

The project started with an introduction to the VLSI partitioning problem by the team mentor. The team then did a literature search, reading journal and conference papers to learn about recent developments in VLSI partitioning. As a team, we choose first to implement two "classic" methods for partitioning. The first, the Kernighan-Lin algorithm [2], is the forefather of all move-based algorithms. Move-based methods explore the solution space of all possible partitions by moving from one solution to another; they start with one or more arbitrary partitionings of the circuit, and then iteratively search for better solutions by making local modifications to (or "moves" from) existing solutions. The second, the Fiduccia-Mattheyses algorithm [3], while relatively old, is still widely used, and is the universal benchmark to which modern algorithms are compared.

The team then decided to experiment with several modern techniques for partitioning. We implemented a genetic algorithm for partitioning, and experimented with the use of various kinds of circuit element clustering to improve the efficiency of the Fiduccia-Mattheyses algorithm. We also wrote software capable of parsing structural circuit descriptions written in the VHDL language, to make it possible to work with a wider variety of benchmark circuits.

Results and Conclusions

This section describes results for the three major parts of the project: the VHDL parser, a genetic algorithm for partitioning, and an investigation of the effects of clustering on partitioning.

A major part of the project, peripheral to VLSI partitioning, was development of a parser for structural circuit descriptions written in VHDL. Standardized by IEEE in 1985, VHDL has become widely used by the CAD community, and is supported by nearly every major CAD software vendor. The parser handles structural VHDL code according to the IEEE Std 1076-1993 VHDL Language Reference Manual. Team member Julie Szmyd led the effort to develop the parser, which was designed using basic principles of compiler construction. A lexical analyzer breaks the VHDL input file into recognizable tokens such as identifiers, numbers, and keywords. A syntax analyzer then maps sequences of tokens to the grammar rules of VHDL. The syntax analyzer was complicated by the fact that some of the grammar rules of VHDL are very similar. There are some cases for which the analyzer has to look ahead several tokens to determine which rule to follow, and some cases for which even look-ahead is not sufficient to distinguish the rules due to ambiguity in the VHDL language. Information from the syntax analyzer is used to generate output files containing information about the connectivity of the circuit. Overall, the parser accepts a VHDL file as input and, if the file is syntactically correct, it generates a net list file, a component list file, and a symbol table file. The net list file contains information regarding the connectivity of nets to components while the component list file is exactly the opposite, giving details of components and the nets that are connected to them. The symbol table contains a list of each component name and the integer identifier for the component, as well as each net name and its integer identifier. These component list and net list files are passed as input to the partitioning algorithms. A genetic algorithm for VLSI partitioning was developed, implemented and tested by team member Kim Robarge. Genetic algorithms are motivated by Darwin's theory of natural selection in evolution, where "superior" members of species produce more offspring in succeeding generations than "inferior" members [1]. Genetic

algorithms start with an initial population, or set of feasible but non-optimal partitions, and use crossover and mutation operations on the members of the initial population to create a new generations of "offspring" members. Then, a fitness function is used to decide whether a particular offspring solution should survive to participate in crossover and mutation to create the next generation. The algorithm was inspired by the work of Saab and Rao [4]. Circuit elements that are clearly on the wrong side of the partition (i.e., that have only a few connections to components within its own side of the partition, and many connections to components on the other side) are marked as "bad". New offspring are created by randomly swapping "bad" elements from one side to the other. The fitness function involves a judgement of whether a particular offspring partition is a good one. Two different judgement functions were used, both involving the number of wires cut by the partition. Experiments were run on benchmark circuits, monitoring solution quality and the number of iterations required for the solution to converge. Using the benchmarks, the judgement functions were compared, and an early termination condition was developed in an attempt to reduce execution time without sacrificing solution quality. Another aspect of the project is a study of the effects of clustering on the efficiency of the Fiduccia-Mattheyses algorithm. This work, done by team member Marina Rabinovich, will become a Masters degree project, and is ongoing. Clustering algorithms group together heavily connected elements, merging them to form larger elements (see [5], for example). Clustering can be used to great advantage in conjunction with iterative improvement partitioning algorithms. Clustering significantly reduces the complexity of the graph, grouping together elements that clearly should be together into a single graph node, so that partitioning algorithms can run in a much smaller time on the condensed graph with very little degradation in solution quality. Ultimately, this part of the project will incorporate several kinds of clustering, some from the literature and one novel approach, with the Fiduccia-Mattheyses algorithm, to see the effects on execution time and solution quality.

All software work was done in C programming language using Microsoft Visual Development Studio for Windows 98/NT platform. While implementing the algorithms, the team members learned very important concepts for the software development cycle, such as choosing an appropriate programming language and appropriate data structures, reuse of existing modules, importance of software testing, and setting realistic deadlines. In the process of implementing our project, we also got valuable experience in programming, and using powerful commercial tools such as Microsoft Visual Development Studio. We learned a great deal about working as a team. Two of the team members have decided to pursue Masters degrees, and the third is currently using what she learned about VLSI in a co-operative experience in industry. We found that this project provided an opportunity for mentoring that was otherwise sorely lacking in our department.

Team Members

Marina Rabinovich is a computer engineering major in her last year at Case Western Reserve University. She is continuing the work she began on the CRA-CREW project, and will soon complete a Masters degree involving the use of clustering to improve the efficiency of algorithms for VLSI partitioning. Marina is obsessed with music and has a huge MP3 music archive. In her free time, she enjoys ice skating and parachuting.

Kim Robarge is a computer engineering major at Case Western Reserve University. She plans to graduate with a Masters degree in May of 2000. She gained knowledge about VLSI circuits, research, and teamwork while participating in this project. She feels fortunate for having the chance to meet and work with Joan, Julie, and Marina. She will continue to try to model herself after Joan's kindness, Julie's optimism, and Marina's humor. She wishes these three women the best in all of their future endeavors.

Julie Szmyd is a junior computer science major at Case Western Reserve University. She looks forward to graduating in May of 2000 and hopes to find a job with a large computer corporation. With the experience that she gained from the project, Julie was able to obtain a co-op position for the upcoming summer and fall semesters with Hewlett Packard. She is currently working in the VLSI Technology Center in Fort Collins, Colorado. She is extremely active and enjoys running, swimming, biking, and rock climbing in her free time.

Team mentor Joan Carletta is an Assistant Professor in the Department of Electrical Engineering and Computer Science. She has been with Case Western Reserve University for four years. She teaches computer engineering classes, ranging from digital logic and computer organization to VLSI design automation. Her research interests are in the area of VLSI design automation, with special interest in high level synthesis and design-for-testability.

References

- [1] C. J. Alpert and A. B. Kahng, "Recent Directions in Netlist Partitioning: A Survey," Integration: the VLSI Journal, 19(1-2), 1995, pp. 1 - 81.
- [2] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," Bell Systems Technical Journal, 49(2), 1970, pp. 291 - 307.
- [3] C. M. Fiduccia and R. M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions," Proceedings of the ACM/IEEE Design Automation Conference, 1982, pp. 175 - 181.
- [4] Y. G. Saab and V. B. Rao, "Fast Effective Heuristics for the Graph Bisectioning Problem," IEEE Transactions on Computer-Aided Design. Vol. 9, No. 1, January 1990.
- [5] J. Cong and M. Smith, "A Parallel Bottom-up Clustering Algorithm with Applications to Circuit Partitioning in VLSI Design," Proceedings of the ACM/IEEE Design Automation Conference, 1993, pp. 755 - 760.