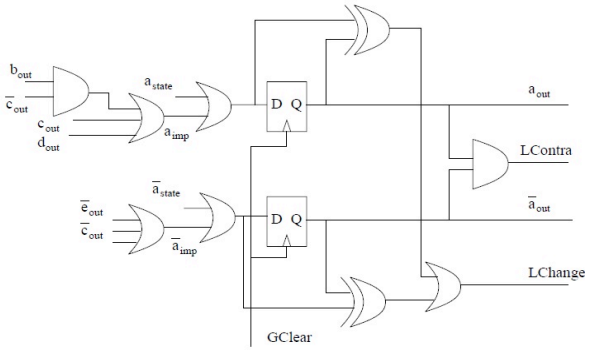
Accelerating Boolean Satisability with Configurable Hardware

Peixin Zhong, Margaret Martonosi, Pranav Ashar, Sharad Malik



Year of publication: *1998*
Area: *Applications*

This paper addresses accelerating Boolean satisfiability, a problem of great interest in a number of computer-aided design (CAD) applications and beyond. Satisfiability is the problem of assigning the variables of a Boolean expression in such a way that the expression evaluates to true. If there is no such assignment, a counter example is produced. Satisfiability is used in many automated proof tools to prove the validity of a statement. It is also used for automatic test pattern generation and routing, among other applications.

Boolean satisfiability was the first problem to be shown to be NP-complete. Many optimization problems are cast into Boolean satisfiability, allowing them all to leverage advances in satisfiability solvers. Using parallelism, such as the parallelism available in FPGA fabrics, helps to accelerate specific instances of the Boolean satisfiability problem.

This paper represents one of the early, successful approaches to implementing a SAT solver on FPGAs and is an early example of the advantages of specializing a circuit for a particular problem instance. It has several excellent features, including software that translates the Boolean satisfiability problem to FPGAs. An interesting discussion of compile times versus run times in the paper is still relevant to current problems in heterogeneous acceleration. In addition, the FPGA implementation ran on state-of-the-art hardware for the time, a DEC Pamette board and an IKOS logic emulator designed for emulating processors. SAT solving speedups of several hundred times were measured for problems of up to four hundred variables and one thousand clauses.

This paper represents the basis of much of the research that followed in this hugely important application area. SAT solving continues to be an important application today, and FPGA implementations of SAT continue to be researched and published. Researchers are interested in increased speed, but also in the ability to solve much larger problems with orders of magnitude more variables and clauses.

This work also had enormous impact on pure software satisfibility solvers. The huge speedups compared to software motivated the developers to understand where time was going in their software solvers. The resulting understanding, initiated a series of optimizations to the software solvers, starting with Chaff, that provided orders of magnitude speedup without FPGA hardware.

Miriam Leeser

DOI: http://dx.doi.org/10.1109/FPGA.1998.707896