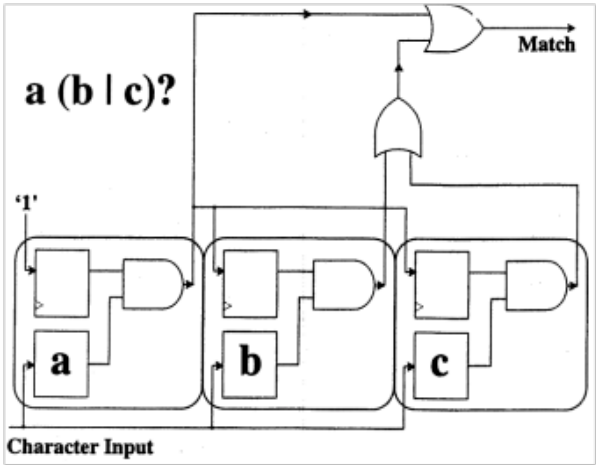
Assisting Network Intrusion Detection with Reconfigurable Hardware

Brad L. Hutchings, Rob Franklin, Daniel Carver

Year of publication: *2002*
Area: *Applications*



This paper was pioneering in that it introduced an integrated domain-specific tool flow for generating efficient FPGA circuitry for carrying out the regular expression matching required in practical network intrusion detection (NID).

The work built upon the prior work of Sidhu and Prasanna (FCCM 2001), who showed how regular expressions could be mapped to a modular FPGA implementation of Nondeterministic Finite Automata (NFA) without intermediate Deterministic Finite Automata representations.

The central contributions of the paper were threefold. First, the NID module generator was driven by pattern matching rules used by a standard software system, specifically the open-source SNORT system. This showed how an FPGA-neutral, domain-specific language could automatically be mapped to an efficient FPGA implementation and set a trend for future researchers, who adopted the SNORT rule database as a standard benchmark. The use of SNORT rules meant adding support for extended regular expression features beyond the basics of concatenation, choice, and Kleene star.

Second, the paper made good practical use of the JHDL hardware design tool kit (FCCM 1998) to underpin the NID module generator, by providing a convenient means to build modular implementations of the NFA required to perform regular expression matching.

Third, the paper showed how the module generator provided a complete flow from the domain-specific NID rules through to optimized FPGA implementations, and illustrated this with performance results that showed the scalable benefits of the FPGA-based approach, compared with the limitations of a standard software-based approach.

Through these contributions, this paper exemplifies what is desirable in an FCCM applications paper. It was not just a direct implementation of some application on an FPGA. Rather, it showed a reusable tool and methodology in a specific domain. Although NID was the motivating application, the work is more generally applicable to situations where regular expression matching is required. While the actual FPGA resource counts and performance data are now somewhat dated, and other researchers progressively improved upon the results, the approach used in this work has stood the test of time.

Gordon Brebner

DOI: http://dx.doi.org/10.1109/FPGA.2002.1106666