Building and using a highly parallel programmable logic array

Maya Gokhale, William Holmes, Andrew Kopser, Sara Lucas, Ronald Minnich, Douglas Sweely, Daniel Lopresti

Published: *IEEE Computer*, vol. 24, no. 1, pp. 81--89, January 1991*.*

This paper presents Splash, a system that pioneers the adoption of field-programmable gate array (FPGA) technology to solve customised computing applications. It is remarkable in covering not only Splash's multi-chip architecture but also facilities for programming and debugging Splash; it describes how a single-board Splash can be significantly faster for DNA sequence comparison than the much larger and more expensive Cray-2 supercomputer.

Instruction-based processors have dominated computer systems for the last 70 years, but their drawbacks are well known: they are much slower than special-purpose architectures customised to a given application. However, it is not cost-effective to design and manufacture a large variety of hard-wired, special-purpose computing machines. Splash shows that the emergence of FPGA technology provided an opportunity to build programmable special-purpose processors that can support customised solutions to a variety of applications by programming the FPGA devices. Released to the user community of the Supercomputing Research Center in Maryland, USA in 1989, just a few years after the first commercial FPGAs came to market, each Splash logic array board contains 32 Xilinx 3090 FPGAs and 32 memory chips, with a Sun-3 or Sun-4 workstation as host. Each pair of FPGA and memory chips are connected to their neighbours to form a linear systolic array of programmable processing elements. The composite Splash system foreshadows today's system-on-chip FPGAs with integrated programmable logic, memory, and processors.

Developing high-performance applications for FPGAs is a challenge even today. To enable rapid turn around, some notable tools were developed for Splash. One of the tools facilitates the design and replication of systolic cells in a language-based environment. Another tool supports symbolic debugging in a way similar to software debuggers and can be used with the state readback facility for FPGAs to support debugging of applications mapped onto Splash. This early work on in-circuit debugging shows how much Splash was ahead of its time, since in-circuit debugging is still an active area of research more than 25 years after this publication. The comprehensive performance evaluation of Splash against eight other systems, from workstations to supercomputers to special-purpose systolic circuits, also helped establish the potential opportunities for FPGA-based computing and make this paper a classic.

Endorsement by: Wayne Luk, Professor, Imperial College London

DOI: http://doi.org/10.1109/2.67197