Processor reconfiguration through instruction–set metamorphosis

Peter M. Athanas and Harvey F. Silverman

Published: *IEEE Computer, March 1993, Volume:26, Issue:3, Pages 11–18.*

This paper describes a software compiler that generates hardware and software images from a single C description to target a MC68010 microprocessor and four attached Xilinx 3090 FPGAs. The system accelerates an application by 1) synthesizing computationally intensive functions to the FPGA hardware and 2) configuring the remaining software and system so that the hardware versions of these synthesized functions are efficiently executed on the attached FPGAs at run-time.

Prior systems generally required programmers to describe the software and hardware of their systems using two different descriptions and two different tool flows. Software was written in C (for example) and compiled using standard compilers. Hardware was described in HDL or schematic form and synthesized. The user also had to manage the run-time configuration of the software and hardware. PRISM evolved the art to the next step by implementing a compiler that could generate both the software and hardware components from a single C description. PRISM also configured the software and hardware at run-time.

PRISM was an early, visionary prototype that demonstrated the feasibility and advantages of using a single description for combined software/hardware implementation. It would be many years before this capability could be provided at large scale with vendor-provided software (e.g., Open-CL-based FPGA tools) and to find wide use by practitioners of configurable computing.

Endorsement by: Brad Hutchings, Professor, Brigham Young University

DOI: https://doi.org/10.1109/2.204677