Design versus Programming in Custom Computing Machine Applications: Experiences Using the Algorithmic State Machine Method[1]

James P. Davis, Nirish Namilae and Heather A. Wake
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29208 USA
jimdavis@cse.sc.edu

Abstract

In this paper, we explore some of the methodological issues associated with creating high-performance computing applications on programmable logic devices forming the "fabric" of a reconfigurable computing platform. Many custom computing applications deal with the coding of a data stream for bit-serial transmission in a variety of noisy media—such as twisted pair cabling, air, or the vacuum of space. These applications are typically formulated as a collection of arithmetic data path and control logic blocks, and have been subject to analysis, architecture and design using custom logic methods for some time. However, one of the predominant issues of custom computing is eliminating the "hardware" dimension associated with moving algorithms onto programmable logic substrates—such that the task of creating custom computing applications is one of writing programs subject to compilation, where the burden of making "design choices" is placed in the hands of compiler designers and the compilation algorithms they develop. In this paper, we explore the issues contrasting these two different approaches to systems design—what are referred to in other literature as "language based" versus "model based" techniques—and we report on some results using the ASM method (found now in most texts on digital logic design) to explore the design space associated with programmable logic and custom computing, contrasting this graphical approach--which embraces design problem-solving--against the predominant language-based programming model. We use examples of Congruential Sieves and Reed-Solomon coding for error correction as a basis for discussing and contrasting these language and model-based methods.

[1] The work has been partially supported by the NSA Lucite program.