Preface

The Splash 2 project began at the Supercomputing Research Center[1] in September of 1991 and ended, with success, in the spring of 1994. Splash 2 is an attached processor system using Xilinx XC4010 FPGAs as its processing elements. As such, it is a *custom computing machine*. That is to say that much of what would be the instruction set architecture of the processing elements is not specified except in the details of the program developed by the application programmer. Although a higher-level block diagram of processing elements, memories, interconnect, and dataflow exists in the hardware structure of Splash 2, the details of the instruction set architecture level of the machine will vary from one application to the next.

The Splash 2 project is significant for two reasons. First, Splash 2 is part of a complete computer system that has achieved supercomputer-like performance on a number of different applications. By "complete computer system" we mean that SRC created or caused to be created an extant hardware system (replicated a dozen times), a complete programming and runtime environment, and a collection of application programs that exercised the unique hardware.

The second significant aspect of Splash 2 is that we were fortunate enough to be able to build a large system, capable of performing real computations on real problems. One common complaint about performance results on novel computing machines or environments is that results on small problems cannot be accurately extrapolated to large problems. The Splash 2 system that was designed and built is a full-sized machine and does not suffer from this defect.

To get to the point: why a book?

[1]Renamed the Center for Computing Sciences in May 1995, but referred to throughout this book as SRC.

This is a novel computing machine. In order to understand what happens when the application programmer is permitted, indeed required, to design the processor architecture of the machine that will execute his program, it is necessary to see the system as a whole. Programmability and problems to be run on this machine both had major influences on its architecture, just as its architecture and its unique nature influence the kinds of problems one would expect to program for this machine and the nature of that programming. And standing between the user and the machine, as the old joke goes, is a new kind of programming environment and an evolving understanding of how this environment must allow the use of the hardware, without forcing every programmer to be a hardware-design engineer.

At the first IEEE Workshop on FPGAs for Custom Computing Machines, one of the industrial attendees remarked that, although nearly everyone would agree, as part of a coffee-room discussion or the like, that it would be *interesting* to think about building a "computer" using FPGAs, no one in management (except perhaps at SRC and DEC PRL) had put up the commitment necessary in time and money to do a real test of the idea. It was then remarked that, given the nature of the marketplace and of engineering management, these first attempts had to be successful in order to open the door for future work. We feel we have been successful, and we offer in this book an in-depth look at one of the handful of data points in the design space of this new kind of machine.

We would hope that this book has a broad appeal and is readable with understanding by nearly all computer scientists and computer engineers. To the hardware designer, perhaps we can offer a new look at programming applications on a moderately general FPGA-based computing machine instead of designing circuits for a specific board incorporating FPGAs. The engineering world has viewed FPGAs, to a great extent, as the next logical step in a continuum of electronic devices; we offer, we feel, the option of viewing them much more broadly than that. To the computer architect we offer a variant hardware platform and an indication of how that general platform can be used. Much of computer architecture is a compromise between the functionality desired and the limits of what can be built given existing technology; we offer the use of a new technology that can offer, to a limited extent now, and could offer much more generally later, greatly increased functionality. For some of those who have hard problems in computation, we offer much of the power of special-purpose hardware without the inflexibility and uncertain delivery times of hardware. The long-term task is not to map a high-level language to a particular architecture or range of architectures, but in some sense to create for each application program a suitable architecture to which the high-level language will be mapped. And to the language designers and compiler writers we offer a world to conquer. We have presented one imperfect but usable approach to programming such a computing machine, and we trust that others interested in the critical problem of making these machines programmable can learn both from what we did right and what we did wrong.

Chapter 1 discusses the general concept of Custom Computing Machines, of which Splash 2 is one example. Chapters 2 and 3 describe at a high level and then in some detail the hardware architecture of Splash 2. Chapter 4 covers the design considerations and decisions in arriving at the second-generation Splash 2 architecture. We present this chapter at the end of the section on hardware, on the basis that it is easier to understand variations in a design when those variations are compared against something concrete.

Chapters 5 and 6 describe, also first at a higher and then at a lower level, the software architecture of Splash 2. All the application programs in the latter chapters were done using VHDL as an applications programming language and these tools in support. The main goal of the Splash 2 project was to show that software, as described in Chapters 5 and 6, could make a computer using FPGAs as its processing elements into something that reasonable people would call "programmable," and, in that sense, the heart of the Splash 2 project is in Chapters 5 and 6. Throughout the life of Splash 2, however, there has been an alternative view of programming. This view is reflected in Chapter 7 on the Splash 2 version of the programming language dbC. The approach taken in dbC is to permit the programmer to use a version of C as the programming language. It is the compiler which then becomes responsible for, in essence, recognizing the instruction set architecture necessary to execute the program and then creating in the FPGA the requisite registers, logic units, and control.

Chapters 8 through 11 then describe four different applications programmed to conclusion on Splash 2. The first of these—the sequence comparison problem—was the driving application, in the sense that funding for Splash 2 was based on its perceived ability to perform this computation. This and the text processing application were done at SRC.

The Splash 2 project team was fortunate in that SRC's parent company, the Institute for Defense Analyses, issued two contracts, to Virginia Polytechnic Institute and to Michigan State University, for applications work on Splash 2 in image processing and fingerprint identification. Both applications seemed good matches with the Splash 2 architecture but lay outside the normal realm of SRC's research program. The faculty members involved have each prepared a chapter on these applications.

We close in Chapter 12 with some opinions and speculations about the future. In an appendix, the project manager presents a chronology of the entire Splash 2 project.

It is incumbent on us, and a genuine pleasure, to thank the Center for Computing Sciences of the Institute for Defense Analyses and the CCS Director, Francis Sullivan, for supporting us in our writing and editing of this book. All royalties will be donated to the Center for Excellence in Education, formerly known as the Rickover Foundation, in McLean, Virginia. The Center for Excellence in Education supports science and engineering education through its sponsorship of the Research Science Institute each summer for high school seniors, its Role Models and Leaders Project in Washington, D.C., Los Angeles, and Chicago for promising women and minority high school students intending to study science and engineering, and its support and preparation of the United States Informatics Olympiad team each year.

The Splash 2 project was a success in large part due to the ability of those who were involved nearly full-time, but it might not have taken the course it did had the hard-core Splash 2 players not had the chance to get advice and occasional help from a much larger group of experts both at SRC and elsewhere.

We acknowledge, therefore, the help and advice of Nate Bronson, Dan Burns, Bill Carlson, Neil Coletti, Maripat Corr, Steve Cuccaro, Hillory Dean, Chuck Fiduccia, Brad Fross, Charles Goedeke, Maya Gokhale, Frank Hady, Dzung Hoang, Bill Holmes, Amy Johnston, Elaine (Davis) Keith, Dan Kopetzky, Andy Kopser, Steve Kratzer, Jim Kuehn, Sara Lucas, Michael Mascagni, Marge McGarry, John McHenry, Ron Minnich, Lindy Moran, Fred More, Mark Norder, Lou Podrazik, Dan Pryor, Craig Reese, Paul Schneck, Brian Schott, Nabeel Shirazi, Doug Sweely, Dennis Sysko, Mark Thistle, Bob Thompson, Ken Wallgren, Alice Yuen, Neal Ziring, and Jennifer Zito.

Duncan A. Buell
Jeffrey M. Arnold
Walter J. Kleinfelder
Bowie, Maryland
March 1996