Practical FPGA Programming in C

Download EBook
High-performance FPGA-accelerated software applications are a growing demand in fields ranging from communications and image processing to biomedical and scientific computing. This text introduces powerful, C-based parallel-programming techniques for creating these applications, verifying them, and moving them into FPGA hardware.

Let's start on the positive side. This book is very current - it mentions the recent Cray and SGI FPGA accelerators as part of the reason for programmers to learn about FPGAs. It goes through some interesting and practical examples, showing how a C program can be used to specify the synthesizable logic for triple DES, including time/space tradeoffs. It gives some hardware awareness, without trying to turn a programmer into an EE. Best of all, it shows practical use of Impulse C, an ANSI C extension that supports pragma-driven pipelining and parallelism. Impulse C, by the way, seems to have an exceptional pedigree. It seems to descend from Maya Gokhale's work at Los Alamos, and there isn't much better parentage for such a product to claim. There are some problems with this book, however. It relies overwhelmingly on the Impulse C product, to the exclusion of Handel C and a flock of other products - it's really an Impulse C how-to. That's not necessarily a bad thing, but it's not what I assumed from the title. It emphasizes streaming data, like radar input, video, audio, or all the other traditional DSP applications of FPGAs. Streaming is good,
but a poor match to the Cray and SGI coprocessors. The biggest problems in this book come from the basic approach of trying to turn an FPGA into a CPU. Back when iron was first introduced for bridge-building, iron beams were built to imitate wooden ones. The result was a wooden bridge made of iron. It worked, but used far more material than was needed for the job, and got relatively poor performance from the material. Ditto what happens when FPGA logic goes into a soft CPU. The result is a little like the classic "Fortran program written in C," only worse.

Download to continue reading...