

Whither Configurable Computing?

Carl Ebeling

Department of Computer Science and Engineering
University of Washington

Configurable computing has captured the imagination of many architects who want the performance of application-specific hardware combined with the flexibility of general-purpose computers. Despite the efforts of many research groups over the past decade, successes have been rare: Configurable computers so far exhibit poor cost performance for most common applications. To make things worse, configurable computers are notoriously hard to program.

Commercial FPGAs are not well-suited to most applications. These FPGAs are necessarily very fine-grained so they can be used to implement arbitrary circuits, but the overhead of this generality exacts a very high price in density and performance. Compared to general purpose processors (including DSPs), which use very optimized function units that operate in bit-parallel fashion on long data words, FPGAs are very inefficient for performing ordinary arithmetic and logical operations. FPGA-based computing has the advantage only when it comes to complex bit-oriented computations like count-ones, find-first-one or complicated masking and filtering.

Because FPGAs are so fine-grained and general purpose, programming an FPGA-based configurable computer is akin to designing an ASIC. The programmer either uses synthesis tools that deliver poor density and performance, or designs the circuit manually which requires both intimate knowledge of the configurable architecture and substantial design time. Neither alternative is attractive, especially if the computation itself is relatively uncomplicated and can be described in a few lines of C code.

We are certainly willing to pay some price for configurability but the question is how large a price will we put up with? The current cost-performance price of configurable computers is a factor of about 100x, and the programming price in terms of expertise and time is many orders of magnitude greater. This combined price is much too high for most applications and most users.

Conclusion 1 - FPGA-based configurable computers will be used only in niche applications where cost is of little concern or that require substantial bit-level data computation. New architectures like the Xilinx 6200 will provide some improvement for some applications, but expecting dramatic improvements is unrealistic. Progress in making configurable computers easier to program will be disappointing.

Conclusion 2 - New configurable computers will appear that are based on new coarse-grained architectures more suitable for conventional arithmetic-intensive tasks. Research examples include the Matrix (MIT) and RaPiD (University of Washington) configurable architectures.

Conclusion 3 - Progress in programming configurable computers will require a coordination between the architecture model of computation, the application domain, and the programming language. One need only look to the successful silicon compilers that have been developed for DSP applications such as Cathedral (IMEC) and Lager (Berkeley) to see the advantage of this approach. Although traditional FPGA-based architectures can benefit from this methodology, the newer coarse-grained architectures can take full advantage of it from the ground up.

Conclusion 4 - Systems will appear that incorporate dynamically programmable components in new and interesting ways that allow conventional computing to be blended with application-specific computing at a fine-grained level. Initial attempts include PRISC (Harvard), DISC (BYU) and Brass (Berkeley).

In summary, future progress in configurable computers will result not from continued research along the same FPGA-based path, but from a diversity of approaches including more coarse-grained configurable architectures and constrained programming models that allow more powerful compilation techniques.