

Features:

Standards-based Reconfigurable Computing for HPC

by Michael D'Amour

Chief Operating Officer, DRC Computer Corporation



Users and developers have long recognized the enormous performance gains that reconfigurable hardware can provide for certain types of computationally intensive problems. However, despite well-known advantages, the technology has historically struggled to gain a strong foothold in the high-performance computing (HPC) marketplace. While there were many reasons for this lack of early widespread acceptance, one major issue has been a lack of standards surrounding reconfigurable processors and the absence of a standardized system architecture that can effectively employ them.

Now, that's beginning to change. Many of the technological barriers to widespread use of reconfigurable computers have been overcome, and with standards-based reconfigurable interfaces and hardware plus a growing body of standard language compilers to support the technology, reconfigurable computing is poised to break through as a viable solution for a wide range of commercial and HPC applications.

To understand how we have reached this point, it is important to understand the architectural issues surrounding reconfigurable processors, including both the benefits of reconfigurable computing and the architectural limitations that historically have prevented it from gaining more traction in the market. And, while new tools and standards-based approaches are poised to open the door to widespread adoption of reconfigurable computing, it is also important to understand the steps that vendors and developers are taking to make this vision a reality.

Making the Case for Reconfigurable Computing

Before one can address the architectural issues surrounding reconfigurable computers, it helps to specifically define the term. Generally, a reconfigurable computer is a system that includes a standard CPU attached to an array of configurable hardware, specifically, one or more standard field programmable gate arrays (FPGAs). In operation, the reconfigurable processing unit, or RPU, is loaded to run a particular task or solve a specific problem in hardware, at hardware speeds. Once that task is complete, the hardware is reconfigured to perform the next required function.

While the general-purpose CPU has advantages of flexibility and ease of programming, the RPU provides a medium to accelerate some types of computations many times faster -- in some cases orders of magnitude faster -- than what can be achieved using a standard CPU. Reconfigurable computers achieve these dramatic gains by parallelizing in reconfigurable hardware algorithms that would otherwise run serially in software, providing the hardware-implemented function with greater memory bandwidth than is available to the processor, and hard-wiring function and interconnect. In addition, within the reconfigurable hardware, no instruction fetching is required and logic only runs when data is present for processing. Effectively, users gain the ability to run particular sets of algorithms in hardware that is custom-built to accelerate them. This can provide performance improvements of 10 to 100 times. For some applications such as gene sequencing and facial recognition, the performance is more than 1000 times what can be achieved using general-purpose CPUs.

While such dramatic application acceleration would be attractive to any commercial user, it is particularly compelling for HPC environments, where algorithms in production may be running on hundreds and thousands of servers in one datacenter. In such environments, improving the function-for-function performance of a server by several times (even after the realities of Amdahl's Law are considered) can have a significant effect on both price-performance and power-performance. Ultimately, these performance improvements can be profound -- even allowing users to address problems that were previously computationally intractable.

Consider an RPU implementation that delivers a 10X algorithm performance improvement for 80 percent of the program plus modest optimizations in the remaining 20 percent (control and I/O), as detailed in Table 1. (The I/O may have never been optimized as it previously represented only a minor portion of the run time).

Table 1: Illustration of 10X RPU Acceleration in an HPC Datacenter

	Original System Time Allocation	System Time Allocation with RPU Acceleration
Algorithm	80%	8%
Control + I/O management	20%	14%
Total of original system time required	100%	22%

Even this relatively modest example of RPU acceleration can deliver significant benefits to the HPC user. In this scenario, improving the overall run-time of the program by 4.5 times means that the reconfigurable computer can compute in 16 nodes what previously required 72 nodes. Price-performance levels of 2:1 to 3:1 are achievable at these levels. In addition, the RPU running the function is likely to use 50

percent of the power required by a standard CPU and, with the attendant reduction in the number of nodes, power supplies, and cooling, a reconfigurable computer will provide equal performance for less than 20 percent of the overall power. With the 5:1 reduced space and power footprint, already jammed HPC datacenters can improve their output by multiples without hitting power, cooling, and space limitations.

Barriers to Widespread Adoption of Reconfigurable Computers

With the potential to realize such dramatic advantages through RPU acceleration, a natural question arises: Why hasn't reconfigurable computing taken off in the industry? After all, the concept of reconfigurable computing has existed since the 1960s. And yet, to date, no broadly viable example of reconfigurable computers has taken hold in the market. Historically, there have been several barriers to adoption of the technology, but the most significant barrier has been the lack of a standardized architecture that can accommodate reconfigurable coprocessors.

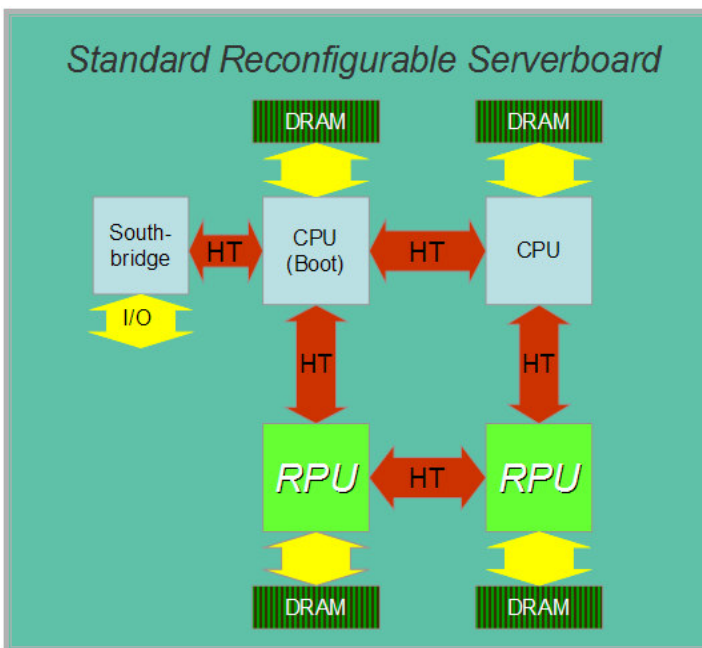
Until recently, reconfigurable computers were based on proprietary system architectures. As a result, the technology was unable to tap the great financial and intellectual energy being put into commercially available, standards-based computers and servers. Certainly, Cray's FPGA-augmented XD1, the SGI Altix with RASC option, and the SRC machines that support FPGAs have found applications in specialized markets. However, these systems have gained only limited acceptance because all were and are proprietary architectures.

As a possible alternative to these architectures, some vendors developed FPGA cards as peripheral boards (which follow PCI bus standards). But implementing reconfigurable hardware as a peripheral presents a number of problems, including issues in slot and rack logistics. Standard servers often have no available slots for FPGA PCI cards, and the cards are often unable to fit in slim form-factors. Currently, no standard server configuration from first- or second-tier server suppliers can be found that includes reconfigurable peripheral cards.

Today, however, a new breed of standards-based reconfigurable computer is emerging. By taking advantage of open-standard processor busses, standard operating systems, standard BIOSs, standard server motherboards, and standard form factors, reconfigurable computing is finally opening up to the rich possibilities of building on the creativity and financial and logistical resources of the commercial computer industry.

Reconfigurable Computing in the Socket

The most important factor in allowing reconfigurable computing to finally break through to the mainstream market has been the emergence of a technical toolkit that allows the RPU to function in a standard system architecture as a true coprocessor. For two decades, the industry has been advancing the FPGA architecture, automated synthesis tools, and high-level language compilers, with an avid development and research community espousing reconfigurable computing. However, it was the emergence of the open-standard HyperTransport bus that completed the technical kit required to make reconfigurable computing widely available. Steve Casselman of DRC Computer Corp., and eventually others, recognized the technical significance of such an open standard in allowing the RPU to function as a full peer processor in a CPU socket.



Once the RPU can be installed in a CPU-grade server board socket, it can be employed more easily and efficiently than ever before. With a direct processor bus providing very low latency between the CPU and coprocessor RPU, much more fine-grained applications become viable in a reconfigurable environment. Implemented in the socket, RPUs are capable of providing under 250ns read latencies -- less than half the latency seen on PCIe peripherals and far better than the microseconds latency experienced on PCI-X.

In addition, the RPU gains large processor-class memory available locally and direct access to all remote memory through the processor bus network. As a result, the RPU can function as a true peer processor, able to load, execute, and store without the need for the processor to intervene. This means that the RPU can process data and return it to the standard CPU's local memory and continue with the next queued request. Further, with the socket capable of driving a power-hungry processor, ample power is available to drive the largest RPU and even additional on-module memory. Applications that previously were compute-bound on the CPU can now digest much more memory bandwidth with hardware acceleration. Still more good news, RPUs on a larger HyperTransport network are capable of

feeding each other directly in a dataflow fashion, creating chains of very powerful coprocessors.

With the ability to implement RPUs as full peer processors using a standard HyperTransport bus, reconfigurable computing is finally poised to gain a much larger foothold in the market. AMD's Torrenza program has opened their HyperTransport to all accelerators and, with Intel's recently-announced access to their front-side bus, reconfigurable computer manufacturers are now welcomed to the processor socket. We will soon see standard reconfigurable server boards offered by high-volume vendors incorporating RPUs as standard components. As a further

endorsement of the standard bus concept, Cray Inc. announced in 2006 their intention to incorporate RPU accelerators into their supercomputers, placing them into standard processor sockets.

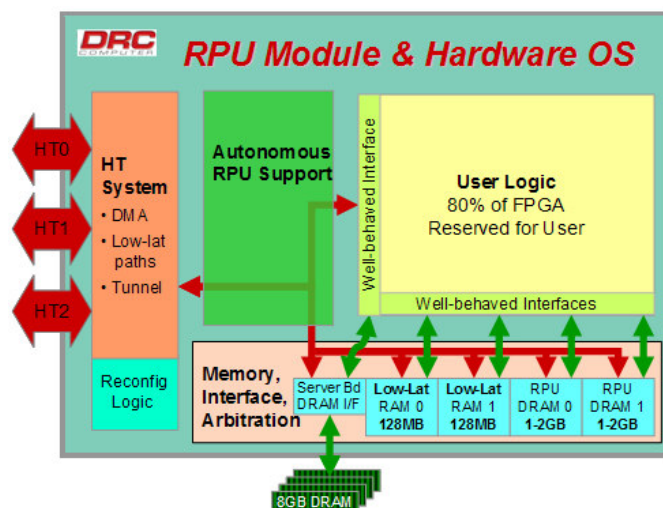
By implementing RPUs in the socket, commercial users stand to see the benefits of reconfigurable computing in the coming years as well, since this model makes it easier and less expensive for commercial vendors to incorporate reconfigurable processors. The standards-based aspect of modular reconfigurable computing allows server manufacturers to benefit from advances in FPGA and memory technology without having to redesign their base machines. They need only choose the latest reconfigurable modules from those who make it their business to adapt new reconfigurable technologies to the standard sockets.

Streamlining Reconfigurable Computers

While the advantages of RPU acceleration are too compelling to be ignored, implementing an RPU function today requires additional steps beyond standard software development. If existing software code is being converted to run on the reconfigurable computer, the developer will profile the code to find the portions requiring the most computation and targeting those portions for acceleration through the software-to-HDL (hardware design language) tool chain. If new code is being developed, the "hot spots" are often known in advance, but surprises often occur. The higher-level program calls to these subroutines are adapted to pass parameters through the RPU software APIs to interact with user logic installed on the RPU. If reconfigurable computing is to reach its full potential in the market, this process must be made as simple and straightforward as possible. Both RPU vendors and developers of RPU tools have important roles to play in making this a reality.

For the RPU vendor, the largest part of providing the reconfigurable element is providing standardized hardware and software interfaces for the RPU -- that is, providing the RPU Hardware OS. To facilitate the process of developing reconfigurable algorithms, the RPU vendor provides all the difficult high-speed interfaces to processor bus and memories, normalizing these interactions to what appear to the compilers (or aggressive expert users) as well behaved busses, regardless of the speed at which the implemented user logic runs. The CPU messaging and DMA usage is also abstracted from the user, bypassing DMA when low-latency calls are in order. Some FPGA vendors allow these interfaces to be locked down so that their timings need not be perturbed with each new build of user logic, greatly shortening compilation time.

The RPU vendor must also provide the ability to load new user logic on demand -- and to do so quickly enough to make the extra steps involved in using the RPU worth while. FPGAs in embedded systems are generally loaded from a PROM associated with the device. In a reconfigurable computer, the CPU loads the user logic as required by the current program running on the CPU. This run-time reconfiguration (RTR) is key to reconfigurable computing and the time required to rewrite new firmware, typically a couple of hundred milliseconds, in many ways determines the granularity of tasks for which the RPU will be used. Obviously, a task will want to run in real time at least 10 times longer than its load time to make the conversion worthwhile. In HPC, tasks are running for hours and days and therefore are amenable to this treatment. As RPU vendors begin to provide partial RTR capability, where only a small portion of the coprocessor is rewritten in a few milliseconds, other classes of tasks will become viable on reconfigurable computers.



In addition to RPUs themselves becoming easier to use, tools to allow software-to-hardware conversion are improving rapidly and, with standard reconfigurable computers available, that improvement will accelerate. Vendors such as Celoxica and Impulse Accelerated Technology translate C or a C variant to HDL. Both do creditable jobs on the conversion. Both also provide controls that will allow those inclined to become expert in the process to direct the tool to more efficient implementation. Perhaps very interesting to financial algorithm developers a considerable amount of work has been done on automatic conversion of Matlab to HDL, potentially allowing quick conversion of algorithms from that language of choice into the RPU.

As important as these automatic tools (and, in fact, contributing to their maturation) is the community of configware developers beginning to take shape around reconfigurable computing and their implementations of the core algorithms that will make reconfigurable computing easier to employ. Universities have done much research in this area, and the literature is rich with work to be further productized. In addition, more and more trained practitioners are being graduated with fully developed reconfigurable computing mindsets. Also, many development groups have partnered with standard reconfigurable computer providers, now checked out and ready to provide conversion services to end-user customers while the tools are firming up for standard software practitioners..

Standards Align Industry Energy

While the evolution of viable, widespread reconfigurable computing has been a slow process, the convergence of open standards with RPU logic size and new tools capabilities is making the payoff for reconfigurable computing more compelling to growing numbers of users and developers. Over the next two years, research now being productized will produce ever easier and more automatic RPU implementations, making reconfigurable computing standard practice in HPC. With the advent and market uptake of the standard reconfigurable computer, FPGA architectures are responding ever faster to the needs of HPC. Tools improvement is also accelerating with critical mass development beyond research, and macro function components are becoming increasingly available on a license and open-source basis. It is really the press of accumulated intellectual development and industry-wide financing that forces breakthroughs and drives technologies into the

mainstream. As reconfigurable computing standards and new tools and capabilities align, a new era of reconfigurable computers is now on the horizon.

About the Author

Michael R. D'Amour is Chief Operating Officer with DRC Computer Corporation. Prior to joining DRC, Michael D'Amour was co-founder and executive vice president of R&D and international operations at Quickturn Design Systems, the industry's largest and most successful reconfigurable emulation Electronic Design Automation (EDA) company. Previously, he was a founding team member at Daisy Systems Corporation where he served as vice president of R&D for multiple generations of the company's EDA products. He brings exceptional start-up experience to DRC and a background in reconfigurable technology that includes delivery of over \$500 million in reconfigurable systems to the marketplace while at Quickturn.

Leading HPC Solution Providers



HPCwire Readers' and Editors' Choice Awards

