

A Novel Processor Architecture for FPGA Supercomputing

The Mitrion Virtual Processor is a fine-grain massively parallel reconfigurable processor for FPGAs.

by Stefan Möhl
CTO
Mitrionics Inc.
stefan.mohl@mitrionics.com

Göran Sandberg
Vice President of Product Marketing
Mitrionics Inc.
goran.sandberg@mitrionics.com

Ever-increasing computing performance is the main development driver in the world of supercomputing. Researchers have long looked at using FPGAs for co-processors as a promising way to accelerate critical applications. FPGA supercomputing is now generating more interest than ever, with major system vendors offering general-purpose computers equipped with the largest Xilinx® Virtex™-4 FPGAs, thus making the technology accessible to a wider audience.

There are compelling reasons to give serious consideration to FPGA supercomputing, especially given the significant performance benefits you can achieve from using FPGAs to accelerate applications. Typical applications can be accelerated 10-100x compared to CPUs, making FPGAs very attractive from a price/performance perspective.

What is even more important is that FPGAs use only a fraction of the power per computation compared to CPUs. Increased computing power with traditional clusters requires more and more electrical power. Today, the cooling problems in large computing centers have become major obstacles to increased computing performance.

Plus, the development of FPGA devices is still keeping up with Moore's law. In practice, this means that the performance of FPGA-based systems could continue to double every 18 months, while CPUs are struggling to deliver increased performance.

FPGAs as Supercomputing Co-Processors

FPGAs are highly flexible general-purpose electronic components. Thus, you must consider their particular properties when using them to accelerate applications in a supercomputer.

Compared to a CPU, FPGAs are Slow

It may seem like a contradiction to claim that FPGAs are slow when they can offer such great performance benefits over CPUs. But the clock speed of an FPGA is about 1/10th of a CPU. This means that the FPGA has to perform many more operations in parallel per clock cycle if it is to outperform a CPU.

FPGAs are Not Programmable

(From a Software Developer's Perspective) Software developers taking their first steps in FPGA supercomputing are always surprised to find out that the P for programmable in the acronym FPGA means "you can load a circuit design." Without such a circuit design, an FPGA does nothing – it is just a set of electronic building blocks waiting to be connected to each other.

Creating a circuit design to solve a computational problem is not "programming" from a software developer's point of view, but rather a task for an experienced electrical engineer.

FPGAs Require Hardware Design

With their general-purpose electronic components, FPGAs are usable in any electronic hardware design. That is why they work so

well as computing co-processors in computers, provided that you have loaded the proper circuit design. Traditional development methods for FPGAs are focused on the hardware design aspects, which are very different from software development (see Table 1).

FPGAs in Supercomputers

Enable a New Paradigm

Putting FPGAs in computers to accelerate supercomputing applications changes the playing field and rules dramatically. Traditional hardware design methods are no longer applicable for several reasons:

- Supercomputer programmers are typically software developers and researchers who would prefer to spend their time on research projects rather than acquiring the expertise to design electronic circuits.
- The sheer complexity of the designs required to compute real-world supercomputing problems is prohibitive and usually not viable.
- Hardware design methods are targeted at projects where a single circuit design is used in a large number of chips. With FPGA supercomputers, each chip will be configured with many different circuit designs. Thus, the number of different circuit designs will be greatly multiplied.
- Hardware design tools and methods focus on optimizing the size and cost of the final implementation device. This is significantly less important in a supercomputer because the investment in the FPGA has already occurred.
- Programs run in supercomputers are often research tasks that continuously change as the researcher gains deeper

understanding. This requires the program to evolve over its life cycle. In contrast, circuit designs are usually created once and then used without changes for the life span of the target application.

The good news is that in a supercomputer, the FPGA is there to perform computing. This allows you to create software development methods for FPGAs that circumvent the complexity and methodology of general hardware design.

The Mitrion Virtual Processor

The key to running software in FPGAs is to put a processor in the FPGA. This allows you to program the processor instead of designing an electronic circuit to place in the FPGA. To obtain high performance in an FPGA, the circuitry of the processor design is adapted to make efficient use of the resources on the FPGA for the program that it will run. The result is a configuration file for the FPGA, which will turn it into a co-processor running your software algorithm. This approach allows you as a software developer to focus on writing the application instead of getting involved in circuit design. The circuit design process has already been taken care of with the Mitrion Virtual Processor from Mitronics.

A Novel Processor Architecture

How do you get good performance from a processor on an FPGA when typical FPGA clock speeds are about 10 times slower than the fastest CPUs? Plus, the FPGA requires and consumes a certain amount of overhead for its reconfigurability. This means that it has a 10-100x less efficient use of available chip area compared to a CPU. Still, to perform computations 10-100x faster than a CPU, we need to put a massively parallel processor onto the FPGA. This processor should also take advantage of the FPGA's reconfigurability to be fully adapted to the program that it will run. Unfortunately, this is something that the von Neumann processor architecture used in traditional CPUs cannot provide.

To overcome the limitations of today's CPU architecture, the Mitrion Virtual Processor uses a novel processor architecture that resembles a cluster-on-a-chip.

Hardware Design	Software Development
Driven by Design Cycle	Driven by the Code-Base Life Cycle
Main Concern: Device Cost (Size, Speed)	Main Concern: Development Cost and Maintenance
Precise Control of Electrical Signals	Abstract Description of Algorithm

Table 1 – Hardware/software development comparison

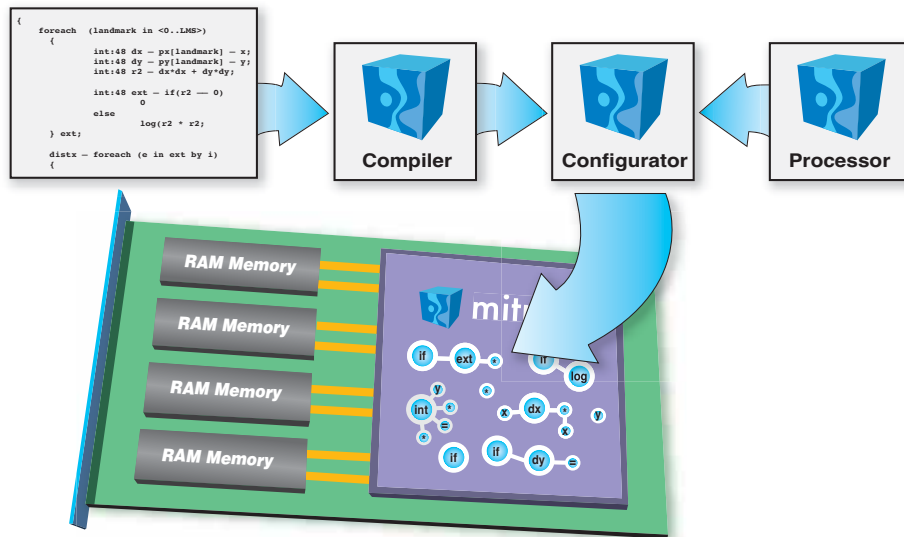


Figure 1 – The Mitrion Platform

Normal computer clusters comprise a number of compute nodes (often standard PCs) connected in a fixed network. The basic limitation of such a cluster is the network latency, ranging from many thousands up to millions of clock cycles, depending on the quality of the network. This means that each node has to run a large block of code to ensure that it has sufficient work to do while waiting for a response to a message. Lower latency in the network will enable each node to do less work between communications.

In the architecture of the Mitrion Virtual Processor, the entire cluster is on the same FPGA device. The network is fully adapted in accordance with the program requirements, creating an ad-hoc, disjunct network with simple point-to-point connections where possible, switched only where required. This allows the network to have a guaranteed latency of a single clock cycle.

The single clock cycle latency network is the key feature of the Mitrion Virtual Processor architecture. With a single-cycle latency network, it becomes possible for the nodes to communicate on every clock cycle. Thus, the nodes can run a block of code comprising only a single instruction. In a node that runs only one instruction, the instruction scheduling infrastructure of the node is no longer necessary, leaving

only the arithmetic unit for that specific instruction. The node can also be fully adapted to the program.

The effect of this full adaptation of the cluster in accordance with the program is that the problem of instruction scheduling has been transformed into a problem of data-packet switching. For any dynamic part of the program, the network must dynamically switch the data to the correct node. But packet switching, being a network problem, is inherently parallelizable. This is in contrast to the inherently sequential problem of instruction scheduling in a von Neumann architecture.

This leaves us with a processor architecture that is parallel at the level of single instructions and fully adapted to the program.

Mitrion-C

To access all of the parallelism available from (and required by) the Mitrion Virtual Processor, you will need a fully parallel programming language. It is simply not sufficient to rely on vector parallel extensions or parallel instructions.

We designed the Mitrion-C programming language to make it easy for programmers to write parallel software that makes the best use of the Mitrion Virtual Processor. Mitrion-C has an easy-to-learn C-family syntax, but the focus is on describing data dependencies rather than order of execution.

The syntax of Mitrion-C is designed to help you achieve high performance in a parallel machine, just like ANSI C is designed to achieve high performance in a sequential machine. Thus the Mitrion-C compiler extracts all of the parallelism of the algorithm being developed. We should note, though, that Mitrion-C is purely a software programming language; no elements of hardware design exist.

The Mitrion Platform

Together with Mitrion-C and the Mitrion Virtual Processor, the Mitrion Software Development Kit (SDK) completes the Mitrion Platform (see Figure 1). The Mitrion SDK comprises:

- A Mitrion-C compiler for the Mitrion Virtual Processor
- A graphical simulator and debugger that allows you to test and evaluate Mitrion-C applications without having to run them in actual FPGA hardware
- A processor configuration unit that adapts a Mitrion Virtual Processor to the compiled Mitrion-C code

The Mitrion Platform is tightly coupled to Xilinx ISE™ Foundation™ software for synthesis and place and route, targeting Xilinx Virtex-II and Virtex-4 devices. It features a diagnostic utility that analyzes the output from these applications and can report any problems it encounters in a format that does not require you to have FPGA design skills.

Conclusion

The novel, massively parallel processor architecture of the Mitrion Virtual Processor to run software in an FPGA is a unique solution. It is a solution readily available on the market today that addresses the major obstacles to the widespread adoption of FPGAs as a means to accelerate supercomputing applications. With the Mitrion Virtual Processor, you will not need hardware design skills to make software run in FPGAs.

For more information about the Mitrion Virtual Processor and the Mitrion Platform, visit www.mitrionics.com or e-mail info@mitrionics.com