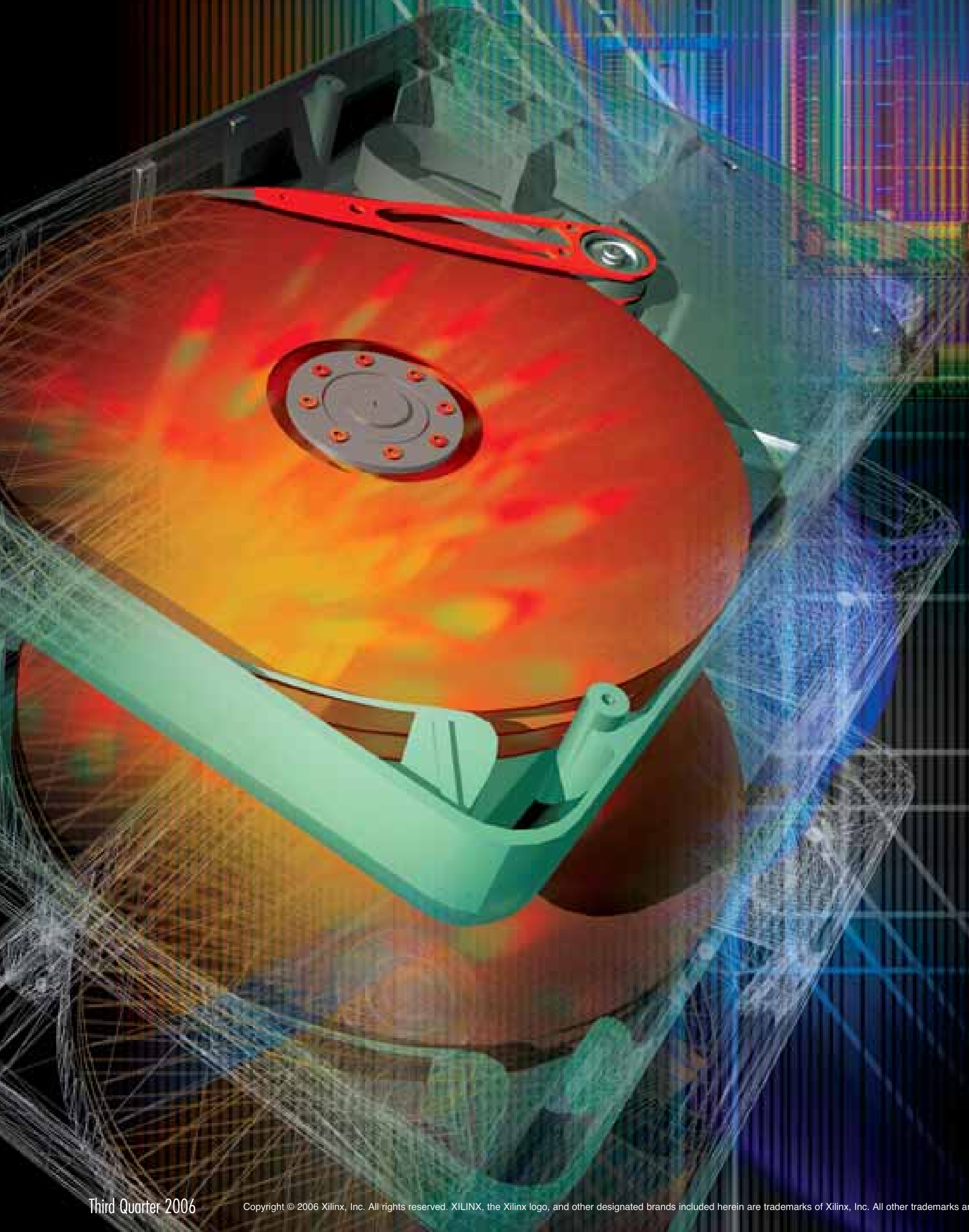
# Scalable Cluster-Based FPGA HPC System Solutions

Nallatech introduces an HPC family of FPGA system solutions optimized for clustering, with performance capability rising to multi-teraflops in a single 19-inch rack.



by Allan Cantle
President and Founder
Nallatech
*a.cantle@nallatech.com*

The high-performance computing (HPC) markets have taken a serious interest in the potential of FPGA computing to help solve computationally intensive problems while simultaneously saving space and reducing power consumption. However, to date few capable products and tools are tailored to suit HPC industry requirements.

In the last 15 years, the industry has also moved away from customized, massively parallel computing platforms (like those offered by Cray and SGI) to clusters of industry-standard servers. The majority of HPC solutions today are based on this clustered approach.

To serve the HPC market, Nallatech has introduced a family of scalable cluster-optimized FPGA HPC products, allowing you to either upgrade your existing HPC cluster environments or build new clusters with commercial-off-the-shelf FPGA computing technology.

Nallatech has developed a tool flow that also allows HPC users to continue to use a standard C-based development environment to target FPGA technology. Once familiar with this basic tool flow, you can then learn more about FPGA computing, leveraging more advanced features to fully exploit the technology.

### A Scalable FPGA HPC Computing Platform

The traditional approach to HPC computing comprises a large array of microprocessors connected together in a symmetrical fashion using the highest bandwidth and lowest latency communications infrastructure possible. This approach provides a highly scalable structure that is relatively easy to manage.

Although easily scalable and manageable, these regular computing structures are rarely the best architecture to solve different algorithmic problems. Algorithms have to be tailored to fit the given computing platform.

With FPGAs, by contrast, you can build a custom computer around any specific algorithmic problem. This approach yields significant benefits in performance, power, and physical size and has been widely adopted in the embedded computing community. However, embedded computers are typically designed with one application in mind – and therefore a custom computer is crafted for each application.

Nallatech's FPGA HPC architectures are designed to merge the best of both worlds from the embedded and HPC computing industries. Our FPGA HPC architecture is symmetrical with regular communications between FPGA processors, but also has an inherent infrastructure that allows you to handcraft custom computing architectures around given algorithmic problems.

Recognizing the popularity of clustering, Nallatech selected and targeted the architecture for IBM BladeCenter computing platforms. With this approach, you can mix and match traditional microprocessor blades and FPGA computing blades in a proportion that suits the amount of code targeted at the FPGA.

Figure 1 shows the four basic types of configurations possible with Nallatech's FPGA HPC computing platforms. The wide scaling capability allows you to scale the FPGA computing resource in a way that is identical to standard cluster scaling. The deep and wide scaling capability of this architecture additionally permits scaling to more complex custom computing algorithms by treating multiple FPGAs as a single, large FPGA.

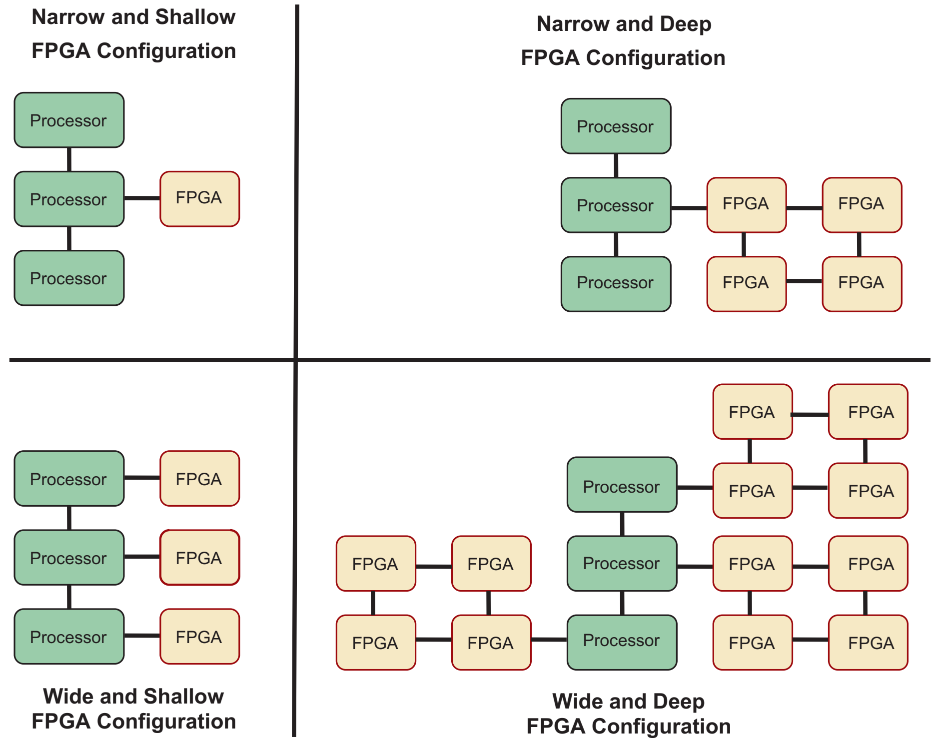



*Figure 1 – Four types of processor-to-FPGA cluster configurations*

A 7U IBM BladeCenter chassis with a protruding Nallatech FPGA computing blade is shown in Figure 2. You can mix this FPGA blade with host processor blades in configurations similar to those detailed in Figure 1. Each blade is configurable with between one and seven FPGA processors providing the deep scaling. Several blades installed into one or more BladeCenter chassis provide the wide scaling necessary for more complex HPC applications.

Floating-point implementations of code in the fields of seismic processing and computational fluid dynamics have demonstrated performance improvements more than 10 times that of standard processors, while achieving power savings well in excess of 20 times.

*Figure 2 – IBM BladeCenter chassis with protruding Nallatech FPGA blade*

### Creating Custom Virtual Processors

Nallatech's approach to FPGA HPC computing provides an extremely flexible architecture that enables you to create exceptionally high-performance computing solutions with a highly efficient use of power and space. However, without the appropriate development tools, using this type of platform would be virtually impossible. As mentioned previously, any physical instantiation of Nallatech's FPGA computing platform can effectively be treated as either:

- A number of physical FPGA processors that exist in the computer connected together
- One massive FPGA processor combining multiple FPGAs and creating one custom compute engine with these processor chips
- Thousands of tiny processors with many processors existing in each FPGA

Whenever a new design commences, it is counterproductive to think of the problem in terms of the physical computing architecture at hand. Instead, consider how you can construct a custom computer to solve the specific compute problem. Effectively, you can develop a virtual processor that is explicitly designed to solve your specific algorithmic problem.

There are many ways to create a virtual processor for FPGAs, including:

- Using standard language compilers to FPGAs such as C and Fortran, pulling together various computing library functions available for the FPGA
- Using high-level algorithmic tools such as Simulink from The MathWorks's together with Xilinx® System Generator
- Designing a dedicated processor from low-level hardware languages such as VHDL

Most HPC software engineers prefer the language-based approach for obvious reasons, so language-based compilers are now maturing and becoming a viable approach to creating virtual processors.

As a high-performance FPGA computing systems provider, Nallatech has taken an approach to try to support as many compiler tools as possible. Today, you can develop virtual processors using three approaches:

- Nallatech's DIME-C enabling compiler
- Impulse Technologies's Impulse C compiler
- Xilinx System Generator for The MathWorks's Simulink environment

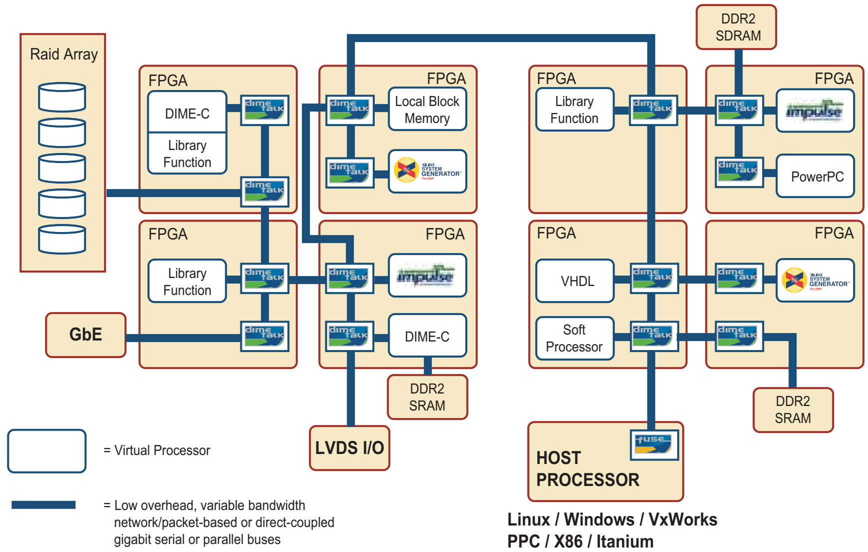



*Figure 3 – Example DIMEtalk network implementing many different virtual processors*

Nallatech's DIME-C compiler is purely based on a subset of ANSI-C, with no requirement for pragmas or other specific actions. This means that you can design and simulate code with the tools with which you are familiar. Once the C code is functioning correctly, you have the freedom to compile directly through to a bitstream and run the code on a chosen FPGA processor. Nallatech tools provide all of the APIs necessary to access the FPGA through simple function calls from the host processor.

## Maximizing Data Throughput and Integrating Virtual Processors

In many cases, creating a virtual FPGA processor using a language-based software environment and implementing the resulting executable on the allocated FPGA coprocessor can substantially improve the performance of the chosen application.

However, this approach does not necessarily yield the most optimum implementation. You can often benefit from splitting your algorithmic problem into several independent processes. In these circumstances it is often worth it to create a virtual processor for each process, as this will allow you to decouple one processor from the other and give you more architectural implementation flexibility.

DIMEtalk allows you to effectively place many virtual processors across the FPGA computing resources in an HPC platform while minimizing the communications overhead between processing elements. DIMEtalk also allows you to effectively maximize the four levels of memory hierarchy that Nallatech's FPGA HPC platforms provide.

Figure 3 shows an example of a complex compute engine implemented across multiple FPGAs in the HPC platform. There are several virtual processors constructed from different compiler tools; these are connected together through the DIMEtalk packet-switched communications network.

## Conclusion

FPGA computing is deliverable on familiar cluster-based computing platforms; plus, with an ANSI-standard C compiler, you can achieve substantial performance gains. Looking forward, FPGA technology will likely advance HPC performance at a far greater rate than traditional processor-based technologies. For traditional software engineers, compiler tools will also become far more advanced and easy to use.

Nallatech is providing a special bundle of its entry-level HPC card with complete enabling tools software and a DIME-C to FPGA run time tool flow. Please visit *www.nallatech.com/hpc* for more details or to place an order.

For additional information, contact Nallatech at *hpc@nallatech.com* or *contact@nallatech.com.*