White Paper

Accelerating High-Performance Computing With FPGAs

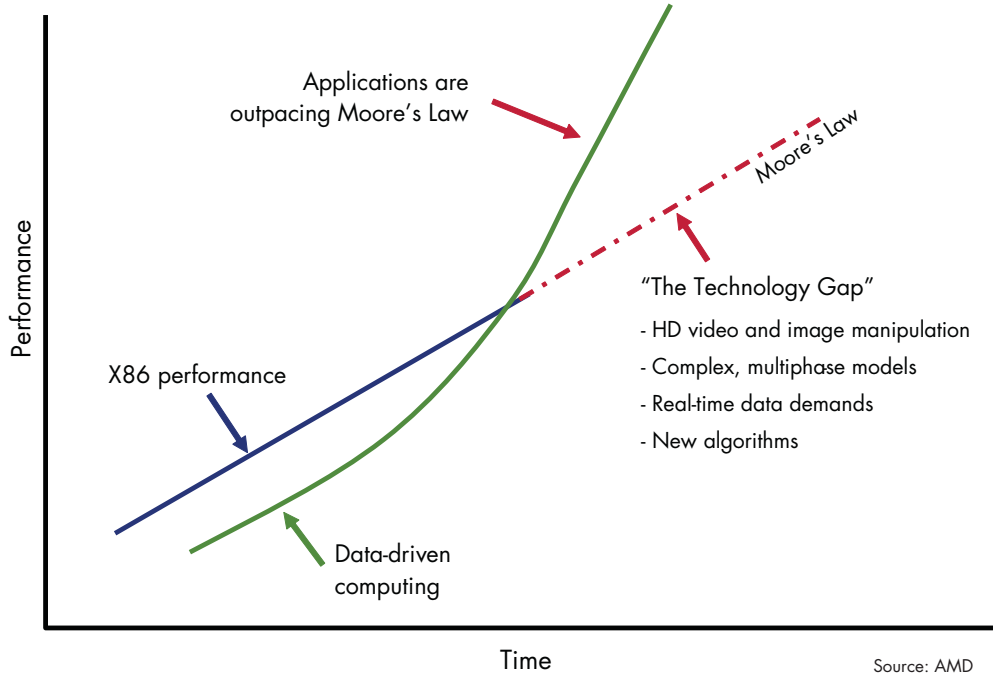
Introduction

Application demands have outpaced the conventional processor's ability to deliver. The solution is hardware acceleration that augments processors with application-specific coprocessors. The right combination of price, performance, and ease-of use, along with significant power savings, comes from using FPGAs as the basis of coprocessor design.

Since its invention nearly four decades ago, the microprocessor has made huge performance strides. Performance doubled every 18 months or so during those decades, a trend affectionately known as “Moore's Law.” As a result, processors have risen from a near-play toy to devices surpassing the performance of the once-exotic Cray-1 supercomputer.

Each new round of processor improvements has made more complex and demanding applications achievable, and expanded user expectations for the next round. For most of the microprocessor's history, application demands have risen in response to processor improvements, allowing processors to stay ahead of demand. In the last few years, however, the situation has changed. High-performance computing (HPC) applications are now demanding more than processors alone can deliver, creating a technology gap between demand and performance (see Figure 1).

Figure 1. Industry Landscape: The Technology Gap



There are many reasons for this dramatic upturn in applications demand. One example is in financial analytics, where financial options trading houses seek to gain a competitive advantage in the market by accelerating their applications. Financial application acceleration enables the financial trading houses to make trades faster and with more accuracy than the competition. The application speed-up enables the financial traders to achieve better results faster, resulting in big financial gains for the trading house. Achieving the necessary performance boost requires at least a doubling of processing power to achieve, so that improvement on all fronts together requires an order of magnitude more processing power.

Many traditional applications have increased their demand on processors by implementing more and more complex algorithms. Algorithms for communications encoding and decoding (CODEC), data compression and warehousing, and encryption, for instance, have increased dramatically in complexity in order to increase to applications capability. Some are even following a self-stimulating spiral in complexity. Security, for instance, continually requires ever-more sophisticated encryption as hackers learn to bypass existing measures.

Applications Pushing Performance Envelope

A number of new applications have also arisen and become widespread as their performance thresholds were met. Medical imaging, including ultrasound, computer-aided tomography (CAT) scanning, and magnetic resonance imaging (MRI), arose when processing power reached high enough levels, and have continually demanded increased resolution ever since. Scientific simulation and modeling have likewise appeared and become mainstream, with ever-increasing model complexity. Even the financial services community has embraced HPC, using models and simulation to provide real-time assistance in financial decision-making.

In addition to these demands for increased capability has been a demand for faster results. Some of this has been due to rising user expectations that set new baselines for applications performance-users have come to expect that applications will keep running faster and have become intolerant of low responsiveness. But another, and even stronger, push to exceed current capacity is now coming from the competitive advantage that compute speed can provide. In gene-based medical research, for instance, the ability to get simulation results faster can make the difference between being first-to-market and being an also-ran. In the financial market's computer-controlled derivatives trading, milliseconds can mean millions in profit or loss. Higher performance now equals higher profits, and the market is not willing to wait for processors to rise to the occasion.

Even as performance demand increases have begun exceeding Moore's Law, processors have begun faltering. Much of the performance increase seen in the last 40 years has been the result of improvements in semiconductor technology, leading to better throughput and system performance. But such technology scaling is reaching hard limits. As circuits become more complex, development costs have risen to millions of dollars per design, and the creation of new production capacity has risen to billions of dollars per plant. As clock speeds increase, device power dissipation also rises and has now reached levels where the heat cannot be removed fast enough.

Fortunately, increasing processor performance is not the only way to address rising application demands. Augmenting a processor with an application-specific processor has long proven to be a viable approach to solving performance shortfalls. Use of such coprocessors began with specialized I/O processing such as modems and Ethernet controllers, extending to the use of graphics rendering engines as display demands rose and encryption engines for increased security. More general-purpose coprocessors also arose, first in the form of math accelerators to handle multiplication and division. Digital signal processors then arose as coprocessors designed to handle complex mathematical algorithms by incorporating built-in math hardware and employing new architectures that featured pipeline and parallel structures.

Such specialized structures are not quite the answer to today's technology gap, however, because they address only a single aspect of the problem. This has been acceptable in the past because processor performance was adequate for most purposes and only used assistance in specific applications where the costs and benefits made coprocessors attractive. Addressing today's technology gap requires a more diverse set of attributes than traditional coprocessors provide. Further, designs need to avoid software-based implementations such as digital signal processors because they face the same scaling limitations as the main processor.

Application-Specific Coprocessors Required

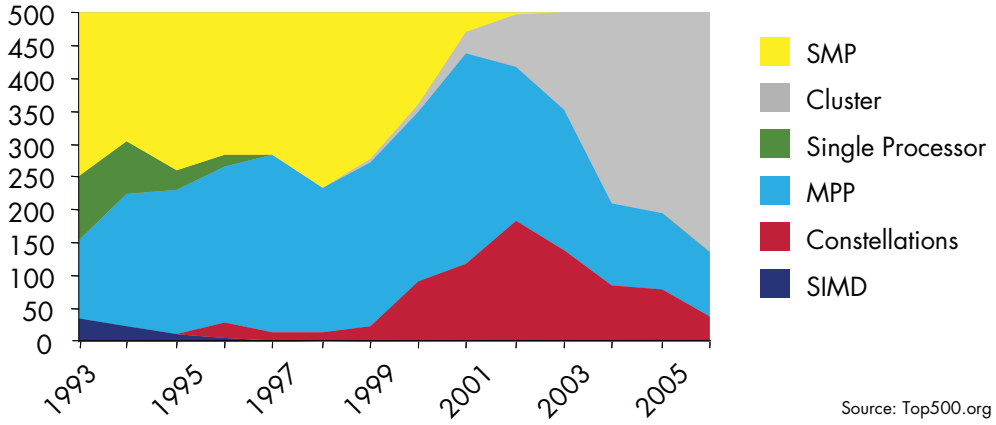
Today's ideal coprocessor must be a hardware-based design that provides three major capabilities. One is that the design needs to provide specific hardware acceleration for key processes within the application, whatever the application may be. Another is that the coprocessor design needs to be scalable in performance through the use of pipeline and parallel structures so that it can keep pace with rising performance demands. Finally, the coprocessor needs to provide high-bandwidth, low-latency interfaces to both the main processor and to system memory.

In addition to the hardware requirements, the ideal coprocessor needs to address the "four Ps" of HPC market needs: performance, productivity, power, and price. The performance need in the HPC market is at least for the acceleration of whole algorithms, not just individual steps, and ideally for the acceleration of the entire application. The productivity need comes from the end user. The coprocessor must be easy to install in existing computer systems and provide a simple method for configuring the system to accelerate existing applications software.

The power need of the HPC market arises from limitations that power consumption imposes on the installation and use of computing systems. For many users, there is a fixed amount of space available for the computers. The less power a computing system requires, then, the less cooling is needed to keep that space within appropriate temperature limits. Lower power for the coprocessor, therefore, represents either a lower operating cost for the computing systems or the ability to put more computing systems within the space.

Price is becoming an increasingly important element in the HPC market. A decade ago the few applications that needed performance beyond what a single processor could provide embraced specialized architectures such as massively parallel processing (MPP) and symmetric multiprocessing (SMP) (see Figure 2). Such systems required the use of custom processor elements and proprietary data pathways, however, and were expensive both to build and to program.

Figure 2. HPC Architectures Shift Toward Cluster Computing



Notes:
(1) SMP = symmetric multiprocessors, MPP = MOTIVE model compilation, and SIMD = single instruction multiple data

Today's HPC market has abandoned such expensive approaches in favor of the more cost-effective cluster computing approach. Cluster computing uses commodity standard architectures such as those from Intel and AMD, industry-standard interconnects such as Gigabit Ethernet and InfiniBand, and standard programming languages such as C running applications under low-cost Linux operating systems. Today's coprocessor design, then, must integrate smoothly into the commodity cluster computing environment with a cost point comparable to adding another node to the cluster.

Designing a coprocessor to provide application-specific hardware acceleration represents a major challenge under these market conditions. Developers have addressed application needs in the past by designing different coprocessors for each of several general application spaces, such as graphics and Ethernet controllers. Long development times and high development costs preclude this approach, however, because the payback is too low for most applications and users do not want to wait. Further, the diversity in applications needing acceleration, along with the need to integrate smoothly into existing computing clusters, makes the traditional approach cost-prohibitive.

FPGAs Power Ideal Implementation

There is, however, an alternative to designing a different coprocessor for each application need: use an FPGA that can be configured as needed for each application. The same semiconductor technology advances that have brought processors to their performance limits have turned FPGAs from simple glue logic collectors to highly capable programmable fabrics. FPGAs are fully capable of addressing the “four Ps” of the HPC market.

Today's FPGAs offer tremendous performance potential. They can support pipeline structures of variable depth and provide parallel compute resources numbering in the thousands, allowing even highly complex functions to be implemented with single-clock execution. The programmability of FPGAs ensures that they can be tuned to meet the specific needs of an application without the cost or delay of designing a custom coprocessor. If the FPGA is also reprogrammable, it can provide highly customized coprocessing for a wide range of applications in a single chip.

The presence of on-board memory in an FPGA also has significant performance benefits. For one, having memory on-chip means that the coprocessor logic's memory access bandwidth is not constrained by the number of I/O pins the device has. Further, the memory is closely coupled to the algorithm logic and reduces the need for an external high-speed memory cache. This, in turn, avoids power-consuming cache access and coherency problems. The use of internal memory also means that the coprocessor does not need additional I/O pins in order to increase its accessible memory size, simplifying design scaling. An FPGA with greater capacity can occupy the same board footprint as an older device, allowing performance upgrades without board changes.

As a result of the structures and resources available in today's high-performance FPGAs (i.e., the Altera® Stratix® III family of FPGAs), they can serve as hardware acceleration coprocessors for a wide array of applications and provide significant performance boosts. As shown in Table 1, practical experience with FPGA-based coprocessors shows at least a ten-fold improvement in the execution speed of algorithms as compared to processors alone. Speeds more than 100 times faster are common.

Table 1. FPGA Algorithm Acceleration

Application	Processor Only	FPGA Coprocessing	Acceleration Factor
Hough and Inverse Hough Processing *(1)*	12 minutes processing time Pentium 4-3 GHz	2 seconds of processing time at 20 MHz	370X
Spatial Statistics (Two-Point Angular Correlation Cosmology) *(2)*	3,397 CPU hours with 2.8-GHz Pentium (approximate solution)	36 Hours (exact solution)	96X
Black-Scholes (Financial Application (single precision FP 2M points)) *(2)*	2.3M experiments/sec with a 2.8-GHz Processor	299M experiments/sec	130X
Smith Waterman SS search 34 from FASTA *(1)*	6461 sec processing time (Opteron)	100-sec FPGA processing	64X
Prewitt Edge Detection (compute intensive video and image processing) *(3)*	327M clocks (1-GHz processing power)	131K clocks at 0.33 MHz	83X
Monte Carlo Radiative Heat Transfer *(1)*	60-ns processing time (3-GHz processor)	6.12 ns of processing time	10X
BJM Financial Analysis (5M paths) *(1)*	6300 sec processing time (Pentium 4-1.5-GHz)	242 sec of processing at 61 MHz	26X

Notes:
(1) Source: Celoxica
(2) Source: SRC Computers
(3) Source: XtremeData

Tools Simplify Customization

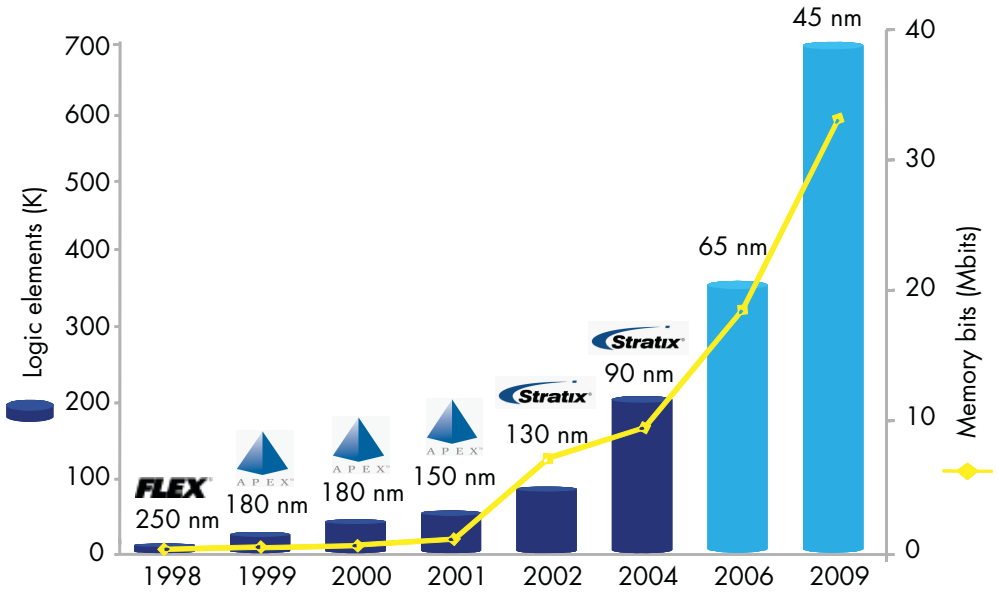
As compelling as these performance boosts appear, they would be of limited value if they required significant development effort to obtain. Fortunately, FPGAs have the support of many mature development tools, such as C-to-hardware tools from ImpulseC and SRC Computers coupled with the superior place-and-route tool found in Altera’s Quartus® II development software, that simplify and streamline the process of turning FPGAs into application-specific coprocessors. Code analysis tools can take a user's C code and identify functions that would benefit from hardware acceleration. Compilers can automatically structure the object code for these functions for

parallel and pipelined execution to maximize the effectiveness of acceleration. Design tools can then map that object code into a hardware description language (HDL) that FPGA layout tools automatically turn into the final coprocessor design.

The availability of this full design tool chain means that users do not have to struggle to accelerate their applications. The tool chain can take their existing code and automatically configure the FPGA coprocessor to provide significant performance enhancements. Further, these tools do not require the user to have FPGA hardware expertise in order to be effective, nor do they require that users rewrite their source code for coprocessing. This scenario is ideal for applications, such as financial services, where the software is tightly regulated and changes involve costly or time-consuming re-certification.

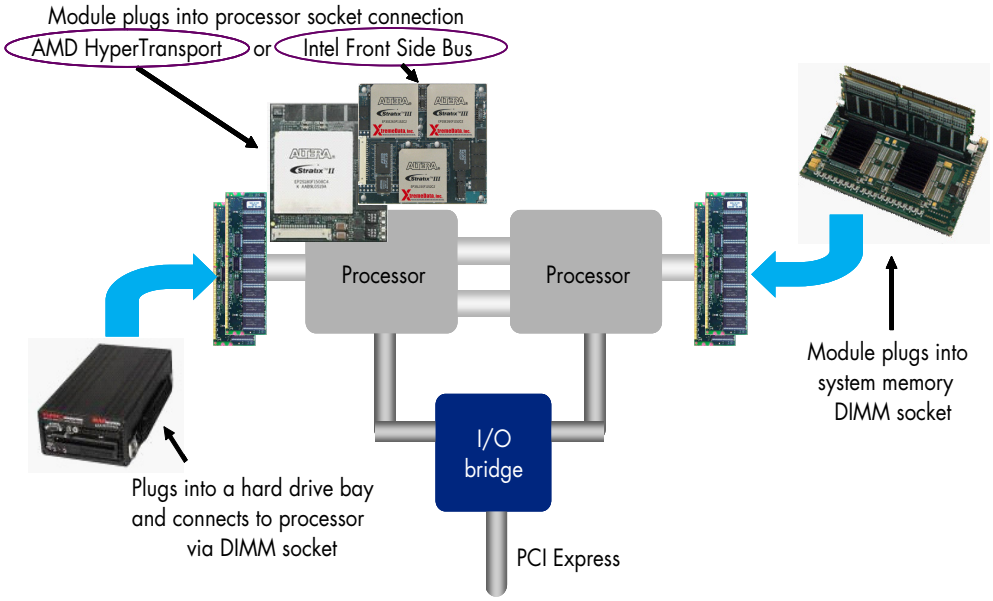
For power, the third "P," FPGAs have a significant advantage over processors. The massive parallel resources of an FPGA allow the hardware to execute functions in only a few clock cycles instead of the hundreds to thousands of clock cycles that the sequential operation of a processor would require. Because so few clock cycles are required, FPGAs can operate with much slower clocks and still provide a performance boost. The lower clock speeds result in lower power consumption, making an FPGA coprocessor much more power-efficient than a processor (see Figure 3).

Figure 3. FPGA Memory and Bandwidth Continue to Scale



Price is the fourth "P," and here, too, FPGAs meet the challenge. The cost of an FPGA coprocessor is comparable to, and often less than, that of a processor with similar performance. As a result, the component cost of a processor and FPGA coprocessor is no more than that of two processors in a standard cluster design. The FPGA coprocessor also offers many options for its addition to a standard cluster design without added cost. These options include replacement of a processor in a multiprocessor board design, connection to the processor though the memory interface, and insertion as a high-performance peripheral card (see Figure 4). An FPGA simply has to implement the appropriate interface as part of its fabric.

Figure 4. Coprocessing Solutions



Altera's Products Perfectly Positioned

Altera's Stratix III family of FPGAs is ideally suited to serve as HPC coprocessing elements and offers several advantages over other FPGAs. One advantage is the balanced mix of logic and DSP blocks. Because of this balance, the Stratix III FPGA is ideally suited for high-performance functions such as double-precision FP. In addition, the Stratix III FPGA is field configurable, so that a coprocessor design can be adapted to whatever application a user wants to run, allowing it to continue providing acceleration even as the user's needs change.

Another advantage of Stratix III FPGAs is the columnar architecture, which is similar to that used by memory suppliers. The use of this architecture allows Altera to create the highest-density parts available for a given semiconductor process. This not only provides a high logic gate count, it ensures adequate on-chip memory resources for any application.

Stratix III FPGAs provide a balanced mix of adder logic and multipliers in its resource matrix. This mix makes them ideal for compute-intensive applications that require matrix multiplication and double-precision floating-point mathematics. Representative implementations of an IEEE-standard floating-point multiplication algorithm, for instance, have produced a peak performance measured at 48 billion operations per second (GFLOPs). In addition to the balanced resource mix, Stratix III FPGAs offer the benefit of the Avalon® routing fabric. This universal interconnect structure allows the ready creation of optimum pipeline structures.

Altera's advantages go beyond the silicon. The company enjoys close partnerships with major processor vendors like Intel and AMD, allowing it to offer high-performance, low-latency interfaces to their processor families. Altera's relationship with Intel, for instance, has resulted in the development of a Front Side Bus (FSB) *(1)* interface for the FPGAs to connect with Intel processors. Unlike the commonly used North Bridge (NB) bus for peripheral connections, an FSB allows the FPGA to act as though it were another processor in a multiprocessor design. Thus, Stratix III FPGAs can be configured to be a drop-in processor replacement in a standard cluster design, bringing hardware acceleration to a multiprocessor board with no hardware changes. Altera's relationship with Advanced Micro Devices (AMD) has resulted in a HyperTransport™ *(2)* interface that provides the same kind of drop-in replacement in an AMD-based design.

Looking Beyond Silicon

Beyond its relationships with semiconductor vendors, Altera has extensive partnerships throughout the design tool community. This ensures that a host of tools are available for creating application-specific coprocessors from Altera devices. Working with tool developer Impulse, for instance, has resulted in Impulse C, a design tool that compiles applications code in C and optimizes it for acceleration in FPGA logic. The output of Impulse C is compatible with Altera's Quartus II development software as well as Visual Studio, Eclipse, and other standard tools, so that developers can quickly create a coprocessor optimized for their application code. The compatibility that has resulted from Altera's tool partnerships can eliminate weeks of development effort optimizing a prototype design.

Altera also partners with product developers to ensure creation of best-in-class designs using Stratix III FPGAs. In the HPC market, Altera has worked with several product developers to create coprocessor modules that end users can readily insert into their computer systems to obtain an immediate performance boost. Xtreme Data, for instance, has developed a module that is a drop-in replacement for an AMD Opteron processor in motherboards, with no board changes required. Users can mix and match Opteron processors and FPGA coprocessors as needed in a multiprocessor design.

Another vendor, SRC, has worked with Altera to develop its MAP product family of coprocessors. These modules connect to both AMD and Intel processors through the memory bus interface, achieving a data bandwidth of 14 Gbytes/second. Smaller MAP modules are available that replace a DIMM memory card in its socket. Larger coprocessor assemblies are also available that fit into a disk drive bay and use the drive's power connection along with the memory card data interface to provide greater capacity. SRC also offers its Carte Tool Chain for converting C or FORTRAN programs to run faster on the FPGA coprocessor.

Conclusion

The coprocessors from Xtreme Data and SRC represent the next step in HPC. The competitive advantages that application speed gives to users ensures that the demand for speed will continue to outpace what processors alone can achieve. Coprocessors based on Stratix III FPGAs provide the high-speed, low-latency interfaces that hardware acceleration requires, while the tool chains and other support that Altera's partners offer simplify the creation of customized acceleration that HPC users demand. These tools and products are commercially available today as proven, high-performance solutions, and Altera's continuing partnership efforts ensure that these parts, products, and tools will continue to keep pace with future HPC needs.

Further Information

1. Front Side Bus (FSB) is a term describing a processor (CPU)-to-system memory data bus. It has also known as CPU bus speed, external CPU speed, memory bus and system bus. It is the speed that the CPU communicates with RAM (memory). The front side bus on a computer connects the processor to the north side bridge, which comprises the memory bus, PCI bus and AGP bus. In general, a faster FSB bus means higher processing speeds and a faster computer:
   www.intel.com/technology/platforms/quickassist/index.htm?iid=platform_home+qa

2. HyperTransport technology is a high-speed, low latency, point-to-point link designed to increase the communication speed between integrated circuits in computers, servers, embedded systems, and networking and telecommunications equipment up to 48 times faster than some existing technologies. HyperTransport technology helps reduce the number of buses in a system, which can reduce system bottlenecks and enable today's faster microprocessors to use system memory more efficiently in high-end multiprocessor systems:
   www.hypertransport.org

Acknowledgements

- Bryce Mackin, Strategic Marketing Manager, Compute and Storage, Applications Business Group, Altera Corporation

101 Innovation Drive
San Jose, CA 95134
www.altera.com