White Paper

Extraordinary Acceleration of Workflows with Reconfigurable Application-specific Computing from SGI

November, 2004

Table of Contents

1.0 Introduction

Application specific acceleration using hardware methods is not new to computing, nor to SGI. Such technologies as digital signal processors (DSPs) or custom ASICs have been deployed in several HPC application areas. For example, SGI's Origin® 2000 and Origin® 3000 systems along with the SGI® Tensor Processing Unit (TPU) accelerate applications which require FFT style calculations. The TPU provided approximately forty times (40X) performance improvement. Attempts in the industry to deploy reconfigurable solutions using Field Programmable Gate Arrays (FPGAs) have also been made in such industries as manufacturing, government, research and media. Applications have included image recognition, decryption, gene comparison, video format translation, data compression and color intermediaries in the special effects industry. However, these efforts have met with mixed success, due to historical limitations in FPGA performance and host system integration. Also, providing an easy-to-use programming environment while achieving the order(s) of magnitude performance increases is a nontrivial problem.

Recent advances in capability, low power consumption and a volume cost profile for FPGAs combined with the breakthrough I/O capabilities of new SGI® Altix® systems, promise extraordinary performance breakthroughs for a variety of problems due to increasing levels of parallelism that can be achieved. And as scalar microprocessor-only based solutions continue on the path of increasing power consumption, more applications will benefit from this method of acceleration.

2.0 Trends in FPGA Technology Point to New Opportunity for HPC

Prior to the late 1990's FPGA technology lagged that of ASIC microprocessor technology. However, R&D by FPGA industry giants Xilinx and Altera and fabrication improvements by IBM and others have led to faster and faster FPGAs. Also, the increasing emphasis for a 'wired' world has led to the demand for faster and more capable FPGAs and given them the attractive cost structure of a volume consumer technology. Finally, with the number of gates increasing to one million or more in an FPGA, more useful work can be accomplished in an FPGA than was possible five or so years ago.

Current offerings of FPGAs allow:

- 500Mhz clock rate
- Up to 11Gbps serial IO
- Onboard scalar processor cores
- 90nm process allowing for low power consumption
- Terabyte/sec memory bandwidth, 10's of Tops/sec

3.0 Common Industry Approaches for FPGA Deployment Limit their Potential

System hardware:

Although variations on the theme do exist, the typical instantiation of an FPGA in a system is in a co-processor model with the FPGA available via an I/O bus, typically PCI(X) or VME as well. Scalar processors on a CPU mother board arrangement provide the housekeeping functions and run the operating system and user interface.

As in most co-processor models, data is loaded into the FPGA in a DMA operation, and results are loaded back to main memory. This model has various advantages. First of all, PCI and VME buses are ubiquitous through all general-purpose computing platforms and provide a solid and well-known development environment. It's also an inexpensive solution that speeds time to market and leverages existing ASIC simulation software solutions.

The disadvantages to this architecture are many-fold. Because of the slave I/O nature of the interface, latency to the FPGA is high and there's relative limited bandwidth back to main memory due to the I/O bus limitations on bandwidth. It's also difficult to integrate into a high-bandwidth network and filesystem because of the limited PCI bandwidth.

System software:

A variety of software stacks exist in the reconfigurable computing marketplace today. Some are nothing more than the very same tools that hardware designers use. This results in programming to a Hardware Design Language (HDL) that is then used to create a bitstream that is downloaded into the FPGA. HDL use is both foreign and difficult for most ISV or end-user programmers.

As an answer to those folks who did not want to program in HDL, a number of schematic-based tools exist that allow the FPGA programmer to take existing modules for specific computing functions (say an FFT module) and link together with other modules. A bitstream is created by the tool for download to the FPGA.

Finally to answer the needs of scientists, engineers, and other researchers who want to program in higher language a number of C-like (but not ANSI C) languages exist. A compiler converts the C-like constructs into HDL and a bitstream can be created using traditional approaches. The drawback of this method is that the high-level language compiler technology is still in its infancy and needs to be optimized further (as in the early days of C to assembly compilers).

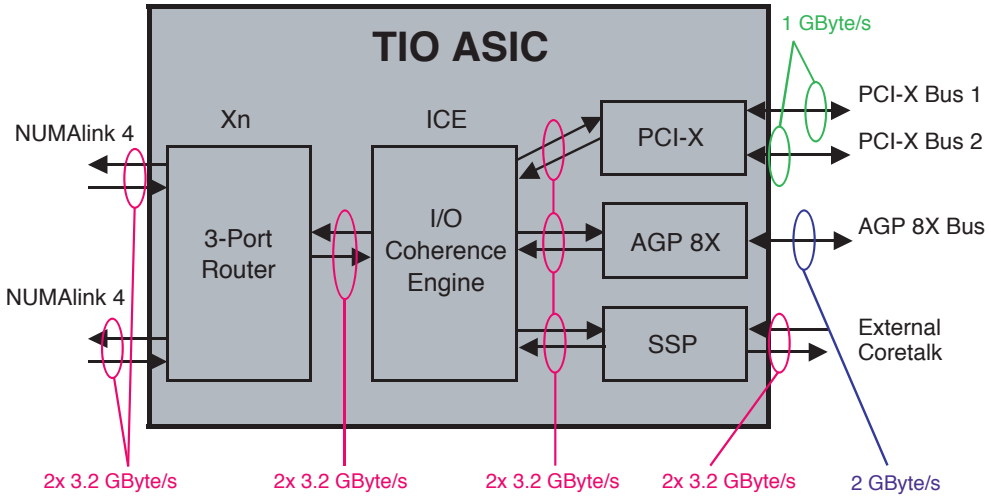

Figure 1. Diagram of SGI's Scalable Systems Port based on the TIO ASIC

There are a variety of reconfigurable logic development tools available in the marketplace today. They can answer the needs of a variety of development skills from the hardware designer to a scientist. The disadvantage of the current tools is that existing code in higher languages (C, C++, FORTRAN) will need to be re-programmed (or sections thereof) in either HDL and/or C-derivative languages. This results in substantial work as well as a risk of suboptimal results, including bitstreams that fail to run "at speed" of their performance targets.

4.0 SGI Approach – Reconfigurable Application Specific Computing or RASC

SGI believes that FPGAs and other reconfigurable chips offer significant – some times orders of magnitude – performance improvements at lower power and heat than conventional microprocessors when integrated into a system with extra-ordinary I/O performance. A high performance scalar micro-processor such as IBM® POWER™ and Intel® Itanium® 2, makes compromises to fulfill adequately the needs of a wide range of applications and to deal with increasing power consumption. For some applications, these compromises place significant, even dramatic limitations on workflow. For example, what if your application needs to do many 'add' operations? Why constrain yourself to the 12 adder units available on an Itanium 2 proces-sor? Instead you might develop your own unique core – which is the design you create on the FPGA – to have logic that allow 120 adds per cycle – a 10X improvement.

Re-configurable application specific computing (RASC) is the term that SGI has coined in reference to a family of capabilities generally addressed by FPGA, technology. Chips built around FPGAs, as opposed to ASIC technology, can be programmed by a customer for a specific task. Certain problems, when programmed into an FPGA, can see orders of magnitude speed improvement over the use of a conventional scalar microprocessor.

While this paper mainly discusses the FPGA elements in RASC, additional application specific computing categories which also benefit from the very high I/O of new Altix systems include Graphics Processing Units (GPU's), DSP's (Digital Signal Processors), math or floating-point processors, and custom ASICs.

SGI Approach to Hardware Architecture:

SGI is working with several third party vendors to make available their PCIX-based solutions and development tools on SGI's Altix servers and Silicon Graphics Prism™ graphics systems. Nallatech is one vendor that has ported their boards and development software (Fuse). For customers who only require bandwidths out to PCIX bandwidths, but also need some of the key features of the Altix platform (NUMA-based big memory, high I/O bandwidth, etc.) this solution makes an ideal develop-ment platform.

For the highest performing RASC solution, SGI is using its Scalable Systems Port (SSP) enabled via the new TIO ASIC to provide a high bandwidth, low-latency interface to FPGAs. Placing the RASC co-processor on the SSP link has a number of benefits:

- Higher bandwidth (up to 3.2GB/s) in each direction
- Lower latency
- Direct access to the memory coherency domain

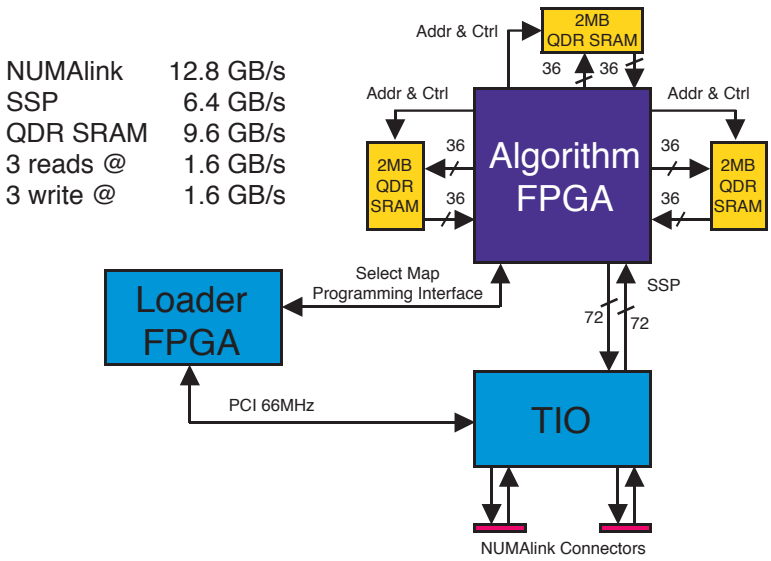

Figure 2. Diagram and specifications for SGI's "SA Brick"

The performance benefits of higher bandwidth, lower latency and coupling of the FPGA directly to system memory are perhaps obvious – dramatic performance improvement. But the simplified programming model in virtual memory brings these benefits home. With virtual memory, the user can map a virtual address to access the memory of the FPGA allowing for global pointers. Without virtual memory, the user would need to read and write data back and forth between user and kernel space (similar to a disk drive access). Another added benefit of using virtual memory is that common synchronization constructs such as barriers and semaphores can be used to synchronize the workings of multiple FPGA cards.

SGI has already deployed a number of developer bundles that involve what we are calling an SA brick with an SGI developed FPGA board connected via NUMAlink™ with an Altix® 350 system.

SGI Approach to Software Architecture:
FPGA adoption in broader-based applications has been hampered in part by a development environment more geared to hardware design engineers than end users. Scientists and engineers who might be interested in deploying them would be far more comfortable with a development environment based upon languages such as FORTRAN and C, and debugging tools such as gdb, the open source GNU debugger. SGI has designed RASC with a user-friendly development environment based on familiar, industry standard elements.

SGI's software architecture for RASC is best described in layers. The upper layer resides in user space. This layer allows an application to manage an FPGA "device" in the familiar context of a Linux device; run algorithms (which may or may not be FPGA-accelerated); call libraries (which may or may not have FPGA-accelerated routines in them); and finally debug via an FPGA-aware version of gdb.

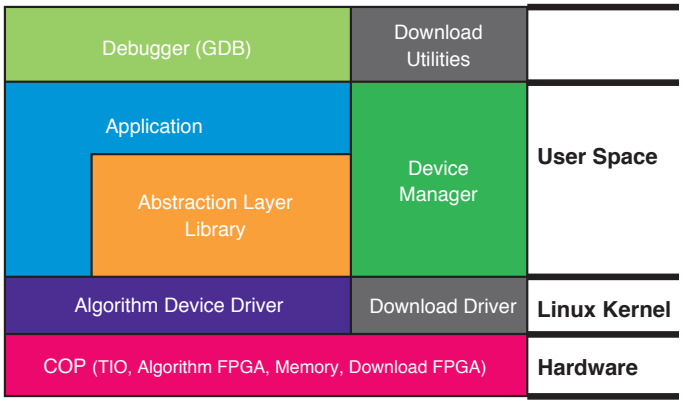

Figure 3. SGI software architecture for RASC

The next layer resides in the Linux kernel and is the device driver layer which includes device-specific code to manipulate the FPGA hardware. It includes a download driver to download a bitstream into an FPGA, and another driver to manage the synchronization of data movement to/from the FPGA as well as algorithm execution. The final layer is the reconfigurable element, commonly called the core, which contains the FPGA logic. When an FPGA 'routine' is called by a program, and that program is being debugged, the debugger typically is unable to 'step' through the program and then smoothly into the 'routine' as if it were a non-FPGA routine. SGI has done work with gdb to support a much more interactive debugging environment that allows stepping through an FPGA routine.

In the future, SGI believes that a development environment of 'libraries' optimized for FPGAs, will be the optimal development environment for scientists and engineers. The library can be included on the compile line. When the program is run, and the library included, a test is made for available FPGAs. If they exist, and are not in use by another program, then the FPGA-optimized routine is run. Otherwise a routine is run on a scalar microprocessor. SGI plans to accelerate some common routines in existing libraries such as SCSL (SGI Scientific Library), as well as accelerate some other popular libraries such as VSIP (Vector Signal Image Processing Library).

Application Areas:
SGI envisions the application use of RASC in the following areas:

1. Compute. To offload parts of computation that lend themselves easily to FPGA processing
2. I/O. An FPGA as a big DMA engine, allowing some preprocessing, and perhaps scatter/gather of data to and from host memory
3. Graphics & Digital Media
4. Application specific
5. Novel networking protocol implementations

FPGA use lends itself very well to integer data and bit data (rather than floating point, which uses up a lot of chip real estate), although as FPGA sizes continue to increase and special embedded hard micros are added, customers will have success in designing FPGAs to perform floating point operations, and even double precision problems. The bit data capabilities of FPGAs are superior to those of scalar micro-processors and lend themselves to some scientific, filtering, and cryptanalysis applications.

In addition, signals processing algorithms such as Fourier Transforms, Fast Fourier Transforms (FFT's), convolutions, digital filtering, etc. lend themselves very well to FPGA use (as well as DSP). These algorithms are computationally rich, with relatively few decision structures. The advantage of the FPGA approach is that new algorithms can be quickly tried and trade-off's tested in hardware and sometimes reprogrammed "in the field."

The convergence of integer operations performance and signal processing leads to a large benefit of FPGAs in the application areas of signals and image processing–which quickly leads into the military and homeland security space. Image recognition in the visual, radar and infrared spaces is one of the most popular application areas today of FPGA technology as well as signals and electronics intelligence 'snooping'. There are also related commercial applications such as Doppler radar.

Signal processing is also important in the areas of vibration and acoustics which have wide application in a lot of industries. The need to run many different simulations of a particular model to ensure optimal quietness in an automobile puts significant stress on a scalar microprocessor. With an FPGA, the number of simulations that can be run as well as the complexity of each one can be improved.

FPGAs have seen wide use already in format conversion in the telecommunications industry with an extension into the media broadcast industry. FPGAs might be used to create color intermediaries in digital movie production, a technique to simulate the various effects that used to be available through applying different development techniques to celluloid film. In addition, format conversion from Standard Definition (SD) to High Definition (HD) to 2K to 4K and back, and/or encryption of valuable digital assets, might be accomplished via an FPGA. The 'dual port' nature of a lot of FPGA designs which allows data to "pass through" the FPGA allows pre- and post-processing of data.

Other application areas that show promise for FPGAs:

- Some seismic processing algorithms
- Data compression and data search of databases
- Format conversion, data encryption, color intermediaries, and digital watermarks in the video and motion picture industry.
- Data encryption and decryption in the homeland security area.

5.0 Conclusion

Combining the high perfomance of new FPGAs and the extra-ordinary I/O capability of the new Altix system, SGI's Recon-figurable Application Specific Computing platform raises the bar for several classes of important HPC applications. The RASC solution brings this capability to users through a software environment that allows control of powerful yet unfamiliar FPGAs in the friendly terms of C and common Linux tools. SGI's RASC solution is being deployed today in development mode for such applications as image processing and encryption. The complete RASC solution on SGI Altix is emerging as the leading platform for reconfigurable computing, with the capability to deliver extraordinary performance breakthroughs for many HPC problems of critical importance to science, industry and national security.

Corporate Office
1500 Crittenden Lane
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

North America +1 800.800.7441
Latin America +55 11.5509.1455
Europe +44 118.925.7500
Japan +81 3.5488.1811
Asia Pacific +1 650.933.3000

3721 [11.04.2004]

J14747