FPGA SUPERCOMPUTING PLATFORMS: A SURVEY

Mariette Awad *, *IEEE member*

Electrical and Computer Department / American University of Beirut
PO Box 11-0236, Riad El Solh
Beirut 1107 2020, Lebanon
email: mariette.awad@aub.edu.lb

ABSTRACT

Field Programmable Gate Arrays (FPGAs) inherent reconfigurable nature and their low power consumption have made them so complementary to microprocessors that many are advocating their inclusion in all supercomputing clusters. Today FPGAs are included in few mainstream computer systems for accelerating application specific performance. Among the numerous areas in reconfigurable computing FPGA have been encroaching into, we focus our literature review mainly on the area of high performance computing. Moving from FPGA general features to the evolution of FPGA supercomputing architecture, its roadmap, we reference selected applications lately developed for accelerating large simulation tasks using FPGA based supercomputers before presenting concluding remarks on challenges yet to be overcome.

1. INTRODUCTION

The introduction of Programmable logic into computing environment for accelerating purposes dates back to the early 1960s when the original reconfigurable computing concept was attributed to Gerald Estrin in his description of a hybrid computer [1]. However it was only with the advent of Field Programmable Gate Arrays (FPGAs) that the technology took off in 1986 when Xilinx launched the first commercial SRAM based FPGA [CD+86]. Due to the constant progress in VLSI architecture design, FPGAs have evolved to become multi-million-gate computing platforms where the newest FPGA generations integrate logic blocks, embedded memory, fast routing matrices, and microprocessors all on one silicon die [2]. FPGAs offer enhanced performance and density while complementing computational ASIC and processor building blocks with their post fabrication configuration. Among the numerous current applications of FPGAs and to name some: digital signal processing, software-defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, target and speech recognition, cryptography, bioinformatics, computer hardware emulation, optimization and a growing range of other areas such as consumers electronics in cars and portable applications, we focus our literature review on the area of high performance computing. Supercomputer refers in this context to the state of the art computer, expected to have several microprocessors running in parallel to provide the highest processing capabilities at the time it is released on the market. However with the technological revolution observed in computing, Moore's Law and the economy of scale, any today's supercomputer is nothing but tomorrow's ordinary computer. This statement can be easily justified by a smere extrapolation of New York World newspaper release when it first introduced the term 'supercomputing' to describe in 1920 the large custom-built tabulators IBM produced for Columbia University.

In the remaining of this paper, we provide a review on FPGA architecture and performance. More specifically, section 2 highlights FPGA general features and briefly compares it to CPU and graphics processing unit (GPU) platforms. Section 3 describes the evolution of FPGA supercomputing architecture, its roadmap and the novel architecture of supercomputers. Section 4 references selected applications developed for accelerating large simulations using FPGA based supercomputers. Finally, section 5 closes with concluding remarks and perspectives on the future of research in FPGA supercomputer architecture and reconfigurable computing.

2. FPGAS: GENERAL OVERVIEW

FPGA technology field has experienced a rich history of mergers, acquisitions and market departures leaving thus only few big fables companies - among which we name Altera (SRAM, Flash) , Actel (Antifuse) , Lattice (SRAM, flash), Quicklogic (Antifuse) and Xilinx (SRAM) - with significant market share due to their design innovations and main technologies. Because of the strong competition and compelled by the desire to differentiate products designed for specific applications and functional architecture, these manufacturers offer a wide range of different device "families".

*Sponsor acknowledgment for University Research Board # 113030-988328

978-1-4244-3892-1/09/$25.00 ©2009 IEEE

To reduce the significant cost associated with the fine grained reconfigurable logic devices that empower FPGA to implement a vast variety of functions, researchers have further investigated coarse grained reconfigurable computing arrays (CGRA) which resulted in a large number of architectures. And to name some work, we mention Raw Chip from MIT which is the most coarse grained, PipeRench from Carnegie Mellon University for run-time reconfiguration of hardware virtualization, PACT XPP developed commercially by PACT Information technology GmbH for signal or media processing, and the Field Programmable Object Array introduced by MathStar as a cost effective, rapid time to market solution. More details on previous CGRA architectures can be found in [3] where Hartensteins's provided an overview of 19 different CGRA and highlighted their main limitations in effectively generalizing for any problem domain.

On the Software side for FPGAs, new tools and techniques enable engineers and companies traditionally using compilers such as C for programming microcomputers to seamlessly migrate towards FPGAs without having to spend resources for learning new languages and environments. Among the numerous algorithmic languages available, we mention Prism Compiler as the first sequential C kernels for RC , NAPA C as the pragma-driven approach for computationally expensive kernels , Garp- C as the ultimate for hardware/software co-compilation, Mitrion-C as one of the few full implicit parallel processing algorithms supporting SGI RASC and CRAY X1, Impulse-C as the standard C compiler used for both processor accelerator and module generation, Streams-C for parallel processing either on a hardware or software level, SAC optimized for image processing applications with its notion of convolution window, Carte C and Carte Fortran from SRC Computer for they allow single compilation of high level language into a unified executable and assign subsections of it to either CPU or FPGA and Handel – C from Celoxica that has their own C dialects for the C – to gate tools [4]. Also few high level graphical programming development tools such as the CoreFire from Annapolis Micro Systems, the Viva from Starbridge Systems, the System Generator from Xilinx, the Reconfigurable Computing Toolbox from DSPlogic, are made available as application development tools and frameworks for users and researchers in the FPGA area.

Compared to CPUs, modern FPGAs exhibit very low power consumption (for instance Xilinx Virtex-5 claims around 3 Watts consumption whereas Intel Core Extreme requires 60-70 Watts). Because of its architecture, a microprocessor deals with an application as a linear flow of instructions whereas an FPGA fragments it into a set of independent and optimized logic blocks capable of meeting severe time-bound limits. FPGA would hold the lead in the technological advance over microprocessors for the coming 15 years at least since FPGA follow the International Technology for Semiconductors roadmap, rather than the microprocessors roadmap. Basically, FPGAs is growing at the rate of their enabling technology now in cutting edge CMOS, which shrinks its feature size by a factor of 1.26 per year according to Noyce thesis observed in the early sixties. A detailed study comparing FPGA to CPU and the viability of FPGA supercomputing can be found in [5].

With the evolution observed in changing GPUs from fixed-function pipelines to flexible general-purpose computing engines, GPUs now offer double-precision support and easy-to-use programming platforms. Nvidia has been bringing its graphics technology such as Tesla S1070 1U GPU supercomputing system into HPC spaces where systems traditionally have been powered by CPUs and claims x250 speedup over standard PC when powered with 960 parallel processing cores. By creating many small independent threads, exploiting data locally and conserving memory bandwidth, GPU programming is becoming attractive for scientific computing such as advanced MRI reconstructions [6]. GPUs are also proving to be competitive and complementary to FPGA supercomputing platforms as demonstrated for instance in [7] where a Cray XD-1 and NVIDIA GeForce 7900 GTX GPU are used for matched filter computation or in [8] where FPGA and GPUs are utilized for accelerating the Quantum Monte Carlo chemistry application used to study quantum many-body systems. Because of the results reported in [9], after optimally combining GPUs, FPGAs, and standard CPU in an attempt to produce a system rivaling the computational power of HPC, future computing architectures are expected to be hybrid systems with parallel-core GPUs working in tandem with multi-core CPUs and FPGAs to offer an increased hardware performance and programmability for supercomputing platforms.

3. EVOLUTION OF FPGA BASED SUPERCOMPUTERS ARCHITECTURE

Historically and up until the mid 1980s, supercomputers had different operating systems and even incompatible vectorizing and parallelizing Fortran compilers. In the early 1990, Algotronix CHS2x4 was introduced as the world's first commercial reconfigurable computer that was designed as an array of CAL1024 processors in ISA format coupled with 8 FPGAs each having 1024 programmable cells to combine routing and computing functions into a single primitive. Other research work such as Triscend's E5, National Semiconductor's NAPA and Berkely's GARP, Pleiades, Teremac, the FPGA High Performance Computing Alliance's (FHPCA) Maxwell etc… followed using heterogeneous architectures.

As of 2006, most of the supercomputers produced by the Top500 list have the same top- level architecture and use variants of Linux or UNIX as operating systems. SRC computers, DRC Computer Corporation Cray, SGI, XtremeData etc… now offer few high performance clusters featuring reconfigurable computing (RC) capabilities. Parallel processing of RC operations can be either at instruction or task levels. However despite the different technologies such as vector processing, liquid cooling, non-uniform memory access (NUMA), striped disks, parallel file systems etc… developed to overcome concerns in reliability, latency and performance, many challenges still face supercomputers. The improved performance observed is often attributed to the memory hierarchy which insures the processor is always fed with instructions/data, to the I/O systems that support high bandwidth and low latency and to the FPGAs modules that are used as acceleration nodes in the high performance clusters. Novel architectures vary radically with respect to processors' interconnect and scalability connections for parallel systems. With yet a unifying taxonomy to be suggested for classification purposes, architectures are mainly compared based on granularity, rate of reconfiguration, host coupling, routing, scalability and tool flow.

In early systems, architectures were described based on the interconnect type. Connected coprocessors consisting of a central microprocessor connected to several FPGA-based boards, I/O Bus accelerators or loosely couple processors were the classic co-processor models for supercomputer architecture. The earliest reconfigurable computers that emerged in the early 90's such as Splash 1, Splash 2, PerLe used the I/O Bus accelerator architecture where I/O boards contained FPGAs, inter-FPGA interconnect, on board SRAM/DRAM and high speed serial interfaces to external devices as well as interface to the host computer I/O bus. Currently SRC MAP processors are connected via memory bus, XtremeData and DRC plug directly into the CPU sockets, whereas SGI RC100 hangs on SGI's NUMA link fabric. The implementation cost of this architecture is relatively very low but so is its performance in parallel applications. Data in that scenario is acquired and processed by FPGA for bandwidth reduction and the result is presented to the microcomputer for evaluation. As long as the serial bus is not overloaded – which is not often the case-transferring the data to and from FPGA will be computationally more effective than implementing the solution on the CPU. Modern implementations such as the one produced by Annapolis Micro systems, DRS Technologies, Nallatech Ltd, Virtual Computer Corporation and Xess Corporation, offer commercial FPGA I/O boards with only one or two FPGAs on board to overcome the interconnect problem.

The tightly connected coprocessor architecture extended the loosely coupled coprocessor model to allow direct communication between FPGAs using some fast interconnect such as Inifiniband or point to point networks. This accounts for the limitations of the serial bus encountered in loosely- connected coprocessors and offers more parallel processing capabilities by having the FPGA and CPU networks available for simultaneous usage. From a topology perspective, wiring and chip floor planning are reasonably more complicated than the simple I/O accelerator model which makes this architecture more costly but still relatively acceptable compared to the high performance gain it achieves. Maxwell which was built in 2007 in Scotland by the FHPCA used this topology of plug in coprocessors with direct dedicated connections where the 64 FPGAs didn't involve interaction with the host processor (2.8 Xeon processor) during algorithm execution.

In some systems, network fabric connection connects FPGAs directly to the host network via NIC. In this manner, FPGAs communications are not contingent on a specific CPU. Instead, a dedicated high speed point to point network with possibility of communication over the host network is made available. However the downside of this type of connection is the need for some communication protocol to implement contention regulation, timing scenes and packet allocation. By combining AMD Opteron 64 bit CPU with Xilinx Virtex-4 LX160 FPGAs, Cray XD1 introduced an entry level supercomputer in 2005 using RapidArray as an embedded switching fabric interconnect and reported four times less MPI latency than Infiniband and 30 times faster than Gigabit Ethernet. [9] also used Cray XD1 with the Mitrion platform where the entire cluster is on one FPGA to further validate performance improvement for the gravity simulator 'gravit'. Mitrion SDK 2.0 has been optimized for applications requiring high throughput and low latency, such as genome informatics, proteomics, Internet and database search, and business process optimization challenges. Accelerated applications running on the virtual processor increase application performance by 10x to100x versus non-accelerated systems while consuming ninety percent less power [10].

Nowadays, to better describe scalability and parallel measures of FPGA systems, the interconnection mechanism is grouped based on the types of nodes being connected together. They can be either uniform nodes non uniform systems (UNNS) or non uniform nodes uniform systems (NNUS) as suggested by [11]. UNNS would have either FPGA or microprocessor for nodes which will be linked via an interconnection network to interact with the globally shared memory. SRC-6/SRC-7, Altix/RASC and the MPAS Series C, are illustrative examples of UNNS architecture. They support a variable ratio of FPGA to microprocessor nodes set by the vendor to meet application demands. Connection to the Hi-bar switch communication layer is achieved through SNAP interface. The major disadvantage of this architecture is the increased latency between the nodes, code portability and competition for the overall bandwidth between FPGA nodes and the microprocessor nodes. NNUS architectures support only one type of node which can contain both FPGAs and microprocessor, where the former are directly connected to the later inside the node. Cray XD1 and reconfigurable clusters are typical examples of NNUS. Differently from the UNNS, the latency between the microprocessor and its FPGA coprocessor is low and performance for the data-intensive applications is relatively high.

4. APPLICATIONS USING FPGA BASED SUPER-COMPUTERS

This section aims to inform readers on selected research that provide a clear measure of supercomputers maximum practical performance. It doesn't seek to cover every recent research performed on FPGA based supercomputers. Readers interested in older techniques and surveys need to consult for instance [12], [13], [14] because they provide an in depth coverage of other approaches.

4.1. FPGA in Cryptography Applications

Because internet traffic security is a fundamental concern for researchers and because cryptographic algorithms require the flexibility to adapt to the evolving requirements of Internet Protocol Security (IPSec) for high speed networks, the fine granularity of FPGAs has made them attractive for Advanced Encryption Standards (AES) at instruction, data and task levels.

For the Data Encryption Standard (DES), [15] compared performance and reported a speedup of 6757, 12162 and 28514 when comparing performance of the SRC-6, Cray XD1 and SGI Altix4700 to a 100 processor Beowulf cluster respectively. The Cray XD1 improved results compared to the SRC-6 stem from the fact that its FPGAs run at 200 MHz twice as fast.

T. Guneysu et al in [16] proposed a novel and massively parallel cluster system (COPACOBANA) based on low-cost FPGAs to perform cryptanalytical operations. Despite the fact that parallel applications in the field of cryptography are complex for FPGAs and also require the availability of at least moderate communication and memory facilities for the arithmetic intensive applications, COPACOBANA was capable to efficiently host high-performance digital signature generation according to the elliptic curve digital signature algorithm and integer factorization based on the elliptic curve method .

4.2. FPGA in DSP

Digital signal processing (DSP) applications are also well suited to FPGA supercomputer architecture because FPGA would perform all the resources intensive computing operations of signal processing.

Cameron in [17] employed deeply pipelined processors with FPGA to effectively implement a ray tracing strategy and improved the rate of ray tracing and calculations for succeeding steps. He assigned 839 AMD Opteron processors and reached a 97.9% parallel efficiency which translates into a 77x speedup over the Moderate Resolution Imaging Spectro-radiometer (MODIS) technique used in the two low orbit earth observing satellites in [18].

El-Araby et al, in [19] used SRC-6E for on- board preprocessing of hyper-spectral imagery. Their work exploited both the fine and coarse-grain parallelism provided by RCs to reach optimal speedup gains. El Araby reported in [20] an order of magnitude speedup over traditional processing techniques.

4.3. FPGA in Dynamics and Bioinformatics

Nanoscale molecular dynamics (NAMD) [21] is a parallel molecular dynamics code extensively used by the computational biophysics community for high-performance simulation of large biomolecular systems. NAMD's computational kernel is highly optimized to run on conventional von Neumann processors. Kindratenko and Pointer in [22] presented the first case study for porting NAMD written in C++ to FPGA language. They used SRC-6 MAP for their target platform and achieved 2 seconds of computation time for one simulation step involving 92224 atoms in just over 2 seconds, which is a 3 times performance increase of NAMD optimized code.

Advances in the field of bio-technology highly encouraged by the Human Genome Initiative and the overwhelming data generated by genetic sequences has led to an increasing demand for searching and matching DNAs in a timely manner. The Smith-Waterman algorithm which is designed to compute the distance between two DNA or protein sequences using dynamic programming approaches has been implemented on different FPGA platforms as reported by [23] and a speedup performance was demonstrated.

5. CONCLUSION

The inherent parallelism of FPGA logic resources enable offloading time intensive operations from software to FPGA and considerable compute throughput to be performed on the FPGA side instead of the microprocessor despite the design complexity when compared to conventional software development and the computational requirements of current design tools that requires few hours for updating minor changes in the source code. With time, as the research and development communities address architecture and CAD tools limitations, FPGA will shape even more strongly most digital logic designs and implementations. As of now, there are few limiting factors for a wide adoption of FPGAs specifically in high performance computing aside than cost. With an improved balanced data access and compute scheme, and an efficient floating point calculation strategy, FPGA would impact even more the HPC arena and deliver order of magnitude performance gains. More reduction in memory usage and time to reconfigure would be achieved if optimization and partial reconfiguration are investigated in more depth. Also FPGA architecture would need to address the bottleneck of the routing interconnect that correlates positively with the FPGA high capacity devices according to Rent's rule. Hierarchical interconnect and multi- FPGA systems now part of the FPGA devices themselves together with microprocessors, DSP and SRAM arrays would provide optimized solutions under the challenges to overcome of the Field Programmable Systems – on – a Chip for which no company yet had emerged as the main solution provider. And given the future silicon technologies trends and increased leakage observed in advanced nodes, energy dissipation and heat extraction will still be a major concern - unless dynamic power management techniques are implemented- because of the high operating frequency and transistor density despite the evident reduction in power on and operating voltage. Finally, bandwidth, latency and energy consumption for communication need to be also optimized based on some effective Network on Chip communication protocol tailored for FPGAs.

6. REFERENCES

[1] I. Grout, Digital Systems Design with FPGAs and CPLDs, Elsevier Science and Technology Books, ISBN-13:9780750683975, 2008.

[2] G. Estrin," Organization of Computer Systems – The Fixed plus Variable Structure Computer", Proc Western Joint Computer Conference, New York, 1960, pp.33-40.

[3] R. Hartenstein,"A decade of Reconfigurable Computing: a Visionary Retrospective", DATE '01: Proceedings of the Conference on Design, Automation and Test in Europe, IEEE Press, 2001, pp.642-649.

[4] B. Holland, et al, " Survey of C- based application mapping tools for reconfigurable computing", in Proceedings of the 8th International Conference on Military and Aerospace Programmable logic Devices (MAPLD), 2005.

[5] S. Craven and P. Athanas, " Examining the Viability of FPGA Supercomputing", EURASIP Journal on Embedded Systems, 2007, Article ID 93652. .

[6] S. Stone, et al., "Accelerating Advanced MRI Reconstructions on GPUs", Proceedings of the 5th International Conference on Computing Frontiers, May 5-7, 2008.

[7] Z.K. Baker, et al, "Matched Filter Computation on FPGA, Cell and GPU", Field-Programmable Custom Computing Machines, 23-25 April 2007, pp. 207-218.

[8] A. Gothandaraman , "Accelerating Quantum Monte Carlo Applications using Emerging Architectures", Supercomputing Conference 2008.

[9] E. Kelmelis et al, "Accelerated modeling and simulation with a desktop supercomputer", Proc. SPIE, Vol. 6227, 62270N (2006); DOI:10.1117/12.668281.

[10] http://www.mitrionics.com

[11] A. Dellson, G. Sandberg and S. Möhl, "Turning FPGAs into Supercomputers Debunking the Myths About FPGA based Software Acceleration", Proceedings of the 48th Cray User Group Conference, Lugano, Switzerland, May 2006.

[12] T. El-Ghazawi, et al., " The Promise of High- Performance Reconfigurable Computing", IEEE Computer Society, Feb. 2009, pp. 69-76.

[13] K.Compton and S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software", ACM Computing Survey, vol, 34, no.2, 2002, pp. 171-210.

[14] S. Margerm, "Reconfigurable Computing in Real- World Applications", FPGA and Structured ASIC Journal, 10(5), February 2006.

[15] S. Hauck and A. DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing, Morgan Kaufman, 2008.

[16] T. Guneysu, et al, "Enhancing COPACOBANA for advanced applications in cryptography and cryptanalysis", Field Programmable Logic and Applications, 2008, pp. 675 – 678.

[17] C. Cameron, "Using FPGA to supplement Ray- Tracing Computations on the Cray XD-1", IEEE Comput. HPCMP-UGC DoD, 2007, pp. 359-363.

[18] E. Waluschka et al.," Modeling Studies for the MODIS Solar Diffuser Attenuation Screen and Comparison with on-orbit Measurements ", Proceedings of SPIE, vol. 5542, no.47, 2004, pp.342-353.

[19] E. El-Araby, et al, "Wavelet Spectral Dimension Reduction of Hyperspectral Imagery on a Reconfigurable Computer," Proc. IEEE 2004 Conference on Field Programmable Technology, Brisbane, Australia, Dec. 6-8, 2004, pp. 399-402.

[20] E. El-Araby, et al, "System-Level Parallelism and Throughput Optimization in Designing Reconfigurable Computing Applications", Reconfigurable Architectures Workshop, RAW 2004, Proc. International Parallel and Distributed Processing Symposium (IPDPS) Workshops 2004, Santa Fe, New Mexico, USA, April, 2004.

[21] J.C. Phillips, et al, "Scalable molecular dynamics with NAMD", Journal of Computational Chemistry, 26, 2005, pp. 1781-1802.

[22] V. Kindratenko and D. Pointer, "A Case Study in Porting a Production Scientific Supercomputing Application to a Reconfigurable Computer", 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'06).

[23] S. Margerm ," Reconfigurable Computing in Real – World Applications ", FPGA and Structured ASIC Journal, 10 (5), February 2006.