Eric STAHLBERG [1], Bogdan OLECH [2]
[1] WITTENBERG UNIVERSITY
[2] WEST POMERANIAN UNIVERSITY OF TECHNOLOGY, SZCZECIN, FACULTY OF COMPUTER SCIENCE & IT

Standards for global interoperability in accelerated computing: hardware, software and organizational

Ph.D. Eric STAHLBERG

He received Ph.D. degree in computational chemistry from the Ohio State University and worked for: Argonne National Laboratory, Cray Research Inc., Oxford Molecular Group, Vital Images, Inc., Chemical Abstracts Service and the Ohio Supercomputer Center. Now, he is Computational Science Director at Wittenberg University. Being main creator and also President of OpenFPGA he is integrating international effort to advance the use of reconfigurable computing in high-performance computing applications.

e-mail: estahlberg@gmail.com

Ph.D. Bogdan OLECH

He received the M.Sc. degree from Szczecin University of Technology, Poland, in 1986 and Ph.D. degree from Warsaw University of Technology in 1999. He is assistant professor at West Pomeranian University of Technology, Computer Science & IT, also cooperates with high-tech industry. His main research interests are: engineering methodology, high performance reconfigurable computing, precision measurement systems, RF applications and mechatronics.

e-mail: bolech@wi.zut.edu.pl

Abstract

Computing accelerators are increasingly being called upon to provide improved application performance, as microprocessor clock rates have leveled off. Ranging from low-level devices (FPGAs or FPOAs) to more specialized accelerators, like ClearSpeed, they can be found in desktops or in large supercomputers. Presented paper outlines efforts of OpenFPGA to foster interoperability and open interchange of information among organizations. To do that, several activities and emerging services are introduced to enable an increasingly technically and economically sustainable mainstream of reconfigurable computing ecosystem.

Keywords: HPC, FPGA, accelerating computing.

Standardy przenośności dla podsystemów wspomagania obliczeń: sprzęt, oprogramowanie, organizacja

Streszczenie

Ponieważ, częstotliwości pracy mikroprocesorów osiągnęły pewien graniczny poziom, aby sprostać wymaganiom związanym z mocą obliczeniową dla określonych aplikacji, coraz częściej sięga się do rozwiązań dodatkowo zwiększających moc obliczeniową komputerów. Zakres możliwości jest bardzo szeroki i rozciąga się, począwszy od operujących na poziomie bramki lub obiektu reprogramowalnych tablic logicznych (FPGA lub FPOA) aż po bardziej specjalizowane rozwiązania takie, jak technologia *ClearSpeed*. Znamiennym dla tych rozwiązań jest wykorzystywanie równoległości prowadzenia obliczeń zarówno na poziomie algorytmu jak i danych, a także strukturalne prowadzenie działań z użyciem zwielokrotnionych zasobów sprzętowych z cechą elastyczności tych struktur, wynikającą z możliwości ich rekonfiguracji. Skuteczność stosowania takich, dodatkowych podsystemów wspomagania obliczeń udokumentowana już została licznymi przykładami, zarówno w odniesieniu do komputerów typu PC, jak i w odniesieniu do superkomputerów. Prezentowana publikacja charakteryzuje w zarysie wysiłek podjęty w ramach organizacji OpenFPGA, dotyczący zamysłu spełnienia warunku pełnej przenośności standardów użytkowania podsystemów przyspieszających obliczenia, a także zapewnienia wymienności informacji pomiędzy organizacjami kreującymi tego typu nowe technologie. Szereg działań oraz nowych usług, zaoferowanych na platformie organizacji, ma na celu poszerzenie ponadnarodowej współpracy pozwalającej sprostać wymaganiom techniczno-ekonomicznego rozwoju utrwalającego się nurtu komputerów rekonfigurowalnych wraz z towarzyszącym jemu, swoistym ekosystemem.

Słowa kluczowe: obliczenia dużej mocy, rekonfigurowalne tablice logiczne, wspomaganie obliczeń.

1. Introduction

Accelerated and reconfigurable computing returns now as seriously considered idea after decade of doubts. The reason is that today we have other needs and other technology available than before. Needs are driven by the progress of the civilization. Science and engineering face challenges being before out of dreaming. The technology offer makes economically reasonable to explore ideas, sometimes even not new, but given up before because of their implementation inability. There are still implementation limitations; this will never change, however, on the other level of explored efficiency.

Appearance of the microprocessor changed the way of computer design. Old, good, but expensive designs slowly faded away. Great effort made on establishing good computing practice was almost forgotten. The market started to be divided into professional and personal area. Personal computing made revolution in computing, but with typical for revolutions ignorance in respect to the continuation of previous generations' effort. We could observe racing, some way technological, but rather concentrated on popularization and simplification. The last 20 years in computing it is Personal Computer (PC) domination [1], with its weaknesses in reliability of software and hardware. What was more surprising, the more advanced propositions failed the PC racing, like these from Apple Inc. for example.

PC era was the reason of traditional computer market gradual collapsing. It appeared that, in spite of rather not promising at the beginning, PC performance was growing. People started to prefer cheap, well known, easy to use, familiar boxes than work stations working with traditional UNIX required a lot of typing. Mile-stone companies, like Digital Equipment Corp. disappeared, others, like Silicon Graphics Inc. had to dramatically fight for survival. But, the popularity of the PC opened the rather hermetic box of Computer Science and made it popular. Having PCs, people started to learn early, a lot of traditional applications of engineering and business has been ported to PC systems. Appearance of Information Technology in context of Internet was the next, secondary to PC, revolution. Eventually, it was good turn, regardless of the losses of the previous empire of supercomputing and work stations, economy and civilization made significant improvements.

These few companies building general purpose microprocessors where concentrated on the effort of encapsulating the computer of previous epoch into one silicon chip. It succeeded as far as the integration had allowed. Some advances like CISC/RISC dilemma where present in between [2]. But for many years microprocessor clock speed was the main parameter compared to estimate (guess) a computing efficiency. But, for the time being physical layer limitations limit also the clock speed. And again, there is something to borrow from traditional computing, like explicitly parallel computing. Multiple core microprocessors are already common, however, suffering from their possible utilization - difficult programming.

Eventually, it is possible to say that every application type requires different programming method to be executed as fast as possible. Any algorithmic or data parallelism, with any granularity should be exploited up to the limit. But, multiprocessing limits algorithmic granularity differentiations to be executed in parallel, similarly, processor architecture limits algorithmic and data granularity. Not every algorithmic graph, on any level can be accessed for parallelization; data word may not have optimal size to make room for other data words to be executed in parallel. The answer would be a computer where its detailed hardware structure can match exact, from parallelism point of view optimized programming, optimally utilizing resources [3]. Even if it would be possible to build it would be practically impossible to program this computer, no one wants to make so customized programming [4, 5]. Another solution would be to build compilers able to recognize any parallelism and automatically configure the hardware, also very difficult. To make hardware to dynamically recognize the nature of program under-run may be also considered (lesson learned from late super-scalar to EPIC migration [6]).

As usual, some compromise is needed. The computer architecture and organization can not change to quite new paradigm, even if it would be technologically possible on so great scale. Programming methods also can not change into quite new direction. The Hybrid Computing may be possible answer. Multiprocessor/multi-core system with a bus, optimizing computing on process level, completed with accelerating components distributed some way in the system. The nature of accelerating components may differ, from logical level as for FPGA to specialized (multi)/processors, with FPOA in between. FPGAs offer most “soft” nature of hardware allowing configuration of a structure at bit level. Configured computing structure may completely follow any concurrency recognized in an algorithm on the lowest level. But again, programming is the issue [7, 8].

In this paper the authors wanted to outline a collaborative effort which is being made by the international community to make the reconfigurable computing standardized to make it directly adoptable for every demanding application. OpenFPGA is the organization where this ambitious idea is turning to the reality.

After general introduction where the background of reconfigurable computing is outlined, in section 2, observations of how the reconfigurable computing is growing are presented. Section 3 it is the presentation of OpeFPGA organization. Working activities, mission and a project example are presented there. The paper is then shortly concluded in section 4.

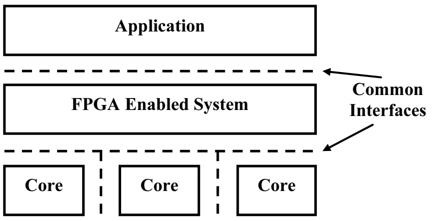

Fig. 1. Reconfigurable system with all components and interfaces are fully standardized
Rys. 1. System rekonfigurowalny ze wszystkimi komponentami oraz interfejsami w pełni standardowymi

2. Reconfigurable computing growth

Let us compare reconfigurable computing from perspective of near past and near future. For the year 2004 the fragmentation was characteristic, there where many activities made, but in isolation, without cross-communication. From the common challenges and needs perspective, tools where expensive and development time was very long. There was no portability across vendors and across product versions. It was challenging and risky to try to enter the market. No common practices were in use. Current situation was similar to the massage-passing and SMP computers, solutions available where vendor specific, without common functionality. General application market was out of reach. However, some technological priorities influenced the future architectures; hardware interfaces design and standards also inter-component or inter-system design and standards started to appear.

Four years later (2008) some improvements where observed, but the effort was still, relatively isolated; Intel AAL, OpenFPGA and CHREC made significant progress. Tools where still expansive and development time long, interoperability and hybrid integration remained emerged, still limited portability between vendors and across product versions was faced, absence of critical commercial applications and challenging marked to enter observed. However, there were characteristic changes in the practices: industry validated FPGAs and other accelerating technologies, vendor specific functionality where directed to common aims. The technology already poised to enter general application market.

Year 2010 horizon promises great changes in the software environments. Affordable and common programmable API, for preservation of investment, should be available. Implementations will be transferable over time and platforms with readily implemented algorithms. New APIs should be extensible, adaptable to new innovations and challenges. Software will be abstracted from technological domain. Core implementations will be portable, reliable and defined in the standard manner. What is more, the new culture and climate will be present; with reconfigurable technology appreciation, industry standards characterized with industry needs and technology deployed in business critical operations. Figure 1 illustrates this situation.

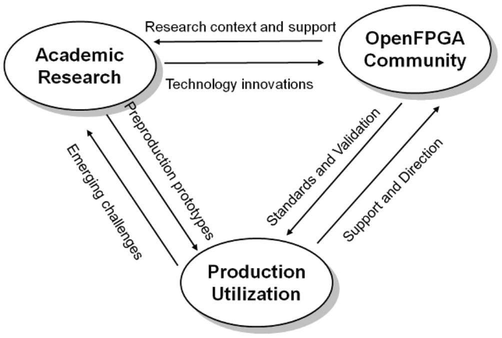

Fig. 2. Bringing research to practice treated as general rule of activity
Rys. 2. Przenoszenie wyników działalności naukowej do praktyki, jako podstawowa reguła działania

3. Organizational overview

OpenFPGA is a non-profit consortium of organizations aiming to support and accelerate the incorporation of reconfigurable computing technology activities. High-performance and enterprise applications are considered as the main area of interest. OpenPFGA is serving the worldwide community through the activities of working groups, technology discussions, mailing lists and collaborative development of best practices and industry standards in reconfigurable computing. The initiative started in 2004, spearheaded with leadership by an international cross-sector steering group and support of the University of Dayton. Presently, through the mailing list, working groups and other avenues, OpenFPGA includes participation from over 200 organizations spanning 40 countries worldwide. Figure 2 shows the key aspect of tight cooperation leading to bring research results to the practice.

The policy of the organization is to create a future of computing, where:

- Advances in reconfigurable computing technology will be easily and eagerly adopted, incorporated across critical high-performance computing and enterprise applications.
- OpenFPGA will be a catalyst for innovation and a vehicle for establishing standards for portability, performance and interoperability in the heterogeneous supercomputing community.

- OpenFPGA will advance contexts for reconfigurable computing research and channels for moving innovations into production use.

And the mission of the organization:

"The mission of OpenFPGA is to promote the use of Field Programmable Gate Arrays in high level and enterprise applications by collaboratively defining, developing and sharing critical information, technologies and best practices."

OpenFPGA working groups and their responsibilities:

T-HLLANG: Technical working group on high-level language specifications, created to define elements needed to transform algorithm specifications to implementation for FPGA systems.

T-APPLIB: Technical working group on application specific FPGA libraries, created to define elements of or needed for using portable software callable application specific APIs to incorporate FPGA functionality in traditional software applications.

T-GENAPI: Technical working group on general FPGA functionality APIs, created to define elements of or needed for an application independent common interface for basic FPGA functionalities (application independent, similar to OpenMP and MPI which are not application specific).

T-CORELIB: Technical working group on core libraries and interoperability, to define elements of or needed for increasing reusability and interoperability of FPGA functional core definitions incorporated into applications.

T-BENCH: Technical working group on FPGA benchmarks. The purpose of this working group is to develop an OpenFPGA Benchmark Suite. This suite would provide a standard set of interesting problems to compare not only performance, but also implementation ease, time to solution, compile time (place and route, etc).

A project example: OpenAccelerator Portable Accelerated Medical Application Libraries Project

Scope: The project focuses on developing an industry standard set of portable accelerated software libraries for use in medical and health research applications. The software libraries will provide a portable set of standard software APIs that have been designed to support application acceleration using a variety of application accelerators including FPGAs, Cell processors, multi-core CPUs of many types and GPGPUs. APIs will be openly published along with necessary validation datasets used to confirm correct operation. Non-accelerated reference implementations will also be provided. There will be several releases of the libraries developed through an iterative, spiral development process. Each cycle will follow the approximate stages described below:

Phase 0 – Review medical and health information application workflows with focus on usage coverage, performance bottlenecks and end-user requirements. The outcome of Phase 0 is the subset of those application scenarios demonstrating the most potential benefit if accelerator technologies are integrated into the workflows. For these workflows datasets for benchmarks are collected and evaluated.

Phase 1 – Review existing medical application libraries in use within medical and health information applications of Phase 0. This includes a review of existing image processing libraries, libraries for encryption/decryption and data compression. Examples of such current libraries include VSIPL.

Phase 2 – Functional capabilities of each application library component are cataloged and classified.

Phase 3 – Available functionalities will be contrasted to current industry data processing bottlenecks in medical application performance. Functionalities will be placed in one of four quadrants: a) high utilization and causing little bottleneck; b) high utilization and causing significant bottleneck; c) low utilization and causing little bottleneck and d) low utilization and causing significant bottleneck

Phase 4 – Hybrid accelerator friendly application programmer interfaces (APIs) will be defined for functions having high utilization yet causing significant bottlenecks. Depending on used methods and data types the API may cover all or a subset of accelerator devices.

Phase 5 – Validation and performance measuring datasets will be identified for each API

Phase 6 – A first release of accelerated medical application libraries will be proposed with working and openly available reference implementations.

Project details:

Implementations of accelerated libraries will be accomplished globally by partnering research, academic and commercial organizations. Existing industry standard libraries, datasets and benchmarks will be incorporated where each remains consistent with the objectives of the project. OpenFPGA partners, members and project sponsors using processes established to minimize risk to end use applications, will facilitate certification of correctness and performance.

4. Conclusion

One of the most important issues limiting the growth of reconfigurable computing is the diversity of accelerating hardware offerings, their programming architecture and interfaces to application level. Also programming methods are not unified; there are in use hardware description languages and C-like languages to implement reconfigurable hardware. Standardization is needed to allow directed growth in the area of reconfigurable computing.

5. References

[1] P.E. Ceruzzi: A view from 20 year as a historian of computing. Annals of the History of Computing, IEEE, Volume 23, Issue 4, pp. 49 – 55, Oct-Dec 2001.

[2] T. Jamil: RISC versus CISC. Potentials, IEEE, Volume 14, Issue 3, pp. 13-16, Aug-Sep 1995.

[3] M.C. Smith, S.L. Drager, L. Pochet, G.D. Peterson: High performance reconfigurable computing systems. Circuits and Systems, 2001, MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on, Volume 1, pp. 462 – 465, 2001.

[4] Tarek El-Ghazawi at. al.: The Promise of High-Performance Computing. IEEE Computer, February 2008.

[5] E. El-Araby, P. Nosum, T. El-Ghazawi: Productivity of High-Level Languages on Reconfigurable Computers: An HPC Perspective. Field-Programmable Technology, 2007. ICFPT 2007. International Conference on, pp. 257 – 260, 12-14 Dec. 2007.

[6] M.S. Schlansker B.R. Rau: EPIC: Explicitly Parallel Instruction Computing. Computer, Volume 33, Issue 2, pp. 37 – 45, Feb. 2000.

[7] J.J. Koo, D. Fernandez, A. Haddad, W.J. Gross: Evaluation of a High-Level-Language Methodology for High-Performance Reconfigurable Computers. Application -specific Systems, Architectures and Processors, 2007. ASAP. IEEE International Conf. on, pp. 30 – 35, 9-11 Ju-ly 2007.

[8] E. El-Araby, I. Gonzalez, T El-Ghazawi: Virtualizing and sharing reconfigurable resources in High-Performance Reconfigurable Computting Systems. High-Performance Reconfigurable Computing Technology and Applications, 2008. HPRCTA 2008. Second International Workshop on, pp. 1 – 8, 16-16 Nov. 2008.

Artykuł recenzowany