Computing and Informatics, Vol. 29, 2010, 989–1000

ANALYSIS OF THE BASIC IMPLEMENTATION ASPECTS OF HARDWARE-ACCELERATED DENSITY FUNCTIONAL THEORY CALCULATIONS

Maciej Wielgosz

AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Krakow
&
ACK Cyfronet AGH
ul. Nawojki 11, 30-950 Cracow, Poland
e-mail: wielgosz@agh.edu.pl

Grzegorz Mazur, Marcin Makowski

Faculty of Chemistry, Jagiellonian University
ul. R. Ingardena 3, 30-060 Cracow, Poland
e-mail: {mazur, makowskm}@chemia.uj.edu.pl

Ernest Jamro, Paweł Russek, Kazimierz Wiatr

AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Krakow
&
ACK Cyfronet AGH
ul. Nawojki 11, 30-950 Cracow, Poland
e-mail: {russek, wiatr, jamro}@agh.edu.pl

Manuscript received 11 February 2010
Communicated by Jacek Kitowski

Abstract. This paper presents a Field Programmable Gate Array (FPGA) implementation of a calculation module for exponential part of Gaussian Type Orbital

(GTO). The module is composed of several specially crafted floating-point modules which are fully pipelined and optimized for high performance. The hardware implementation revealed significant speed-up for the finite sum of the exponential products calculation ranging from 2.5x to 20x in comparison to a general-purpose Central Processing Unit (CPU) version. Calculating values of GTOs is one of computationally critical parts of the Kohn-Sham algorithm. The approach proposed in the paper aims to increase the performance of a part of the quantum chemistry computational system by employing FPGA-based accelerator. Several issues are addressed, such as identification of code fragments which benefit most from hardware acceleration, porting a part of the Kohn-Sham algorithm to FPGA, data precision adjustment and data transfer overhead. The authors' intention was also to make hardware implementation of calculating the orbital function universal and easily attachable to different quantum-chemistry software packages.

Keywords: High performance reconfigurable computing, FPGA, quantum chemistry, floating-point operations

1 INTRODUCTION

High-performance computing (HPC) has been recognized for many years as an area dominated only by general-purpose microprocessors. The rapid increase of the logic resources of modern Field Programmable Gate Arrays (FPGAs) creates an opportunity to hardware-accelerate calculations which involve processing large volumes of data. Such a vital rise in the computational capabilities of FPGAs has stimulated studies on reconfigurable hardware accelerators [1].

The proposed hardware module aims to speed up quantum-chemical calculations and provide compatibility with the standard floating point data format. The RASC [2] platform has been chosen as the hardware accelerator because of the computational capabilities it offers for SMP supercomputers. Since achievable computational speed-up strictly depends on the selected part of the algorithm, the choice of the proper section of the code becomes the crucial issue. In our studies, performed using the niedoida computational chemistry package [3], the finite sum of the exponential functions was found a proper candidate for hardware acceleration. This operation is a significant part of Density Functional Theory (DFT) [4] computations. The finite sum of the exponential functions is also present in other types of calculations, like Quantum Monte Carlo (QMC) [5, 6] or Time-Dependent Density Functional Theory (TDDFT) [7].

However, the implementation described here should be considered as the first step only. The ultimate goal is to design a hardware unit capable of generating the exchange-correlation potential matrix, which is one of the most computationally intensive routines within the Kohn-Sham (KS) formulation of DFT [4, 8].

2 KOHN-SHAM METHOD

The Kohn-Sham formalism is an approach to quantum-chemical calculations recently very popular for its competitive speed and accuracy. From the algorithmic point of view, it is a Self-Consistent Field (SCF) type procedure. The generalized eigenproblem of the Kohn-Sham operator F,

$$FC = SCE \tag{1}$$

is solved iteratively until the electron density is converged. The iterative procedure is required due to the fact that F depends on the eigenvectors C.

The Kohn-Sham operator matrix is defined as

$$F = T + V^{ext} + J + V^{xc} \tag{2}$$

where T is the kinetic energy matrix, V^{ext} is the external electrostatic potential matrix, J is the Coulomb matrix and V^{xc} stands for the exchange-correlation potential matrix. In vast majority of the molecular calculations, the matrices are in the atomic orbital (AO) basis [9], which, in turn, are linear combinations of Gaussian Type Orbitals (GTO) [9].

The exchange-correlation potential is the central notion of the Density Functional Theory in the Kohn-Sham formulation. At the same time, it is the only part of the Kohn-Sham operator matrix which is calculated by means of numerical integration [4].

In order to generate the V^{xc} matrix, the integral is approximated by finite sum over a three-dimensional grid

$$V^{xc}_{\mu\nu} = \iiint \chi_\mu(\mathbf{r})\hat{V}^{xc}\chi_\nu(\mathbf{r})\mathrm{d}^3\mathbf{r} \approx \sum_{\mathbf{g}\in G} w_{\mathbf{g}}\chi_\mu(\mathbf{g})\hat{V}^{xc}\chi_\nu(\mathbf{g}) \tag{3}$$

where $\hat{V}^{xc}$ is the exchange-correlation operator, $w_{\mathbf{g}}$ is the volume (weight) associated with the grid point $\mathbf{g}$ and χ stands for an atomic orbital. Therefore, the integration procedure has to calculate the AO values in every grid point.

The GTO-type atomic orbital is expressed by [9]

$$\chi_{klm}(\mathbf{r}) = \mathbf{r}_x^k\mathbf{r}_y^l\mathbf{r}_z^m \sum_i C_i e^{-\alpha_i \mathbf{r}^2} \tag{4}$$

where $k, l, m \in \mathbb{N}_0$ denote the type of the particular orbital. The C_i, α_i coefficients are predefined for particular orbitals in the so-called basis set. Therefore, it can be easily seen that generating the values of the exponential function is an important part of the exchange-correlation potential matrix evaluation.

3 ALGORITHM

The formula of Equation (4) was split into several sections [10] to allow adoption of a modular design approach. Consequently, a fully pipelined structure was obtained.

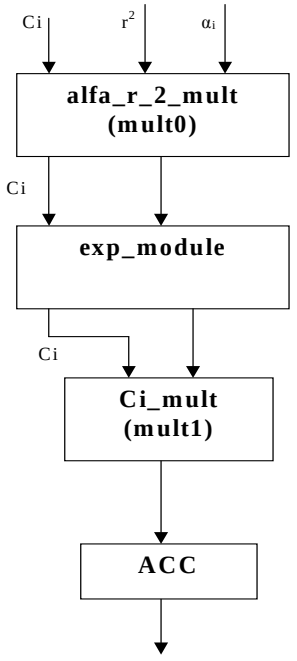


Fig. 1. EP module block diagram

Three different units have been employed within the EP module (Figure 1). All of them are specially designed, fully pipelined floating-point modules [11, 12] optimized to high speed performance. Input widths as well as their internal data path scale from single to double data precision. The input and output data format complies with the IEEE-754 floating-point standard. Nevertheless, intermediate data representations employ different non-standard floating-point format in order to reduce hardware resources.

Multiplier. For double precision data format inputs mantissa is 53-bit (54-bit including leading one) wide, therefore the product width is 108-bit wide. Such a bit-width is far beyond the required precision, therefore the LSBs of the product are usually disregarded [13]. Consequently the LSB's part of the multiplier performs operations which are not used in the next arithmetic operations. As a result, in the proposed architecture some of the LSBs logic is not implemented at all. This approach allows for significant area reduction. A more detailed description of the multiplier will be available in a separate paper.

Exponential function. This is mixed table-polynomial implementation of the exp function [11, 12], based on commonly known mathematical identities

$$e^x = 2^x * \log_2(e) = 2^{x_i} * e^{x - x_i / \log_2 e} \tag{5}$$

where x_i is an integer part of $x * \log_2 e$.

Accumulator. This is a fully pipelined mixed precision floating point unit. It can process one datum per clock cycle due to its advanced and optimized structure. A more detailed description of the accumulator will be available in a separate paper.

4 PLATFORM

The Altix 4700 series is a family of multiprocessor distributed shared memory (DSM) computer systems that currently ranges from 8 to 512 CPU sockets (up to 1 024 processor cores) and can accommodate up to 6 TB of globally shared memory in a single system while delivering a teraflop of performance in a small-footprint rack. The RASC module communicates with the Altix system in the same manner as any CPU does, i.e. employing NUMALink interconnection. Data transfer between the FPGA and NUMALink bus is realized by an application-specific integrated circuit (ASIC) denoted as TIO.

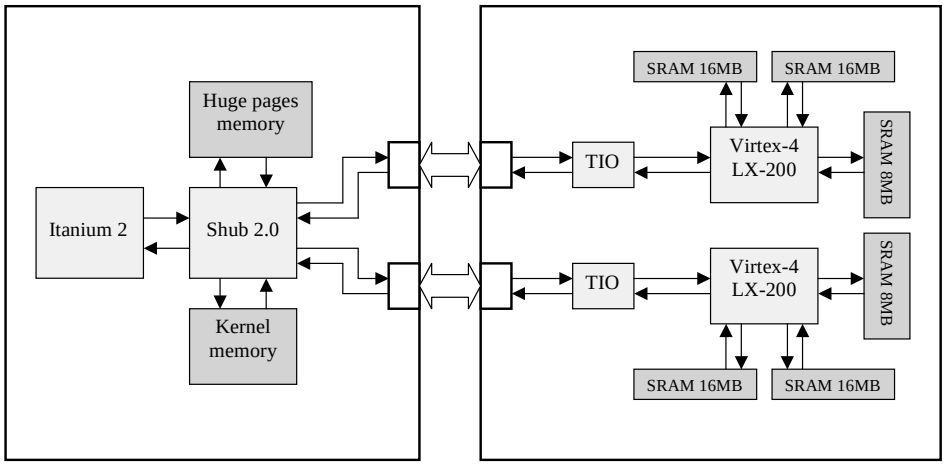


Fig. 2. RASC architecture and the host-FPGA data link

SGI RASC RC100 Blade consists of two Virtex-4 LX 200 [2] FPGAs, with 40 MB of external SRAM logically organized as two 16 MB blocks (as shown in Figure 2) and an 8 MB block. Each QDR SRAM block is capable of transferring 128-bit data every clock cycle (at 200 MHz) both for reads and writes. The RC100 Blade is

connected using the low latency NUMALink interconnect to the SGI Altix 4700 Host System, for a rated peak bandwidth of 3.2 GB per second in each direction. The RASC algorithm execution procedure, from the processor's perspective, is composed of several functions which reserve resources, queue commands and perform other preparation steps. The resource reservation procedure, once conducted, allows many runnings of the algorithm – which amounts to huge time savings, since the procedure takes approximately 7.5 ms. Therefore optimal structure of the hardware-accelerated application should contain as few initializations of the FPGA chips as possible so the FPGAs configuration time will be only a fraction of the algorithm execution time.

5 IMPLEMENTATION AND ACCELERATION RESULTS

The EP module implemention results on the RASC [2] platform are presented below. Core services (CS), provided by SGI together with the RASC module, is an extra logic incorporated in an FPGA which provides communication with the on-board hub. Units which the EP module is built of are strongly parametrized, therefore resources occupation varies depending on the data width (different calculation precision) of the module.

Implementation results	# 4-input LUT	# flip-flops	# 18 Kb BRAMs
EP module alone	2 229 (1 %)	1 975 (1 %)	2 (0.006 %)
EP module with the CS	11 560 (7 %)	15 922 (9 %)	25 (7 %)

Table 1. Implementation results of the EP module – single precision

Implementation results	# 4-input LUT	# flip-flops	# 18 Kb BRAMs
EP module alone	8 684 (4.8 %)	7 891(4 %)	6 (0.02 %)
EP module with the CS	18 015 (10 %)	21 838 (12 %)	29 (8 %)

Table 2. Implementation results of the EP module – double precision

Implementation results	# 4-input LUT	# flip-flops	# 18 Kb BRAMs
Exp()	820 (0.5 %)	920 (0.5 %)	2 (0.006 %)
Multiplier	549 (0.3 %)	444 (0.25 %)	0
Accumulator	860 (0.5 %)	605 (0.4 %)	0

Table 3. Implementation results of the elementary modules – single precision

The overall pipeline latency of the exp module is 33 clock cycles for the single precision data format. The latency strongly depends on the width of input data.

It is worth underlining that all the modules have adjustable widths of data paths within the range from single to double precision. The Hartree-Fock algorithm is executed differently, depending on the set of molecules it is employed for.

Implementation results	# 4-input LUT	# flip-flops	# 18 Kb BRAMs
Exp()	5 025 (3 %)	5 223 (3 %)	6 (1.8 %)
Multiplier	2 316 (1 %)	1 850 (1 %)	0
Accumulator	1 343 (0.5 %)	818 (0.4 %)	0

Table 4. Implementation results of the elementary modules – double precision

Module	Multiplier	Exp()	Accumulator	EP module
Pipeline delay [clk]	4	21	8	33

Table 5. Delays introduced by the EP module's components – single precision

This in turn impacts the precision requirements at different stages of the computations. It may happen that for a certain chemical substance some sections of the algorithm process much more data than others. Besides, in the Hartree-Fock algorithm the result precision increases (error decreases) with every algorithm iteration, therefore the calculation precision may be somehow scaled with the iteration number. Utilizing FPGAs in Hartree-Fock computations allows for the adjustment of data buses widths within the system so the proper precision is maintained across all the stages of the algorithm execution. The FPGA configuration time (7.5 ms) is significantly shorter than the overhead introduced by the oversized data format. It may appear that some sections of calculations can be successfully computed at single precision (or any other precision, e.g. 48-bits), saving a huge amount of resources that would otherwise be wasted if double precision data format was adopted. A hardware approach allows gradual switching from single to double data format together with the algorithm's rising demand for precision.

The authors' main concern is to determine sufficient precision at each stage of algorithm performance. To address it, specialized software has been designed that generates test vectors and monitors output of the algorithm's subroutines (Figure 3). This method is based on the observation of the impact of the decreasing input data precision on the output results. It is a very straightforward approach, which moreover does not require much chemical knowledge which is indispensable to apply some more sophisticated error calculation formulas. Provided that the proposed approach is employed with high granularity within a computational routine, it gives reliable results. Original pieces of code are replaced with fragments taken from the library containing components of an arbitrarily chosen precision, which allows the adoption of data length within the range from single to double precision.

Module	Multiplier	Exp()	Accumulator	EP module
Pipeline delay [clk]	5	30	10	45

Table 6. Delays introduced by the EP module's components – double precision

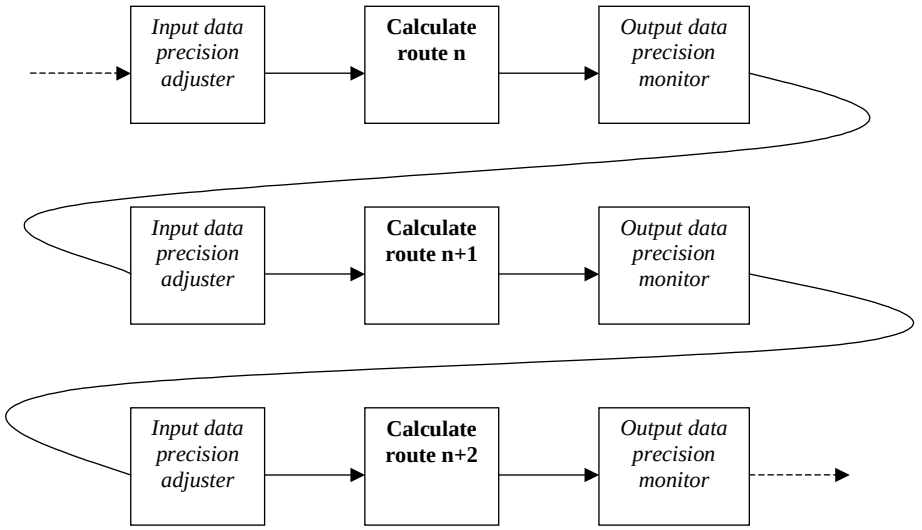


Fig. 3. Data width determination method

The EP module is intended to be a part of a larger system performing exchange-correlation potential calculation, so some preliminary tests were carried out in order to determine top achievable performance of the module on the RASC platform.

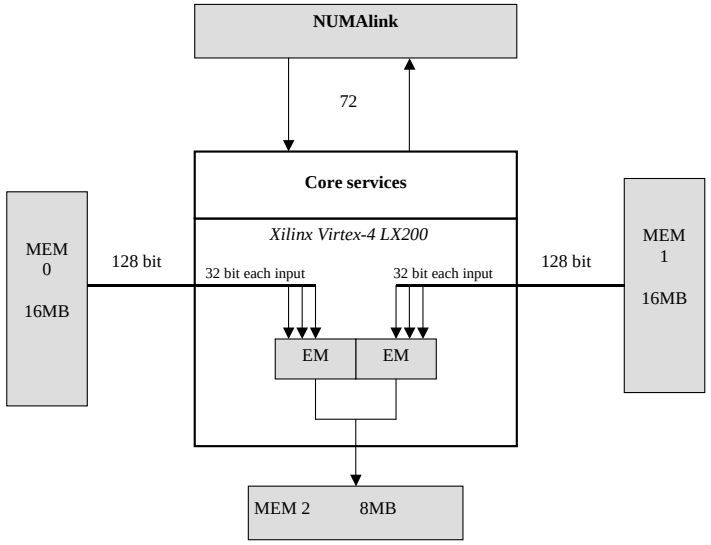


Fig. 4. Configuration of EP modules on the RASC platform

Each of the EP modules performs calculations in single precision so it is possible to aggregate two of them to increase overall throughput. Additionally due to the algorithm's structure some of the data (atom base coefficients) is used multiple times. So only a single 32-bit data chunk is streamed in and out the FPGA. This in turn allows to aggregate up to four EP modules (single precision) on the single FPGA. This holds as the parallization degree is limited by the external memory data transfer rate rater than the FPGA resources. Single exponential operation calculation together with two-directional data transfer takes roughly 8 ns on the RASC platform whereas the same operation executed on the Itanium takes roughly 20 ns for a highly optimized code. Both values were obtained as the result of a test conducted with the 400 000 input data vector.

Number of EP modules	[RASC/Itanium] speed-up
1	2.5
2	5
4	10

Table 7. Acceleration results

Employing both Xilinx Virtex 4 chips available on the RASC platform leads to 20 × speed-up compared to Itanium 2 1.6 GHz processor.

6 SUMMARY

It is worth noting that a finite sum of the exponential products (Equation 4) is, as a computational routine, ubiquitous in scientific calculations because of its universal function. Many different processes in the real world can be described by exponential function. Combination of the exp() leads to an even more uniform formula. Performance tests revealed that the FPGA is much faster than a processor – even with data stored in the processor's memory and the full optimization provided. It is worth taking into consideration that the Kohn-Sham algorithm, due to its iterative execution, allows the adoption of gradually adjustable data precision, which will give a speed boost to FPGAs in the final application. So there is still a huge potential in hardware implementation of quantum chemistry computational routines.

REFERENCES

[1] Underwood, K. D—Hemmert, K. S.—Ulmer, C.: Architectures and APIs: Assessing Requirements for Delivering FPGA Performance to Applications. Proceedings of the 2006 ACM/IEEE Conference on Supercomputing (Tampa, Florida, November 11–17, 2006). SC '06. ACM, New York, NY, 111.

[2] Silicon Graphics, Inc. Recon_gurable Application-Specific Computing User's Guide, Ver. 005, January 2007, SGI.

[3] Mazur, G.—Makowski, M.: Development and Optimization of Computational Chemistry Algorithms. Computing and Informatics, Vol. 28, 2009, No. 1, pp. 115–125.

[4] Koch, W.—Holthausen, M. C.: A Chemist's Guide to Density Functional Theory. Wiley 2001.

[5] Gothandaraman, A.—Peterson, G.—Warren, G.—Hinde, R.—Harrison, R.: FPGA Acceleration of a Quantum Monte Carlo Application. Parallel Computing, Vol. 34, 2008, Nos. 4-5, pp. 278–291.

[6] Gothandaraman, A.—Warren, G. L.—Peterson, G. D.—Harrison, R. J.: Reconfigurable Accelerator for Quantum Monte Carlo Simulations in N-Body Systems. Proceedings of the 2006 ACM/IEEE Conference on Supercomputing SC '06. ACM, New York, NY, 177.

[7] Casida, M. E.: Time-Dependent Density-Functional Theory for Molecules and Molecular Solids. J. Mol. Struc.-Theochem. 2009, Vol. 914, pp. 3–18.

[8] Kohn, W.—Sham, L. J.: Self-Consistent Equations Including Exchange and Correlation Effects. Phys. Rev. A, Vol. 140, 1965, p. 1133.

[9] Helgaker, T.—Jorgensen, P.—Olsen, J.: Molecular Electronic-Structure Theory. Wiley 2000.

[10] Omondi, A. R.: Computer Arithmetic Systems. Prentice Hall, Cambridge 1994.

[11] Wielgosz, M.—Jamro, E.—Wiatr, K.: Highly Efficient Structure of 64-Bit Exponential Function Implemented in FPGAs. ARC 2008, LNCS 4943, Springer-Verlag, London, pp. 274–279.

[12] Jamro, E.—Wielgosz, M.—Wiatr, K.: FPGA Implementation of 64-bit Exponential Function for HPC. FPL Netherlands, August 27–29, 2007, FPL Proceedings.

[13] Parhi, K. K.—Chung, J. G.–Lee, K. C.—Cho K. J.: Low-Error Fixed-Width Modified Booth Multiplier. US Paten: 957244.

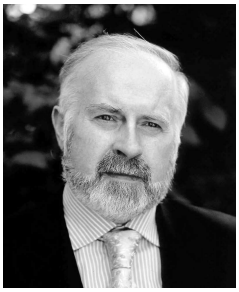
Maciej Wielgosz received his M. Sc. and Ph. D. degrees from the AGH University of Science and Technology (Poland, Cracow) in 2005 and 2010, respectively. His research interests include parallel computing on FPGAs, image processing, neural networks. He is currently developing an FPGA-based accelerator of quantum chemistry computations.

Grzegorz Mazur is an Assistant Professor at the Department of Computational Methods in Chemistry of Jagiellonian University in Cracow (Poland). He attained his Ph. D. in chemistry in 2001 at the same university. His current research interests include theoretical description of the excited states properties and optimization of quantum chemistry algorithms.

Marcin Makowski is an Assistant Professor at the Department of Theoretical Chemistry of Jagiellonian University in Cracow (Poland). He attained his M. Sc. degree in 2001 and his Ph. D. in 2004, both in chemistry, at the same university, and his B. Sc. degree in computer science at University of Mining and Metallurgy in Cracow in 2004. His current research interests include theoretical molecular spectroscopy, linear scaling methods in quantum chemistry and optimization of quantum chemistry algorithms. He participates in several research projects related with the development of theoretical chemistry formalisms and numerical methodologies that allows to calculate efficiently electronic structure of large molecular systems.

Ernest Jamro recieved his M. Sc. degree in electronic engineering from the AGH University of Science and Technology (UST), Cracow (Poland) in 1996; his M. Phil. degree from the University of Huddersfield (U.K.) in 1997; his Ph. D. degree from the UST in 2001. He is currently an Assistant Professor in the Department of Electronics, UST. His research interests include reconfigurable hardware (especially Field Programmable Gate Arrays – FPGAs), reconfigurable computing systems, System on Chip design, artificial intelligence.

Paweł Russek received his Ph. D. degree in electrical engineering from the AGH University of Science and Technology in Cracow in 2003. His research interests concern custom computing architectures, reconfigurable computing accelerators and digital design methods and tools. He is a Manager of Computing Acceleration Group in Academic Computing Center "Cyfronet" AGH. As a lecturer in Department of Electronics of AGH he provides lectures and laboratories in Embedded Systems and Programmable Logic Devices. He is an author or co-author of papers in professional journals and conference proceedings which regards reconfigurable logic.



Kazimierz Wiatr received the M. Sc. and Ph. D. degrees in electrical engineering from the AGH University of Science and Technology, Cracow, Poland, in 1980 and 1987, respectively, and the D. Hab. degree in electronics from the University of Technology of Łódź in 1999. He received the Professor degree in 2001. His research interests include design and performance of dedicated hardware structures and reconfigurable processors employing FPGAs for acceleration computing. He received 9 research grants from Polish Committee of Science Research. These works resulted in above 200 publications, including 3 books, the recent one being Acceleration Computing in Video Processing Systems. He is also the author of 5 patents and 35 industrial implementations. He was the reviewer of: IEEE Expert Magazine: Intelligent Systems, IEE Computer and Digital Techniques, IEE Electronic Letters, International Journal Eng. App. of Artificial Intelligence, IEEE Transactions on Neural Networks, Journal Machine Graphics and Vision, Eurasip Journal on Applied Signal Processing. He currently is the Director of Academic Computing Centre CYFORNET AGH, and is the head of PIONIER council – Polish Optical Internet. Last but not least, he is a member of the Polish Parliament (Senate), and the head of the Senate Science and Education Committee.