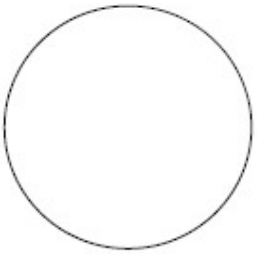

The IPSI Transactions
on
Advanced Research

Multi-, Inter-, and Trans-disciplinary Issues in Computer Science and Engineering

A publication of IPSI Internet Research Society New York, Frankfurt, Tokyo, Belgrade
January 2024 Volume 20 Number 1 (ISSN 1820-4511)

Table of Contents:

The IPSI Internet Research Society

The IPSI Internet Research Society is an association of people with professional interest in the field of the Internet. All members will receive IPSI Transaction Journals upon registering at the Society.

Member copies of Transactions are for personal use only
IPSI TRANSACTIONS ON ADVANCED RESEARCH
www.ipsitransactions.org

Editorial Board

Recent Advancements in DataFlow Computing

Korolija, Nenad; Salom, Jakob; and Milutinovic, Veljko

Abstract: ***Continues advancements in FPGAs along with ever increasing need for high performance computing further encouraged researchers to exploit FPGAs for dataflow computing. As a result, a wide range of applications have emerged. This work tries to encapsulate recent trends in dataflow computing. Research is presented in an uniform manner and further research directions are estimated. Results indicate that the utilization of FPGAs for high performance computing will further rise and it is expected that new fields of usage will appear.***

Index Terms: ***Dataflow, High performance computing, FPGAs, Hybrid architectures***

1. Introduction

This research tries to encapsulate new trends in dataflow computing that are based on utilization of Field Programmable Gate Arrays (FPGAs). The presentation is given in a uniform manner. Special attention is paid on future research directions, encouraging researchers to exploit available tools for enabling energy-efficient high performance computing.

There are numerous processor architectures that are based on von Neumann principles, but, nevertheless, they also use dataflow computing to increase instruction-level parallelism. In some cases, the compiler is responsible for the allocation of processing units (PUs), creating the schedule for the execution of instructions on the PUs, and the intercommunication between PUs. However, in traditional von Neumann processors, the common internal bus is an inherent limiting factor in exploiting the instruction-level parallelism.

In order to improve the processing speed, many processors are based on a hybrid von Neumann and dataflow computing model in which instructions that are repeatedly executed are scheduled for the execution on a dataflow hardware, and the communication between the processing elements is performed by sending semi-results directly from producer to the consumer, avoiding utilizing a common bus that would present a bottleneck.

Dataflow systems can roughly be divided onto software and hardware dataflow systems. While hardware dataflow systems are usually based on FPGAs, software dataflow systems run on control-flow architectures, exploiting the available parallelism that can be obtained using dataflow graphs. In this work, we are interested in hardware dataflow systems, but we also present software dataflow systems that are also applicable for the dataflow hardware.

Scientific applications of dataflow are numerous. Fields range from geoscience [BBĐ+23] and general physics [CKV22], to scientific [KT19]. One of the most important aspects of dataflow systems is scheduling tasks [KBHM22]. Another important aspect of dataflow systems is related to the capacity of universities to teach this paradigm [MSV+17], [BBB+23].

2. Existing Research

Authors in their work [SB23] define a dataflow process networks, as a model for computation. With fixed sets of nodes, the networks are also used as dataflow graphs as an intermediate program representation by compilers to

Manuscript received December 15, 2023.

N. K. is with School of Electrical Engineering, University of Belgrade, Serbia (e-mail: nenadko@etf.rs).

J. S. is with IPSI Belgrade, Serbia (e-mail: jakob.salom@yahoo.com).

V. M. is with University of Indiana, USA (e-mail: vmilutin@iu.edu).

uncover instruction-level parallelism of sequential programs. They introduce an abstraction of hybrid dataflow/von Neumann architectures [MTK+17] and consider the mapping of the dataflow graph to the actual processing units dataflow process network architecture, being careful to avoid conflicts with mapping multiple graph nodes to the same processing unit. They express both allocation and scheduling constraints to be handled by the propositional logic in the dataflow graph.

Authors [SKT+23] present the programming model that enable experimentation for non-experts users. Their goal architecture is a general purpose GPU cluster. They demonstrate how Celerity C++ programming model can be utilized for the development of distributed applications. Authors claim that the Celerity is scalable for multiple benchmarks on up to 128 GPUs.

Transforming the program so that the number of future uses of all variables can be accurately determined in order to predict the size and allocate the required buffer sizes in the later compilation phases is presented in [BS23]. Experimental results show an improvement of the processor performance with increasing buffer sizes and demonstrate the potential of buffered hybrid dataflow architectures for a scalable use of instruction level parallelism.

Dataflow tasks can have inter- or intra-application flows of data, creating dependencies among the tasks and causing resource constraints on shared storage systems. These constraints cause performance issues that are often solved manually, which demands holistic knowledge about the data dependencies and require knowing the infrastructure. Authors [CDNM+22] design DFMan, a graph-based dataflow management and optimization framework whose aim is maximizing bandwidth by leveraging the powerful storage stack on HPC systems, by devising a graph-based optimization algorithm, automatically carrying out co-scheduling of tasks and data placement. They achieve speedup factors between 1.29 and 5.42 on a wide variety of scientific workflows.

Recently, researchers have introduced a new computer architecture consisting of multiple different hardware[MK22], [MBJK], [KJMM22], [M+21], [MKR+21]. Usually, this hardware comprise of a control-flow and dataflow hardware. Some researchers estimate that these architectures might find their place in computer clouds [MPK22].

Researchers [Đ+23] have developed a method for using scratchpad memories for real-time systems. Their algorithm can predict the worst case memory access scenario and limit it, so that it is almost proportional to the required memory size.

3. Analysis of Existing Research and Potentials

This chapter presents an analysis of recent advancements in dataflow computing based on FPGAs. In general, all of the presented work adds certain complexity, but also benefits computer architectures based on FPGAs for future high performance computers. The progress ranges from defining the dataflow process network, dataflow management and optimization frameworks, and new programming models, to hybrid control-flow and dataflow chips.

As the computer architectures based on dataflow principles are proven to be faster and more energy efficient compared to their control-flow counterpart, they are becoming increasingly utilized in high performance computing. There is a vast range of applications of dataflow computer architectures, and so is the case with research on these architectures.

This analysis aims to help industries, but also researchers, to split existing research into categories, and apply what best suits them. The recent research presented in this paper is only one example. The model is applicable to any dataflow research.

First, orthogonal directions of research are chosen, and then explained. These directions include:

- Dataflow and control-flow synchronization,
- Data manipulation,
- Algorithm transformations,
- Automatic translation of algorithms,

- Frameworks for dataflow computing,
- Dataflow architectures,
- Cluster and cloud computing.

The applicability of dataflow computing is limited to high performance computing applications that exhibit repeatable execution of the same set of instructions over a prolonged time. Many algorithms suitable for the dataflow computing model also include data initialization and saving results, which are operations that dataflow hardware might not be suitable for. Therefore, dataflow hardware is usually utilized along with a control-flow processor. This opens a vast research field of synchronizing two types of hardware to achieve at least one of the following goals: reduced execution time, reduced resource requirements, and improved power efficiency.

Data manipulation is related to reorganizing the data so that the dataflow hardware achieves better performances. This direction is partially related to algorithm transformations and automatic translation of algorithms.

Control-flow algorithms are usually programmed solely for control-flow architectures. This is often not suitable for dataflow architectures. Therefore, algorithm transformations are often needed so that the optimal performance is achieved using the dataflow hardware.

Transforming any software created for control-flow algorithms into their dataflow representation requires certain effort. Researchers have been working on developing models for automatic translation of control-flow algorithms. They can be further divided into those that produce a hardware representation, or a semi-result, i.e. intermediate transformation, so that the result can be applied to multiple dataflow hardware types.

Frameworks for dataflow computing are becoming more important, as the dataflow computing is being recognized as a promising way to achieve better performance and lower power consumption. They usually include tools that help programmers develop dataflow algorithms without having to know dataflow hardware specifics.

Certain research is directed towards improving dataflow architectures. There are general dataflow architectures, and specialized ones for some purposes.

On top of the dataflow research, both programming models and dataflow architectures can be applied to clusters and computing clouds.

Any scientific paper about dataflow computing can belong to one of the mentioned research directions. In rare cases, a paper can overlap two or more directions of research.

The work in [SB23] aims to automatically translate algorithms. It defines networks with fixed sets of nodes, and compilers should be able to detect instruction-level parallelism. The work also defines an abstraction of hybrid dataflow/von Neumann architectures, including dataflow and control-flow synchronization processes.

Automatic translation of algorithms is the aim of many researchers. In the work [MTK+17], authors consider the mapping of the dataflow graph to the actual architecture, avoiding conflicts with mapping multiple graph nodes to the same processing unit.

The work [SKT+23] has two directions of research. It first presents the programming model, i.e. a framework for dataflow computing, that enables experimentation for non-expert users. The target architecture is a cluster, so the research is also oriented towards cluster and cloud computing.

The work presented in [BS23] is related to the algorithm transformation research. The research is on transforming a program, so that certain requirements can be known in advance.

Authors [CDNM+22] design a graph-based dataflow management and optimization framework, which overlaps with the research direction called data manipulation. The work is related to automatic translation of algorithms, as they automatically carry out co-scheduling of tasks and data placement.

Recently, many researchers worked on defining new computer architectures with multiple types of hardware on the same chip die [MK22], [MBJK], [KJMM22], [M+21], [MKR+21]. Therefore, they define new dataflow architectures, but their research is also related to the dataflow and control-flow synchronization. In some cases, authors also work

in the domain of cluster and cloud computing [MPK22].

The research directions of dataflow architectures also includes those architectural models where only a given set of an architecture is modified, i.e. memory system [Đ+23].

4. Conclusion

Dataflow systems are becoming increasingly important for numerous reasons. They have proven to be capable of faster execution of the same high performance computing algorithms comparing to their control-flow counterparts. At the same time, the size of the hardware can be considerably smaller. One of the most important aspects of modern computers is their energy efficiency. Dataflow hardware works on smaller frequencies compared to the architectures based on von Neumann principles. This leads to a lower power consumption. Researchers have reported energy consumption that is even couple of orders of magnitude lower. It is reasonable that the dataflow hardware is more efficient, as it doesn't have the bottleneck like common internal bus. Recent research suggests that the dataflow hardware is more applicable when combined with the control-flow hardware on the same chip die, so that the context switch is reasonably low. It is estimated that such systems will become the general computing systems. The ratio between the dataflow hardware and the control-flow hardware is to be determined in the future.

References

[Đ+23] Borislav Đorđević et al. Scratchpad memory unit in hybrid control-flow and dataflow architectures. In *Sinteza 2023-International Scientific Conference on Information Technology and Data Related Research*, pages 159–165. Singidunum University, 2023.

[BBĐ+23] Zoran Babović, Branislav Bajat, Vladan Đokić, Filip Đorđević, Dražen Drašković, Nenad Filipović, Borko Furht, Nikola Gačić, Igor Ikodinović, Marija Ilić, et al. ed civil-engineering and related scientific fields, using machine learning and big data: an overview of open problems. *Journal of Big Data*, 10(1):1–21, 2023.

[BBB+23] Zoran Babović, Branislav Bajat, Dusan Barac, Vesna Bengin, Vladan Đokić, Filip Đorđević, Dražen Drašković, Nenad Filipović, Stephan French, Borko Furht, et al. Teaching computing for complex problems in civil engineering and geosciences using big data and machine learning: synergizing four different computing paradigms and four different management domains. *Journal of Big Data*, 10(1):89, 2023.

[BS23] Anoop Bhagyanath and Klaus Schneider. Program balancing in compilation for buffered hybrid dataflow processors. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 57–66. IEEE, 2023.

[CDNM+22] Fahim Chowdhury, Francesco Di Natale, Adam Moody, Kathryn Mohror, and Weikuan Yu. Dfman: A graph-based optimization of dataflow scheduling on high-performance computing systems. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 368–378. IEEE, 2022.

[CKV22] Dušan Cvijetić, Nenad Korolija, and Marko Vojinović. Possibilities for parallelizing simplicial complexes simulation, 2022.

[KBHM22] Nenad Korolija, Dragan Bojić, Ali R Hurson, and Veljko Milutinović. A runtime job scheduling algorithm for cluster architectures with dataflow accelerators. In *Advances in computers*, volume 126, pages 201–245. Elsevier, 2022.

[KJMM22] Nenad Korolija, Vladisav Jelisavčić, Zlatogor Minchev, and Veljko Milutinović. Towards hybrid control-flow and dataflow architectures. *Archibald Reiss Days*, 12, 2022.

[KT19] Nenad Korolija and Roman Trobec. An overview of selected dataflow applications in physics simulations. *Exploring the DataFlow Supercomputing Paradigm: Example Algorithms for Selected Applications*, pages 229–240, 2019.

[M+21] Veljko Milutinović et al. The ultimate dataflow for ultimate supercomputers-on-a-chip, for scientific computing, geo physics, complex mathematics, and information processing. In *10th Mediterranean Conference on Embedded Computing (MECO) 2021*. IEEE, 2021.

[MBJK] Danko Miladinović, Miroslav Bojović, Vladisav Jelisavčić, and Nenad Korolija. Hybrid manycore dataflow processor.

[MK22] Kent Milfeld and Nenad Korolija. Towards hybrid supercomputing architectures. *Journal of Computer and Forensic Sciences*, 1(1):47–54, 2022.

[MKR+21] Veljko Milutinović, Miloš Kotlar, Ivan Ratković, Nenad Korolija, Miljan Djordjevic, Kristy Yoshimoto, and Mateo Valero. The ultimate data flow for ultimate super computers-on-a-chip. In *Handbook of Research on Methodologies and Applications of Supercomputing*, pages 312–318. IGI Global, 2021.

[MPK22] Danko Miladinović, Jovan Popović, and Nenad Korolija. The evolution of big data analytics solutions in the could. In *IX IcETRAN Conference*, pages 6–9, 2022.

[MSV+17] Veljko Milutinovic, Jakob Salom, Dragan Veljovic, Nenad Korolija, Dejan Markovic, and Luka Petrovic. *DataFlow supercomputing essentials: research, development and education*. Springer, 2017.

[MTK+17] Veljko Milutinović, Nemanja Trifunović, Nenad Korolija, Jovan Popović, and Dragan Bojić. Accelerating program execution using hybrid control flow and dataflow architectures. In *2017 25th Telecommunication Forum (TELFOR)*, pages 1–4. IEEE, 2017.

[SB23] Klaus Schneider and Anoop Bhagyanath. Consistency constraints for mapping dataflow graphs to hybrid dataflow/von neumann architectures. *ACM Transactions on Embedded Computing Systems*, 22(5):1–25, 2023.

[SKT+23] Philip Salzmann, Fabian Knorr, Peter Thoman, Philipp Gschwandtner, Biagio Cosenza, and Thomas Fahringer. An asynchronous dataflow-driven execution model for distributed accelerator computing. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 82–93. IEEE, 2023.

Nenad Korolija is with School of Electrical Engineering, University of Belgrade, Serbia. He received a PhD degree in electrical engineering and computer science in 2017. His interests and experiences include developing software for high performance computer architectures and Dataflow architectures. During 2008, he worked on the HIPEAC FP7 project at the University of Siena, Italy. In 2013, he was an intern at the Google Inc., Mountain View, California, USA. In 2017, he worked for Maxeler Ltd., London. During 2021, he worked for Johns Hopkins University on parallelizing the protein formation simulator.

Jakob Salom is with IPSI Belgrade, Serbia. His research interests reflects in his books, focusing on dataflow supercomputing, Mind Genomics, and Smart Cities. He is the co-editor-in-chief for IPSI BgD Transactions on Internet Research (TIR) and IPSI BgD Transactions on Advanced Research (TAR).

Veljko Milutinovic (1951) received his PhD from the University of Belgrade in Serbia, spent about a decade on various faculty positions in the USA (mostly at Purdue University and more recently at the University of Indiana in Bloomington), and was a co-designer of the DARPAs pioneering GaAs RISC microprocessor and the related GaAs Systolic Array with about 14000 GaAs microprocessors. Later, for almost three decades, he taught and conducted research at the University of Belgrade, in EE, MATH, BA, and PHYS/CHEM. His research is mostly in datamining algorithms and DataFlow computing, with the emphasis on mapping of data analytics algorithms onto fast energy efficient architectures. Most of his research was done in cooperation with industry (Intel, Fairchild, Honeywell, Maxeler, HP, IBM, NCR, RCA, etc...). For 10 of his books, forewords were written by 10 different Nobel Laureates with whom he cooperated on his past industry sponsored projects. He published 40 books (mostly in the USA), he has over 100 papers in SCI journals (mostly in IEEE and ACM journals), and he presented invited talks at over 400 destinations worldwide. He has well over 1000 Thomson-Reuters WoS citations, well over 1000 Elsevier SCOPUS citations, and about 4000 Google Scholar citations. He is a Life Fellow of the IEEE and a Member of The Academy of Europe and a Foreign Member of The Montenegro National Academy of Sciences and Arts.

For more details on semantics of your paper, see the following:

V. Milutinovic, "The Best Method for Presentation of Research Results," IEEE TCCA Newsletter, September 1997, pp. 1-6.

`http://www.computer.org/tab/tcca/news/sept96/sept96.htm`

Computing Based on Mapping of Structures and Algorithms onto Graphs for Hardware Implementation: 40 Years Later

Milutinović, Veljko

Abstract: The first written mention of computing based on mapping of algorithms onto graphs to be built into hardware dates back to January 1984, when the author of this text submitted a related proposal to the Defense Department of RCA, Camden, New Jersey, USA [1]. A preliminary study concluded that the optimal structure for mapping algorithms onto hardware is not of a square but of a hexagonal shape referred to as "honeycomb". That is how the term honeycomb was created in the context of mapping structures and algorithms into hardware. According to the open literature [2,3] the first structures ever mapped onto the honeycomb architecture are interconnection networks and the first algorithms ever mapped onto the honeycomb architecture are neural networks. If a structure or an algorithm is mapped onto a graph to be built into the hardware, conditions are met for data to flow from inputs to outputs, driven by the voltage difference between inputs and outputs. If that goal is accomplished, potentials are generated for data to move faster, which means that the algorithms are executed faster. If the data movement is accomplished with a slower clock, less energy is consumed during the execution of an algorithm. Consequently, since no control mechanisms have to be implemented, the volume of the engine becomes smaller. Finally, since the structure of the data path is formed at the time of mapping the graph into hardware, the width of the data path could be made as wide as needed and no wider than needed, which leads to a lot better precision at no cost in the domain of execution time. Of course, the described paradigm shift became utilizable in practice only after the reconfigurable hardware was invented. That is why the first data flow machines, based on the same or similar concept, appeared only after the FPGA circuits became widespread. However, the full utilization of this paradigm will be possible only after the Analog See of Gates circuits become feasible [4], which enables a lot lower operational frequency (a lot lower power consumption) and a lot higher speed (due to fewer obstacles on the data propagation paths). This article sheds light on the complexity and speed related to the mapping of structures and algorithms.

1. INTRODUCTION

The quality of computing is primarily judged via the following four parameters: (a) Speed, (b) Power, (c) Precision, and (d) Volume. The lower the Power and Volume, the better [5]. The higher the Speed and Precision, the better. Higher speed means less time for the final result. Lower power means that the electricity bills at the end of the month become lower. Higher precision means that hidden knowledge could be noticed, and smaller volume means that rental fees become lower.

The currently dominating Control Flow paradigm achieves better speed via the exploitation of parallelism and changes in technology. In Control Flow, precision could be increased only at the expense of execution slowdowns due to the time needed for processing multiple data. The Control Flow paradigm burns more energy due to the fact that speed-ups are often achieved via a higher clock. In Control Flow machines the volume is larger due to the fact that the majority of the transistor budget is not spent for arithmetic or logic, but for control mechanisms of all kinds (data caches, instruction caches, data prediction, branch prediction, memory management, i/o management, etc).

In the Data Flow machines, all four desired characteristics are achieved via paradigm changes. The essence of the paradigm is best noticed when one analysis the mapping of structures and algorithms. Consequently, the stress of this article is on the effective mapping of structures and algorithms.

For mapping of structures, interconnection networks have been chosen, since they represent the essential aspect of any structure that moves data. The area analysis and the time analysis results are based on a PhD thesis from the 1990s [6].

For mapping of algorithms, the essence is in transforming an algorithm into a graph, so that data flows through the given graph and at the end of the flow, data flow produces the results inherent to the algorithm used. In other words, the mapping of algorithms boils down to the mapping of structures. This has been shown through the example of neural networks [7].

2. Problem Statement

The mapping of algorithms into a graph has been researched for decades now. Mapping of graphs onto hardware is a research avenue that has started with the appearance of programmable logic. Both research avenues have to be correlated for the maximum effectiveness of the final results. This means that mathematicians and computer scientists have to cooperate.

In conditions when the main contributors to speed slowdown and power dissipation are longer edges of the graph, the main problem is how to generate a graph with edges as short as absolutely possible, having in mind the real geometries of the reconfigurable hardware to be used later to host the developed graph, after the mapping process is completed.

The problem is important because the higher speed and the lower dissipation are crucial for many applications. The importance of the problem will grow over time because the pressure on speed and power keeps increasing. Speed and power also could impact the precision of computing and the size of equipment.

3. Existing Solutions

Existing solutions in the domain of implementation of interconnection networks are based on wiring that follows various topologies of importance for the algorithms to be using the implemented interconnection networks. Research activity of H.J. Siegel states that essentially there are 17 different types of interconnection networks for parallel processing, some more important than others [8].

Existing implementations of neural networks are done mostly in software and 40 different algorithms were elaborated in [9]. Hardware implementations are based on interconnection networks, so the presentation of the mapping of interconnection networks also sheds light on the mapping of neural networks.

On the system level, it is worth mentioning here the research efforts related to: Sea of Gates VLSI starting with [10], Cellular Arrays [11], Connectionist Machine [12], and the most recent FPGA-related efforts. In all these cases, interconnection networks play an important role, so the following pages summarize the types and characteristics of the major eight types of interconnection networks.

4. Mapping of Interconnection Networks onto the Honeycomb Architecture

The honeycomb architecture consists of an array of ALUs that you connect to neighboring ALUs with wires of zero length or almost zero length, which enables minimal power dissipation and maximal internal speed, due to the small VLSI area.

Each ALU could be used either to perform a simple ALU operation or to pass the data signal to the next destination. Passing to the next

destination could follow the rules of different interconnection network types and consequently implement the given interconnection network, with fast propagation times.

Details of the mapping process, as well as the time and area analysis, could be found in the PhD thesis of Dragana Milutinovic at the University of Belgrade in 1995, based on the work done from 1984 to 1989. The same methodology could be used for the mapping of any other structure or algorithm, as well as for the time and area analysis.

5. CONCLUSION

The ideas behind the honeycomb architecture and the mapping of algorithms onto it have developed into many different aspects of mapping the algorithm onto the FPGA structures, using the execution graph structures of algorithm representation.

These ideas will become more relevant once the new and more sophisticated types of FPGAs become available. From the point of view of the proposed honeycomb paradigm, the ideal circuits are those that enable data signals to propagate from inputs to outputs without obstacles (flip-flops).

The newly open problems are related to the mapping of all possible algorithms, plus the determination of area and time functions for each and every algorithm mapped. Once the clear picture is known for all relevant algorithms, one will be able to predict execution times and VLSI costs more precisely.

REFERENCES

[1] Milutinovic, V., “Proposal for Advanced Research in Computer Architecture,” RCA, Camden, New Jersey, USA.

[2] Milutinovic, D., “Mapping of Interconnection Networks for Parallel Processing onto the Advanced Sea-of-Gates VLSI,” Journal of Systems Architecture, Elsevier, 1997.

[3] Milutinovic, V., “Mapping of Neural Networks on the Honeycomb Architecture,” Proceedings of the IEEE, 1989.

[4] Duchene, P., Declecq, M.J., “A Highly Flexible Sea of Gates Structure for Digital and Analog Applications,” IEEE Journal of Solid State Circuits, 1989.

[5] Milutinovic, et al, “Guide to DataFlow SuperComputing,” Springer Nature, 2015.

[6] Milutinovic, D., “Mapping of Interconnection Networks onto a Special VLSI Structure,” Ph.D. Thesis, University of Belgrade, Serbia, 1995.

[7] Antogneti, P., Milutinovic, V. (editors), “Neural Networks: Concepts, Applications, and Implementations,” Prentice Hall, 1991.

[8] Siegel, H.J., “Interconnection Networks,” Purdue University Technical Report, TR-EE 83-11, 1983.

[9] Mueller, B., et al, “Neural Networks,” Springer 1995.

[10] Myer, E, “Jambo Arrays Scale 250,000 Gates,” Computer Design, March 1, 1990, pp.28-36.

[11] Kan, T., et al, “Parallel Processing on the CAP,” Proceedings of the COMCON, 1984.

[12] Hills, W.D., “The Connectionist Machine,” MIT Press, 1985.

Veljko Milutinovic received his PhD from the University of Belgrade in Serbia, spent about a decade on various faculty positions in the USA (mostly at Purdue University and more recently at the University of Indiana in Bloomington), and was a co-designer of the DARPAs pioneering GaAs RISC microprocessor and the related GaAs Systolic Array with about 14000 GaAs microprocessors. Later, for almost three decades, he taught and conducted research at the University of Belgrade, in EE, MATH, BA, and PHYS/CHEM. His research is mostly in datamining algorithms and DataFlow computing, with the emphasis on mapping of data analytics algorithms onto fast energy efficient architectures. Most of his research was done in cooperation with industry (Intel, Fairchild, Honeywell, Maxeler, HP, IBM, NCR, RCA, etc...). For 10 of his books, forewords were written by 10 different Nobel Laureates with whom he cooperated on his past industry sponsored projects. He published 40 books (mostly in the USA), he has over 100 papers in SCI journals (mostly in IEEE and ACM journals), and he presented invited talks at over 400 destinations worldwide. He has well over 1000 Thomson- Reuters WoS citations, well over 1000 Elsevier SCOPUS citations, and about 4000 Google Scholar citations. He is a Life Fellow of the IEEE and a Member of The Academy of Eu- rope and a Foreign Member of The Montenegro National Academy of Sciences and Arts.

Reviewers:

Australia
Abramov, Vyacheslav; Monash University
Begg, Rezaul; Victoria University
Bem, Derek; University of Western Sydney
Betts, Christopher; Pegacat Computing Pty. Ltd.
Buyya, Rajkumar; The University of Melbourne
Chapman, Judith; Australian University Limited
Chen, Yi-Ping Phoebe; Deakin University
Hammond, Mark; Flinders University
Henman, Paul; University of Queensland
Palmisano, Stephen; University of Wollongong
Ristic, Branko; Science and Technology Organisation
Sajjanhar, Atul; Deakin University
Sidhu, Amandeep; University of Technology, Sydney
Sudweeks, Fay; Murdoch University

Austria
Derntl, Michael; University of Vienna
Hug, Theo; University of Innsbruck
Loidl, Susanne; Johannes Kepler University
Linz Stockinger, Heinz; University of Vienna
Sutter, Matthias; University of Innsbruck

Brazil
Parracho, Annibal; Universidade Federal
Fluminense Traina, Agma; University of Sao Paulo
Traina, Caetano; University of Sao Paulo
Vicari, Rosa; Federal University of Rio Grande

Belgium
Huang, Ping; European Commission

Canada
Fung, Benjamin; Simon Fraser University
Grayson, Paul; York University Gray,
Bette; Alberta Education
Memmi, Daniel; UQAM
Neti, Sangeeta; University of Victoria
Nickull, Duane; Adobe Systems, Inc. Ollivier-Gooch,
Carl; The University of British Columbia Paulin,
Michele; Concordia University
Plaisent, Michel; University of Quebec Reid, Keith;
Ontario Ministry og Agriculture Shewchenko,
Nicholas; Biokinetics and Associates Steffan,
Gregory; University of Toronto Vandenberghe,
Christian; HEC Montreal

Czech Republic
Kala, Zdenek; Brno University of Technology
Korab, Vojtech; Brno University of technology
Lhotska, Lenka; Czech Technical University

Finland
Lahdelma, Risto; University of Turku
Salminen, Pekka; University of Jyvaskyla

France
Cardey, Sylviane; University of Franche-Comte
Klinger, Evelyne; LTCI – ENST, Paris
Roche, Christophe; University of Savoie
Valette, Robert; LAAS - CNRS

Germany
Accorsi, Rafael; University of Freiburg
Glatzer, Wolfgang; Goethe-University
Gradmann, Stefan; Universitat Hamburg
Groll, Andre; University of Siegen
Klamma, Ralf; RWTH Aachen University
Wurtz, Rolf P.; Ruhr-Universitat Bochum

Hungary
Mester, Gyula; Óbuda University, Budapest

India
Pareek, Deepak; Technology4Development
Scaria, Vinod; Institute of Integrative Biology
Shah, Mugdha; Mansukhlal Svayam

Ireland
Eisenberg, Jacob; University College Dublin

Israel
Feintuch, Uri; Hadassah-Hebrew University

Italy
Badia, Leonardo; IMT Institute for Advanced
Studies Berrittella, Maria; University of Palermo
Carpaneto, Enrico; Politecnico di Torino

Japan
Hattori, Yasunao; Shimane University
Livingston, Paisley; Linghan University
Srinivas, Hari; Global Development Research Center
Obayashi, Shigeru; Institute of Fluid Science,
Tohoku University

Netherlands
Mills, Melinda C.; University of Groningen
Pires, Luís Ferreira; University of Twente

New Zealand
Anderson, Tim; Van Der Veer Institute

Portugal
Cardoso, Jorge; University of Madeira
Natividade, Eduardo; Polytechnic Institute of
Coimbra Oliveira, Eugenio; University of Porto

Singapore
Tan, Fock-Lai; Nanyang Technological University

South Korea
Kwon, Wook Hyun; Seoul National University

Spain
Barrera, Juan Pablo Soto; University of Castilla
Gonzalez, Evelio J.; University of La Laguna
Perez, Juan Mendez; Universidad de La
Laguna Royuela, Vicente; Universidad de
Barcelona Vizcaino, Aurora; University of
Castilla-La Mancha Vilarrasa, Clelia Colombo;
Open University of Catalonia

Sweden
Johansson, Mats; Royal Institute of Technology

Switzerland
Niinimaki, Marko; Helsinki Institute of Physics
Pletka, Roman; AdNovum Informatik AG
Rizzotti, Sven; University of Basel
Specht, Matthias; University of Zurich

Taiwan
Lin, Hsiung Cheng; Chienkuo Technology University
Shyu, Yuh-Huei; Tamkang University
Sue, Chuan-Ching; National Cheng Kung
University

United Kingdom
Ariwa, Ezendu; London Metropolitan University
Biggam, John; Glasgow Caledonian University
Coleman, Shirley; University of Newcastle
Conole, Grainne; University of Southampton
Dorfler, Viktor; Strathclyde University
Engelmann, Dirk; University of London
Eze, Emmanuel; University of Hull
Forrester, John; Stockholm Environment Institute
Jensen, Jens; STFC Rutherford Appleton
Laboratory Kolovos, Dimitrios S.; The University
of York McBurney, Peter; University of Liverpool
Vetta, Atam; Oxford Brookes University
WHYTE, William Stewart; University of Leeds
Xie, Changwen; Wicks and Wilson Limited

USA
Bach, Eric; University of Wisconsin Bolzendahl,
Catherine; University of California Bussler,
Christoph; Cisco Systems, Inc. Charpentier,
Michel; University of New Hampshire Chong,
Stephen; Cornell University
Collison, George; The Concord Consortium
DeWeaver, Eric; University of Wisconsin -
Madison Gans, Eric; University of California
Gill, Sam; San Francisco State University
Hunter, Lynette; University of California Davis
Iceland, John; University of Maryland
Kaplan, Samantha W.; University of Wisconsin
Langou, Julien; The University of Tennessee
Liu, Yuliang; Southern Illinois University
Edwardsville Lok, Benjamin; University of Florida
Minh, Chi Cao; Stanford University
Morrissey, Robert; The University of Chicago
Mui, Lik; Google, Inc
Rizzo, Albert ; University of Southern
California Rosenberg, Jonathan M. ; University
of Maryland Shaffer, Cliff ; Virginia Tech
Sherman, Elaine; Hofstra University
Snyder, David F.; Texas State University
Song, Zhe; University of Iowa
Wei, Chen; Intelligent Automation, Inc.
Yu, Zhiyi; University of California

Welcome to IPSI Conferences and Journals!

http://tar.ipsitransactions.org

http://www.ipsitransactions.org

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd
ISSN 1820 – 4511
The IPSI Transactions
on Advanced Research
COBISS.SR - ID 119128844