

- [55] R. W. Hartenstein, R. Kress, H. Reinig: A Dynamically Reconfigurable Wavefront Array Architecture for Evaluation of Expressions; Proc. Int'l Conference on Application-Specific Array Processors, ASAP'94, San Francisco, IEEE Computer Society Press, Los Alamitos, CA, Aug. 1994
- [56] K. Schmidt: A Program Partitioning, Restructuring, and Mapping Method for Xputers; Ph.D. Thesis, Kaiserslautern 1994
- [57] R. Hartenstein, J. Becker, M. Herz, R. Kress, U. Nageldinger: A Parallelizing Programming Environment for Embedded Xputer-based Accelerators; High Performance Computing Symposium '96, Ottawa, Canada, June 1996
- [58] R. W. Hartenstein, J. Becker: Hardware/Software Co-Design for data-driven Xputer-based Accelerators; Proc. of 10th Int'l. Conf. on VLSI Design, January 4-7 1997, Hyderabad, India
- [59] A. Ast, J. Becker, R. W. Hartenstein, R. Kress, H. Reinig, K. Schmidt: Data-procedural Languages for FPL-based Machines; 4th Int'l Workshop on Field Programmable Logic and Applications, FPL'94, Prague, Sept. 7-10, 1994, Lecture Notes in Computer Science, Springer, 1994
- [60] R. Hartenstein, A. Hirschbiel, M. Weber: MoM - a partly custom-designed architecture compared to standard hardware; Proc. COMP EURO, Hamburg, Germany, 1989; IEEE Press 1989
- [61] A. Hirschbiel: A Novel Processor Architecture Based on Auto Data Sequencing and Low Level Parallelism; Ph.D. Thesis, Kaiserslautern, 1991
- [62] R. W. Hartenstein, M. Riedmüller, K. Schmidt, M. Weber: A Novel Asic Design Approach Based on a New Machine Paradigm; Special Issue of IEEE Journal of Solid State Circuits on ESSCIRC'90, July 1991
- [63] G. J. Lipovski: On a Stack Organization for Microcomputers; in [64]
- [64] R. Hartenstein, R. Zaks: Microarchitecture of Computer Systems; North Holland 1975
- [65] H. Corporaal, H. Mulder: MOVE: A framework for high-performance processor design; Proc. Supercomputing '91, Albuquerque, IEEE Computer Society Press, November 1991
- [66] H. Corporaal, P. van der Arend: MOVE32INT, a Sea of Gates realization of a high performance Transport Triggered Architecture; Microprocessing and Microprogramming vol. 38, pp. 53-60, North-Holland, 1993
- [67] H. Corporaal: Evaluating Transport Triggered Architectures for scalar applications; Transport Triggered Architecture; Microprocessing and Microprogramming vol. 38, pp. 45-52, North-Holland, 1993
- [68] H. Corporaal: Transport Triggered Architectures; Ph. D. thesis, Tech. University of Delft, Holland, 1995
- [69] R. Hartenstein, A. Hirschbiel, K. Schmidt, M. Weber: A novel Paradigm of Parallel Computation and its Use to implement Simple High-Performance Hardware; Future Generation Computing Systems 7 (1991/92), (invited reprint from [72])
- [70] T. Ungerer: Datenflußrechner; Teubner, 1993
- [71] J.-L. Gaudiot, L. Bic (Eds.): Advanced Topics in Data Flow Computing; Prentice-Hall, 1991
- [72] R. Hartenstein, A. Hirschbiel, M. Weber: A Novel Paradigm of Parallel Computation and its Use to Implement Simple High Performance Hardware; InfoJapan'90- International Conference memorating the 30th Anniversary of the Computer Society of Japan, Tokyo, Japan, 1990
- [73] R. Hartenstein, M. Riedmüller, K. Schmidt, M. Weber: A Novel ASIC Design Approach based on a New Machine Paradigm; IEEE Journal of Solid State Circuits, July 1991 (invited reprint from [62])
- [74] see world-wide web address: http://xputers.informatik.uni-kl.de/index_academic.html
- [75] R. W. Hartenstein, J. Becker, R. Kress, H. Reinig, K. Schmidt: A Reconfigurable Machine for Applications in Image and Video Compression; Conference on Compression Technologies and Standards for Image and Video Compression, Amsterdam, The Netherlands, March 1995
- [76] J. Becker, R. W. Hartenstein, R. Kress, H. Reinig: A Reconfigurable Parallel Architecture to Accelerate Scientific Computation; Proc. Int'l Conf. on High Performance Computing, New Delhi, India, Dec 1995



- [14] R. Hartenstein, J. Becker, R. Kress: An Embedded Accelerator for Real Time Image Processing; 8th EUROMI-CRO Workshop on Real Time Systems, L'Aquila, Italy, June 1996
- [15] S. Guccione, M. Gonzales: Classification and Performance of Reconfigurable Architectures; FPL '95 - Int'l Symposium on field-programmable Logic and Applications, Oxford, UK, 29-31 August 1995
- [16] B. K. Fawcett: FPGAs as Configurable Computing Elements; Int'l Workshop on Reconfigurable Architectures @ IPPS '95 - 9th Int'l Parallel Processing Symposium, Santa Barbara, CA, 24-29 April 1995
- [17] A. Koch, U. Golze (Technical University Braunschweig, Germany): A Universal Co-Processor for Workstations; in: W. R. Moore, W. Luk (eds.): More FPGAs; Abbingdon EE&CS Books, Oxford, UK 1993 (selection from Proc. Int'l Symposium on field-programmable Logic and Applications, Oxford, UK, Sept. 1993)
- [18] R. Gupta, G. De Micheli: Hardware-Software Cosynthesis for Digital Systems; IEEE Design & Test, Sep 1993
- [19] R. Hartenstein, J. Becker, R. Kress: Two-Level Hardware/Software Partitioning Using CoDe-X; Int'l IEEE Symp. on Engineering of Computer Based Systems (ECBS), Friedrichshafen, Germany, March 1996
- [20] A. Agarwal: Hot Machines; Proc. Int'l Conf on High Performance Computing; Dec. 1995, New Delhi, India
- [21] S. Y. Kung: VLSI Array Processors; Prentice-Hall, 1988
- [22] N. Petkov: Systolische Algorithmen und Arrays; Akademie-Verlag, Berlin 1989
- [23] A. Postula, D. Abramson, P. Logothetis: Synthesis for Prototyping of Application-specific Processors; Proc. 3rd Asia Pacific Conf. on Hardware Description Languages (APCHDL'96), Bangalore, India, Jan. 1996
- [24] R. Hartenstein, J. Becker, R. Kress: Application Specific Design Methodologies: General Model vs. Tinker Toy Approach; GI/ITG Workshop Custom Computing Machines, Schloß Dagstuhl, Wadern, Germany, Juni 1996
- [25] A. Postula, D. Abramson, P. Logothetis: Synthesis for Prototyping of Application Specific Processors; Proc. 3rd Asia Pacific Conference on Hardware Description Languages (APCHDL'96), Bangalore, India, Jan. 1996
- [26] Ing-Jer Huang, A. Despain: Synthesis of Application Specific Instruction Sets; IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 6, June 1995
- [27] H. Akaboshi, H. Yasuura: COACH: A Computer Aided Design Tool for Computer Architects; IEICE Trans. Fundamentals, Vol. E76-A, No. 10, Oct. 1993
- [28] J. Sato, M. Imai, T. Hakata, A. Y. Alomary, N. Hikichi: An Integrated Design Environment for Application Specific Integrated Processor; Proc. IEEE Int'l Conf. on Computer Design: ICCD 1991, pp. 414-417, Oct 1991
- [29] D. Buell, K. Pocek: IEEE Worksh. on FPGAs for Custom Computing Machines (FCCM'93) Napa, CA, April 1993
- [30] D. Buell, K. Pocek: IEEE Worksh. on FPGAs for Custom Computing Machines (FCCM'94), Napa, CA, April 1994
- [31] P. Athanas, K. Pocek: IEEE Worksh. on FPGAs for Custom Computing Machines (FCCM'95), Napa, CA, April 1995
- [32] J. Arnold, K. Pocek: IEEE Worksh. on FPGAs for Custom Computing Machines (FCCM'96), Napa, CA, April 1996
- [33] R. Hartenstein: Custom Computing Machines (opening keynote); DMM'95 - Int'l Symp. on Design Methodologies in Microelectronics, Smolenice Castle, Slovakia, September 1995
- [34] R. Hartenstein: Custom Computing Machines; aktuelles Schlagwort; GI Informatik-Spektrum 18: (p. 228-229, Springer-Verlag, Juni, 1995)
- [35] N. N.: FPL Bibliography; <http://splash.ee.byu.edu/bib/bibpage.html> - (ask wirthlin@fpga.ee.byu.edu)
- [36] S. Guccione: List of FCCMs; http://www.io.com/~guccione/HW_list.html - (ask guccione@io.com)
- [37] K. Buchenrieder: Hardware/Software Co-Design; IT Press, Chicago 1995 [ISBN 0-9639887-7-8]
- [38] R. Gupta: Hardware/Software Co-Design; Proc. 9th Int'l Conf. on VLSI Design, Bangalore, India, Jan 1996
- [39] R. Gupta: Co-Synthesis of Hardware and Software for Digital Embedded Systems; Kluwer, 1995
- [40] R. Hartenstein: Hardware/Software Co-Design; aktuelles Schlagwort; GI Informatik-Spektrum 18 (p. 286-287, Springer-Verlag, Oktober, 1995)
- [41] Proc. 1st Int'l Worksh. on Hardware/Software Co-Design CODES/CASHE '92, Estes Park, Colorado, 1992
- [42] Proc. 2nd Int'l Workshop on Hardware/Software Co-Design CODES/CASHE '93, Innsbruck, Austria, 1993
- [43] Proc. 3rd Int'l Worksh. on Hardware/Software Co-Design CODES/CASHE '94, Grenoble, France, Sept 1994
- [44] Proc. 4th Int'l Workshop on Hardware/Software Co-Design CODES/CASHE '96, Pittsburgh, USA, March 1996
- [45] B. K. Fawcett: FPGAs as Configurable Computing Elements; Int'l Worksh. on Reconfigurable Architectures, @ ISPS '95 - 9th Int'l Parallel Processing Symposium, Santa Barbara, 24. - 29. April 1995
- [46] R. Gupta: Cosynthesis of Hardware and Software for Digital Embedded Systems; Kluwer 1995
- [47] R. Gupta, G. de Micheli: Hardware/Software Co-Synthesis for Digital Systems; IEEE D&T o'Computers, Sept 1993
- [48] K. Buchenrieder: <http://ti-ibm06.informatik.uni-tuebingen.de/~buchen/> (ask Buchenrieder@zfe.siemens.de)
- [49] J. N. Riis Grode: <http://www.ifi.uio.no/~olavlok/co-design.html> (list of projects, ask jnrg@it.dtu.dk)
- [50] R. W. Hartenstein (key note): High Performance Computing: über Szenen und Krisen; GI/ITG Workshop über Custom Computing Machines, Schloß Dagstuhl, Wadern, Germany, Juni 1996
- [51] R. W. Hartenstein, J. Becker, M. Herz, R. Kress, U. Nageldinger: Co-Design and High Performance Computing: Scenes and Crisis; Proc. of Reconfigurable Technology for Rapid Product Development & Computing, Part of SPIEs International Symposium 96, Boston, USA, Nov. 1996
- [52] P. Treleaven, M. Pacheco, M. Vellasco: VLSI Architectures for Neural Networks, IEEE Micro, Vol. 9, Nr. 6
- [53] R. Hartenstein, R. Kress: A Datapath Synthesis System for the Reconfigurable Datapath Architecture; Asia and South Pacific Design Aut. Conf., ASP-DAC'95, Makuhari, Chiba, Japan, Aug. 29 - Sept. 1, 1995
- [54] R. Kress: A Fast Reconfigurable ALU for Xputers; Ph.D. Thesis, University of Kaiserslautern, 1996



5. Summary

The paper has introduced a fundamentally new machine paradigm, which takes into account, that hardware has become soft. Along with a compilation technique, this paper has introduced its implementation and application by illustrating the rDPA (reconfigurable data path array), an integrated circuit designed by Rainer Kress and submitted for fabrication. This is the first implementation of a field-programmable data path array (FPDPA), which uses configurable ALU blocks (CABs) instead of CLBs (configurable logic blocks). This novel platform provides highly efficient means of instruction level parallelism: for many applications by orders of magnitude more efficient than the traditional parallelism used in MPC or HPC or supercomputing.

6. Conclusions

We have introduced a paradigm, which integrates programming and resources for computing in time and for computing in space. This novel d-paradigm has mainstream potential. It has the chance to break the monopoly of the von Neumann paradigm as a conceptual backbone of computer science. It could become as important for structurally programmable information processing systems, as the von Neumann paradigm is for procedural computing systems. In Industry and academia already for many years system designs have been practiced, which integrate both worlds, although more by tinker toy approaches. However, the discipline of systolic arrays, where both computing worlds are overlapping, has developed more formal mapping methods, which are strong links between these two worlds. The new paradigm shows a way toward a systematic dichotomy of computing sciences, which integrates computing in time and computing in space.

Literature

- [1] T. Lewis: The next 10,000₂ years: part I; IEEE Computer, April 1996 - part II: IEEE Computer, May 1996
- [2] H. Grünbacher, R. Hartenstein (Editors.): Field-Programmable Gate Arrays: Architectures and Tools for Rapid Prototyping; Second International Workshop on Field-Programmable Logic and Applications, Vienna, Austria, Aug./Sept. 1992; Lecture Notes in Computer Science 705, Springer-Verlag, 1992
- [3] R. Hartenstein, M. Servít (Editors.): Field-Programmable Logic: Architectures, Synthesis and Applications; Fourth International Workshop on Field-Programmable Logic and Applications, Prague, Czech Republic, Sept. 1994; Lecture Notes in Computer Science 849, Springer-Verlag, 1994
- [4] W. Moore, W. Luk (Editors.): Field-Programmable Logic and Applications; Fifth International Workshop, Oxford, United Kingdom, Aug./Sept. 1995; Lecture Notes in Computer Science 975, Springer-Verlag, 1995
- [5] M. Glesner, R. Hartenstein (Editors.): Field-Programmable Logic and Applications; Sixth International Workshop, Darmstadt, Germany, Sept. 1996; Lecture Notes in Computer Science, Springer-Verlag, 1996
- [6] N. N.: WILDFIRE Custom Configurable Computer WAC4010/16; Document # 11502-0000, Rev. C, Anapolis Micro Systems, Inc., April 1995
- [7] P. Chan: A Field-Programmable Prototyping Board: XC4000 BORG User Guide; UCSRC-CRL-94-18, Apr 1994
- [8] S. Casselman, J. Schewel, M. Thornburg: H.O.T. (Hardware Object Technology) Programming Tutorial; Release 1, Virtual Computer Corporation, January 1995
- [9] H. Chow, S. Casselman, H. Alunuweiri: Implementation of a Parallel VLSI Linear Convolution Architecture Using the EVC1; in [10]
- [10] N.N.: EVC-1 Info 1.1; Virtual Computer Corporation, 1994
- [11] A. Koch, U. Golze (Technical University Braunschweig, Germany): A Universal Co-Processor for Workstations; in: W. R. Moore, W. Luk (eds.): More FPGAs; Abbingdon EE&CS Books, Oxford, UK 1993 (selection from Proc. Int'l Symposium on field-programmable Logic and Applications, Oxford, UK, Sept. 1993)
- [12] J. Becker, R. Hartenstein, R. Kress, H. Reinig: High-Performance Computing Using a Reconfigurable Accelerator; Proc. Workshop on High Performance Computing, Montreal, Canada, July 1995
- [13] R. Hartenstein, J. Becker, R. Kress, H. Reinig: High-Performance Computing Using a Reconfigurable Accelerator; CPE Journal, Special Issue of Concurrency: Practice and Experience, John Wiley & Sons Ltd., 1996 (invited reprint from [12])



Notice: This document has been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a noncommercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

sion bits (figure 8). Without the classical “instruction fetch“ only one other mode of evoking an operation is available: “transport-triggered“ operations execution. The “instruction fetch“ has been migrated from run time to compile time.

We get a non von Neumann processor, which is not driven by control flow, but by data sequencing [69]. Such a data-procedural processor (d-Processor, which we also call Xputer¹) operates deterministically — in contrast to the co-called data flow machine [70] [71], where the order of execution is unpredictable because it is driven by arbitration (compare figure 9). Instead of program jumps and program loops a d-Processor carries out data (address) jumps and data (address) loops, i. e. jump destinations are data locations instead of instruction locations [59]. The d-paradigm is as universal as the von Neumann paradigm ([58] - [73]).

Figure 9. Comparing different computing paradigms.

nputer Neumann machine)	xputer (d-paradigm)	data flow machine
cedural sequencing ministic)		sequencing by arbiter (indeterministic)
ontrol-flow-driven	data-driven	

A d-processor may be used in stand-alone mode, if desirable. But it may be also used as a flexible, or even as a general purpose accelerator co-processor to a von Neumann host ([19] - [57]). Several typical Xputer mechanisms having been published elsewhere (for directory and papers see [74]) provide at low hardware cost an enormous acceleration potential for many algorithms, such as in DSP, image processing, multimedia applications and others. ([12] - [14]). Some of these acceleration mechanisms are parallelism at data path level, and the avoidance of several kinds of overhead, as well as by massive run-time-to-compile-time migration (r2c migration). Such an r2c migration seems to be a remedy of the fine granularity communication switching explosion known from the field of “traditional“ massively parallel computer systems. d-processors are a highly promising general purpose platform, which has been shown by a number of speedup factors obtained experimentally, in a special case by more than three orders of magnitude ([13], [33], [60], [72] - [76]).

A paradigm shift requires enormous mental efforts. Is it worth such efforts? The most important aspect is: data sequencers are universal, whereas instruction sequencers ALU-specific. Whenever you change the ALU of a von Neumann computer, you always need a new data sequencer. But even if you change the Xputer’s ALU dramatically, you do not need another data sequencer, even if you switch from a single rALU to a large rALU array ([53], [54]). This is a fundamentally important advantage for in using soft hardware, where you may change the ALU within milliseconds. The d-paradigm is the ideal backbone paradigm for using reconfigurable data paths and even configurable data path arrays, even when linked to a von Neumann host.

1. non von Neumann: please, do not mix up with transputers, which are von Neumann processors (X “trans“)



Notice: This document has been provided by the contributing authors as a means to ensure timely dissemination of scholarship and technical work on a noncommercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author’s copyright. These works may not be reposted without the explicit permission of the copyright holder.

At Kaiserslautern a compiler has been developed, which features 2-level hardware/software partitioning. It accepts source “programs“ written in a C dialect X-C (Xputer C). The first level partitioner separates sequential code for the (von Neumann) host from the tasks for the accelerator (Xputer). The second level partitioner generates sequential code for the data sequencer(s), as well as configuration code for the rDPA. The DPSS has been integrated as the back end of the rDPA code subcompiler. The compilation method and its implementation details have been published elsewhere ([19], [56] - [58]). X-C includes a sublanguage MoPL, a C extension to elegantly express generic data sequencing patterns (which we call scan patterns [58], [59]). More about scan patterns and MoPL has been published elsewhere ([60] - [62]).

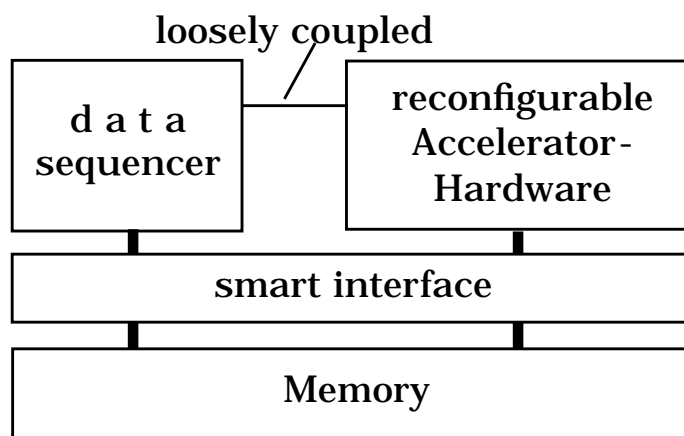
4. Life beyond von Neumann

From scenes dealing with HPC only the area of application-specific array processors (ASAPs: systolic arrays etc.) looks very well structured. I has a strong backbone paradigm, which looks like having mainstream capability. It combines data scheduling with locality of data, of operations, and, of resources. Time and space appear within the same formula. The von Neumann paradigm has disappeared. From its instruction sequencer only the clock has survived. Wavefront arrays do not even use clocking. Without turning away from the von Neumann paradigm such a well structured system of theories would never have been possible.

About Hardware/Software Co-Design Rajesh Gupta complains the lack of a general model [38], so that the scene is paralysed by the wide variety of architectures. The same complaint also holds for the scene of FPGA-based Custom-Computing Machines, a sister of the Hardware/Software Co-Design scene. A major difference between both disciplines is the implementation platform: reconfigurable versus hardwired. Another difference is the importance of compatibility: extensions to a commercially available general purpose machine versus optimized hardware/software trade-off. But both areas have one problem in common: the missing backbone paradigm.

Coupling a reconfigurable accelerator to a host resembles changes the hosts ALU hardware. That’s why CCM designers interface such accelerators through the host’s back door — to avoid the need for changing the instruction sequencer and thus destroying the host architecture. From a von Neumann point of view this means a misuse of memory addresses (figure 2): accelerator calls are “transport-triggered“ ([63] - [68]), and are not issued by “instruction fetch“,

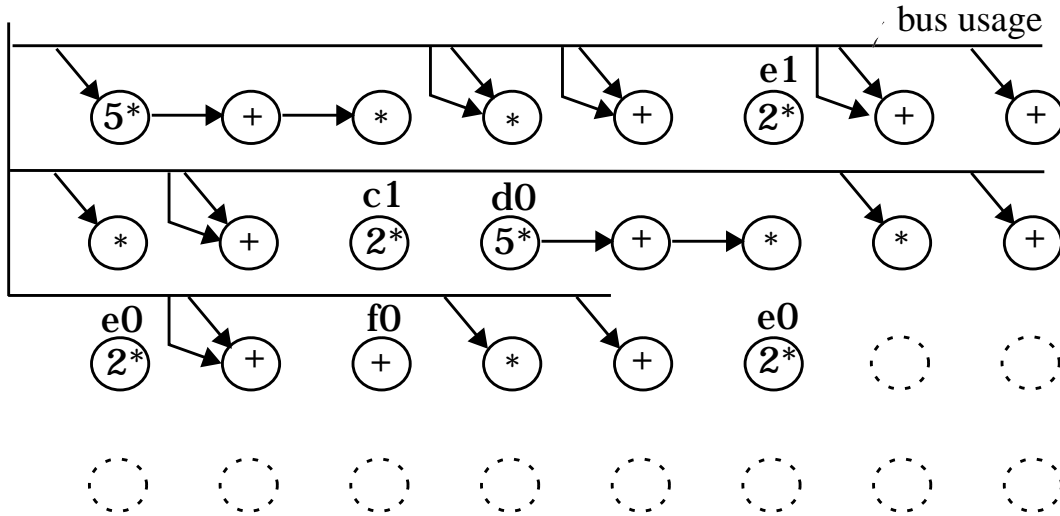
Figure 8. Xputer: procedurally data-driven computer.



A reconfigurable ALU or other accelerator can be reconfigured within milliseconds. How can be avoided, that the architecture falls apart already after partial ALU reconfiguration? A solution is the “Data Sequencer“ [60] [61]. It is only very loosely coupled to the data path resources, only by a few deci-

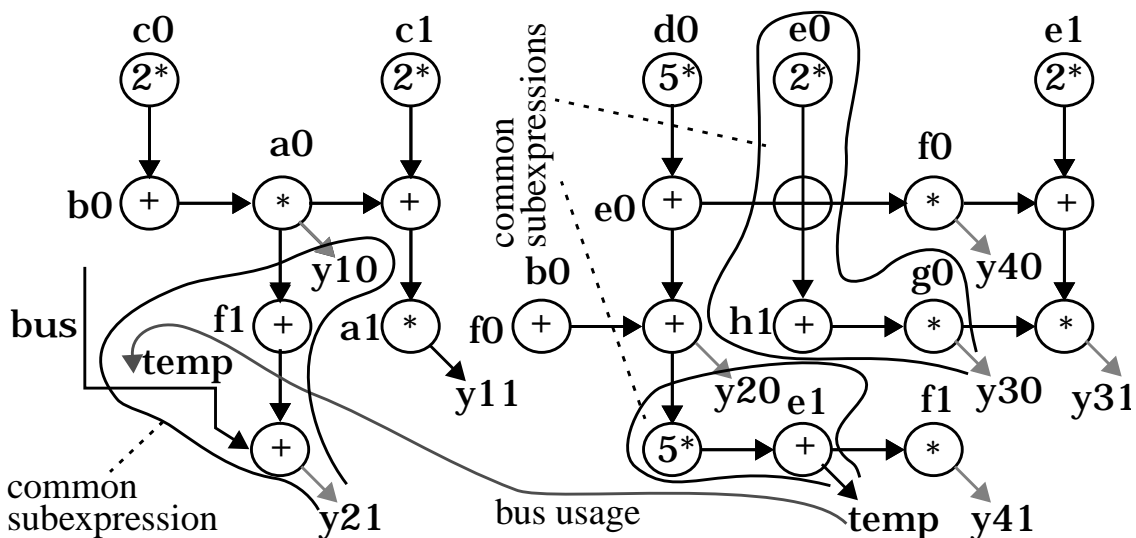


Figure 6. A first solution for the rDPA-Routing-and-Placement to the problem in figure 4



The 8 equations in figure 5 are used as a design problem demo. figure 6 shows a first brute force solution, which needs 17 bus cycles and makes a bottleneck out of the only available bus. After a few seconds a simulated annealing optimizer produces a much better solution (figure 7), where the bus is needed for a single cycle only. All other communication is restricted to nearest neighbour interconnect between rDPs. The possibility to use rDPs also for routing only, instead of using it as an operator, provides more flexibility to obtain better solutions from the optimization algorithm. The optimizer is part of the data path synthesis system (DPSS) developed by Rainer Kress, which also features smart data prefetching strategies ([53], [54])

Figure 7. DPSS-optimized solution of rDPA routing and placement to equations in figure 4.



(application-specific instruction set processors [25] - [28]), as FCCMs (FPGA-based custom computing machines: [29] - [36]), or by Hardware-Software-Co-Design ([37] - [49]).

The more radical ASIP approach creates completely application-specific instruction sets. Also a completely new compiler, operating system, simulator and emulator have to be generated for each new ASIP design alternative. In the scenes of FCCMs und H/S-Codesign, however, only a few application-specific extensions are patched to the (standardized) instruction set of a general purpose processor by adding external accelerator hardware (AH). The advantage is the use of commercially available software. But because of the very tight coupling between ALU and sequencer the von Neumann paradigm requires a new instruction sequencer as soon as the instruction set is modified. To avoid the need of a new instruction sequencer for each new design the AH is interfaced through the back door by memory address stealing (figure 2). That's why the results from FCCM or Hardware/Software Co-Design tend to be patchwork ([33], [50], [51])

Figure 5. An application problem example for rDPA structural programming .

$y_{10} := a_0 * (b_0 + 2 * c_0);$	$y_{11} := a_1 * (y_{10} + 2 * c_1);$
$y_{20} := 5 * d_0 + e_0 + (f_0 + b_0);$	$y_{21} := 5 * y_{20} + e_1 + (f_1 + y_{10});$
$y_{30} := g_0 * (h_0 + 2 * e_0);$	$y_{31} := y_{30} * (y_{40} + 2 * e_1);$
$y_{40} := (5 * d_0 + e_0) * f_0;$	$y_{41} := (5 * y_{20} + e_1) * f_1;$

3. From FPGA to Field-programmable Data Path Array (FPDPA)

Word level reconfigurable arrays [52] demonstrate, that structural programming does not necessarily lead to Sauerkraut structures. The simplicity of programming such arrays is shown by an example using an rDPA (reconfigurable Data Path Array) by Rainer Kress ([53], [54], [55]). figure 3 illustrates the rDPs (reconfigurable Data Paths) and figure 4 shows its structural programmability. An rDPA design with an array of 2 by 3 rDPs on a chip has been submitted for fabrication. Each rDP is 32 bits wide and is mainly a configurable ALU. But by a small microprogram it may carry out more complex operators, such as e. g. floating point. Interconnect between rDPs uses an asynchronous interface. To save chip area, only nearest neighbour interconnect between rDPs is possible. But an rDP may serve as a routing element only, which provides sufficient flexibility for routing and placement. The rDPA concept is fully scalable. Several rDPA chips may be interconnected in a transparent way. The rDPA is a first step from FPGA to FPDPA by using (word level) configurable ALU blocks (CABs) instead of the (bit level) CLBs (configurable logic blocks) used in platforms available commercially.

Currently a next generation rDPA chip concept is being developed, which also allows to map data sequencers on it. The goal of this is to minimize the number of designs needed for Xputer implementation. We have not yet decided, whether we would prefer two chip designs (one for sequencer and rDPA, and another one for the smart memory interface, compare figure 8), or a single one which might include a kind of "slice" of the smart memory interface per chip.



Figure 3. Example of an rDPA (reconfigurable Data Path Array)..

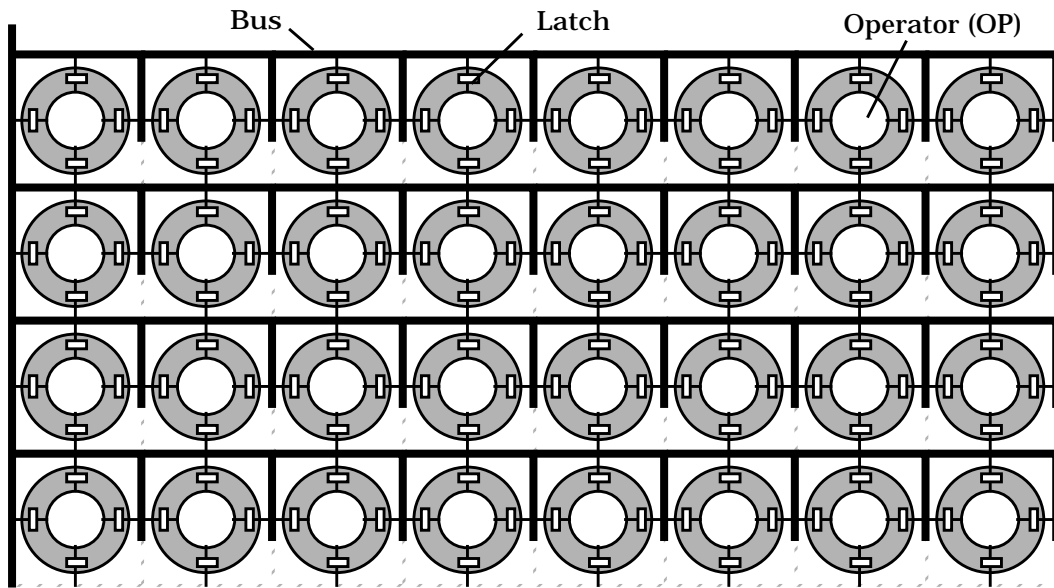
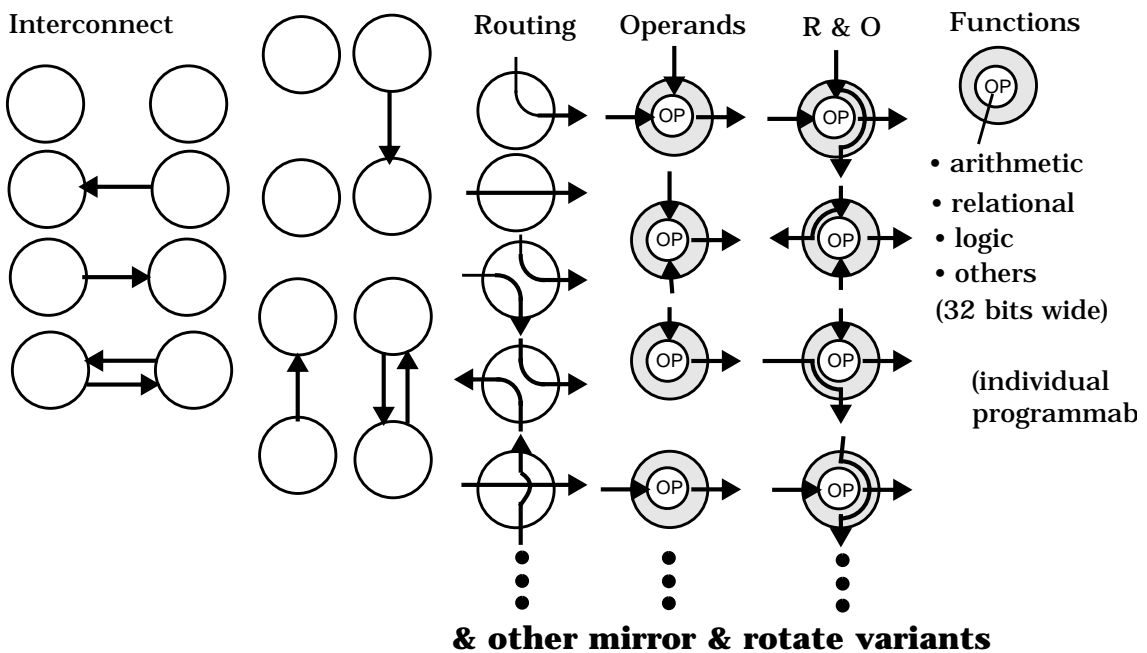


Figure 4. Some programmability synopses of the rDPA-II (also see figure 3).

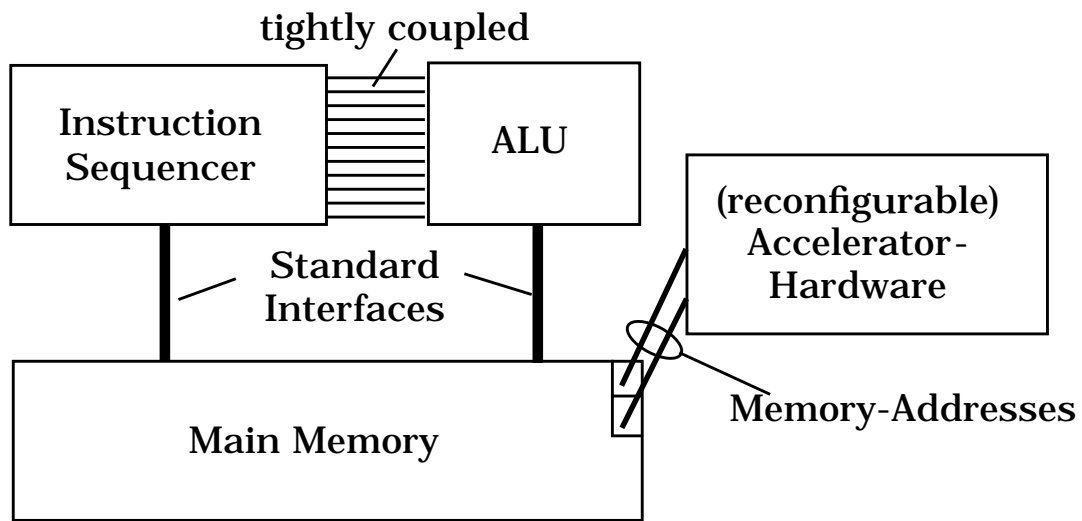


Computing has been dominated by the von Neumann paradigm for about 50 years. But VLSI design problem capture experienced a paradigm shift roughly every 6 years: from polygon pushing, over transistor circuit design, logic synthesis to RT level synthesis (high level synthesis and VHDL or Verilog usage. Now we are facing a shift to the processor level. The market for application-specific processors, microcontrollers or other subsystems is growing faster than that of general purpose processors. Application-specific microprocessors are pushed within several different R&D scenes ([23], [24]): as ASIPs



Notice: This document has been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a noncommercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Figure 2. F-CCMs as a tinker toy approach: accelerator hardware is linked through the back door. Due to the von Neumann paradigm a direct instruction set augmentation is prohibited..



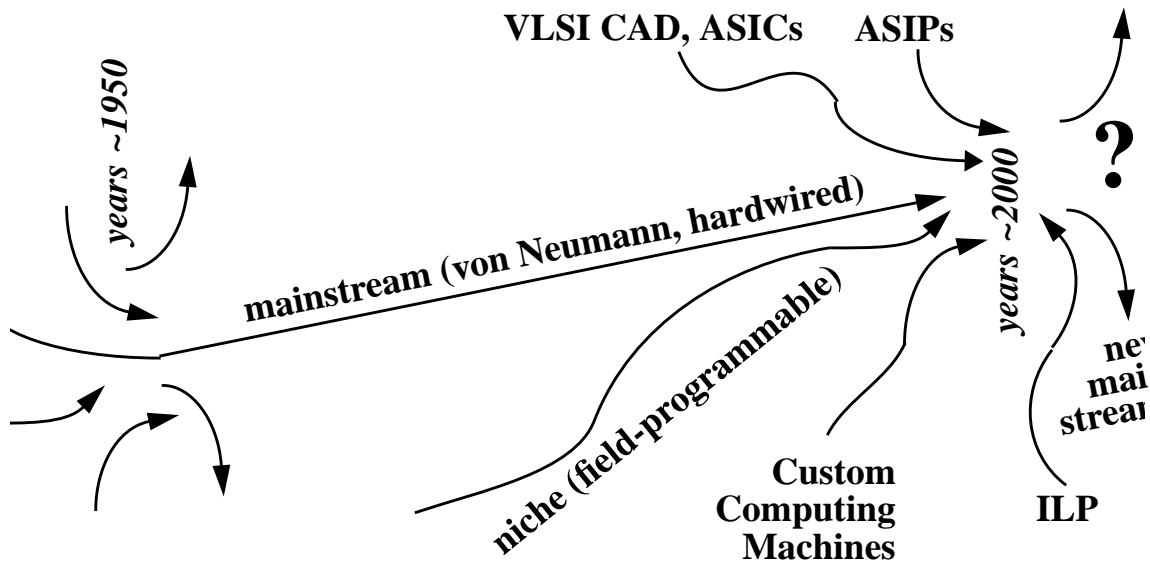
2. Introduction

Meanwhile field-programmable devices have reached a market volume of almost four Billion US Dollars (worldwide). Hardware has become soft and structural programming is gaining momentum towards traditional procedural programming ([2] - [5]). Also field-programmable boards are commercialized (e. g. [6] - [10]). An important application are embedded reconfigurable accelerators (for example [11] - [14]). Reconfigurable architectures (e. g. also [15] - [18]) start to compete with structurally programmable processors. Why did all the MPC and HPC scenes ignore this platform of efficient parallelism all the time?

Why are "Real Time Systems", "Multimedia Hardware", "DSP" und "Custom Computing Machines" strongly disjoint scenes? They all have the same goal: high performance for a reasonable price. Also a new IEEE Technical Committee called "Engineering of Computer-Based Systems (ECBS)", tries to tackle the chaos [19]. But what is missing is an interdisciplinary model as powerful as von Neumann, which helps to link these still isolated scenes. Also the TC-ECBS has no solution, not yet. Von Neumann does not help, since it is not a communication paradigm. And it is not suitable for using reconfigurable hardware, which has been recently discovered by HPC people. At the High Performance Computing Symposium in New Delhi by end '1995 field-programmable logic has been officially introduced to the scene by the opening keynote of a speaker from MIT [20]. He said: "computing by the yard" replaces "computing in time". This is a mile stone insofar, as this scene had completely ignored this technology platform for decades. But probably it will take decades to a marriage between computing in time and computing in space. Systolic arrays ([21], [22]) look more like an illegitimate child, the result of an early episode



Figure 1. History of turbulences in Computing, Parallel and/or High Performance Computing.



Some of these MPC people now have discovered the system on a chip (SOC). A beautifully architected network of processors on a single chip is now their dream. So they are ready to repeat all the mistakes from the last 30 years on a new technology platform. But more and more controversial panels are questioning the future of MPC and HPC. In turbulent times crowds gather in the market square, ready to follow new leaders promising a way out of the dead road. Within a few years communication channels featuring tens or hundreds of Gigabytes per second will be a commodity at all levels: LAN, WAN, and even world-wide (also see [1]). This will cause a tendency away from the heavy metal of the MPC or super computing scenes toward distributed HPC. Load balancing for massively parallel computing will become the job of networking people: from linking workstation clusters to utilizing idling computing power from the internet. The authors of this paper strongly believe, that the only way to implement HPC concentrated on a single board or a single chip will be the way how embedded system designers do it — more pragmatic and less hypnotized by the von Neumann paradigm. We believe, that a new paradigm would be very helpful. This paper tries to convince the reader



Notice: This document has been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a noncommercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

A General Approach in System Design Integrating Reconfigurable Accelerators

R. W. Hartenstein, J. Becker, M. Herz, U. Nageldinger
Universitaet Kaiserslautern
Postfach 3049, D-67653 Kaiserslautern, Germany
Fax: +49 631 205 2640, e-mail: hartenst@rhrk.uni-kl.de
URL: <http://xputers.informatik.uni-kl.de/>

Key Words: Xputer, accelerator, application-specific processor, digital system, ECBS, embedded system, emerging technologies, field-programmable, FPGA, hardware/software co-design, ILP, instruction level parallelism, microprocessor, non von Neumann, parallelism, partitioning, rapid prototyping

Abstract

This paper introduces a fundamentally new machine paradigm, which takes into account, that hardware has become soft. Along with a compilation technique, this paper introduces its implementation and application by illustrating the rDPA (reconfigurable data path array), an integrated circuit having been designed at Kaiserslautern and submitted for fabrication. This is the first implementation of a field-programmable data path array (FPDPA), which uses configurable ALU blocks (CABs) instead of CLBs (configurable logic blocks). This novel platform provides highly efficient means of instruction level parallelism: for many applications by orders of magnitude more efficient than the traditional parallelism used in all scenes of high performance computing.

1. Preface

This paper deals with high performance computing for the many. Traditionally we have two completely different worlds: embedded systems as well as massively parallel computing (MPC), supercomputing, high performance computing (HPC), or, whatever it may be called. This means: application-specific versus general purpose hardware. In the field of embedded systems application-specific hardware extensions are used for acceleration. The goal of the other world has been a general purpose hardware platform for MPC or HPC. But these scenes are now in a heavy crisis. For decades the mainstream of the crowd has followed the von Neumann paradigm bandwagon (figure 1). The market is shrinking, but ivory tower activities are exploding. There are more than a hundred conferences per year, worldwide.



Notice: This document has been provided by the contributing authors as a means to ensure timely dissemination of scholarship and technical work on a noncommercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.