Report

The current status of reconfigurable computing

Author(s):
Enzler, Rolf

Publication Date:
1999

Permanent Link:

https://doi.org/10.3929/ethz-a-004704215 →

The Current Status of Reconfigurable Computing

Rolf Enzler
Swiss Federal Institute of Technology
Electronics Laboratory
CH-8092 Zürich
enzler@ife.ee.ethz.ch

Technical Report, July 1999

Abstract

With reconfigurable computing a novel computing paradigm has emerged in the last few years that promises to be a valuable alternative to the traditional approaches like processors and ASICs. This report discusses the novel computing approach of reconfigurable computing and outlines the state of affairs in research and industry. The various approaches and directions are classified and discussed. The major challenges and potentials of reconfigurable computing are identified and future conceivable scenarios considered. In order to provide a representative overview of the activities in this young research field, a detailed list of projects and products is appended.

1 Introduction

In the last decade, a lot of effort in research and development has been dedicated to computer and processor architecture. One of the fundamental trade-offs in the design of computing systems and devices involves the balance between flexibility and performance.

At one end of the spectrum are processors such as general-purpose processors (GPPs) or digital signal processors (DSPs), which have an instruction-set architecture. They provide the possibility of processing arbitrary computations due to their architectural concept. In consequence of the overhead paid for the flexibility, processors are rather inefficient regarding performance and power consumption.

At the other end of the spectrum are application-specific integrated circuits (ASICs), which contain dedicated circuits specialized to a particular set of tasks. Therefore, the architecture is optimally suited for the tasks at hand which is why ASICs are efficient regarding performance and power consumption, but they lack flexibility, as no programmable resources are provided.

With reconfigurable computing (RC), an alternative has emerged in the last ten years that is located between these traditional approaches. Reconfigurable systems[1] are implemented with programmable logic, which allows to alter the hardware circuits. Therefore, reconfigurable computing combines features of

[1] often referred to as Custom Computing Machines (CCMs)

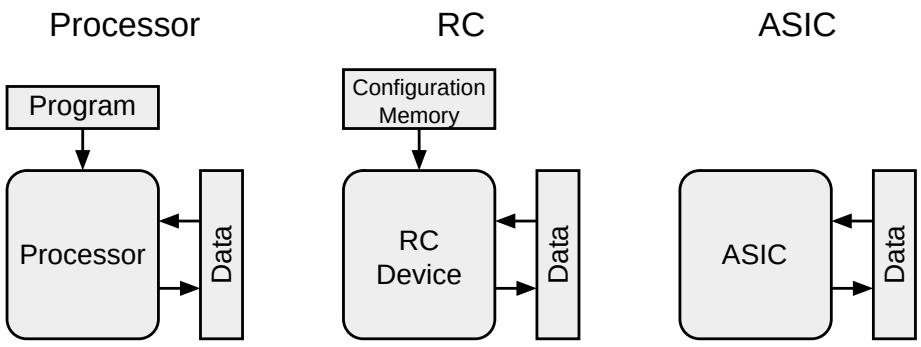

Figure 1: Reconfigurable computing is located between the processor and the ASIC concept.

processor and ASIC approaches (figure 1). The realization of reconfigurable systems was enabled through the introduction of the field-programmable gate array (FPGA) in the mid eighties. FPGAs share with ASICs the capability to implement application-specific circuits, with the key difference that FPGA circuits are programmed by means of a configuration datastream that specifies the logical functionality and connectivity.

This report intends to provide an overview of the field of reconfigurable computing. Section 2 elucidates the reconfigurable computing paradigm and discusses its novelties. Section 3 outlines the composition of reconfigurable systems and their peculiarities. Sections 4 and 5 explore the current status of research and industry, respectively. Section 6 summarizes the insights and reflects about the future of reconfigurable computing. Finally, Appendix A lists research projects and commercial products mentioned in Sections 4 and 5.

2 Reconfigurable Computing Paradigm

Processors have a general, fixed architecture that allows to implement tasks by *temporally* composing atomic operations provided e. g. by the ALU or the floating-point unit. In contrast, ASICs implement tasks by *spatially* composing operations provided by dedicated functional units like adders or multipliers (figure 2). Reconfigurable computing combines both approaches. Reconfigurable systems provide programmable logic, which allows to implement tasks both in a spatial manner similar to ASICs and in a temporal manner comparable to processors. A new computing paradigm emerges that enables several computation approaches:

Multi-mode hardware Several different algorithms are executed concurrently or sequentially on the same reconfigurable hardware.

Temporal partitioning An algorithm is partitioned into several sections, each implemented by an individual circuit and executed sequentially.

Co-processor Execution is sped up by implementing critical parts of an algorithm in reconfigurable hardware.

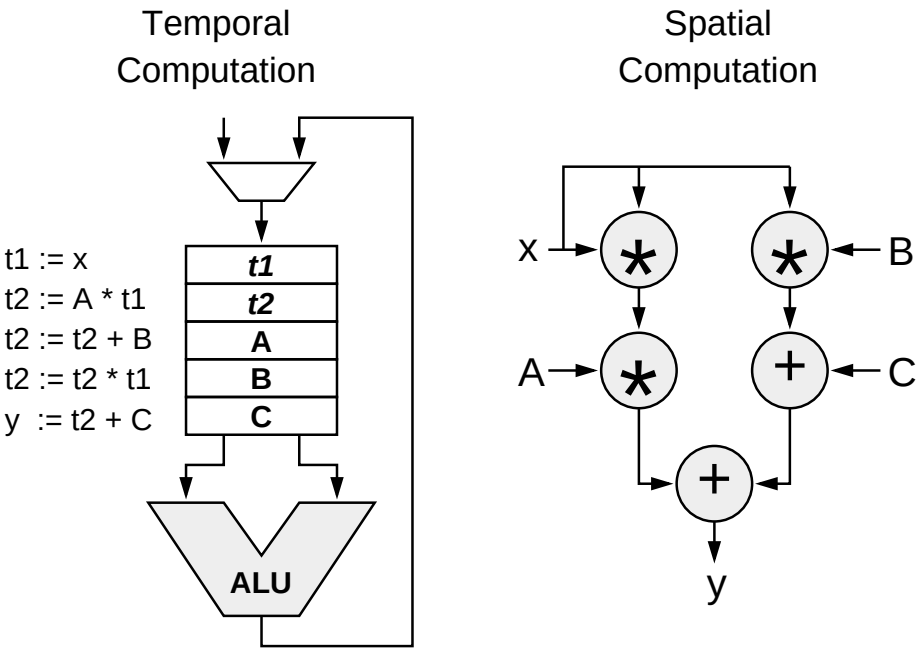

Figure 2: Temporal vs. spatial computation of the expression $y = Ax^2 + Bx + C$

Hardware-on-demand Functions are switched on command, i. e. at arbitrary, not predefined points in time.

Dynamic adaptation The algorithm implementation is adapted at run-time depending on the incoming data. Different scenarios are conceivable, from only adapting constants to dynamically generating whole circuits. Examples are neural networks, adaptive filters and constant propagation [117].

3 Reconfigurable Systems

To apply the reconfigurable computing paradigm, systems are needed that provide the required functionality. Reconfigurable platforms consist on one hand of a reconfigurable computing engine that provides the required programmable logic, of a configuration management mechanism that controls the execution of the tasks on the available reconfigurable resources and of a software environment (figure 3). The complexity of the configuration management mechanism and the form of its implementation can widely vary depending on the application and the computation approach, respectively. On the other hand, the development system is a vital part of a reconfigurable platform. Crucial elements are a well defined design flow that is adapted for reconfigurable computing, compilers that allow easy and fast compilation of high-level software descriptions into hardware circuits, and CAD tools that support RC mechanisms like e. g. run-time reconfiguration.

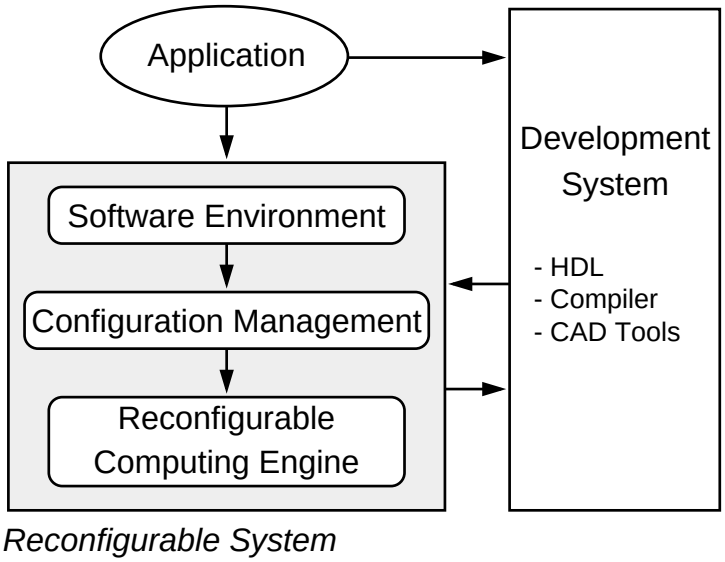

Figure 3: Composition of a reconfigurable system and its environment

As the discussion in the previous section shows, the reconfigurable computing paradigm leaves many degrees of freedom. However, some peculiarities can be identified that characterize the nature of reconfigurable systems and allow a classification.

System composition Due to the wide spectrum of reconfigurable computing many different architectures are imaginable. Reconfigurable systems are composed of programmable logic and eventually of processor or specialized custom components. Accordingly, both pure FPGA systems and hybrid systems are conceivable approaches.

Run-time reconfiguration (RTR) RTR, i. e. the capability of reprogramming the hardware circuits at run-time, is a major and vital feature of reconfigurable systems [61]. Some concepts can only be realized if the reconfiguration time is very short. For that reason, the overhead introduced by RTR is a crucial factor for the functionality and the performance of reconfigurable systems.

Configuration granularity The granularity is defined by the number of partitions (configurations) an algorithm consists of and their size, respectively. Therefore, the granularity depends strongly on the amount of resources that a reconfigurable system provides.

Configuration scheduling Basically two approaches are conceivable, namely static scheduling and run-time scheduling. Responsible for the scheduling strategy is the configuration management (figure 3).

4 Research

The first reconfigurable systems that reported remarkable performance were Splash [10, 49] and PeRLe-1 [111] in the early nineties, both multi-FPGA systems attached to a host computer. They proved to achieve higher performance than any other architecture for applications requiring highly parallel, bit-level operations. The Splash 2 system e. g. outperformed any contemporary supercomputer implementation of genetic string matching by several orders of magnitude [60]. Other application implementations achieved similar results [92, 109, 111]. In the last years, various single-FPGA and multi-FPGA systems succeeded (see Appendix A).

However, for many applications current reconfigurable devices, nowadays mainly FPGAs, and hence pure FPGA systems are ill-suited. For that reason, research has tended towards developing appropriate FPGA architectures, novel configurable devices and systems and tools supporting them.

The major research directions are outlined here. A structured overview containing the according references is provided in Appendix A.

FPGA architecture In order to better support the needs of reconfigurable computing, many FPGA architectures have been proposed with novel compositions of the complex logic blocks (CLBs) and the interconnection structure.

Since reconfiguration time is a crucial feature of reconfigurable systems, novel configuration schemes have emerged to speed-up reconfiguration on both architectural and conceptual level. Examples are partial RTR, compressed RTR [57, 58], configuration prefetching [54], Wormhole RTR [17], Incremental Pipeline RTR [98], Autonomous RTR [83] and optical RTR [108].

Even a step further go approaches of context-switching FPGAs, where several FPGA configurations ("contexts") are stored on-chip, which allows to switch quickly between the configurations [34, 97, 104]. This enables virtual hardware concepts analogous to virtual memory concepts.

Another tendency is to get away from bit-oriented architectures towards word-oriented architectures (8-, 16-bit) because many applications cannot be handled efficiently by bit-oriented computation.

Other issues investigated are high-speed and asynchronous FPGA architectures.

Hybrid devices and engines Since it has become clear that pure FPGA solutions are ill-suited for many kinds of applications, many research activities target hybrid devices and engines. Hybrids are composed of reconfigurable logic as well as of processor or specialized custom components. Systems emerged on board level, where discrete components are coupled on a PCB, and on device level, where the components are integrated on the same die. The stronger coupling on a single die intends to overcome the emerging communication bottleneck between the system components. The feature of reconfigurability can be applied in various ways. Many approaches merge a processor with reconfigurable resources in order to augment the executional capabilities. Similar systems have emerged that combine reconfigurable logic with functional units (FUs), or that mix a variety of different FPGAs. They all have the idea in common to execute

every operation on the most appropriate resource.
Other concepts have been presented that make use of reconfigurability for the interconnection network between the components. These approaches benefit from research in related fields like reconfigurable meshes [20, 19] or PMMLAs [47]. What has emerged as well are frameworks that finally intend to manufacture an ASIC rather than serving as a target platform themselves.

Fully reconfigurable processors Processors have been fully implemented in reconfigurable logic. Since many structures cannot be mapped efficiently onto reconfigurable logic, these approaches have to be seen as attempts to study the schemes of reconfigurable computing rather than to propose a target platform.

Configuration management In order to exploit the capabilities provided by RC engines, a run-time configuration management is essential, which controls the reconfiguration procedure and the available resources. Various concepts have been proposed.

Tools One of the most complex challenges in the reconfigurable computing area are the software tools. Two peculiarities emerge that are not found in the standard ASIC design: run-time reconfiguration, and timing restrictions for the compilation process as it is not acceptable for many applications to wait for hours until a new circuit is generated.
Fast compilers are needed that manage issues like automatic partitioning, circuit synthesis, and mapping of circuits onto reconfigurable devices. The goal is the easy and fast compilation of high-level software descriptions into software and/or hardware circuits that run on an RC engine. For that reason, a variety of new hardware description languages (HDLs) and corresponding compilers have been introduced.
On the other hand, CAD tools are needed that support the development of reconfigurable computing applications in the design stage [74].

Benchmarks In order to objectively compare reconfigurable systems among themselves and especially versus traditional approaches, benchmarks are needed that express the suitability of the candidates. The challenge is to find a fair comparison due to the heterogeneous nature and the complexity of the problem which is all but trivial. However, some benchmark suites for RC purposes have been proposed [12, 68].

Applications A multitude of applications has been realized which is why it is impossible to give a comprehensive overview. The most applications have been reported in the fields of signal processing (e. g. wireless communication, transformations, neural networks, adaptive filters), image processing (e. g. computer vision, target recognition), cryptography, DNA sequence comparison, data compression, boolean satisfiability problems and search problems.

5 Industry

Up to now, hardly any system has proven attractive or competitive enough to establish a commercial presence, not even in the most promising areas like signal or image processing, where the suitability of reconfigurable systems has been proven in the research labs.

Actually, to the knowledge of the author, there exists only one commercial system that really deserves to be called a reconfigurable system: Compugen's BioXL and its predecessor the Bioccelerator. Both are multi-FPGA systems attached to a host processor and are used as hardware accelerators for searching genomic and protein databases. They are typical niche products.

However, only few companies show interest in reconfigurable computing. Products that are commercially available now are some rather modest RC platforms without a particular target application. On the market available are single-FPGA, multi-FPGA and some hybrid boards (see Appendix A). Some of them are the commercialized counterparts of successful research projects.

Regarding RC devices, the situation is even worse. Besides the established mainstream FPGA devices, very few products are on the market. The most promising ones are Triscend's E5 embedded processor, which implements the I/O functionality in reconfigurable logic, and Philips' R.E.A.L, although this is more a framework with an ASIC implementation as target. Some other promising products have been withdrawn from the market like Motorola's Core+, which merged a processor core with an FPGA on the same die, and Xilinx' XC6200 FPGA family, which provided some nice features like partial reconfiguration and a sophisticated reconfiguration interface. Actually, neither the established FPGA nor microprocessor manufacturers show a profound interest in the field.

Trying to understand the reasons for this situation, some fundamental problems can be identified. A major concern of the industry is that reconfigurable computing has no clearly defined borders to the alternatives like processors or ASICs. The situation is even worse, because the borders are not only blurred but also continuously changing. With every new technology step or processor generation, the lucrative area for reconfigurable computing changes. Consequently, companies that intend to become active in the reconfigurable computing area face the difficulty of positioning their products in the market. Since processor and ASIC solutions share the entire market, there is no distinct market for reconfigurable computing. Furthermore, there exist no CAD tools that support any kind of reconfigurable computing mechanisms so that the implementation of an application entails a big effort. Consequently, the high risk keeps companies from entering the field.

As a positive fact, reconfigurable computing allows product differentiation, according to [105] one of the driving forces of embedded system implementations beside time-to-market, performance and costs. In this aspect, RC solutions have a clear advantage over most of the alternatives what could help to establish reconfigurable systems in the marketplace.

6 Conclusions and Outlook

A basic problem reconfigurable computing faces is that it will never provide the best solution in the design space regarding a single criterion. An RC solution will always be a trade-off and has therefore to represent an optimum between different criteria. In general, an ASIC solution is better regarding performance, density, power consumption and costs (in high volumes), while processors are more flexible. A standard estimate says that custom logic is one order of magnitude faster and up to two orders of magnitude smaller than an equivalent FPGA solution [4, 55].

Therefore, the competitive benefit is only real for applications where the RC trade-off represents a valuable optimum, or where an ASIC or processor implementation is just not feasible, e. g. due to the limited ASIC size or insufficient processor performance.

Multi-FPGA systems have proved to be suitable for deeply pipelined, highly parallel, bit-oriented computations. However, for less regular computations current systems are ill-suited. Furthermore, reconfigurable computing today is mainly based on commercially available FPGAs that are insufficient for the demands of this domain. The research community tries to overcome these limitations, as the lively activities show. Investigation on novel concepts is in progress in many areas of reconfigurable computing such as device and system architecture, compilers, CAD tools, applications and benchmarking.

As in any other new computing technology a high effort is necessary in the early development stages where systems and tools lag behind the needs. Due to the complexity and the variety of challenges that have to be addressed, it is likely that reconfigurable computing will have to spend more time in the development stage than some other technologies.

For the improvement of processors and ASICs in the recent years especially advances in process technology and computer architecture have been responsible. While process technology will still serve as an innovative force during the next years, this is not so clear for computer architecture. As the latest processor generations show, novel architectural features are complex and implicate an enormous effort (e. g. wide datapaths, deep pipelines, branch prediction, out-of-order execution, superscalar execution, caching mechanisms, functional units).

Reconfigurable computing can profit from these tendencies. Since architectural novelties become more and more complex and technology will allow to integrate more functions onto the same die, reconfigurable concepts become more feasible and attractive. Consequently, a close coupling between processor and reconfigurable components can be achieved, which is crucial to overcome the communication bottleneck.

From the commercial point of view, reconfigurable computing is currently a niche solution. The promising achievements in research have not yet turned into commercial products. Only some rather small companies have committed themselves to reconfigurable computing, in contrast to the established FPGA vendors (Xilinx, Altera, etc.), which show no interest at the moment. Therefore, the RC community must find a way to make their desired features attractive

for the FPGA vendors to integrate them into their mainstream products.

In order to gain more commercial impact, the RC community has to define its competition area and therein to point out the benefits over processors on one hand and over ASICs on the other hand. For that reason, cheap and accessible applications are needed that demonstrate the power of reconfigurable computing, e. g. DES encryption/decryption for e-mail or data compression/decompression for Internet applications. Furthermore, meaningful benchmarks would help to quantify reconfigurable systems and enable a comparison with processors and ASICs. Another major key for RC to enter mainstream will be to adapt its development process to the traditional development process. The design flow and the computation models have to be brought into a practical context, in order to be competitive with the traditional processor and ASIC approaches.

All the topics outlined above suggest that reconfigurable computing will need some time to enter the commercial marketplace. The first commercial systems are expected to emerge in the field of image and signal processing. In a more mainstream manner, embedded systems will probably first adopt some features of reconfigurable computing. Until reconfigurable components will be integrated into processors and RC will have a serious impact on general-purpose processing in a real mainstream environment, much more time will pass. But the vision is still alive.

References

[1] A. Abnous and J. Rabaey. Ultra-low-power domain-specific multimedia processors. In *VLSI Signal Processing, IX*, pages 461–470. IEEE Signal Processing Society, 1996.

[2] A. Abnous, K. Seno, Y. Ichikawa, M. Wan, and J. Rabaey. Evaluation of a low-power reconfigurable DSP architecture. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 55–60. Springer, 1998.

[3] A. A. Aggarwal and D. M. Lewis. Routing architectures for hierarchical field programmable gate arrays. In *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors, ICCD'94*, pages 475–478, 1994.

[4] O. T. Albaharna, P. Y. K. Cheung, and T. J. Clarke. On the viability of FPGA-based integrated coprocessors. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'96*, pages 206–215, 1996.

[5] Alex Computer Systems. APAC509. http://www.alexusa.com/pg13.htm.

[6] Altera. ARC-PCI. http://www.altera.com/html/programs/phd.html.

[7] R. Amerson, R. Carter, W. Culbertson, P. Kuekes, G. Snider, and L. Albertson. Plasma: An FPGA for million gate systems. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'96*, pages 10–16, 1996.

[8] R. Amerson, R. J. Carter, W. B. Culbertson, P. Kuekes, and G. Snider. Teramac – configurable custom computing. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'95*, pages 32–38, 1995.

[9] Annapolis Micro Systems. WILDFIRE. http://www.annapmicro.com/wfhtm.html.

[10] J. M. Arnold, D. A. Buell, and E. G. Davis. Splash 2. In *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures, SPAA'92*, pages 316–322, 1992.

[11] P. M. Athanas and H. F. Silverman. Processor reconfiguration through instruction-set metamorphosis. *IEEE Computer*, 26(3):11–18, March 1993.

[12] J. Babb, M. Frank, V. Lee, E. Waingold, R. Barua, M. Taylor, J. Kim, S. Devabhaktuni, and A. Agarwal. The RAW benchmark suite: Computation structures for general purpose computing. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 134–143, 1997.

[13] P. J. Bakkes and J. J. du Plessis. Mixing fixed and reconfigurable logic for array processing. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'96*, pages 118–125, 1996.

[14] P. Bellows and B. Hutchings. JHDL - an HDL for reconfigurable systems. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 175–184, 1998.

[15] V. Betz and J. Rose. Using architectural families to increase FPGA speed and density. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'95*, pages 10–16, 1995.

[16] J.-L. Beuchat, J.-O. Haenni, and E. Sanchez. Hardware reconfigurable neural networks. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 91–98. Springer, 1998.

[17] R. Bittner and P. Athanas. Wormhole run-time reconfiguration. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'97*, pages 79–85, 1997.

[18] R. A. Bittner, Jr., P. M. Athanas, and M. D. Musgrove. Colt: An experiment in wormhole run-time reconfiguration. In *High-Speed Computing, Digital Signal Processing, and Filtering Using Reconfigurable Logic. Proceedings of SPIE*, volume 2914, pages 180–186. SPIE, 1996.

[19] V. V. Bokka, H. Gurla, S. Olariu, and J. L. Schwing. Constant-time algorithms for constrained triangulations on reconfigurable meshes. *IEEE Transactions on Parallel and Distributed Systems*, 9(11):1057–1072, November 1998.

[20] K. Bondalapati and V. K. Prasanna. Reconfigurable meshes: Theory and practice. In *Proceedings of the Reconfigurable Architectures Workshop, RAW'97*, 1997.

[21] K. Bondalapati and V. K. Prasanna. DRIVE: An interpretive simulation and visualization environment for dynamically reconfigurable architectures. In *Field-Programmable Logic and Applications, Proceedings of FPL'99*, volume 1673 of *Lecture Notes in Computer Science*. Springer, 1999.

[22] G. Borriello, C. Ebeling, S. Hauck, and S. Burns. The Triptych FPGA architecture. *IEEE Transactions on VLSI Systems*, 3(4):491–501, December 1995.

[23] J. Burns, A. Donlin, J. Hogg, S. Singh, and M. de Wit. A dynamic reconfiguration run-time system. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 66–75, 1997.

[24] S. Cadambi, J. Weener, S. C. Goldstein, H. Schmit, and D. E. Thomas. Managing pipeline-reconfigurable FPGAs. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'98*, pages 55–64, 1998.

[25] H. C. Card, G. K. Rosendahl, D. K. McNeill, and R. D. McLeod. Competitive learning algorithms and neurocomputer architecture. *IEEE Transactions on Computers*, 47(8):847–858, August 1998.

[26] Compaq Systems Research Center. PCI Pamette. http://www.research.digital.com/SRC/pamette/.

[27] V. C. Chan and D. M. Lewis. Area-speed tradeoffs for hierarchical field-programmable gate arrays. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'96*, pages 51–57, 1996.

[28] Compugen. Bioccelerator. http://cgen.com/.

[29] Compugen. BioXL. http://cgen.com/.

[30] S. A. Cuccaro and C. F. Reese. The CM-2X: A hybrid CM-2/Xilinx prototype. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'93*, pages 121–130, 1993.

[31] D. Davis, M. Barr, T. Bennett, S. Edwards, J. Harris, I. Miller, and C. Schanck. A Java development and runtime environment for reconfigurable computing. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 43–48. Springer, 1998.

[32] D. Davis and J. Harris. ACEcard: A high-performance architecture for run-time reconfiguration. In *Proceedings of the First Merged International Parallel Processing Symposium & Symposium on Parallel and Distributed Processing, IPPS/SPDP'98*, pages 616–619, 1998.

[33] A. DeHon. DPGA-coupled microprocessors: Commodity ICs for the early 21st century. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'94*, pages 31–39, 1994.

[34] A. DeHon. DPGA utilization and application. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'96*, pages 115–121, 1996.

[35] D. E. Van den Bout. The Anyboard: Programming and enhancements. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'93*, pages 68–77, 1993.

[36] T. H. Drayer, W. E. King IV, J. G. Tront, and R. W. Conners. MORRPH: A modular and reprogrammable real-time processing hardware. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'95*, pages 11–19, 1995.

[37] C. Ebeling, D. C. Cronquist, and P. Franklin. Configurable computing: The catalyst for high-performance architectures. In *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures and Processors, ASAP'97*, pages 364–372, 1997.

[38] M. Eisenring, M. Platzner, and L. Thiele. Communication synthesis for reconfigurable embedded systems. In *Field-Programmable Logic and Applications, FPL'99*, volume 1673 of *Lecture Notes in Computer Science*. Springer, 1999.

[39] Embedded Solutions. Handel-C. http://www.embedded-solutions.ltd.uk/ProdApp/index.htm.

[40] Embedded Solutions. RC1000-PP. http://www.embedded-solutions.ltd.uk/ProdApp/index.htm.

[41] W. Fornaciari and V. Piuri. Virtual FPGAs: Some steps behind the physical barrier. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 7–12. Springer, 1998.

[42] P. C. French and R. W. Taylor. A self-reconfiguring processor. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'93*, pages 50–59, 1993.

[43] B. Fross, D. M. Hawver, and J. P. Peterson. WILDFIRE heterogeneous adaptive parallel processing system. In *Proceedings of the First Merged International Parallel Processing Symposium & Symposium on Parallel and Distributed Processing, IPPS/SPDP'98*, pages 611–615, 1998.

[44] D. Galloway. The Transmogrifier C hardware description language and compiler for FPGAs. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'95*, pages 136–144, 1995.

[45] S. Gehring and S. Ludwig. The Trianus system and its application to custom computing. In *Field-Programmable Logic: Smart Applications, New Paradigms and Compilers, FPL'96*, volume 1142 of *Lecture Notes in Computer Science*, pages 176–184. Springer, 1996.

[46] S. Gehring, S. Ludwig, and N. Wirth. A laboratory for a digital design course using FPGAs. In *Field-Programmable Logic: Architectures, Synthesis and Applications, FPL'94*, volume 849 of *Lecture Notes in Computer Science*, pages 385–396. Springer, 1994.

[47] G. Ghare and S.-Y. Lee. Dynamic reconfiguration of a PMMLA for high-throughput applications. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 1998.

[48] Giga Ops. Spectrum. http://www.gigaops.com/rcprodov.htm.

[49] M. Gokhale, W. Holmes, A. Kopser, S. Lucas, R. Minnich, D. Sweely, and D. Lopresti. Building and using a highly parallel programmable logic array. *IEEE Computer*, 24(1):81–89, January 1991.

[50] S. Goldstein, H. Schmit, M. Budiu, S. Cadambi, and M. Moe. PipeRench: A co-processor for the future. In *Proceedings of the International Symposium on Computer Architecture, ISCA'99*, 1999.

[51] S. Guccione. List of FPGA-based computing machines. `http://www.io.com/~guccione/HW_list.html`.

[52] J.-O. Haenni, J.-L. Beuchat, and E. Sanchez. RENCO: A reconfigurable network computer. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 288–289, 1998.

[53] R. W. Hartenstein, M. Herz, T. Hoffmann, and U. Nageldinger. On reconfigurable co-processing units. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 67–72. Springer, 1998.

[54] S. Hauck. Configuration prefetch for single context reconfigurable coprocessors. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'98*, pages 65–74, 1998.

[55] S. Hauck. The future of reconfigurable systems. In *Canadian Conference on Field Programmable Devices, FPD'98*, 1998.

[56] S. Hauck, T. W. Fry, M. M. Hosler, and J. P. Kao. The chimaera reconfigurable functional unit. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 87–96, 1997.

[57] S. Hauck, Z. Li, and E. J. Schwabe. Configuration compression for the Xilinx XC6200 FPGA. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 138–146, 1998.

[58] S. Hauck and W. D. Wilson. Runlength compression techniques for FPGA configurations. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'99*, 1999.

[59] J. R. Hauser and J. Wawrzynek. Garp: A MIPS processor with a reconfigurable coprocessor. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 24–33, 1997.

[60] D. T. Hoang. Searching genetic databases on Splash 2. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'93*, pages 185–191, 1993.

[61] B. L. Hutchings and M. J. Wirthlin. Implementation approaches for reconfigurable logic applications. In *Field-Programmable Logic and Applications, FPL'95*, volume 975 of *Lecture Notes in Computer Science*, pages 419–428. Springer, 1995.

[62] C. Iseli and E. Sanchez. Spyder: A reconfigurable VLIW processor using FPGAs. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'93*, pages 17–24, 1993.

[63] C. Iseli and E. Sanchez. Spyder: A SURE (SUper and REconfigurable) processor. *The Journal of Supercomputing*, 9(3):231–252, 1995.

[64] J. Jean, K. Tomko, V. Yavagal, R. Cook, and J. Shah. Dynamic reconfiguration to support concurrent applications. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 302–303, 1998.

[65] B. Kastrup. Automatic hardware synthesis for a hybrid reconfigurable CPU featuring Philips CPLDs. In *Proceedings of the PACT'98 Workshop on Reconfigurable Computing*, pages 5–10, 1998.

[66] A. Kaviani and S. Brown. Hybrid FPGA architecture. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'96*, pages 3–9, 1996.

[67] P. Kievits, E. Lambers, C. Moerman, and R. Woudsma. R.E.A.L. DSP technology for telecom baseband processing. In *Proceedings of the International Conference on Signal Processing Applications and Technology, ICSPAT'98*, 1998.

[68] S. Kumar, L. Pires, D. Pandalai, M. Vojta, J. Golusky, S. Wadi, and H. Spaanenburg. Benchmarking technology for configurable computing system. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 273–274, 1998.

[69] Y.-T. Lai and P.-T. Wang. Hierarchical interconnection structures for field programmable gate arrays. *IEEE Transactions on VLSI Systems*, 5(2):186–196, June 1997.

[70] M. Leeser, W. M. Meleis, M. M. Vai, S. Chiricescu, W. Xu, and P. M. Zavracky. Rothko: A three-dimensional FPGA. *IEEE Design & Test of Computers*, 15(1):16–23, January-March 1998.

[71] D. M. Lewis, D. R. Galloway, M. van Ierssel, J. Rose, and P. Chow. The Transmogrifier-2: A 1 million gate rapid prototyping system. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'97*, pages 53–61, 1997.

[72] G. Lu, H. Singh, M.-H. Lee, N. Bagherzadeh, F. Kurdahi, and E. M. C. Filho. The MorphoSys parallel reconfigurable system. In *Euro-Par'99, Parallel Processing*, volume 1685 of *Lecture Notes in Computer Science*. Springer, 1999.

[73] P. Lysaght and J. Stockwood. A simulation tool for dynamically reconfigurable field programmable gate arrays. *IEEE Transactions on VLSI Systems*, 4(3):381–390, September 1996.

[74] W. H. Mangione-Smith. Application design for configurable computing. *IEEE Computer*, 30(10):115–117, October 1997.

[75] G. McGregor, P. Lysaght, and D. Robinson. A hardware/software co-design environment for reconfigurable logic systems. In *Field-Programmable Logic and Applications: From FPGAs to Computing Paradigm, FPL'98*, volume 1482 of *Lecture Notes in Computer Science*, pages 258–267. Springer, 1998.

[76] O. Mencer, M. Morf, and M. J. Flynn. PAM-Blox: High performance FPGA design for adaptive computing. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 167–174, 1998.

[77] MiroTech. Aristotle. http://www.mirotech.com/product/aristotle.htm.

[78] E. Mirsky and A. DeHon. MATRIX: A reconfigurable computing architecture with configurable instruction distribution and deployable resources. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'96*, pages 157–166, 1996.

[79] E. Mirsky and A. DeHon. MATRIX: A reconfigurable computing device with configurable instruction distribution. Hot Chips Symposium, 1997.

[80] M. Moe, H. Schmit, and S. C. Goldstein. Characterization and parameterization of a pipeline reconfigurable FPGA. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 294–295, 1998.

[81] L. Moll and M. Shand. Systems performance measurement on PCI pamette. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 125–133, 1997.

[82] Motorola. MPACF250 – MPA's CORE+ reconfigurable system. Product Brief, April 1998.

[83] K. Nagami, K. Oguri, T. Shiozawa, H. Ito, and R. Konishi. Plastic cell architecture: Towards reconfigurable computing for general-purpose. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 68–77, 1998.

[84] M. Nakkar. *Evaluation of Dynamically Programmable Cache Machine with Low Power Field Programmable Gate Arrays (FPGAs) and 3-Dimensional Multi-Chip-Module Package*. PhD thesis, North Carolina State University, Department of Electrical and Computer Engineering, 1999.

[85] M. Nakkar, J. Harding, D. Schwartz, P. Franzon, and T. Conte. Dynamically programmable cache. In *Configurable Computing: Technology and Applications. Proceedings of SPIE*, volume 3526, pages 218–226. SPIE, 1998.

[86] I. Ouaiss, S. Govindarajan, V. Srinivasan, M. Kaul, and R.Vemuri. An integrated partitioning and synthesis system for dynamically reconfigurable multi-FPGA architectures. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 31–36. Springer, 1998.

[87] I. Page. Constructing hardware-software systems from a single description. *Journal of VLSI Signal Processing*, 12(1):87–107, January 1996.

[88] I. Page. Reconfigurable processor architectures. *Microprocessors and Microsystems*, 20(3):185–196, 1996.

[89] R. Payne. Self-timed FPGA systems. In *Field-Programmable Logic and Applications, FPL'95*, volume 975 of *Lecture Notes in Computer Science*, pages 21–35. Springer, 1995.

[90] Philips Semiconductors. R.E.A.L. DSP core for low-cost low-power telecommunication and consumer applications. White Paper, 1998.

[91] J. M. Rabaey. Reconfigurable computing: The solution to low power programmable DSP. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'97*, pages 275–278, 1997.

[92] N. K. Ratha and A. K. Jain. Computer vision algorithms on reconfigurable logic arrays. *IEEE Transactions on Parallel and Distributed Systems*, 10(1):29–43, January 1999.

[93] R. Razdan and M. D. Smith. High-performance microarchitectures with hardware-programmable functional units. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pages 172–180, 1994.

[94] D. Robinson, P. Lysaght, and G. McGregor. New CAD framework extends simulation of dynamically reconfigurable logic. In *Field-Programmable Logic and Applications: From FPGAs to Computing Paradigm, FPL'98*, volume 1482 of *Lecture Notes in Computer Science*, pages 1–8. Springer, 1998.

[95] C. R. Rupp, M. Landguth, T. Garverick, E. Gomersall, H. Holt, J. M. Arnold, and M. Gokhale. The NAPA adaptive processing architecture. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 28–37, 1998.

[96] S. M. Scalera, J. J. Murray, and S. Lease. A mathematical benefit analysis of context switching reconfigurable computing. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 73–78. Springer, 1998.

[97] S. M. Scalera and J. R. Vázquez. The design and implementation of a context switching FPGA. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 78–85, 1998.

[98] H. Schmit. Incremental reconfiguration for pipelined applications. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 47–55, 1997.

[99] S. Sezer, J. Heron, R. Woods, R. Turner, and A. Marshall. Fast partial reconfiguration for FCCMs. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'98*, pages 318–319, 1998.

[100] H. Singh, M.-H. Lee, G. Lu, F. J. Kurdahi, N. Bagherzadeh, and E. M. C. Filho. MorphoSys: A reconfigurable architecture for multimedia applications. In *Proceedings of the PACT'98 Workshop on Reconfigurable Computing*, pages 34–39, 1998.

[101] A. Stansfield and I. Page. The design of a new FPGA architecture. In *Field-Programmable Logic and Applications, FPL'95*, volume 975 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 1995.

[102] J. Stockwood and P. Lysaght. A simulation tool for dynamically reconfigurable field programmable gate arrays. In *Proceedings of the IEEE International ASIC Conference and Exhibit, ASIC'95*, pages 167–170, 1995.

[103] E. Tau, D. Chen, I. Eslick, J. Brown, and A. DeHon. A first generation DPGA implementation. In *Proceedings of the Third Canadian Workshop on Field-Programmable Devices, FPD'95*, pages 138–143, 1995.

[104] S. Trimberger, D. Carberry, A. Johnson, and J. Wong. A time-multiplexed FPGA. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pages 22–28, 1997.

[105] Triscend Corporation. Configurable processors: An emerging solution for embedded system design. White Paper, Version 1.0, 1998.

[106] W. Tsu, K. Macy, A. Joshi, R. Huang, N. Walker, T. Tung, O. Rowhani, V. George, J. Wawrzynek, and A. DeHon. HSRA: High-speed, hierarchical synchronous reconfigurable array. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'99*, pages 125–134, 1999.

[107] A. Tsutsui and T. Miyazaki. YARDS: FPGA/MPU hybrid architecture for telecommunication data processing. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'97*, pages 93–99, 1997.

[108] M. Vasilko and D. Ait-Boudaoud. Optically reconfigurable FPGAs: Is this a future trend? In *Field-Programmable Logic: Smart Applications, New Paradigms and Compilers, FPL'96*, volume 1142 of *Lecture Notes in Computer Science*, pages 270–279. Springer, 1996.

[109] J. Villasenor and B. Hutchings. The flexibility of configurable computing. *IEEE Signal Processing Magazine*, 15(5):67–84, September 1998.

[110] Virtual Computer Corporation. H.O.T. Works. http://www.vcc.com/Prod1.html.

[111] J. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, and P. Boucard. Programmable active memories: Reconfigurable systems come of age. *IEEE Transactions on VLSI Systems*, 4(1):56–69, March 1996.

[112] J. Walrath and R. Vemuri. A performance modeling and analysis environment for reconfigurable computers. In *Parallel and Distributed Processing, Proceedings of IPPS/SPDP'98 Workshops*, volume 1388 of *Lecture Notes in Computer Science*, pages 19–24. Springer, 1998.

[113] M. Wazlowski, L. Agarwal, T. Lee, A. Smith, E. Lam, P. Athanas, H. Silverman, and S. Ghosh. PRISM-II compiler and architecture. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'93*, pages 9–16, 1993.

[114] S. J. E. Wilton, J. Rose, and Z. G. Vranesic. Architecture of centralized field-configurable memory. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'95*, pages 97–103, 1995.

[115] M. J. Wirthlin and B. L. Hutchings. A dynamic instruction set computer. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'95*, pages 99–107, 1995.

[116] M. J. Wirthlin and B. L. Hutchings. Sequencing run-time reconfigured hardware with software. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'96*, pages 122–128, 1996.

[117] M. J. Wirthlin and B. L. Hutchings. Improving functional density through run-time constant propagation. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'97*, pages 86–92, 1997.

[118] M. J. Wirthlin, B. L. Hutchings, and K. L. Gilson. The Nano Processor: A low resource reconfigurable processor. In *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, FCCM'94*, pages 23–30, 1994.

[119] R. D. Wittig and P. Chow. OneChip: An FPGA processor with reconfigurable logic. In *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'96*, pages 126–135, 1996.

[120] XESS. XS40. http://www.xess.com/FPGA/ho04000.html.

[121] A. K Yeung and J. M. Rabaey. A 2.4 GOPS data-driven reconfigurable multiprocessor IC for DSP. In *IEEE International Solid-State Circuits Conference, ISSCC'95, Digest of Technical Papers*, pages 108–109, February 1995.

[122] H. Zhang, M. Wan, V. George, and J. Rabaey. Interconnect architecture exploration for low-energy reconfigurable single-chip DSPs. In *Proceedings of the IEEE Computer Society Workshop on VLSI, WVLSI'99*, 1999.

A Overview of RC Activities

Legend:
- ITEM research
- ITEM commercially available
- ITEM† has been withdrawn from the market

A.1 Boards attached to a Host

Single-FPGA boards

- H.O.T. Works [110] Virtual Computer Corporation
- RC1000-PP [40] Embedded Solutions
- and many more (see [51])

Multi-FPGA boards

- Splash [49], Splash 2 [10]
- WILDFIRE [9, 43] Annapolis Micro Systems
- PeRLe-1 [111] DEC
- Pamette [26, 81] Compaq Systems Research Center, formerly DEC
- Teramac [8] Hewlett-Packard
- AnyBoard [35]
- Transmogrifier-2 [71]
- Spyder [62, 63]
- ARC-PCI [6] Altera
- Spectrum [48] Giga Operations
- Bioccelerator [28], BioXL [29] Compugen
- and many more (see [51])

Hybrid boards

- PRISM [11, 113]
- CM-2X [30]
- YARDS [107]
- RENCO [16, 52]
- HARP [88]
- Neurocomputer [25]
- ACEcard [32] TSI TelSys
- Aristotle [77] MiroTech
- APAC509 [5] Alex Computer Systems
- XS40 [120] XESS
- MIX [13]
- MORRPH [36]
- PMMLA [47]
- Reconfigurable Meshes [19, 20]

A.2 FPGA Devices

Novel CLB and interconnect composition

- MATRIX [78, 79]
- RaPiD [37]
- Triptych / Montage [22]
- CAM-based FPGA [88, 101]
- Kress ALU Array [53]
- Raw μP [12]
- Rothko [70]
- Hierarchical FPGA [3, 27, 69]
- HSRA [106]
- PipeRench [50, 80]
- Plasma [7]
- Hybrid FPGA [66]
- Centralized Field-Configurable Memory [114]
- Siblings Family[15]

Novel configuration schemes

- DPGA [33, 34, 103]
- CSRC FPGA [96, 97]
- Colt / Stallion [17, 18]
- Time-Multiplexed FPGA [104]
- Striped FPGA [98]
- Dynamically Programmable Cache (DPC) [85, 84]
- OR-FPGA [108]
- Plastic Cell Architecture (PCA) [83]
- XC6200† Xilinx

Asynchronous FPGAs

- Montage [22]
- STACC [89]

A.3 Hybrid Devices

Reconfigurable computation unit

- Garp [59]
- Dynamically Programmable Cache (DPC) [85, 84]
- MorphoSys [72, 100]
- CORE+† [82] Motorola
- PRISC [93]
- Chimaera [56]
- OneChip [119]
- Hybrid Reconfigurable CPU [65] Philips
- NAPA [95] National Semiconductor

Reconfigurable I/O

- E5 [105] Triscend

Reconfigurable interconnection network

- PADDI-2 [121]

Framework with ASIC as target

- Pleiades: P1 [1, 2, 91], Maia [122]
- R.E.A.L. [67, 90] Philips

A.4 Fully Reconfigurable Processors

- DISC [115], DISC-II [116]
- Nano Processor [118]
- Spyder [62, 63]
- Self-Reconfiguring Processor [42]

A.5 Tools

HDLs and Compilers

- Handel [87], Handel-C
- Handel-C [39] Embedded Solutions
- JHDL [14]
- Transmogrifier C [44]
- Lola [46]
- SFL [83]
- and many more

CAD Tools supporting RTR

- Trianus [45, 46]
- Dynamic Circuit Switching (DCS) Simulation Tool [73, 75, 94, 102]
- ARC [112]
- DRIVE [21]
- SPARCS [86]
- PARTHENON [83]
- PAM-Blox [76]
- CORES/HASIS [38]
- and many more

A.6 Configuration Management

- Virtual FPGA [41]
- Virtual Hardware [99]
- Dynamic Reconfigurable System [64]
- PipeRench [24]
- RAGE [23]
- DISC [116]
- ACEruntime [31, 32] TSI TelSys

A.7 Benchmarks

- RAW [12]
- Stressmarks [68]