Chapter 18

The Paramountcy of Reconfigurable Computing

Reiner Hartenstein

Abstract. Computers are very important for all of us. But brute force disruptive architectural developments in industry and threatening unaffordable operation cost by excessive power consumption are a massive future survival problem for our existing cyber infrastructures, which we must not surrender. The progress of performance in high performance computing (HPC) has stalled because of the "programming wall" caused by lacking scalability of parallelism. This chapter shows that Reconfigurable Computing is the silver bullet to obtain massively better energy efficiency as well as much better performance, also by the upcoming methodology of HPRC (high performance reconfigurable computing). We need a massive campaign for migration of software over to configware. Also because of the multicore parallelism dilemma, we anyway need to redefine programmer education. The impact is a fascinating challenge to reach new horizons of research in computer science. We need a new generation of talented innovative scientists and engineers to start the beginning second history of computing. This paper introduces a new world model..

19.1 Introduction

In Reconfigurable Computing, e. g. by FPGA (Table 15), practically everything can be implemented which is running on traditional computing platforms. For instance, recently the historical Cray 1 supercomputer has been reproduced cycle-accurate binary-compatible using a single Xilinx Spartan-3E 1600 development board running at 33 MHz (the original Cray ran at 80 MHz) 1. Reconfigurable Computing is the paramount issue for continuing the progress of computing performance and for the survival of world-wide computing infrastructures. Section 19.2 of this chapter stresses, that all our computer-based infrastructures worldwide are extremely important3, also for avoiding a massive crisis of the global and local economy. Section 19.3.1 warns of the future unaffordability of the electricity consumption of the entirety of all computers worldwide, visible and embedded and, that low power circuit design5 and other traditional "green computing"9-11, although important and welcome, are by far not sufficient to guarantee affordability and not at all to support further progress for future applications of high performance computing. Thousands of books have been published about world economy, energy, CO2, climate, survival on the globe, water, food, health, etc. Hundreds of them are about peak oil. I have listed just a few of them 17-95.

In contrast to the currently still dominant von Neumann (vN) machine, Reconfigurable Computing⁹⁶⁻⁹⁹ (RC), the second RAM-based machine paradigm, introduced in section 19.4, offers a drastic reduction of the electric energy budget and speedup factors by up to several orders of magnitude - compared to using the von Neumann paradigm, now beginning to loose its domination.

nance¹⁰⁰⁻¹⁰¹. Sections 19.5.4 and 19.6.2 stress the urgency of moving RC from niche to main-stream and urges, that we need a worldwide mass movement of a larger format than that of the VLSI design revolution around 1980, where only an urgently needed designer population has been missing ¹⁰²⁻¹¹⁴. This time a properly qualified programmer population is missing. But we need to push the envelope into two different directions. The VLSI design revolution has been the most effective project in the modern history of computing. But this time we need even more. A dual rail effort (section 19.6) is needed for simultaneously developing the scene toward parallel programming for manycore architectures and to structural programming for reconfigurable computing (RC), as well as heterogeneous systems including the cooperation of both paradigms.

Currently the dominance of the basic computing paradigm is gradually wearing off (see section 19.4) with the growth of the area of Reconfigurable Computing (RC) applications - bringing profound changes to the practice of scientific computing, cyber physical systems (CPS) and ubiquitous embedded systems, as well as new promises of disruptive new horizons for affordable very high performance computing. Due to RC also the desk-top personal supercomputer is near. To obtain the payoff from RC we need a new understanding of computing and supercomputing, as well as of the use of accelerators (section 19.6.3). For bridging the translational gap, the software / configware chasm, we need to think outside the box3.

19.2 Why Computers are Important

Computers are very important for all of us. By many millions of people around the world computers are used everywhere. Typical orders of magnitude in the computer application landscape are: hundreds of applications, consisting of ten-thousands of programs, with millions of lines of code, having been developed by expenditure of thousands of man-years investment volumes up to billions of dollars3. We must maintain these important infrastructures. Wiki "Answers pages" nicely tell us, why computers running this legacy software are indispensable in the world115. The Computer is an electronic device used in almost every field even where it is most unexpected ¹¹⁷. Now we cannot imagine a world without computers. These days' computers are the tools for not only engineers and scientists but also they are used by many millions of people around the world.

The computer has become very important nowadays not only because it is accurate, fast and can accomplish many tasks easily. Otherwise to complete many tasks manually much more time is required (Fig. 1). It can do very big calculations in just a fraction of a second. Moreover it can store huge amount of data in it. We also get information on many different aspects using internet on our computer. But there are more reasons, why computers are important. Many more different kinds of local or even worldwide infrastructures will be controlled by networks of computers ¹¹⁷. Here computer crashes or software crashes may cause widely spread disasters by domino effects ⁸⁹⁻⁹⁵. In his novel Hermann Maurer depicts the worldwide total chaos caused by a network crash in the year 2080 where millions of people die and the life of billions is threatened ⁹⁰⁻⁹¹. A thrilling novel? Yes, but also a textbook about possible solutions.

BANKS use computers to keep record of all transactions and other calculations. It provides speed, convenience, security. Communication is another important aspect, very easy through internet and E-mail. Computer communicates by telephone lines and wireless. Through E-mail we can send messages to anybody in any part of the world in just a second while if we write letter

then it will reach in some days. So the internet has made the earth a global village and above all saves time. This would not be possible without computers. Internet helps to find information on every topic. It's the easiest and fastest way of research. Computer network makes the user capable of accessing remote programs and databases of same or different organizations. Without computers we also would not have any automated teller machines (ATM).

Business. Computers have now become an integral part of corporate life. They can do business transactions very easily and accurately and keep the record of all the profit and loss. Today computers can be found in every store, supermarkets, restaurants, offices etc. special software is used in these computers to calculate the huge bills within seconds. One can buy and sell things online, bills and taxes can be paid online and can also predict the future of business using artificial intelligence software. It also plays a very important role in the stock markets.

Business Information Systems. For the economy: business information systems are as essential as materials, energy and traffic. Without business information systems the economy would be extremely ineffective and inefficient. Business information systems are essential for globalization. Their significance for each enterprise: improving the productivity of the business processes (= rationalization), mastering complexity and volume, making information available fast and everywhere: for any operations, for decisions, as well as strategically for entrepreneurial planning on the creation of new business opportunities, i. e. by e-business. If automobile manufacturers would not have PPC systems (product planning & control system), cars could not be manufactured in desired wide variety. It would be like at the early times of Henry Ford, who said: cars can be delivered in any color, provided it is black.

Biological And Medical Science. Dagnostics of diseases and also treatments can be proposed with the help of computer. Many machines use computers which allows the doctor to view the different organs of our body like lungs, heart, kidneys etc. There is special software which helps the doctor during the surgery.

Education. Today the computer has become an important part of one's education because we are using computers in every field and without the knowledge of computer we cannot get a job and perform well in it. So computers can secure better jobs prospects. Knowledge about computer is must in this time.

Media. Almost every type of editing and audio- visual compositions can be made by using special software especially made for this purpose. Some software can even make three dimensional figures which are mostly used in the cartoon films. Special effects for action and science fiction movies are also created on computer.

Travel And Ticketing. Computers do all the work of plane and train reservation. It shows the data for vacant and reserved seats and also saves the record for reservation. Let us imagine, Lufthansa would handle reservations like in 1960 (Fig. 1). To-day they could not handle their flight operations by this method.

Weather Predictions are possible by experts using supercomputers: another important application.

Sports. It is also used for umpiring decisions. Many times the umpire has to go for the decision of a third umpire where the recording is seen again on computer and finally reaches to the accurate and fair decision. Simulation software allows the sportsman to practice and improve his skills.

Car Safety. Here the ultimate goal is a zero-fatality vehicle. Auto companies use computers for crash simulations to figure out how to build safer cars. GM, Ford, Honda, Mercedes Benz and other companies use this technology¹¹⁹. For computer simulations, such as one vehicle crashing into another, carmakers have the supercomputing power in-house. It has been publicly demonstrated that even a complex simulation of a full crash test with 1 million elements can take just five minutes to render using a cluster of Intel Xeon 5500 processors¹¹⁹. The latest HPC technology has enabled GM to move to an interactive design process for the entire vehicle, and run a simulation with up to four million elements. American Honda has more than 3000 processors devoted to crash analysis¹¹⁹. Mercedes-Benz is now running approximately 5,000 crash simulations for every new vehicle design. The ultimate goal is a zero-fatality vehicle. More sophisticated technology should help make much safer cars a reality in the not too distant future¹¹⁹.

Other HPC applications. HPC is pervasive enough so that it is used today not just by government and university researchers but to design products ranging from cars and planes to golf clubs, microwave ovens, animated films, potato chips, diapers and many other products.

Cyber Physical Systems (CPS) are computers or computer networks ready for real-time response, directly coupled to biological organisms or systems, to sensor networks, to organizations, or to technical networks and much more. Figure 2 lists some CPS application examples.

Daily Life. Computers are everywhere. We operate washing machines, microwave oven and many other products using software. Moreover we can store all the information about our important work, appointments schedules and list of contacts. Computers are playing a very important role in our lives. We cannot imagine the world without computers. And this technology is advancing in both, industry and home. It is creating new mass markets by a variety of wireless smart portable devices¹²¹. It is necessary for everyone to have some basic computing knowledge. Otherwise he cannot get a job as computers have invaded almost all the fields.

Survival risk of mainframe software. Tracing their roots back to IBM's System/360 from the 60ies mainframes became popular in banking, insurance, and other industries. Quite a number of companies still employ older mainframes with 3270 terminal emulator and Disk Operating System (DOS)¹²². Representing cutting-edge technology when the oldest Baby Boomers were still teenagers, mainframe survival is in danger because of recruiting problems.

Riscs of Domino Effects. The computer-based worldwide interconnectedness of all areas of life is highly risky. For instance public power supply infrastructures are computer-controlled by intelligent networks, so that any local malfunction can trigger cross-border blackouts, causing widely spread breakdowns: employees do not reach their workplace, subcontracted supply does not reach the assembly line, perishable goods do not reach their destination and/or cannot be cooled, etc. Highly vulnerable are also all our communication infrastructures. For instance, blackouts have a lot of follow-up problems: phones and even cell phones do not work. Due to highly complex global interconnectedness a minor bug may cause by chains of reactions a huge disaster. The World Economic Forum (WEF) comes to the conclusion⁸⁹, that by a wide variety of reasons such

riscs have a very high probability (Fig. 3). Our computer-controlled economic and technical infrastructures have reached such an enormous complexity that we hardly can estimate all possible domino effects.

19.2.1 Computing for Sustainable Environment

Computing for the Future of the Planet is more and more important for us because computing (computers, communications, applications) will make a major and crucial contribution to ensuring a sustainable future for society and the planet. The "Power Down for the Planet" challenge is a national competition to fight global warming by pledging to reduce the amount of energy used by computers on campus and elsewhere. Also here computers are important: computing for our sustainable environment is important for preserving our civilization, for avoiding its collapse¹⁷⁻⁹⁵.

The goal of Green Computing is simple ¹²³: reduction in the use of harmful materials, maximize energy efficiency, and promote recyclability. Green Computing ³⁴⁸ is the science of efficient and effective designing, manufacturing, using, disposing, and recycling of computers and computer related products (servers, networks, peripherals, etc.), also by creating technologies helping reduce harmful impact on the environment and to preserve natural resources. Wasting energy is costly and leading to a climatic change from burning coal and oil.

With the major goal to minimize the carbon footprint of computing 130 "Green IT" consists of 3 parts: 1) designing products that are less polluting, less energy-consuming and easier to recycle, 2) more efficient data centres, 3) innovative projects that will enable, via IT contributions, the building of a more sustainable world. Green IT supports smart measuring the energy consumption of housing, public buildings and other facilities, in order to be able to optimize the use of energy (smart meters)¹²⁸. Green IT also supports education of the data centre operators on enhanced energy optimization (green data centers9⁻¹¹) - Use of telecommuting and teleconferencing in order to reduce travel requirements. - Installation of web sites offering better information on carpooling or public transport possibilities in order to reduce the traffic on our roads. Optimization of road traffic and transport logistics. Andy Hopper even sees four levels 132 at which innovation-driven developments in computing being effective: 1) Simulation and modelling are important tools which will help predict global warming and its effects. Much more powerful computing systems, are required to make the predictions better, more accurate, and relevant. 2) The amount of infrastructure making up the digital world is continuing to grow rapidly and starting to consume significant energy resources. 3) Computing will play a key part in optimizing use of resources in the physical world. 4) We are experiencing a shift to the digital world in our daily lives as witnessed by the wide scale adoption of the world wide web. Let me add a fifth level: 5) To help generate momentum and achieve these goals, it is important that a coordinated set of challenging international projects are investigated.

The World Economic Forum proposed to help existing institutions by IT networking to enable existing institutions to unleash public value, catalyzing initiatives and unleashing human capital in the world¹³⁵. Klaus Schwab: "Our existing global institutions require extensive rewiring to confront contemporary challenges in an effective, inclusive and sustainable way." Organizations like UN, GATT, G8 and G20 are becoming increasing inept at fixing what ails the world: goals of

economic growth, climate protection, poverty eradication, conflict avoidance, human security and promotion of shared values. The topics are the three Rs: Rethink, Redesign and Rebuild¹³⁴.

The Forum's "Global Redesign Initiative" report notes 136 ,,how the digital world has brought about cross-border integration by new technologies enabling virtual interaction have created a world that is much more complex and more bottom-up than top-down." The world has become economically, politically and environmentally more interdependent - without a new set of international bureaucracies piled on the existing ones. It has been argued for a global system with graphic visualization tools to measure success, for a complete redesign the global legal system, for a global vaccine protocol, global intellectual property system, global risk management, etc. This means taking a Wikinomics approach -- embracing more agile structures enabled by global networks for new kinds of collaboration such, that do not need a new set of international bureaucracies piled on the existing ones.

Governments need to launch a new paradigm to involve the citizens of the world through mass collaboration by a new medium of communications including tools like digital brainstorms and town hall meetings: decision-making initiatives like citizen juries and deliberative polling; execution tools like policy wikis and social networks within government and evaluation programs. This initiative demonstrates how important to reinvent computing and the growth of IT and the internet is for broad engineering issues insuring sustainability issues of the world like smart energy production and distribution, intelligent water management, strengthening welfare, dealing with ageing and young population - mitigating riscs ¹³⁸.

19.3 Performance Progress stalled

Not only disruptive architectural developments in industry stall further progress of IT with respect to energy-inefficiency and performance improvements. Unaffordable operation cost by excessive power consumption are a massive future survival problem for existing cyber infrastructures, which we must not surrender. Because of the inevitable manycore architecture contemporary computer systems are in an all-dominant programmability crisis. The progress of performance is massively stalled because of this "programming wall" caused by lacking scalability of parallelism and an ubiquitous programmer productivity gap ^{139,140,143}. Later in this chapter we show that reconfigurability is the silver bullet to obtain massively better energy efficiency as well as much better performance by the upcoming heterogeneous methodology of HPRC (high performance reconfigurable computing). We also believe in the need for a massive campaign for migration of software over to configware. Also because of the multicore parallelism dilemma, we anyway need to reinvent programmer education ¹⁴⁶. The impact is a fascinating challenge to reach new horizons of research in computer science. We need a new generation of talented innovative scientists and engineers to start the beginning second history of computing. This chapter discusses its new world model.

19.3.1 Unaffordable Energy Consumption of Computing

The future of our world-wide total computing ecosystem is facing a mind-blowing and growing electricity consumption, together with a trend toward growing cost and shrinking availability of energy sources. The electricity consumption by the internet alone causes more Greenhouse Gas

emission than the world-wide air traffic. Will The Internet Break?^{89-91,147} Consumer broadband connections in North America, Mexico and Western Europe have reached 155 millions by the end of 2007 and are predicted to reach 228 millions 4 years after 2007. The network if the internet is being stressed more than ever with new technologies and larger e-mails, and the trend will accelerate. An explosion in services integrating video and software will intensify by increasing popularity of games, massive use of video on demand, high-definition video and pay-TV to the living room, as well as by newer services by mobile phone companies. and multiple connected PCs¹⁴⁸ and devices using connections¹⁴⁷. The internet service providers need to be able to assess how much more bandwidth will be required and how much headroom they have.

It has been predicted that by the year 2030, if current trends continue, worldwide electricity consumption by ICT infrastructures will grow by a factor of 30¹⁵³, reaching much more than the current total electricity consumption of the entire world for everything, not just for computing. The trends are illustrated by an expanding wireless internet, and by a growing number of internet users. as well as with tendencies toward more video on demand, HDTV over the internet, shipping electronic books, efforts toward more cloud computing¹⁵⁵ and many other services. Other estimations claim, that already now the greenhouse gas emission from power plants generating the electricity needed to run the internet is higher than that of the total world-wide air traffic. For more predictions see^{156.158}.

Already for to-day's petascale (10¹⁵ calculations/second) supercomputer systems with the annual power and cooling cost exceeding the acquisition cost of servers, the power consumption has become the leading design constraint¹⁴⁹. Extrapolating from today's petascale systems to future exascale machines (10¹⁸ calculations/second, a processing capability close to that of the human brain 152), the overall power consumption is estimated to be on the order of 10 GW 149,152 v, twice the power budget of New York City with a population of 16 millions (an earlier estimation for one system is 120 MW¹⁵¹). The electricity bill is a key issue not only for Google, Microsoft, Yahoo and Amazon with their huge data farms at Columbia River¹²⁵ (Fig. 4). That's why Google recently submitted an application asking the Federal Energy Regulatory Commission for the authority to sell electricity 160, and has a patent for water-based data centers, using the ocean to provide cooling and power (using the motion of ocean surface waves to create electricity: Fig. 5)¹⁶². Already in the near future the electricity bill of most facilities will here be substantially higher than the value of their equipment ¹⁶³. Already in 2005, Google's electricity bill was with about 50 million US-\$higher than the value of its equipment. Meanwhile the cost of a data center is determined solely by the monthly power bill, not by the cost of hardware or maintenance 164. Google's employee L. A. Barroso said 165: "The possibility of computer equipment power consumption spiraling out of control could have serious consequences for the overall affordability of computing."

Rapidly growing energy prices are predicted since the oil production has reached its peak by about the year 2009¹⁶⁷⁻¹⁶⁹. Already currently 80% of crude oil is coming from decline fields (Fig. 6). However, the demand is growing because of developing standards of living in China, India, Brazil, Mexico and newly industrializing countries. The world Energy Council estimates, that the demand will double until the year 2050¹⁷⁰. We need at least "six more Saudi Arabias for the demand predicted already for 2030" (Fatih Birol, Chief Economist IEA¹⁷¹). I believe, that these predictions do not yet consider the rapidly growing electricity consumption of computers. Maybe, we will need 10 more Saudi Arabias. About 50% of the shrinking oil reserves are under water¹⁷³. In

consequence of the Gulf of Mexico oil spill not all deepwater explorations will be allowed, insurance rates rise, and the crude oil prices will go further up (Fig. 7). Transitions from carbon fuels to renewables cannot completely fill the gap within at least two decades. This will cause a massive future survival problem for running our cyber infrastructures, which we must not surrender because this is an important global economy issue. Or, should we dig more coal ¹⁷⁴? It makes sense, to measure computing performance not just by MIPS (million instructions per second), but by MIPS/watt or FLOPS/watt instead ¹⁷⁵.

19.3.2 Crashing into the Programming Wall

For 40 years, semiconductor technology has followed Moore's Law. Until about 2004 we obtained better performance by just waiting for the next generation microprocessor with higher clock speed. Because of this free ride on Gordon Moore's law the improvement of software performance has been the successful job of hardware designers. This development ended, when the microprocessor industry changed strategy from a single CPU on a chip to manycore by increasing the number of on-chip processor cores instead of growing clock frequency. The "golden" CMOS era is gone 140. Technology scaling does not deliver anymore significant performance speedup and increasing power density poses severe limitations. High performance is no more the job of hardware designers. We hit the "programming wall" since high performance now requires rare parallel programming skills 140,176. Great challenges for RC provide the answer 140,143.

This "programming wall"¹⁷⁸ we know from supercomputing is not new. The dead supercomputer society list¹⁸⁰ demonstrates, that almost all much earlier supercomputing projects and start-ups failed, since parallel programming has been required, which crashed into the parallel programming wall. This list is not even complete. More dead projects are listed elsewhere ¹⁸¹. Even to-day the vast majority of HPC or supercomputing applications was originally written for a single processor with direct access to main memory. But the first petascale supercomputers employ more than 100,000 processor cores each, and distributed memory. Three real-world applications have broken the petaflop barrier (10¹⁵ calculations/second) (Jaguar at ORNL)¹⁷⁸. A slightly larger number have surpassed 100 teraflops (100 x 10¹² calculations/second), mostly on IBM and Cray¹⁷⁸. The scene hopes, that dozens of applications are inherently parallel enough to be laboriously decomposed, sliced and diced, for mapping onto such highly parallel computers. But a large number of applications is only modestly scalable. More than 50% of the codes do not scale beyond eight cores, only about 6% can exploit more than 128 PE, still a tiny fraction of 100,000 or more available cores¹⁷⁸.

A very important issue is saving energy ^{14,182}. But multicore processors tend to have tuned-down speeds. Down from 3 to 4 GHz (single-core) each core meanwhile runs at about half that speed. Some HPC or supercomputing sites report that some of their applications were running more slowly on their newest HPC system ¹⁷⁸. With almost 70% of this market x86-based intel or AMD processors are dominant (in 2009). But, multiplied collective peak performance comes without corresponding increases in NoC bandwidth, making it difficult to move data into and out of each core fast enough to keep the cores busy ^{178,183}. We have to rethink not only Amdahl's law. Adding accelerators via a slow PCI bus adds to the problem. Both hardware and software advances are urgently needed.

We see that massive hardware parallelism from skyrocketing core counts is racing ahead of programming paradigms and programmer productivity. This parallel performance "wall" will reshape the nature of HPC code design and system usage ¹⁴⁰. The evolutionary path is not addressing the fundamental problems. A large number of HPC applications will need revolutionary changes to be fundamentally rethought and rewritten within the next five to 10 years by serious algorithm development. We've seen examples of mathematical models and algorithms that broke when pushed beyond. There aren't enough people with the right kind of brainpower^{12,13}. Universities should produce more.

Semiconductor technology has followed Moore's Law throughout 4 decades. But continuing that pattern will require not only a breakthrough in energy-efficient design. With a very high probability we will be forced to seek an entirely new paradigm¹⁸⁶. This crisis and its key issues such as software scalability, memory, IO, and storage bandwidth, and system resiliency stems from the fact that processing power is outpacing the capabilities of all the surrounding technologies.

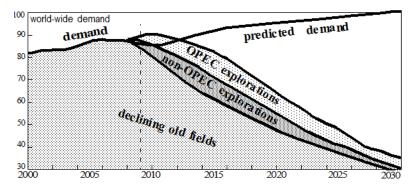


Fig. 1. Beyond Peak Oil: massively declining future crude oil production [22].

2.1.1 Green Computing: important, but limited

Green Computing tends to use conservative methods to save energy by more efficient modules and components. For example LED flat panel displays need much less power than LCD-based displays. Also much more power-efficient power supply modules are possible. The potential to save power is substantially less than an order of magnitude: maybe, a factor of about 3 to 5. A scene separate from Green Computing is Low Power Circuit Design, now also called Low Power System on Chip Design (LPSoCD). Its most important conference series are about 30 years old: the PATMOS (oldest) and the ISLPED conference series.

Several methods are known for LPSoCD, such as: Active Body Bias (ABB), Adaptive Voltage Scaling (AVS), Dynamic Voltage Scaling (DVS), Multiple Supply Voltages (MSV), Multi-Threshold CMOS (MTCMOS), Power Gating (PG), Power Gating with Retention (RPG), etc. [16]. However, the order of magnitude of the benefit to be expected from this subarea LPSoCD is rather low. By MSV in using 3 Vdds the power reduction ratio at best is about 0.4 [16]. LPSoCD is a matter of ASIC design, e. g. of hardwired

accelerator design. Only 3 % of all design starts are ASIC designs (Fig. 2) with a trend leading further down. But in fact, low power design is also used for developing better power-efficient FPGAs - to the benefit of Reconfigurable Computing. But we need a much higher potential of saving energy because "Energy cost may overtake IT equipment cost in the near future" [9]. "Green Computing has become an industry-wide issue: incremental improvements are on track" [23], But "we may ultimately need revolutionary new solutions." [25] Let me correct this statement by "we will ultimately also need revolutionary solutions (like reconfigurable computing), since we need much higher efficiency."

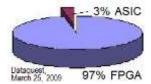


Fig. 2. FPGA to ASIC design start ratio..

2.1.2 Massively Saving Energy by RC

The idea of saving energy by using Reconfigurable Computing is not new [26, 27]. Being very important to massively reduce the energy consumption of computing, by up to several orders of magnitude, Reconfigurable Computing is extremely important for the survival of the world economy. Already a partial paradigm shift promises to save electricity by orders of magnitude. Dozens of papers (ref. in [28]) have been published on speed-ups obtained by migrating applications from software running on a CPU, over to configware for programming FPGAs [28]. It has been reported already more than a decade ago, that for a given feature size, microprocessors using traditional compilers have been up to 500 times more power hungry than a pure hardware mapping of an algorithm in silicon [27] (Fig. 3). Speedup factors up to 4 orders of magnitude have been reported from software to FPGA migrations [26-49]. Here the energy saving factor is roughly about 10% of the speedup factor, i.e., still up to >3 orders of magnitude.

Fig. 4 shows a few speedup factors picked up from literature, reporting a factor of 7.6 in accelerating radiosity calculations [46], a factor of 10 for FFT (fast Fourier transform), a speedup factor of 35 in traffic simulations [47]. A speedup by a factor of 304 is reported for an R/T spectrum analyzer [48]. For digital signal processing and wireless communication, as well as image processing and multimedia, speed-ups by 2 to almost 4 orders of magnitude have been reported. In the DSP area for MAC operations a speedup factor of 100 has been reported compared to the fastest DSP on the market (2004) [49]. Already in 1997, a speedup between 7 and 46 has been obtained over the fastest DSP [26]. In the multimedia area we find factors ranging from 60 to 90 in video rate stereo vision [34] and from 60 to 90 in real-time face detection [35], and of 457 for hyperspectral image compression [36]. In communication technology we find a speedup by 750 for UAV radar electronics [37]. For cryptography speed-ups by 3 to >5 orders of magnitude have been obtained. For a commercially available Lanman/NTLM Key Recovery Server [50] a speedup of 50 - 70 is reported. Another cryptology application reports a factor of 1305. More recently for DES braking a speed-up by x 28514 has been reported [51] (Table 1).

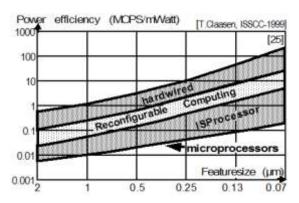


Fig. 3. Better power efficiency by accelerators..

Table 1. Recent speed-up / power save data from software to configware migration [51].

SGI Altix 4700 w. RC 100	speed-up	sa	ve factor	
RASC vs. Beowulf cluster	factor	power	cost	size
DNA & Protein sequencing	8723	779	22	253
DES braking	28514	3439	96	1116

For Bioinformatics applications [52] (also see [29]) speed-ups have been obtained by 2 to 4 orders of magnitude. Compared to software implementations sensational speed-up factors have been reported for software to FPGA migrations. A speedup of up to 30 has been shown in protein identification [30], by 133 [31] and up to 500 [32] in genome analysis. The Smith-Waterman algorithm, which is used for protein and gene sequence alignment, is basically a string-matching operation that requires a lot of computational power [52]. Here another study demonstrates speedups of 100x using Xilinx Virtex-4 hardware matched against a 2.2 GHz Opteron [53]. A speedup by 288 has been obtained with Smith-Waterman at the National Cancer Institute [33]. More recently a speed-up higher by more than an order of magnitude has been obtained here [45]. The CHREC project (supported by 24 industry partners [54]) reports running Smith-Waterman on a Novo-G supercomputer, a cluster of 24 Linux servers, each housing four Altera Stratix-III E260 FPGAs. According to this CHREC study, a four-FPGA node ran 2,665 times faster than a single 2.4 GHz Opteron core [55]. Another Smith-Waterman DNA sequencing application that would take 2.5 years on one 2.2 GHz Opteron is reported to take only 6 weeks for 150 Opterons running in parallel. Using 150 FPGAs on NRL's Cray XD1 (speedup by 43) is reported to further reduce this time to 24 hours, which means a total speedup of 7,350X over a single Opteron [42]. These are just a few examples from a wide range of publications [29-51] reporting substantial speedups by FPGAs.

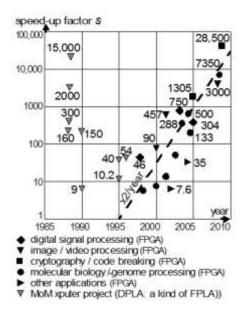


Fig. 4. Speed-up factors.

Recently not only energy saving factors have been reported, roughly one order of magnitude lower than the speed-up. More recently has been reported [51] for DES breaking (a crypto application): 28,500 (speed-up) vs. 3439 (saving energy) and for DNA sequencing 8723 (speed-up) vs. 779 (saving energy) etc. (Table 1). This paper also reports factors for saving equipment cost (up to x96) and equipment size (up to 1,116, see Table 1). No hangar full of equipment is needed when FPGAs are used in Scientific Computing. The Pervasiveness of FPGAs is not limited to embedded systems, but is also spread over practically all areas of scientific computing, where high performance is required and access to a supercomputing center is not available or not affordable. The desk-top supercomputer is near.

2.1.3 A Mass Movement needed as soon as possible

This subsection emphasizes that RC is a critical survival issue for computing-supported infrastructures worldwide and stresses the urgency of moving RC from niche to mainstream. It urges acceptance of the massive challenge of reinventing computing, away from its currently obsolete CPU-processor-centric Aristotelian CS world model, over to a twin-paradigm Copernican model. A massive software to configware migration campaign is needed. First this requires clever planning to optimize the effort versus its expected results. Which software packets should be migrated first. All this requires massive R&D and education efforts taking many years. Lobbying for the massive funding should be started right now. We should address politicians at all levels: community level, state level, national level, and European Union level.

To explain all this to politicians is very difficult. Since politicians always watch the sentiment of their voter population, we efficiently have to teach the public, which is a challenge. Without a strong tailwind from the media a successful lobbying seems to be almost successless. All this has to be completed as soon as possible, as long as we can still afford such a campaign. To succeed with such a challenging educational campaign the foundation of a consortium is needed for running an at least Europe-wide project.

2.2 Reconfigurable Computing

This section introduces a flavor of Reconfigurable Computing, its history, its more recent developments and, the massive impact on the efficiency of computing it promises. It is not easy to write this section such that it may be (almost) readable for non-experts - like the booklet "FPGAs for Dummies" which may help a little bit [56]. A Classical application for reconfigurable computing subsystems is the use as an accelerator to support the CPU ("central processing unit"). According to the state of the art in the 90's, having been the tail wagging the dog, this typically was and is a non-von-Neumann accelerator [57]. But we have to distinguish two kinds of such accelerators: made from hardwired logic or from field-programmable logic. These two kinds are distinguished by binding time of their functionality: (1) before fabrication for fixed logic or hardwired logic devices (HWD) vs. (2) after fabrication for (field-)programmable logic devices (PLD). The term "field-programmable" indicates, that by reconfiguration the functionality can be changed also at the users site by receiving new configuration code: from some memory, or, even over the internet.

First field-programmable blocks from the early 80's have been so-called FPLAs featuring very area-efficient layout similar as known from ePROM memory for the price of being able to compute only Boole-an functions in sum-of-product form. Very high speed-up could be obtained by matching hundreds of boolean expressions within a single clock cycle instead of computing them sequentially by a microprocessor. Together with a reconfigurable address generator [58] this brought a speed-up by factor up to 15,000 [59-63] for a grid-based design rule checker - already in the early 80's. Via the multi project chip organization of the E.I.S. project such a FPLA (which was called DPLA) has been manufactured on a multi-project chip of the multi university E.I.S. project: the German contribution to the Mead-&-Conway VLSI design revolution. This DPLA has the capacity of 256 first FPGAs (field-programmable gate array) just appearing on the market (by Xilinx in 1984). This again demonstrates the massive area-inefficiency of FPGAs contributing to the Reconfigurable Computing Paradox (see section 2.3.2) and the very early high speed-ups (Fig. 4).

The usual acronyms (Table 2) FPLA and FPGA are highly confusing being really not intuitive. From the straight-forward language feeling there does not seem to be any difference between "logic" in logic array (LA) and "gate" in Gate Array (GA). What is really different with FPGAs? In fact, FPGAs feature much more flexibility by introducing CLBs and routable wiring fabrics for interconnect between CLBs (Fig. 5). In contrast to FPLAs, the CLB in FPGAs allows for instance to select one of 16 logic functions from simple LUTs (look-up tables, Fig. 6). However, PLAs [64,65] are not routable and allow only to implement Boolean functions in sum-of-product form.

Table 2. List of acronyms.

acronym	meaning	acronym	meaning	acronym	meaning	
ALU	Arithmetic/Logic Unit	ESL	electronic system-level	MIPS	mio instructions p. second	
ASM	auto-sequencing memory		design	MoPL	map-oriented PL	
CE	computer engineering	EU	European Union	PC	personal computer	
CS		FIR	finite impulse response	PL	programming language	
THE STATE OF	computer science	FPGA	field-programmable GA			
CHREC	NSF Center for High-	FPL	field-programmable logic	PLA	programmable LA	
	Performance Reconfi- gurable Computing	FPLA	field-programmable PLA	PLD	programmable logic device	
CLB	configurable logic block	GA	(routable) gate array	PROM	programmable ROM	
CPU	vN central processing unit	GPP	general purpose processor	PU	processing unit	
DNA	Deoxyribonucleic acid	GPGPU		RAM	random access memory	
DPA	datapath array	GPU	on graphics processing units	RC	reconfigurable computing	
DPU	datapath unit		graphic processing unit	rDPA	reconfigurable DPA	
		HPC	high performance computing	rDPU	reconfigurable DPU	
DSP	digital signal processing	HPRC	high performance RC	rΕ	reconfigurable element	
EDA	electronics design automation	HWD	hard-wired device	ROM	read-only memory	
EPLA	e-programmable LA	ICT	information and com- munication technology	R&D	research & development	
EPP	extensible programmable platform	IEA	intern'l energy agency	SMP	symmetric multiprocessor	
ePROM.	e-programmable PROM	IT	information technology	TM	transactional memory	
	+ +	LA	(compact) logic array			
ERC ESA	electrical rules checker embedded software	LUT	look-up table	VLSI	yery large scale integrated (circuits)	
ESA	automation	MAC		vN	von Neumann	

Beyond such fine grained reconfigurability the progress of Moore's law leads to higher abstraction levels with "coarse-grained reconfigurability" featuring also CFBs (configurable function blocks), which may be adders, multipliers and/or many other functions. The next step is coarse-grained "platform FPGAs", which also include one or several microprocessors, like the PowerPC in earlier platform FPGAs from Xilinx.

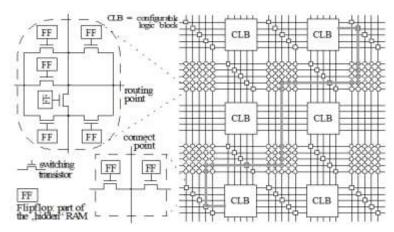


Fig. 5. Interconnect fabrics example of a routable GA; grey line: example of one routed wire connecting 2 CLBs with each other..

n			ab	W4 100		
**	800	801	810	811	f(A, B)	
0	0	0	0	0	0	
1	0	.0	0	1	A and B	
2	0	0	1	0	B disables A	
3	0	0	1	1	A	
4	0	1	0	0	A disables B	
5	0	1	0	1	В	
6	0	1	1	0	A exor B	
7	0	1	1	1	A or B	
8	1	0	0	0	not(A or B)	
9	1	0	0	1	A coin B	
10	1	0	1	0	no (B)	
11	1	0	1	1	B implies A	
12	1	1	0	0	not (A)	
13	1	1	0	1	A implies B	
14	1	1	1	0	not(A and B)	
15	1	1	1	1	1	
1	5 60%	801	\$10 20/03	811	00 01 10 f(A, B	

Fig. 6. LUT example.

2.2.1 Embedded Systems vs. Supercomputing

A growing trend is the use of FPGAs in embedded systems: ERC (Embedded Reconfigurable Computing). Originally there has been a feeling that FPGAs are too slow, power-hungry and expensive for many embedded applications. This has changed. With low power and a wide range of small packages, particular FPGAs can be found in the latest handheld portable devices, including smartphones, eBooks, cameras, medical devices, industrial scanners, military radios, etc.

But embedded designers just don't like FPGAs with CPUs inside [66]. FPGAs in this context have been very much seen as a hardware engineer's domain, with the softies allowed in to play at some late stage [67]. Xilinx pre-announced a new family of devices "going beyond the FPGA". This "Extensible Programming Platform (EPP)" has a hardwired area with a top-end twin-core ARM Cortex-A9M processor unit and with a NEON multimedia processor, memory interfacing, peripherals, and a programmable fabric [67]. Instead of communicating across an FPGA, the two processors are connected by 2,500 wires, providing much capacity for an AMBA-AXI bus and other communications protocols. Xilinx was stressing that this approach recognizes the increasingly dominant role of software in systems and is pushing EPPs as a way to first define the system in software and then carry out software and hardware design in parallel.

EPPs make the processor the centre of the device with the programmable fabric as an extra. And this, argues Xilinx, now puts the software engineer first with the hardies following behind. In EPPs the FPGA logic and the CPU will be programmable separately. FPGA configuration will be handled by the processor(s) directly, not by a serial ROM. In other words, you have to tell the FPGA you want it configured. That's very un-FPGA-like [66]. That's EPP-like. The approach of using both a processor and programmable fabric allows design to start at high level and the system to be implemented as software [67].

Xilinx's first attempt at this was an FPGA with a processor inside. This time around, it's a processor with an FPGA grafted on. That's not just semantic hair-splitting: it's the big difference between these chips and the old ones. The new chips will boot up and run just like normal microprocessors, meaning there's no FPGA configuration required at all [68].

EPPs are a result of the new research topic Network-on-Chip (NoC) [69], which is a new paradigm for designing the on-chip hardware communication architecture based on a communication fabric, also including on-chip routers. NoC CAD tool flows also support mapping applications into NoC.

Apart from ERC (Embedded Reconfigurable Computing) we have another reconfigurable computing scene: HPRC (High Performance Reconfigurable Computing), This last is a relatively new area, but has attracted a lot of interest in recent years, so much so that this entire new phrase has been coined to describe it [71]. HPRC uses FPGAs as accelerators for supercomputers [72]. Large HPC vendors are already supplying machines with FPGAs ready-fitted, or have FPGAs in their product roadmaps. What are the benefits of using FPGAs in HPC? Also here the first and most obvious answer is performance. HPC is renowned as that area of computing where current machine performance is never enough. A problem yet to be solved here is programmer productivity [78-80]. It is an educational challenge, that programmers with the needed mix of skills are hardly available. Will FPGAs have a tough road ahead in HPC?

2.2.2 The Reconfigurable Computing Paradox

Technologically FPGAs are much less efficient than microprocessors [68, 70]. The clock speed is substantially lower. The routable reconfigurable wiring fabrics causes a massive wiring area overhead. There is also massive other overhead: reconfigurability overhead, since of 200 transistors e. g. maybe about 5 or even less of them (Fig. 9 in [70]) serve the application, whereas the other 195 are needed for reconfigurability (Fig. 6). Often there is also routing congestion, so that not all CLBs can be used, what causes further degradation of efficiency. Software to configware migration yield massive improvements in speed and power consumption, although FPGAs are a dramatically worse technology. We call this the "Reconfigurable Computing Paradox". by orders of magnitude better performance with a drastically worse technology? What is the reason? It's the von Neumann paradigm's fault. The next subsection goes into details.

2.2.3 Why von Neumann is so inefficient

The von Neumann paradigm has been criticized often [76-79]. Peter Newman had for 15 years each month the highly critical "computers at risk" back pages of Communications of the ACM [80]. Nathan's law (by Nathan Myhrvold, a former CTO of Microsoft) said that software is a gas, which fills any available storage space: on-chip memory, extra semiconductor memory located outside the processor chip, as well as hard disks. A lady (I forgot her name) said that it even fills the internet. Nicklaus Wirth's pre-manycore interpretation of Moore's law is, that "software is slowing faster than hardware is accelerating" [78].

Why is von Neumann so inefficient? It is the von Neumann syndrome [82] caused by the fact, that instruction streams are very memory-cycle-hungry. We can distinguish 2 different reasons: algorithmic complexity required by the von Neumann paradigm, and, architectural issues. There are also other attempts to explain at least particular symptoms of this syndrome (Fig. 8 [83]) [84]. A well known architectural problem is the memory wall [85,86] (Fig. 7): the access time to RAM outside the processor chip is slower by a factor of about 1000, than to on-chip memory [86]. This difference is growing by 50% every year. It is a dramatic software engineering issue, that multiple levels of instruction stream overhead leads to massive code sizes which hit the memory wall [86]. However, rDPUs and rDPAs do not suffer from Nathan's law, since at run time no instruction streams are running through.

How data are moved is a key issue. CPUs usually move data between memories requiring instruction streams (first line, Table 3). This means the movement of data is evoked by execution of instructions due to the von Neumann paradigm. Also the execution of operations inside a CPU requires reading and decoding of instructions (Fig. 8 gives an idea of the overhead of the main components for contemporary CPUs).

However, after a full migration to static reconfigurable computing an algorithm is run by data streams only. Instead of a hardwired program counter reconfigurable data counters are used which do not require instruction sequences for address computation.

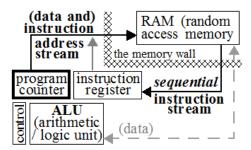


Fig. 7. von Neumann principles.

Also, how data are moved inside the data paths is a key issue, and pipe network structures to interconnect rDPUs avoid moving data through memory blocks because data are moved directly from DPU to DPU [87]. This means, that operation execution inside a DPU (not having a program counter) is "transport-triggered" (second line, Table 3). It is triggered via handshake by the arrival of the data item, not needing an instruction to call it. Not looking at dynamically reconfigurable systems ([88] only for advanced courses) we see, that reconfigurable fabrics don't perform any instruction sequencing at run time.

Table 3. Twin paradigm fundamental terminology.

		controlled by	machine	state register		
# 1	term		paradigm	type	location	
1	software	instruction streams	von Neumann	progam counter	in CPU	
2	configware	(configuration memory)	none	none	(hidden)	
3	flowware	reconfigurable address generator	data stream machine	data counter	in ASM memory block	

Beyond such fine grained reconfigurability the progress of Moore's law leads to higher abstraction levels with "coarse-grained reconfigurability" featuring also CFBs (configurable function blocks), which may be adders, multipliers and/or many other functions. The next step is coarse-grained "platform FPGAs", which also include one or several microprocessors, like the PowerPC in earlier platform FPGAs from Xilinx

But by a migration sometimes also the amount of data streams may be minimized by changing the algorithm. Here an illustration example for reducing the algorithmic complexity is given by the migration of the well-known O(n2) complexity bubble sort algorithm away from von Neumann. The algorithmic complexity turns from O(n2) into O(n) [8]. In a similar manner, other well-known algorithmic methods can be transformed to explore parallelism and locality, like in dynamic programming as presented in [88]. The combination of these effects leads to massive speed-up and massive saving of energy.

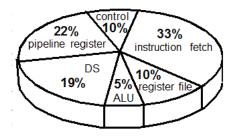


Fig. 8. All but ALU is overhead: x20 efficiency [83].

Of course, the data entering or leaving such an array (Fig. 9) have to be stored. The datastream machine paradigm uses auto-sequencing Memory blocks (asM). Each asM has a reconfigurable address generator and data counter inside. so that no instruction streams are needed for address computation. All this data streams can be programmed via data-imperative languages [90], being a kind of sisters of classical instruction-imperative programming languages. Data-imperative languages are easy to teach since both classes of imperative languages use the same primitives (Table 4). But there is one exception: data-imperative languages also support parallelism inside loops (Table 4). This also contributes to the benefit by reconfigurable computing. The simultaneous use of both classes of languages we call "twin-paradigm approach" (Table 5 and Fig. 10).

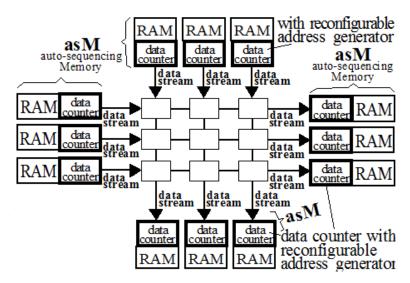


Fig. 9. Example for data stream processor principles.

Table 4. imperative language twins: program counter vs. data counters.

	instruction stream languages	data stream languages
sequencing primitives	read next instruction goto (instruction address) jump (to instruction address) instruction loop instruction loop nesting escapes instruction stream branching	read next data item goto (data address) jump (to data address) data loop data loop nesting escapes data stream branching
asymmetry	loops not internally parallel	yes: loops internally parallel

2.3 Why We Need To Reinvent Computing.

We cannot afford to relinquish RC. We will urgently need this technology to cope with threatening unaffordable operation cost by excessive power consumption of the entirety of all von Neumann computers world-wide. We need to migrate many application packages from software over to configware. This is a challenge to reinvent computing to provide the qualifications needed since disruptive developments in industry have caused the many-core programming crisis. Intel's cancellation the Tejas and Jayhawk processors indicated in May 2004 the end of frequency scaling's dominance to improve performance. "Multicore computers shift the burden of software performance from VLSI designers over to software developers." [89]. For Gary Smith from GS-EDA the three biggest disruptions are not only (1) many-core silicon, but also (2) non-vN architectures, and (3) parallel software, and the center of gravity shifts from EDA to programming (not "software" how Gary named it [93], compare Table 3 and Fig. 10).

To use manycore we need to rewrite our software: our biggest problem. RTL verification moves up to ESL. EDA Industry and ESA need to merge. Especially for Embedded Software Automation (ESA) we need tools to develop parallel software. He calls for an approach of using both, processor and programmable fabric, allows design to start at high level and the system to be implemented as progware (programware, see Fig. 10), e. g. with tools like LabView, MatLab, or others. From Xilinx and ARM hardware and software IP and tool chains are available useful to speed up time-to-market and reducing risk. It's easier to take advantage of accelerators than to try to integrate and program more processors.

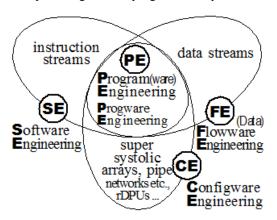


Fig. 10. New CS world model image.

To rewrite the software the qualified programmer population is not existing: a huge challenge to provide new educational approaches to qualify for heterogeneous systems including both, parallel software and configware. This requires much more than just bridging the traditional hardware/software chasm in education [91]. We need robust and fast implementations of adequate compilers and design tools, e. g. automated by formal techniques based on rewriting [92]. The biggest payoff will come from Putting Old ideas into Practice and teaching people how to apply them properly [94]. Dimensionality-rooted scaling laws favor reconfigurable spatial computing over temporal computing. Time to space mapping even dates back to the early 70's and even the late 60's, years before the first hardware description languages came up [91, 95, 96]. "The decision box (in the flow chart) turns into a demultiplexer. This is so simple! Why did it take 30 years to find out?" [97].

The impact is a fascinating challenge to reach new horizons of computer science. We need new generations of talented innovative scientists and engineers to start the second history of computing.

2.3.1 The "parallel programming problem"

Parallel computer programs are difficult to write: performance is affected by data dependencies, race conditions, synchronization, and parallel slowdown. The problem is: how to optimize parallel Computing despite Amdahl's Law? The "parallel programming problem" has been addressed high performance computing for more than 25 years with very disappointing results [100-102]. Programming languages research has stalled for several decades [103]. Informal approaches are not working. For the multi-core era, we must adopt a systematic approach informed by insight into how programmers think [104].

I do not agree. We have to teach them how to think. We have to teach programmers how to "think parallel", how to find concurrent tasks, how to synchronize free from deadlock and race conditions, how to schedule tasks at the right granularity, and, how to solve the data locality problem. Perhaps a new visual programming paradigm is required [68]. What are the right models (or abstractions) to avoid typical problems [105]: multi-core version applications running slower, problems with race conditions, and strategies for migrating code to multi-core. We see a promising new horizon: a model-based twin-paradigm methodology to master the hetero of all 3: Single-core, Multicore, & Reconfigurable Computing.

We need a different way of thinking. "The shift to multicore processor architectures really is stressing existing programming models" said Richard C. Murphy at Sandia aiming at redesigning memory systems to move computation as close to memory as possible to eliminate the traditional load-store approach where big systems use more resources moving data around than for actually computing [106]. Since Linpack doesn't measure performance for actual problems in many application areas, Sandia has proposed Graph500 as a new rating system for testing skills in analyzing large, graph-based structures that link huge numbers of data points [107]. Studies show that moving data around (not computations) will be the dominant energy problem.

2.3.2 Why FPGAs Should Win

Most ASIC design world-wide has stopped [108]. Only 3% of all design starts are ASIC designs (Fig. 3) from mega-funded companies with gigantic-volume products that can afford latest generation custom SoC development, and niche players that continue doing ASIC design with older-generation processes. The enormous 97% gap can be filled best by hybrid FPGA/hard-core devices - by FPGA companies with the required technology and infrastructure to sell and support them.

"FPGAs have become incredibly capable with respect to handling large amounts of logic, memory, digital-signal-processor (DSP), fast I/O, and a plethora of other intellectual property (IP)" [109]. At 28-nm, FPGAs deliver the equivalent of a 20- to 30-million gate application-specific integrated circuit (ASIC). At this size, traditionally used FPGA design tools begin to break down and can no longer design and verify these devices in a reasonable amount of time.

This positions FPGA companies in the best place they've ever been - a place from where they could capture huge segments of the standard parts and ASSP business with semi-standard parts that include FPGA fabric for application-specific customization. Instead of today's still-very-general-purpose FPGAs, we'll see more devices with a narrower application focus without too much general-purpose overhead on the die. With each passing process generation, the cost of that overhead shrinks, and the cost of developing the traditional alternative solutions goes up.

Industry abandoned the "pure" FPGA [110]. Countless failed FPGA start-ups have proven that the magic is not in the fabric. The real keys are tools, IP, and support for enabling the customer/designer to get the fabric to do what they want as easily as possible, and with minimal risk.

It turns out that the solution is a mixture of FPGA fabric and hard logic coming from the FPGA companies. Instead of putting FPGA fabric in our custom SoC designs, we are getting custom SoC in our FPGAs.

Today's FPGA are hybrid parts with optimized hardwired cells (like multipliers, memory, IO and even processors) and FPGA fabric living on the same die.

It is an important factor that there is an order of magnitude more software engineers than hardware guys. Usually it is the software community that selects the processor, not the hardware team. To gain the approval of software engineers the FPGA vendors realized that promoting "a processor with FPGA accelerators" is more attractive than an "FPGA with a processor inside"

The FPGA business and the processor business looks like "chip" business, but actually are more "tools, software, IP, and services" businesses. With the coming together of the embedded processing world [111] and the FPGA world, we will see if FPGA companies like Xilinx can be convincing enough in their ability to support the embedded software developer, or if companies like Intel can be convincing enough in their ability to support the FPGA designer.

"Customers are increasingly turning to FPGAs and expert 3rd party providers to design progressively complex products within shrinking time to market budgets. "As the industry is developing more complex designs on programmable solutions, competent and trusted providers are required to deliver key IP, software, and services to meet the tight delivery schedules of today's system companies and to allow customers to find the right qualified 3rd party provider easier and faster than before - avoiding, that for SoC designers it's a nightmare using IPs delivered by 3rd parties or internal IP teams? An Electronics IP core, a semiconductor intellectual property core, or IP block is a reusable unit of logic, cell, or chip layout design that is the (legal) intellectual property of one party.

2.3.3 Problems We Must Solve

Furthermore this chapter outlines the educational barriers we have to surmount and the urgent need for major funding on a global scale to run a world-wide mass movement, of a dimension as far reaching as the Mead-&-Conway-style microelectronics revolution in the early 80ies. Problems We Must Solve:

- (1.) A mass migration from software to configware for the benefit of massively saving energy, of much higher performance, and, of gaining high flexibility.
- (2.) developing a most promising migration priority list.
- (3.) to reeducating the programmer population for such a mass movement campaign [68], and upgrading our highly obsolete curricula for three reasons:
 - (a.) to realize that parallel programming qualifications are a must,
 - (b.) to resolve the extreme shortage of programmers qualified for RC, and
 - (c.) twin paradigm programming skills are a must to program hetero systems (like modern FPGAs featuring all 3: reconfigurable fabrics, hardwired function blocks, and CPUs).

As a consequence we need innovative undergraduate programming courses [98] which also teach a sense for locality, not only needed for classical parallel programming, is already coming along in RC with time to space mapping required to structurally map an application to the datastream side of the twin paradigm approach. This means, that teaching the structural programming of RC also exercises the sense of locality needed for traditional parallel programming. The extension of the non-sequential part of education should be optimized not to scare away undergraduate students. Twin-paradigm lab courses should be model-based, may be MathWorks-supported, mainly at the abstraction level of pipe networks [99].

2.3.4 How to introduce Reconfigurable Computing

Since software has to be rewritten anyway, this is the occasion for the twin-paradigm approach to massively reduce the energy consumption of our computing infrastructures.

Meanwhile FPGAs are also used everywhere for high performance in scientific computing, where this is really a new computing culture - not at all a variety of hardware design. Instead of H/S codesign we have here software/configware co-design (SC co-design), which is really a computing issue. This major new direction of developments in science will determine how academic computing will look in 2015 or even earlier. The instruction-stream-based mind set will loose its monopoly-like dominance and the CPU will quit its central role - to be more an auxiliary clerk, also for software compatibility issues.

An introduction to Reconfigurable Computing (RC) [90,112,113] should regard the background to be expected from the reader. This chapter of the book mainly addresses a bit IT-savvy people in the public and its mass media, as well as "software engineers". Here an introduction is difficult, since in both communities people typically know nothing or almost nothing about RC. To move RC from its niche market into mainstream massive funding is needed for R&D and to reinvent programming education. To yield the attention of media and the politicians we need a highly effective campaign by mass media.

RC should urgently become mainstream. Several reasons have prevented RC from truly becoming mainstream [114]. The execution model is inherently different from the traditional sequential paradigm were we can reason about state transition sequences much better than in a hardware or a concurrent execution model. As a consequence, the development and validation of tools is substantially a traditional hardware mind set

Tools are limited and above all fairly bridle. This means programmers must master the details of not only software development but also of hardware design. Such a set of skills is also not taught as part of major electrical engineering courses severely constraining the pool of engineering with the "right" mindset for programming RC to a selected few. Moreover the recent evolution of FPGAs and to some extent coarsegrain RC architecture make programmer and performance portability difficult at best.

One of the objectives of the REFLECT project (chapter 11) is lowering the barrier of access of RC to the average programmers, by retaining the "traditional" imperative programming mindset in a high-level language such as MATLAB and rely on the concepts of Aspects to provide a clean mechanism (at the source code level) for the advanced user to provide key information for a compilation and synthesis tool to do a good job in mapping the computation to hardware. The approach should be by no means fully automatic [114]. Instead, we have the programmer involved but controlling the high-level aspects of the mapping while the tools takes care of the low-level, error-prone steps.

We extend the "traditional" imperative programming mindset (for software) by a twin-paradigm imperative mind (subject of subsection 2.4.4.) also including an imperative datastream programming methodology (for "flowware" - for terminology see Table 3) [90]. We obtain an almost fully symmetric methodology: the only asymmetry is intra-loop parallelism, possible for data streams, however not for instruction streams (Table 4). The semantic difference of these machine paradigms is the state register: the program counter (located with the ALU) for running the instruction streams in executing software, and data counter(s) (located in memory block(s) [60, 61]) for running data streams in executing flowware.

2.3.5 Toward a New World Model of Computing

The traditional CPU-centric world model of the CS world is obsolete. It resembles the old Aristotelian geo-centric world model. Its instruction-stream-based software-only tunnel view perspective hides structural and data stream aspects - massively threatening the progression of system performance, where we have to confront a dramatic capability gap. We need a generalized view, comparable to the Copernican world model not being geo-centric. We need a hetero model which also includes structures and data streams and supports time to space mapping, since scaling laws favor reconfigurable spatial computing over temporal computing. Exercising time to space mapping, also by programming data streams and by software to configware migration, provides important skills: e. g. locality awareness, understanding and designing efficient

manycore architectures and their memory organization being essential to cope with bottlenecks caused by bandwidth problems.

This new direction has not yet drawn the attention of the curriculum planner within the embedded systems scene. For computer science this is the opportunity of the century, of decampment for heading toward new horizons, and, to preserve the affordability of its electricity consumption. This should be a wake-up call to CS curriculum development. Each of the many different application domains has only a limited view of computing and takes it more as a mere technique than as a science on its own. This fragmentation makes it very difficult to bridge the cultural and practical gaps, since there are so many different actors and departments involved. We need the new CS world model to avoid the capability gap caused by that fragmentation. Computer Science should take the full responsibility to merge Reconfigurable Computing into CS curricula for providing Reconfigurable Computing Education from its roots. CS has the right perspective for a trans-disciplinary unification in dealing with problems, which are shared across many different application domains. This new direction would also be helpful to reverse the current down trend of CS enrolment.

Not only for the definition of the term "Reconfigurable Computing" (RC), it makes sense to use a clear terminology – not only to improve education about how to reinvent computing. It is a sluttish use of terms if "soft" or "software" is used for everything, which is not hardware. The term "software" should be used only for instruction streams and their codes. However, we generalize the term "programming" (Fig. 6) such, that procedural programming (in time domain) creates sequential code, like instruction streams (software), or data streams, which we call "flowware", and, that "structural programming" (programming in space) creates "structural code", which we call "configware", since it can be used for the configuration of FPGAs (Field-Programmable Gate Arrays) or other reconfigurable platforms. Summary: Table 3.

This established terminology reveals (see Table 3 for the terms we should use and Table 4 for the terms that usually make some confusion), that a software to configware migration means a paradigm shift, away from the traditional programmer's CPU-centric world model of computing, resembling the geo-centric Aristotelian world model. To reinvent computing we need a multi paradigm hetero system world model of computing science (Fig. 9), which models the co-existence of, and the communication between: (1.) the traditional imperative software programming language mind set with the CPUs running by software (instruction streams), (2.) the reconfigurable modules to be structurally programmed by configware, and (3.) an imperative datastream programming language mind set with [90] data stream machines programmed by flowware for generating and accepting data streams (asM in Table I stands for "auto-sequencing Memory", also containing the data counter inside a reconfigurable address generator). We obtain an almost fully symmetric methodology: the only asymmetry is intra-loop parallelism, possible for data streams, however not for instruction streams (Table 4). The semantic difference of these machine paradigms is the state register: the program counter (located with the ALU) for running the instruction streams in executing software, and the data counter(s) (located in memory block(s) [60-66]) for running data streams in executing flowware

Table 4. Confusing terms which should not be used.

term	once introduced for
dataflow	indeterministic exotic machines
firmware microcode microprogram	nested von Neumann machines
software or "soft"	no use other than for instruction streams

Fig. 10 illustrates this triple-paradigm "Copernican" world model replacing the von-Neumann-only-centric obsolete "Aristotelian" narrow tunnel view perspective of classical software engineering, which hides almost everything, which is not instruction-stream-based. (The term "supersystolic" in Fig. 10 stands for the generalization of the systolic array: non-linear and non-uniform pipes are allowed like spiral, zigzag and any excessively irregular shapes.) This generalized model will help us to come up with a new horizon of programmer education which masters overcoming the hardware / software chasm, having been a typical misconception of the ending first history of computing. The impact is a fascinating challenge to reach new horizons of research and development in computer science. We need a new generation of talented innovative scientists and engineers to start the beginning second history of computing, not only for the survival of our important computer-based cyber infrastructures, but also for developing and integrating exciting new innovative products for the transforming post PC era global information and communication markets [90].

Not yet discussed in this paper is the accelerator role of GPUs (graphics processors [115]) which for some authors seem to be the FPGA's competitor w. r. to speed-up and power efficiency [116-118]. Meanwhile the very busy hype on the accelerator use of GPGPU seems to be over-exaggerated [117]. FPGAs from a new Xilinx 28nm high-performance, low-power process, developed by Xilinx and TSMC-optimized for high performance & low power are massively better off than GPUs. E.g., for the Smith-Waterman algorithm the following normalized performance is reported: 584 for FPGA, 25 for GPU, and, 1 for GPP [119]. Since a compute-capable discrete GPU can draw much more than 200 watts, other authors call this massive power draw a serious roadblock to the adoption, not only in embedded systems, but even for data centers [120].

But going hetero by interweaving instruction stream parallelism and structural parallelism is a massive challenge requiring to master many difficult problems. The existence of thousands of languages did not prevent a stall of language research in the past 2 decades [104]. Being speaker in 7 tutorials at Supercomputing 2010 [121] Tim Mattson of Intel complaints about what he calls "choice overload" and calls to arms. Another design tool problem is hitting the moving target of the complex value chain in SoC design: the rapidly growing segment of the electronics industry called "Electronics Intellectual Property" or "Electronics IP", where currently the designers have a nightmare using IPs delivered by 3rd parties or internal IP teams [122]. Masses of highly qualified new kinds of jobs must be created to meet the fascinating challenges of reinventing computing sciences, following the wide horizon of the new world model [121].

2.4 Conclusions

This chapter has emphasized that Reconfigurable Computing (RC) is a critical survival issue for computing-supported infrastructures worldwide and has stressed the urgency of moving RC from niche to main-stream. Since a qualified programmer population does not exist we need to use Reconfigurable Computing to Reinvent Computing (R2R) and to Rewrite Textbooks for R2R (RT4R2R), and many of us should become Reinvent Computing Evangelists (RCE). We urgently need a world-wide mass movement of R&D and education to be more massively funded and supported than the Mead-&-Conway VLSI design revolution in the early 80's, which so far has been the most effective project in the history of modern computing science. This chapter urges acceptance of the massive challenge of reinventing computing, away from its currently obsolete CPU-processor-centric Aristotelian CS world model, to a twin-paradigm Copernican model.

For energy cost reasons, a massive software to configware migration campaign is needed. First this requires clever planning to optimize all its aspects. We also need to develop plans deciding, which software packets need to be migrated, and, which of them should be migrated first. All this requires many years, probably a decade of massive R&D and education efforts. We cannot afford to hesitate. Lobbying for the

massive funding should be started right now. We should address politicians at all levels: community level, state level, national level, and European Union level. To explain all this to politicians is very, very difficult. Since politicians always watch the sentiment of their voter population, we efficiently have to teach the public, which is a dramatic challenge. Without the support by a strong tailwind from the media a successful lobbying does not seem to have any chance. All this has to be completed as soon as possible, as soon as we can still afford such massive activities. To succeed with such a challenging educational campaign the foundation of a powerful consortium to be funded at all levels is needed for running an at least Europe-wide project.

2.5 References

- 1. Clive Maxfield: Reproducing a Cray 1 supercomputer in a single FPGA; Sep 3, 2010, EETimes2
- http://www.eetimes.com/design/programmable-logic/4207359/Reproducing-a-Cray-1-supercomputer-in-a-single-FPGA
- 3. R. Hartenstein: The Grand Challenge To Reinvent Computing; XXX Congress of the SBC, 20-23 July 2010, Belo Horizonte, MG, Brazil4
- 4. http://www.inf.pucminas.br/sbc2010/anais/pdf/semish/st03 02.pdf
- 5. Raj Jammy (keynote): Emerging Low Power Technologies: CMOS or Beyond CMOS? 2010 Int'l Symp. on Low Power Electronics and Design (ISLPED), Austin, TX, USA, Aug 18-20, 2010
- 6. Hamed Dadgour, Kaustav Banerjee: A New Paradigm in the Design of Energy-Efficient Digital Circuits Using Laterally-Actuated Double-Gate NEMS; 2010 International Symposium on Low Power Electronics and Design (ISLPED), Austin, TX, USA August 18-20, 2010
- 7. Dimin Niu et al.: Low-power Dual-element Memristor-Based Memory Design; 2010 Int'l Symp. on Low Power Electronics and Design (ISLPED), Austin, TX, USA Aug 18-20, 2010
- 8. Ajith Amerasekera (keynote): Ultra Low Power Electronics in the Next Decade; 2010 Int'l Symp. on Low Power Electronics and Design (ISLPED), Austin, TX, USA Aug 18-20, 2010
- 9. Celeste LeCompte: Green Data Centers; March 11, 2010, GigaOM (BusinessWeek)
- 10. Olga Kharif: Green Data Centers Blossom; May 30, 2007, Business Week
- 11. Fei Li Yan Lin, Lei He, Jason Cong: Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics; 2004 ACM/SIGDA 12th int'l symp. on FPGAs, 2004, Monterey, CA, USA
- 12. Mark Guzdial, Judy Robertson: Too Much Programming too soon? March 2010, Comm_ACM
- 13. Mark Guzdial: How We Teach Introductory Computer Science is Wrong; http://cacm.acm.org/blogs/blog-cacm/45725
- 14. Susan Albers: Energy-efficient Algorithms; May 2010, Comm ACM
- 15. Joao M. P. Cardoso: New Challenges in Computer Science Education; 10th ann. SIGCSE conf. on Innovation and technology in computer science education, June 27-29, 2005, Monte de Caparica, Portugal
- V. A. Chandrasetty, S. R. Laddha: A Novel Dual Processing Architecture for Implementation of Motion Estimation Unit of H.264 AVC on FPGA", in IEEE Symposium on Industrial Electronics and Applications, 2009, Kuala Lumpur, Malaysia
- 17. Bruce Nussbaum: The World after Oil; June 29, 1984, Touchstone
- 18. Th. Hartmann, G. Kretzschmar: Unser ausgebrannter Planet: Von der Weisheit der Erde und der Torheit der Moderne; March 1, 2000, Riemann Verlag
- 19. Jeremy Leggett: Carbon War: Global Warming and the End of the Oil Era; April 1, 2001, Routledge
- Stanley Donaldson: Giant Earthquake will drain world's oil supplies like a flushing toilet; Dec 24, 2002, Weekly World News
- 21. C. J. Campbell: The Coming Oil Crisis; April 1, 2004, Multi-Science Publishing Co. Ltd.
- 22. R. Heinberg: Powerdown: Options and Actions for a Post-Carbon World; Sep 1, 2004, New Society
- 23. M. C. Ruppert, C. A. Fitts: Crossing the Rubicon: The Decline of the American Empire at the End of the Age of Oil; October 1, 2004, New Society Publishers
- 24. David Goodstein: Out of Gas: The End of the Age Of Oil; Feb 17, 2005, W. W. Norton & Company
- 25. Vaclav Smil: Energy at the Crossroads: Global Perspectives and Uncertainties; Apr 1, 2005, The MIT Press
- 26. Paul Roberts: The End of Oil: On the Edge of a Perilous New World; April 5, 2005, Mariner Books

- 27. Eva Marbach: Jenseits des Ölgipfels: Ein Peak-Oil-Roman; May 5, 2005, Verlag Solare Zukunft
- 28. R. Heinberg: The Party's Over: Oil, War and the Fate of Industrial Societies; June 1, 2005, New Society
- 29. Roger D. Blanchard: The Future of Global Oil Production: Facts, Figures, Trends And Projections, by Region; November 30, 2005, McFarland & Company
- 30. J. H. Kunstler: The Long Emergency: Surviving the End of Oil, Climate Change, and Other Converging Catastrophes of the Twenty-First Century; March 2, 2006, Grove Press
- 31. Wolfgang Gründinger: Die Energiefalle: Ein Rückblick auf das Erdölzeitalter; March 15, 2006, Beck
- 32. Anthony H. Cordesman, Khalid R. Al-Rodhan: The Global Oil Market: Risks And Uncertainties; March 20, 2006, Center for Strategic & Int'l Studies
- 33. Sonia Shah: Crude: The Story of Oil; May 1, 2006, Seven Stories Press
- 34. Leonardo Maugeri: The Age of Oil: The Mythology, History, and Future of the World's Most Controversial Resource; May 30, 2006, Praeger
- 35. Oystein Noreng: Crude Power: Politics and the Oil Market; May 30, 2006, I. B. Tauris
- 36. Matthew R. Simmons: Twilight in the Desert: The Coming Saudi Oil Shock and the World Economy; June 5, 2006, Wiley
- 37. Kenneth S. Deffeyes: Beyond Oil: The View from Hubbert's Peak; June 13, 2006, Hill and Wang
- 38. D. A. Pfeiffer: Eating Fossil Fuels: Oil, Food and the Coming Crisis in Agriculture; Oct 1, 2006, New Society
- 39. Robert Mabro: Oil in the Twenty-First Century: Issues, Challenges, and Opportunities; November 9, 2006, Oxford University Press, USA
- 40. Matthew R. Simmons: Wenn der Wüste das Öl ausgeht: Der kommende Ölschock in Saudi-Arabien Chancen und Risiken; Dec 14, 2006, FinanzBuch Verlag
- 41. C. J. Campbell, F. Liesenborghs, J. Schindler, W. Zittel, H. Roth: Ölwechsel!: Das Ende des Erdölzeitalters und die Weichenstellung für die Zukunft; Jan 1, 2007, DTV
- 42. Stephen Leeb, Glen Strathy: The Coming Economic Collapse: How You Can Thrive When Oil Costs \$200 a Barrel; February 21, 2007, Business Plus
- 43. Karen R. Merrill: The Oil Crisis of 1973-1974; February 22, 2007, Bedford/St. Martin's
- 44. P. Tertzakian: A Thousand Barrels a Second: The Coming Oil Break Point and the Challenges Facing an Energy Dependent World; May 9, 2007, McGraw-Hill
- 45. Wade Adams, Abdul Samad Al-Awadi, Ray McCormack, Basil Gelpke: A Crude Awakening The Oil Crash; DVD, July 31, 2007, DOCURAMA
- 46. R. Heinberg: Peak Everything: Waking Up to the Century of Declines; Oct 16, 2007. New Society
- 47. L. Maugeri: The Age of Oil: What They Don't Want You to Know About the World's Most Controversial Resource; November 1, 2007, The Lyons Press
- 48. David Strahan: The Last Oil Shock: 2007, John Murray
- 49. J. Moroney: M. Simmons: Power Struggle: World Energy in the Twenty-First Century; May 30, 2008, Praeger
- 50. Jürgen Petermann: Sichere Energie im 21. Jahrhundert; June 2008, Hoffmann und Campe
- 51. A. Raggam, K. Faißner: Zukunft ohne Öl: Lösungen für Verkehr, Wärme und Strom; Aug 1, 2008, Stocker
- 52. Eva Marbach: Peakoil Reloaded; Aug 26, 2008, Marbach, Eva Verlag
- 53. Robin M. Mills: The Myth of the Oil Crisis: Overcoming the Challenges of Depletion, Geopolitics, and Global Warming; August 30, 2008, Praeger
- 54. John Michael Greer: The Long Descent: A User's Guide to the End of the Industrial Age; Sep 2008, New Society
- 55. Peter Gruss: Die Zukunft der Energie: Die Antwort der Wissenschaft. Ein Report der Max-Planck-Gesellschaft; Sep 22, 2008, C.H. Beck
- 56. R. Heinberg, M. Bayer, B. Hickisch: Öl-Ende: "The Party's Over" Die Zukunft der industrialisierten Welt ohne Öl; Sep 22, 2008, Riemann Verlag
- 57. Peter Gruss: Die Zukunft der Energie: Die Antwort der Wissenschaft. Ein Report der Max-Planck-Gesellschaft; Sep 22, 2008, C.H. Beck
- 58. Sascha Müller-Kraenner: Energy Security; October 2008, Earthscan Publications Ltd.
- David G. Victor, Amy M. Jaffe, Mark H. Hayes: Natural Gas and Geopolitics: From 1970 to 2040; October 14, 2008, Cambridge University Press
- 60. Andreas Eschbach: Ausgebrannt: Thriller; Dec 18, 2008, Bastei Verlag
- 61. Thomas Seifert, Klaus Werner: Schwarzbuch Öl; Sep 1, 2008, Ullstein
- 62. D. Clarke: The Battle for Barrels: Peak Oil Myths & World Oil Futures; Feb 1, 2009, Profile Books
- 63. Cathal Kelly: Take Peak Oil seriously it'll be here much sooner than you think; Feb 15 2009, Toronto Star
- 64. R. Bryce: Gusher of Lies: Dangerous Delusions of "Energy Independence"; March 2, 2009, Public Affairs
- 65. J. W. Rawles: Patriots: A Novel of Survival in the Coming Collapse; April 7, 2009; Ulysses Press

- 66. Thom Hartmann, William Rees, James Jandak Wood: Crude Impact; DVD, Apr 14, 2009, DOCURAMA
- 67. Werner Zittel, Jörg Schindler: Geht uns das Erdöl aus?: Wissen was stimmt; 5. May 5, 2009, Herder V.
- 68. Jeff Rubin: Why Your World Is About to Get a Whole Lot Smaller; May 19, 2009, Random House
- 69. P. Tertzakian, K. Hollihan: The End of Energy Obesity: Breaking Today's Energy Addiction for a Prosperous and Secure Tomorrow; June 29, 2009, Wiley
- 70. Richard Heinberg: Blackout Coal, Climate and the Last Energy Crisis; July 1, 2009, New Society
- 71. H.-J. Zillmer: Der Energie-Irrtum: Warum Erdgas und Erdöl unerschöpflich sind; July 7, 2009, Herbig
- 72. Ch. Steiner: \$20 Per Gallon: How the Inevitable Rise in the Price of Gasoline Will Change Our Lives for the Better; July 16, 2010, Grand Central Publishing
- 73. Peter Maass: Crude World: The Violent Twilight of Oil; September 22, 2009, Knopf
- 74. Vandana Shiva: Leben ohne Erdöl: Eine Wirtschaft von unten gegen die Krise von oben; Sept 28, 2009, Rotpunktverlag, Zürich
- 75. J. W. Rawles: How to Survive the End of the World as We Know It: Tactics, Techniques, and Technologies for Uncertain Times; September 30, 2009, Plume
- 76. J. M. Greer: The Ecotechnic Future: Envisioning a Post-Peak World; Oct 1, 2009, New Society Publishers
- 77. Jörg Schindler, Martin Held, Gerd Würdemann: Postfossile Mobilität: Wegweiser für die Zeit nach dem Peak Oil; Nov 28, 2009, Vas-Verlag für Akademische Schriften
- 78. Michael C. Ruppert, Colin Campbell: Confronting Collapse: The Crisis of Energy and Money in a Post Peak Oil World; December 15, 2009, Chelsea Green Publishing
- 79. Carlos Pascual, Jonathan Elkind: Energy Security: Economics, Politics, Strategies, and Implications; December 29, 2009, Brookings Institution Press
- 80. George Friedman: The Next 100 Years: A Forecast for the 21st Century, January 26, 2010, Anchor
- 81. Leonardo Maugeri: Beyond the Age of Oil: The Myths, Realities, and Future of Fossil Fuels and Their Alternatives; February 26, 2010, Praeger
- 82. Joseph Stiglitz: Freefall; Oct 2010, Penguin
- 83. Meinhard Miegel: Exit: Wohlstand ohne Wachstum; March 2, 2010, Propyläen
- 84. Jeff Rubin, Karsten Petersen. Warum die Welt immer kleiner wird: Öl und das Ende der Globalisierung; March 4, 2010, Hanser
- 85. Jeff Rubin: Why Your World is About to Get a Whole Lot Smaller: Oil and the End of Globalisation; March 4, 2010, Virgin Books
- 86. Michael Lewis: The Big Short; March 15, 2010, Penguin
- 87. Michael Lewis: The Big Short: Inside the Doomsday Machine; March 15, 2010, W. W. Norton & Co
- 88. Robert Bryce: Power Hungry: The Myths of "Green" Energy and the Real Fuels of the Future; April 27, 2010, Public Affairs
- 89. Hansjakob Ginsburgh et al.: Angst vor dem Dominoeffekt; Apr 26, 2010, Wirtschaftswoche
- 90. Hermann Maurer: Das Paranetz Zusammenbruch des Internet; Roman: 2004, Freya Verlag
- 91. Hermann Maurer: The Paranet; novel: 2004, Freya Verlag
- 92. Nouriel Roubini, Stephen Mihm, Jürgen Neubauer, Petra Pyka: Das Ende der Weltwirtschaft und ihre Zukunft: Crisis Economics; May 11, 2010, Campus Verlag
- 93. Michael Ruppert: Collapse; DVD, June 15, 2010, MPI HOME VIDEO
- 94. Tom Bower: Oil: Money, Politics, and Power in the 21st Century; June 3, 2010, Grand Central
- 95. Ron Rhodes: The Coming Oil Storm: The Imminent End of Oil...and Its Strategic Global Role in End-Times Prophecy; July 1, 2010, Harvest House Publishers
- 96. T. El-Ghazawi et al.: "The promise of high-performance reconfigurable computing," IEEE Computer, vol.41, no.2, pp.69-76, Feb. 200897
- 97. http://ite.gmu.edu/~kgaj/publications/journals/GWU GMU UIUC USC Computer 2008.pdf
- 98. C. Bobda: Introduction to Reconfigurable Computing: Architectures; Springer, 2007
- 99. C. Chang et al: The Biggascale Emulation Engine (Bee); summer retreat 2001, UC Berkeley
- 100. Reiner Hartenstein: Morphware and Configware; chapter 11 in: Albert Zomaya (editor): Handbook of Nature-Inspired and Innovative Computing; 2005, Springer
- 101. Reiner Hartenstein: Basics of Reconfigurable Computing; ch. 20 in: J. Henkel, S. Parameswaran (editors): Embedded Computing A Low Power Perspective, Springer-Verlag, 2007
- 102. P. Wallich: Profile: Lynn Conway—Completing the Circuit; Scientific American Magazine, Dec 2000.
- 103. L. Conway: The MPC Adventures: Experiences with the Generation of VLSI Design and Implementation Methodologies; Microprocessing and Microprogramming The Euromicro Journal, Vol. 10, No. 4, Nov 1982104
- 104. http://ai.eecs.umich.edu/people/conway/VLSI/MPCAdv/MPCAdv.pdf

- 105. National Research Council: Funding a Revolution; National Academies Press; January 8, 1999
- 106. N. N.: "The book that changed everything", Electronic Design News, Feb. 11, 2009
- 107. D. Kilbane: Lynn Conway A trailblazer on professional, personal levels; El. Design, Oct, 2003
- 108. http://partners.academic.ru/dic.nsf/dewiki/362215
- 109. C. Mead, L. Conway: Introduction to VLSI Systems; Addison-Wesley, 1980
- 110. L. Conway: The MIT '78 VLSI System Design Course; August 1979, Xerox PARC
- 111. R. Hon, C. Sequin: A Guide to LSI Implementation; September 1978, Xerox PARC
- 112. L. Conway, A. Bell, M. Newell: MPC79: A Large-Scale Demonstration of a New Way to Create Systems in Silicon; LAMBDA, the Magazine of VLSI Design, 2nd Quarter 1980
- 113. http://lexikon.freenet.de/Mikrochip-Entwurfs-Revolution
- 114. R. Hartenstein (invited paper): Reconfigurable Computing: Paradigmen-Wechsel erschüttern die Fundamente der Informatik; Prof. Glesners 60th birthday workshop, Aug 29, 2003 Darmstadt, Germany. http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/HartensteinGlesner60.pdf
- 115. http://wiki.answers.com/QWhy are computers important in the world
- 116. E. Denert: Erfahrungen und Einsichten eines Software-Unternehmers; Oct 28, 2004, TU Kaiserslautern.
- 117. Jürgen Teich: Invasive Computing Basic Concepts And Forseen Benefits; ARTIST Summer School Europe 2010, September 5-10, 2010, Autrans, France
- 118. Edward A. Lee: Cyber-Physical Systems Are Computing Foundations Adequate?; Position Paper for NSF Workshop On Cyber-Physical Systems; Oct 16 17, 2006, Austin, TX
- 119. N. N.: HPC Technology Makes Car Safety Job 1; HPCwire, August 05, 2010120
- 120. http://www.hpcwire.com/news/HPC-Technology-Makes-Car-Safety-Job-1-100088164.html
- 121. P. Cowhey, J. Aronson: Transforming Global Information and Communication Markets, MIT Press, 2009
- 122. Rachael King: Big Tech Problem as Mainframes Outlast Workforce; August 2, 2010, Bloomberg Business Week
- 123. Rebecca Scudder: Definitions in Green Computing; Jan 29, 2010 BRIGHT HUB124
- 124. http://www.brighthub.com/environment/green-computing/articles/62742.aspx
- 125. R. Katz: Tech Titans Building Boom Google, Microsoft, and other Internet giants race to build the mega data centers that will power cloud computing; IEEE Spectrum, February 2009 126
- 126. http://www.spectrum.ieee.org/feb09/7327
- 127. Pelamis Wave Power Ltd., Edinburgh, Scotland http://www.pelamiswave.com
- 128. Rebecca Scudder: Top Rated Energy Efficient Computer Monitors; Mar 7, 2010, BRIGHT HUB129
- 129. http://www.brighthub.com/environment/green-computing/reviews/65304.aspx#ixzz0vzotrQYK
- 130. Peter Marwedel, Michael Engel: Plea for a Holistic Analysis of the Relationship between Information Technology and Carbon-Dioxide Emissions; ARCS 2010, Feb 22-25, 2010, Hannover, Germany131
- 131. http://ls12-www.cs.tu-dortmund.de/publications/papers/arcs-10-marwedel.pdf
- 132. Andy Hopper: Computing for the Future of the Planet; 2008, Trans. R. Soc. A 366(1881)133
- 133. http://www.cl.cam.ac.uk/research/dtg/~ah12/CFP%20paper.pdf
- 134. N. N.: Davos: rethink, redesign and rebuild; 27 January 2010. Channel 4 News135
- 135. http://www.channel4.com/news/articles/business money/davos+rethink+redesign+and+rebuild/3517427
- 136. Don Tapscott: The Dubai Summit On Redesigning Global Cooperation And Problem Solving; Nov 26, 2009, The Huffington Post137
- 137. http://www.huffingtonpost.com/don-tapscott/the-dubai-summit-on-redes_b_371730.html
- 138. Giovanni De Micheli: Nano-systems: Devices, Circuits, Architectures and Applications; ARTIST Summer School Europe 2010, Autrans, France
- 139. Claudio Brunelli, Fabio Campi, Damien Picard, Fabio Garzia, Jari Nurmi: Reconfigurable Hardware: The Holy Grail of Matching Performance with Programming Productivity; 18th Int'l Conference on Field Programmable Logic and Applications (FPL 2008), Sep 8 10, 2008, Heidelberg, Germany
- 140. Syed Zahid Ahmed, Gilles Sassatelli, Lionel Torres, Laurent Rougé: Survey of new trends in Industry for Programmable hardware: FPGAs, MPPAs, MPSoCs, Structured ASICs, eFPGAs and new wave of innovation in FPGAs; 20th International Conference on Field Programmable Logic and Applications (FPL 2010), Aug 31- Sep 2, 2010, Milano, Italy141,142
- 141. http://www.lirmm.fr/~ahmed/files/FPL10/Zahid FPL10 SurveyPaper.pdf
- 142. http://conferenze.dei.polimi.it/FPL2010/presentations/W1 B 1.pdf
- 143. Syed Zahid Ahmed, JulienEydoux, Laurent Rougé, Jean-Baptiste Cuelle, Gilles Sassatelli, Lionel Torres: Exploration of power reduction and performance enhancement in LEON3 processor with ESL reprogrammable eFPGAin processor pipeline & as a co-processor; The Design, Automation, and Test in Europe (DATE) conference, April 20 24, 2009, Nice, France 144, 145

- 144. http://www.date-conference.com/proceedings/PAPERS/2009/DATE09/PDFFILES/03.3 4.PDF
- 145. http://www.lirmm.fr/~ahmed/files/DATE09/DATE09 SZA 03.3 4 slides.pdf
- 146. Robert N. Charette: Why Software Fails; IEEE Spectrum, Sep 2005
- 147. J. Garlick, Will the Internet Break? ISP Economics Assessment to 2012; Oct 2008, Screen Digest
- 148. P. Somavat, S. Jadhav, and V. Namboodiri: Accounting for the Energy Consumption of Personal Computing Including Portable Devices; e-Energy 2010, April 13-15, 2010, Passau, Germany
- 149. Wu-chun Feng: On the Second Coming of Green Destiny?; International Conference on Energy-Aware HPC; Sep 16–17, 2010, Hamburg, Germany150
- 150. http://ena-hpc.org/index.html
- 151.Peter Kogge (editor): ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems; Sep 28, 2008 http://www.cse.nd.edu/Reports/2008/TR-2008-13.pdf
- 152. Stephen S. Pawlowski: Exascale Science The next frontier in High Performance Computing; 24th International Conference on Supercomputing, June 1-4, 2010, Tsukuba, Japan
- 153. G. Fettweis: ICT Energy Consumption Trends and Challenges; WPMC'08, Lapland, Finland, Sep 8 -11, 2008154
- 154. http://www.vodafone-chair.com/publications/2008/Fettweis G WPMC 08.pdf
- 155. P. Ross: Cloud Computing's Killer App: Gaming; IEEE Spectrum, March 2009
- 156.D. Estrin et al: "Internet Predictions," IEEE Internet Computing, vol. 14, no. 1, pp. 12-42, Jan./Feb. 2010, doi:10.1109/MIC.2010.12
- 157. G. Da Costa, et al.: Multi-Facet Approach to Reduce Energy Consumption in Clouds and Grids: The GREEN-NET Framework; e-Energy 2010, Apr 13-15, 2010, Passau, Germany
- 158. Niels Boeing: Wie bändigt man ein Energiemonster?; Oct 26, 2007, Heise Technology Review159
- 159. http://www.heise.de/tr/artikel/Wie-baendigt-man-ein-Energiemonster-280483.html
- 160. R. Nuez: Google is Going into the Electricity Business; The Huffington Post, JUNE 6, 2010161
- 161. http://www.huffingtonpost.com/ramon-nuez/google-is-going-into-the b 417035.html
- 162. http://en.wikipedia.org/wiki/Pelamis_Wave_Energy_Converter
- 163. A. Zomaya et al.: Interweaving Heterogeneous Metaheuristics Using Harmony Search; Int'l Parallel & Distr. Proc. Symp., May 23–29, Rome, Italy
- 164. J. Rabaey: Low Power Design Essentials; Springer Verlag, 2009
- 165. L. A. Barroso: The Price of Performance; Oct 18, 2005, ACMqueue¹⁶⁶
- 166. http://queue.acm.org/detail.cfm?id=1095420
- 167. W. Blendinger: Post Peak Abstieg vom Peak Oil; ASPO annual conference, Berlin, 18 Mai 2010
- 168. K. Aleklett: Post Peak The Future of the Oil-Production; ASPO annual conf., Berlin, 18 Mai 2010
- 169. J. S. Gabrielli de Azevedo: Petrobras e o Novo Marco Regulatório; São Paulo, December 1, 2009
- 170. Stefan Hajek: Dynamik unterschätzt; Wirtschaftswoche 34, August 8, 2010
- 171. F. Birol: Leave oil before it leaves us; THE INDEPENDENT (UK) March 2, 2008172
- 172. http://www.youtube.com/watch?v=m377Is4tGF0
- 173. H. Ginsburg: Unterm Wasser liegt die Zukunft; Wirtschaftswoche 23, June 7, 2010
- 174. Mark P. Mills: The Bottomless Well: The Twilight of Fuel, the Virtue of Waste, and Why We Will Never Run Out of Energy; Basic Books, 2006
- 175. Costas Bekas, Alessandro Curioni: A new Energy Aware Performance Metric; International Conference on Energy-Aware HPC; Sep 16–17, 2010, Hamburg, Germany150
- 176. R. Hartenstein (keynote): Directions of Programming Research: Seeking a Needle in the Haystack? BGME 2010, 1st Brazilian-German Workshop on Micro and Nano Electronics, Oct 6–8, 2010, Porto Alegre, RS, Brazil177
- 177. Abstract at: http://hartenstein.de/Reiners-keynote-Porto-Alegre.pdf
- 178. Steve Conway: Multicore Processing: Breaking through the Programming Wall Big challenges lie on the software side in efficiently corralling hardware's runaway parallelism; Aug 10, 2010, Scientific Computing179
- 179. http://www.scientificcomputing.com/articles-HPC-Multicore-Processing-Breaking-through-the-Programming-Wall-081010.aspx
- 180. Kevin D. Kissell: The Dead Supercomputer Society; http://www.paralogos.com/DeadSuper/
- 181. Karyn R. Ames, Alan Brenner (editors): Frontiers of Supercomputing II: A National Reassessment; February 10, 1994, University of California Press
- 182. Erik Saxe: Power-efficient Software; Feb 2010 Comm ACM
- 183. Mary Jane Irwin: Shared Caches in Multicores: The Good, The Bad, and The Ugly"; 37th ISCA, June 19-23, 2010, Saint Malo. France
- 184. D. Patterson: The Trouble with Multicore; IEEE Spectrum, July 2010
- 185. Patrick Lysaght (keynote): The Programmable Logic Perspective; FPL-2010, Sep 2010, Milano, Italy

- 186. J. Izydorczyk, M. Izydorczyk: Microprocessor Scaling: What Limits Will Hold? Computer, Aug 2010
- 187. N. N. (Sandia National Labs): DARPA selects Sandia National Laboratories to design new supercomputer prototype; August 18, 2010, PHYSORG.com¹⁸⁸
- 188. http://www.physorg.com/wire-news/43576779/darpa-selects-sandia-national-laboratories-to-design-new-superco.html
- 189. M. Ben-Ari: Objects Never? Well, Hardly Ever! Comm. of the ACM vol 53, no. 9 (Sep 2010)
- 190. T. Ohtsuki, Albert Ruehli, Reiner Hartenstein: Circuit analysis, simulation, and design; 1987, Elsevier
- 191. Reiner Hartenstein (editor): Hardware Description Languages; 1986, Elsevier
- 192. Melvin Breuer, Reiner Hartenstein (editors): Computer Hardware Description Languages and their Applications; 1981, Elsevier
- 193. Reiner Hartenstein: Fundamentals of Structured Hardware Design; 1977, Elsevier
- 194. D. Selwood: Showing off in San Jose; Embedded Technology J., June 8, 2010
- 195. Bill Dally (keynote): Moving the needle: Effective Computer Architecture Research in Academy and Industry; 37th ISCA, June 19-23, 2010, Saint Malo, France
- 196. R. Hartenstein (invited paper): The von Neumann Syndrome; Stamatis Vassiliadis Memorial Symp. "The Future of Computing",, Sep 28, 2007, Delft, NL197
- 197. http://helios.informatik.uni-kl.de/staff/hartenstein/Hartenstein-Delft-Sep2007.pdf
- 198. T. Mattson, M. Wrinn: Parallel Programming: Can we PLEASE get it right this time? DAC 2008, Anaheim, CA, J
- 199. J. Turley: Whither Embedded Part Deux; Embedded Technology Journal, May 25, 2010200
- 200. http://www.techfocusmedia.net/embeddedtechnologyjournal/feature_articles/20100525-esc2/
- 201. J. Udell: Loose coupling and data integrity; Jon's Radio: Jon Udell's Radio Blog, Aug 6, 2002202
- 202. http://radio.weblogs.com/0100887/2002/06/19.html
- 203. M. Anderson: Understanding Multi-Core in Embedded Systems; The PTR Group, Falls Church VA, USA, June 15, 2010
- 204. N. N. (HPC Advisory Council): HPC Advisory Council Forms Worldwide Centers of Excellence; August 17, 2010, HPCwire
- 205. R. Hartenstein: KressArray: what is the achievement? Generalization of the Systolic Array206
- 206. http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/GeneralizedSystolicArray.pdf
- 207. R. Hartenstein, R. Kress: A Datapath Synthesis System for the Reconfigurable Datapath Architecture; Asia and South Pacific Design Automation Conference, ASP-DAC'95, Nippon Convention Center, Makuhari, Chiba, Japan, Aug./Sept. 1995
- 208. R. Kress: A Fast Reconfigurable ALU for Xputers, Ph. D. Dissertation 1996, Kaiserslautern University of Technology; available as a booklet only.
- 209. S. Y. Kung: VLSI Array Processors; Prentice-Hall, Inc., 1988
- 210. N. Petkov: Systolic Parallel Processing; North Holland Publishing Co, 1992
- 211. http://en.wikipedia.org/wiki/List_of_programming_languages
- 212. J. Larus: Spending Moore's Dividend; Comm ACM, May 2009
- 213. R. Jacobi, M. Ayala-Rincón, L. Carvalho, C. Llanos, R. Hartenstein: Reconfigurable systems for sequence alignment and for general dynamic programming; Genetics and Molecular Research 2005
- 214. M. Ayala, C. Llanos, R. P. Jacobi, R. Hartenstein: Prototyping Time and Space Efficient Computations of Algebraic Operations over Dynamically Reconfigurable Systems Modeled by Rewriting-Logic; ACM TODAES 11(2), 2006
- 215. Fig. 13. in R. Hartenstein: The History of KARL and ABL; in: J. Mermet: Fundamentals and Standards in Hardware Description Languages; Kluwer, 1993216
- 216. http://xputers.informatik.uni-kl.de/staff/hartenstein/karlHistory.html
- 217. N. N. (I forgot the celebrity's name): Computer Magazine, 1970 or 1971
- 218. D. L. Parnas (keynote): Teaching for Change; 10th Conf. on Softw. Engrg Education and Training (CSEET), April 13-16, 1997, Virginia Beach, VA, USA
- 219. R. Hameed et al.: Understanding Sources of Inefficiency in General-Purpose Chips; 37th ISCA, June 19-23, 2010, Saint Malo, France
- 220. T. C. Chen et al., "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder," Circuits and Systems for Video Technology, IEEE Transactions on Circuits and Systems for Video Technology, vol.16, no.6, pp. 673-688, June 2006.
- 221. R. Hartenstein (invited paper): The Microprocessor is no more General Purpose: why Future Reconfigurable Platforms will win; Proc. Int'l Conf. on Innovative Systems in Silicon, ISIS'97, Austin, Texas, USA, Oct 8-10, 1997
- 222. N. Tredennick (keynote): Computing in transition; Nov. 11, 2007, SC 2007: Reno, Nevada, USA

- 223. Reiner Hartenstein: Wozu noch Mikrochips? Einführung in Methoden der Technischen Informatik, 1964, IT Press Verlag, Bruchsal/Chicago
- 224. K. Morris: Kicking a Dead Horse FPGAs Going the Distance Against ASIC; FPGA and Structured ASIC Journal, April 7, 2009 http://www.fpgajournal.com
- 225. N. Conner: FPGAs for Dummies FPGAs keep you moving in a fast-changing world; Wiley, 2008
- 226. K. Paulsson, M. Hübner, J. Becker: Exploitation of dynamic and partial hardware reconfiguration for on-line power/performance optimization; FPL 2008
- 227.J. Becker, M. Hübner (invited talk): Run-time Reconfigurabilility and other Future Trends; SBCCI 2006, Sept 2006, Ouro Preto, Brazil
- 228. Dan Isaacs: De-Mystifying FPGAs for Software Engineers; ESC, Sep 2010, Boston, USA
- 229. E. El-Araby et al.: Exploiting Partial Runtime Reconfiguration for High-Performance Reconfigurable Computing. ACM Trans on Reconfigurable Technology and Systems (TRETS) 1(4): (2009)
- 230. E. El-Araby et al.: Productivity Of High-level Languages On Reconfigurable Computers: An HPC Perspective; ICFPT 2007 Int'l Conf. on Field-Programmable Technology, Dec 2007, Japan
- 231. E. El-Araby et al.: Comparative Analysis Of High Level Programming For Reconfigurable Computers: Methodology And Empirical Study; III Southern Conf. on Programmable Logic (SPL2007), Mar Del Plata, Argentina, Feb 2007
- 232. D. Gajski et al.: A second opinion on data-flow machines and languages; IEEE Computer, Feb. 1982.
- 233. M. Feldman: Reconfigurable Computing Prospects on the Rise; HPCwire, December 3, 2008234
- 234. http://www.hpcwire.com/features/Reconfigurable-Computing-Prospects-on-the-Rise 35498449.html
- 235. Rob Evans: Treat programmable hardware design as a high level system task; Apr 13, 2010, EETimes236
- 236. http://www.eetimes.com/design/programmable-logic/4008929/Treat-programmable-hardware-design-as-a-high-level-system-task
- 237. Shepard Siegel: Applying Open Standards to FPGA IP; 11th Annual Workshop on High Performance Embedded Computing: MIT Lincoln Laboratory, September 18-20, 2007, Lexington, Mass, USA238 ----- paradigm twins --- vN syndrome
- 238. http://www.ocpip.org/uploads/documents/HPEC07 Mercury ssiegel poster v1.pdf
- 239. Ashok K. Sharma: Programmable Logic Handbook: PLDs, CPLDs, and FPGAs; 1998, McGraw-Hill
- 240. M. Herz et al.: Memory Organisation for Stream-based Reconfigurable Computing; IEEE ICECS 2002, Sep 15-18, 2002, Dubrovnik, Croatia241
- 241. http://www.michael-herz.de/publications/AddressGenerators3.pdf
- 242. http://xputer.de/fqa.html#anchor81257
- 243. R. Hartenstein, A. Hirschbiel, M.Weber: A Novel Paradigm of Parallel Computation and its Use to Implement Simple High Performance Hardware; InfoJapan'90, Tokyo, Japan, 1990244
- 244. R. Hartenstein et al. (invited reprint243): A Novel Paradigm of Parallel Computation and its Use to Implement Simple High Performance Hardware; Future Generation Computer Systems, no. 7, pp. 181-198 (North-Holland Publ. Co., 1991/92)
- 245. M. Weber et al.: Automatic Synthesis of Cheap Hardware Accelerators for Signal Processing and Image Preprocessing; 12. DAGM-Symp. Mustererkennung, Oberkochen-Aalen, Germany 1990
- 246. Reiner Hartenstein (invited embedded tutorial): Coarse grain reconfigurable architectures; Asia and South Pacific Design Automation Conference, Jan 30-Feb 2. 2001, Yokohama, Japan
- 247. Reiner Hartenstein (embedded tutorial): A Decade of Reconfigurable Computing: a Visionary Retrospective; Design Automation and Test in Europe (DATE), March 2001, Munich, Germany248
- 248. http://hartenstein.de/date01hartenstein.pdf
- 249. U. Nageldinger: Design-Space Exploration for Coarse Grained Reconfigurable Architectures; Ph.D. diss., 2000, TU Kaiserslautern, Germany250
- 250. http://xputers.informatik.uni-kl.de/papers/publications/NageldingerDiss.html
- 251. N. Voros, R. Nikolaos, A. Rosti, M. Hübner (editors): Dynamic System Reconfiguration in Heterogeneous Platforms The MORPHEUS Approach; Springer Verlag, 2009
- 252. T. El-Ghazawi (keynote): Computing with FPGAs: Where does it stand and what is next? FPGAworld'2010, Sep. 6, 2010, Copenhagen, Denmark
- 253. J. Turley: How Many Times Does CPU Go Into FPGA? Embedded Technology J., June 8, 2010
- 254. D. Selwood: EPP A Platform to Bridge a Gap? Embedded Technology Journal June 8, 2010
- 255. R. Hartenstein, M. Herz, T. Hoffmann, U. Nageldinger: Exploiting Contemporary Techniques in Reconfigurable Accelerators; Proceedings FPL'98, Aug 31- Sep 3, 1998, Tallinn, Estonia256
- 256. http://www.michael-herz.de/publications/paper100.pdf

- 257. S. Reddy G. M, P. C. Reddy: Design and FPGA Implementation of High Speed, Low Power Digital Up Converter for Power Line Communication Systems; European Journal of Scientific Research, Vol.25 No.2 (2009) http://www.eurojournals.com/ejsr_25_2_07.pdf
- 258. N. N. (XILINX): Xilinx Redefines Power, Performance, and Design Productivity with Three New 28 nm FPGA Families; FPGA and Programmable Logic Journal259
- 259. http://www.techfocusmedia.net/fpgajournal/ondemand/2010062501-xilinx/
- 260. http://www.chrec.org/
- 261. R. Baxter et al.: High-Performance Reconfigurable Computing the View from Edinburgh; AHS 2007, Aug 5-8, 2007, Edinburgh, UK
- 262. Wang Yi (keynote): Cache-Aware Scheduling and Analysis for Multicores; CSE '09. Int'l Conf. on Computational Science and Engineering, Aug 29-31, 2009, Vancouver, BC, Canada
- 263. M. Hübner, D. Goehringer, J. Noguera, J. Becker: Fast dynamic and partial reconfiguration Data Path with low Hardware overhead on Xilinx FPGAs; Proc. RAW 2010, Atlanta, USA, April, 2010
- 264. Marco D. Santambrogio (keynote): From reconfigurable architectures to self-adaptive autonomic systems; CSE '09. International Conference on Computational Science and Engineering, Aug 29-31, 2009, Vancouver, BC, Canada
- 265. S. Borkar: Designing reliable systems from unreliable components: The challenges of transistorvariability and degradation; IEEE Micro, 25(6):10–16, 2005.
- 266. Ioannis Sourdis, Georgi Gaydadjev: HiPEAC: Upcoming Challenges in Reconfigurable Computing: in 267
- 267. J. M. P. Cardoso, M. Hübner (editors): Reconfigurable Computing, 2011, Springer Verlag
- 268. Fernanda Lima Kastensmidt, Luigi Carro, Ricardo Reis: Fault-Tolerance Techniques for SRAM-Based FPGAs; Dec 2009, Springer US
- 269. M. Flynn: Accelerating computations using data parallelism; CE TU Delft colloquium, Sept. 2009270
- 270. http://ce.et.tudelft.nl/cecoll/slides/09/0925 Flynn.pdf
- 271. R. Hartenstein: Steinbuch; Neue Deutsche Biographie (NDB) vol. 25, Duncker & Humblot, Berlin 2010
- 272. Bernhard Widrow, Reiner Hartenstein, Robert Hecht-Nielson: 1917 Karl Steinbuch 2005 Eulogy; IEEE Computational Intelligence Society Newsletter, August 2005
- 273. K. Steinbuch: Die Lernmatrix; Kybernetik 1 (1961), S. 36-45
- 274. K. Steinbuch, U. Piske (1963): Learning matrices and their applications" IEEE Trans EC, Dec 1963
- 275. R. Rojas: Neural Networks; Springer Verlag, 1996
- 276. R. Hecht-Nielson: Neurocomputing, Addison-Wesley, 1990;
- 277. C. A. Mead: Analog VLSI and Neural Systems, Addison-Wesley, 1989
- 278. André L. S. Braga, Carlos H. Llanos, Mauricio Ayala-Rincón, Jürgen Becker, Michael Hübner, Diana Göhringer, Jonathan Obie, Reiner Hartenstein: Evaluating Performance, Power Consumption and Resource Utilization of Hardware / Software Artificial Neural Networks; BGME 2010, 1st Brazilian-German Workshop on Micro and Nano Electronics, Oct 6–8, 2010, Porto Alegre, RS, Brazil
- 279. L. Chua LO: Memristor—The missing circuit element. IEEE Tran Circuits Theory 18 (1971)
- 280. R. Stanley Williams: How We Found the Missing Memristor; IEEE Spectrum, Dec. 2008
- 281. R. Hartenstein: Why we need Reconfigurable Computing Education; 1st Int'l Workshop on Reconfigurable Computing Education (RC education 2006), March 1, 2006, KIT Karlsruhe Insitute of Technology, Germany http://helios.informatik.uni-kl.de/RCeducation06/RCe-i1.pdf
- 282. G. Lienhart: Beschleunigung Hydrodynamischer N-Körper-Simulationen mit Rekonfigurierbaren Rechensystemen; Joint 33rd Speedup & 19th PARS Workshop; Basel, Switzerland, March 19-21, 2003
- 283. B. C. Schäfer et al.: Implementation Of The Discrete Element Method Using Reconfigurable Computing (FPGAs); 15th ASCE Engineering Mechanics Conf., June 2-5, 2002, New York, NY284
- 284. http://www.civil.columbia.edu/em2002/proceedings/papers/126.pdf
- 285. W. Roelandts (Keynote): FPGAs and the Era of Field Programmability; FPL, Aug. 29 Sep 1, 2004, Antwerp,
- 286. A. A. Gaffar and W. Luk: Accelerating Radiosity Calculations; FCCM 2002
- 287. M. Gokhale et al.: Acceleration of Traffic Simulation on Reconfigurable Hardware; 2004 MAPLD Int'l Conf., Sep 8-10, 2004, Washington, D.C., USA
- 288. J. Hammes, D. Poznanovic: Application Development on the SRC Computers, Inc. Systems; RSSI Reconfigurable Systems Summer Inst. July 11-13, 2005, Urbana-Champaign, IL,USA
- 289. E. Chitalwala: Starbridge Solutions to Supercomputing Problems; RSSI Reconf. Syst. Summer Inst., July 11-13, 2005, Urbana-Champaign, IL, USA
- 290. S. D. Haynes, P. Y. K. Cheung, W. Luk, J. Stone: SONIC A Plug-In Architecture for Video Processing; FPL 99
- 291. J. Rabaey: Reconfigurable Processing: The Solution to Low-Power Programmable DSP, ICASSP 1997

- 292. A. Darabiha: Video-Rate Stereo Vision on Reconfigurable Hardware; M. Thesis, U. Toronto, 2003
- 293. R. McCready: Real-Time Face Detection on a Configurable Hardware Platform; M. thesis, U. Toronto
- 294. T. Fry, S. Hauck: Hyperspectral Image Compression on Reconfigurable Platforms; IFCCM 2002
- 295. P. Buxa, D. Caliga: Reconfigurable Processing Design Suits UAV Radar Apps; COTS Journal, Oct. 2005296
- 296. http://www.srccomp.com/ReconfigurableProcessing UAVs COTS-Journal Oct05.pdf
- 297. R. R. Colenbrander et al.: Co-design and Implementation of the H.264/AVC Motion Estimation Algorithm Using Co-simulation; Proc. 11th EUROMICRO Conf. on Digital System Design Architectures, Methods and Tools (DSD 2008), Sep 3-5, 2008, Parma, Italy, pp 210-215
- 298. F. Dittrich: World's Fastest Lanman/NTLM Key Recovery Server Shipped; Picocomputing, 2006299
- 299. http://www.picocomputing.com/press/KeyRecoveryServer.pdf
- 300. K. Gaj, T. El-Ghazawi: Cryptographic Applications; RSSI Reconfigurable Systems Summer Institute, July 11-13, 2005, Urbana-Champaign, IL, USA301
- 301. http://www.ncsa.uiuc.edu/Conferences/RSSI/presentations.html
- 302. Y. Gu, et al.: FPGA Acceleration of Molecular Dynamics Computations; FCCM 2004303
- 303. http://www.bu.edu/caadlab/FCCM05.pdf
- 304. Anish Alex, Jonathan Rose et al.: Hardware Accelerated Novel Protein Identification; FPL 2004
- 305. N. N. Nallatech, press release, 2005
- 306. H. Singpiel, C. Jacobi: Exploring the benefits of FPGA processor technology for genome analysis at Acconovis; ISC 2003, June 2003, Heidelberg, Germany307
- 307. http://www.hoise.com/vmw/03/articles/vmw/LV-PL-06-03-9.html
- 308. M. Feldman: In Fermi's Wake, a Place for FPGAs? HPCwire, Oct. 15, 2009
- 309. N. N. (Starbridge): Smith-Waterman pattern matching; National Cancer Inst., 2004
- 310.O. O. Storaasli, D. Strenski: Experiences on 64 and 150 FPGA Systems; Reconfig. Syst. Summer Inst., July 7-9, 2008, Urbana-Champaign, IL, USA
- 311. M. Feldman: The FPGA crowd reacts; HPCwire Oct. 15, 2009
- 312. M. Kuulusa: DSP Processor Based Wireless System Design; Tampere Univ. of Technology, Publ. No. 296313
- 313. http://edu.cs.tut.fi/kuulusa296.pdf
- 314. D. Knuth: The Art of Computer Programming, Vol. 3, Sorting and Searching, Addison-Wesley, 1973
- 315. R. Hartenstein, A. Hirschbiel, M. Weber: MOM-map-oriented machine-a partly custom-designed architecture compared to standard hardware; Proc. IEEE CompEuro, Hamburg, Germany, May 1989
- 316. M. Duhl: Incremental Development and Description of a Shuffle Sort Array Circuit in hyperKARL from the Algorithm Representation of the Bubble Sort Algorithm; Projektarbeit, Informatik, Univ Kaiserslautern 1988
- 317. S. McKee: Reflections on the Memory Wall; 1st conf. on Computing Frontiers, Ischia, Italy, 2004
- 318.O. O. Storaasli, D. Strenski: Exceeding 100X Speedup/FPGA Cray XD1 Timing Analysis Yields Further Gains; Proc. Cray Users Group, 2009 http://www.nccs.gov/wp-content/uploads/2010/01/Storaasli-paper.pdf
- 319. Ian Kuon, Russell Tessier, Jonathan Rose: FPGA Architecture: Survey and Challenges; Foundations and Trends in Electronic Design Automation: Vol. 2: No 2, 2008, pp. 135-253.
- 320. Jih-Sheng Shen, Pao-Ann Hsiung (editors): Dynamic Reconfigurable Network-On-Chip Design: Innovations for Computational Processing and Communication; Information Science Publishers, April 2010
- 321. R. Hartenstein: Implications of Makimoto's Wave; http://hartenstein.de/ImplicationsMakimotosWave2.pdf
- 322. http://en.wikipedia.org/wiki/ENIAC
- 323. Martin H. Weik: The ENIAC Story; January-February 1961 issue of ORDNANCE (Journal Amerian Ordnance Association) http://ftp.arl.mil/~mike/comphist/eniac-story.html
- 324. N. N.: Chaos; 1995, The Standish Group http://www.projectsmart.co.uk/docs/chaos-report.pdf
- 325. J. Backus: Can programming be liberated from the von Neumann style?; Comm ACM, Aug1978
- 326. E. Dijkstra: The GOTO Considered Harmful; Comm. ACM 11,3, March 1968
- 327. N. Wirth: A Plea for Lean Software, IEEE Computer, 28, 2, (Feb. 1995)
- 328. Arvind et al.: A critique of Multiprocessing the von Neumann Style; ISCA 1983
- 329. Peter G. Neumann 1985-2003: 216x "Inside Risks" (18 years in ack cover, each issue of C_ACM)
- 330. R. Hartenstein, G. Koch: The universal Bus considered harmful; EUROMICRO Workshop on the Microarchitecture of Computer Systems, June 23-25, 1975, Nice, France
- 331. J. L. Hennessy, D. A. Patterson, Computer Architecture: a Quantitative Approach, Morgan-Kaufman, 1990
- 332. J. Becker, et al.: Data-procedural Languages for FPL-based Machines; FPL'94, Sep 7-10, 1994, Prague
- 333. http://gpgpu.org/
- 334. V. W. Lee et al.: Debunking the 100X GPU vs. CPU myth; 37th ISCA, June 19-23, 2010, Saint-Malo, France,
- 335. R. Vaduc et al: On the Limits of GPU Acceleration; USENIX Workshop HotPar'2010, June 14-15,2010, Berkeley,

- 336. R. Bordawekar, U. Bundhugula, R. Rao: Believe it or Not! Multicore CPUs can Match GPUs for FLOP-intensive Applications! IBM Research Report, April 23, 2010, Yorktown Heights, NY, USA
- 337. Thomas Scogland, Heshan Lin, Wu-chun Feng: A First Look at Integrated GPUs for Green High-Performance Computing; In'l Conf. on Energy-Aware HPC; Sep 16–17, 2010, Hamburg, Germany150
- 338. Sangjin Han, Keon Jang, KyoungSoo Park, Sue Moon: PacketShader: a GPU-Accelerated Software Router; SIGCOMM'10, Aug 30–Sep 3, 2010, New Delhi, India
- 339. Michael Feldman: A GPU on Every Chip; August 19, 2010, HPCwire340
- 340. http://www.hpcwire.com/blogs/A-GPU-on-Every-Chip-101104834.html
- 341. Lawrence Latif: AMD turns up the heat on Nvidia's GPGPUs; Aug 17, 2010, The Inquirer342
- 342. http://www.theinquirer.net/inquirer/feature/1727688/amd-heat-nvidias-gpgpus
- 343. Ian King: Intel to Make Processor With Built-in Graphics, Challenging AMD; Sep 9, 2010, Bloomberg344
- 344. http://www.bloomberg.com/news/2010-09-08/intel-to-show-off-new-chip-with-graphics-tackle-amd-challenge.html?cmpid=yhoo
- 345. T. Claasen: High Speed: Not the Only Way to Exploit the Intrinsic Computational Power of Silicon; ISSCC-1999, pp. 22-25, Feb. 1999
- 346. Jan Rabaey (keynote): Going Beyond Turing: Energy-efficiency in the Post-Moore Era; 2010 Int'l Symp. on Low Power Electronics and Design (ISLPED), Austin, TX, USA Aug 18-20, 2010
- 347. Kevin Nowka (keynote): Technology variability and uncertainty implications for power efficient VLSI systems; 2010 Int'l Symp. Low Power Electronics & Design (ISLPED), Austin, TX, USA Aug 18-20, 2010
- 348. M. Immitzer: Information and Communication Technologies Enable More Energy-Efficient Societies; e-Energy 2010, April 13-15, 2010, Passau, Germany
- 349. L. Chiaraviglio, I. Matta: GreenCoop: Cooperative Green Routing with Energy-Efficient Servers; e-Energy 2010, April 13-15, 2010, Passau, Germany
- 350. C. Freeman: Power Reduction for FPGA using Multiple Vdd/Vth.; Monday April 3, 2006351
- 351. http://deimos.eos.uoguelph.ca/sareibi/TEACHING_dr/ENG6530_RCS_html_dr/outline_W2010/docs/LECTURE_dr/TOPIC15_dr/2006_dr/Cecille_topic_presentation_april3.pdf
- 352. H. Simon (invited presentation): Leibniz-Rechenzentrum, TU Munich, 2009, Garching, Germany
- 353.H. Simon: The Greening of HPC Will Power Consumption Become the Limiting Factor for Future Growth in HPC? HPC User Forum, Oct 13-14, 2008, Stuttgart, Germany
- 354. Stacey Higginbotham: Watch Out Intel! Marvell to Make ARM-based Server Chips; May. 11, 2010, GigaOM (Business Week)
- 355. Colin Holland: GE seek early adopters for FPGA boards; Aug 3, 2010, EETimes356
- 356. http://www.eetimes.com/electronics-news/4205834/GE-seek-early-adopters-for-FPGA-boards
- 357. Nicola Leske, Noel Randewich: Intel buys Infineon unit and expands wireless offer; Aug 30, 2010, Reuters http://www.reuters.com/article/idUSTRE67Q3J820100830
- 358. Van Smith: The Coming War: ARM versus x86; April 8, 1020, Van's Hardware Journal359
- 359. http://vanshardware.com/2010/04/the-coming-war-arm-versus-x86/
- 360. Tom Krazit: iPhone 3.0 code hints at future Apple hardware; March 19, 2009, CNET News http://news.cnet.com/8301-13579_3-10200292-37.html
- 361. http://www.fpgaworld.com/modules.php?name=News&file=article&sid=297
- 362. T. Tuan, A. Rahman, S Das, S. Trimberger, S. Kao: A 90-nm Low-Power FPGA for Battery-Powered Applications; Feb. 2007, IEEE Trans CAD-26
- 363.R. Porter: Evolution on FPGAs for Feature Extraction; Ph.D. thesis; Queensland Un. of Technology Brisbane, Australia, 2001364
- 364. http://www.pooka.lanl.gov/content/pooka/green/Publications files/imageFPGA.pdf
- 365. S. Moore: Multicore Is Bad News For Supercomputers; IEEE Spectrum, Nov. 2008
- 366. Jagath C. Rajapakse, Amos R. Omondi (editors): FPGA Implementations of Neural Networks; 2006, Springer
- 367. R. Hartenstein (keynote): Reconfigurable Computing: boosting Software Education for the Multicore Era; Proc. SPL 2010) Porto Galinhas Beach, Brazil, March 24-26, 2010
- 368. Arnaud Grasset, Richard Taylor, Graham Stephen, Joachim Knäblein, Axel Schneider: Specification Tools for Spacial Design; in332
- 369. personal communication from authors of chapters of the book267:
- 370. Falk Salewski, Dirk Wilking, Stefan Kowalewski: Diverse Hardware Platforms in Embedded SystemsLab Courses: A Way to Teach the DifferencesThe first Workshop on Embedded Systems Education WESE 2005Sep 22, 2005, Jersey City, New Jersey, USA

- 371. Kiran. Bondalapati, Viktor K. Prasanna: Mapping Loops onto Reconfigurable Architectures; 8th Int'l Workshop on Field-Programmable Logic and Applications, Sep 1998, Tallinn, Estonia
- 372. S. Bhattacharyya et al.: OpenDF, A Dataflow Toolset for Reconfigurable Hardware and Multicore Systems; Swedish Workshop on Multicore Computing (MCC-08), Ronneby, Sweden, 27-28 Nov, 2008
- 373. C. Bell et al: The Description and Use of Register-Transfer Modules (RTM's); IEEE Trans-C21/5, May 1972
- 374. K. Morris: Who Are You People? ...and What Do You Want? FPGA and Structured ASIC Journal375
- 375. http://www.fpgajournal.com/articles 2008/20080715 survey.htm
- 376. F. L. Bauer: (coined "software crisis"); 1st NATO Software Engrg Conf. 1968, Garmisch, Germany.
- 377. http://en.wikipedia.org/wiki/Software_crisis
- 378. E. Dijkstra (Turing award lecture): The Humble Programmer; Comm. ACM, 15,10 (Oct 1972)
- 379. C. Walter: Kryder's Law; Scientific American Magazine, August 2005380
- 380. http://www.sciam.com/article.cfm?id=kryders-law
- 381. N. Wirth: A Plea for Lean Software; Computer, 28,2, Feb. 1995
- 382. Wang Yi: Towardes Real-time Applications on Multicore Plartforms: The Timing Problem and Possible Solutions; ARTIST Summer School Europe 2010, Autrans, France
- 383. Hiroaki Takada: Challenges of Hard Real-Time Operating Systems; ARTIST Summer School Europe 2010, September 5-10, 2010, Autrans, France
- 384. Sanjoy Baruah: Scheduling Issues in Mixed-criticality Systems; ARTIST Summer School Europe 2010, September 5-10, 2010, Autrans, France
- 385. Rolf Ernst: Formal Performance Analysis and Optimization of Safety-related Embedded Systems; ARTIST Summer School Europe 2010, September 5-10, 2010, Autrans, France
- 386. Luca Benini: Programming Heterogeneous Many-core Platforms in Nanometer Technology; ARTIST Summer School Europe 2010, September 5-10, 2010, Autrans, France
- 387. Madeleine Faugere et al.: MARTE: Also an UML Profile for Modeling AADL Applications; 12th IEEE Int'l Conf. on Engrg Complex Computer Systems (ICECCS 2007), Auckland, New Zealand
- 388. Jorn Janneck: Dataflow Programming; ARTIST Summer School Europe 2010, Autrans, France
- 389. http://x10.codehaus.org/
- 390. N. N.: Developing Parallel Programs A Discussion of Popular Models; Sep 2010, Oracle White paper391
- 391. http://www.oracle.com/technetwork/server-storage/solarisstudio/documentation/oss-parallel-programs-170709.pdf
- 392. Miroslav Lícko, Jan Schier, Milan Tichý, Markus Kühl: MATLAB/Simulink Based Methodology for Rapid-FPGA-Prototyping; 13th International Conference on Field Programmable Logic and Applications (FPL 2003), Sep 1-3, 2003, Lisbon, Portugal
- 393. Gary W. Johnson, Richard Jennings: LabVIEW Graphical Programming; August 2006, Mcgraw-Hill
- 394. John Essick: Hands on Introduction to LabVIEW for Scientist and Engineers; Nov 2008, Oxford Univ Press
- 395. Dick Selwood: Drag-and-Drop vs. HDL? FPGA and Programmable Logic Journal396
- 396. http://www.techfocusmedia.net/fpgajournal/feature_articles/20100803-ni/
- 397. Joseph A. Roback, Gregory R. Andrews: Gossamer: A Lightweight Programming Framework for Multicore Machines; HOTPAT10, 2nd USENIX workshop on Hoot Topics in Parallelism, June 14 15, Berkeley, CA http://www.usenix.org/events/hotpar10/tech/
- 398. K. Asanovic et al.: A view of the parallel computing landscape. Commun. ACM 52, 10 (2009)
- 399. Xilinx Press release: Xilinx 7 Series FPGAs Slash Power Consumption by 50% and Reach 2 Million Logic Cells on Industry's First Scalable Architecture; June 21, 2010, San Jose, CA, USA400
- 400. http://news.thomasnet.com/companystory/Xilinx-7-Series-FPGAs-Slash-Power-Consumption-by-50-and-Reach-2-Million-Logic-Cells-on-Industry-s-First-Scalable-Architecture-581023
- 401. R. Hartenstein, M. Herz, Th. Hoffmann, U. Nageldinger: KressArray Xplorer: A New CAD Environment to Optimize Reconfigurable Datapath Array Architectures; 5th Asia and South Pacific Design Automation Conference (ASP-DAC), Jan 25-28, 2000, Yokohama, Japan402
- $402. \, http://www.michael-herz.de/publications/aspdac00.fin.pdf$
- 403.4th Workshop on Programmability Issues for Heterogeneous Multicores (MULTIPROG-2011), January 23, 2011, Heraklion, Crete, Greece http://multiprog.ac.upc.edu/
- 404. Gerd Michelsen, Jasmin Godemann (Herausgeber): Handbuch der Nachhaltigkeits-kommunikation: Grundlagen und Praxis; 2005, Oekom Verlag
- 405. Clive Maxfield: C-to-FPGA integration gives prototyping 10X boost! Sep 3, 2010, EETimes406
- 406. http://www.eetimes.com/electronics-products/electronic-product-reviews/fpga-pld-products/4207362/C-to-FPGA-integration-gives-prototyping-10X-boost-