Implications of Makimoto's Wave

Reiner Hartenstein
TU Kaiserslautern
http://hartenstein.de

Makimoto's wave [1] [2] models the history and predicts the future of mainstream integrated circuit (IC) applications (figure 1, slide 1). Hartenstein also used Makimoto's wave as a coordinate system to model the history of computing [3] [4] (slide 2), where the mainframe age was before Makimoto's wave. Makimoto's law is as important as Gordon Moore's law. Both laws are synchronized with the market introduction of the integrated circuit.

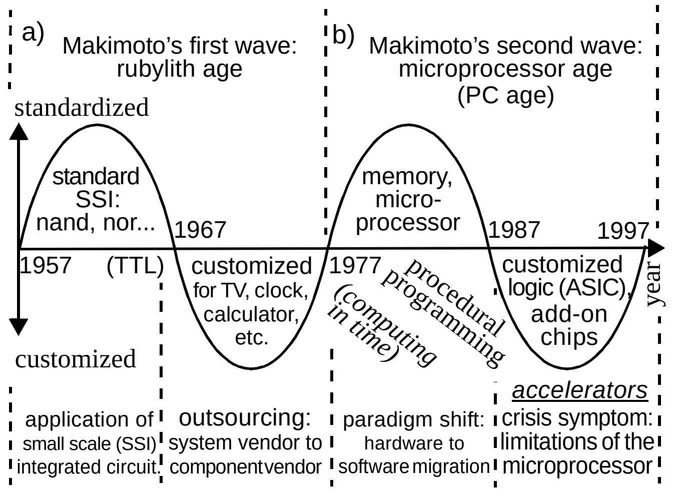


Fig. 1: Makimoto's wave: history of change in integrated circuit use first wave (rubylith age), and, second wave (PC age)

The first wave is determined by hardwired designs implemented by LSI CAD (figure 2 a). Nick Tredennick classifies this phase of IC mainstream application history by „resources fixed and algorithms fixed" (figure 3 a). The number of designs was exploding going toward a dominance of customized designs. Increasing design complexity due to Moore's law lead to the first design crisis, first discussed in 1975.

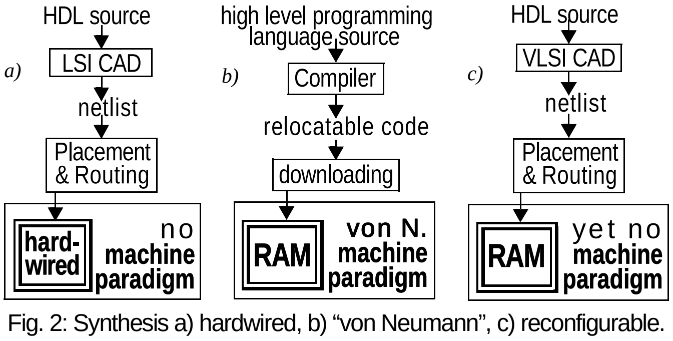


Fig. 2: Synthesis a) hardwired, b) "von Neumann", c) reconfigurable.

Makimoto's second wave has been caused by the introduction of the microprocessor and memory microchips (figure 1 b) as standard components (first half of 2nd wave). Nick Tredennick classifies this phase of IC application history by „resources fixed and algorithms variable" (figure 3 b). This „new" kind of design flow (figure 2 b), being a „replacement of the soldering iron by the keyboard" [5] [6], has been a major revolution. But preparing a revolution takes time. It has been reported, that intel needed to give courses to a quarter million people to be able to sell microprocessors at all.

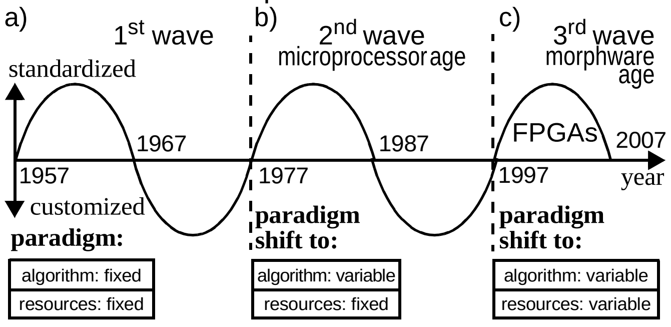


Fig. 3: Makimoto's wave & Tredennick's paradigm shift model.

Accelerators. Due to the microprocessor's sequential nature of operation its weakness requires an increasing number of accelerator co-processor designs of growing complexity. Finally most of the silicon real estate is covered by auxiliary circuitry needed for support [5] [6]. This heavily growing customized circuit demand has triggered the 2nd half wave of Makimoto's second wave (figure 1). A transition to a software / hardware split design flow is the consequence (figure 4 b), which became typical to the embedded systems design community. *This hardware / software chasm* by this splitting of the design flow is *a severe educational problem* causing billions of dollars of damage each year: the second design crisis.

Design Crises. Each Makimoto cycle so far has been concluded by a design crisis ([5] [6] figure 5). The first design crisis around the mid 70ies stimulated massive academic research efforts by the Mead-&-Conway rush [7] and the foundation of the EDA industry (Electronic Design Automation industry). The impact of the Mead-&-Conway rush caused a major restructuring of EE and EECS

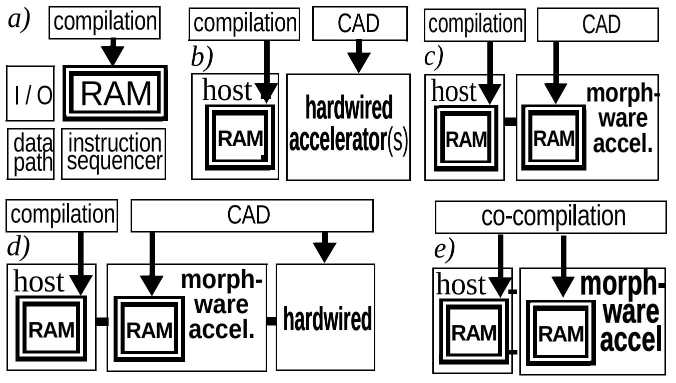


Fig. 4: History of computing platforms: a) "von Neumann"-based b) embedded systems, c) with programmable accelerators, d) with mixed accelerators, e). future co-compilation (needs specific silicon

curricula world-wide - during the time of the first half of Makimoto's second wave (figure 1). The rapid growth of the designer population during that time was one of the reasons of an exploding number of customized IC designs: second half of Makimoto's second wave (figure 1 b).

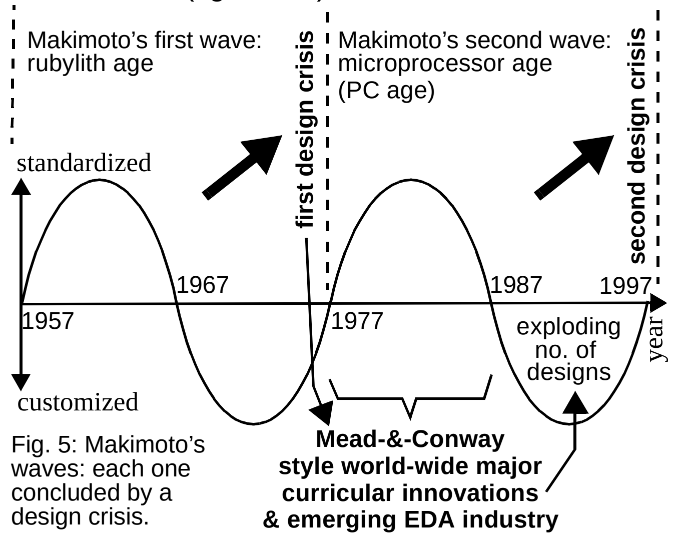

Makimoto's first wave: rubylith age
Makimoto's second wave: microprocessor age (PC age)
first design crisis
second design crisis
standardized
customized
1957
1967
1977
1987
1997
year
exploding no. of designs
Mead-&-Conway style world-wide major curricular innovations & emerging EDA industry

Fig. 5: Makimoto's waves: each one concluded by a design crisis.

The second design crisis. The still existing second design crisis stems from the facts, that designer productivity is growing much slower than design complexity, and, that rapidly increasing design cost comes along with shrinking product life cycles. EDA industry cannot meet the tool quality requirements. A poll held during FCCM at Napa, California, in 1998 has revealed, that more than 80% of all designers hate their tools.

Morphware has become mainstream what has fulfilled the prediction of Makimoto's third wave: the accelerators have become programmable (figure 2 c). Nick Tredennick classifies this phase of IC application history by „resources variable and algorithms variable" (figure 3 c).

The configware / software chasm. Many accelerators have become programmable. But their application development still uses (a modified form of) hardware design methods (figure 4 c and d). The *hardware / software chasm* has turned into a *configware / software chasm*. This chasm is caused by massive deficits in CS education and is a main reason of the current second design crisis [8] [9].

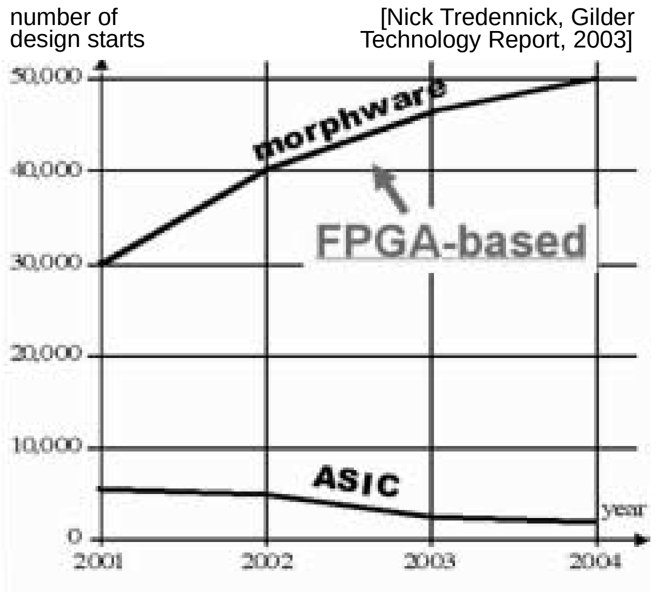


Fig. 6: Design start statistics: ASICs versus FPGA designs.

FPGAs replacing silicon foundries? The first half of Makimoto's third wave shows the dominance of FPGAs (figure 3), being a kind of standardized, since logic gates are general purpose, and, by being commodities. In contrast to ASICs, FPGAs do not require specific silicon suffering from exponentially growing mask cost and other NRE cost: a reason for the defeat of ASIC design starts by FPGA-based design starts (figure 6). Also the adoption rate of silicon foundries is declining (figure 7).

The FPGA Efficiency Paradox. Because most of the area of an FPGA microchip is covered by wiring patterns, its integration density (transistors per chip) is lower than the Moore curve by 2 orders of magnitude (figure 8). But already in 1995 the *physical integration density* of FPGAs had already exceeded that of the microprocessor (figure 8). Since FPGA layout is regularly structured like that of

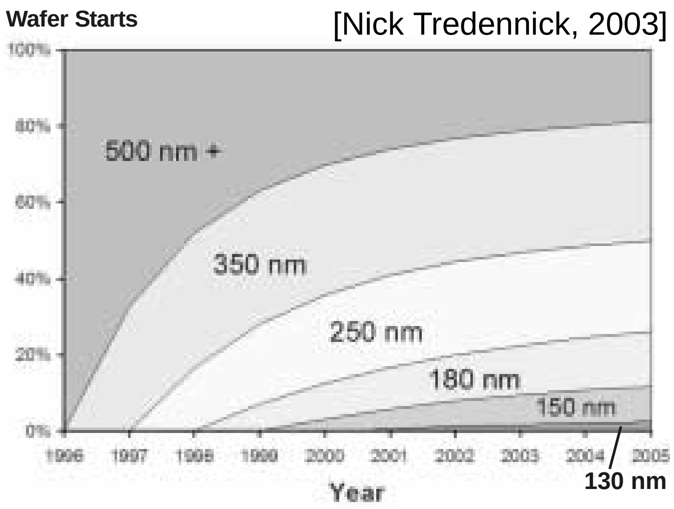


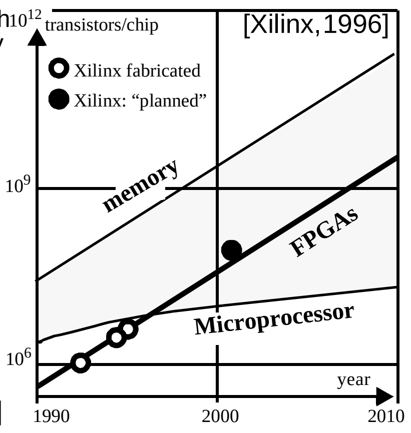
Fig. 7: Silicon Foundry Adoption Rate by Process Technology.

Fig. 8: FPGA high growth rate of integration density — compared to memory and microprocessor.

Progress: parallel to Gordon Moore Curve

FPGA pre-design is of full custom style

Transistor count exceeds that of the microprocessor [Xilinx]

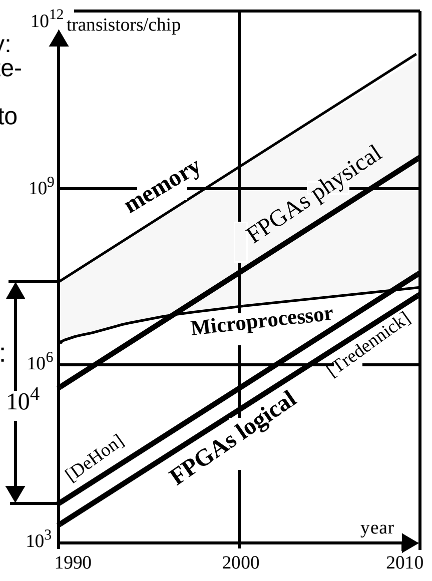


a memory microchip the growth of integration density goes in parallel with Gordon Moore’s curve (figure 8). But due to reconfigurability overhead only about one out of about 100 to 200 transistors directly serves the application, so that the *logical integration density* of FPGAs is behind that of memory (Moore curve) by 4 orders of magnitude (figure 9). But there are numerous publications reporting substantial speed-ups mostly ranging up to 2 orders of magnitude, obtained from software-to-FPGA migration of a variety of applications ([3], see slide 3, [4]) - although the clock speed of an FPGA is about a factor of 5 slower than that of a microprocessor of comparable technology.

Bypassing the memory wall. The FPGA efficiency paradox is explained by the much higher degree of parallelism possible within an FPGA, as well as by avoiding the massive memory cycle overhead inevitable with the instruction-stream-based classical machine paradigm (where memory cycles are slower than processor clock cycles by more than two orders of magnitude). Bypassing the memory wall is one of the key speed-up factors coming early with the time of Makimoto’s third wave. Another speed-up factor is the multiple level

ig. 9: The hardware ap of reconfigurability: hysical vs. logical inte-ration density of PGAs — compared to icroprocessors and emory chips.

reconfigurability overhead in FPGAs: logical chip area uses only 1 % of physical chip area [DeHon] - 1 logical transistor per 200 physical transistors [Tredennick]



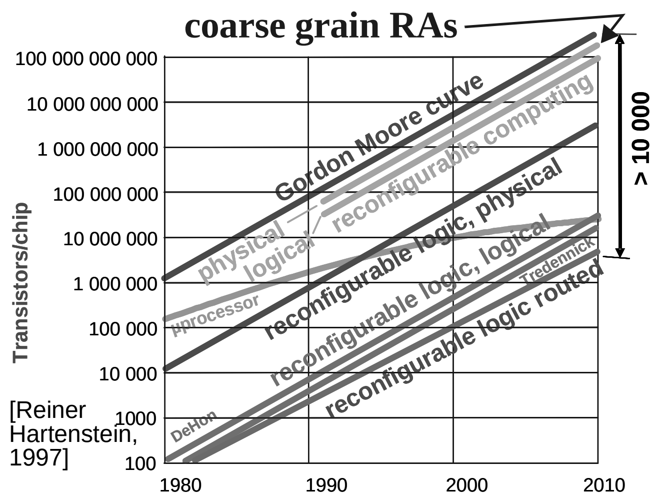


Fig. 10: Integration density of coarse grain reconfigurable array

parallelism being much more flexible than classical parallelism obtained only by concurrent software processes.

Reconfigurable Computing. Implementing an application onto an FPGA is a design activity at gate level, where a CLB is about one bit wide - unless not yet really available good tools hide this abstraction level from the users point of view. However, the use of coarse grain morphware arrays (rDPAs: reconfigurable Data Path Arrays, also commercially available [10] slide 5) with word-width rDPUs (reconfigurable Data Path Units, 32 bits wide, for instance) directly featuring number crunching functionality turns programming from logic synthesis to computing [11] [12] [13].

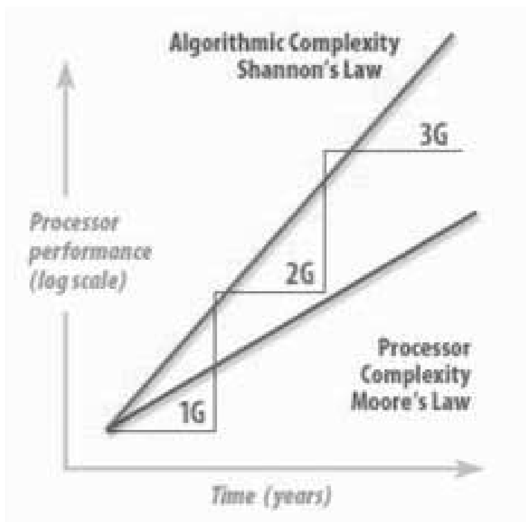
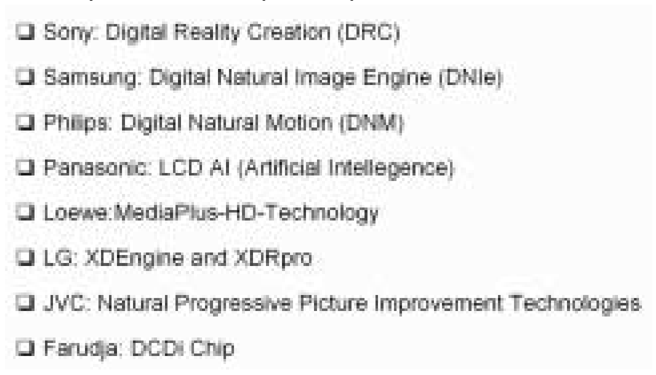


Fig. 11: Cellular wireless communication: performance requirements growing faster than Moore’s law [Jan Rabaey].

High Area Efficiency. When a coarse grain morphware array is well designed like a processor core in full custom structured design style using wiring by abutment, it reaches almost the integration density of Gordon Moore‘s curve (figure 10). Because of the high processing power of a rDPU only a small array is needed for most applications, like around a hundred rDPUs, for instance.

Compared to modern FPGAs with up to a million CLBs this massively decreases reconfigurable interconnect fabrics requirements. The availability of 7 or more metal layers allows the interconnect fabrics can be layouted over the rDPU so no extra interconnect areas are needed outside rDPU clocks. Also configuration memory is drastically reduced, so that the configuration time is massively reduced. Physical and logical integration density are almost the same (figure 10). Morphware also yields more MOPS per milliWatt (slide 4).



- Sony: Digital Reality Creation (DRC)
- Samsung: Digital Natural Image Engine (DNIe)
- Philips: Digital Natural Motion (DNM)
- Panasonic: LCD AI (Artificial Intellegence)
- Loewe:MediaPlus-HD-Technology
- LG: XDEngine and XDRpro
- JVC: Natural Progressive Picture Improvement Technologies
- Farudja: DCDi Chip

Fig. 12: Main battlefield: Picture quality

Coarse Grain Applications. For coarse grain morphware platforms there is a wide variety of applications like any kind of HPC, but also consumer electronic products and other embedded systems. An example is *cellular wireless communication*, where performance requirements are growing faster than Moore's law (figure 11) what cannot bed met even by the most powerful DSPs. Coarse grain morphware platforms provide the flexibility needed to cope with multiple (de)coding standards and multiple media changing on the fly (voice, voice over IP, image, video, SMS, e-mail, and others). Here coarse grain platforms are already well practicable for base stations. But because of low power requirements we need a future generation of morphware platforms for handies and similar battery-driven appliances.

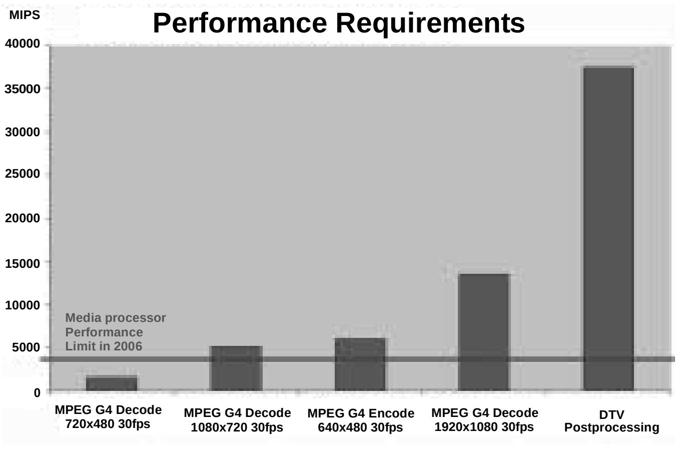


Fig. 13: Picture processing: performance requirements

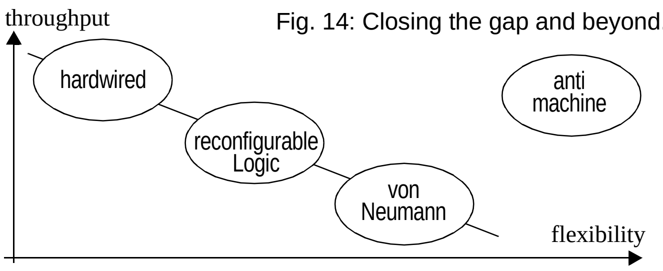


Fig. 14: Closing the gap and beyond.

Coarse grain for (H)DTV. Another important area of coarse grain morphware application area is *(H)DTV* where picture quality is the main battlefield (figure 12). Enormous performance is required (figure 13). Examples of processing tasks are noise reduction, picture improvement, artefact removal (e. g. smoothing horizontal and vertical lines), scaling of display size, scan rate and frame rate conversion, multiple standard video (de)coding, variable file format conversions, variable content security formats, and many other processing tasks. In this field of applications programmability is inevitable because casting improved algorithms onto silicon is too expensive, because ASIC chip development takes too long, and, because of ASIC inflexibility hampers adding new features. Coarse grain arrays bring continuity by programmability and drastically more implementation efficiency by morphware re-use. For (H)DTV implementations on morphware a benefit has been reported by a factor of 4 in development time, and, by a factor of 5 in development cost [10].

The Anti Machine. The main problem of supercomputing stems from the instruction-stream-driven basic paradigm, sometimes called von Neumann paradigm. The benefit of the morphware-based anti machine goes far beyond bridging the gap between microprocessor and ASIC (figure 14). Meanwhile is has been recognized, that the only roadmap to drastically higher HPC efficiency, and to solve the problem of deficits in education [14] [15], is the introduction of a second machine paradigm, which is data-stream-driven - the anti machine [15] [16] [17] [18], the direct counterpart to the von Neumann paradigm. We need a duality of 2 machine paradigms (figure 4 e). But for the anti machine one or more data counters are used instead of a program counter (slide 6). But there are unsymmetries: in anti machines the data counter is co-located with a memory bank, so that the memory bank is an auto-sequencing memory module (asM), being able to autonomously generate a data stream (slide 6).

Configware replacing Software. Anti machines cannot be programmed by software (programming in time, which is instruction-stream-based). To program morphware we need configware instead (for configuration: i. e. programming in space) and flowware to program the asMs for the data streams needed run time (see slide 7). The co-compiler we need has to generate three different blocks of three different kinds of code (figure 15, slide 8): software code (for the host), configware code, and, flowware code. Figure 15 shows the structure of an automatically partitioning academic software / configware / flowware co-compiler [19] [20]. Configware compilation [21] [22] and automatic partitioning [23] [24] can be easily implemented by simulated annealing. Figure 16 summarizes a commercial example of an integrated development environment for coarse grain morphware [10].

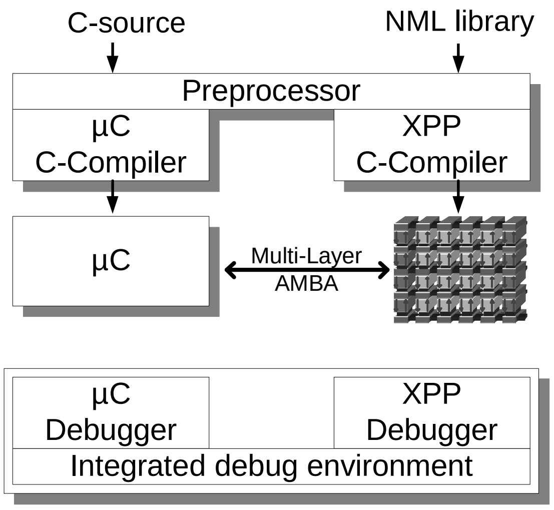
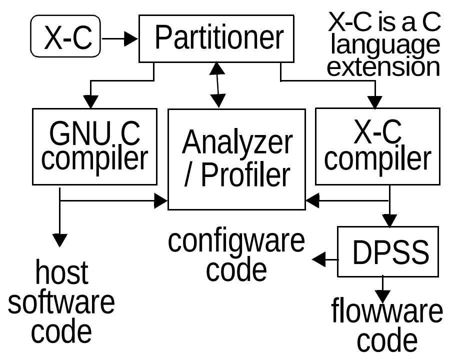


Fig. 16: PACT corp. Integrated Development Environment

Makimoto's last wave. Reiner Hartenstein has merged Makimoto's law with Nick Tredennicks paradigm shift model (figure 3). A remaining question is: will Makimoto's third wave really a wave, and how long will it take? The conclusion from Nick Tredennicks paradigm shift model is, that there will not be a fourth Makimoto's wave, because all degrees of freedom have already been exhausted by the third wave. The consequence is, that Makimoto's third wave will last much longer than 20 years: probably it will last for all the rest of the life time of the silicon IC industry.

Reconfigurable HPC. Hartenstein predicted [11] [12] [13] [25] [26], that the second phase of Makimoto's third wave will have a different shape (figure 17). Because of their very high area

Fig. 15: CoDe-X automatically partitioning Co-Compiler.



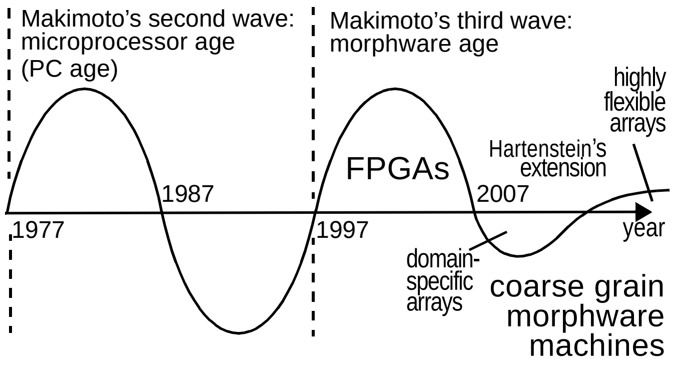


Fig. 17: 3rd Makimoto wave: Hartenstein's extension (2nd half wa

efficiency and very high throughput coarse grain morphware will become important, also driven by the coming rush from classical high performance computing (HPC) to reconfigurable HPC [3] [8] [14] [17]. But general purpose rDPAs may still be unrealistic, so that domain-specific arrays are more likely. This means somewhat limited flexibility: i. e. only slightly customized (compare figure 17).

Mapping HPC onto FPGAs ? Featuring multi-context coarse grain morphware architectures future rDPA might become almost general purpose. Mapping rDPAs onto FPGAs might make sense with future high-density FPGA architectures. So the final waveform of Makimoto's third wave should go up to slightly standardized (figure 17).

The Personal Supercomputer (PS). New CPU technology will not help to beef up old HPC architectures because this does not solve the memory wall problem. A remedy can be obtained only by a fundamental paradigm shift. Overcoming the sustained performance barrier and drastic hardware cost reduction is possible only by software to configware migration using coarse grain morphware platforms. The Reconfigurable Computing HPC rush is already running [17] [18] [27] [28]. Cheap supercomputing power at every home and office by a morphware extension card to the PC, featuring Reconfigurable High Performance

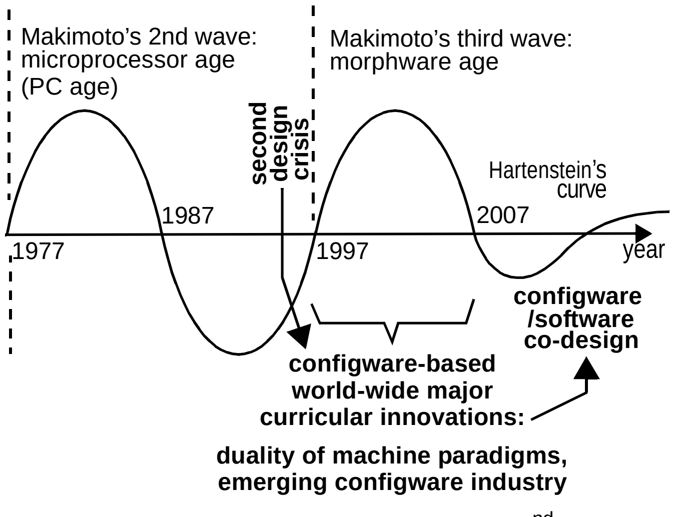
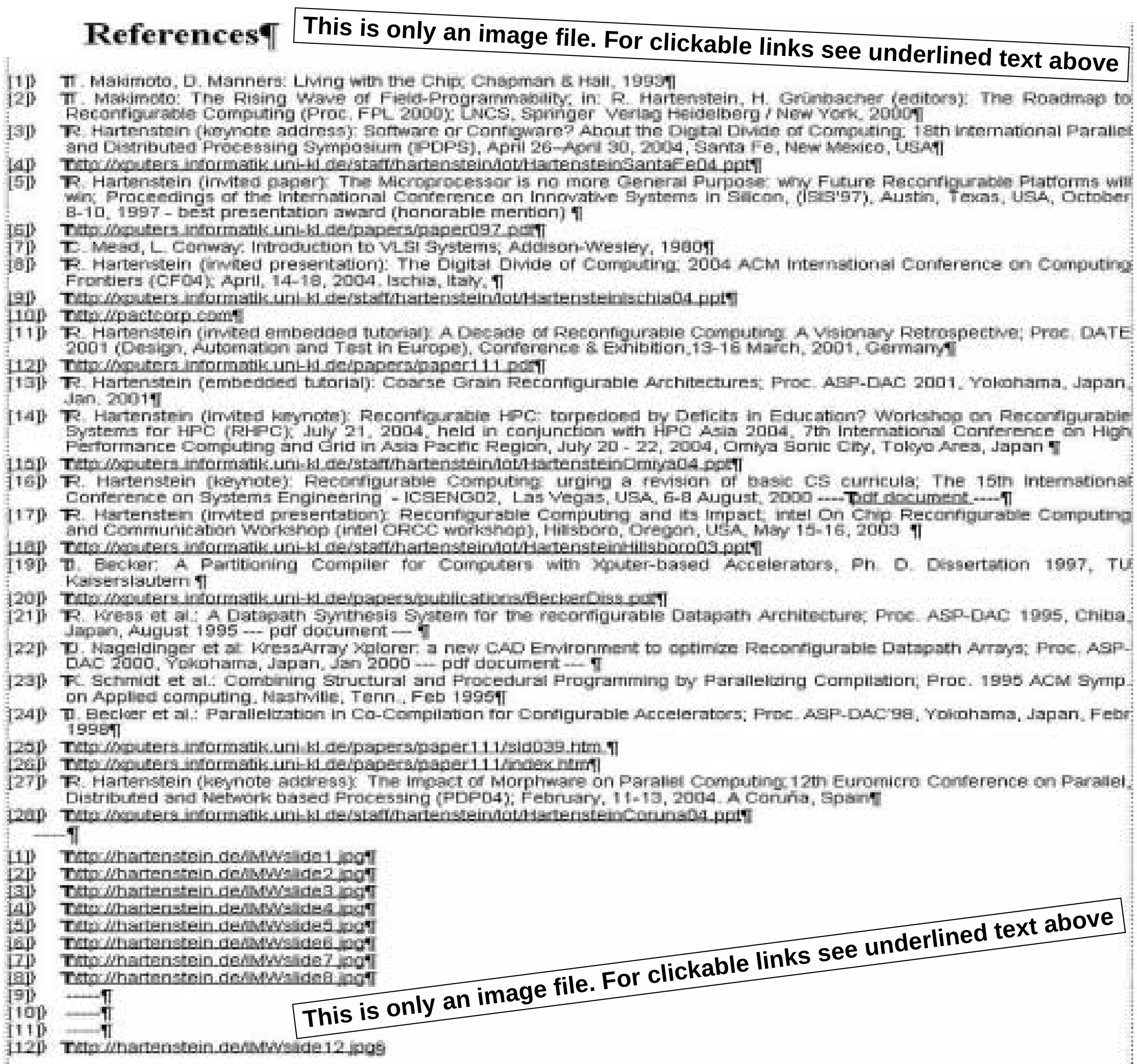


Fig. 18: Makimoto's 3rd waves: the impact of the 2nd design cris

Computing will bring a revival of the PC business. But intel seems not yet having recognized this trend. The success of the PS will repeat the success story of the software industry for the already existing configware industry (slide 12). Who will be the Bill Gates o the configware industry ? Figure 18 shows how the current configware rush fits to Makimoto’s third wave.

For more References click here



This is only an image file. For clickable links see underlined text above

References¶

[1] T. Makimoto, D. Manners: Living with the Chip; Chapman & Hall, 1993¶

[2] T. Makimoto: The Rising Wave of Field-Programmability; in: R. Hartenstein, H. Grünbacher (editors): The Roadmap to Reconfigurable Computing (Proc. FPL 2000); LNCS, Springer Verlag Heidelberg / New York, 2000¶

[3] R. Hartenstein (keynote address): Software or Configware? About the Digital Divide of Computing; 18th International Parallel and Distributed Processing Symposium (IPDPS), April 26–April 30, 2004, Santa Fe, New Mexico, USA¶

[4] http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/HartensteinSantaFe04.ppt¶

[5] R. Hartenstein (invited paper): The Microprocessor is no more General Purpose: why Future Reconfigurable Platforms will win; Proceedings of the International Conference on Innovative Systems in Silicon, (ISIS'97), Austin, Texas, USA, October 8-10, 1997 - best presentation award (honorable mention) ¶

[6] http://xputers.informatik.uni-kl.de/papers/paper097.pdf¶

[7] C. Mead, L. Conway: Introduction to VLSI Systems; Addison-Wesley, 1980¶

[8] R. Hartenstein (invited presentation): The Digital Divide of Computing; 2004 ACM International Conference on Computing Frontiers (CF04); April, 14-18, 2004, Ischia, Italy, ¶

[9] http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/HartensteinIschia04.ppt¶

[10] http://pactcorp.com¶

[11] R. Hartenstein (invited embedded tutorial): A Decade of Reconfigurable Computing: A Visionary Retrospective; Proc. DATE 2001 (Design, Automation and Test in Europe), Conference & Exhibition,13-16 March, 2001, Germany¶

[12] http://xputers.informatik.uni-kl.de/papers/paper111.pdf¶

[13] R. Hartenstein (embedded tutorial): Coarse Grain Reconfigurable Architectures; Proc. ASP-DAC 2001, Yokohama, Japan, Jan. 2001¶

[14] R. Hartenstein (invited keynote): Reconfigurable HPC: torpedoed by Deficits in Education? Workshop on Reconfigurable Systems for HPC (RHPC); July 21, 2004, held in conjunction with HPC Asia 2004, 7th International Conference on High Performance Computing and Grid in Asia Pacific Region, July 20 - 22, 2004, Omiya Sonic City, Tokyo Area, Japan ¶

[15] http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/HartensteinOmiya04.ppt¶

[16] R. Hartenstein (keynote): Reconfigurable Computing: urging a revision of basic CS curricula; The 15th International Conference on Systems Engineering - ICSENG02, Las Vegas, USA, 6-8 August, 2000 ---- pdf document ----¶

[17] R. Hartenstein (invited presentation): Reconfigurable Computing and its Impact; intel On Chip Reconfigurable Computing and Communication Workshop (intel ORCC workshop), Hillsboro, Oregon, USA, May 15-16, 2003 ¶

[18] http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/HartensteinHillsboro03.ppt¶

[19] J. Becker: A Partitioning Compiler for Computers with Xputer-based Accelerators, Ph. D. Dissertation 1997, TU Kaiserslautern ¶

[20] http://xputers.informatik.uni-kl.de/papers/publications/BeckerDiss.pdf¶

[21] R. Kress et al.: A Datapath Synthesis System for the reconfigurable Datapath Architecture; Proc. ASP-DAC 1995, Chiba, Japan, August 1995 --- pdf document --- ¶

[22] U. Nageldinger et al: KressArray Xplorer: a new CAD Environment to optimize Reconfigurable Datapath Arrays; Proc. ASP-DAC 2000, Yokohama, Japan, Jan 2000 --- pdf document --- ¶

[23] K. Schmidt et al.: Combining Structural and Procedural Programming by Parallelizing Compilation; Proc. 1995 ACM Symp. on Applied computing, Nashville, Tenn., Feb 1995¶

[24] J. Becker et al.: Parallelization in Co-Compilation for Configurable Accelerators; Proc. ASP-DAC'98, Yokohama, Japan, Febr 1998¶

[25] http://xputers.informatik.uni-kl.de/papers/paper111/sld039.htm ¶

[26] http://xputers.informatik.uni-kl.de/papers/paper111/index.htm¶

[27] R. Hartenstein (keynote address): The Impact of Morphware on Parallel Computing;12th Euromicro Conference on Parallel, Distributed and Network based Processing (PDP04); February, 11-13, 2004. A Coruña, Spain¶

[28] http://xputers.informatik.uni-kl.de/staff/hartenstein/lot/HartensteinCoruna04.ppt¶

-----¶

[1] http://hartenstein.de/IMWslide1.jpg¶

[2] http://hartenstein.de/IMWslide2.jpg¶

[3] http://hartenstein.de/IMWslide3.jpg¶

[4] http://hartenstein.de/IMWslide4.jpg¶

[5] http://hartenstein.de/IMWslide5.jpg¶

[6] http://hartenstein.de/IMWslide6.jpg¶

[7] http://hartenstein.de/IMWslide7.jpg¶

[8] http://hartenstein.de/IMWslide8.jpg¶

[9] -----¶

[10] -----¶

[11] -----¶

[12] http://hartenstein.de/IMWslide12.jpg

This is only an image file. For clickable links see underlined text above

For more References click here