Configware / Software Co-Design: be prepared for the Next Revolution!
Semiconductor Revolutions

"Mainstream Silicon Application is switching every 10 Years"

"The Programmable System-on-a-Chip is the next wave"

Makimoto's Wave

Treadennick's Paradigm Shifts

1957

1st Design Crisis

1967

2nd Design Crisis

1977

1987

1997

standard

hardwired

procedural programming

structural programming

reconfigurable

algorithm: fixed
resources: fixed

algorithm: variable
resources: fixed

algorithm: variable
resources: variable

Published in 1989

TTL

LSI, MSI

μproc., memory

ASICs, accel's

© 2001, reiner@hartenstein.de

http://KressArray.de
Impact of Makimoto’s wave

Software Industry’s Secret of Success

Repeat Success Story by new Machine Paradigm!

Personalization (CAD) before fabrication

Procedural personalization via RAM-based Machine Paradigm

structural personalization: RAM-based before run time


© 2001, reiner@hartenstein.de
http://KressArray.de
Reconfigurable goes mainstream

Topic adopted by congresses: ASP-DAC, DAC, DATE, ISCAS, SPIE ....

• FCCM, FPGA (founded 1992), and FPL (founded 1991 at Oxford, UK): International Conference on Field-programmable Logic and Applications

• FPL 2002, La Grande Motte (Montpellier, France), Sept. 2 - 4

http://www.lirmm.fr/fpl2002/

FPL 2002: 214 Submissions - sensational increase by 83%
"...Adoption of VHDL was one of the biggest mistakes in the history of design automation, causing users and EDA vendors to waste hundreds of millions of dollars..."

— Joe Costello, Cadence Design Systems, 1995

"EDA industry paradigm switching every 7 years

82% of designers hate their tools

Makimoto’s 3rd wave

1978

Transistor entry: Applicon, Calma, CV ...

1985

Schematics entry: Daisy, Mentor, Valid ...

1992

Synthesis: Cadence, Synopsys ...

1999

(Co-) Compilation & data-stream-based (r)DPAs

2006

Paradigm Shift

Mainstream Tornado

82% of designers hate their tools

http://KressArray.de
Ignoring reconfigurable computing by teaching computing fundamentals within our CS curricula is one of the biggest mistakes in the history of information technology application causing the waste billions of dollars.
now the area is going mainstream: a rapidly widening audience of non-specialists gets interested...

severe communication gaps due to educational deficits

not only to users: still many hardware and EDA experts ask: isn't it just logic design on a strange platform?

it is time to clarify and popularize fundamental aspects and to explain, that it is a fundamentally different culture
• Introduction
• The CPU Machine Paradigm
• DPU & DPA: Antimatter of Computing
• Reconfigurable Computing
• Dominance of Embedded Systems
Reconfigurable or Hardwired: Datastream-based Computing is the Antimatter of classical Computing.

This Antimatter has been mainly ignored.

“Antimatter: where is it?”
Paul Dirac predicted a complete anti universe consisting of antimatter.

“There are regions in the universe, which consist of antimatter ..... But there are asymmetries”

when a particle hits its antiparticle, both are converted into energy: Annihilation

We are not aware, that there is a new area in computing sciences, which consists of antimatter of computing.

Reconfigurable Computing is made from this antimatter, especially if it’s data-stream-based.
• 1928: Paul Dirac: „there should be an anti electron having positive charge“ (Nobel price 1933)

• 1932: Carl David Anderson detected this „positron“ in cosmic radiation (Nobel price 1936)

• 1954: new accelerators: cyclotron, like Berkeley's Bevatron

• 1955 Owen Chamberlain et al. create anti proton on Bevatron

• 1956: anti neutron created on Bevatron

• 1965: creation of a deuterium anti nucleus at CERN

• 1995: hydrogen anti atom created at CERN - by forcing positron and anti proton to merge by very low energy.
The World of Matter
Machine paradigm: the Atom

Anti Matter
Machine paradigm: Anti Atom

Electron spinning
Positron spinning
Informatics: Matter & Antimatter: Machine and Anti Machine

CPU

1936 1st el. computer (Konrad Zuse)
1946 v. N. machine paradigm
1971 1st microprocessor (Ted Hoff)
1979 „data streams“ (systolic array: Kung / Leiserson)
1990 Xputer machine paradigm
1995 rDPA / DPSS (supersystolic: Rainer Kress)

Machine paradigm: „von Neumann“ Computer

DPU

Anti Machine

data-procedural: Xputer

© 2001, reiner@hartenstein.de

http://KressArray.de
The CPU Machine Paradigm

- Introduction
- The CPU Machine Paradigm
  - Concurrency
- DPU & DPA: Antimatter of Computing
  - DPA
- rDPA: Reconfigurable Computing
  - FPGA boom
- Dominance of Embedded Systems

http://www.uni-kl.de
CPU: RAM-based + simple machine paradigm + scalability + relocatability + compatibility = secret of success of software industry
Nasty Matter

CPU

Central Processing Unit

Data Path

Instruction sequencer

RAM

Instruction Fetch Overhead

Address Computation Overhead

central von Neumann bottleneck

the wrong machine paradigm

aiw. new instruction sequencer needed

reconfigurable?

extremely power hungry
and area inefficient

performance problems

© 2001, reiner@hartenstein.de http://KressArray.de

University of Kaiserslautern

Xputer Lab
Introduction

The CPU Machine Paradigm
- Concurrency

DPU & DPA: Antimatter of Computing
- DPA

rDPA: Reconfigurable Computing
- FPGA boom

Dominance of Embedded Systems
Parallelism by Concurrency

independent instruction streams
Concurrent Computing

- CPU extremely inefficient
- Bus(es) or switch box
- massive switching activity at runtime
- may affect far beyond Amdahl's law
<table>
<thead>
<tr>
<th>Company</th>
<th>Company</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACRI</td>
<td>DAPP</td>
<td>MasPar</td>
</tr>
<tr>
<td>Alliant</td>
<td>Denelcor</td>
<td>Meiko</td>
</tr>
<tr>
<td>American Supercomputer</td>
<td>Elexsi</td>
<td>Multiflow</td>
</tr>
<tr>
<td>Ametek</td>
<td>ETA Systems</td>
<td>Myrias</td>
</tr>
<tr>
<td>Applied Dynamics</td>
<td>Evans and Sutherland</td>
<td>Numerix</td>
</tr>
<tr>
<td>Astronautics</td>
<td>Computer</td>
<td>Prisma</td>
</tr>
<tr>
<td>BBN</td>
<td>Floating Point Systems</td>
<td>Tera</td>
</tr>
<tr>
<td>CDC</td>
<td>Galaxy YH-1</td>
<td>Thinking Machines</td>
</tr>
<tr>
<td>Convex</td>
<td>Goodyear Aerospace MPP</td>
<td>Saxpy</td>
</tr>
<tr>
<td>Cray Computer</td>
<td>Gould NPL</td>
<td>Scientific Computer Systems</td>
</tr>
<tr>
<td>Cray Research</td>
<td>Guiltech</td>
<td>Systems (SCS)</td>
</tr>
<tr>
<td>Culler-Harris</td>
<td>ICL</td>
<td>Soviet Supercomputers</td>
</tr>
<tr>
<td>Culler Scientific</td>
<td>Intel Scientific Computers</td>
<td>Supertek</td>
</tr>
<tr>
<td>Cydrome</td>
<td>International Parallel Machines</td>
<td>Supercomputer Systems</td>
</tr>
<tr>
<td>Dana/Ardent/ Stellar/Stardent</td>
<td>Kendall Square Research</td>
<td>Suprenum</td>
</tr>
<tr>
<td></td>
<td>Key Computer Laboratories</td>
<td>Vitesse Electronics</td>
</tr>
</tbody>
</table>
• Introduction
• The CPU Machine Paradigm
  – Concurrency
• DPU & DPA: Antimatter of Computing
  – DPA
• rDPA: Reconfigurable Computing
  – FPGA boom
• Dominance of Embedded Systems
Some differences: CPU versus DPU

- CPU: Data Path
  - Instruction sequencer
  - Instruction stream routed here
  - No vN bottleneck: multiple ports
  - No instruction fetch at run time: no overhead

- DPU: Data Path Unit
  - Data streams
  - Transport-triggered
  - RAM data sequencer
  - RAM data streams scheduled elsewhere
  - External signal or nothing central
“von Neumann” Computer: the wrong Machine Paradigm

Xputer: The Soft Machine Paradigm

there are some differences
data stream spec

Compiler
Scheduler
Datapath
Array

(multiple) sequencer

RAM

loosely coupled by decision data bits only

data counters

reconfigurable

also for hardwired

(anti machine)
### Machine Paradigms

<table>
<thead>
<tr>
<th>machine category</th>
<th>Computer (&quot;v. Neumann&quot;)</th>
<th>Xputer (no transputer!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>driven by:</td>
<td>Instruction streams</td>
<td>data streams (no “dataflow”)</td>
</tr>
<tr>
<td>engine principles</td>
<td>instruction sequencing</td>
<td>sequencing data streams</td>
</tr>
<tr>
<td>state register</td>
<td>program counter</td>
<td>(multiple) data counter(s)</td>
</tr>
<tr>
<td>communication path set-up</td>
<td>at run time</td>
<td>at load time</td>
</tr>
<tr>
<td>data path resource</td>
<td>DPU (e.g. single ALU)</td>
<td>DPU or DPA (DPU array) etc.</td>
</tr>
<tr>
<td>data path operation</td>
<td>sequential</td>
<td>parallel pipe network etc.</td>
</tr>
</tbody>
</table>
## Programming Language Paradigms

<table>
<thead>
<tr>
<th>language category</th>
<th>Computer Languages</th>
<th>Xputer Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>both deterministic</td>
<td>procedural sequencing: traceable, checkpointable</td>
<td>read next data item, goto (data addr.), jump (to data addr.), data loop, loop nesting, parallel loops, escapes, data stream branching</td>
</tr>
<tr>
<td>operation sequence driven by:</td>
<td>read next instruction, goto (instr. addr.), jump (to instr. addr.), instr. loop, loop nesting no parallel loops, escapes, instruction stream branching</td>
<td>read next data item, goto (data addr.), jump (to data addr.), data loop, loop nesting, parallel loops, escapes, data stream branching</td>
</tr>
<tr>
<td>state register</td>
<td>program counter</td>
<td>data counter(s)</td>
</tr>
<tr>
<td>address computation</td>
<td>massive memory cycle overhead</td>
<td>overhead avoided</td>
</tr>
<tr>
<td>Instruction fetch</td>
<td>memory cycle overhead</td>
<td>overhead avoided</td>
</tr>
<tr>
<td>parallel memory bank access</td>
<td>interleaving only</td>
<td>no restrictions</td>
</tr>
</tbody>
</table>

© 2001, reiner@hartenstein.de

http://KressArray.de
• Introduction

• The CPU Machine Paradigm
  - Concurrency

• DPU & DPA: Antimatter of Computing
  - DPA

• rDPA: Reconfigurable Computing
  - FPGA boom

• Dominance of Embedded Systems
nothing central!

Parallelism at data stream level
mapping before fabrication
no switching overhead at run time

configured after fabrication

reconfigurable interconnect fabrics
machine paradigm: some differences

CPU

DPA

DPU

no. of streams \geq 1

matter

antimatter

© 2001, reiner@hartenstein.de

http://KressArray.de
DPA = DPU array

coherent data streams spinning around
DPA-based entire System

memory communication architecture

mapping before fabrication

DPA

RAM

RAM

RAM

RAM

RAM

RAM

RAM

RAM
DPA-based entire System


reconfigurable memory communication architecture

mapping after fabrication

reconfiguration RAM

rDPA 1995  rDPA Rainer Kress
rDPA: Reconfigurable Computing

- Introduction
- The CPU Machine Paradigm
  - Concurrency
- DPU & DPA: Antimatter of Computing
  - DPA
- rDPA: Reconfigurable Computing
  - FPGA boom
- Dominance of Embedded Systems

http://www.uni-kl.de
XPU family:
PACT AG, Munich

http://pactcorp.com

Commercial rDPA

http://www.fpl.uni-kl.de
SNN filter KressArray Mapping

array size: 10 x 16 = 160 rDPUs à 32 bits

rout thru only

backbus connect

not used

http://kressarray.de

© 2001, reiner@hartenstein.de

http://KressArray.de
Super Pipe Networks

The key is *mapping*, rather than architecture.

<table>
<thead>
<tr>
<th>array</th>
<th>applications</th>
<th>pipeline properties</th>
<th>mapping</th>
<th>scheduling (data stream formation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>shape</td>
<td>resources</td>
<td></td>
</tr>
<tr>
<td><strong>systolic array</strong></td>
<td>regular data dependencies only</td>
<td>linear only</td>
<td>uniform only</td>
<td>linear projection or algebraic synthesis</td>
</tr>
<tr>
<td><strong>super-systolic DPA</strong></td>
<td>no restrictions</td>
<td></td>
<td>simulated annealing or P&amp;R algorithm</td>
<td>(e.g. force-directed) scheduling algorithm</td>
</tr>
</tbody>
</table>

*) KressArray [ASP-DAC-1995]
The FPGA boom

- Introduction
- The CPU Machine Paradigm
  - Concurrency
- DPU & DPA: Antimatter of Computing
  - DPA
- rDPA: Reconfigurable Computing
  - The FPGA boom
- The Dominance of Embedded Systems
Top 4 FPGA Manufacturers 2000

- Xilinx: 42%
- Altera: 37%
- Lattice: 15%
- Actel: 6%

Top 4 PLD Manufacturers 2000

- Altera: 37%

total: $3.7 Bio

Top 4 PLD Manufacturers 2000

- fastest growing semiconductor market segment
- PLD vendors’ and their alliances provide libraries of “soft IPs”

Configware Market
Soft rDPA?

• Rapid technology progress
• 50 million system gates soon
• FPGAs f. relocatable configware code?
• Compatibility at configuration code level?
• Slower clock: compensated by more parallelism: low power
• Even large rDPAs as a soft IP become feasible
• By >2005: don’t care about area efficiency?
The Dominance of Embedded Systems

- Introduction
- The CPU Machine Paradigm
  - Concurrency
- DPU & DPA: Antimatter of Computing
  - DPA
- rDPA: Reconfigurable Computing
  - FPGA boom
- The Dominance of Embedded Systems

http://www.uni-kl.de
Ubiquitous embedded systems

20 billion µprocessors (2001)
> 90% in embedded systems

10 times more programmers will write embedded applications than computer software by 2010

That’s where our graduates will go

Embedded systems means:

- **hardware** / **software** co-design
- **configware** / **software** co-design
- **hardware** / **configware** / **software** co-design
Annihilation?

avoidable by tools ....
Jürgen Becker’s **Co-DE-X Co-Compiler** supporting platform-based design

**Computer** machine paradigm

**Computer**

- **X-C** is C language extended by MoPL

- **Partitioner**
  - **GNU C compiler**
  - **Analyzer / Profiler**
  - **X-C compiler**

- **Xputer** machine paradigm

- **Loop Transformations**

- **Host Software**
  - **KressArray Configware**

- **DPSS**

- **Resource Parameters**

- **supporting different platforms**
Configware / Software Co-Design?  
Hardware / Software Co-Design?
However, current CS Education ….

Hardware invisible: 
under the surface

Algorithm

Software

procedural high level
Programming Language

Assembly Language

Hardware invisible: 
under the surface

Software Faculty Colleagues
shy away from the Paradigm Shift:
their Brain hurts? - can't be:
this Half has been amputated
Hardware and Software as Alternatives

procedural  structural

Algorithm

partitioning

Brain Usage: both Hemispheres

Hardware, Configware
Software

Hardware only
Software & Configware only
Dominance of the Submarine Model ...

(procedural) structurally disabled

... indicates, that our CS education system produces zillions of mentally disabled Persons

... completely disabled to cope with solutions other than software only

It's time to attack the software faculty dictatorship. Get involved!
thank you for your patience