Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

ICECS 2002

Reiner Hartenstein,
University of Kaiserslautern, Germany
http://hartenstein.de

Trends in Reconfigurable Logic and Reconfigurable Computing

>> Outline
- The Computer Architecture Crisis
- The Impact of Reconfigurable Platforms
- The Dichotomy of Models
- Parallelism
- Conclusions

Flag ship example: annual IEEE ISCA conference series
Statistics (David Padua, John Hennessy, et al.)

The Computer Architecture Crisis
- ACRI
- Alliant
- American Supercomputer
- Analytic
- Applied Dynamics
- Astronautics
- BBN
- CDC
- Convex
- Cray Computer
- Cray Research
- Culler Harris
- Culler Scientific
- Cybersome
- Data/Aden/Stanford

The Impact of Reconfigurable Platforms
- DAPP
- Danecor
- Easel
- ETA Systems
- Evans and Sutherland
- Computer
- Floating Point Systems
- Galaxy/TH1
- Goodyear Aerospace MPP
- Gould/MPL
- Galax
- ICL
- Intel Scientific Computers
- International Parallel Machines
- Kendall Square Research
- Key Computer Laboratories

The Dichotomy of Models
- MassPar
- Meko
- Multiflow
- Amd
- Numeric
- Pram
- Tera
- Thinking Machines
- Saegy
- Scientific Computer Systems (SCS)
- Soviet Supercomputers
- Superbyte
- Supercomputer Systems
- Supersym
- Vitesse Electronics

Parallelism
- CS: young? dynamic?
- but the von Neumann Paradigm is still the dominant doctrine...
- still pushing he basic models from the times of mainframe dinosaurs
- Microelectronics is ignored (except falling cost of computational effort)
- Re-orientation silicon conserv.
- computing sciences are ultra conservative...

© 2002, reiner@hartenstein.de
http://KressArray.de

University of Kaiserslautern
Xputer Lab

© 2002, reiner@hartenstein.de
http://KressArray.de

http://www.uni-kl.de

CS: young? dynamic?
- after >10 technology generations...
- the von Neumann Microprocessor is a methusela, the steam engine of the silicon age.
- Re-orientation silicon conserv.

Dead Supercomputer Society
(Gordon Bell, keynotes at ISCA 2000)

- ACRI
- Alliant
- American Supercomputer
- Analytic
- Applied Dynamics
- Astronautics
- BBN
- CDC
- Convex
- Cray Computer
- Cray Research
- Culler Harris
- Culler Scientific
- Cybersome
- Data/Aden/Stanford

© 2002, reiner@hartenstein.de
http://KressArray.de

http://www.uni-kl.de

© 2002, reiner@hartenstein.de
http://KressArray.de

http://www.uni-kl.de

>> Paradigm Shifts
- The Computer Architecture Crisis
- The Impact of Reconfigurable Platforms
- The Dichotomy of Models
- Parallelism
- Conclusions

© 2002, reiner@hartenstein.de
http://KressArray.de

http://www.uni-kl.de

© 2002, reiner@hartenstein.de
http://KressArray.de

http://www.uni-kl.de
Reiner Hartenstein, University of Kaiserslautern, Germany  
http://hartenstein.de

- Better to go for reconfigurable platforms
  - Fastest growing segment of semiconductor market
  - TP reuse and silicon reuse
  - FPGAs are going into every type of application

Why coarse grain?

Throughput vs. Efficiency

Throughput vs. Flexibility

Terminology

Paradigm Shifts: Nick Tredennick's view

Nick Tredennick's view
Reiner Hartenstein, University of Kaiserslautern, Germany  
http://hartenstein.de

---

**Compilation for (r)DPA of anti machine**

- **streamware**
  - high level source program (software notation)
  - parameters
  - wrapper
  - morphware
  - DPU library
  - configware
  - code generators
  - scheduler

---

**Why fine grain?**

- no specific silicon: low production volume (aerospace, automotive, military, industrial controllers, et al.)
- the spare part problem
- design flow

---

**Evolution of FPGA and its design flow**

- as soon as F16 FPGA is available

---

**软 CPU 的实例**

- **core**
  - architecture
  - platform

- **core**
  - architecture
  - platform

---

**软 CPU 在学术教学中**

- UCSC 1990
- Marijuana University
- Chalmers University
- Cornell University
- Gray Research
- Georgia Tech
- Hiroshima City Univ.
- Michigan State
- Univ. de Valladolid
- Virginia Tech
- Washington U. St. Louis
- New Mexico Tech
- UC Riverside
- Tokai University

---

**ASIC emulation**

- ASIC emulation / Rapid Prototyping: to replace simulation
- Quickturn (Cadence), IKOS (Synopsys), Celero (Mentor)
- hours of compilation run: inefficient since netlist-based: ...
- ASIC emulators will become obsolete soon
- by RTR: in-circuit execution debugging instead of emulation
- new business model: upgradable morphware is the product
- emulation for solving the spare part problem in many areas
The microelectronics spare part problem

- Demand: several decades of availability
- e.g. car price: ~25% electronics
- ICs do not survive storage time
- Original fab line is no more existing

The microelectronics spare part problem

- Demand: several decades of availability
- ICs do not survive storage time
- Original fab line is no more existing

The dichotomy of models

- The computer architecture crisis
- The impact of reconfigurable platforms
- The dichotomy of models
- Parallelism
- Conclusions

Matter & Antimatter

- The world of anti matter: machine paradigm: Anti Atom
- Electron spinning
- The world of matter: machine paradigm: the Atom
- Positron spinning

Matter & Antimatter of informatics:

- Anti machine paradigm
- Instruction stream spinning (von Neumann)
- Nothing central!
- Data stream spinning

Computing paradigms and methodologies:

- Instruction-stream-based vs. data-stream-based

1946: machine paradigm (von Neumann)
1980: data streams (Kung, Leiserson)
1989: anti machine paradigm introduced
1990: anti machine implementation methodology
1990: rDPU (Rabaey)
1994: anti machine high level programming language
1995: super systolic rDPA (Kress)
1996+: SCCG (LANL), SCORE, ASPRC, Bee (UCB), ...
1997: configure / software partitioning compiler (Becker)
2000: generator for rDPA with high memory bandwidth (tutorials and courses available on all this)
Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

Nasty Matter

- Performance Problems: extremely power hungry and area inefficient
- von Neumann bottleneck

DPU

Matter vs. Antimatter: CPU vs. DPU

- DPU: Data Path Unit
- DPU: Data Path Unit

Success Factors

- RAM-based
  - Instruction stream based
  - Coarse grain
  - Available
  - Available
  - Feasible
  - Feasible

- Machine paradigm
  - Concurrency
  - Available
  - Available
  - Feasible
  - Feasible

- Compatibility
  - RAM-based
  - Hardwired
  - Hardwired
  - Hardwired

- Scalability
  - RAM-based
  - Hardwired
  - Hardwired
  - Hardwired

- Code relocatability
  - RAM-based
  - Hardwired
  - Hardwired
  - Hardwired

- FOR configware industry is missing:
  - FPGA compatibility,
  - Fully scalable FPGA,
  - Reconfigurable configuration code

- rDPUs and rDPAs do much better than FPGAs

Success of software industry

CPU: RAM-based

- Simple machine paradigm
- Scalability
- Relocatability
- Compatibility
= Secret of success of software industry

>> Problems with Concurrency

- The Computer Architecture Crisis
- The Impact of Reconfigurable Platforms
- The Dichotomy of Models
- Parallelism
- Conclusions

http://www.uni-kl.de
Parallelism by Concurrency
independent instruction streams

Difficult coordination

Bus(es) or switch box
massive run time overhead

Data-stream-based Parallelism

See my other talk

Memory Organisation for Data-stream-based Reconfigurable Computing

The Dominance of Embedded Systems

• The Computer Architecture Crisis
• The Impact of Reconfigurable Platforms
• The Dichotomy of Models
• Parallelism
• Conclusions

Summary of the Anti Machine Paradigm

• anti language primitives are almost the same (slightly extended)
• anti machine execution potential is dramatically more powerful
• provides drastically more flexibility
• not always replacing von Neumann

Conclusions

• the anti machine is the way to go for massive parallelism, also data-intensive applications
• reconfigurable anti machine for high performance with short product life cycles, unstable standards
• reconfigurable for low cost low volume production
• sparepart problem: needs new infrastructures
• Giga FPGAs highly promising - only by a new design flow: configure could repeat the success of software industry

Thank you for your patience

Computing Sciences are in a severe crisis

CS ignores > 90% µprocessors in embedded systems: languages hardware parallel memory IP core and module generator vendors

keynotes 2001 / 2002

Computing curricula are obsolete because of strictly compilation techniques

SBCCI, Brasilia, 2001
DATE. Munich, 2001
invited talks 2001 / 2002

sequencing methodology: hw & sw

Computer Architecture and related areas have hw / sw partitioning methodology

Xputer Lab


Machine Paradigms

<table>
<thead>
<tr>
<th>machine category</th>
<th>Computer (the Machine: &quot;v. Neumann&quot;)</th>
<th>The Anti Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>driven by</td>
<td>instruction streams</td>
<td>data streams</td>
</tr>
<tr>
<td>engine principles</td>
<td>instruction sequencing</td>
<td>sequencing data stream</td>
</tr>
<tr>
<td>state register</td>
<td>single program counter</td>
<td>(multiple) data counter(s)</td>
</tr>
<tr>
<td>Communication path setup (&quot;instruction fetch&quot;)</td>
<td>at run time</td>
<td>at load time</td>
</tr>
<tr>
<td></td>
<td>resource</td>
<td>DPU (e.g. single ALU)</td>
</tr>
<tr>
<td></td>
<td>operation</td>
<td>sequential</td>
</tr>
</tbody>
</table>

programming Language paradigms

<table>
<thead>
<tr>
<th>language category</th>
<th>Computer Languages</th>
<th>Languages f. Anti Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>both deterministic</td>
<td>procedural sequencing</td>
<td>procedural sequencing</td>
</tr>
<tr>
<td>operation sequence</td>
<td>instruction driven by: data item</td>
<td>instruction driven</td>
</tr>
<tr>
<td></td>
<td>next instruction, data item</td>
<td>next instruction, data item</td>
</tr>
<tr>
<td></td>
<td>jump (data item), loop nesting</td>
<td>jump (data item), loop nesting</td>
</tr>
<tr>
<td></td>
<td>parallel loop, escape, data stream</td>
<td>parallel loop, escape, data stream</td>
</tr>
<tr>
<td></td>
<td>instruction branch</td>
<td>instruction branch</td>
</tr>
<tr>
<td></td>
<td>parallel pipe network etc.</td>
<td>parallel pipe network etc.</td>
</tr>
</tbody>
</table>

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

Reiner Hartenstein, University of Kaiserslauern, Germany http://hartenstein.de
Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

The Situation in Computing Sciences

- Computing Sciences are in a severe crisis
- New fundamentals and R&D directions are inevitable
- My mission: getting you involved
- All knowledge needed is readily available...
- ...even from Computing Sciences
- Silicon application and EDA provide useful concepts
- Reconfigurable Computing has the remedy

The edu gap has dramatic consequences

- Key R&D scenes are drying out or dying
- Because of a lack of qualified researchers
- The embedded system design crisis gets worse
- Because of a lack of qualified designers
- Many innovative products cannot be sold
- Because of a lack of qualified customers
- The edu gap is widening dramatically
- Because of a lack of qualified educators

Super Pipe Networks

The key is mapping, rather than architecture

array applications pipeline properties mapping scheduling (data stream formation)

systolic array regular data dependencies only linear only linear projection or algebraic synthesis

regular data dependencies only uniform only linear projection or algebraic synthesis

supersystolic DPA no restrictions simulated annealing or PSR algorithm (e.g. force-directed) scheduling algorithm

---

Jürgen Becker’s Co-DE-X Co-Compiler
supporting platform-based design

Computer machine paradigm

Partitioner

GNU C compiler

Analyzer/Profiler

X-C compiler

X-C is C language extended by MoPL

Loop Transformations

supporting different platforms

Resource Parameters

Configware

KressArray Configure

Host Software

DPSS

Impact of Makimoto’s wave


Procedural personalization via RAM-based Machine Paradigm

Software Industry’s Secret of Success

Repeat Success Story by new Machine Paradigm!
Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

“von Neumann” Computer: the wrong Machine Paradigm

Xputer: The Soft Machine Paradigm

Reconfigurable semiconductor market

Semiconductor Revolutions

Impact of Makimoto’s wave

Impact of Data-stream-based

Embedded Hardware/Configware Industry

Configware Industry

Repeat Success Story by new Machine Paradigm

Qualified people are not available

Qualified people are not available

Software Industry’s Secret of Success

Procedural personalization via RAM-based Machine Paradigm

Procedural personalization via RAM-Based Machine Paradigm
Reiner Hartenstein, University of Kaiserslautern, Germany
http://hartenstein.de

Rapidly growing CS education gap

- Our computing curricula are obsolete
- Introduction is strictly "procedural-only"
- VN-only use of terms like "computer organisation", "computer structures", "computer architecture"
- Graduates are not prepared to the real world
  - Most applications for embedded systems (>90% by 2010)
- Our graduates are unable to compete with EE graduates
- Only a few % curricula need to be changed
- My mission: getting you involved

Why Coarse Grain instead of FPGA?

Timing vs. Computing Domain

- Binding time (setup of communication channels at run time)
- Compilation time (at load time)
- Fabrication time (before load time)

- Reconfigurable computing: computing in space and time
- "Instruction" fetch at compile time
- Structural programming
- Data scheduling
- Also hardware implementations
- "Instruction" fetch before fabrication

What are the differences?

- VN* computing:
  - Computing in time
  - Instruction fetch at run time
  - Procedural programming
  - Instruction scheduling
  - Also hardwired implementations**

- Reconfigurable computing:
  - Computing in space and time
  - "Instruction" fetch at compile time
  - Structural programming
  - Data scheduling
  - I.e. Data-stream-based

---

Sources: Proc ISSCC, ICSPAT, DAC, DSPWorld

Why Coarse Grain instead of FPGA?

- Physical vs. Logical
- Reconfigability
- Loading time vs. Benefits

- VN computing:
  - Computing in time
  - Instruction fetch at run time
  - Procedural programming
  - Instruction scheduling

- Reconfigurable computing:
  - Computing in space and time
  - "Instruction" fetch at compile time
  - Structural programming
  - Data scheduling
  - I.e. Data-stream-based
  - Also hardwired implementations**

---

Basics of Binding Time

- "Instruction" generalized:
  - Including complex expressions and other datapaths
  - Strong impact on the machine paradigm!

- Time of "Instruction Fetch"
  - Run time
  - Loading time
  - Compile time

---