Reconfigurable Supercomputing: What are the Problems? What are the Solutions?
The Supercomputing Paradox

Promising technology

COTS processor decreasing cost
Increasing number of processors running in parallel
Rapidly growing listed Teraflops
Almost stalled application implementation progress
Often limited sustained Teraflops
Very high total cost of the Tera(?)flops
Scientists waiting for affordable compute capacity
dangerously telling this to the supercomputing people:

You ... used the wrong roadmap the past 20 years !!!
N.N.: „Innovation for Prosperity“ (2)

shrinking time-to-insight?

However, time-to-insight .... stalled for more than a decade in high performance computer architecture
3 Reconfigurable Computing Paradoxes

Reconfigurable Computing Education Paradox

The low power paradox

The high performance paradox
The Pervasiveness of RC

search “FPGA and ....”

Reconfigurable Embedded Computing
- pattern recognition
- signal processing
- video, vision
- radar, sonar
- wireless
- control
- coding
- crypto
- music
- fuzzy
- video
- HDTV
- defense
- aerospace
- automotive
- multimedia
- manufacturing
- image processing

mainly experts with hardware background

Hardware / Configure / Software Co-Design

Embedded Systems Applications

Reconfigurable Scientific Computing
- artificial intelligence
- environmental
- mathematics
- mechanics
- petroleum
- vector
- bio
- DNA
- genetic
- weather
- chemistry
- molecular
- oil and gas
- astrophysics
- fluid dynamics
- neural network
- crash simulation

mainly experts without hardware background

Software / Configure Co-Compilation

Scientific Computing Applications

# of hits by Google

647,000
1,490,000
398,000
1,620,000
915,000
272,000

171,000
194,000
127,000
113,000
158,000
162,000
Almost 10 million hits

<table>
<thead>
<tr>
<th>keyword</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td>9,960,000</td>
</tr>
<tr>
<td>Reconfigurable Computing</td>
<td>256,000</td>
</tr>
<tr>
<td>Configware</td>
<td>13,400</td>
</tr>
<tr>
<td>(Kress/Kung) Anti Machine</td>
<td>18,800</td>
</tr>
</tbody>
</table>

Fig. 1. (Jan. 2006) found almost 10 million times

<table>
<thead>
<tr>
<th>FPGA and ...</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>... embedded</td>
<td>3,280,000</td>
</tr>
<tr>
<td>... wireless</td>
<td>1,490,000</td>
</tr>
<tr>
<td>... automotive</td>
<td>915,000</td>
</tr>
<tr>
<td>low power</td>
<td>541,000</td>
</tr>
<tr>
<td>... medical</td>
<td>710,000</td>
</tr>
<tr>
<td>... music</td>
<td>398,000</td>
</tr>
<tr>
<td>... physics</td>
<td>508,000</td>
</tr>
<tr>
<td>... chemical</td>
<td>247,000</td>
</tr>
<tr>
<td>... defense</td>
<td>287,000</td>
</tr>
<tr>
<td>... bio</td>
<td>140,400</td>
</tr>
<tr>
<td>... weather</td>
<td>128,000</td>
</tr>
<tr>
<td>... chemistry</td>
<td>115,800</td>
</tr>
<tr>
<td>... molecular</td>
<td>113,000</td>
</tr>
<tr>
<td>high performance</td>
<td>706,000</td>
</tr>
<tr>
<td>supercomputing</td>
<td>65,800</td>
</tr>
<tr>
<td>n body problem</td>
<td>28,200</td>
</tr>
<tr>
<td>... oil and gas</td>
<td>22,300</td>
</tr>
</tbody>
</table>

Fig. 2. (Jan. 2006) Going to every application area.
We now also have the hardware / configware / software chasm.

Curricula still ignore these extremely hot new challenges.

The Reconfigurable Computing Education Paradox:

its run-away accelerated pervasiveness, despite of all these educational deficits.
Within about 500 pages the term reconfigurable is not found – nor its synonyms.
von Neumann's monopoly inside curricula is obsolete
von Neumann is not the common model

software

microprocessor age:

instruction-stream-based

data-stream-based

co-processors

hardware

morphware

mainframe age:

von Neumann instruction-stream-based machine

von Neumann bottleneck

von Neumann program counter

von Neumann RAM memory

vN paradigm dominance?

the tail is wagging the dog

© 2006, reiner@hartenstein.de
modern FPGA bestsellers:

The new model is reality: FPGA fabrics, together with several µprocessors, several memory banks, and other IP cores, on the same COTS microchip.
Speech by Bill Gates at a summit meeting of US state governors:
"American high schools are obsolete."

"The high schools of today teach kids about today's computers like on a 50-year-old mainframe.

„Without re-design for the needs of the 21st century, we will keep limiting - even ruining - the lives of millions of Americans every year."
The most important cultural revolution since the invention of text characters:

it's not the mainframe

It is the Microchip!
<table>
<thead>
<tr>
<th>RC education</th>
<th>International Workshop on Reconfigurable Computing Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td></td>
</tr>
</tbody>
</table>

The 1st International Workshop on Reconfigurable Computing Education
March 1, 2006, Karlsruhe, Germany

35 submissions from
Australia, Brasil, India, USA, and throughout Europe

http://fpl.org/RCeducation/
Reconfigurable Computing Paradoxes

Reconfigurable Computing Education Paradox

The low power paradox

The high performance paradox
The awful technology of FPGAs:

„very power-hungry“ [Rick Kornfeld*]

FPGAs run at lower clock frequencies, draw much more power and are more expensive.

*) personal communication

Reducing the electricity bill by an order of magnitude and more by supercomputer 2 FPGA migration
telling this to the low power design people?

you ... used the wrong roadmap
the past 15 years: use FPGAs!

ISLPED,
Oct 4 - 6,
Tegernsee

PATMOS,
Sep 13 - 15,
Montpellier

1991: Kaiserslautern, Germany
1992: Paris, France
1993: Montpellier, France
Reconfigurable Computing Paradoxes

Reconfigurable Computing Education Paradox

The low power paradox

The high performance paradox
The High Performance Paradox

The awful technology of FPGAs:

Effective integration density **much worse** than the Gordon Moore curve: by a factor of more than 10,000

FPGAs run at lower clock frequencies, and are more expensive.

85% of all designers hate their tools
fine-grained RC: 1st DeHon's Law

[1996: Ph. D, MIT]

transistors / microchip

10^9

10^6

10^3

10^0

1980 1990 2000 2010

FPGA routed

FPGA logical

FPGA physical

density: overhead:

wiring overhead

reconfigurability overhead

>> 10,000

immense area inefficiency
coarse-grained RC: Hartenstein's Law

area efficiency very close to Moore's law

[1996: ISIS, Austin, TX]
Claassen's Law

MOPS / milliWatt

- hardwired
- FPGAs (fine grained reconf.)
- instruction set processors
- standard microprocessor

µ feature size

© 2006, reiner@hartenstein.de
http://hartenstein.de
Selection of published speed-up factors

Image processing, Pattern matching, Multimedia

DSP and wireless

Bioinformatics

Astrophysics

Microprocessor

Memory

1980 1990 2000 2010

10^9

10^6

10^3

10^0

Grid-based DRC ("fair comparison")

SPIHT wavelet-based image compression

video-rate stereo vision

pattern recognition

real-time face detection

Los Alamos traffic simulation

protein identification

Blast

molecular dynamics simulation

Lee Routing (DPLA by TU-KL)

2-D FIR filter (no FPGA: DPLA by TU-KL)

FFT

100 000

50%/yr

7%/yr

MoM architecture

Xputer architecture

100 000

x 2/yr

1000

100

http://xputers.informatik.uni-kl.de/faq-pages/fqa.html

http://hartenstein.de
Computational Density

The computational capacity of FPGAs exceeds that of CPUs and the gap is *increasing.*
The three RC Paradoxes

- poor technology
- poor tools
- very poor education

brilliant results
Why supercomputing / HPC failed

because of the interconnect network architecture
the wrong way, how the data are moved around
instruction-stream-based: memory-cycle-hungry
instruction fetch overhead
sequencing overhead
address computation overhead
and other overhead

The law or More:

- Researchers spending more time on software development
- Problem will get more acute as the number of processors increases
- Software done...Machine obsolete
moving data around inside the Earth Simulator

Crossbar weight: 220 t, 3000 km of cable,

ES 20: TFLOPS

5120 Processors, 5000 pins each
data moved around by software

i.e. by memory-cycle-hungry instruction streams which fully hit the memory wall

P&R: move locality of operation, not data!

extremely unbalanced

stolen from Bob Colwell

http://laughingriot.incide.net
Progress stalled by the software/configware chasm

Useful simple archetype not widely accepted

An archetype common model should provide ....

Guidance for organizing efficient solutions

Make the project manageable

Allow to share lessons between applications and between disciplines

support undergraduate education
The new paradigm: how the data are traveling

no, not by instruction execution
transport-triggered: an old hat
pipeline, or chaining
asynchronous (via handshake)
systolic array
wavefront array
Flowware defines:... which data item at which time at which port

(pipe network) DPA

Def.: data streams (flowware)

(input data streams)

(output data streams)

source and sink?
Data streams source and sink: not my job

Not my Job!
distributed memory

input data streams

output data streams

implemented by distributed on-chip memory

On-chip Auto-Sequencing Memory

ASM

GAG

RAM

© 2006, reiner@hartenstein.de

http://hartenstein.de
How the data are moved

DMA,

vN move processor [Jack Lipovski, EUROMICRO, Nice, 1975]

ASM use **GAG** generic address generator [TU-KL publ.: Tokyo 1989 + NH journal]

by the way: **GAG** st.... by TI [TI patent 1995]

Henk Corporaal coins the term “transport-triggered”

MoM: **GAG**-based storage scheme methodology [Herz*]

Application-specific distributed memory [Catthoor et al.]

*) [see Michael Herz et al.: ICECS 2002 (Dubrovnik)]
The dual paradigm approach

von Neumann paradigm

Kress-Kung paradigm

Software Engineering

Configware Engineering

© 2006, reiner@hartenstein.de

http://hartenstein.de
algebraic methods

i. e., linear projections

yields only uniform arrays w. linear pipes

only for applications with regular data dependencies
Coarse-grained reconfigurable arrays are a Generalization of the Systolic Array ....

discard algebraic synthesis methods

use optimization algorithms instead, for example: simulated annealing

the achievement: also non-linear and non-uniform pipes, and even more wild pipe structures possible

now reconfigurability really makes sense
Coarse grain is about computing, not logic.

Example: mapping onto rDPA by DPSS: based on simulated annealing.

SNN filter on KressArray (mainly a pipe network)

Array size: 10 x 16 = 160 rDPUs

No CPU

rDPU, 32 bit

Tool: KressArray Xplorer: diss. Ulrich Nageldinger (downloadable)
Software / Configware Co-Compilation

[Juergen Becker’s CoDe-X, 1996]

C language source

Partitioner

SW compiler

Analyzer / Profiler

CW compiler

FW Code

Resource Parameters

supporting different platforms

simulated annealing

"vN" machine paradigm

anti machine paradigm

SW code

CW Code

FW Code

© 2006, reiner@hartenstein.de

http://hartenstein.de
Software / Configware Co-Compilation

[Juergen Becker’s CoDe-X, 1996]

C language source

For thesis see book exhibit rack at library entrance

“vN” machine paradigm

simulated annealing

supporting different paradigms

Partitioner

Analyzer / Profiler

SW compiler

CW compiler

SW code

CW Code

FW Code

Resource Parameters

© 2006, reiner@hartenstein.de

http://hartenstein.de
Distributed Memory Parallelism Capability

example: 10 x 16
Applications for coarse-grained arrays

(on-chip distributed memory for intermediate results)

with steady I/O data streams at constant speed:

Multi-standard world HDTV receiver

Wide variety of multimedia applications

Wide variety of real-time applications

Many other applications
The wrong mind set ....

„but you can't implement decisions!“

(remark of a high-ranked industrial research head - discussion after a talk by Ulrich Nageldinger - RAW Orlando)
a tiny section of the pipe network
The wrong mind set ....

section of a very large pipe network:

not knowing this solution:

“but you can't implement decisions!”

symptom of the hardware / software chasm

and the configware / software chasm
introducing hardware description languages
(in the mid' seventies)

“The decision box becomes a (de)multiplexer”

This is so simple: why did it take decades to find out?

The wrong mind set – the wrong road map!
hypothesis example to illustrate software-to-configware migration

\[ S = R + (\text{if } C \text{ then } A \text{ else } B \text{ endif}); \]

C = 1

simple conservative CPU example

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Memory Cycles</th>
<th>Nano Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>read instruction</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>if C then read A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>instruction decoding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read operand*</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>operate &amp; reg. transfers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>if not C then read B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read instruction</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>instruction decoding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>add &amp; store</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read instruction</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>instruction decoding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>operate &amp; reg. transfers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>store result</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>total</td>
<td>5</td>
<td>500</td>
</tr>
</tbody>
</table>

*) if no intermediate storage in register file

section of a major pipe network on rDPU

no memory cycles; speed-up factor = 100

clock 200 MHz (5 nanosec)
why the RC paradigm shift is so important

Move the stool or the grand piano?

by Software

by Configware
... understand only this parallelism solution:

the instruction-stream-based approach

the data-stream-based approach has no von Neumann bottleneck

... cannot cope with this one

von Neumann bottlenecks

http://hartenstein.de
What means Reconfigurable Computing?

- switching the multiplexers?
- routing ALU result to a register?
- microprogramming?
- concurrency of 64 or 256 CPUs on a single chip?
- it means using the Kress/Kung machine paradigm!
vN paradigm loosing its dominance

RAMP project proposes:
Run LINUX on FPGAs
vN paradigm loosing its dominance

Xilinx inside!
Recommended Pentium successor

Discard most caches
Have 64* cores
with clever interconnect for:
concurrent processes,
for multithreading, and,
Kung-Kress rDPA array

The Desk-top Supercomputer!
What means Reconfigurable Computing?

The key issue: which is the underlying paradigm?

Operation **not** based on instruction-streams at run time

No instruction fetch at run time

Machine paradigm is data stream-based: Kress-Kung

Undergraduate education needs a dual paradigm approach: symbiosis of von Neumann / Kress-Kung
thank you
END
Backup for Discussion:
Term to be used for „soft hardware“
unfortunately “Morphware” is trademarked

<table>
<thead>
<tr>
<th>accelware</th>
<th>gateware</th>
<th>performware</th>
<th>shuntingware</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptware</td>
<td>gateroutware</td>
<td>perfware</td>
<td>speedware</td>
</tr>
<tr>
<td>adjustware</td>
<td>hpcware</td>
<td>perware</td>
<td>speedupware</td>
</tr>
<tr>
<td>altware</td>
<td>LUTware</td>
<td>pipeware</td>
<td>suiteware</td>
</tr>
<tr>
<td>alterware</td>
<td>matchware</td>
<td>platformware</td>
<td>switchware</td>
</tr>
<tr>
<td>arrangeware</td>
<td>modiware</td>
<td>railware</td>
<td>switchingware</td>
</tr>
<tr>
<td>changeware</td>
<td>morphware®</td>
<td>rangeware</td>
<td>streamware</td>
</tr>
<tr>
<td>conformware</td>
<td>morfware</td>
<td>RCware</td>
<td>structware</td>
</tr>
<tr>
<td>doughware</td>
<td>mouldware</td>
<td>ressourceware</td>
<td>transferware</td>
</tr>
<tr>
<td>fabricsware</td>
<td>muxware</td>
<td>routware</td>
<td>transware</td>
</tr>
<tr>
<td>fabrixware</td>
<td>parware</td>
<td>routewear</td>
<td>variware</td>
</tr>
<tr>
<td>fitware</td>
<td>paraware</td>
<td>routingware</td>
<td>varyware</td>
</tr>
<tr>
<td>flexware</td>
<td>passware</td>
<td>RTware</td>
<td>warpware</td>
</tr>
<tr>
<td>formware</td>
<td>pathware</td>
<td>shapeware</td>
<td>xferware</td>
</tr>
<tr>
<td>FPWare</td>
<td>patchware</td>
<td>shuntware</td>
<td>xware</td>
</tr>
</tbody>
</table>

send your proposal to:
reiner@hartenstein.de
Compilation: Software vs. Configware

Software Engineering

source program

software compiler

software code

Configware Engineering

source "program"

mapper

configware code

data

scheduler

flowware code

placement & routing

C, FORTRAN, MATHLAB

© 2006, reiner@hartenstein.de

http://hartenstein.de

TU Kaiserslautern

62
Co-Compilation

C, FORTRAN, MATHLAB

software compiler

Software / Configware Co-Compiler

automatic SW / CW partitioner

mapper

configware compiler

data scheduler

software code

configware code

flowware code

© 2006, reiner@hartenstein.de
Why use Reconfigurable Computing instead of software? instead of spec. hardware?

Exploit spatial parallelism, and ..

... high bandwidth and low latency memory access

... and fine-grained parallelism when useful

Ride the technology curve avoiding specific silicon

Adapt to change: standards, trends, ..... 

Adapt to application / deployment requirements

Reduce risk
Pre-1990s:

EE+ CE
HARDWARE

CS
SOFTWARE

IS
BUSINESS

Post-1990s:

EE
HARDWARE

CE

CS
SOFTWARE

SE

IT

IS

ORGANIZATIONAL NEEDS

Figure 2.1. Harder choices: How the disciplines might appear to prospective students.
2.2.1. … how it should be

CONFIGWARE

MORPHWARE

and configware added
FP7 Timetable

- 6th April 2005  Commission Proposal on FP7
- May-June 2005  Thematic Consultations
- 21 Sept 2005  Commission Proposal on Specific
                 Programmes and Rules for participation
                 Impact Assessment
- 23 Nov 2005  Commission Proposal on Art 169 and 171
- 1st half of 2006  Council and Parliament debate
- July 2006  Council and Parliament adoption
- October 2006  Commission proposal on Workprogrammes
- November 2006  Publication of first calls for proposals