Reconfigurable Computing
The Pervasiveness of RC

ECE-savvy scene (mainstream many years)

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

Hardware / Configure / Software Co-Design

Embedded Systems Applications
- and many more areas

mainly experts with hardware background

Math/SW-savvy scene (more recently: 2-3 years)

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

Software / Configure Co-Compilation

Scientific Computing Applications
- and many more areas

mainly experts without hardware background

# of hits by Google
- FPGA and ...
- 647,000
- 1,490,000
- 398,000
- 1,620,000
- 272,000

© 2006, reiner@hartenstein.de

2
The dominance of Configware

Most compute power is coming from Configware

More MIPS migrated to Configware than running as Software
Reconfigurable Supercomputing (VHPC) going commercial

Cray XD1

silicon graphics RASC

... and other vendors
Outline

- Reconfigurable Computing Paradox
- The Supercomputing Paradox
- We are using the wrong model
- Coarse-grained Reconfigurable Devices
- Super Pentium for Desktop Supercomputer
The Reconfigurable Computing Paradox

**poor FPGA technology:**
area-inefficient, slow, power-hungry, expensive

**poor tools:**
tools and languages unacceptable by most users
even most hardware experts (86%**) hate their tools

**poor education:** RC education: extremely poor, if at all
- ignored by CS curricula

CS taught like for a 50 year old mainframe ...

**) DeHon '98**
FPGA integration density

what paradox?

However, brilliant results everywhere

the effective integration density of plane FPGAs is behind Moore’s law by more than 4 orders of magnitude
platform FPGAs: better area efficiency

DeHon's 1st Law (1996) was for plane FPGAs

DSP platform FPGA [courtesy Xilinx Corp.]

- 500MHz PowerPC™ Processors (680DMIPS) with Auxiliary Processor Unit
- 500MHz Flexible Soft Logic Architecture 200K Logic Cells
- 1Gbps Differential I/O
- 500MHz Programmable DSP Execution Units
- 500MHz multi-port Distributed 10 Mb SRAM
- 500MHz DCM Digital Clock Management
- 0.6-11.1 Gbps Serial Transceivers
pre FPGA era: Why DPLA* was so good

1. Large arrays of canonical boolean expressions - classical PLA layout highly area-efficient: close to Moore's law

Mid' 80ies: first only very tiny FPGAs available: 1 DPLA replaced 256 of them

2. ASM: A generalization of the DMA**

Speed-up factor of 20 by reducing memory cycles which is the key issue

**) for a survey by IMEC & TU-KL see: [M. Herz et al.: ICECS 2003, Dubrovnik]

GAG Generic Address Generator** to avoid address computation overhead

*) fabricated 1984 by E.I.S. multi university project

© 2006, reiner@hartenstein.de
taxonomy of algorithms, better tools and better education
New dimensions of low power:

Application migration [from supercomputer] resulting not only in massive speed-ups

Electricity bills reduced by an order of magnitude and even more you may get for free

…. up to millions of $ dollars per year

„Saves more than $10,000 in electricity bills per year (7¢ / kWh) - …. per 64-processor 19" rack“

[Herb Riley, R. Associates]

(also a matter of national energy policy)
• Reconfigurable Computing Paradox
• The Supercomputing Paradox
• We are using the wrong model
• Coarse-grained Reconfigurable Devices
• Super Pentium for Desktop Supercomputer
The Supercomputing Paradox

Growing listed Teraflops

Increasing number of processors running in parallel

COTS processor decreasing cost

promising technology
HPC by classic supercomputing methodology

poor results

Extreme shortage of affordable capacity

More parallelism absorbs programmer productivity

Program ready: hardware obsolete

The law of More

Not for high performance embedded computing

Lack of scalability: progress only by innovation
• Reconfigurable Computing Paradox
• The Supercomputing Paradox
• We are using the wrong model
• Coarse-grained Reconfigurable Devices
• Super Pentium for Desktop Supercomputer
Why traditional supercomputing / HPC failed

because of the wrong multi-core interconnect architecture

memory-cycle-hungry instruction-stream-based:

the wrong way, how the data are moved around

stolen from Bob Colwell
http://laughingriot.incide.net
moving data around inside the Earth Simulator

Crossbar weight: 220 t, 3000 km of thick cable,
discarding the wrong road map

with a paradigm shift
the same performance is feasible on a single 19” rack
Bringing together data and processor

Moving data to the processor: by Software

moving the grand piano
Key issues in very High Performance Computing (vHPC)

reducing memory cycles is the key issue

this needs a paradigm shift

away from the dominance of instruction streams
Here is the common model

it's not von Neumann
we need
dual paradigm education

the vN monopoly in our curricula is severely harmful

software code
configware code

instruction-stream-based
data-stream-based

Von Neumann:
the tail is wagging the dog

CPU
reconfigurable accelerator
hardwired accelerator

very high performance
& electricity bill issues

legacy issues
The wrong basic mind set

we need a
da dual paradigm approach

our IT expert labor force
lacks the rite basic mind set

this is a severe
educational challenge
For high school and undergraduate education

we need a

an archtype simple common model

instead of a wide variety of

sophisticated architectures

this is a severe

educational challenge
• Reconfigurable Computing Paradox
• The Supercomputing Paradox
• We are using the wrong model
• Coarse-grained Reconfigurable Devices
• Super Pentium for Desktop Supercomputer
the effective integration density of plane FPGAs behind Moore’s law by more than 4 orders of magnitude.

the effective integration density of rDPAs* may come close to Moore’s law.

*) reconfigurable DataPath Arrays (coarse-grained reconfigurability)
Coarse grain is about computing, not logic

SNN filter on KressArray (mainly a pipe network)

array size:
10 x 16
= 160 rDPUs

no CPU

reconfigurable Data Path Unit, e.g. 32 bits wide
SW 2coarse-grained CW migration example
Compare it to software solution on CPU

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

<table>
<thead>
<tr>
<th>C = 1 simple conservative CPU example</th>
<th>memory cycles</th>
<th>nano seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>if C then read A</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>read operand*</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>operate &amp; reg. transfers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>then 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>
hypothetical branching example to illustrate software-to-configware migration

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

\begin{tabular}{|l|l|l|}
\hline
\textbf{C = 1} & \textbf{memory cycles} & \textbf{nano seconds} \\
\textbf{simple conservative CPU example} & & \\
\hline
if C then read A & read instruction & 1 \hspace{1cm} 100 \\
& instruction decoding & \\
& read operand* & 1 \hspace{1cm} 100 \\
& operate & reg. transfers & \\
\hline
if not C then read B & read instruction & 1 \hspace{1cm} 100 \\
& instruction decoding & \\
\hline
add & store & read instruction & 1 \hspace{1cm} 100 \\
& instruction decoding & \\
& operate & reg. transfers & \\
& store result & 1 \hspace{1cm} 100 \\
\hline
\textbf{total} & & 5 \hspace{1cm} 500 \\
\hline
\end{tabular}

*) if no intermediate storage in register file

clock 200 MHz (5 nanosec)
Why the speed-up? What's the difference?

moving the locality of operation into the route of the data stream by P&R

instead of moving data by instruction streams
Bringing together data and processor

Place the location of execution into the data pipe

Move the stool

by Configware

http://hartenstein.de

© 2006, reiner@hartenstein.de
Data-stream-based execution should be transport-triggered instead of instruction-triggered transport should be done within compiled pipelines, not by move engines* which are instruction-stream-based!

*)
For high school and undergraduate education

we should send CTOs and professors back to school

this is a severe educational challenge
The wrong model

upon this schematics ...

... question by a Japanese Corporate vVIP

SNN filter on KressArray (mainly a pipe network)

array size: 10 x 16 = 160 rDPUs

no CPU

reconfigurable Data Path Unit, e. g. 32 bits wide

http://hartenstein.de
The wrong mind set ....

(Question by a Japanese Corporate vVIP: [RAW’99])

„but you can‘t implement decisions!“

not knowing this solution:
symptom of the hardware / software chasm
and the configware / software chasm

We need Reconfigurable Computing Education
• Reconfigurable Computing Paradox
• The Supercomputing Paradox
• We are using the wrong model
• Coarse-grained Reconfigurable Devices
• Super Pentium for Desktop Supercomputer
some Goals

Universal HPC co-architecture for:
  embedded vHPC (nomadic, automotive, ...)
  desktop vHPC (scientific computing ...)

Application co-development environment for
  Hardware non-experts, ....
  Acceptability by software-type users, ...

Meet product lifetime >> embedded syst. life:
  FPGA emulation logistics from development
downto maintenance and repair stations
examples: automotive, aerospace, industrial, ..
Architecture: A potential Pentium successor

Discard most caches

have 64* cores, 0.5 - 1 GHz

with clever interconnect for:

▪ concurrent processes and

▪ and for multithreading,

▪ Kung-Kress pipe network

The Desk-top Supercomputer!

*) CPU mode / DPU mode capability

39

© 2006, reiner@hartenstein.de
“Super Pentium” configuration example

twin paradigm machine
e. g.: \(~ 8 \times 8 \text{ rDPA: all feasible under } 500 \text{ MHz}\)

World TV &
game console &
multi media center

- Variable resolutions and refresh rates
- Variable scan mode characteristics
- Noise Reduction and Artifact Removal
- High performance requirements
- Variable file encoding formats
- Variable content security formats
- Variable Displays
- Luminance processing
- Detail enhancement
- Color processing
- Sharpness Enhancement
- Shadow Enhancement
- Differentiation
- Programmable de-interlacing heuristics
- Frame rate detection and conversion
- Motion detection & estimation & compensation
- Different standards (MPEG2/4, H.264)
- A single device handles all modes

http://pactcorp.com
feasible under 500 MHz

means low electricity cost and allows very high integration density
apropos compiled pipeline ...
Dual Paradigm Application Development Support

Placement & routing in the compiler optimizes interconnect bandwidth by preferring nearest neighbor connect.
Software / Configware Co-Compilation

C language source

Brainstormer

Partitioner

SW compiler

CW compiler

Placement & Routing

(Move the Locality of Operation)

Resource Parameters

supporting different platforms

Juergen Becker's CoDe-X, 1996

http://hartenstein.de

© 2006, reiner@hartenstein.de
Software / Configware very high level Synthesis

Math formula ....

term-rewriting-based vhl synthesis system

software code

instruction-stream-based

configware code
data-stream-based

CPU

reconfigurable accelerator

hardwired accelerator

[Arvind, or, Mauricio Ayala]

© 2006, reiner@hartenstein.de

http://hartenstein.de
• Reconfigurable Computing Paradox
• The Supercomputing Paradox
• We are using the wrong model
• Coarse-grained Reconfigurable Devices
• Super Pentium for Desktop Supercomputer
• Conclusions
Objectives
for every area which needs:

cheap, compact vHPC
rapid prototyping, field-patching, emulation
avoiding specific silicon
flexibility (for accelerators)
Reconfigurable Computing opens many spectacular new horizons:

Cheap vHPC without needing specific silicon, no mask ....
Cheap embedded vHPC
Cheap desktop supercomputer (a new market)
Replacing expensive hardwired accelerators
Fast and cheap prototyping
Flexibility for systems with unstable multiple standards by dynamic reconfigurability
Supporting fault tolerance, self-repair and self-organization
Emulation logistics for very long term sparepart provision and part type count reduction (automotive, aerospace ...)
Massive reduction of the electricity bill: locally and national
Needed: Universal vHPC co-architecture demonstrator

Conclusion (2)

For widely spreading its use successfully:

- The compilation tool problem to be solved
- Language selection problem to be solved
- Education backlog problems to be solved

Use this to develop a very good high school and undergraduate lab course

select killer applications for demo

A motivator: preparing for the top 500 contest
thank you
backup
Compilation: Software vs. Configware

**Software Engineering**
- source program
  - software compiler
    - software code

**Configware Engineering**
- source "program"
  - mapper
    - configware compiler
      - configware code
        - data
          - scheduler
            - flowware code

Languages:
- C
- FORTRAN
- MATHLAB
Nick Tredennick’s Paradigm Shifts explain the differences

**Software Engineering**

```
<table>
<thead>
<tr>
<th>CPU</th>
<th>resources: fixed</th>
<th>1 programming source needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>software</td>
<td>algorithm: variable</td>
<td></td>
</tr>
</tbody>
</table>
```

**Configware Engineering**

```
<table>
<thead>
<tr>
<th>configware</th>
<th>resources: variable</th>
<th>2 programming sources needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowware</td>
<td>algorithm: variable</td>
<td></td>
</tr>
</tbody>
</table>
```
Co-Compiler for Hardwired Kress/Kung Machine
[e. g. Brodersen]

source

automatic SW / CW partitioner

Software / Flowware Co-Compiler

software code

flowware code

software compiler

Flowware compiler

data scheduler
The first archetype machine model

Software Industry

Software Industry's Secret of Success

Procedural personalization

Instruction-stream-based mind set

"von Neumann"

Compile or assemble

Simple basic.

Machine Paradigm

Personalization: RAM-based
The 2nd archetype machine model

Configware Industry

Configware Industry's Secret of Success

- compile
- structural personalization
- data-stream-based mind set
- "Kress-Kung"
- reconfigurable accelerator

Machine Paradigm

simple basic.

RAM-based
Co-Compiler Enabling Technology

is available from academia

only a small team needed for commercial re-implementation

on the road map to the Personal Supercomputer
define: ... which data item at which time at which port

(pipe network) DPA

implemented by distributed memory

50 & more on-chip ASM are feasible

H. T. Kung paradigm (systolic array)

Data streams

input data stream

output data streams

ASM: Auto-Sequencing Memory

50 & more on-chip ASM are feasible
The Generalization of the Systolic Array

Kress-Kung paradigm
super systolic array

only for applications with regular data dependencies

remedy? discard algebraic synthesis methods

[R. Kress]: use optimization algorithms e. g.: simulated annealing

reconfigurability makes sense

Achievement: also non-linear and non-uniform pipes, and even more wild pipe structures possible
(Kress-Kung machine paradigm) drastically reducing memory cycles

ASM: Auto-Sequencing Memory

Data Counter instead of Program Counter

Generalization of the DMA

GAG & enabling technology: multiple publications 1989 ...
Survey paper: [M. Herz et al.]: IEEE ICECS 2003, Dubrovnik

Storage Scheme optimization methodology, etc.*

*) IMEC, Leuven & TU-KL
Technology: fine-grained RC: 1st DeHon's 1st Law

[1996: Ph. D, MIT]

transistors / microchip

10^12

10^9

10^6

10^3

10^0

1980 1990 2000 2010

density: overhead:

FPGA physical

wiring overhead

>> 10000

FPGA logical

reconfigurability overhead>

FPGA routed

routing congestion

immense area inefficiency

(Gordon Moore curve)
coarse-grained RC: Hartenstein's amendment of DeHon's 1st Law

[1996: ISIS, Austin, TX]

area efficiency very close to Moore's law
More compute power by Configware than Software (a very cautious estimation**)

75% of all (micro)processors are embedded

25% embedded µProc. accelerated by FPGA(s)

-> Every 2nd µProc accelerated by FPGA(s)  -> 1 : 1

avarage acceleration factor >2  -> rMIPS* : MIPS > 2

Conclusion: most compute power from Configware
(difference probably an order of magnitude)

*) rMIPS: MIPS replaced by FPGA compute power

**) Dataquest interaction pending
Conclusion (3)

Self-Repair and Self-Organization methodology

Embedded r-emulation logistics methodology

Universal vHPC co-architecture demonstrator

For widely spreading its use successfully:

select a killer application for demo
Dual Paradigm Application Development Support

MATLAB

adapter

high level language

software/configware co-compiler

software code

instruction-stream-based

CPU

reconfigurable accelerator

data-stream-based

other example

hardwared accelerator

http://hartenstein.de

© 2006, reiner@hartenstein.de