IPDPS 2004
Santa Fe, NM, April 26–30, 2004

Reiner Hartenstein
TU Kaiserslautern

Software or Configware?
About the Digital Divide of Parallel Computing

The White House, Sept 2000:
Bill Clinton condemns the Digital Divide in America:
access to the internet
World Economic Forum 2002:
The Global Digital Divide,
disparity between the "haves" and "have nots"
The Digital Divide of Parallel Computing:
Access to Configware (CW) Solutions

The "havenots"
Configware methodology to
move data around more efficiently:
"havenots" are found in the HPC community
Configware engineering as a qualification
for programming embedded systems:
The "havenots" are our typical CS graduates
Reconfigurable HPC is torpedoed
by deficits in education:
curricular revisions are overdue

Software to Configware Migration
Software to Configware Migration
is the most important source of speed-up
Hardware is just frozen Configware
this talk will illustrate the performance benfit
which may be obtained from Reconfigurable Computing
stressing coarse grain Reconfigurable Computing (RC),
point of view, this talk hardly mentions FPGAs
But coarse grain may be always mapped onto FPGAs

"HPC"

Moving data around inside the Earth Simulator
Crossbar weight: 220 t, 3000 km of cable,
ES 20: TFLOPS
5120 Processors, 5000 pins each

Reiner Hartenstein (keynote address)
The 18th International Parallel and Distributed Processing
Symposium (IPDPS), April 26–30, 2004, Santa Fe, NM, USA
data are moved around by software
i.e. by memory-cycle-hungry instruction streams which fully hit the memory wall
(slower than CPU clock by 2 orders of magnitude)

... understand only this parallelism solution:
the instruction-stream-based approach

... path of least resistance:
avoiding a paradigm shift
Many researchers seem never to stop working on sophisticated solutions for marginal improvements ...
... continuously ignoring methodologies promising speed-ups by orders of magnitude ....

<< Embedded Computing >>

• HPC
• Embedded Computing
• The wrong Roadmap
• Configware Engineering
• Dual Machine Paradigms
• Speed-up Examples
• Final Remarks

History of Machine Models

mainframe age
compile
main frame
procedural mind set
instruction-stream-based
(coordinated by Makimoto wave)

computer age (PC age)
compile
μProc. accel.

scientific computing example:
molecular dynamics, astrophysics, plasma physics, hydrodynamics:
MD GRAPE-2 PCI board [1997] 4 chips to nobody simulated
converts a PC to 64 GFlops
user: RIKEN institute, ARI, Heidelberg, etc.

by hardware guys
by structural design
structural mind set:
data-stream-based
(coordinated by Makimoto wave)
the hardware / Software Chasm:

It's the gap between procedural (instruction-stream-based) and structural (datastream-based) mind set.

Typical programmers don't understand function evaluation without machine mechanisms (counters, state registers).

Growth Rate of Embedded Software

Already today, more than 98% of all microprocessors are used within embedded systems.

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

typical CS graduates: the „havenots“

Today, typical CS graduates are unqualified for this labor market.

... cannot cope with Hardware / Configware / Software partitioning issues.

... cannot implement Configware.

the current CS mind set is based on the Submarine Model

This model does not support Hardware / Configware / Software partitioning.

Hardware invisible: under the surface.

Software / Configware / Hardware Partitioning skills urgently needed

Software to Configware Migration is the most important source of speed-up.

Hardware is just frozen Configware.

or: to cope with any combination of co-design.

By the way...

... the oldest and largest conference in the field:

International Conference on Field-Programmable Logic and Applications (FPL)

http://fpl.org

Aug. 20 - Sept 1, 2004, Antwerp, Belgium

... going into every type of application.

288 submissions! They all work on high performance.
Dominance of the Submarine Model ...

(procedural) structurally disabled

Hardware

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

... disabled to cope with solutions other than instruction-stream-based

CS Education

You cannot teach Hardware to a Programmer

*efficiently

have structural natural

have not procedural

But to a Hardware Guy you always can teach Programming

CS Education

procedural

You cannot teach Hardware to a Programmer

*efficiently

But to a Hardware Guy you always can teach Programming

Structural

Natural

Hardware

CS Education

procedural

You cannot teach Hardware to a Programmer

*efficiently

But to a Hardware Guy you always can teach Programming

Structural

Natural

But to a Hardware Guy you always can teach Programming

>> the wrong Roadmap <<

• HPC
• Embedded Computing
• the wrong Roadmap
• Configureware Engineering
• Dual Machine Paradigms
• Speed-up Examples
• Final Remarks

Completely wrong roadmap

beef up old architectural principles by new technology?

growth factor

area efficiency

performance

"Pollack's Law" (simplified)

... the CPU is a methusela, the steam engine of the silicon age

Completely wrong mind set

The key problem, the memory wall, cannot be solved by new CPU technology

The vN paradigm is not a communication paradigm

Its monopoly creates a completely wrong mind set

We need a 2nd machine paradigm (a 2nd mind set ...)

We need an architectural communication paradigm

But we need both paradigms: a dichotomy

3rd machine model became mainstream

mainframe age computer age (PC age)

morphware age programmable

μProc. rDPA

3rd machine model became mainstream

mainframe age computer age (PC age)

morphware age programmable

μProc. rDPA

Reiner Hartenstein (keynote address)
The 18th International Parallel and Distributed Processing Symposium (IPDPS), April 26-30, 2004, Santa Fe, NM, USA
The 18th International Parallel and Distributed Processing Symposium (IPDPS), April 26–30, 2004, Santa Fe, NM, USA
Compilation: Software vs. Configware

Flowware defines: ... which data item at which time at which port

Flowware programs data streams

Flowware: not new

*no confusion, please: no "dataflow machine"

1980: data streams (Kung, Leiserson: systolic arrays)
1989: data-stream-based Xputer architecture
1990: rDPU (Rabaey)
1994: Flowware Language MoPL (Becker et al.)
1995: super systolic array (rDPA) + DPSS tool (Kress)
1996+: Stream-C language, SCCC (Los Alamos), SCORE, ASPRC, Bee (UC Berkeley), ...
1996+: streaming languages (Stanford et al.)
1996+: configware / software partitioning compiler (Becker)

Why a new machine paradigm ???

The anti machine as the 2nd paradigm is the key to curricular innovation

... a Troyan horse to introduce the structural domain to the procedural-only mind set of programmers

Programming by flowware instead of software is very easy to learn (... same language primitives)

Flowware education: no fully fledged hardware expert needed to program embedded systems
von Neumann vs. anti machine

von Neumann bottleneck
instruction stream machine (von Neumann etc.)

asM: auto-sequencing Memory
asMA: auto-sequencing Memory Array

data stream machine (anti machine)

DPU vs. CPU

von Neumann bottleneck

DPU

asMA: auto-sequencing Memory Array

Behavior of the Counter

DPU

Programmed by firmware

asMA

Data stream machine (anti machine)

asMA

Counters: the same micro architecture?

Yes, is possible, but for data counters ...

... a much better AGU methodology is available*

*) for history of AGUs see Herz et al.: Proc. ICECS 2002, Dubrovnik, Croatia

commercial rDPA example:

PACT XPP - XPU128

• Full 32 or 24 Bit Design working silicon
• 2 Configuration Hierarchies
• Evaluation Board available, and
• XDS Development Tool with Simulator

mapping algorithms efficiently onto rDPA

by DPSS: based on simulated annealing

SNN filter on KressArray

array size: 10 x 16 = 160 rDPUs

reiner@hartenstein.de

The 18th International Parallel and Distributed Processing Symposium (IPDPS), April 26–30, 2004, Santa Fe, NM, USA
The 18th International Parallel and Distributed Processing Symposium (IPDPS), April 26–30, 2004, Santa Fe, NM, USA
rDPA (coarse grain) vs. FPGA (fine grain)

<table>
<thead>
<tr>
<th>Status</th>
<th>rDPA</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>hardwired 3</td>
<td>hardwired 4</td>
</tr>
</tbody>
</table>

Why the speed-up ...

... although FPGA is clock slower by x 3 or even more (most know-how from high level synthesis discipline)

support operations: no clock nor memory cycle
decisions without memory cycles nor clock cycles
moving operator to the data stream (before run time)
most „data fetch“ without memory cycle

First Indications of Change

PARS & Speed-up, Basel, Switzerland, March 2003: keynote address
10th RAW at IPDPS, Nice, France, April 2003: after a decade of non-overlap: first IPDPS people coming
PDP04, La Coruña, Spain, Febr. 2004: keynote address
IPDPS, Santa Fe, NM, USA, April 2004: keynote address

Conclusions

We need an academic grass roots movement, for ...
RC has become mainstream in all kinds of applications
CS education deficits: a curricular revision is overdue ...
... by a merger with the embedded systems mind set
We need an academic grass roots movement, for ....
...free material & tools for undergraduate lab courses to program and emulate small SW/CW/HW examples
all know-how needed readily available:
get involved!
END