The new machine paradigm

Configware is going mainstream
not only in embedded systems
Hardware / Configware / Software do-design is the new mind set for digital systems engineering
a dichotomy of machine paradigms is needed for qualification
a co-education for a symbiosis of instruction-stream-based and data-stream-based concepts

Software to Configware Migration

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

Terminology: „soft hardware“?

Terminology: „soft hardware“?

Programming sources:

Configware: for configuring morphware
Flowware: for scheduling data streams
Software: for scheduling instruction streams

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

Extremely unbalanced.

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 .... Continue to bang their heads against the memory wall instead of blinding to ignore the impact of morphware.

... Understand only this parallelism solution: the instruction-stream-based approach. The data-stream-based approach has no von Neumann bottleneck.

... The History of Paradigm Shifts

Makimoto’s 3rd Wave

- Fine Grain Subsystems (FPGAs):
  - 1st half of 3rd wave
  - Universal (but less efficient)
- Coarse Grain Subsystems:
  - 2nd half of 3rd wave
  - Domain-specific
  - Much more flexible than 2nd half of 2nd wave

>> Embedded Computing <<

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

Makimoto’s Wave

“Mainstream Silicon Application is switching every 10 Years”

“The Programmable System-on-a-Chip is the next wave”

What replaced in 1969?
**How’s next Wave?**

- Standard: 1947
- Procedural/programming: 1967
- Structural programming: 1977
- Library: 1987
- FPGA: 1997
- µProc, accel: 2007

**History of Silicon Application**

- Hardware people
- CS people
- New breed needed

Common terminology needed

**Tredennick’s Paradigm Shifts**

- Algorithm: fixed
  - Resources: fixed
  - Hardwired
- Algorithm: variable
  - Resources: fixed
  - Procedural programming

**History of Machine Models**

- Mainframe age
- Computer age (PC age)
- Procedural mind set: instruction-stream-based
- Structural mind set: data-stream-based

**The hardware / Software Chasm:**

- It's the gap between procedural (instruction-stream-based) and structural (data-stream-based) mind set
- Typical programmers don't understand function evaluation without machine mechanisms (counters, state registers)

**History of Machine Models**

- Mainframe age
- Computer age (PC age)
- Procedural mind set: instruction-stream-based
- Structural mind set: data-stream-based

**History of Machine Models**

- Mainframe age
- Computer age (PC age)
- Procedural mind set: instruction-stream-based
- Structural mind set: data-stream-based

**History of Machine Models**

- Mainframe age
- Computer age (PC age)
- Procedural mind set: instruction-stream-based
- Structural mind set: data-stream-based
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"

To-day, typical CS graduates are unqualified for this labor market

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

... cannot implement Configware

Hardware / Configware / Software Partitioning skills urgently needed

Algorithm

partitioning

to cope with each of it: SW, CW, HW

model: Hardware is just frozen Configware

or: to cope with any combination of co-design

CS Education

You cannot "teach Hardware to a Programmer" *efficiently.

have procedural

have structural natural

But to a Hardware Guy you always can teach Programming

>> the wrong Roadmap <<

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

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

μPro“accel. going into every type of application

they all work on high performance
future HPC: completely wrong mind set
beef up old architecture principles by new technology?
The key problem, the memory wall, cannot be solved by new CPU technology — not execution!
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

3rd machine model became mainstream

The 2nd machine paradigm.

De facto Duality of RAM-based platforms

We now have 2 types of programmable platforms.

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

Software Industry

Growing Configware Industry

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

... the brain hurts

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

From Software to Configware Industry

Growing Configware Industry

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

Future HPC: completely wrong mind set
beef up old architecture principles by new technology?
The key problem, the memory wall, cannot be solved by new CPU technology — not execution!
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

The 2nd machine paradigm.

De facto Duality of RAM-based platforms

We now have 2 types of programmable platforms.

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

Software Industry

Growing Configware Industry

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

... the brain hurts

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

From Software to Configware Industry

Growing Configware Industry

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

... the brain hurts

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.

From Software to Configware Industry

Growing Configware Industry

The HPC scene believed to be smart, when smiling about us CW guys.
Others experienced, that the brain hurts, when trying the paradigm shift.
morphware: fastest growing sector of the IC market.
CW has become mainstream...
... going into every type of application.
benefit from RAM-based & 2nd paradigm

1. RAM-based platform needed for:
   - flexibility, programmability
   - avoiding the need of specific silicon
   - mask cost: currently 2 mio $ - rapidly growing

2. Simple 2nd machine paradigm needed as a common model:
   - to avoid the need of circuit expertise
   - needed to educate zillions of programmers

Software Engineering

CPU software
resources: fixed
algorithm: variable
1 program needed

Configware Engineering

configware
resources: variable
algorithm: variable
2 programs needed

Compilation: Software vs. Configware

Software Engineering

source program
software compiler
software code

Configware Engineering

source program
mapper
configware compiler
data scheduler
configware code

Compilation: Software vs. Flowware

Flowware Engineering

source program
flowware compiler
data scheduler
flowware code

Flowware programs data streams

Flowware: not new

*) no confusion, please:

no "dataflow machine"

Flowware: data stream* ...

around 1975 - 1980

1957
mainframe age

1967
computer age (PC age)

1977
morphware age

1997
DPA

input data streams

output data streams

DPA
data streams*: not new

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+: Streams-C language, SCCC (Los Alamos), SCORE, ASPRC, Bee (UC Berkeley), DSP-C, Brook, ...
1996: configure / software partitioning compiler (Becker)

Why a new machine paradigm ??

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

Behavior of the Counter

Counters: the same micro architecture?

* 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

© PACT AG, http://pactcorp.com

XPP64A: Platform Development Board
- SDR Board In Debug Phase
- XPP64A Chips from STMicro Fab
- Assembly & Test / Available March 2003

mapping algorithms efficiently onto rDPA:
by DPSS: based on simulated annealing

Compilers play a key role in mapping a problem to a platform.

Bill Dally, WCAE’04

Many problems are better solved at
compile time

Not moving data to
operator inefficiently
at run time, but place
rDPU into data stream
at compile time

Microcoding wave

Julich, 1957

Ultra SPARC

Morphware age

Compilers play a key role in mapping algorithms efficiently onto rDPA: by DPSS, based on simulated annealing. Compilers play a key role in mapping a problem to a platform. Many problems are better solved at compile time. Not moving data to operator inefficiently at run time, but place rDPU into data stream at compile time.
Better solutions by Configware instead of software
methodologies not new: high level synthesis (1980+)
loop transformations (1970+)
many other areas
Memory cycles minimized
e.g.: no instruction fetch at run time & other effects
No cache misses!
Memory access for data: caches do not help anyhow
Loop xforms: no intra-stream data memory cycles
Complex address computation: no memory cycles

hypothetical branching example to illustrate time-to-space migration
S = R + (if C then A else B endif);

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

speed-up examples
key issue: algorithmic cleverness

rDPA (coarse grain) vs. FPGA (fine grain)
Status: ~1998

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

Final Remarks

- HPC
- Embedded Computing
- The wrong Roadmap
- Configware Engineering
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks
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
- PDPO4, La Coruna, Spain, Feb. 2004: keynote address
- IPDPS, Santa Fe, NM, USA, April 2004: keynote address
- HPC Asia 2004 - 7th Int’l Conference on High Performance Computing, July 20-22, 2004 Omiya Sonic City, Tokyo Area, Japan: Workshop on Reconfigurable Systems f. HPC (RHPC) + keynote address

HPC experts coming...

- Simulation of Star Clusters: x10 speed-up by supercomputer-to-morphware migration (also molecular biology et al.)
- Embedded and reconfigurable architectures

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
- ...free material & tools for undergraduate lab courses to program and emulate small SW/CW_HW examples
- all know-how needed readily available:

Edholm’s Law of Bandwidth

- Copyright © 1996 by Masayuki Suzuki

END