Reiner Hartenstein, Informatik Department,
University of Kaiserslautern, Germany
http://hartenstein.de
reiner@hartenstein.de

WCAE '04
at ISCA 2004

June 19, 2004, Munich, Germany

Reiner Hartenstein

TU Kaiserslautern

The Changing Role of Computer Architecture Education within CS-related Curricula

TU Kaiserslautern

Preface

Challenges to all CS-related curricula

we need a duality of machine paradigms with a new basic common platform model to join the "von Neumann" machine paradigm

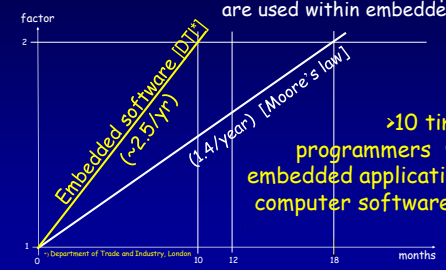
for a new mind set to dwarf "classical" speed-up methodologies in high performance computing (HPC)

to qualify programmers for embedded systems

TU Kaiserslautern

Growth Rate of Embedded Software

already to-day, more than 98% of all microprocessors are used within embedded systems



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

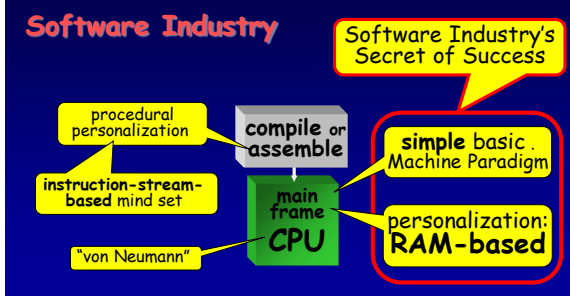
TU Kaiserslautern

>> common platform models <<

- Common platform models
- HPC going Configware
- Configware vs. Software
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks

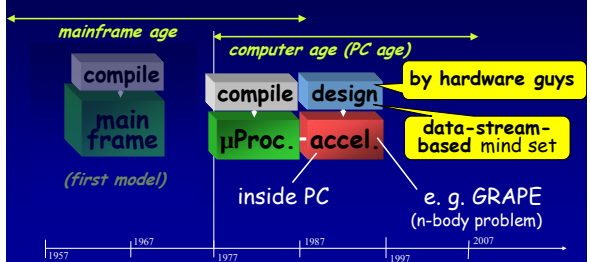
TU Kaiserslautern

The first basic machine model



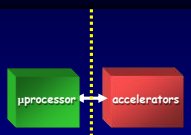
TU Kaiserslautern

The second basic machine model



Reiner Hartenstein (keynote address): The Changing Role of Computer Architecture Education within CS-related Curricula; Workshop on Computer Architecture Education (WCAE 2004) at ISCA 2004, June 19, 2004, Munich, Germany

the Hardware / Software Chasm:

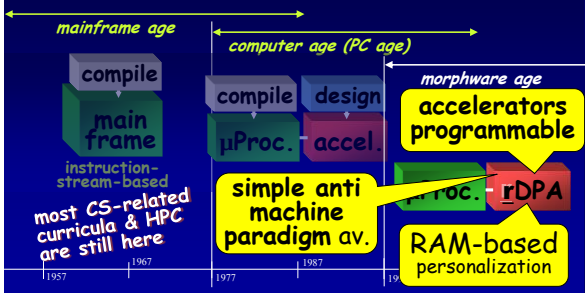


The gap between procedural (instruction-stream-based) and structural (datastream-based) mind set

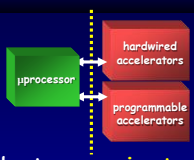
Most CA courses teach CPU instruction set architecture. But only few people architecture CPUs. [Bill Dally, WCAE'04]

CA courses should bridge the Hardware / Software chasm.

3rd machine model became mainstream



the Hardware / Configware / Software Chasm



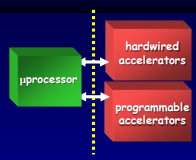
hardware

configware running on morphware

the gap between instruction-stream-based and datastream-based mind set

typical programmers cannot program morphware: i. e. they cannot implement configware

typical programmers are unqualified ....

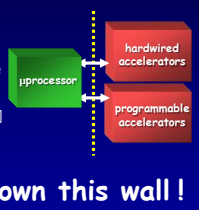


typical programmers cannot cope with Hardware / Configware / Software partitioning issues

typical programmers are unqualified to implement embedded applications

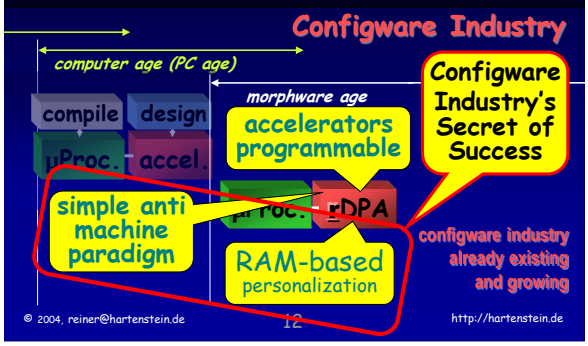
the Computing Education Wall

Simple models of area, power, and delay can be taught in a week or two. [Bill Dally, WCAE'04]



Tear down this wall !

Configware Industry going Mainstream



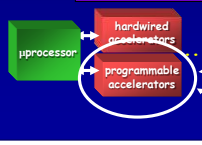
By the way ...
... the oldest and largest conference in the field:

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

http://fpl.org

early registration: deadline July 7

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



... going into every type of application
289 submissions !
they all work on high performance

TU Kaiserslautern

>> HPC going configware <<

- Common platform models
- HPC going Configware
- Configware vs. Software
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks

TU Kaiserslautern

Software to Configware Migration

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

The Reconfigurable Computing mind set
dwarfs the efficiency of current PHC methodologies

Morphware: fastest growing sector of the IC market

CW has become mainstream ...

... going into every type of application

TU Kaiserslautern

HPC experts coming ...

example: N-body problem going configware

paper already at FPL 1999 http://fpl.org

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

Configware by
Reinhard Maenner, University of Mannheim
HPC pioneer since 1976 (Physics Dept Heidelberg)

Astrophysics by
Rainer Spurzem, University of Heidelberg
ARI, Astrononisches Rechen-Institut, founded 1700 in Berlin, moved 1945 to Heidelberg by August Kopff

TU Kaiserslautern

moving data around inside the Earth Simulator

Crossbar weight: 220 t, 3000 km of cable,

No longer is the task to obtain speed-up at any expense
[Bill Dally, WCAE'04]

TU Kaiserslautern

data are moved around by software

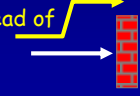
i.e. by memory-cycle-hungry instruction streams which fully hit the memory wall

(slower than CPU clock by >2 orders of magnitude)

... continue to bang their heads against the memory wall

path of least resistance: avoiding a paradigm shift

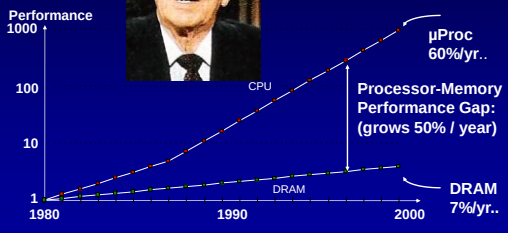
instead of



TU Kaiserslautern

The memory wall

Tear down this wall !



TU Kaiserslautern

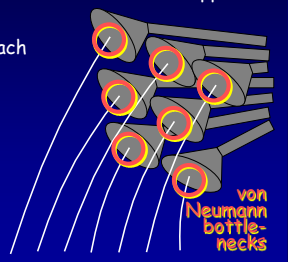
… understand only this parallelism solution:

the instruction-stream-based approach

the data-stream-based approach

has no von Neumann bottle-neck

… cannot cope with this one



TU Kaiserslautern

Completely wrong mind set

beef up old architectures by new technology?

Memory wall not removable by new CPU technology

The vN paradigm is not a communication paradigm

Its monopoly creates a completely wrong mind set

We need the mind set of Reconfigurable Computing

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

We need an architectural communication paradigm

But we need both paradigms: a dichotomy

TU Kaiserslautern

>> Configware vs. Software <<

- Common platform models
- HPC going Configware
- Configware vs. Software
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks

TU Kaiserslautern

de facto Duality of RAM-based platforms

We now have 2 types of programmable platforms

2nd paradigm

	traditional	new
RAM-based platform	CPU	morphware (FPGA, rDPA ..)
configured onto it:	(hardwired)	configware
„running" on it:	software	flowware
machine paradigm	von Neumann etc.: instruction-stream-based	anti machine: data-stream-based

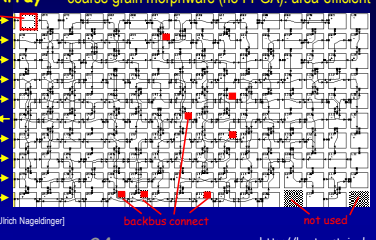
TU Kaiserslautern

mapping algorithms efficently onto rDPA:

by DPSS: based on simulated annealing

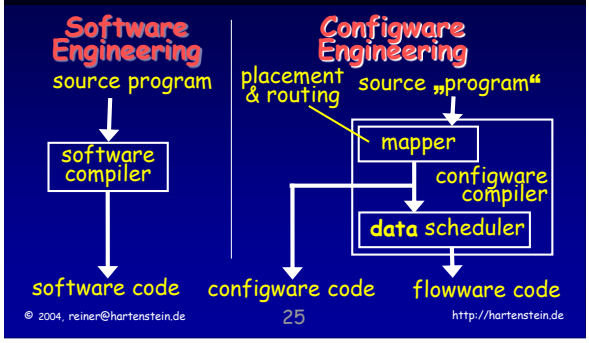
Compilers play a key role in mapping a problem to a platform. [Bill Dally, WCAE'04]

SNN filter on KressArray coarse grain morphware (no FPGA): area-efficient



Reiner Hartenstein, Informatik Department,
University of Kaiserslautern, Germany
http://hartenstein.de
reiner@hartenstein.de

Compilation: Software vs. Configware

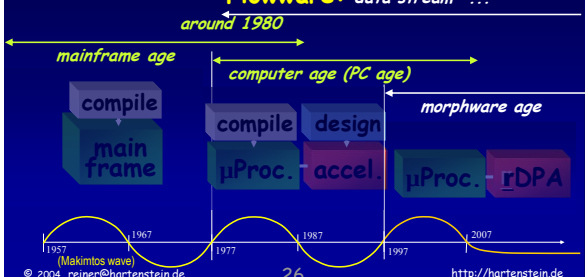


 25

Flowware: not new

*) no confusion, please: no „dataflow machine“ !!!

Flowware: data stream* ...



 26

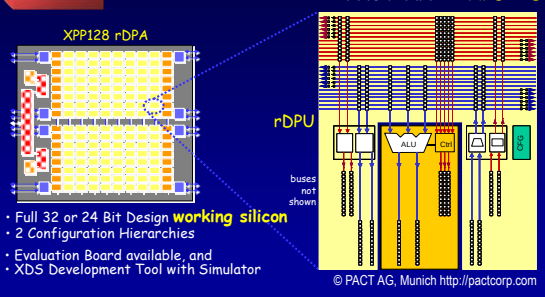
Data Streams: Flowware is not new

- 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: configware / software partitioning compiler (Becker)

 27

(r)DPA commercial rDPA example:

PACT XPP - XPU128

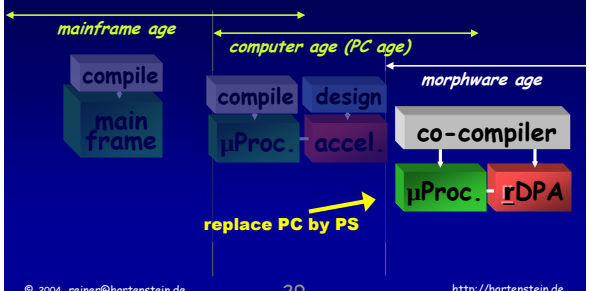


- Full 32 or 24 Bit Design **working silicon**
- 2 Configuration Hierarchies
- Evaluation Board available, and
- XDS Development Tool with Simulator

© PACT AG, Munich http://pactcorp.com

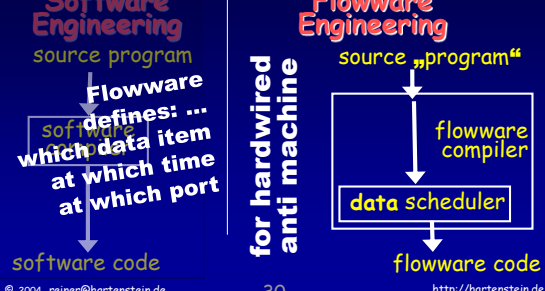
 28

symbiosis of machine models



 29

Compilation: Software vs. Flowware



 30

Reiner Hartenstein (keynote address): The Changing Role of Computer Architecture Education within CS-related Curricula; Workshop on Computer Architecture Education (WCAE 2004) at ISCA 2004, June 19, 2004, Munich, Germany

>> Dual Machine Paradigms <<

- Common platform models
- HPC going Configware
- Configware vs. Software
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks

CS-related Education

You cannot *teach Hardware to a Programmer

*) efficiently

have not procedural

have structural natural

But to a Hardware Guy you always can teach Programming

Why a new machine paradigm ???

µprocessor → rDPA

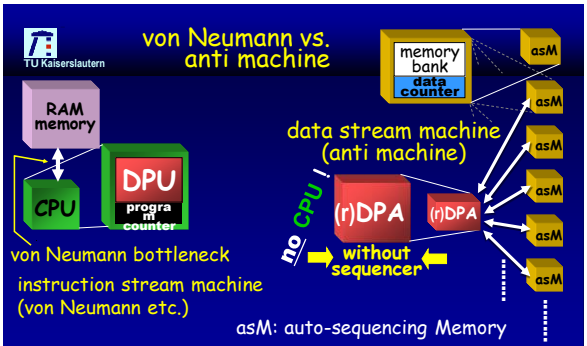
The anti machine as the 2nd paradigm is the key to curricular innovation

... a Troyan horse to introduce data-stream-based issues to the classical mind set of programmers

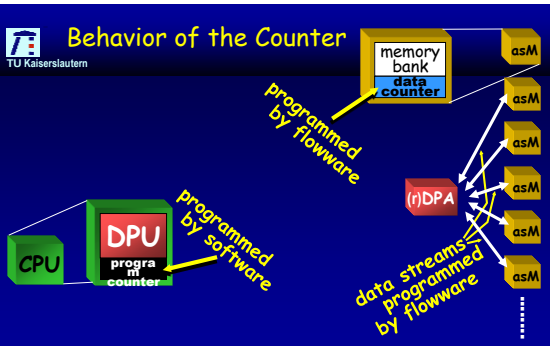
Programming by flowware instead of software is very easy to learn (... same language primitives)

Flowware education: no fully fledged hardware expert needed to program configware

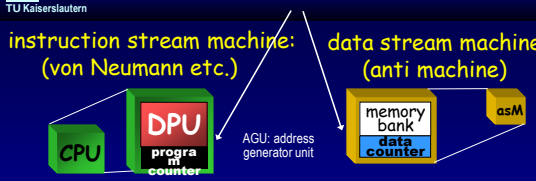
von Neumann vs. anti machine



Behavior of the Counter



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

TU Kaiserslautern

>> Speed-up Examples <<

- Common platform models
- HPC going Configware
- Configware vs. Software
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks

TU Kaiserslautern

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

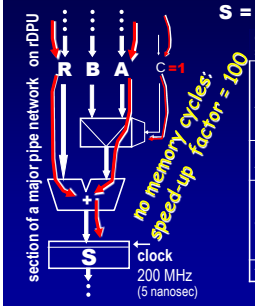
Loop xforms: no intra-data-stream memory cycles

Complex address computation: no memory cycles

TU Kaiserslautern

hypothetical branching example to illustrate time-to-space migration

S = R + (if C then A else B endif);



C = 1 simple conservative CPU example		memory cycles	nano seconds
if C then read A	read instruction	1	100
	instruction decoding		
	read operand*	1	100
	operate & reg. transfers		
if not C then read B	read instruction	1	100
	instruction decoding		
add & store	read instruction	1	100
	instruction decoding		
	operate & reg. transfers		
	store result	1	100
total		5	500

*) if no intermed. storage in register file

TU Kaiserslautern

speed-up examples
key issue: algorithmic cleverness

platform	application example	speed-up factor	method
PACT Xtreme 4-by-4 array [2003]	16 tap FIR filter	x16 MOPS/mW	straight forward
MoM anti machine with DPLA* [1983]	grid-based DRC** 1-metal 1-poly nMOS*** 256 reference patterns	> x1000 (computation time)	multiple aspects
CPU 2 FPGA [FPGA 2004]	migrate several simple application exampes	x7 – x46 (compute time)	hi level synthesis
DSP 2 FPGA [Xilinx 2004[2]]	from fastest DSP: 10 gMACs to 1 teraMAC	X 100 (compute time)	not spec.

*) DPLA: MPC fabr. via E.I.S. multi univ. project **) Design Rule Check ***) for 10-metal 3-poly cMOS expected: >> x10,000 2) Wim Roelandts

TU Kaiserslautern

>> Final Remarks <<

- Common platform models
- HPC going Configware
- Configware vs. Software
- Dual Machine Paradigms
- Speed-up Examples
- Final Remarks

TU Kaiserslautern

benefit from RAM-based 2nd paradigm

RAM-based platform needed for:
- flexibility, programmability
- avoiding the need of specific silicon

mask cost:
currently 2 mio $
- rapidly growing

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

TU Kaiserslautern

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

PDP'04, La Coruna, Spain, Febr. 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*

SBAC-PAD 2004 - 16th Symposium on Computer Architecture and High Performance Computing, Foz do Iguacu, PR, Brazil, October 27-29, 2004: topic area explicitely: Reconfigurable Systems

HPCA-11, 11th International Symposium on High-Performance Computer Architecture, San Francisco, Febr. 12-16, 2005: topic area explicitely: Embedded and reconfigurable architectures

 43 *) keynote speaker: http://hartenstein.de

TU Kaiserslautern

Conclusions

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 !

 44 http://hartenstein.de

TU Kaiserslautern

END

 45 http://hartenstein.de