invited in-depth presentation; 9th International Forum on Embedded MPSoC and Multicore, 2-7 Aug. 2009, Savannah, Georgia, USA

MPSoC '09

Reiner Hartenstein

reiner@hartenstein.de

Multicore Programming and the CS Education Dilemma

Don‘t forget Reconfigurable Computing (RC)

Technische Universität Kaiserslautern

Teaching for Change: an early martyr

„Turing is irrelevant“

David Lorge Parnas (keynote), "Teaching for Change“, 10th Conf. on Software Engineering Education and Training (CSEET '97)
http://www.sigsoft.org/SEN/parnas.html

The von Neumann model is the emulation of a tape machine

„The von Neumann syndrome“: coined ~ a decade later

Prof. C.V. Ramamoorthy, (UC Berkeley), SDPS 2006, San Diego, CA

Dijkstra 1968: The Goto considered harmful

R.H. & G. Koch 1975: The universal Bus considered harmful

Backus, 1978: Can programming be liberated from the von Neumann style?

Arvind et al., **1983:** **A critique of Multiprocessing the von Neumann Style**

Peter G. **Neumann** 1985-2003: 216x “Inside Risks“ (18 years inside back cover C_ACM)

B. Cox 1990: Planning the Software Industrial Revolution

L. Savain 2006: Why Software is bad …

Outline (1)

- The Power Consumption of Computing
- The Single-Core Approach
- The Multicore Scenario
- The Silver Bullet?
- A CPU-centric Flat World
- The Generalisation of Software Engineering
- Conclusions

3

Impact of the von Neumann Syndrome

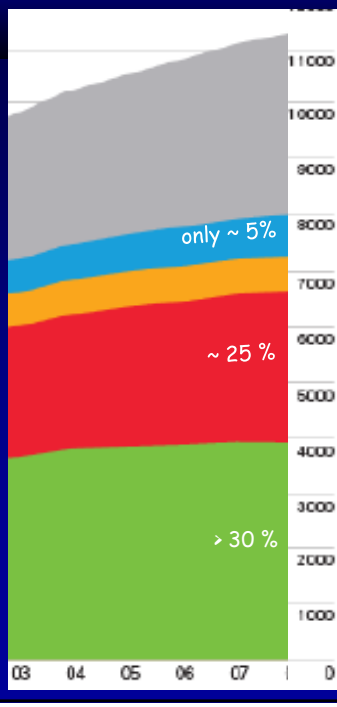
NACHTEILE DES VON-NEUMANN-PRINZIPS
[1989 from a student at Kaiserslautern]

never run out of energy?

Dig more coal -- the PCs are coming
Peter W. Huber, Mark P. Mills, 05.31.99
Forbes.com
http://www.forbes.com/forbes/1999/0531/6311070a.html

4

never run out of energy?



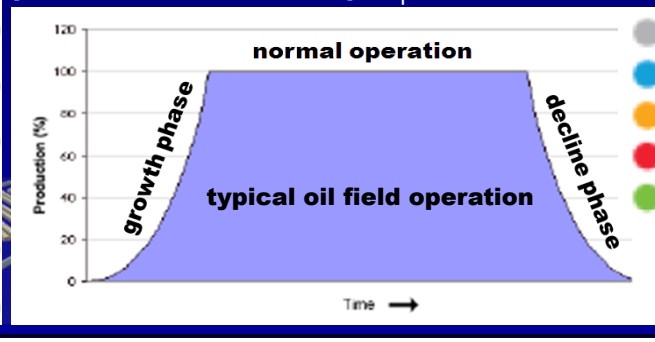
2007:

80% crude oil coming from decline fields

natural gas: similar situation

(worldwide population growth)

[Fatih Birol, Chief Economist IEA]. https://www.theoildrum.com/



coal
hydro
nuclear
gas
oil

Server Farms

at banks of the Columbia river:

[Randy Katz: IEEE Spectrum, Febr. 2009]

Power consumption by internet:
x30 til 2030 if trends continue

G. Fettweis, E. Zimmermann: ICT Energy Consumption - Trends and Challenges; WPMC'08, Lapland, Finland, 8 –11 Sep 2008 error corrected

(YAHOO! and)
Microsoft
at Quincey:
48 MW

amazon.com at Boardman

the electricity bill is a key issue

6

Power Consumption of Computers

... has become an industry-wide issue:
incremental improvements are on track,

but „we may ultimately need revolutionary new solutions"

[Horst Simon, LBNL, Berkeley]

Energy cost may overtake IT equipment cost in the near future

[Albert Zomaya]

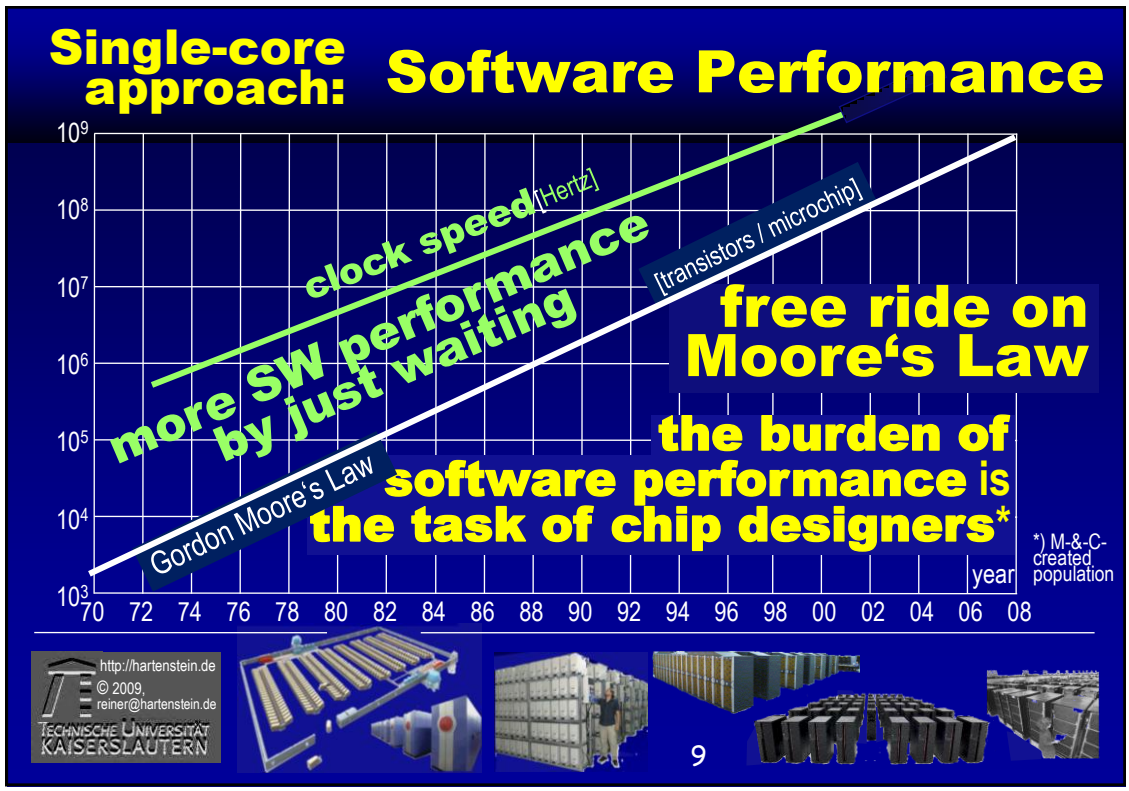
Current trends lead to unaffordable future operation cost of our cyber infrastructure

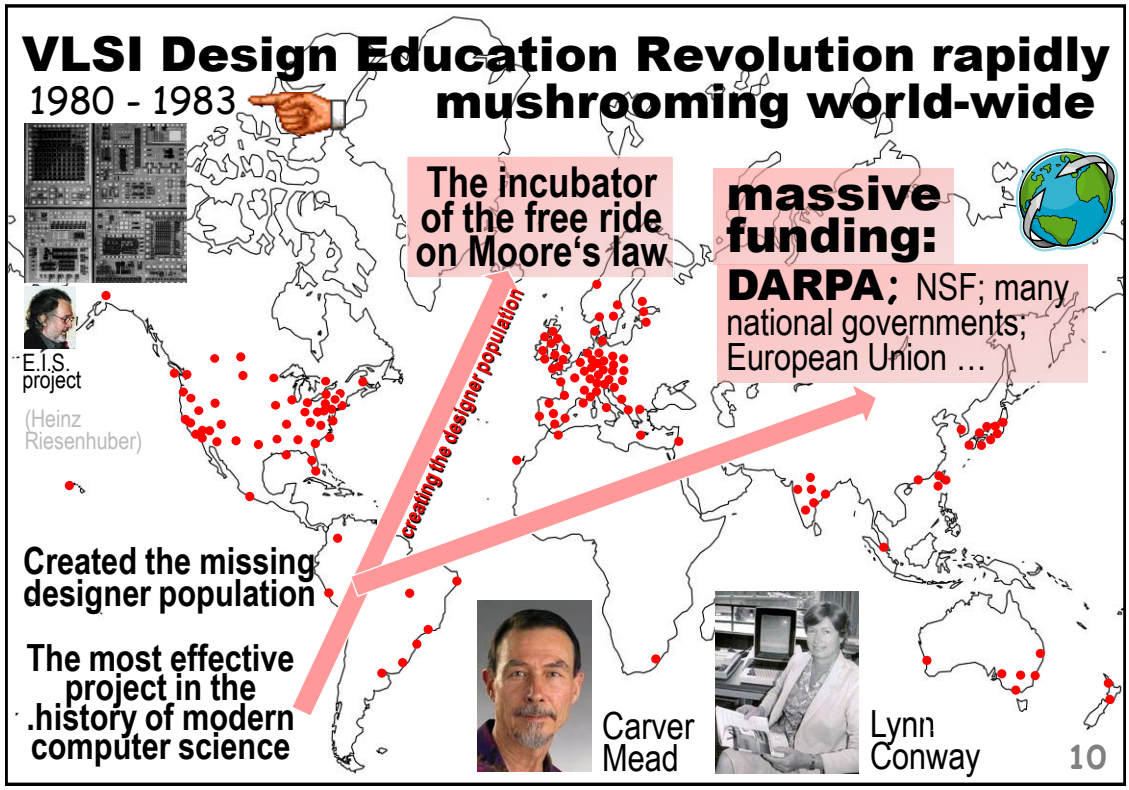
7

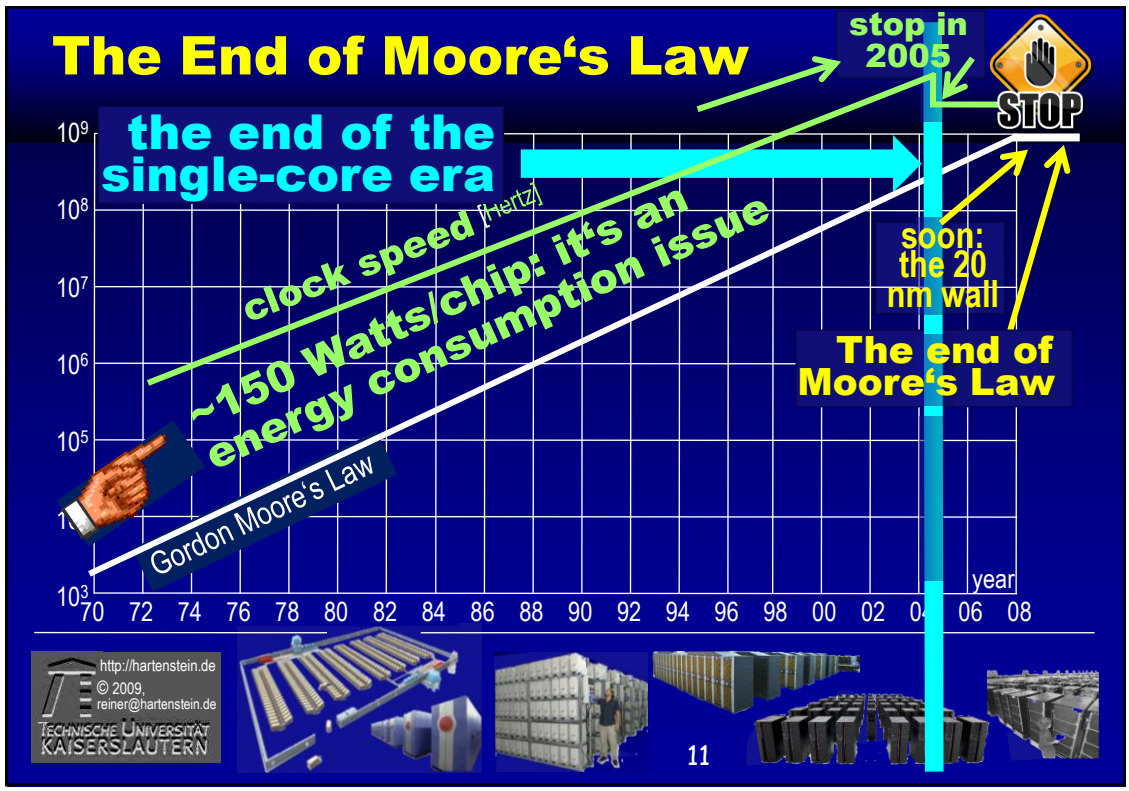
Outline (2)

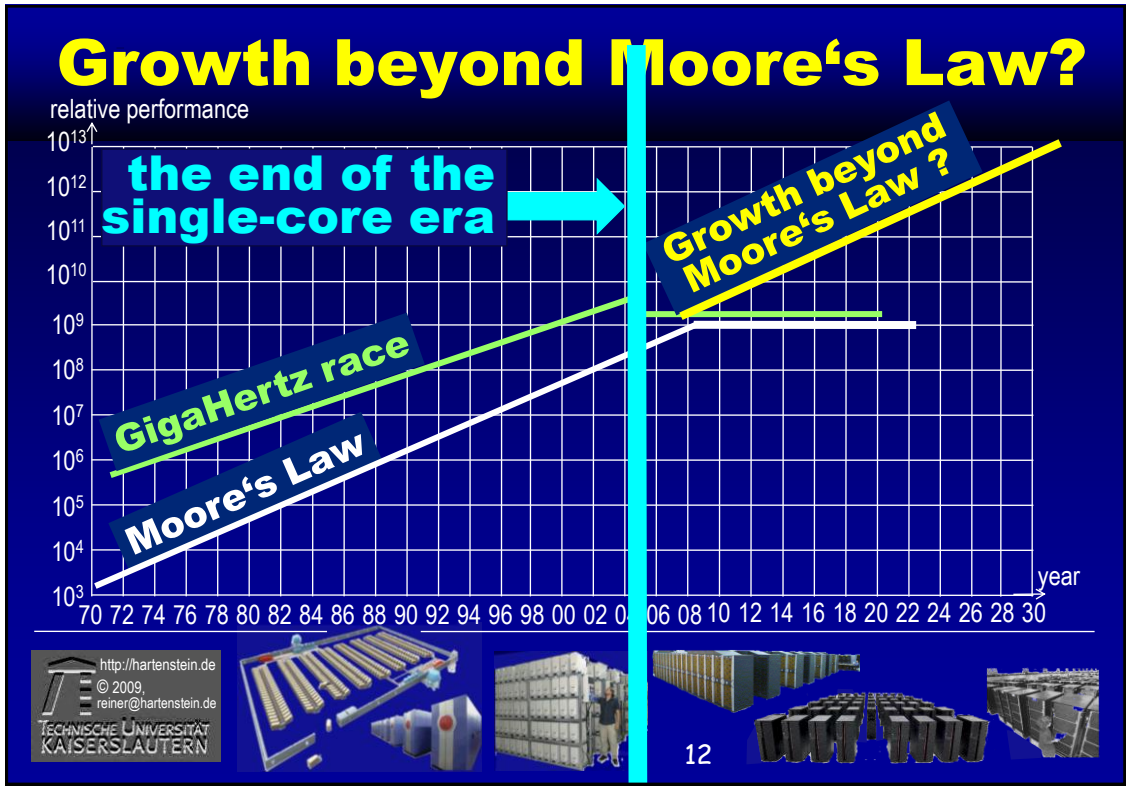
- The Power Consumption of Computing
- The Single-Core Approach
- The Multicore Scenario
- The Silver Bullet?
- A CPU-centric Flat World
- The Generalisation of Software Engineering
- Conclusions

8


Single-core approach:
Software Performance
clock speed [Hertz]
more SW performance by just waiting
[transistors / microchip]
free ride on Moore's Law
the burden of software performance is the task of chip designers*
Gordon Moore's Law
*) M-&-C-created population
year
http://hartenstein.de
© 2009, reiner@hartenstein.de
TECHNISCHE UNIVERSITÄT KAISERSLAUTERN
9


VLSI Design Education Revolution rapidly mushrooming world-wide
1980 - 1983
The incubator of the free ride on Moore's law
massive funding:
DARPA; NSF; many national governments; European Union …
creating the designer population
E.I.S. project
(Heinz Riesenhuber)
Created the missing designer population
The most effective project in the history of modern computer science
Carver Mead
Lynn Conway
10


The End of Moore's Law
stop in 2005
STOP
the end of the single-core era
clock speed [Hertz]
~150 Watts/chip: it's an energy consumption issue
soon: the 20 nm wall
The end of Moore's Law
Gordon Moore's Law
year
http://hartenstein.de
© 2009, reiner@hartenstein.de
Technische Universität Kaiserslautern
11


Growth beyond Moore's Law?
relative performance
the end of the single-core era
Growth beyond Moore's Law ?
GigaHertz race
Moore's Law
year
http://hartenstein.de
© 2009, reiner@hartenstein.de
Technische Universität Kaiserslautern
12

ICT is at an inflection point

e.g., the living room commercially more important than the comparatively small PC market.

„Future prosperity depends on network capacity, ..., efficient pricing, and flexible platforms"

Cheap Revolution:
affordable broadband
software performance

Senior Counselor to the U.S. Trade Representative (USTR) on strategy and negotiations.

„Broadband is significant at the inflection point, prompting major market governance changes"

Cowhey's & Aronson's Law: **massive funding needed**

13

Funding major market governance changes

RUS Broadband Initiatives Program (BIP)

NTIA Broadband Technology Opportunities Program (BTOP).

other sources ?

DARPA ?

ARPA-E ?

EFRCEs ?

EU-FP7 ?

The portal to apply for broadband funding under the American Recovery and Reinvestment Act of 2009

http://www.broadbandusa.gov/

hurry up !!

Energy Frontier Research Centers 777 bio $

14

Outline (3)

- The Power Consumption of Computing
- The Single-Core Approach
- The Multicore Scenario
- The Silver Bullet?
- A CPU-centric Flat World
- The Generalisation of Software Engineering
- Conclusions

15

16

Multicore is not new

Dead (Super)Computer Society [Gordon Bell, keynote, ISCA 2008]

- ACRI
- Alliant
- American Supercomputer
- Ametek
- Applied Dynamics
- Astronautics
- BBN
- CDC
- Convex
- Cray Computer
- Cray Research
- Culler-Harris
- Culler Scientific
- Cydrome
- Dana/Ardent/ Stellar/Stardent
- DAPP
- Denelcor
- Elexsi
- ETA Systems
- Evans and Sutherland Computer
- Floating Point Systems
- Galaxy YH-1
- Goodyear Aerospace MPP
- Gould NPL
- Guiltech
- ICL
- Intel Scientific Computers
- International Parallel Machines
- Kendall Square Research
- Key Computer Laboratories
- MasPar
- Meiko
- Multiflow
- Myrias
- Numerix
- Prisma
- Tera
- Thinking Machines
- Saxpy
- Scientific Computer
- Systems (SCS)
- Soviet Supercomputers
- Supertek
- Supercomputer Systems
- Suprenum
- Vitesse Electronics

only 2 or 3 successes

most in 1985-1995 - mainly research

the single core sequential mind set was the winner

EDA: a SE issue

„Multicore computers shift the burden of Software Performance to Software Developers."

[J. Larus: Spending Moore's Dividend; C_ACM, May 2009]

Multicore: SE crisis tightens the EDA crisis

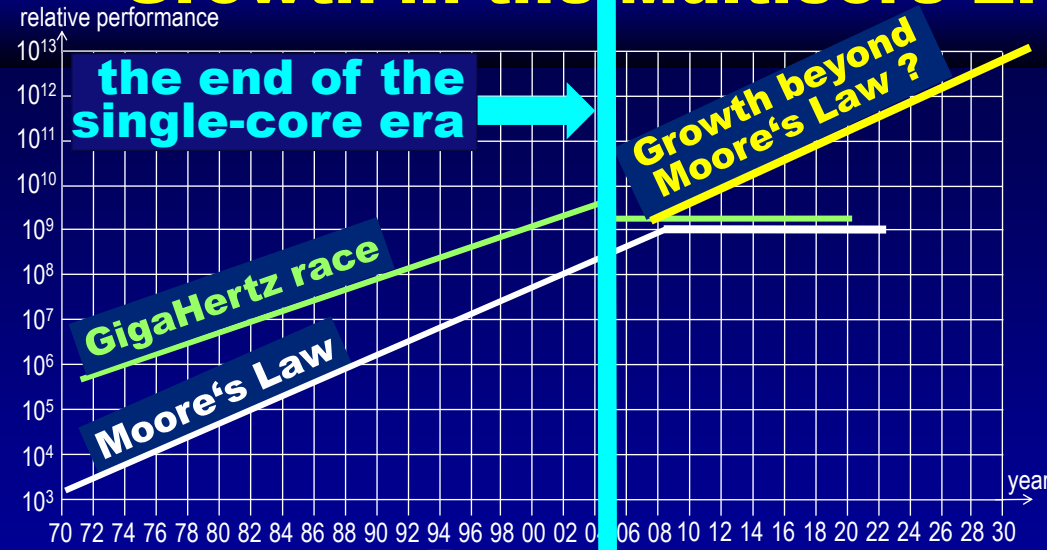
SE has been an EDA issue all the time

However, EDA must become a SE issue

SE scene hesitating: **this is our job !!**

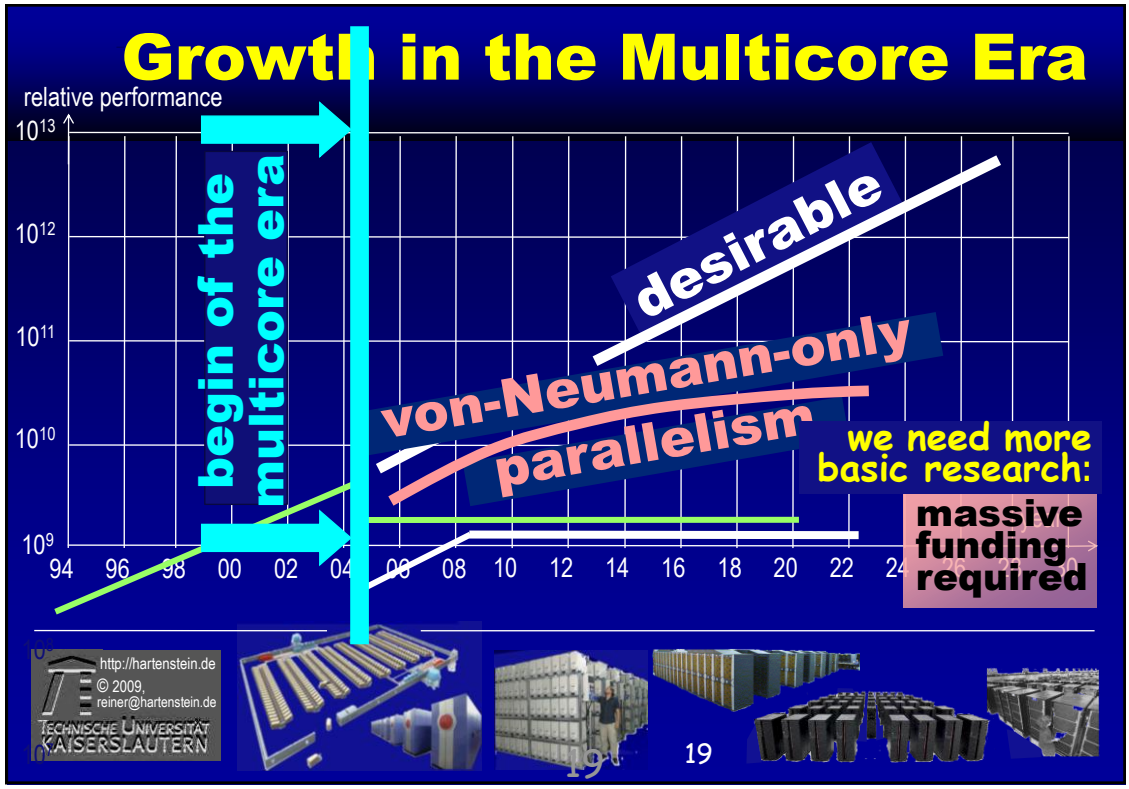
17

Growth in the Multicore Era



18

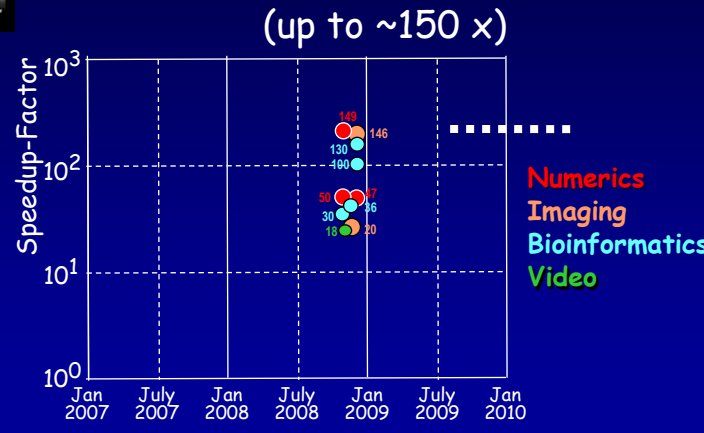
Growth in the Multicore Era



Speed-up factors by GPGPUs (1)

[Michael Garland, NVIDIA Research: Parallel Computing on Manycore GPUs; IPDPS, Rome, Italy, June 25-29, 2009]

The power efficiency is disputable



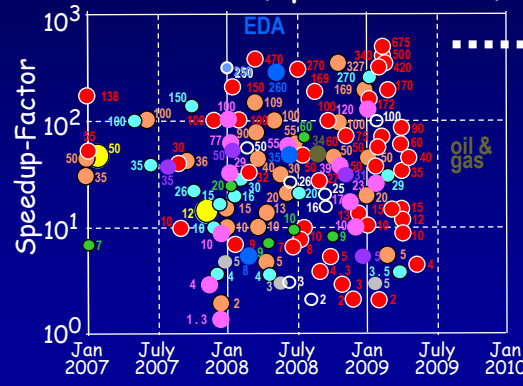
20

Speed-up factors by GPGPUs (2)

http://www.nvidia.co.uk/object/cuda_home_uk.html#state=home
CUDA ZONE pages [NVIDIA Corp.]:
non-reviewed CUDA user submissions

power consumption not reported!

(up to ~600 x)



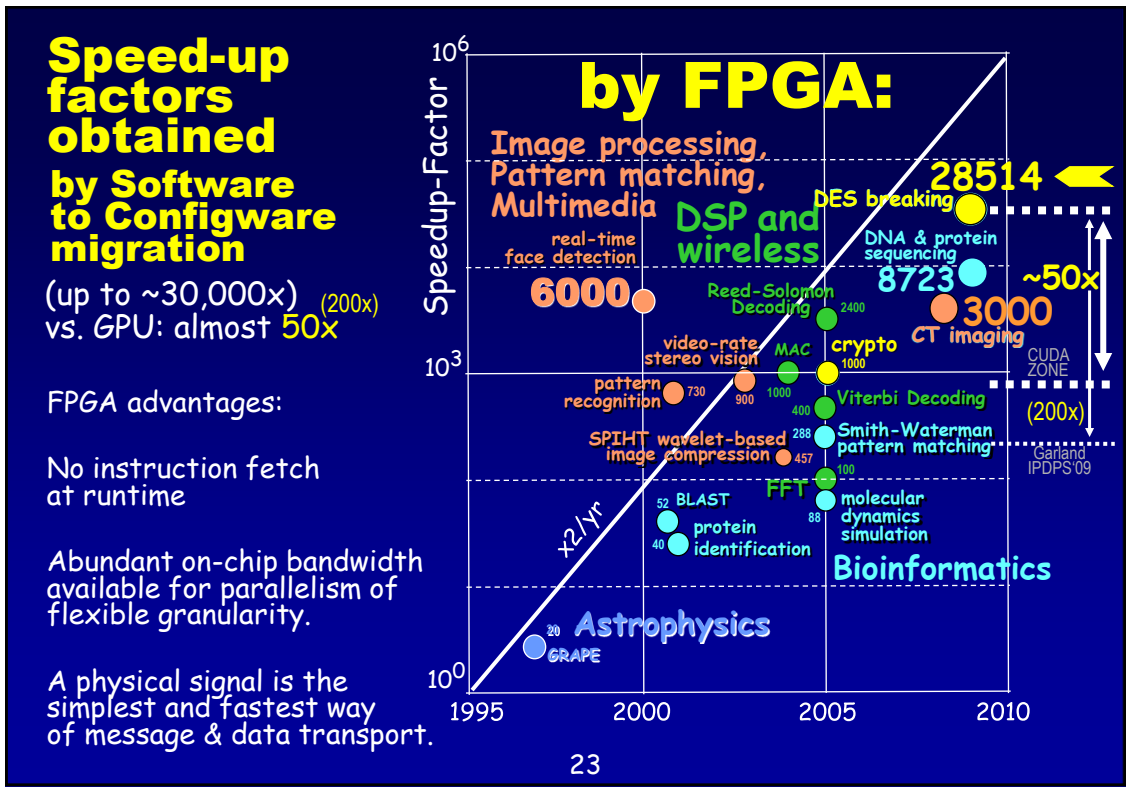
Astrophysics
Bioinformatics
CFD Computational Fluid Dyamics
Cryptography
DCC Digital Content Creation
DSP Digital Signal Processing
Graphics
Imaging
Numerics
Video & Audio

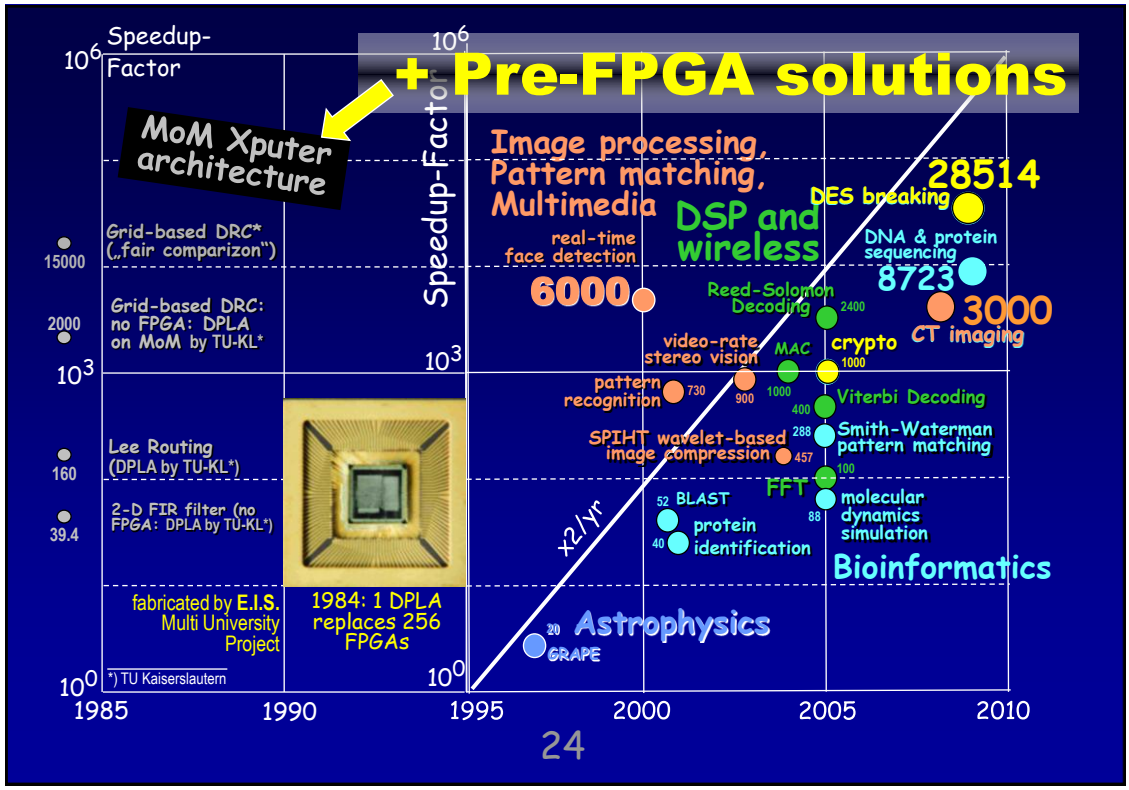
21

Outline (4)

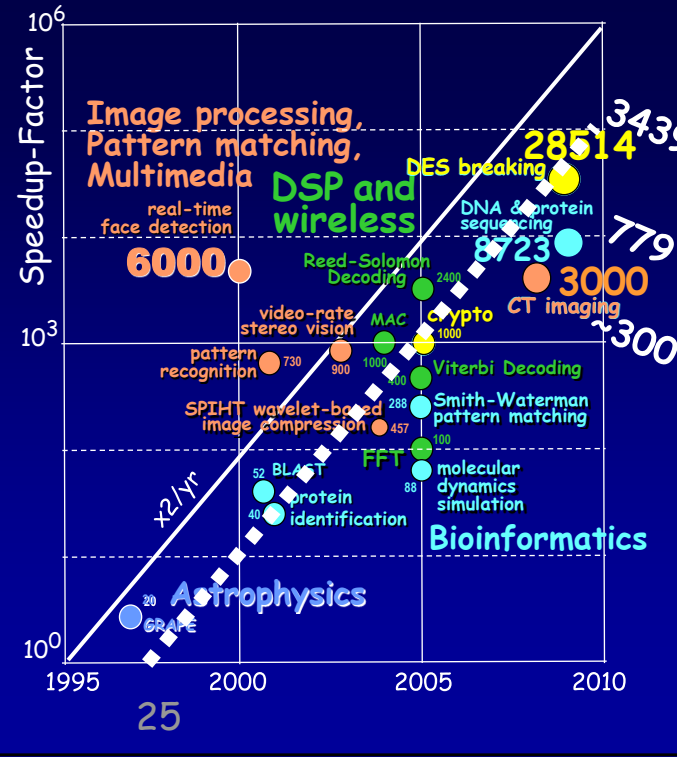
- The Power Consumption of Computing
- The Single-Core Approach
- The Multicore Scenario
- The Silver Bullet?
- A CPU-centric Flat World
- The Generalisation of Software Engineering
- Conclusions

22


Speed-up factors obtained
by Software to Configware migration
(up to ~30,000x) (200x)
vs. GPU: almost 50x
FPGA advantages:
No instruction fetch at runtime
Abundant on-chip bandwidth available for parallelism of flexible granularity.
A physical signal is the simplest and fastest way of message & data transport.
by FPGA:
Speedup-Factor
Image processing, Pattern matching, Multimedia
DSP and wireless
Bioinformatics
Astrophysics
real-time face detection
6000
DES breaking
28514
DNA & protein sequencing
8723
CT imaging
3000
~50x
CUDA ZONE
(200x)
Garland IPDPS'09
Reed-Solomon Decoding
2400
video-rate stereo vision
MAC
crypto
1000
pattern recognition
730
900
1000
400
Viterbi Decoding
SPIHT wavelet-based image compression
457
288
Smith-Waterman pattern matching
100
FFT
52
BLAST
protein identification
40
molecular dynamics simulation
88
x2/yr
20
GRAPE
1995
2000
2005
2010
23


+ Pre-FPGA solutions
Speedup-Factor
MoM Xputer architecture
Grid-based DRC* („fair comparizon")
15000
Grid-based DRC: no FPGA: DPLA on MoM by TU-KL*
2000
Lee Routing (DPLA by TU-KL*)
160
2-D FIR filter (no FPGA: DPLA by TU-KL*)
39.4
fabricated by E.I.S. Multi University Project
*) TU Kaiserslautern
1984: 1 DPLA replaces 256 FPGAs
Image processing, Pattern matching, Multimedia
DSP and wireless
Bioinformatics
Astrophysics
real-time face detection
6000
DES breaking
28514
DNA & protein sequencing
8723
CT imaging
3000
Reed-Solomon Decoding
2400
video-rate stereo vision
MAC
crypto
1000
pattern recognition
730
900
1000
400
Viterbi Decoding
SPIHT wavelet-based image compression
457
288
Smith-Waterman pattern matching
100
FFT
52
BLAST
protein identification
40
molecular dynamics simulation
88
x2/yr
20
GRAPE
1985
1990
1995
2000
2005
2010
24

Energy saving factors: ~10% of speedup



25

RC: Demonstrating the intensive Impact

[T. Elghazawi et al.: IEEE COMPUTER, Febr. 2008]

SGI Altix 4700 with RC 100 **RASC** compared to Beowulf cluster

Application	Speed-up factor	Savings		
		Power	Cost	Size
DNA and Protein sequencing	8723	779	22	253
DES breaking	28514	3439	96	1116

FPGA has a very high internal bandwidth compared to a CPUs with a few buses.

Much less equipment needed

Massively saving energy

Much less memory and bandwidth needed

26

Why such Speed-up Factors?

FPGAs: much worse technology

Wiring overhead

Reconfigurability overhead

Routing congestion growing with FPGA size

➡ The Reconfigurable Computing Paradox

Reason #1: **the von Neumann Syndrome**

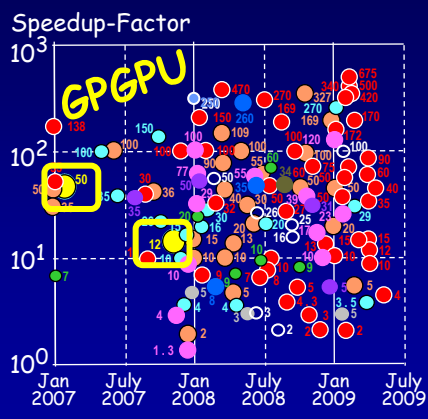
Reason #2: recently more „platform FPGAs"

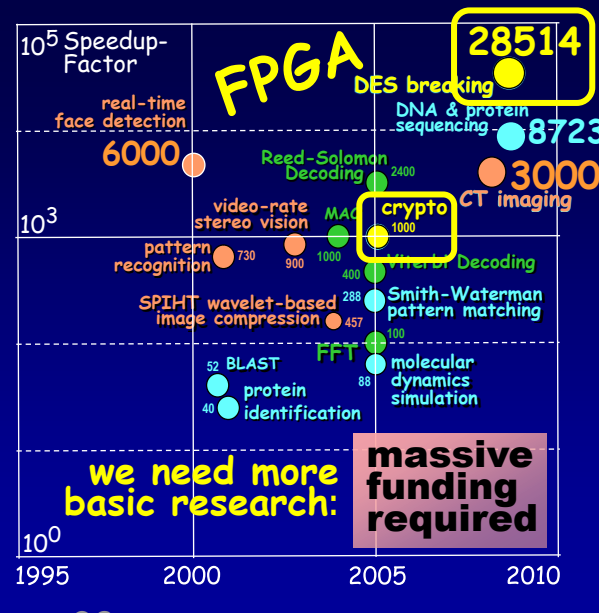
27

Missing Taxonomy of Algorithms

missing: compare GPGPU vs. FPGA per application

● 12 / 50 vs. 1000 / 28500 ?





28

RC versus Multicore

RC: speed-up often higher by orders of magnitude — Sure !

RC: energy-efficiency often higher: very much, or, by orders of magnitude ?

probably – recommend a study

this is the silver bullet

We need **both**: Multicore and RC

For a Booming Multicore Era

von-Neumann-only is not the silver bullet
Reconfigurable Computing is indispensable!

Outline (5)

- The Power Consumption of Computing
- The Single-Core Approach
- The Multicore Scenario
- The Silver Bullet?
- A CPU-centric Flat World
- The Generalisation of Software Engineering
- Conclusions

CPU-centric flat world

(Aristotelian model)

typical programmer qualification:

sequential-only mind set –

CPU-"centric" but no hardware know-how

The Software-centric world model is obsolete

Machine Model of the Mainframe Era

Machine model	resources		sequencer		
	property	programming source	property	programming source	state register
CPU	hardwired	-	programmable	**Software** (instruction streams)	**program counter**

33

Computer Machine Model of the PC Era

Machine model	resources		sequencer		
	property	programming source	property	programming source	state register
ASIC accelerator	hardwired	-	hardwired	SE: no change	
CPU	hardwired	-	programmable	**Software** (instruction streams)	**program counter**

"the tail is wagging the dog"

ISIS 1997
Austin, TX

Application-Specific Integrated Circuit & other accelerators: e.g. graphics processor

34

Nathan Myhrvold

"Software" stands for extremely memory-cycle-hungry instruction streams

Nathan's Law: Software is a gas.
It expands to fill the container it is in ...
... until being limited by Moore's [& Kryder's] Law

The von Neumann Syndrome: overhead piles up to code sizes of **astronomic dimensions**

Dave Patterson

Patterson's Law:
"The Memory Wall"
coined by Sally McKee

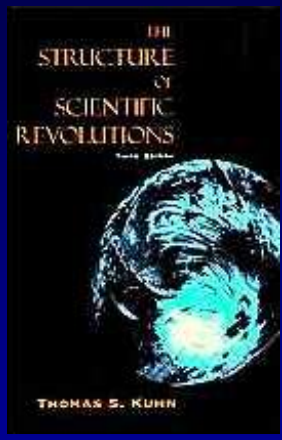
from earlier talks:

from earlier talks:

35

Thomas S. Kuhn

Thomas S. Kuhn 1969:
The Structure of Scientific Revolutions



40 years SE crisis

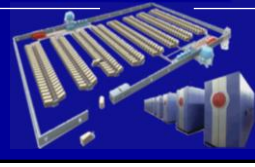
Science does not progress continuously,

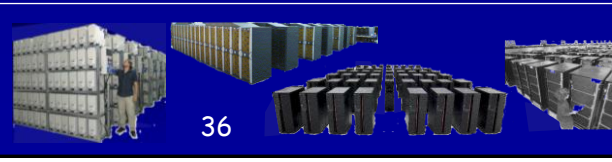
... shortcomings in an established paradigm produces **a crisis that may lead to a revolution**

F. L. Bauer:
„SE crisis"
coined 1968

? in which the established paradigm is overthrown and replaced. ?

The von Neumann paradigm?





36

No Revolution !

ACM / IEEE-CS joint taskforce

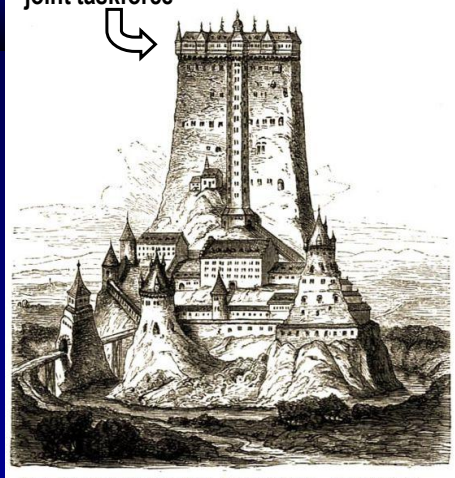


Fig. 1. Burg Fleckenstein im Elsaß. (11. bis 16. Jahrh. — Nach Specklin.)

ACM/IEEE-CS Curriculum recommendations

unconquerable: ***the Fleckenstein Effect***

„We do not want to discuss any details !!"

Algorithms and Complexity (p. 44): „....the **parallelism topics remain listed here as elective**, ..."

the role of parallelism throughout the curriculum needs to be considered."

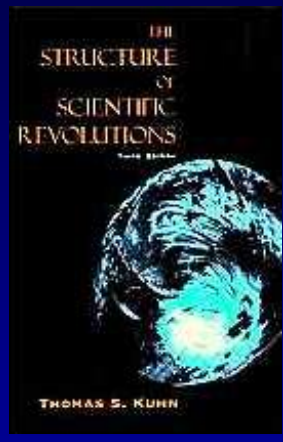
only end of last page (p. 108):
„Co-processor techniques including **GPU**, Cell, **FPGA**, characteristics of co-processor programming methodologies

Computer Science Curriculum 2008: An Interim Revision of CS 2001; Dec. 2008, ACM, IEEE-CS
http://www.acm.org//education/curricula/ComputerScience2008.pdf

37

(2) Thomas S. Kuhn

Thomas S. Kuhn 1969: The Structure of Scientific Revolutions



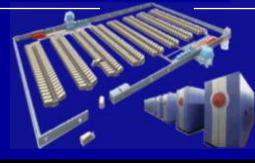
shortcomings in an established paradigm produces **a crisis that may lead to a revolution**

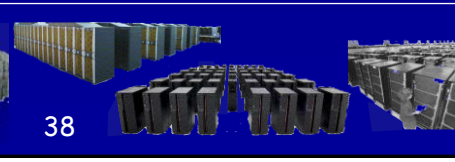
25 years EDA crisis

? in which the established paradigm is overthrown and replaced. ?

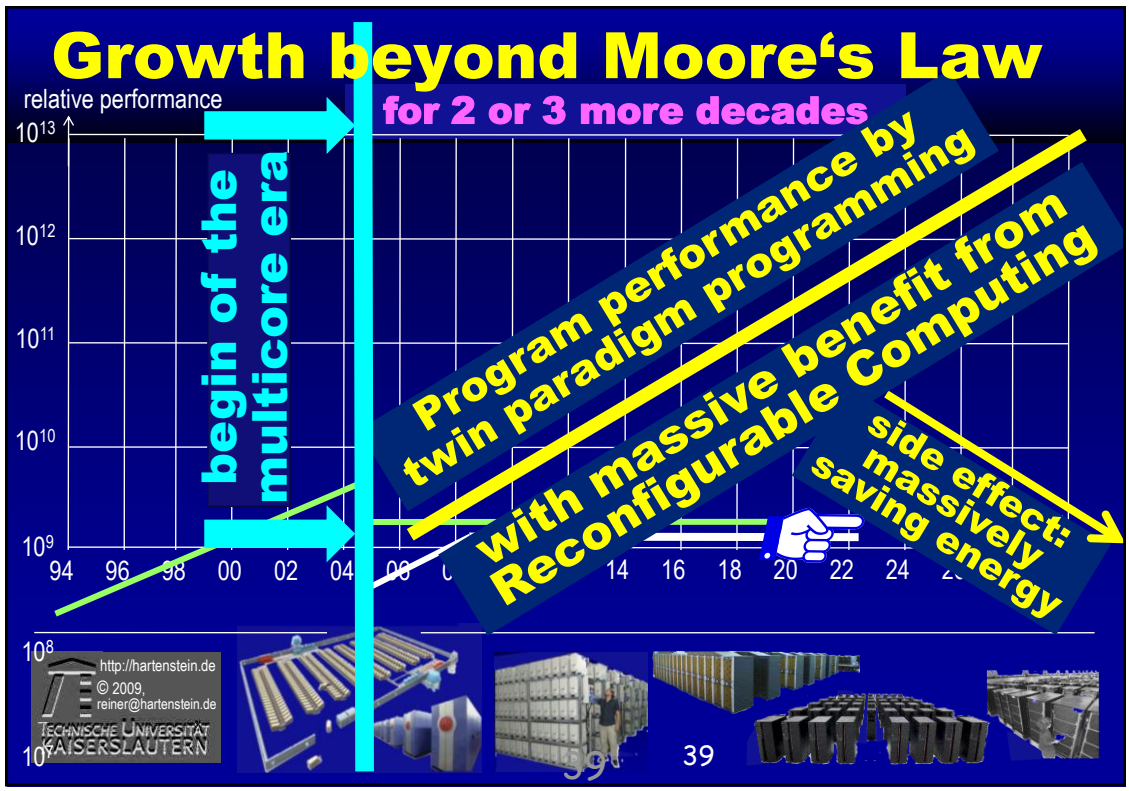
is there any established paradigm in EDA?

If yes: which one is it?





38



Outline (6)

- The Power Consumption of Computing
- The Single-Core Approach
- The Multicore Scenario
- The Silver Bullet?
- A CPU-centric Flat World
- The Generalisation of Software Engineering
- Conclusions

From CPU to RPU

Reconfigurable Processing Unit

Machine model	resources		sequencer		
	property	programming source	property	programming source	state register
ASIC accelerator	hardwired	-	hardwired	-	
CPU	hardwired	-	programmable	**Software** (instruction streams)	**program counter**
RPU accelerator 30:1 dominance	programmable	**Configware** (configuration code)	programmable	**Flowware** (data streams)	**data counters**

Now accelerators are programmable we need 2 more program sources

41

40 years Software Crisis

Thomas Kuhn is right !

… in which the established paradigm is **overthrown and replaced.”**

[Thomas S. Kuhn 1969: The Structure of Scientific Revolutions]

However, **not the von Neumann paradigm** will be overthrown and replaced.

The CPU-centric world model of Software Engineering will be replaced – at least rel. to the CMP/EDA scene

42

RC outside a CPU-centric flat world?

For the Multicore era we need a new model (Copernican)

43

Program Performance

„Multicore computers shift the burden of software performance to ~~software~~ program developers."

[J. Larus: Spending Moore's Dividend; C_ACM, May 2009]

People have to write code differently

We urgently need a Software Engineering Education Revolution

Missing programmer population and methodology:
a scenario like before the Mead-&-Conway revolution

But, what programmer qualifications are needed ?

44

Which Direction ?

multi-threading only ?
hopeless ? GPGPU ?
x86 multicore ?
FPGA ? hetero ?
what language ? C ?
imperative ? HDL ?
concurrent ? OO ?
learn'g hdw needed ?
lean hdw qualification ?
transactional memory ?

auto-parallelizing compiler ?

we've a developing huge tool market
very important objective: not to shy away undergraduate students

45

A Multicore Submarine Model

(this is not a lecture on brain regions)

46

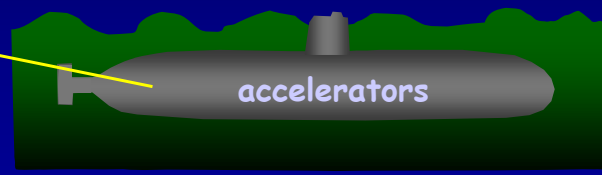
Brain usage:
procedural-only
(time domain only)

SE domain untouched

application ?

vN automatically parallelizing compiler

space domain invisible



this is not the silver bullet

mapping parallelism just into the time domain:
"abstracting" away the space domain is fatal

The Computing Education Wall

hardware must be visible!

limited perspective

von Neumann paradigm

not yet really a paradigm

for world-wide action:

massive funding required

µprocessor ⟷ accelerators

time domain: instruction streams

time domain: data streams
+ space domain: pipe networks
- as a paradigm

lean hardware qualification by dual dichotomy
not scaring away undergraduate students

D. Parnas [1997]

Putting **O**ld **i**deas **i**nto **P**ractice

teaching flowware in addition to software

47

A Heliocentric CS Model

special Flowware Engineering

FE

time to space mapping

instruction streams

data streams

CPU SE

special

Software Engineering

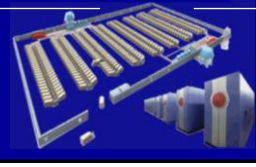
PE

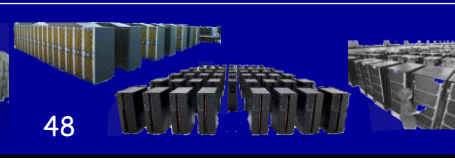
general

Program Engineering

The Generalization of Software Engineering —

A Twin Paradigm Dual Dichotomy Approach.





48

Our Contemporary Computer Machine Model

Machine model	resources		sequencer		
	property	programming source	property	programming source	state register
ASIC accelerator	hardwired	-	hardwired	-	
CPU	hardwired	-	programmable	**Software** (instruction streams)	**program counter in CPU**
RPU accelerator	programmable	**Configware** (configuration code)	programmable	**Flowware** (data streams)	**data counters in RAM**

data counters of reconfigurable address generators in asM (auto-sequencing) data memory blocks

twin Paradigm Dichotomy

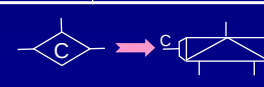
the same language primitives!

Time to Space Mapping

Machine model	resources		sequencer		
	property	programming source	property	programming source	state register
ASIC accelerator	hardwired	-	hardwired	-	
CPU	hardwired	time to space mapping	programmable	**Software** (instruction streams)	**program counter**
RPU accelerator	programmable	**Configware** (configuration code)	programmable	**Flowware** (data streams)	**data counters**

loop turns 2 pipeline

Relativity Dichotomy



„The biggest payoff will come from Putting Old ideas into Practice and teaching people how to apply them properly."

Flowware

Flowware means parallelism resulting from time 2 space migration

Generalization of th systolic array:

Any wild free form pipe networks

spiral, zigzag, more wild, fork and join,

fully or partially bidirectional

e.g. bubble sort conversion

Flowware: scheduling data streams

Fifos, stacks, registers, register files, RAM blocks

51

The 3 Walls

the Energy Wall,
the Memory Wall,
the Education Wall.

„Tear down these walls, Mr. Gorbatschov"

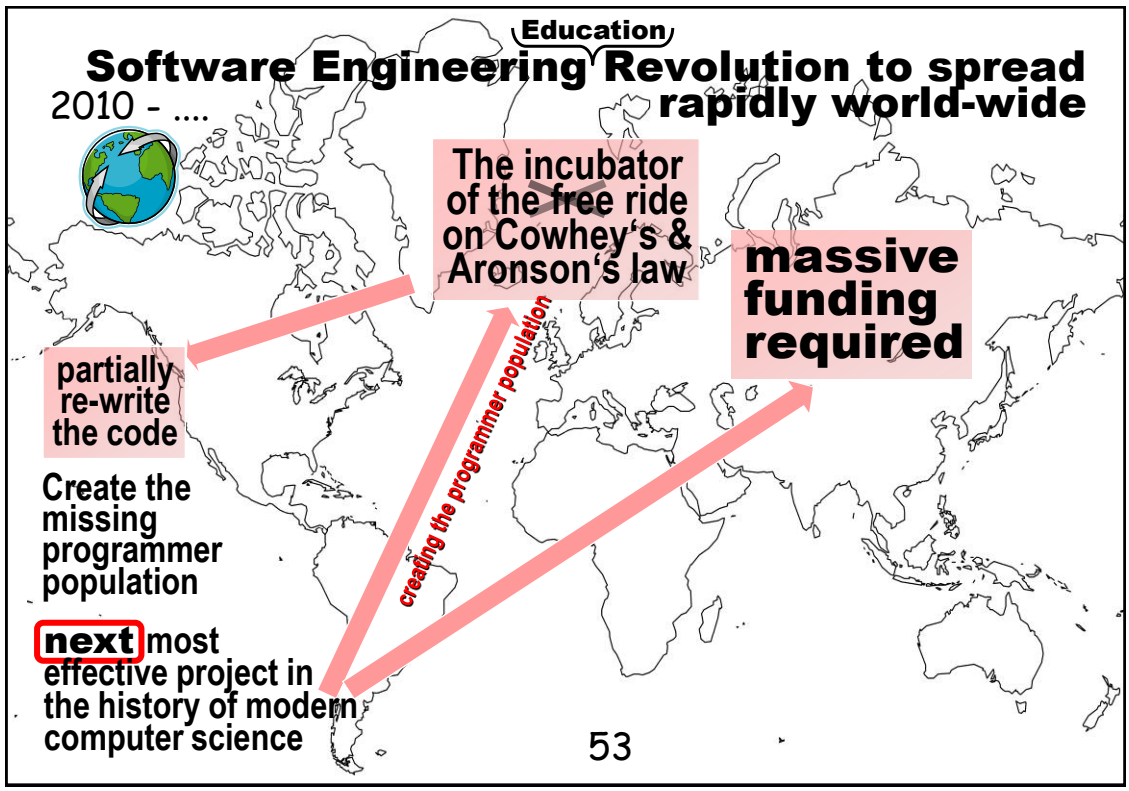
... by a Software Engineering Revolution

Education

for world-wide action:

massive funding required

52



SE Education Revolution

Software Engineering

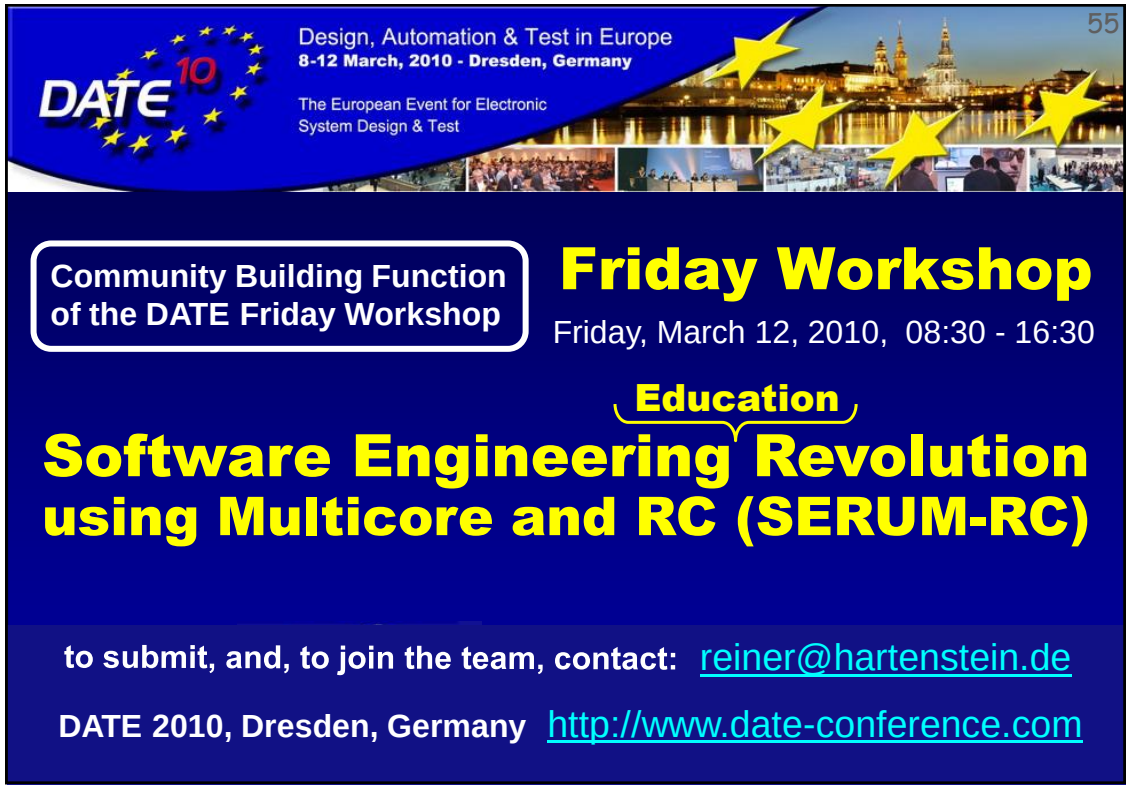
by simultanous dual domain co-education:

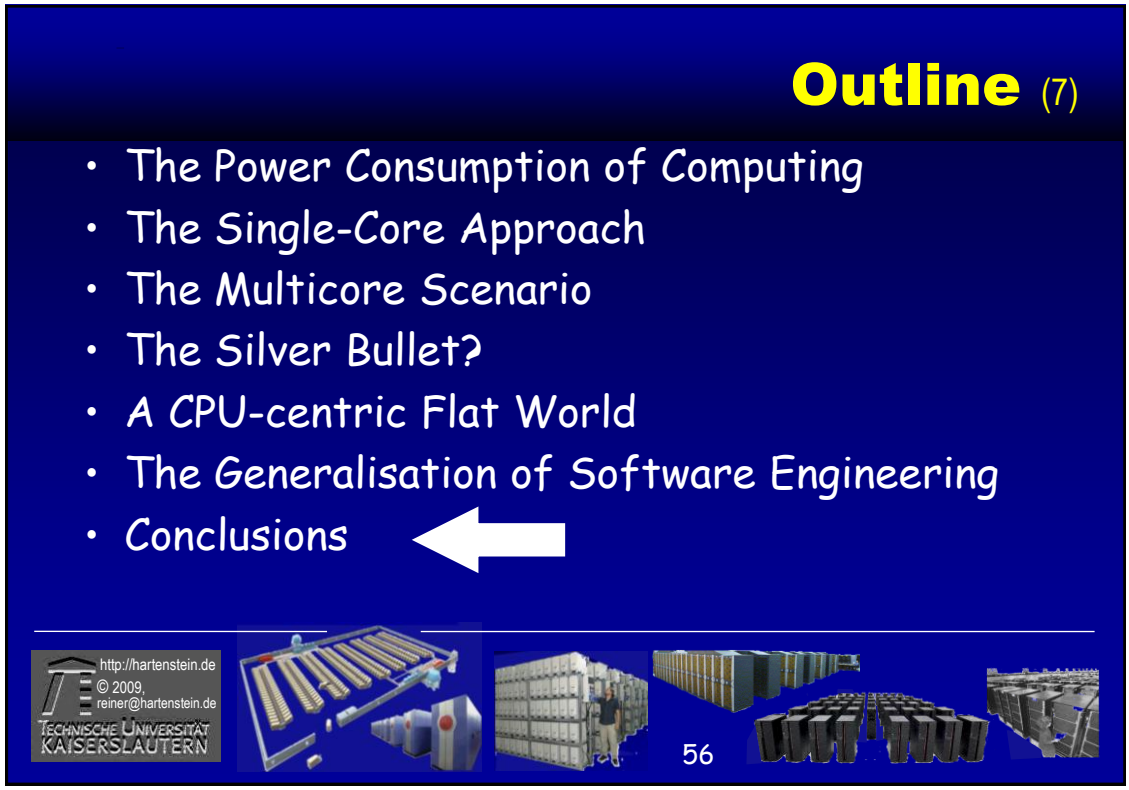
traditional qualification in the time domain

\+ lean qualification in the space domain
= lean hardware modeling qualification at a higher level of abstraction

lean ?
not to scare away undergraduates

➡ viable methodology for dual rail education
(only a few % curricula need to be changed)


55
DATE 10
Design, Automation & Test in Europe
8-12 March, 2010 - Dresden, Germany
The European Event for Electronic System Design & Test
Community Building Function of the DATE Friday Workshop
Friday Workshop
Friday, March 12, 2010, 08:30 - 16:30
Education
Software Engineering Revolution using Multicore and RC (SERUM-RC)
to submit, and, to join the team, contact: reiner@hartenstein.de
DATE 2010, Dresden, Germany http://www.date-conference.com


Outline (7)
• The Power Consumption of Computing
• The Single-Core Approach
• The Multicore Scenario
• The Silver Bullet?
• A CPU-centric Flat World
• The Generalisation of Software Engineering
• Conclusions
http://hartenstein.de
© 2009, reiner@hartenstein.de
TECHNISCHE UNIVERSITÄT KAISERSLAUTERN
56

Conclusions (1)

To maintain a Booming Multicore Era Reconfigurable Computing is indispensable!

57

58

Conclusions

key issues: performance and energy consumption of programs

Textbook on the way

Flowware / Configware skills are additional essential qualifications for programmers.

The main problem to solve:

which advanced lab course tools for training this programmer community ?

We need to master hetero of all 3: Singlecore, Multicore, & Reconfigurable Computing

Mead-&-Conway-dimension SE Revolution toward twin-paradigm education is is urgently needed

SERUM-RC

hetero development tools, environments & lab courses are a cardinal problem

massive long term R&D funding required
like known from DARPA

thank you for your patience

END

61

backup for discussion

62

Double Dichotomy

1) Twin Paradigm Dichotomy

von Neumann Machine
instruction stream
(Software-Domain)

⟷

Datastream Machine
data stream
(Flowware-Domain)

2) Relativity Dichotomy

time:
-Procedure
(Software-Domain)

⟷

space:
-Structure
(Configware-Domain)

Relativity Dichotomy

time ⟶ space *(time ⟶ time/space)*

time domain:
procedure domain

2 phases:
1) programming instruction streams
2) run time von Neumann Machine

space domain:
structure domain
3 phases:
1) reconfiguration of structures
2) programming data streams
3) run time Datastream Machine

time-iterative to space-iterative

n time steps, 1 CPU

a time to space mapping

1 time step, n DPUs

loop transformation methodogy: 70ies and later

e. g. example: bubble sort migration

Often the space dimension is limited

n*k time steps, 1 CPU

a time to space/time mapping

n time steps, k DPUs

POTTP

Strip mining

[D. Loveman, J-ACM, 1977]

65

time to space mapping

time domain: procedure domain

space domain: structure domain

time algorithm → space algorithm

program loop

n time steps, 1 CPU

pipeline

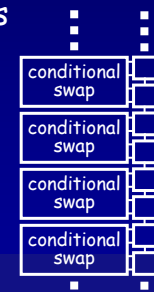
1 time step, n DPUs

Bubble Sort

n x k time steps, 1 „conditional swap" unit

Shuffle Sort

k time steps, n „conditional swap" units



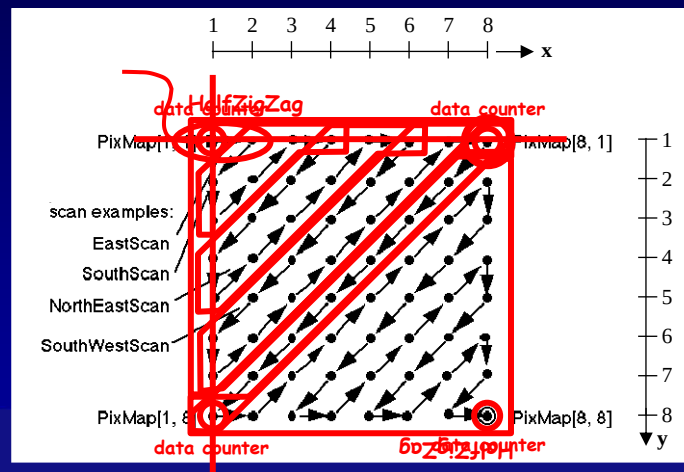
time algorithm → space/time algorithm s

66

JPEG zigzag scan pattern

Flowware language example (MoPL): programming the datastream

```
*> Declarations

4 EastScan is
    step by [1,0]
  end EastScan;

2 SouthScan is
    step by [0,1]
  endSouthScan;

3 NorthEastScan is
    loop 6 times until [*,1]
    step by [1,-1]
    endloop
  end NorthEastScan;

1 SouthWestScan is
    loop 7 times until [1,*]
    step by [-1,1]
    endloop
  end SouthWestScan;

  HalfZigZag is
    EastScan
    loop 3 times
      SouthWestScan
      SouthScan
      NorthEastScan
      EastScan
    endloop
  end HalfZigZag;

Main program:
  goto PixMap[1,1]
  HalfZigZag;
  SouthWestScan
  uturn (HalfZigZag)
```



67

(an animation)

CV of Reiner Hartenstein

Called* „The father of Reconfigurable Computing" (also pre-FPGA era)

Author of KARL**, the most successful trailblazing HDL before VHDL: 85 million ECU (pre-€) European Union grant for complete EDA framework around KARL

1981: visiting professor at UC Berkeley

Back from Berkeley: founder of the German Mead-&-Conway VLSI design scene: multi university „E.I.S project" (grant 1983: 35 million Deutschmark)

IEEE fellow, SDPS fellow, FPL fellow, and several other awards

Prof. em. TU Kaiserslautern, all degrees from Karlsruhe Institute of Technology

*) by Viktor Prasanna, as session chair at IPDPS 2004

**) R. Hartenstein: The History of KARL and ABL; in: J. Mermet (ed.): Fundamentals and Standards in Hardware Description Languages; ISBN 0-7923-2513-4, Kluwer*, September 1993.
also see: http://xputers.informatik.uni-kl.de/karl/karl_history_fbi.html

*) now: Springer Verlag

http://hartenstein.de
reiner@hartenstein.de

68