Disruptive Trends urging to rethink Embedded System Implementation

Reiner Hartenstein
The impact of shifting to multicore performance

4 P issues:
- performance
- programmer productivity
- program efficiency
- power consumption

market trends
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]

IPCC ?

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

"Google causes 2% of the world’s electricity consumption" (Google denied)

Energy cost may overtake IT equipment cost in the near future

© 2010, reiner@hartenstein.de

http://hartenstein.de
vN: a Massive Power Guzzler

it's a symptom of the von Neumann Syndrome:

**Software** is extremely power-hungry - by massively memory-cycle-hungry instruction streams

**Software**: has often very bad performance

we need an approach using much less **Software**

**triple paradigm**
Growth beyond Moore’s Law?

The end of the single-core era

GigaHertz race vs. Moore’s Law

Performance drops, productivity & other problems...

“Multicore shifts the burden of Performance from Chip Designer to Software Developers.”

“Spending Moore’s Dividend”

We need to learn parallel programming

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10^3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10^13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2010, reiner@hartenstein.de

http://hartenstein.de
Multimedia in the Multicore Era

Performance growth needed:
- Audio: 800 MIPS
- Graphics: 11 GOPS
- Video: 160 GOPS
- Digital TV: 900 GOPS

Needed performance growing faster than Moore’s law

[Courtesy E. Sanchez]
ICT market at an inflection point

The battle for the living room & mobile is more important than the PC market.

Prosperity depends on network capacity, ..., efficient pricing, flexible platforms, & ...

... Cheap Revolution: • low power
• affordable broadband
• software performance

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

Cowhey’s & Aronson’s Law & massive funding needed

© 2010, reiner@hartenstein.de

http://hartenstein.de

Senior Counselor to the U.S. Trade Representative (USTR) on strategy and negotiations.
Performance Growth by Multicore? & massive programmer productivity problems

begin of the multicore era & much slower than Moore's law

von-Neumann-only parallelism

von-Neumann-only is not the silver bullet

Reconfigurable Computing is indispensable!

Maximum Theoretical Speedup from Amdahl's Law
Learning from history?
(Multicore is not really new)

• 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

the single core sequential mind set was the winner
John Hennessy: widespread confusion and competing claims, "I would be panicked if I were in industry"

Hastily knitted compilers for the heavy lifting?

e.g. automatically parallelizing compilation via multi-threading, and many other ad-hoc solutions?

new types of bugs introduced

new types of bugs introduced

easy fix?
Michael Wrinn, (keynote at SIGCSE2010):
Suddenly, All Computing Is Parallel:
Seizing Opportunity Amid the Clamor


“Foundational change will disrupt traditional habits throughout the discipline ....“

“The proud era of von Neumann architecture passes into history.“

works to bring parallel computing into mainstream of undergraduate education

He also works with the ACM Education Council to bring industrial perspective to curriculum evolution.

... especially how students are to be introduced ....
HPRC: High Performance Reconfigurable Computers

programming dilemma... ... a taxonomy of design flows
### Application Speed-up factor | Savings
<table>
<thead>
<tr>
<th></th>
<th>Power</th>
<th>Cost</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DNA and Protein sequencing</strong></td>
<td>8723</td>
<td>779</td>
<td>22</td>
</tr>
<tr>
<td><strong>DES breaking</strong></td>
<td>28514</td>
<td>3439</td>
<td>96</td>
</tr>
</tbody>
</table>

much less memory and bandwidth needed
massively saving energy
much less equipment needed

no **software** used!
Speed-up factors obtained by Software to Configware migration

No instruction fetch at runtime:

no **software**!

Abundant on-chip bandwidth available for parallelism of flexible granularity (by FPGA).

A physical signal is the simplest and fastest way of message & data transport.

“cyber-physical computing”
Power save factors obtained

Energy saving factors: ~10% of speedup

GPGPU and x86 multicore: no energy saving data available

Low Power Circuit Design: PowerOpt™ (ChipVision Design Systems): divides power consumption by up to 4
Why such Speed-up Factors ...

... with FPGAs: a much worse technology!

- massive wiring overhead & slower clock
- massive reconfigurability overhead
- routing congestion growing with FPGA size

→ The „Reconfigurable Computing Paradox“

main reason: no von Neumann Syndrome!

no software!

using Configware and Flowware instead
Isn’t NVIDIA the solution?

begin of the multicore era

relative performance

year

von-Neumann-only parallelism

maybe, a few exceptions ...
Speed-up factors by GPGPUs (1)

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!

Drawbacks:
von Neumann syndrome,
Programmer productivity

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

http://hartenstein.de
© 2009, reiner@hartenstein.de

Drawbacks:
von Neumann syndrome,
Programmer productivity
power consumption not reported!

(up to ~600 x)
Speed-up factors obtained (2) by Software to Configware migration
(up to ~30,000x) vs. GPU: almost 50x
RC versus Multicore

RC: speed-up often higher by orders of magnitude

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

this is the silver bullet

We need both: Multicore and RC

Sure!
Patterson’s Law: bandwidth gap grows 50% / year
Dave Patterson has reached >1000x

“The Memory Wall” coined by Sally McKee (& co-author)

Nathan’s Law: Software is a gas. It expands to fill its containers …
until being limited by Moore’s Law [& Kryder’s Law]
„even fills the internet“

Wirth’s Law [Niklaus Wirth]
“Software is slowing faster than hardware is accelerating”

The von Neumann Syndrome:
C.V. Ramamoorthy

overhead piles up to code sizes of astronomic dimensions

The ugliness of this term "Software" stands for extremely memory-cycle-hungry instruction streams

© 2010, reiner@hartenstein.de
http://hartenstein.de
50 years Software Crisis

Max Planck:
Replacement of false doctrines by new insights needs 50 years waiting for not only old professors but also their scholars to die off.

Parkinson's Law
bureaucracy growth independent of actual work to be done

The time has come

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

L. Savain 2006:
Why Software is bad

Software Engineering critics is not new:
F. L. Bauer 1968, coined the term „Software Crisis“
N. N. 1995: THE STANDISH GROUP REPORT
Anthony Berglas 2008: Why it is Important that Software Projects Fail
CPU-centric flat world model

(Aristotelian model)

typical programmer qualification:
sequential-only mind set –

CPU-“centric” but no hardware know-how
(Kind of tunnel view)

This software-centric world model is obsolete

not visible from SE
von Neumann versus Anti-machine (data stream machine).

PE: the Generalization of Software Engineering — First Step

*) do not confuse with „dataflow“!
### Procedural Languages Twins

<table>
<thead>
<tr>
<th><strong>imperative Software Languages</strong></th>
<th><strong>systolic Flowware Languages</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>read next instruction</td>
<td>read next data item</td>
</tr>
<tr>
<td>goto (instruction address)</td>
<td>goto (data address)</td>
</tr>
<tr>
<td>jump to (instruction address)</td>
<td>jump to (data address)</td>
</tr>
<tr>
<td>instruction loop</td>
<td>data loop</td>
</tr>
<tr>
<td>instruction loop nesting</td>
<td>data loop nesting</td>
</tr>
<tr>
<td>instruction loop escape</td>
<td>data loop escape</td>
</tr>
<tr>
<td>instruction stream branching</td>
<td>data stream branching</td>
</tr>
<tr>
<td><strong>no:</strong> no internally parallel loops</td>
<td><strong>yes:</strong> internally parallel loops</td>
</tr>
</tbody>
</table>

But there is the Asymmetry for data parallelism

© 2010, reiner@hartenstein.de

http://hartenstein.de

TU Kaiserslautern
### Machine twins: different data movement

Who moves operand to operator if not an instruction? / from

<table>
<thead>
<tr>
<th>#</th>
<th>moving data between</th>
<th>data transport</th>
<th>execution triggered by</th>
<th>strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>von Neumann CPU cores</td>
<td>via common memory</td>
<td>instruction stream</td>
<td>moving data at run time</td>
</tr>
</tbody>
</table>

Remember the Memory Wall (Patterson's Law) if not Software?
A Heliocentric CS Model needed

Triple Paradigm Dual Dichotomy Approach.
The Generalization of Software Engineering —

Instruction streams

Time to space mapping issue

Data streams

*) do not confuse with „dataflow”!

Structure

Pipe network model

Configware Engineering

Reconfigurable Data Path Unit

Reconfigurable Data Path Array

Software Engineering

Program Engineering

Flowware Engineering

CPU

SE

PE

CE

TU Kaiserslautern

© 2010, reiner@hartenstein.de

http://hartenstein.de
Triple Paradigm Compilation

Software Engineering
- source program
- software compiler
- software code
- instruction scheduler
- instruction streams

Configware Engineering
- source „program“
- mapper
- configware code
- configuration

Code-X
- mid' 90ies:
  - Jürgen Becker
- configware compiler
- data scheduler
- flowware code
- data streams

C, FORTRAN, MATHLAB, …
by triple paradigm co-education:

traditional qualification in the time domain

+ lean qualification in the space domain

= lean hardware modeling qualification
at a higher level of abstraction
We urgently need a Software Education Revolution for using Multicore and Reconfigurable Computing (SERUM-RC*)

*) Reconfigurable Computing

We urgently need a Mead- & Conway-dimension text book on triple-paradigm programming education and a few new Matlab/Simulink boxes for a model-based lean instruction approach to undergraduate students
Conclusions (2)

To maintain a Booming Multicore Era: possible for 2 or 3 more decades? Not without Reconfigurable Computing!

Since high growth rate is indispensable

Relative performance

The end of the singlecore era

Von-Neumann-only parallelism

GPGPU

x86

Side effect: massively saving energy

Year

04 08 12 16 20 24 28 30
thank you
extra pages for discussion:
The Systolic Array

(H. T. Kung paradigm)

Algebra experts' hobby, early 80ies

(introducing Data streams

1978, ...

no instruction streams needed)

nice time/space notation - defines: ...
which data item
at which time
at which port

(input data stream

output data streams)

DPA*

*) DataPath Array
(array of DPUs)

DataPath Unit has no program counter!
it's no CPU!

DataPath

The Systolic Array

© 2010, reiner@hartenstein.de

http://hartenstein.de
The von Neumann Syndrome

The instruction-stream-based von Neumann approach:

The data-stream-based anti machine approach:

has no von Neumann bottle-necks

per CPU!

has several von Neumann overhead phenomena
Data meeting the Processing Unit (PU) ...

We have 2 choices

routing the data by memory-cycle-hungry instruction streams thru shared memory

remember Pattersons law!

data-stream-based: placement* of the execution locality ...

pipe network generated by configware compilation

by Configware

by Software

*) before run time

http://hartenstein.de
EastScan is steps by [1, 0] end EastScan;

SouthScan is step by [0, 1] end SouthScan;

NorthEastScan is loop 8 times until [1, 1] step by [1, -1] endloop end

SouthWestScan is loop 8 times until [1, 1] step by [-1, 1] endloop end

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

goto PixMap[1, 1] HalfZigZag;

SouthWestScan uturn (reverse (HalfZigZag))
Double Dichotomy

Paradigm Dichotomy

von Neumann instruction stream (Software-Domain) ↔ Anti Machine data stream (Flowware-Domain)

Relativity Dichotomy

Procedure (Software-Domain) ↔ Structure (Configware-Domain)
Paradigm Dichotomy: an old hat

paradigm mapping causes a time to space mapping

decision box:

ENABLE

CONDITION

B0

B1
demultiplexor:

ENABLE

CONDITION

1

0

B0

B1

HDL scene ~1970:
decision box turns into demultiplexor

“That’s so simple! why did it take 30 years to find out ?”

reductionists’ tunnel view

C. G. Bell et al: IEEE Trans-C21/5, May 1972
RTM as DEC product available: 1973

David Parnas: Put [very] Old Ideas Into Practice

PvOIIIP

© 2010, reiner@hartenstein.de
Paradigm Dichotomy

**von Neumann**

- Instruction stream
  - (Software-Domain)

**Anti Machine**

- Data stream
  - (Flowware-Domain)

Software to flowware mapping?

Relativity Dichotomy

- Procedure
  - Time
    - (Software-Domain)

- Structure
  - Space
    - (Configware-Domain)
Relativity Dichotomy

Paradigm Dichotomy

von Neumann

Anti Machine

instruction stream
(data stream)

(Software-Domain)

(flowware-Domain)

Procedure

(time to space mapping)

(space structure)

(software-Domain)

(configware-Domain)
Relativity Dichotomy (2)

time domain: space domain

procedure domain

2 phases:
1) programming
   instruction streams
2) run time

3 phases:
1) reconfiguration of structures
2) programming data streams
3) run time
time-iterative to space-iterative

the space dimension is limited (e.g. because of the chip size)

loop transformation methodology: 70ies and later

Strip mining
[D. Loveman, J-ACM, 1977]
POIIP: Loop turns into Pipeline [1979]

(reconfigurable) DataPath Unit:

loop:

Pipeline:

CPU

Memory

loop body

complex loop body

nested loops

complex rDPU or pipe network inside rDPU

complex pipe network
The Bubble Sort Algorithm

```
loop i = 2 ... N
  loop j = 2 ... N
    if key [j-1] > key [j] then swap (key [j-1], key [j])
  endif;
  endloop j;
endloop i;
```

© 2010, reiner@hartenstein.de http://hartenstein.de
architecture instead of synchro

bubble sort example

direct time to space mapping

accessing conflicts

only half of the number of blocks

modification: with shuffle-function

"Shuffle Sort"
time 2 space mapping

Time domain: Procedure-Domain
Program loop
- n time steps, 1 CPU

Bubble Sort
- n x k time steps: 1 „conditional swap“ unit

space-Domain: Structure-Domain

Pipeline
- 1 clock steps n DPUs

Shuffle Sort
- k clock steps, n „conditional swap“ units

time-Algorithm → space-Algorithms

conditional swap
x
y

time-Algorithm → space-/time-Algorithm

http://hartenstein.de

© 2010, reiner@hartenstein.de