The Impact of Reconfigurable Computing on Manycore Programming Trends

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

Impact of the von Neumann Syndrome

- Dig more coal -- the PCs are coming
- never run out of energy?
- Power consumption by internet:
  x30 til 2030 if trends continue

Opening keynote, the 6th FPGAworld Conference, 10 Sep 2009, Stockholm, Sweden
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. Current trends will lead to an unaffordable future operation cost of our cyber infrastructure.

Outline

• 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

Outline (2)

Rapid VLSI Design Education Revolution

1980 - 1983

The incubator of the free ride on Moore’s Law

massive funding: DARPA; NSF; many national governments; European Union …

Created the missing designer population

The most effective project in the history of modern computer science

Carver Mead

Lynn Conway

The End of Moore’s Law

the end of the single-core era

traditional instruction-based computing is running out of steam

-150 Watts/chip? It’s an energy consumption issue

The End of Moore’s Law

the end of the single-core era

stop in 2005

The End of Moore’s Law

Growth beyond Moore’s Law?

the end of the single-core era

performance beyond the end of Moore’s Law?

the key issue: processor cores number of transistors doubles every 2 years

Clock speed

Gigahertz race

Traditional instruction-based computing is running out of steam

The End of Moore’s Law

Opening keynote, the 6th FPGAworld Conference, 10 Sep 2009, Stockholm, Sweden
Multimedia in the Multicore Era

ICT is at an inflection point

IKT is at an inflection point

Funding market changes

Multicore has been around for decades

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

Handheld & 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

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

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

Multicore has been around for decades

Dana/Ardent/ Stellar/Stardent
Cydrome
Culler Scientific
Culler
Cray Computer
Convex
CDC
BBN
Astronautics
Applied Dynamics
Ametek
American Supercomputer

The Power Consumption of Computing

The Silver Bullet?

The Generalisation of Software Engineering

The Multicore Scenario

The Single-Core Approach

Conclusions

Opening keynote, the 6th FPGAworld Conference, 10 Sep 2009, Stockholm, Sweden
The Power Consumption of Computing

The Silver Bullet?

The Multicore Scenario

Conclusions

The Generalisation of Software Engineering

Opening keynote, the 6th FPGAworl Conference, 10 Sep 2009, Stockholm, Sweden
In the 6th FPGAworld Conference, 10 Sep 2009, Stockholm, Sweden
Why such Speed-up Factors ...

... with FPGAs: a much worse technology!

- Massive wiring overhead
- Massive reconfigurability overhead
- Routing congestion growing with FPGA size

The "Reconfigurable Computing Paradox"

Main reason: NO von Neumann Syndrome!

More recently also: more "platform FPGAs"

RC versus Multicore

RC: speed-up often higher by orders of magnitude

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

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
### Machine Model of the PC Era

<table>
<thead>
<tr>
<th>Machine model</th>
<th>resources</th>
<th>sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP accelerator</td>
<td>hardwired</td>
<td>-</td>
</tr>
<tr>
<td>CPU</td>
<td>hardwired</td>
<td>programmable</td>
</tr>
</tbody>
</table>

*the tail is wagging the dog*  
Application-Specific Integrated Circuit & other accelerators, e.g., graphics processor

---

### From CPU to RPU (Reconfigurable Processing Unit)

<table>
<thead>
<tr>
<th>Machine model</th>
<th>resources</th>
<th>sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP accelerator</td>
<td>hardwired</td>
<td>-</td>
</tr>
<tr>
<td>CPU</td>
<td>hardwired</td>
<td>programmable</td>
</tr>
<tr>
<td>RPU accelerator</td>
<td>programmable</td>
<td>-</td>
</tr>
</tbody>
</table>

*now accelerators are programmable!"non-von-Neumann"

---

### 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

---

### 40 Years Software Crisis

- **Wirth's Law** [Niklaus Wirth]: "software is slowing faster than hardware is accelerating"
- **Nathan's Law**: Software is a gas. It expands to fill its container … until being limited by Moore’s Law & Kryder’s Law

---

### 40 Years Software Crisis

- **Patterson's Law**
- **The Memory Wall**
- **The von Neumann Syndrome**
- **Software** stands for extremely memory-cycle-hungry instruction streams

---

### Thomas S. Kuhn

- Science does not progress continuously.
- shortcomings in an established paradigm produces a crisis that may lead to a revolution
- in which the established paradigm is overthrown and replaced.

*The von Neumann paradigm?*
Thomas Kuhn is right!

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

The CPU-centric world model of Software Engineering will be replaced by removing the tunnel view perspective.
Time to Space Mapping

<table>
<thead>
<tr>
<th>Machine model</th>
<th>resources</th>
<th>sequencer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>property</td>
<td>property</td>
</tr>
<tr>
<td></td>
<td>programming</td>
<td>programming</td>
</tr>
<tr>
<td>ASIC accelerator</td>
<td>hardwired</td>
<td>-</td>
</tr>
<tr>
<td>CPU</td>
<td>hardwired</td>
<td>programmable</td>
</tr>
<tr>
<td>RPU accelerator</td>
<td>programmable</td>
<td>programmable</td>
</tr>
<tr>
<td></td>
<td>configurable code</td>
<td>programmable</td>
</tr>
<tr>
<td></td>
<td>data streams</td>
<td>data counter</td>
</tr>
</tbody>
</table>

Relativity Dichotomy

"The biggest payoff will come from putting old ideas into practice and teaching people how to apply them properly."

David Parnas

Software Education (R)evolution:

step by step, not overthrowing the SE scene

by simultaneous 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

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

We need a Software Education Revolution

2010 - ....

The incubator
of the free ride
on Cowhey’s & Aronson’s law

massive funding required

partially re-write the code

Create the missing programmer population

next most effective project in the history of modern computer science

DOS to Windows

took 10 years

RAW 2010

17th Reconfigurable Architectures Workshop
April 19-20, 2010, Atlanta (Georgia), USA


Manuscript due: October 18, 2009
Notification of acceptance: December 14, 2009
Camera-ready Papers Due: February 1, 2010

in conjunction with:

Community Building Function of the DATE Friday Workshop

Friday Workshop
Friday, March 12, 2010, 08:30 – 17:00

Software Education Revolution for using Multicore - and RC* (SERUM-RC*)
to submit, and, to join the team, contact: reiner@hartenstein.de
DATE-Conference, Dresden, DE: http://www.date-conference.com
CIP: http://fpd.org/cfp/

* Reconfigurable Computing

"How to hide the ugliness from the user" [Henmar Schmit]

"...Adoption of VHDL was one of the biggest mistakes in the history of design automation, causing users and EDA vendors to waste hundreds of millions of dollars..."

— Joe Costello, Cadence Design Systems, 1995

POiIP
Outline

- 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

Conclusions

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

We need „une’ Levée en Masses“

Thank you for your patience

END
Double Dichotomy

1) Paradigm Dichotomy

von Neumann Machine (Software-Domain)

Datastream Machine (Flowware-Domain)

2) Relativity Dichotomy

-time: -Procedure (Software-Domain)

-space: -Structure (Configware-Domain)

Relativity Dichotomy

2 phases:
1) programming
2) run time

time domain: procedure domain

space domain: structure domain

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

Time to Space Mapping

\text{time to space mapping}

\begin{align*}
\text{time domain:} & \quad \text{structure domain} \\
\text{space domain:} & \quad \text{space domain}
\end{align*}

program loop

pipeline

-Bubble Sort

-\text{Strip mining (D. Loveman, JACM, 1977)}

-\text{Loop transformation method: 70ies and later}

-\text{e.g., example: bubble sort migration}

Opening keynote, the 6th FPGAWorld Conference, 10 Sep 2009, Stockholm, Sweden
Flowware means parallelism resulting from time to space migration.

Flowware: scheduling data streams - from a generalization of the systolic arrays supports any wild free form of pipe networks: spiral, zigzag, fork and join, and even more wild, unidirectional and fully or partially bidirectional, Fifos, stacks, registers, register files, RAM blocks...

Acceleration Mechanisms

- parallelism by multi bank memory architecture
- auxiliary hardware for address calculation
- address calculation before run time
- avoiding multiple accesses to the same data
- avoiding memory cycles for address computation
- optimization by storage scheme transformations
- optimization by memory architecture transformations

New boundary constraints are the limiting factor

Legacy scientific applications: predominantly sequential

The entire software ecosystem will need to evolve (including curricula): O/S, libraries, software development environments, compilers and languages additional levels of parallelism: chaining, pipelining, systolic, super-systolic, wavefront arrays additional data structures and storage organization: the new distributed memory discipline
**old Paradigms and Methodologies**

- 1946: Machine Paradigm (von Neumann)
- 1980: Datastreams (Kung, Leiserson)
- 1989: Anti Machine** (TU-KL)
- 1990: first rDPA (Robson)
- 1994: higher Anti Machine** Programming Language (Flowware: TU-KL)
- 1995: super systolic array: rDPA (Kress)
- 1996+: SCCC (LANL), SCORE, ASPRC, Bee (UCB),...
- 1997+: Discipline of Distributed Memory Architectures (IMEC…)
- 1997: first automatically partitioning Configware/Software Co-Compiler (TU-KL)

Note: ** indicates a stream machine.

* rDPA = reconfigurable Data Path Array

---

**Teaching computing fundamentals**

Ignoring Reconfigurable Computing in teaching computing fundamentals within our CS curricula is one of the biggest mistakes in the history of information technology application causing to waste billions of dollars.