Clipping from this presentation:

The Tunnel Vision Syndrome: Massively Delaying Progress

Generalization of the Systolic Array

Outline

- Preface
- History of Computing
- The Systolic Array
- The Kress Array
- The Xputer Paradigm
- The Twin Paradigm Approach
- More Tunnel Vision
- We must Reinvent Computing
- Conclusions
M. J. Foster and H. T. Kung: The Design of Special-Purpose VLSI Chips...

IEEE 7th ISCA, La Baule, France, May 6-8, 1980

The first prominent step back toward data-streams

Just only Special purpose VLSI chips? (Tunnel Vision)

Karl Steinbuch

M. J. Foster and H. T. Kung: "The Design of Special-Purpose VLSI Chips..."

why not general purpose?

It is not sufficient to invent something. You need to recognize, that you have invented something.

http://www.fpl.uni-kl.de/papers/publications/karl-steinbuch.html
What Synthesis Method?
of course algebraic! (linear projection)

( Mathematics Point of View )
supports only applications with
strictly regular data dependencies

1995: 
Rainer Kress replaced it by *simulated annealing*: 
supports also any irregular & wild form pipe networks

The super-systolic array: a 
generalization of the systolic array

© 2012, reiner@hartenstein.de

The Data Streams?

Who generates* the datastreams?

*) and receives

© 2012, reiner@hartenstein.de

Featured Invited Talk; The 24th IEEE International Conference
on Application-specific Systems, Architectures and Processors
(ASAP 2013), 5-7 June 2013, Washington, DC, USA
It’s not our job

Who generates the data streams?

Another Tunnel Vision Symptom

without a sequencer: missed to invent a new machine paradigm

Who generates the data streams?

The data stream computing model

The anti-machine paradigm* (Xputer): data counters(s) instead of a programm counter

the straight-forward model of reconfigurable computing, since no instruction streams

*) publ. 1989

http://xputer.de/

http://data-streams.org/
**Duality of procedural Languages**

<table>
<thead>
<tr>
<th>Software Languages</th>
<th>Flowware Languages</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> internally parallel loops</td>
<td><strong>yes:</strong> internally parallel loops</td>
</tr>
</tbody>
</table>

But there is an Asymmetry

© 2012, reiner@hartenstein.de

---

**Heterogeneous Taxonomy**

The extension into this huge design space is mainly ignored by curricula and even by most R&D scenes

© 2012, reiner@hartenstein.de
The tunnel vision of the pre-manycore age

Conclusions

We need by orders of magnitude more parallelism and power-efficiency

A reductionist attitude of most R&D areas massively delays the solution of urgent problems

Disruptive research is urgently required and fundamental issues have to be revisited

We must disruptively reivent CS education

We must overcome the von Neumann syndrome and the widespread Tunnel Vision dementia