Karl Steinbuch about how to invent something

Reiner Hartenstein, TU-Kaiserslautern and ITIV

If you invent something, you may suffer from the Tunnel Vision Syndrome. Karl Steinbuch’s point of view has been illustrated during my featured invited presentation at ASAP 2013 on George Washington University, at Washington, DC, USA.

The colleagues who introduced the Systolic Array 1980 at ISCA in La Baule, France, suffered from this syndrome. They did not at all recognize, that they had re-invented a very important old machine paradigm. They recognized only having invented a very special design method for very special purpose VLSI chips.
Systolic Arrays

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

changed R&D of my group

Systolic Arrays (2)

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

why not general purpose?

Just only Special purpose VLSI chips? (Tunnel Vision)

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

Karl Steinbuch

http://www.fpl.uni-kl.de/papers/publications/karl-steinbuch.html
What synthesis method? Because H. T. Kung is a mathematician: of course algebraic by using linear projections! This is its very strong disadvantage, since it just supports only applications with strictly regular data dependencies – which is only a very small minority of VLSI microchip applications.

That time my student, Rainer Kress proposed to use simulated annealing instead, massively improving systolic array generalization by also supporting all very strange applications with wildly irregular data dependencies. Tunnel vision went even further: “generating the data streams is not our job!” (an EE student with a soldering iron will connect it to some signal source). Kung’s team has not been aware missing to re-invent this very important old machine paradigm (data-stream-based).
The Data Streams?

Who generates the datastreams?

*) and receives

“IT’s not our job”

Who generates the datastreams?

Another Tunnel Vision Symptom

without a sequencer: missed to invent a new machine paradigm

Winner of the "Not My Job" Award - ADOT
Litchfield Park, AZ 85
The Data Stream-based Machine Paradigm.

Being a pipeline network the systolic array operation is data-stream-based. The first important electrical computer was data-stream-based. In 1884 its prototype was ready for mass production: the Hollerith Tabulator. Following the scientific revolution in the mid 40ies this paradigm has been replaced by the instruction-stream-based von Neumann machine paradigm.

Discarding datastream-based computing by adoption of the von Neumann paradigm much more than half a century ago was the biggest mistake in the history of computing, meanwhile having caused users and vendors altogether to waste quadrillions of dollars.

Meanwhile it has been predicted, that – if the trends will not change - in about a decade the electricity bill for running all computers and ICT infrastructures worldwide will become unaffordable because von Neumann Computers are extremely inefficient in using electric power. Returning back to data-stream-based computing is absolutely unavoidable.

Computers are facing a seismic shift since vanquishing von Neumann will be extremely difficult, since more than a half century of software sits squarely on top. We need almost millions of programmers to re-implement this legacy software.
Who generates the data streams?

http://xputer.de/

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

© 2012, reiner@hartenstein.de

What Parallelism?

why such massive speed-up?

datastream parallelism: no von Neumann bottleneck

Look at the Reconfigurable Computing Paradox

instead of:

instruction stream parallelism:

many von Neumann bottlenecks

© 2012, reiner@hartenstein.de
A quantum leap in computing efficiency can be unlocked by "soft hardware" (FPGAs etc.), i.e. by Reconfigurable Computing (RC). Satnam Singh calls it “computing without processors”. FPGAs are also very good platforms to implement systolic arrays. The FPGA came to market in 1984 by Xilinx, exactly one century after the Hollerith tabulator prototype was ready for mass production. By migration of applications from von-Neumann-based implementation massive speed-up factors have been obtained: mostly by several orders of magnitude, like, for instance by a factor of 28.500 as published by Tarek El-Ghazawi.
The Reconfigurable Computing Paradox

These very high speed-up factors are a surprise, since because of reconfigurability overhead an FPGA needs up to about 50 times more transistors than are required for execution by a microprocessor. From this point of view the FPGA technology is worse by several orders of magnitude.

The von Neumann Syndrome

What is the explanation of this highly surprising paradox? It is the unbelievably extreme inefficiency of the von Neumann paradigm, so that the paradigm shift produces such a benefit even by using a drastically more inefficient technology. For the wide variety of performance and organization problems with the memory hierarchy Prof. C. V. Ramamoorthy (UC Berkeley) has coined the term “von Neumann Syndrome”. 
massive speed-up (the RC paradox)