Challenges and Opportunities in the Design of Future Systems on Chip

Tobias G. Noll
RWTH Aachen University

With the continuous progress of CMOS technology the integration of complete, very complex systems on a single die of silicon becomes feasible, featuring a huge computational power by operating billions of transistors at GHz-clock frequencies. But aside of the well known advantages of such systems-on-chip (SoCs) their exploding gate count complexity, the physical complexity of future deep-sub-micron technologies as well as the ever decreasing time-to-market window are facing future SoC designers by serious, today still unsolved issues. Actually the design of competitive SoCs enforces fundamental changes in system conception and design methodology.

The extensive use of SW programmable kernels in such SoCs offers the highest available flexibility but, although their performance continuously increases, suffers from a too small computational performance for many applications and even more severe, is paid for by an unacceptable power penalty. Adding high performance co-processor blocks dedicated to standard functionalities (like filtering, error decoding etc. in frequently used DSP processing), featuring orders of magnitude better power efficiency and applying an optimized HW/SW split significantly reduces programmable kernel load, releasing flexible resources for added value applications. An additional implementation alternative becoming more and more attractive is the use of application domain specific re-configurable FPGA-like building blocks.

This leads to heterogeneous System-on-Chip architectures, where the most important challenges are to find the ideal HW/SW and analog / digital partitioning for minimizing system costs as silicon area, power dissipation, design effort, testability, etc.. Keys to the success of this approach are 1) an early interaction between system level requirements and implementation issues, and 2) short development times. Parameterized, fairly accurate cost models here enable the evaluation of the high-dimensional design space with fast iteration cycles on algorithmic system level and are the basis for tradeoffs between system requirements and implementation cost already in an early phase of the system development.

From the on-chip communications point of view also power dissipation forces to change from today's mostly memory centric to interconnect centric architectures. Again only a quantitative cost and performance evaluation of alternatives like point-to-point, bus or even network-on-chip based alternatives will enable the design of competitive products. But appropriate cost models for the underlying infrastructure like interfaces, switches and routers are not available yet.

Even more challenging, because of decreasing reliability, future SoCs will require the use of fault tolerant architectures and circuits to cope with "static" fabrication defect or fluctuation and aging problems as well as with "dynamic" soft error, noise, and worst case timing problems. As power and area efficient strategies to implement fault tolerance today are available only for memory and FPGA-like blocks, these are missing for combinational and especially sequential logic blocks and this may lead to the most serious challenge in future SoC design.

Challenges and Opportunities in the Design of Future Systems-on-Chip

Tobias G. Noll

Chair of Electrical Engineering and Computer Systems
RWTH Aachen University - Germany

tgn@eecs.rwth-aachen.de • www.eecs.rwth-aachen.de

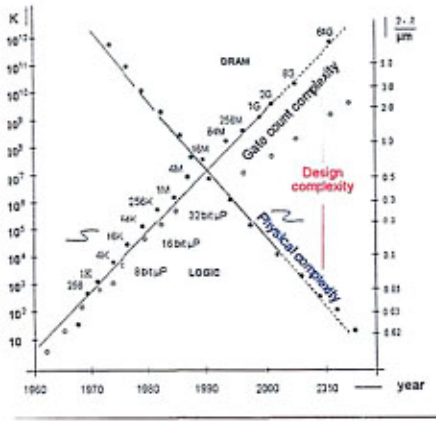
EECS RWTH

Outline

- Introduction
- Design of SoC kernels
- Partitioning / mapping methodologies
- Cost Modeling of SoC kernels
- On-SoC communication issues
- Fault tolerance
- Outlook: SoCs vs SiPs

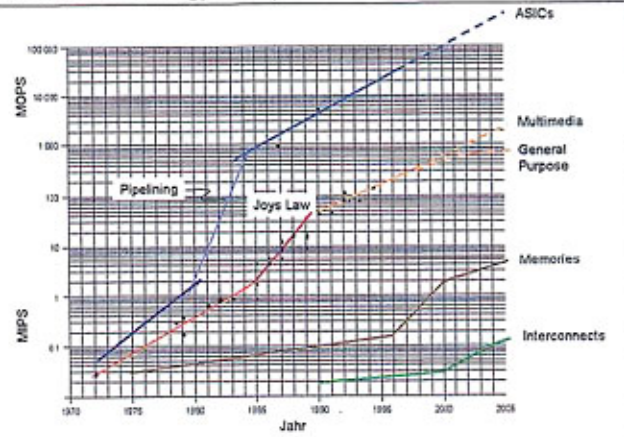
EECS RWTH

Trend of Transistor Complexity K and Smallest Feature Size $L = 2\lambda$



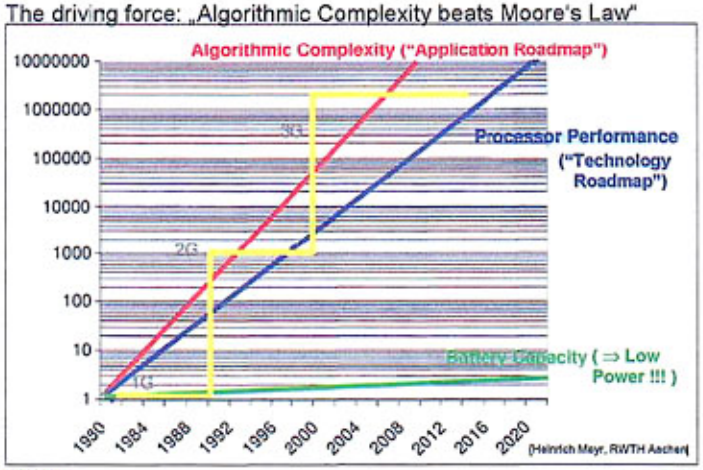
EECS RWTH

Trends in VLSI-Technology: Computational Power



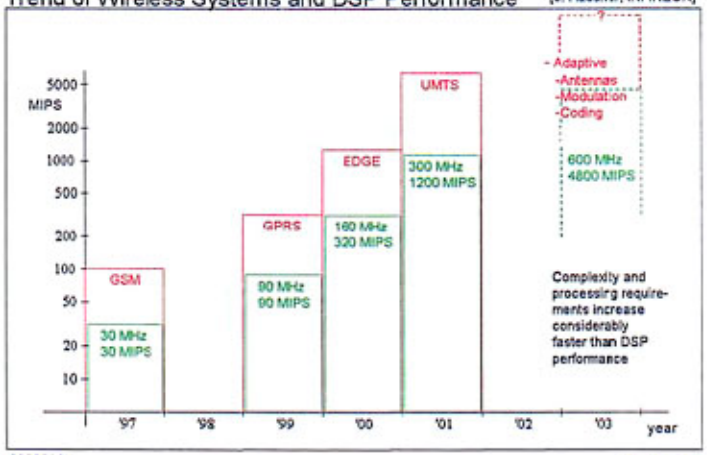
EECS RWTH

The driving force: „Algorithmic Complexity beats Moore's Law"



Trend of Wireless Systems and DSP Performance

[J. Hausner, INFINEON]

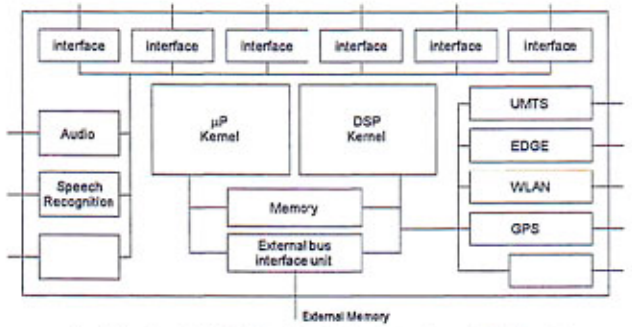


Example of UMTS receiver processing requirements @384 kbps:

Digital Filtering (RRC, channelization)	~3600 MIPS ⇨ HW
Searcher (frame, slot, delay path estimation)	~1500 MIPS ⇨ HW
RAKE receiver	~650 MIPS ⇨ HW
Maximum-ratio combining (MRC)	~24 MIPS ⇨ HW
Channel estimation	~12 MIPS
AGC, AFC	~10 MIPS
Deinterleaving, rate matching	~14 MIPS ⇨ HW
Turbo-decoding	~52 MIPS
Total	**~5860 MIPS**

[J. Hausner, INFINEON]

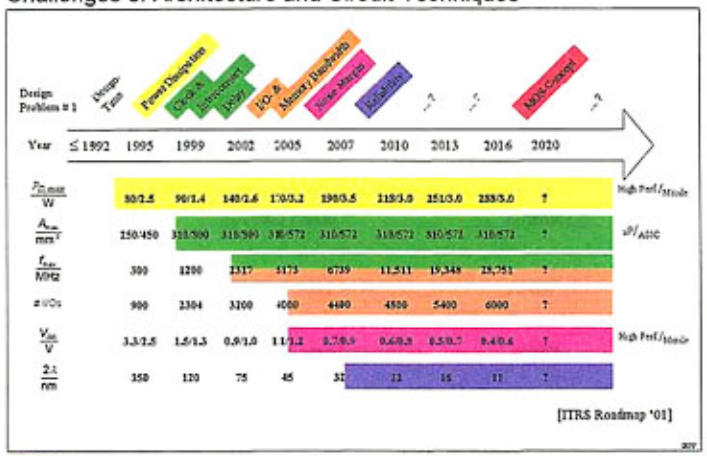
UMTS Solution



⇨ Optimized HW/FW Split significantly reduces DSP load, releasing DSP for added value applications

[J. Hausner, INFINEON]

Challenges of Architecture and Circuit-Techniques



Power Dissipation

- Power dissipation will limit further increase of the product of gate count complexity and clock frequency
- Today the major (dynamic) part of power is burned due to charging and discharging of (interconnect) capacitances
- A vastly increasing (static) part of power dissipation is due to leakage and tunneling currents in the devices

Power is and will be a Primary Design Problem

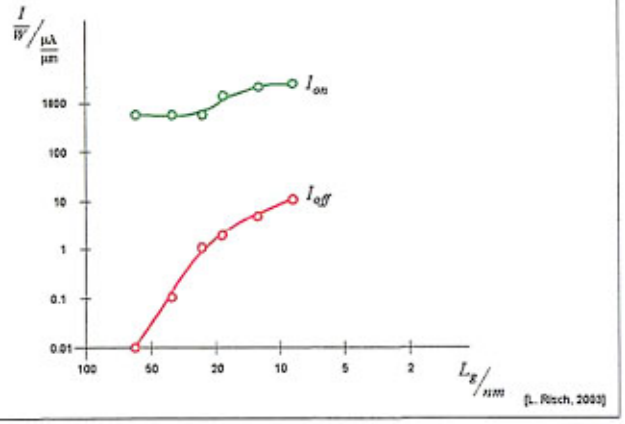
DSP Arithmetic
- Preservation of locality
- Clustered supply voltage
- Optimized device sizing
- ...

$$P = \sigma \cdot f \cdot V_{dd}^2 \cdot C + \underbrace{\sigma \cdot f \cdot V_{dd} \cdot Q_{SC}}_{\leq 10\%} + \underbrace{V_{dd} \cdot I_{eff}}_{\text{in the past}}$$

Communication Systems with long idle times
- Clock gating
- MTCMOS
- Stacked transistors
- "Random modulation"
- ...

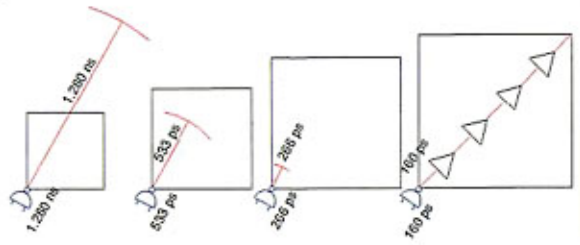
in the past (increases dramatically due to deep sub-µm effects)

Increase of Leakage Current



Trend of One-Gate Delay Wiring Distance

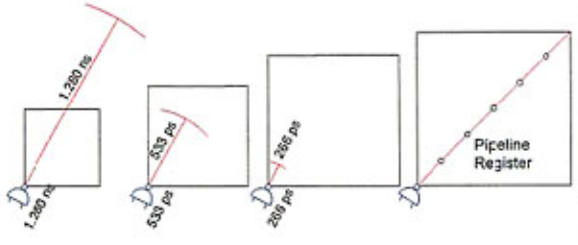
Year	1986 (1.2 um)	1991 (0.6 um)	1996 (0.3 um)	2001 (0.18um)
Silicon Area	10 x 10 mm²	13 x 13 mm²	17 x 17 mm²	20 x 20 mm²
$l_{max,\ Gate\ Delay}$	24.5 mm	10 mm	3.16 mm	1.47 mm



EECS RWTH

Trend of One-Gate Delay Wiring Distance

Year	1986 (1.2 um)	1991 (0.6 um)	1996 (0.3 um)	2001 (0.18um)
Silicon Area	10 x 10 mm²	13 x 13 mm²	17 x 17 mm²	20 x 20 mm²
$l_{max,\ Gate\ Delay}$	24.5 mm	10 mm	3.16 mm	1.47 mm



Long interconnections become extremely expensive in terms of power dissipation

EECS RWTH

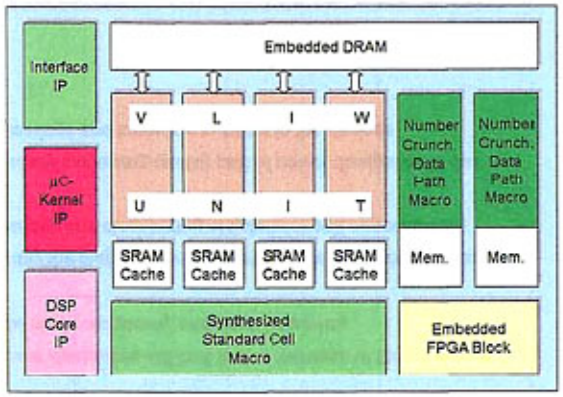
Coping with Gate Count Complexity

• '60 ... '80	Transistor Level	GDSII Layout
• '80 ... '90	Register Transfer Level	VHDL
• '90 ... '00	Behavioural Level	VHDL
• '00 ...	IP Core Level	System C

⇒ IP-based, Platform-based system-on-chip design

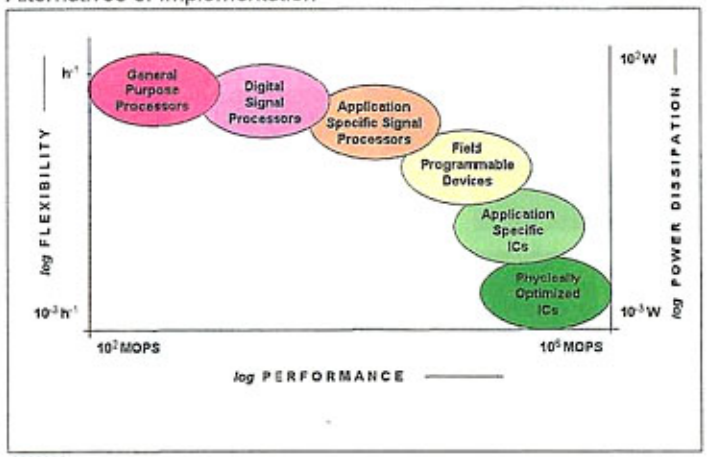
EECS RWTH

Heterogenous Target Architecture

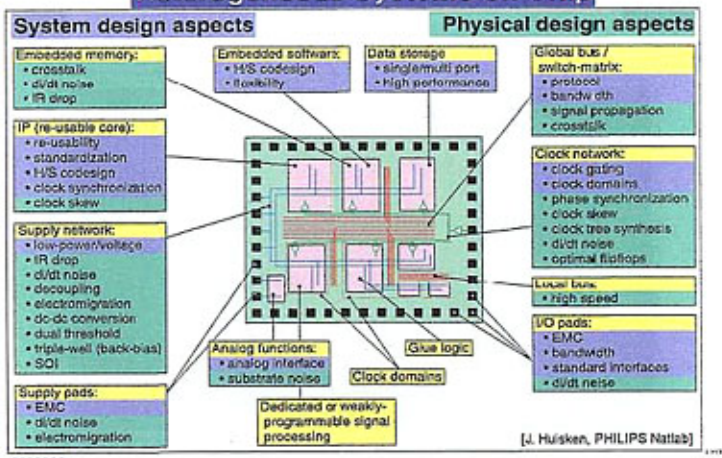


EECS RWTH

Alternatives of Implementation



Heterogeneous Systems-on-Chip

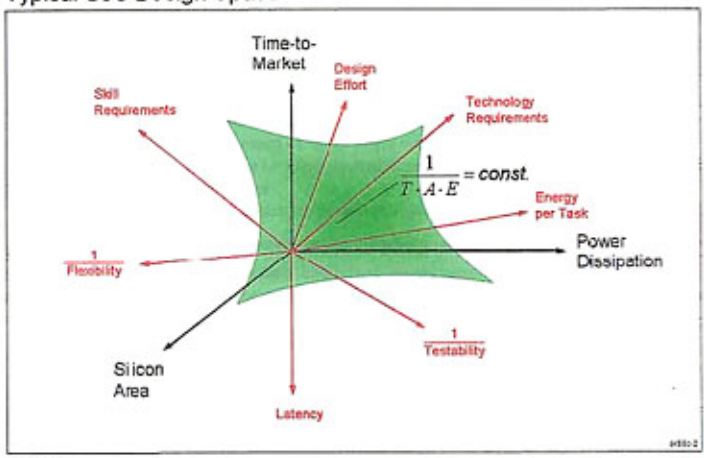


Major Challenges in SoC Design

Initial idea: Integrating all components for "complete" system functionality into a single chip with heterogeneous architecture

- Ever increasing application demand for computational power
  ⇒ exploding gate count and clock frequency
  ⇒ increasing power cissipation
- Decreasing time-to-market window due to shrinking product cycles
  ⇒ increasing demand for flexibility
- Deep-sub-micron physical complexity (interconnect delay, signal integrity, ...)
- Increasing design anc especially mask costs
- Need for fault tolerance (decreasing reliability, ...)
- Specification, Parametrization, Interfaces, ...

Typical SoC Design Space



Outline

- ...
- Design of SoC kernels
  - Automated, physically optimized hard core design
  - ...
  - ...
- ...

Datapath Generator: Basic Ideas

DSP Algorithms are iterative in time and functionality

⇒ DSP Datapaths are commonly **regular** what can be exploited to reduce design effort

Split into < **synthesized** non-iterative control & glue logic, and
DPG generatable time-, area- and power-critical datapaths

⇨ Heteregenous Design Methodology

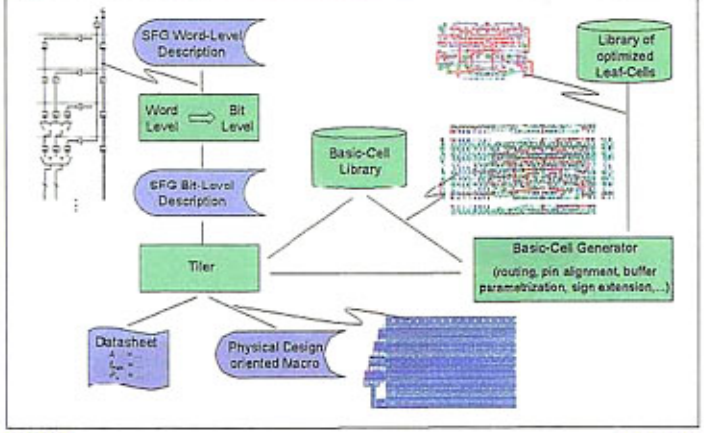
DSP Algorithms are based on only a few types of simple basic functions:
addition, delay, compare, multiplexing, simple boolean bit manipulations

⇒ DSP Datapaths **basic functions** can be **highly optimized** at low design effort

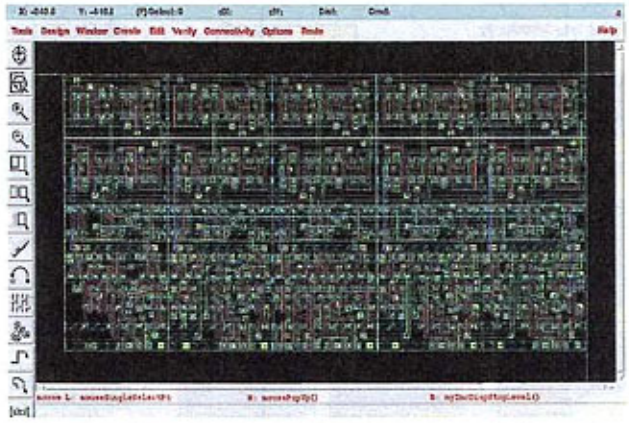
DSP Algorithms inherently contain a high degree of **locality** or can be localized by transformations. Locality allows:
Short interconnects i.e. small interconnect capacitances,
minimum sized devices i.e. small device apacitances,
short and simple critical pathes i.e. small glitching activity

⇒ Preservation of **locality ensures** high throughput, small silicon area, and even more important low power dissipation

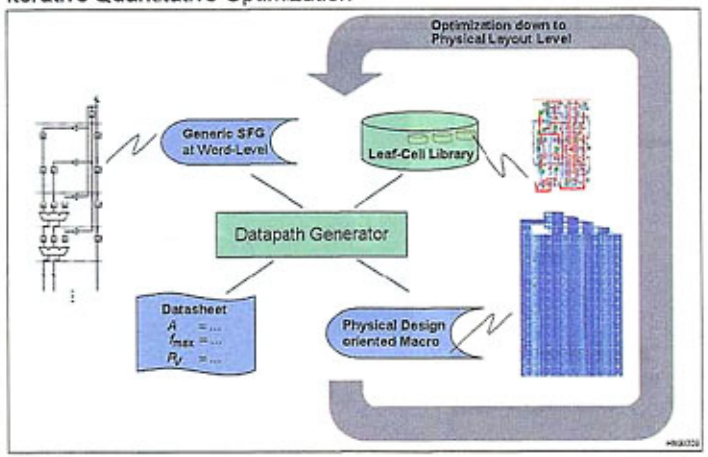
Datapath Generator



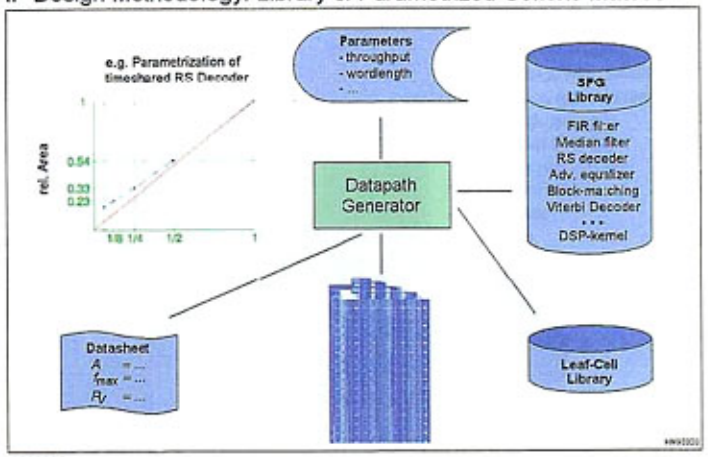
DPG-based Macro Generation



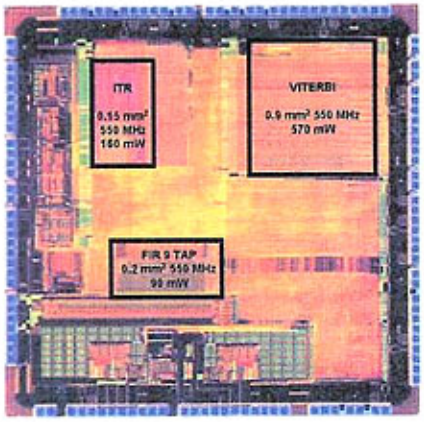
Iterative Quantitative Optimization



IP Design Methodology: Library of Parametrized Generic Macros



500 Mbit/s Read/Write Channel for Hard Disk Applications



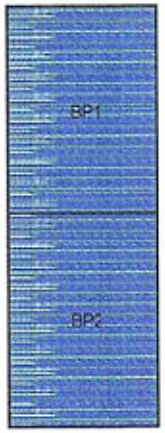
Infineon technologies

1999 Hard Disk RW-Channel

Features:

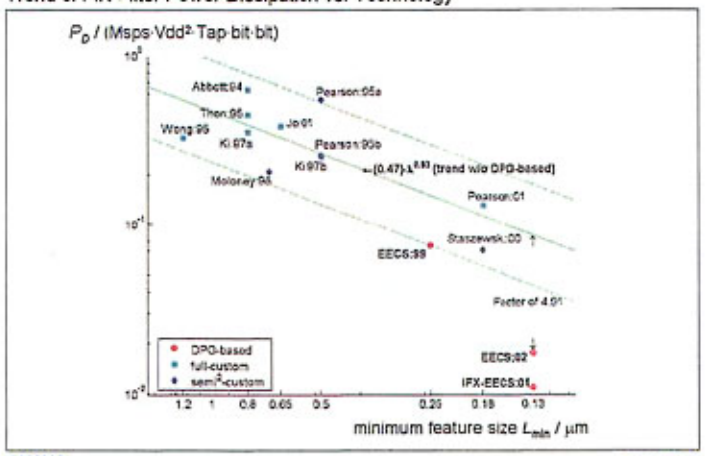
- 48/51 EPRML
- 500 Mbit/s Maximum Transfer Rate
- Advanced 0.25 µm CMOS Technology
- 10mm² Chip Area
- Digital Prefered Architecture
- Ready for System-on-Chip (SOC)

Modified Bitplane Filter

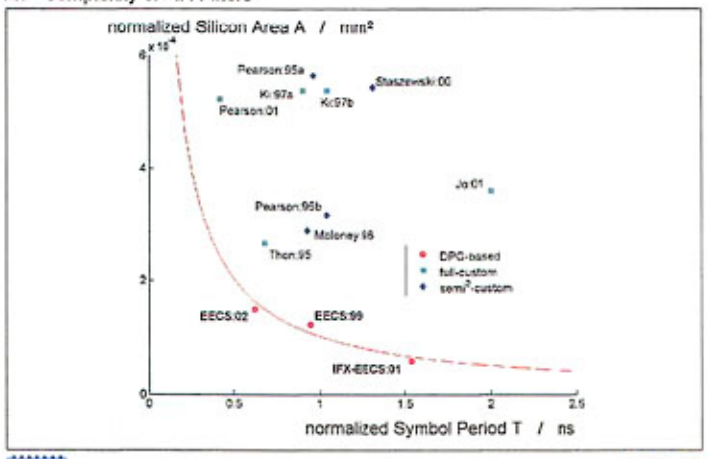


taps	10
w_{sample} / w_{coeff}	10 bit / 8 bit
technology	0.25-µm 4-LM CMOS
silicon area	0.09 mm²
throughput rate	450 MHz @ 2.25V
power dissipation	60mW (σ_{coeff}=0.25)

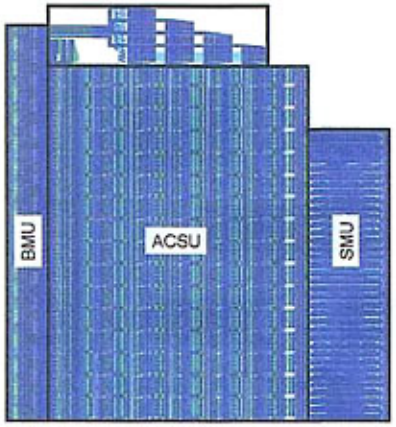
Trend of FIR Filter Power Dissipation vs. Technology



AT - complexity of FIR Filters



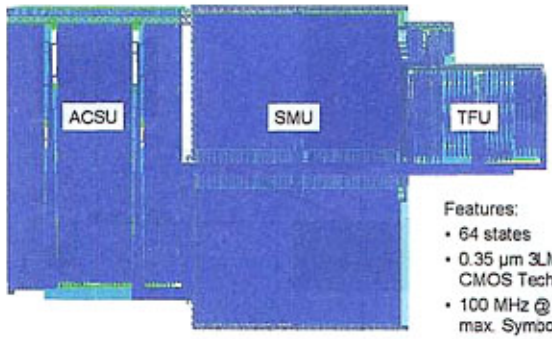
VLSI Implementation of Viterbi Macro for RW-Channel



Pipelined Carry-Save, Radix-4, 2-fold time-shared Add-Compare-Select Unit

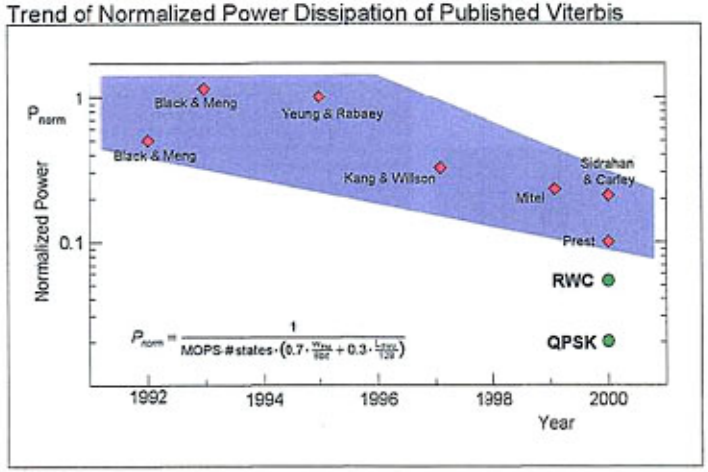
Features:
- 16 States
- 0.25 µm 4LM CMOS Technology
- 550 MHz @ 2.25V max. Symbol Rate
- 0.92 mm² Silicon Area
- 460 mW Power Dissipation
- 3 MM Design Effort

VLSI Implementation of Viterbi Macro for DVB QPSK-Decoder

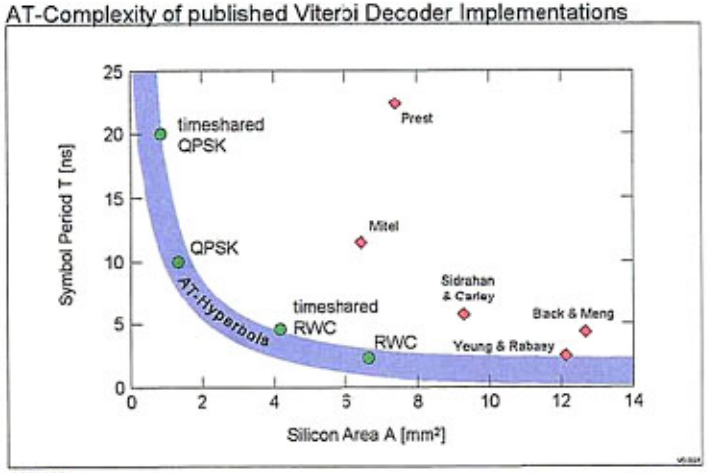


Features:
- 64 states
- 0.35 µm 3LM CMOS Technology
- 100 MHz @ 3.0V max. Symbol Rate
- 1.35 mm² Silicon Area
- 130 mW Power Dissipation
- 6 MM Design Effort

Trend of Normalized Power Dissipation of Published Viterbis



AT-Complexity of published Viterbi Decoder Implementations

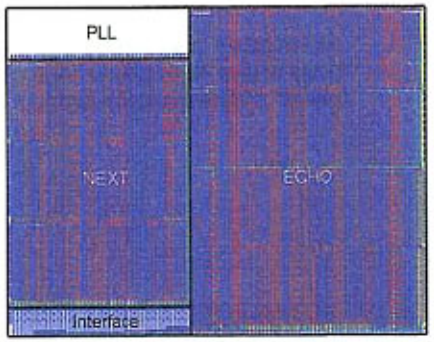


Techno cgy Scaled Features

Design	0.35µm @ 3.0V	0.25µm @ 2.25V	0.18µm @ 1.6V	0.12µm @ 1.3V
Timeshared Low Power	50MHz, 70mW	70MHz, 40mW	100MHz, 20mW	150MHz, 11mW
Low Power	100MHz, 130mW	140MHz, 70mW	200MHz, 35mW	300MHz, 20mW
Timeshared High Performance	200MHz, 450mW	280MHz, 250mW	400MHz, 130mW	600MHz, 80mW
High Performance	400MHz, 820mW	550MHz, 460mW	800MHz, 230mW	1,2GHz, 150mW

[Realized Viterbi Decoder Implementations]

Gigabit Ethernet Echo / Next Canceller Macro



- Echo Block:
  120 Taps x 4 Dimensions
  6...10 bit Coefficients
- Next Block:
  3 x 40 Taps x 4 Dimensions
  6 bit Coefficients
- Input Data: Quinary Symbols
- 1.654 Mio. Xtrs
- 0.18 um TSMC CMOS
- Silicon Area 4mm²
- Clock Frequency 125MHz
- P = 230mW@ 1.8V

Similar Approach: „Direct Mapping of Standard Cells"

A Design Environment for High Throughput, Low Power Dedicated Signal Processing Systems

W. Rhett Davis, Ning Zhang, Kevin Camera, Fred Chen, Dejan Markovic, Nathan Chan, Borivoje Nikolic, Robert W. Brodersen

University of California at Berkeley
Berkeley, California, USA

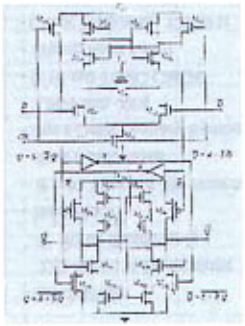
Recent studies of improved design methodologies stress ASIC designer control over physical structure (4) and early floorplanning (5). To achieve the full benefit of direct-mapping, however, control of the physical structure and early floorplanning is needed by system designers as well. To accomplish this, a new kind of design environment is needed which offers fast, automatic generation of physical information from a system-level description. This contrasts the majority of recent work, such as (3), which focuses on automatic generation of physical information from register-transfer level (RTL) descriptions.

RWTH

Today's High Performance Semi-Custom Design

Achieving 550 MHz in an ASIC Methocology

D. G. Chinnery, B. Nikolic, K. Keutzer
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
{chinnery, bora, keutzer}@eecs.berkeley.edu



"Semi-custom designs" already supplemented by device sizing, direct placement, additions to cell libraries, ...

RWTH

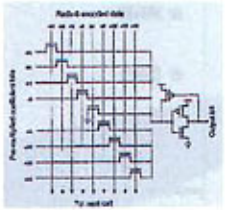
Today's High Performance Semi-Custom Design (cont'd)

A 550-MSample/s 8-Tap FIR Digital Filter for Magnetic Recording Read Channels

Robert Hegdon Strzelecki, Khurram Muhammad, and Piras Ruhani

III. Standard Cell Design

This FIR filter is an integral part of a larger read channel device and we decided to follow, as with all other digital blocks, a commercial CMOS standard cell [14] digital flow methodology. The design was modeled using a VHDL description code for system simulations (described in [13]), was synthesized from a *register transfer level* RTL description subset of VHDL and then auto-placed and auto-routed with timing-driven constraints, together with other high-speed digital back-end blocks, such as an LMS coefficients adaptation block, phase and gain detectors, etc. A few standard cells were created specifically for this design and made part of the tactical cell library: the 9-way multiplexer cell of Fig. 6 and its variants, and the low-power latch. The intended floor-plan forced on the auto-place tool, with some help of cell grouping regions, is shown in Fig. 7.



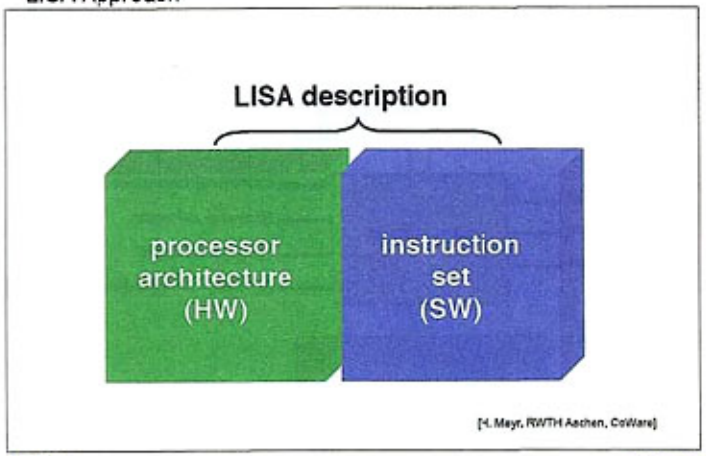
RWTH

Outline

- ...
- Design of SoC kernels
  - ...
  - Application specific instruction processor kernel design
  - ...
- ...

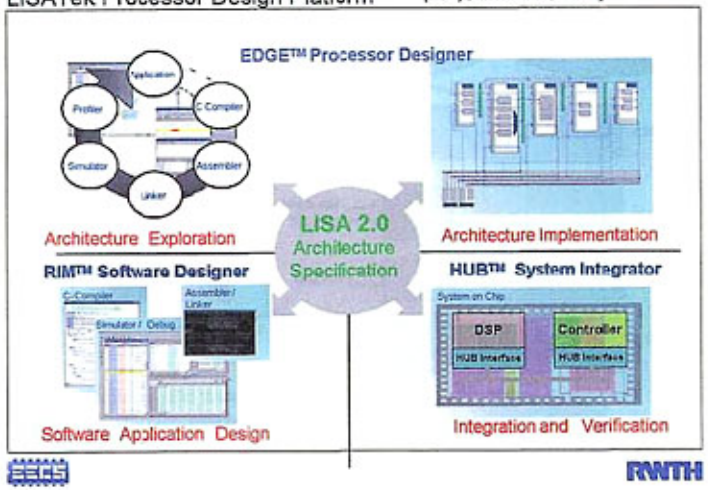
RWTH

LISA Approach



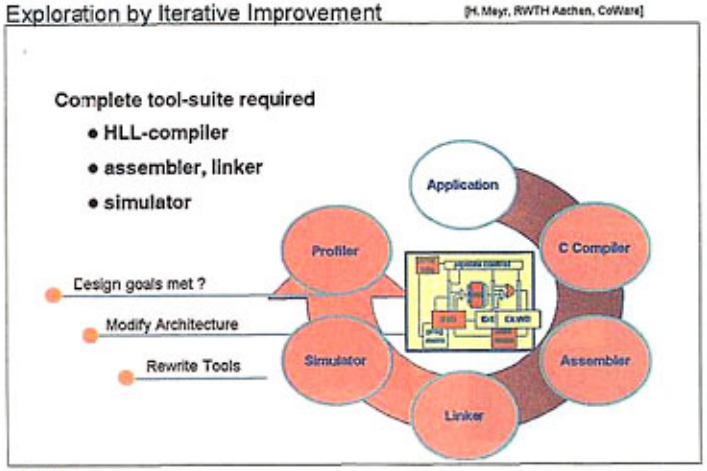
LISATek Processor Design Platform

[H. Meyr, RWTH Aachen, CoWare]



Exploration by Iterative Improvement

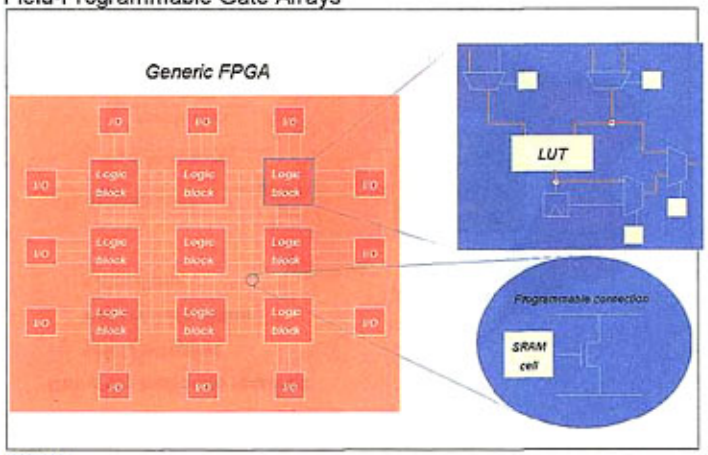
[H. Meyr, RWTH Aachen, CoWare]



Outline

- ...
- Design of SoC kernels
  - ...
  - ...
  - Potential for application domain specific embedded FPGAs
- ...

Field-Programmable Gate Arrays



eFPGA Optimization for Platform SoCs

Goal:
Optimizing an FPGA-architecture for dedicated applications without restricting generality

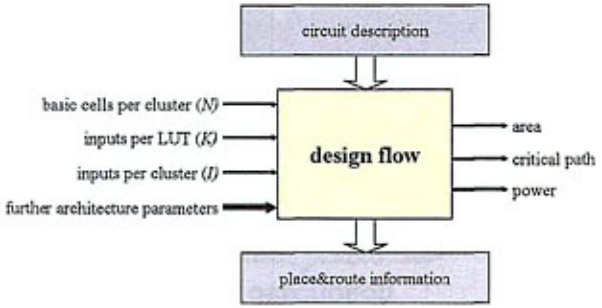
"Make the frequent case fast"

Application (domain) specific embedded FPGAs

New design sub-space

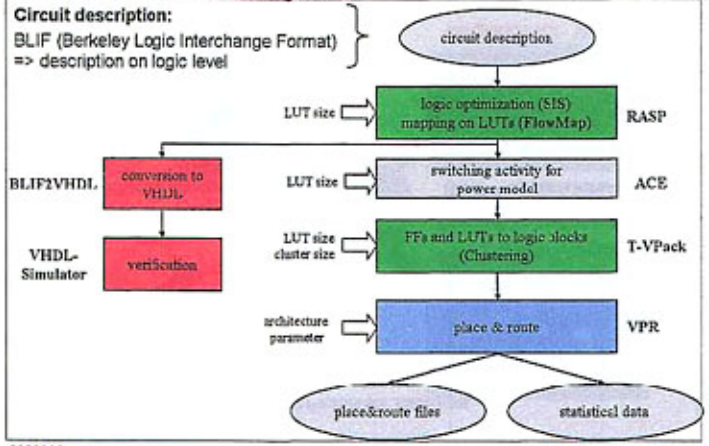
Parametrizable FPGA-design flow

Idea: Variation of the architecture for optimizing the FPGA



Parametrizable FPGA Design Flow

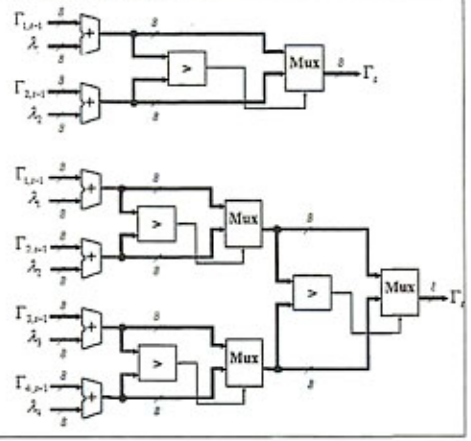
Circuit description:
BLIF (Berkeley Logic Interchange Format)
=> description on logic level



Implementation of an ACSU: Bit parallel

- Radix-2
- 8 bit parallel

- Radix-4
- 8 bit parallel
- here:
  3 comparators
- also possible:
  6 comparators

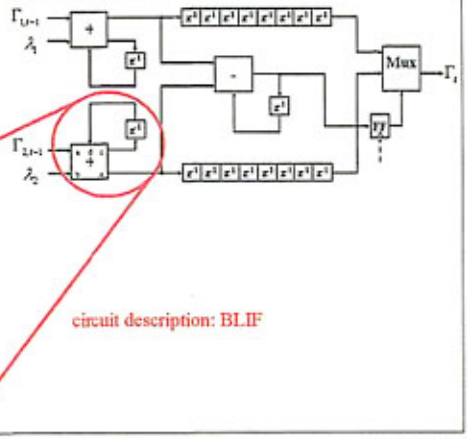


Implementation of an ACSU: Bit serial

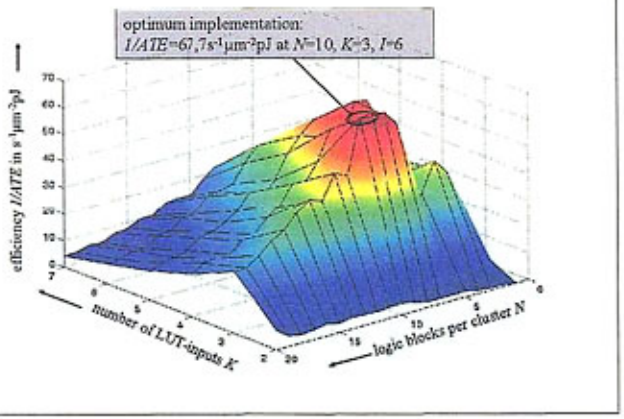
- radix-2
- 8 bit serial
- here: LSB-first, carry-ripple

```
.model fulladder
.inputs a b clk
.outputs s

.names a b d s  # Sum
001 1
010 1
100 1
111 1
.names a b d c  # Carry
011 1
101 1
110 1
111 1

.latch c d re clk 0

.end
```

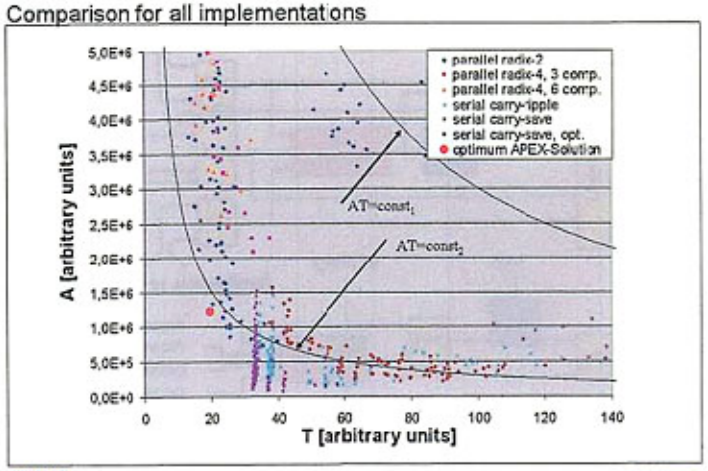
circuit description: BLIF



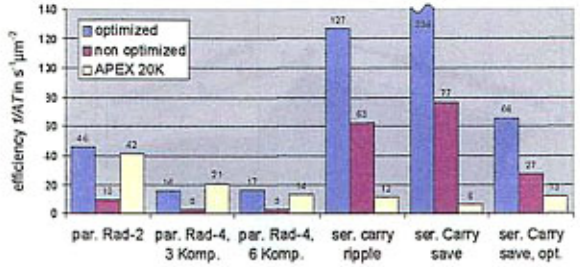
Efficiency *1/ATE* for a bit serial radix-2 ACSU



Comparison for all implementations



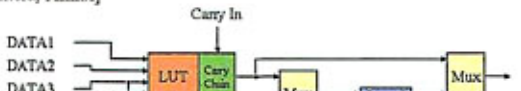
Comparison with APEX-based implementations



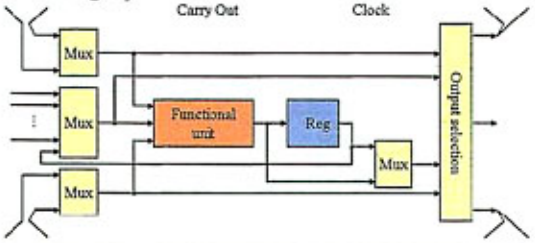
- using the same configuration, APEX-based implementations for parallel ACSU are more efficient
- optimized implementation is always at least as efficient as APEX-based implementation
- increase of efficiency by a factor of ~5 through optimization

FPGA Logic Cell Architectures

LUT-based [Altera, Xilinx]

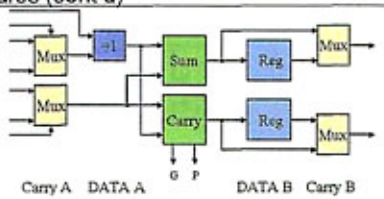


Triptych [Univ. of Washington]

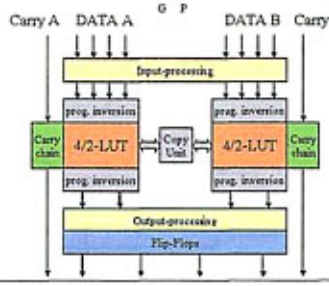


FPGA Logic Cell Architectures (cont'd)

RHW
(Reconfigurable HardWare)
[NEC 2001]



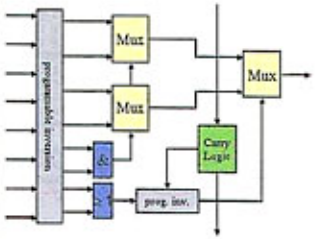
LUT-based architecture
[Philips/Natlab 2002]



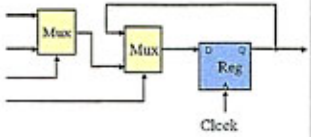
FPGA Logic Cell Architectures (cont'd)

Actel Axcelerator

C-cell (combinatorial):



R-cell (sequential):



- SuperCluster uses 4 „C-cells" and 2 „R-cells"
- any 3-input boolean function

Outline

- ...
- ...
- Partitioning / mapping methodologies
- ...

Partitioning / Mapping

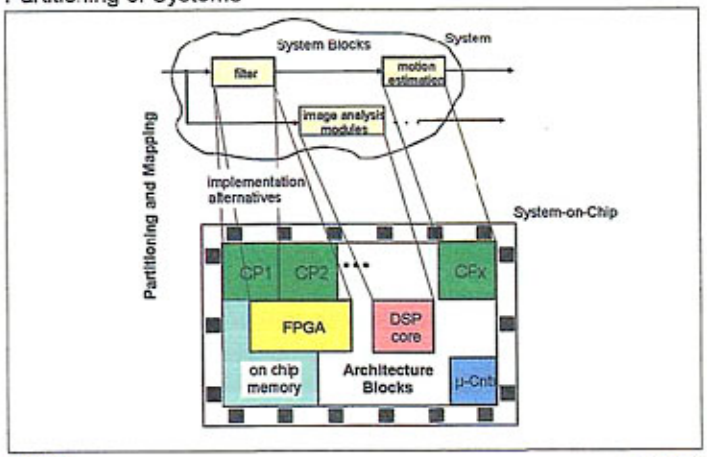
- HW / SW
- Digital / Analog
- ...

Today done manually and relying on designers experience

⇒ Has to be quantitatively optimized !

⇒ Need for fairly accurate parametrized cost models

Partitioning of Systems



Cost Functions and Scaling Rules

$$cost = A \cdot T \cdot E_{per\ sample}$$

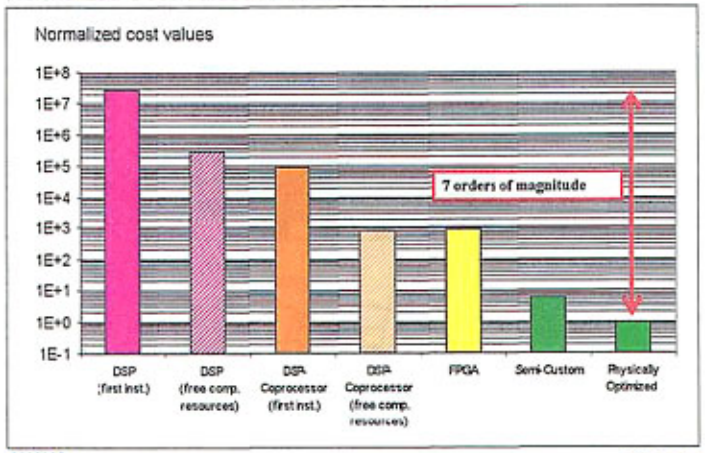
Technology dependencies to be eliminated by scaling ...

Could also use ...

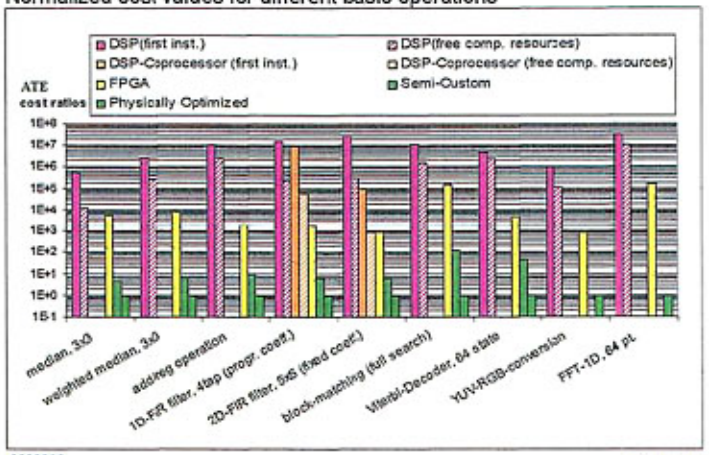
Energy Efficiency: MOPS/mW = OPs/nJ

Area Efficiency: MOPS/mm²

Normalized Cost Values for a 2D-FIR Filter



Normalized cost values for different basic operations



Flexibility has a Price Tag

SW solutions on programmable processor kernels are extremely power consumptive, because of

- Time-shared use of one or few computational units
- Control overhead
- Large code and data memories

"There is no software / hardware tradeoff !" [Bob Brodeson, Feb. 2003]

- The difference between hardware and software is so large that there is no "tradeoff"
- It is reasons other than power, energy, performance or cost that drives a software solution (business, legacy, ...)

Good Reasons for Flexibility

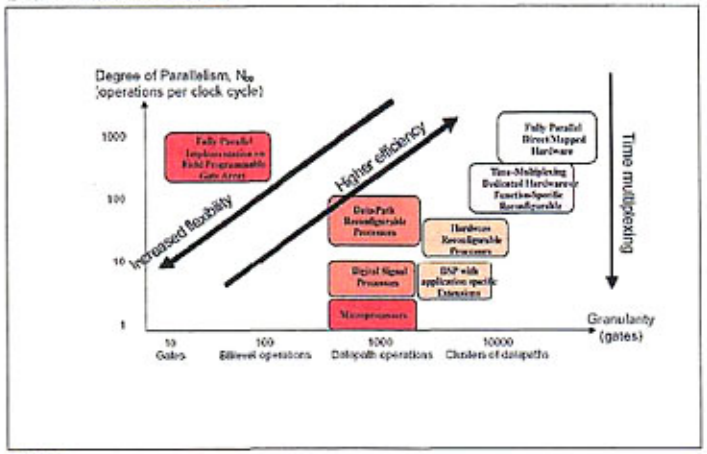
- One design for a number of SoC customers (high volume)
- Customers able to provide added value and uniqueness
- Unsure of specification or can't make a decision
- Backwards compatibility with debugged software
- Risk, cost and time of implementing hardwired solutions

"But": - These are business, not technical reasons

- The cost of flexibility is orders of magnitude of inefficiency over an optimized solution

[Bob Broderson, Feb. 2003]

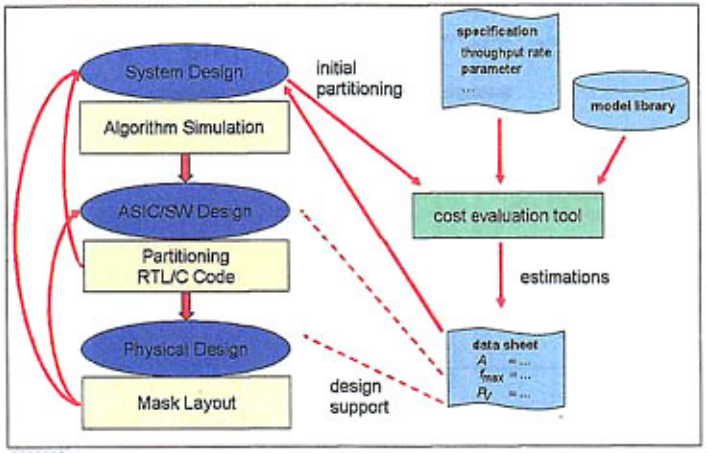
Bob Brodersen's View



Outline

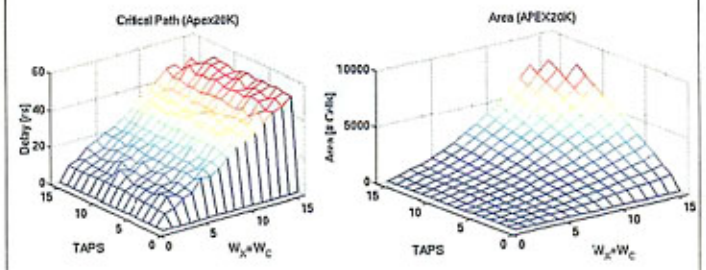
- ...
- ...
- Cost Modeling
  - FPGA blocks
  - hard cores
  - ...
- ...

Cost Evaluation Tool



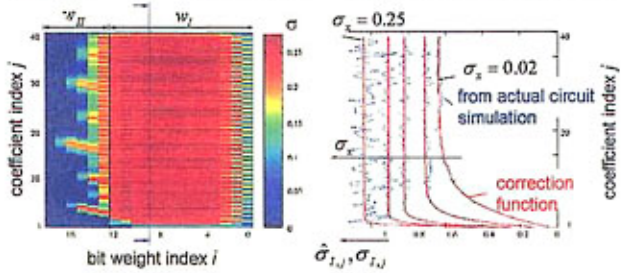
Fit Modelling of an FIR Filter on an FPGA

- Non-symmetrical coefficients
- Without multiplier pipelining
- Optimized word length in datapath



Simple Arithmetic Power Models

FIR Bitplane $\sigma_x = 0.25, \ldots, 0.02; \sigma_c \approx 0$; equally distributed coefficients



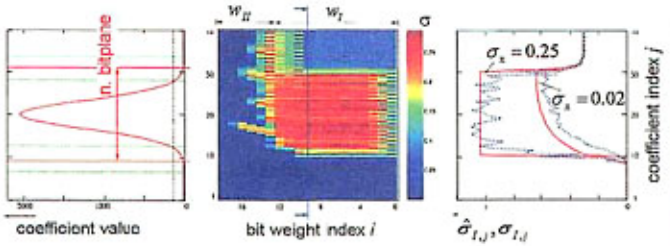
Fitting typical switching activity e.g with the fit parameters

$$\hat{\sigma}_{H,j} = a_H \cdot \left[1 - b_H \cdot e^{-j/c_H}\right] \Rightarrow P_H = f \cdot w_H \cdot \sum_{j=1}^{N} \hat{\sigma}_{H,j} \cdot E_{cell} \qquad a_H, b_H, c_H = f(\sigma_x)$$

Non-Equally Distributed Coefficients

Chosen example: Standard distributed coefficient values,
Standard deviation σ=0.5, coefficient wordlength w_c = 12

Non-zero coefficient bits according to a bitplane specific threshold

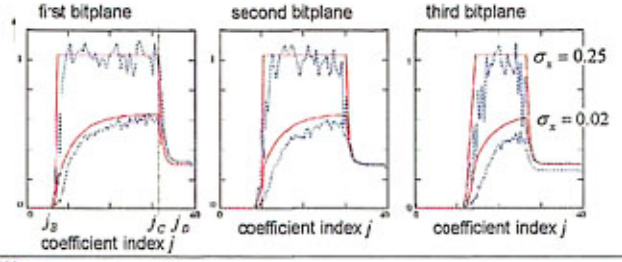


Non-Equally Distributed Coefficients (cont'd)

Switching activity according to a bitplane specific threshold

$$\hat{\sigma}_{H,j} = \begin{cases} 0 = const & ; j = 1 \ldots j_B \\ a_B \cdot \left[1 - b_B \cdot e^{-j/c_B}\right], & j = j_B + 1 \ldots j_C \\ a_C \cdot \left[1 - b_C \cdot e^{-j/c_C}\right], & j = j_C + 1 \ldots j_D \\ a_D = const & ; j = j_D + 1 \ldots N \end{cases}$$

with $a_\nu, b_\nu, c_\nu \ (\nu = B, C)$ fitted as before

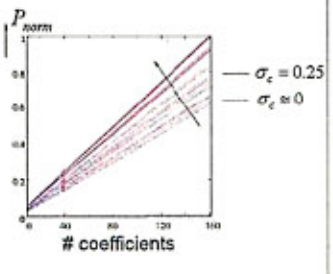


FIR Filter Example: Simulation and Modeling Results

Filter examples

$w_x = 10$, $w_c = 8$		P_{norm} 160 coefficients		
σ_x	σ_c	model	simulation	Error [%]
0.25	≈ 0	1.688	1.656	1.9
0.02	≈ 0	1.274	1.234	3.1
0.25	0.25	2.000	1.982	0.9
0.02	0.25	1.870	1.838	1.7
$w_x = 10, w_c = 8$		10 coefficients		
σ_x	σ_c	model	simulation	Error [%]
0.25	≈ 0	0.098	0.094	4.1
0.25	0.25	0.118	0.116	1.7

bitplane



FIR Filter Example: Simulation And Modeling Results (cont'd)

Filter implementation of an industrial partner

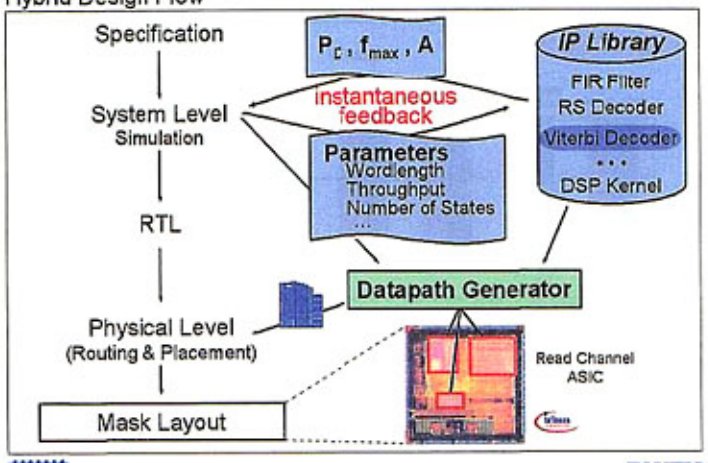
- interleaved bitplane structure
- different cell implementation

$\sigma_x = 0.25$ $\sigma_c \approx 0$	model	simulation	Error [%]
$P_{(clock)norm}$	0.436	0.448	2.7
$P_{(logic)norm}$	0.569	0.552	3.0
$P_{(\Sigma)norm}$	1.005	1	0.5

- different filter implementation
- same power model
- applying proper energy conversion figures

RWTH

Hybrid Design Flow



RWTH

Outline

- ...
- ...
- Cost Modeling
  - ...
  - SW kernels
- ...

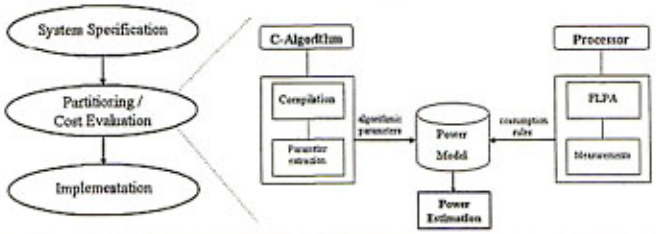
RWTH

Functional Level Power Analysis (FLPA) of DSP Systems

Power Estimation is divided into two steps:

- power model definition for DSP architecture
- parameter extraction for algorithm

Introduced by [Laurent, Univ. South Brittany, 01]



RWTH

Case study of the TI TMS320C6416 DSP

The TMS320C6416 is a last generation TI DSP featuring:

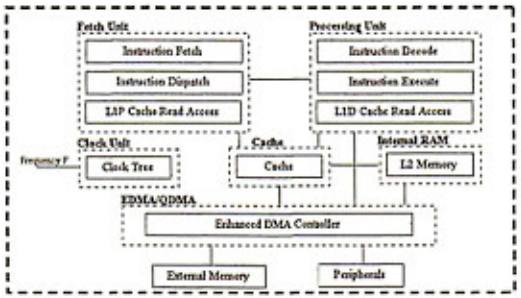
- deep pipeline (11 stages)
- VLIW instruction set
- parallelism capabilities (up to 8 instructions in parallel)
- two-level internal memory
- enhanced DMA controller

Complex architectures of modern DSPs require consideration of:

- pipeline stalls
- program and data cache misses

RWTH

FLPA for the TI TMS320 C6416 CPU



- DSP is divided into functional blocks
- Consumption rules for each block can be found by measurements / simulations
- Algorithmic parameters determine the utilization of each block

RWTH

Power Models

Consumption rules:

$P_{Fetch} = (a^* \alpha^2 + b^* \alpha + c)^* d^* F^* (1\text{-}PSR)$

$P_{Proc} = (a^* \alpha + b)^* c^* F^* (1\text{-}PSR)$

$P_{Clock} = (a^* F + b)^* c$

$P_{Memory} = (a^* \beta + b^* \gamma)^* c^* F^* (1\text{-}PSR)$

$P_{Cache} = (a^* \delta + b^* \varepsilon + c)^* d^* F^* (1\text{-}PSR)$

$P_{EDMA} = (a^* \zeta + c)^* d^* F^* (1\text{-}PSR)$

α: parallelism rate

PSR: pipeline stall rate

β: mem. read access rate

γ: mem. write access rate

δ: L1P cache miss rate

ε: L1D cache miss rate

ζ: EDMA activity rate

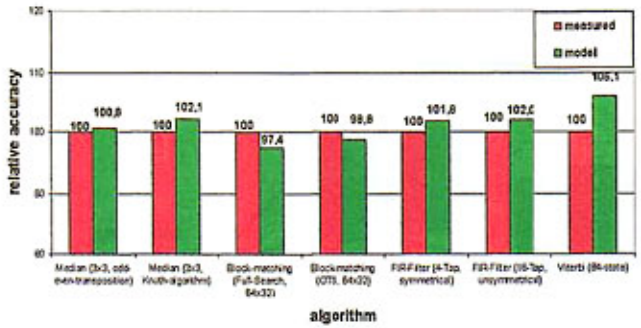
Verification:

Estimation of power consumption of a first set of signal processing applications yields a maximum error of 6%

RWTH

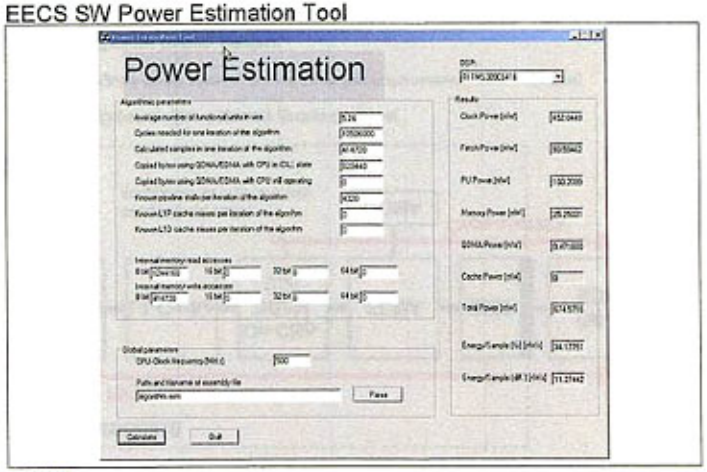
Model vs. Measurement Results

Exemplary results of energy/sample estimation for different algorithms using FLPA

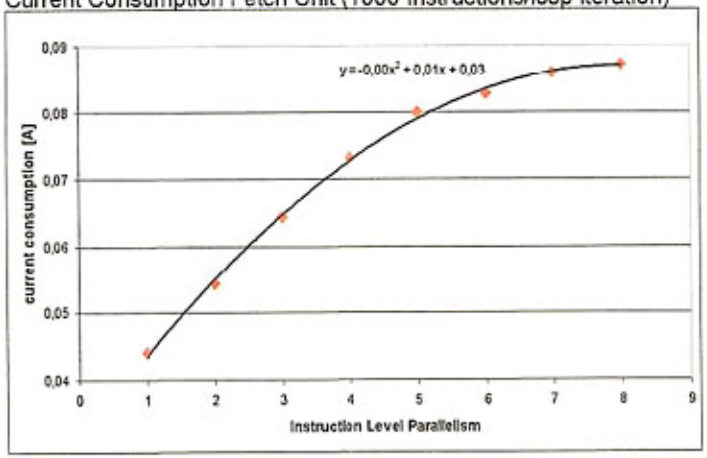


RWTH

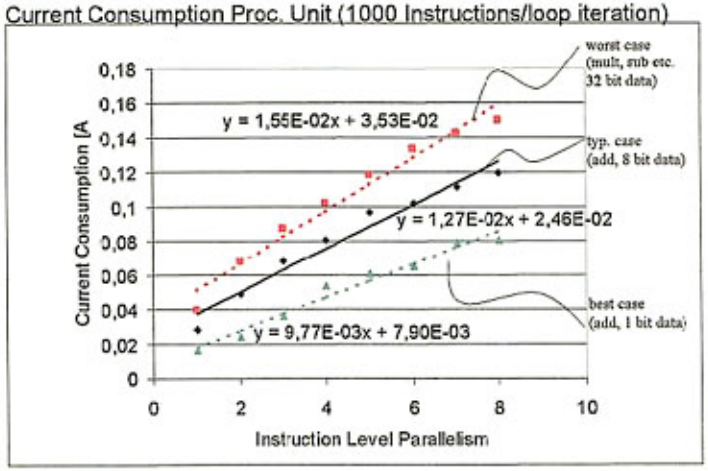
EECS SW Power Estimation Tool



Current Consumption Fetch Unit (1000 Instructions/loop iteration)



Current Consumption Proc. Unit (1000 Instructions/loop iteration)



Outline

- . . .
- . . .
- On-SoC Communication Issues
- . . .

On-SoC Communication

Today: Future:

Memory centric ⇒ Interconnect centric

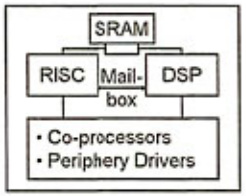
What interconnect architecture
- Point-to-point ?
- Busses ?
- Network on Chip ?
- Dynamically reconfigurable ?
- Hierarchical ?

⇒ Need ... - Physical level cost models (wires, interfaces, ...)
- Application level traffic models (protocols, ...)

... for quantitative optimization

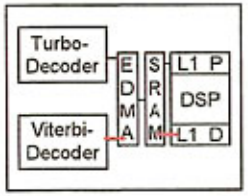
Exemplary Heterogeneous Systems on Chip

Texas Instruments OMAP
Open Multimedia Applications Platform



- Shared Memory (Read/Write communication)
- Synchronization applying proprietary mailbox mechanism
- Simulation of on chip communication not documented

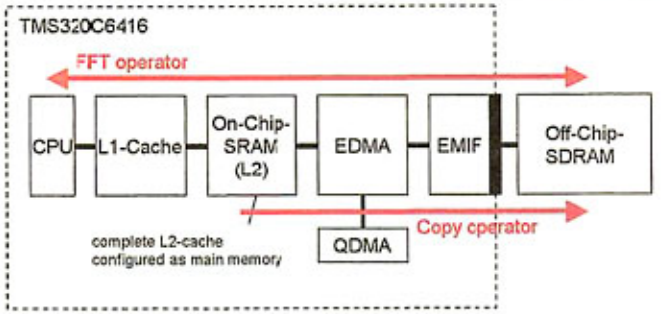
Texas Instruments TMS320C6416



- SRAM: applied as L2-Cache and / or RAM
- communication with coprocessors via SRAM / EDMA (Read/Write communication) synchronized by interrupts

EDMA = Enhanced Direct Memory Access

Example for a Joint Resource Access of two Units



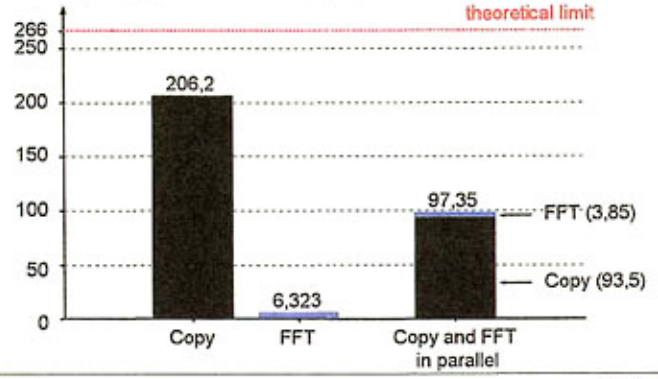
EDMA = Enhanced Direct Memory Access Controller

QDMA = Quick Direct Memory Access Controller (CPU relieving copy operation)

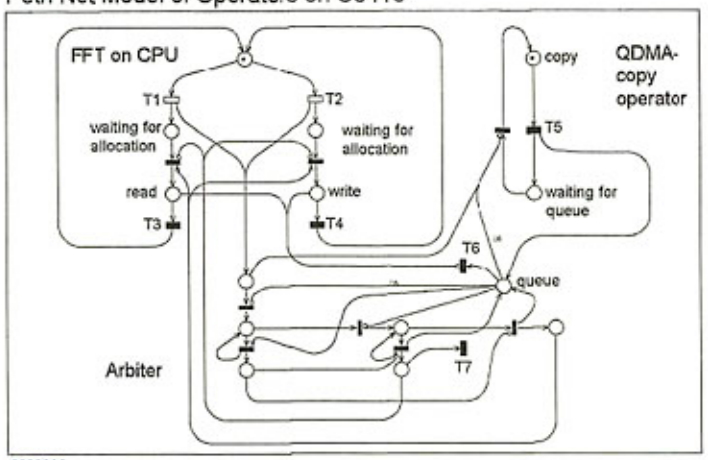
EMIF = External Memory Interface

Measured values

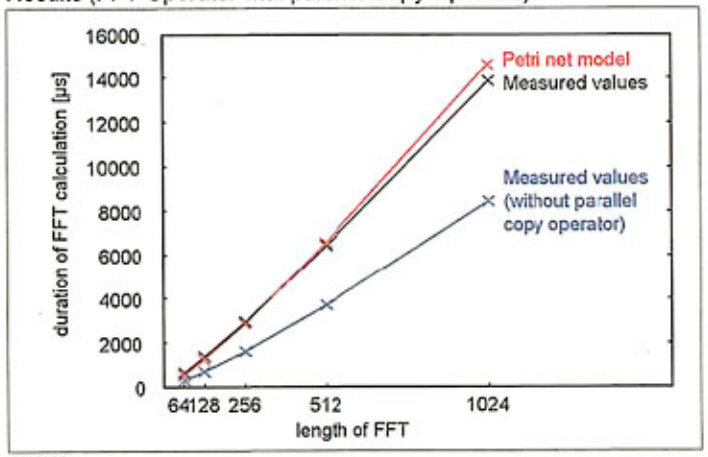
Memory throughput in 32-Bit-words per µs



Petri Net Model of Operators on C6416



Results (FFT-Operator with parallel Copy Operator)



Outline

- ...
- ...
- Fault tolerance
- ...

Fault Tolerant Building Blocks

⇒ Need for Fault Tolerance against

- Permanent Faults as
  - Fabrication Faults (opens, shorts, stuck-ats, ...)
  - Aging Faults (device degradation, electromigration, ...)
  - Insufficient Test Capabilities
- Transient Faults as
  - Noise (worst case coupling, EMC, timing glitches, ...)
  - Alpha Particles
  - ...

⇒ Add Redundancy

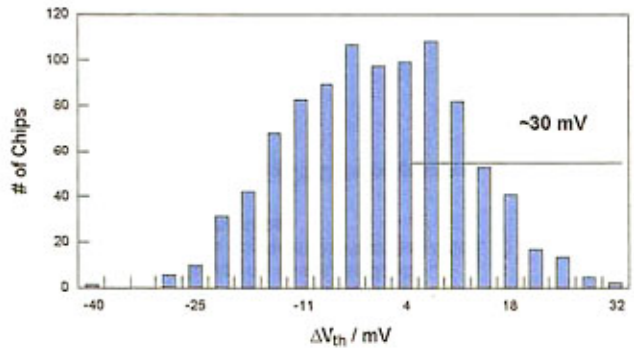
Issues:
- Overhead ?
- Fault Distribution ?

Increased Noise due to

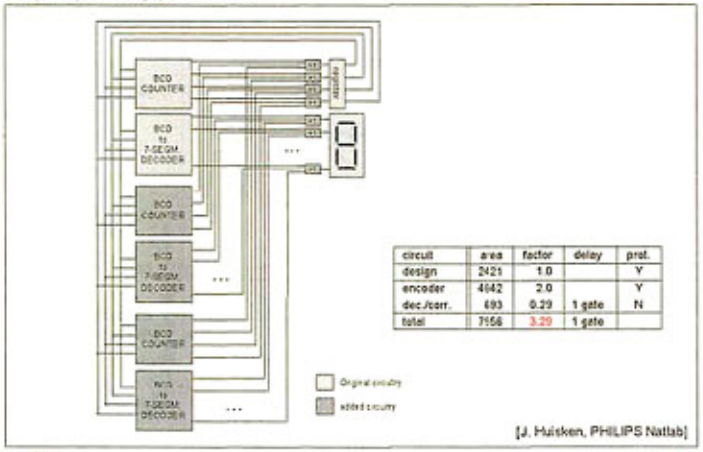
- Increased mutual coupling
  - capacitive
  - inductive
- Increased voltage drop
  - resistive $\Delta V = R \cdot I$
  - inductive $\Delta V = L \cdot {}^{dI}/_{dt}$ ⚠
- Increased slew rates (due to higher clock frequency)
- Increased noise propagation (due to high-speed circuit techniques)
- Soft errors (α-hits)

⇒ Noise budget doesn't scale

V_{th} Fluctuation due to Discrete Dopand Effects



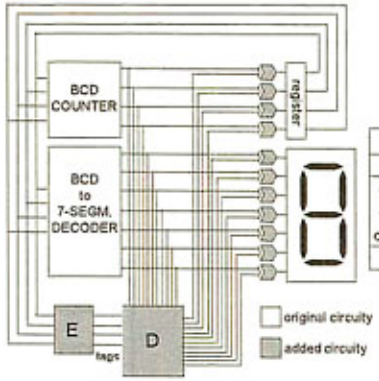
Majority voting protection



circuit	area	factor	delay	prot.
design	2421	1.0		Y
encoder	4842	2.0		Y
dec./corr.	693	0.29	1 gate	N
total	7556	3.29	1 gate	

[J. Huisken, PHILIPS Natlab]

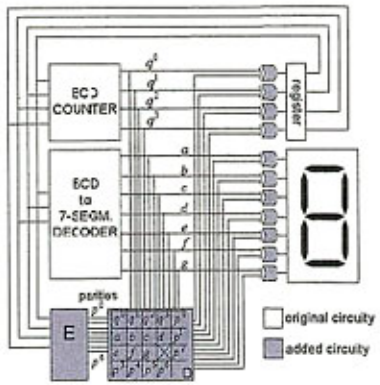
Hamming Coce Protection



circuit	area	factor	delay	prot.
design	2421	1.0		Y
encoder	333	0.14		Y
decoder	2538	1.05	4 gates	N
correction	594	0.25	1 gate	N
total	5886	2.44	5 gates	

[J. Huisken, PHILIPS Natlab]

Product Code Protection



circuit	area	factor	delay	prot.
design	2421	1.0		Y
encoder	756	0.31		Y
decoder	810	0.33	3 gates	Y
correction	990	0.41	2 gates	N
total	4977	2.05	5 gates	

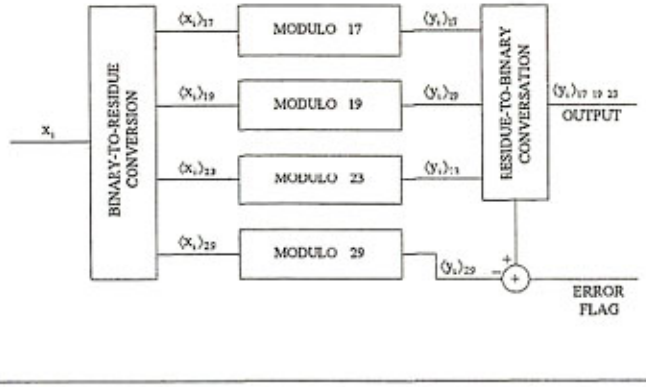
[J. Huisken, PHILIPS Natlab]

Results On A Larger Design

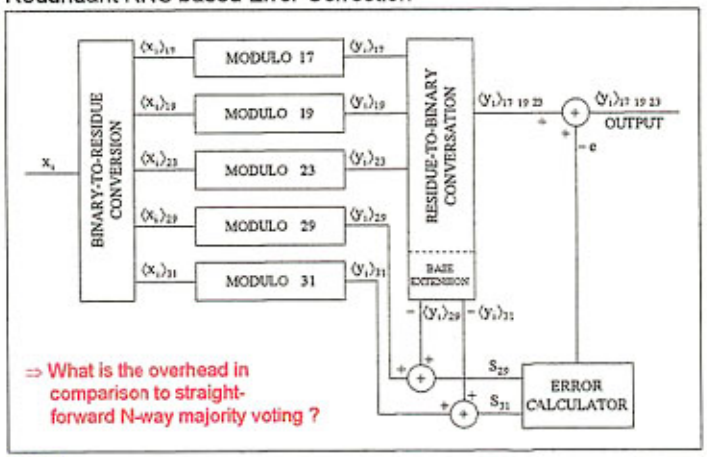
Area µm²	orig.	Maj. v.	Ham.	product
design	88605	88605	88605	88605
encoder		177210	43911	82512
decoder		862	3510	1800
corrector		implied	756	756
factor	1.0	3.01	1.54	1.96
unprot.	100%	1.0%	4.5%	1.4%

[J. Huisken, PHILIPS Natlab]

Redundant RNS based Error Detection



Redundant RNS based Error Correction



⇒ What is the overhead in comparison to straight-forward N-way majority voting ?

Fault Tolerance: State-of-the-Art

Eficient fault tolerance schemes available for

- Memories ✓
- FPGAs ✓
- Processor ALUs (✓: error detection and re-issuing)
- Combinational logic ?
- Sequential logic ???

If we are not able to provide EFFICIENT fault tolerance decreased reliability may become the show stopper for Moore's Law !

RWTH

Outline

- . . .
- . . .
- Outlook: SoCs vs SiPs

RWTH

Systems-on-Chip vs. Systems-in-Package

The progress in interconnect and packaging technology allows SiP solutions with reduced chip-to-chip overhead

- Avoids technology trade-offs for individual components (logic, mems, analog/RF, ...)
- Eases electrical compatibility dramatically (noise, ...)
- Reduces fabrication costs (dedicated technologies, testing, ...)

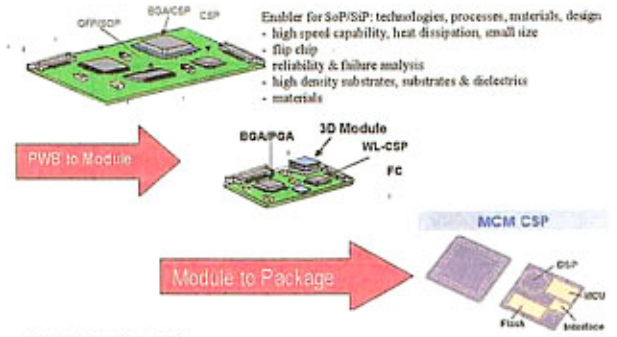
Associated cost savings can overcompensate increased packaging costs

⇒ Need for quantitative optimization

RWTH

Design for Forward Integration Including Memories

- Intelligence meets package (Chip/Package/Board) SiP vs. SoC



RWTH

Thank you for your attention !