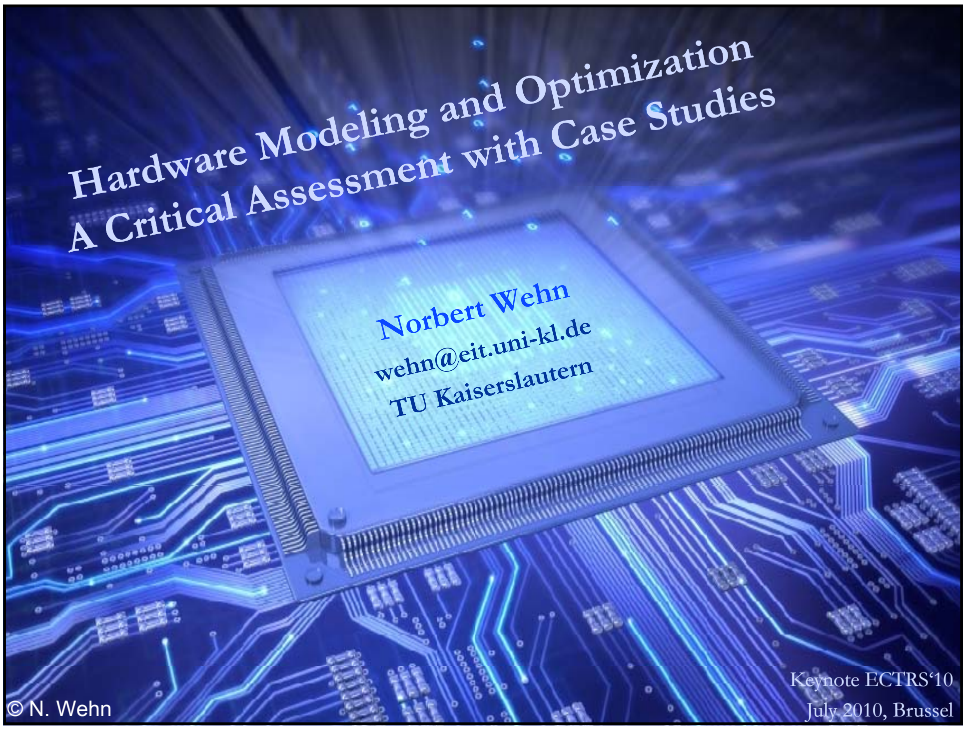
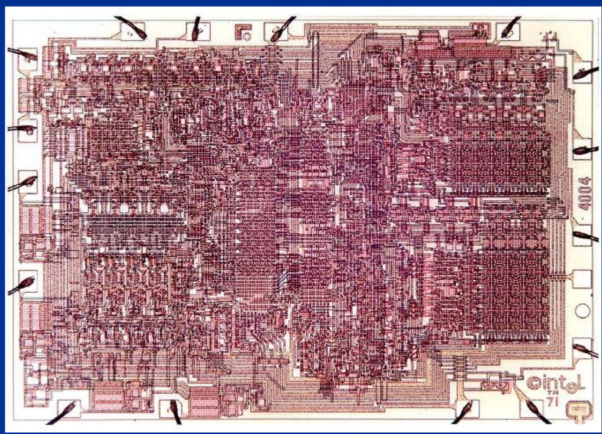


Outline

- Progress in microelectronics
- Limits for Moore's Law
- Power/Energy Wall
- Architecture Impact
  - Homogeneous versus heterogeneous architectures
- Power/Energy Models
  - CPU Cores
  - DRAMs
  - WSN
- Outlook

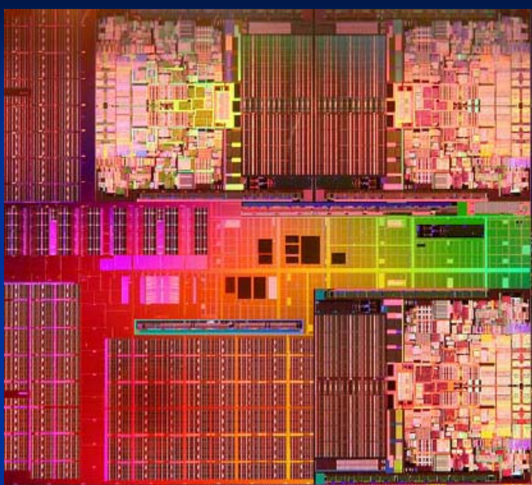
Progress in Microelectronics

- First microprocessor 1972
- 2250 transistors, 108KHz, 10um, 11 mm^2

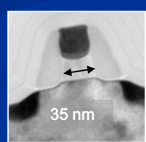


Source: Intel

Intel Dunnington



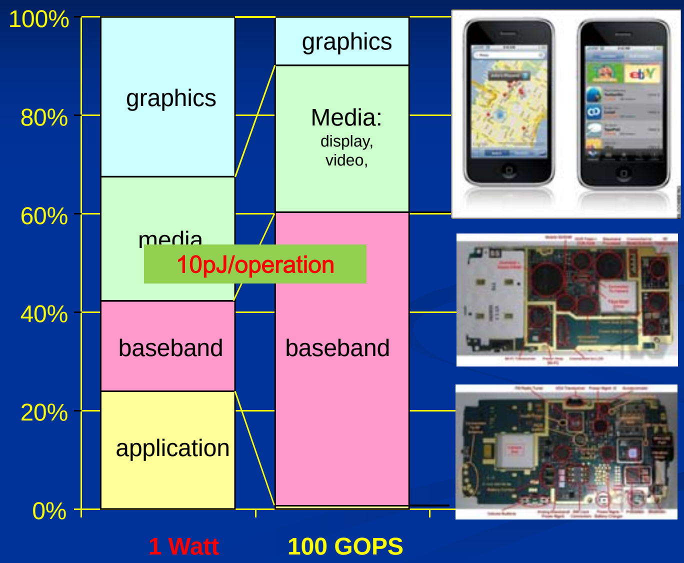
- 6 Core processor
- 1900 Mill. Trans.
- 2.66 GHz, 45nm
- 403 mm^2
- >130 Watt



#Transistors	x 10^6	#Frequency	x 30.000
#Feature size	x 200	# Chip size	x 40

Source: Intel

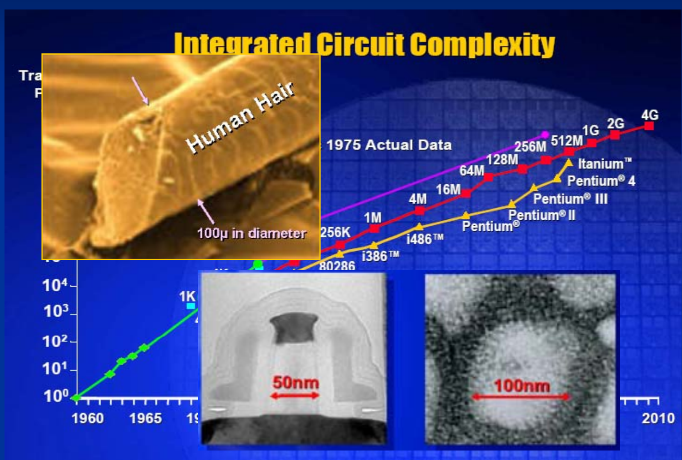
Modern 3.5G Smartphone



© N. Wehn

Source: Kees van Berkel, DATE2009

Moore's Law (1965)



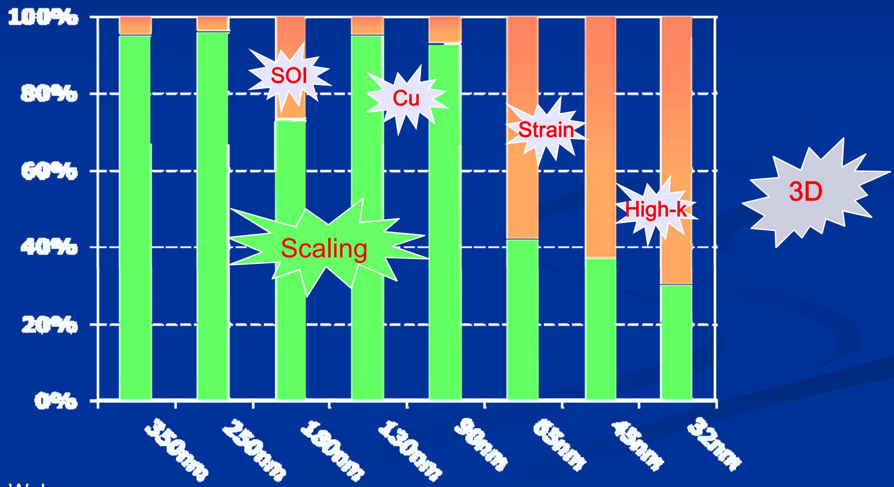
- Scaling based manufacturing process (lithography)

© N. Wehn

Source: Intel

Technology Progress

- Technology Innovation

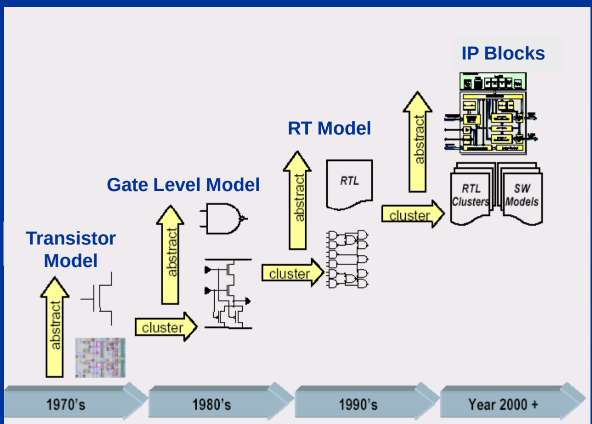


© N. Wehn

Design Methodology Progress

Disciplined engineering design methodology

- Clustering and abstraction, models and sophisticated algorithms



© N. Wehn

Limits of Moore's Law

Not technology but cost

- **E.g. 22nm technology node**
  - Fab 4.5-6.5 Billion $
  - Process R&D: 1.3 Billion $
  - Design 140 Mill. $
- **Design starts in first 5 years**
  - 45nm: 526
  - 32nm: 244
  - 22nm: 156

Design Complexity

Power/Energy

Power/Energy Wall

Covers entire spectrum of market sectors

- Large datacenters (MW), mobile devices (W), sensor platforms (uW)

Economic impact

- Worldwide CO_2 emission due to I&C technology ~ airplane emission
- 20% increase/year
- Datacenters: energy cost dominate overall costs, cooling cost about 40% of total energy

Thermal issues/hot spots

Reliability

Limited energy resources for mobile and sensor applications

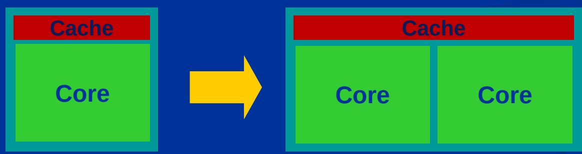
Show stopper for further integration

⇨ **Parallel and heterogeneous architectures**

Why Multi-core

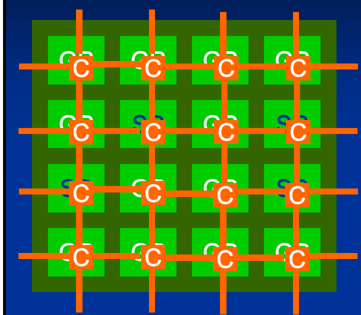
Pushing clock speed for getting higher performance in uP stopped
INTEL: Speed/Power Trade-off =fundamental theorem of multi-core

Frequency reduction by	yields performance reduction	and power reduction
20%	13%	50 %



Freq	= 1		Freq	= 0.8
Area	= 1		Area	= 2
Power	= 1		Power	= 1
Perf.	= 1		Perf.	= 2 x 0,87=1.74

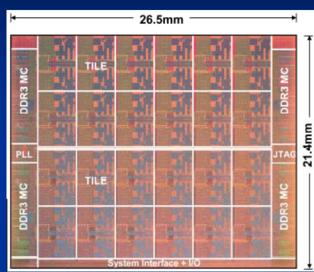
(Future) GPP Multi-core Platform



General Purpose Cores

Special Purpose HW

Scalable Interconnect



- E.g. Intel's 48 core computer (ISSCC'10)
  - 48 Pentium IA-32 processor cores, 576mm², 45nm, 1.3 Billion transistors
  - 6x4 2D Mesh NoC
  - Speed/Power Trade-Off

    1.5GHz@1.3V ⇨ 200W/50C°

    1.0GHz@1.14V ⇨ 125W/50C°

    125MHz@0.7V ⇨ 25W/50C°

Source: ISSCC'10

Homogeneous vs Heterogeneous Architectures

- **Homogeneous (HPC)**
  - Regularity simplifies hardware design, validation and manufacturing
  - Simplified programming model ⇨ software development
  - Large flexibility, no application specific computing platform ⇨ lowers cost
  - Many optimization opportunities to operating system i.e. run-time scheduling
- **But what about energy efficiency?**
  - Simpler cores are more energy efficient than complex cores
    - E.g. calculating cloud resolving climate model

      AMD Opteron 2.8GHz 1.700.000 Cores ⇨ 179MW

      Tensilica Xtensa 500MHz 10.000.000 Cores ⇨ 3MW
  - Processor performance has to match task workload and its *characteristic*
    - Voltage scheduling
    - Dynamic task scheduling
    - But think on its overhead (energy, latency)!

Heterogeneous Parallel Architectures

Role of Software
- Provides large flexibility
- Metric in SW: functionality, modularity and reusability
- SW can never improve the energy efficiency, it can just enable it
- Reality: SW often disables energy efficiency

E.g. "Computing" in smart phone 100 GOPS@1W
- SW implementation on embedded ARM11 processor: 20W
- SW implementation on DSP processor: 2..5W
- Dedicated HW implementation: 0.2-0.5W

  E.g. MPEG decoding: HW ⇨ SW ⇨ HW

Fundamental Trade-off Flexibility/Energy

⇨ Heterogeneous architectures

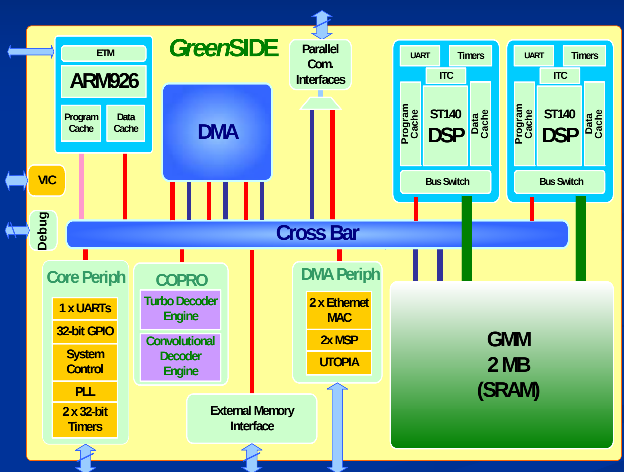
Homogeneous vs Heterogeneous Architectures

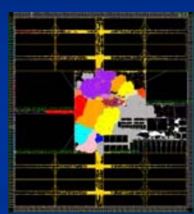
- **Heterogeneous**
  - Driven by energy and latency constraints ⇨ application specific
  - Latency and energy efficient tasks ⇨ dedicated /optimized hardware blocks
  - Flexibility for run-time optimization very limited, mainly static scheduling and mapping at design time
  - Increased complexity in hardware design and validation
  - Bound to an application class ⇨ higher cost
- **Energy and latency critical applications (e.g. mobiles)**
  - Heterogeneous architectures are dominating

⇨**Multi-Processor-System-on-Chip (MPSoC)**

Heterogeneous MPSoC (STM/TU Kaiserslautern)

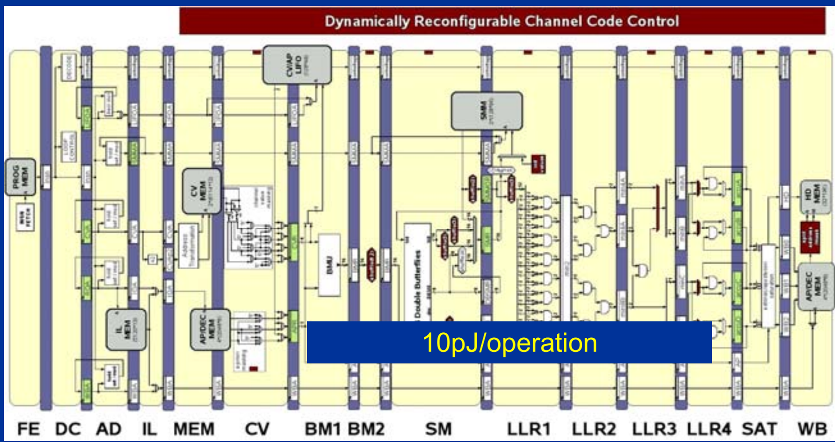
- **2.5G and 3G Baseband Modem Chip**





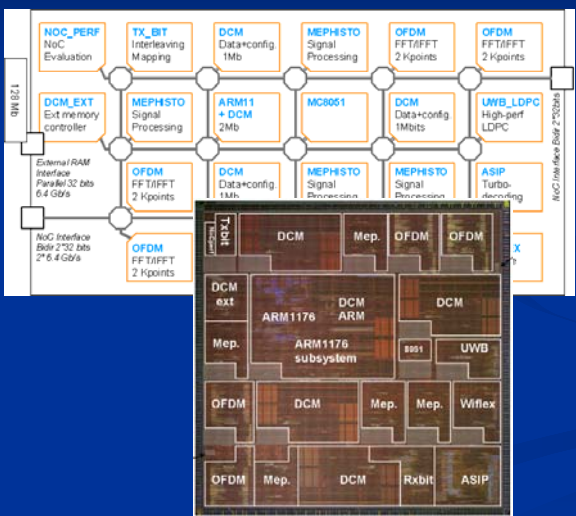
ASIP (TU Kaiserslautern)

- Blurring border HW and SW: Application Specific Instruction Set Processor
  E.g. ASIP for channel decoding in SDR
  65nm technology, 385MHz, 0.7 mm^2, 100mW



Magali Chip
(LETI / TU Kaiserslautern)

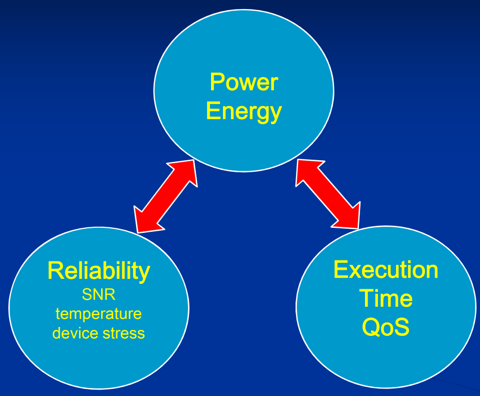
- 477mW NoC Based Digital Baseband for MIMO 4G SDR (ISSCC'10)
- 96Mtransistors, 27mm^2, 65nm technology



- 22 processing units:
  5 VLIW processors
  ARM11 processor
  ASIP processor
  Many HW accelerators
- 15 asynchronous NoC router
- Distributed power management

Source: ISSCC'10

Power/Energy



- Accurate power/energy models are key
- Modelling of power/energy of key building blocks
  - CPU, DRAM, Wireless Sensor Nodes

Power/Energy Models

- Frequently rely on simple assumptions
- Standard equations for frequency and voltage scaling/scheduling (DVFS)

$$P_{active} \sim f_{(V_{DD})} \cdot V_{DD}^2$$

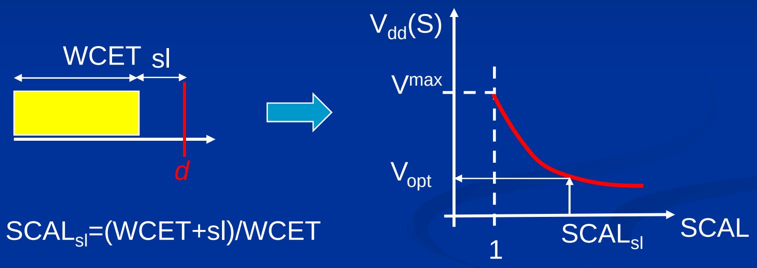
$$E_{active} \sim P_{active} \cdot t = \frac{1}{T} \cdot V_{DD}^2 \cdot T \cdot cycles = V_{DD}^2 \cdot cycles$$

⇨ Frequency scaling only impacts power and not energy

⇨ Lowest possible voltage/frequency yields power/energy optimum

Dynamic Voltage Scaling/Scheduling

- **Given task with known WCET and deadline d**
- **Find minimum V_{DD}/f such that d is fulfilled on a processor**



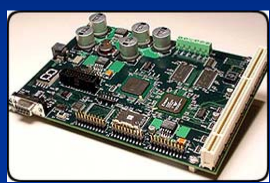
Trade-off: energy/performance

- **Lowering V_{DD} lowers E quadratic but decreases f**
- **Accurate relation between V_{DD}, frequency f , performance, energy**
- **Accurate estimation of execution time**

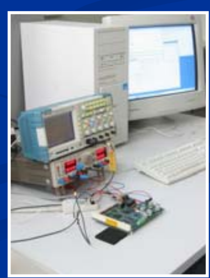
⇨ Understanding system behavior

A Case Study: XScale Board

- **XScale 80200**
  - 32-Bit CPU with optimized RISC Pipeline
  - Dynamic Frequency Scaling (333MHz to 733MHz)
  - Dynamic Voltage Scaling
  - 32kB data cache, 32kB instruction cache

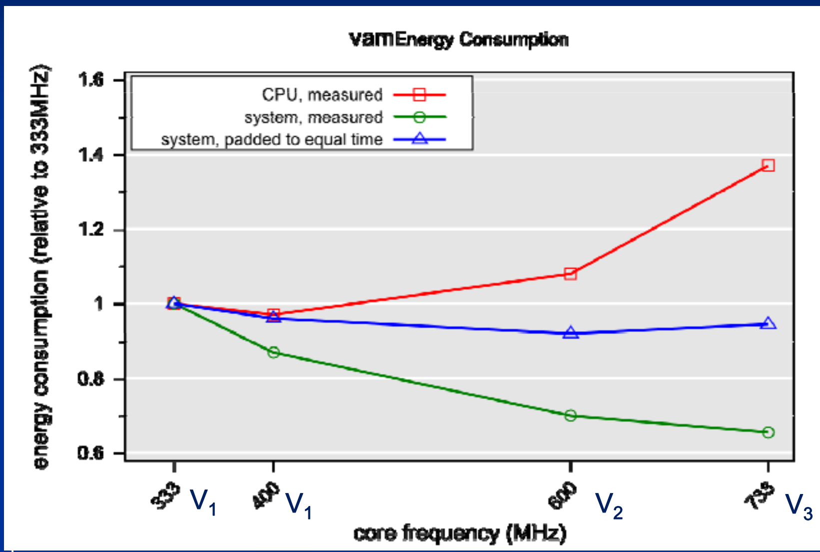


- **XScale Evaluation Board for measurements**
  - Memory: SDRAMs
  - Memory Controller implemented in FPGA



Energy Measurements

- Performed measurements at different voltages V_1 V_2 V_3



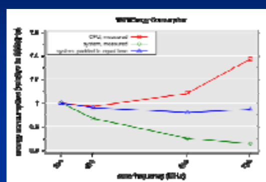
Lessons Learnt for Energy

You have to take into account the whole system

- Voltage and frequency scaling does not affect memory

$$E_{active} = E_{scal}(V_{DD}^{scal}) + P(V_{DD}^{fix})\, t$$

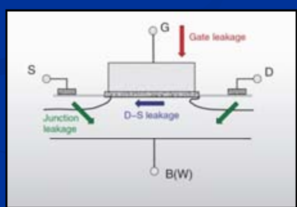
- Higher frequencies can be more energy efficient



active

inactive

standby/ leakage



$$E_{total} = E_{active} \cdot \delta + P_{leakage\_standby} \cdot t \cdot (\delta - 1) + E_{overhead}$$

Further Lessons for Energy

- In reality only some discrete voltages possible
- Any voltage change implies overhead (DC/DC converter, PLL)
  - Latency: x 1.000 cycles
  - Energy overhead

Multi-Core architectures

- Processor core energy (performance) is often not dominating
  E.g. INTEL 48 core computer
  - Maximum Speed: Cores@1GHz, NoC@2GHz
    ⇨125W@1.14V@50C°: 69% cores, 30% NoC and DRAM interface
  - Low Power Mode: Cores@125MHz, NoC@255MhZ
    ⇨ 25W@0.7V@50C°: 21% cores, 70% NoC and DRAM interface

Execution Time Estimation

Execution_time $t = \frac{1}{f_{V_{DD}}} \cdot \text{cycles}$

- **Relation between f and V_{DD}** $f_{V_{DD}} \sim \frac{(V_{DD} - V_t)^\alpha}{V_{DD}}$ with $\alpha = 1..2$

⇨ **But fitting model in valid voltage operation range:** $f_{max} \sim V_{DD}^{1.75}$

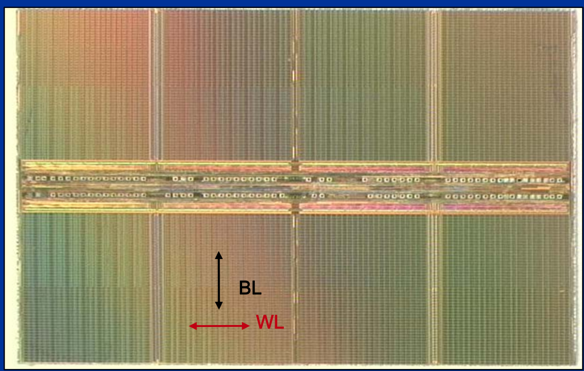
- **Cycles: processor cycles + cycles for external memory accesses**

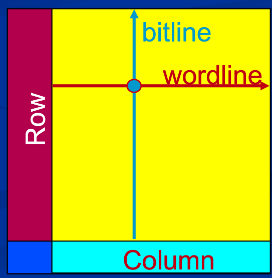
External memory accesses

- **Cache miss modeled in many models/simulators**
- **Cache miss ⇨ external memory (DRAM) has to be accessed**

A modern 1 Gbit DDR2 SDRAM

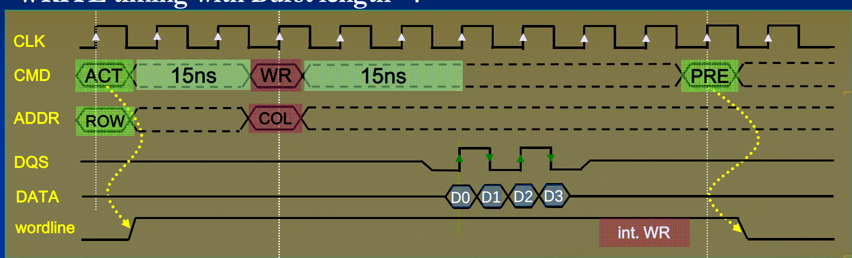
- Minimize cost/bit ⇨ minimize cell area (cell size 0.02um² 20..30fF)
- Access time to individual memory cell nearly constant over time
  ⇨ CAS latency improvement < 7%/year
- Throughput improvements in periphery/interfaces
  ⇨ complex interface protocol



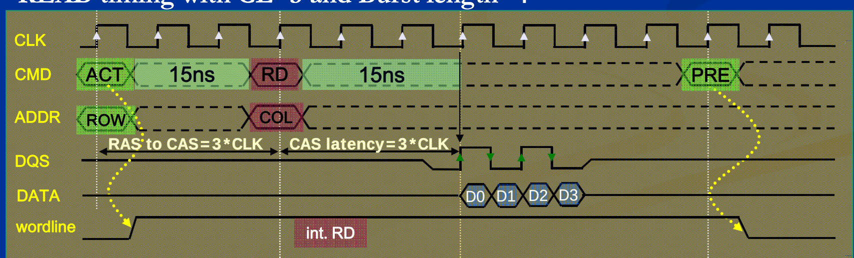


DDR2 SDRAM Timing Protocols

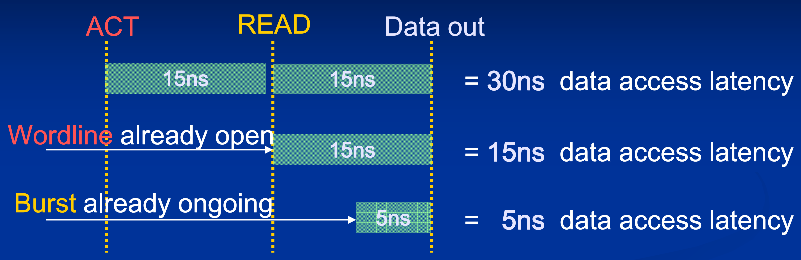
WRITE timing with Burst length=4



READ timing with CL=3 and Burst length=4



DDR2 Timing



- Large difference in timing (factor 6)
- Each activate (ACT) of a wordline is power hungry

State-of-the art in many models/simulators

- Fixed latency for memory access and fixed energy/access

Energy and performance optimization

- Re-ordering of the DRAM accesses to avoid re-opening of rows

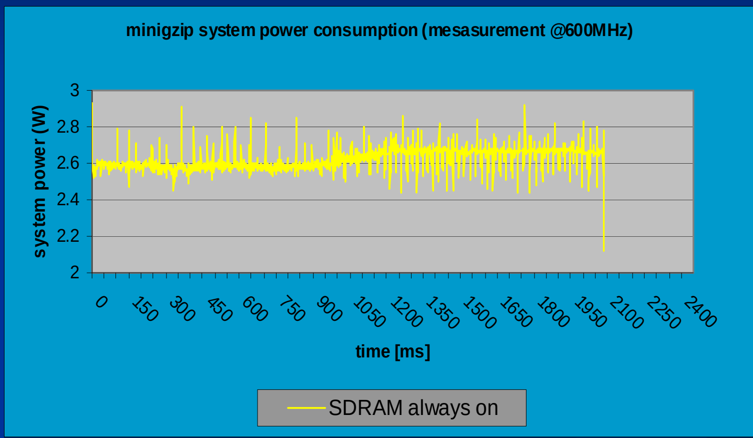
DRAM Power Models

- Many power modes for DRAMs
  - E.g. active (3nJ), standby (0.8nJ), power down (0.005nJ)
- SDRAM Power model from manufacturer Micron available
  - State based model
  - Worst case assumptions
  - Similar models from Rambus
- These models are base of existing simulators and optimizations
- Power model suggests aggressive use of DRAMs low-power modes
- Measurements with modified memory controller

  Minigzip ⇨ high memory activity
  Djpeg ⇨ medium memory activity
  Vam ⇨ very low memory activity

Power Measurements

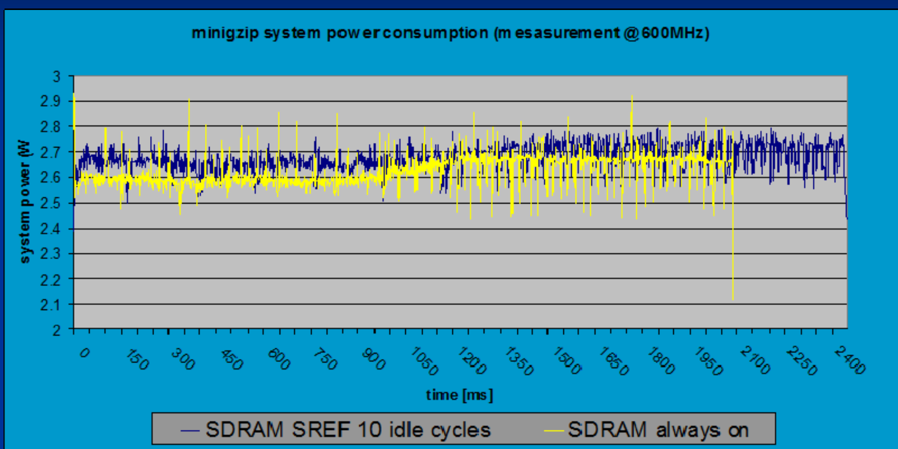
- Measured power consumption of the minigzip benchmark



- Switching SDRAM to low power mode after 10 idle cycles on memory bus

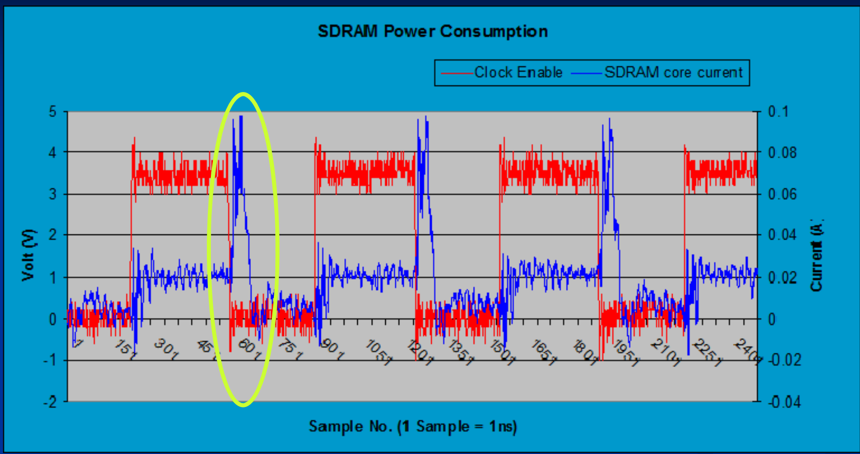
Aggressive Use of Low Power Modes

- Predicted reduction of average power (Micron model): 173 mW



- Increase in program runtime due to transition time active ⇔ low power state
- Average power consumption rises by 100mW (prediction -173mW)

Power Analysis



- When switching to memory power down mode, a power peak is observed due to a refresh: valid for all DDRx too (JEDEC standard)
- Not modeled in Micron's power model, not taken into account in any previous publication we know of

Lessons Learnt

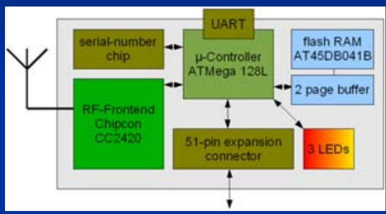
- **DRAM access protocols are complex and show large latency and energy variations**
  - Fixed DRAM access latency and energy is wrong assumption
- **Theoretic power models for SDRAM are misleading**
  - Overestimate power consumption and energy saving potential
  - Neglect important effects like transition energy
  - Not only wrong absolute numbers but also wrong trends
- **Aggressive SDRAM power management is not always beneficial**

New Xscale/DRAM simulator (www.inf.u-szeged.hu/xeemu)

- Average error: 3.0% (runtime), 1.6% (core energy), 3.3% (DRAM energy)

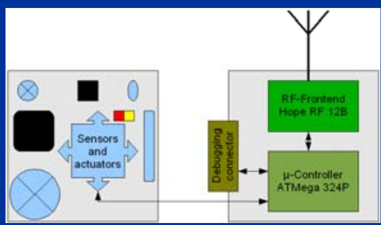
Wireless Sensor Nodes

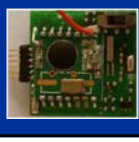
- **Key devices in „swarm systems“**



MICAz

Source: Crossbow Technology





AmICA
TU Kaiserslautern

Common Assumptions in WSN

Energy: Transmit energy dominates energy consumption
Long distance: Multi-hop is the preferred solution
Robust wireless communication: ARQ is more (energy) efficient than forward error correction (FEC)

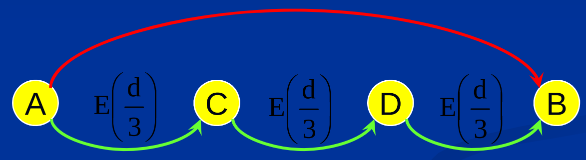
Based on theoretical assumptions and simplified models
Accurate Power models for many sensor nodes missing
Our approach:
- State based model
- Flowcharts to represent valid traces the FSM

$$E = \sum_{state} P_{state} \cdot t_{state} + \sum_{state} P_{transition} \cdot t_{transition}$$

Short Hops Versus Long Hops in WSN

- Transmission energy has exponential growth with distance d
  E(d) ~d^α with α path loss exponent (1 < α < 4)

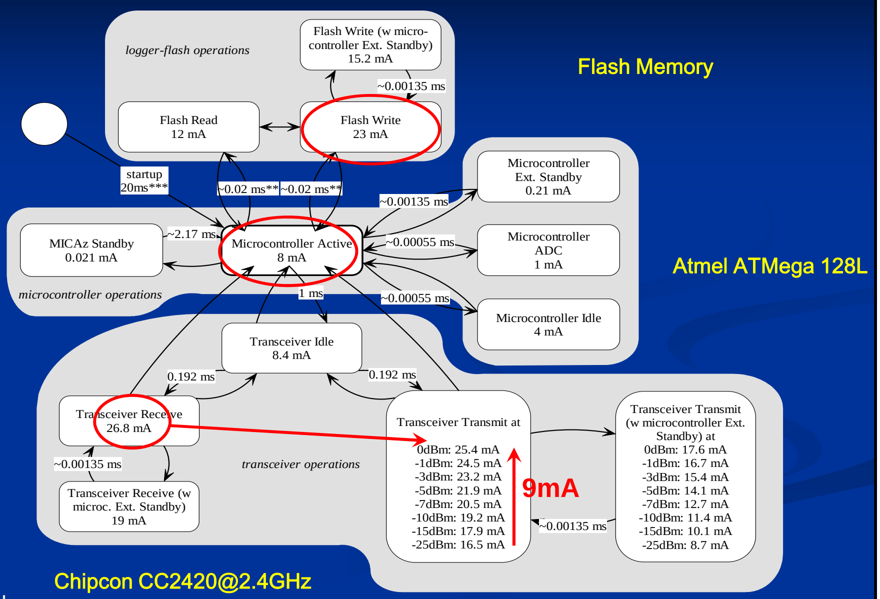
$$E(d) \sim d^{\alpha}$$



$$\text{Multi\_hop}: \; n \cdot E\left(\frac{d}{n}\right) = n \cdot \left(\frac{d}{n}\right)^{\alpha} = \frac{1}{n^{\alpha-1}} E(d)$$

- Theory favours many short hops
- Forward Error Correction inefficient since E(FEC) > E(d) for small d

State-Based MICAz Power Model



Observations

Communication vs Computation

- $E_{compute}$ ~ 2nJ/operation
- E_{send} ~ 230nJ/useful bit

⇨ E_{send}(127 bytes) ~ E_{uC}(100.000 cycles)
E_{send}(1 bit) ~ 100…4000 x $E_{compute}$(1 instruction)

Computation vs Flashstorage

- E_{flash_write}(127 bytes) ~ E_{uC}(300.000 cycles)

Communication

- $P_{receive}$ ~ $P_{transmit}$

⇨ Large energy for ACK based protocols
  - E.g. frame length with 60 bytes
  - Energy_RX_ACK/total_energy: 80% (10ms), 30% (0.5ms)

Consequences

Frame loss and relaying have to be minimized for energy efficiency

Instead of ARQ

- Use of FEC to trade-off communication versus computation energy
- Only theoretical investigations known

Many applications: single hop asymmetric structure with central powerful node for information aggregation

Measurements in Lab environment

- ARQ with CRC Checksum
- Repetition codes (1/3, 1/6) with majority voting
- Turbo-Code (UMTS, 1/3)

© N. Wehn

Energy Measurements Results

- 3 MicaZ nodes running in parallel, ~ 1% BER in noisy WLAN environment
- 1 frame/sec sent, measured left-over battery cap after 120 hours runtime

Method	# of sent Messages	Ø Frames / succ. Message (#ARQ)	Energy / succ. Message [µJ]
Only ARQ Battery fully depleted after 48 hours Extrapolated to a runtime of 120h	431,906	2.34	346,049
ARQ + Rep 1/3	431,737	1.18	16,842
ARQ + Turbo-Code	431,728	1.12	11,798

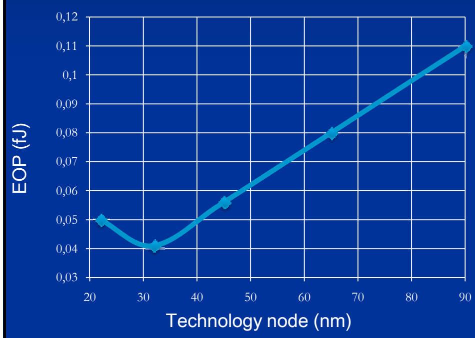
- High number of retransmissions requires a large on-time in receive mode
- Overhead for encoding is more than compensated for by higher reliability

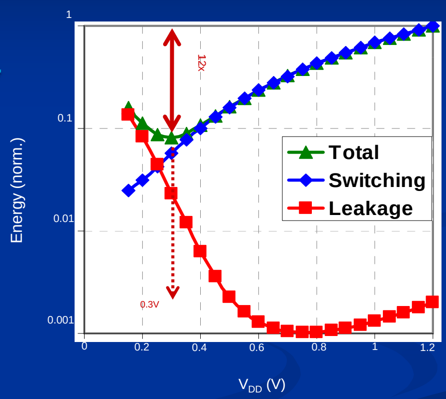
Energy Improvement by >20X compared to only ARQ

Lessons Learnt

- Careful Trade-off computation vs communication energy
- Not only transmit power dominates, receive power is as important
- Multi-Hop is conclusion of wrong power models
- Forward error correction can be very efficient in star shaped networks

Scaling doesn't solve Low Power Problem



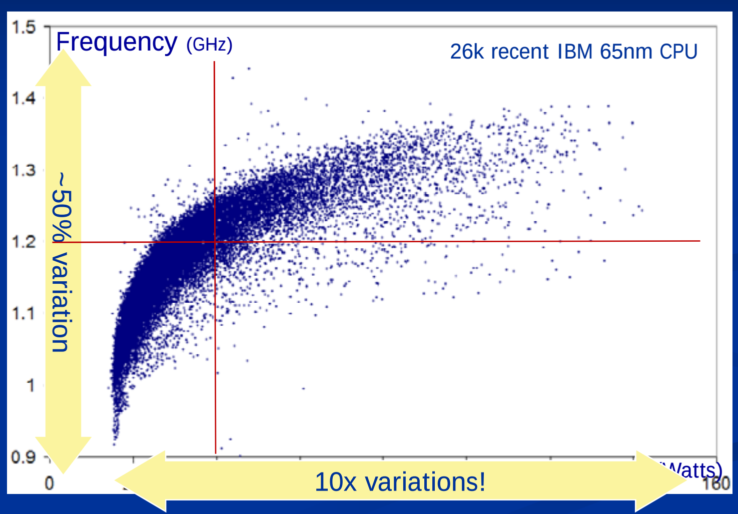


Technology Scaling

Voltage Scaling

Some Look into the Future

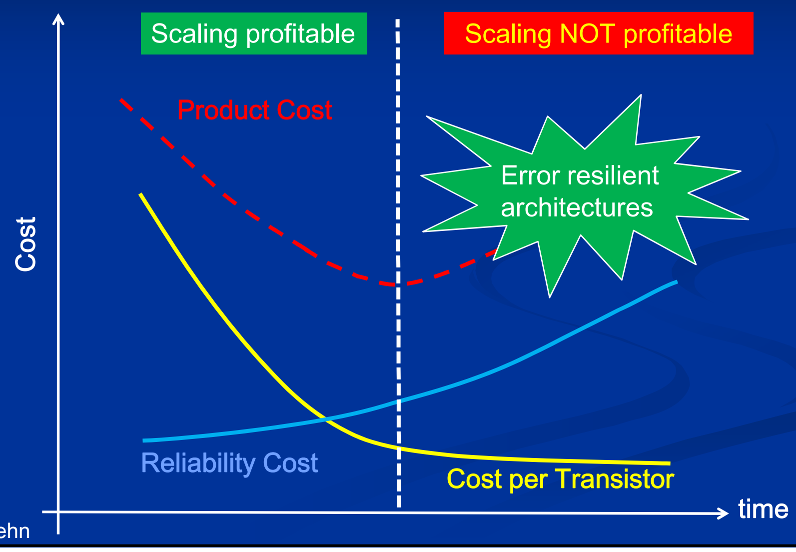
- Smaller technology dimensions: variability impacts predictability



Source: Sani Nassif

End of Worst Case Design Methodology

- **Variability: Worst Case Methodology based on corner cases**



Thank you for attention!

For more information please visit

http://ems.eit.uni-kl.de

MICROELECTRONIC SYSTEMS DESIGN RESEARCH GROUP