Advanced Topics in Computer Architecture

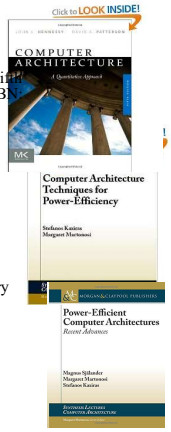
Spring 2023
Tufts University

Instructor: Prof. Mark Hempstead
mark@ece.tufts.edu

Lecture 11: Intro to Power-Aware Computing; Power Models and Metrics

Resources

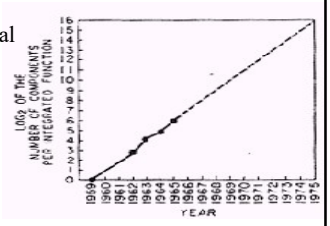
- "Computer Architecture: A Quantitative Approach," Fifth Edition, John L. Hennessy and David A. Patterson, ISBN: 978-0-12-383872-8
- Two Additional References:
  - "Computer Architecture Techniques for Power-Efficiency"
    - By Stefanos Kaxiras, Margaret Martonosi, 2010
  - "Power-Efficient Computer Architectures: Recent Advances"
    - By Magnus Själander, Margaret Martonosi, Stefanos Kaxiras, Dec 2014
  - Part of the Synthesis Lectures on Computer Architecture Series. Available free from the Library (3 concurrent users).
    - http://library.tufts.edu:80/record=b2812045~S1
    - http://library.tufts.edu:80/record=b2812046~S1
- Research papers (will be available on the web)



TRENDS IN MICROPROCESSOR POWER CONSUMPTION

Moore's Law

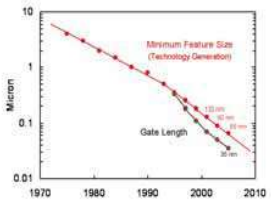
- Gordon Moore's Original observation from 1965 has held consistently since then
- However recently challenges to Moore's law have appeared:
  - Increasing Leakage Power
  - Increasing Power Density (Dark Silicon)
  - Transistor variability
  - Battery life



Electronics, Volume 38, Number 8, April 19, 1965

Dennard Scaling

Device or Circuit Parameter	Scaling Factor
Device dimension *tox, L, W*	1/κ
Doping concentration *Na*	κ
Voltage *V*	1/κ
Current *I*	1/κ
Capacitance *εA/t*	1/κ
Delay time/circuit *VC/I*	1/κ
Power dissipation/circuit *VI*	$1/\kappa^2$
Power density *VI/A*	1



[From Dennard's original paper]

For years transistor scaling was made possible through Dennard Scaling also called constant field scaling. Device dimensions, capacitance ,voltage, power consumption, frequency all scaled equally.

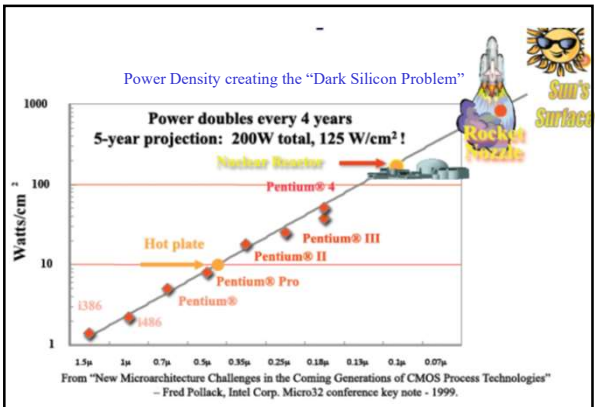
The Triple Play

Using Dennard scaling rules
- Get more transistors, gates (area) $1/\alpha^2$
- Gates get faster, delay scales as α
- Energy per switch is reduced α^3

So we can compute $1/\alpha^3$ as many gate evals/sec
- At the same power and area as the previous design
- Architects take this to improve computer performance

We ignored interconnect!

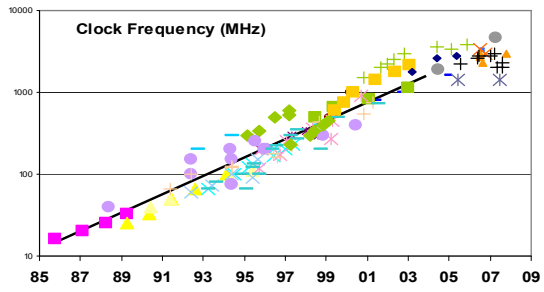
Power Density creating the "Dark Silicon Problem"



[Taylor, DAC and DaSi 2012]

7

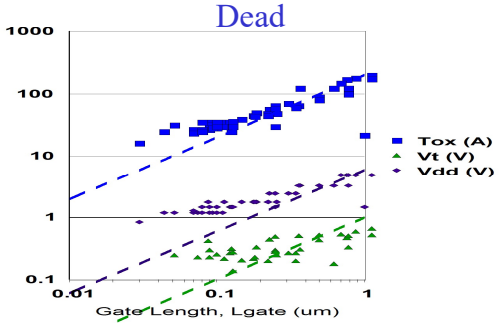
Why Power Increased



EE156/CS140 Mark Hempstead

8

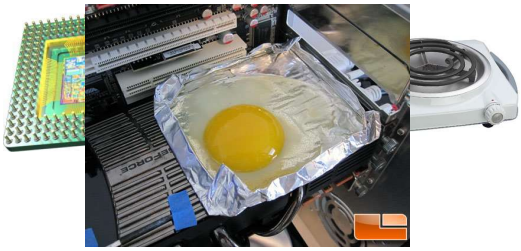
Bad News – Dennard Scaling is Dead



EE156/CS140 Mark Hempstead

9

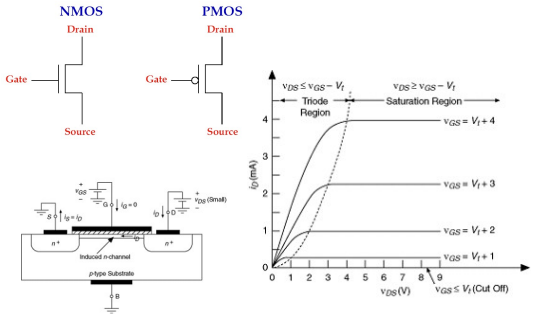
A Microprocessor as a HotPlate



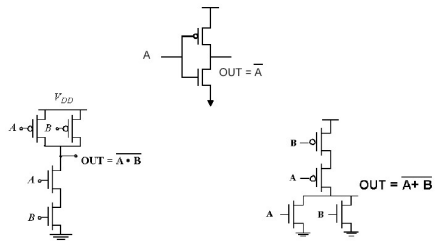
Where is power consumed in a

CIRCUIT PRIMER

Circuit Primer



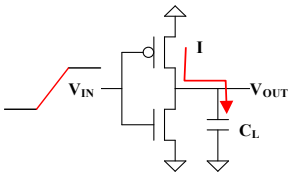
Basic Gates



Power: The Basics

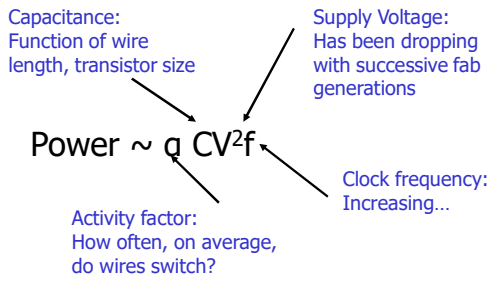
- Dynamic power vs. Static power
  - Dynamic: "switching" power
  - Static: "leakage" power
  - Dynamic power dominates, but static power increasing in importance
  - Trends in each
- Static power: steady, per-cycle energy cost
- Dynamic power: capacitive and short-circuit
- Capacitive power: charging/discharging at transitions from 0→1 and 1→0
- Short-circuit power: power due to brief short-circuit current during transitions.
- Most research focuses on capacitive, but recent work on others

Dynamic (Capacitive) Power Dissipation



- Data dependent – a function of switching activity

Capacitive Power dissipation

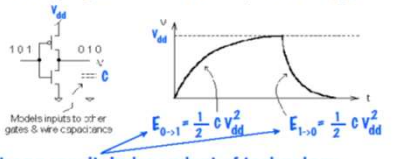


Activity Factor (α)

- Fraction expressing how often a circuit/node or architectural block is active
- Examples:
  - Read α on Cache Port = 0.3 → the cache port is active 30% of the time
  - Nodal α in Adder = 0.2 → the adder is used only 20% of the time
- Architects can reduce dynamic power by reducing α on an architectural block

Switching Energy: Fundamental Physics

Every logic transition dissipates energy.

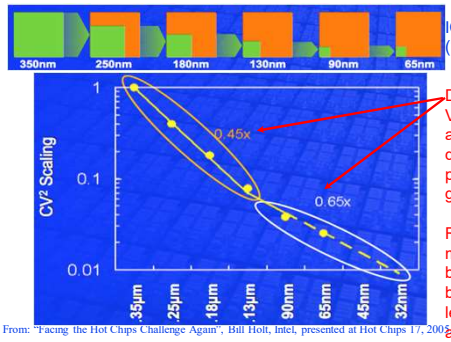


Strong result: Independent of technology.

How can we limit switching energy?

(1) Slow down clock (fewer transitions). But we like speed …
(2) Reduce Vdd. But lowering Vdd limits the clock speed …
(3) Fewer circuits. But more transistors can do more work.
(4) Reduce C per node. One reason why we scale processes.

Scaling switching energy per gate ...



IC process scaling ("Moore's Law")

Due to reducing V and C (length and width of Cs decrease, but plate distance gets smaller).

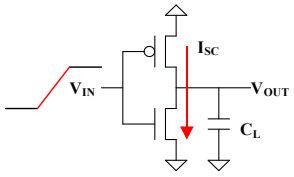
Recent slope more shallow because V is being scaled less aggressively.

From: "Facing the Hot Chips Challenge Again", Bill Holt, Intel, presented at Hot Chips 17, 2005

Lowering Dynamic Power

- Reducing Vdd has a quadratic effect
  - Has a negative (~linear) effect on performance however
- Lowering C_L
  - May improve performance as well
  - Keep transistors small (keeps intrinsic capacitance (gate and diffusion) small)
- Reduce switching activity
  - A function of signal transition stats and clock rate
  - Clock Gating idle units
  - Impacted by logic and architecture decisions

Short-Circuit Power Dissipation



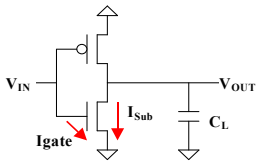
- Short-Circuit Current caused by finite-slope input signals
- Direct Current Path between VDD and GND when both NMOS and PMOS transistors are conducting

Short-Circuit Power Dissipation

$$Power_{SC} \sim t_{sc} V I_{peak}$$

- Power determined by
  - Duration and slope of input signal, t_{sc}
  - I_{peak} determined by transistor sizes, process technology, C_L
- Short circuit power can be minimized
  - Try to match rise/fall times of input and output signals
  - Have not seen many architectural solutions here
  - Good news: relatively, $Power_{SC}$ is shrinking

Leakage Currents



$$I_{DSub} = k \cdot e^{\frac{-q \cdot V_T}{a \cdot k_a \cdot T}}$$

- Subthreshold currents grow exponentially with increases in temperature, decreases in threshold voltage
  - But threshold voltage scaling is key to circuit performance!
- Gate leakage primarily dependent on gate oxide thickness, biases
- Both type of leakage heavily dependent on stacking and input pattern

Lowering Static Power

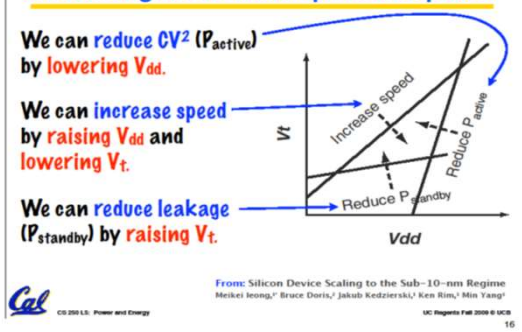
- Design-time Decisions
  - Use fewer, smaller transistors -- stack when possible to minimize contacts with Vdd/Gnd
  - Multithreshold process technology (multiple oxides too!)
    - Use "high-Vt" slow transistors whenever possible
- Dynamic Techniques
  - Reverse-Body Bias (dynamically adjust threshold)
    - Low-leakage sleep mode (maintain state), e.g. XScale
  - Vdd-gating (Cut voltage/gnd connection to circuits)
    - Near zero-leakage sleep mode
    - Lose state, overheads to enable/disable

Device engineers trade speed and power

We can reduce CV^2 (P_{active}) by lowering V_{dd}.

We can increase speed by raising V_{dd} and lowering V_t.

We can reduce leakage ($P_{standby}$) by raising V_t.



From: Silicon Device Scaling to the Sub-10-nm Regime
Meikei Ieong,[1] Bruce Doris,[1] Jakub Kedzierski,[1] Ken Rim,[1] Min Yang[1]

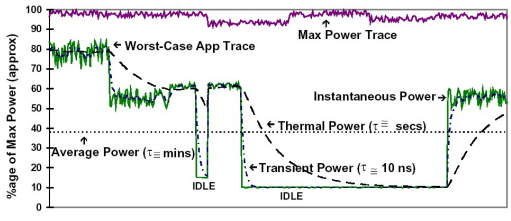
CS 250 L5: Power and Energy

UC Regents Fall 2008 © UCB

16

METRICS

What do we mean by Power?



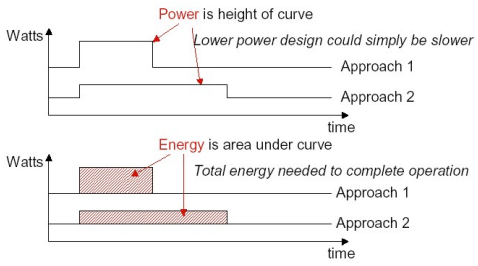
- Max Power: Artificial code generating max CPU activity
- Worst-case App Trace: *Practical* applications worst-case
- Thermal Power: Running average of worst-case app power over a time period corresponding to thermal time constant
- Average Power: Long-term average of typical apps (minutes)
- Transient Power: Variability in power consumption for supply net

27

Power vs. Energy

- Power consumption in Watts
  - Determines battery life in hours
  - Sets packaging limits
- Energy efficiency in joules
  - Rate at which energy is consumed over time
  - Energy = power * delay (joules = watts * seconds)
  - Lower energy number means less power to perform a computation at same frequency

Power vs. Energy



Power vs. Energy

- Power-delay Product (PDP) = P_{avg} * t
  - PDP is the average energy consumed per switching event
- Energy-delay Product (EDP) = PDP * t
  - Takes into account that one can trade increased delay for lower energy/operation
- Energy-delay^2 Product (EDDP) = EDP * t
  - Why do we need so many formulas?!!?
  - We want a voltage-invariant efficiency metric! Why?
  - Power ~ ½ CV^2Af, Performance ~ f (and V)

E vs. EDP vs. ED^2P

- Power ~ CV^2f ~ V^3 (fixed microarch/design)
- Performance ~ f ~ V (fixed microarch/design)
- (For the nominal voltage range, f varies approx. linearly with V)
- Comparing processors that can only use freq/voltage scaling as the primary method of power control:
  - $(perf)^3$ / power, or $MIPS^3$ / W or $SPEC^3$ /W is a fair metric to compare energy efficiencies.
  - This is an ED^2P metric. We could also use: $(CPI)^3$ * W for a given application

E vs. EDP vs. ED^2P

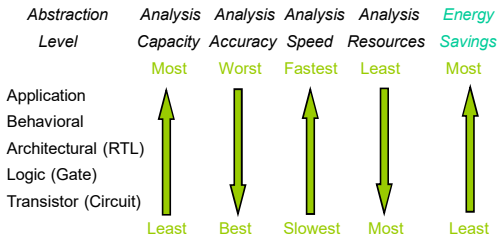
- Currently have a processor design:
  - 80W, 1 BIPS, 1.5V, 1GHz
  - Want to reduce power, willing to lose some performance
  - Cache Optimization:
    - IPC decreases by 10%, reduces power by 20% => Final Processor: 900 MIPS, 64W
    - Relative E = MIPS/W (higher is better) = 14/12.5 = 1.125x
  - Energy is better, but is this a "better" processor?

Not necessarily

- 80W, 1 BIPS, 1.5V, 1GHz
  - Cache Optimization:
    - IPC decreases by 10%, reduces power by 20% => Final Processor: 900 MIPS, 64W
    - Relative E = MIPS/W (higher is better) = 14/12.5 = 1.125x
    - Relative EDP = $MIPS^2/W$ = 1.01x
    - Relative ED^2P = $MIPS^3/W$ = .911x
- What if we just adjust frequency/voltage on processor?
  - How to reduce power by 20%?
  - $P = CV^2F = CV^3$ => Drop voltage by 7% (and also Freq) => .93*.93*.93 = .8x
  - So for equal power (64W)
    - Cache Optimization = 900MIPS
    - Simple Voltage/Frequency Scaling = 930MIPS

MODELING

Modeling: Analysis Abstraction Levels



Power/Performance abstractions

- Low-level:
  - Hspice
  - PowerMill
- Medium-Level:
  - RTL Models
- Architecture-level:
  - PennState SimplePower
  - Intel Tempest
  - Princeton Wattch
  - IBM PowerTimer
  - Umich/Colorado PowerAnalyzer
  - MV5
  - HP Labs McPat
- System Level
  - Fuel Gauge (Android)
  - PowerTutor

Low-level models: Hspice

- Extracted netlists from circuit/layout descriptions
  - Diffusion, gate, and wiring capacitance is modeled
- Analog simulation performed
  - Detailed device models used
  - Large systems of equations are solved
  - Can estimate dynamic and leakage power dissipation within a few percent
  - Slow, only practical for 10-100K transistors
- PrimeTime power and HSIM from (Synopsys)

BSIM Model

$$I_{leak} = \mu_0 C_{OX} \frac{W}{L} e^{a+b*(V_{dd}-V_{dd0})} v_t^2 \left(1 - e^{\frac{-V_{dd}}{v_t}}\right) \exp\left(\frac{-|V_{th0}| - V_{off}}{n \cdot v_t}\right)$$

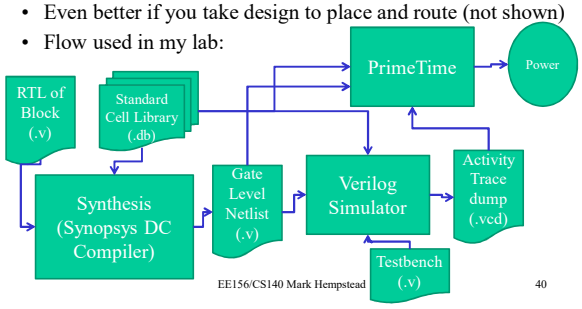
- Industry Standard model used in HSPICE simulations
- http://bsim.berkeley.edu/
- Circuit level model of a transistor in a particular technology
- BSIM4 – includes gate leakage BSIM v3.3 does not
- Newest BSIM6.1 model released in 2015
- Other models for SOI, and common/multi-gate (e.g. FinFET)

Medium-level models: RTL

- Logic simulation obtains switching events for every signal
- Structural VHDL or verilog with zero or unit-delay timing models
- Capacitance estimates performed
  - Device Capacitance
    - Gate sizing estimates performed, similar to synthesis
  - Wiring Capacitance
    - Wire load estimates performed, similar to placement and routing
- Switching event and capacitance estimates provide dynamic power estimates

RTL Based Estimation Flow

- If you have RTL (Verilog/VHDL) you can estimate power of a block with excellent accuracy
- Even better if you take design to place and route (not shown)
- Flow used in my lab:



Architecture level models

- Two major classes:
  - Cycle/Event-Based: Arch. Level power models interfaced with cycle-driven performance simulation
  - Instruction-Based: Measurement/Characterization based on instruction usage and interactions
- Components of Arch. Level power model
  - Could be based on ckt schematic measurements/extrapolation

  Or…
  - Capacitance models

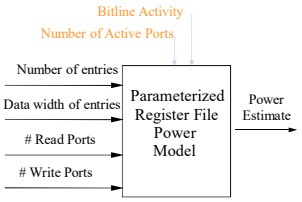
  Both may need to consider…
  - Circuit design styles
  - Clock gating styles & Unit usage statistics
  - Signal transition statistics

Watch: An Overview

Watch's Design Goals
- Flexibility
- Planning-stage info
- Speed
- Modularity
- Reasonable accuracy

Overview of Features
- Parameterized models for different CPU units
  - Can vary size or design style as needed
- Abstract signal transition models for speed
  - Can select different conditional clocking and input transition models as needed
- Based on SimpleScalar
- Modular: Can add new models for new units studied

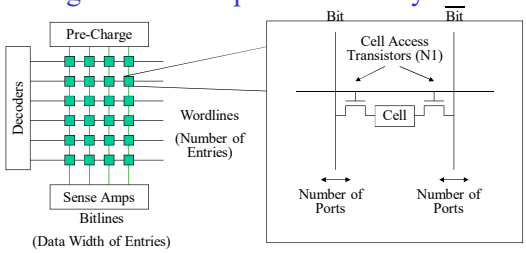
Unit Modeling



Modeling Capacitance
- Models depend on structure, bitwidth, design style, etc.
- E.g., may model capacitance of a register file with bitwidth & number of ports as input parameters

Modeling Activity Factor
- Use cycle-level simulator to determine number and type of accesses
  - reads, writes, how many ports
- Abstract model of bitline activity

Register File: Capacitance Analysis



$$C_{wordline} = C_{diffcapWordlineDriver} + NumberBitlines * C_{gatecapN1} + Wordlinelength * C_{metal}$$

$$C_{bitline} = C_{diffcapPchg} + NumberWordlines * C_{diffcapN1} + Bitlinelength * C_{metal}$$

Register File Model: Validation

Error Rates	*Gate*	*Diff*	*InterConn.*	*Total*
Wordline(r)	1.11	0.79	15.06	8.02
Wordline(w)	-6.37	0.79	-10.68	-7.99
Bitline(r)	2.82	-10.58	-19.59	-10.91
Bitline(w)	-10.96	-10.60	7.98	-5.96

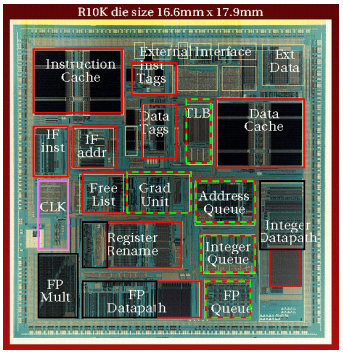
(Numbers in Percent)

- Validated against a register file schematic used in Intel's Merced design
- Compared capacitance values with estimates from a layout-level Intel tool
- Interconnect capacitance had largest errors
  - Model currently neglects poly connections
  - Differences in wire lengths -- difficult to tell wire distances of schematic nodes

One Cycle in Wattch

	Fetch	Dispatch	Issue/Execute	Writeback/ Commit
Power (Units Accessed)	• I-cache • Bpred	• Rename Table • Inst. Window • Reg. File	• Inst. Window • Reg File • ALU • D-Cache • Load/St Q	• Result Bus • Reg File • Bpred
Performance	• Cache Hit? • Bpred Lookup?	• Inst. Window Full?	• Dependencies Satisfied? • Resources?	• Commit Bandwidth?

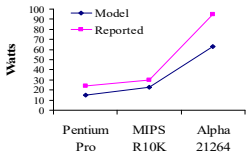
- On each cycle:
  - determine which units are accessed
  - model execution time issues
  - model per-unit energy/power based on which units used and how many ports.



Units Modeled by Wattch

- **Array Structures**
  - Caches, Reg Files, Map/Bpred tables
- **Content-Addressable Memories (CAMs)**
  - TLBs, Issue Queue, Reorder Buffer
- **Complex combinational blocks**
  - ALUs, Dependency Check
- **Clocking network**
  - Global Clock Drivers, Local Buffers

Wattch accuracy



Relative Wattch estimates track well even in cases where absolute accuracy falls short.

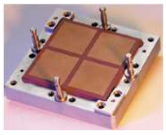
Hardware Structure	*Intel Data*	*Wattch*
Instruction Fetch	22%	21%
Register Alias Table	6%	5%
Reservation Stations	8%	9%
Reorder Buffer	11%	12%
Integer Exec. Unit	15%	15%
Data Cache Unit	11%	11%
Memory Order Buffer	6%	5%
Floating Point Exec. Unit	8%	8%
Global Clock	8%	10%
Branch Target Buffer	5%	4%

Typically 10-15% relative accuracy as compared to low-level industry data.

Wattch Simulation Speed

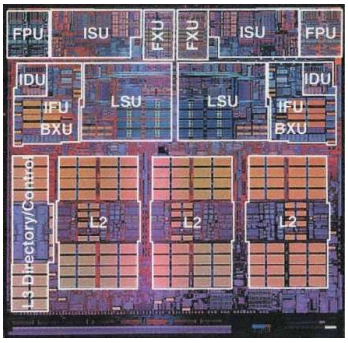
- Roughly 80K instructions per second (PII-450 host)
- ~30% overhead compared to performance simulation alone
  - Could be decreased if power estimates are not computed every cycle
- Many orders of magnitude faster than lower-level approaches
  - For example, PowerMill takes ~1hour to simulate 100 test vectors on a 64-bit adder

IBM Power 4: How does die heat up?

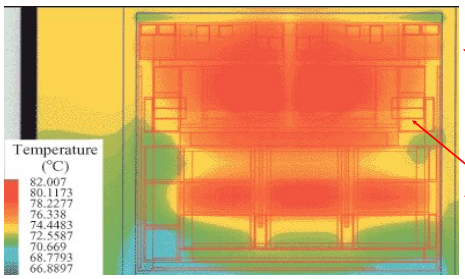


4 dies on a multi-chip module

2 CPUs per die →



115 Watts: Concentrated in "hot spots"



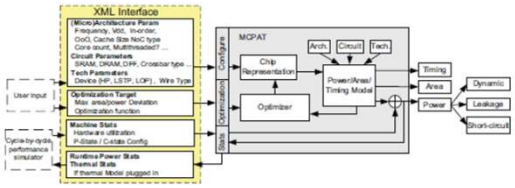
Hot spots

Fixed point units

Cache logic

66.8 C == 152 F 82 C == 179.6 F

McPAT Power Model

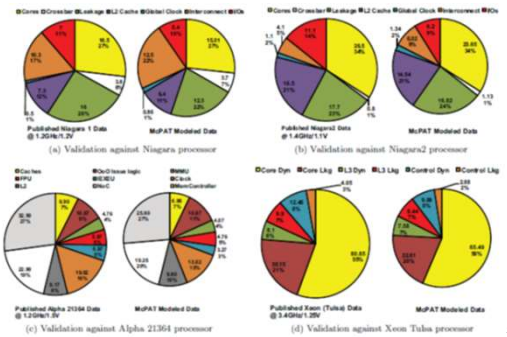


"McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures" MICRO 2009

McPAT

- Developed by HP Labs and is available for download online
  - http://www.hpl.hp.com/research/mcpat/
- integrated power, area, and timing modeling framework
- early stage design space exploration for multicore and manycore processor configurations ranging from 90nm to 22nm and beyond

Validation



CACTI – modeling SRAMS

- HP parameterized model register files, rams and caches:
  http://www.hpl.hp.com/research/cacti/
- http://www.hpl.hp.com/techreports/2009/HPL-2009-85.pdf
  - Version 6.5 now available as a download
  - Version 5.3 is available as a web interface.
- Used in many microarchitecture Power-Models, Ex: McPAT and Orion

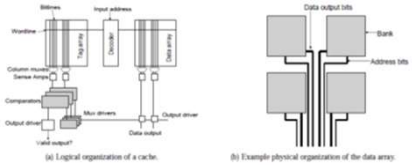


Figure 1. Logical and physical organization of the cache (from CACTI 3.0 [13]).

CACTI must model wires

- Wires and repeater spacing is important

$$\tau = \left(\frac{1}{l} r_s (c_0 + c_p) + \frac{r_s}{s} C_{wire} + R_{wire} s c_0 + 0.5 R_{wire} C_{wire} l\right)$$

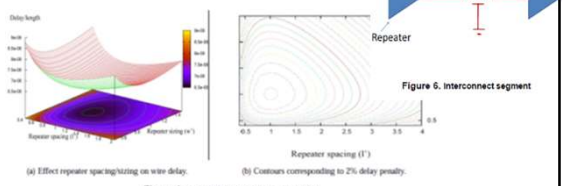


Figure 4. Repeater overhead vs wire delay

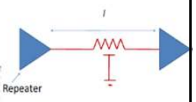


Figure 6. Interconnect segment

Measuring Power Consumption

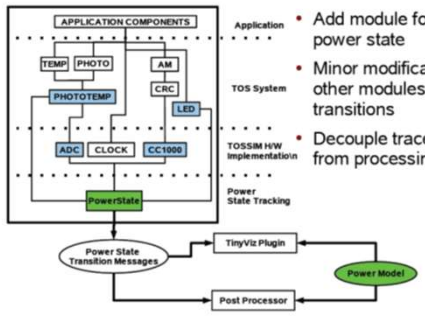
- Empirical measurement is often preferred for its speed and accuracy.
- Limitations:
  - Specific to one device
  - Specific to one technology
  - Requires silicon (not for early stage exploration)
  - Hard to isolate individual micro architecture or system components

Option: Measure with Microbenchmarks

- Microbenchmarks: small snippets of code that exercise one component at a time
- Use a suite of microbenchmarks to build a power model
- Write logger code that tracks state changes in an application
- Example PowerTOSSIM for WSN (SenSys'04) http://www.eecs.harvard.edu/~shnayder/ptossim/

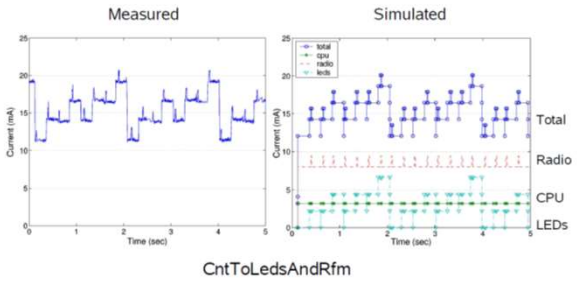
CPU Mode	Current @3V		Radio Mode	Current @3V
Active	8.0 mA		Receive	7.0 mA
Idle	3.2 mA		Transmit Min Power	3.7 mA
Standby	216 µA		Transmit Max Power	21.5 mA
Power-save	110 µA		**Sensor Board**	0.7 mA

PowerTOSSIM Architecture



- Add module for tracking power state
- Minor modifications to other modules to report transitions
- Decouple trace gathering from processing

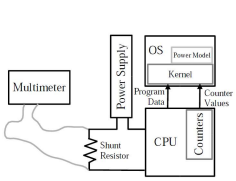
Accuracy

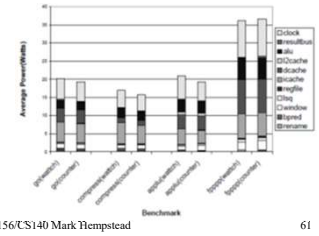


CntToLedsAndRfm

Option2: Use Hardware Performance Counters

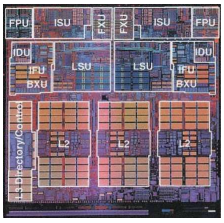
- Extracting microarchitectural detail using microbenchmarks might not be possible
- Joseph et al. (ISLPED'01) showed you can use performance counters to estimate a Pentium Pro
- Contreras and Martonosi showed a newer technique for the XScale in ISLPED'05
- Intel's Sandybridge has a new "energy counter"





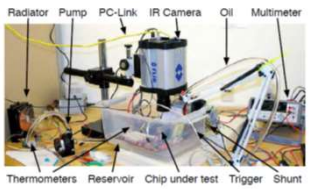
Floorplan (Area) based estimates

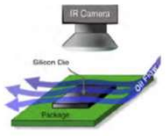
- If the switching factors are similar you can estimate power based on a die photo
- Divide total power measured by area fraction
- However, some structures have different power densities due to different switching rates (logic vs. cache)



Thermal Based Estimates

- If you have an expensive infrared (IR) camera you can find the hotspots on the chip and partition the total power based different temperatures
- The setup is tricky, must keep chip cool (oil is often used) [Mesa-Martinez ISCA'07]





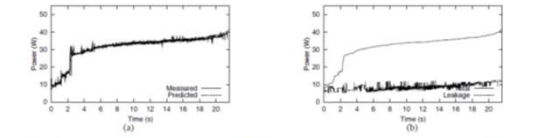


Figure 7: Power prediction accuracy summary for all the applications (a); power breakdown summary for all the applications (b).

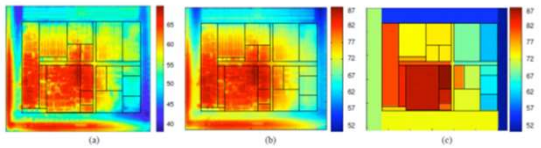


Figure 8: Temperature with overlapped floorplan (a); Hottest captured frame (b); and its average temperature per block (c).

Analytical Models

- Amdahl's Law

$$S(N) = \frac{1}{(1-P) + \frac{P}{N}}$$

- Others leverage these models to make predictions
  - Such as the Dark Silicon Paper
  - Much require populating best on empirical data

Aside: Pareto-Frontier

Example of a Pareto frontier. The boxed points represent feasible choices, and smaller values are preferred to larger ones. Point *C* is not on the Pareto Frontier because it is dominated by both point *A* and point *B*. Points *A* and *B* are not strictly dominated by any other, and hence do lie on the frontier.

(You will often see Pareto Frontiers used for modeling)