Sam Naffziger
AMD Senior Fellow

AMD Smarter Choice

High Performance Processors in a Power Limited World

Outline

Today's processor design landscape

- Trends

Issues making designer's lives difficult

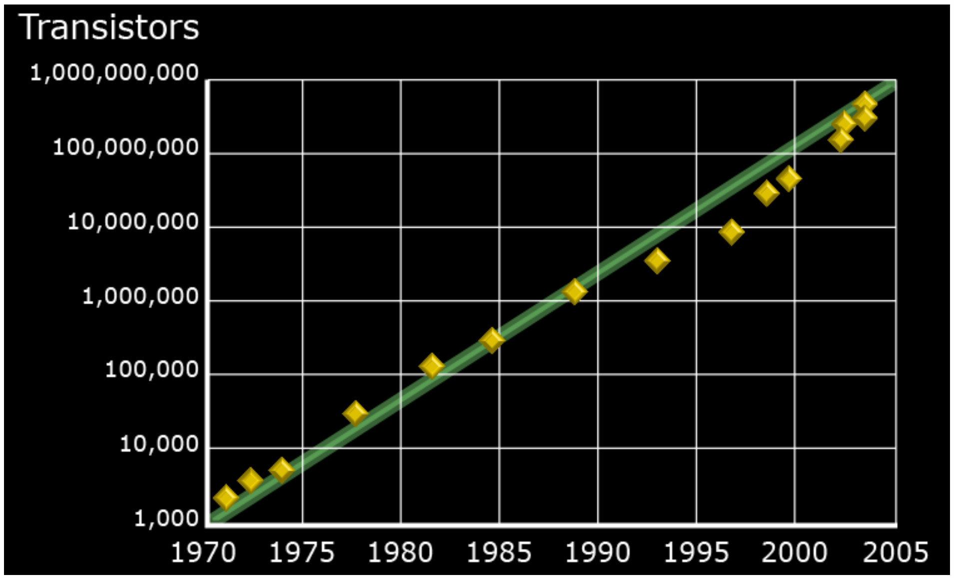
- Power limits
- Scaling effects

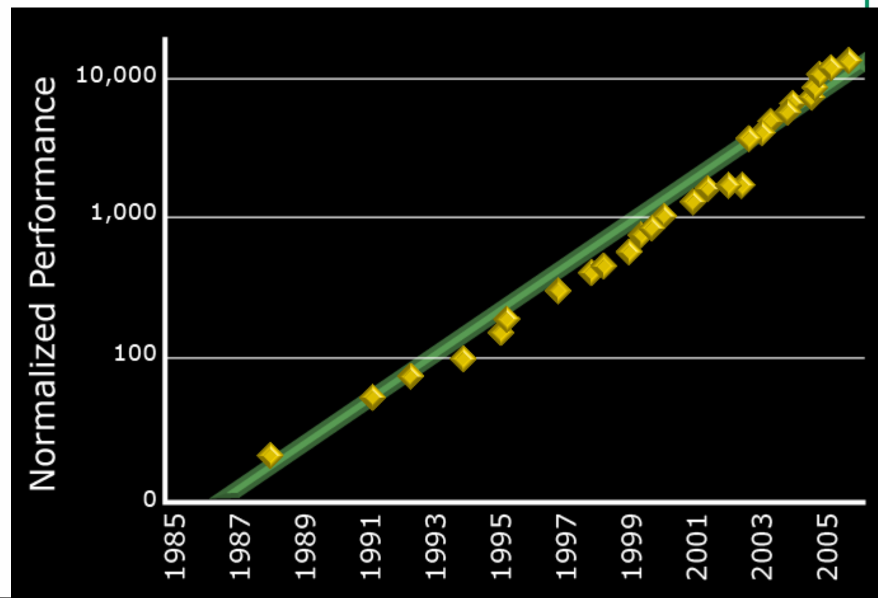
Design opportunities

- Circuit level
- Architectural

Summary

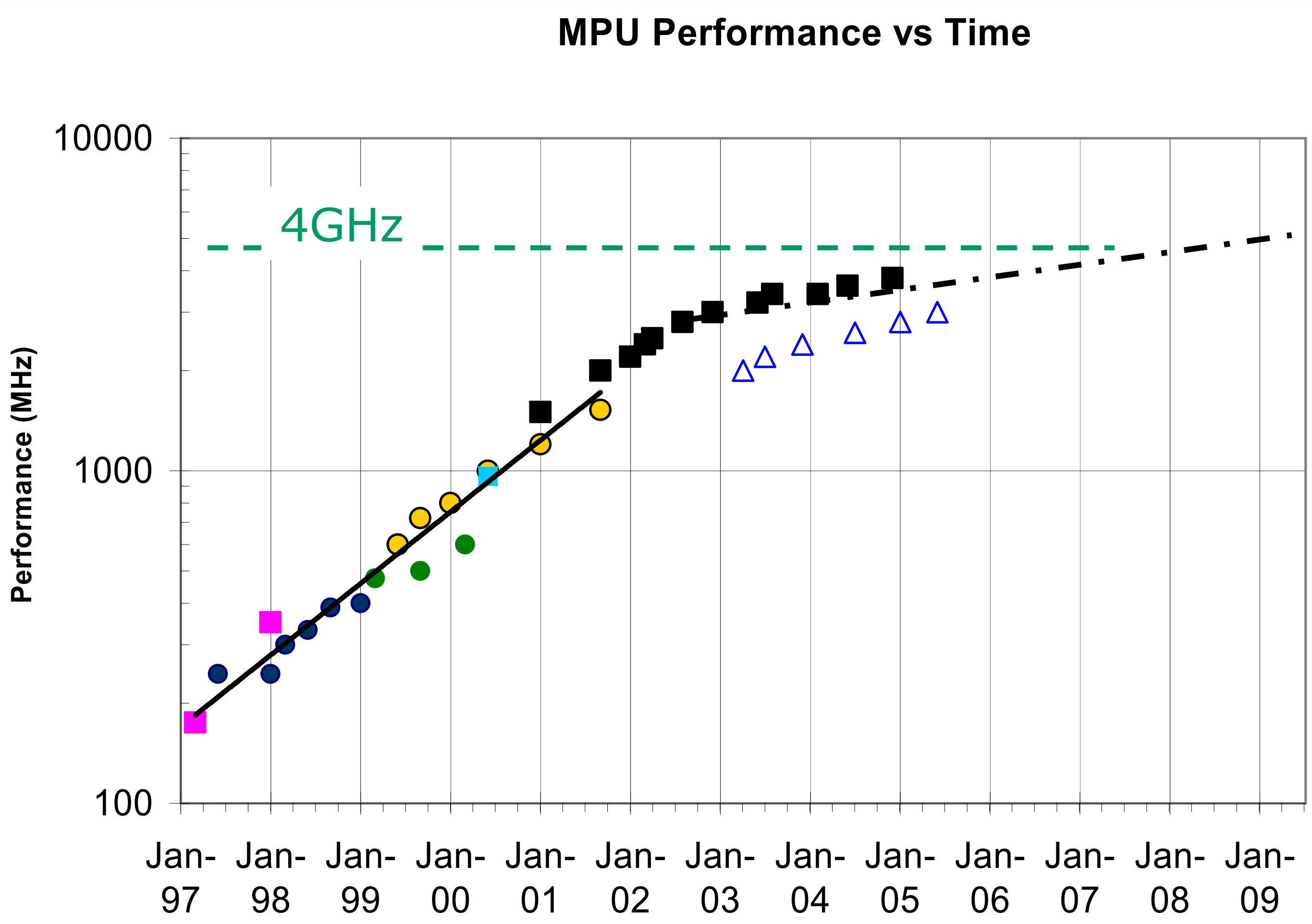
The All Consuming Quest for Greater Performance at Lower Cost





Moore's Law has served us well.

Processor Frequency vs. Time



The amazing frequency increases of the past decade have leveled off – Why? **Power Limits** **Process Issues**

Outline

Today's processor design landscape

- Trends

Issues making designer's lives difficult

- **Power limits**
- Scaling effects

Design opportunities

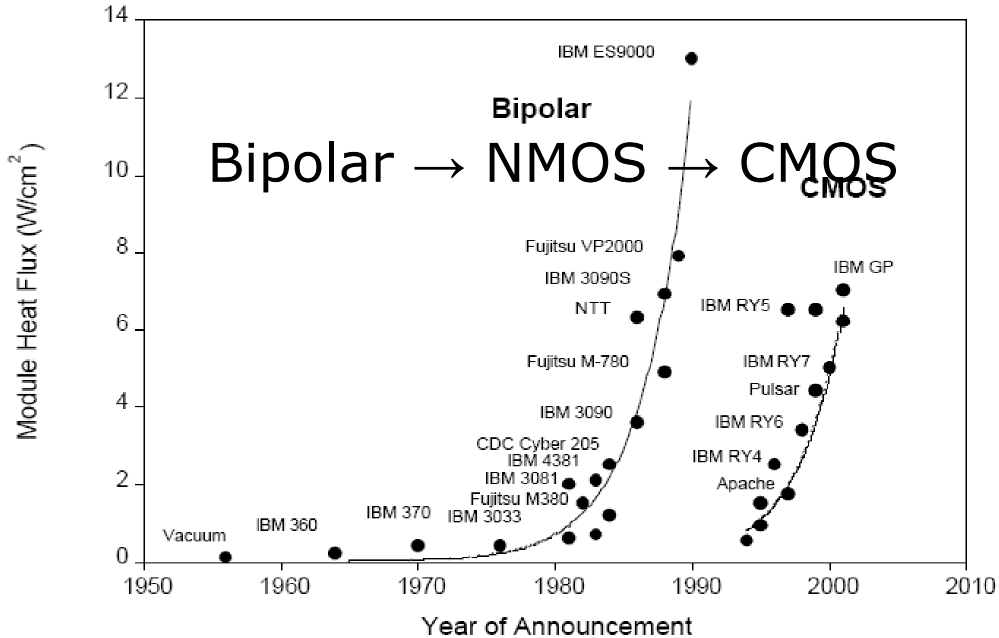
- Circuit level
- Architectural

Summary

Power Consumption Background

"Will it be possible to remove the heat generated by 10's of thousands of components?"

G. Moore, *Cramming more components onto integrated circuits*, Electronics, Volume 38, Number 8, April 19, **1965**



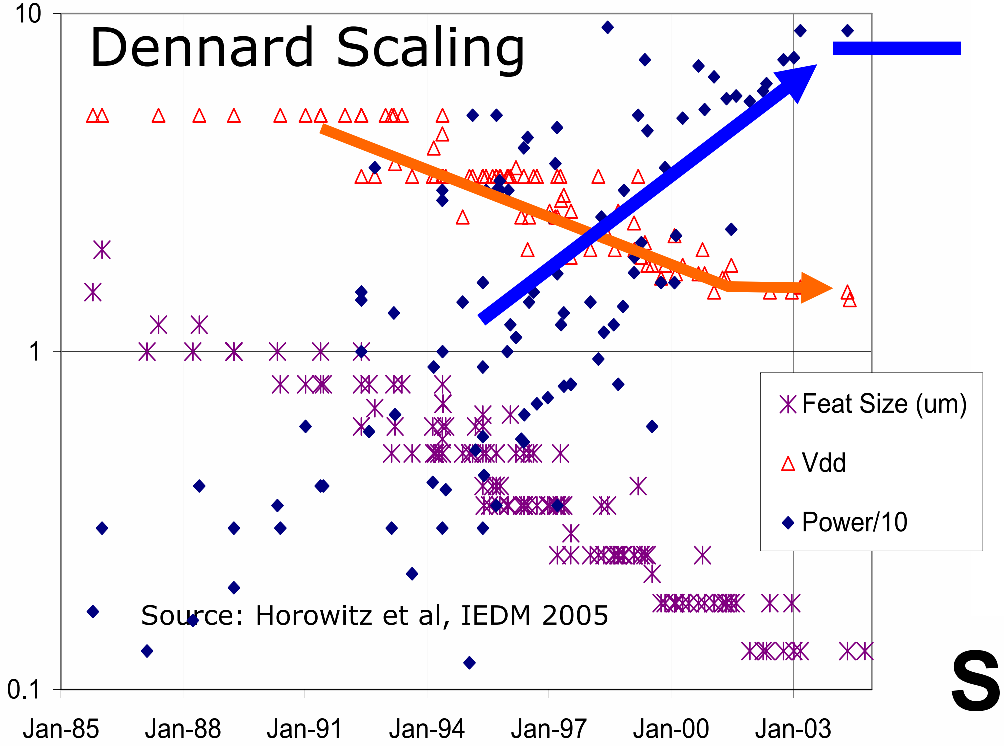
(Ghoshal and Schmidt)

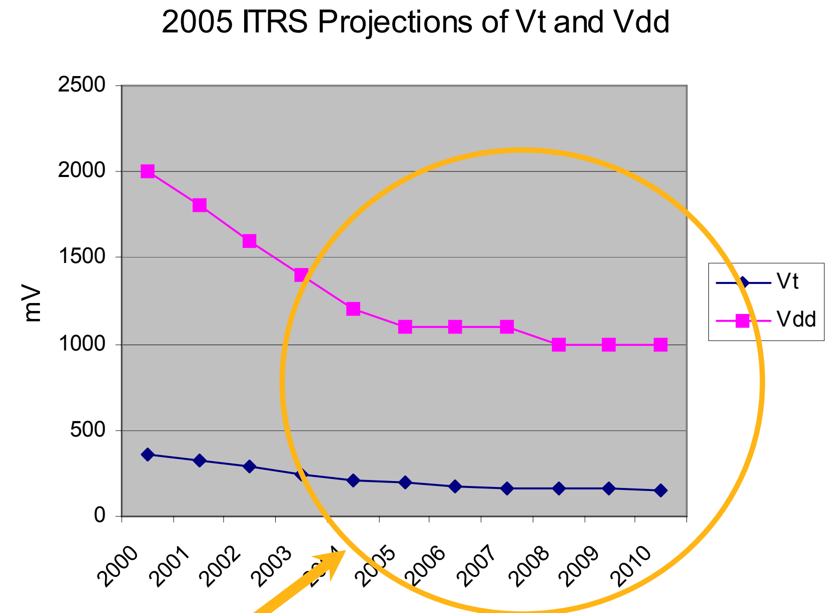
Power has always challenged circuit integration

We've been bailed out by technology in the past

Scaling Background

Realistic power limit





Scaling doesn't bail us out any more

Power Consumption Background

- **Reducing Vdd**
- **Reducing C_{TOT}**
- **Reducing I_{LEAK}, I_{CO}**
- **Reducing α**

The Process guys have had the biggest impact on these

But now, not only are those improvements fading, but we have a host of new challenges

- **Variation**
- **Voltage droop**
- **Wire non-scaling**

The Processor Designer

Switching Power — Crossover Power — Leakage Power

$$P \approx \underbrace{C_{TOT} \cdot \alpha \cdot F \cdot Vdd^2}_{\text{Switching Power}} + \underbrace{N_{TOT} \cdot \alpha \cdot F \cdot Vdd \cdot I_{CO}}_{\text{Crossover Power}} + \underbrace{N_{ON} \cdot I_{LEAK} \cdot Vdd}_{\text{Leakage Power}}$$

Outline

Today's processor design landscape

- Trends

Issues making designer's lives difficult

- Power limits
- **Scaling effects**

Design opportunities

- Circuit level
- Architectural

Summary

The Silicon Age Still on a Roll, But …

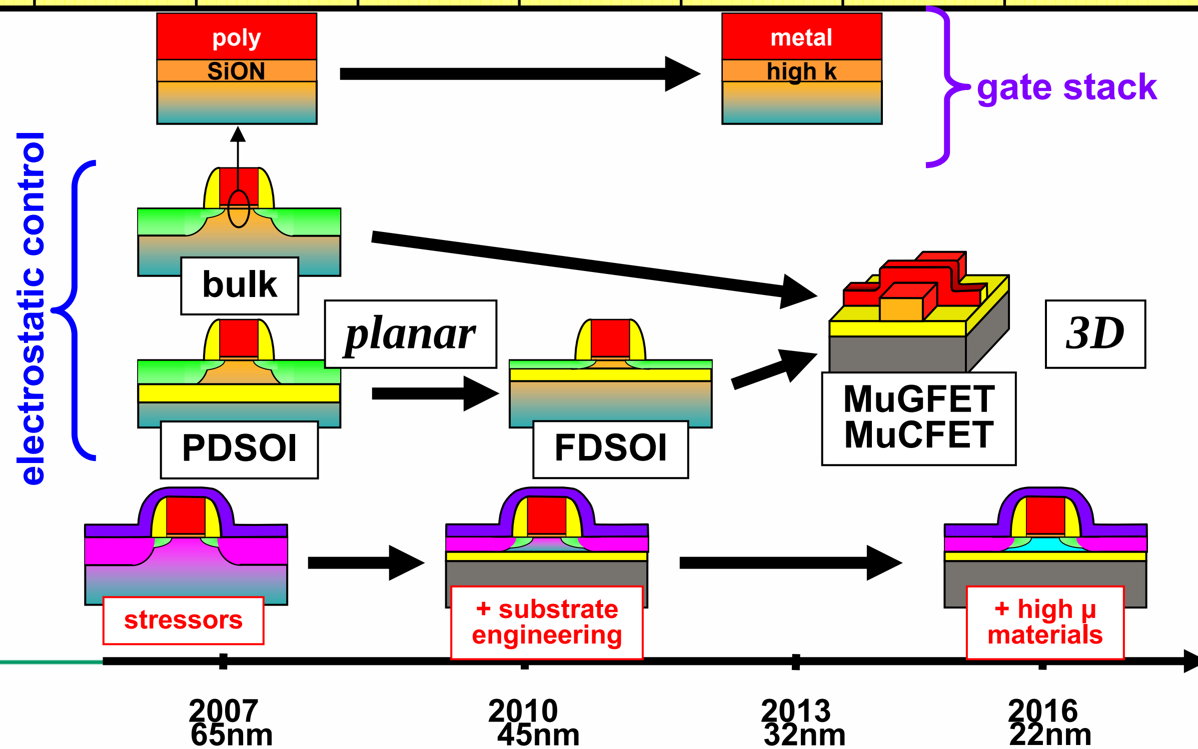
AMD Smarter Choice

	2004	2006	2008	2010	2012	2014	2016	2018
High Volume Manufacturing	2004	2006	2008	2010	2012	2014	2016	2018
Technology Node (nm)	90	65	45	32	22	16	11	8
Delay = CV/I scaling	0.7	~0.7	>0.7	Delay scaling will slow down				
Energy/Logic Op scaling	>0.35	>0.5	>0.5	Energy scaling will slow down				
Bulk Planar CMOS	High Probability					Low Probability		
Alternate, 3G etc	Low Probability					High Probability		
Variability	Medium			High		Very High		
RC Delay	1	1	1	1	1	1	1	1

But … all this scaling has some nasty side effects

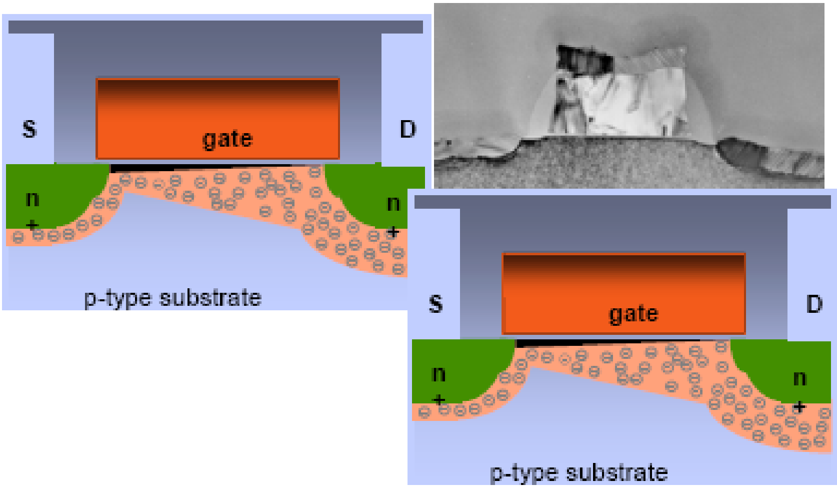
ITRS Roadmap

Source: European Nanoelectronics Initiative Advisory Council (ENIAC)



Device Variation Reverse Scales

Accuracy in 0.25 μm CMOS



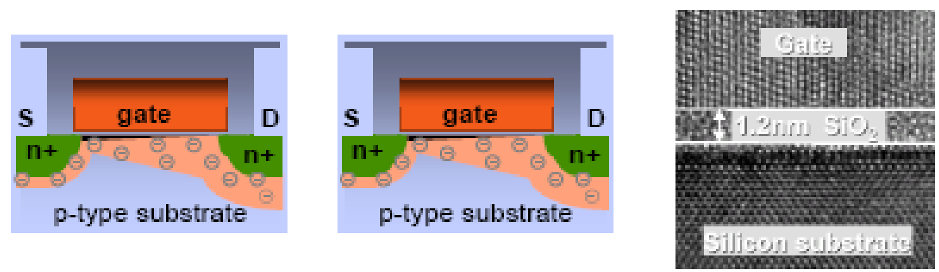
Granularity on molecular level is reached:
0.25/0.25 transistor = 1200 doping atoms

$V_T \propto 1200$

$\sigma_{\Delta VT} \propto \sqrt{1200} \approx 3\% V_T$

Source: Pelgrom, IEEE lecture 5/11/06

The Problem:
Atoms don’t scale



Intel

Granularity on molecular level is reached:
0.1/0.065 transistor = 60-80 doping atoms in depletion region

$V_T \propto 80$

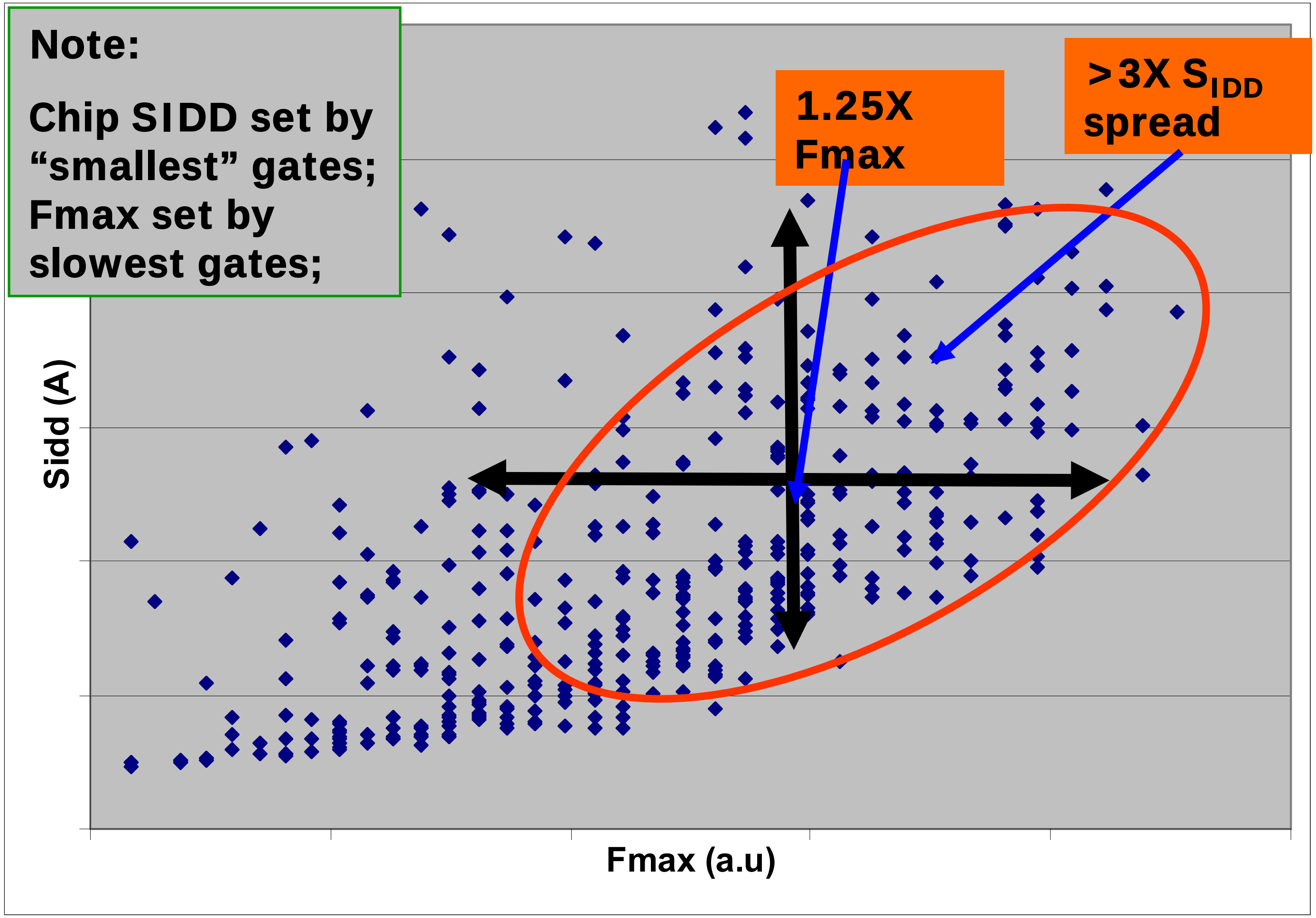
$\sigma_{\Delta VT} \propto \sqrt{80} \approx 11\% V_T$

Variations subtract directly off cycle time

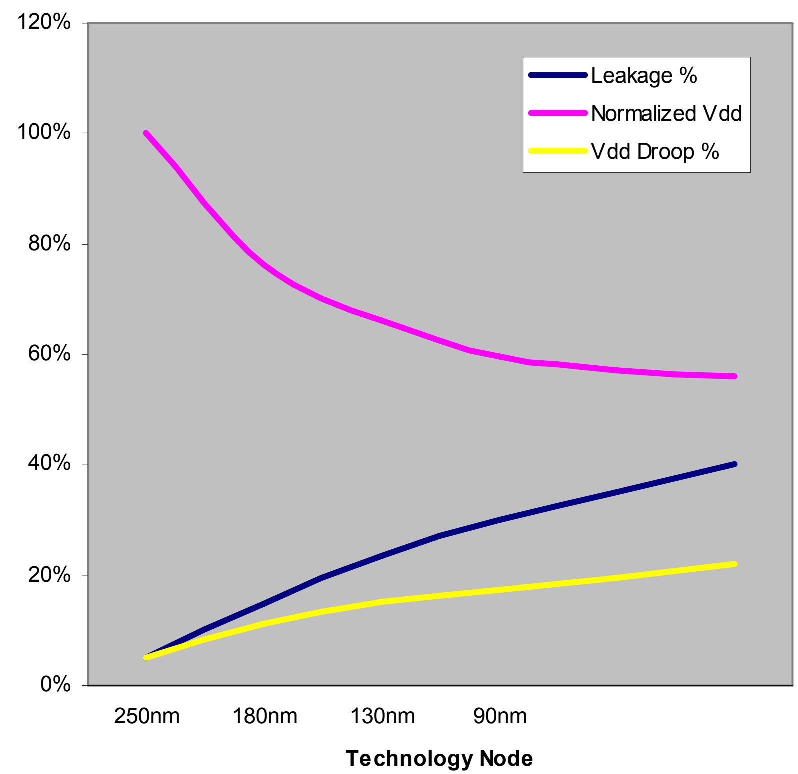
➔ power efficiency drops

➔ Circuit margins degrade

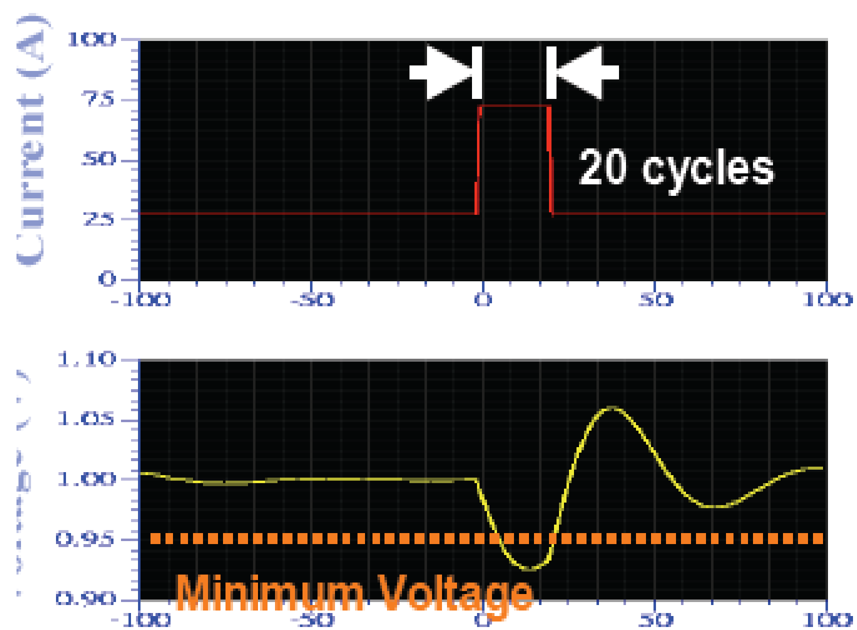
One impact of variation is leakage spreads



Scaling Intrinsically Hurts Supply Integrity



Di/Dt (Vdd/Gnd Bounce)



Source: Bose, Hotchips 17

With power per core staying constant but area, voltage and cycle times dropping, we have a big challenge

Requiring a higher voltage to hit frequency is a quadratic power impact

Outline

Today's processor design landscape

- Trends

Issues making designer's lives difficult

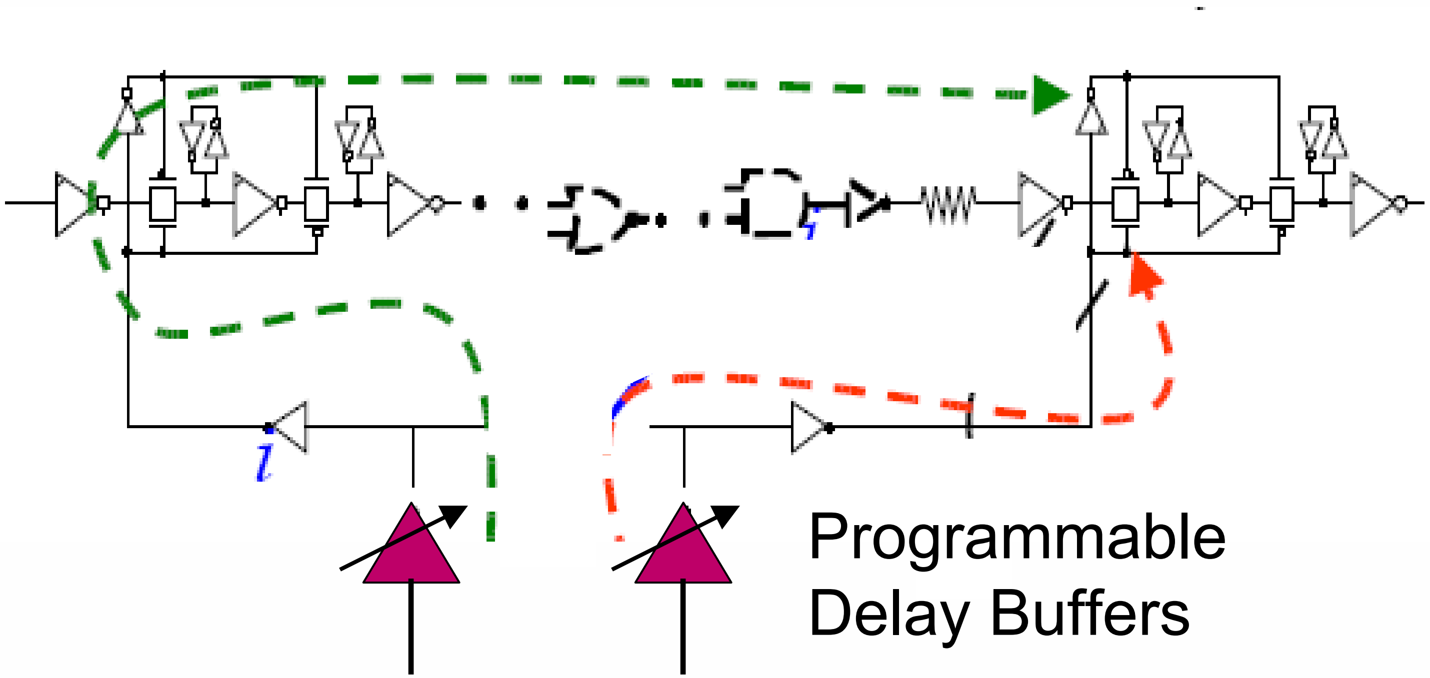
- Power limits
- Scaling effects

Design opportunities

- **Circuit level**
- Architectural

Summary

Some Ways to Shoulder the Variation Burden: Adaptive clocking



Empirically set the clock edge to optimize frequency

Higher granularity → more variation tolerance

L_{BIST} and GA search algorithms show promise for per-part optimization

Some Ways to Shoulder the Variation Burden: Self Healing Designs

Simplest example is cache ECC on memory arrays

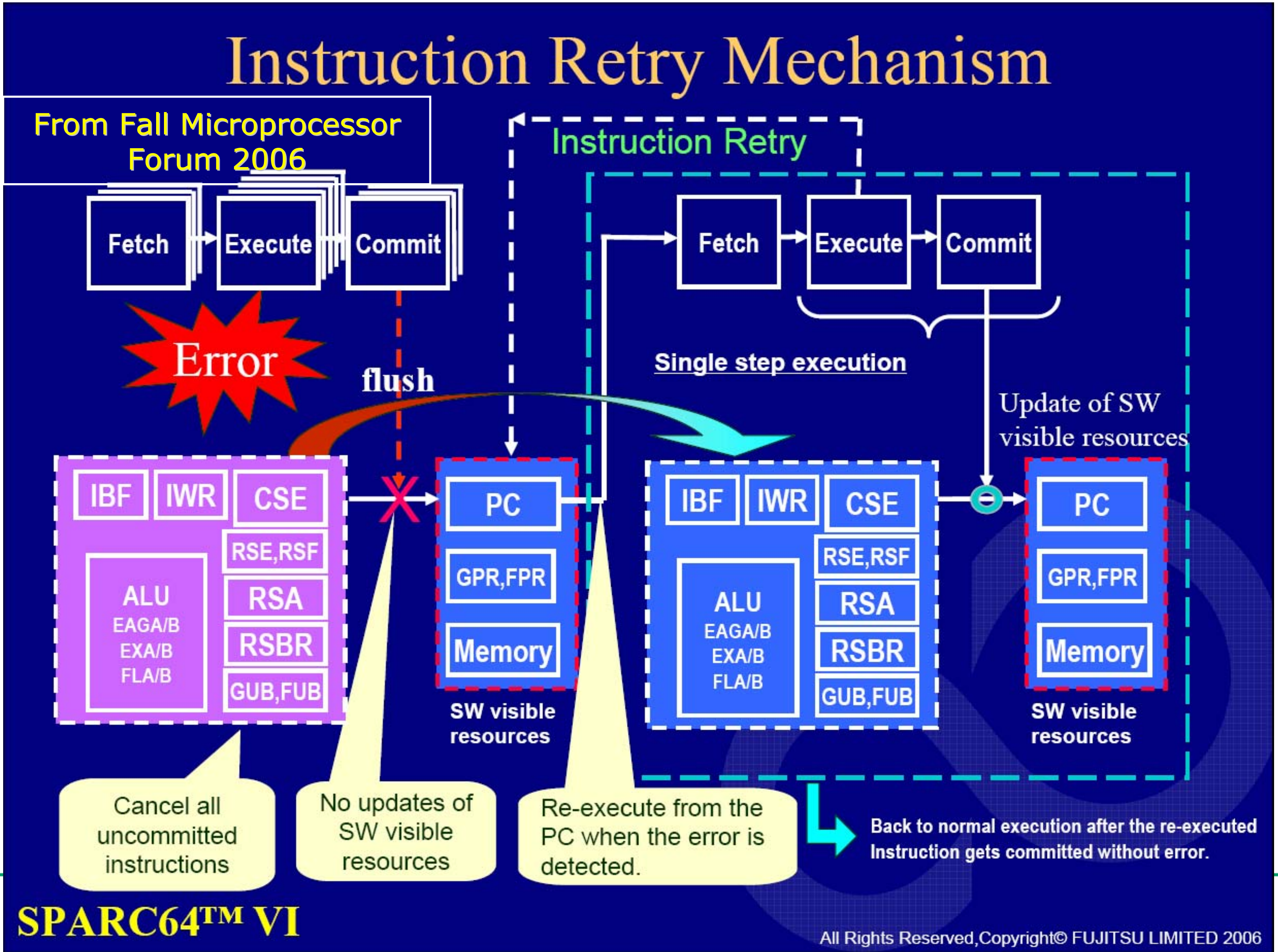
Next level is Intel's Pellston technology implemented on Montecito and Tulsa

- Disable defective lines detected by multiple ECC errors

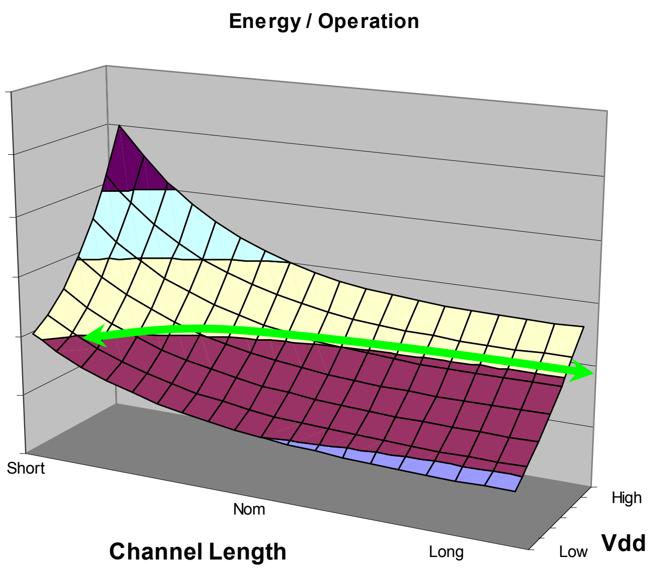
Future directions involve self-checking with redundant logic and retry

- Predict result through parity, residues or redundant logic
- On an error, replay calculation before committing architectural state
- If replay correct, it was a transient error (particle strike, Vdd droop, random noise coupling etc.)
- If incorrect can reduce frequency, increase voltage or retry with an alternate execution path

Some Ways to Shoulder the Variation Burden: Self Healing Designs

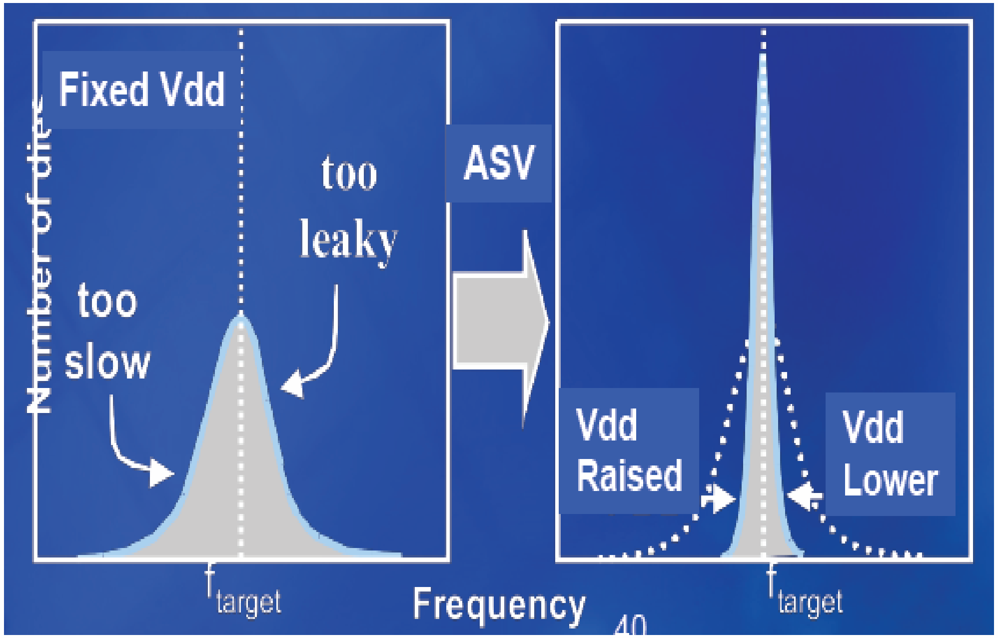


Adaptive Supply Voltage



Per-part and dynamic voltage management are key

More range flexibility and finer grain response will provide differentiation

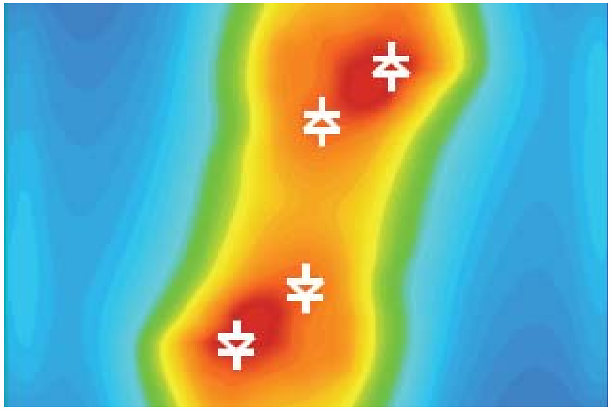


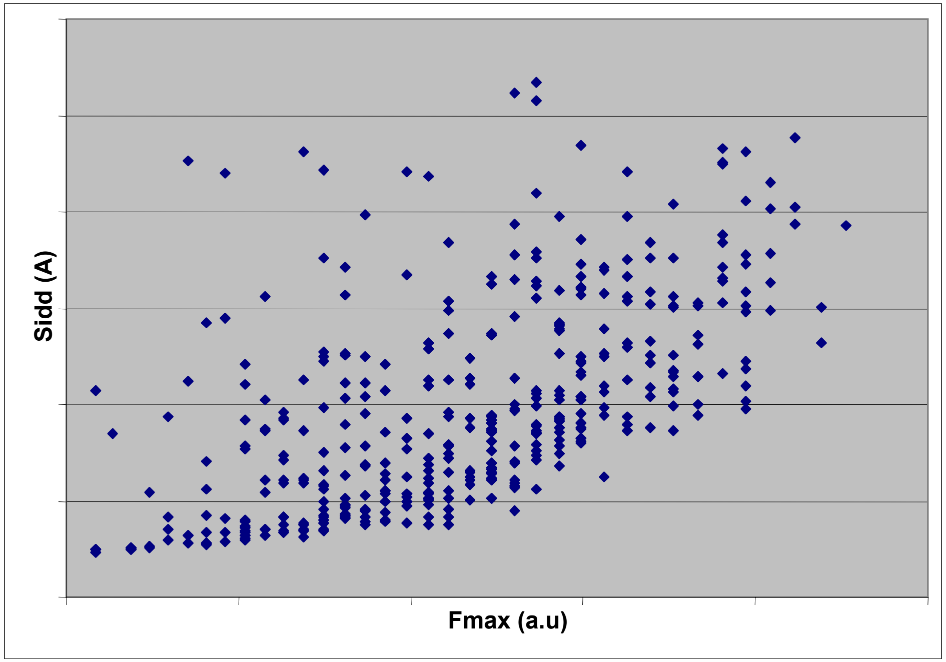
Integrated Power and Thermal Management

"Fuse and forget" is no longer viable

Too much variation in environment, manufacturing and operating conditions

Some means of dynamic optimization needed



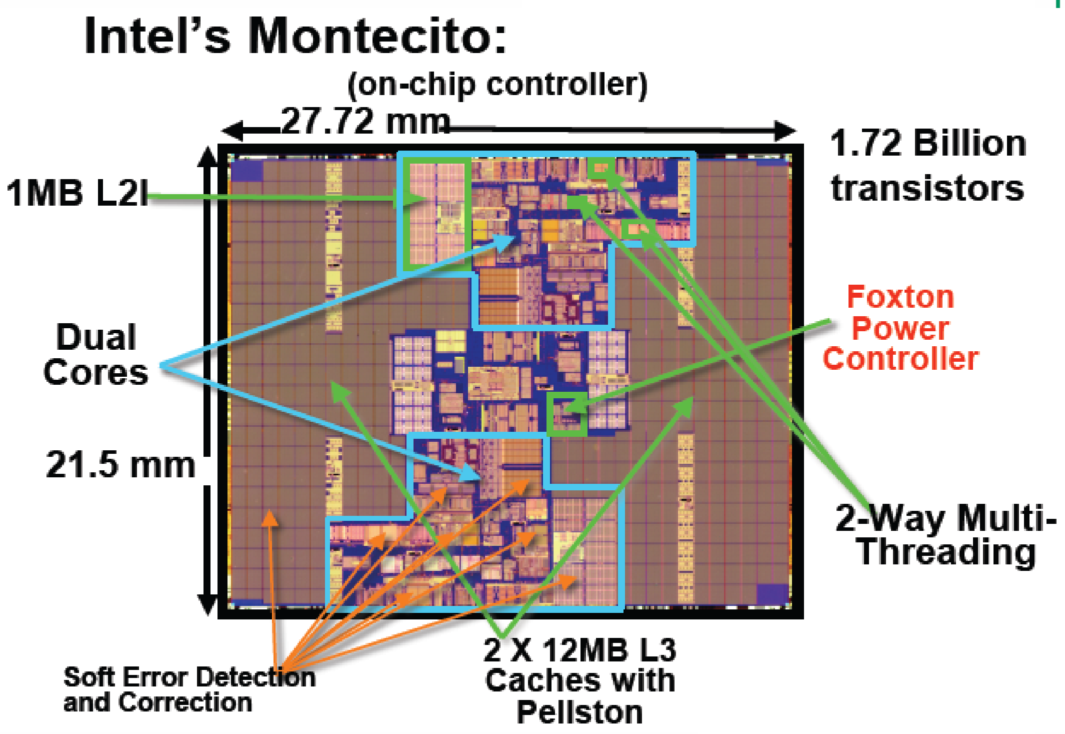


Integrated Power and Thermal Management

An autonomous programmable controller enables real time optimizations

An embedded controller provides the needed flexibility

- OS interfacing
- Multi-core management
- Per-part optimization



Outline

AMD Smarter Choice

Today's processor design landscape

- Trends

Issues making designer's lives difficult

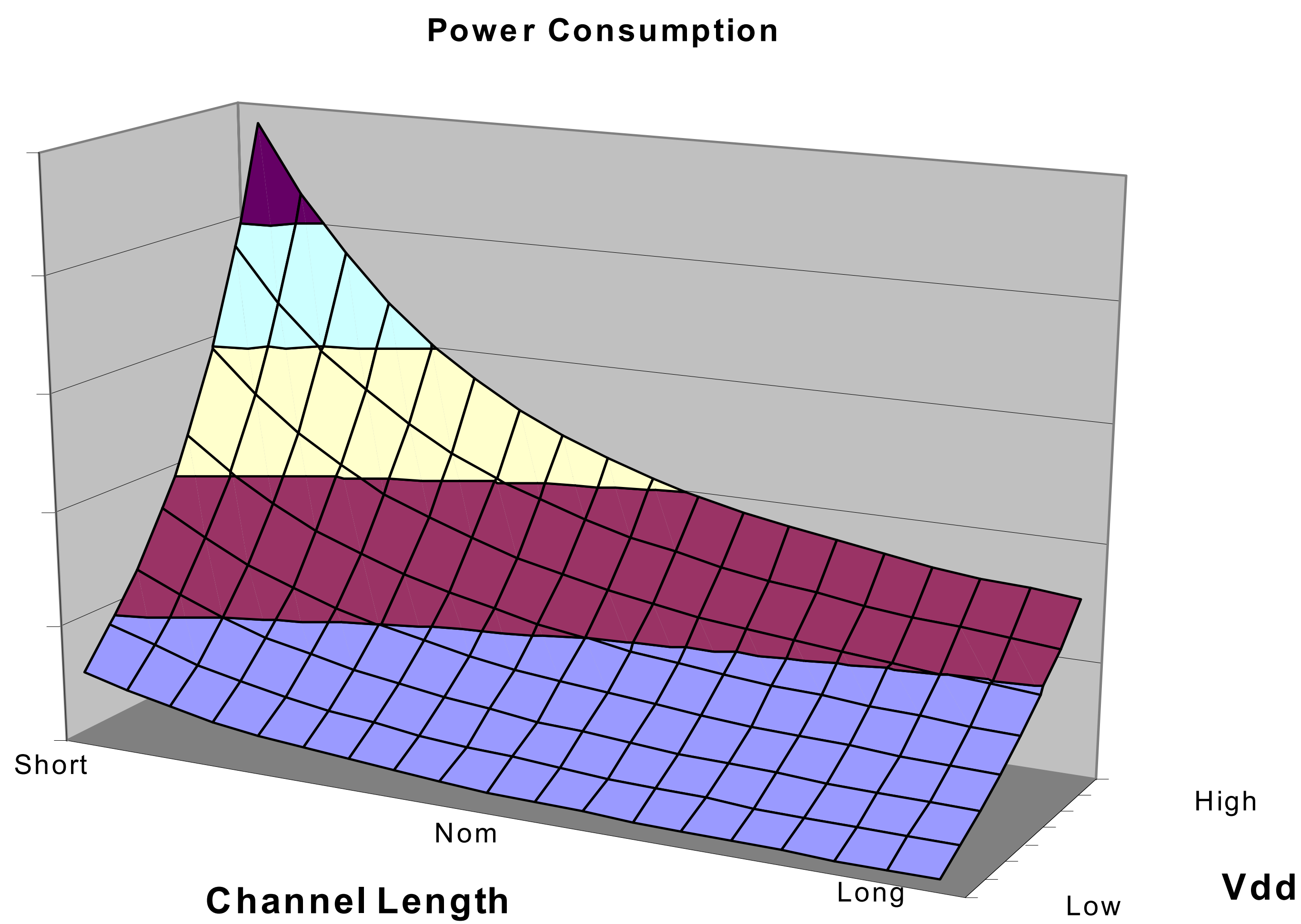
- Power limits
- Scaling effects

Design opportunities

- Circuit level
- **Architectural**

Summary

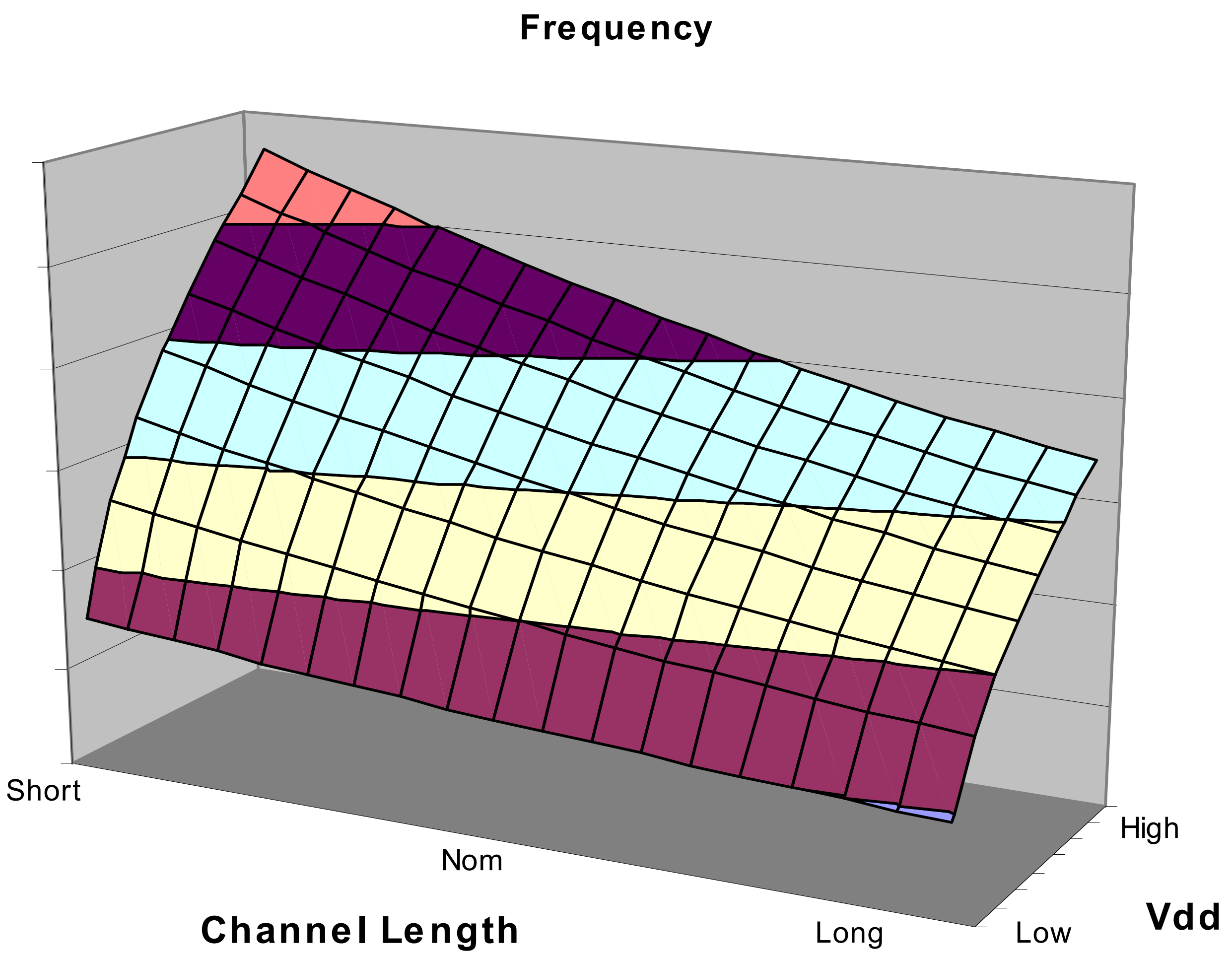
Traversing the Power Contour



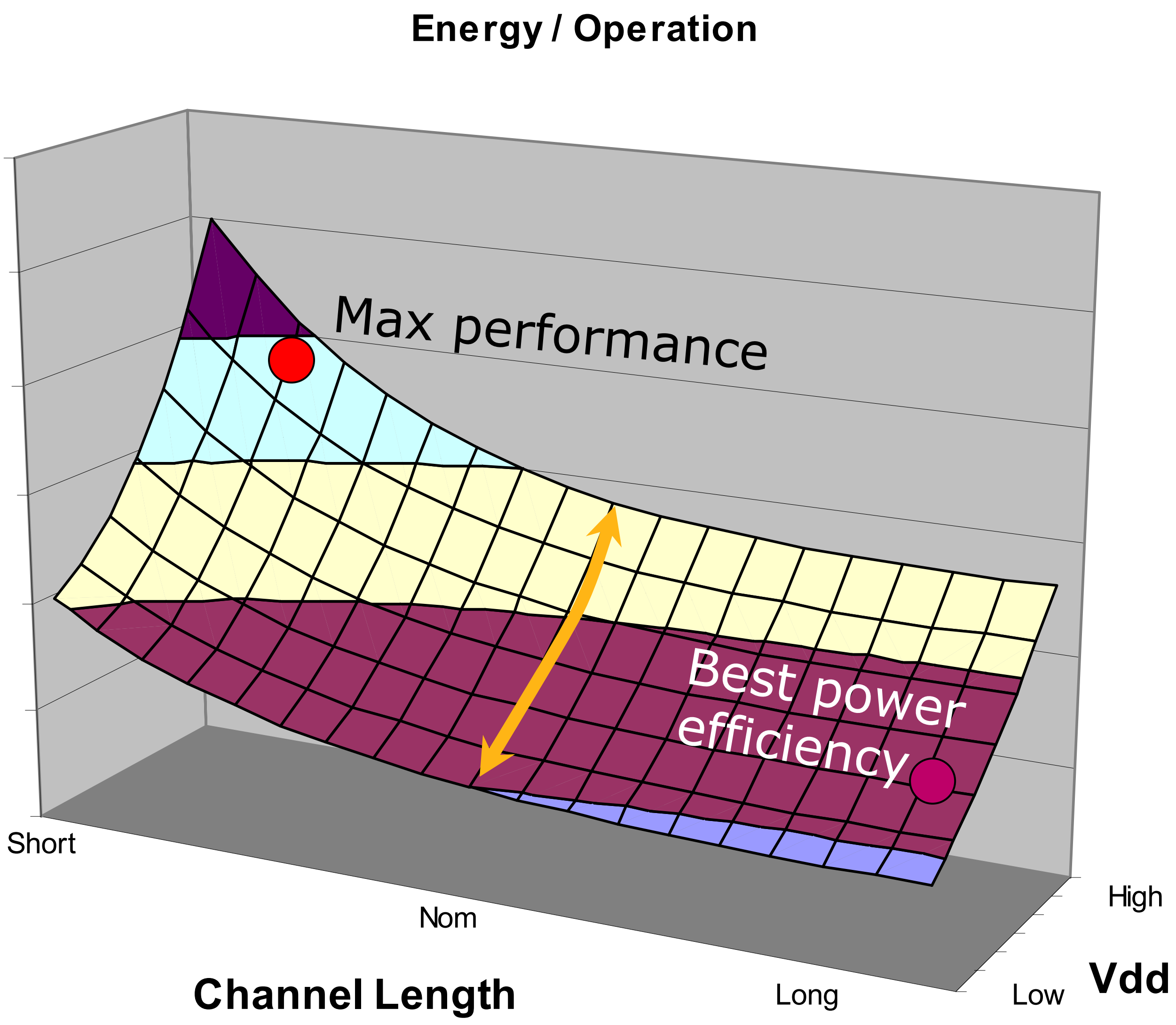
Switching Power | Crossover Power | Leakage Power

$$P \approx \underbrace{C_{TOT} \cdot \alpha \cdot F \cdot Vdd^2}_{\text{Switching Power}} + \underbrace{N_{TOT} \cdot \alpha \cdot F \cdot Vdd \cdot I_{CO}}_{\text{Crossover Power}} + \underbrace{N_{ON} \cdot I_{LEAK} \cdot Vdd}_{\text{Leakage Power}}$$

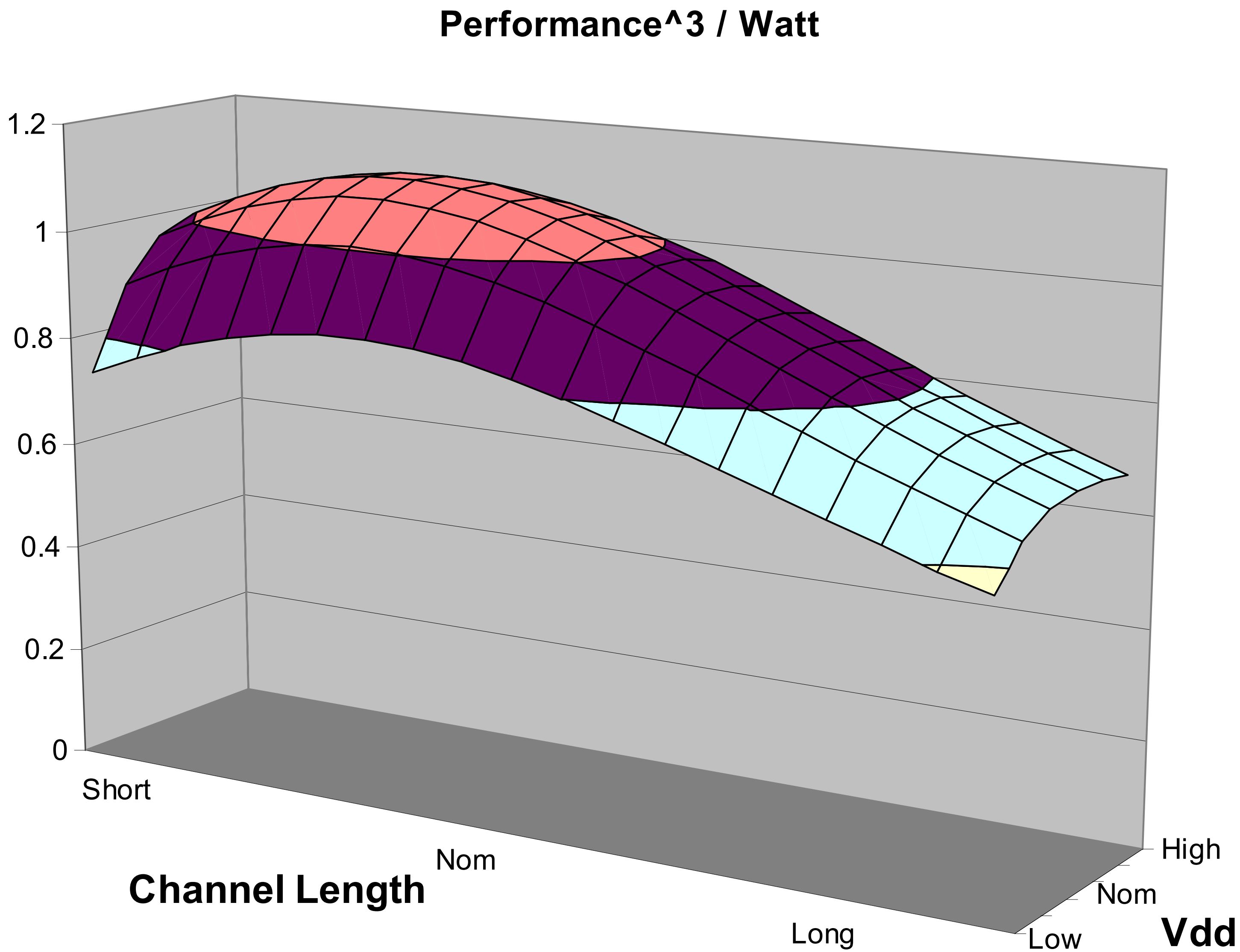
Traversing the Power Contour



Traversing the Power Contour for a Given Implementation

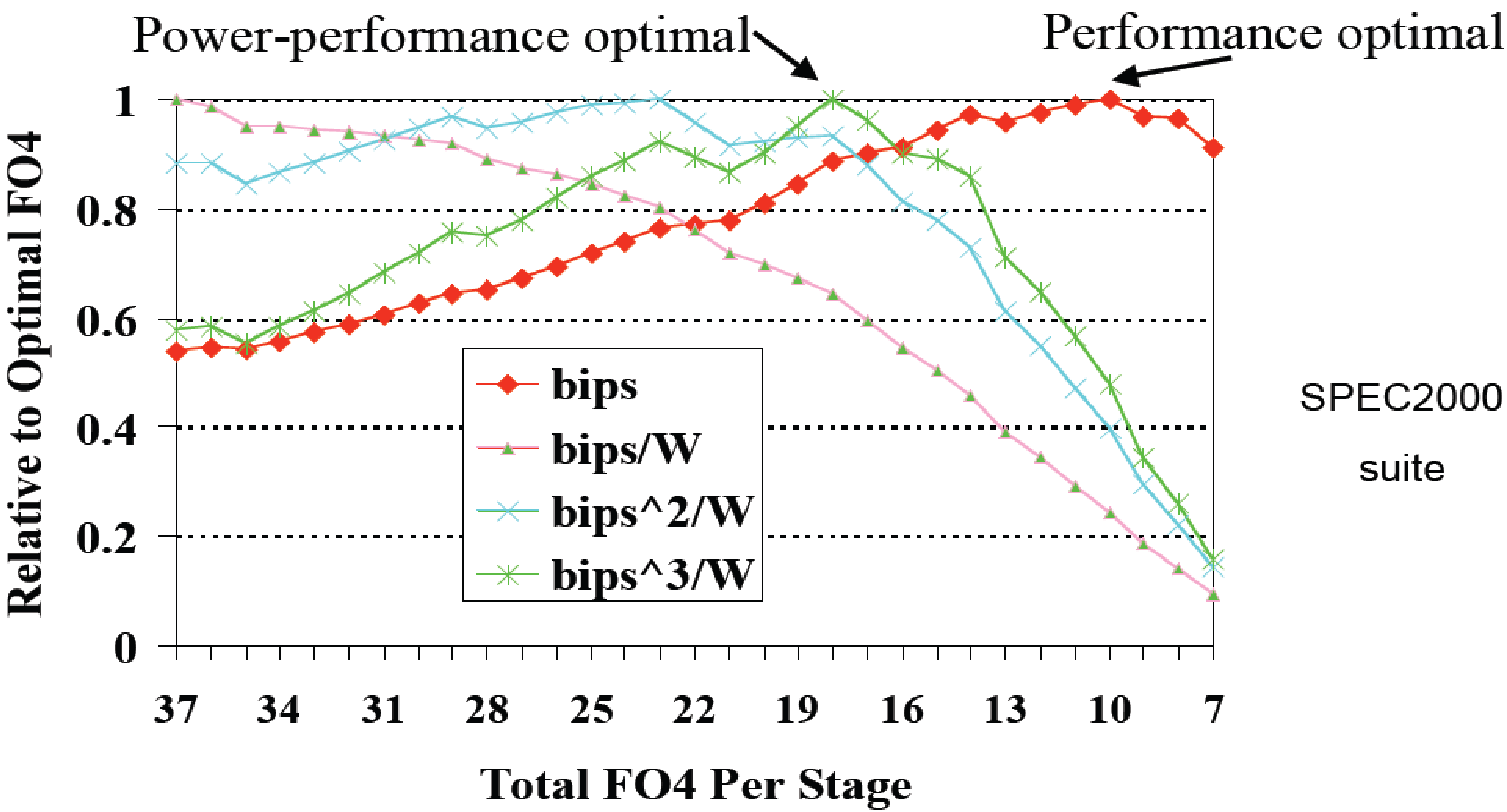


For Comparing Architectural Efficiency, Performance³ / W is most effective



Optimal Pipeline Depth

V. Srinivasan et al., MICRO-35

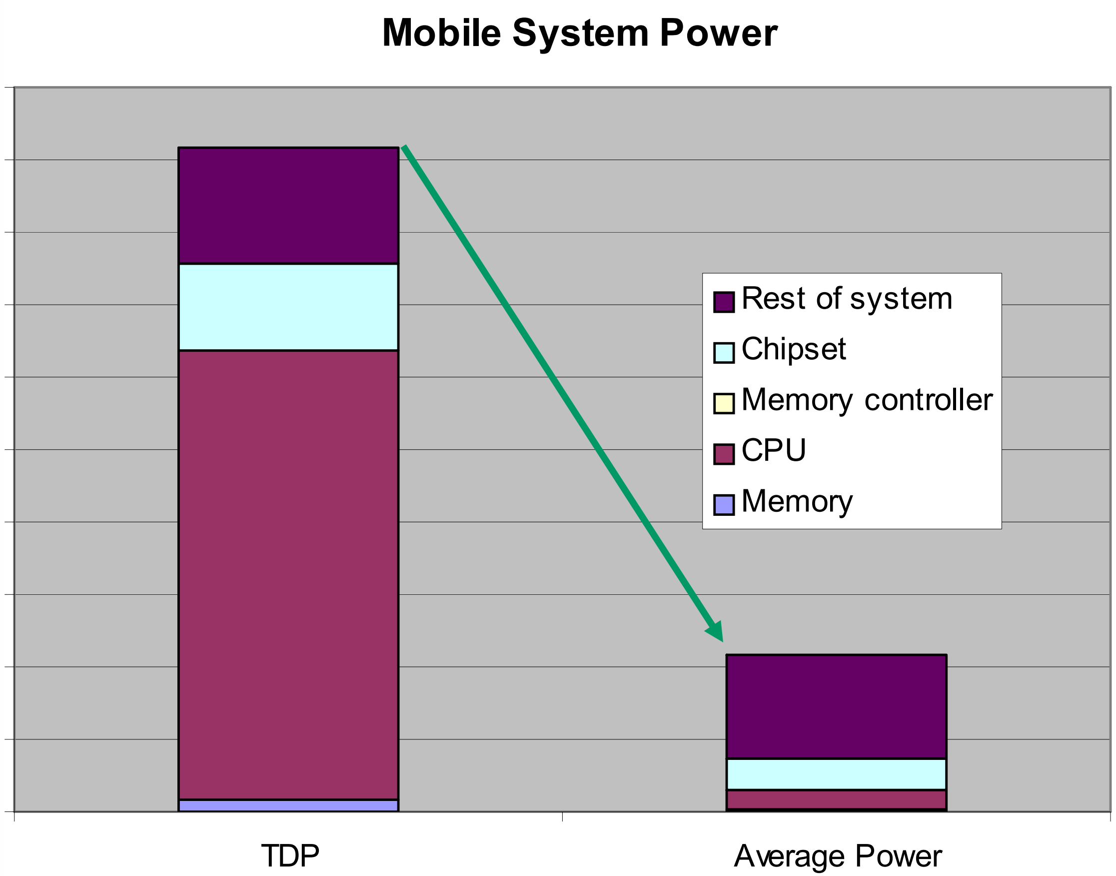


SPEC2000

suite

A Look at Mobile Processor Power

A Look at Mobile System Power

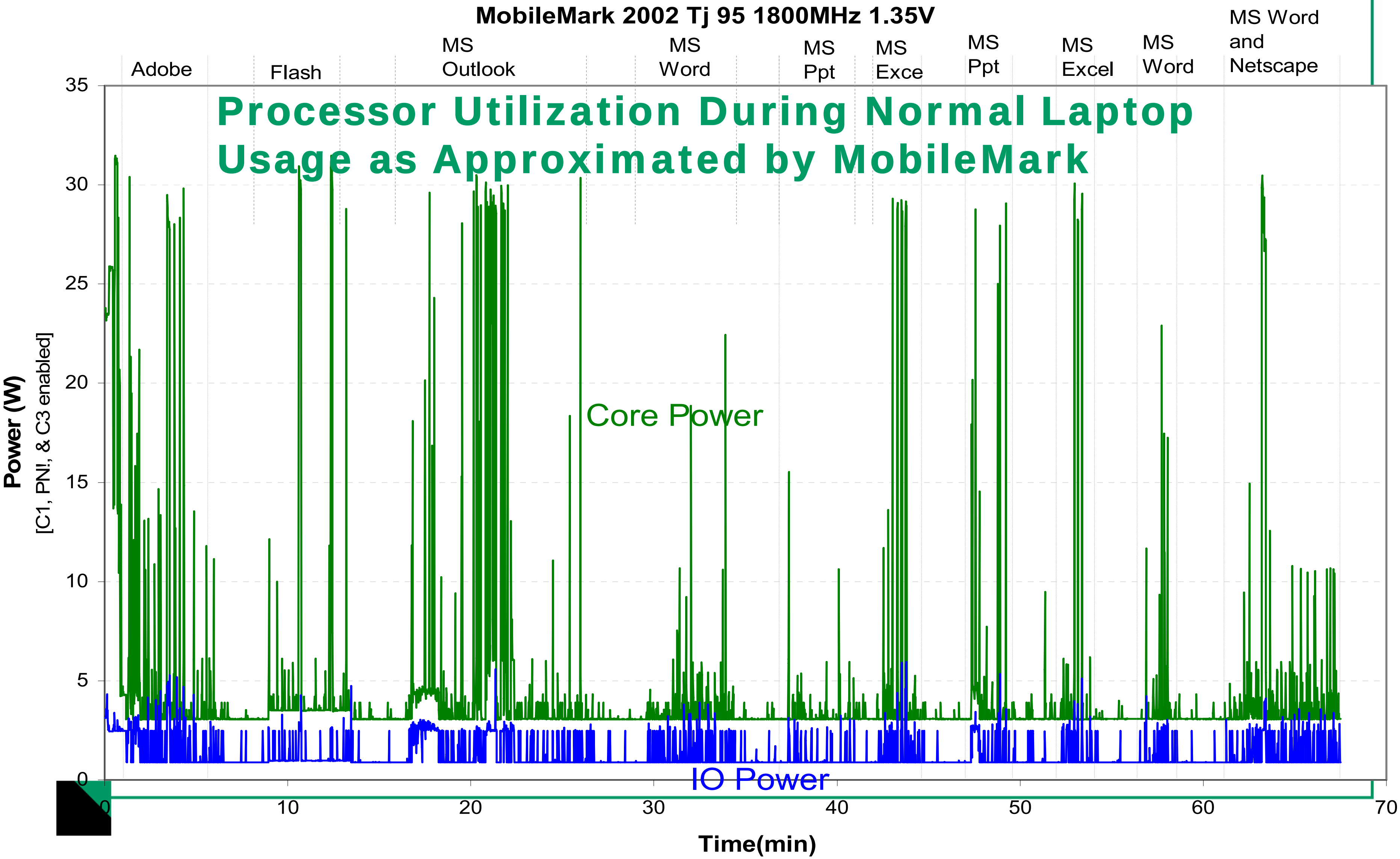


If a laptop burned TDP power all the time, battery life would be measured in minutes

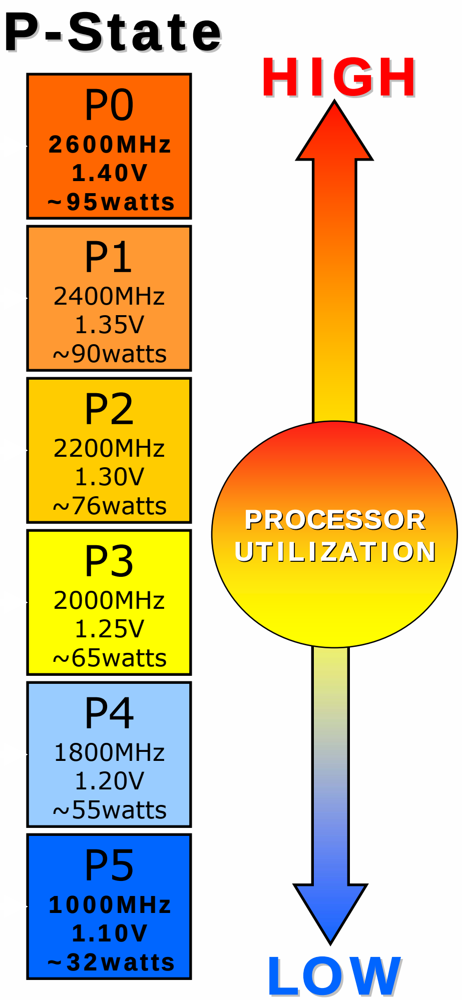
How do we get mobile average power so much lower than TDP?

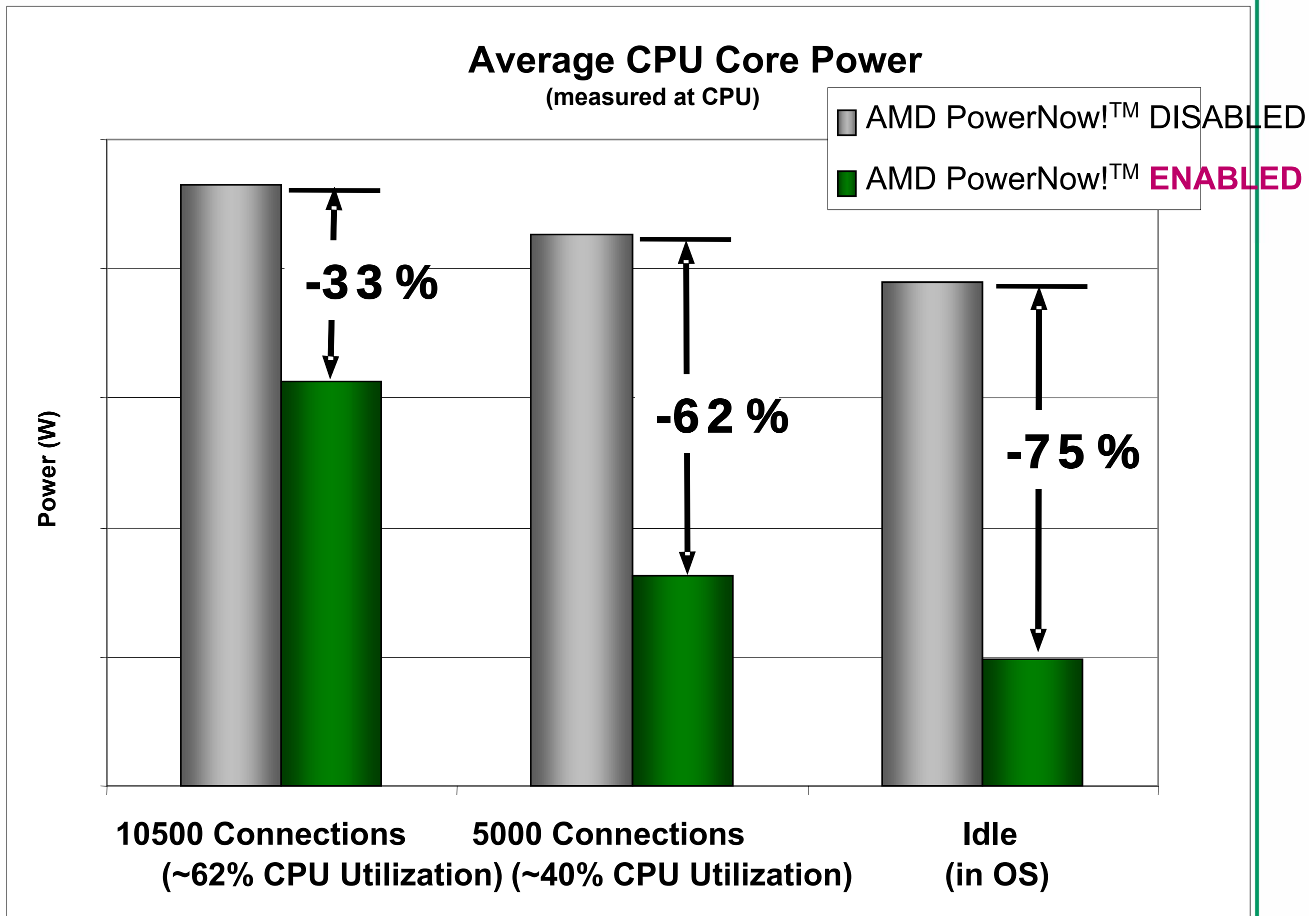
The Answer:
Take Advantage of Typically Low CPU Utilization

AMD Smarter Choice



Reducing Power and Cooling Requirements with Processor Performance States



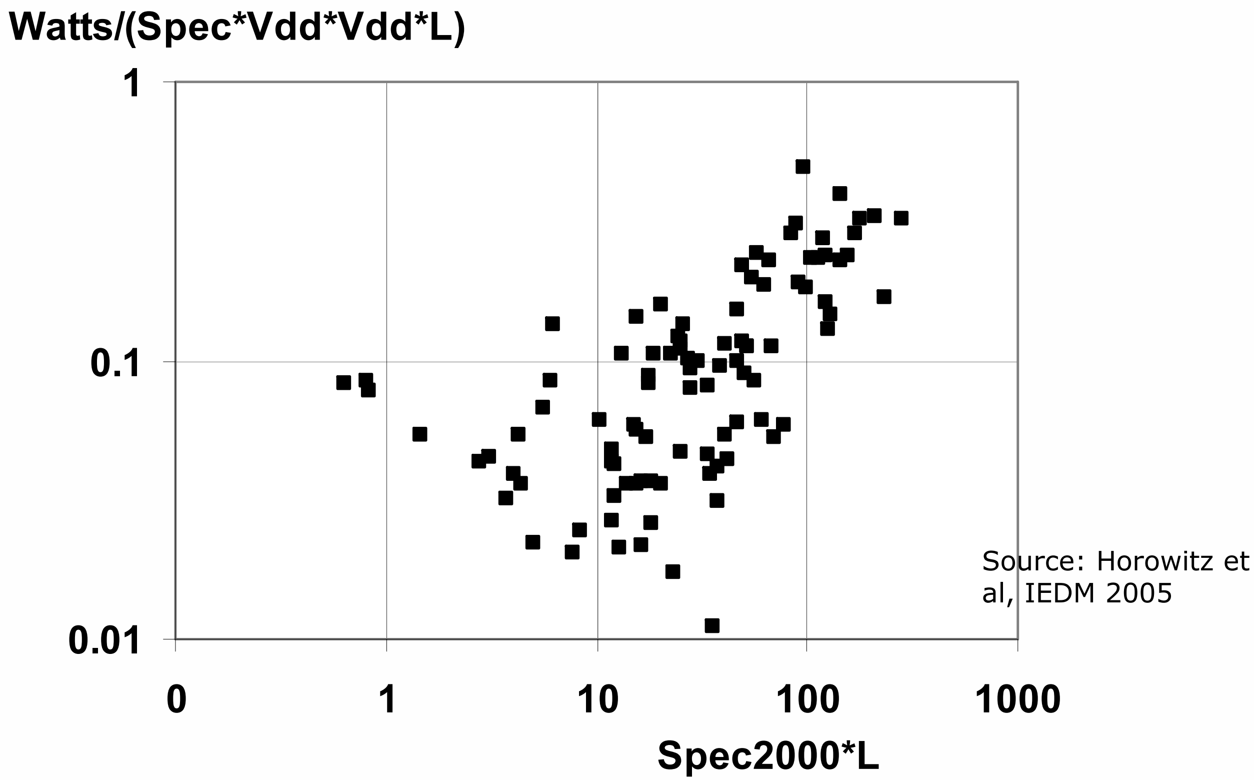


Up to 75% power savings (at idle)!

Additionally "C-states" reduce power further by cutting clocks completely and dropping voltage to retention levels

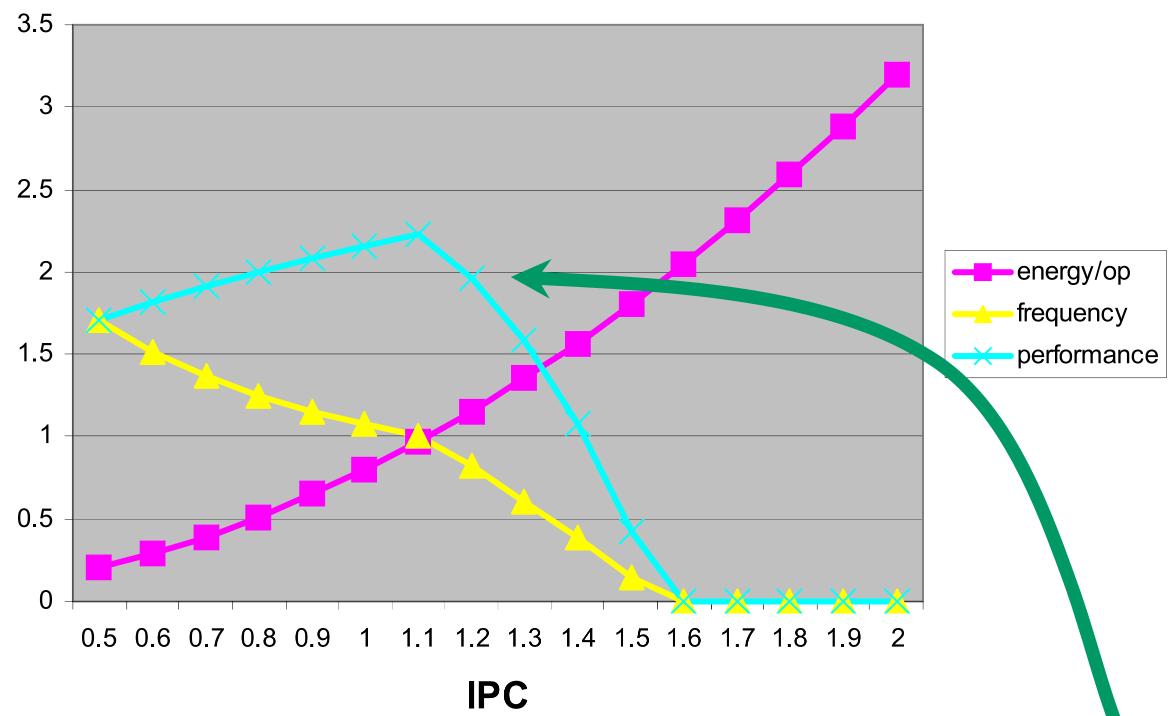
Improving Peak Performance per Watt

Adding Features to Increase Performance



- Increasing execution efficiency has, historically hurt power efficiency
- However, the cubic reduction of power with V/F scaling has tended to make this a good tradeoff

Adding Features to Increase Performance Works with V/F Scaling



Voltage scaling has it limits

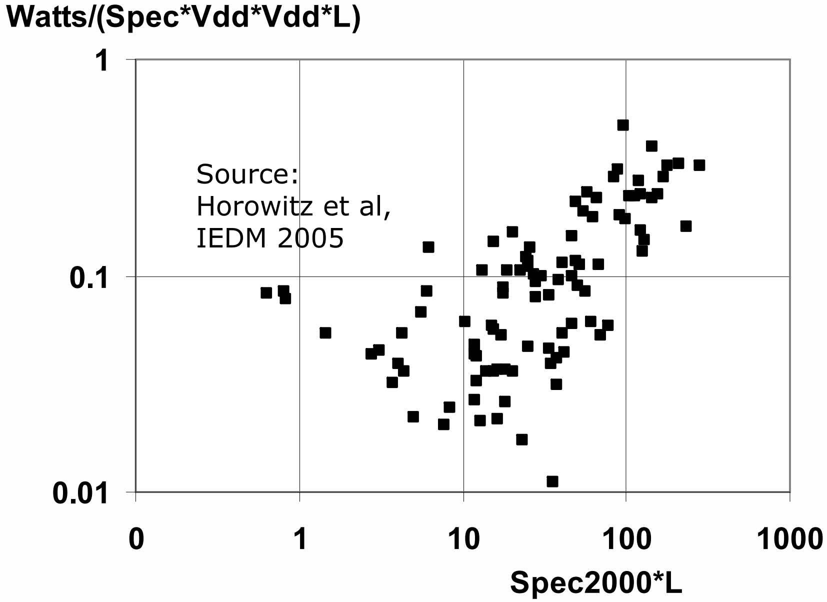
➔More power efficient designs have an advantage

➔High power designs get penalized due to higher di/dt, higher temperatures etc.

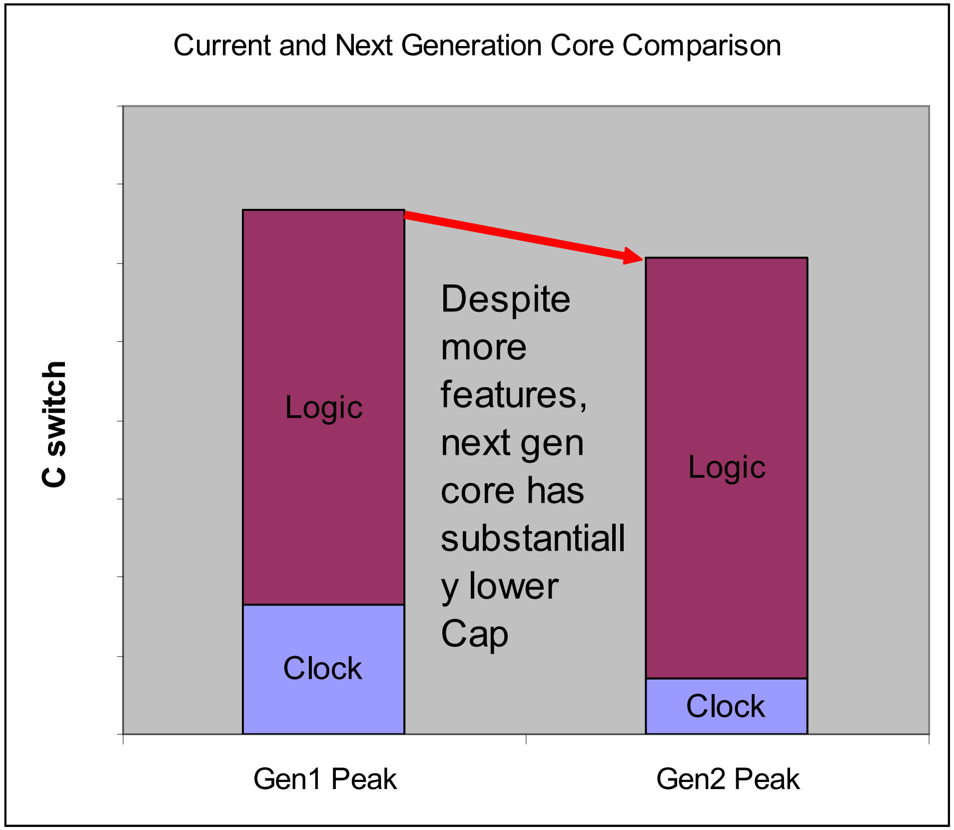
If we hit V_{MIN} however, the game is over

How Hard is Improving Existing Processors?

AMD Smarter Choice



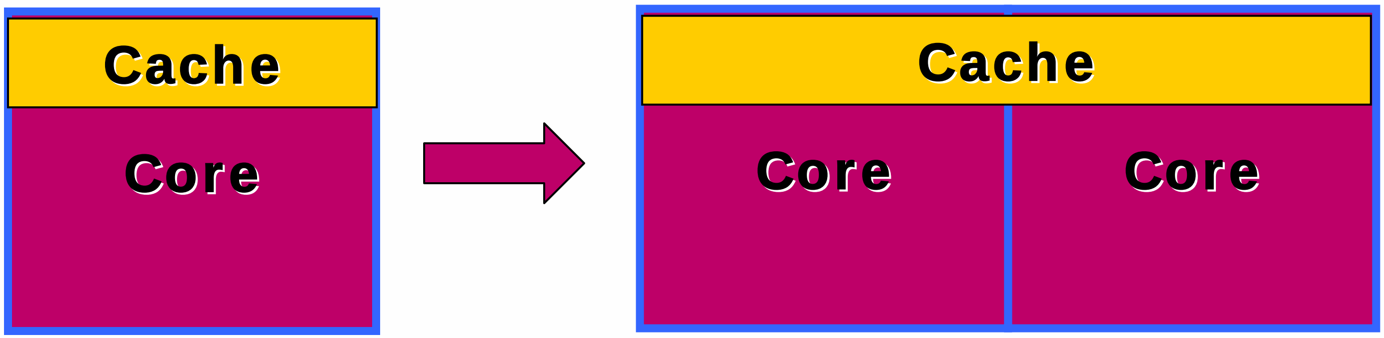
Peak performance costs more energy/operation



Most of the Big hitter improvements have been heavily mined already

Next generation AMD cores have >> 50% of clocks gated off even for high power code

Multi-Core to the Rescue?



Voltage	=1
Frequency	=1
Area	=1
Power	=1
Perf	=1
Perf/Watt	=1

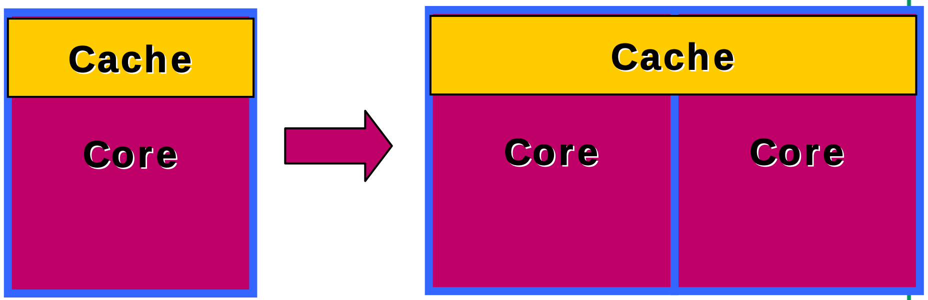
Voltage	=.85
Frequency	=.85
Area	=2
Power	=1
Perf	≈1.7
Perf/Watt	≈1.7

Sounds like a great story, what’s the catch?

Multi-Core to the Rescue?

Some of the catches:

- What if you're already at V_{MIN}? Need to cut frequency in half to stay within power limit ☹
- How much parallelizable code is really out there?
- More compute capacity means more IO and memory bandwidth demands …



Voltage	=1
Frequency	=1
Area	=1
Power	=1
Perf	=1
Perf/Watt	=1

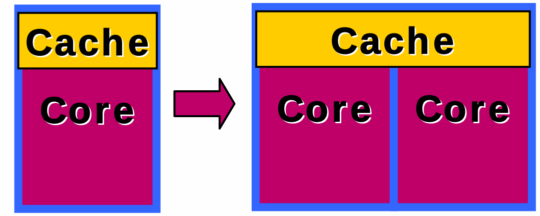
Voltage	=.85
Frequency	=.85
Area	=2
Power	=1
Perf	≈1.7
Perf/Watt	≈1.7

Multi-Core Issues: Amdahl's Law

AMD
Smarter Choice

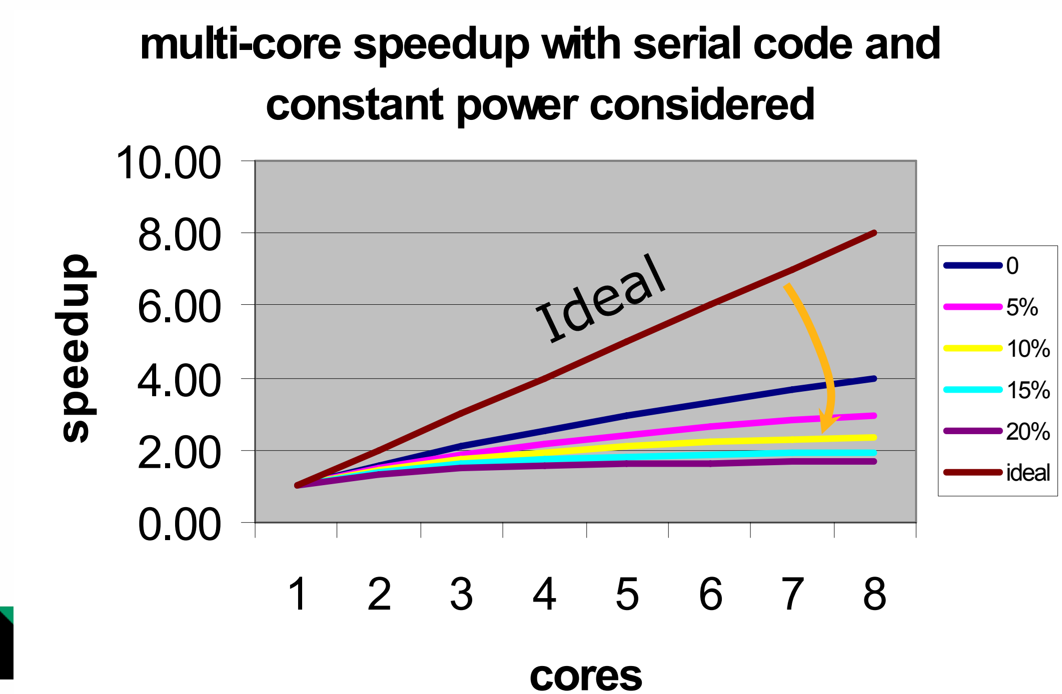
There is almost always a portion of an application that cannot be parallelized

- This portion becomes a bottleneck as the number of threads is increased
- A typical value is in the range of 10%



Voltage	=1
Frequency	=1
Area	=1
Power	=1
Perf	=1
Perf/Watt	=1

Voltage	=.85
Frequency	=.85
Area	=2
Power	=1
Perf	≈1.7
Perf/Watt	≈1.7

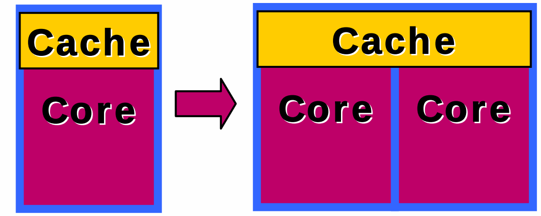


Just 10% serial code drops 8 core performance improvement by 41%

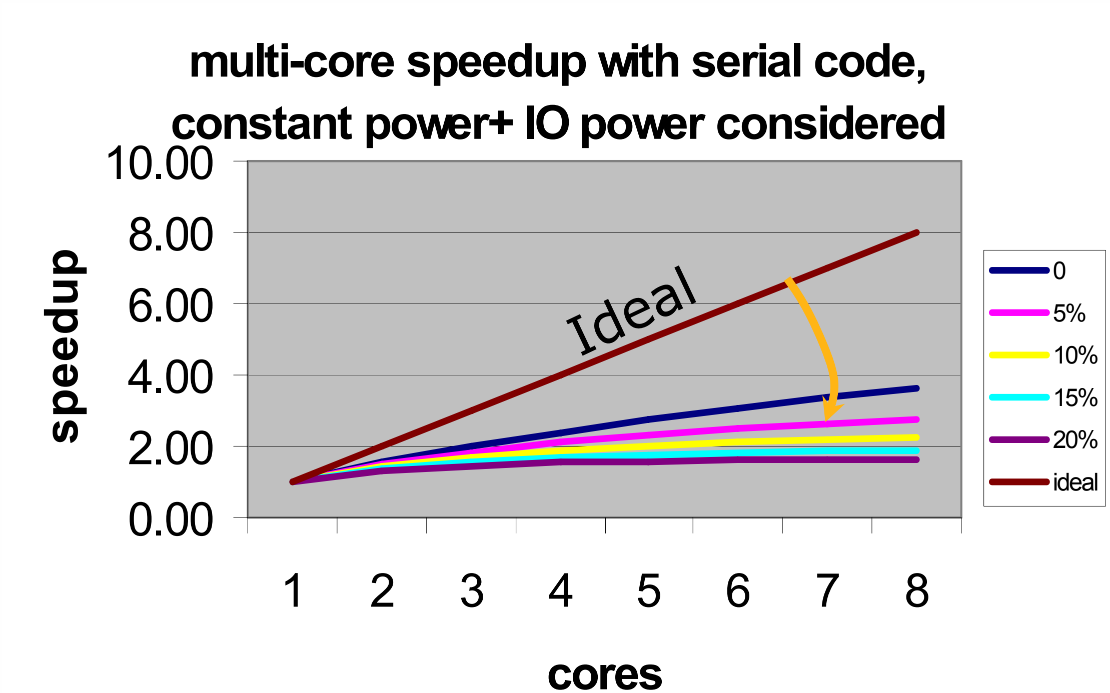
Multi-Core Issues: IO Power

All those extra cores need their own data …

IO power in terms of W/Gb/s has been pretty constant in the range of 20mW for years



Voltage	=1	Voltage	=.85
Frequency	=1	Frequency	=.85
Area	=1	Area	=2
Power	=1	Power	=1
Perf	=1	Perf	≈1.7
Perf/Watt	=1	Perf/Watt	≈1.7



- If we increase IO power accordingly, but hold total chip power constant with V/F scaling, things get worse
- Overall performance drops by another 10% or so …

The Transition to Parallel Applications

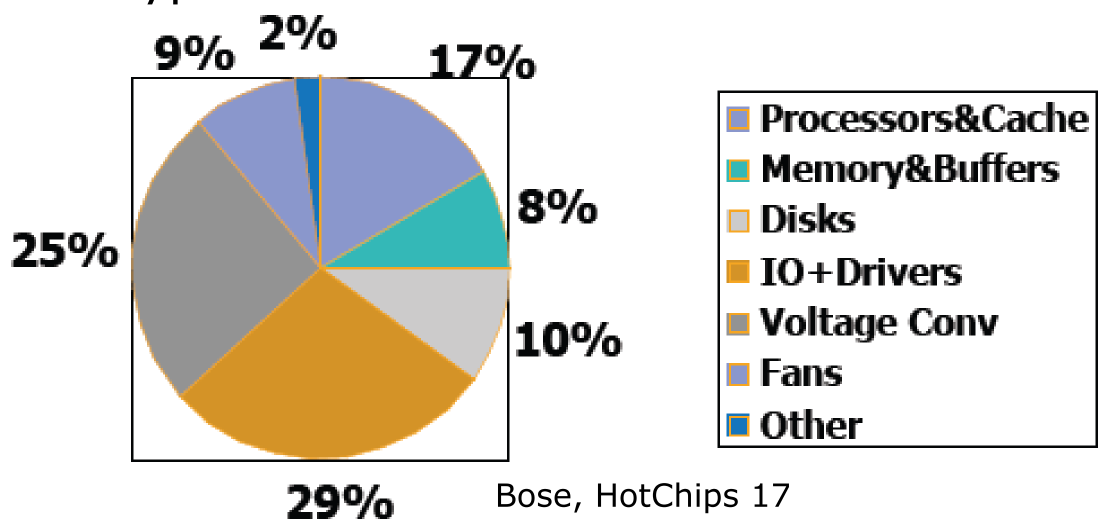
Single-threaded Applications	Parallel Applications
Most of today's applications	Small number of applications (worked by experts for 10+ yrs)
Well understood optimization techniques	Awkward development, analysis and debug environments
Advanced development, analysis and debug tools	Parallel programming is hard!
	Amdahl's law is still a law…
Conceptually, easy to think about	SW productivity is already in a crisis → ***this worsens things!***

Establishing an *appropriate balance*
is key for managing this important transition

Other Architectural Directions: Integration

Not only does the integration of more system components (i.e. memory controllers, IO etc.) improve performance

Typical Server Power Breakdown

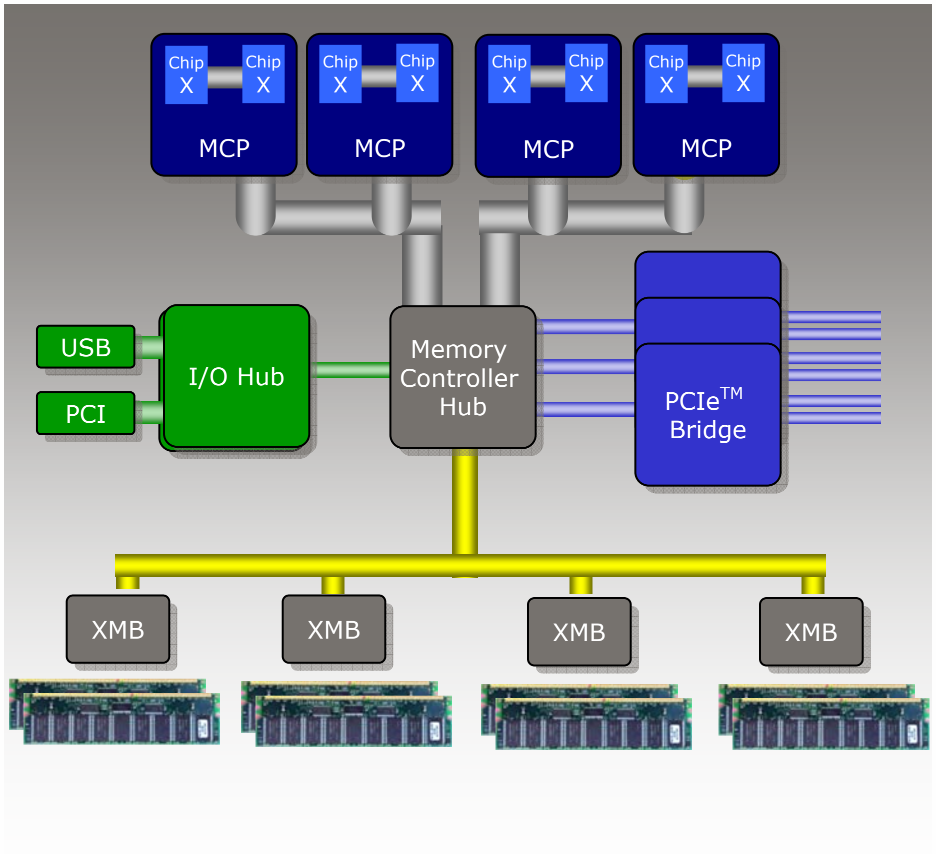


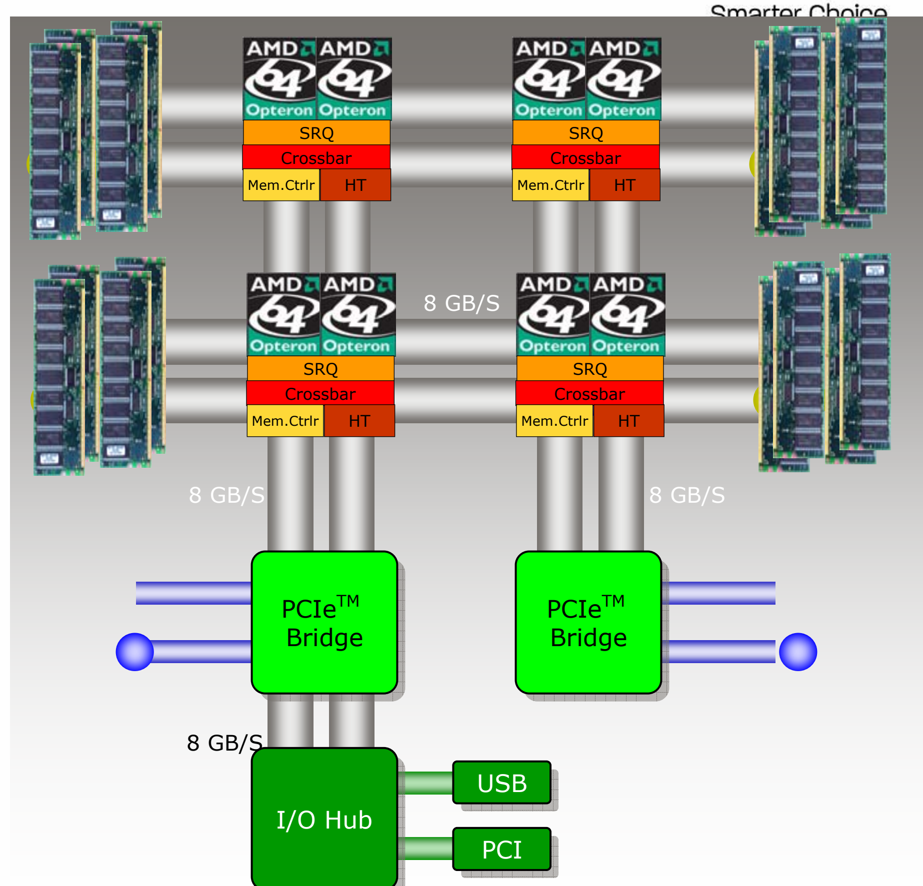
Bose, HotChips 17

Integration reduces power significantly as well

- IO communication overhead drops
- CPU integrated power management can dynamically optimize
- Power efficiency of special function components (i.e. graphics accelerators, network processors etc.) greatly exceeds that of general purpose CPUs

System-level Power Consumption





Dual-Core Packages with legacy technology

- 692 watts for processors (173w each)
- 48 watts for external memory controller

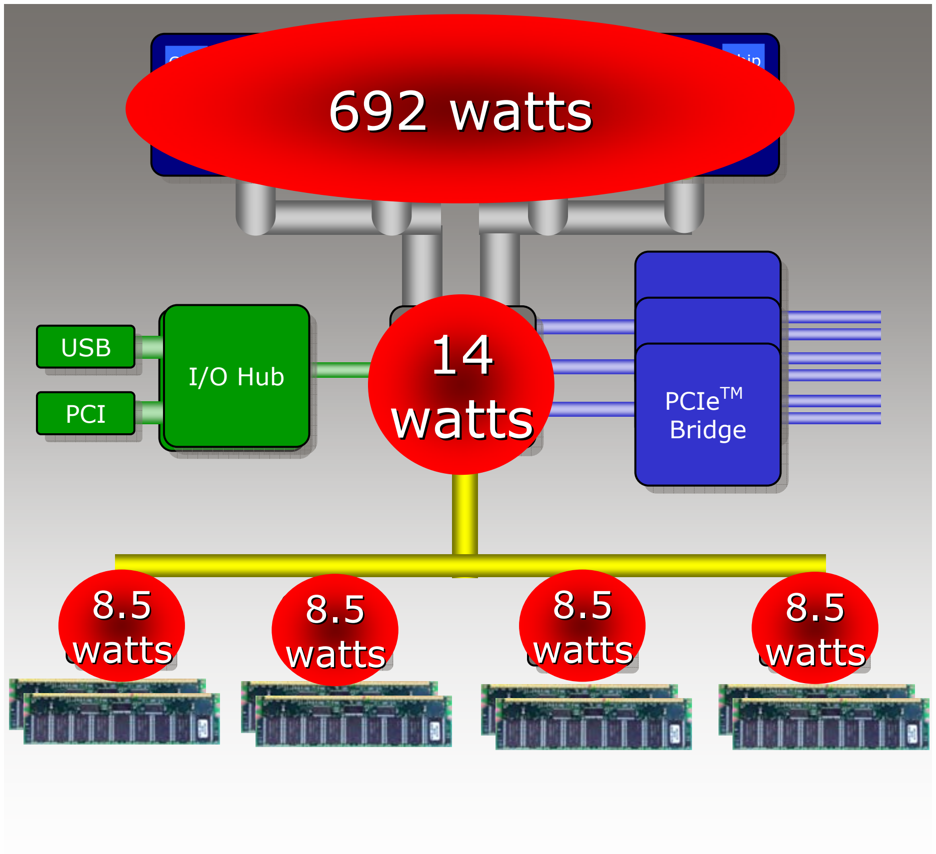
95% More Power

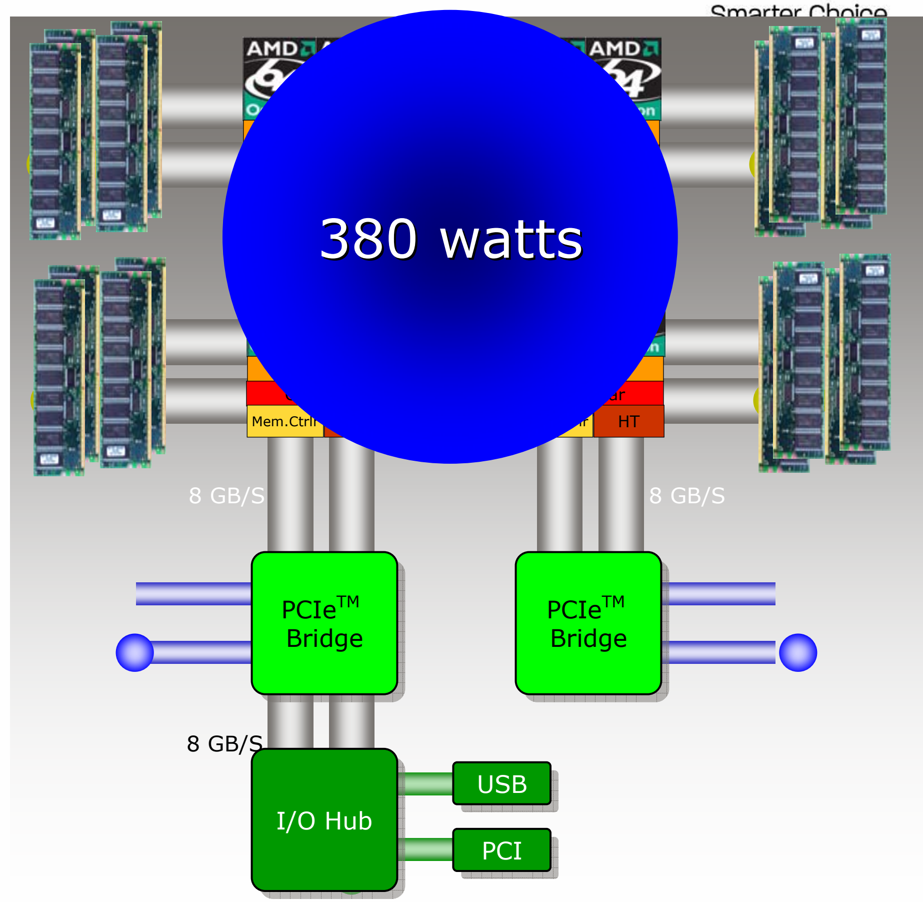
Dual-Core AMD Opteron™ processors

- 380 watts for processors (95w each)
- Integrated memory controllers

System-level Power Consumption

AMD Smarter Choice





Dual-Core Packages with legacy technology

- 692 watts for processors (173w each)
- 48 watts for external memory controller

95% More Power

740 watts

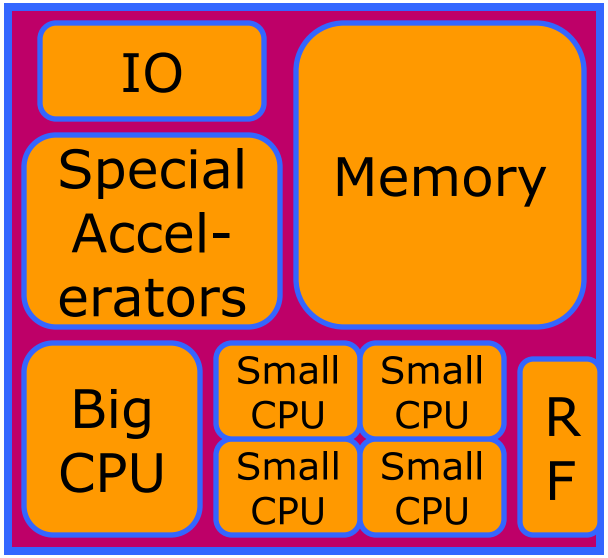
Dual-Core AMD Opteron™ processors

- 380 watts for processors (95w each)
- Integrated memory controllers

380 watts

tual system power measurements may vary based on configuration and components used.

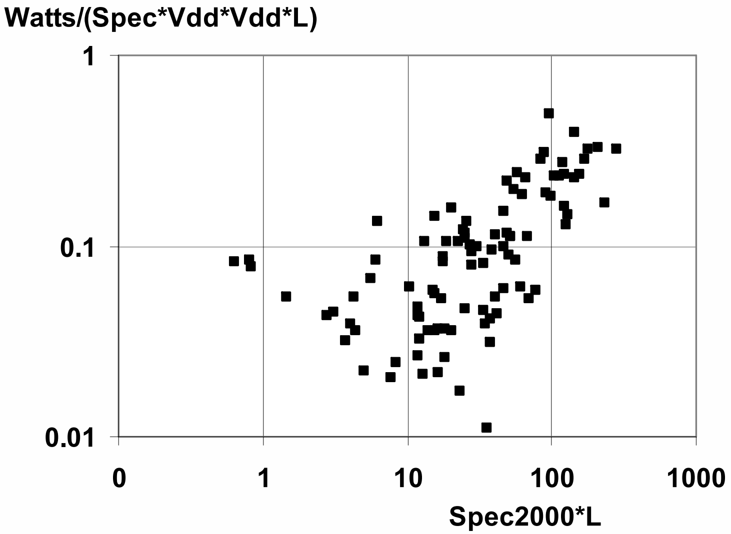
Other Architectural Directions: Integration



Integrating dual designs for processor core enable both peak performance and throughput/watt

Barriers?

- Integration of heterogeneous designs non-trivial
- IP barriers
- Schedule issues with multiple converging components

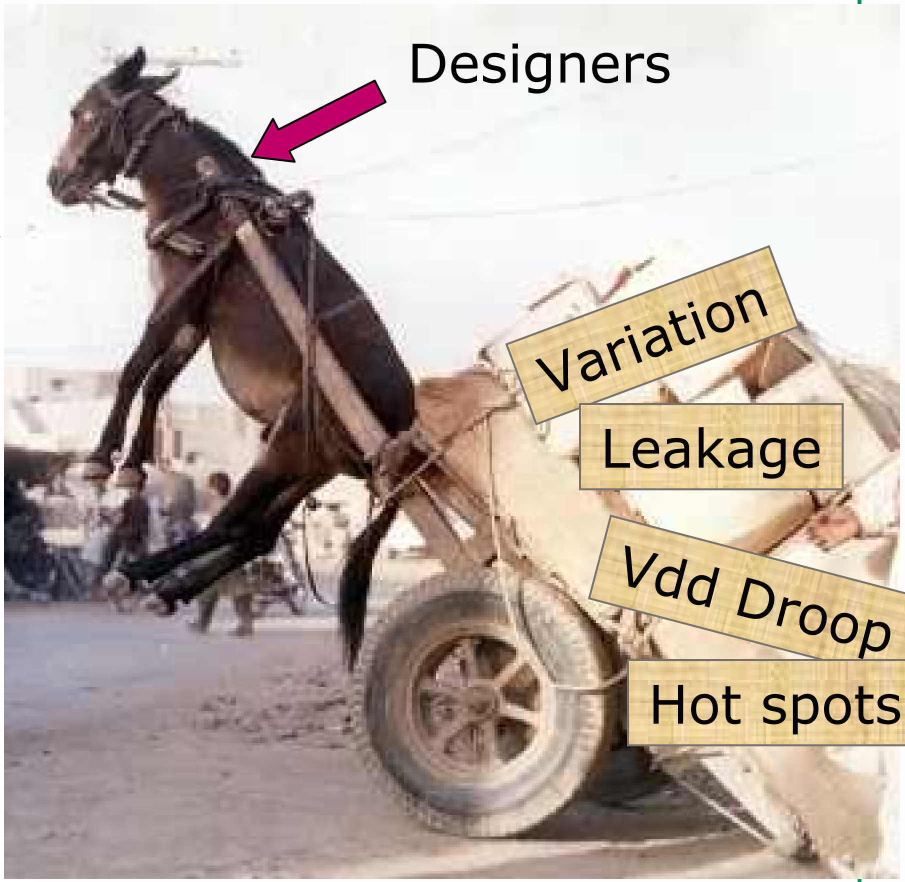


Summary (1 of 2)

Silicon process technology is unlikely to be the major engine of processor performance increases in the future

Major circuit related challenges that we've only just started to address lie ahead:

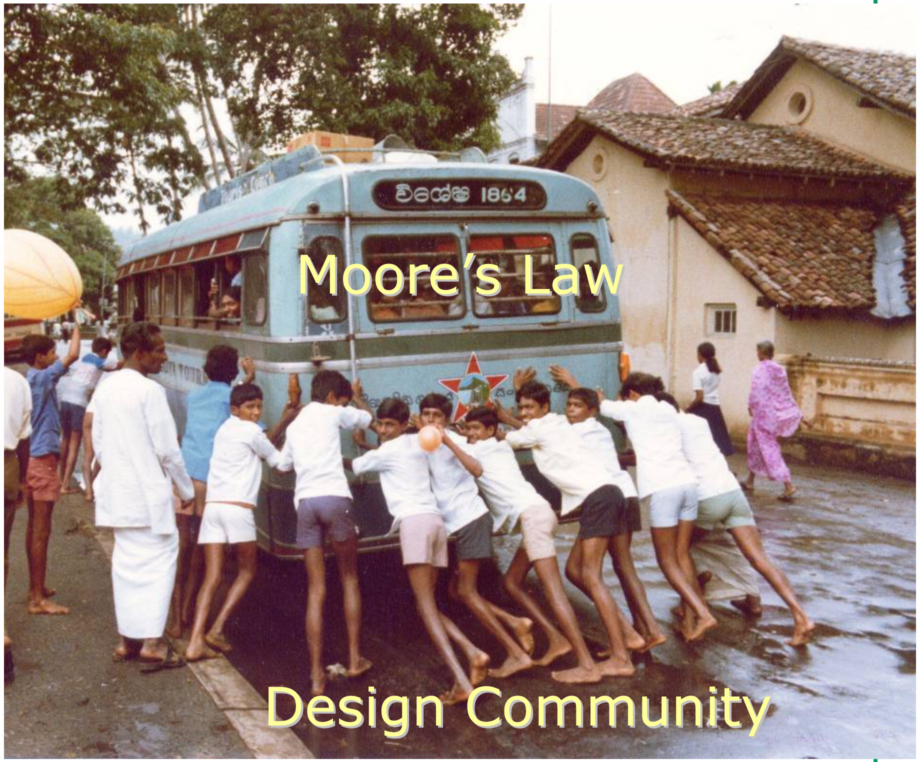
- Design for variation tolerance and mitigation
- Maintaining dynamic voltage headroom within reliability and variation imposed limits
- Adaptive, self-healing techniques are a key direction



Summary (2 of 2)

Silicon process technology is unlikely to be the major engine of processor performance increases in the future

- CPU architectures are converging on modest pipe length, limited issue out of order designs
- Multi-core is good, but has limits in the not too distant future
- Heterogeneous integration is a key direction



We're up to the challenge, but it will be a joint effort …