

Scaling, Power and the Future of CMOS

Mark Horowitz, Elad Alon, Dinesh Patil, Stanford

Samuel Naffziger, Rajesh Kumar, Intel

Kerry Bernstein, IBM

A Long Time Ago

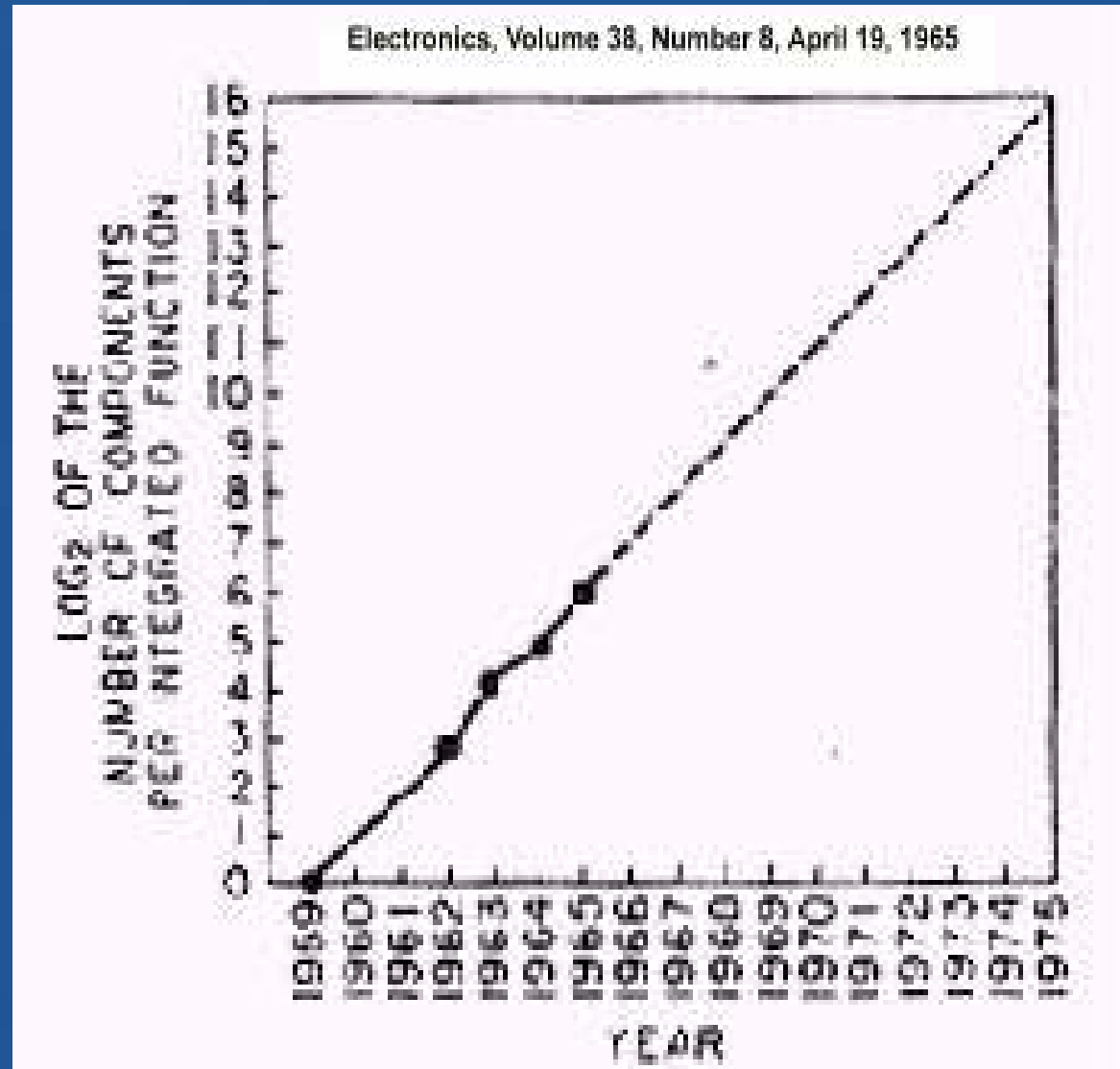
In a building far away

A man made a prediction

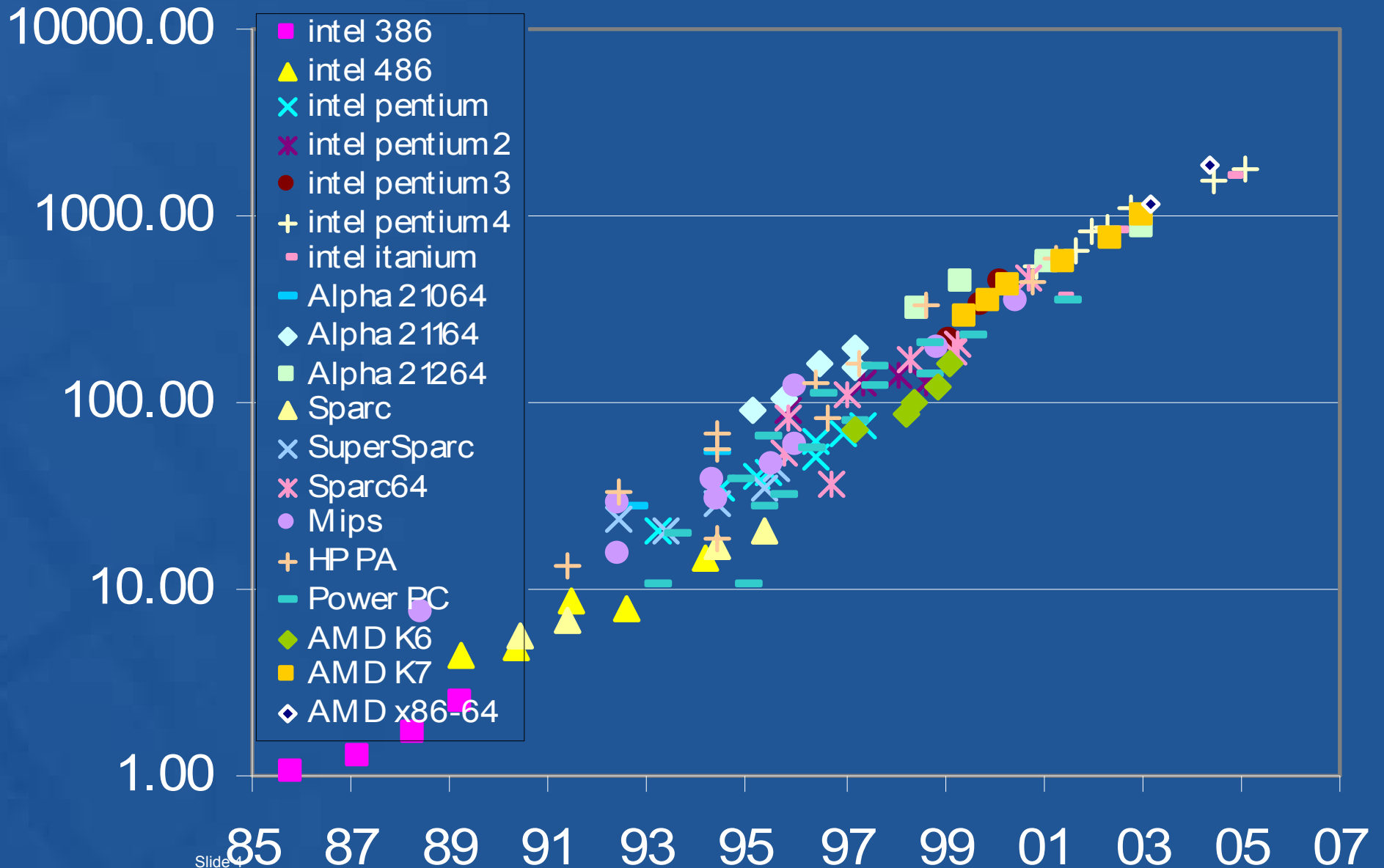
On surprisingly little data

That has defined an industry

Moore's Law



CMOS Computer Performance



Moore's Original Issues

- Design cost
- **Power dissipation**
- What to do with all the functionality possible

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



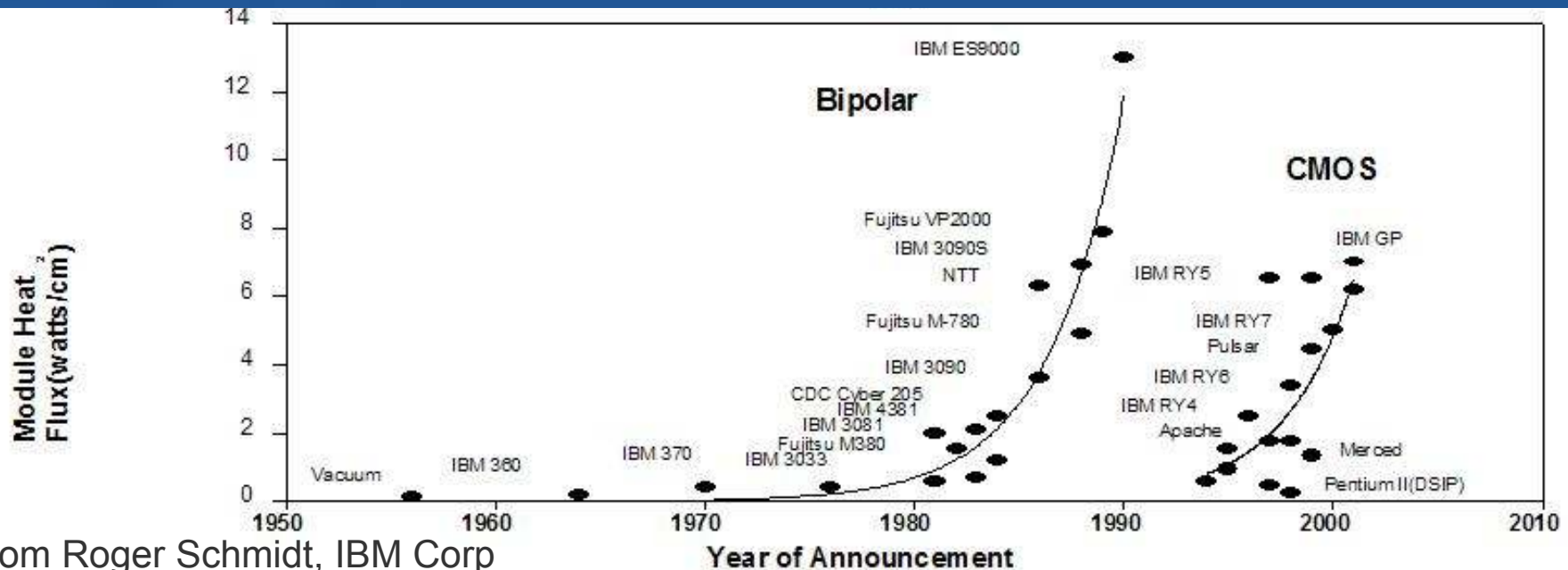
<ftp://download.intel.com/research/silicon/moorespaper.pdf>

Outline

- How designers will deal with poor power scaling
- Origins of the power problem
- An optimization perspective
 - Low power circuits and architectures
 - Cost of variability
- Future scenarios
- What device characteristics matter (to me)

The 80's Power Problem

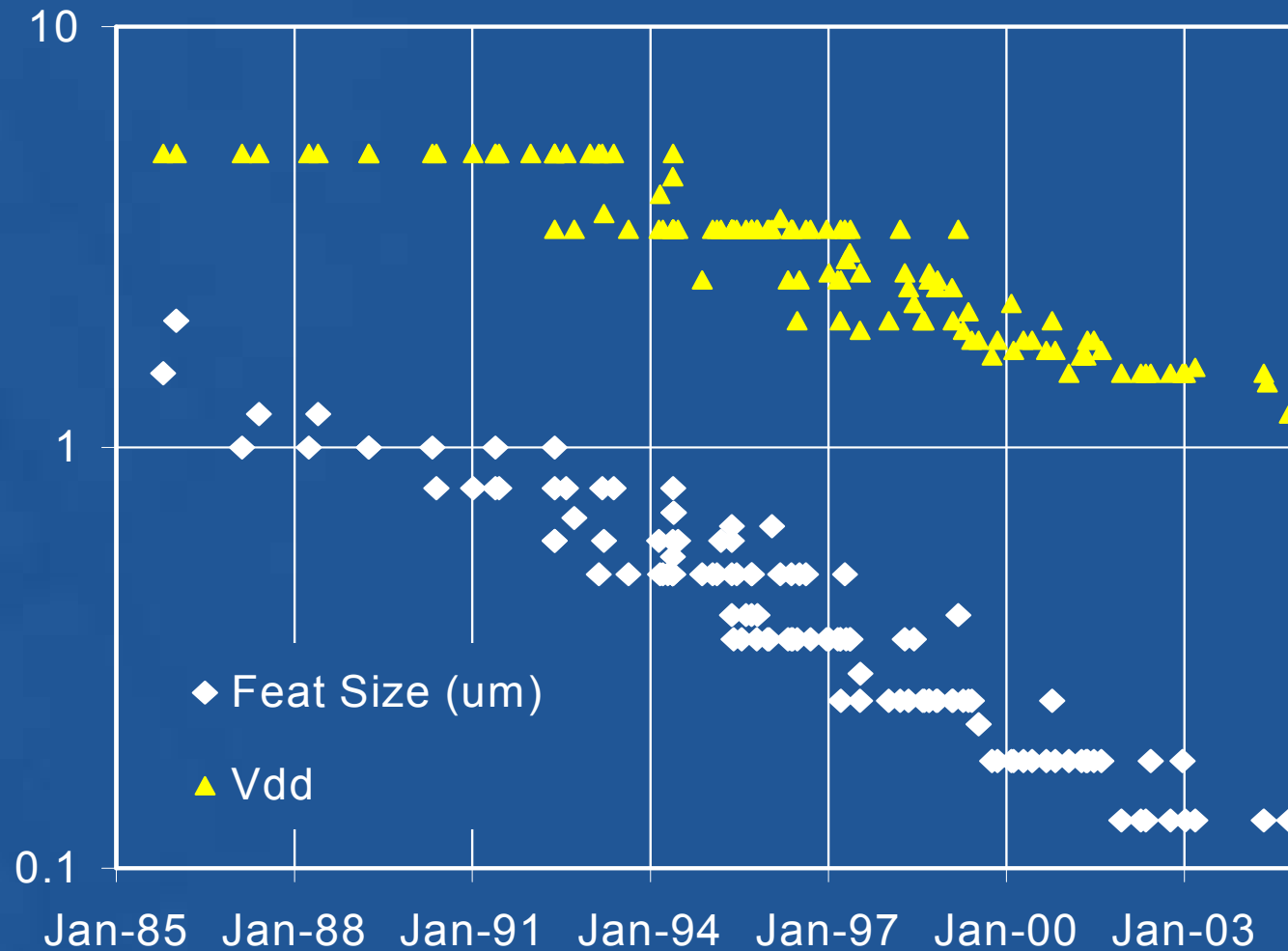
- Until mid 80s technology was mixed
 - nMOS, bipolar, some CMOS
- Supply voltage was not scaling / power was rising
 - nMOS, bipolar gates dissipate static power



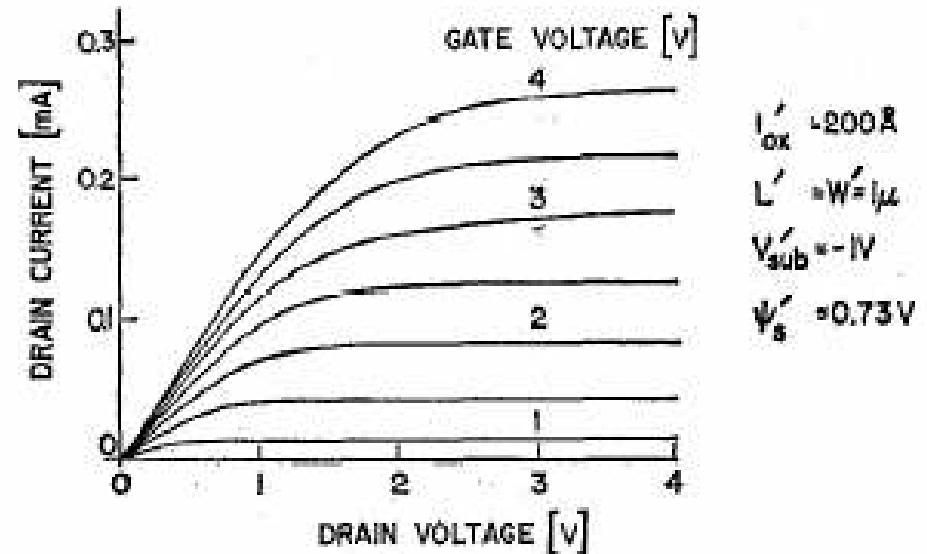
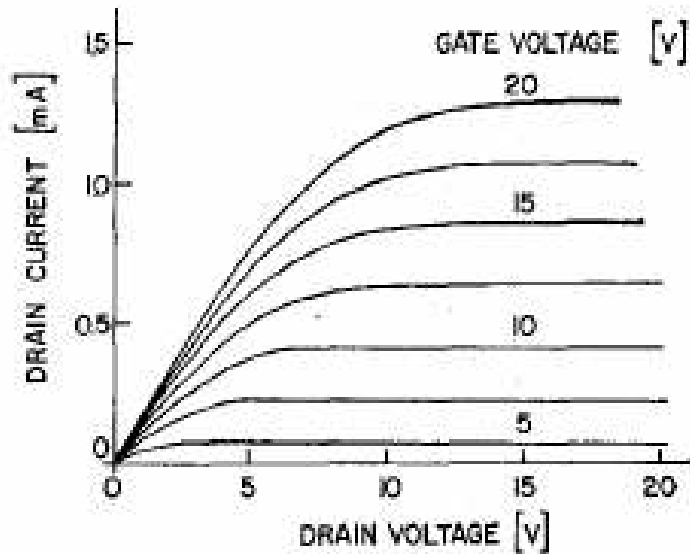
From Roger Schmidt, IBM Corp

Solution: Move to CMOS

- And then scale Vdd



Scaling MOS Devices

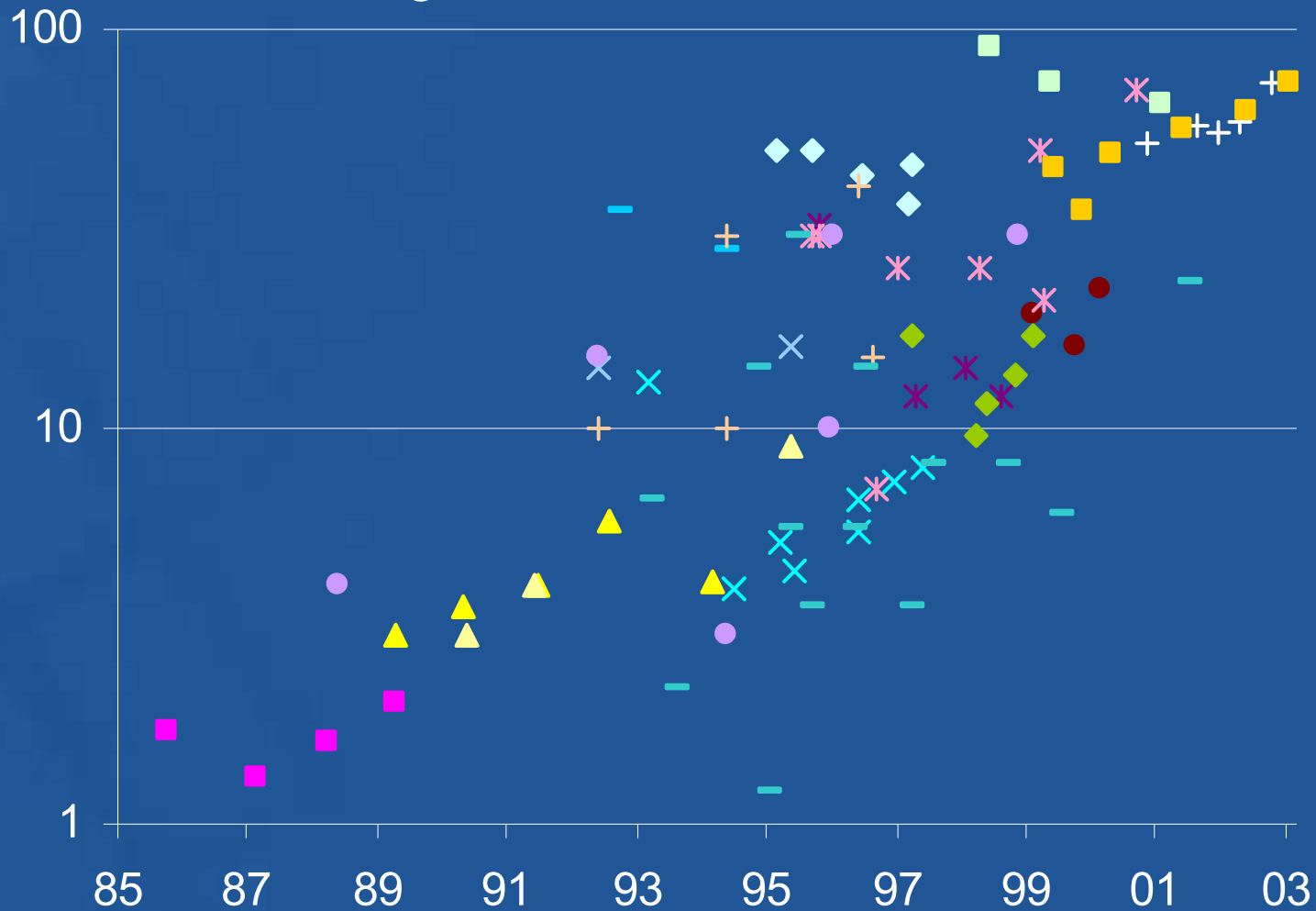


JSSC Oct 74, pg 256

- In this ideal scaling
 - V scales to αV , L scales to αL
 - So C scales to αC , i scales to αi (i/μ is stable)
 - Delay = CV/i scales as α
 - Energy = CV^2 scales as α^3

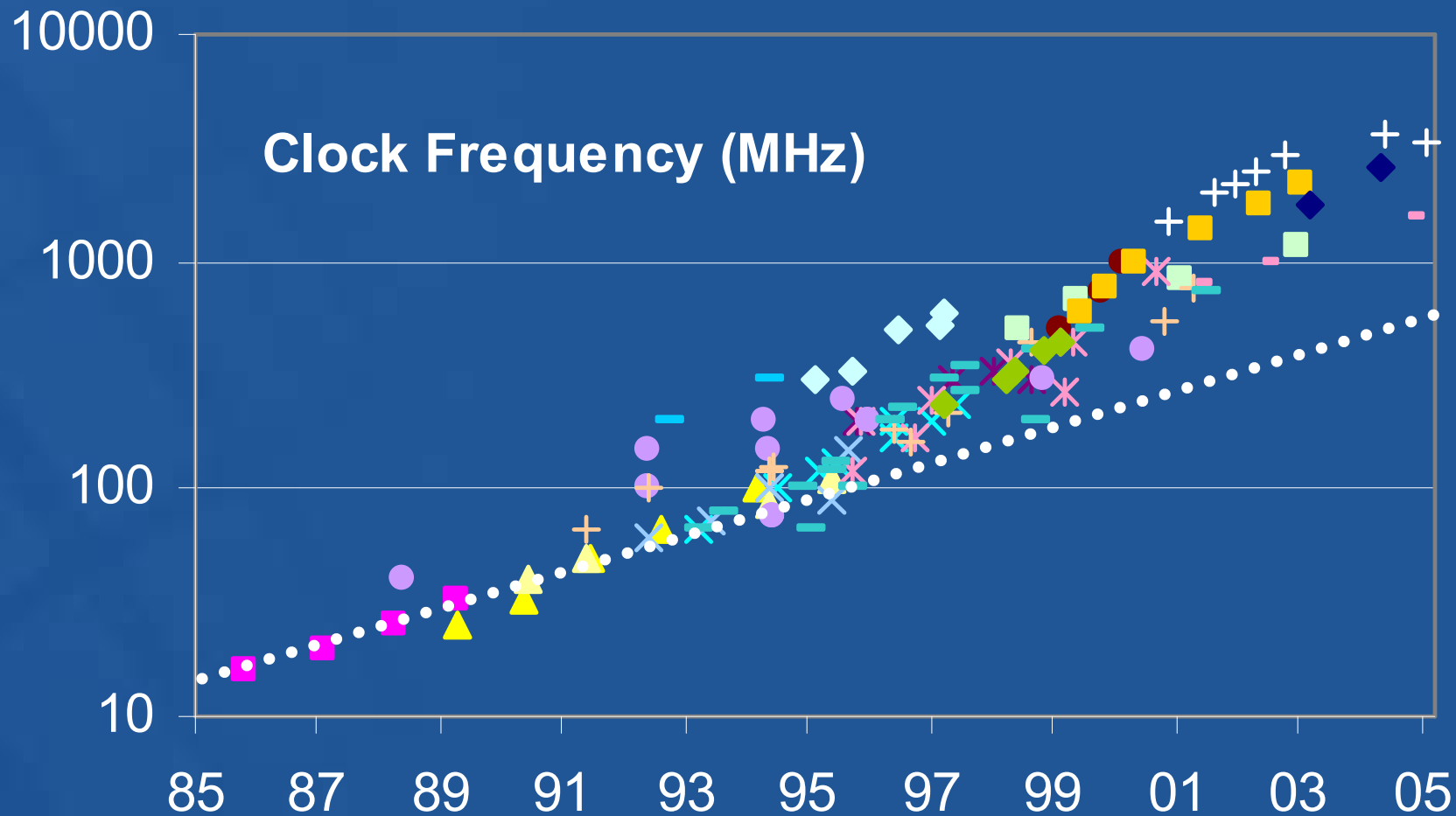
Processor Power

- Continued to grow, even when V_{dd} was scaled



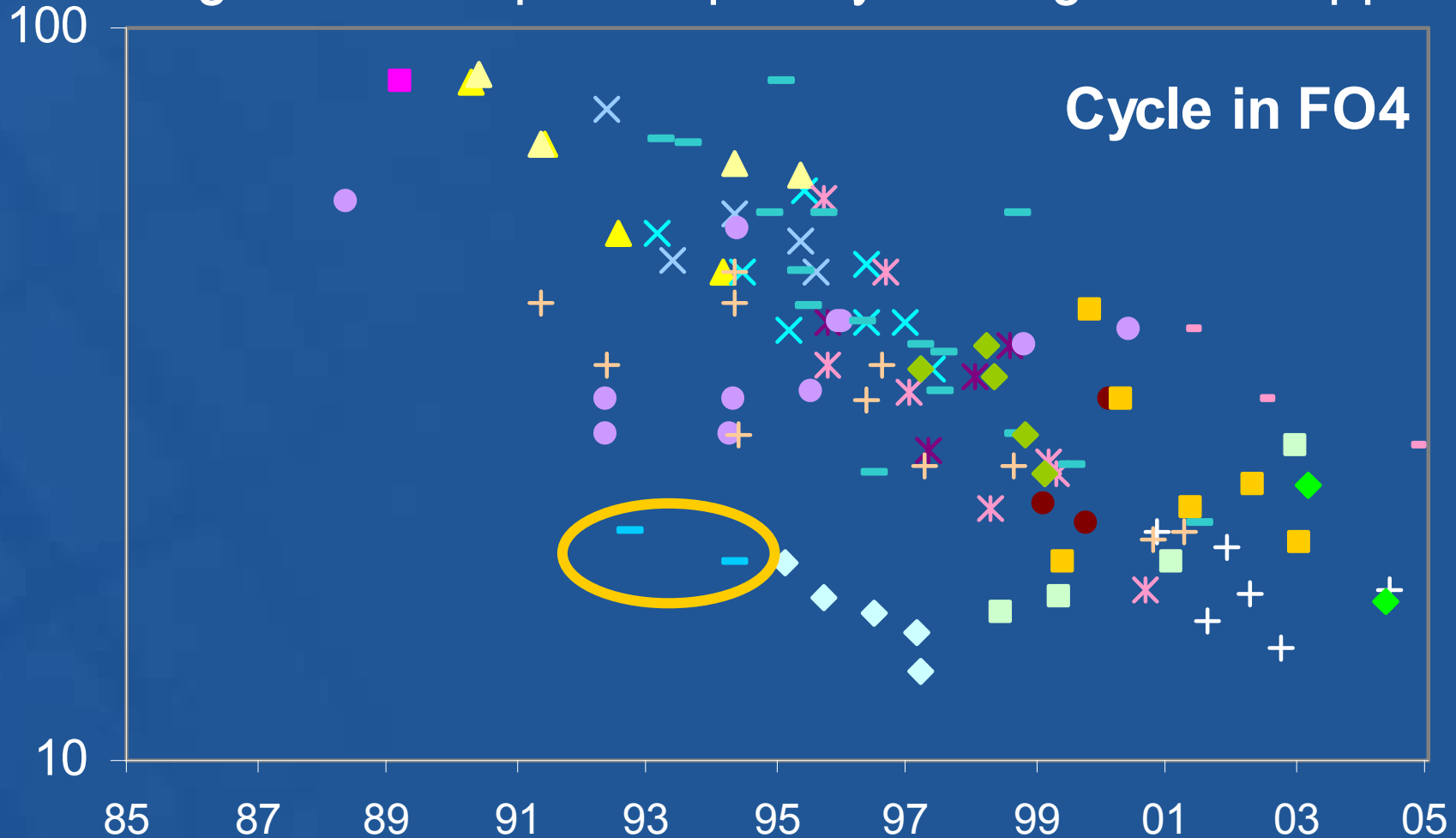
Why Power Increased

- Growing die size, fast frequency scaling



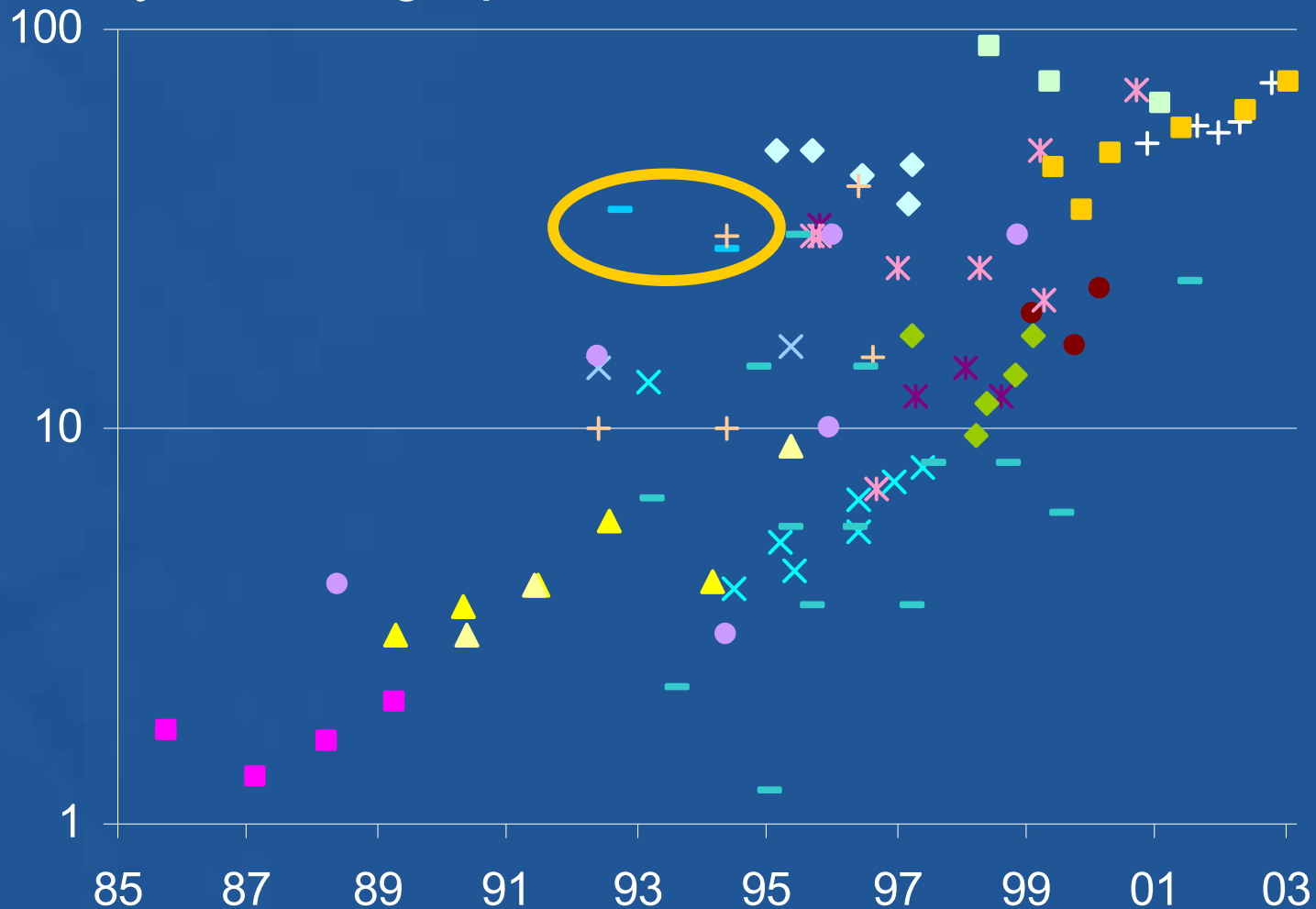
Good News

- Die growth & super frequency scaling have stopped



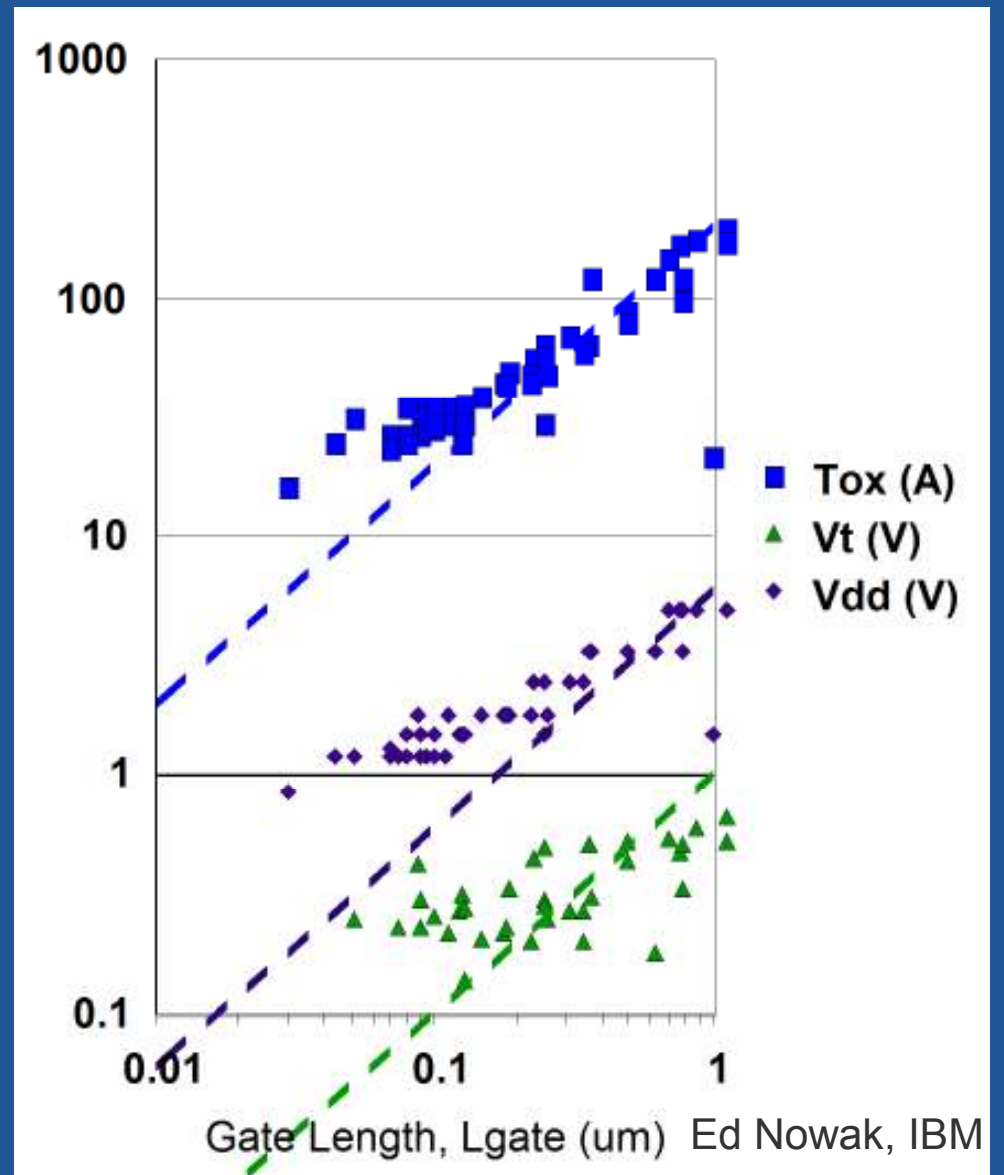
Processor Power

- They were high power too



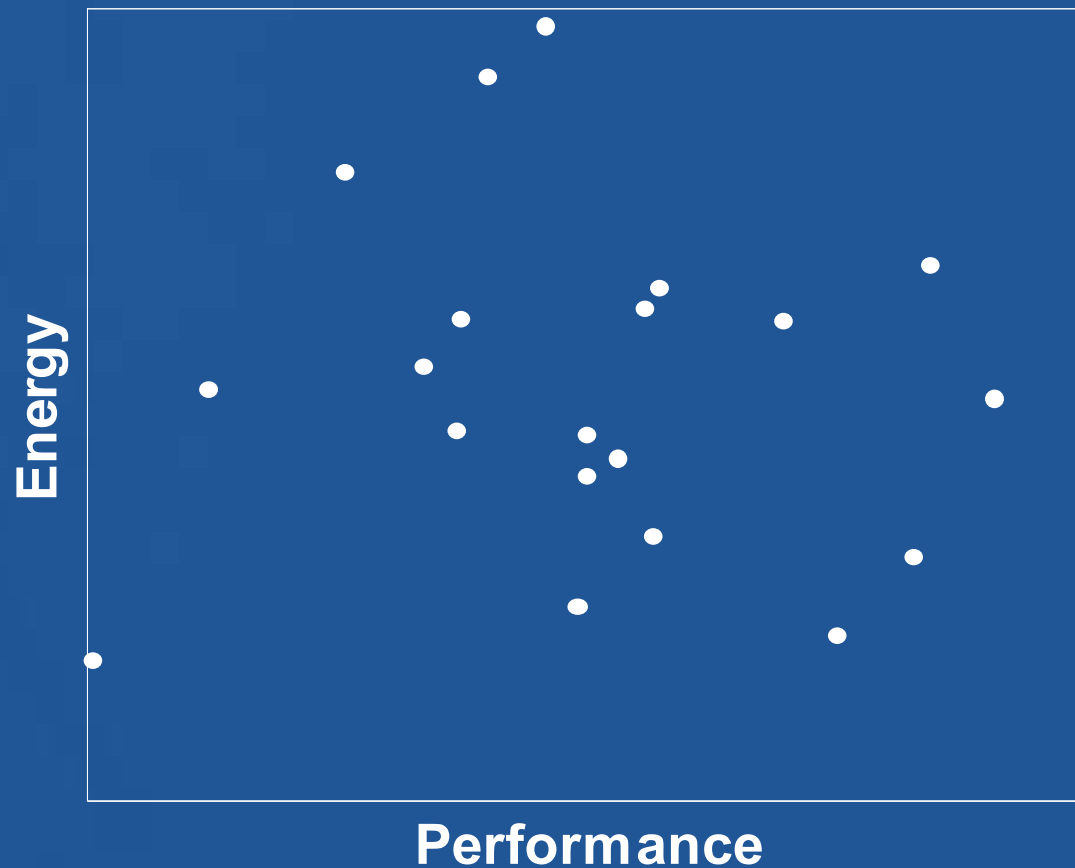
Bad News

- Voltage scaling has stopped as well
 - kT/q does not scale
 - V_{th} scaling has power consequences
- If V_{dd} does not scale
 - Energy scales slowly



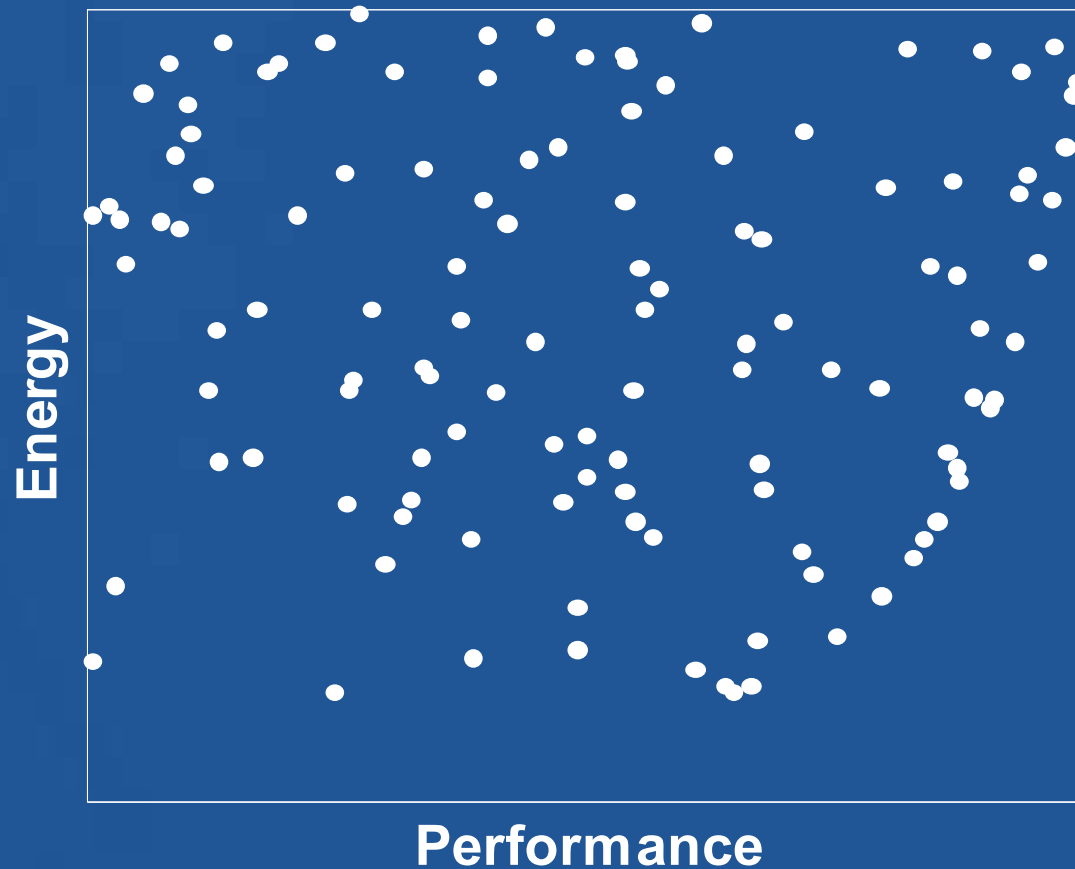
Energy – Performance Space

- Every design is a point on a 2-D plane



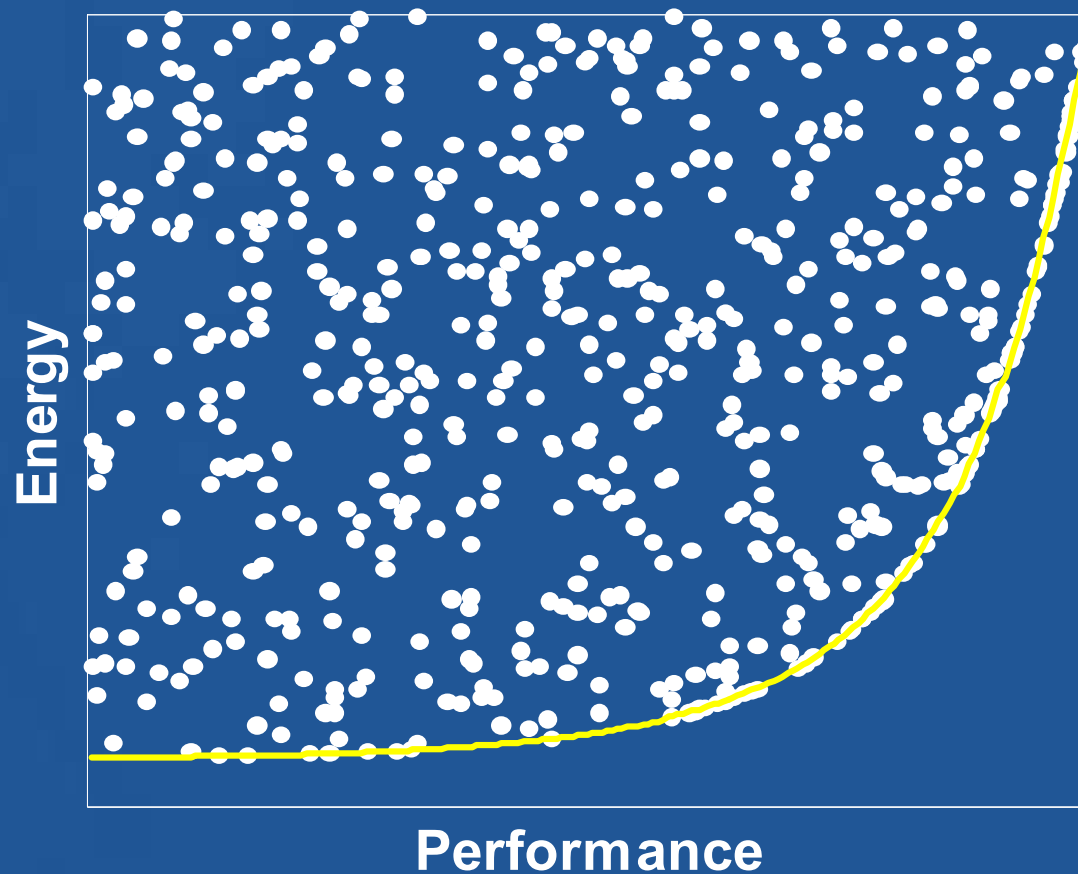
Energy – Performance Space

- Every design is a point on a 2-D plane

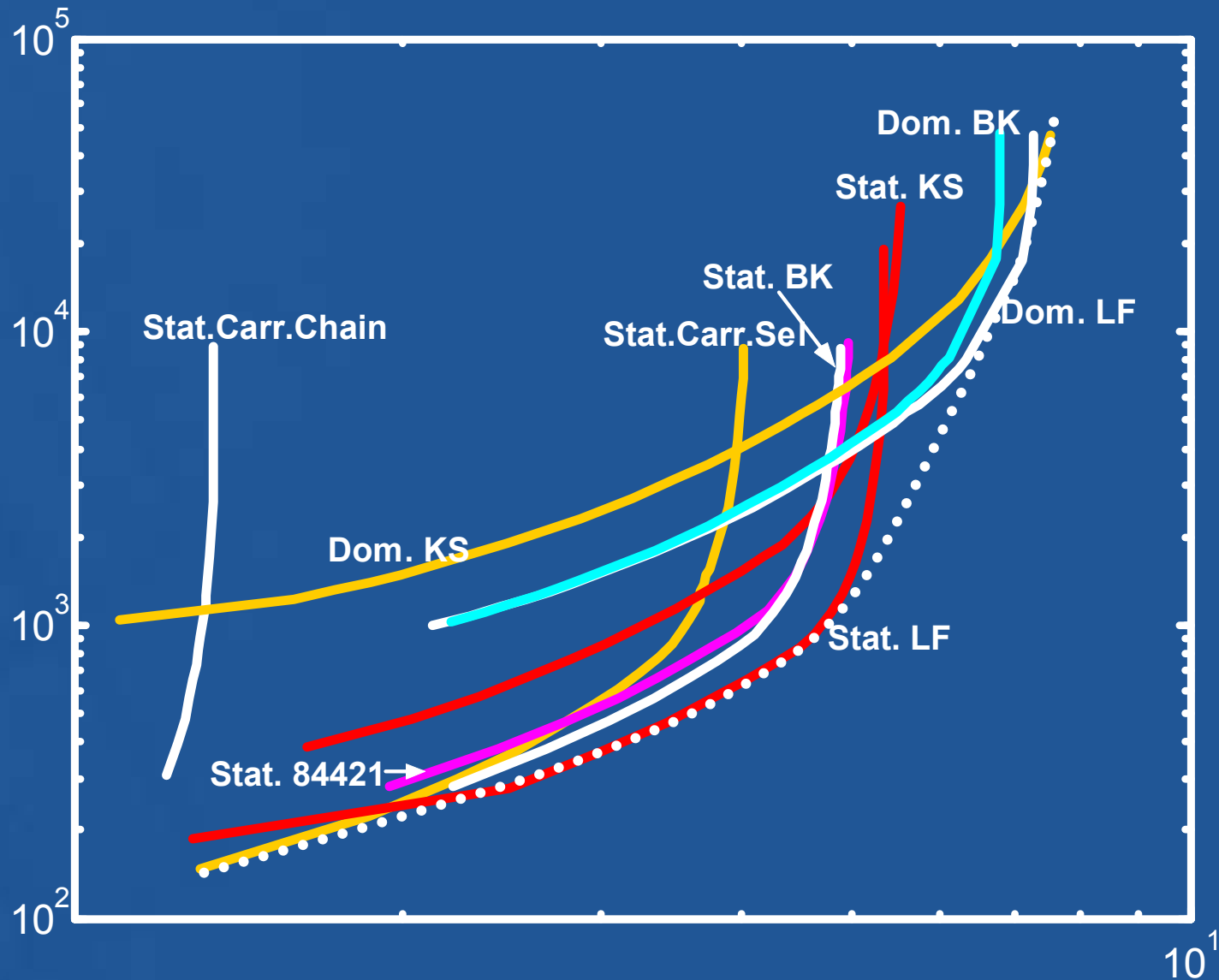


Energy – Performance Space

- Every design is a point on a 2-D plane



Trade-offs for an Adder



Key Observation:

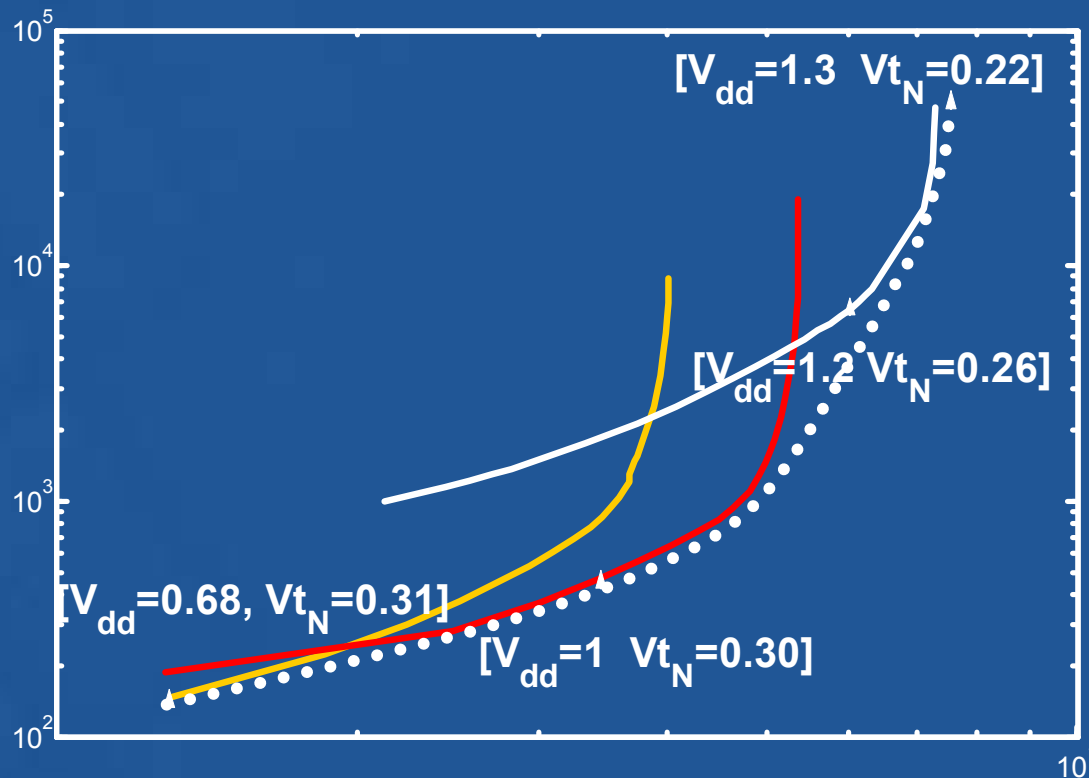
- Define the Energy/Delay sensitivity of parameter
 - For example V_{dd}:

$$Sens(V_{dd}) = - \frac{\frac{\partial E}{\partial V_{dd}}}{\frac{\partial D}{\partial V_{dd}}} \Bigg|_{V_{dd} = V_{dd}^*}$$

- At optimal point, all sensitivities should be the same
 - Must equal the slope of the Pareto optimal curve

What This Means

- V_{dd} and V_{th} are not directly set by scaling
 - Instead set by slope of Pareto optimal curve
 - Leakage rose to lower total system power!



Low Power Design Techniques

Three main classes of methods to reduce energy:

- Cheating
 - Reducing the performance of the design
- Reducing waste
 - Stop using energy for stuff that does not produce results
 - Stop waiting for stuff that you don't need (parallelism)
- Problem reformulation
 - Reduce work (less energy and less delay)

Cheating

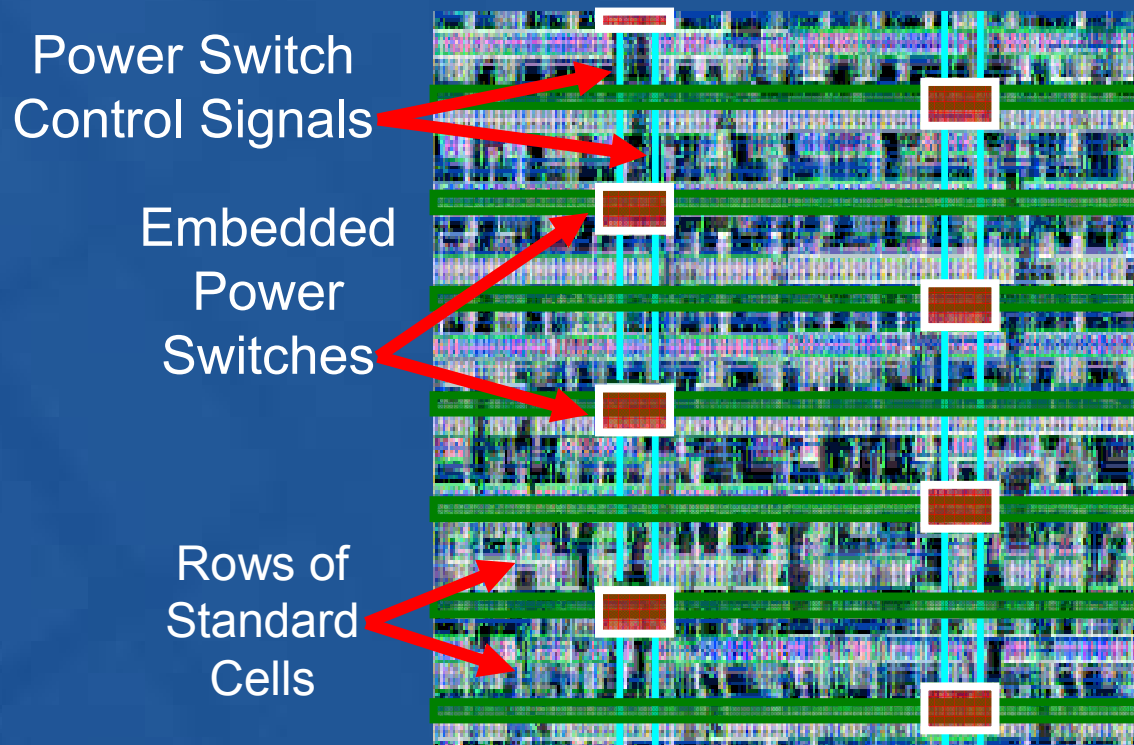
- Many low-power papers talk only about energy
 - Don't consider performance
- Reducing performance can always reduce energy
 - But there are many ways to reduce performance
- Good technique must lower the optimal curve
 - “Sensitivity” of technique
 - Must be better than current curve
 - This depends on location on the curve

Reducing Energy Waste

- Clock gating
 - If a section is idle, remove clock
 - Removes clock power
 - Prevents any internal node from transitioning
- Create system power states
 - Turn on subsystems only when they are needed
 - Can have different “off” states
 - Power vs. wakeup time
 - Disk (do you stop it from spinning?)

Embedded Power Gating

- Since transistors still leak when power is off

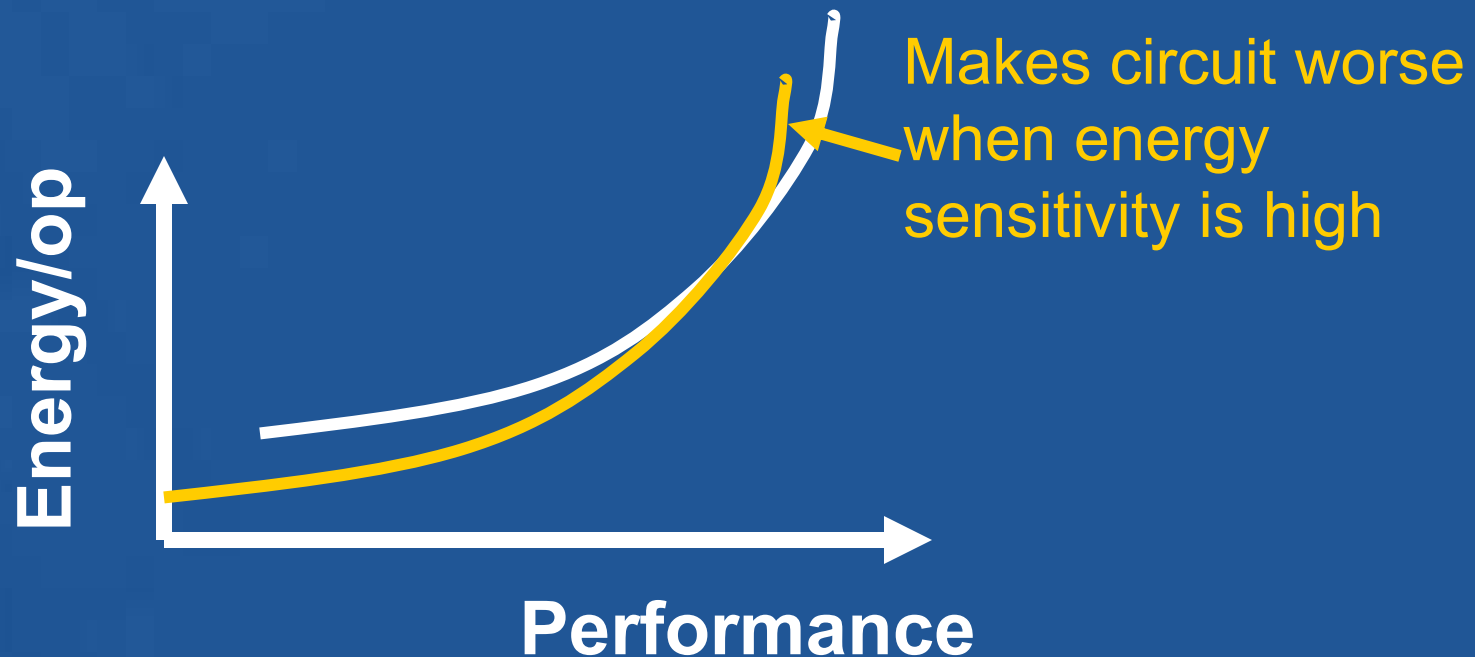


- Can reduce leakage
 - 250x reported
- But costs
 - Performance
 - Drop in Vdd, Gnd

Royannez, et al, 90nm Low Leakage SoC Design Techniques for Wireless Applications, ISSCC 2005

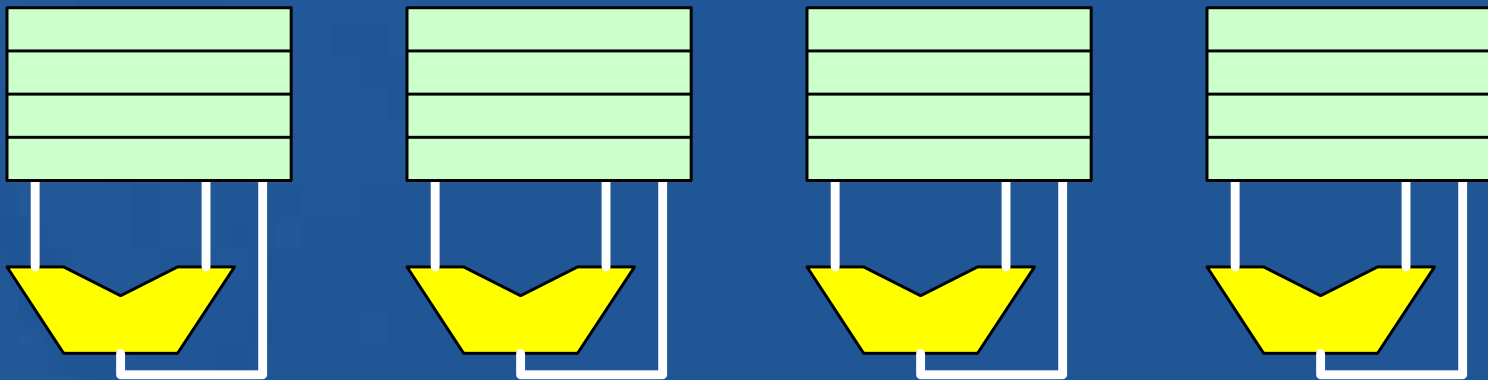
Range of Applicability

- Power supply gating
 - Done to remove leakage power
 - But slows down the circuit
 - Adds series resistance to the supply



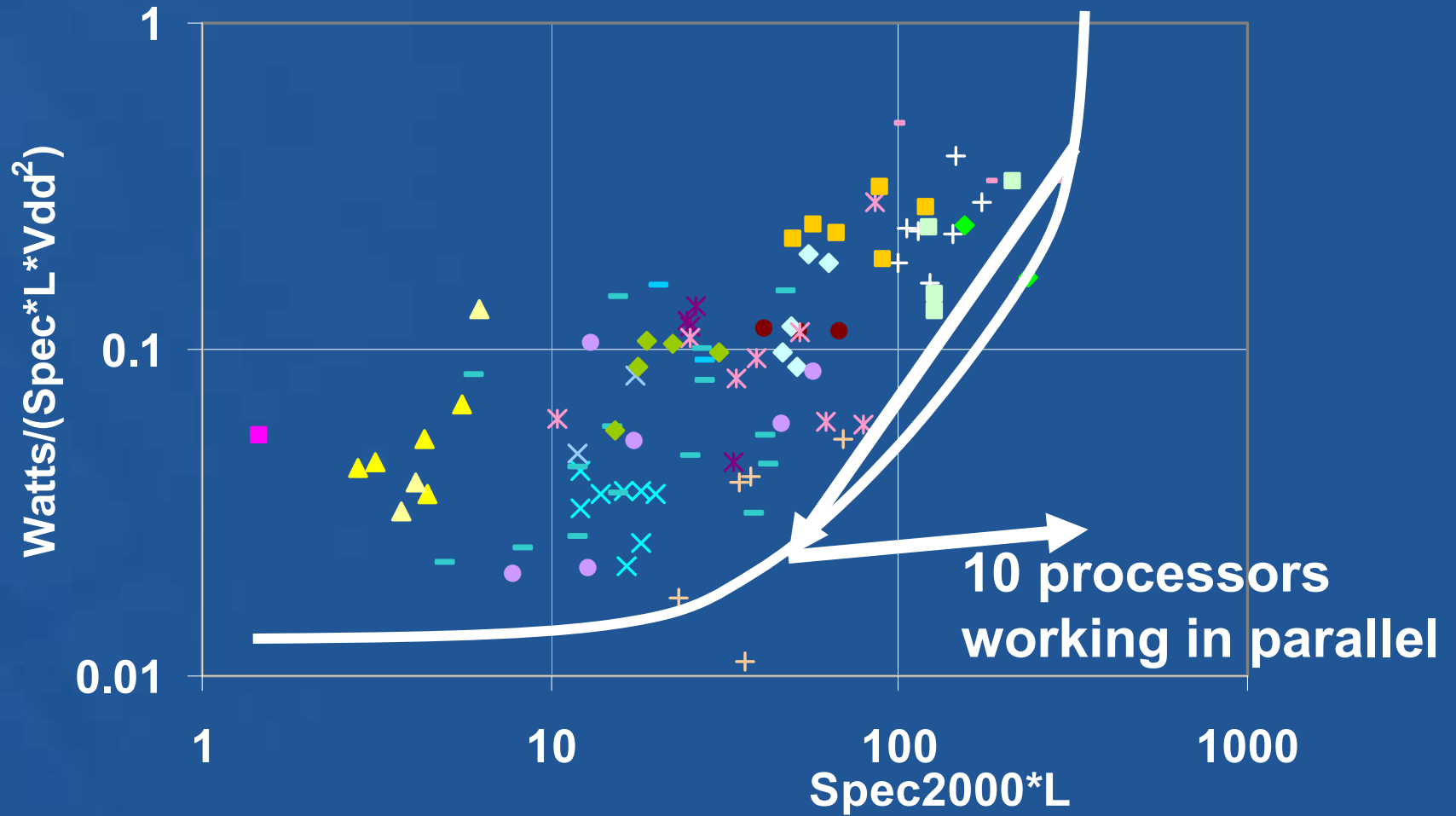
Parallelism

- If the application has data parallelism



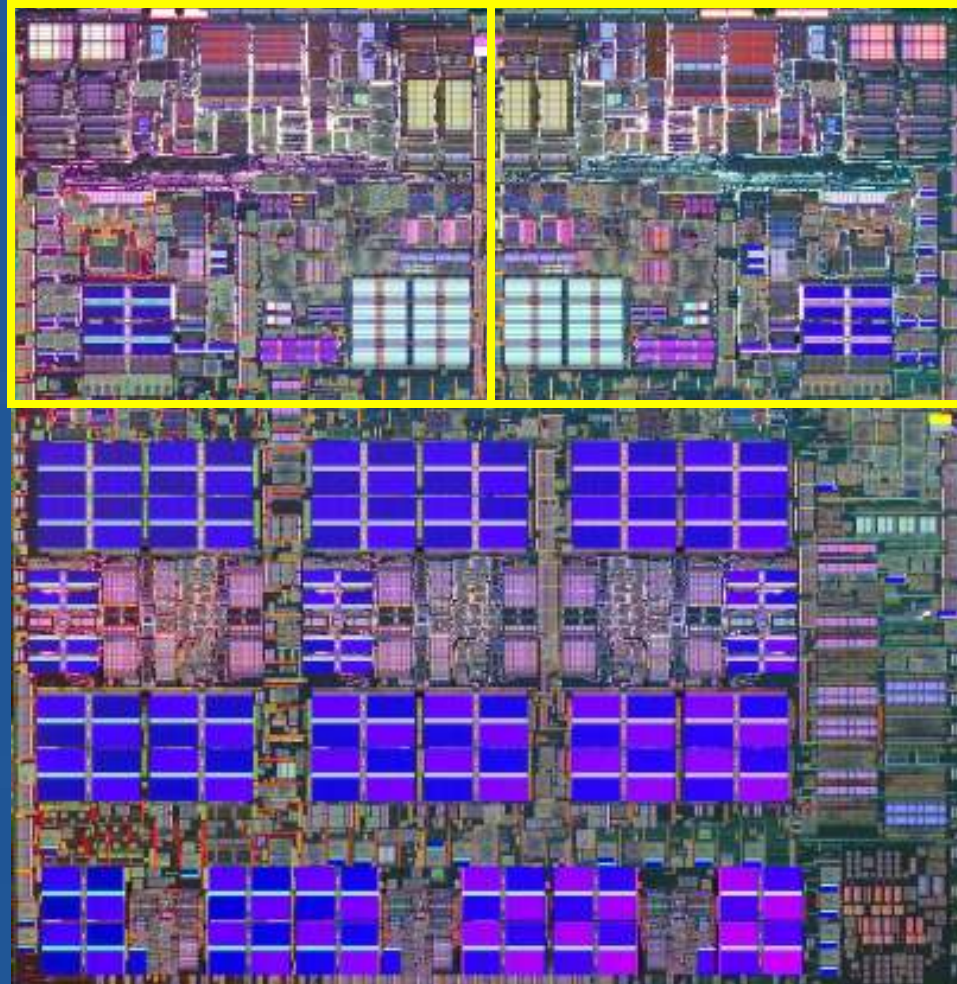
- Parallelism is a way to improve performance
 - With low additional energy cost

Existing Processors



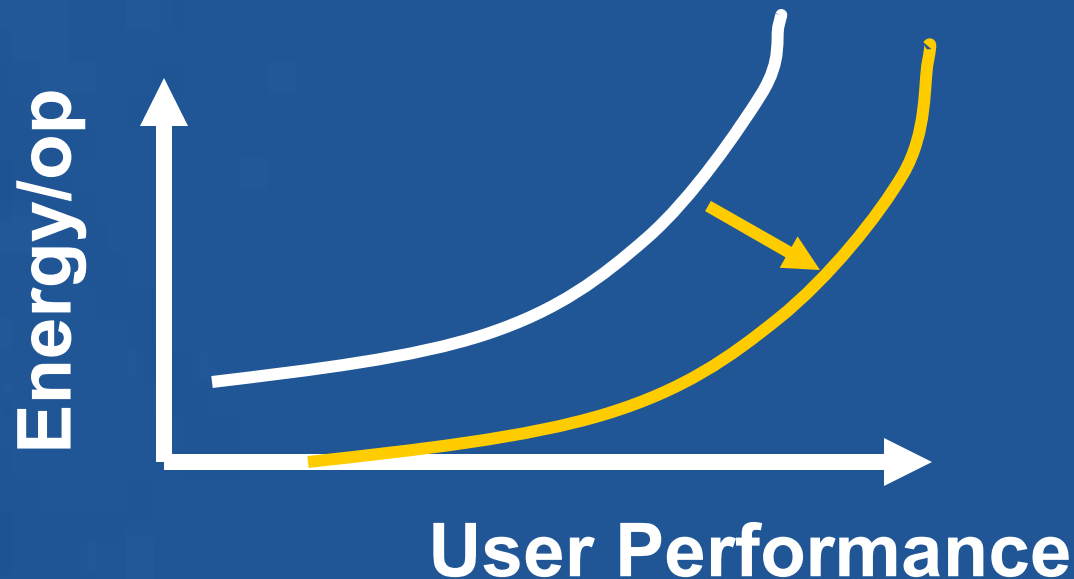
Parallel Server Chip

- Power 5 from IBM



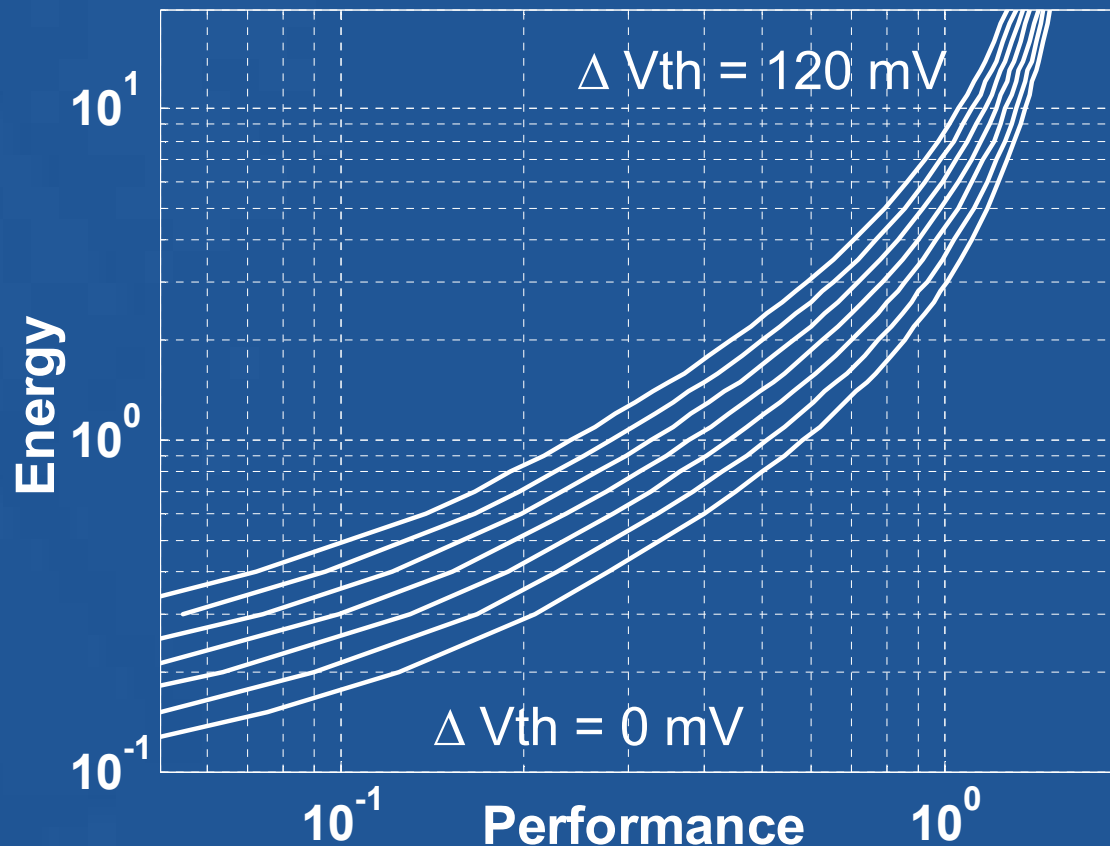
Problem Reformulation

- Best way to save energy is to do less work
 - Energy directly reduced by the reduction in work
 - But required time for the function decreases as well
 - Convert this into extra power gains
 - Shifts the optimal curve down and to the right



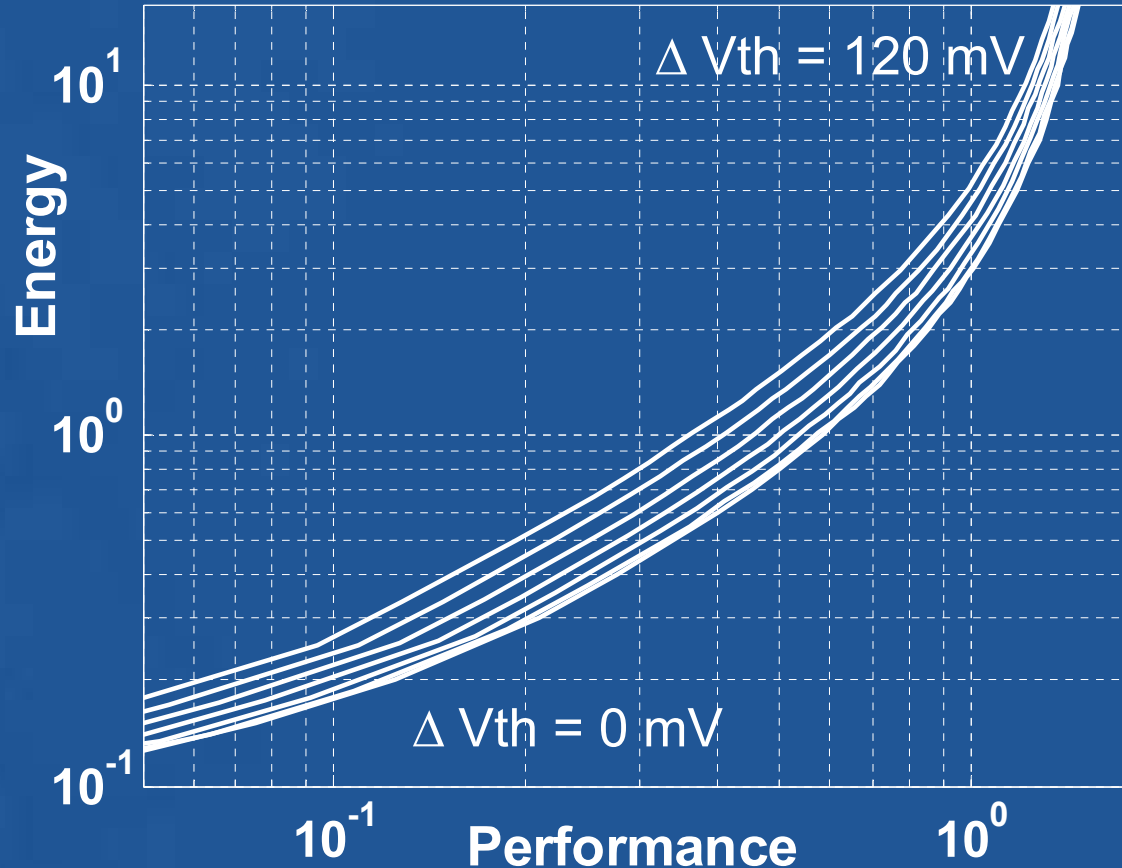
Cost of Variation

- Variability changes position of the optimal curves
 - Need to margin V_{th} , V_{dd} to ensure circuit always works



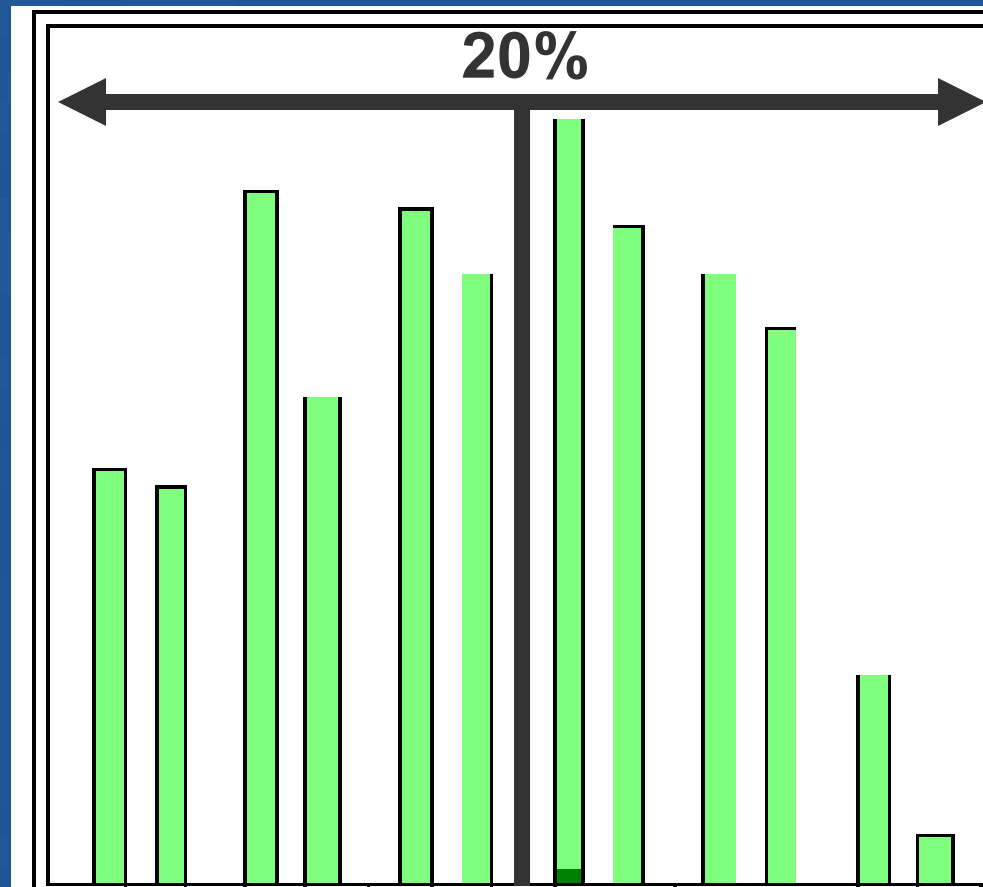
Partial Compensation

- Adjust V_{dd} after you get part back
 - Compensates very well for small deviations in V_{th}



Reducing Voltage Margins

- At test time determine V_{dd} for that part
 - Have private DC-DC converter already

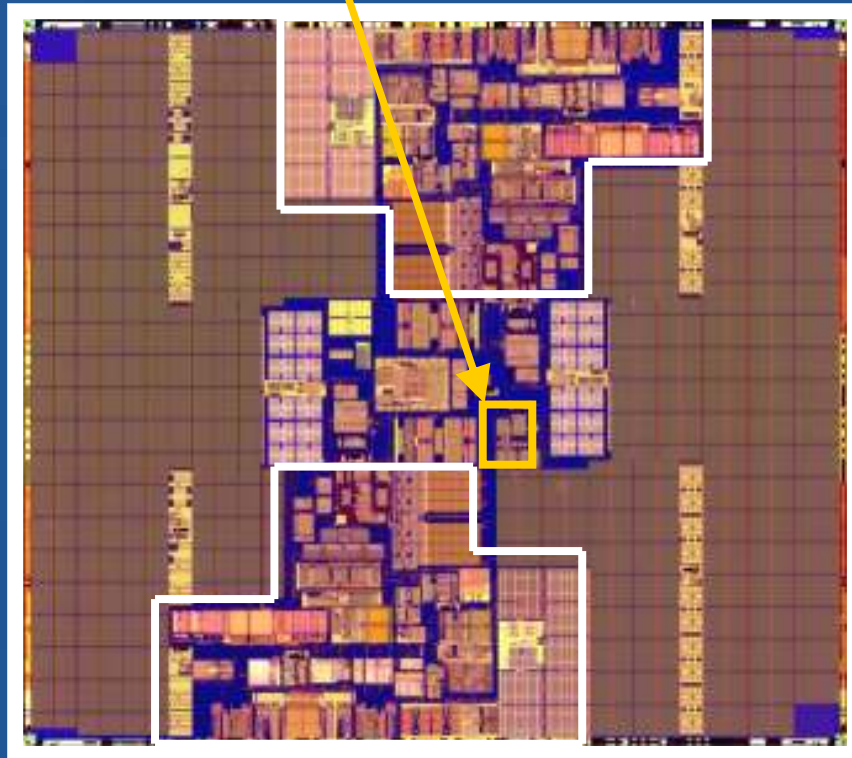


Variable Application Demands

- Try to provide a couple of operating points
 - Application can control speed and energy
 - Hard question is what are valid Vdd, F pairs
 - Usually determined during test
- Dynamic voltage scaling
 - Intel Speed Step in laptop processors
 - 2 performance/power points
 - Transmeta Long Run Technology
 - Many operating points. Test data + formula

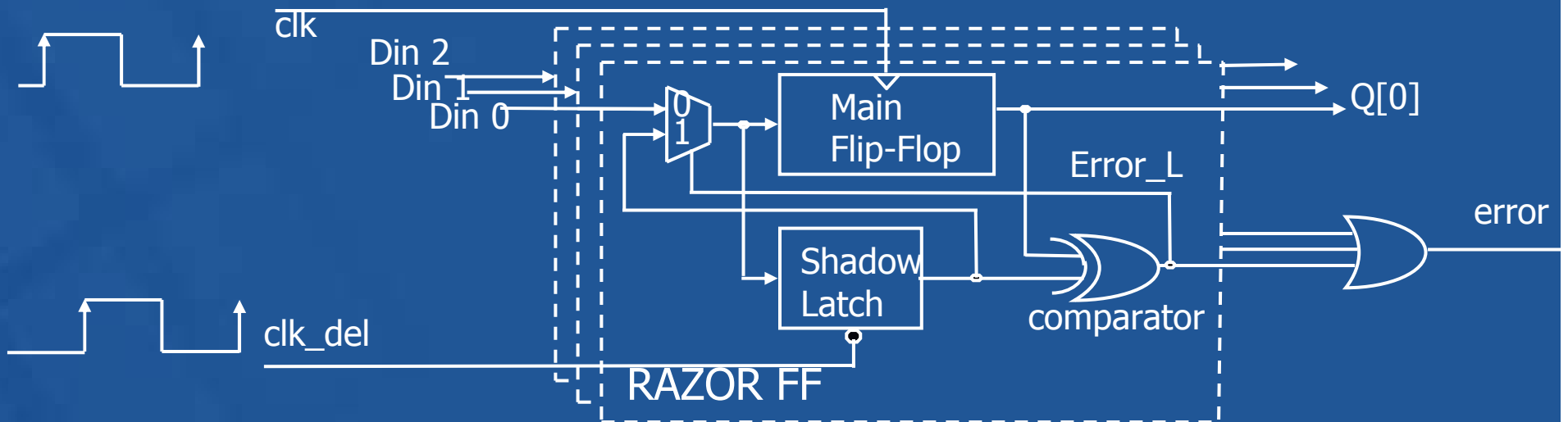
Constant Power Scaling

- Foxtan controller on next-gen. Itanium II
 - Raises V_{dd} /boosts F when most units idle
 - Lowers V_{dd} for parallel code to stay in budget



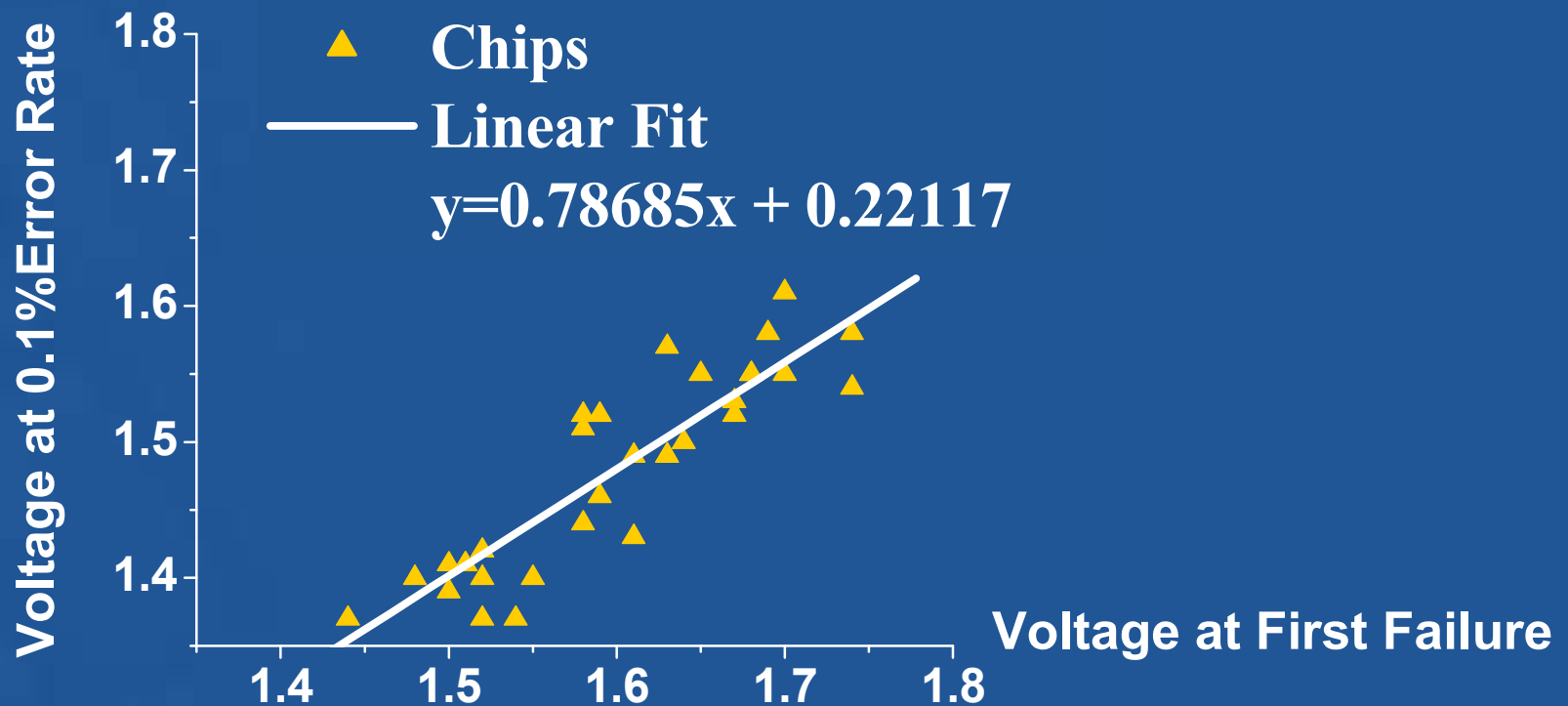
Self Checking Hardware

- Razor (Austin/Blaauw, U of Mich)
 - Use the actual hardware to check for errors
 - Latch the input data twice
 - Once on the clock edge, and then a little later
 - If the data is not the same, you are going too fast



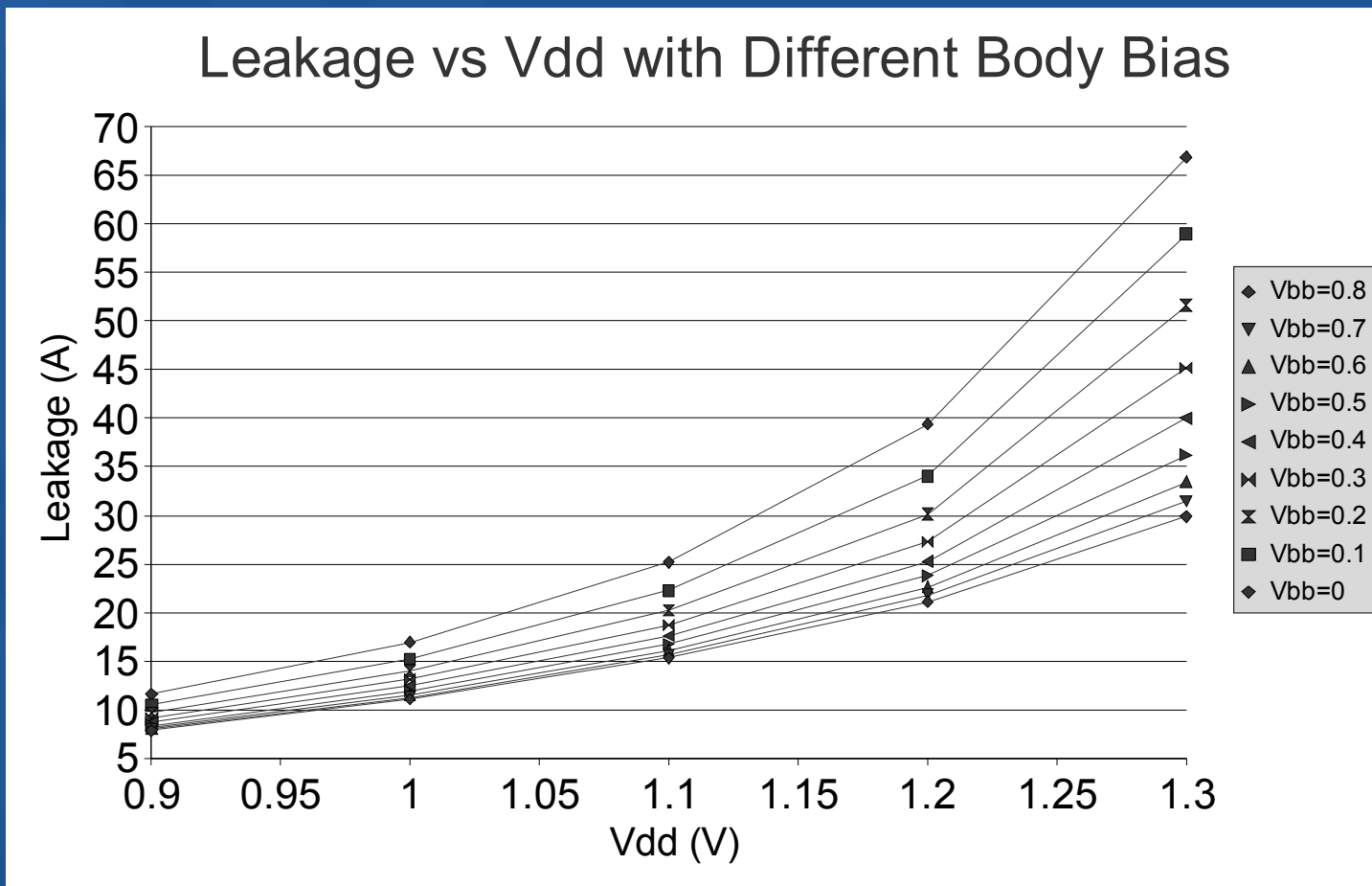
With Error Recovery

- Can run the chip so it makes some errors
 - Chip gets right answer 99.9% of the time
 - 0.1% of the time, the chip must rerun operation



Adjusting V_{th}

- In theory want to adjust V_{th} too
 - Very hard to do with modern transistors

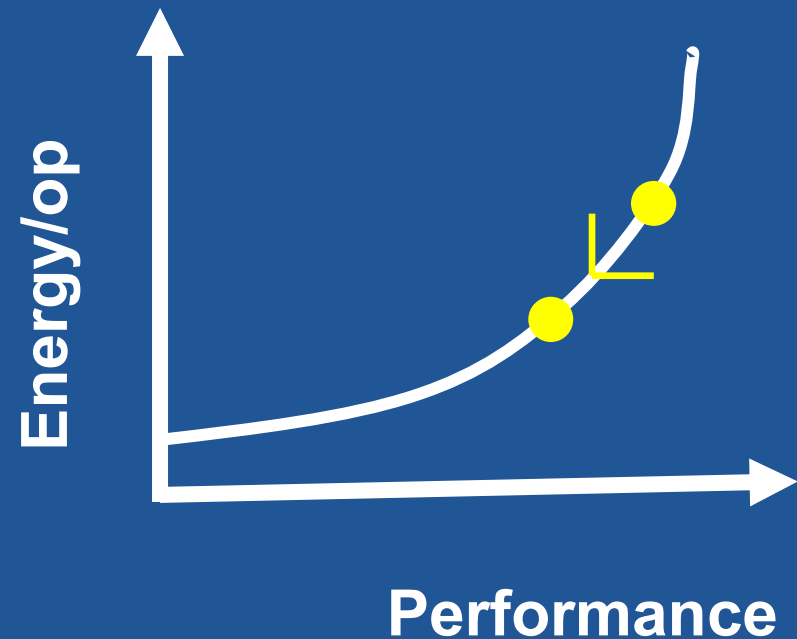


Future Systems

- Some simple math
 - Assume scaling continues
 - Dies don't shrink in size
 - Average power/gate must decrease by 2x / generation
- Since gates are shrinking in size
 - Get 1.4x from capacitive reduction
- Where is the other factor of 1.4x ?

Exploit Parallelism / Scale Vdd

- If you have parallelism
 - Add more function units
 - Fill up new die (2x)
 - Lower energy/op
 - $\Delta E/\Delta P$ will decrease
 - Vdd, sizes, etc will reduce
 - Build simpler architectures

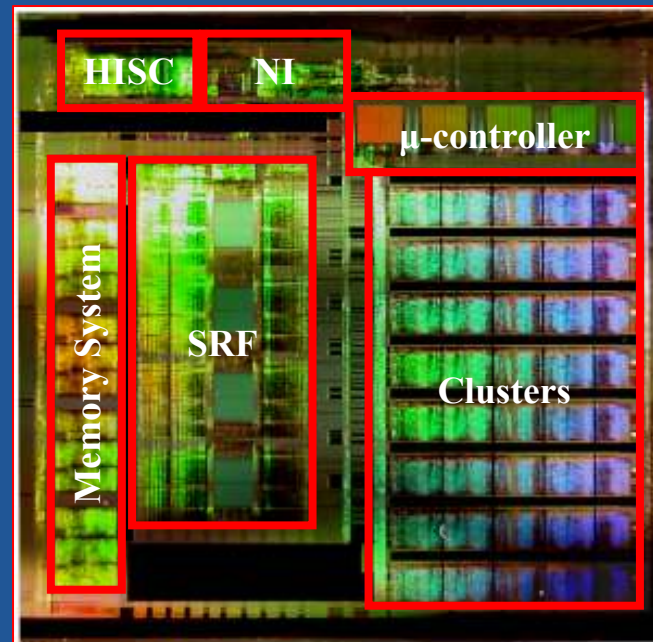


- Works well when $\Delta E/\Delta P$ is large
 - Per unit performance decrease is small

Exploit Specialization

- Optimize execution units for different applications
 - Reformulate the hardware to reduce needed work
 - Can improve energy efficiency for a class of applications
- Stream / Vector processing is a current example
 - Exploit locality, reuse
 - High compute density

Bill Dally et al, Stanford
Imagine



Exploit Integration

- If both those techniques don't work
 - Still can increase integration by at least 1.4x
- Moving units onto one chip
 - Reduces the number of I/Os on system
 - I/O can take significant power today
 - Allows even larger integration

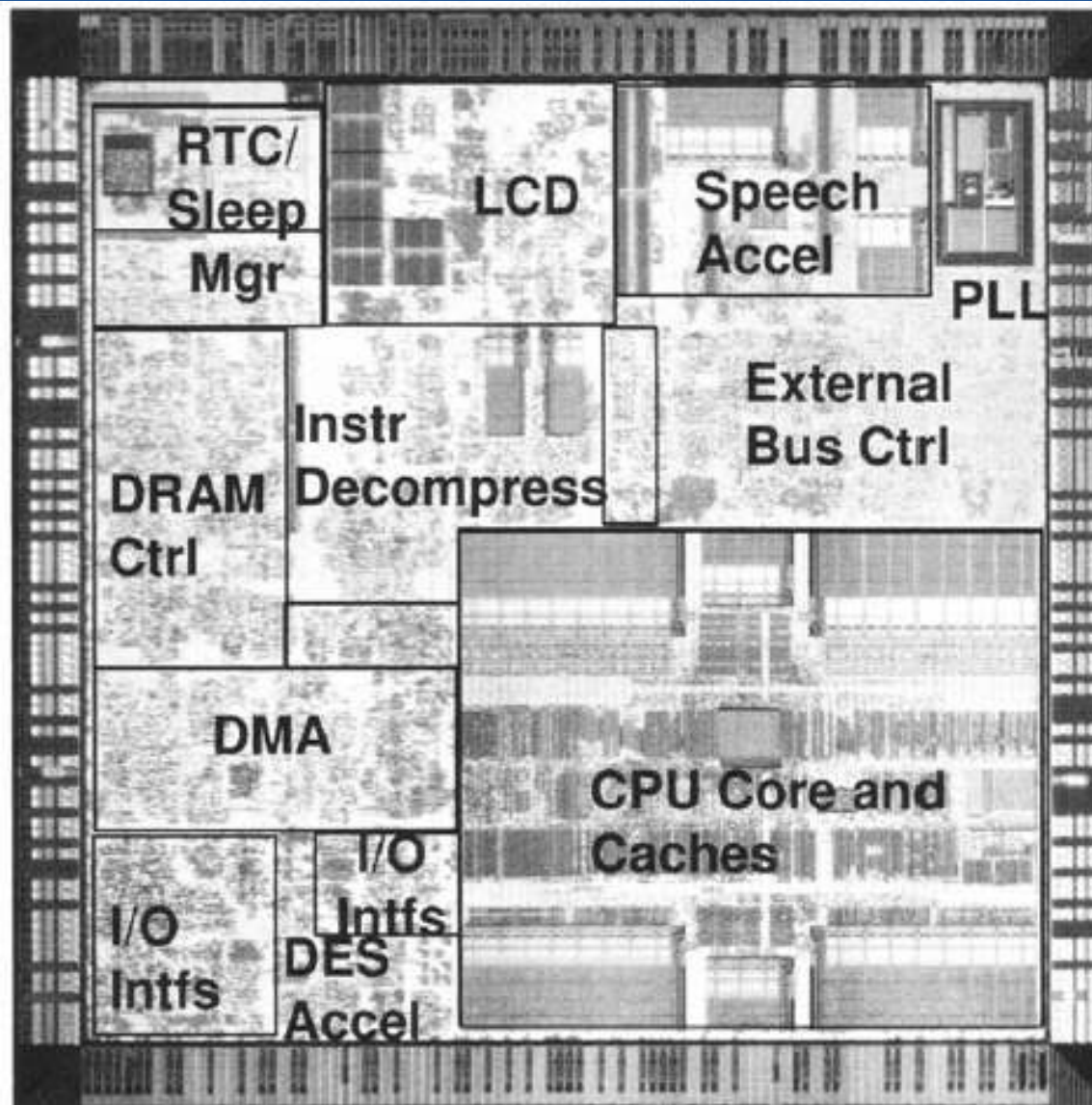
TI - OMAP2420



- Specialization
- And power domains
 - Most units are off
- OMAP 2420
 - 5 Power Domains
 - #1: MCU Core
 - #2: DSP Core
 - #3: Graphic Accelerator
 - #4: Core + Periph.
 - #5: Always On logic

Royannez, et al, 90nm Low Leakage SoC Design Techniques for Wireless Applications, ISSCC 2005

Low-Power PowerPC



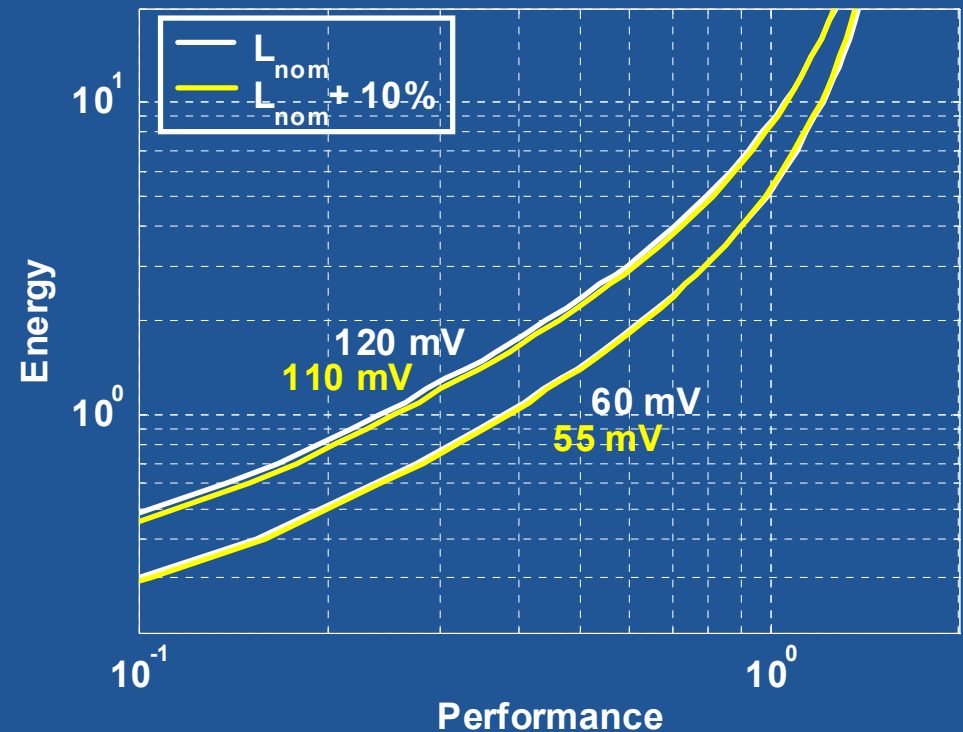
Nowka et al., Low-power PowerPC, ISSCC

What All This Means

- As long as \$/function and cap continue to scale
 - Moving to the new technology will be profitable
 - And will allow designs to be better systems
- In the worst case, active die area will decrease
 - Scale gates by the decrease in gate capacitance
- In most cases, we will do much better
- But how to optimize devices in this new domain?

Radical Idea:

- Scaling channel length may no longer be critical
 - I still want small (i.e. dense) devices
 - But I also want lower variations & external control of V_{th}
- Longer L_{eff} may actually improve energy efficiency
 - Less variability \rightarrow lower energy penalty
 - Especially as move to lower performance (parallelism)



Conclusions

- Unfortunately power is an old problem
 - Magic bullets have mostly been spent
- Power will be addressed by application-level optimization, parallelism/specialized functional units, and more adaptive control
- Need to rethink scaling
 - Still makes things cheaper
 - But what do we want from scaled transistors?

Technology Scaling

Seems simple,

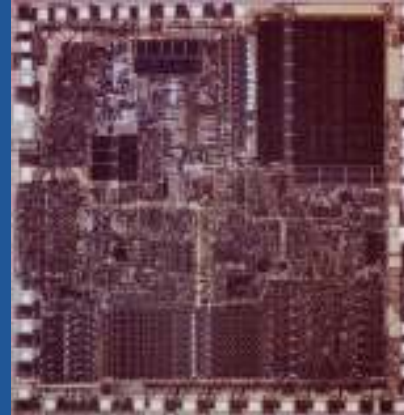
- Every ~~1~~ ~~1.5~~ 2 years
 - Number of transistors double
 - Transistors get faster
 - Gates become lower power (CMOS)
- Life just gets better and better

Reality is a Little Different

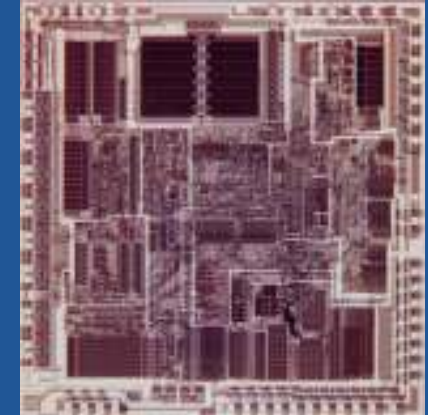
- While scaling has been smooth
 - Almost nothing else has been
- Device and circuit technology has changed
 - DTL, ECL, TTL, pMOS, nMOS, CMOS
- Power periodically becomes a critical issue
 - It is critical again

nMOS, TTL, ECL Were King

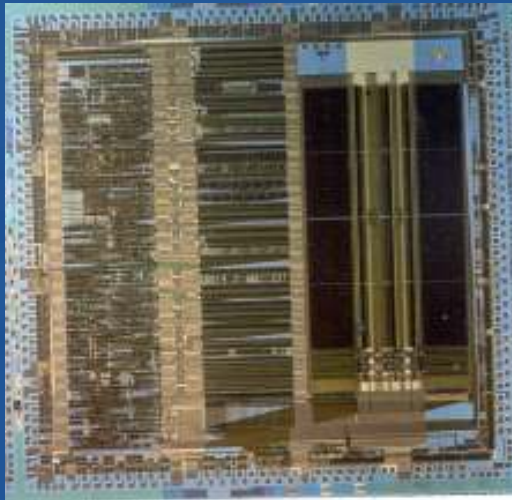
- 1978 – Started in VLSI
 - First design was bipolar/ECL
 - 3 μ m nMOS was hot



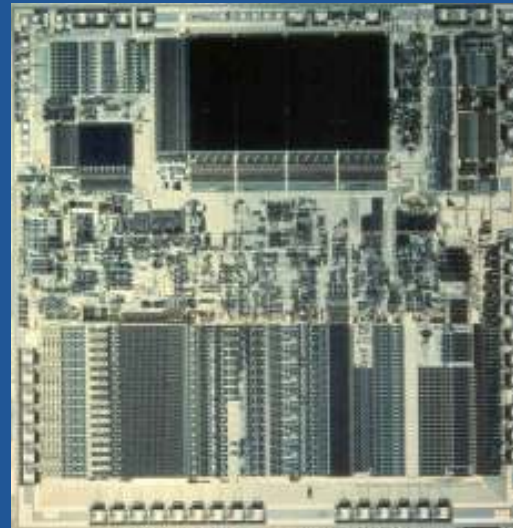
Intel 8086



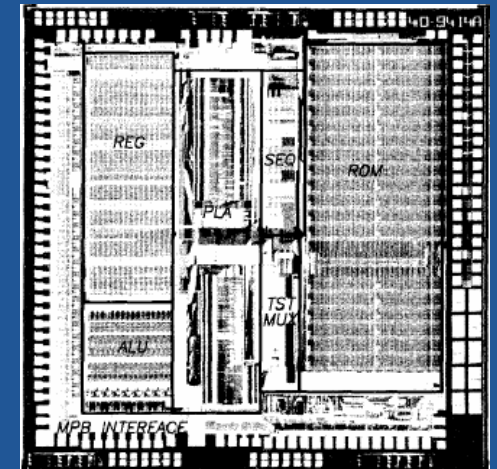
Intel 286



BIT Sparc



DEC μ Vax



HP Focus

MOS Scaling Was Understood

- MOS devices operate on electric fields
- If E fields are the same
 - Relation between E and J is the same
- So if all voltages and lengths scale
 - iV curve retains the same shape, scaled in V

Bob Dennard worked all the math in 74

JSSC Oct 74, pg 256

Dilemma

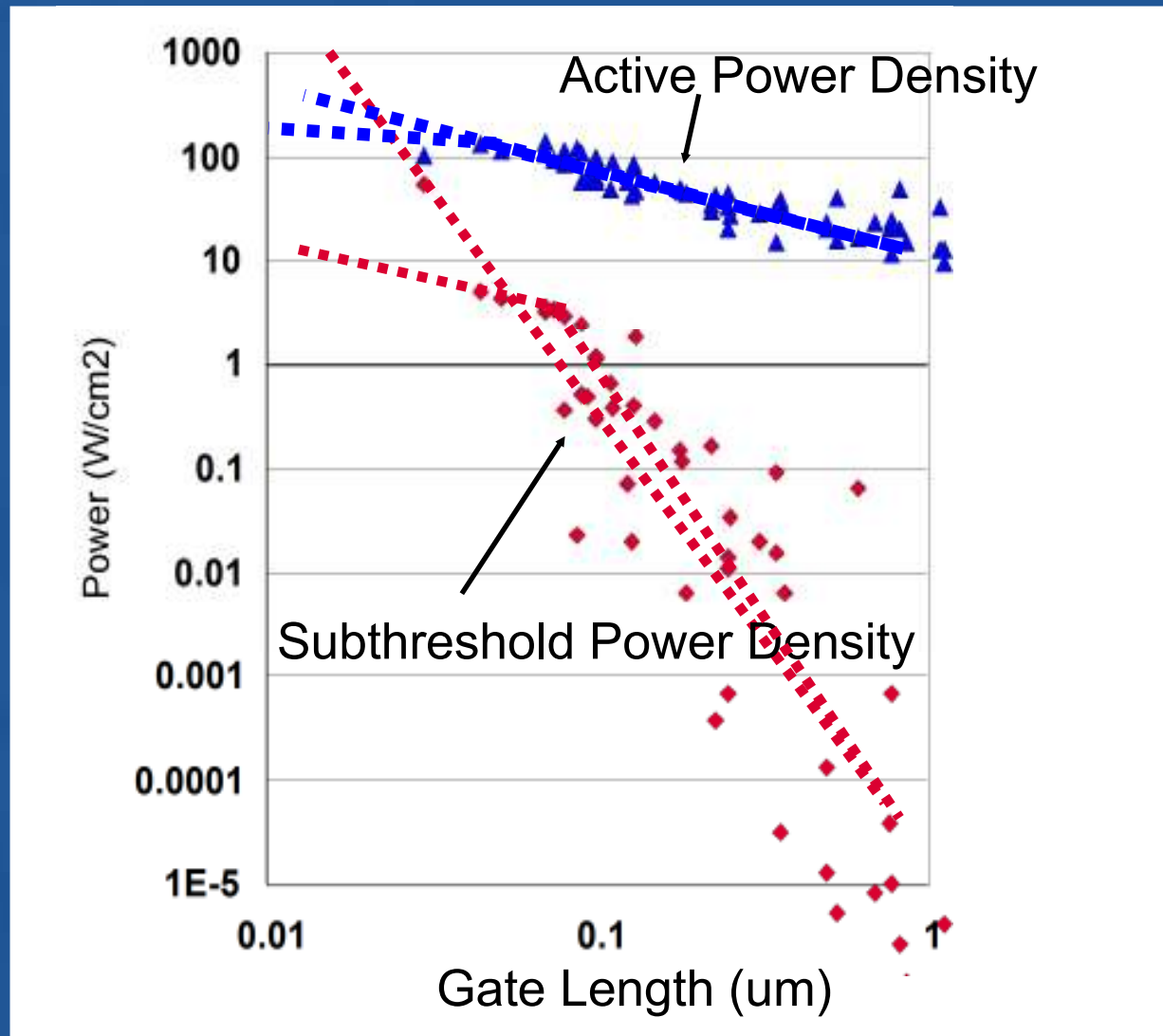
- Processors today are power limited
 - As are many other chips
- Technology scaling will not save us
 - With V_{dd} fixed, energy scaling will be modest
- How does one build more powerful processors?
 - Or other types of chips

When constrained, optimize!

Optimizing the Right Thing

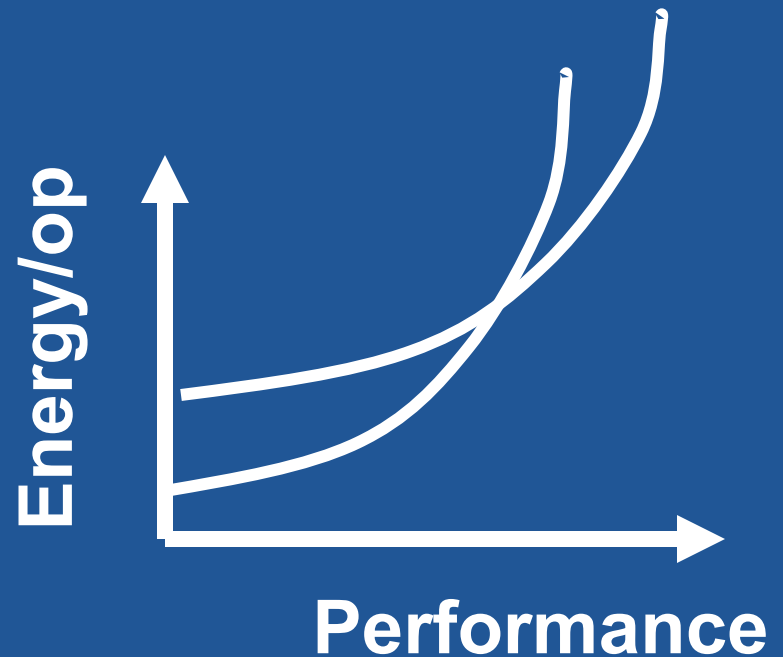
- Given systems are power limited
 - Highest performance system is not interesting
 - Will dissipate too much power
 - Lowest energy solution is also not interesting
 - Will not have enough performance
- Want constrained optimization
 - Highest performance for 20 Watts
 - Lowest power for 100 SPEC

Leakage Trends



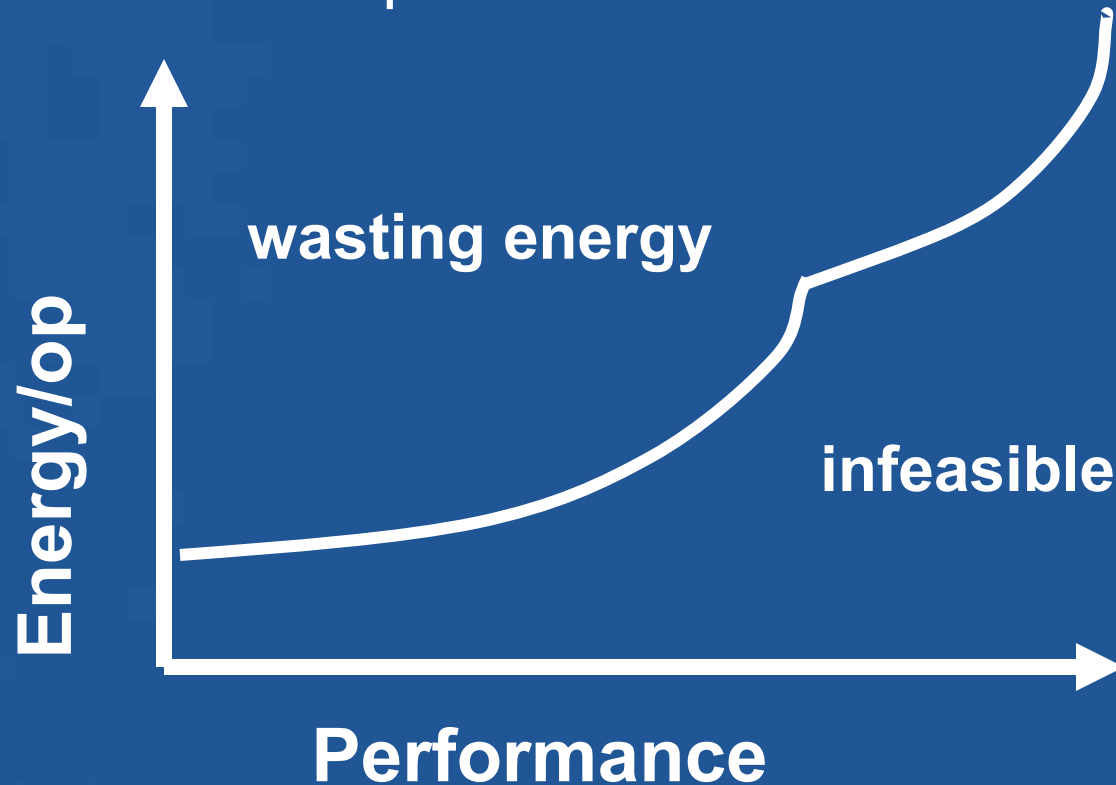
Design Parameters To Adjust

- Circuit
(sizing, supply, threshold)
- Circuit topology
(adder: CLA, CSA, ...)
- Logic style
(domino, pass-gate, ...)
- Micro-architecture
(pipelining, cache design, branch architecture, etc)



Energy Efficient Designs

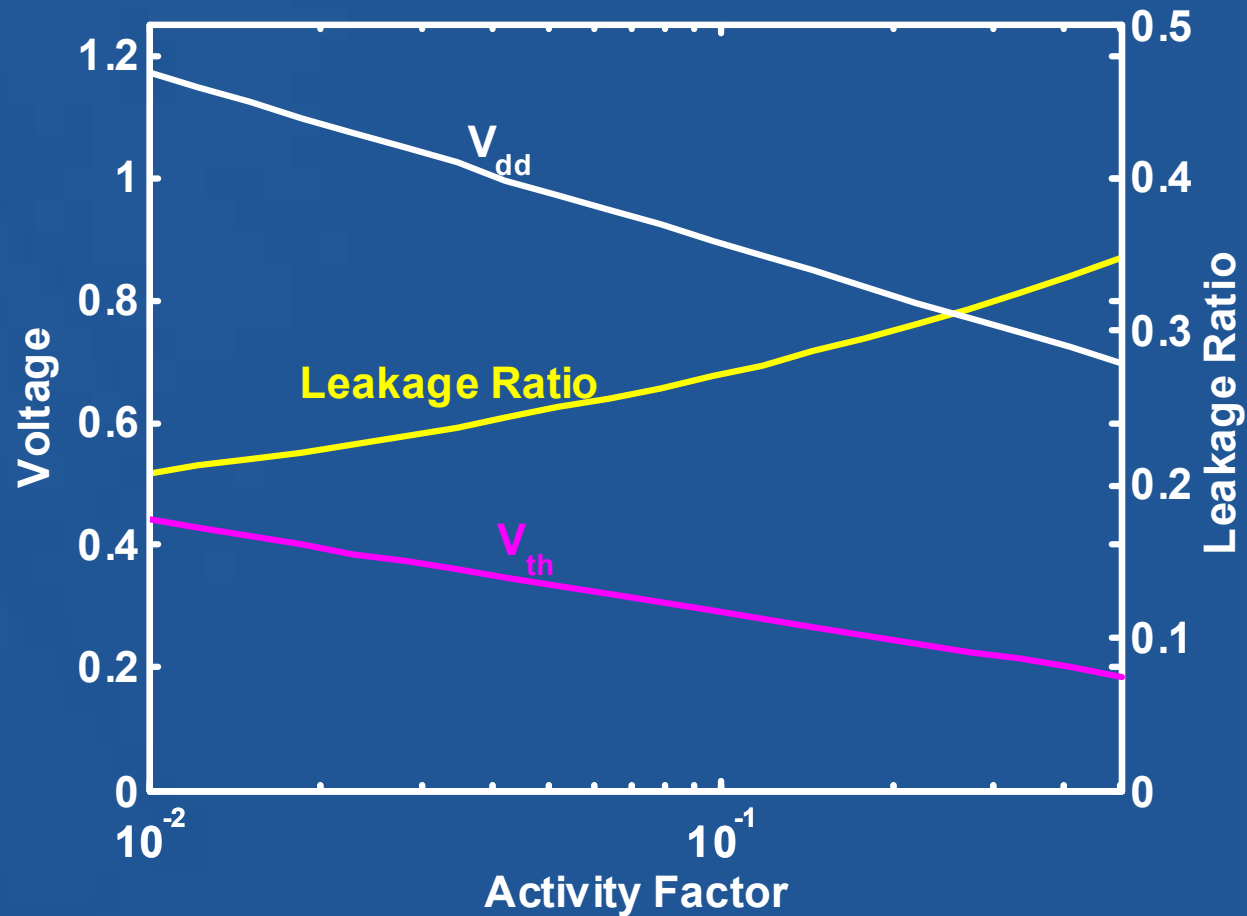
- Are on the Pareto optimal curve



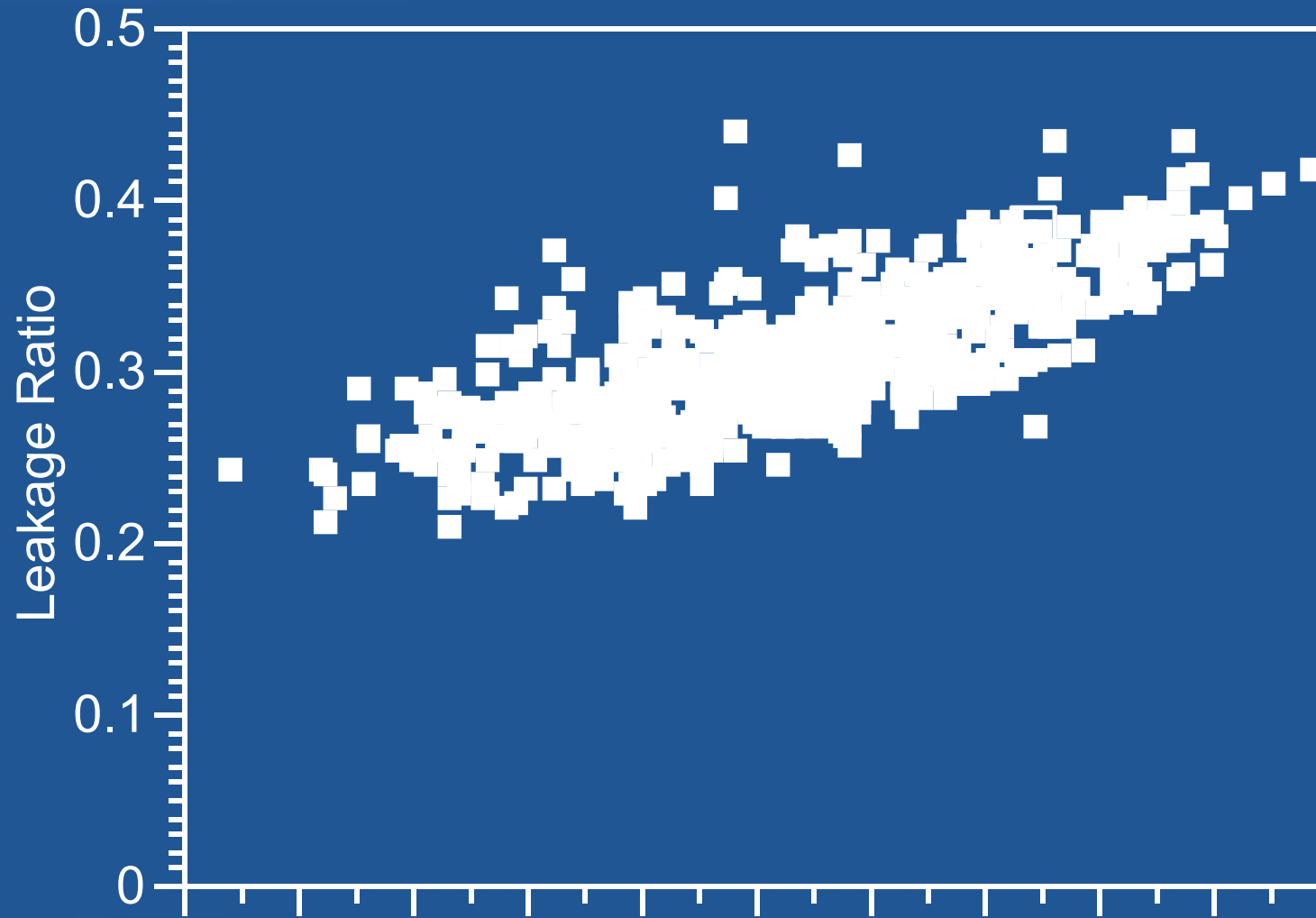
- On this curve design parameters are constrained

Leakage Energy

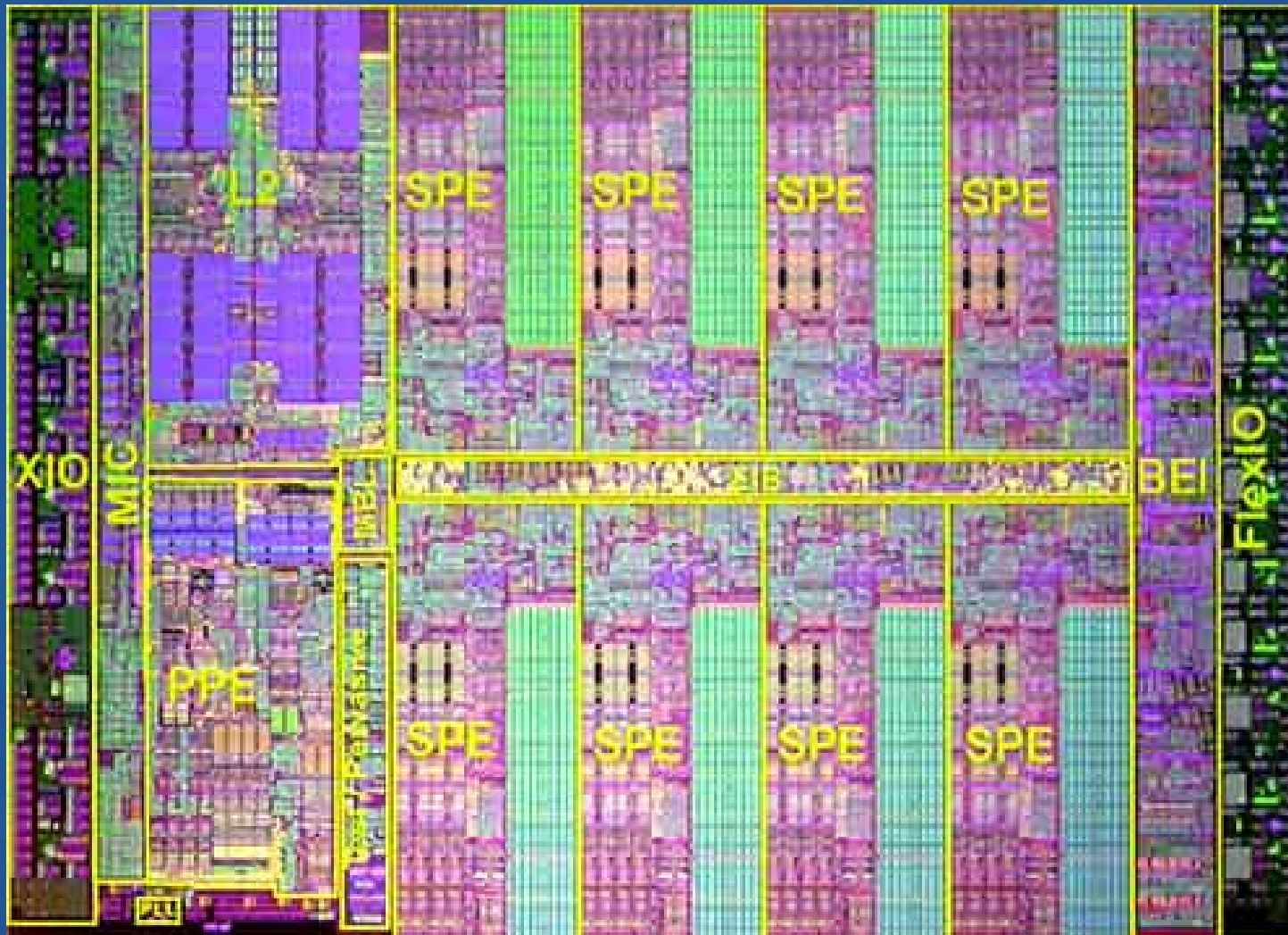
- Matching marginal costs for V_{dd} and V_{th}



Measured Leakage Data

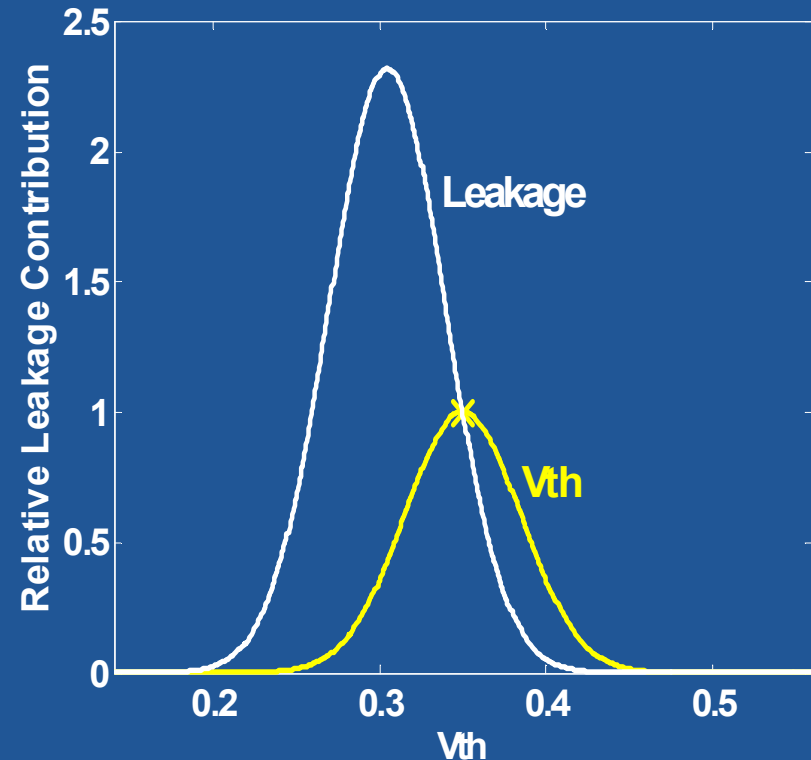
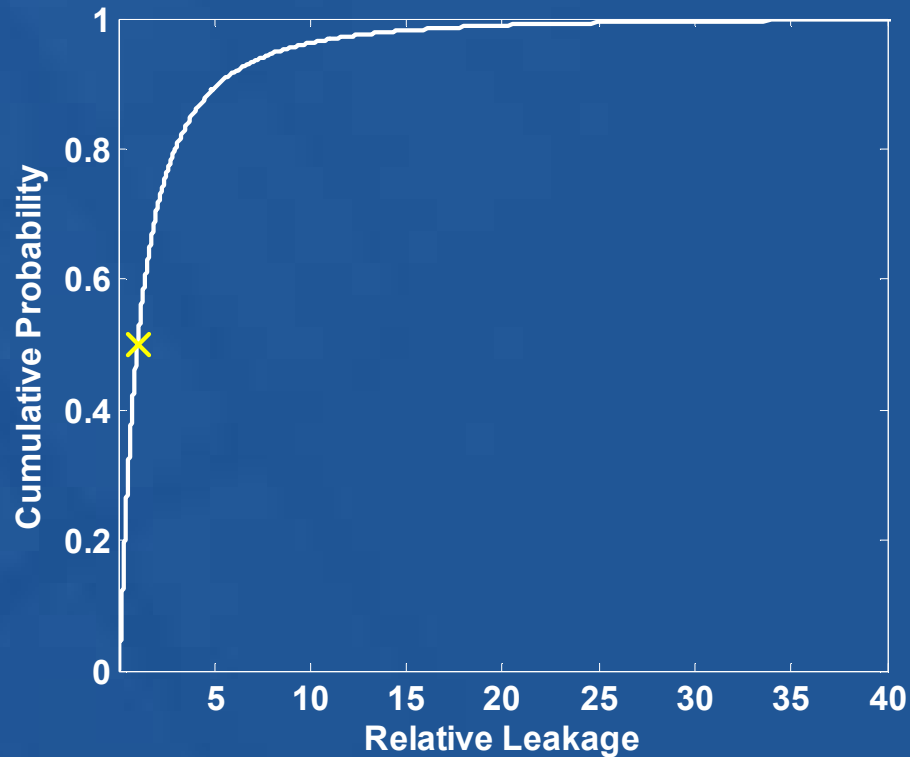


IBM Cell Processor



Vth Variation

- Since leakage is exponential on Vth
 - Average Vth for leakage is not the expected Vth

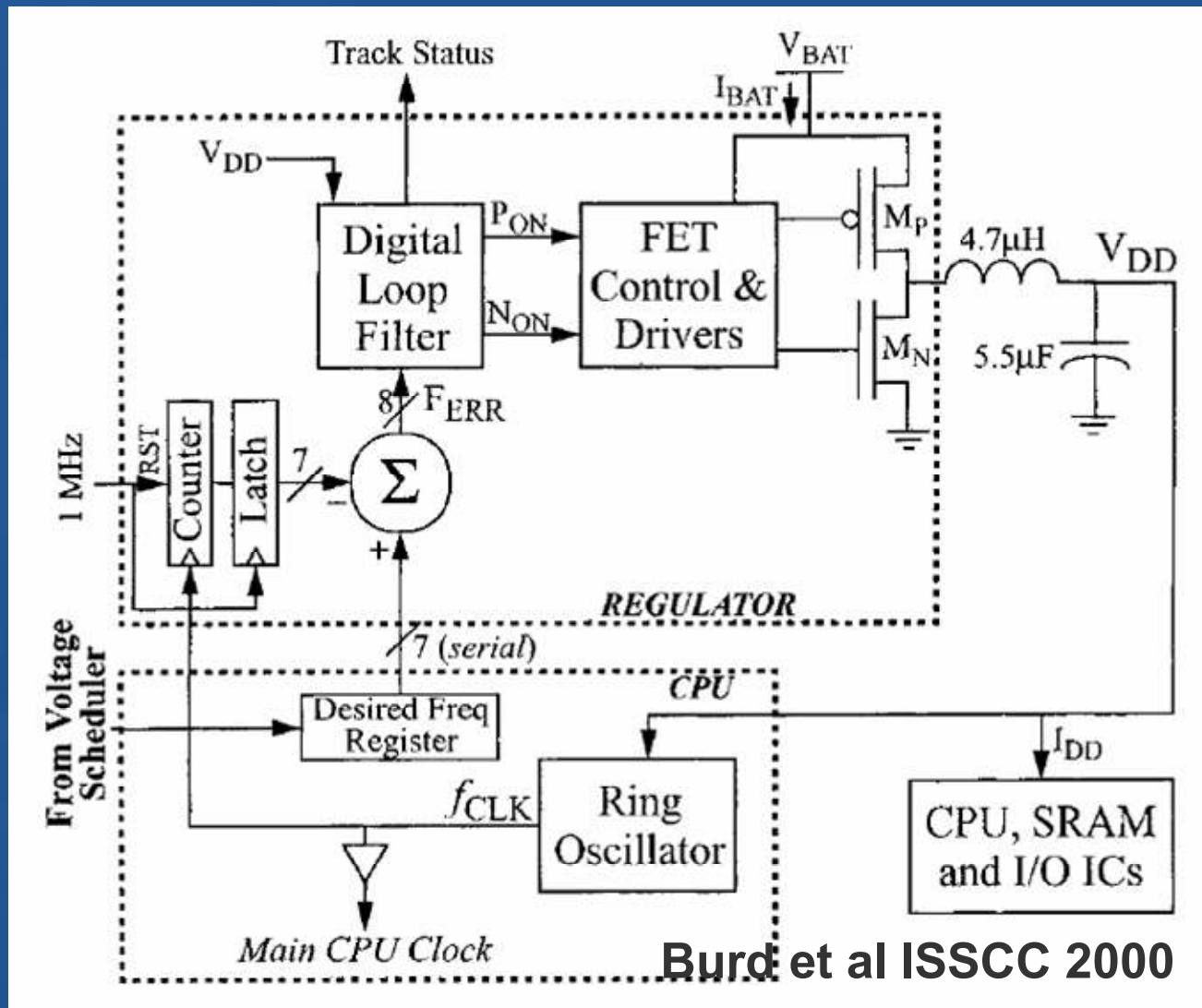


How Else to Save Energy?

- Running faster than needed wastes energy
 - Forces you to run higher on performance curve
- Why do you run faster than needed?
 - Need margins to account for variability
 - From application, environment, or technology

Variations cause waste

Dynamic Voltage Scaling



Dynamic Voltage Scaling

- Dynamic voltage scaling
 - Adjusts V_{dd} to the “right” value for desired performance
- Big problem is how to find the “right” V_{dd}
 - Need to know the relationship between V_{dd} and F
 - Need to have a circuit that matches the critical path
- How do you do this with variations?