CSE 560
Computer Systems Architecture

Technology

Survey: What is Moore's Law?

What does Moore's Law state?

A. The length of a transistor halves every 2 years.

B. The number of transistors on a chip will double every 2 years.

C. The frequency of a processor will double every 2 years.

D. The number of instructions a CPU can process will double every 2 years.

Survey: What is Moore's Law?

What does Moore's Law state?

A. The length of a transistor halves every 2 years.

B. The number of transistors on a chip will double every 2 years.

C. The frequency of a processor will double every 2 years.

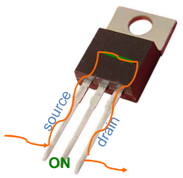
D. The number of instructions a CPU can process will double every 2 years.

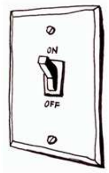
Technology Unit Overview

- **Technology basis**
  - Transistors
  - Transistor scaling (Moore's Law)
- **The metrics**
  - Cost
  - Transistor speed
  - Power
  - Reliability

How do the metrics change with transistor scaling?

How do these changes affect the job of a computer architect?

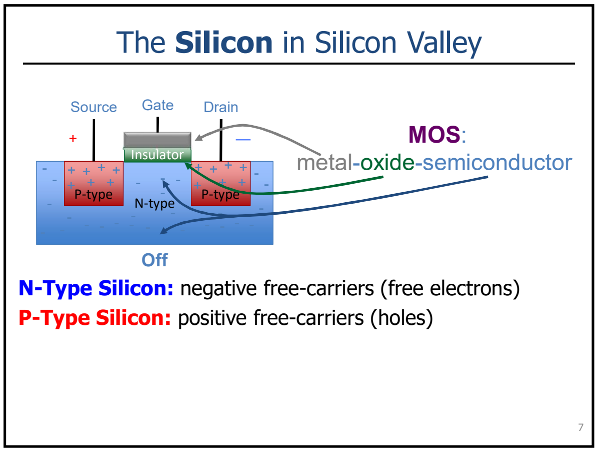




The Transistor

Technology Generations

1950-1959 Vacuum Tubes
1960-1968 Transistors
1969-1977 Integrated Circuit (multiple transistors on chip)
1978-1999 LSI & VLSI (10Ks & 100Ks transistors on chip)
2000-20xx VLSI (millions, now billions transistors on chip)



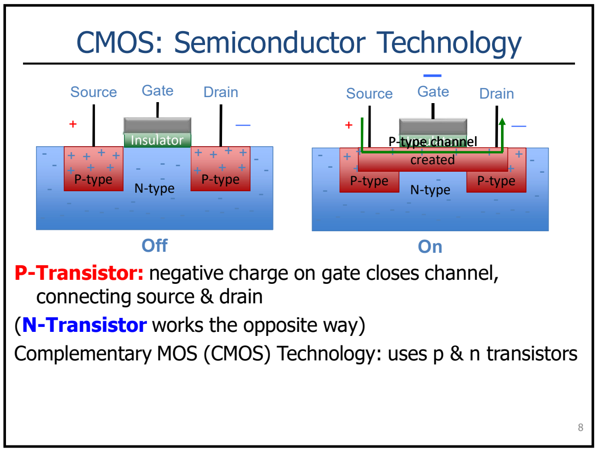
The **Silicon** in Silicon Valley

MOS: metal-oxide-semiconductor

N-Type Silicon: negative free-carriers (free electrons)

P-Type Silicon: positive free-carriers (holes)

7



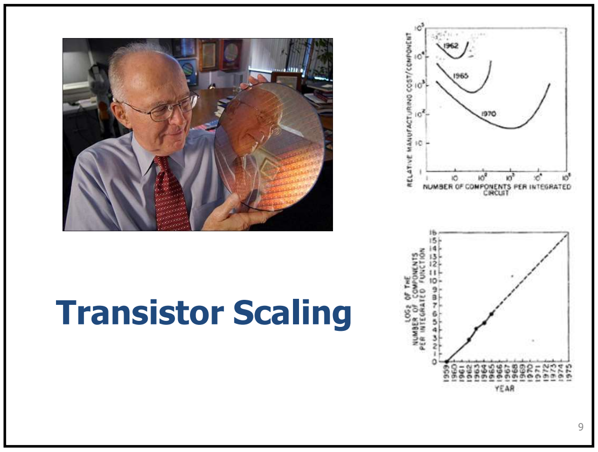
CMOS: Semiconductor Technology

P-Transistor: negative charge on gate closes channel, connecting source & drain

(**N-Transistor** works the opposite way)

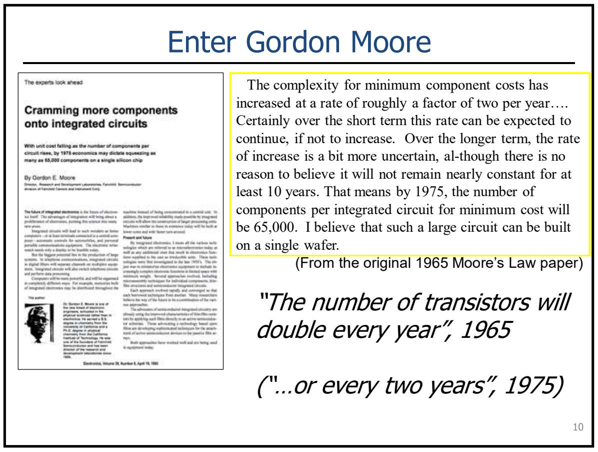
Complementary MOS (CMOS) Technology: uses p & n transistors

8



Transistor Scaling

9



Enter Gordon Moore

The experts look ahead

Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

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

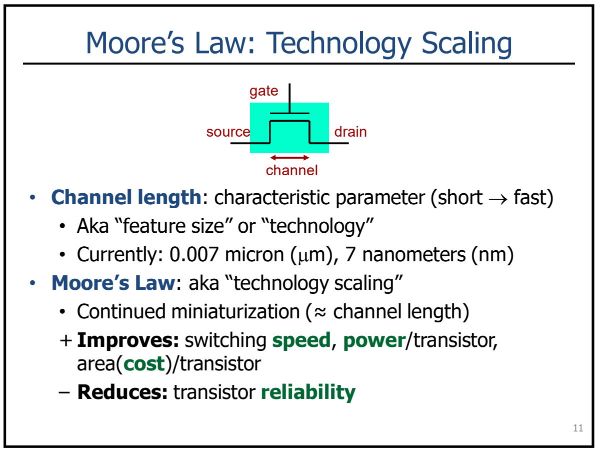
The complexity for minimum component costs has increased at a rate of roughly a factor of two per year…. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, al-though there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer.

(From the original 1965 Moore's Law paper)

"The number of transistors will double every year", 1965

("…or every two years", 1975)

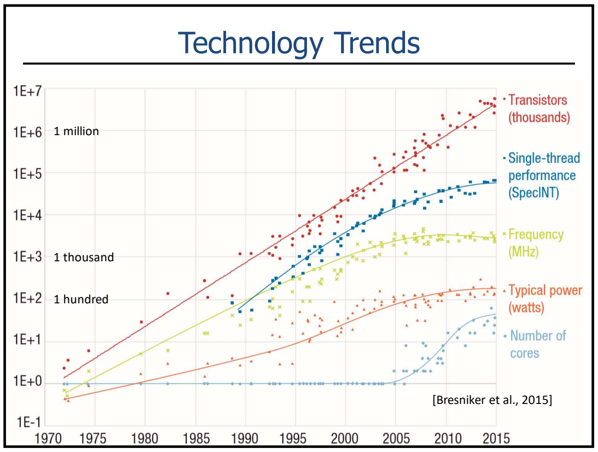
10



Moore's Law: Technology Scaling

- **Channel length**: characteristic parameter (short → fast)
  - Aka "feature size" or "technology"
  - Currently: 0.007 micron (μm), 7 nanometers (nm)
- **Moore's Law**: aka "technology scaling"
  - Continued miniaturization (≈ channel length)
  - + **Improves:** switching **speed**, **power**/transistor, area(**cost**)/transistor
  - – **Reduces:** transistor **reliability**

11



Technology Trends

[Bresniker et al., 2015]

12

Cost

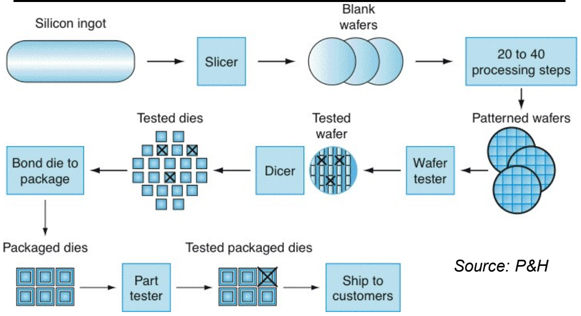
Cost

- Metric: **$**
- CPU = fraction of cost, so is profit (Intel’s, Dell’s)

	Desktop	Laptop	Netbook	Phone
$	$100–$300	$150-$350	$50–$100	$10–$20
% of total	10–30%	10–20%	20–30%	20-30%
Other costs	Memory, display, power supply/battery, storage, **software**			

- We are concerned about *chip cost*
  - **Unit cost:** costs to manufacture individual chips
  - **Startup cost:** cost to design chip, build the manufacturing facility

Unit Costs in Manufacturing Process

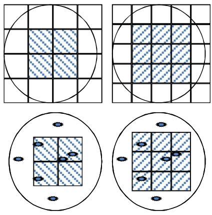


Source: P&H

Lots of unit testing, packaging, re-testing

Unit Cost: Integrated Circuit (IC)

- Cost / wafer is constant, f(wafer size, number of steps)
- Chip (die) cost related to **area**
  - Larger chips → fewer chips/water
    → fewer *working* ones
  - Chip cost ~ chip area$^{\alpha}$
    - α = 2 to 3
    - Why? random defects
- **Wafer yield:** % wafer that is chips
- **Die yield**: % chips that work
  - Yield is increasingly non-binary, fast vs. slow chips



Fixed Costs

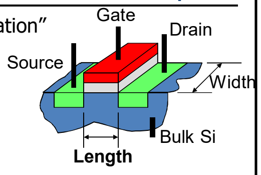
- **For new chip design**
  - Design & verification: ~$100M (500 person-years @ $200K per)
  - Amortized over “proliferations”, *e.g.*, Xeon/Celeron cache variants
- **For new (smaller) technology generation**
  - ~$3B for a new fab
  - Amortized over multiple designs
  - Amortized by “rent” from companies w/o their own fabs
- **Intel’s tick-tock** (smaller → better)

Transistor Speed

Moore's Speed Effect #1: Transistor Speed

Transistor length: "process generation"

45nm = transistor gate length



Shrink transistor length:

+ ↓ resistance of channel (shorter)
+ ↓ gate/source/drain capacitance

Result: switching speed ↑ linearly as gate length ↓

- much of past performance gains

But 2nd-order effects more complicated

– Process variation across chip increasing
  - Some transistors slow, some fast
  - Increasingly active research area: dealing with this

Diagrams © Krste Asanovic, MIT

Moore's Speed Effect #2: More Transistors

Linear shrink in each of 2 dimensions

- 180 nm, 130 nm, 90 nm, 65 nm, 45 nm, 32 nm, 22 nm, 14 nm, 10 nm, 7 nm …
- Each generation is a 1.414 linear shrink
- Results in 2x more transistors (1.414 x 1.414)

More transistors → increased performance

- **Job of computer architect:** figure out what to do with the ever-increasing # of transistors
- *Examples:* caches, branch predictors, exploiting parallelism at all levels

Moore's Speed Effect #3: Psychological

Moore's Curve: common interpretation of Moore's Law

- "CPU performance doubles every 18 months"
- Self fulfilling prophecy: 2x in 18 months is ~1% per week
  - Q: Would you add a feature that improved performance 20% if it would delay the chip 8 months?
- Processors under Moore's Curve (arrive too late) fail spectacularly
  - *E.g.*, Intel's Itanium, Sun's Millennium

Power & Energy

Power/Energy Increasingly Important

- **Battery life** for mobile devices
  - Laptops, phones, cameras
- **Tolerable temperature** for devices without active cooling
  - Power means temperature, active cooling means **cost**
  - No fan in a cell phone, no market for a hot cell phone
- **Electric bill** for compute/data centers
  - Pay for power twice: once in, once out (to cool)
- **Environmental concerns**
  - "Computers" account for growing fraction of energy consumption

Energy & Power

Energy: total amount of energy stored/used

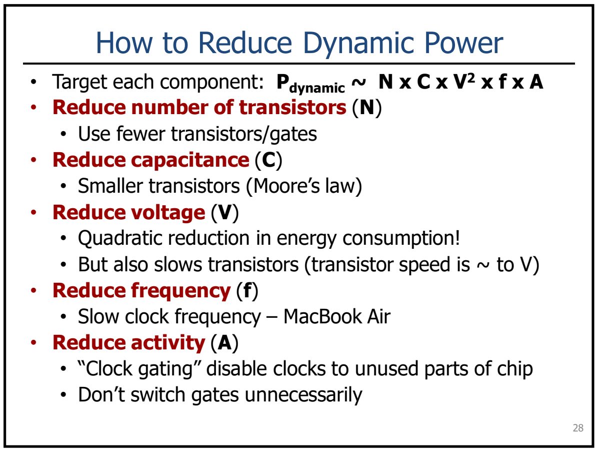
- Battery life, electric bill, environmental impact

Power: energy per unit time

- Related to "performance" (also a "per unit time" metric)
- Power impacts power supply, cooling needs (cost)
- Peak power vs. average power
  - E.g., camera power "spikes" when you take a picture

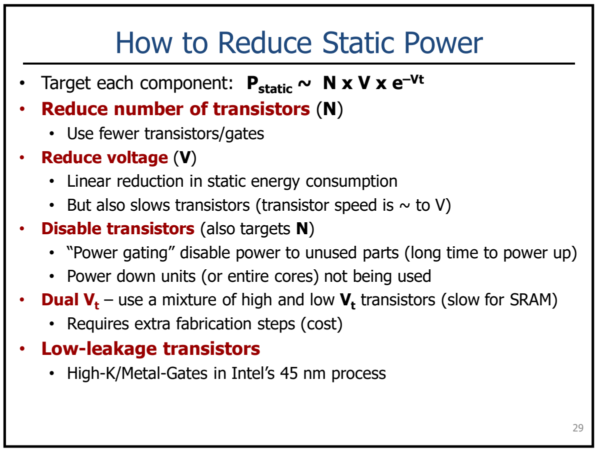
Two sources:

- **Dynamic power**: active switching of transistors
- **Static power**: transistors leak even when inactive



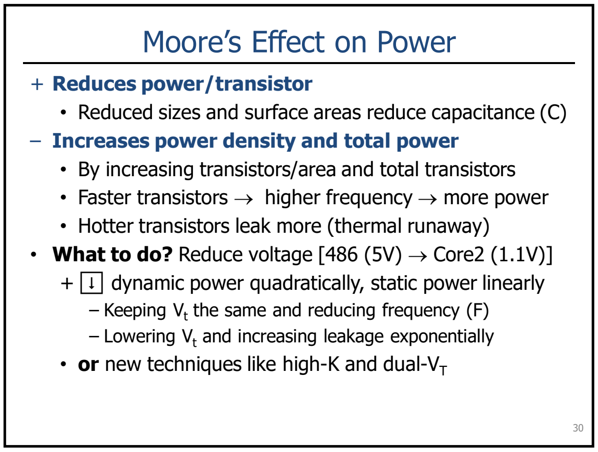
How to Reduce Dynamic Power

- Target each component: $P_{dynamic}$ ~ **N x C x V^2 x f x A**
- **Reduce number of transistors** (**N**)
  - Use fewer transistors/gates
- **Reduce capacitance** (**C**)
  - Smaller transistors (Moore's law)
- **Reduce voltage** (**V**)
  - Quadratic reduction in energy consumption!
  - But also slows transistors (transistor speed is ~ to V)
- **Reduce frequency** (**f**)
  - Slow clock frequency – MacBook Air
- **Reduce activity** (**A**)
  - "Clock gating" disable clocks to unused parts of chip
  - Don't switch gates unnecessarily



How to Reduce Static Power

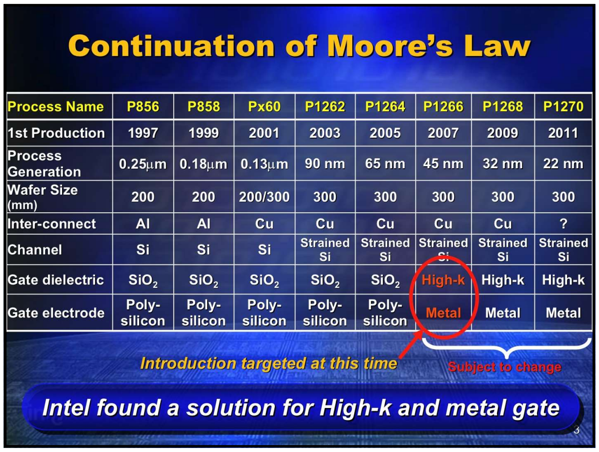
- Target each component: P_{static} ~ **N x V x e^{-Vt}**
- **Reduce number of transistors** (**N**)
  - Use fewer transistors/gates
- **Reduce voltage** (**V**)
  - Linear reduction in static energy consumption
  - But also slows transistors (transistor speed is ~ to V)
- **Disable transistors** (also targets **N**)
  - "Power gating" disable power to unused parts (long time to power up)
  - Power down units (or entire cores) not being used
- **Dual V_t** – use a mixture of high and low V_t transistors (slow for SRAM)
  - Requires extra fabrication steps (cost)
- **Low-leakage transistors**
  - High-K/Metal-Gates in Intel's 45 nm process



Moore's Effect on Power

+ **Reduces power/transistor**
  - Reduced sizes and surface areas reduce capacitance (C)

– **Increases power density and total power**
  - By increasing transistors/area and total transistors
  - Faster transistors → higher frequency → more power
  - Hotter transistors leak more (thermal runaway)
- **What to do?** Reduce voltage [486 (5V) → Core2 (1.1V)]
  - + $\boxed{\downarrow}$ dynamic power quadratically, static power linearly
    - – Keeping V_t the same and reducing frequency (F)
    - – Lowering V_t and increasing leakage exponentially
  - **or** new techniques like high-K and dual-V_T



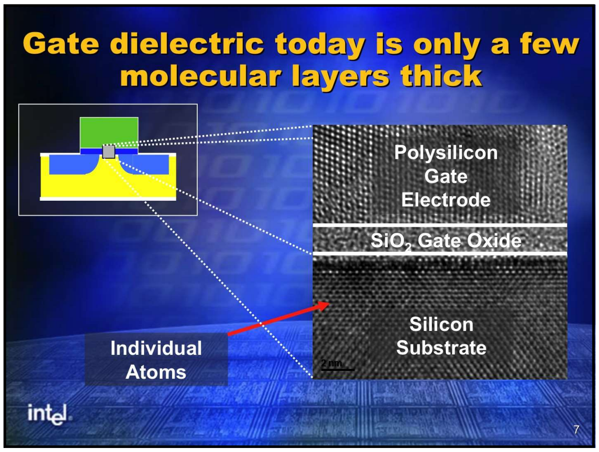
Continuation of Moore's Law

Process Name	P856	P858	Px60	P1262	P1264	P1266	P1268	P1270
1st Production	1997	1999	2001	2003	2005	2007	2009	2011
Process Generation	0.25µm	0.18µm	0.13µm	90 nm	65 nm	45 nm	32 nm	22 nm
Wafer Size (mm)	200	200	200/300	300	300	300	300	300
Inter-connect	Al	Al	Cu	Cu	Cu	Cu	Cu	?
Channel	Si	Si	Si	Strained Si	Strained Si	Strained Si	Strained Si	Strained Si
Gate dielectric	SiO_2	SiO_2	SiO_2	SiO_2	SiO_2	High-k	High-k	High-k
Gate electrode	Poly-silicon	Poly-silicon	Poly-silicon	Poly-silicon	Poly-silicon	Metal	Metal	Metal

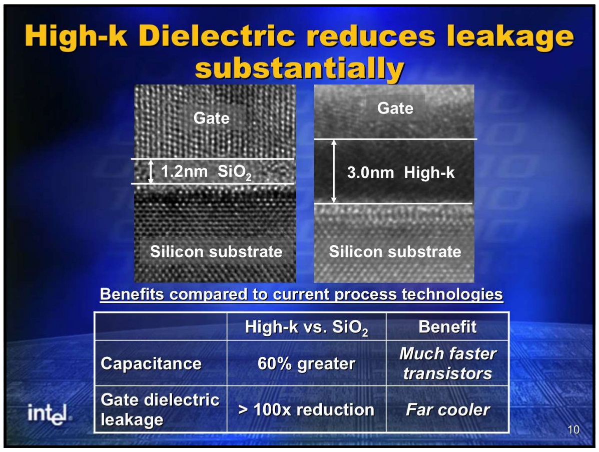
Introduction targeted at this time

Subject to change

Intel found a solution for High-k and metal gate



Gate dielectric today is only a few molecular layers thick

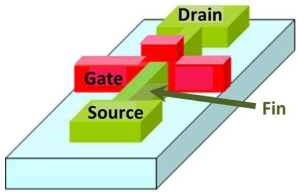


High-k Dielectric reduces leakage substantially

Benefits compared to current process technologies

	High-k vs. SiO_2	Benefit
Capacitance	60% greater	*Much faster transistors*
Gate dielectric leakage	> 100x reduction	*Far cooler*

FinFET



By Irene Ringworm at the English language Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=3833512

36

Reliability

40

Technology Basis for Reliability

- **Transient faults**
  - A bit "flips" randomly, **temporarily**
  - Cosmic rays etc. (more common at higher altitudes!)
- **Permanent (hard) faults**
  - A gate or memory cell wears out, **breaks and stays broken**
  - Temperature & electromigration slowly deform components
- Solution for both: **redundancy** to detect and tolerate

41

Moore's Bad Effect on Reliability

– **Transient faults:**
  - Small (low charge) transistors are more easily flipped
  - Even low-energy particles can flip a bit now

– **Permanent faults:**
  - Small transistors and wires deform and break more quickly
  - Higher temperatures accelerate the process

Wasn't a problem until ~10 years ago (except in satellites)
  - Memory (DRAM): these dense, small devices hit first
  - Then on-chip memory (SRAM)
  - Logic is starting to have problems…

42

Moore's Good Effect on Reliability

- Scaling makes devices less reliable

+ Scaling increases device density to enable **redundancy**

- Examples
  - Error correcting code for memory (DRAM), $s (SRAM)
  - Core-level redundancy: paired-execution, hot-spare, etc.
  - Intel's Core i7 (Nehalem) uses 8 transistor SRAM cells
    - Versus the standard 6 transistor cells
- Big open questions
  - Can we protect logic efficiently? (w/o 2-3x overhead)
  - Can architectural techniques help hardware reliability?
  - Can software techniques help?

43

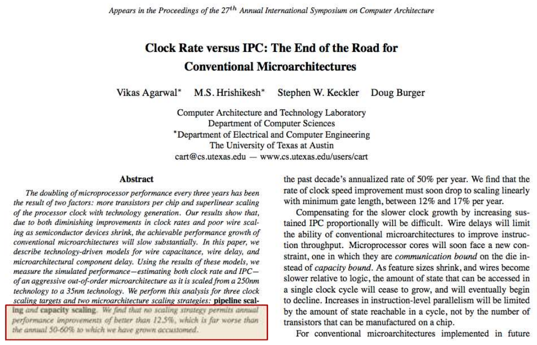
Summary

44

Moore's Law in the Future

- Won't last forever, approaching physical limits
  - But betting against it has proved foolish in the past
  - Likely to "slow" rather than stop abruptly
- Transistor count will likely continue to scale
  - "Die stacking" is on the cusp of becoming main stream
  - Uses the third dimension to increase transistor count
- But transistor performance scaling?
  - Running into physical limits
  - Example: gate oxide is less than 10 silicon atoms thick!
    - – Can't decrease it much further
  - Power is becoming a limiting factor

Moore's Law & Chicken Little



Appears in the Proceedings of the 27th Annual International Symposium on Computer Architecture

Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures

Vikas Agarwal* M.S. Hrishikesh* Stephen W. Keckler Doug Burger

Computer Architecture and Technology Laboratory
Department of Computer Sciences
*Department of Electrical and Computer Engineering
The University of Texas at Austin
cart@cs.utexas.edu — www.cs.utexas.edu/users/cart

Abstract

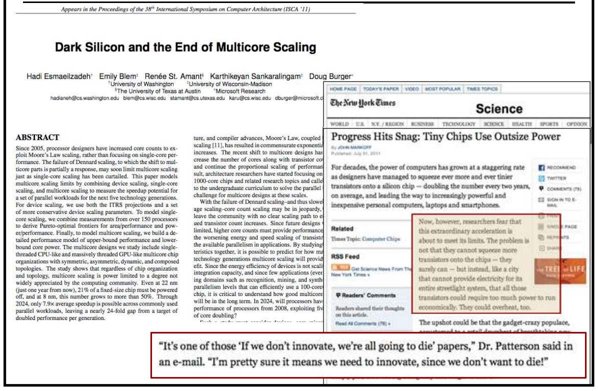
The doubling of microprocessor performance every three years has been the result of two factors: more transistors per chip and superlinear scaling of the processor clock with technology generation. Our results show that, due to both diminishing improvements in clock rates and poor wire scaling as semiconductor devices shrink, the achievable performance growth of conventional microarchitectures will slow substantially. In this paper, we describe technology-driven models for wire capacitance, wire delay, and microarchitectural component delay. Using the results of these models, we measure the simulated performance—estimating both clock rate and IPC—of an aggressive out-of-order microarchitecture as it is scaled from a 250nm technology to a 35nm technology. We perform this analysis for three clock scaling targets and two microarchitecture scaling strategies: **pipeline scaling** *and* **capacity scaling**. *We find that no scaling strategy permits annual performance improvements of better than 12.5%, which is far worse than the annual 50-60% to which we have grown accustomed.*

the past decade's annualized rate of 50% per year. We find that the rate of clock speed improvement must soon drop to scaling linearly with minimum gate length, between 12% and 17% per year.

Compensating for the slower clock growth by increasing sustained IPC proportionally will be difficult. Wire delays will limit the ability of conventional microarchitectures to improve instruction throughput. Microprocessor cores will soon face a new constraint, one in which they are *communication bound* on the die instead of *capacity bound*. As feature sizes shrink, and wires become slower relative to logic, the amount of state that can be accessed in a single clock cycle will cease to grow, and will eventually begin to decline. Increases in instruction-level parallelism will be limited by the amount of state reachable in a cycle, not by the number of transistors that can be manufactured on a chip.

For conventional microarchitectures implemented in future

Moore's Law & Chicken Little, Part 2



Appears in the Proceedings of the 38th International Symposium on Computer Architecture (ISCA '11)

Dark Silicon and the End of Multicore Scaling

Hadi Esmaeilzadeh† Emily Blem‡ Renée St. Amant§ Karthikeyan Sankaralingam‡ Doug Burger°
†University of Washington ‡University of Wisconsin-Madison
§The University of Texas at Austin °Microsoft Research
hadianeh@cs.washington.edu blem@cs.wisc.edu stamant@cs.utexas.edu karu@cs.wisc.edu dburger@microsoft.c

ABSTRACT

Since 2005, processor designers have increased core counts to exploit Moore's Law scaling, rather than focusing on single-core performance. The failure of Dennard scaling, to which the shift to multicore parts is partially a response, may soon limit multicore scaling just as single-core scaling has been curtailed. This paper models multicore scaling limits by combining device scaling, single-core scaling, and multicore scaling to measure the speedup potential for a set of parallel workloads for the next five technology generations. For device scaling, we use both the ITRS projections and a set of more conservative device scaling parameters. To model single-core scaling, we combine measurements from over 150 processors to derive Pareto-optimal frontiers for area/performance and power/performance. Finally, to model multicore scaling, we build a detailed performance model of upper-bound performance and lower-bound core power. The multicore designs we study include single-threaded CPU-like and massively threaded GPU-like multicore chip organizations with symmetric, asymmetric, dynamic, and composed topologies. The study shows that regardless of chip organization and topology, multicore scaling is power limited to a degree not widely appreciated by the computing community. Even at 22 nm (just one year from now), 21% of a fixed-size chip must be powered off, and at 8 nm, this number grows to more than 50%. Through 2024, only 7.9× average speedup is possible across commonly used parallel workloads, leaving a nearly 24-fold gap from a target of doubled performance per generation.

ture, and compiler advances, Moore's Law, coupled
scaling [11], has resulted in commensurate exponentia
increases. The recent shift to multicore designs has
crease the number of cores along with transistor co
and continue the proportional scaling of performan
sult, architecture researchers have started focusing on
1000-core chips and related research topics and calle
to the undergraduate curriculum to solve the parallel
challenge for multicore designs at these scales.

With the failure of Dennard scaling–and thus slowe
age scaling–core count scaling may be in jeopardy,
leave the community with no clear scaling path to e
ued transistor count increases. Since future designs
limited, higher core counts must provide performance
the worsening energy and speed scaling of transisto
the available parallelism in applications. By studying
teristics together, it is possible to predict for how ma
technology generations multicore scaling will provid
efit. Since the energy efficiency of devices is not scali
integration capacity, and since few applications (ever
ing domains such as recognition, mining, and synth
parallelism levels that can efficiently use a 100-core
chip, it is critical to understand how good multicore
will be in the long term. In 2024, will processors hav
performance of processors from 2008, exploiting fiv
of core doubling?

Progress Hits Snag: Tiny Chips Use Outsize Power

By JOHN MARKOFF
Published: July 31, 2011

For decades, the power of computers has grown at a staggering rate as designers have managed to squeeze ever more and ever tinier transistors onto a silicon chip — doubling the number every two years, on average, and leading the way to increasingly powerful and inexpensive personal computers, laptops and smartphones.

Now, however, researchers fear that this extraordinary acceleration is about to meet its limits. The problem is not that they cannot squeeze more transistors onto the chips — they surely can — but instead, like a city that cannot provide electricity for its entire streetlight system, that all those transistors could require too much power to run economically. They could overheat, too.

Related
Times Topic: Computer Chips

RSS Feed
Get Science News From The New York Times »

Readers' Comments
Readers shared their thoughts on this article.
Read All Comments (78) »

The upshot could be that the gadget-crazy populace,

"It's one of those 'If we don't innovate, we're all going to die' papers," Dr. Patterson said in an e-mail. "I'm pretty sure it means we need to innovate, since we don't want to die!"

Summary of Device Scaling

+ Reduces unit cost
  – But increases startup cost

+ Increases performance
  - Reduces transistor/wire delay
  - Gives us more transistors with which to increase performance

+ Reduces local power consumption
  – Quickly undone by increased integration, frequency
  – Aggravates power-density and temperature problems

– Aggravates reliability problem
  + But gives us the transistors to solve it via redundancy