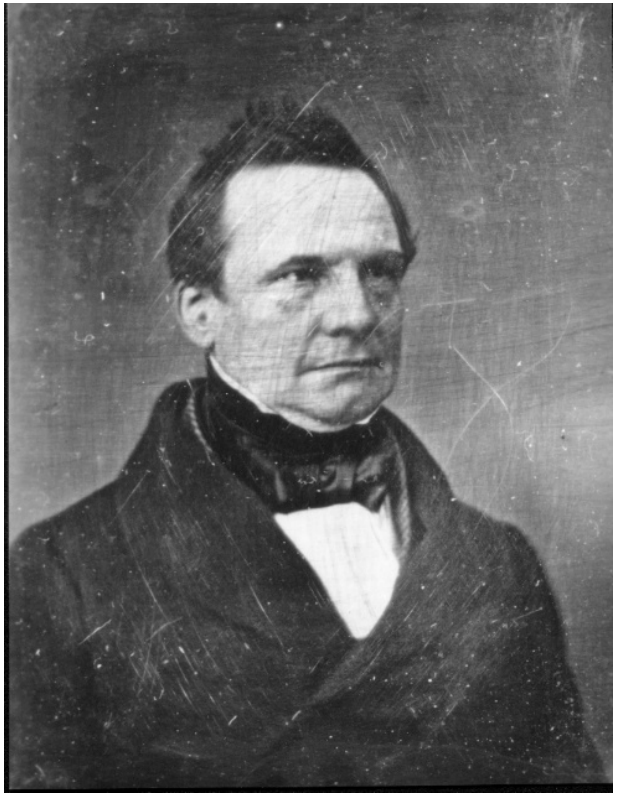
An Incomplete History of Computation



Abacus: ca. 2700 B.C., Sumeria

Charles Babbage 1791-1871

Lucasian Professor of Mathematics,
Cambridge University, 1827-1839
First computer designer



Ada Lovelace 1815-1852

First computer programmer



Adapted from Arvind and Asanovic's MIT course 6.823, Lecture 1

1801: Jacquard Loom



Charles Babbage

- *Difference Engine* 1823

- *Analytic Engine* 1833
  - The forerunner of modern digital computer!

Application

- Mathematical Tables – Astronomy
- Nautical Tables – Navy

Background

- Some efforts at mechanical calculators in the past.

Technology

- mechanical - gears, Jacquard's loom, simple calculators

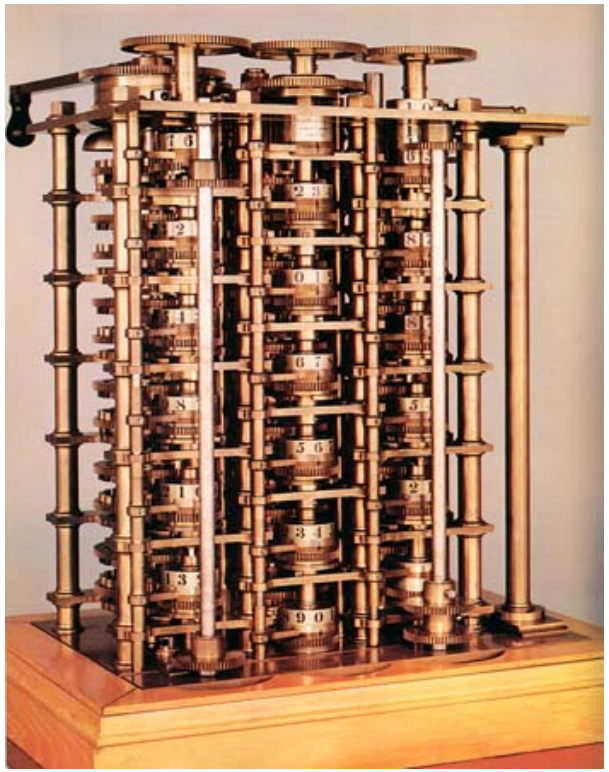
Difference Engine

1823
- Babbage's paper is published

1834
- The paper is read by Scheutz & his son in Sweden

1842
- Babbage gives up the idea of building it;he is on to the Analytic Engine!

1855
- Scheutz displays his machine at the Paris World Fare
- Can compute any 6th degree polynomial
- *Speed:* 33 to 44 32-digit numbers per minute!



Now the machine is at the Smithsonian

Analytic Engine

The first conception of a general purpose computer

1. The *store* in which all variables to be operated upon, as well as all those quantities which have arisen from the results of the operations are placed.
2. The *mill* into which the quantities about to be operated upon are always brought.

An operation in the *mill* required feeding two punched cards and producing a new punched card for the *store.*

An operation to alter the sequence (i.e., a branch) was also provided!

Harvard Mark I

- Built in 1944 in IBM Endicott laboratories
  - Howard Aiken – Professor of Physics at Harvard
  - Essentially electromechanical but had some electro-magnetically controlled relays and gears
  - Weighed *5 tons* and had *750,000* components
  - A synchronizing clock that beat every *0.015* seconds
  - Data entry: mechanical switches
  - Instructions: punched paper tape; create physical "loops" w/ tape!

Performance:

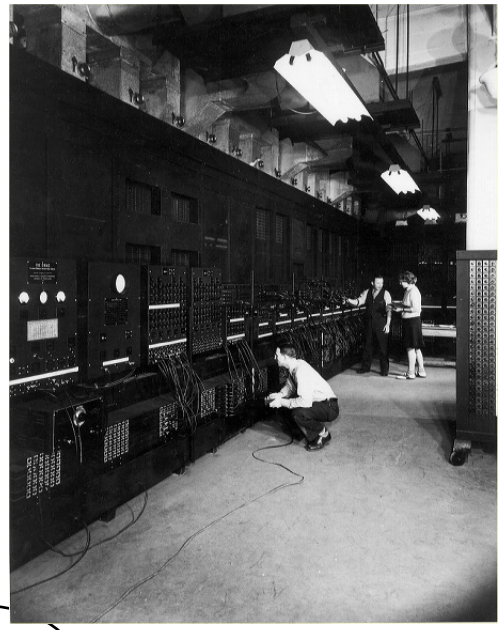
0.3 seconds for addition
6 seconds for multiplication
1 minute for a sine calculation

Broke down once a week!

See it in the Harvard Science Building.

Electronic Numerical Integrator and Computer (ENIAC)

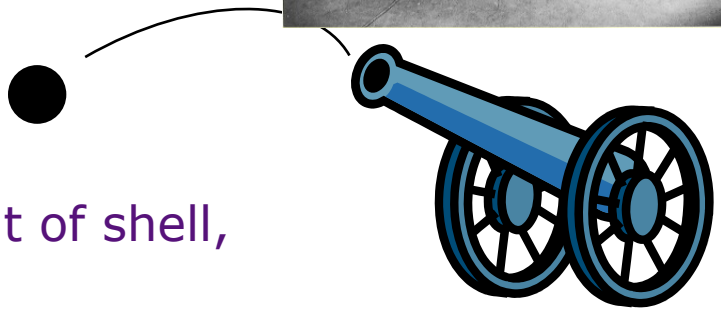
- Inspired by Atanasoff and Berry, Eckert and Mauchly designed and built ENIAC (1943-45) at the University of Pennsylvania
- The first, completely electronic, operational, general-purpose analytical calculator!
  - 30 tons, 72 square meters, 200KW
- Performance
  - Read in 120 cards per minute
  - Addition took 200 μs, Division 6 ms
  - 1000 times faster than Mark I
- Not very reliable!



WW-2 Effort

Application: Ballistic calculations

angle = f (location, tail wind, cross wind,
air density, temperature, weight of shell,
propellant charge, ...)



Electronic Discrete Variable Automatic Computer (EDVAC)

- ENIAC's programming system was external
  - Sequences of instructions were executed independently of the results of the calculation
  - Human intervention required to take instructions "out of order"
- Eckert, Mauchly, John von Neumann and others designed EDVAC (1944) to solve this problem
  - Solution was the *stored program computer*

    ⇒ "*program can be manipulated as data*"
- *First Draft of a report on EDVAC* was published in 1945, but just had von Neumann's signature!
  - In 1973 the court of Minneapolis attributed the honor of *inventing the computer* to John Atanasoff

Stored Program Computer

Program = A sequence of instructions

How to control instruction sequencing?

manual control — calculators

automatic control

- *external (paper tape)* — Harvard Mark I , 1944; Zuse's Z1, WW2
- *internal*
  - *plug board* — ENIAC 1946
  - *read-only memory* — ENIAC 1948
  - *read-write memory* — EDVAC 1947 *(concept)*

– The same storage can be used to store program and data

EDSAC	1950	Maurice Wilkes

first stored program computer

And then there was IBM 701

IBM 701 -- 30 machines were sold in 1953-54

IBM 650 -- a cheaper, drum based machine,
more than 120 were sold in 1954
and there were orders for 750 more!
- eventually sold about 2000 of them

Users stopped building their own machines.

Why was IBM late getting into computer technology?

IBM was making too much money!
Even without computers, IBM revenues were doubling every 4 to 5 years in 40's and 50's.

Dominant Problem: *Reliability*

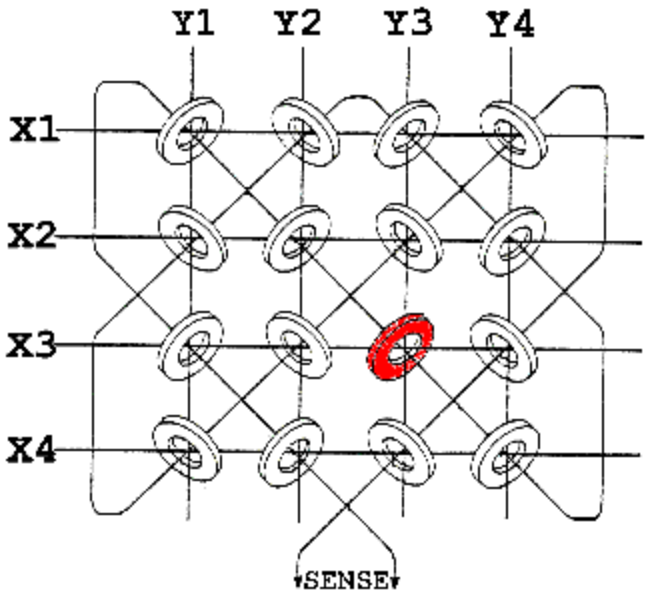
Mean time between failures (MTBF)

MIT's Whirlwind with an MTBF of 20 min. was perhaps the most reliable machine !

Reasons for unreliability:

1. Vacuum Tubes

2. Storage medium
   acoustic delay lines
   mercury delay lines
   Williams tubes
   Selections



Magnetic Core Memory	J. Forrester 1951

-first cheap, reliable memory (~ 1 MHz)
-also called "core" (e.g., "core dump")
-non volatile!
-destructive read cycle
-array of ferrite toroids (or "cores")
-dominant memory technology until 70's (→ ICs)

Adapted from Arvind and Asanovic's MIT course 6.823, Lecture 1

Computers in mid 50's

- Hardware was expensive
- Stores were small (1000 words)
  ⇒ No resident system-software!
- Memory access time was 10 to 50 times slower than the processor cycle
  ⇒ Instruction execution time was totally dominated by the *memory reference time*.
- The *ability to design complex control circuits* to execute an instruction was the central design concern as opposed to *the speed* of decoding or an ALU operation
- Programmer's view of the machine was inseparable from the actual hardware implementation

Into the 60's…:

Compatibility Problem at IBM

By early 60's, *IBM had 4 incompatible lines of computers!*

701 → 7094
650 → 7074
702 → 7080
1401 → 7010

Each system had its own

- Instruction set
- I/O system and Secondary Storage: magnetic tapes, drums and disks
- assemblers, compilers, libraries,...
- market niche
  business, scientific, real time, ...

⇒ *IBM 360*

IBM 360 : Design Premises

Amdahl, Blaauw and Brooks, 1964

http://www.research.ibm.com/journal/rd/441/amdahl.pdf

- Upward and downward, machine-language compatibility across a family of machines
- General purpose machine organization, general I/O interfaces, storage > 32K
- Easier to use (answers-per-month vs. bits-per-second)
- Machine must be capable of *supervising itself* without manual intervention →OS/360 (simple OS's in IBM 700/7000)
- Built-in *hardware fault checking* and locating aids to reduce down time
- Simple to assemble systems with redundant I/O devices, memories etc. for *fault tolerance*

… the use of the "ISA" as a compatibility layer

$5 billion project (1964 dollars)

The Amdahl .. from Amdahl's Law.
The Brooks .. from The Mythical Man-Month.


IBM
SYSTEM 360

IBM 360: *A General-Purpose Register (GPR) Machine*

- Processor State
  - 16 General-Purpose 32-bit Registers
    - *may be used as index and base register*
    - *Register 0 has some special properties*
  - 4 Floating Point 64-bit Registers
  - A Program Status Word (PSW)
    - *PC, Condition codes, Control flags*
- A 32-bit machine with 24-bit addresses
  - No instruction contains a 24-bit address !
- Data Formats
  - 8-bit bytes, 16-bit half-words, 32-bit words, 64-bit double-words

IBM 360: Implementation

	Model 30	*Model 70*
Storage	8K - 64 KB	256K - 512 KB
Datapath	8-bit	64-bit
Circuit Delay	30 nsec/level	5 nsec/level
Local Store	Main Store	Transistor Registers
Control Store	Read only 1μsec	Conventional circuits

IBM 360 instruction set architecture completely hid the underlying technological differences between various models.

With minor modifications it survives till today

High performance competitor to IBM/S360

CDC 6600 *Seymour Cray, 1964*

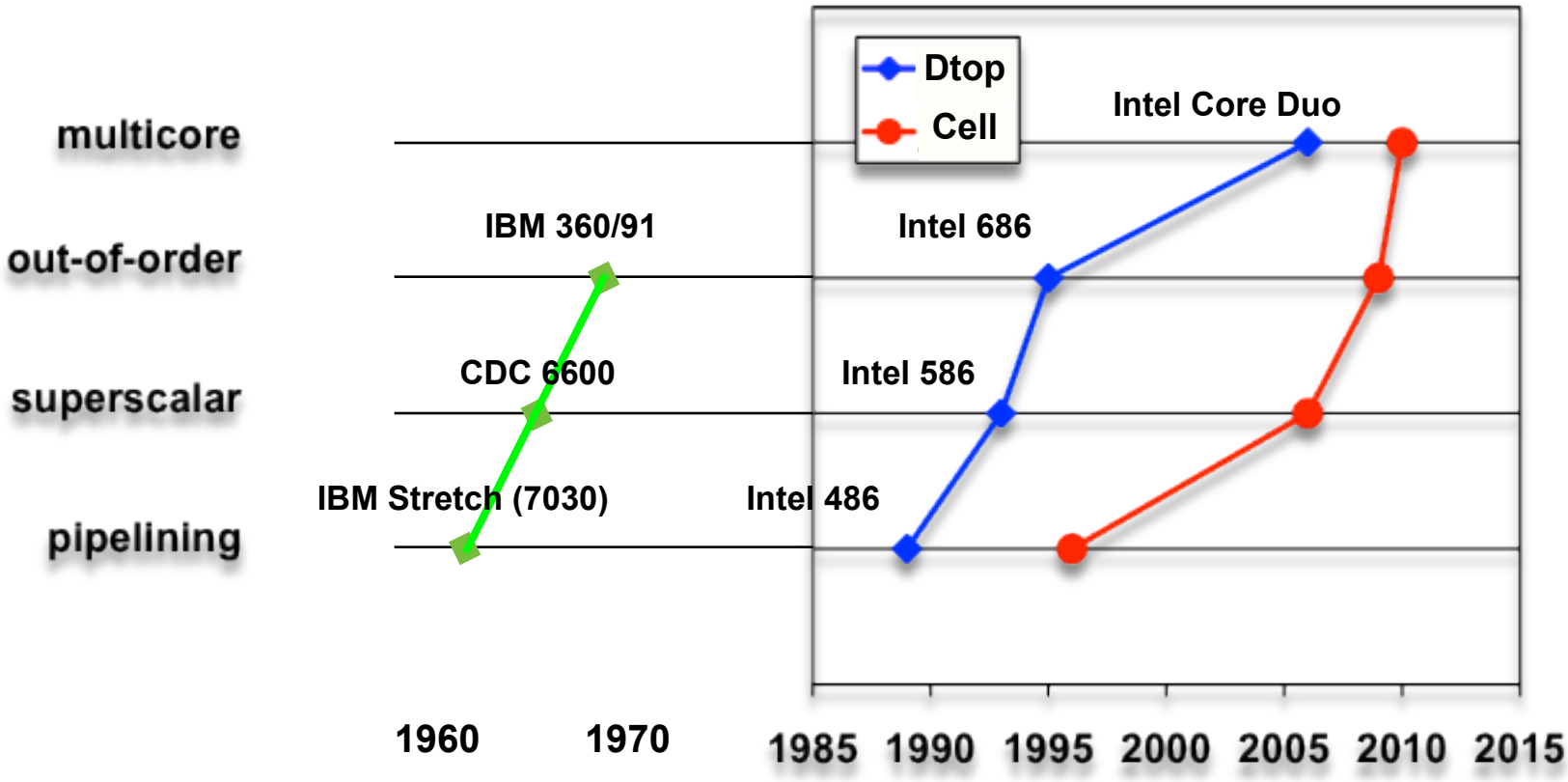
a scientific supercomputer

- A fast pipelined machine with 60-bit words
- Ten functional units
  - Floating Point: adder, multiplier, divider
  - Integer: adder, multiplier
  - ...
- Hardwired control (no microcoding)
- Dynamic scheduling of instructions using a scoreboard (see Appendix A)
- Ten Peripheral Processors for Input/Output
  - a fast time-shared 12-bit integer ALU
- Very fast clock
- Novel freon-based technology for cooling
- Seymour Cray is the original computer architecture rockstar → Cray series of supercomputers

Seymour's 5P's: Packaging, Plumbing (bits and heat flow), Parallelism, Programming, and understanding Programs

Cell Phone ("Mobile Application") Processors Recapitulate Desktop Processors Recapitulate Mainframe Processors

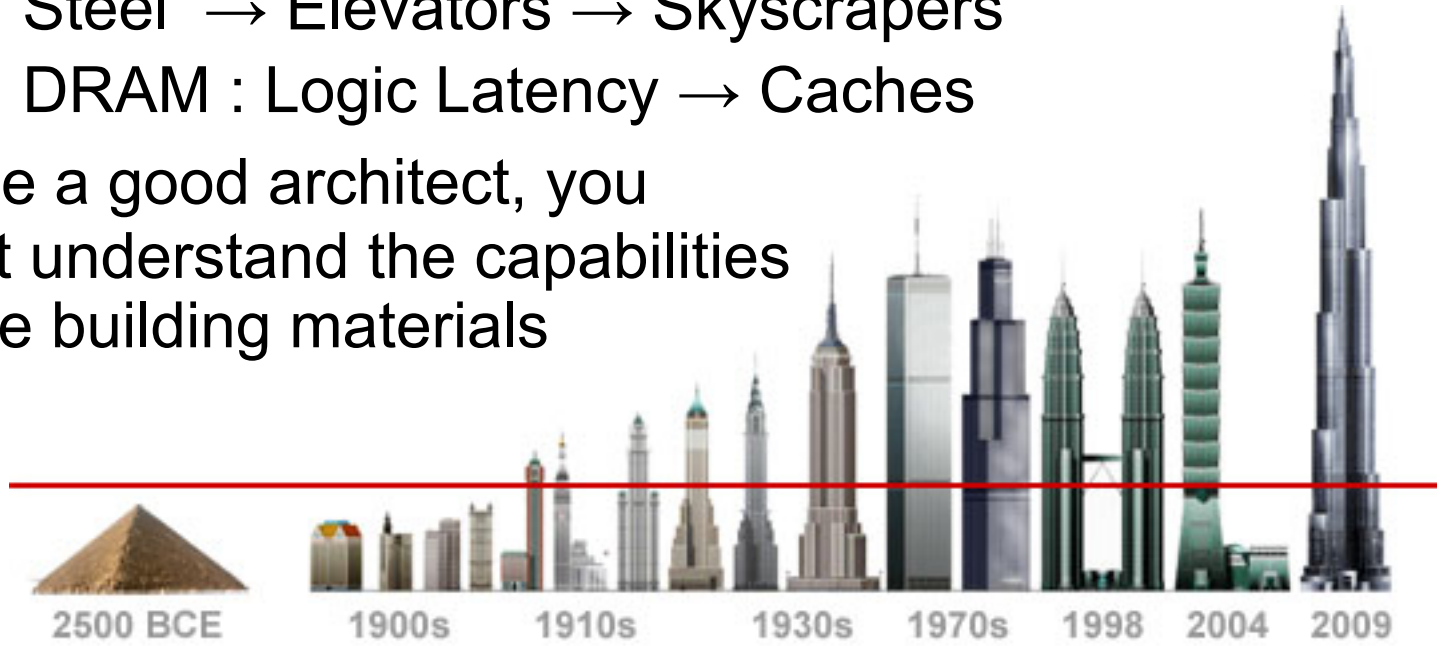
- The evolution of cell phone processors mirrors that of microprocessors, which mirrors that of mainframes



Understanding The Architect's Raw Materials

Modern Day Architecture is a Consequence of Modern Construction Properties

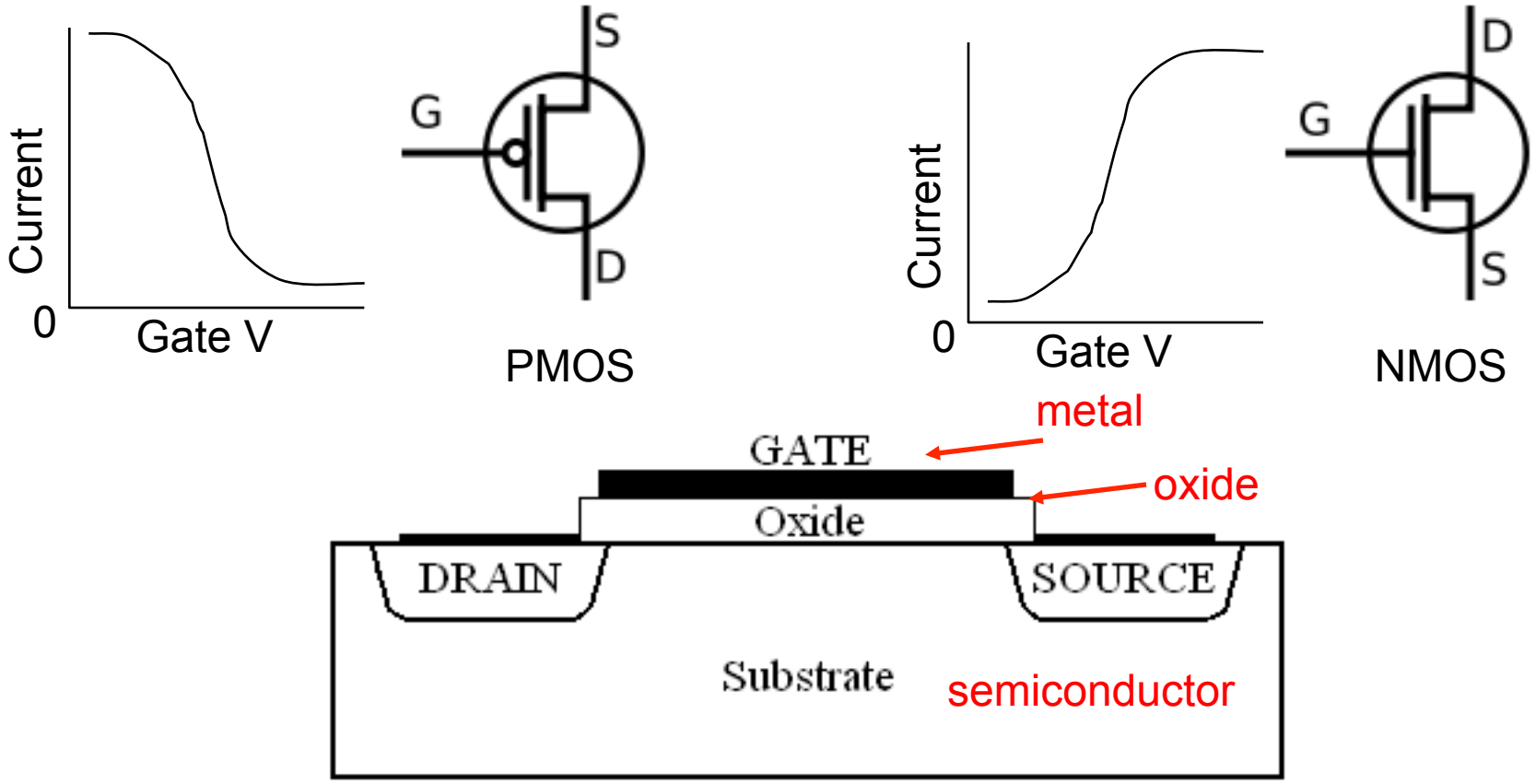
- E.g., Stone → Pyramids
- E.g., Steel → Elevators → Skyscrapers
- E.g., DRAM : Logic Latency → Caches
- To be a good architect, you
- must understand the capabilities
- of the building materials



(buildings: http://www.newraleigh.com/articles/archive/edifice-rex/)

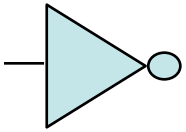
CMOS Logic: One Transistor

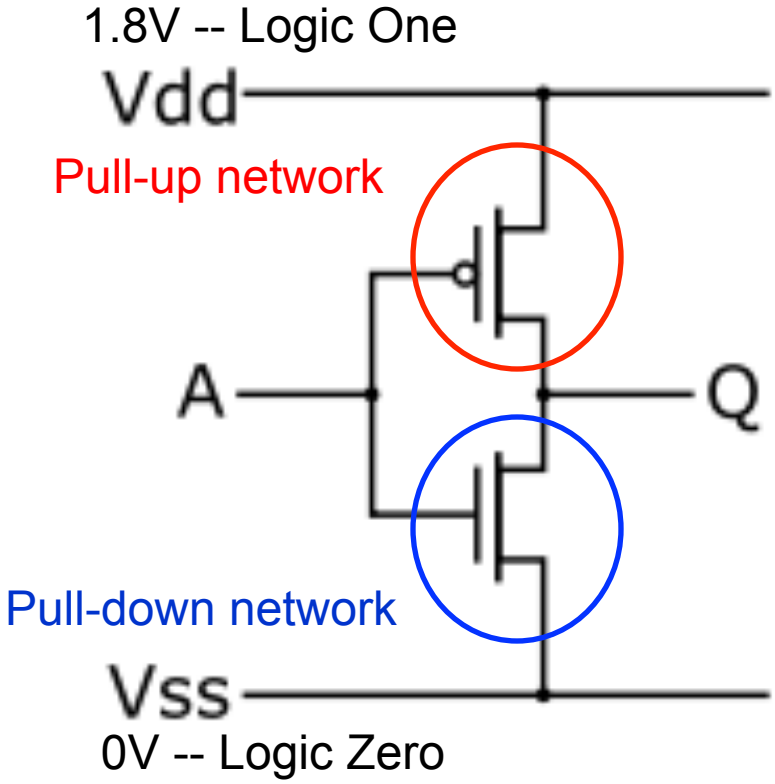
- Complimentary Metal Oxide Semiconductor
  - Logic family (an abstraction!)

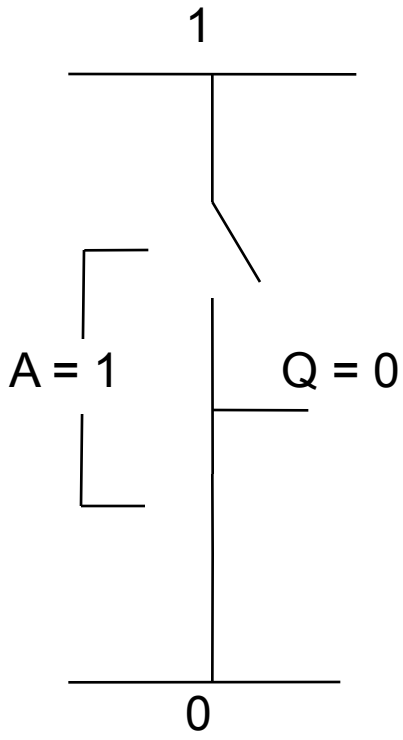


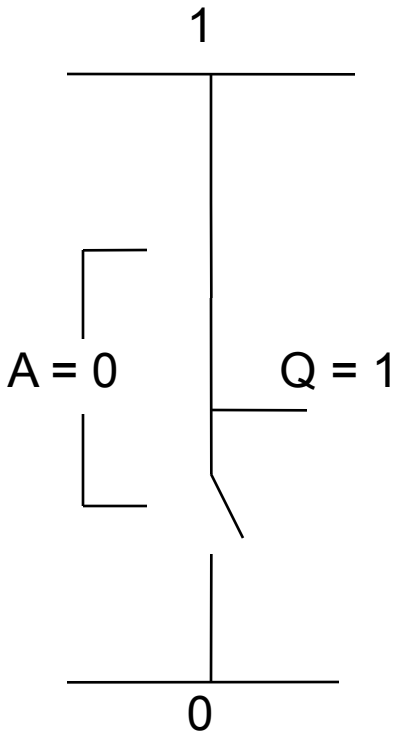
CMOS Logic: One Gate

- Complimentary MOS

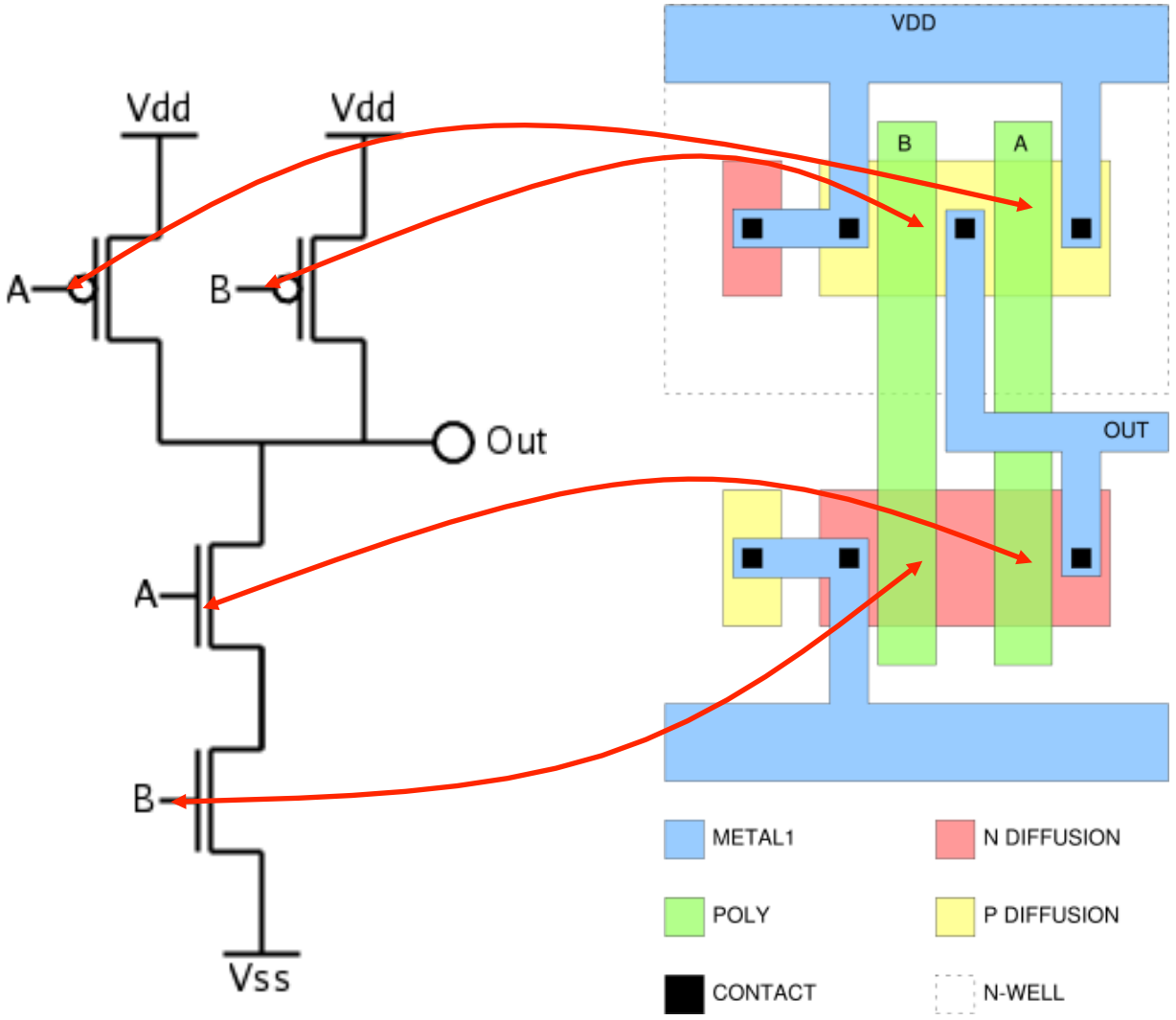






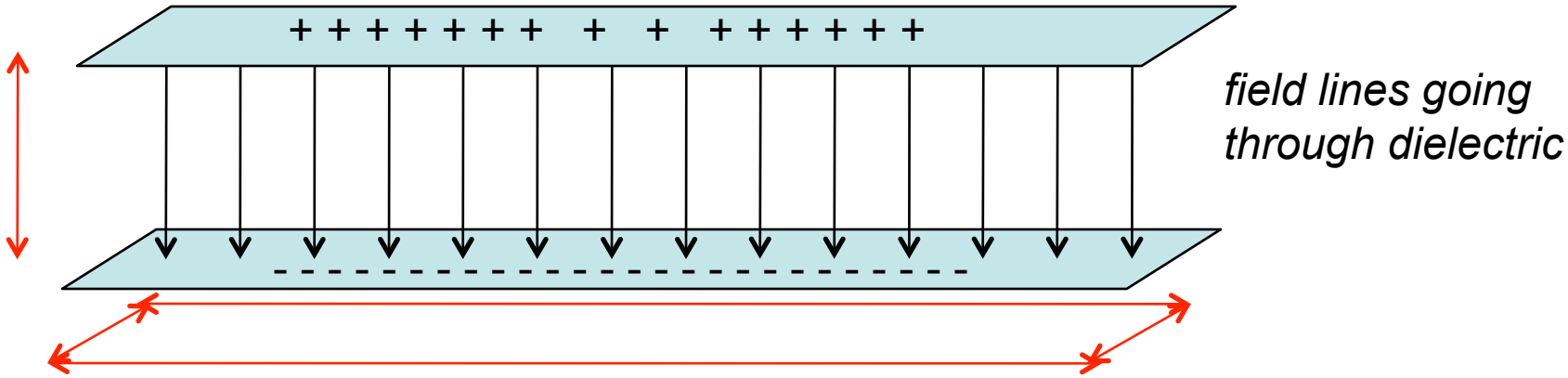


CMOS Gate: Layout



Capacitance: First Order Calc

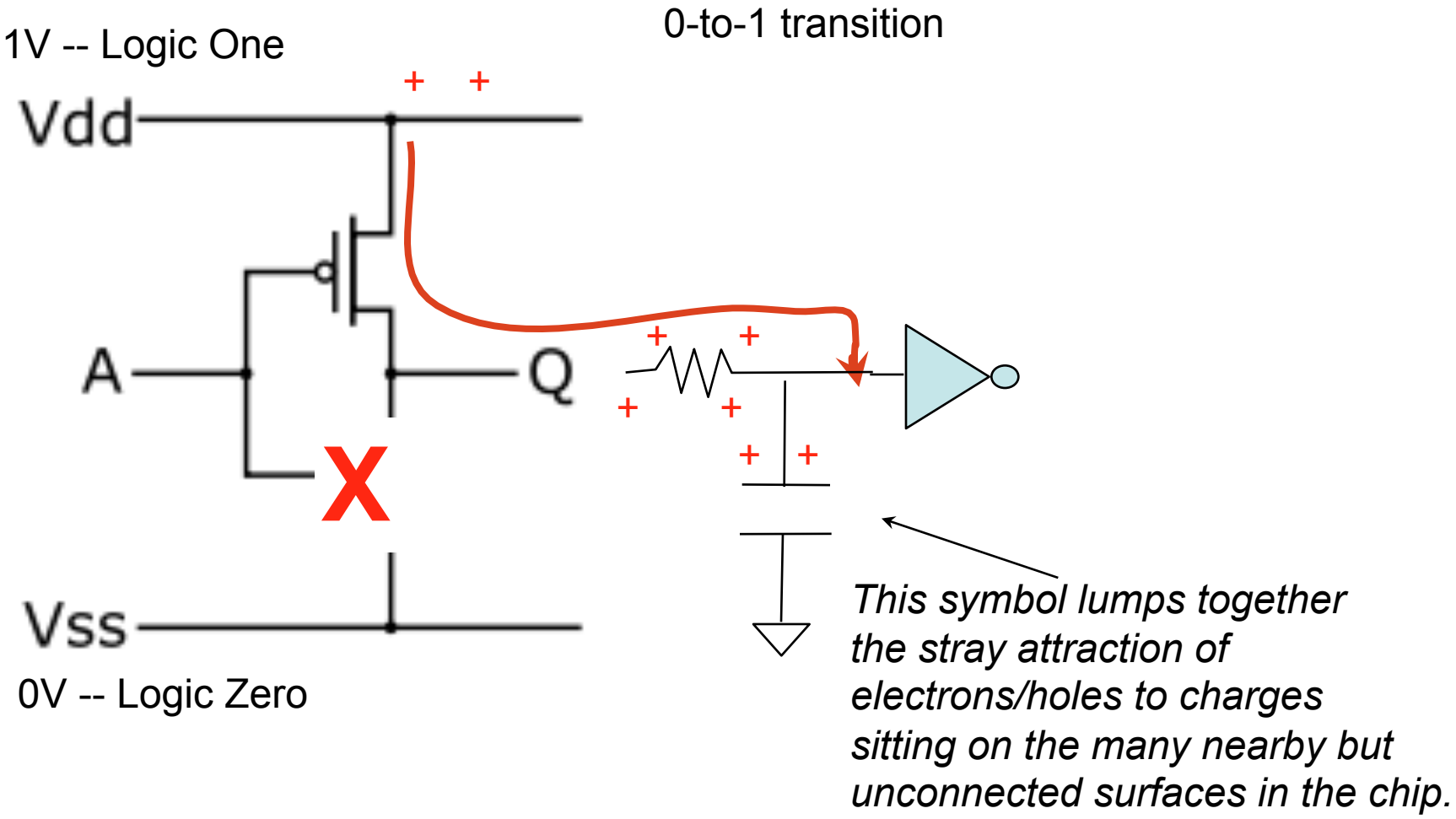
- = ~ $\frac{\text{Surface\_Area} * \text{Dielectric Constant}}{\text{Separation}}$



- If we scale width and separation by the same amount, capacitance is constant.
- If we scale all three dimensions by a factor of S, capacitance drops by S.
- SiO2 dielectric is ~4.1. Vacuum/Air = 1. Currently: FSG@2.7
- More cap is worse except if you are using it to store data!

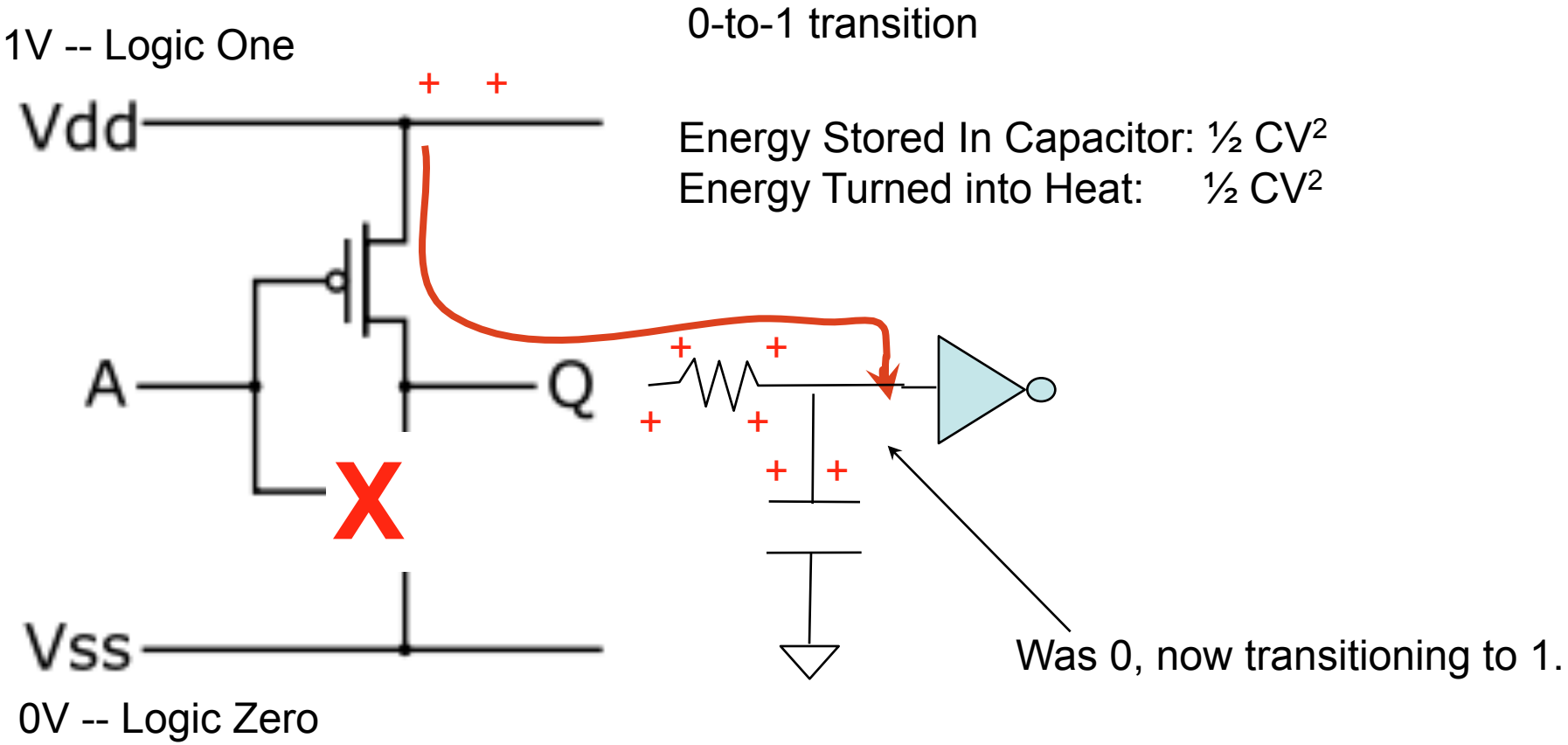
CMOS Logic: Capacitance

- Connection to Vdd transfers charge to the wires and gates on the output; these surfaces form “parasitic” capacitance with other non-connected surfaces at different voltages



CMOS Logic: Cap~Energy

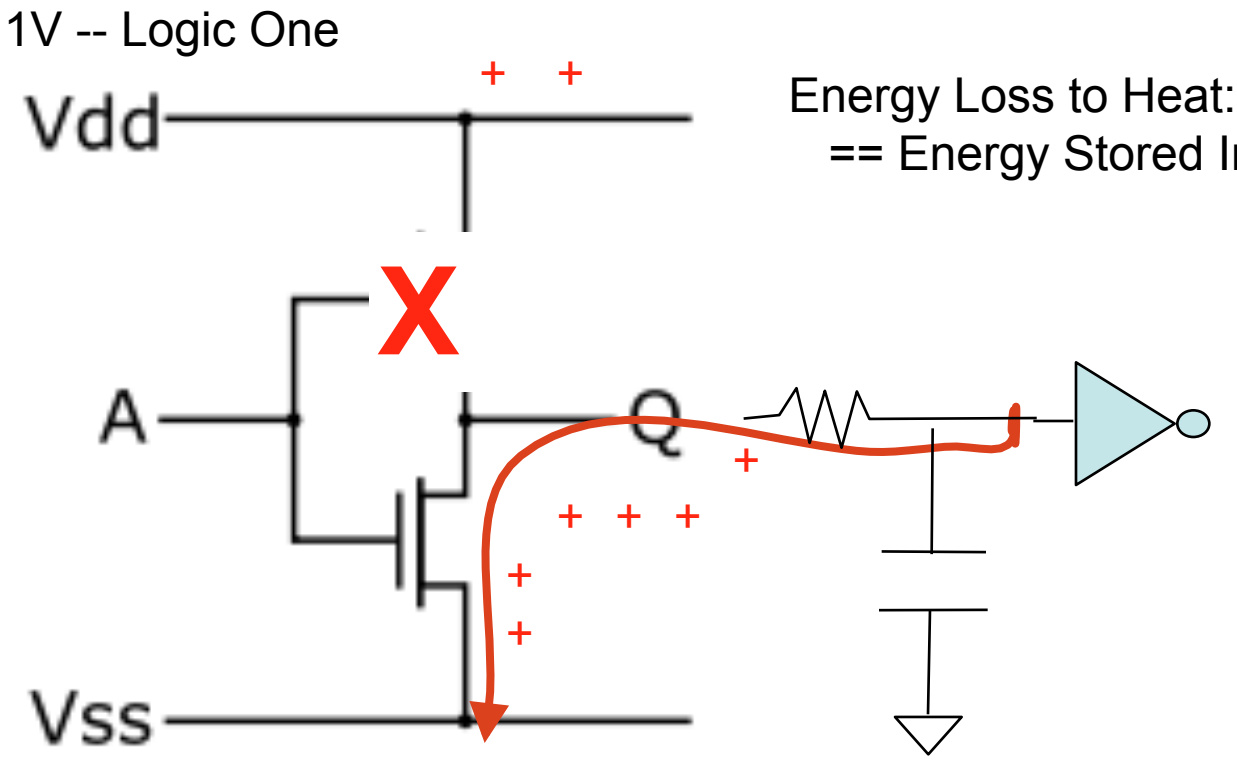
- Energy is expended when charging the parasitic capacitance (i.e 0-to-1 transition);
- Some energy is stored via the parasitic capacitance



Observation: Resistance does not affect energy in CMOS circuits.

CMOS Logic: Cap~Energy

- Connection to Vss discharges those capacitances to the ground rail.
- Where does the energy stored in capacitors go? Heat.

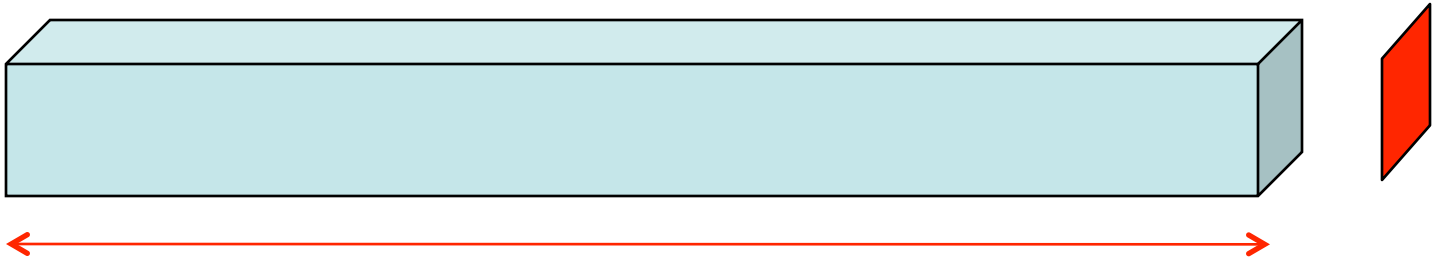


Energy Loss to Heat: ½ CV^2
== Energy Stored In Capacitor

Since # 0→1 transitions ≈ # 1→0 transitions, we often just count one or the other and assign the aggregate cost: CV^2.

Resistance

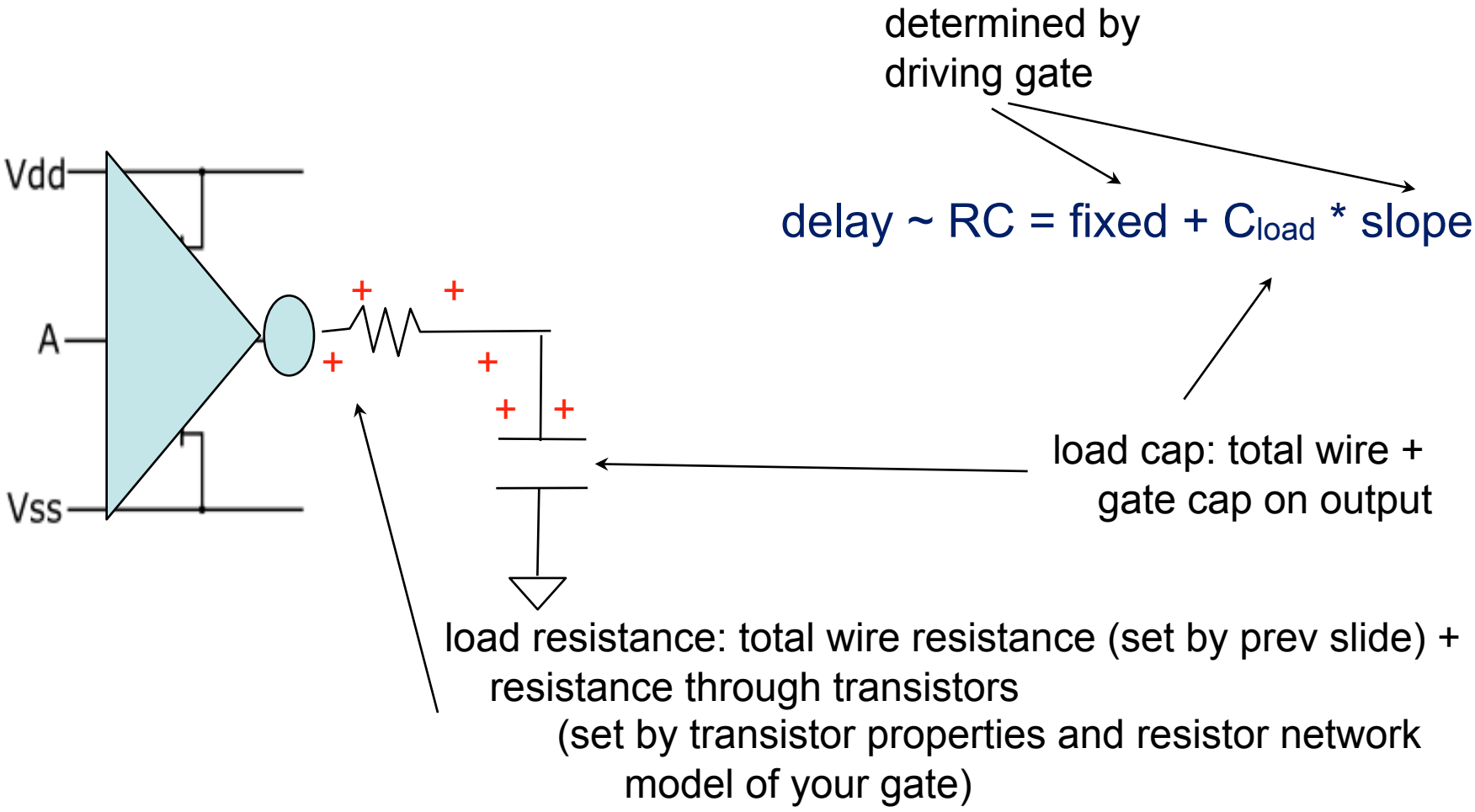
- =~ $\frac{\text{Length} * \text{Resistivity}}{\text{Cross-Section Area}}$



- Wires: Al: 2.7; Cu: 1.7; Au: 2.2; Ag: 1.5
- Superconductor: 0! Actually, really 0.
- Silicon Nanowire / Carbon Nano Tube
  - High contact resistance (~14K ohm)
  - But zero once you are on the wire!

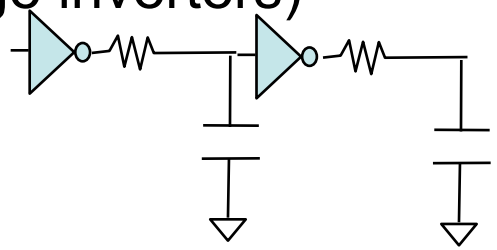
CMOS Logic: Delay

- Delay of switching a transistor corresponds to delay of discharging or charging load capacitance over a resistor
- (i.e. the on-resistance of the charging/discharging transistor.)



Wire Properties

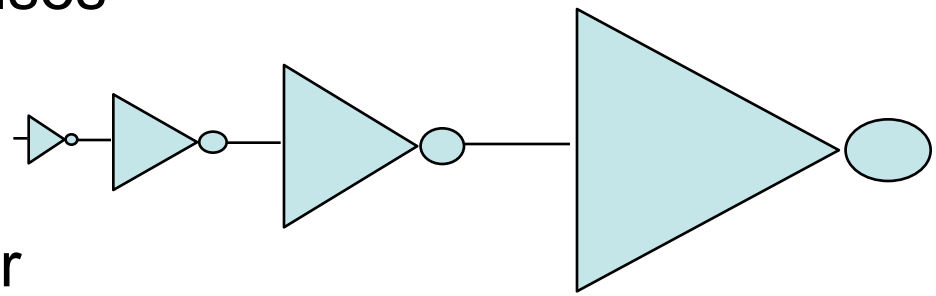
- 1 mm of wire:
  - ~50x-1000x the capacitance of a simple gate
    - scales linearly with wire length
    - fairly process invariant (~.3 fF per um^2)
  - high resistance
    - scales linearly with wire length
    - quadratically worse with process scaling per mm^2 because x-sectional area drops
  - delay ~ RC; power ~ CV^2
    - delays scale quadratically with wire length
    - we can add distributed *repeaters* (large inverters)
      - make delay linear
      - burns additional power
      - tradeoff of delay through repeaters
      - versus delay in wire



CMOS Logic: Driving Large Loads

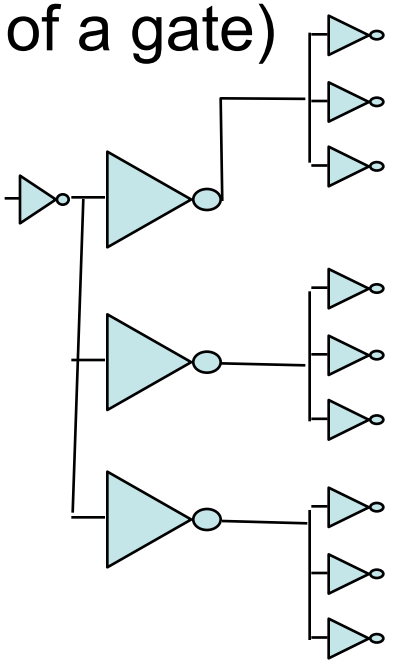
- Best gate to drive large cap loads (e.g. gates+wire): Inverter
  - *Lowest Capacitance and Lowest Resistance*
- The larger the inverter:
  - the larger the transistors delay = fixed + C_{load} * slope
  - the smaller the resistance through the transistor
    - smaller slope
  - the larger the capacitance of the inverter itself
    - larger fixed delay

- Driving large loads typically uses stages of inverters, where the load of the output is 4x the input load of each inverter



CMOS Logic: Driving Large Loads

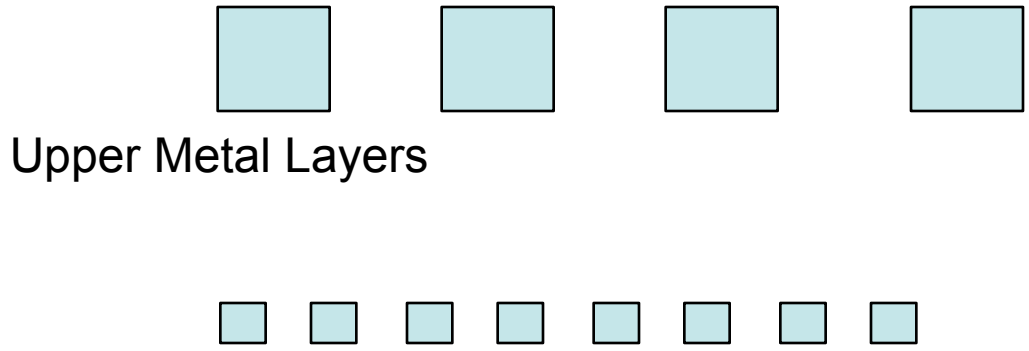
- Examples of large loads:
  - Stall signal in your 5-stage MIPS pipeline
    - fans out to maybe 20 32-bit registers spread across large sections of the chip (wire capacitance!)
    - load: 640x → lg_4(640) -> 4-6 inverter stages
    - but wire cap may be even worse!
  - I/O Driver for Package Pin (~1000x cap of a gate)
  - Clock Tree
    - every register in your design!



Wire Properties

- Latencies:
  - 30 cm in 1 ns (light in a vacuum)
  - 15 cm in 1 ns (light in silicon dioxide or PC Board signal)
  - 1 cm in 1 ns (electrical field; RC limited; top metal)
    - using repeaters on top metal

- How to make a fast wires



Equal capacitance

Upper level can have 40x less resistance.

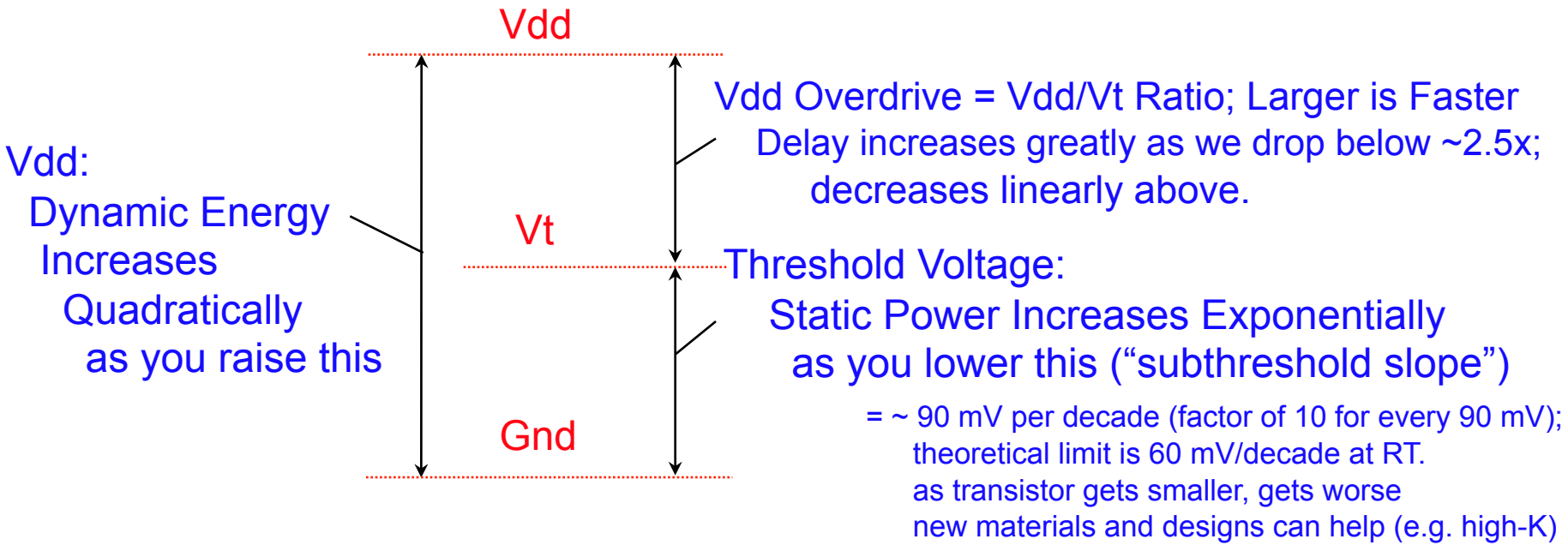
How does latency compare?

How does bandwidth compare?

Optimizing CMOS Power

- Two key parameters are settable on a transistor:
  - Supply power (Vdd) - around 700 mV to 1 V
  - Threshold Voltage (Vt) - around 250 mV to 400 mV

- We're stuck between a rock (dynamic power) and a hard place (static power):
  - ➡ we need to lower Vdd to reduce dynamic power
  - ➡ we need to lower Vt to keep transistors fast (Vdd Overdrive)
  - ➡ we can't lower Vt because leakage will increase exponentially



CMOS Leakage

- Modern processes set Vt to balance between dynamic power, performance, and leakage.
  - "Lower power" technologies set Vt high
    - low standby power
    - greater dynamic power
      - ironically, higher power for actually computing
    - e.g. "slow core" in Nvidia 4+1 Tegra 3 chip on hybrid-Vt technology
  - "General purpose" technologies set Vt lower
    - higher standby power
    - lower dynamic power, and faster
  - Most processors use families of gates with both low-Vt and high-Vt transistors (sometimes three levels) "multi-Vt"
    - low-Vt's: 10x the leakage, ~20% faster
    - 5% of transistors, and shrinking
    - why would we accept this tradeoff?

Silicon Memories

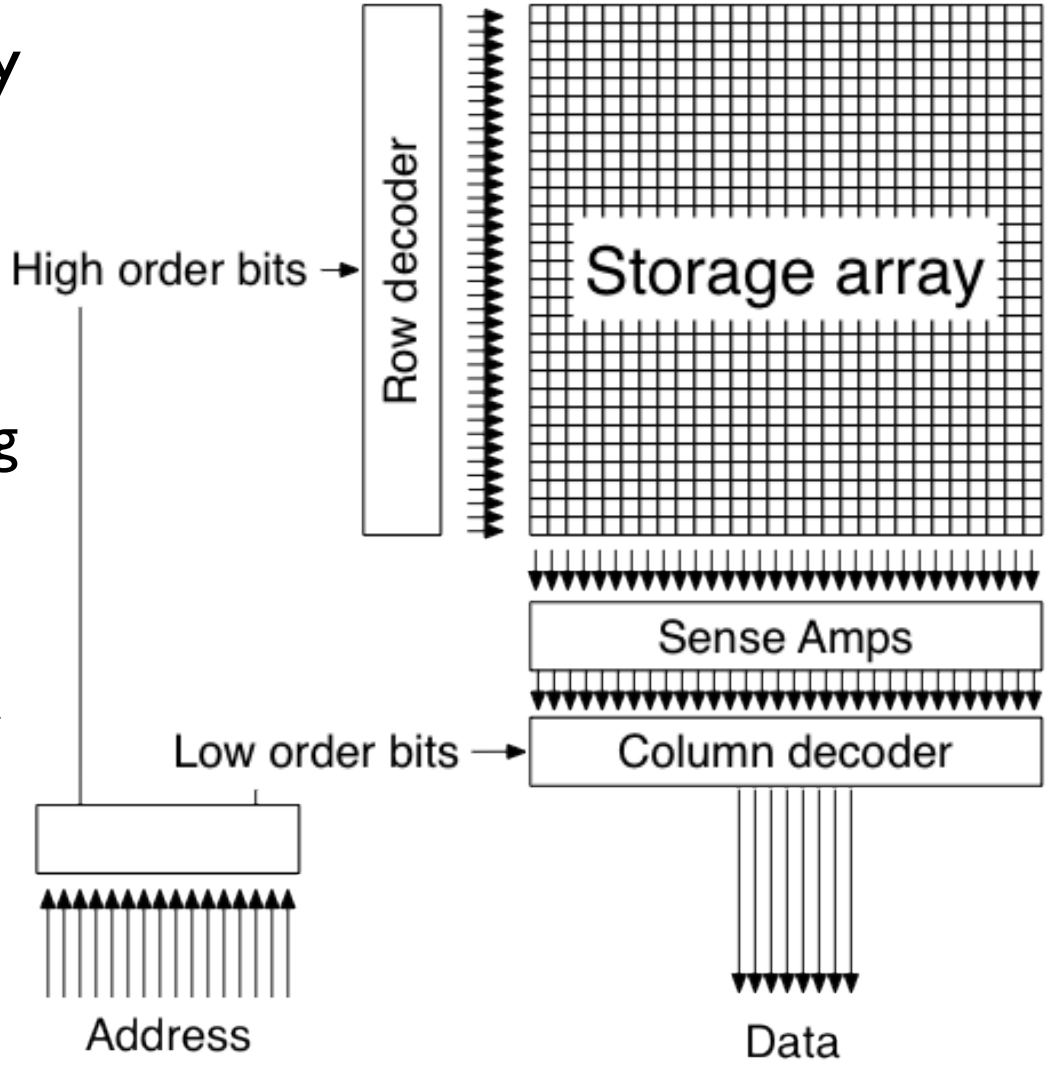
- Why store things in silicon?
  - It's fast!!!
  - Compatible with logic devices (mostly)
- The main goal is to be cheap
  - Dense -- The smaller the bits, the less area you need, and the more bits you can fit on a chip/wafer/through your fab.
  - Bit sizes are measured in F^2 -- the smallest feature you can create.
  - F^2 is a function of the memory technology, not the manufacturing technology.

Questions

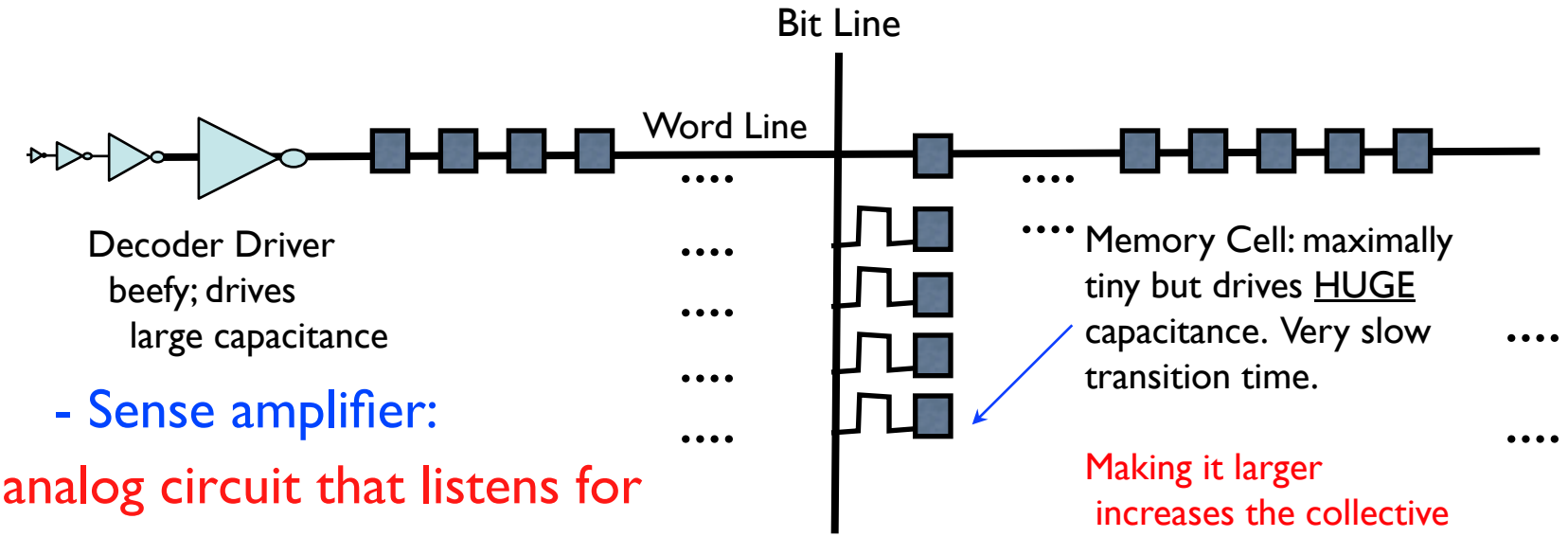
- What physical quantity should represent the bit?
  - Voltage/charge -- SRAMs, DRAMs, Flash memories
  - Magnetic orientation -- MRAMs
  - Crystal structure -- phase change memories
  - The orientation of organic molecules -- various exotic technologies
  - All that's required is that we can sense it and turn it into a logic one or zero.
- How do we achieve maximum density?
- How do we make them fast?

Anatomy of a Memory

- Dense: Build a big array
  - bigger the better
  - less other stuff
  - Bigger -> slower
- Row decoder
  - Select the row by raising a "word line"
  - Binary to One Hot
- Column decoder
  - Select a slice of the row
  - Basically a big mux



CMOS Logic: Driving Large Loads



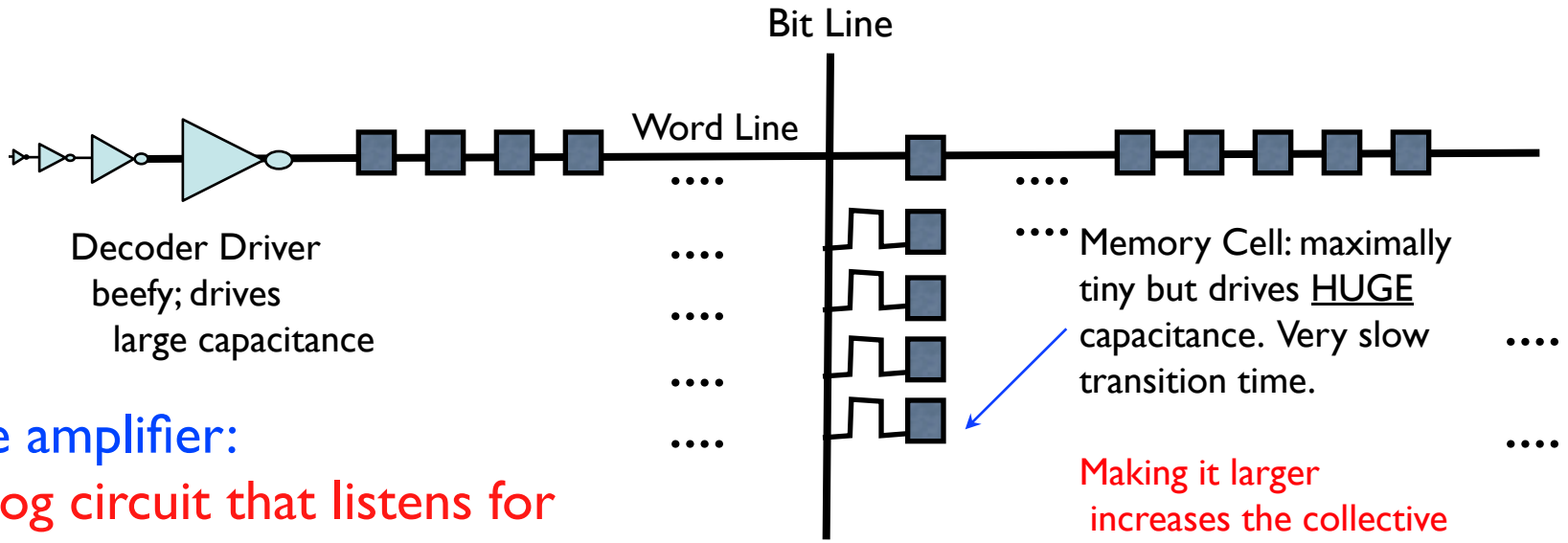
- - Sense amplifier:

  analog circuit that listens for

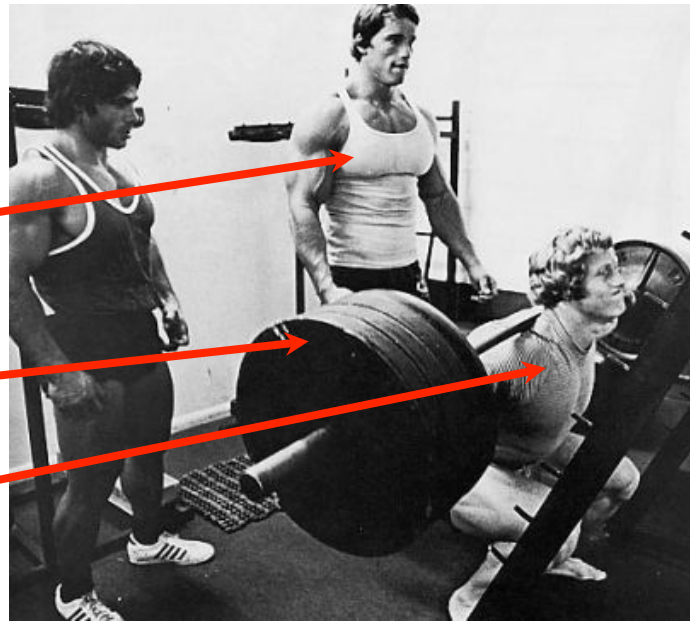
  a change on a wire and amplifies it

- Frequently used in memories
- Why and where do we need sense amps?

CMOS Logic: Driving Large Loads



- Sense amplifier:
  analog circuit that listens for a change on a wire and amplifies it
- Frequently used in memories
- Why and where do we need sense amps?

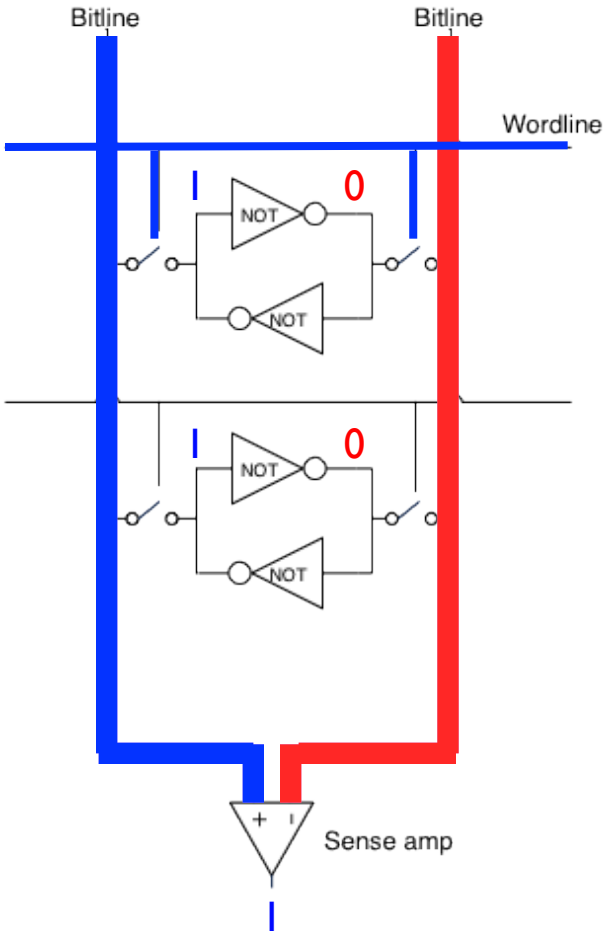


The Storage Array

- Density is king.
  - Highly engineered, carefully tuned, automatically generated.
  - The smaller the devices, the better.
- Making them big makes them slow.
  - Bit/word lines are long (millimeters)
  - They have large capacitance, so their RC delay is long
  - For the row decoder, use large transistors to drive them hard.
  - For the bit cells...
    - There are lots of these, so they need to be as small as possible (but not smaller)

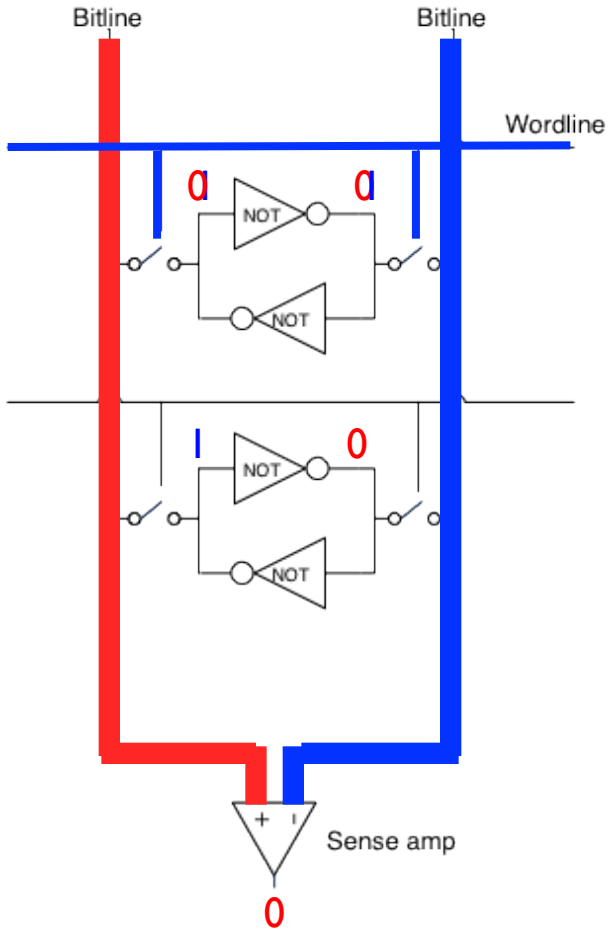
Static Random Access Memory (SRAM)

- Storage
  - Voltage on a pair of cross-coupled inverters
  - Durable in presence of power
- To read
  - Pre-charge two bit lines to Vcc/2
  - Turn on the "word line"
  - Read the output of the sense-amp



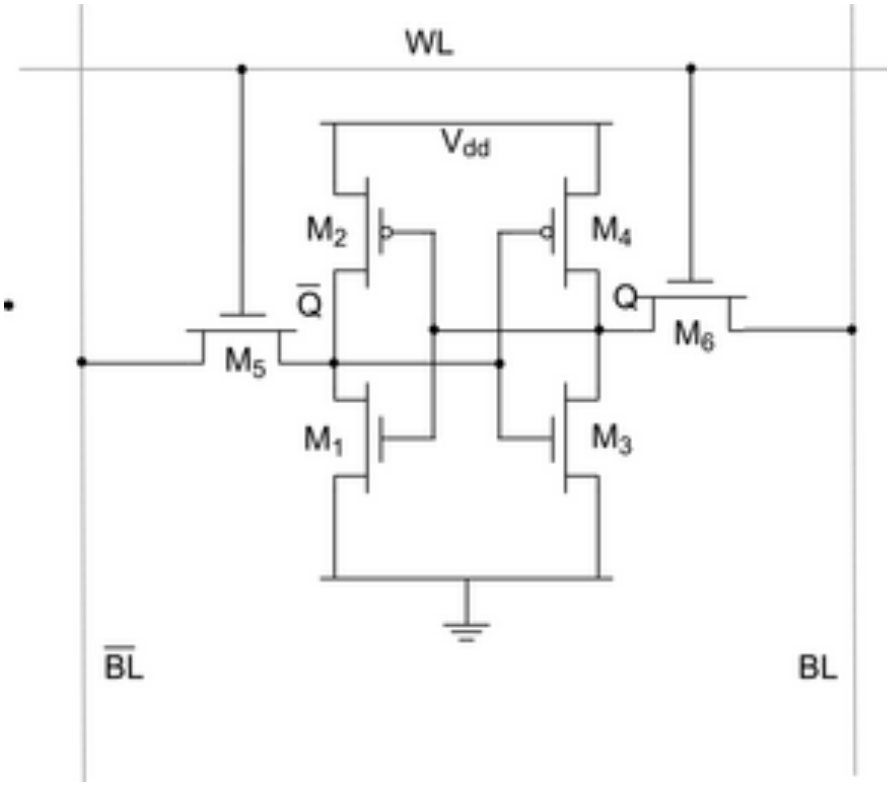
SRAM Writes

- To write
  - Turn off the sense-amp
  - Turn on the wordline
  - Drive the bitlines to the correct state
  - Turn off the wordline



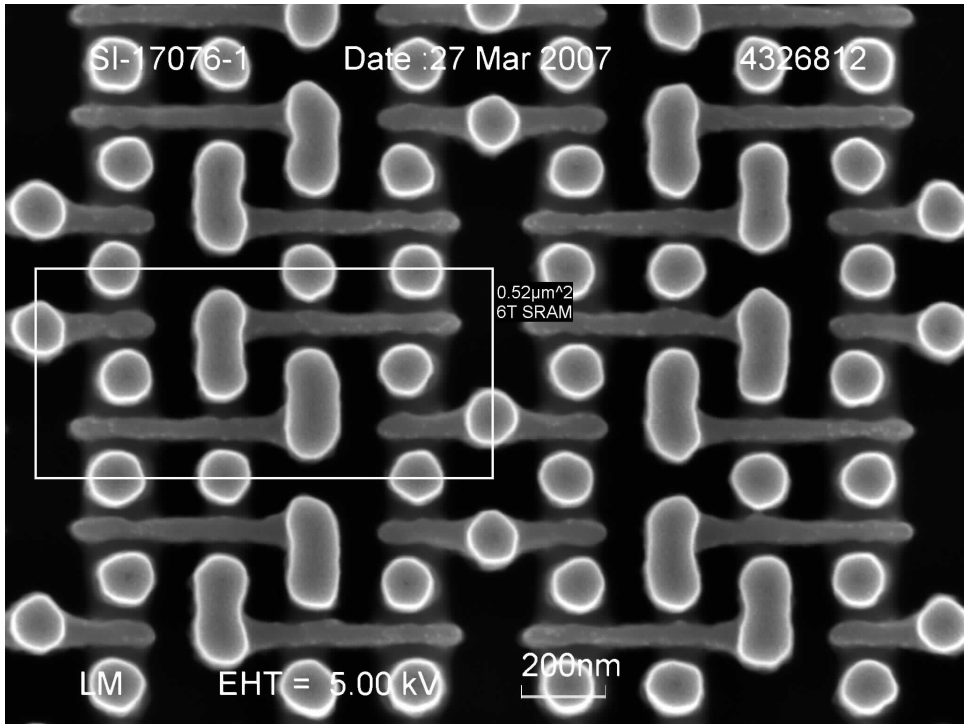
Building SRAM

- This is "6T SRAM"
- 6 "basic devices" is pretty big
- Moving to more T's because process technology is becoming more fickle.



SRAM Density

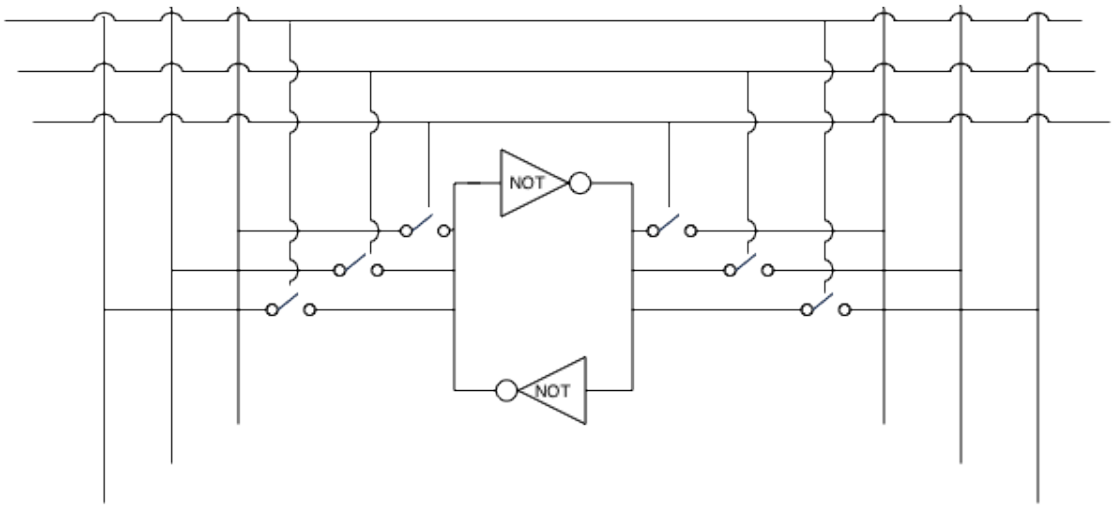
- At 65nm: 0.52um^2
- 123-140 F^2
- 1 F^2 is one "square feature"
- [ITRS 2008]



65nm TSMC 6T SRAM

Multi-Ported SRAM Cells

- Add word and bit lines
- Read/write multiple things at once
- Area grows as square of # of ports
- Quite costly; only used for small #'s of words
  - e.g. register files

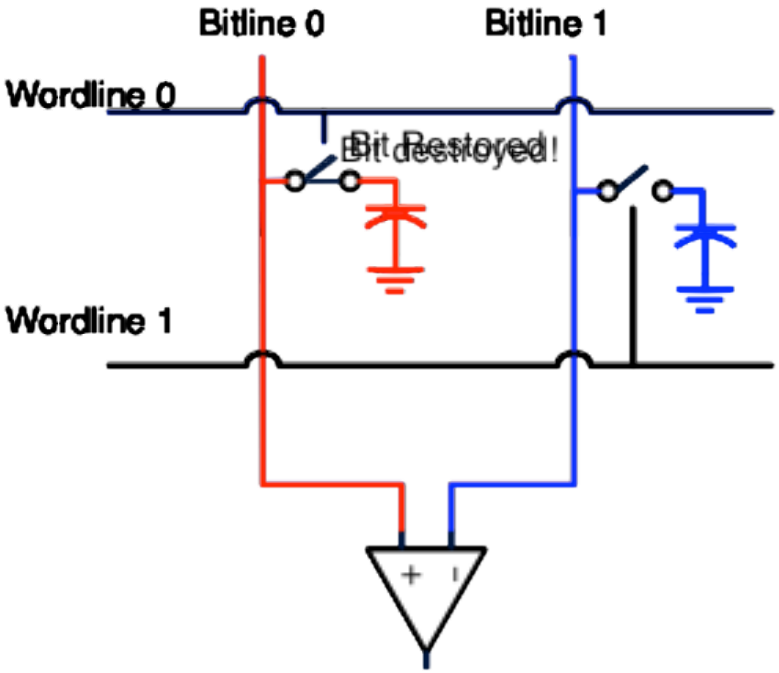


SRAM Performance

- Read and write times
  - 10s-100s of ps
- Bandwidth
  - Registers -- 324GB/s
  - L1 cache -- 128GB/s
  - Samsung K7D323674C -- 3.6GB/s
- Durability
  - Infinite (not quite actually, but very close)

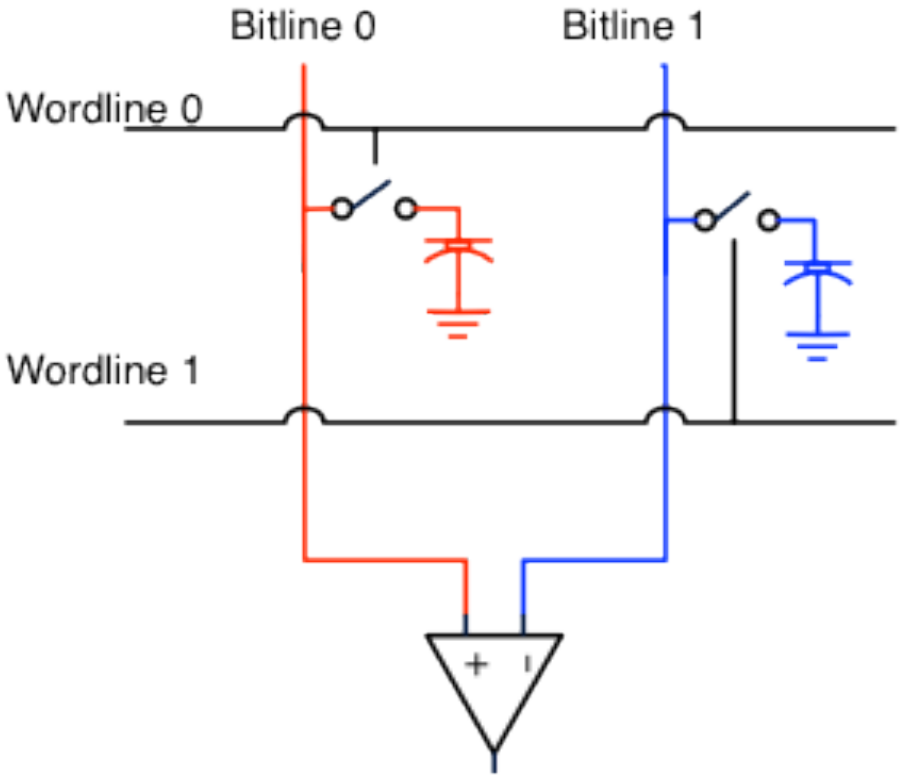
Dynamic Random Access Memory (DRAM)

- Storage
  - Charge on a capacitor
  - Decays over time (us-scale)
  - This is the "dynamic" part.
  - About $6F^2$: 20x better than SRAM
- Reading
  - Precharge
  - Assert word line
  - Sense output
  - Refresh data

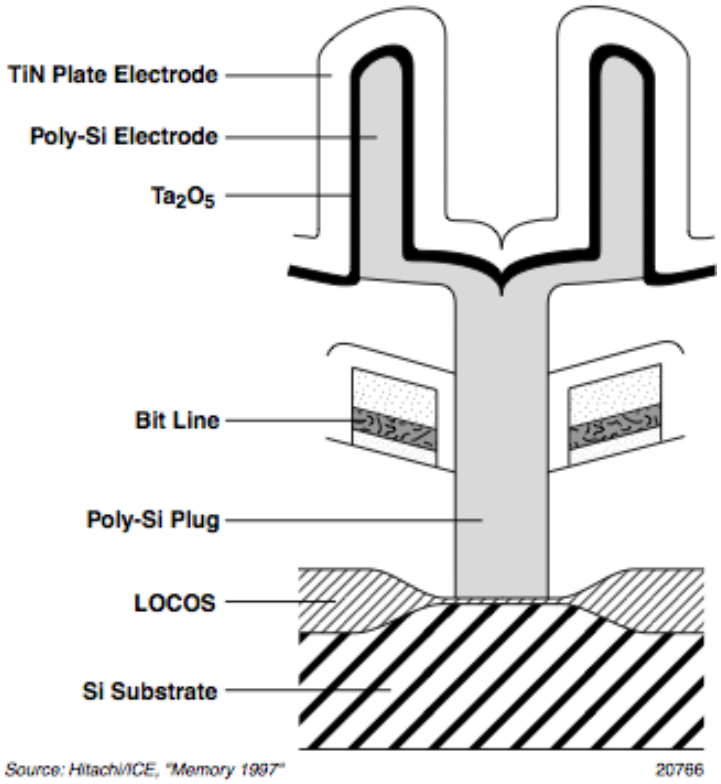


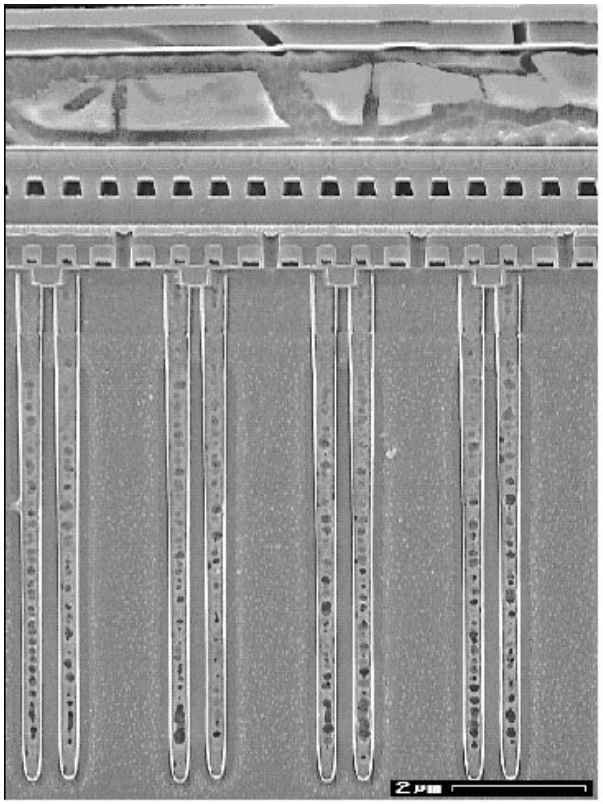
DRAM: Write and Refresh

- Writing
  - Turn on the wordline
  - Override the sense amp.
- Refresh
  - Every few micro-seconds, read and re-write every bit.
  - Consumes power
  - Takes time



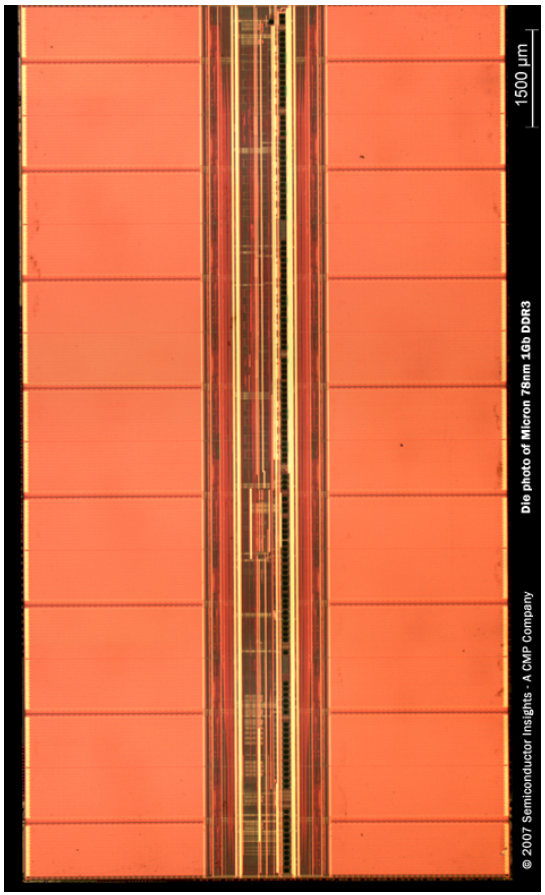
DRAM Lithography





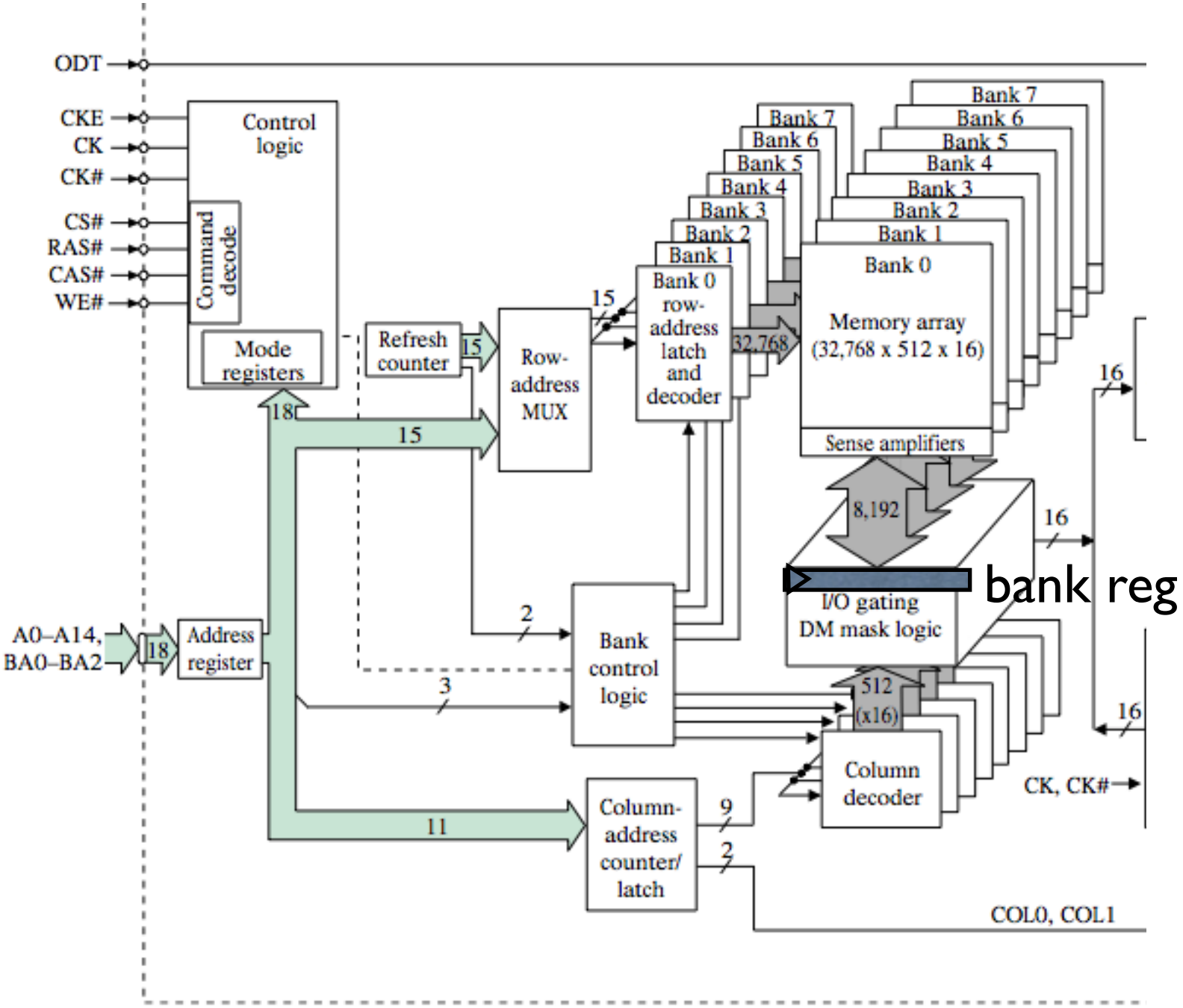
DRAM Devices

- There are many banks per die (16 at left)
  - Multiple can be active at once to hide latencies
  - Parallelism!!!
- Example
  - open bank 1, row 4
  - open bank 2, row 7
  - open bank 3, row 10
  - read bank 1, column 8
  - read bank 2, column 32
  - ...

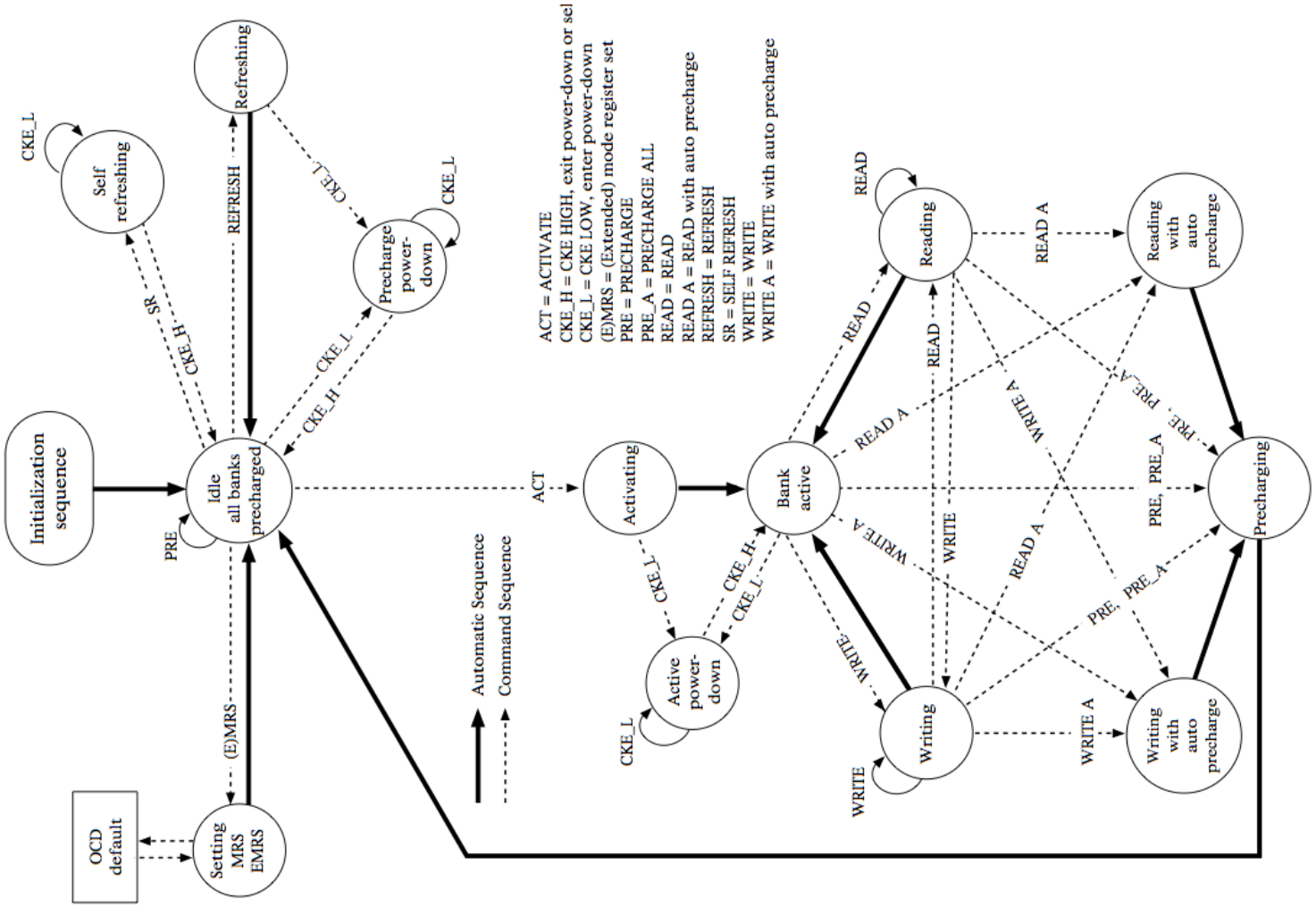


Micron 78nm 1Gb DDR3

DRAM: Micron MT47H512M4



DRAM: Micron MT47H512M4



DRAM Variants

- The basic DRAM technology has been wrapped in several different interfaces.
- SDRAM (synchronous)
- DDR SDRAM (double data-rate)
  - Data clocked on rising *and* falling edge of the clock.
- DDR2
  - Example on previous slides
- DDR3, DDR4, LVDDR2, LVDDR3 ...
- GDDR2-5 -- For graphics cards.
- FB-DIMMS

DDR2 SDRAM

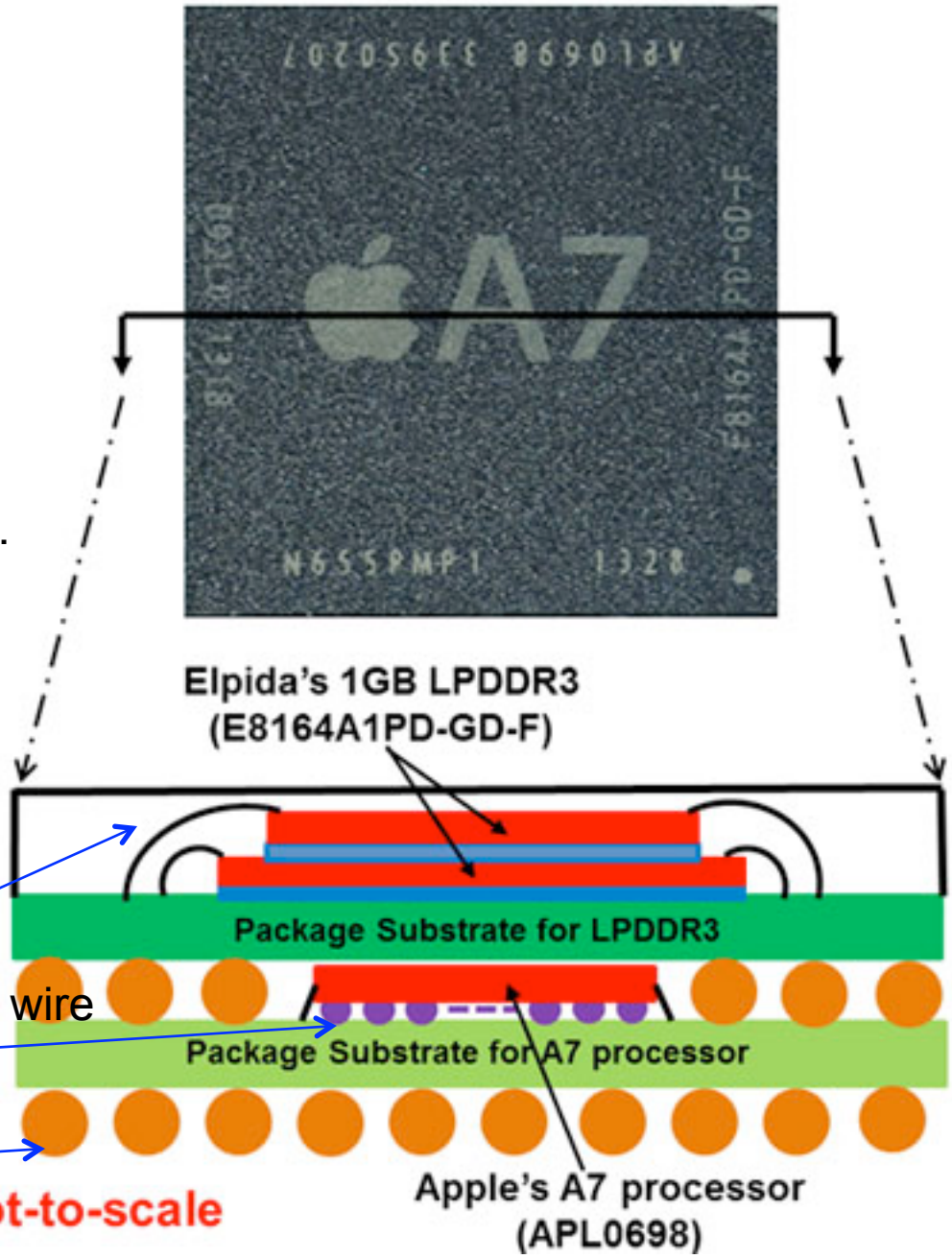
- Fewer, larger banks.
- Pin count per package (DIMM): 14 address, 16 data
- DIMM data path is 64bits
- Data rate: up to 800Mhz DDR (1600Mhz effective)
- Bandwidth per DIMM GTNE: 12.8GB/s
  - *guaranteed not to exceed*
- Multiple DIMMs can attach to a bus
  - Reduces bandwidth/GB (a good idea?)



Package on Package

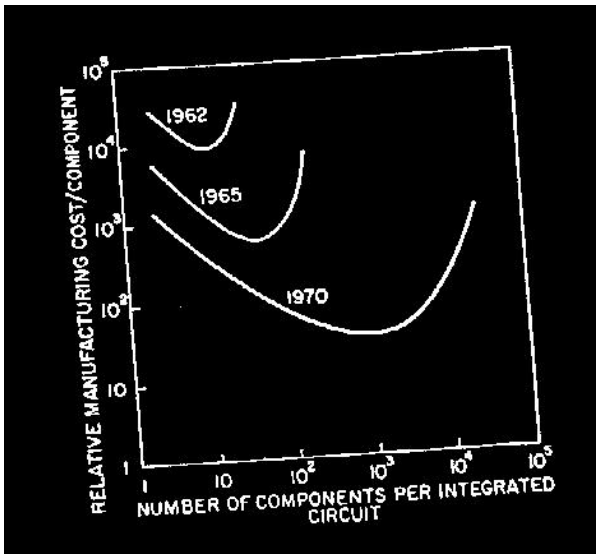
Put the DRAM on top of the processordie.

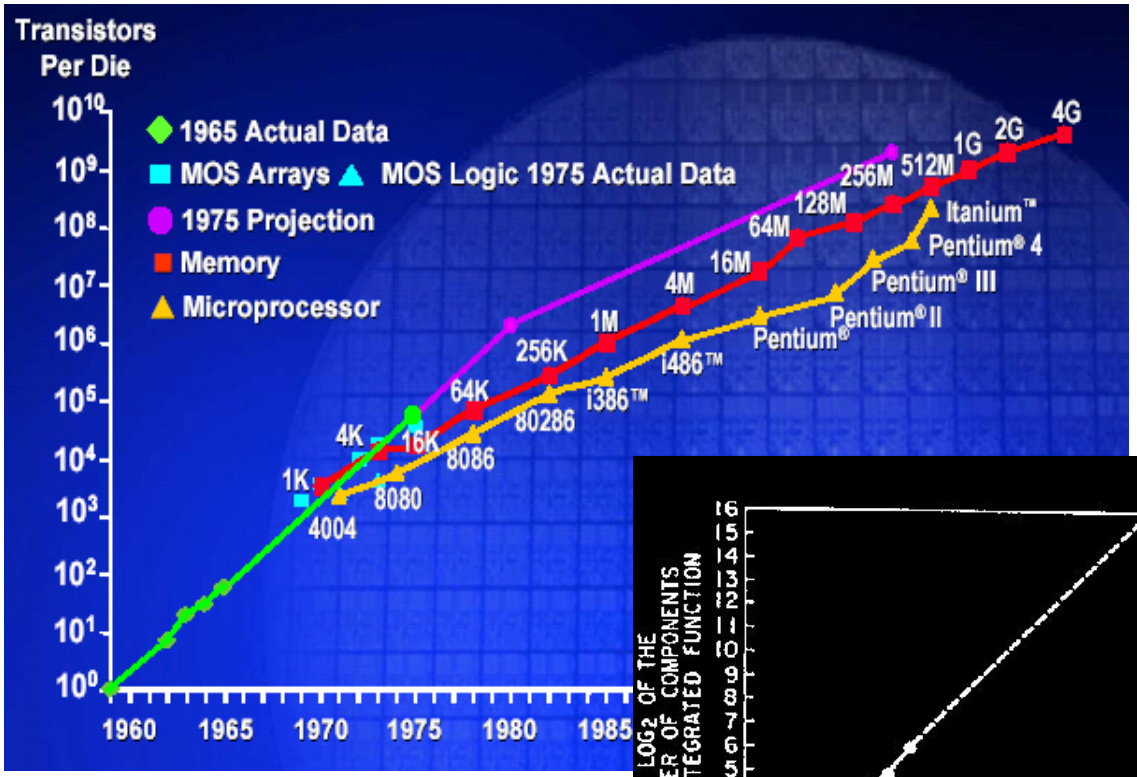
Package substrates are built out of PCB's with ultra-fine space and trace.

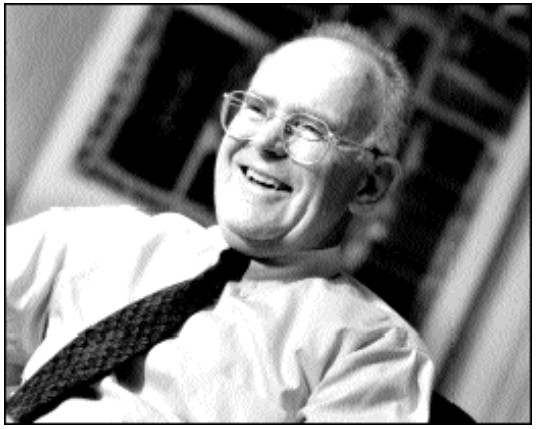


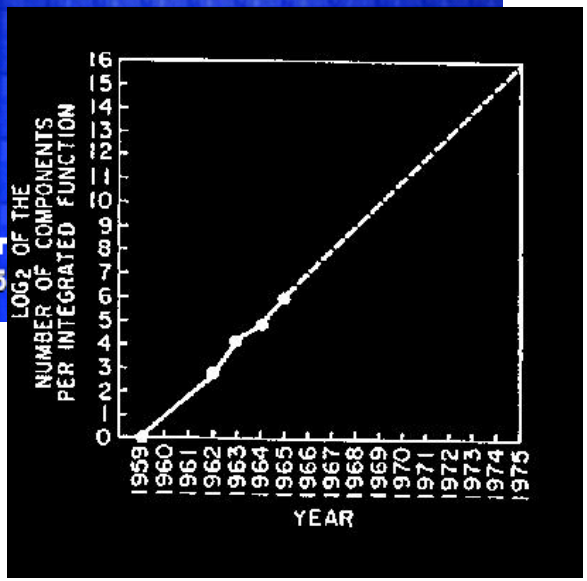
Technology Scaling

Moore's Law: 2X transistors / "year"







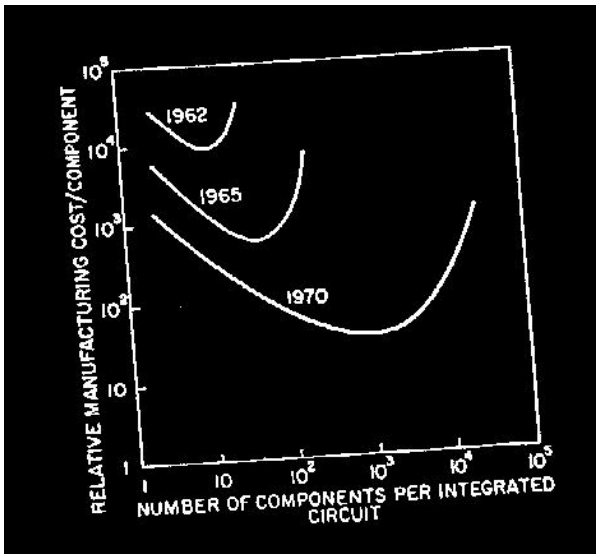


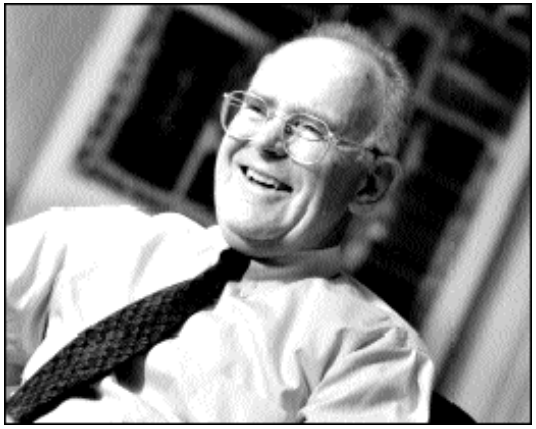
"Cramming More Components onto Integrated Circuits"

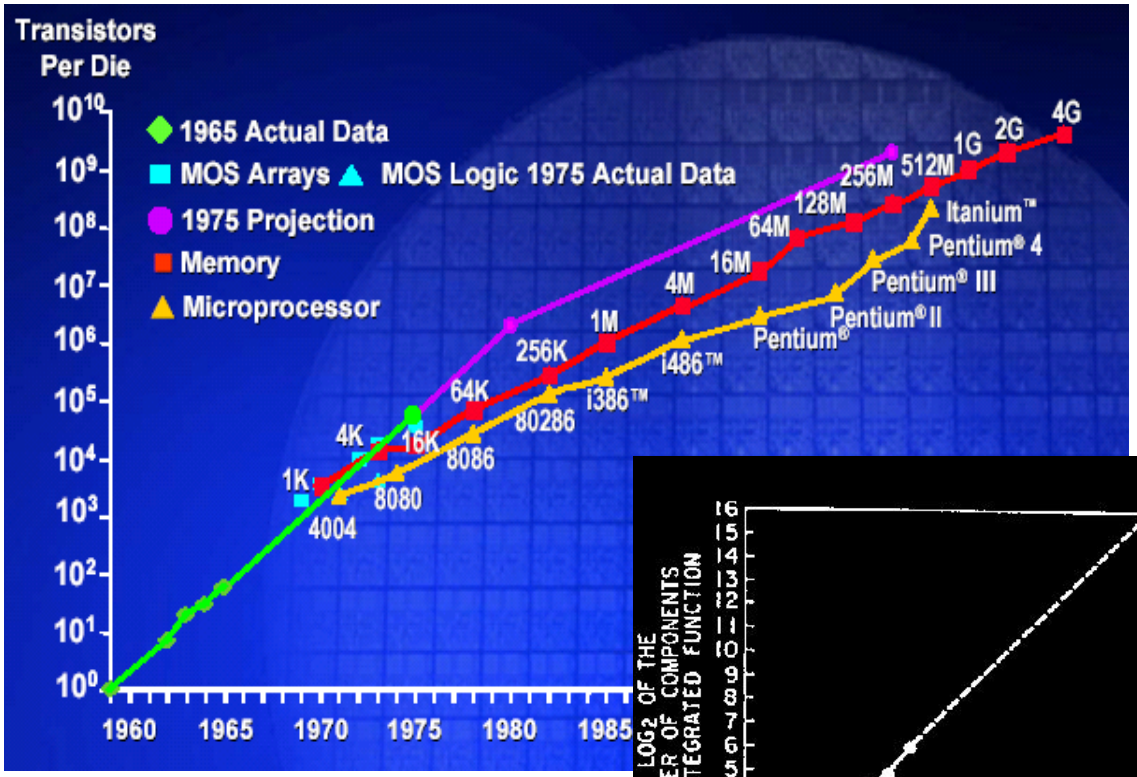
– Gordon Moore, Electronics, 1965

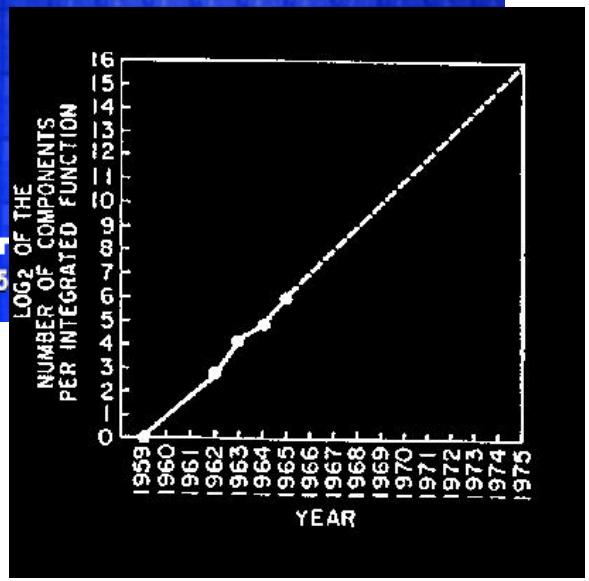
on transistors / cost-effective integrated circuit double every N months ($12 \leq N \leq 24$)

Moore's Law: 2X transistors / "year"









N seems to be rising - now 30?

"Cramming More Components onto Integrated Circuits"

– Gordon Moore, Electronics, 1965

on transistors / cost-effective integrated circuit double every N months ($12 \leq N \leq 24$)

The essence of Moore's Law: scale each dimension by ~1/√2 = ~0.71

each scaling halves the area of a fixed design

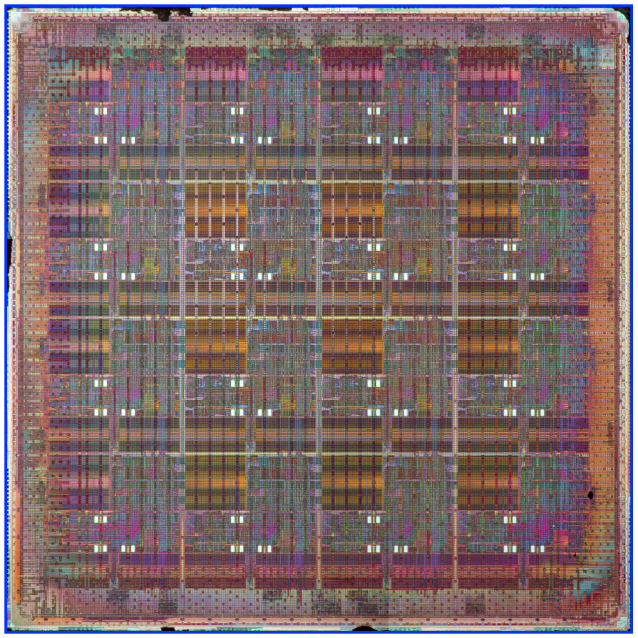
× .72 =

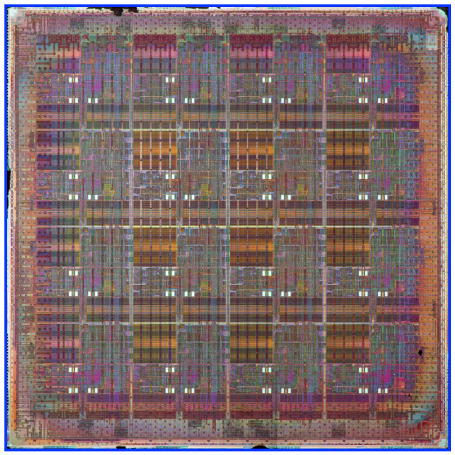
× .69 =

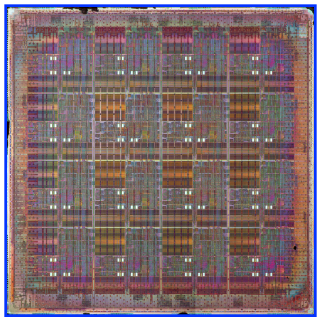
18.2 mm

13.1 mm

9.04 mm

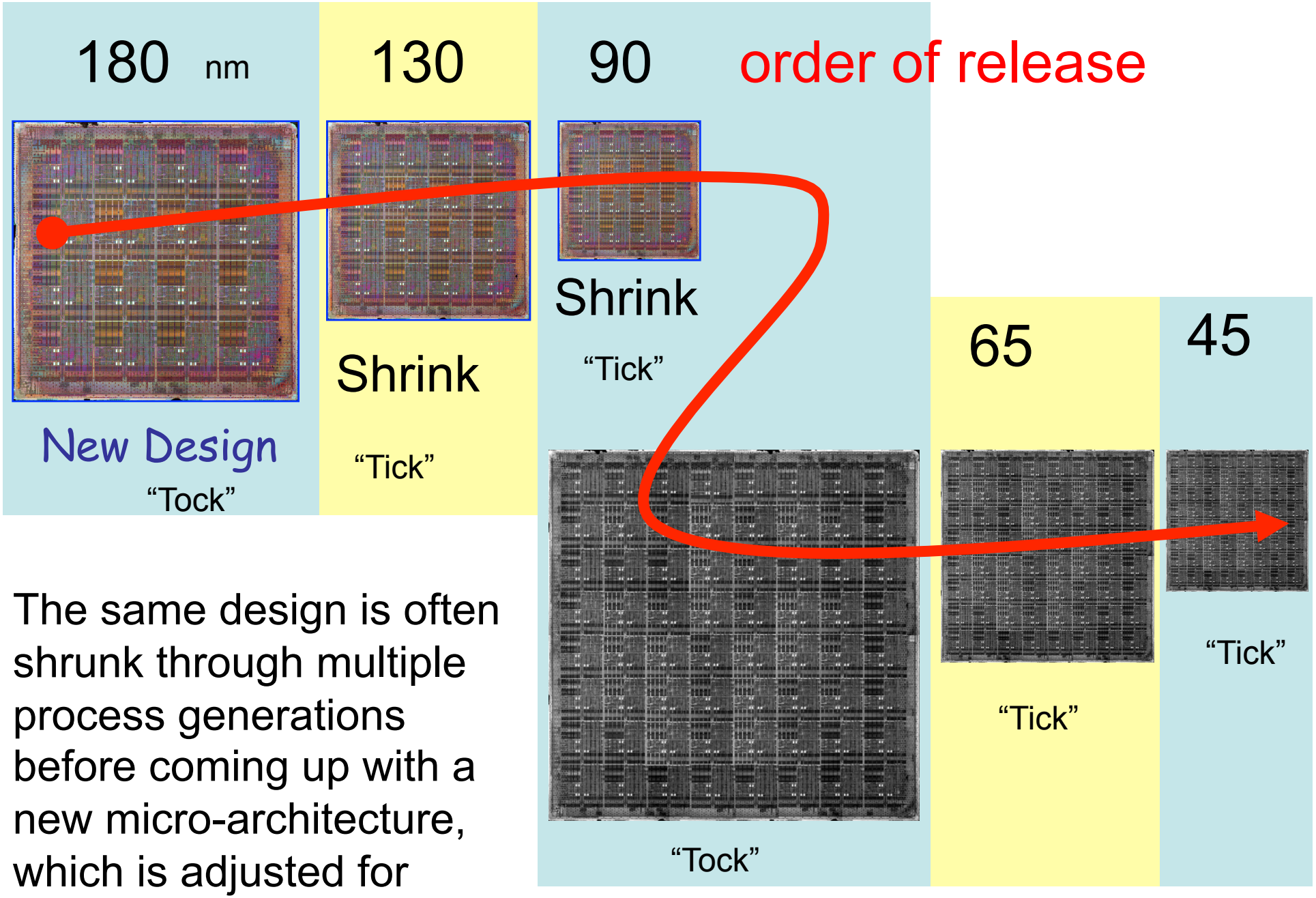




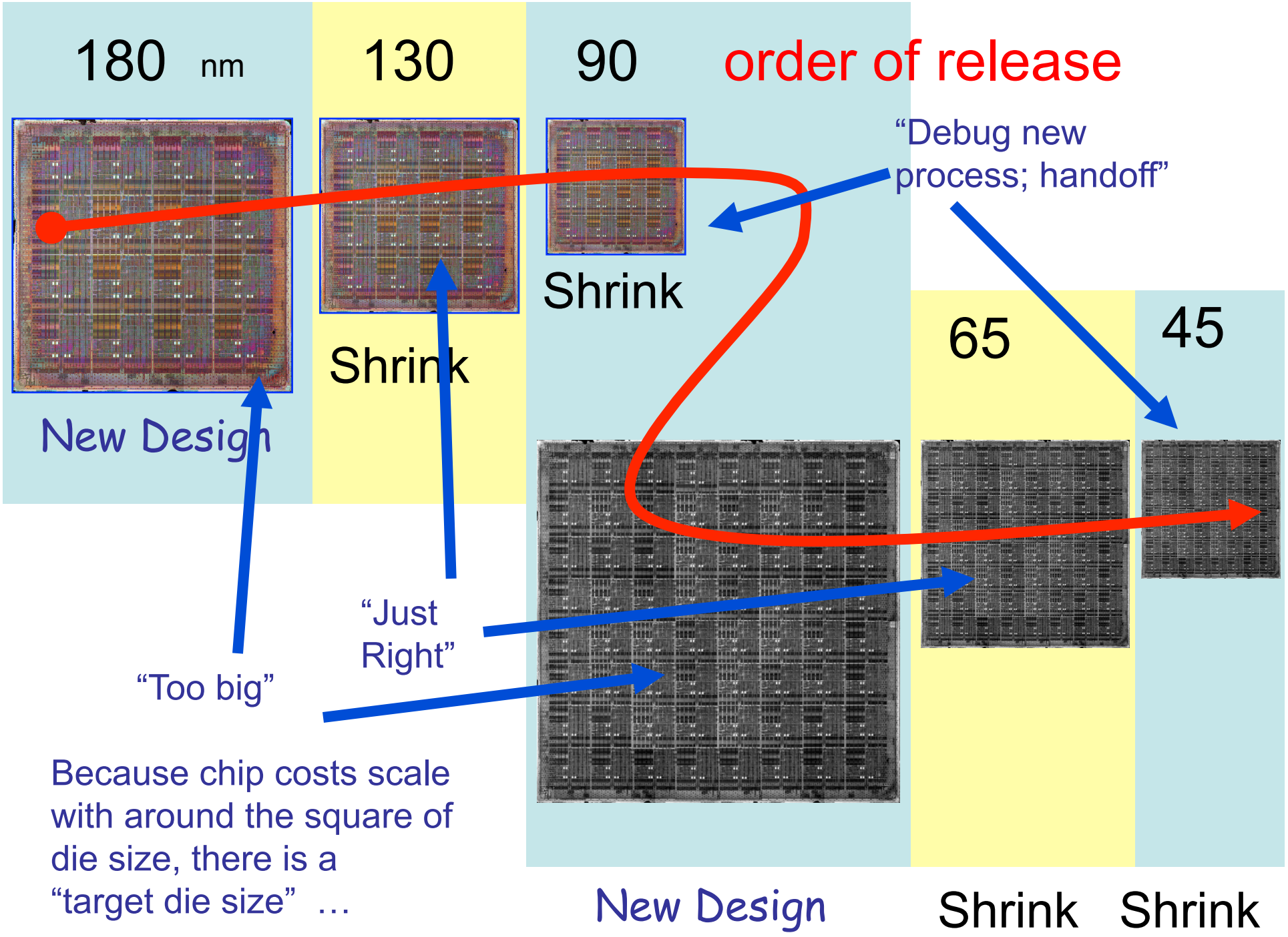


"Process" or "Lithography"
or "Litho" Generation
smallest wire pitch = ~ 2-3x litho

180 nm ⟶ 130 nm ⟶ 90 nm

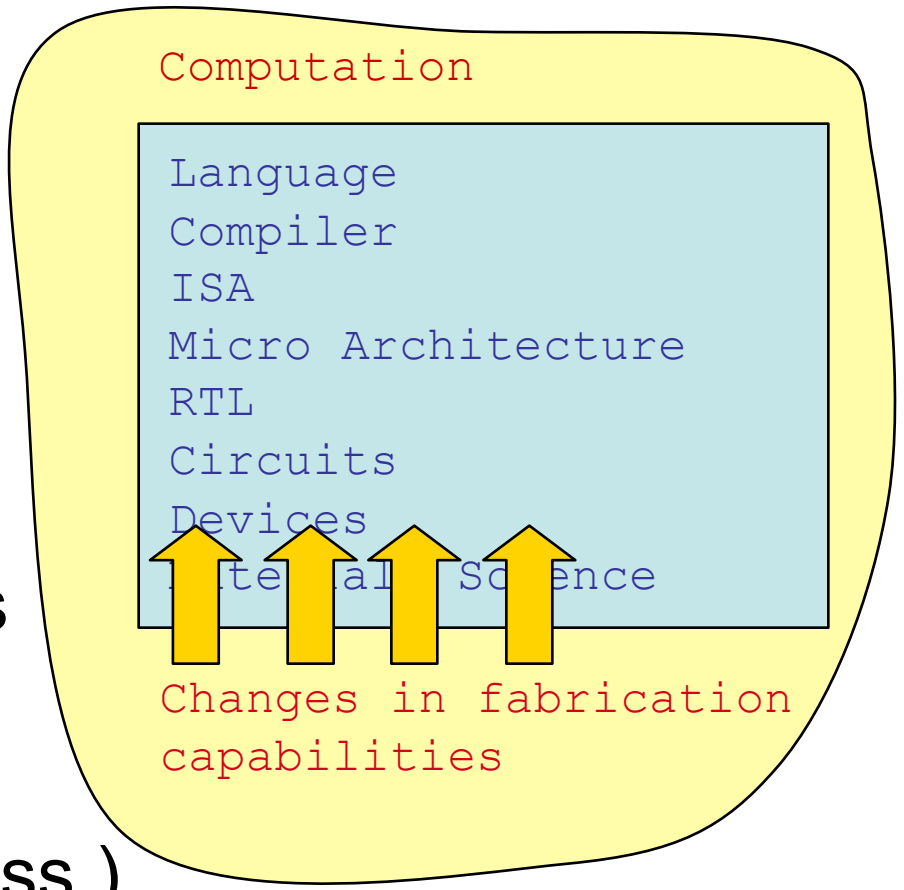


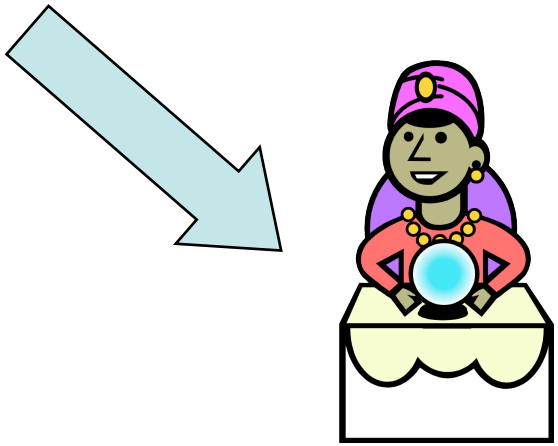
The same design is often shrunk through multiple process generations before coming up with a new micro-architecture, which is adjusted for technology changes.


180 nm
130
90
order of release
"Debug new process; handoff"
Shrink
Shrink
New Design
65
45
"Just Right"
"Too big"
Because chip costs scale with around the square of die size, there is a "target die size" …
New Design
Shrink
Shrink

Tech Trends

Since technology
change is such a
big influence in architecture,
and because it takes 3-6 years
to create a totally new design,
we try to predict & exploit it
(with varying degrees of success.)

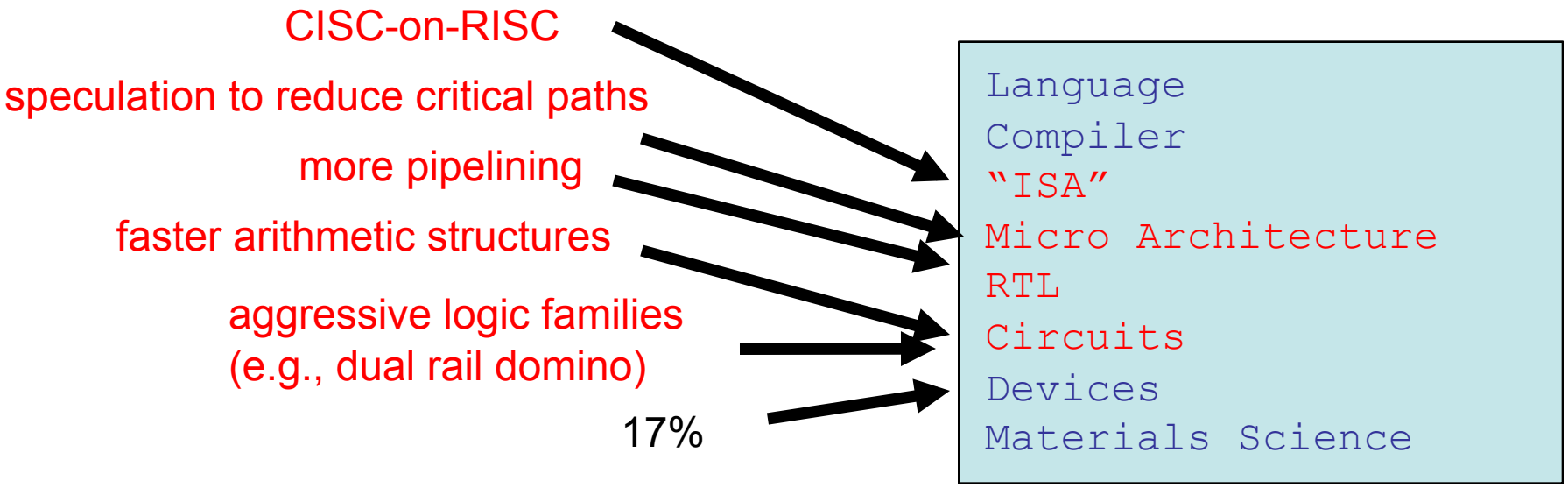




Transistor Frequency Scaling

Transistor (not processor) frequency scales approximately linearly with feature size.(e.g., 1.4x / generation), est. 17%/year

So where did the remainder of the
39% per year and 58% per year come from?

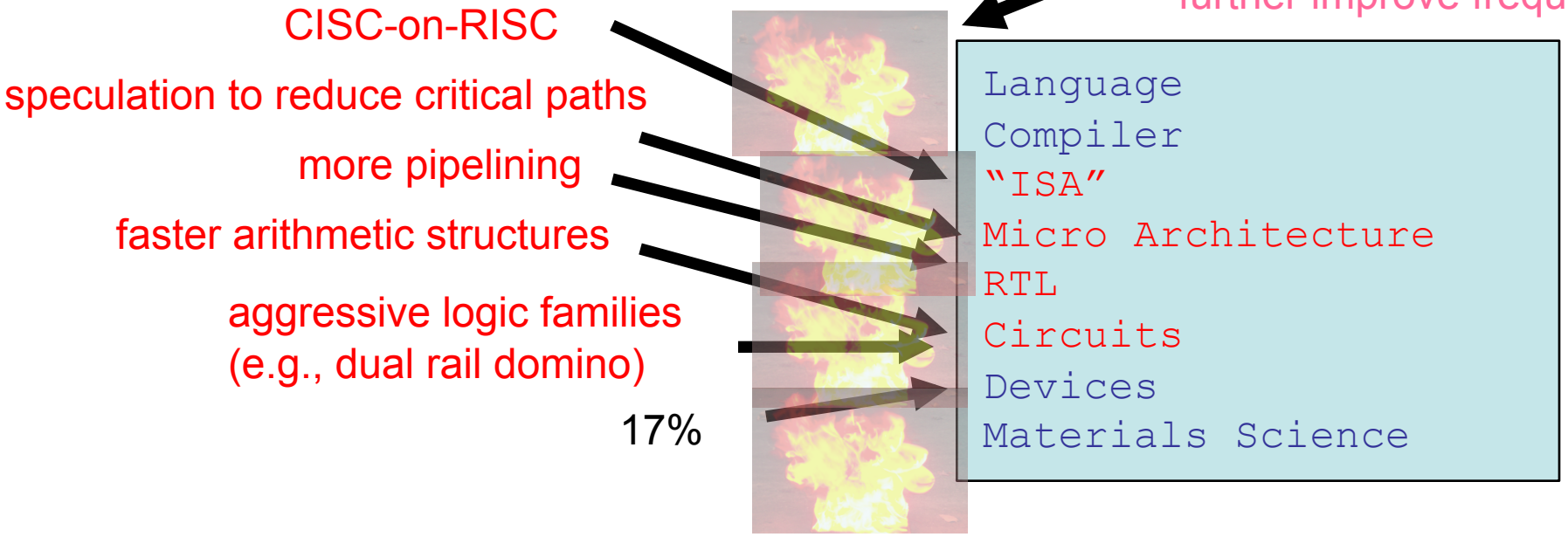


Transistor Frequency Scaling

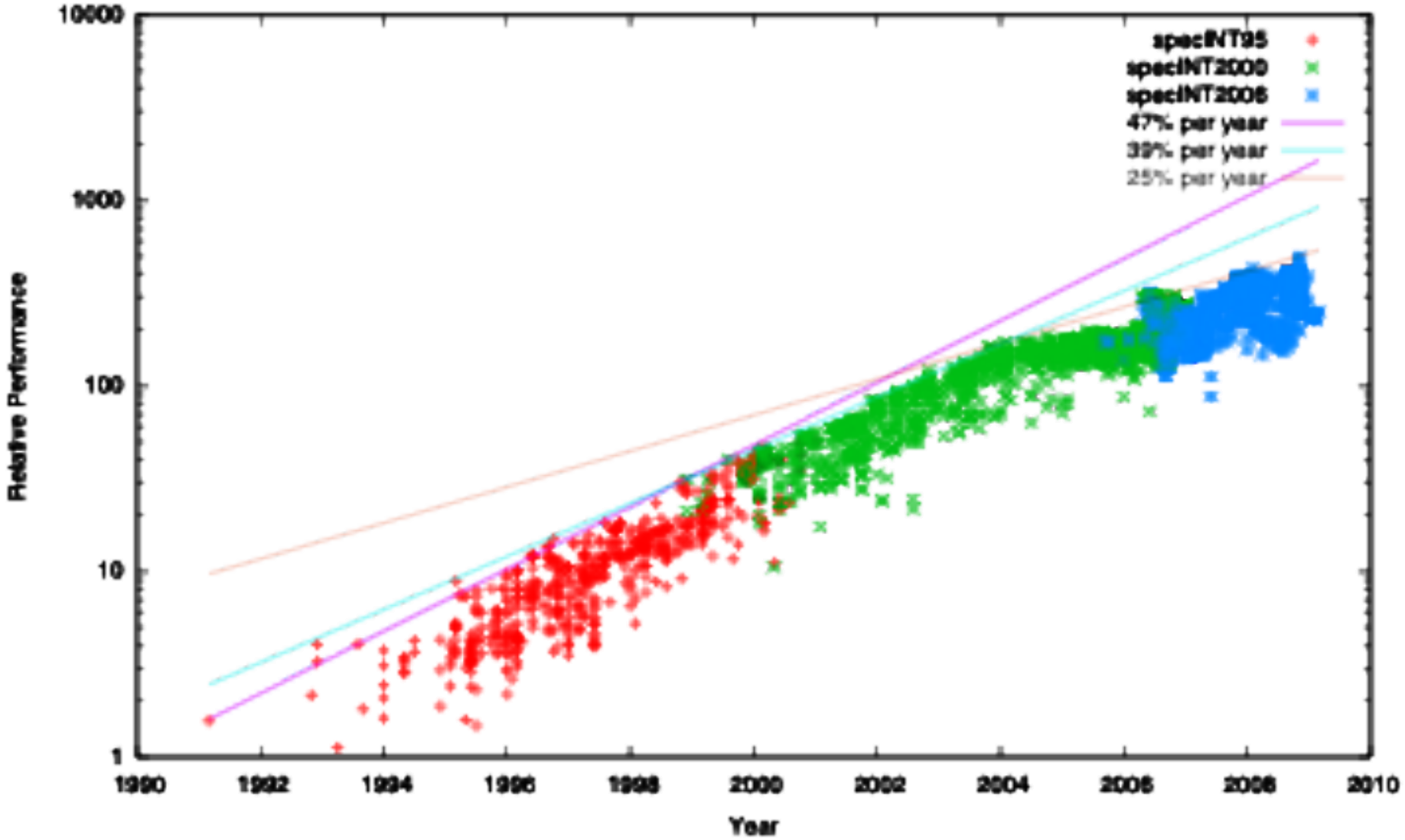
Transistor (not processor) frequency scales approximately linearly with feature size. (e.g., 1.4x / generation), est. 17%/year

So where did the remainder of the 39% per year and 58% per year come from?

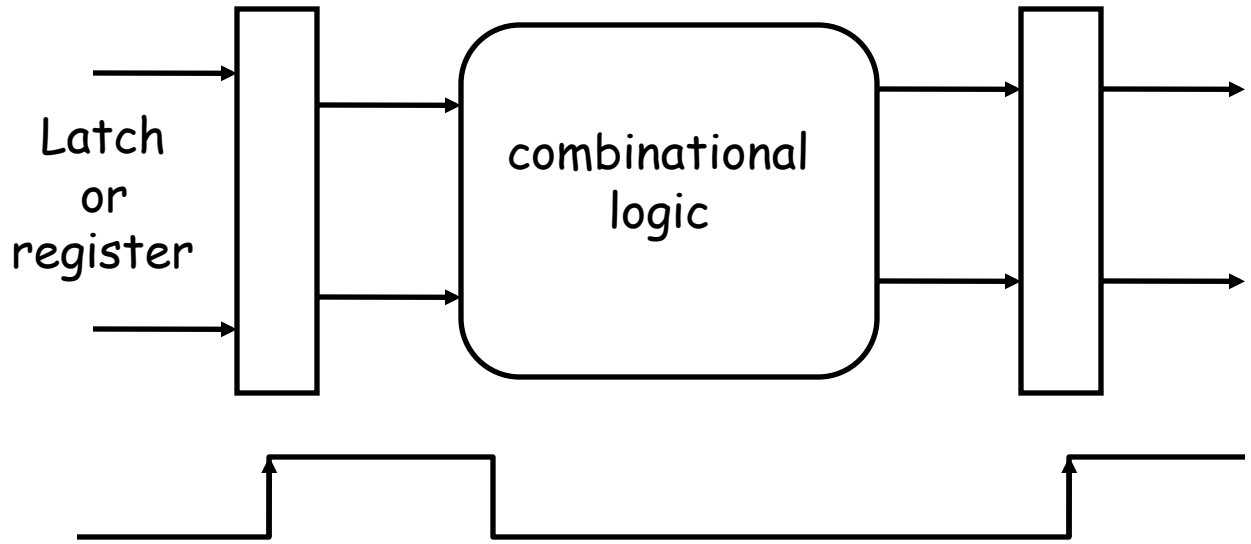
the power wall - has reduced ability of these things to further improve frequency



Computer Performance



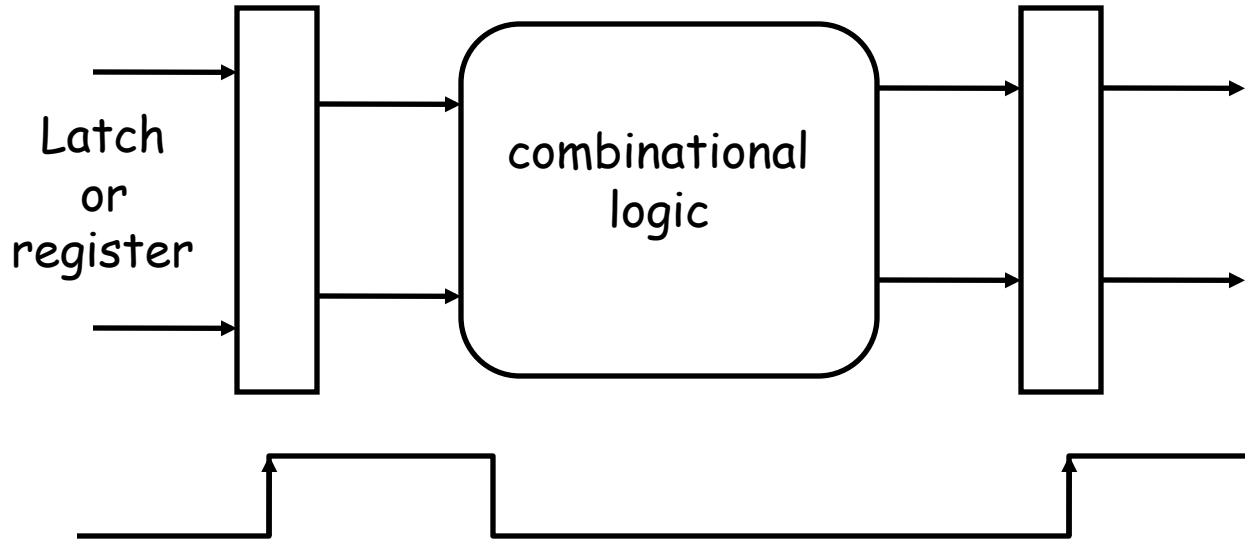
What determines the cycle time today?



- Old days: 10-16 levels of gates
- Today: gate delays, clock overhead, wire delays, + POWER

Adapted from Patterson, CSE 252 Sp06 Lecture 1 © 2006 UC Berkeley.

Processor Frequency and Power



$P = V^2FCa$

V = Processor voltage

F = frequency

C = Capacitance

a = Activity factor

- Lower frequency
  - Less overhead
  - Linear power savings
  - Potentially lower voltage -> quadratic power savings
  - More layers of logic per clock
  - Performance impact?

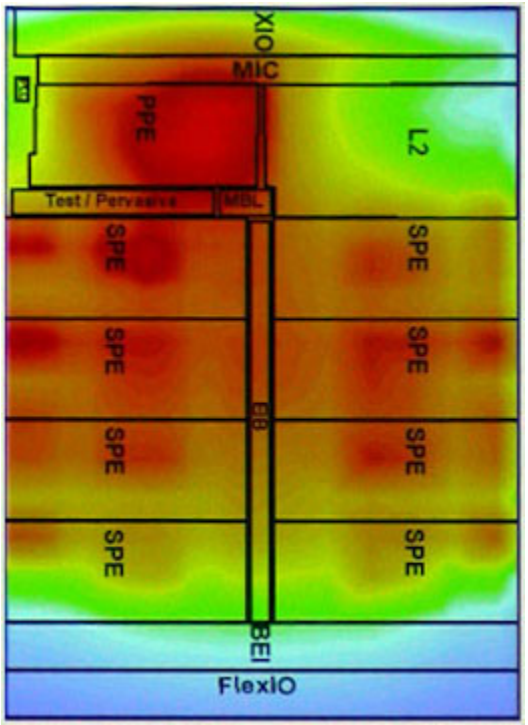
Processor Frequency and Power

Vdd (Volts)	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8	4	4.2	4.4	4.6	4.8	5	5.2
1.3	48C 4W	49C 4W	50C 5W	50C 6W	51C 6W	52C 7W	53C 7W	54C 7W	55C 8W	56C 8W	57C 9W	58C 9W	59C 10W	60C 10W	61C 10W	63C 11W	61C
1.2	39C 2W	39C 3W	40C 3W	41C 4W	42C 4W	42C 4W	43C 5W	44C 5W	45C 5W	45C 5W	46C 6W	47C 6W	47C 7W	48C	49C		
1.1	32C 2W	33C 2W	33C 3W	35C 3W	35C 3W	36C 3W	36C 4W	37C 4W	37C 4W	38C 4W	38C 4W	39C	39C				
1	28C 2W	28C 2W	29C 2W	29C 2W	30C 2W	30C 3W	30C 3W	31C 3W	31C 3W	31C 3W	32C						
0.9	25C 1W	26C 1W	26C 1W	26C 2W	27C 2W	27C 2W	27C	Failed									

Cell Processor SPE
Top Speed: 5.2 GHz
Shipping speed: 3.2 GHz

2 GHz lost due to power constraints

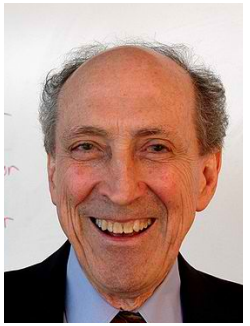
source: ISSCC 2005, p. 135

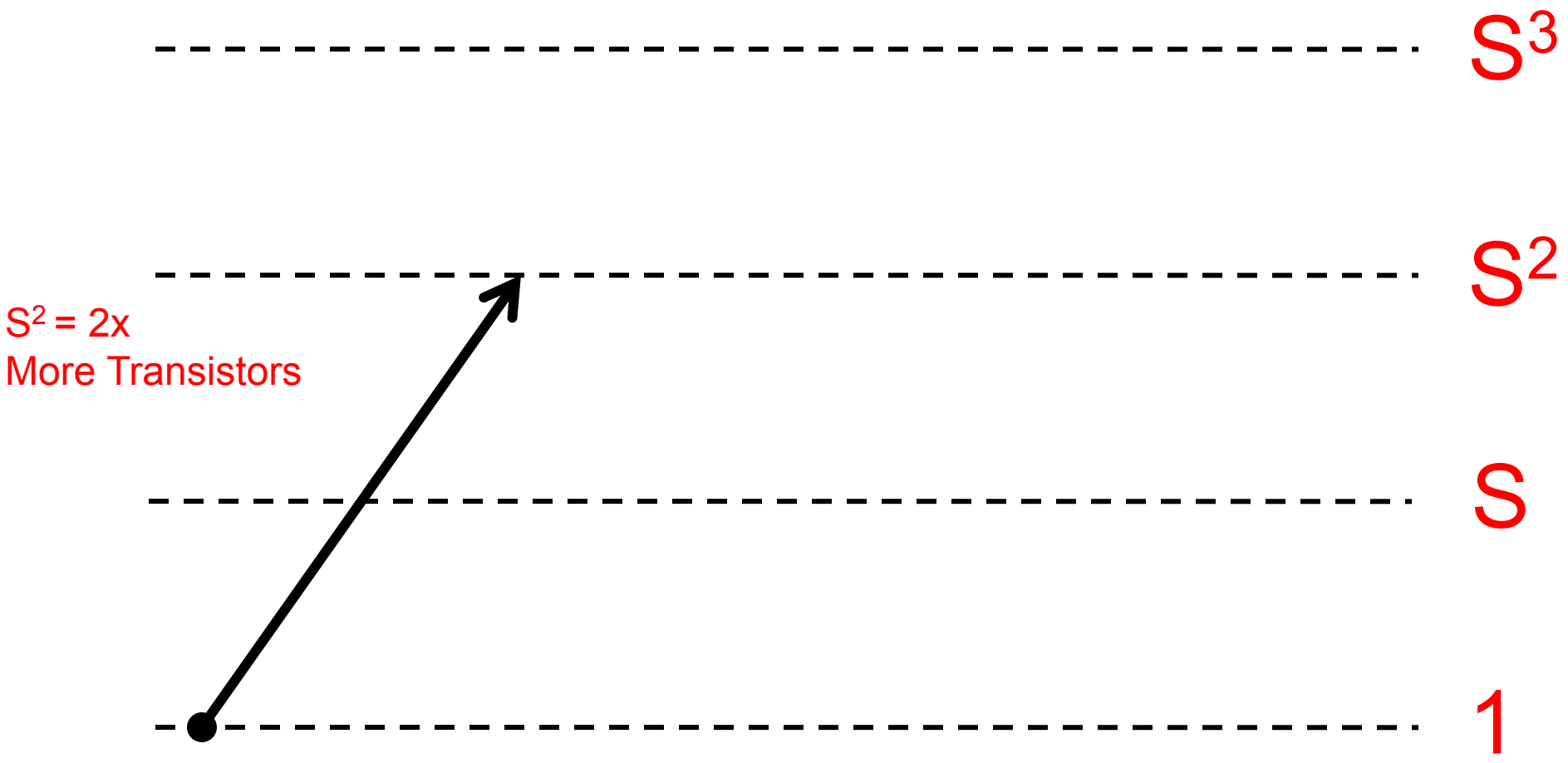


More on Scaling

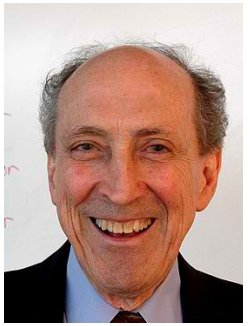
- Seminal paper on scaling is Dennard et. al. “Design of ion-implanted MOSFET's with very small physical dimensions”, 1974
- Lays out how to parameters of transistors should be tweaked to maintain scaling.

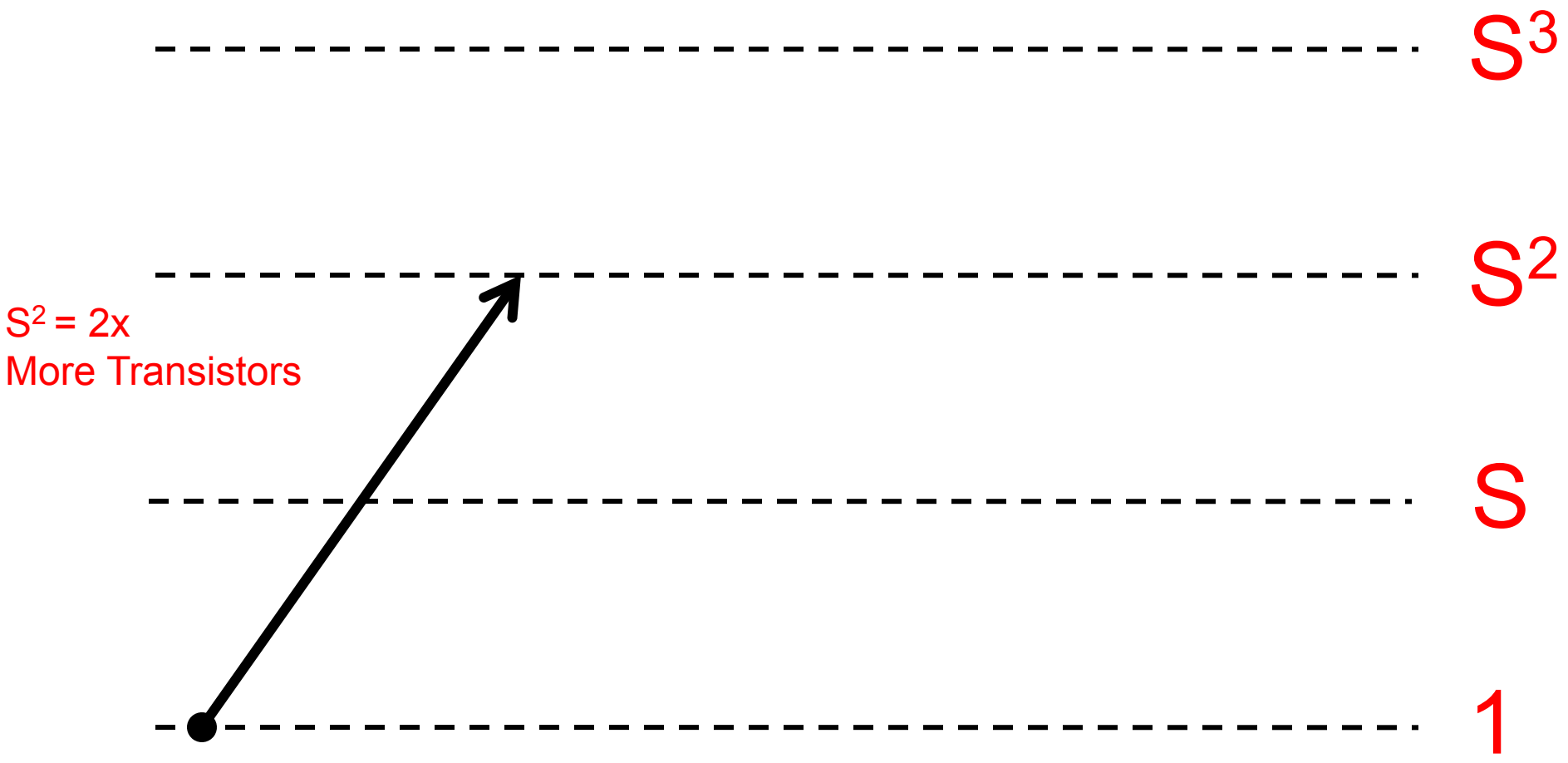
Dennard:
Compute Capabilities with Scaling



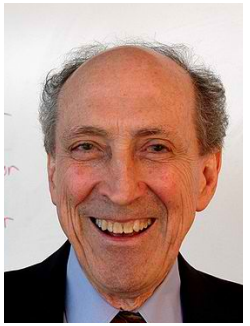


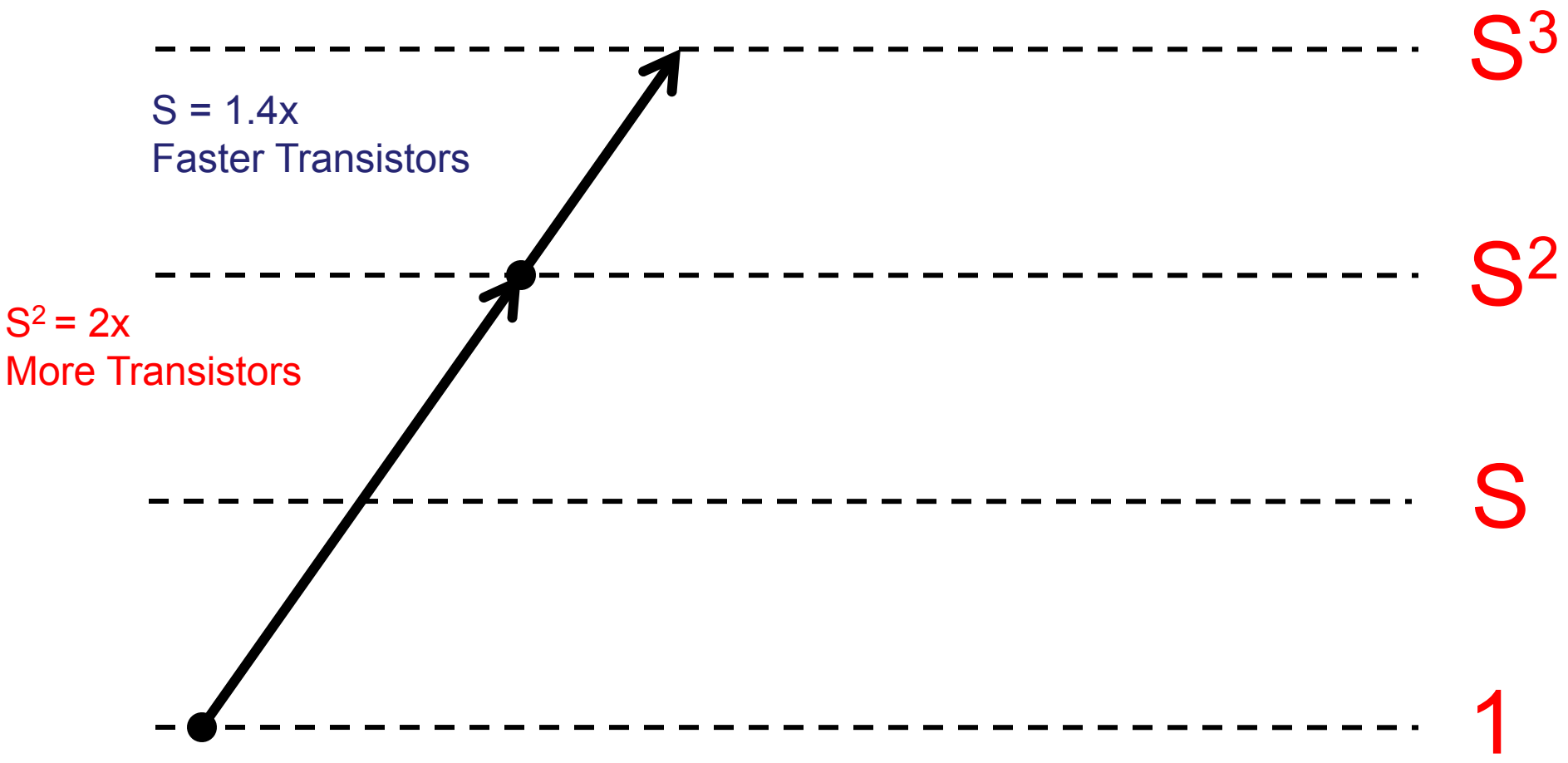
Dennard:
Compute Capabilities with Scaling





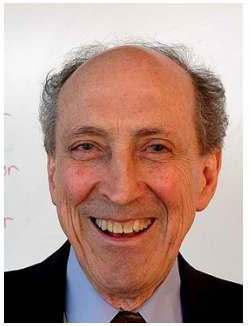
Dennard:
Compute Capabilities with Scaling

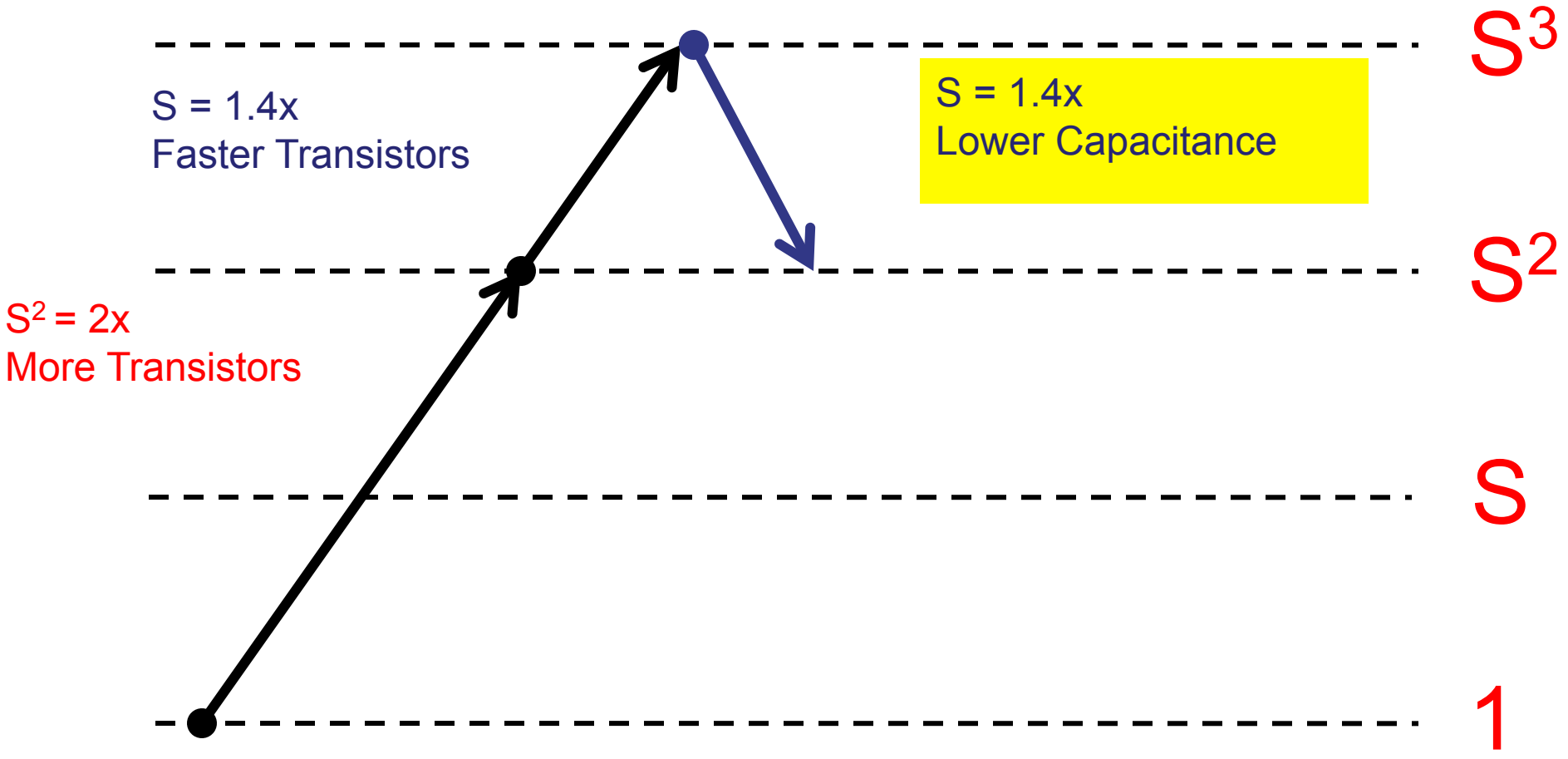




Dennard:

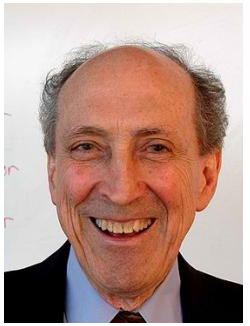
"We can keep power consumption constant"

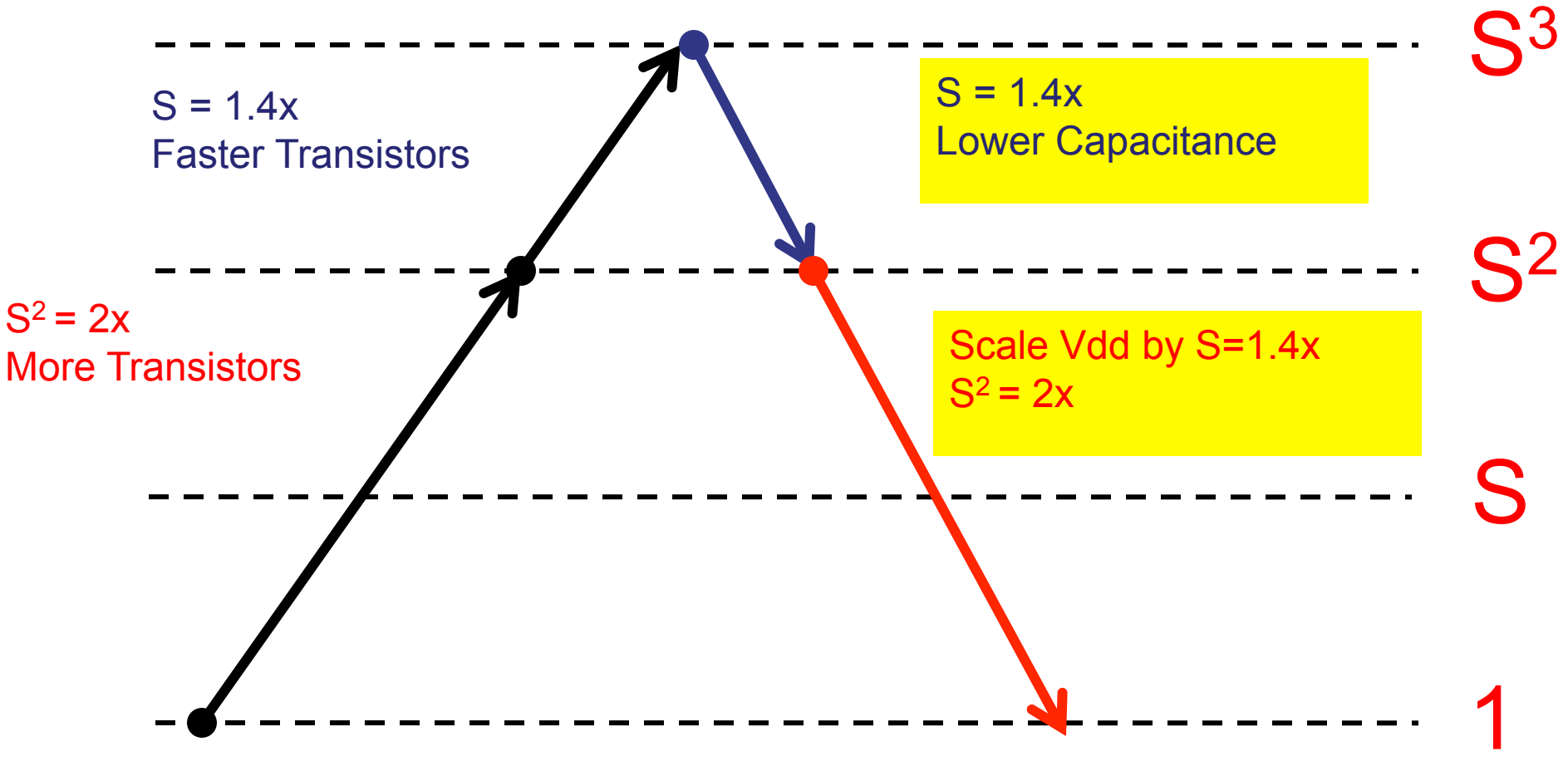




Dennard:

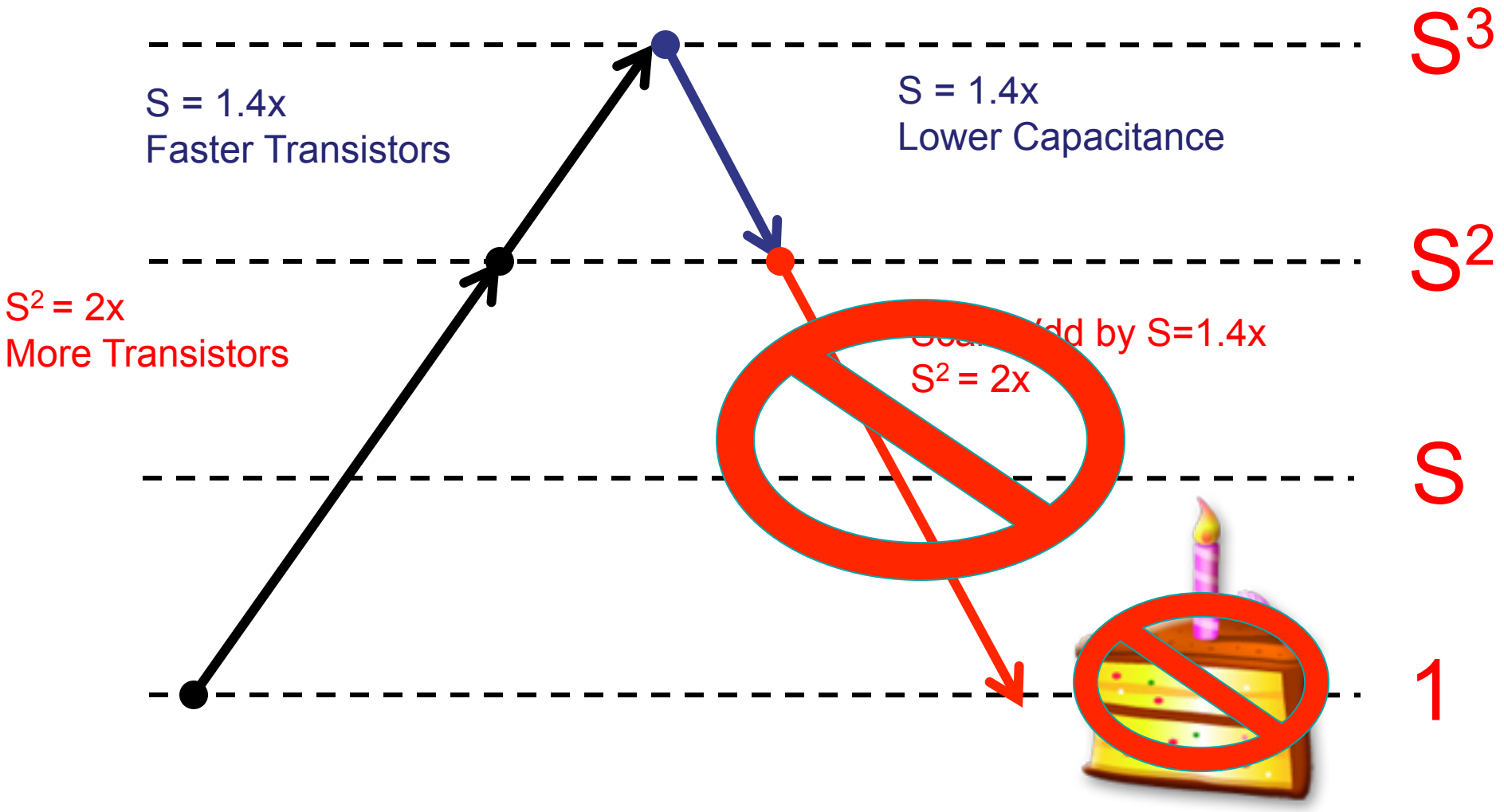
"We can keep power consumption constant"



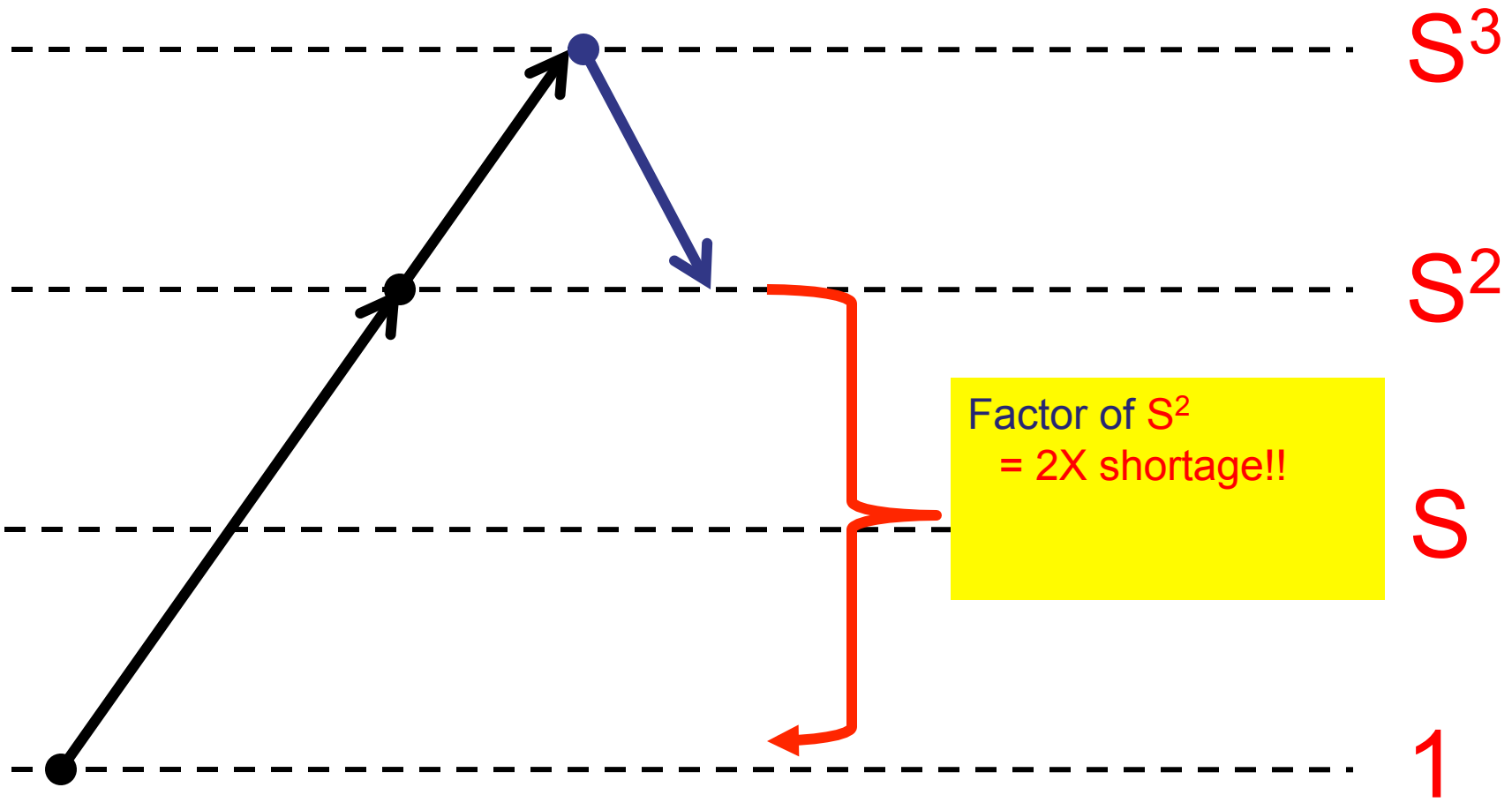


Fast forward to 2005:

Threshold Scaling Problems due to Leakage Prevents Us From Scaling Voltage



Full Chip, Full Frequency Power Dissipation
Is increasing exponentially by 2x with
every process generation



Multicore, thought "the solution to power", also hits the Utilization Wall

Spectrum of tradeoffs between # of cores and frequency

Example:
65 nm → 32 nm (S = 2)

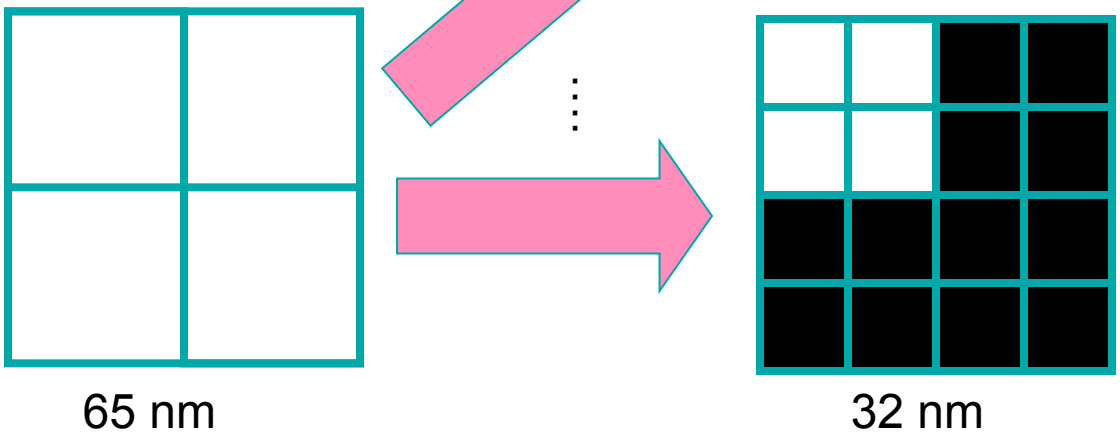
4x4 cores @ .9 GHz

(*GPUs, throughput?)*

2x4 cores @ 1.8 GHz
(8 cores dark, 8 dim)

(*Intel/x86 Choice, next slide*)

4 cores @ 1.8 GHz



65 nm

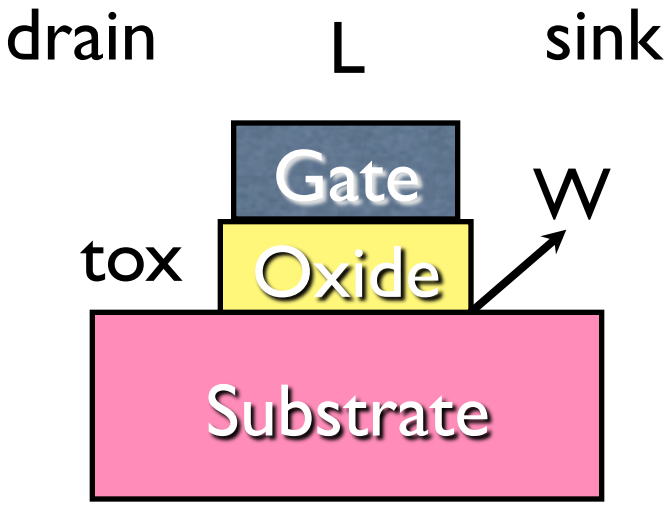
32 nm

4 cores @ 2x1.8 GHz
(12 cores dark)

[Goulding, Hotchips 2010]

Dennardian Scaling

- Given a scaling factor *S*.

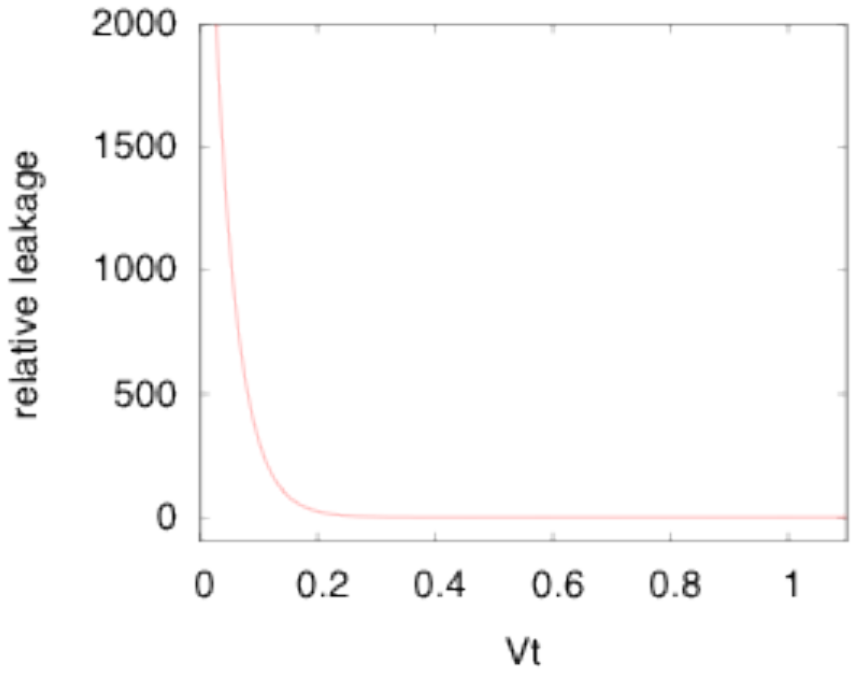


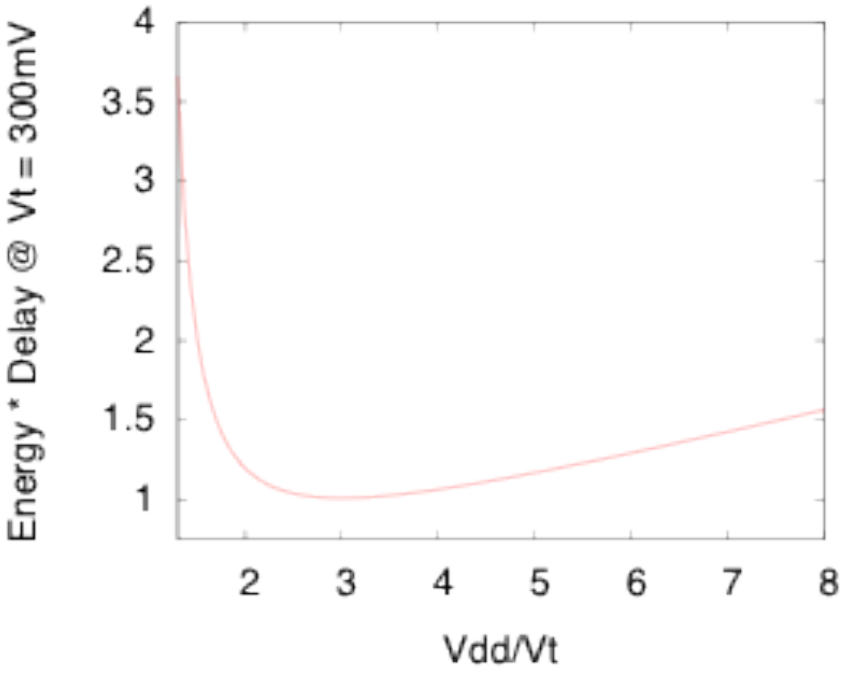
Parameter	Relation	Classical Scaling
power budget		1
chip size		1
V_{dd} (supply voltage)		$1/S$
V_t (threshold voltage)	$1/S$	$1/S$
t_{ox} (oxide thickness)		$1/S$
W, L (transistor dimensions)		$1/S$
C_{gate} (gate capacitance)	WL/t_{ox}	$1/S$
I_{sat} (saturation current)	WV_{dd}/t_{ox}	$1/S$
F (device frequency)	$I_{sat}/(C_{gate}V_{dd})$	S
D (device/area)	$1/(WL)$	S^2
p (device power)	$I_{sat}V_{dd}$	$1/S^2$
P (full die, full frequency power)	$D \times p$	1
U (utilization at fixed power)	$1/P$	1

Per-transistor power scales with density!

Dennardian Breakdown

- The problem with leakage





Post-Dennard Scaling

Parameter	Relation	Classical Scaling	Leakage Limited
power budget		1	1
chip size		1	1
V_{dd} (supply voltage)		$1/S$	1
V_t (threshold voltage)	$1/S$	$1/S$	1
t_{ox} (oxide thickness)		$1/S$	$1/S$
W, L (transistor dimensions)		$1/S$	$1/S$
C_{gate} (gate capacitance)	WL/t_{ox}	$1/S$	$1/S$
I_{sat} (saturation current)	WV_{dd}/t_{ox}	$1/S$	1
F (device frequency)	$I_{sat}/(C_{gate}V_{dd})$	S	S
D (device/area)	$1/(WL)$	S^2	S^2
p (device power)	$I_{sat}V_{dd}$	$1/S^2$	1
P (full die, full frequency power)	$D \times p$	1	S^2
U (utilization at fixed power)	$1/P$	1	$1/S^2$

Per-transistor power is constant!

UCSD Research! [ASPLOS 2010]

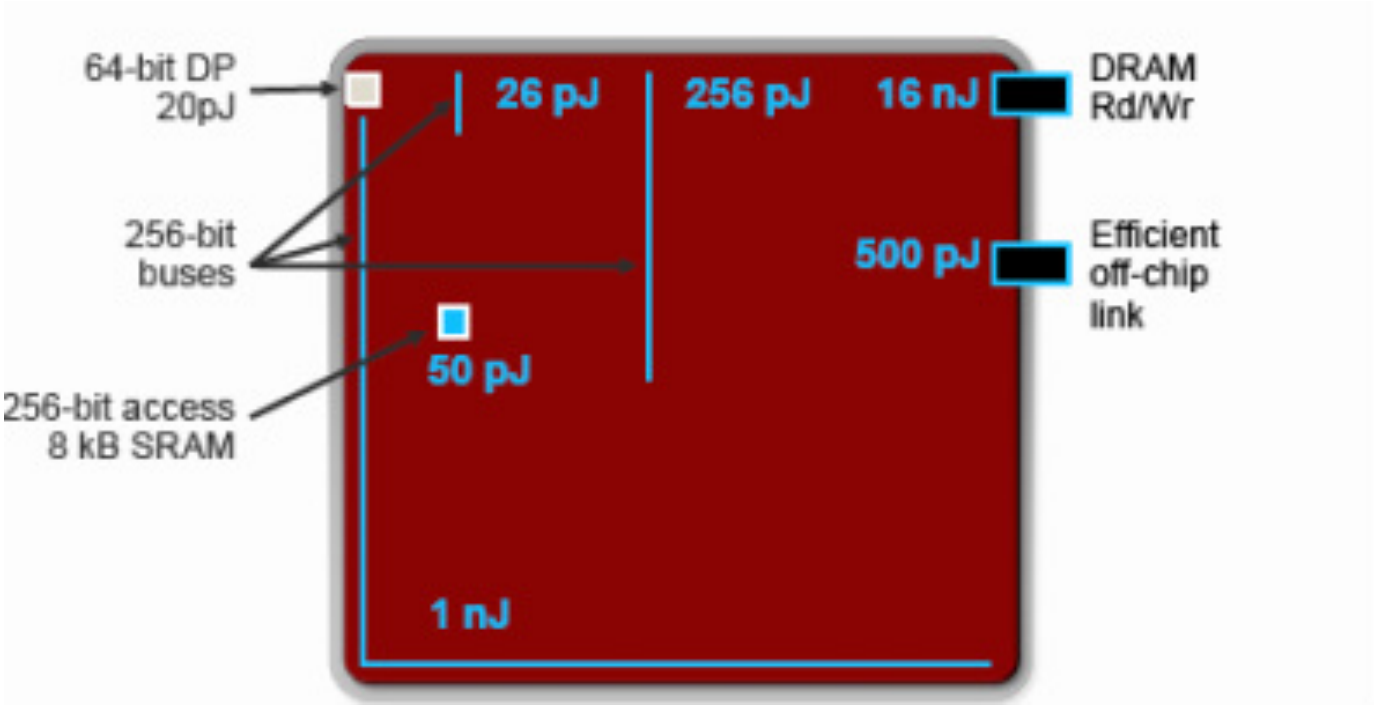
Dennardian Breakdown

Processor numberΔ	L2 cache	L3 cache	Clock speed	Front side bus	System type	Power	Number of cores
45-nm technology							
X7460	9MB	16MB	2.66 GHz	1066 MHz	MP	130W	6
L7455	9MB	12MB	2.13 GHz	1066 MHz	MP	65W	6
L7445	6MB	12MB	2.13 GHz	1066 MHz	MP	50W	4
E7450	9MB	12MB	2.40 GHz	1066 MHz	MP	90W	6
E7440	6MB	16MB	2.40 GHz	1066 MHz	MP	90W	4
E7430	6MB	12MB	2.13 GHz	1066 MHz	MP	90W	4
E7420	6MB	8MB	2.13 GHz	1066 MHz	MP	90W	4
65-nm technology							
X7350	8MB	0	2.93 GHz	1066 MHz	MP	130W	4
L7345	8MB	0	1.86 GHz	1066 MHz	MP	50W	4
E7340	8MB	0	2.40 GHz	1066 MHz	MP	80W	4
E7330	6MB	0	2.40 GHz	1066 MHz	MP	80W	4
E7320	4MB	0	2.13 GHz	1066 MHz	MP	80W	4
E7310	4MB	0	1.60 GHz	1066 MHz	MP	80W	4
E7220	8MB	0	2.93 GHz	1066 MHz	MP	80W	2
E7210	8MB	0	2.40 GHz	1066 MHz	MP	80W	2

Power/Energy Conservation Tricks

- Toggle (or "data") Gating
  - make sure inputs to unused logic do not change
    - e.g. keep FPU inputs steady with flip-flop enable or AND gate, if you are not using the FPU
- Clock Gating
  - if you have a bunch of logic that does not change, then prevent the clock from toggling to save clock power
- Frequency Scaling
  - Reduce clock freq. $P = CV^2 F$ *Reduces power but not energy.*
- Dynamic Voltage and Frequency Scaling (DVFS)
  - Reduces energy and power together $E=CV^2$; $P = CV^2 F$; $\max(F) \sim V$; ($V>2.5V_t$)
  - Can reduce leakage too.
  - But: RAMS have a minimum voltage (.5-.7 V)
- Power Gating
  - Turn off the power supply to region of chip (e.g. an entire core)
  - It can no longer leak. All state is lost unless you use retention flops.
  - Power gates are very, large so they can provide low resistance to power supply
  - Need to *stage* turn on so current inrush noise doesn't disturb neighbor circuits. > 300 ns

Energy of communication v. computation



Final Thoughts on Moore's Law

- Moore's Law is a *conspiracy*

  Webster conspiracy:

  *- **2 :** to act in harmony toward a common end*

The chip (semiconductor) industry consists of many players – equipment manufacturers (e.g. lithography, mask making equipment), chip makers, computer aided design (CAD) companies, and end-sellers. It more or less runs in lock step. No one company can go too far ahead in process generations without the others.

In fact – they all plan together what to shoot for according to a schedule over the next 15 years!

International Technology Roadmap for Semiconductors (ITRS)

Future Target

Yellow: some research will get us there

Red: we have no idea

Year of Production	2005	2006	2007	2008	2009	2010	2011	2012	2013
DRAM ½ Pitch (nm) (contacted)	80	70	65	57	50	45	40	36	32
MPU/ASIC Metal 1 (M1) ½ Pitch (nm)(contacted)	90	78	68	59	52	45	40	36	32
MPU Physical Gate Length (nm)	32	28	25	22	20	18	16	14	13
R_{sd}: Effective Parasitic series source/drain resistance [12]									
Planar Bulk (Ω-µm)	180	170	140	140	120	105	80	70	
UTB FD (Ω-µm)				155	140	125	110	90	75
DG (Ω-µm)							110	100	90
$C_{g,ideal}$: Ideal NMOS Device Gate Capacitance [13]									
Extended Planar Bulk (F/µm)	5.73E-16	5.25E-16	4.69E-16	6.37E-16	6.72E-16	6.78E-16	7.39E-16	6.41E-16	
UTB FD (F/µm)				5.84E-16	5.75E-16	5.65E-16	5.52E-16	5.37E-16	4.98E-16
DG (F/µm)							4.60E-16	4.39E-16	4.48E-16
$C_{g,total}$: Total gate capacitance for calculation of CV/I [14]									
Extended Planar Bulk (F/µm)	8.13E-16	7.65E-16	6.99E-16	8.47E-16	8.42E-16	8.28E-16	8.59E-16	7.51E-16	
UTB FD (F/µm)				8.04E-16	7.55E-16	7.35E-16	6.92E-16	6.67E-16	6.18E-16
DG (F/µm)							6.50E-16	6.29E-16	6.28E-16
$\tau = CV/I$: NMOSFET intrinsic delay (ps) [15]	0.870	0.740	0.640	0.540	0.460	0.400	0.340	0.290	0.250
$1/\tau$: NMOSFET intrinsic switching speed (GHz) [16]	1149	1351	1563	1852	2174	2500	2941	3448	4000

Current

× ~13.5 = 1 “FO4” (gate delay) = 81 GHz

http://www.itrs.net/Links/2009ITRS/Home2009.htm

Est. Technology Params: Memories and Caches

- Cacti Tool can be very useful
  - enter in dimensions, technology node
  - out comes energy, delay, area
- http://quid.hpl.hp.com:9081/cacti/