Introduction to VLSI

Jo Ebergen

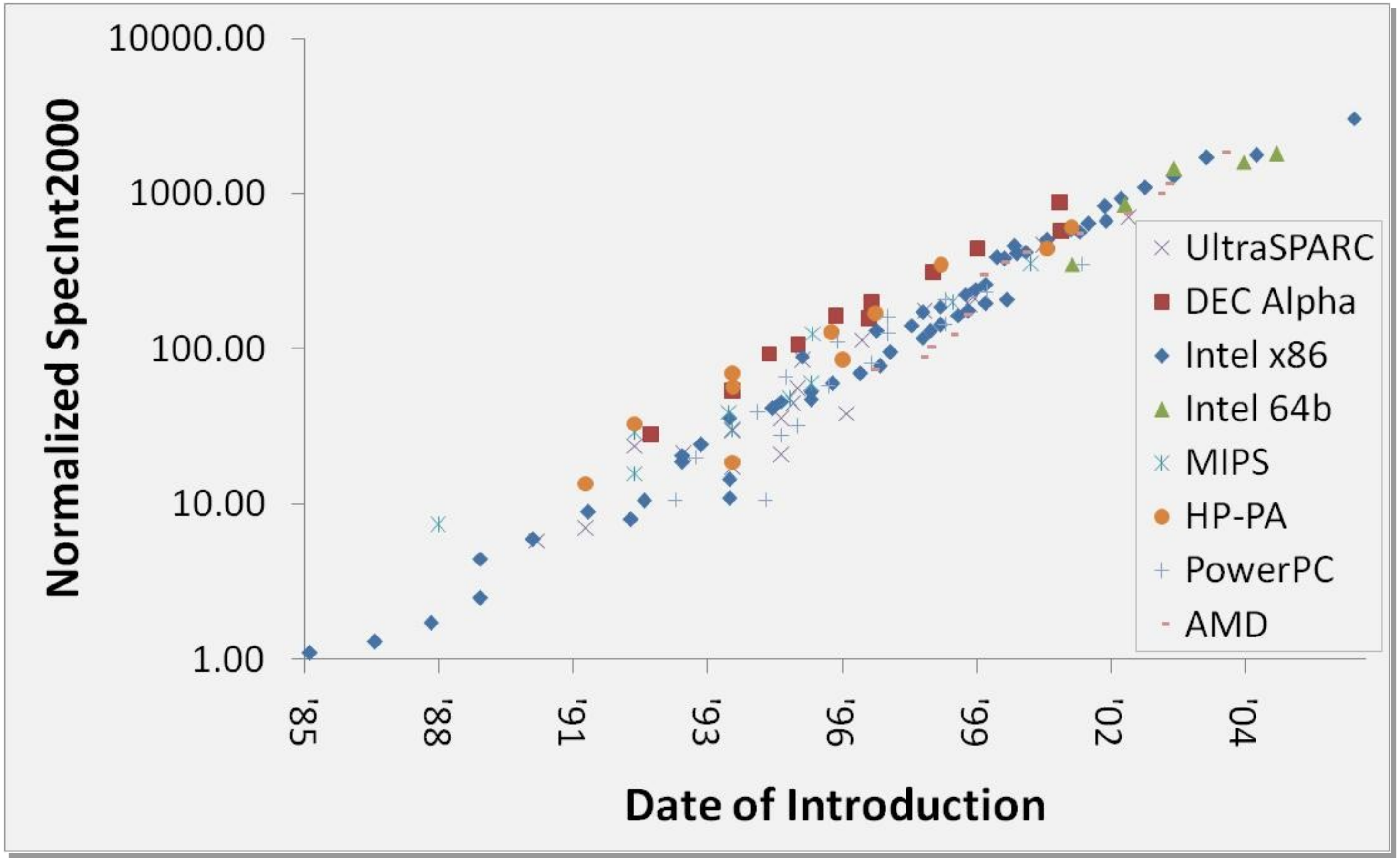
Sun Microsystems Laboratory

jo.ebergen@sun.com

Acknowledgements

- Slides used with permission from
  - Ron Ho (Sun Microsystems)
  - Don Stark (Stanford Univ)
  - David Harris (Harvey Mudd College)
  - Richard Klein (AMD)
- More lecture material available at
  - http://eeclass.stanford.edu/ee371/
  - http://www.cmosvlsi.com/coursematerials.html
  - “CMOS VLSI Design” by Neil Weste and David Harris

VLSI trends – clock rates scaling up

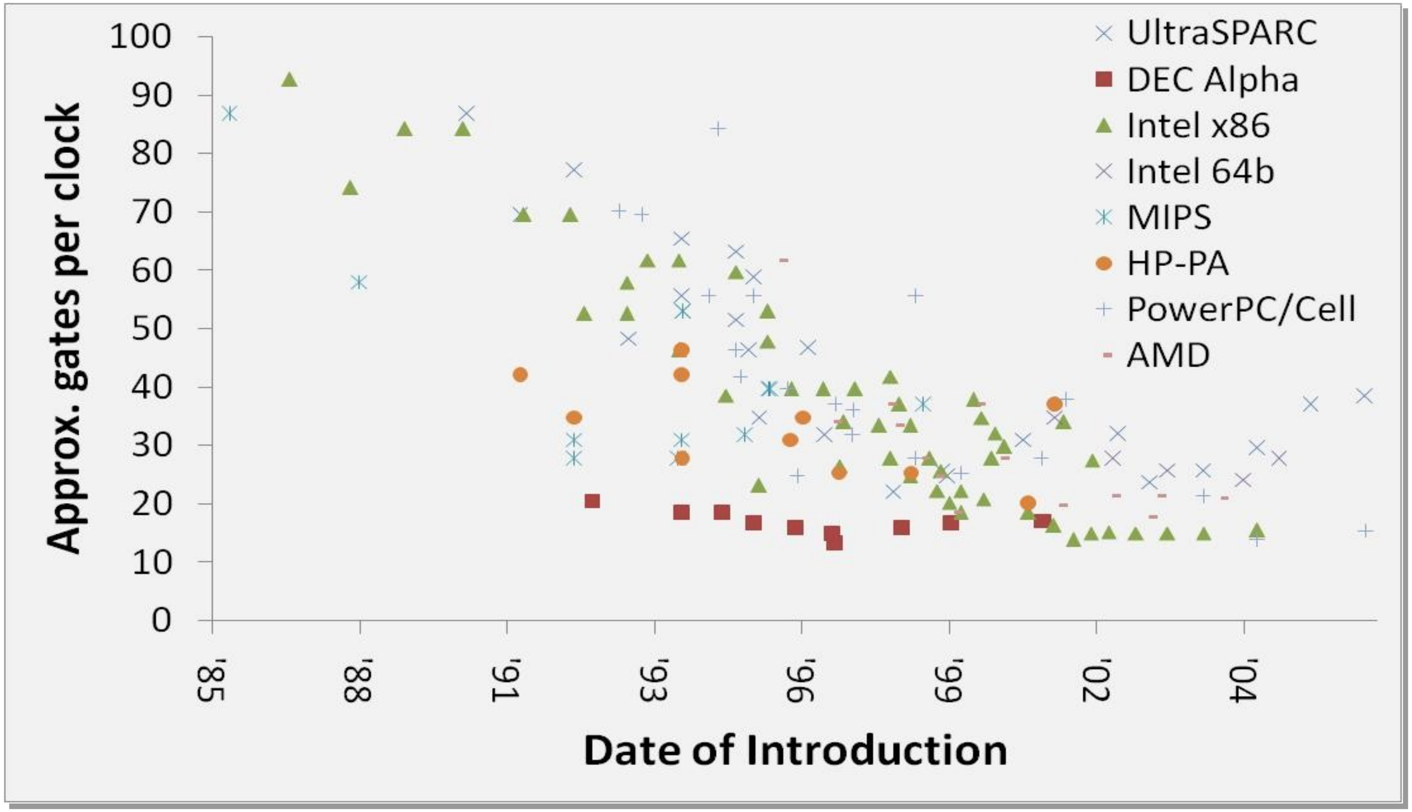


Some data from Horowitz, IEDM 2005

- Not just due to faster technologies

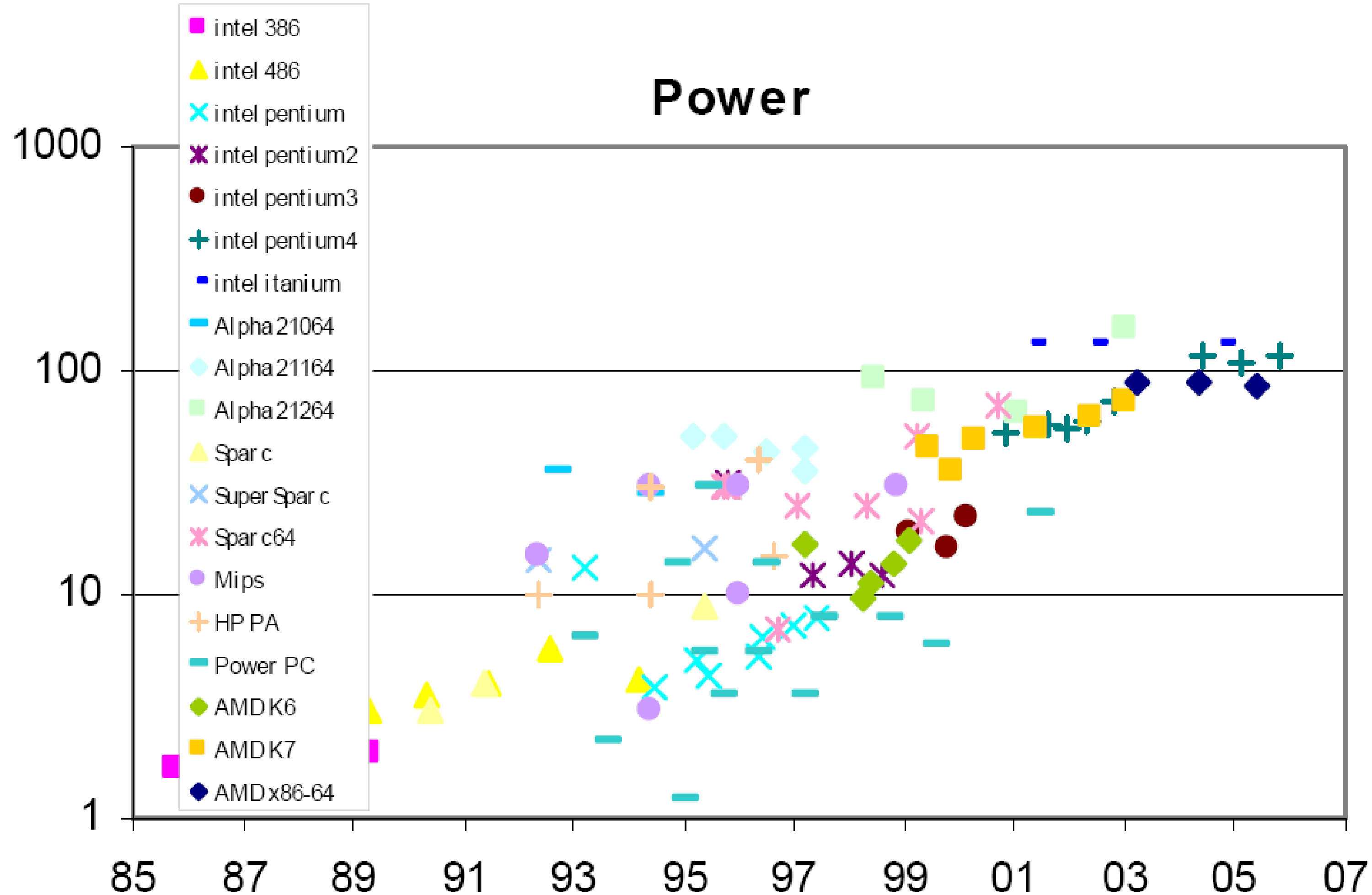
VLSI trends – cycle times in FO4 falling

- In terms of FO4 (fanout-of-four inverter delay) – process independent



Some data from Horowitz, IEDM 2005

VLSI trends – power density

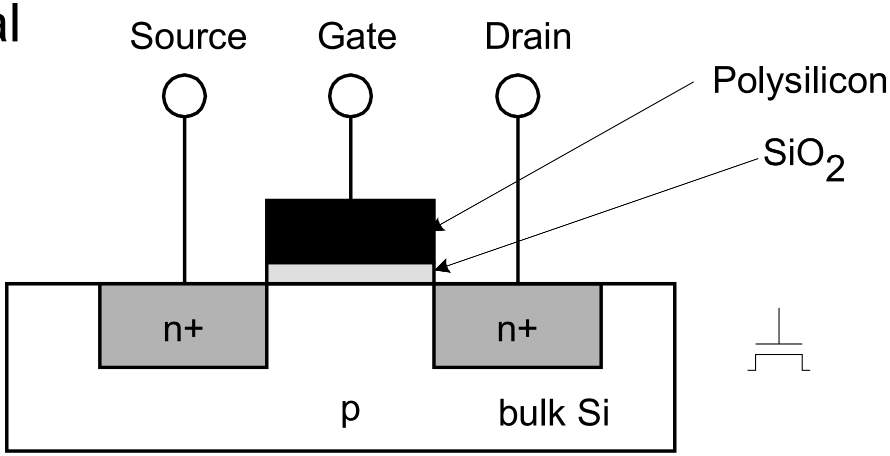


Design objectives have changed

- Prior to 2000 most designs focused on performance
- Today most designs are power limited
  - Chips could run faster, if we were willing to pay power/thermal costs
  - The problem is getting worse moving forward (Vdd is not scaling)
- Additional challenges
  - Soft errors
  - Increasing variability in many parameters affecting performance, power, and yield

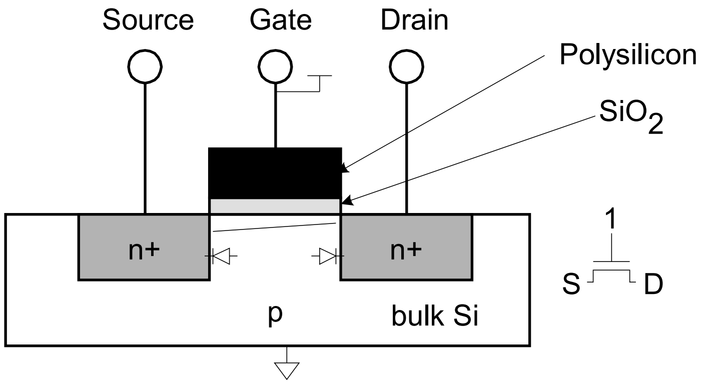
nMOS Transistor

- Four terminals: gate, source, drain, body
- Gate – oxide – body stack looks like a capacitor
  - Gate and body are conductors
  - SiO_2 (oxide) is a very good insulator
  - Called metal – oxide – semiconductor (MOS) capacitor
  - Even though gate is no longer made of metal



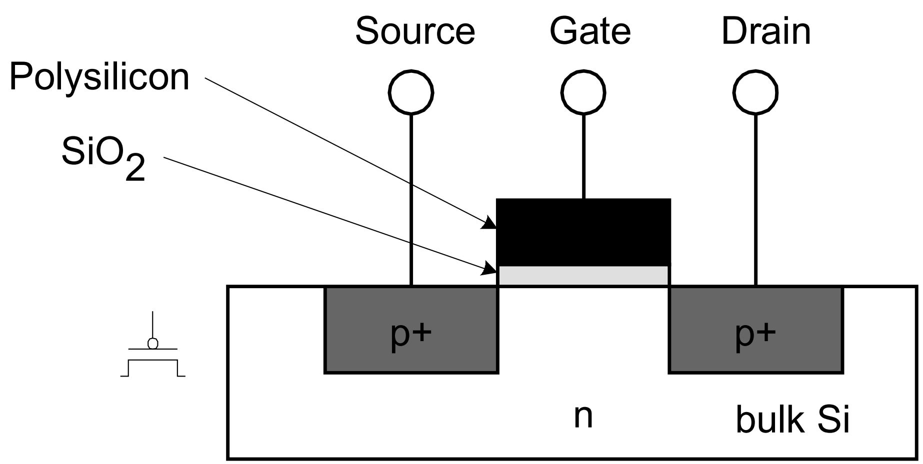
nMOS Operation Cont.

- When the gate is at a high voltage:
  - Positive charge on gate of MOS capacitor
  - Negative charge attracted to body
  - Inverts a channel under gate to n-type
  - Now current can flow through n-type silicon from source through channel to drain, transistor is ON



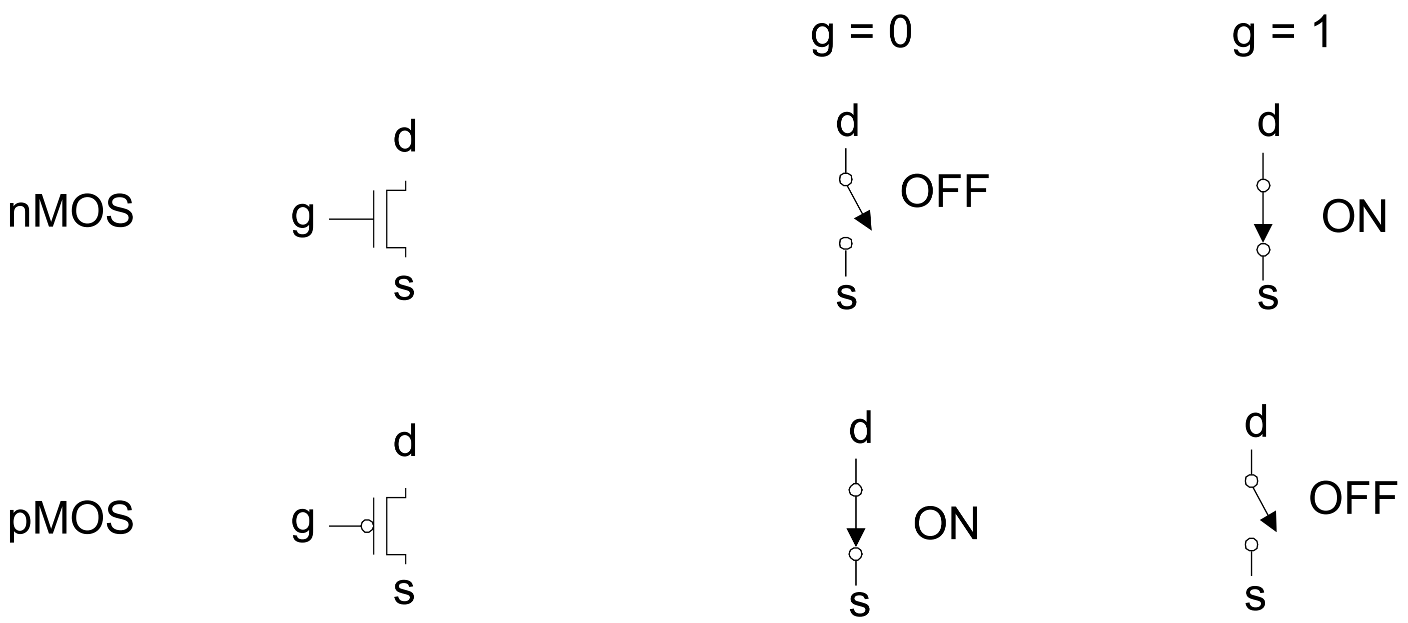
pMOS Transistor

- Similar, but doping and voltages reversed
  - Body tied to high voltage (V_{DD})
  - Gate low: transistor ON
  - Gate high: transistor OFF
  - Bubble indicates inverted behavior



Transistors as Switches

- We can view MOS transistors as electrically controlled switches
- Voltage at gate controls path from source to drain

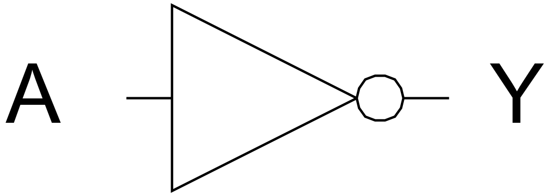


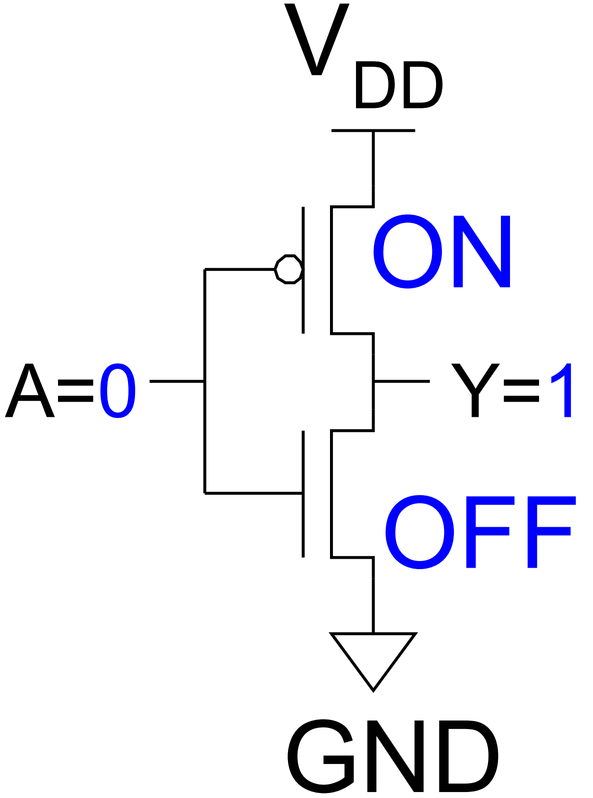
Power Supply Voltage

- GND = 0 V
- In 1980's, V_{DD} = 5V
- V_{DD} has decreased in modern processes
  - High V_{DD} would damage modern tiny transistors
  - Lower V_{DD} saves power
- V_{DD} = 3.3, 2.5, 1.8, 1.5, 1.2, 1.0, …

CMOS Inverter

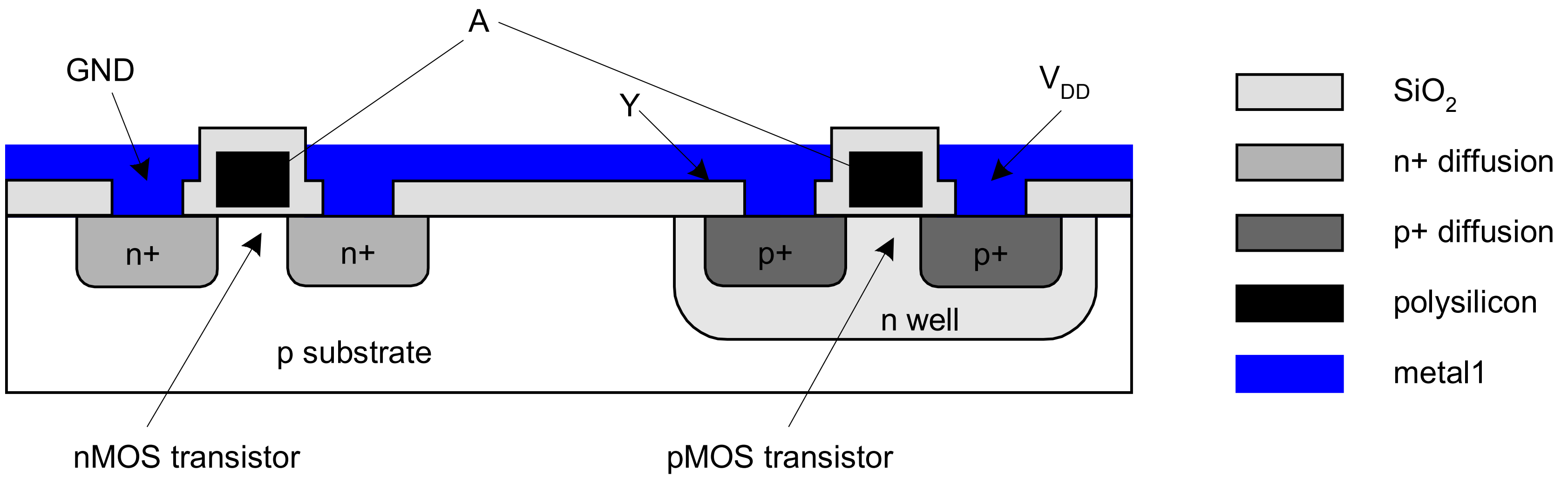
A	Y
0	1
1	0





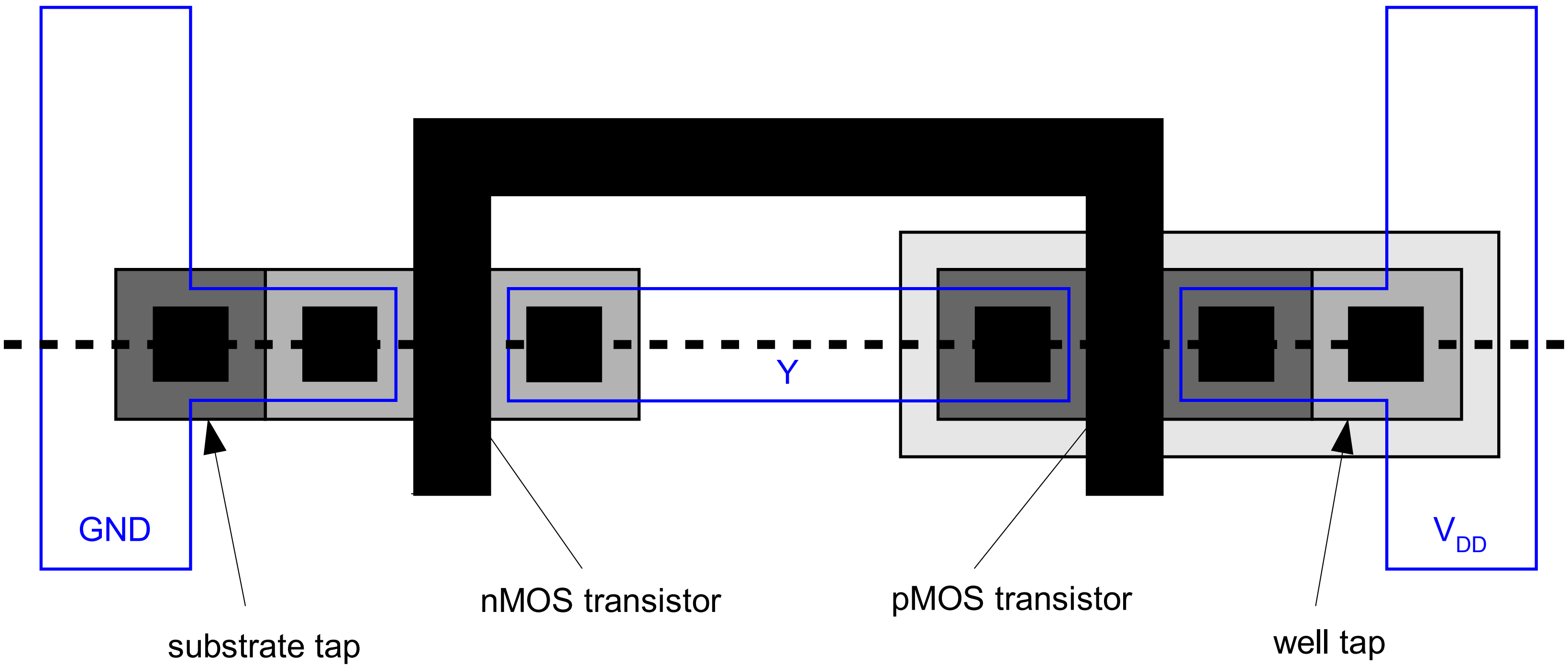
Inverter Cross-section

- Typically use p-type substrate for nMOS transistors
- Requires n-well for body of pMOS transistors



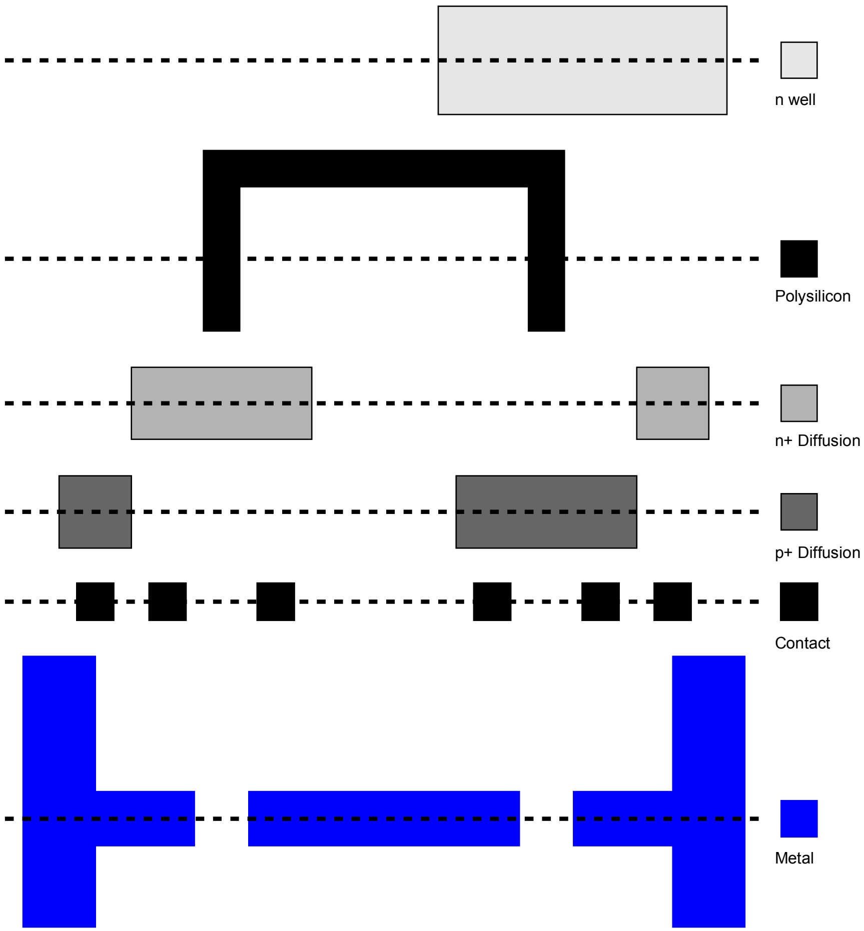
Inverter Mask Set

- Transistors and wires are defined by *masks*
- Cross-section taken along dashed line



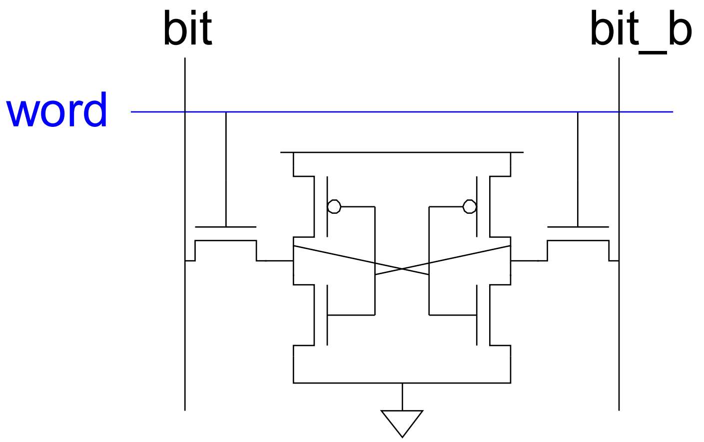
Detailed Mask Views

- Six masks
  - n-well
  - Polysilicon
  - n+ diffusion
  - p+ diffusion
  - Contact
  - Metal



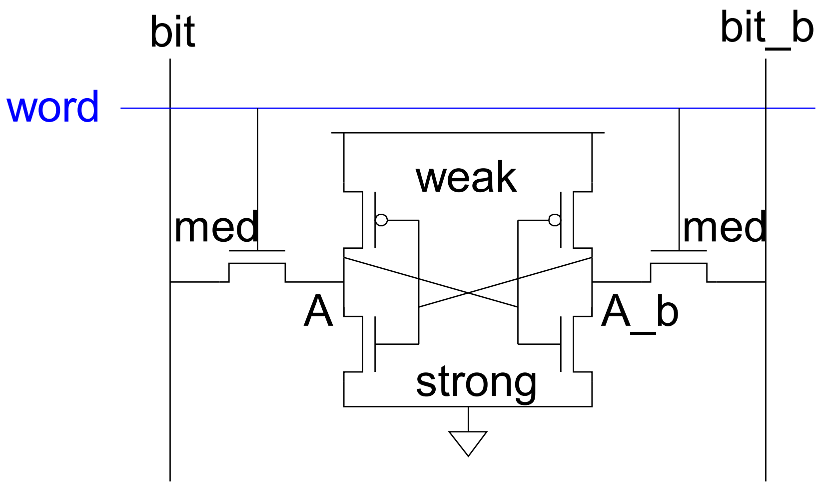
A Memory Cell: 6T SRAM Cell

- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read:
  - Precharge bit, bit_b
  - Raise wordline
- Write:
  - Drive data onto bit, bit_b
  - Raise wordline



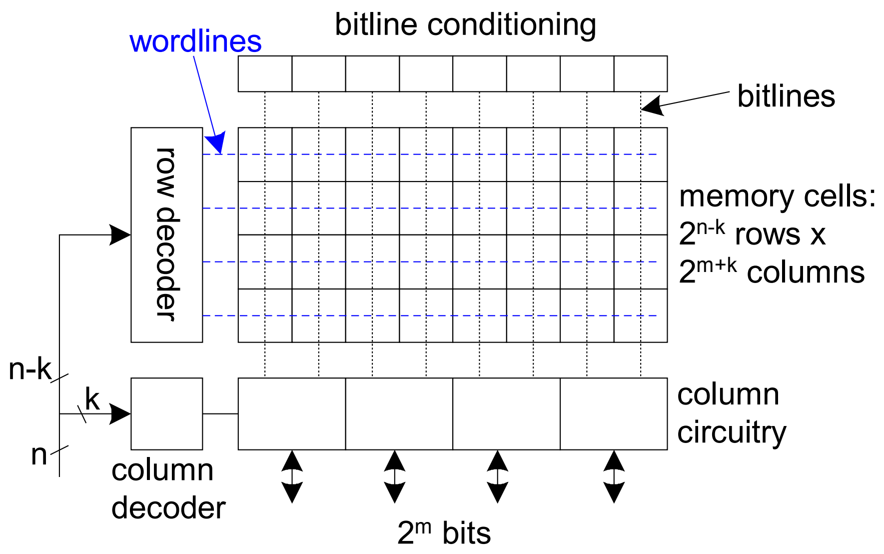
SRAM Sizing

- High bitlines must not overpower inverters during reads
- But low bitlines must write new value into cell



A Memory Array Architecture

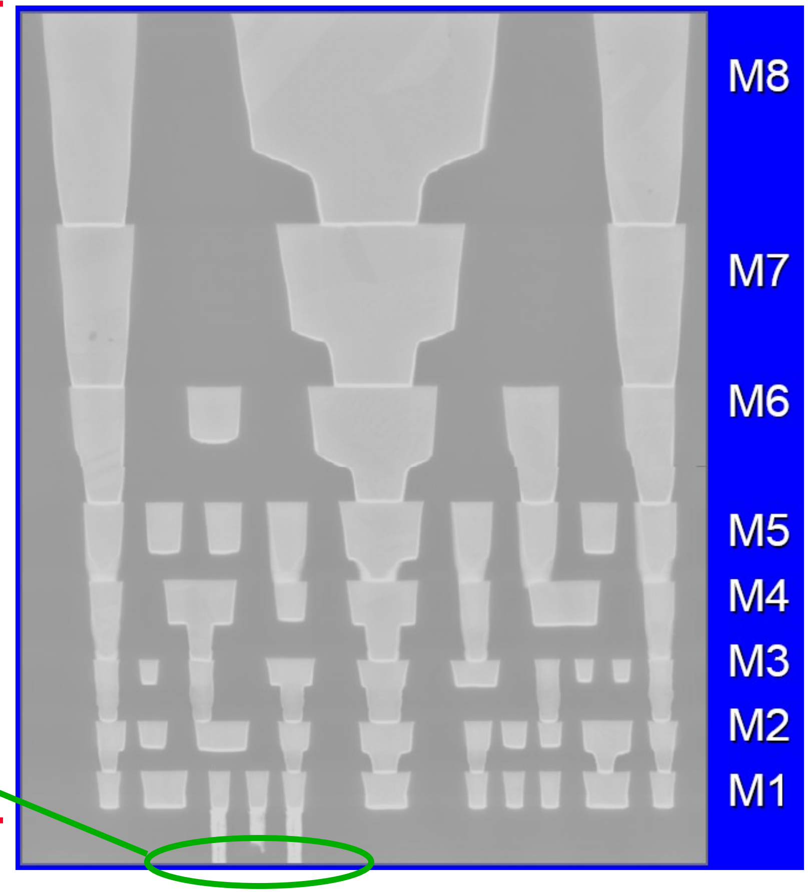
- 2^n *words* of 2^m *bits* each
- If n >> m, fold by 2^k into fewer *rows* of more *columns*



- Good regularity – easy to design
- Very high density if good cells are used

A modern technology is mostly wires

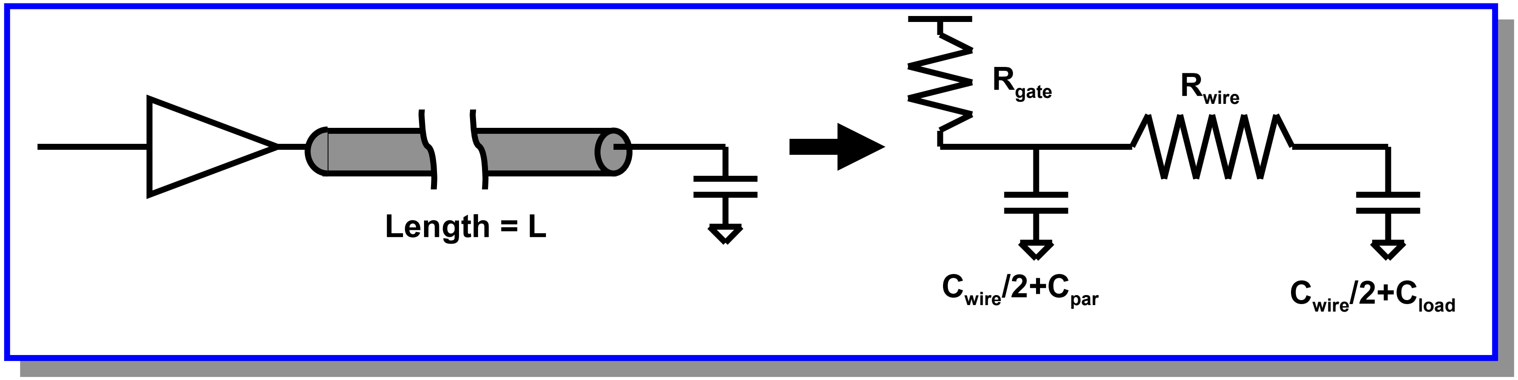
- Cross-section, Intel's 65nm tech
  - 8 metal layers
  - Low-κ_r dielectrics ($\kappa_r = 2.7$)
  - Wires are 2x taller than wide
- Wires are here
- Transistors are here



P. Bai, *et al.*, "A 65nm logic technology featuring 35nm gate lengths, enhanced channel strain, 8 Cu interconnect…," *IEDM* 2004.

Wire performance: latency

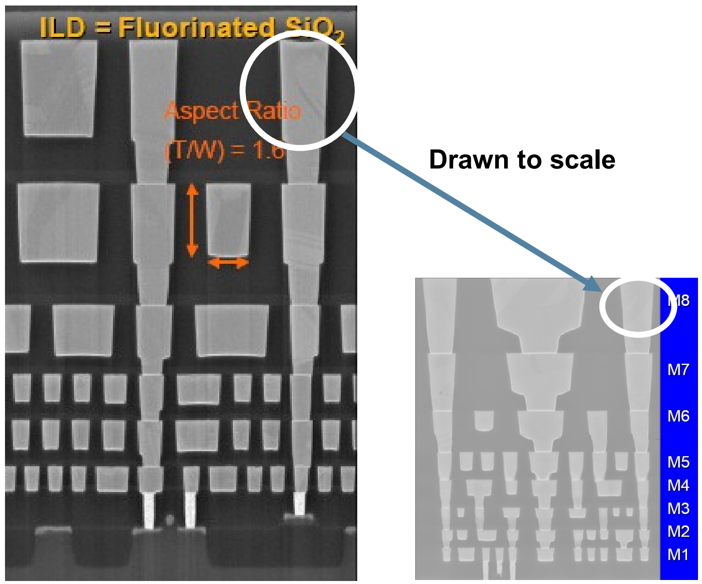
- Delay of gate driving a long wire governed by RC time constants



  - Elmore delay: $D = R_{gate}(C_{wire} + C_{par} + C_{load}) + R_{wire}(0.5C_{wire} + C_{load})$
  - For long wires: $D = k \cdot FO4 + 0.5R_{wire}C_{wire}$
  - $R_{wire}C_{wire}$ is quadratic in wire length L
- For long wires, this delay quickly becomes untenable
  - In a 65nm process, wire delay looks like 2-3 FO4/mm²

Wire scaling basics

- Shrinking wire dimensions changes characteristics
  - R_{wire} depends on cross-sectional areas: grows quickly
  - C_{wire} and L_{wire} depend on the ratio of dimensions: stays constant



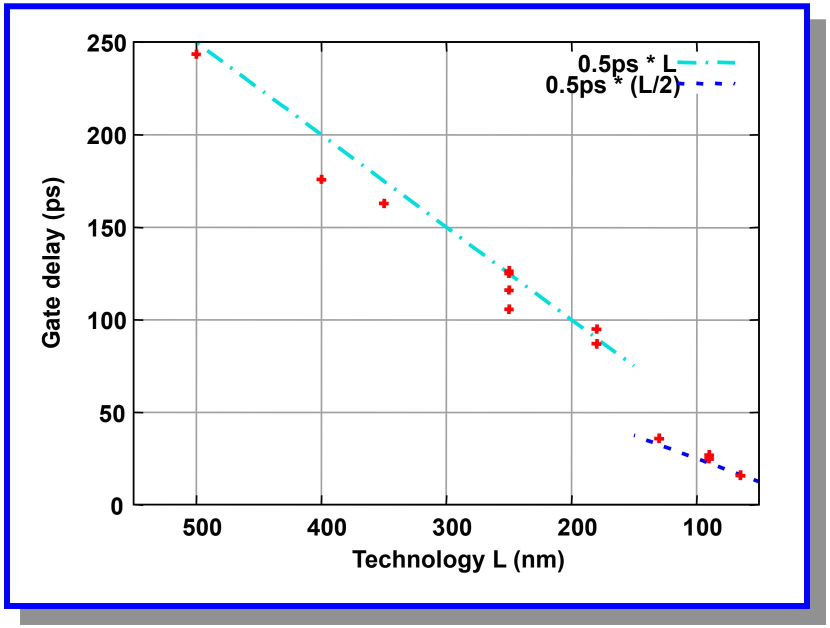
Intel's 130nm process　　Intel's 65nm process

S. Thompson, "An enhanced 130nm generation logic technology featuring 60nm transistors…", *IEDM* 2001.
P. Bai, *et al.*, "A 65nm logic technology featuring 35nm gate lengths, enhanced channel strain, 8 Cu interconnect…," *IEDM* 2004.

Gate delay scaling

- Wire scaling matters only if slower (different) than gate scaling

- Use FO4 as our gate delay metric
  - Delay (in ps) = 0.5 * L_{phys} (in nm)

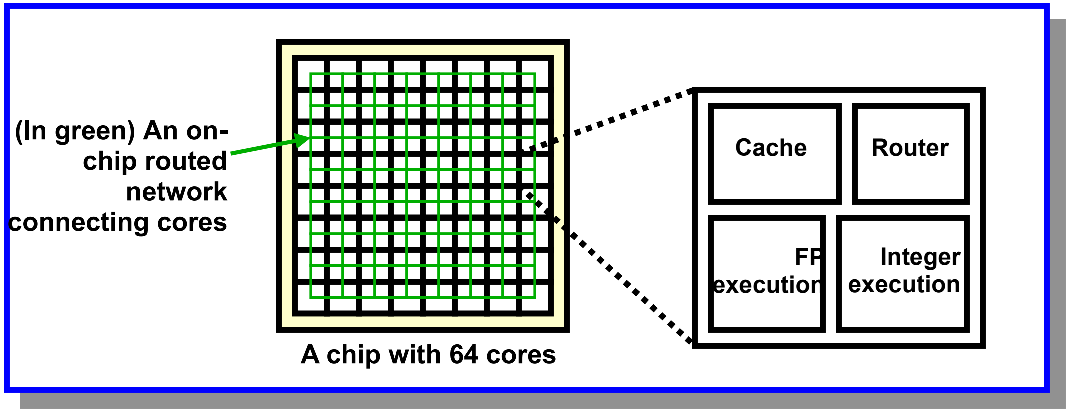


- Poly undercuts change L_{phys}
  - $L_{phys} = L_{drawn}$ up until 180nm
  - $L_{phys} = 0.5 L_{drawn}$ after 130nm

- Wire delays scale worse than gate delays!

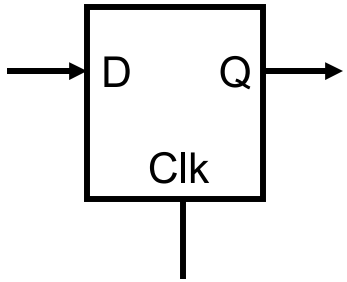
Scaling implications for architecture

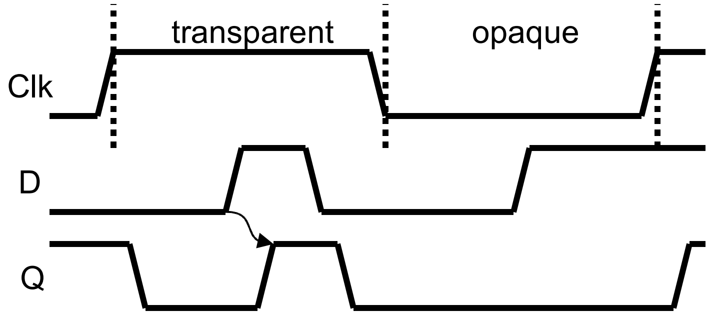
- Look towards architectures that can handle the duality of wires?
  - Slow global communication does not throttle performance
  - Performance is set by fast local (scaled-length) wires
- Answer: modular architectures using multiple tiled cores
  - Cores are built with local wires that are fast and that scale well
  - Global core-to-core communication is explicitly slow



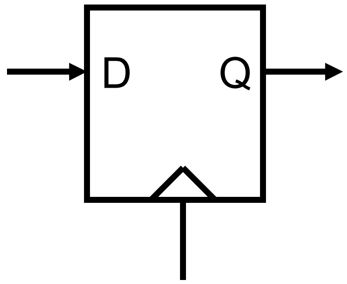
Latches and Flops

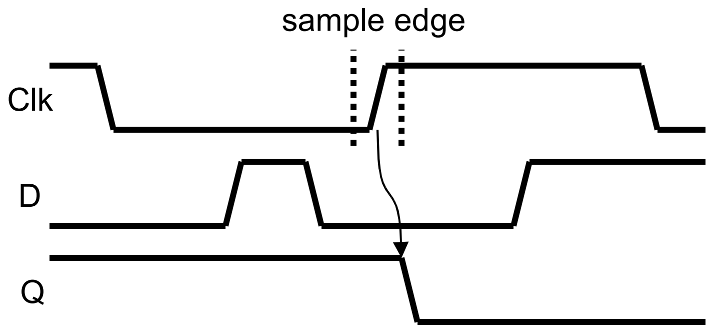
- Latch has soft timing: transparent when clock=1





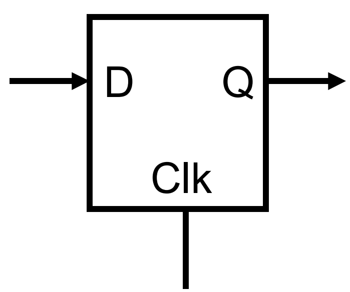
- Flop has hard timing: “transparent” when clock 0→1

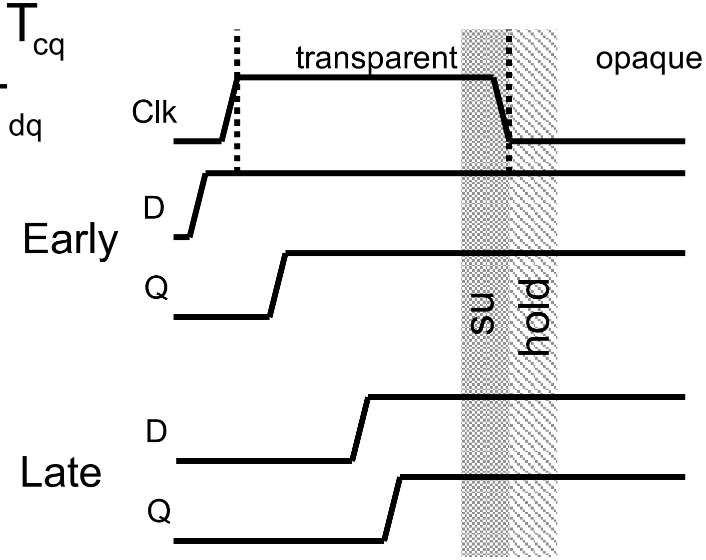




Latch Timing

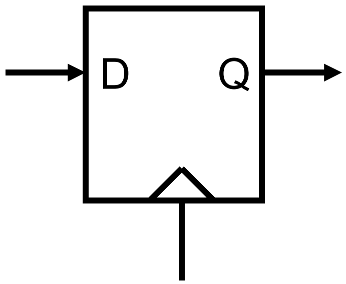
- Setup and hold times are defined relative to the clock **fall**
  - Setup time: how long before the clock fall must the data arrive
  - Hold time: how long after the clock fall must the data not change
- Delay depends on arrival time of data relative to clock **rise**
  - On early data arrival, delay = T_{cq}
  - On late data arrival, delay = T_{dq}

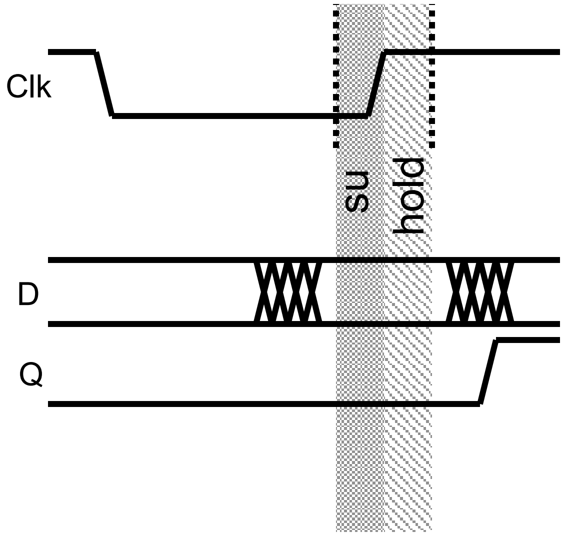




Flop Timing

- Setup and hold times are defined relative to the clock **rise**
  - Setup time: how long before the clock rise must the data arrive
  - Hold time: how long after the clock rise must the data not change
- Delay is always T_{cq}, as long as data hits the setup constraint

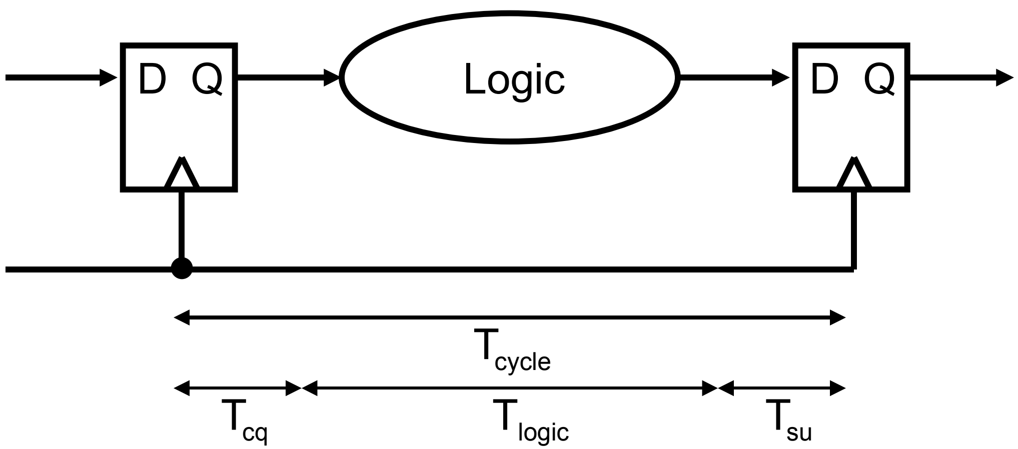




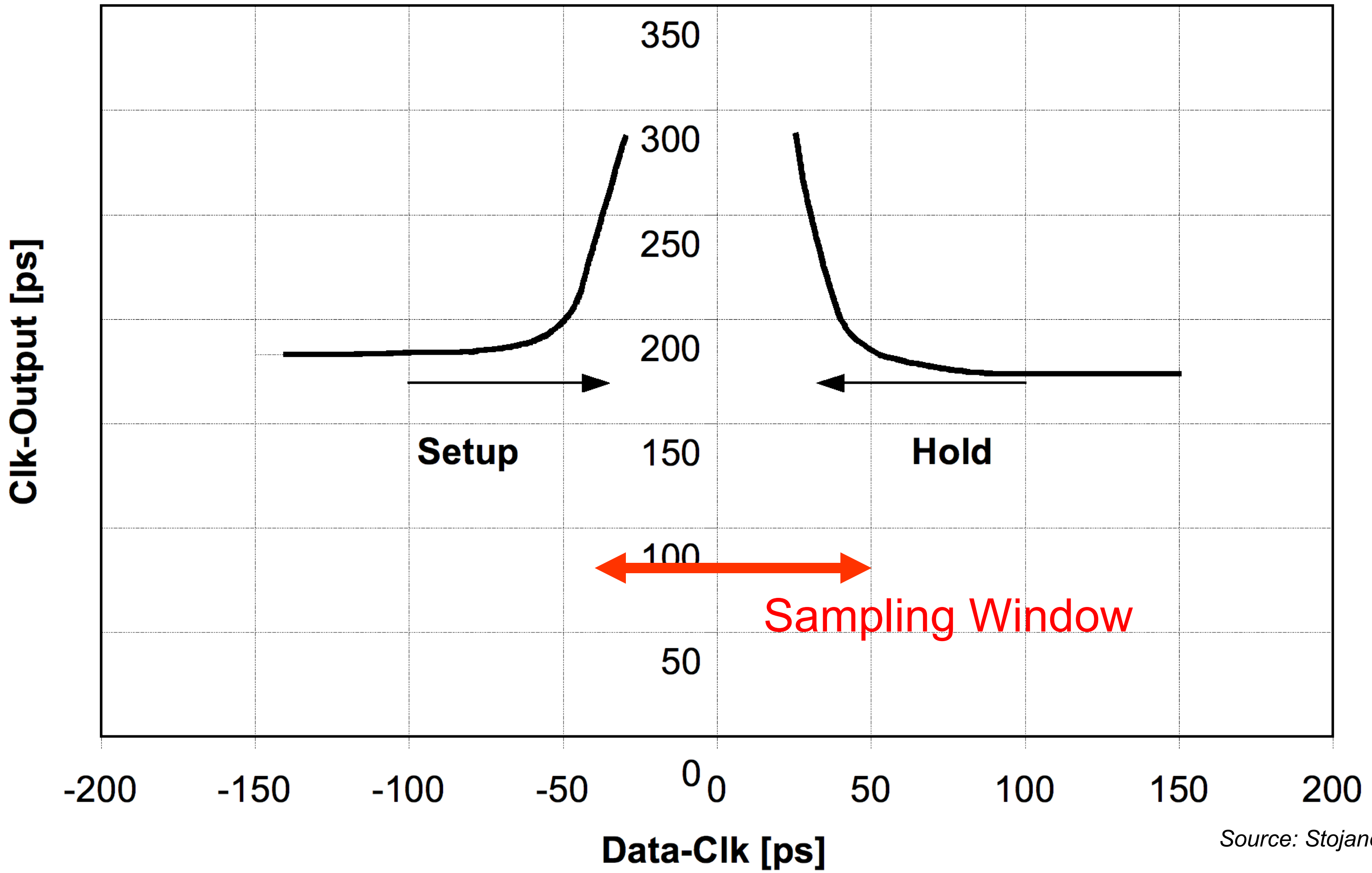
Flop Delay

- What's the delay through a flop?
  - Data must get to the flop by a setup time T_{su}
  - Data must traverse the flop and take up T_{cq}

- So the time available to do logic is what's left
  - $T_{logic} = T_{cycle} – T_{su} – T_{cq} – T_{skew}$



Flop Clk-to-Q Delay Depends On Data-to-Clk



Source: Stojanovic

We Need Faster Clocked Elements

- Today: 16 FO4/cycle in Pentium4; 12 FO4/cycle in Sony Cell
  - If a flop has an overhead of 3FO4, this is 15%-30% overhead
  - We can fill less of the cycle time with “work”

- Making a faster flop helps performance significantly

We Need Lower Power Clocked Elements

- Power scales as $C*V^2f$ and V is not scaling anymore

- Part of this power is for clock distribution
  - Clocking requires 70% of core power in Power4 (2001)
  - Clocking requires 25% of core power in dual-core Itanium (2005)

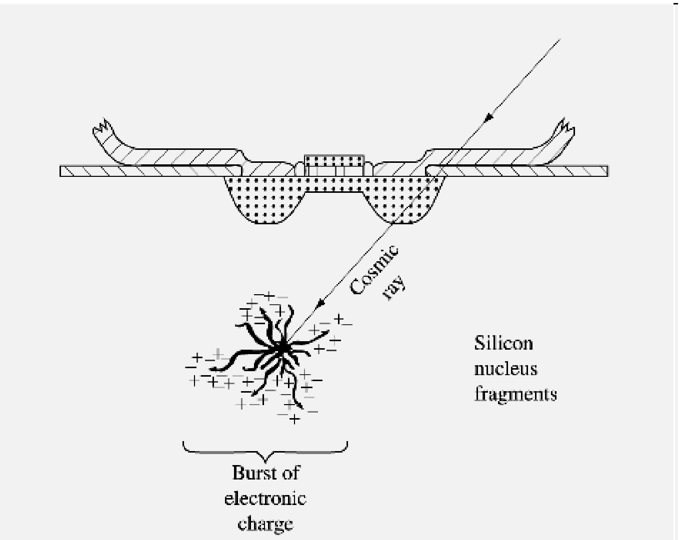
- Almost all of this power is in driving the latches/flops at the ends
  - Only around 20-25% of clock power is in clock transmission
  - Saving power at the ends is better than saving power in the wires

- Making a more efficient flop helps power significantly

Soft Errors

- Some soft errors are caused by radiation from the outside
  - Cosmic rays
  - Alpha particles

- Radiation strikes the chip and smacks into Silicon lattice
  - Generates a flood of h^+ and e^-
  - Charge flows into diffusions
  - Can upset the state of nodes
  - If charge exceeds Q_{crit}



Source: Ziegler, IBM JRD, 1996

- In general, soft errors include
  - EMI, signal integrity
  - Not just radiation…

SEU, FIT

- SEU = “Single Event Upset”
- SEU = change of state caused by a high-energy particle strike to a sensitive node in a micro-electronic device. An SEU can result in a soft error.

- Soft errors are measured in FITs
- FIT = “Failure In Time”
- 1 FIT = 1 failure per billion (10^9) hours of operation
  – Once per 115,000 years
- FIT is usually measured by means of an accelerated test procedure in semiconductor industry

Failure Rates for Soft Errors

- Numbers vary all over the place, depending on who you ask
  - Many memory vendors claim lowest FIT rates for their products

- DRAM FIT rates around 1 FIT/Mb (give or take 10x) today
  - And falling as technology scales
  - DRAM node cap is staying constant(-ish)

- SRAM FIT rates around 1000 FIT/Mb (give or take 10x)
  - And staying constant (falling very slowly) as technology scales

- But both are generally “don’t care” events, because of ECC
  - Unless the memory isn’t ECC, as in small registers or flops

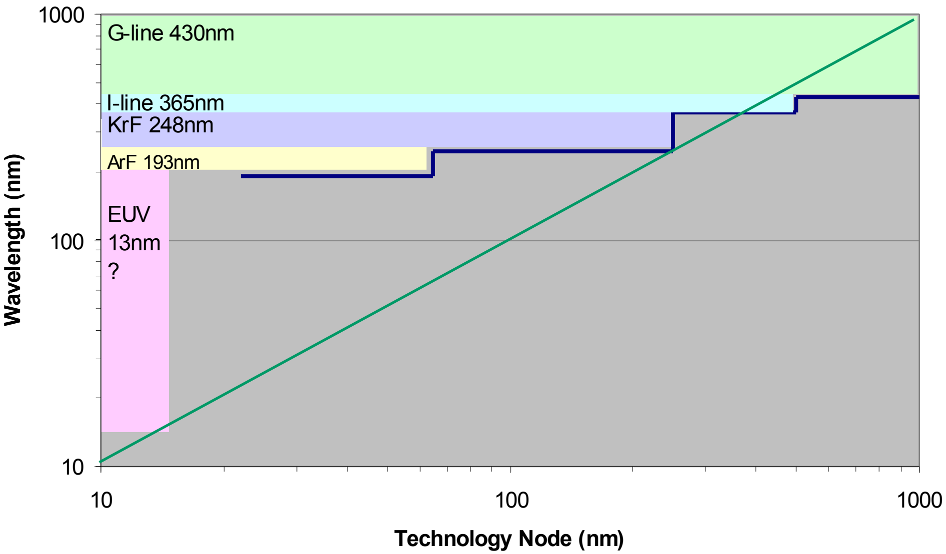
Solutions to Soft Errors

- In order of increasing cost…
- Use ECC in arrays and interleave entries
  - Single strikes can’t affect multiple rows
- Use static circuits instead of dynamic circuits
  - Static circuits can recover from minor charge injection
- “Harden” your latches or circuits
  - Double state, or add “ballast” capacitance
- Multiple CPUs can run in lock-step
  - Check architectural state at interface

Variation

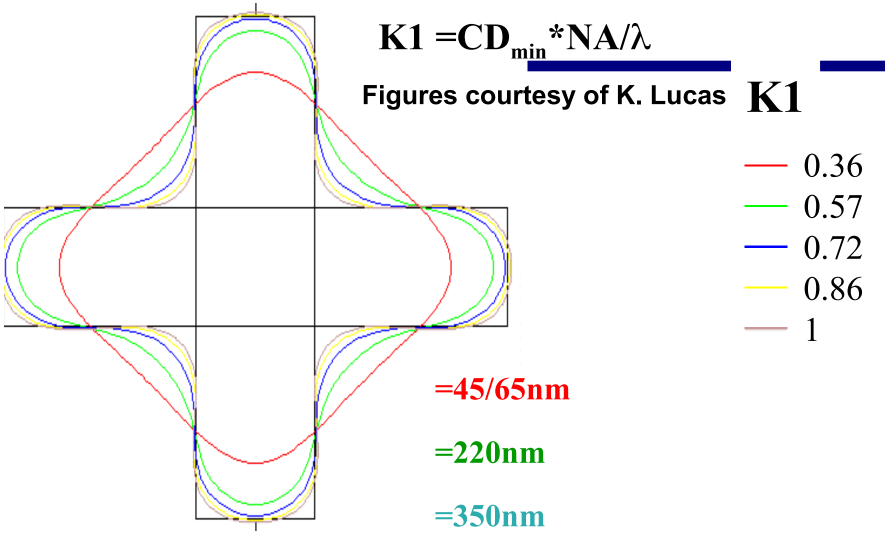
- Variation is becoming more important – why is this?
- We're near the famous "end of the roadmap":
  - Atoms don't scale
  - Photons don't scale (much) (for drawing masks)
  - **Variability** as a percentage of feature size increases
- Sources of Variation are diverse
  - Process-induced
  - Environmental
  - Physical limit-induced variations
- Impact of Variation can be on Performance, Power, Yield

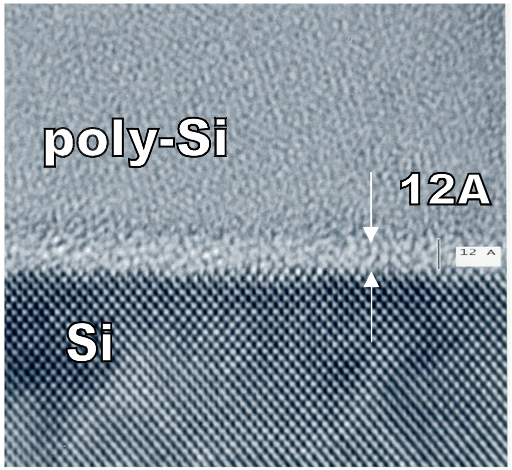
Introduction – Physical Limit Issues

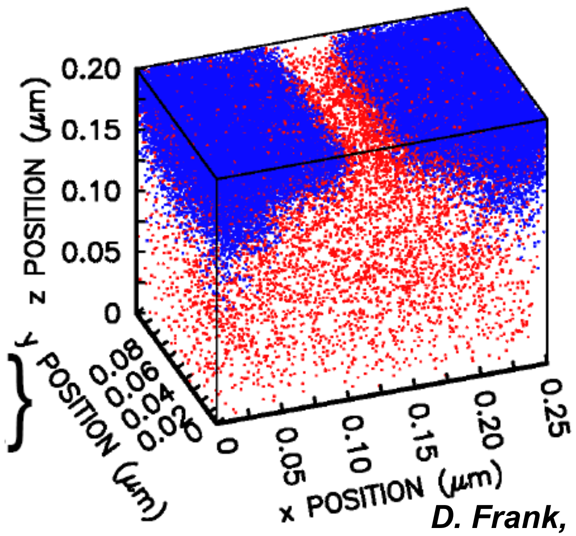


$K1 = CD_{min} * NA/\lambda$

Figures courtesy of K. Lucas

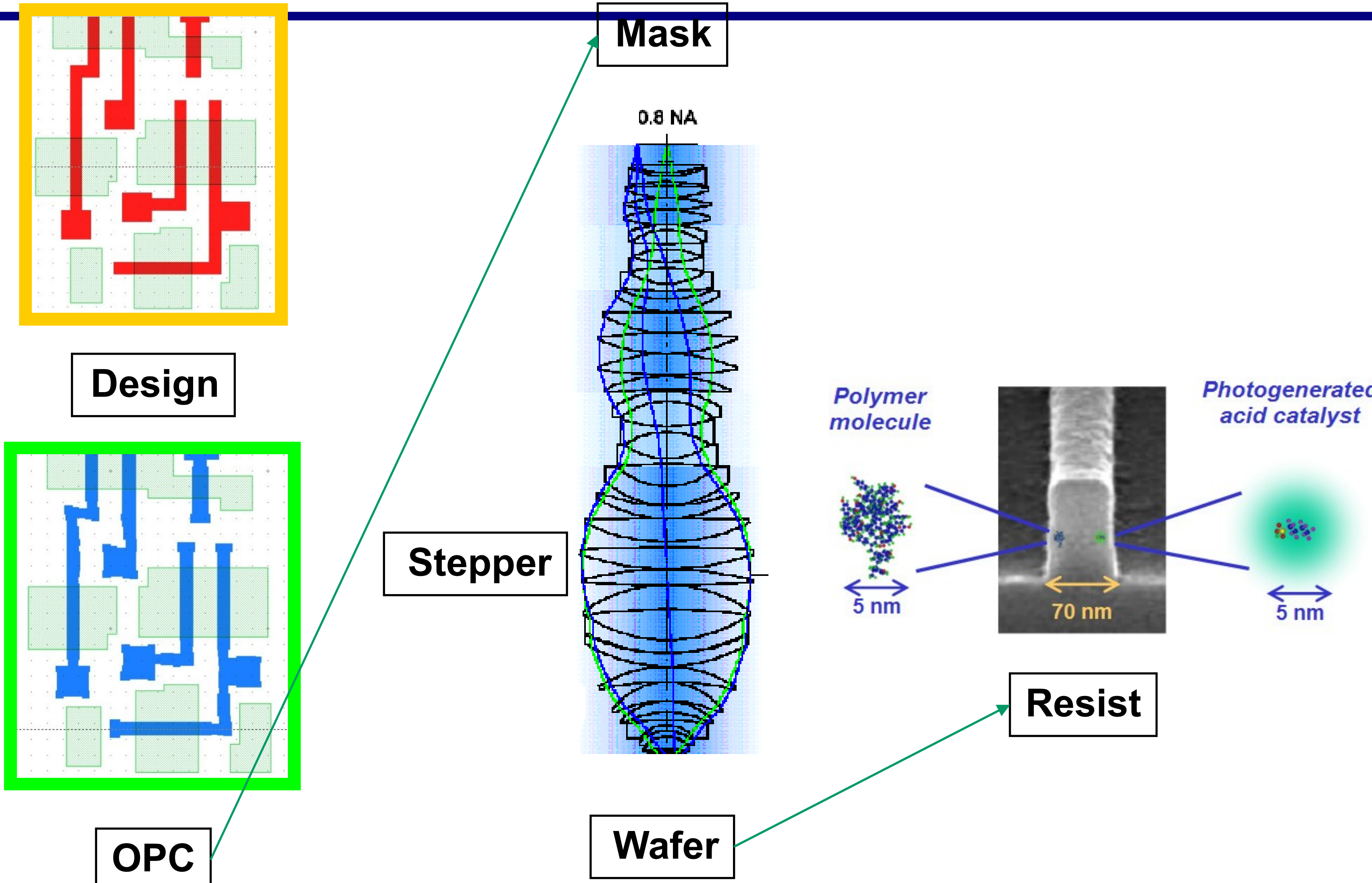






D. Frank, ISSCC 2004

Lithography-Related Variations (Gate CD)



Types of Variation

- “Global” and systematic
  - Covers significant areas of the die
  - Focus, Rapid Thermal Annealing, Polishing, etc.
- “Unmodeled” Deterministic Effects
  - Temperature, voltage, stress, density
- Local & Random
  - Device to device
  - LER (Line Edge Roughness), Dopant fluctuation

Reducing Variations

- Most logic paths are *relatively* benign
  - Use larger, multi-finger devices which reduce variation
  - Use greater logic depth, often spread out across chip
  - Use a variety of metal levels