CS 250
VLSI System Design

Lecture 1 – The Fab/Design Interface

2013-8-29

Professor Jonathan Bachrach

today's lecture by John Lazzaro

TA: Ben Keller

www-inst.eecs.berkeley.edu/~cs250/

Today's lecture plan ...

ON THE SAME PAGE

UNIVERSITY OF CALIFORNIA BERKELEY

Sixty minute tour of semiconductor technology, to put the class outline in context.

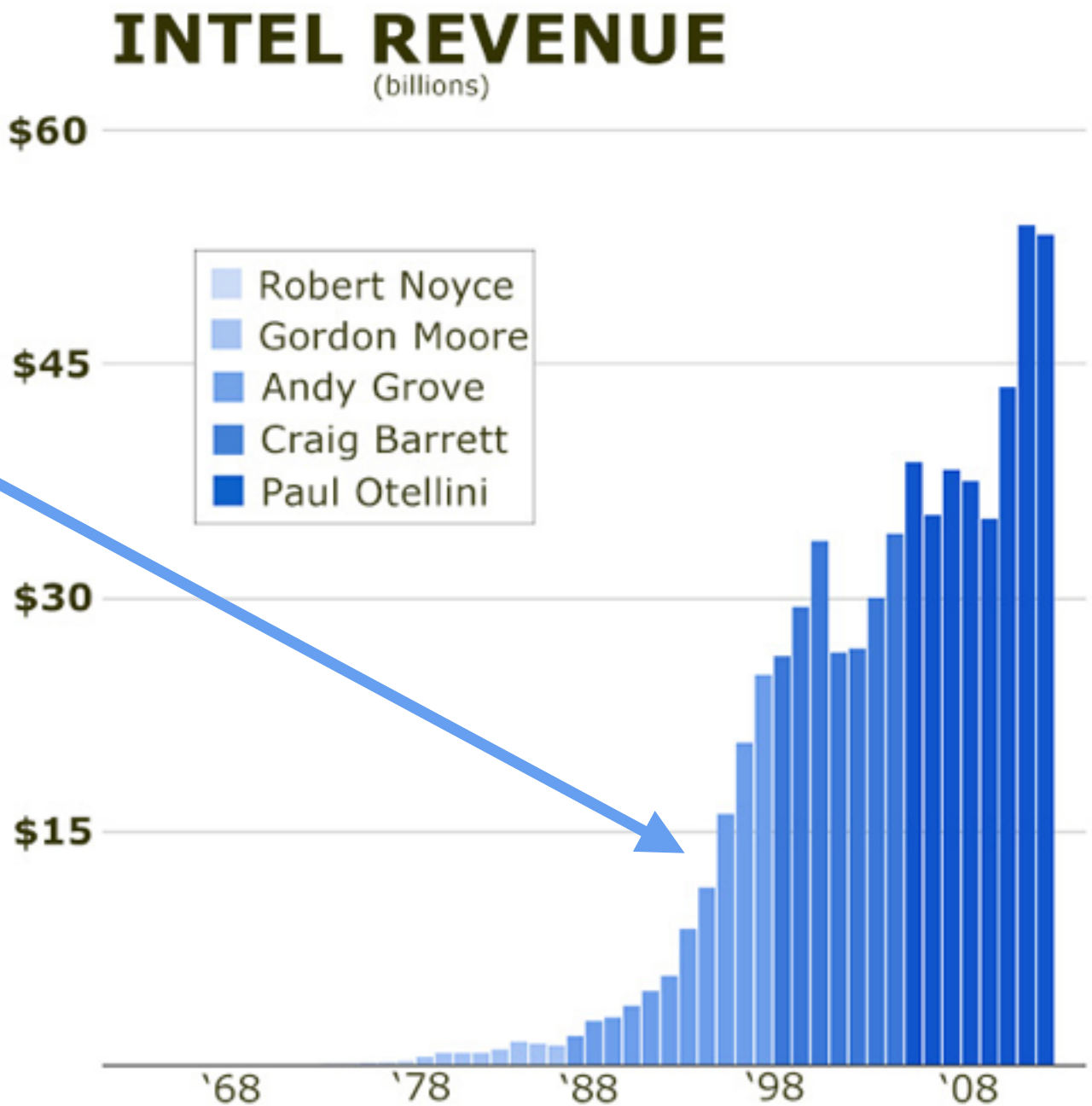
✳ **Short Break.**

✳ **Class Outline.**

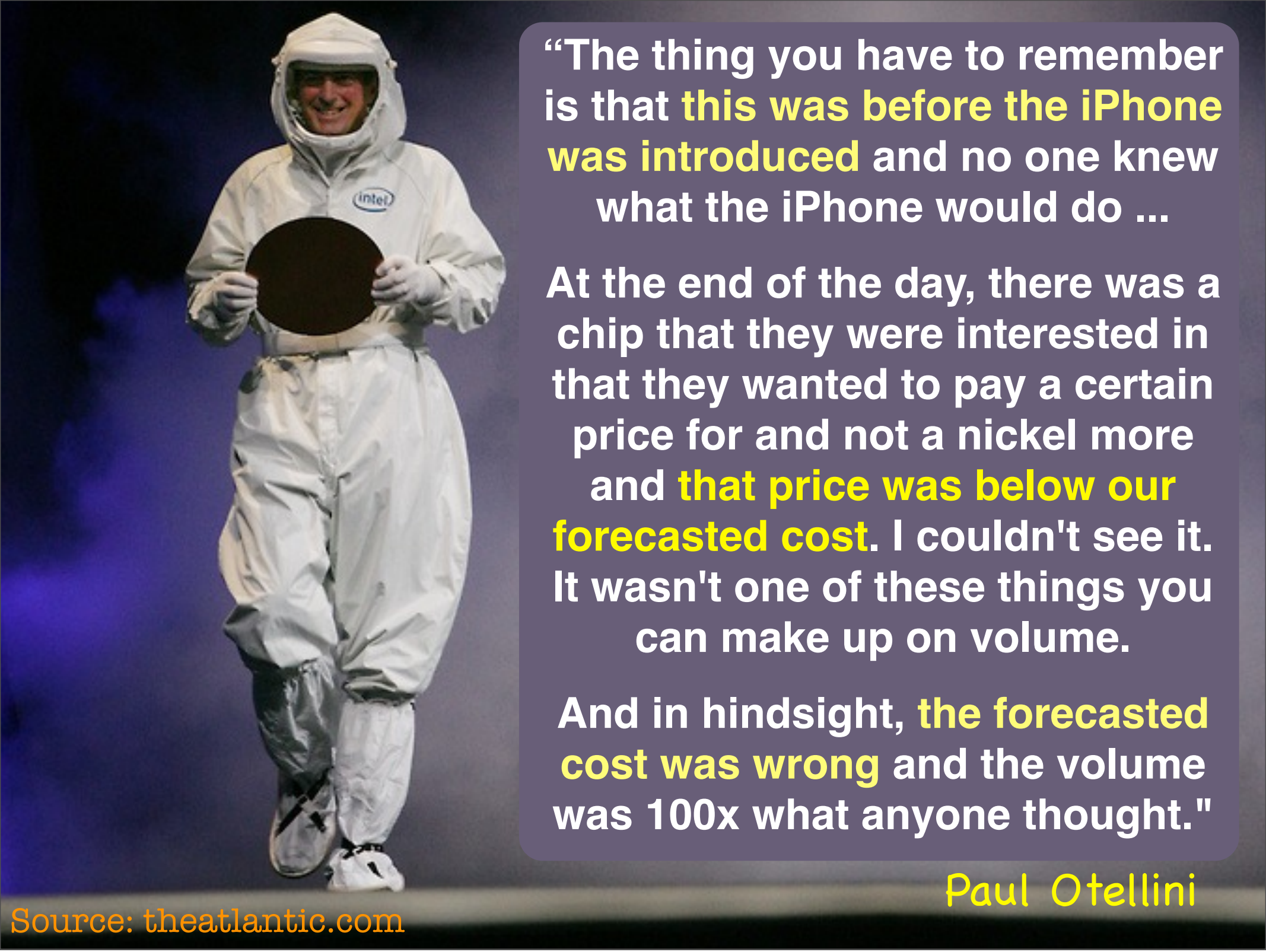
What we'll be doing this semester.

Economics ... inextricably part of chip design

Andy Grove
UCB Ph.D. 1963




Moscone Center, 2005.
Apple switches
the Mac to Intel CPUs.
intel
Steve Jobs
Paul Otellini,
Haas School
MBA 1974



"The thing you have to remember is that this was before the iPhone was introduced and no one knew what the iPhone would do ...

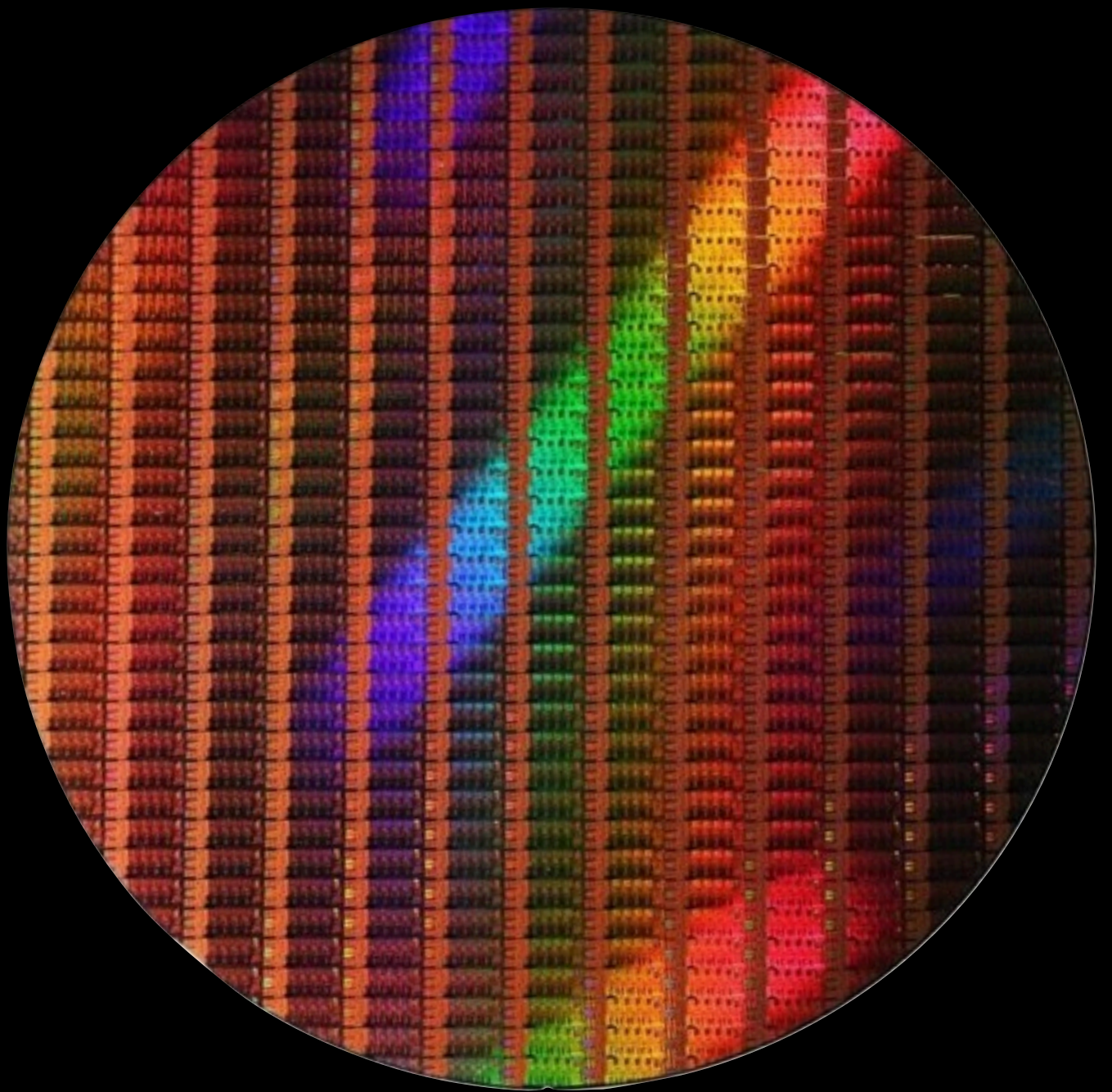
At the end of the day, there was a chip that they were interested in that they wanted to pay a certain price for and not a nickel more and that price was below our forecasted cost. I couldn't see it. It wasn't one of these things you can make up on volume.

And in hindsight, the forecasted cost was wrong and the volume was 100x what anyone thought."

Paul Otellini

Source: theatlantic.com

Intel Haswell wafer (22nm FinFET)

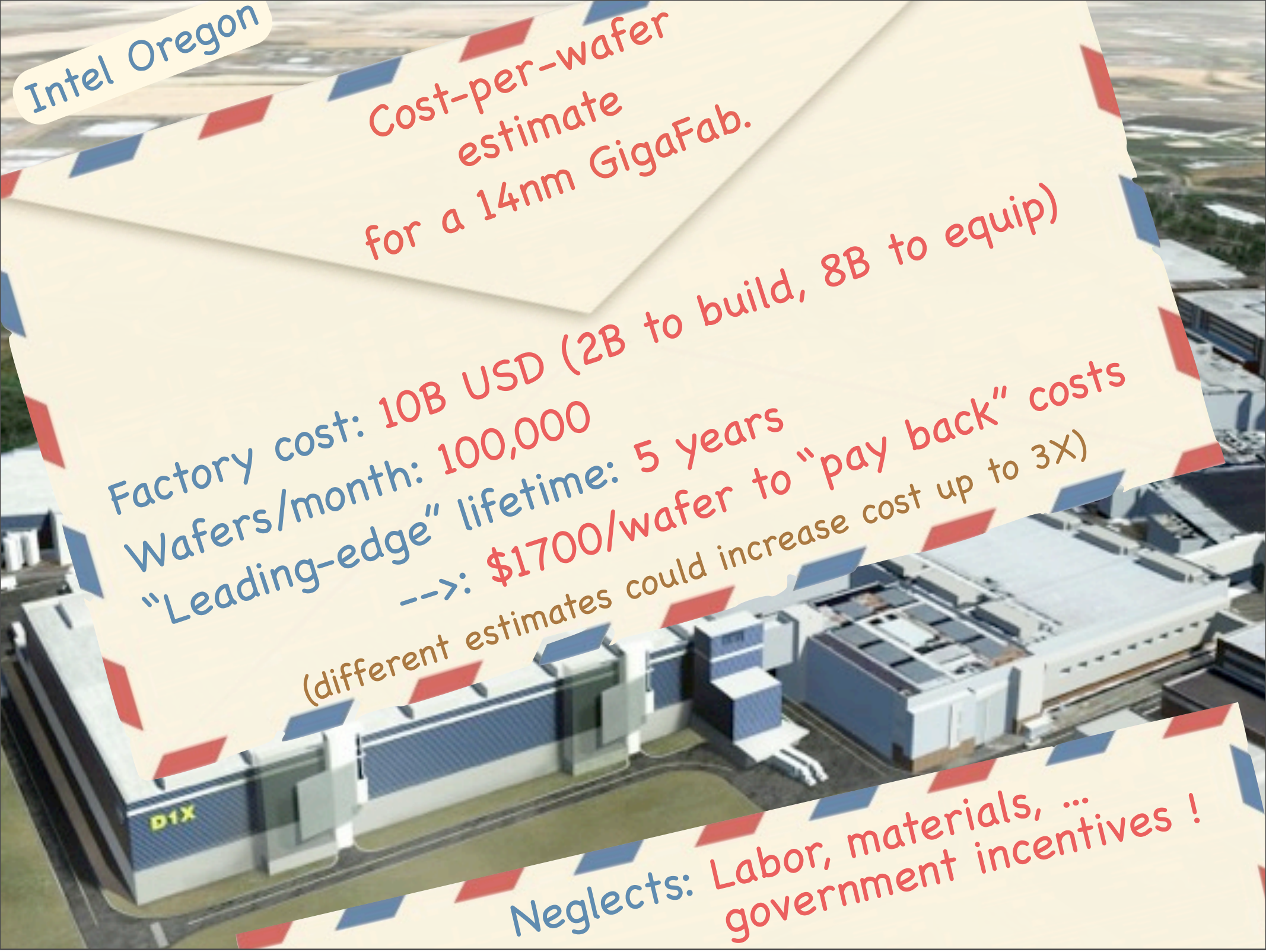


How do you estimate the manufacturing cost of a chip?

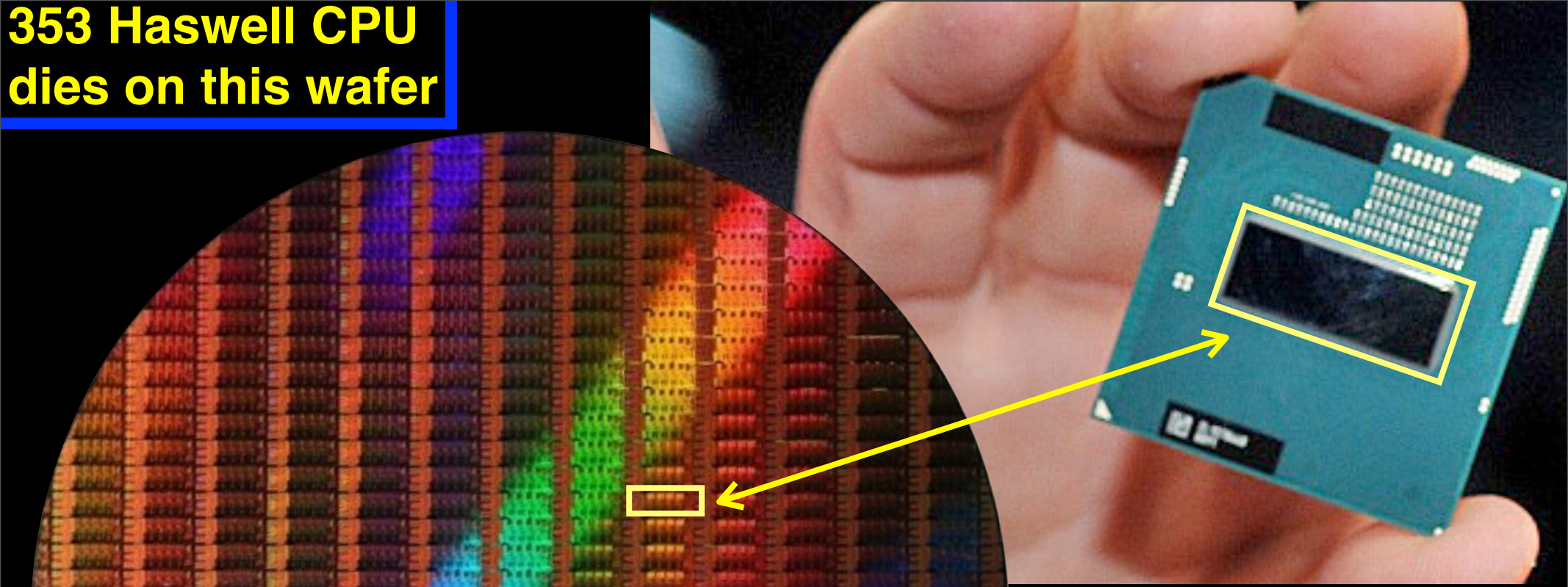
Chip factories ("fabs") process wafers that contain many copies of a chip ("dies")

So, step one is estimating the cost to manufacture a wafer.


Intel Oregon
D1X


Intel Oregon
Cost-per-wafer estimate for a 14nm GigaFab.
Factory cost: 10B USD (2B to build, 8B to equip)
Wafers/month: 100,000
"Leading-edge" lifetime: 5 years
-->: $1700/wafer to "pay back" costs
(different estimates could increase cost up to 3X)
Neglects: Labor, materials, ... government incentives !
D1X

353 Haswell CPU dies on this wafer



==> **$4.80 per die!**

But if die were twice as big ... $9.60 per die!

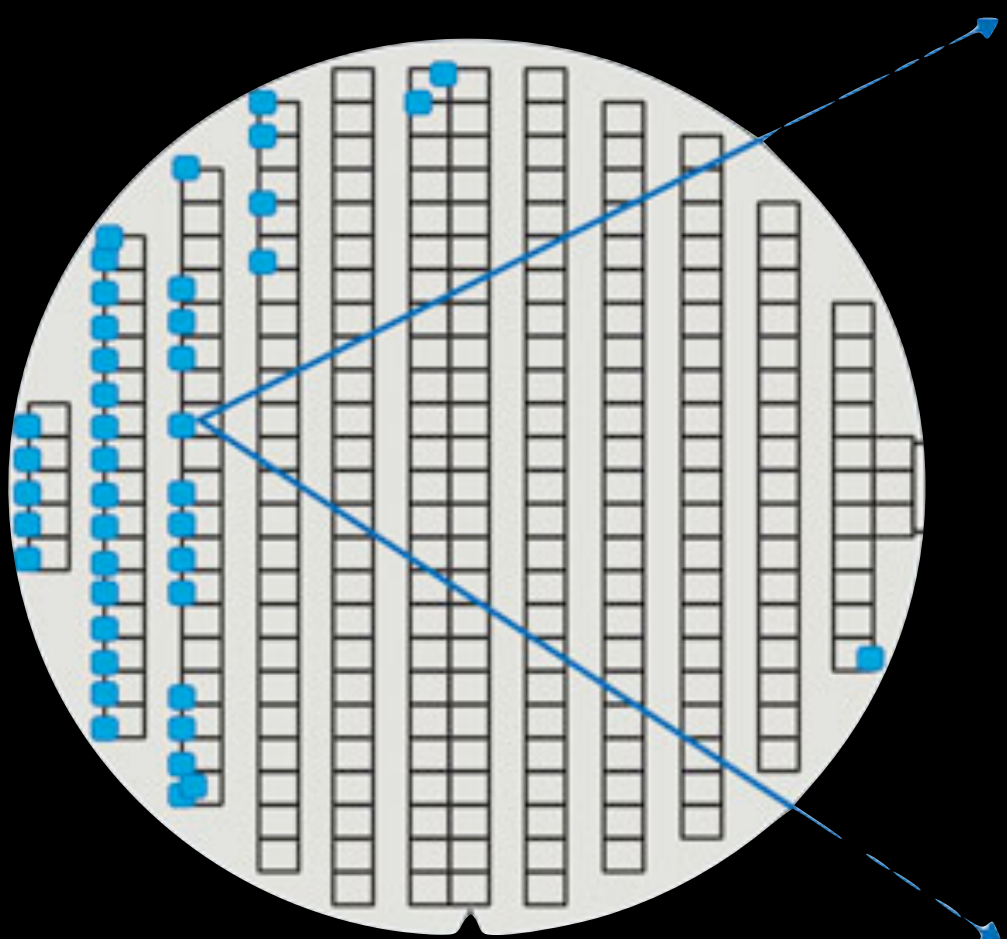
This is one reason why die size matters.

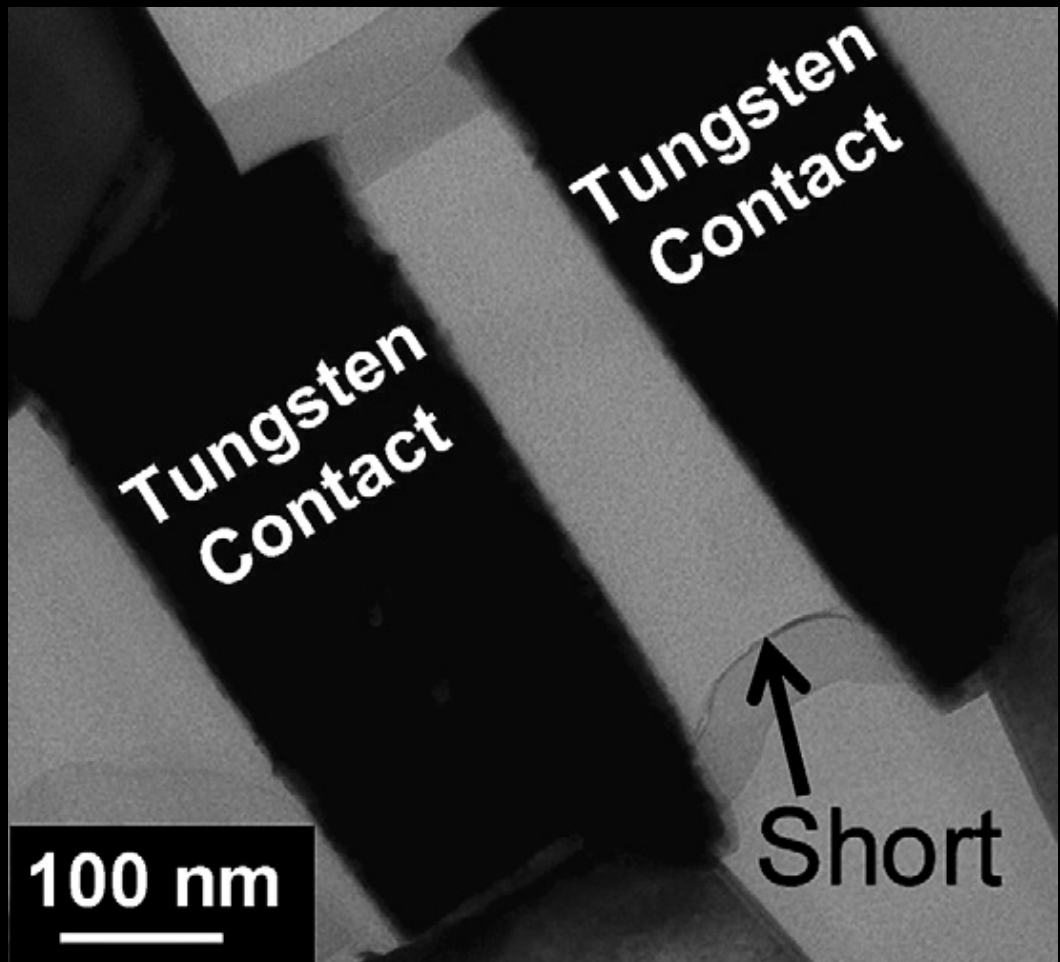
Die counts per column

11 21 27 31 33 35 37 35 33 31 27 21 11

This analysis is optimistic ...

Yield: Not all dies on the wafer will work!





A = die area D_o = defects per unit area

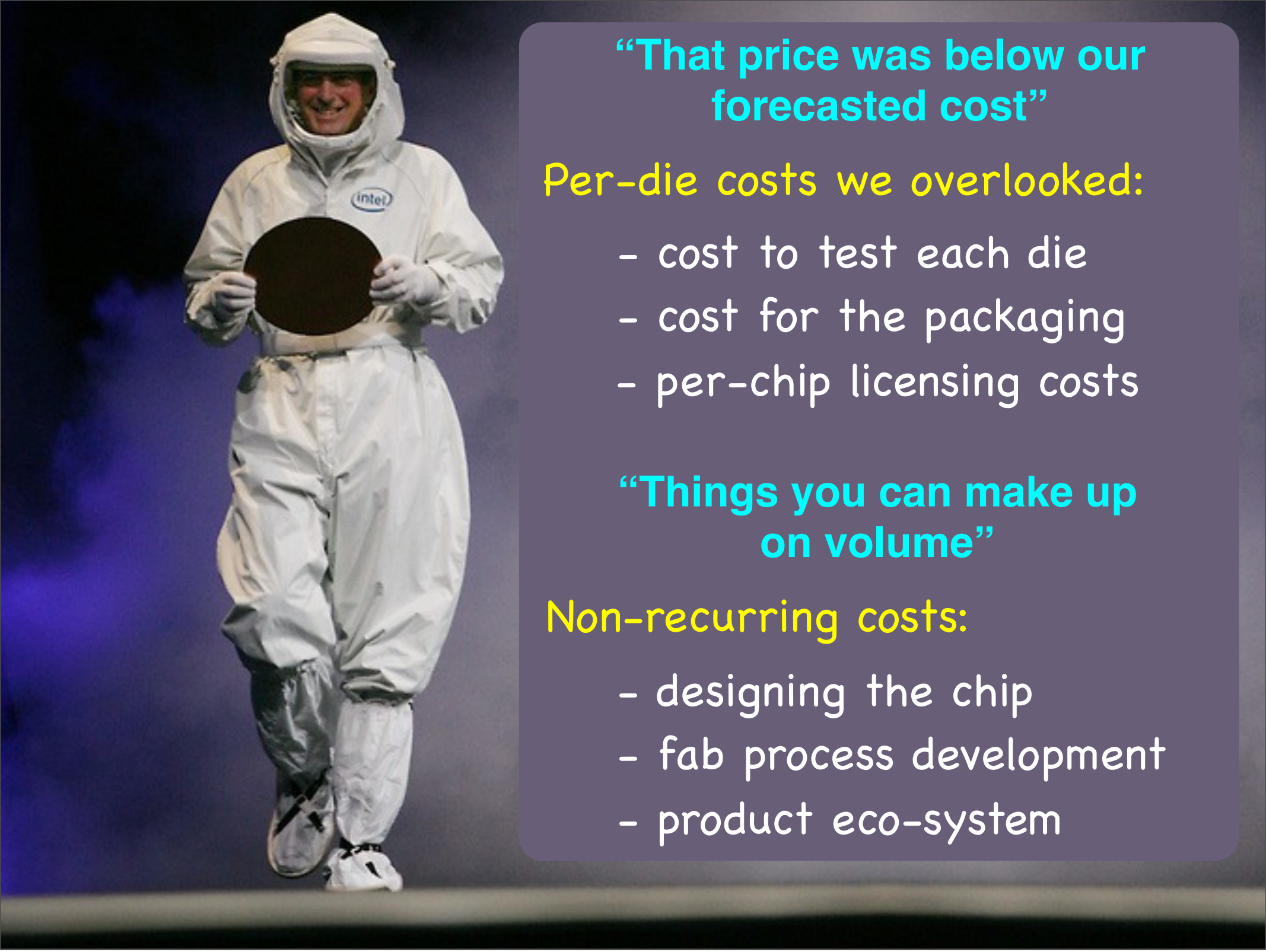
Yield (A) = 100% × exp(-A*D_o)

If A << D_o, yield approach 100%.

If A ≈ 2 D_o, yield is about 13%

Another reason why die size matters!

Note: this "Poisson" model is overly pessimistic, real-world yield models are more complex.

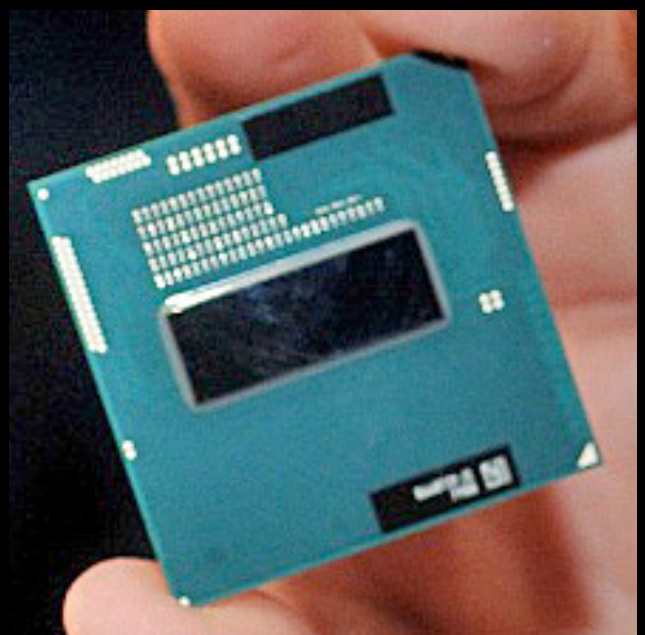

"That price was below our forecasted cost"
Per-die costs we overlooked:
- cost to test each die
- cost for the packaging
- per-chip licensing costs
"Things you can make up on volume"
Non-recurring costs:
- designing the chip
- fab process development
- product eco-system

Industry Organization

i286 design team (1984)

Integrated Device Manufacturers (IDMs)

Intel designs "standard product" chips that sell to many customers (Apple, Dell, HP).

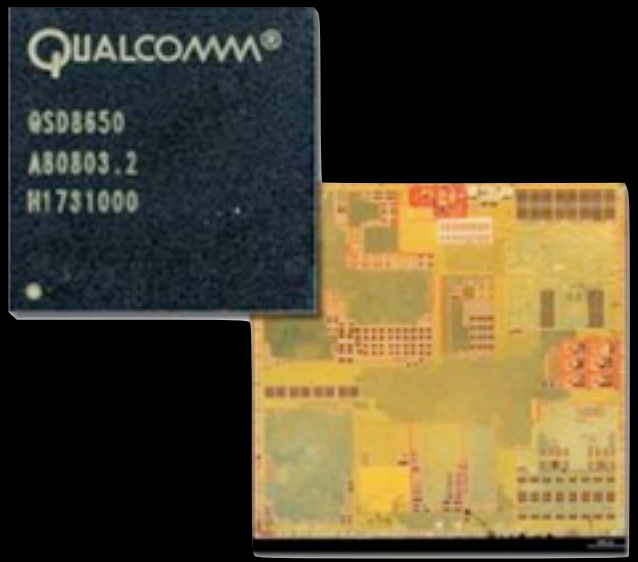


It develops IC process technology, for use in its own wafer fabs.

This was the original industry model.

It avoids "competing with its customers" (Intel doesn't make notebook PCs).

Fabless Merchant Model



Qualcomm designs chips for use in smartphones, but does not own fabs.

TSMC (a "foundry") owns fabs, but does not do chip design. Qualcomm contracts with TSMC to design its chips.

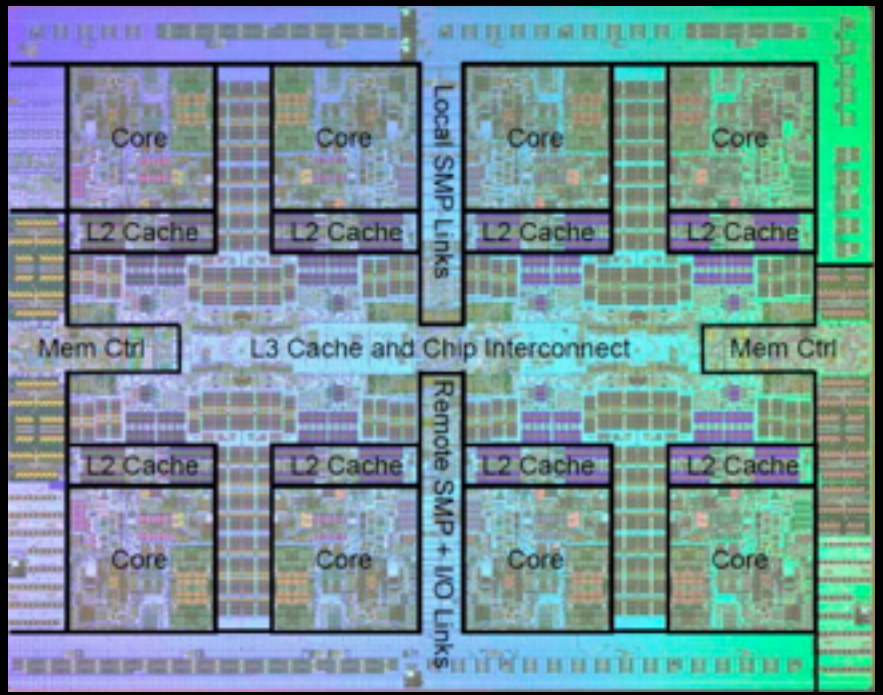
HTC (and many other smartphone companies) buy chips from Qualcomm.

Fabless Captive Model

Apple designs chips (the "A" series) for exclusive use in its iOS-series products (such as the iPhone).

Apple doesn't own a fab, and so it contracts with foundries (currently Samsung!)

Original Captive Model



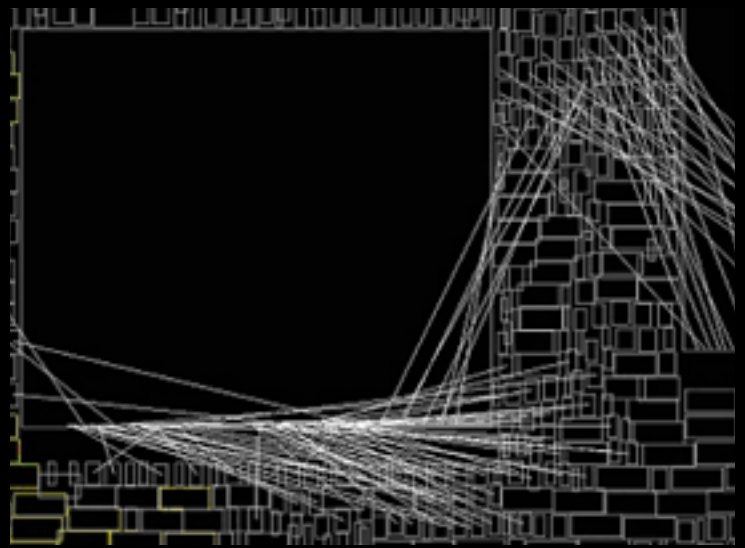
IBM owns fabs, and designs CPU chips, for use in most of its server lines.

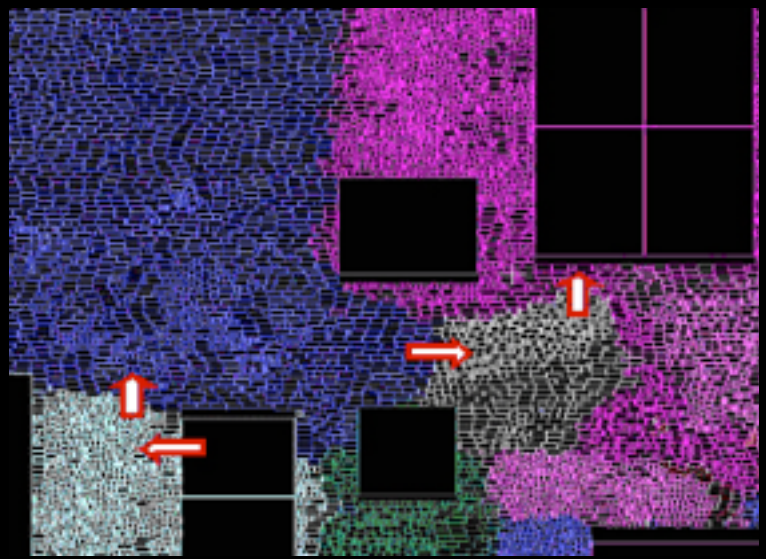
For many decades, it manufactured DRAM chips (Robert Dennard, the inventor of the DRAM, works for IBM).

Entire industries exist to "sell arms" to "semis"

"Electronic Design Automation" (EDA, or CAD)

SYNOPSYS®
Accelerating Innovation





Chip fabrication tools

KLA-Tencor
Accelerating Yield

Wafer manufacturing

Silicate materials form 90% of the earth's crust

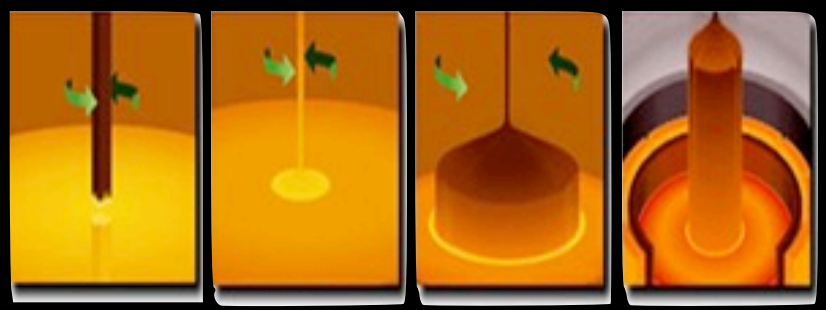
Silica (silicon dioxide) is mined, mostly for use in concrete.

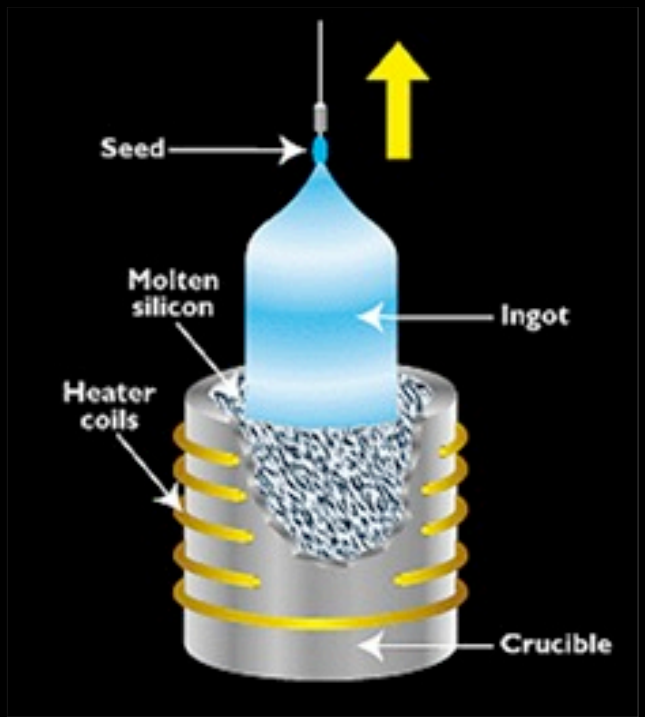
Refined elemental silicon (95% pure) is mostly used for aluminum metallurgy.

Electronic grade silicon requires 99.9999999% purity.

"Single crystalline" composition is also required for wafers.

Silicon "ingots" are grown from a "perfect" crystal seed in a melt, and then purified to "nine nines".





Ingots sliced into 450µm thick wafers, using a diamond saw.

This photo is from Intel's ecommerce site for server chips.

This photo is from Intel's ecommerce site for server chips.

6 inch square Si single-crystal wafer, $2 on Alibaba

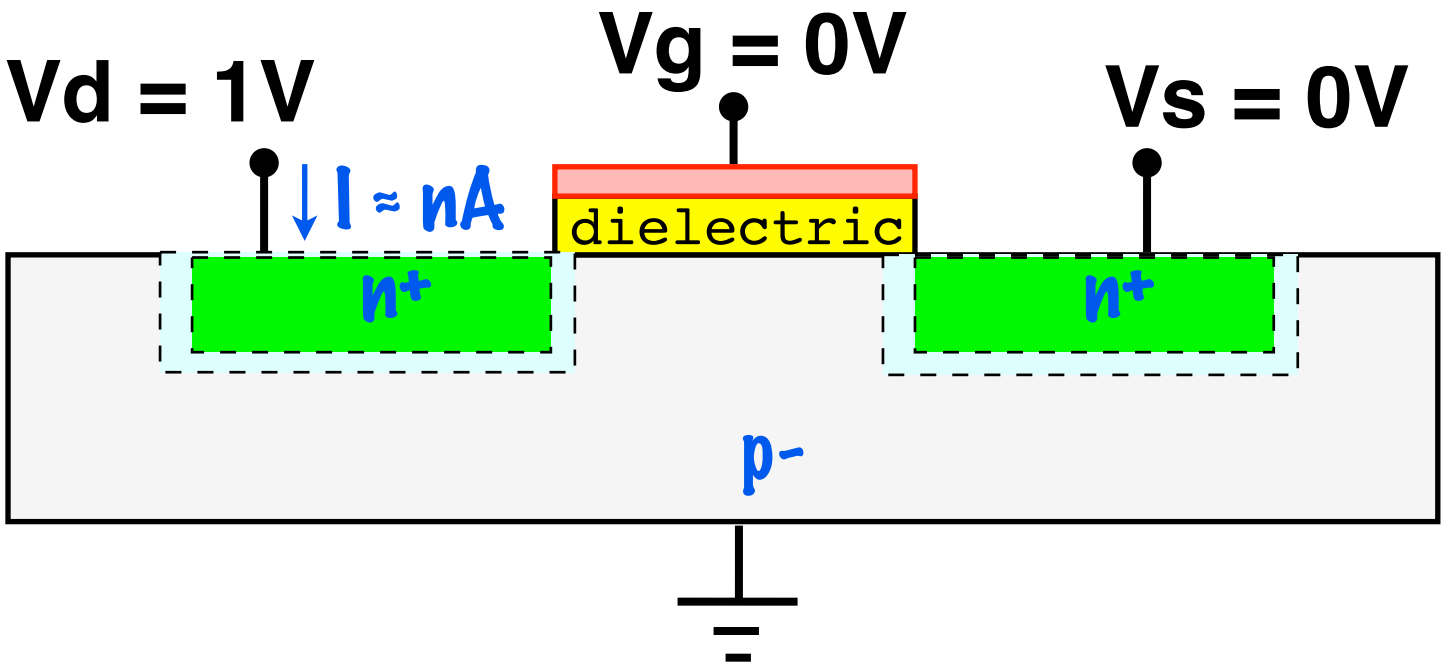
Wafer costs are crucial for solar cells, but irrelevant for chips that perform high-value functions.

Xeon E7-8870, in small quantities, $4600

But yet, photo was shot behind solar cells to look "hi-tech" :-).

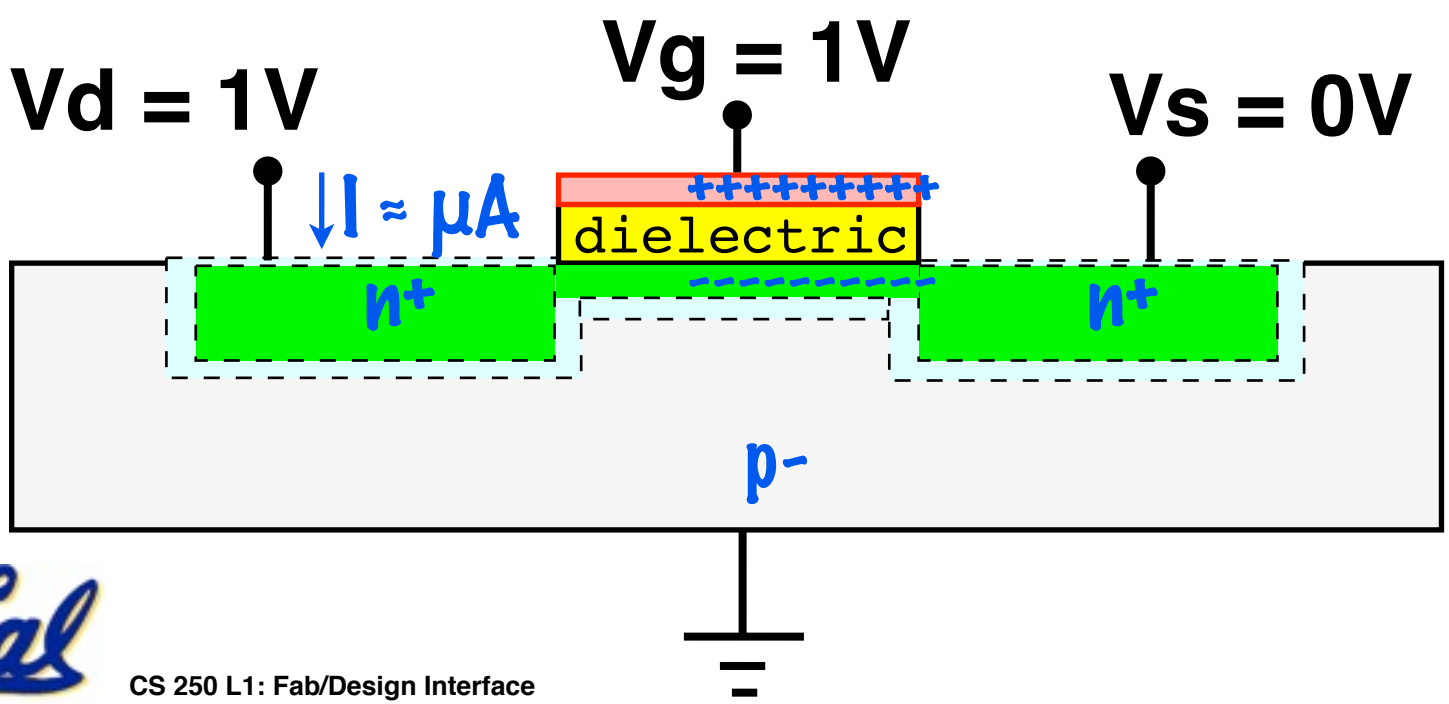
Fabrication

An n-channel MOS transistor (planar)



Polysilicon gate, dielectric, and substrate form a capacitor.

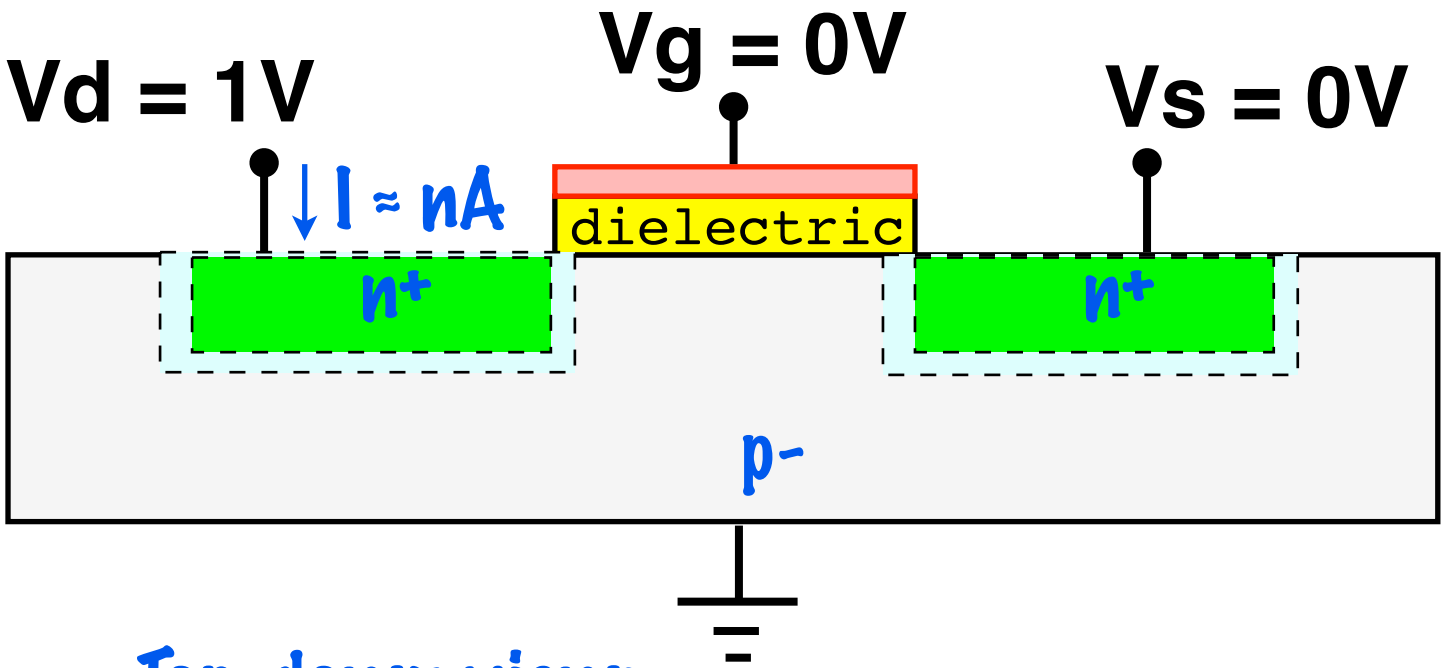
nFet is off
(I is "leakage")



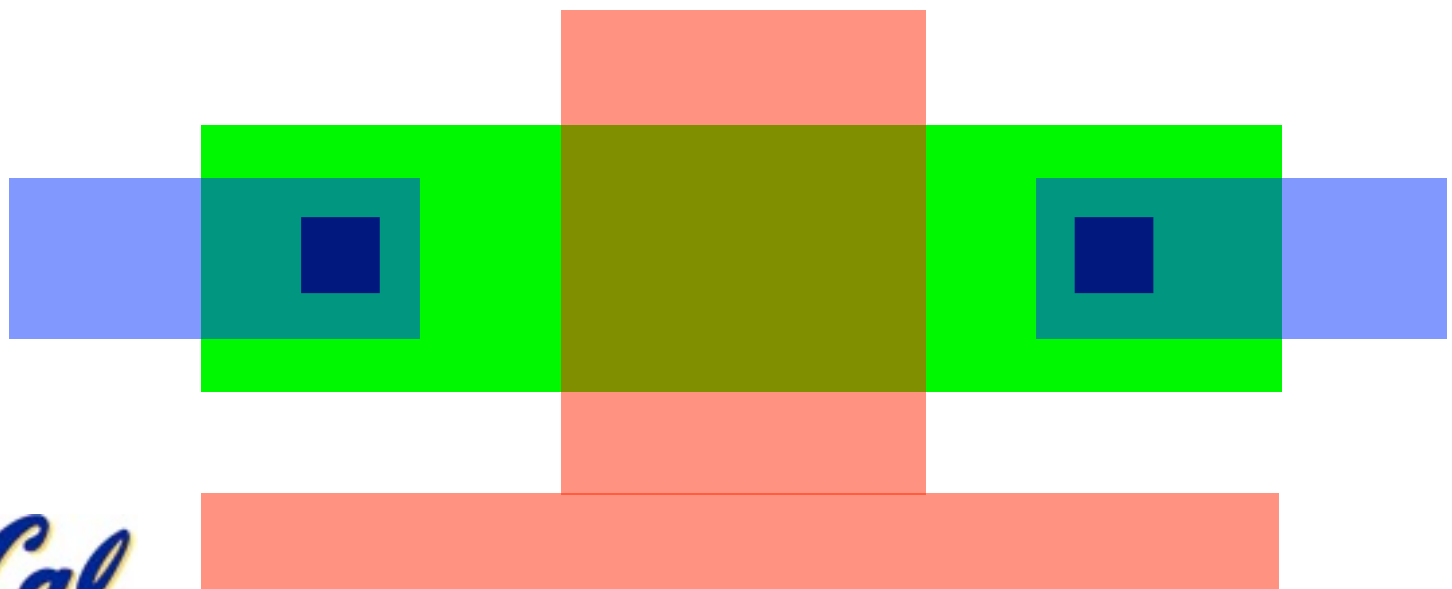
Vg = 1V, small region near the surface turns from p-type to n-type.

nFet is on.

Mask set for an n-Fet (circa 1986)



Top-down view:



Masks

#1: n+ diffusion

#2: poly (gate)

#3: diff contact

#4: metal

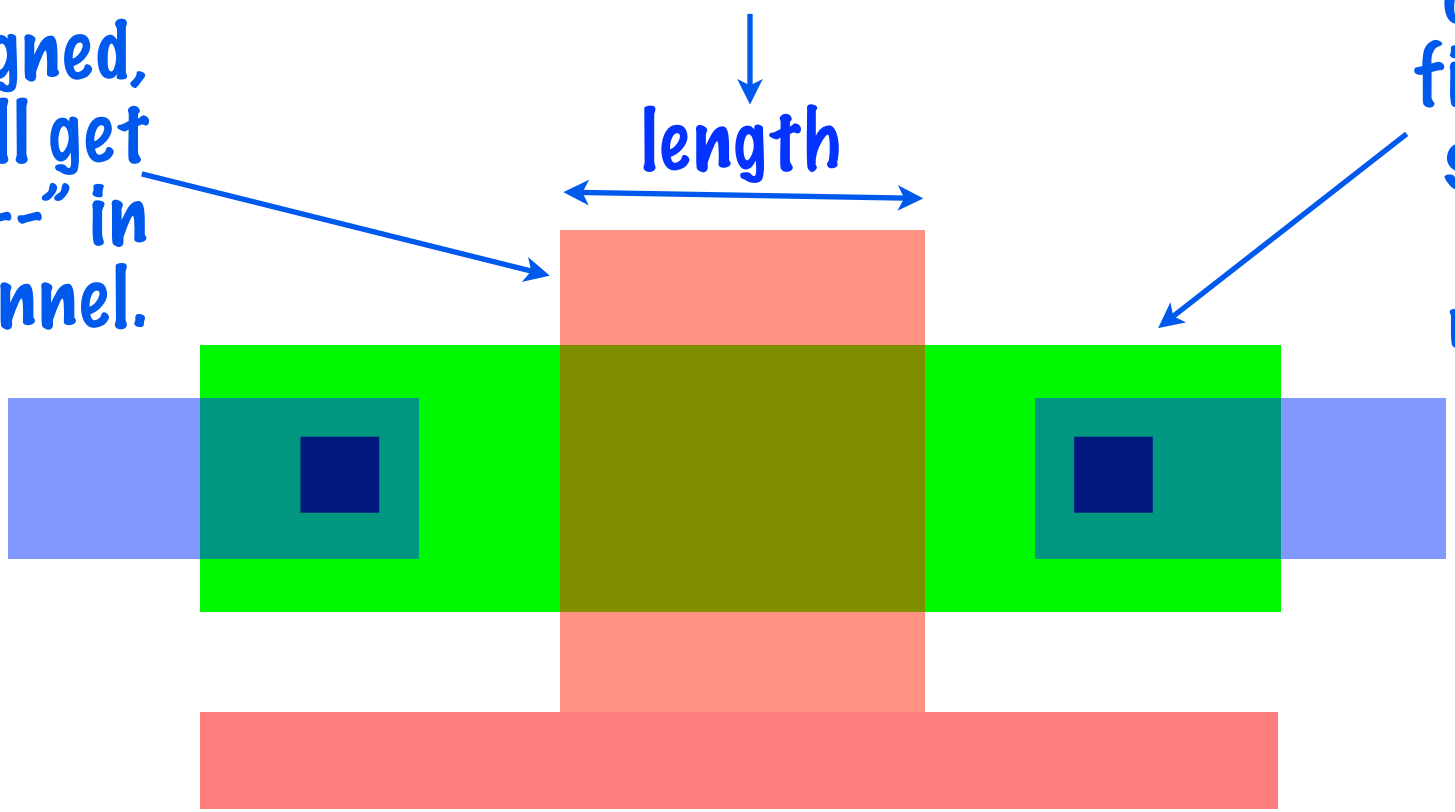
Layers to do p-Fet not shown. Modern processes have 6 to 10 metal layers (or more) (in 1986: 2).

"Design rules" for masks, 1986 ...

Poly overhang. So that if masks are misaligned, we still get "---" in channel.

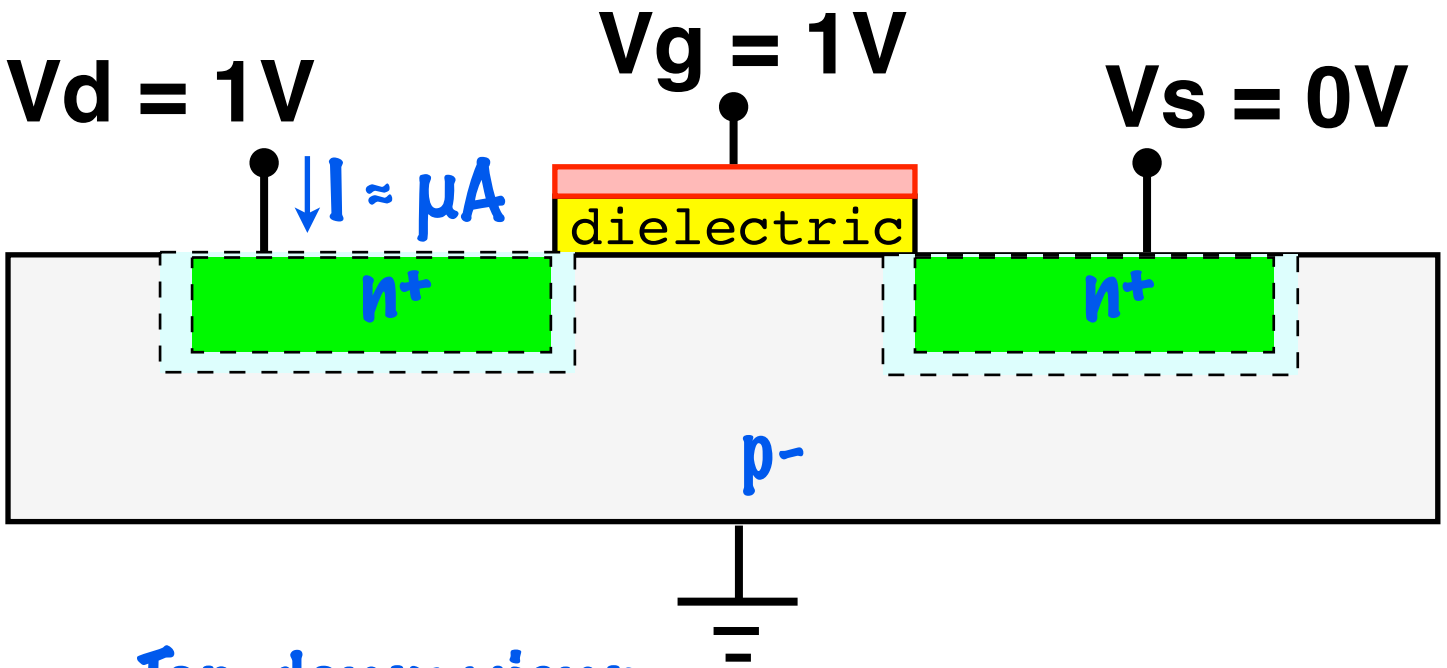
Minimum gate length. So that the source and drain depletion regions do not meet!

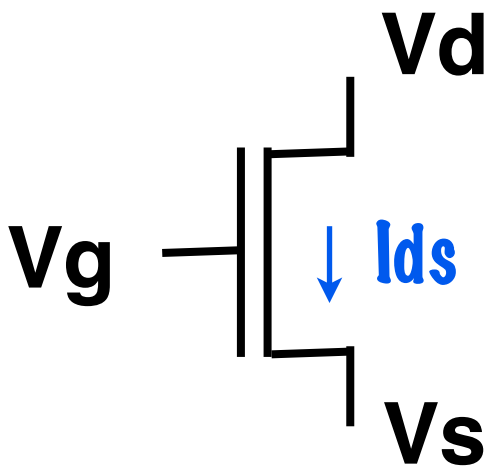
Metal rules: Contact separation from channel, one fixed contact size, overlap rules with metal, etc ...



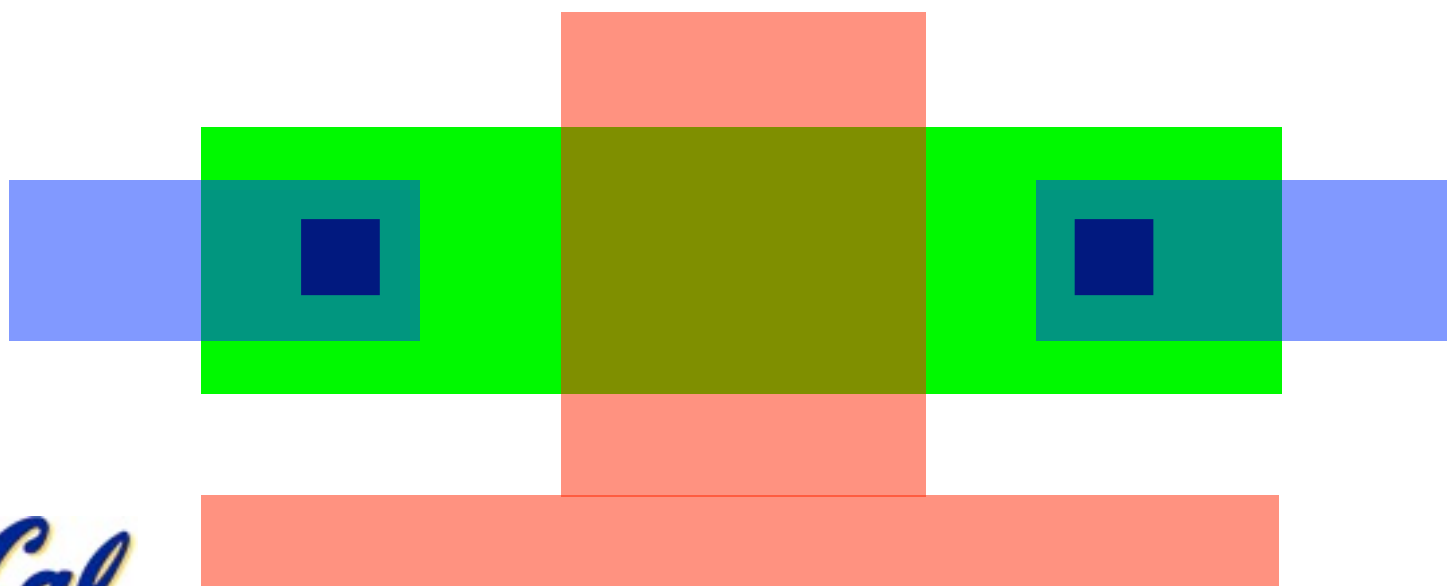
#1: n+ diffusion
#2: poly (gate)
#3: diff contact
#4: metal

How a fab uses a mask set to make an IC





Top-down view:

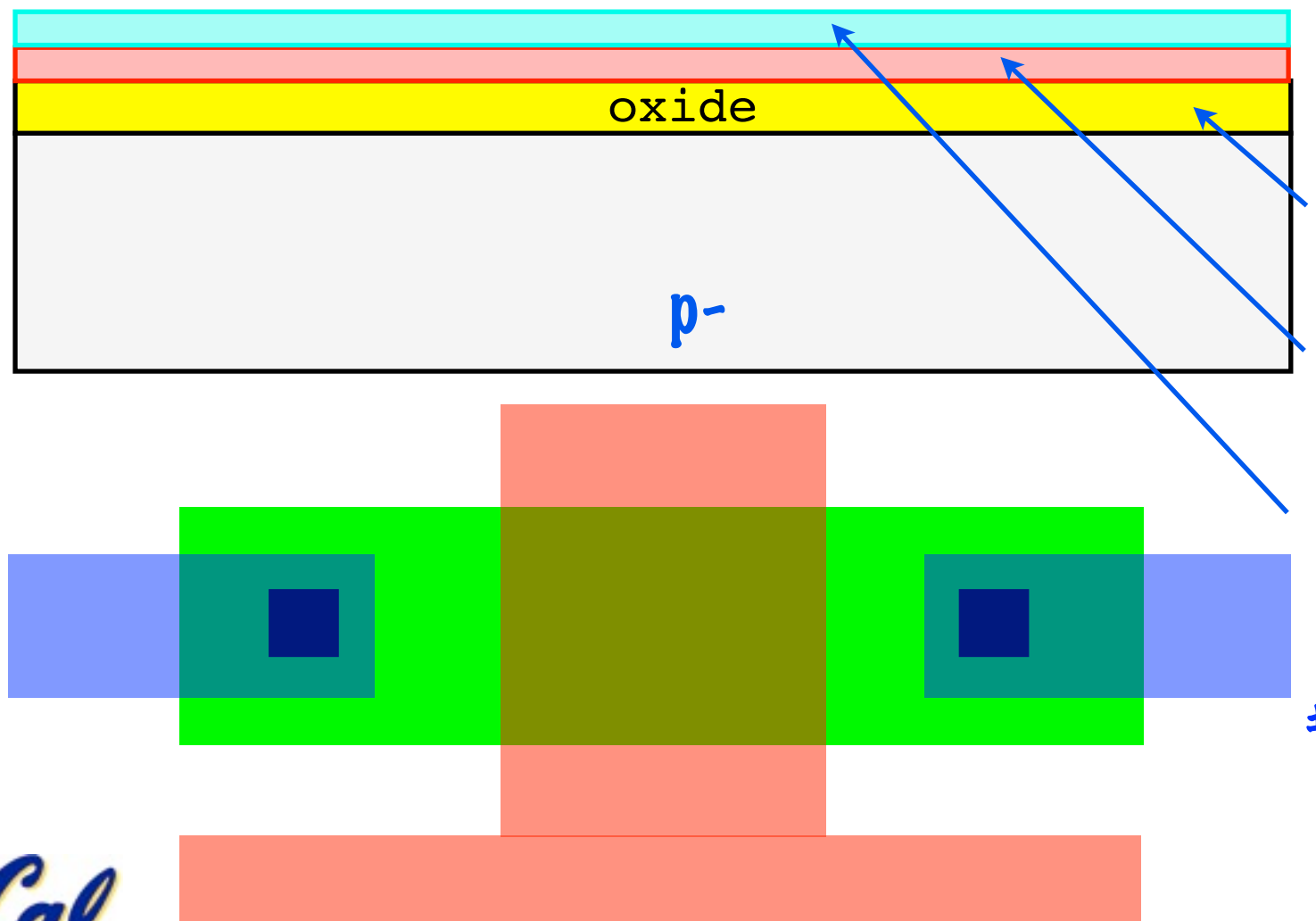


Masks

- #1: n+ diffusion
- #2: poly (gate)
- #3: diff contact
- #4: metal

Start with an un-doped wafer ...

UV hardens exposed resist. A wafer wash leaves only hard resist.



Steps

#1: dope wafer p-

#2: grow gate oxide

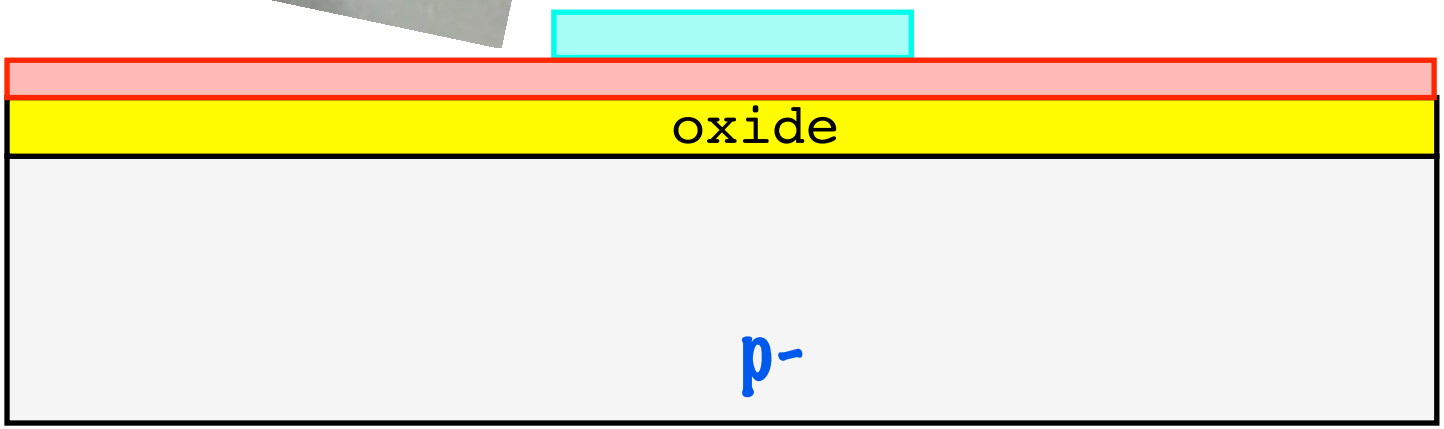
#3: grow undoped polysilicon

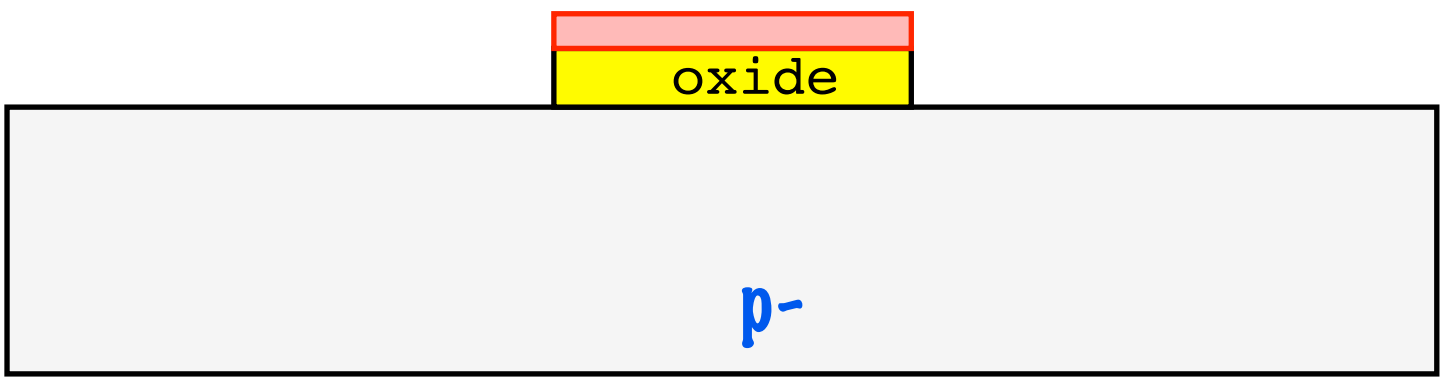
#4: spin on photoresist

#5: place positive poly mask and expose with UV.

Wet etch to remove unmasked ...

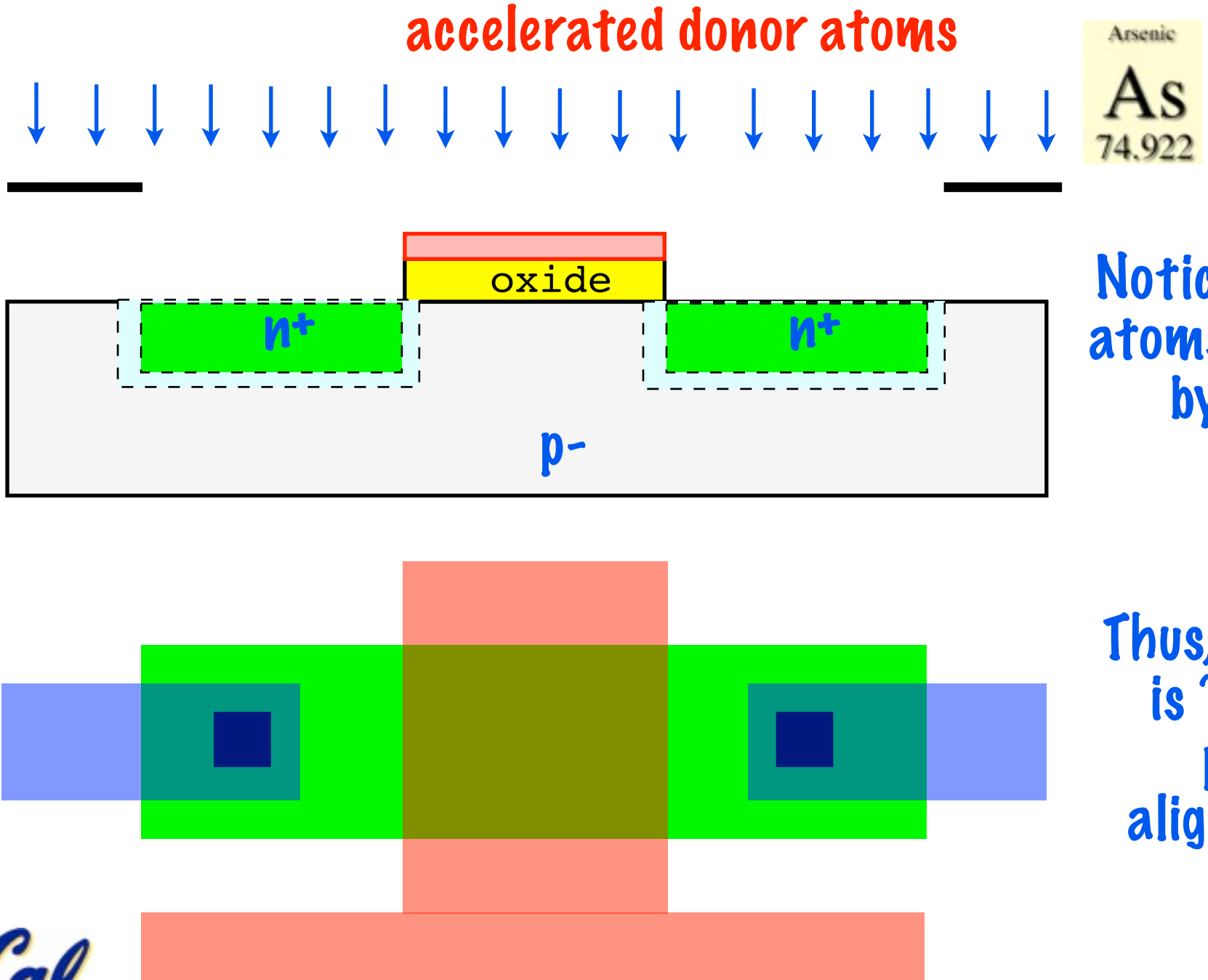
HF acid etches through poly and oxide, but not hardened resist.





After etch and resist removal

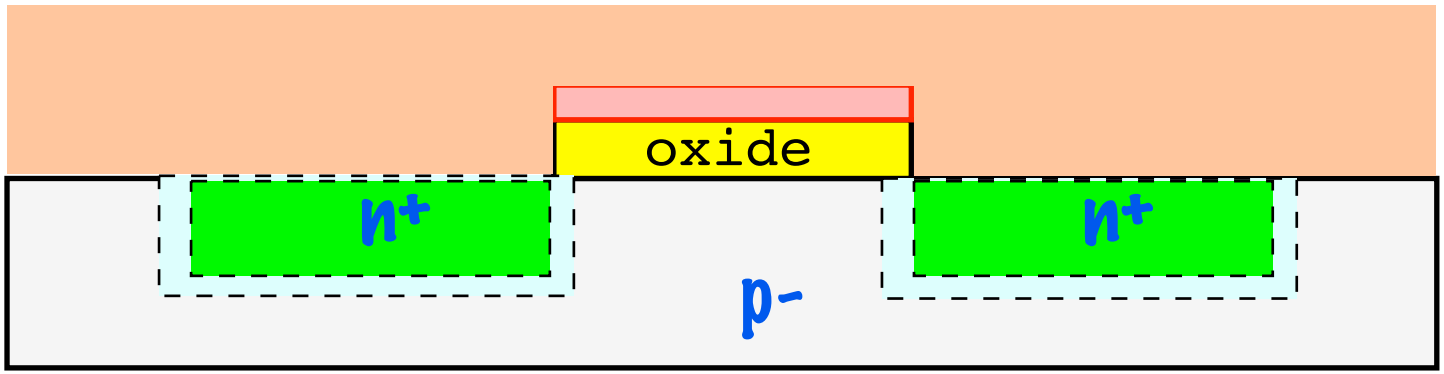
Use diffusion mask to implant n-type



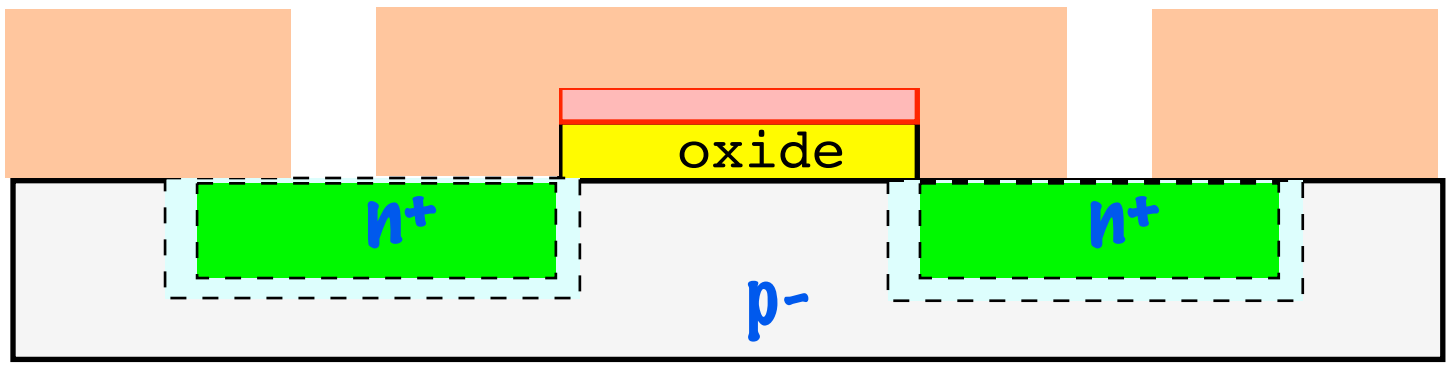
Notice how donor atoms are blocked by gate and do not enter channel.

Thus, the channel is "self-aligned", precise mask alignment is not needed!

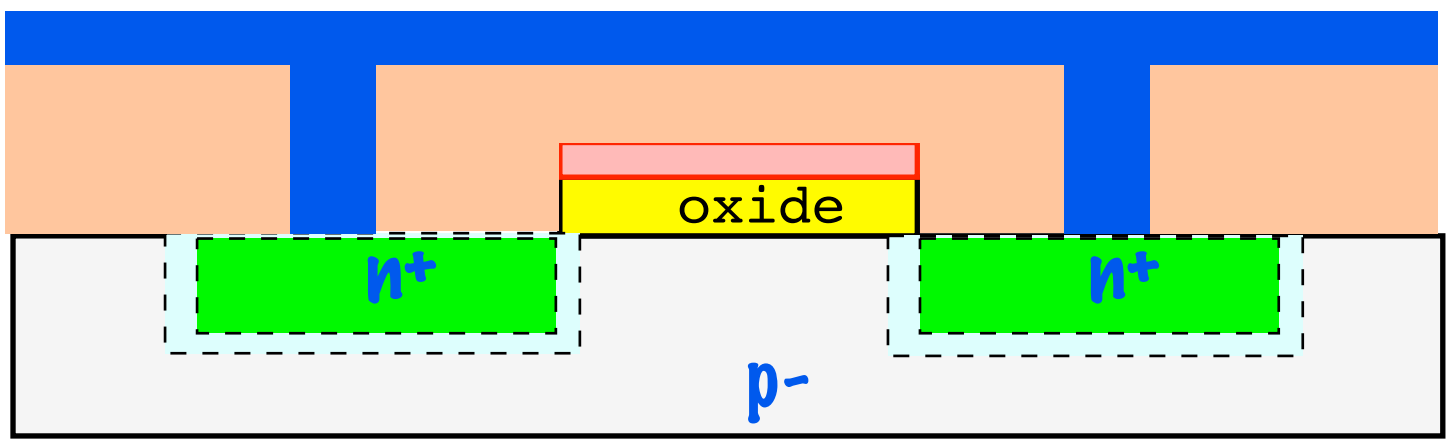
Metallization completes device



Grow a thick oxide on top of the wafer.

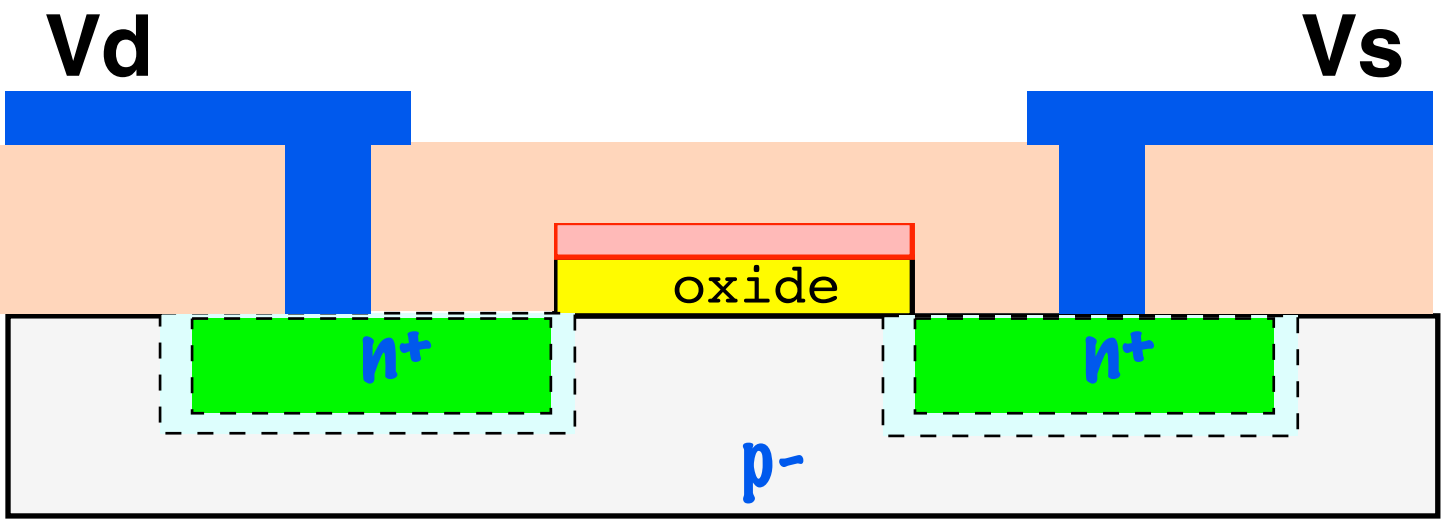


Mask and etch to make contact holes



Put a layer of metal on chip. Be sure to fill in the holes!

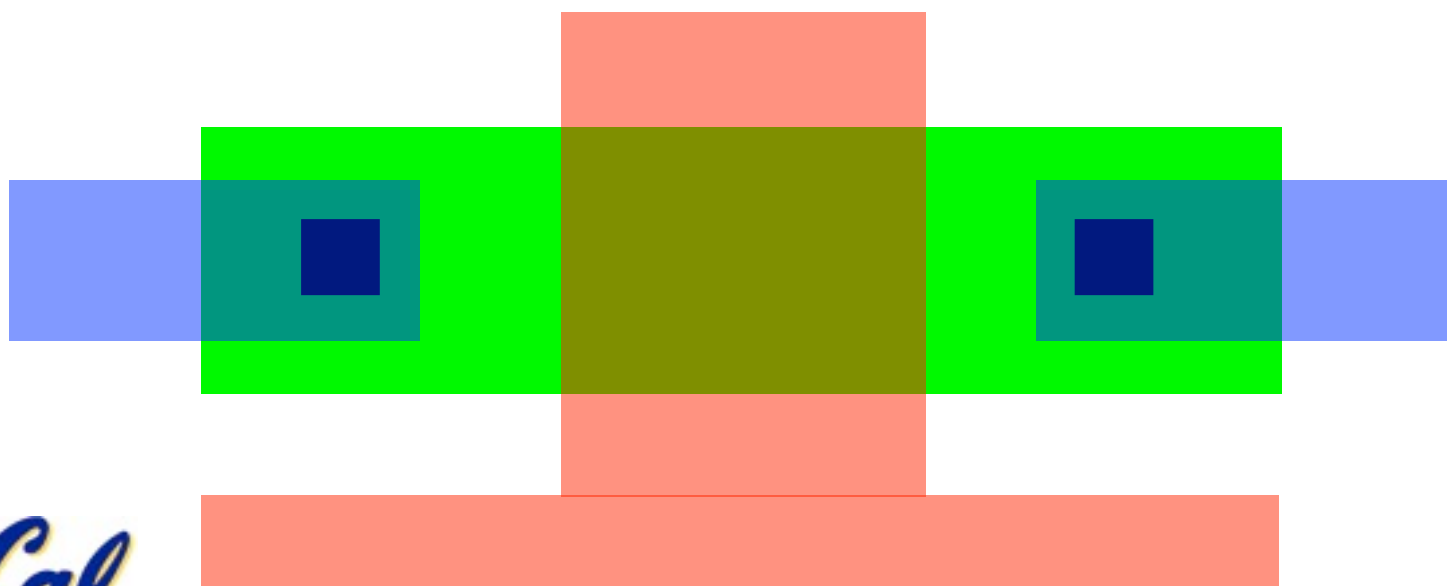
Final product ...

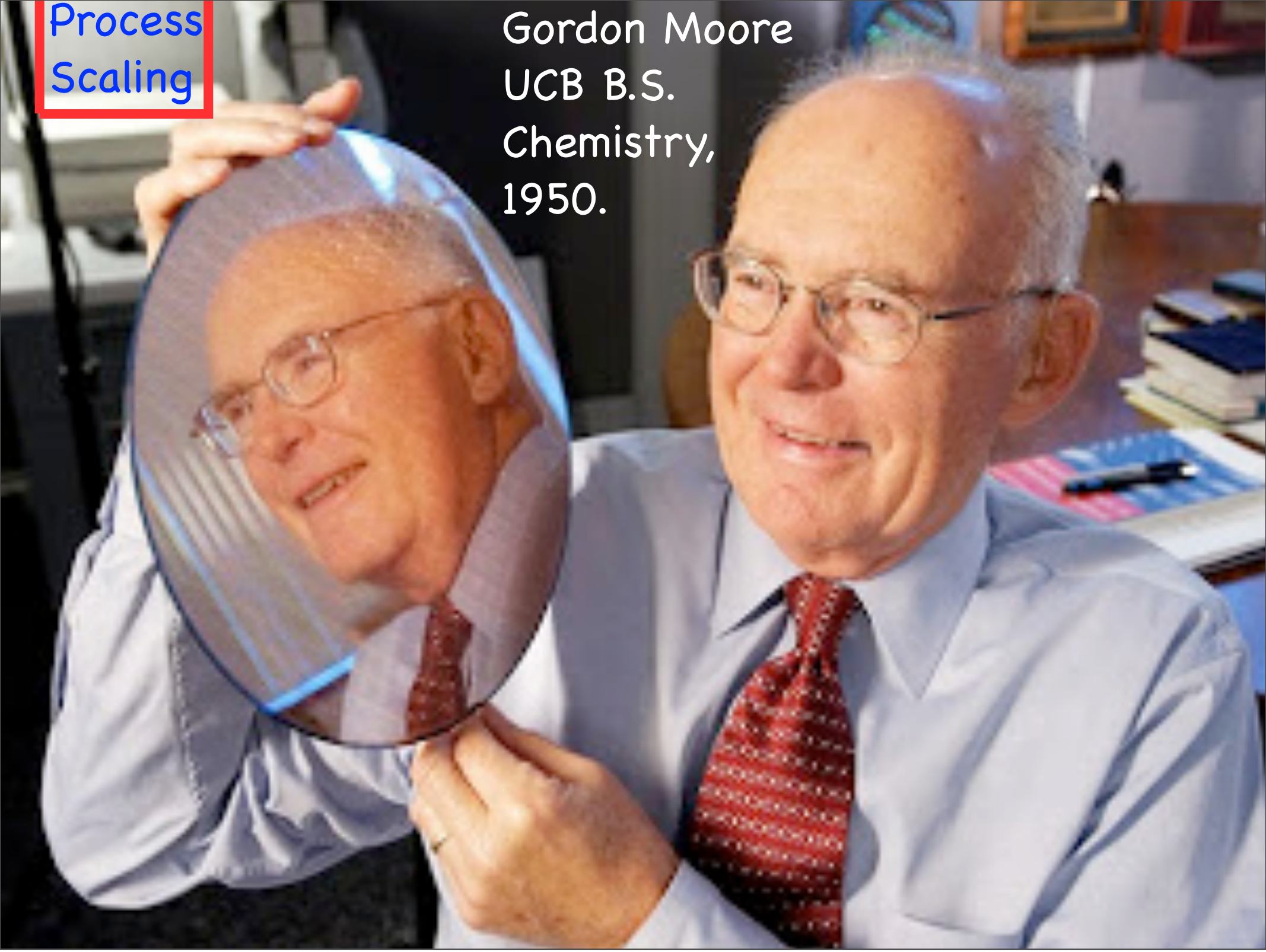


"The planar process"

Jean Hoerni, Fairchild Semiconductor 1958

Top-down view:

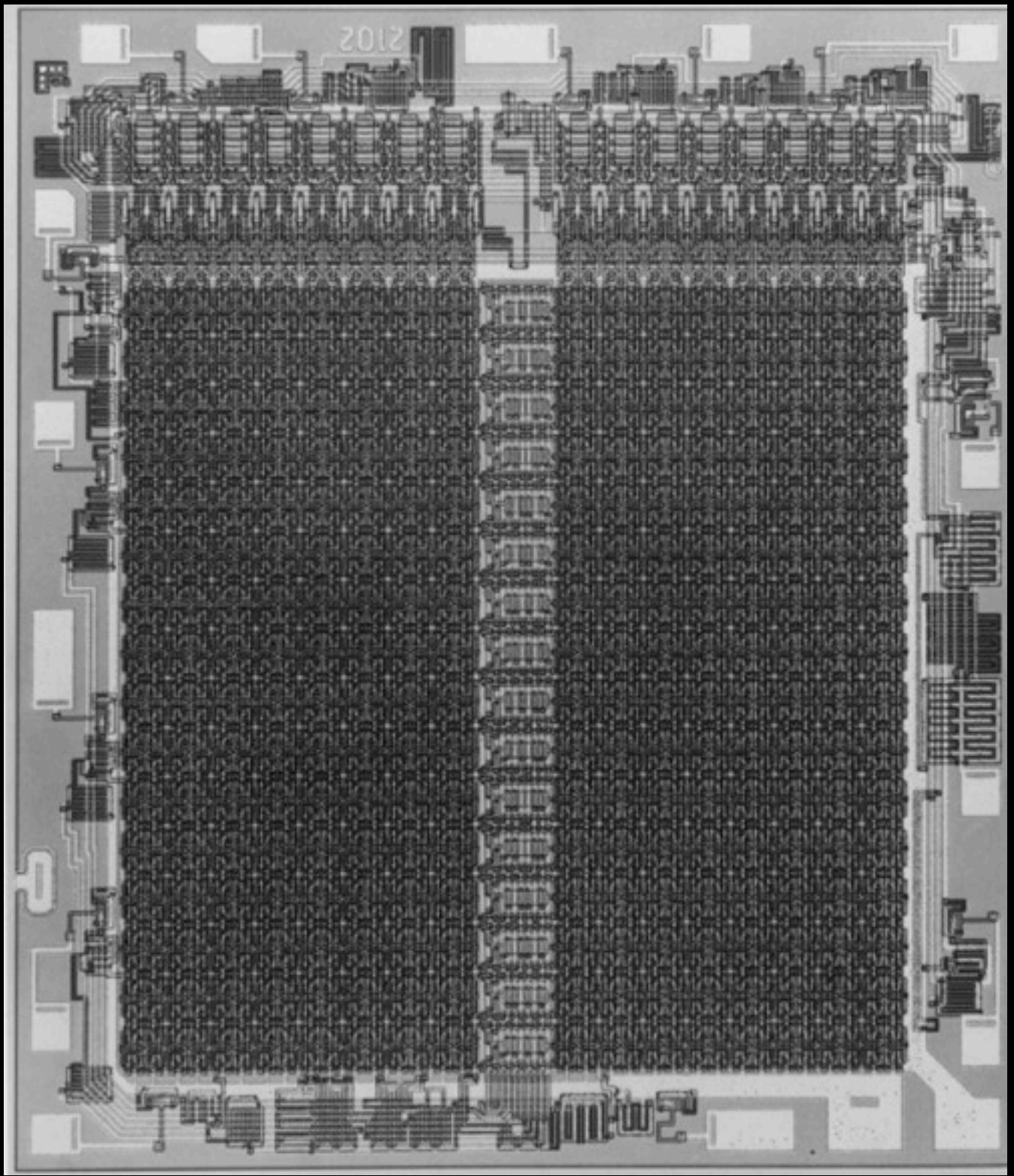



Process Scaling
Gordon Moore
UCB B.S.
Chemistry,
1950.

MOS in the 70s

1971 state of the art.

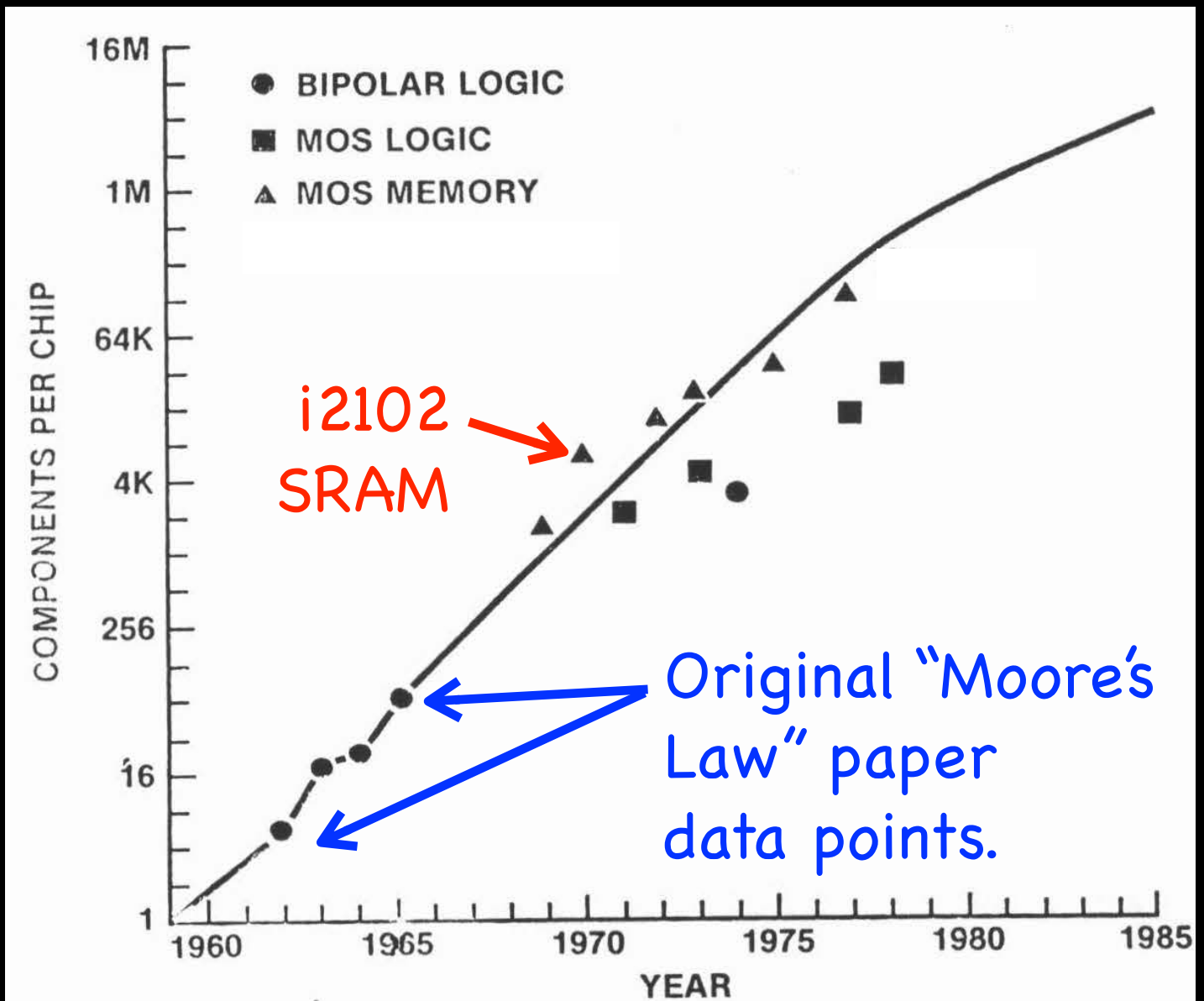
Intel 2102, a 1kb, 1 MHz static RAM chip with 6000 nFETs transistors in a 10 µm process, like the one we just saw.



By 1971, "Moore's Law" paper was already 6 years old ...

But the result was empirical.

Understanding the physics of scaling MOS transistor dimensions was necessary ...



Are We Really Ready for VLSI[2]?
Gordon E. Moore
Intel Corporation

CALTECH CONFERENCE ON VLSI, January 1979

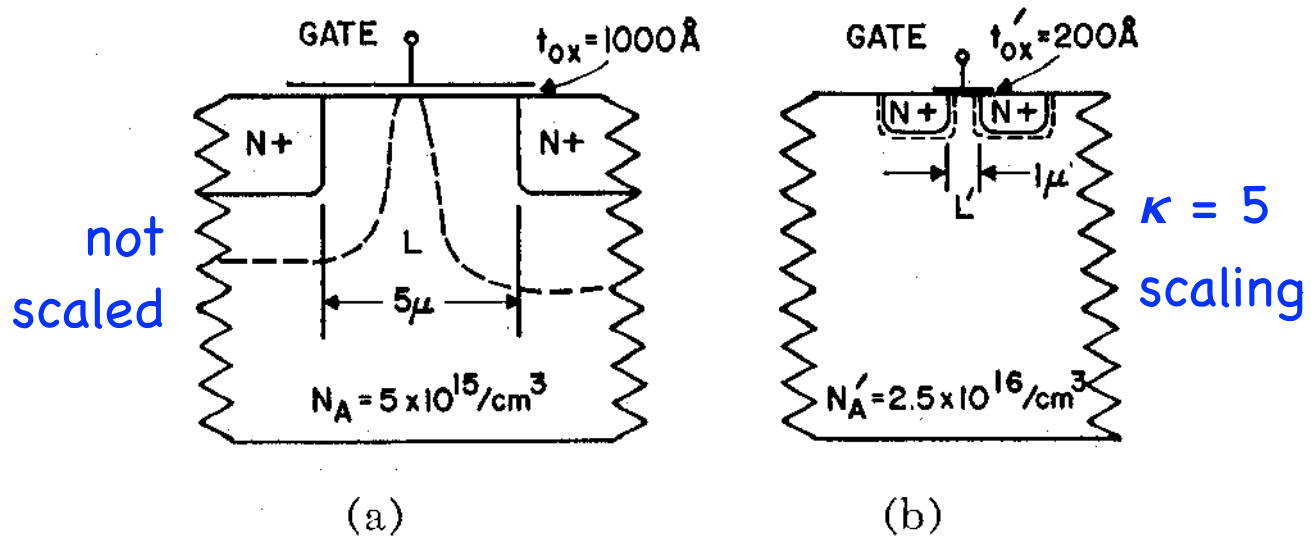
1974: Dennard Scaling

IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. SC-9, NO. 5, OCTOBER 1974

Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions

ROBERT H. DENNARD, MEMBER, IEEE, FRITZ H. GAENSSLEN, HWA-NIEN YU, MEMBER, IEEE, V. LEO RIDEOUT, MEMBER, IEEE, ERNEST BASSOUS, AND ANDRE R. LeBLANC, MEMBER, IEEE

If we scale the gate length by a factor κ, how should we scale other aspects of transistor to get the "best" results?



not scaled

κ = 5 scaling

Fig. 1. Illustration of device scaling principles with $\kappa = 5$. (a) Conventional commercially available device structure. (b) Scaled-down device structure.

Dennard Scaling

Things we do: scale dimensions, doping, Vdd.

What we get: κ^2 as many transistors at the same power density!

Whose gates switch κ times faster!

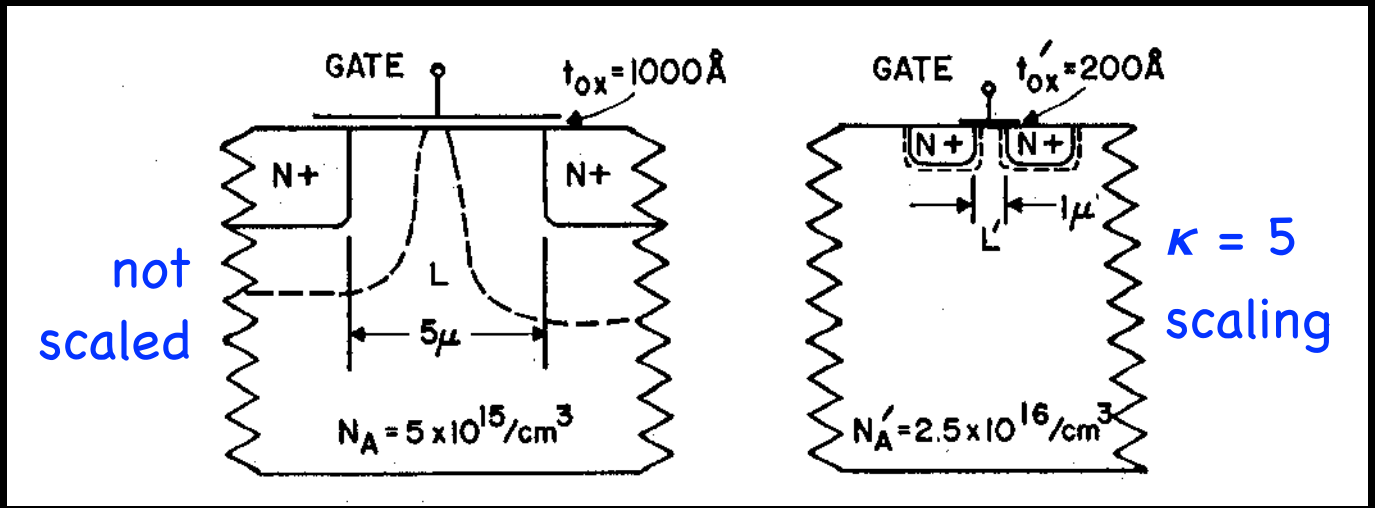


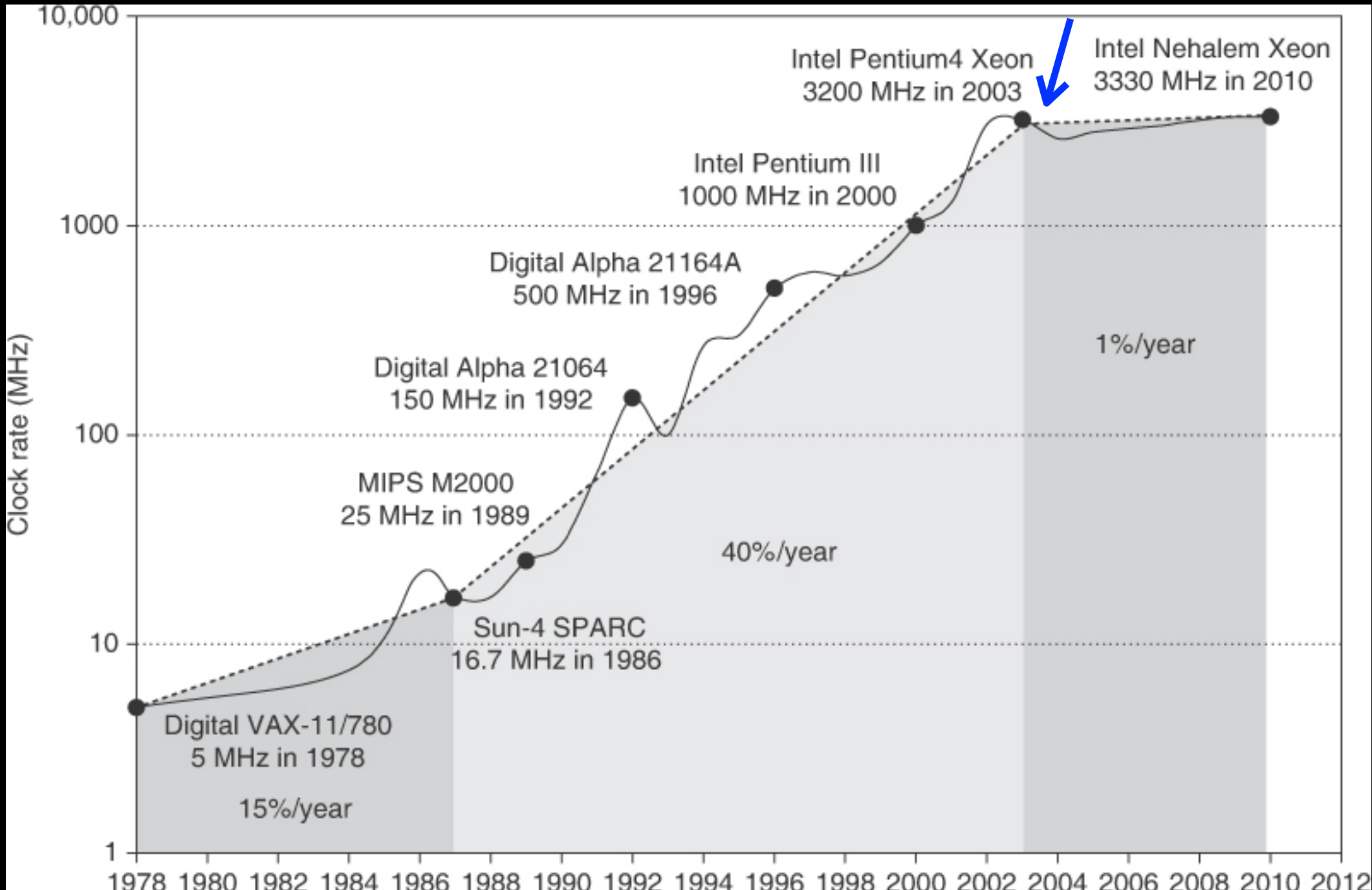
TABLE I

SCALING RESULTS FOR CIRCUIT PERFORMANCE

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/\kappa$
Doping concentration N_a	κ
Voltage V	$1/\kappa$
Current I	$1/\kappa$
Capacitance $\epsilon A/t$	$1/\kappa$
Delay time/circuit VC/I	$1/\kappa$
Power dissipation/circuit VI	$1/\kappa^2$
Power density VI/A	1

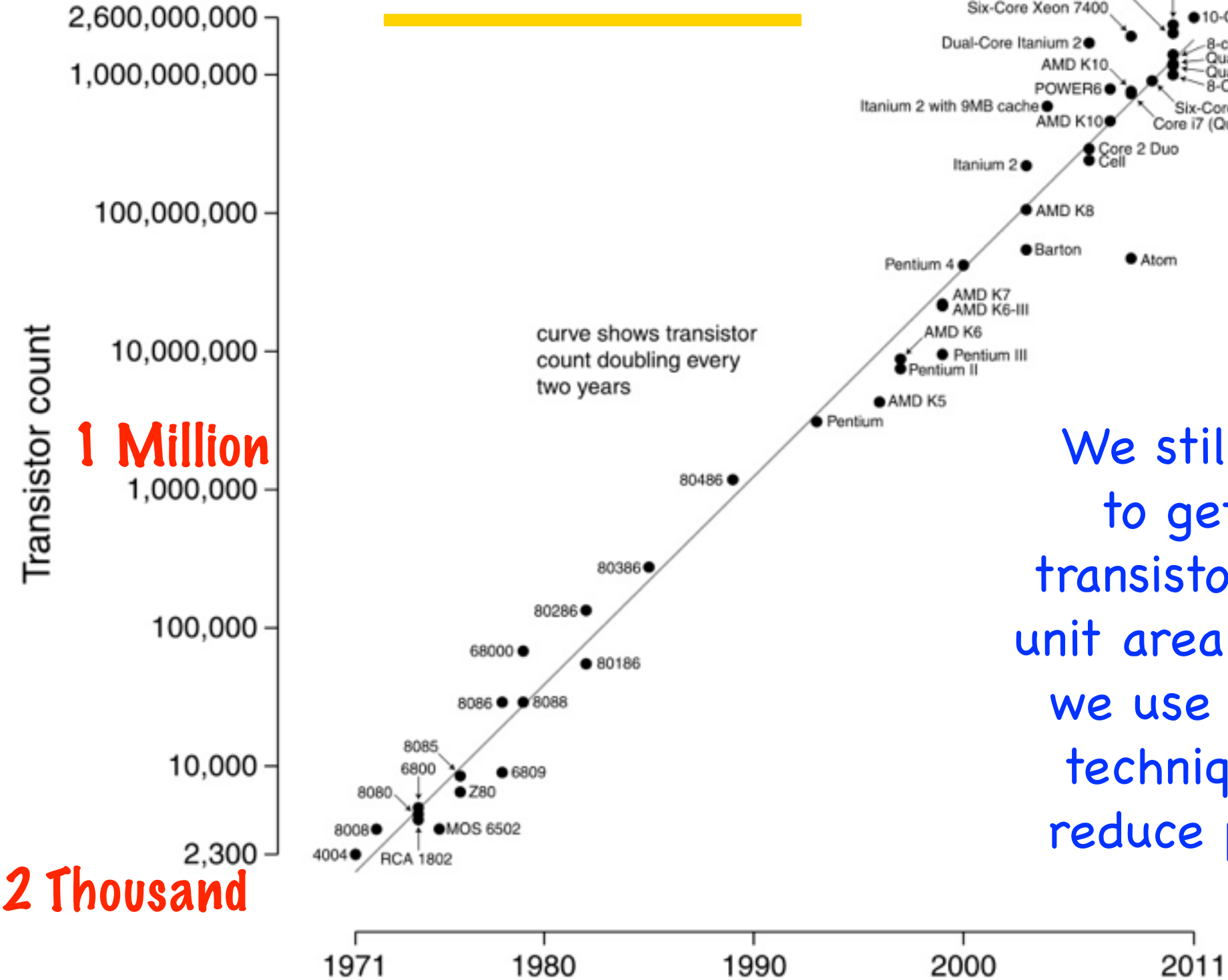
Power density scaling ended in 2003 (Pentium 4: 3.2GHz, 82W, 55M FETs). We could no longer scale Vdd (Lec 4).

Dennard Scaling ended … when we hit the "power wall"



Moore's Law

2.6 Billion



1 Million

2 Thousand

We still scale to get more transistors per unit area ... but we use design techniques to reduce power.

Photo: Global Foundries fab floor, before equipment arrives.

Design Styles

Chip Designers: Head in the Clouds

Process Design Kit: Design Rules and Device Models

IC Process Designers: Feet on the Ground

Photo: Global Foundries fab floor, before equipment arrives.

Structured Custom Design

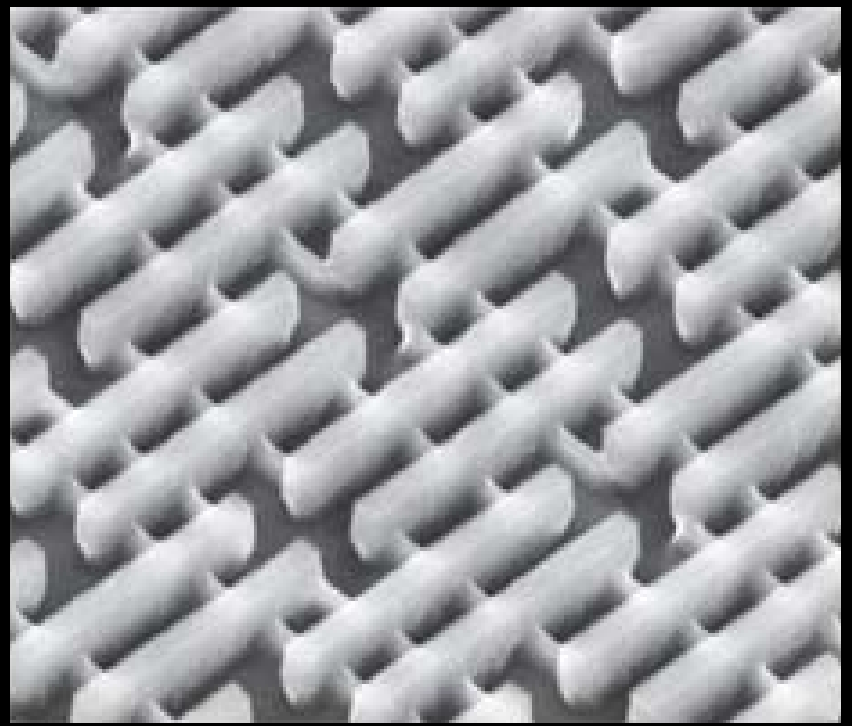
- Wiring by abutment. Rectangular leaf cell layout is hand-crafted so that edge wires "match up" when cells are tiled in 1-D or 2-D.
- Cell compilation. Designers write programs ("cell compilers") to tile leaf cells into larger logic blocks. Wire routing comes "for free".
- Parameters. N-bit datapath compilers.

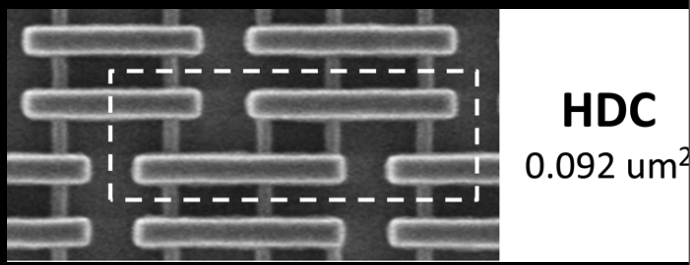
Process Design Kit: Design Rules and Device Models

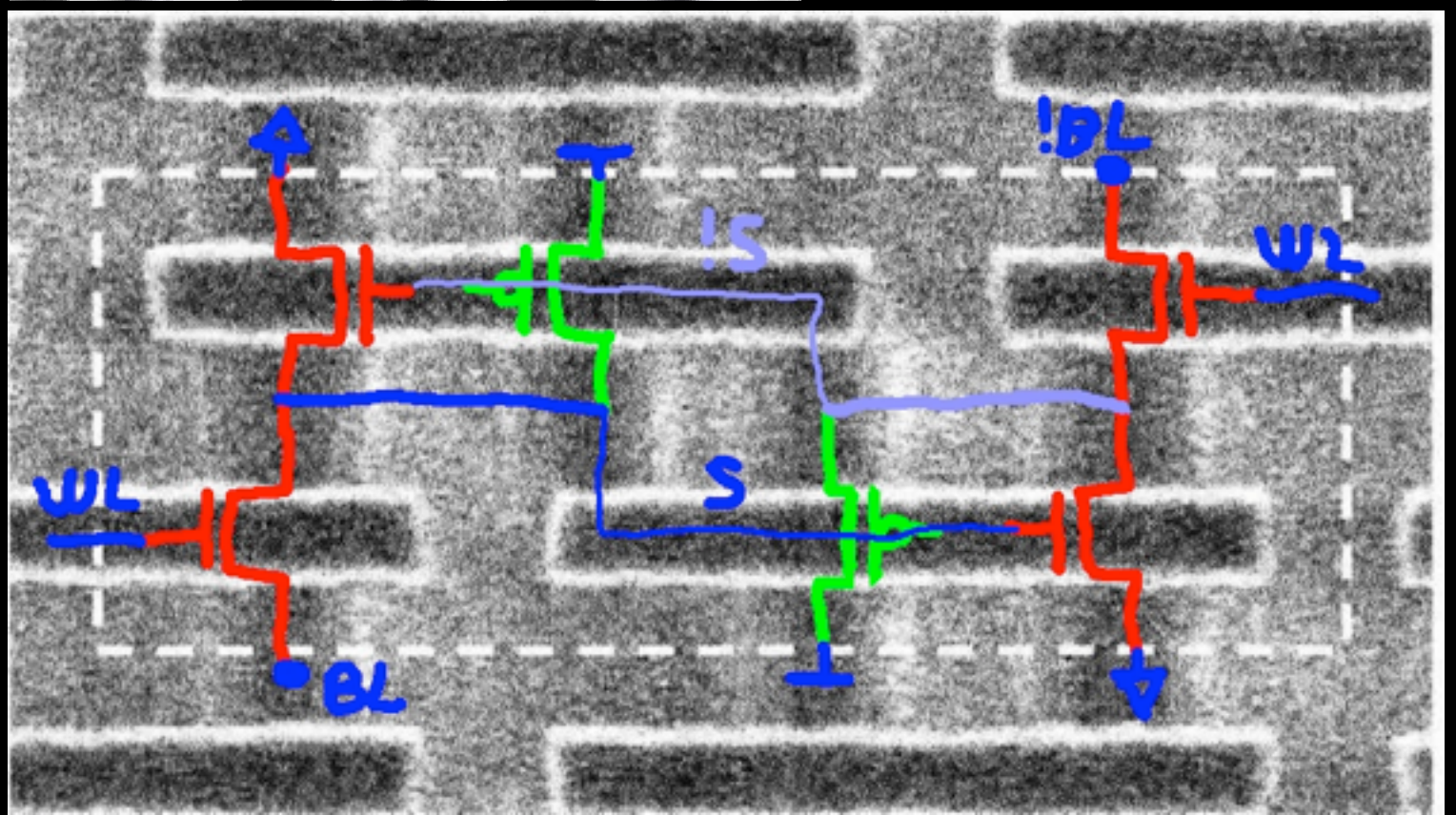
RAM Compilers

On average, 30% of a modern logic chip is SRAM, which is generated by RAM compilers.

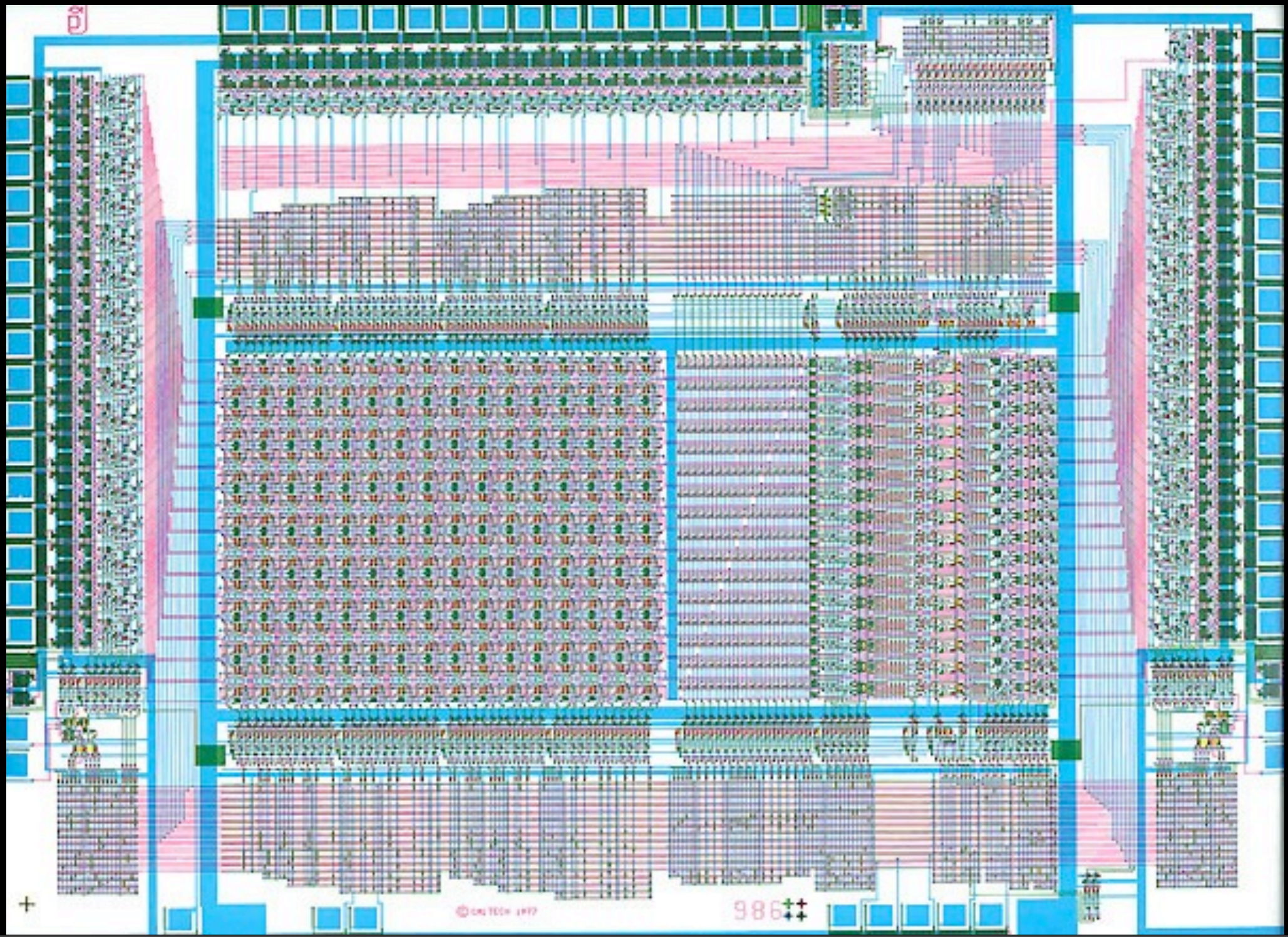
Compile-time parameters set number of bits, aspect ratio, ports, etc.

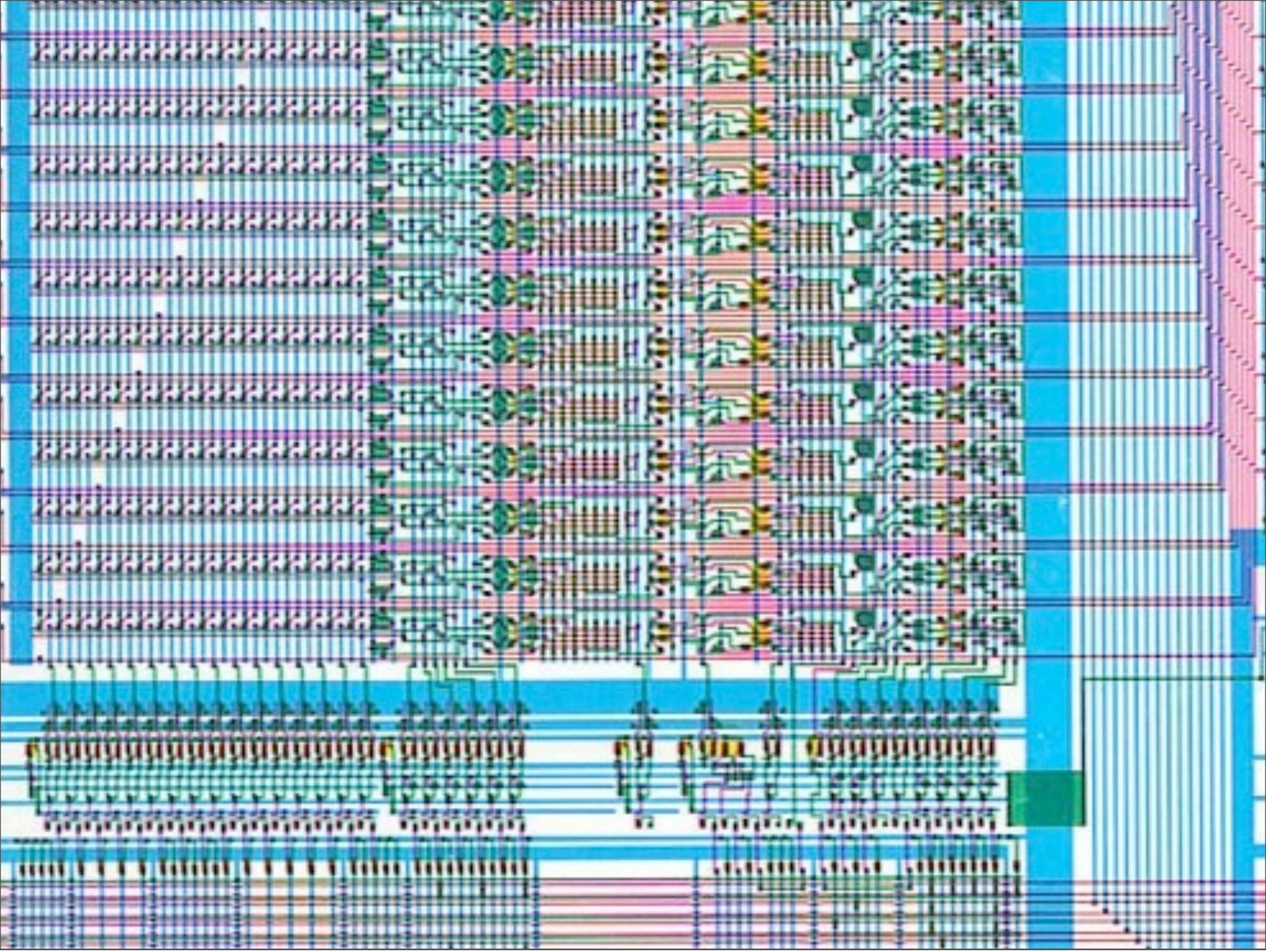






"Structured Custom" CPU (David Johannsen, Caltech, '77)





Limitations

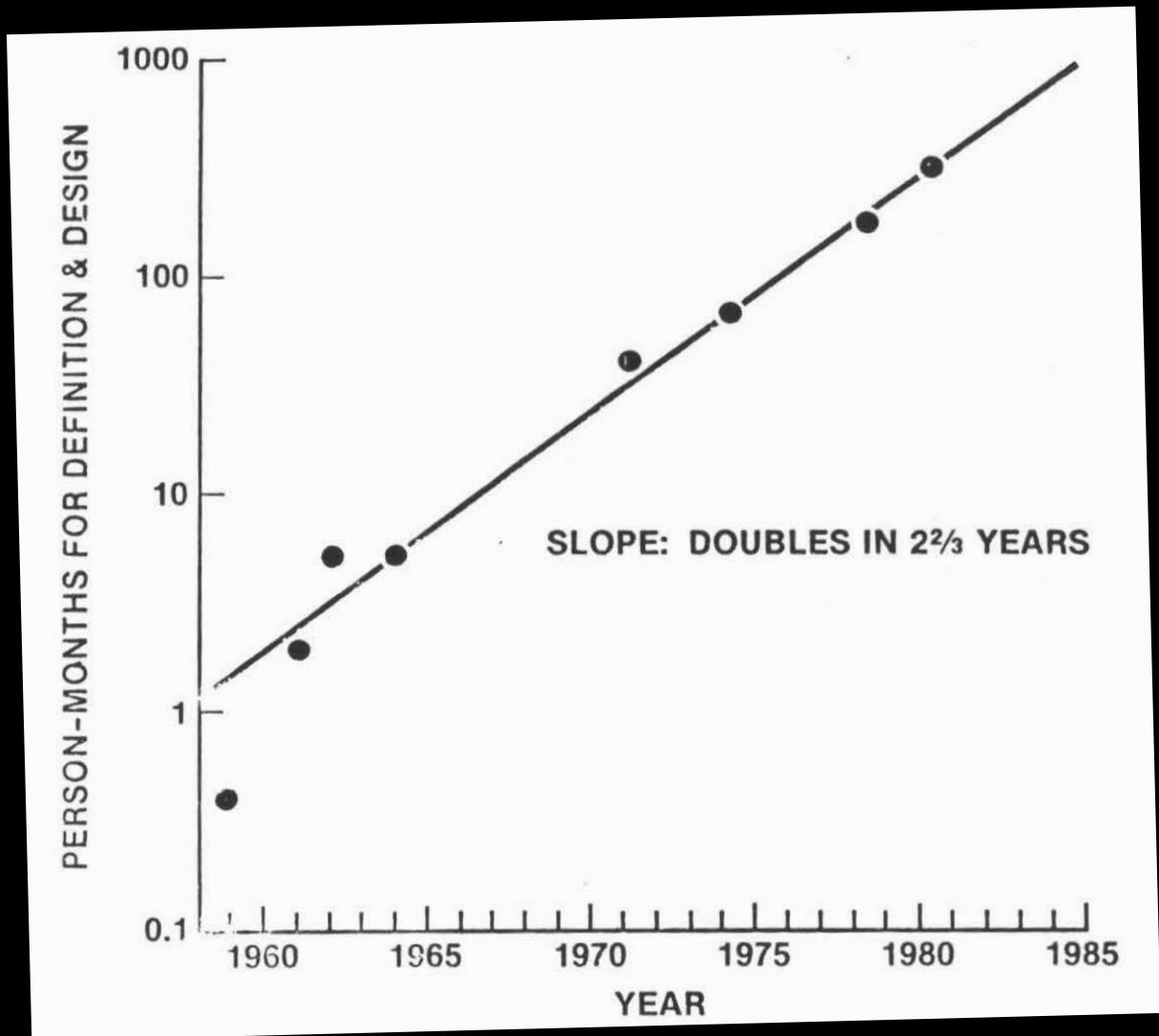
Labor intensive.

Key metric is the number of leaf cells required to efficiently use a given area of silicon

Memory arrays and FPGAs are a good fit.

Still, even today it is common to see custom layout in critical parts of CPU logic.

PERSON-MONTHS FOR DEFINITION & DESIGN



CALTECH CONFERENCE ON VLSI, January 1979

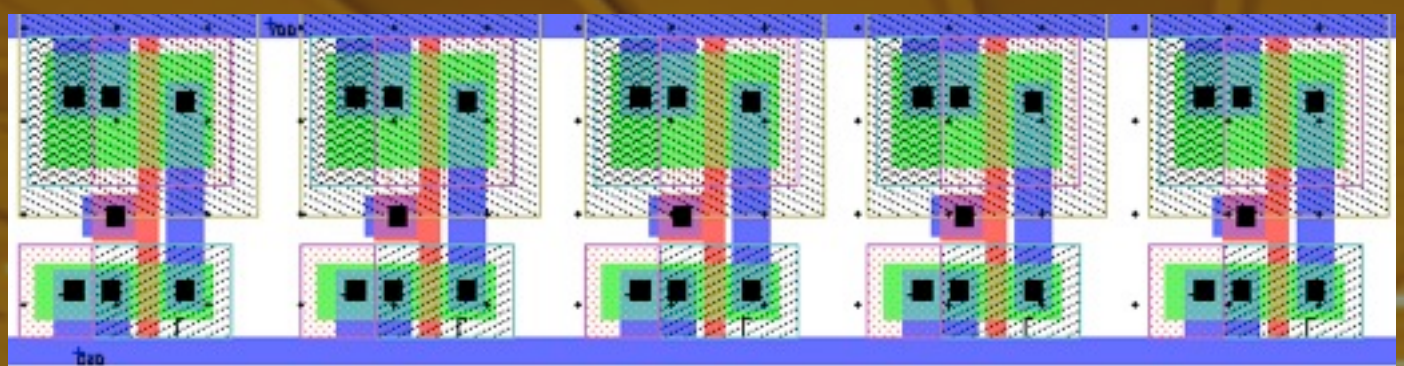
Are We Really Ready for VLSI[2]?

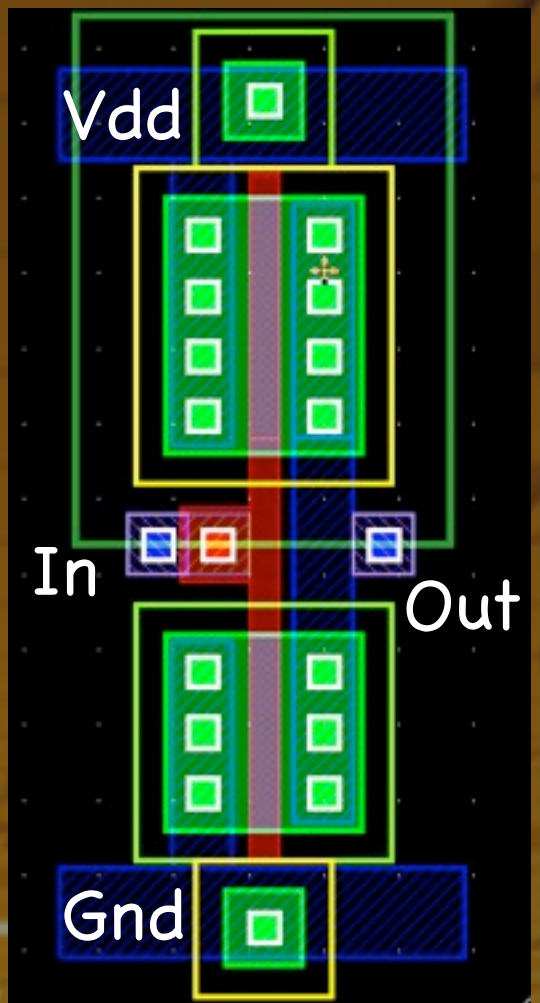
Gordon E. Moore
Intel Corporation

Standard Cell Design

Logic schematics using library gates.

- Gate Library. Fixed-height, to be placed in rows. Vdd and Gnd rails connect by abutment.
- I/O Ports. Auto-router places wires over cell to connect them.





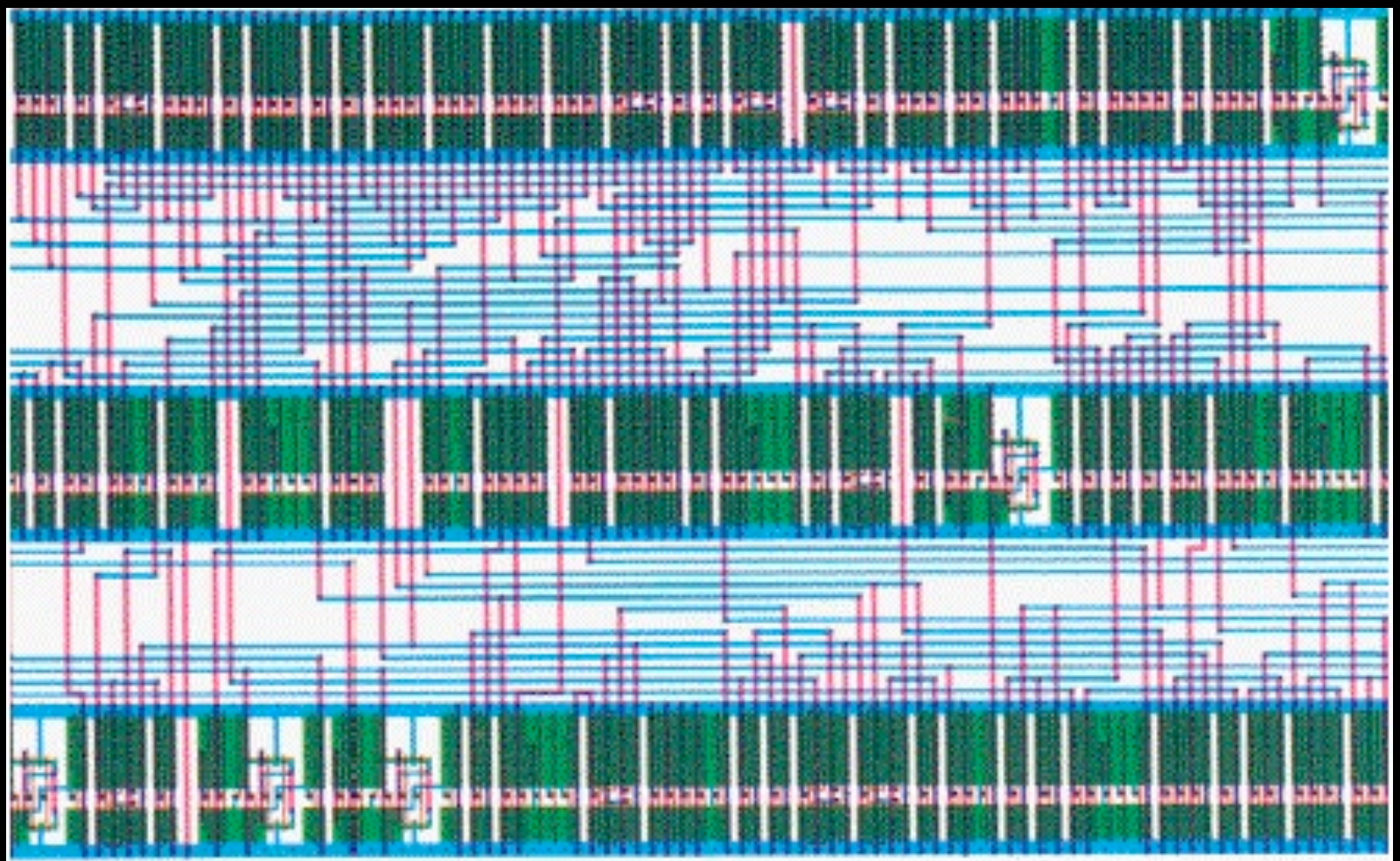
Process Design Kit: Design Rules and Device Models

Place & Route

Router connects gate ports to match schematic.

Router "optimizes" relative lengths of wire to meet constraints.

Software places cells into rows, to "optimize" area, performance, and power constraints.

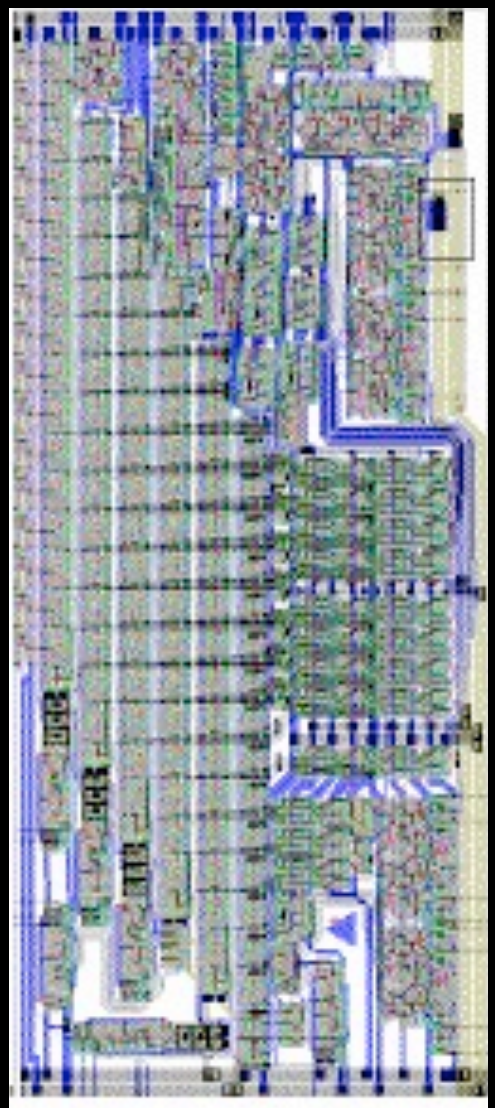


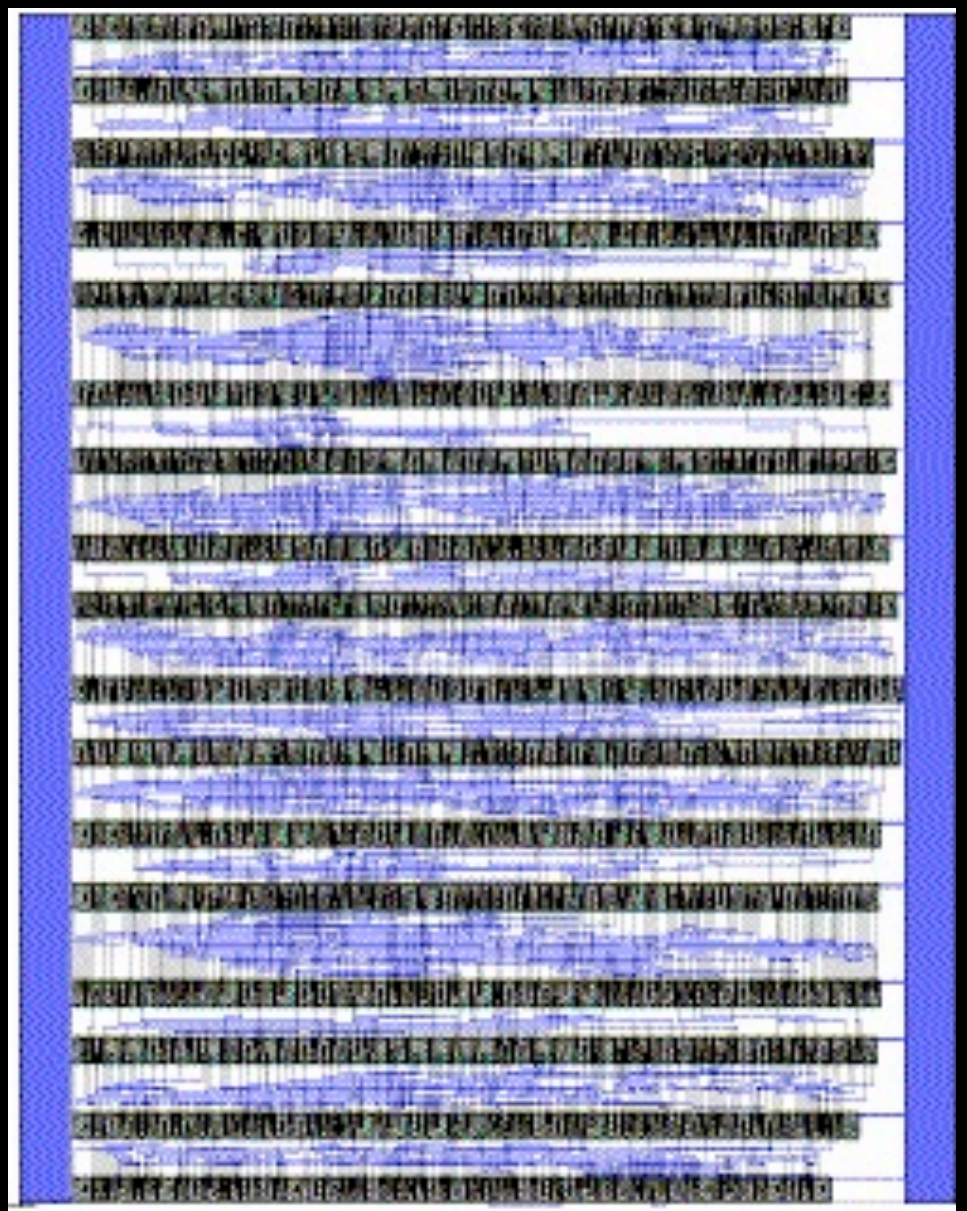
We put "optimize" in quotes to reflect the NP-hard nature of the algorithms behind place & route.

12 x 16 Multiplier in Structured Custom and Standard Cell

Benchmark by a custom design house (Obsidian).

In general, they claim: "30% of the power, twice the speed, and 4 times the density of standard cells".





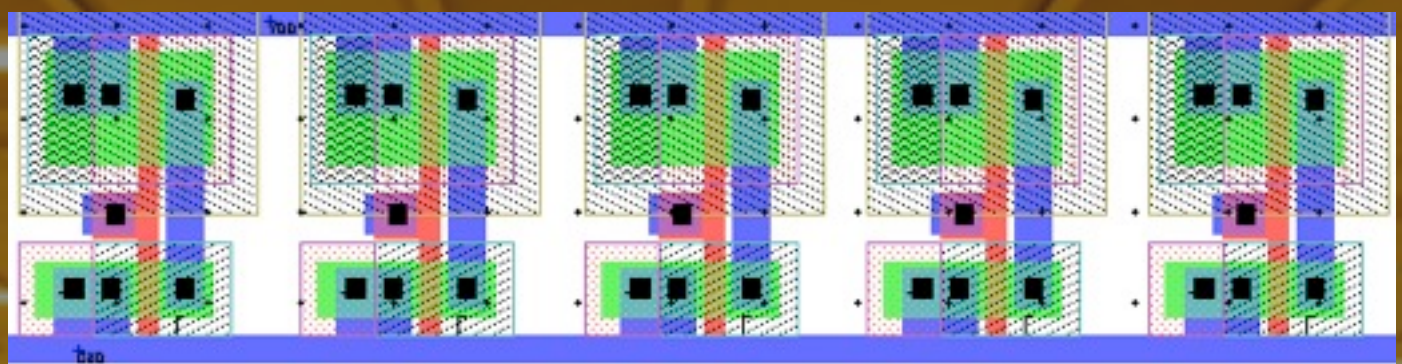
Custom layout (left) is a factor of 2.2 smaller than standard cell layout (right).

Logic Synthesis - The Automation of Logic Design

At the start of the 1980s, the standard-cell flow was driven by hand-drawn schematics.

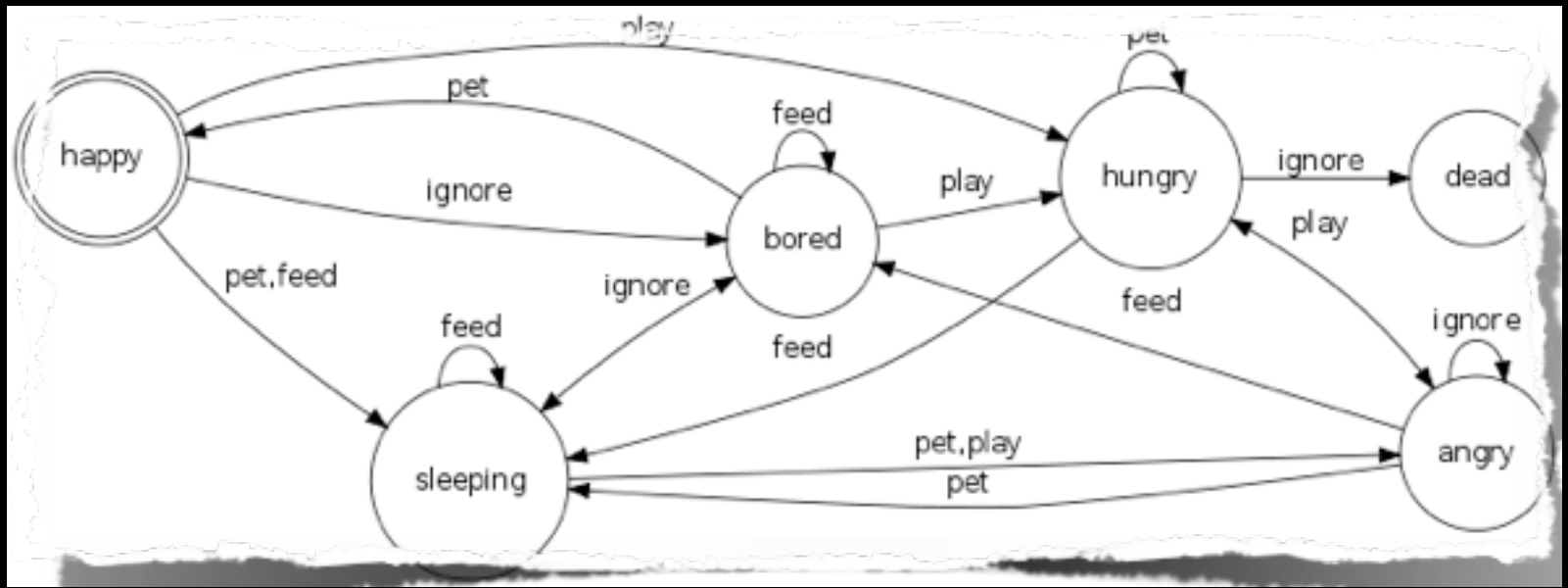
By the early 1990s, schematics were replaced with Verilog/VHDL, to drive logic synthesis, whose output was integrated into standard cell back ends.

Place & Route



Process Design Kit: Design Rules and Device Models

Logic design … as I learned it in 1981 …



$f = m_1 + m_2 + m_5 + m_6 + m_7$

$= x'y'z + x'yz' + xy'z + xyz' + xyz$ FUNCTION

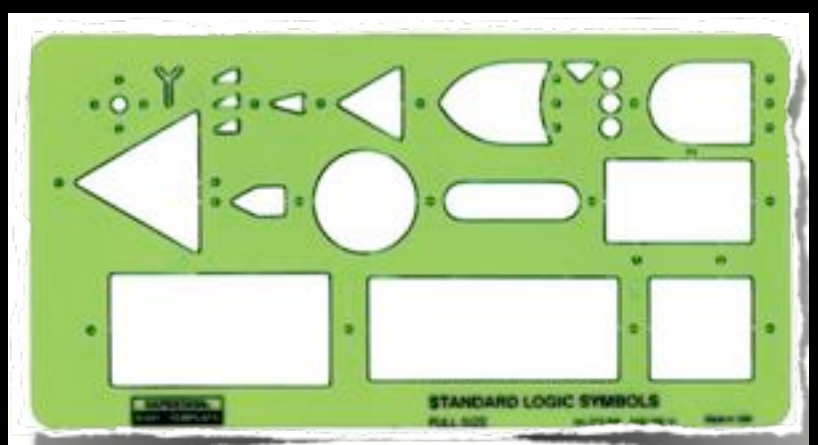
$= x(y'z + yz') + x(y'z + yz') + xyz$ DISTRIBUTIVE

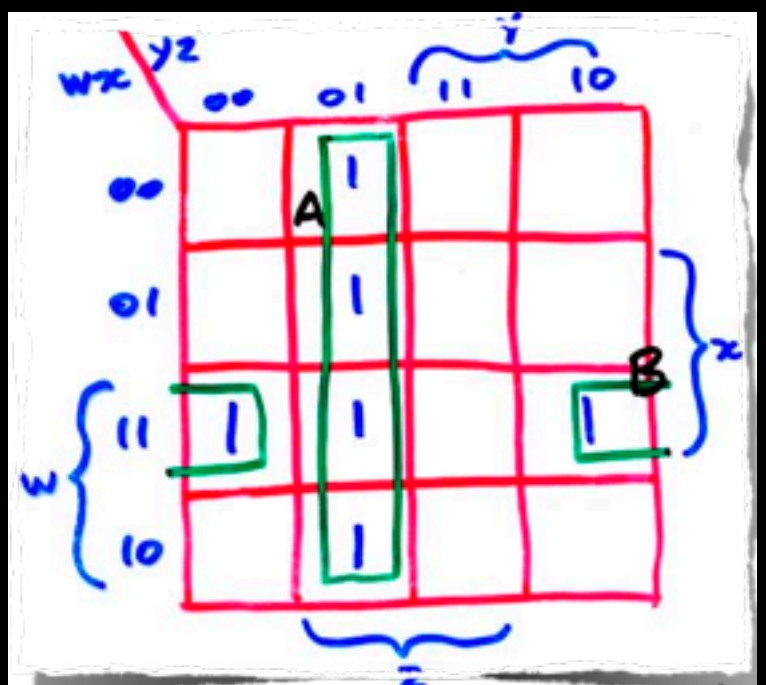
(1, 2) (5, 6)

$= (x' + x)(y'z + yz') + xyz$ DISTRIBUTIVE

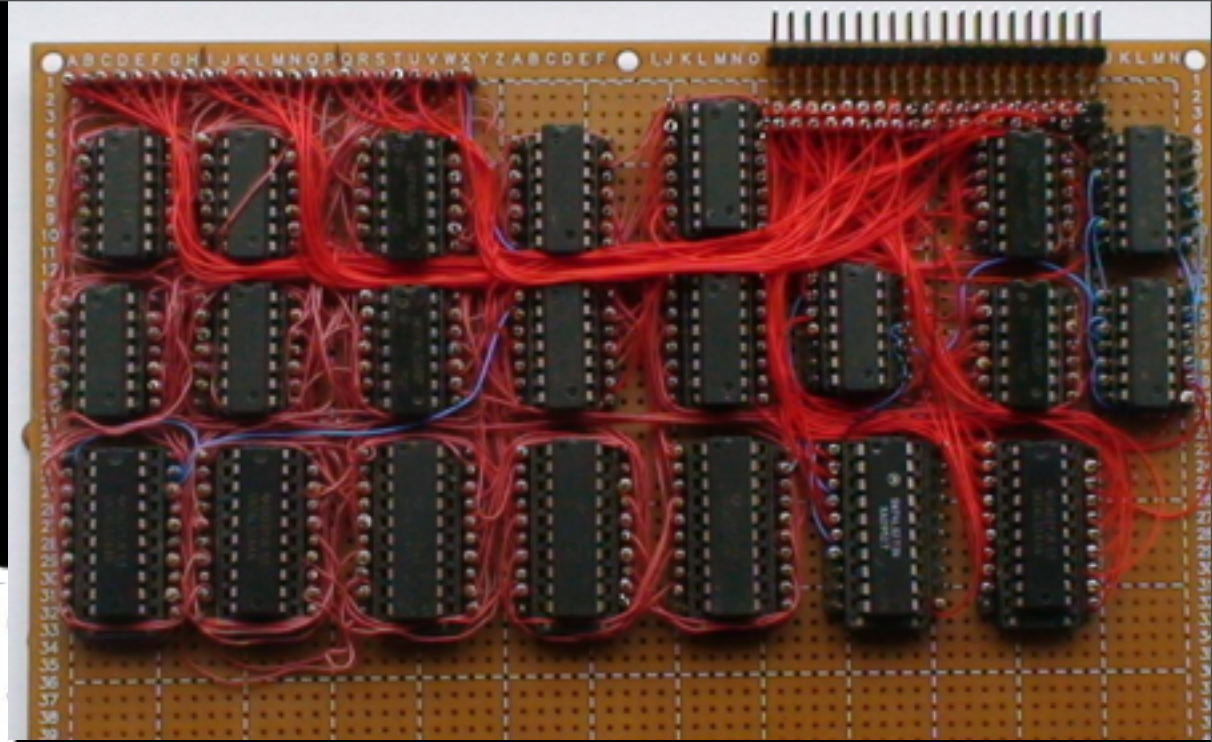
$= 1 \; (y'z + yz') + xyz$ USE OF COMPLEMENTS

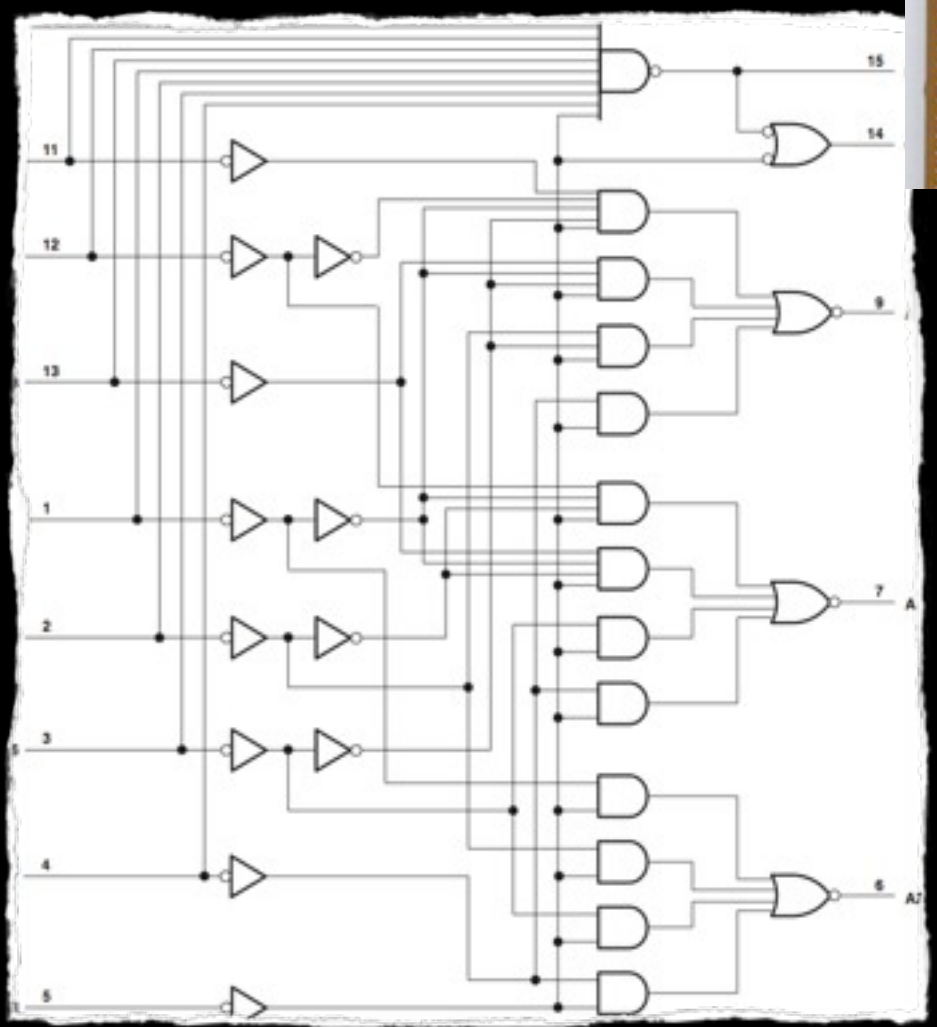
$= y'z + yz' + xyz$ USE OF IDENTITY ELEMENT

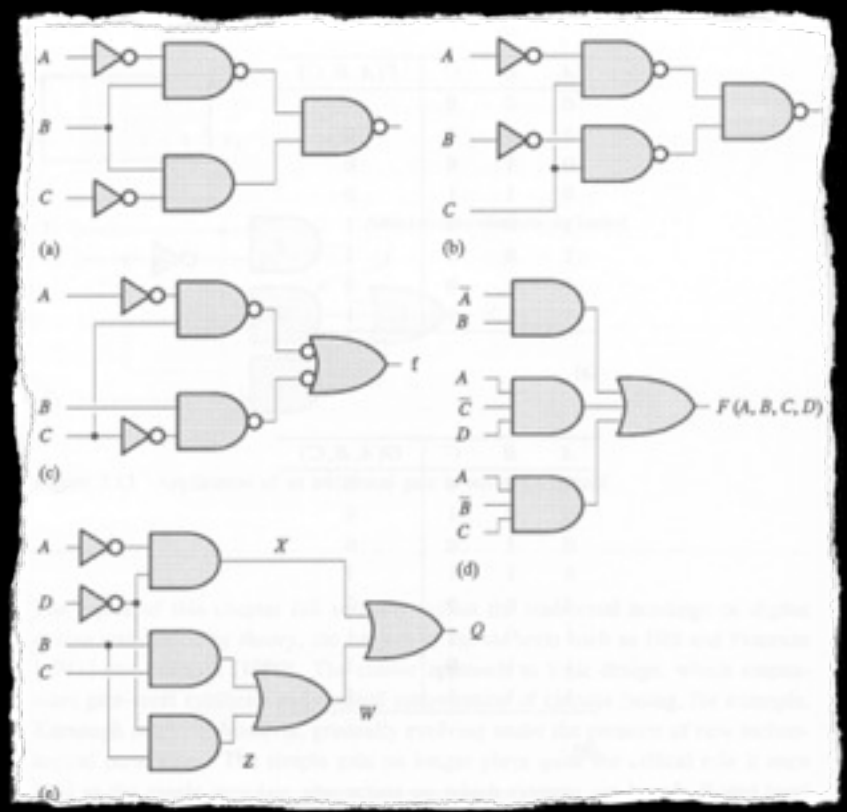




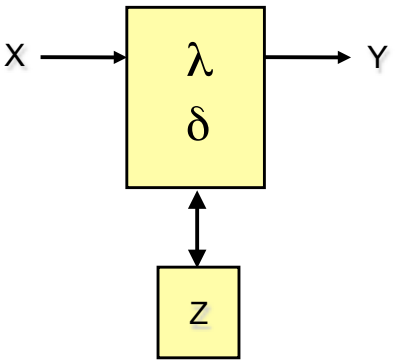
There was one computer in the logic design lab at Penn EE in 1981 … used for "burning" fuse ROMs …





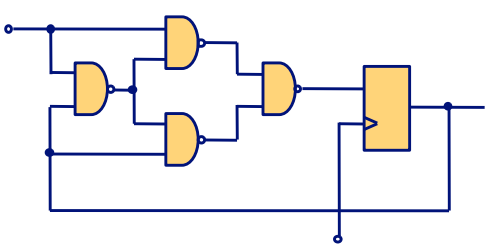


In the early 1980s, progress in academia and industrial labs made the problem domain tractable …



Given: Finite-State Machine F(X,Y,Z, λ, δ) where:

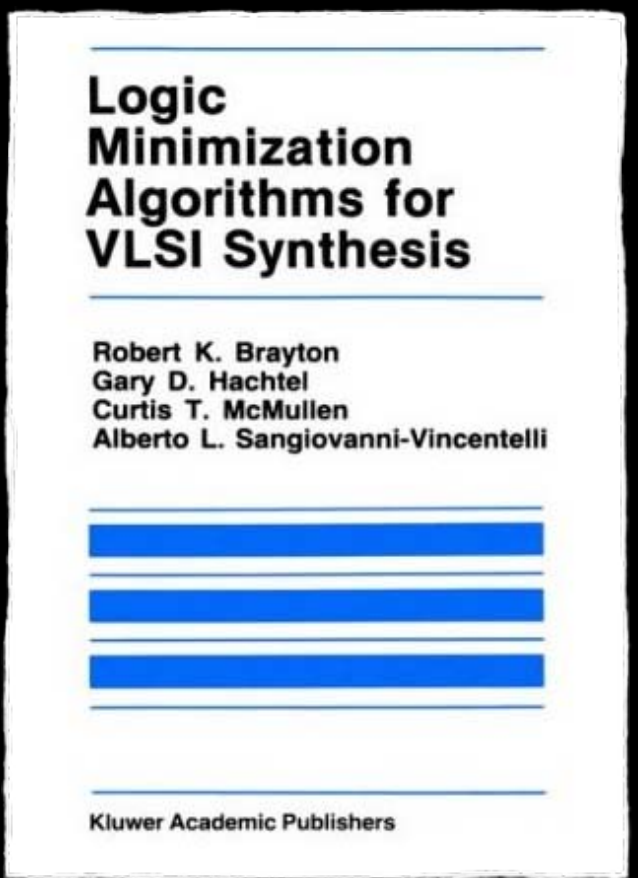
X: Input alphabet
Y: Output alphabet
Z: Set of internal states
: X x Z → Z (next state function)
: X x Z → Y (output function)

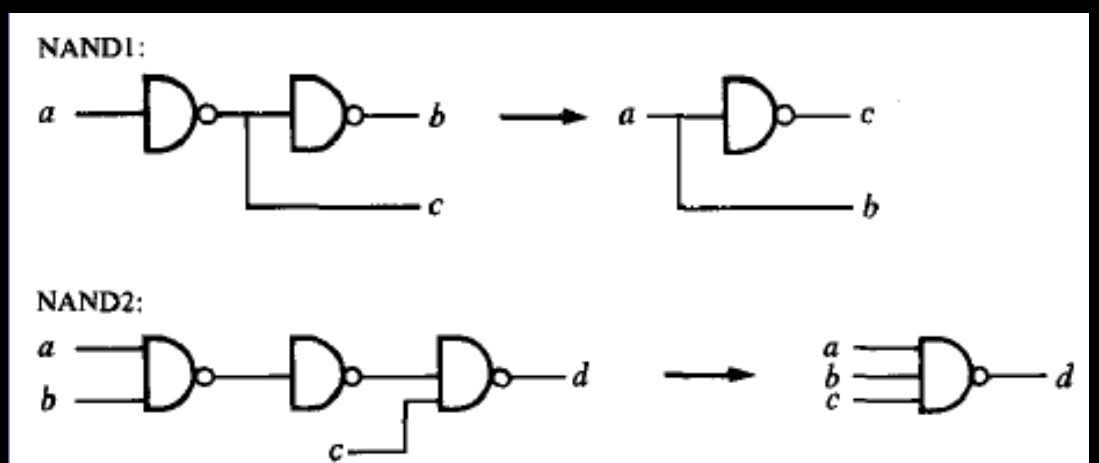


Target: Circuit C(G, W) where:

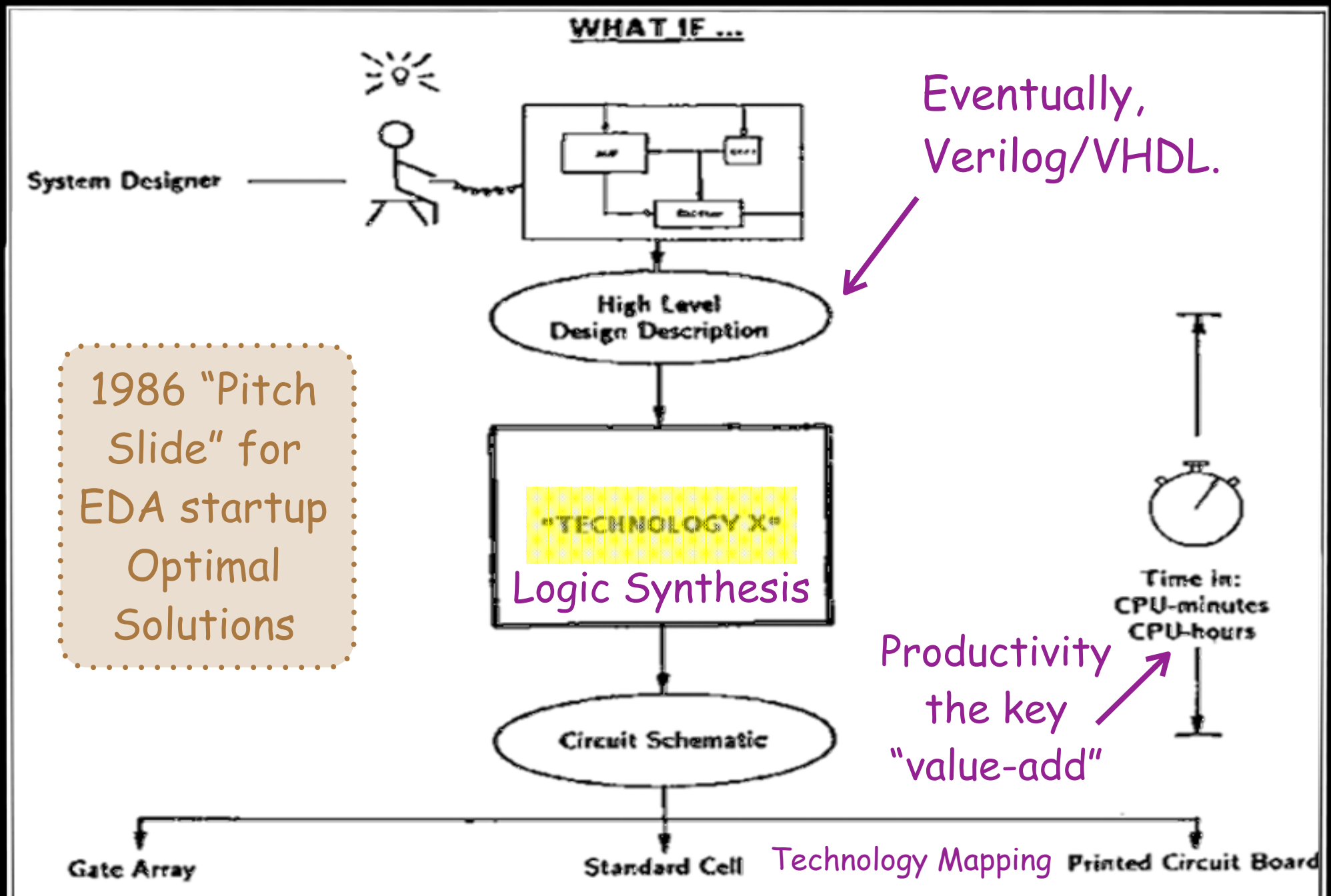
G: set of circuit components g∈{Boolean gates, flip-flops, etc}

W: set of wires connecting G





In the second half of the 1980s, the startup that became Synopsys developed Design Compiler (dc) …



Modern ASIC Methodology and Flow

- **RTL Synthesis Based**

HDL specifies design as combinational logic + state elements

Instantiations needed for blocks not inferred by synthesis (typically RAM)

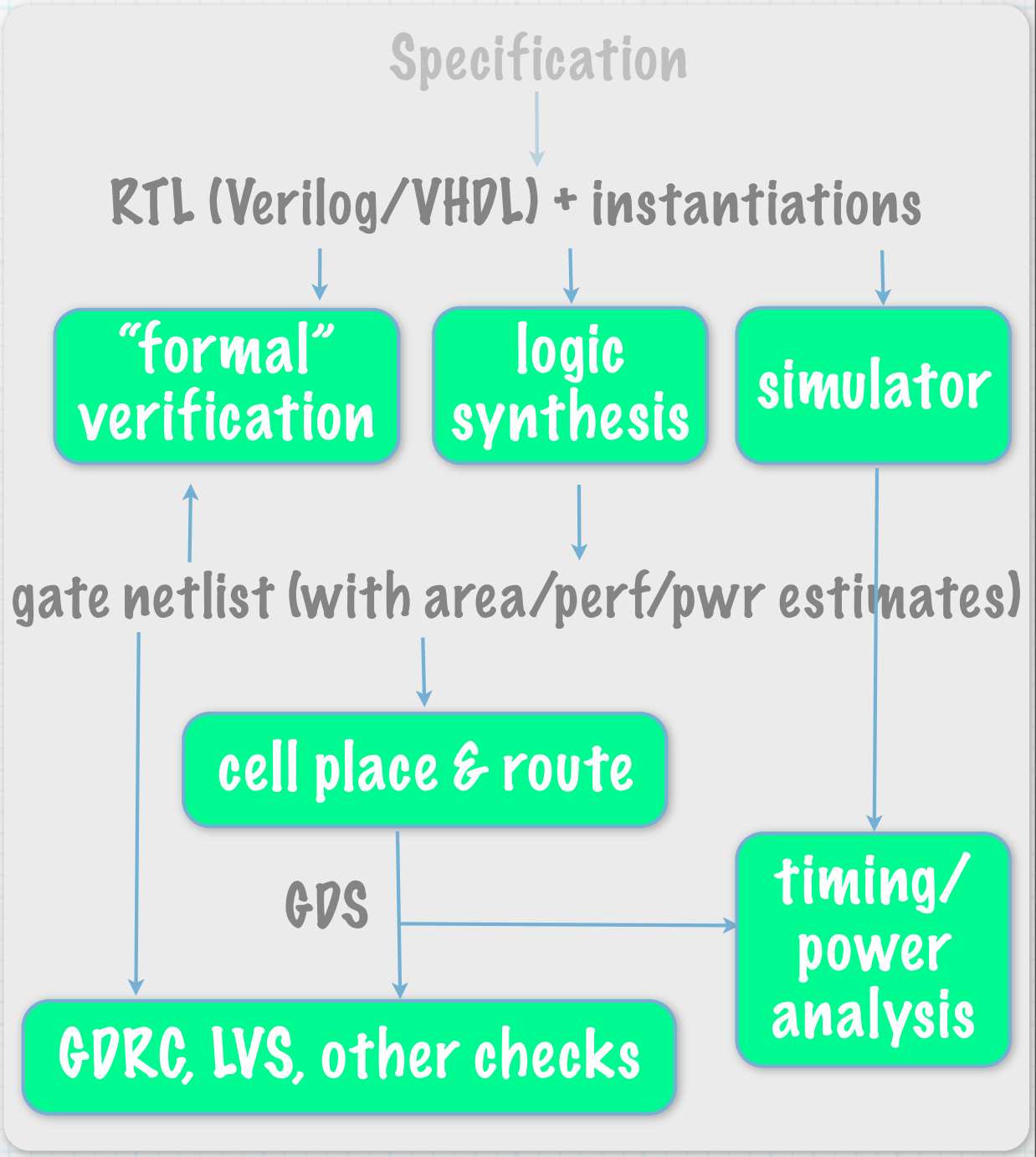
Event simulation verifies RTL

"Formal" verification compares logical structure of gate netlist to RTL

Place & route generates layout

Timing and power checked statically

Layout verified with LVS and GDRC



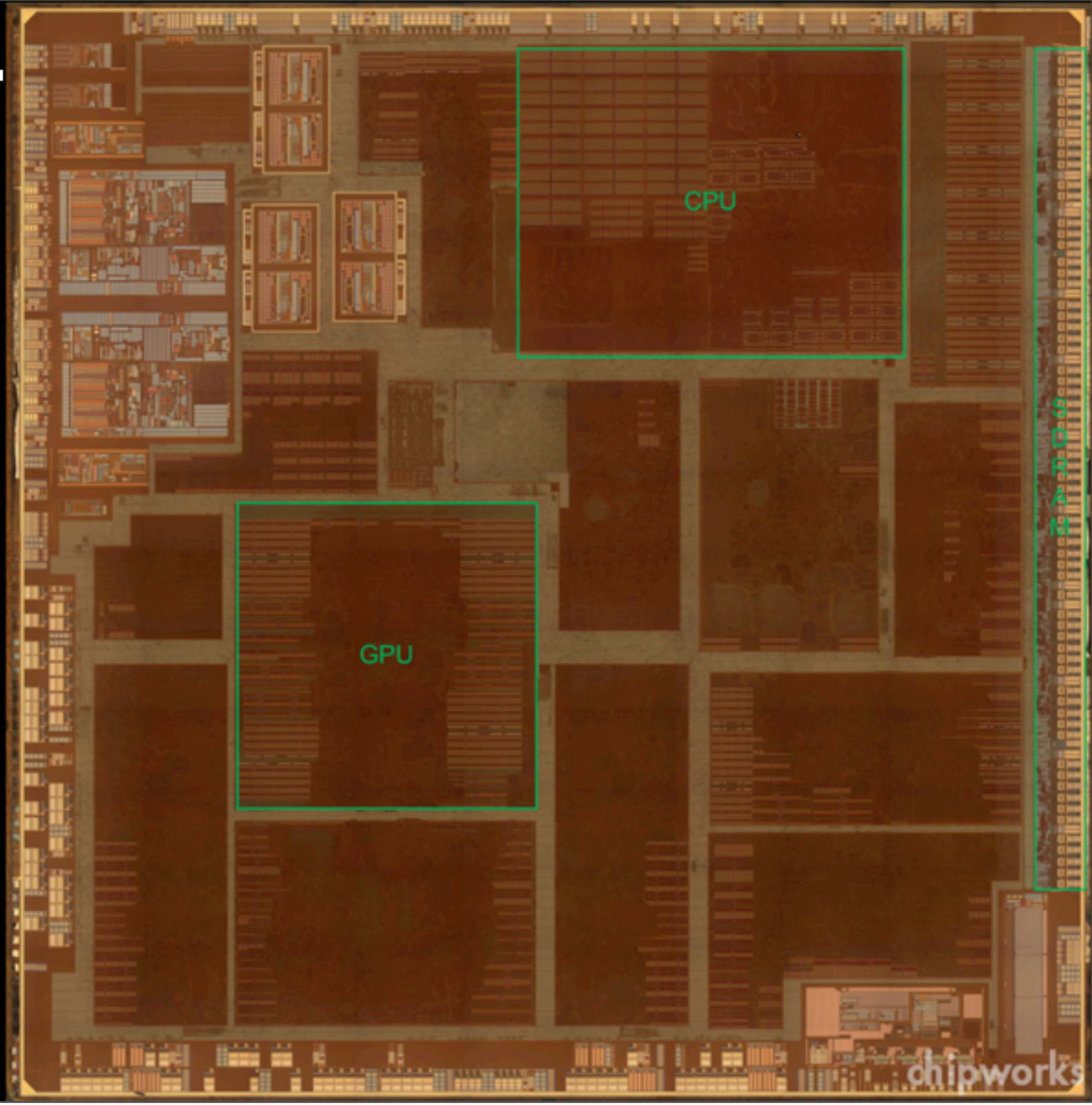
Systems on a Chip (SoCs)

- Today's chips are mosaics. The chip design process often consists of licensing "intellectual property (IP)" from other companies (large like CPUs and GPUs, small like DRAM controllers & analog blocks).
- On chip buses. IP blocks are often designed to hook up to standardized on-chip buses, defined by CPU IP vendors like ARM.

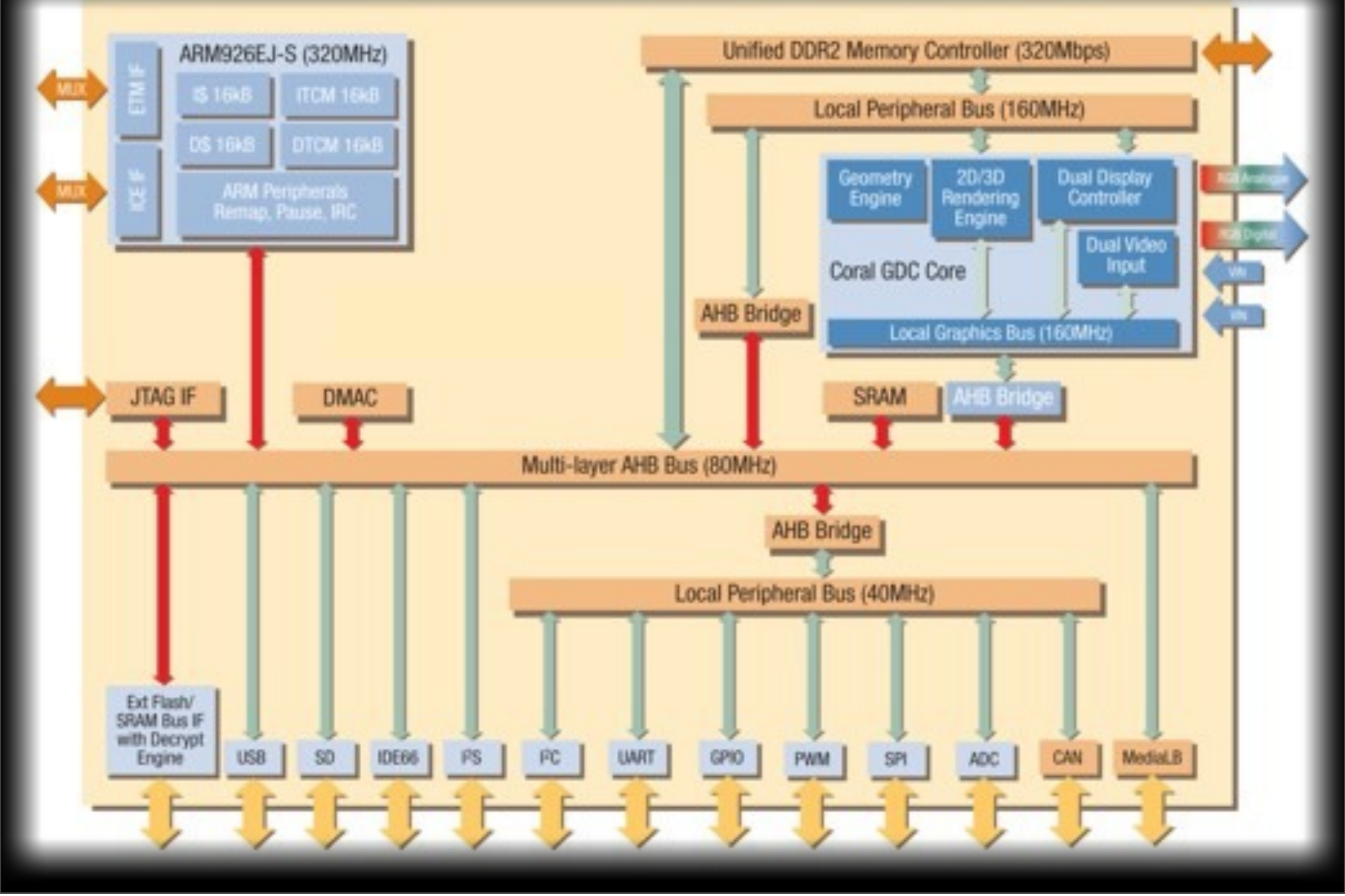
Process Design Kit: Design Rules and Device Models

Apple TV SoC

Chip designed by Apple, but many blocks are licensed from third parties. Some are standard cells, others full custom.



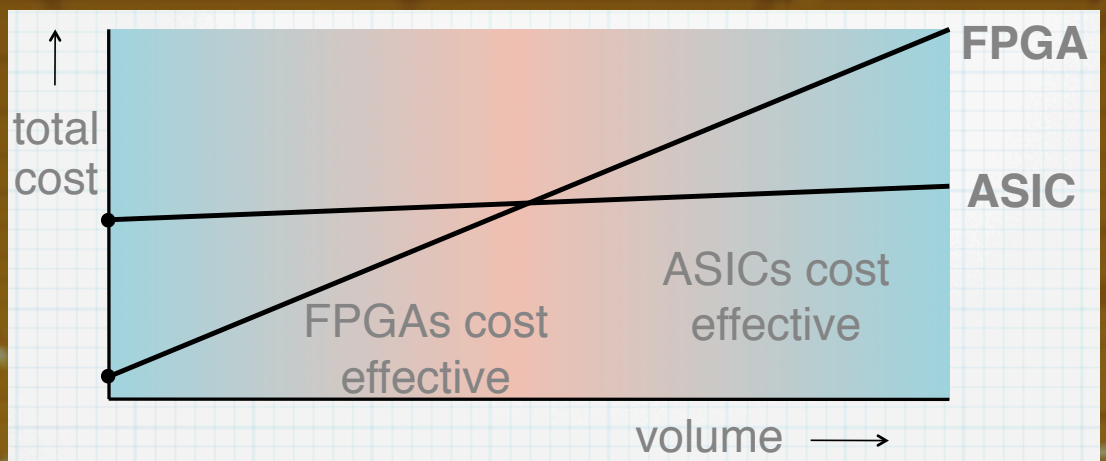
On-chip bus hierarchy for an ARM-based system …



The Programmable Imperative

Instead of doing your own chip, buy a standard-product chip that is programmable in ways more sophisticated than a PC. Examples: FPGAs (Field Programmable Gate Arrays), specialized CPU-based chips.

Build or Buy? "Buy" wins at lower volumes. Cross-over shifting rightward over time

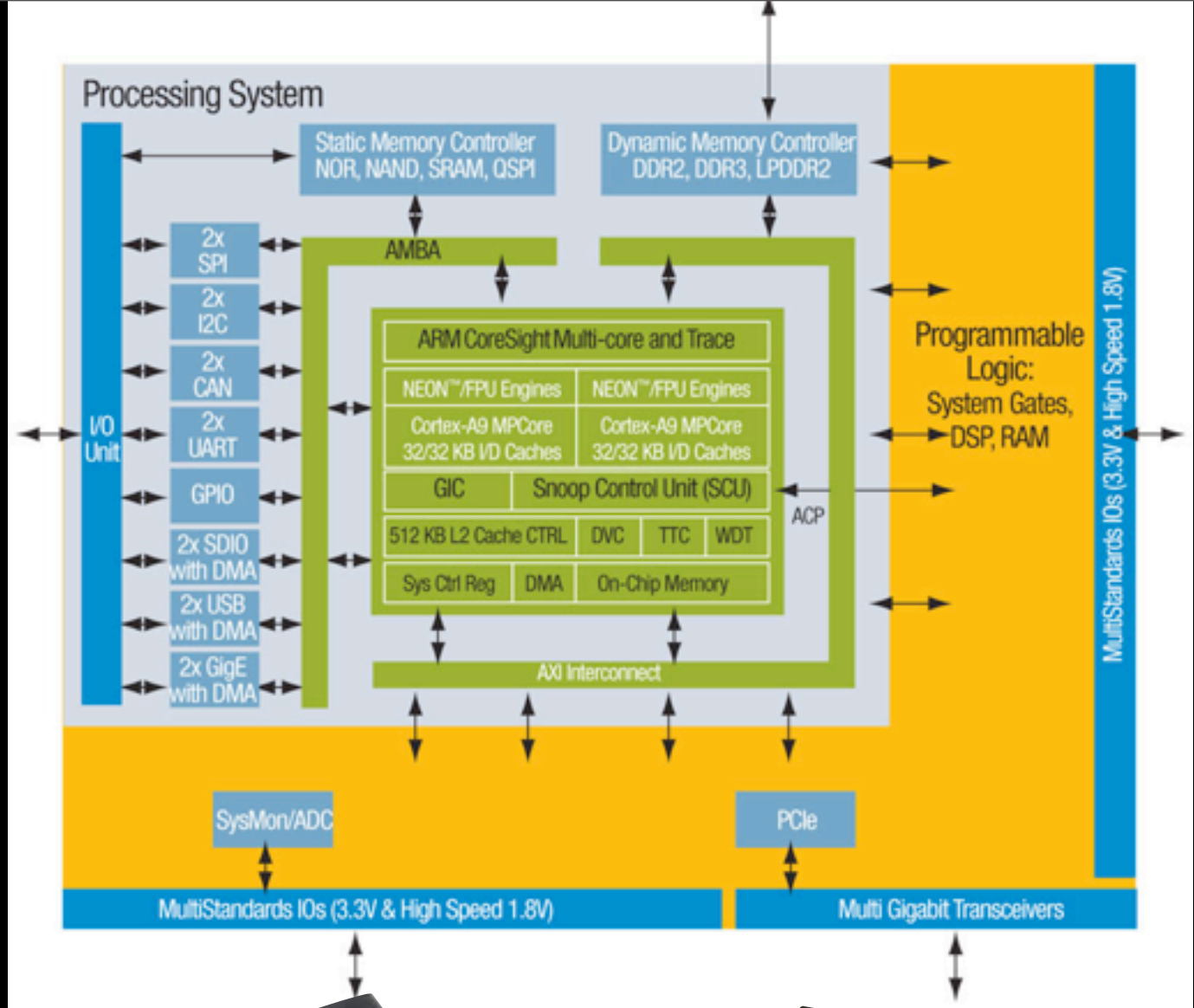


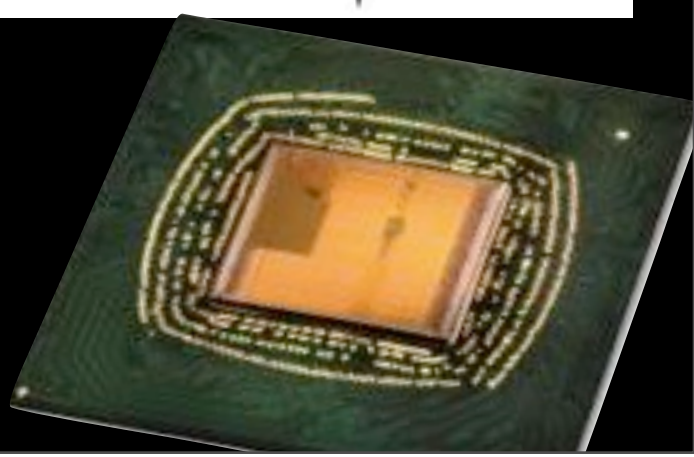
Process Design Kit: Design Rules and Device Models

Xilinx ZYNQ

A dual-core ARM SoC with a full set of peripherals.

Plus, a significant portion of the chip area devoted to Xilinx FPGA elements, that interact with ARM cores efficiently.





Break

Play:

CS 250: What we will do

CS 250 Flowchart

(see website for exact dates)

September++

11 lectures and 4 labs to prepare you for the project.
Each student does the 4 labs by themselves.
Students self-organize into two-person project teams.

Early October Oral & written project proposals.

October, November++

Lecture timeslots used for private meetings between teams and instructors. Also, public progress talks.

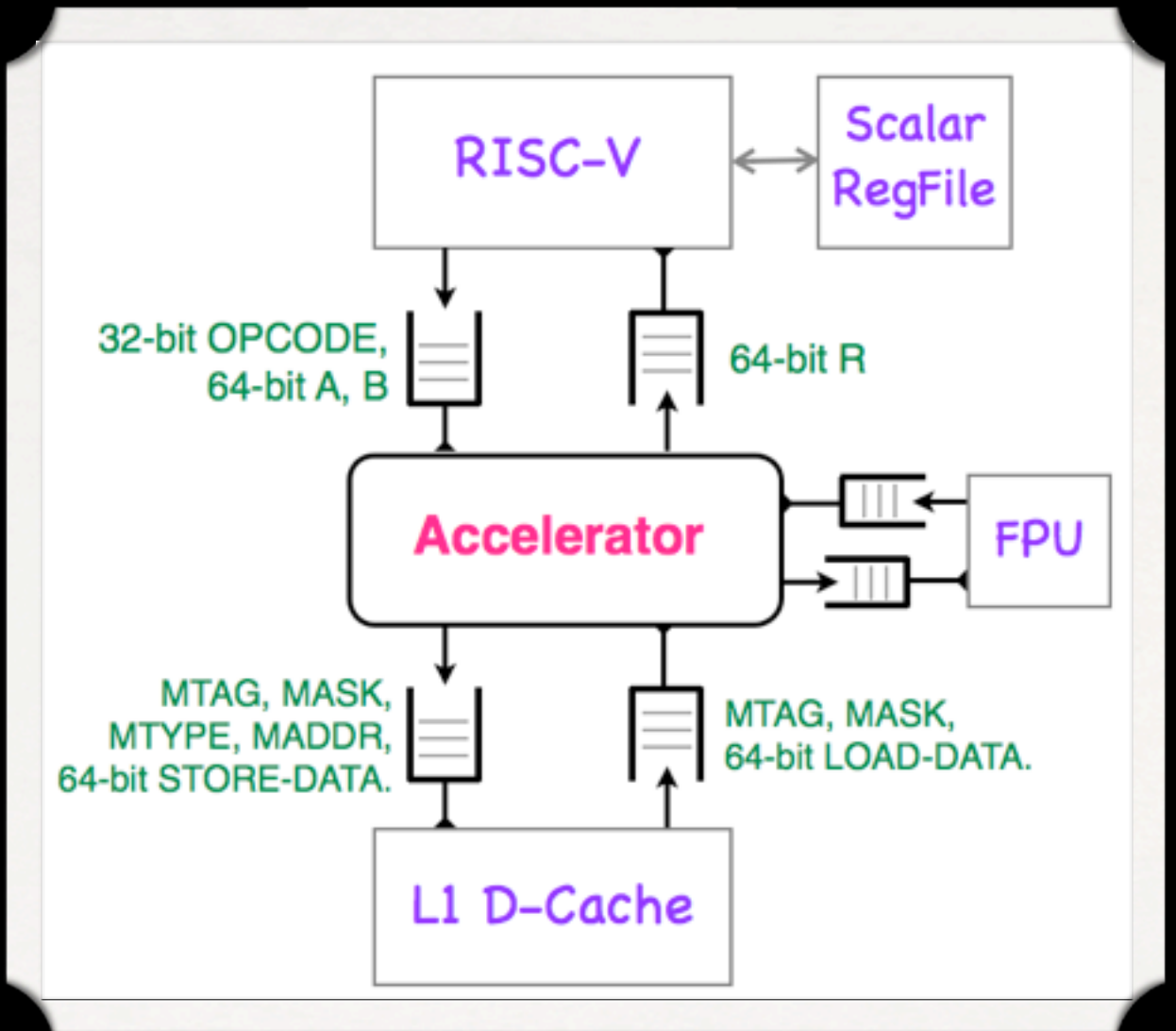
Mid-December: Final project talks and written reports.

Project Background

Rocket, a RISC-V ("risk five") implementation

Rocket is an implementation of Cal's RISC-V ISA. It is a current project in Professor Krste Asanovic's research group

Rocket is meant to be used by SoC designers.

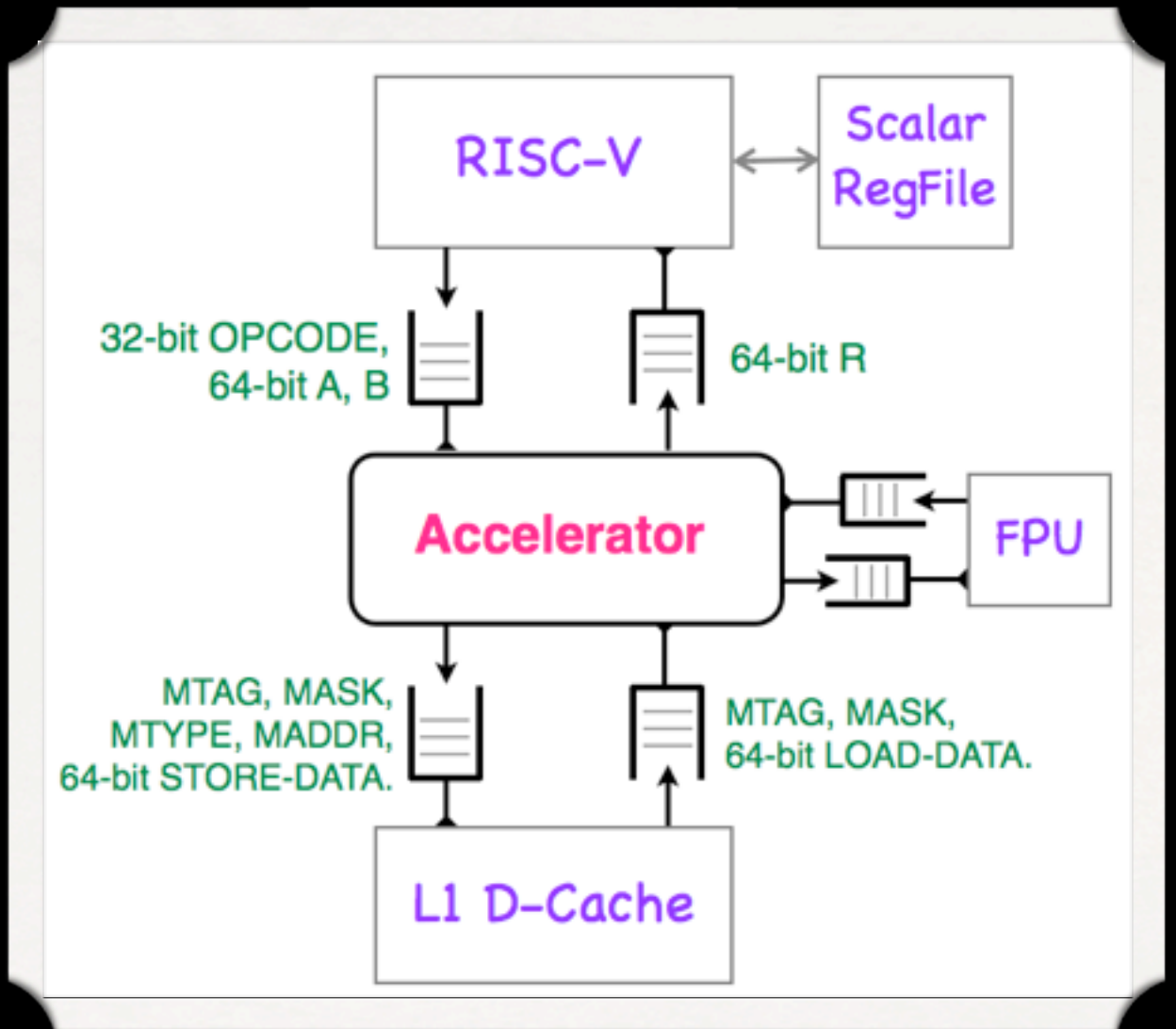


Rocket lets an SoC designer define new instructions in the ISA. Those instructions are executed by an accelerator that the SoC designer provides. Rocket defines the interface that the accelerator uses to interact with the CPU.

The Project

Teams will define an accelerator concept, and implement it using the Synopsis standard-cell flow (logic synthesis, place and route).

Teams will write their accelerator in Chisel, a new HDL by Jonathan Bachrach. Chisel is based on Scala.



Teams will design several implementations of their accelerator, each "Pareto-optimal" for different points in the power/performance/area space.

How we prepare you for the project

11 Lectures

2 lectures on Chisel (also, "Chisel Bootcamp", 9/30).

2 lectures on Rocket and accelerator ideas.

2 lectures on Pareto optimality space.

3 lectures on microarchitectural design patterns.

1 lecture on testing your accelerator.

4 Labs

Using Chisel.

Using Synopsis.

Using the RISC-V and Rocket tool chains.

Administriva, Part I

- Ben runs the class via Piazza.
  - https://piazza.com/ to sign up.
- Tools run on EECS instructional machines.
  - Get the account form from Ben and sign up soon!
- Vote on discussion section time.
  - Discussion section vital for class. Ben will be in touch.
- Laptop/tablet/smartphone in class.
  - Fine for note taking and class-related activities.
  - Please wait till the break to check emails, etc.

Administriva, Part II

Grading: 70% project, 25% labs, 5% class participation.

Undergrad prereq: B+ or better in CS 150.

Grad students: we expect you know digital logic design, contact us for advice if you don't.

See class website for schedule, deadlines.

Class URL on first slide of this lecture.

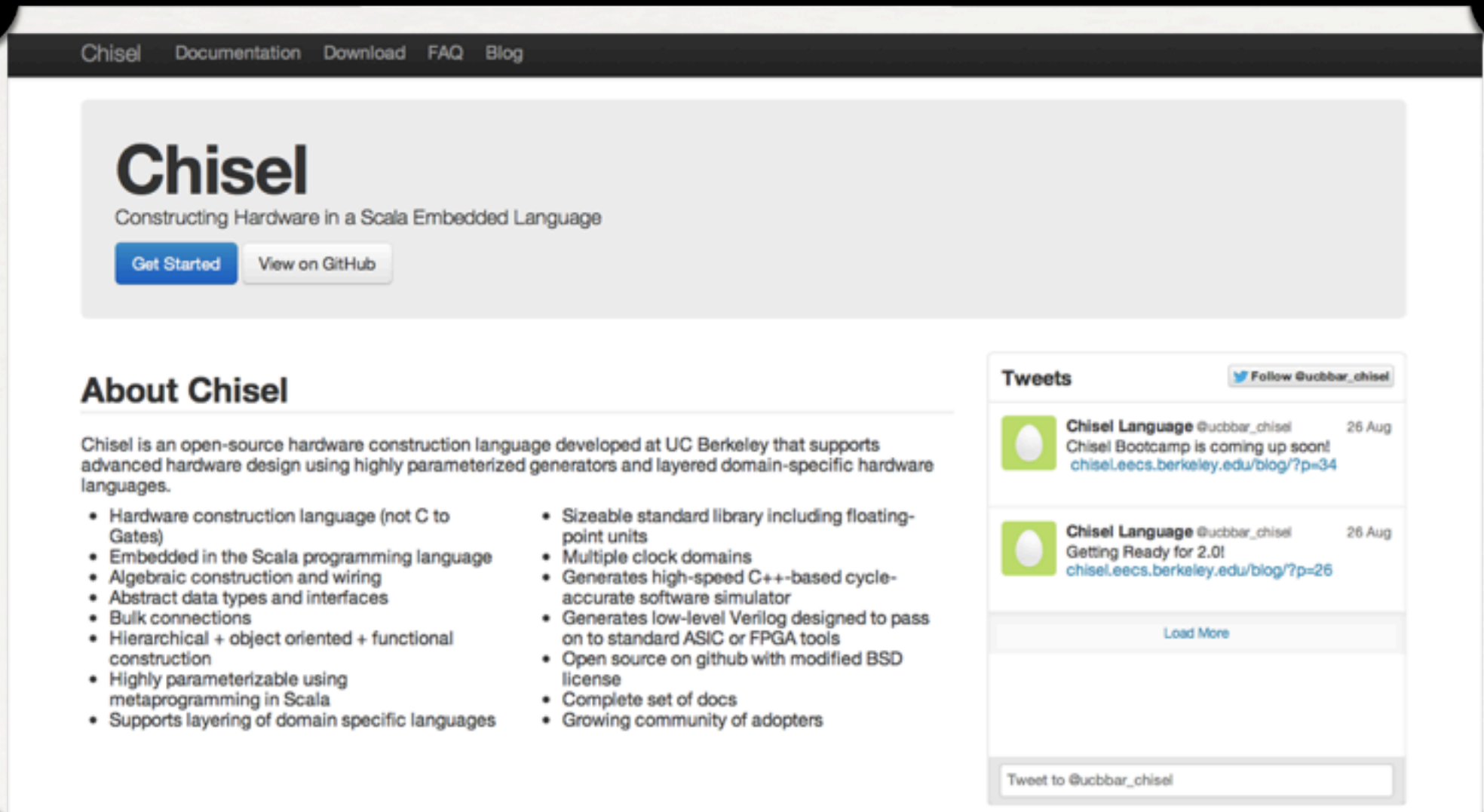
Deadlines.

All project deadlines are "hard" deadlines. See us if you have a serious conflict. For the labs, there are 4 "late days" -- see website and Ben for details.

Collaboration policy

You must do the labs by yourself and write your own lab reports. Discussions about labs with others is OK.

On Tuesday … Chisel, Part One



Tue Sep 3	JB	Lecture 2: Introduction to Chisel hardware description language.