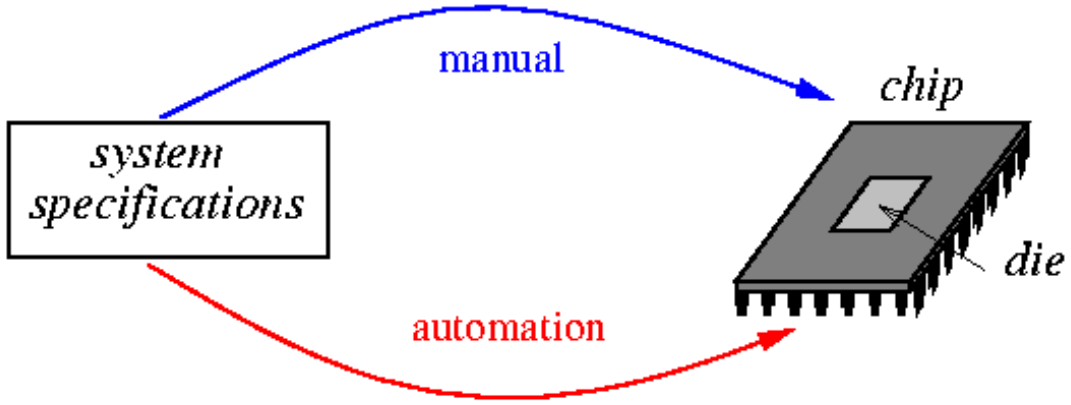
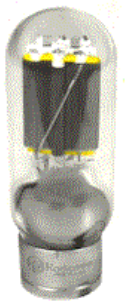
Unit 1: Intro. to Electronic Design Automation

- Today's contents:
  - Introduction to VLSI design flow/methodologies & styles
  - Introduction to VLSI design automation tools
  - Semiconductor technology roadmap
  - CMOS technology
- Readings
  - Chapters 1-2



Milestones for IC Industry

- **1947:** Bardeen, Brattain & Shockly invented the transistor, foundation of the IC industry.
- **1952:** SONY introduced the first transistor-based radio.
- **1958:** Kilby invented integrated circuits (ICs).
- **1965:** Moore's law.
- **1968:** Noyce and Moore founded Intel.
- **1970:** Intel introduced 1 K DRAM.



In 1956 John Bardeen, William Shockley and Walter Brattain shared the Nobel Prize in Physics for their discovery of the transistor.

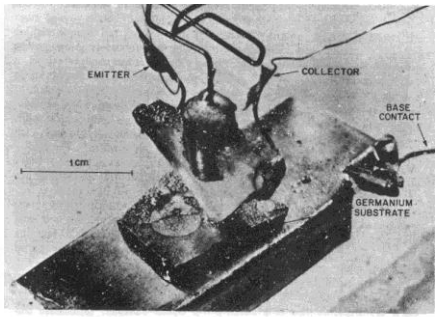
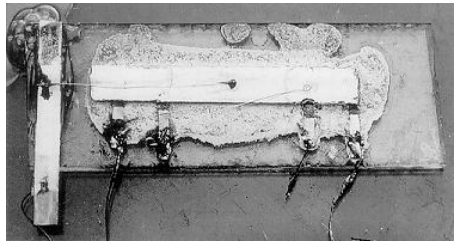
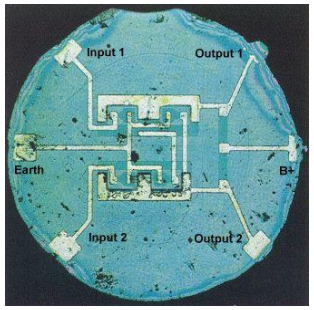


Fig. 1 The first transistor.

First transistor



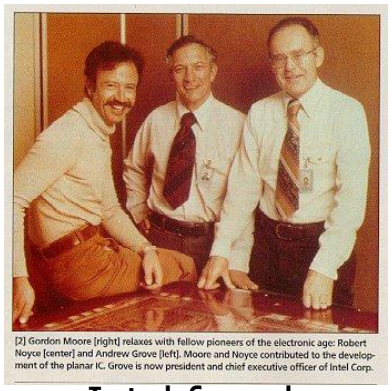
First IC by Kilby



First IC by Noyce

Milestones for IC Industry

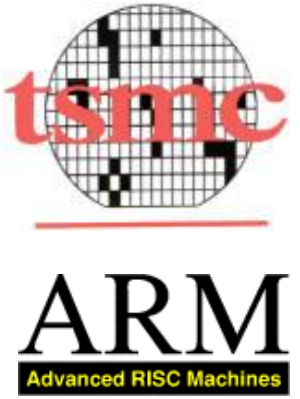
- **1971:** Intel announced 4-bit 4004 microprocessors (2250 transistors).
- **1976/81:** Apple II/IBM PC.
- **1985:** Intel began focusing on microprocessor products.
- **1987:** TSMC was founded (fabless IC design).
- **1991:** ARM introduced its first embeddable RISC IP core (chipless IC design).


[2] Gordon Moore [right] relaxes with fellow pioneers of the electronic age: Robert Noyce [center] and Andrew Grove [left]. Moore and Noyce contributed to the development of the planar IC. Grove is now president and chief executive officer of Intel Corp.

Intel founders

4004

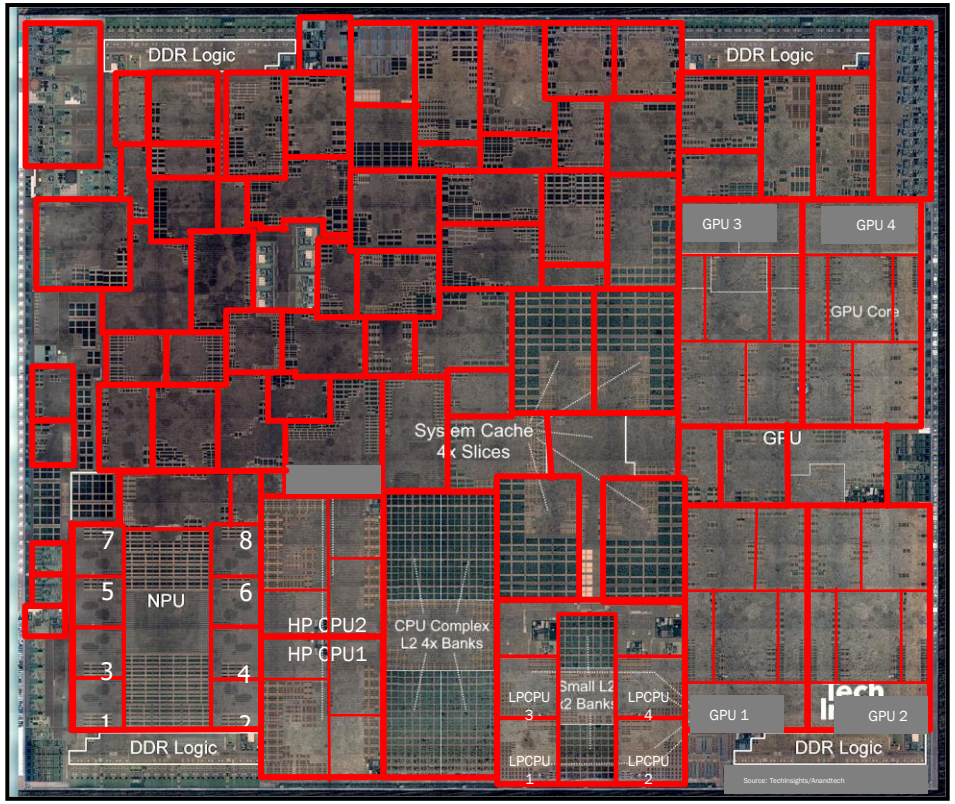
IBM PC



Apple A12 - iPhone Xs (2018)

- 7nm TSMC FinFet
- 9.89 x 8.42 = 83.27 mm2
- 6.9 Billion transistors
- 4 **GPU** cores (~18%): 9 Blocks
- 6 **CPU** cores (~14%): 13 Blocks
  - 4 ‘Tempest’ Low Power CPU cores
  - 2 ‘Vortex’ High Performance CPU cores
  - L2 & L3 caches
- 8 **NPU/TPU** cores (~7%) 4 Blocks
- DDR (~3%) 1 block
- Misc (~57%): 50 Unique Blocks

- Total: ~75 unique blocks
- Q: did Apple design all its IPs?



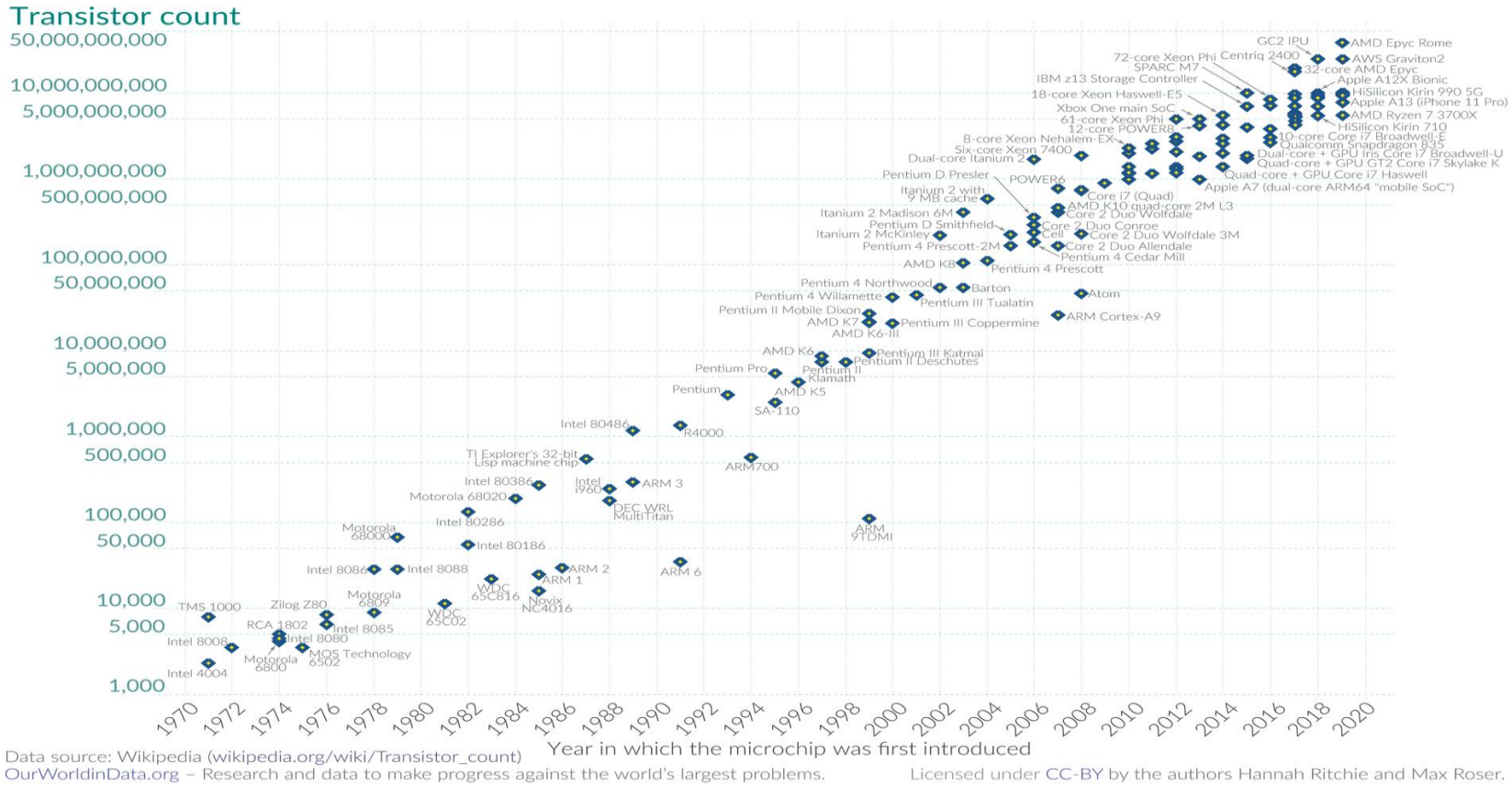
Questions Are

Design flow / methodology

From previous picture

- How was the whole chip partitioned? Verification at the functional level?
- Shall emulation be used to verify the functionalities?
- Which technology node(s) should we use? Pros & Cons?
  - Shall we go with the new node? Or current one? Was foundry ready?
- How was the floorplan decided?
  - Looks like HP(CPU1), HP(CPU2) were mirrored? How many floorplans shall we try?
- Any routing channels? If not, were blocks abutted? How to do pin assignment for each block? Can wiring route cross the blocks?
- How were DRC, LVS, RC extraction done? Whole chip? Or at block level?
  - How to resolve density issue popping up when blocks are grouped together?
- How was the entire chip verified? Using STA (static timing analysis)?
  - If fixes were needed, how to handle the ECO changes? Re-route? Re-place? …
- How was the chip tested?
- Shall we consider heating, aging, …?
- …

Moore's Law: The Number of Transistors Doubles Every 2 Years



TSMC CEO Mark Liu says the rise of 5G, A.I., and the Internet of Things is paving the way for "a golden era for the semiconductor industry."

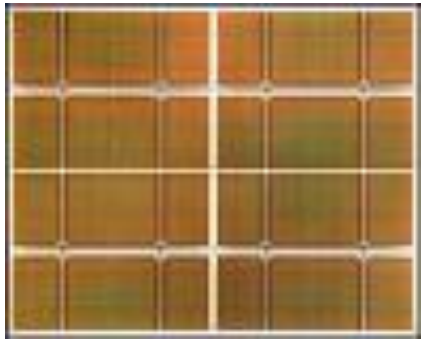
Milestones for IC Industry (Cont'd)

- **1996:** Samsung introduced IG DRAM.
- **1998:** IBM announces1GHz experimental microprocessor.
- **1999/earlier: System-on-Chip (SOC)** methodology applications.
- An Intel P4 processor contains 42 million transistors (1 billion by 2005)

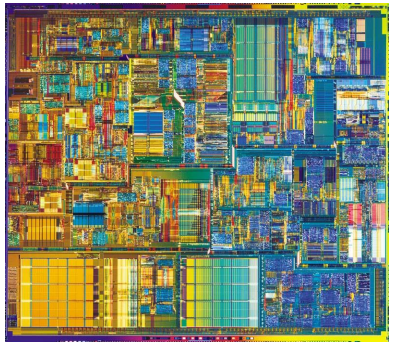
- In **2021**, Apple/TSMC produced 15 billion transistors in a chip

  Apple's A15 Bionic chip powers iPhone 13 with 15 billion transistors, new graphics and AI (@5nm)

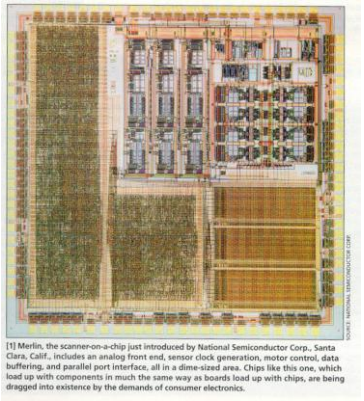
- **Semiconductor/IC: very critical**



4GB DRAM (2001)



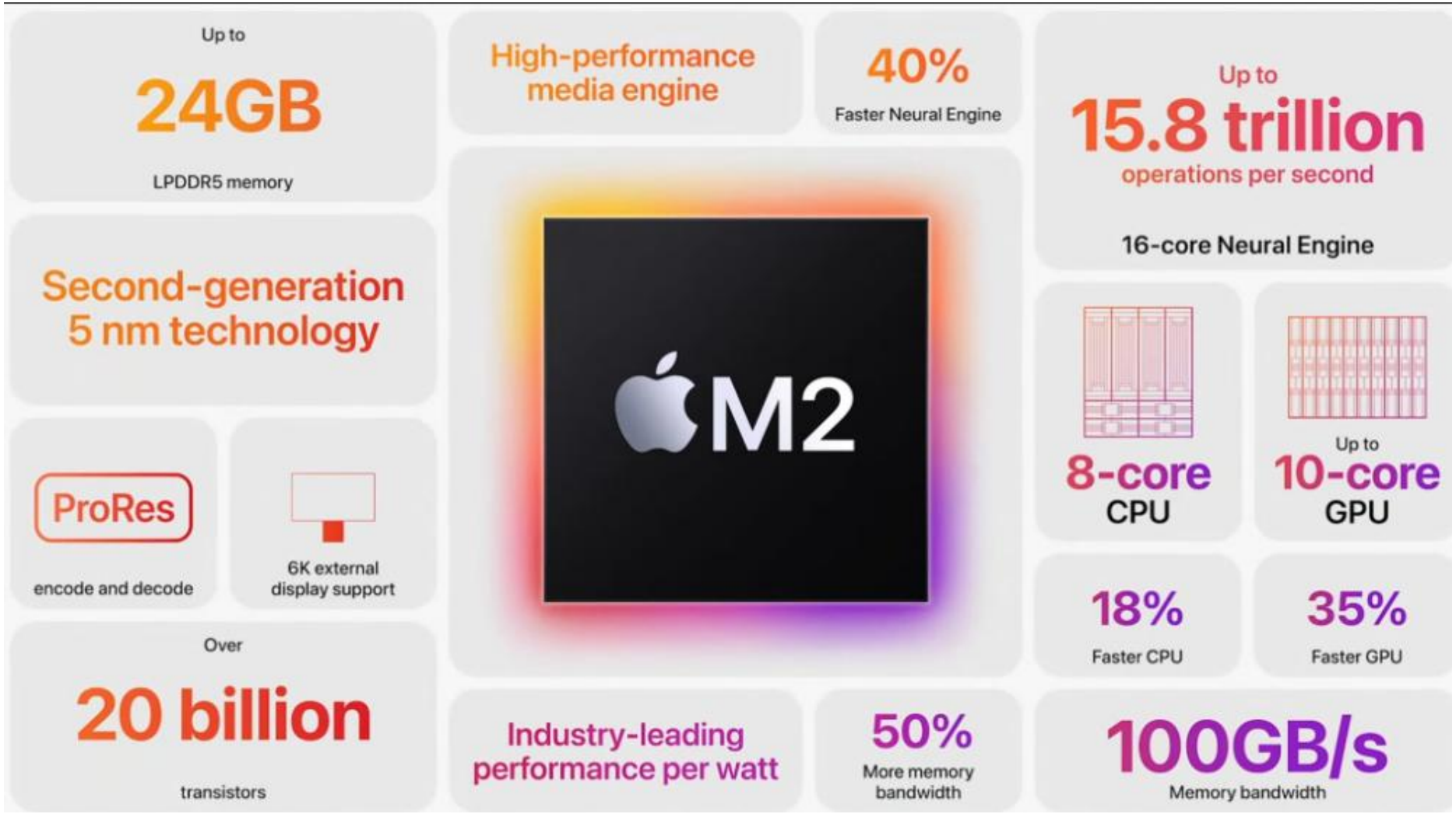
Pentium 4



Scanner-on-chip

Blue tooth technology

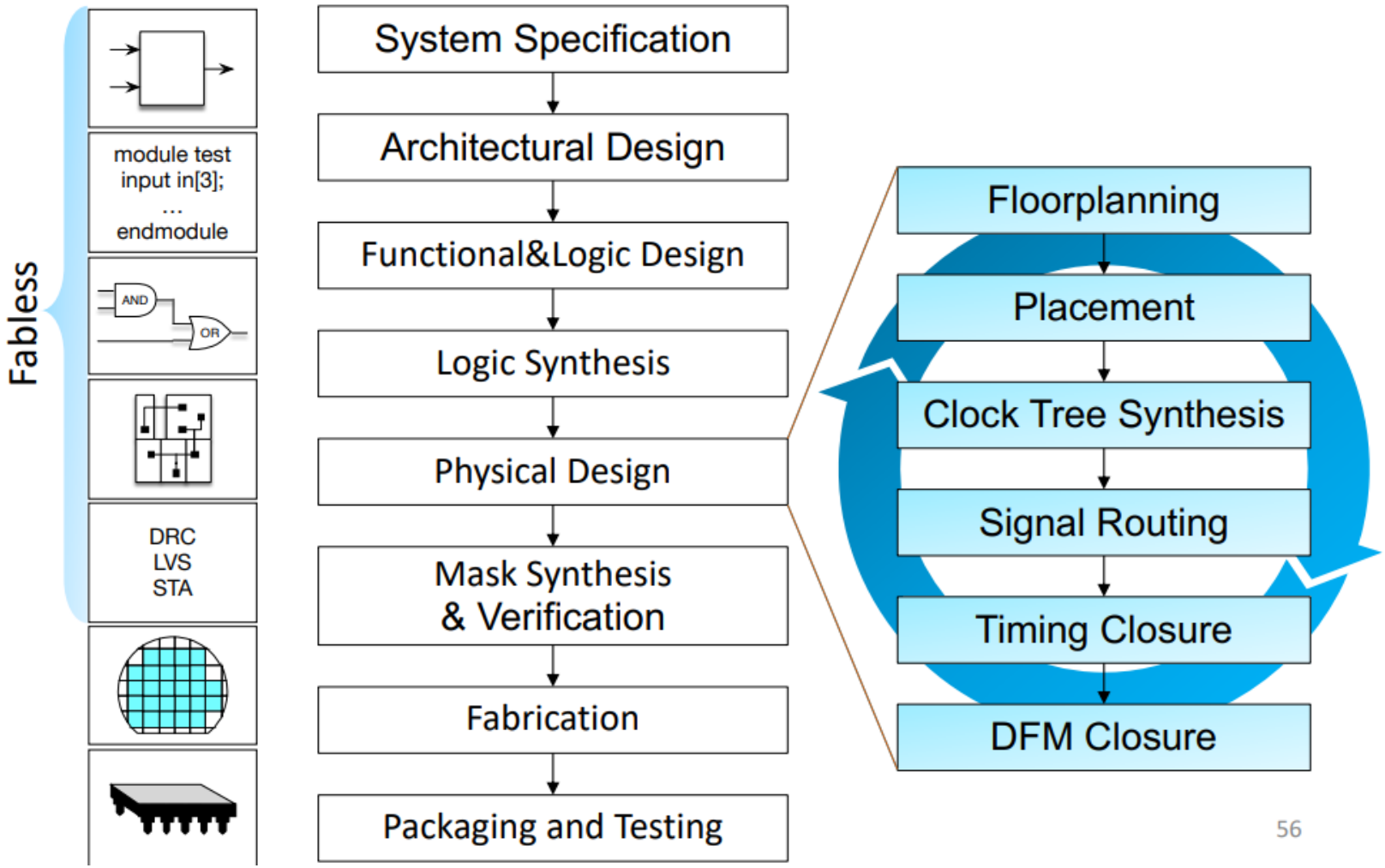
Apple M2 (on TSMC 5nm)



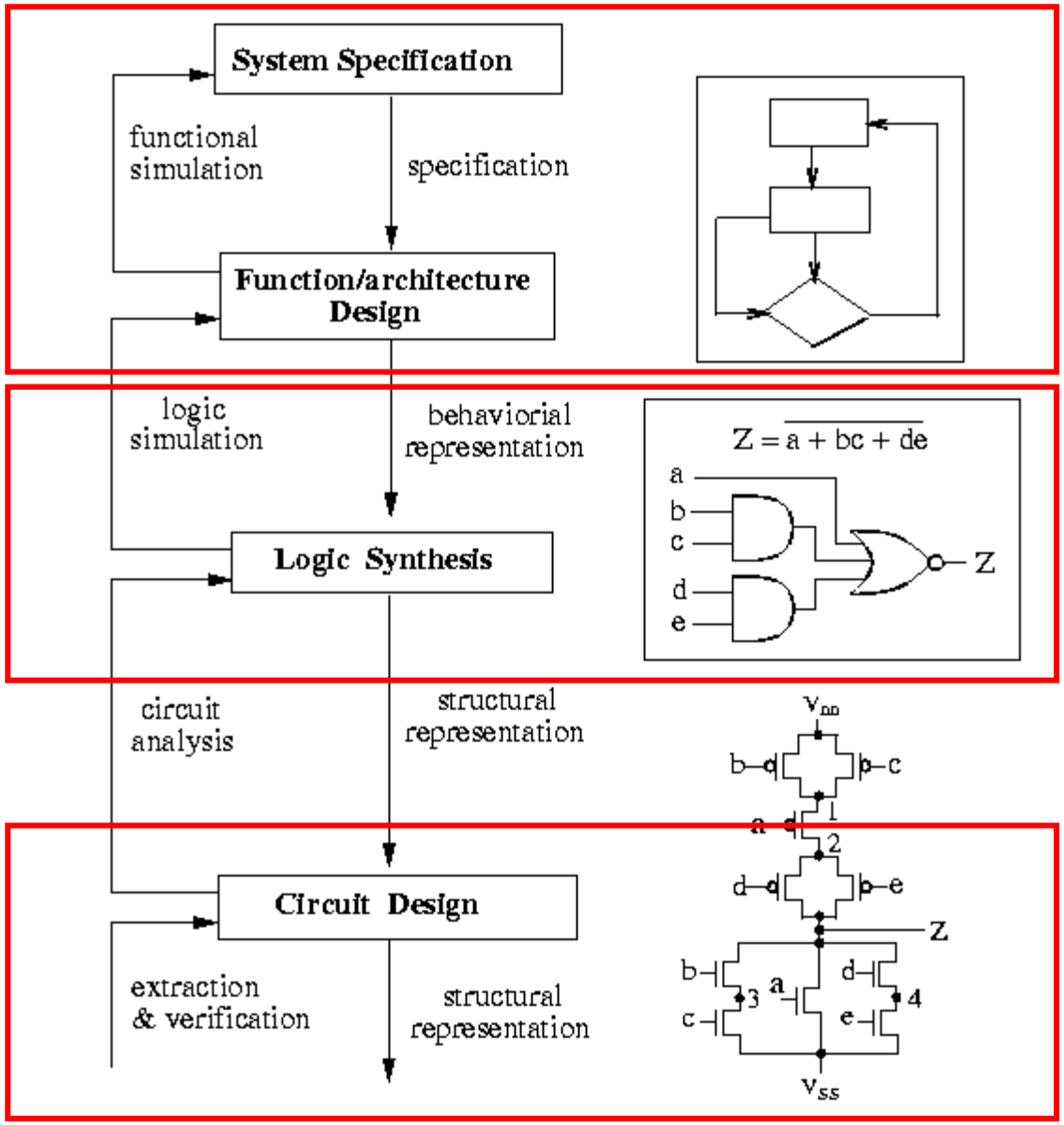
Course Ordering Of The Major Topics

- Digital design flow
- Basic logic synthesis, technology mapping
- CMOS logic gates and Boolean equations
- Algorithms and complexity
- Synthesis, technology mapping
- Compaction
- Partitioning
- Floorplanning
- Placement
- Routing
- High level synthesis

IC Design Flow – Silicon Compiler



Typical VLSI Design Cycle

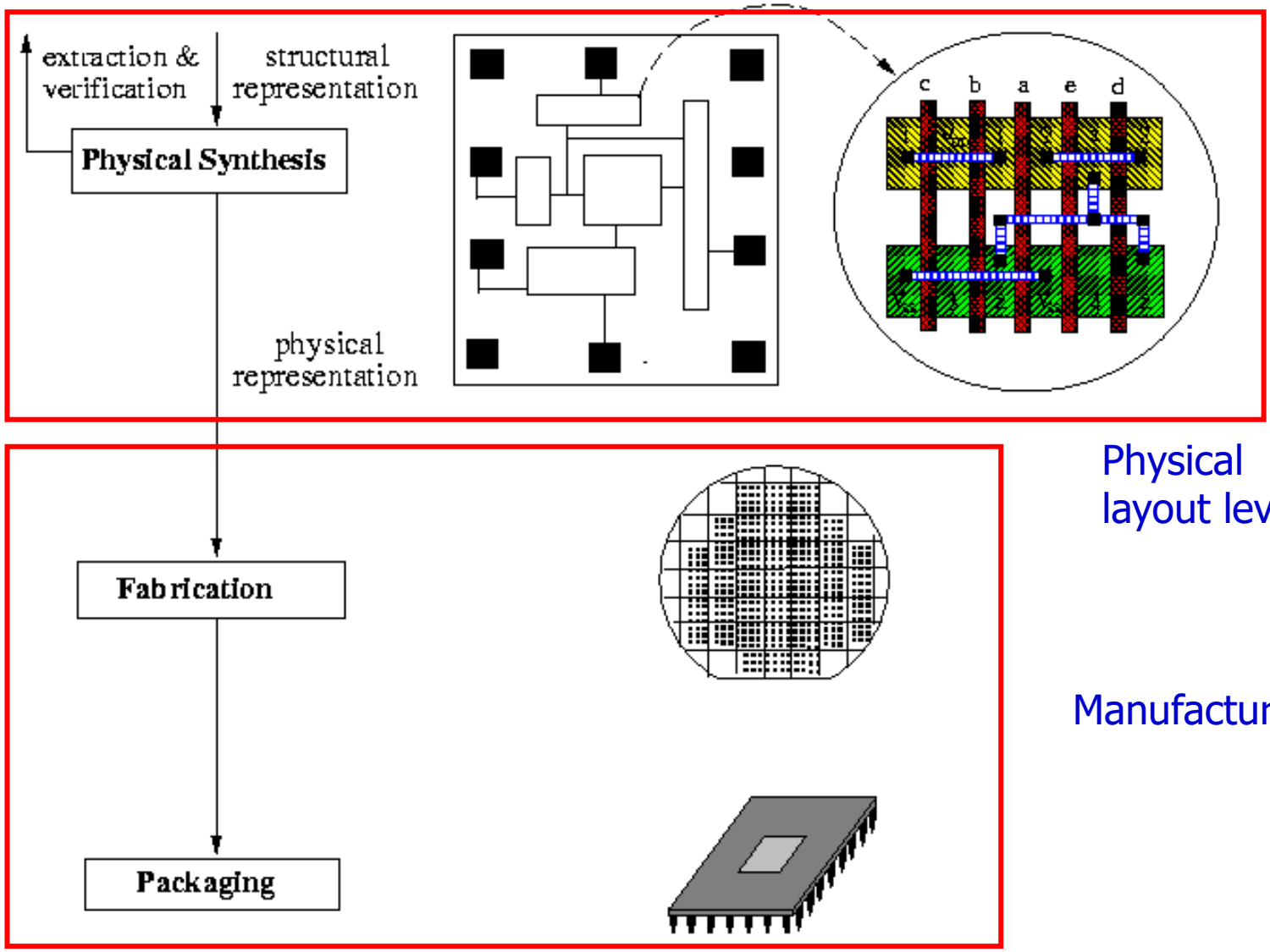


System level

Logic level

Building block

Traditional VLSI Design Flow (Cont'd)

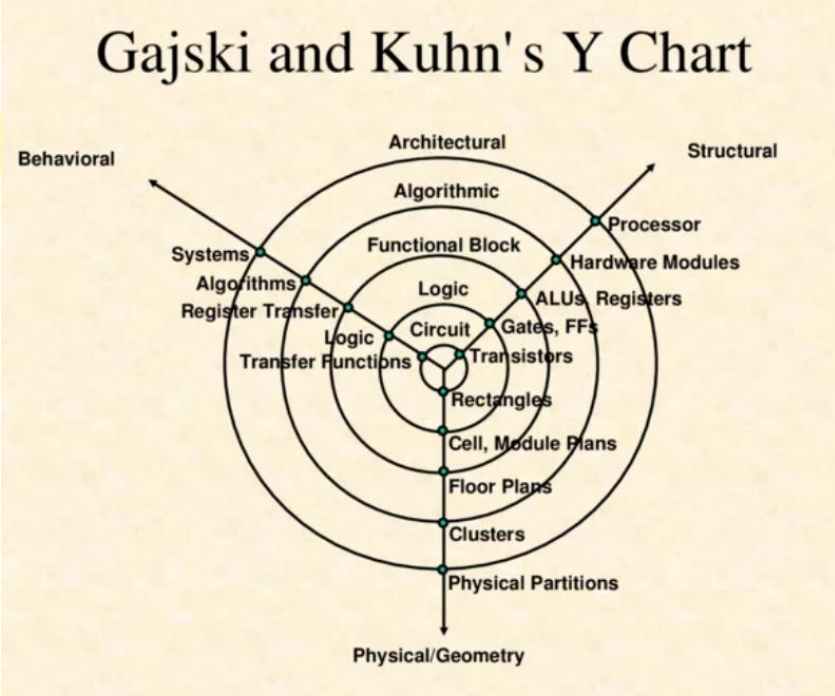


Physical layout level

Manufacturing

Major Components In SoC Design Flows

- On the **digital** SoC (system on a chip)
  - System level design: simulation (behavioral, function level)
  - Logic synthesis; then, physical synthesis with constraints (may need gate level simulation, or formal verification)
  - Layout: floorplan, place-and-route, timing optimization
    - Formal verification to confirm the logic changes in layout
  - Verification:
    - R/C parasitic extraction with corners
    - static timing analysis
    - layout verification (DRC, LVS, PERC, ...)
  - Power network analysis and IR-drop with electro-migration
  - ECO changes

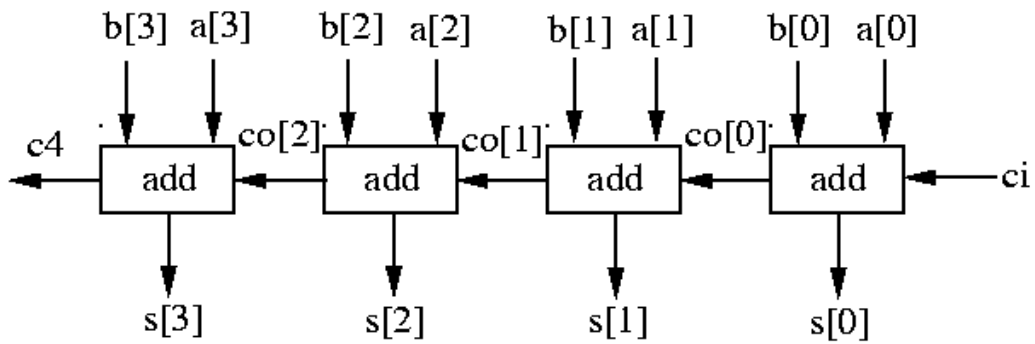


Three Design Views

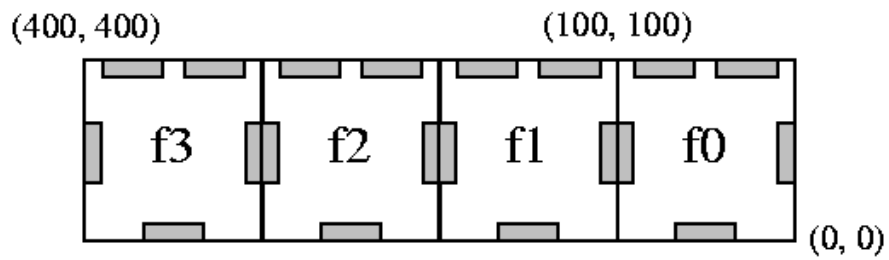
Behavior

```
module add4 (s, c4, ci, a, b);
        input [3:0] a, b;
        input ci;
        output [3:0] s;
        output c4;
        wire [2:0] co;
            add f0 (co[0], s[0], a[0], b[0], ci);
            add f1 (co[1], s[1], a[1], b[1], co[0]);
            add f2 (co[2], s[2], a[2], b[2], co[1]);
            add f3 (c4, s[3], a[3], b[3], co[2]);
endmodule
```

Structural



Physical



Design Issues and Tools

- System-level design
  - Partitioning into hardware and software, co-design, co-simulation, etc.
  - Cost estimation, design-space exploration
- Algorithmic-level design
  - Behavioral descriptions (e.g. in Verilog, VHDL)
  - High-level simulation, waveform viewing
- From algorithms to hardware modules
  - High-level (or architectural) synthesis
- Logic design:
  - Schematic entry
  - Register-transfer level and logic synthesis
  - Gate-level simulation (functionality, power, etc) & waveform viewing
  - Timing analysis
  - Formal verification
- Testing

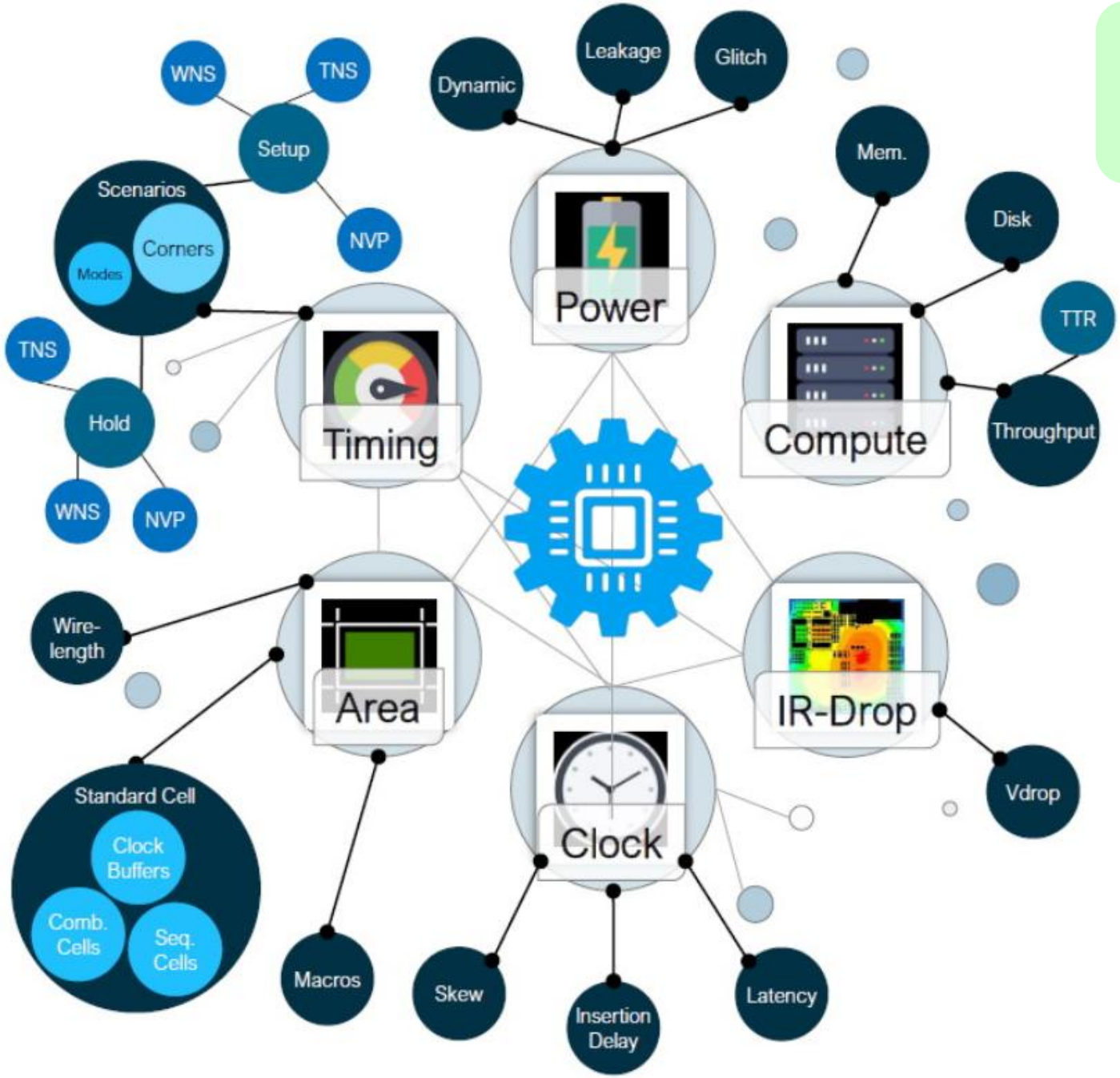
Design Metrics

- Design metrics: (PPA) speed, power dissipation, area; noise/crosstalk, design time, testability, etc.

- Design evolution: interconnect (not gate) delay dominates circuit performance in deep submicron era.
  - Interconnects are determined in physical design
  - Need considering interconnections in early design stages
  - Often post-layout simulation results are different from those in pre-layout

- Other tasks, such as testing, formal verification, …, are not covered here

EDA plays a critical role in digital SoC!



https://semiengineering.com/the-next-incarnation-of-eda/

The Next Incarnation Of EDA (Aug. 2022)

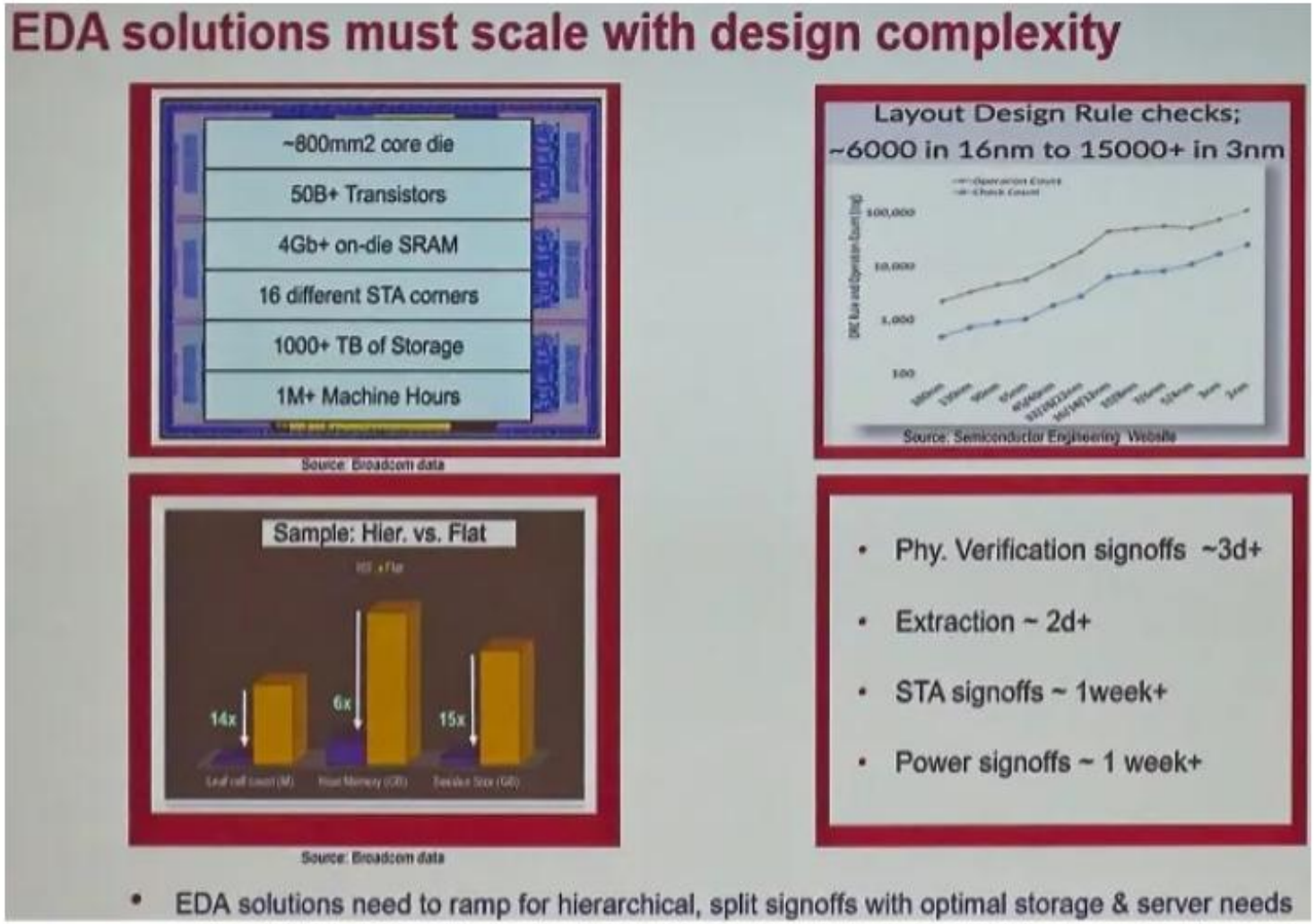
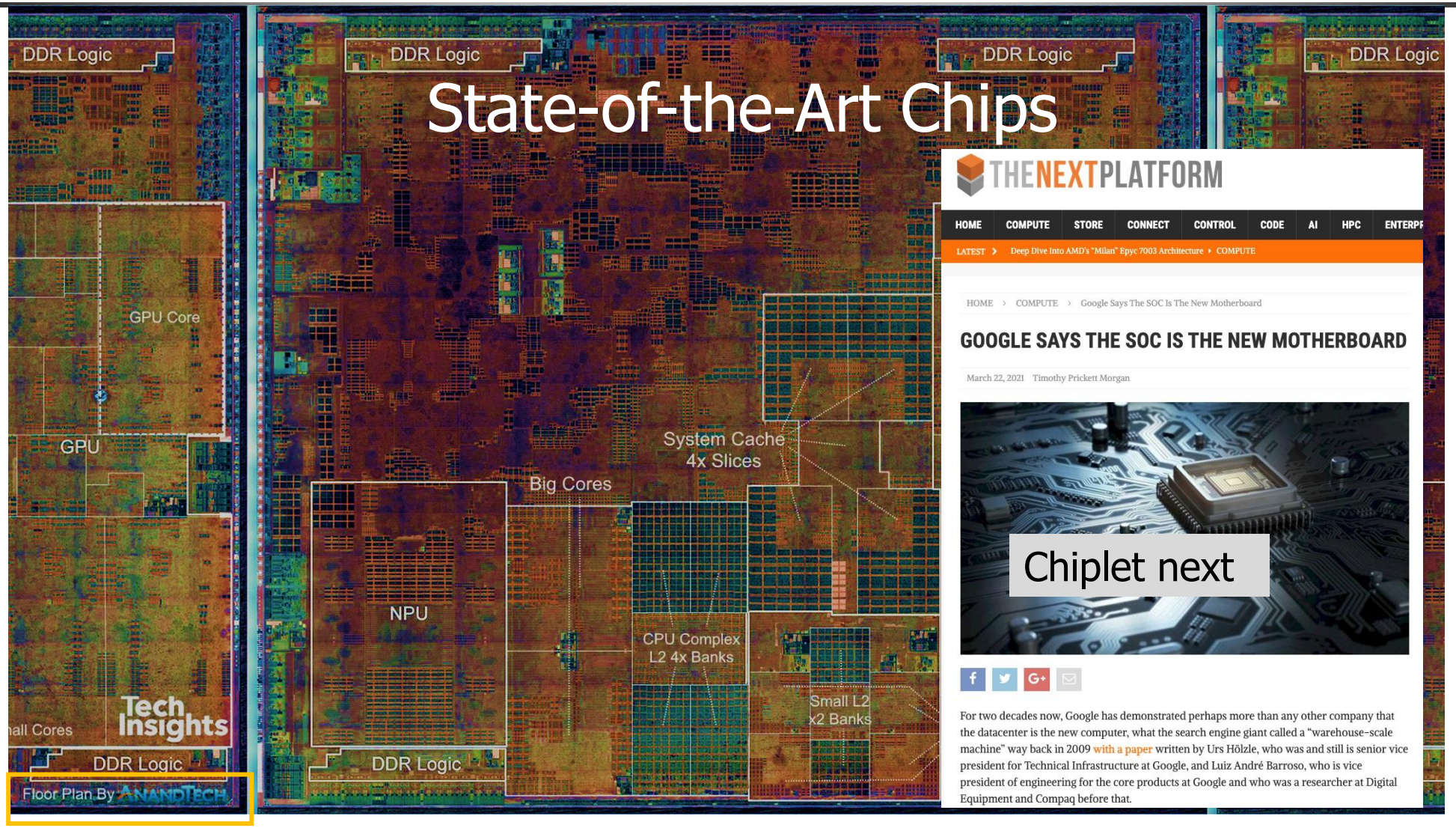


Fig. 1: Design complexity and EDA. Source: Broadcom

State-of-the-Art Chips



Major Components In Custom Design Flows (2)

- On the **custom/analog** design:
  - Design specification (schematic, AMS Verilog, …):
    - Netlisting output, **simulation** (spice, co-sim)
  - Layout:
    - placement with matching (constraint based),
    - special routing, in-design analysis
  - Layout verification: DRC, LVS, and RC extraction
  - Post-layout simulation (with huge RC data)
  - Power network analysis and IR-drop, Electro-migration
  - ECO fixes (schematic, sim, layout, verification)

Same for Manufacturing
- Optical Proximity Correction (OPC)
  - Lithography simulation to get hot spots (using CPU farms)
  - Add extra shapes to mask

Many software development talents are needed;

PDK development also needs talents.

Wafer & Chip

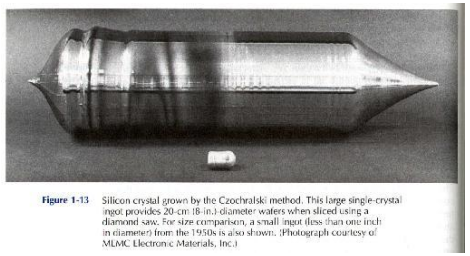
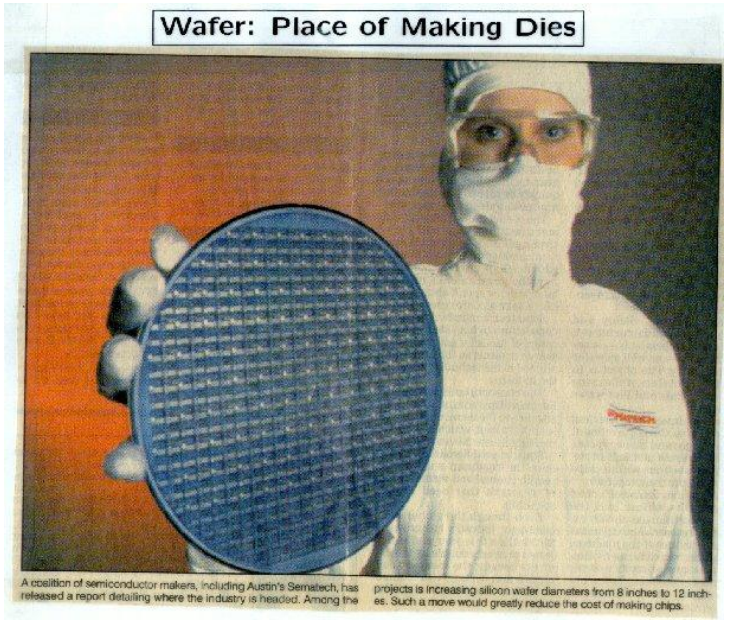


Figure 1-13 Silicon crystal grown by the Czochralski method. This large single-crystal ingot provides 20-cm (8-in.) diameter wafers when sliced using a diamond saw. For size comparison, a small ingot (less than one inch in diameter) from the 1950s is also shown. (Photograph courtesy of MEMC Electronic Materials, Inc.)



A coalition of semiconductor makers, including Austin's Sematech, has released a report detailing where the industry is headed. Among the projects is increasing silicon wafer diameters from 8 inches to 12 inches. Such a move would greatly reduce the cost of making chips.

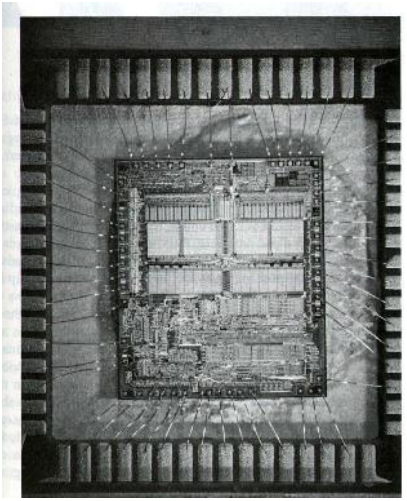
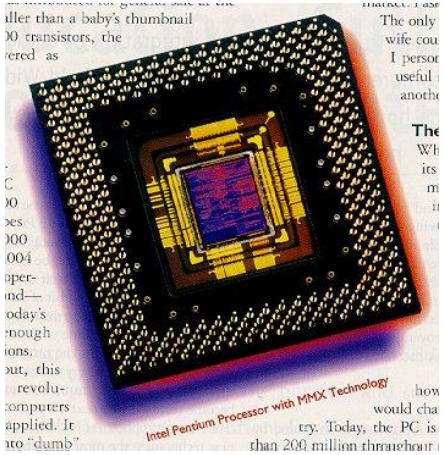
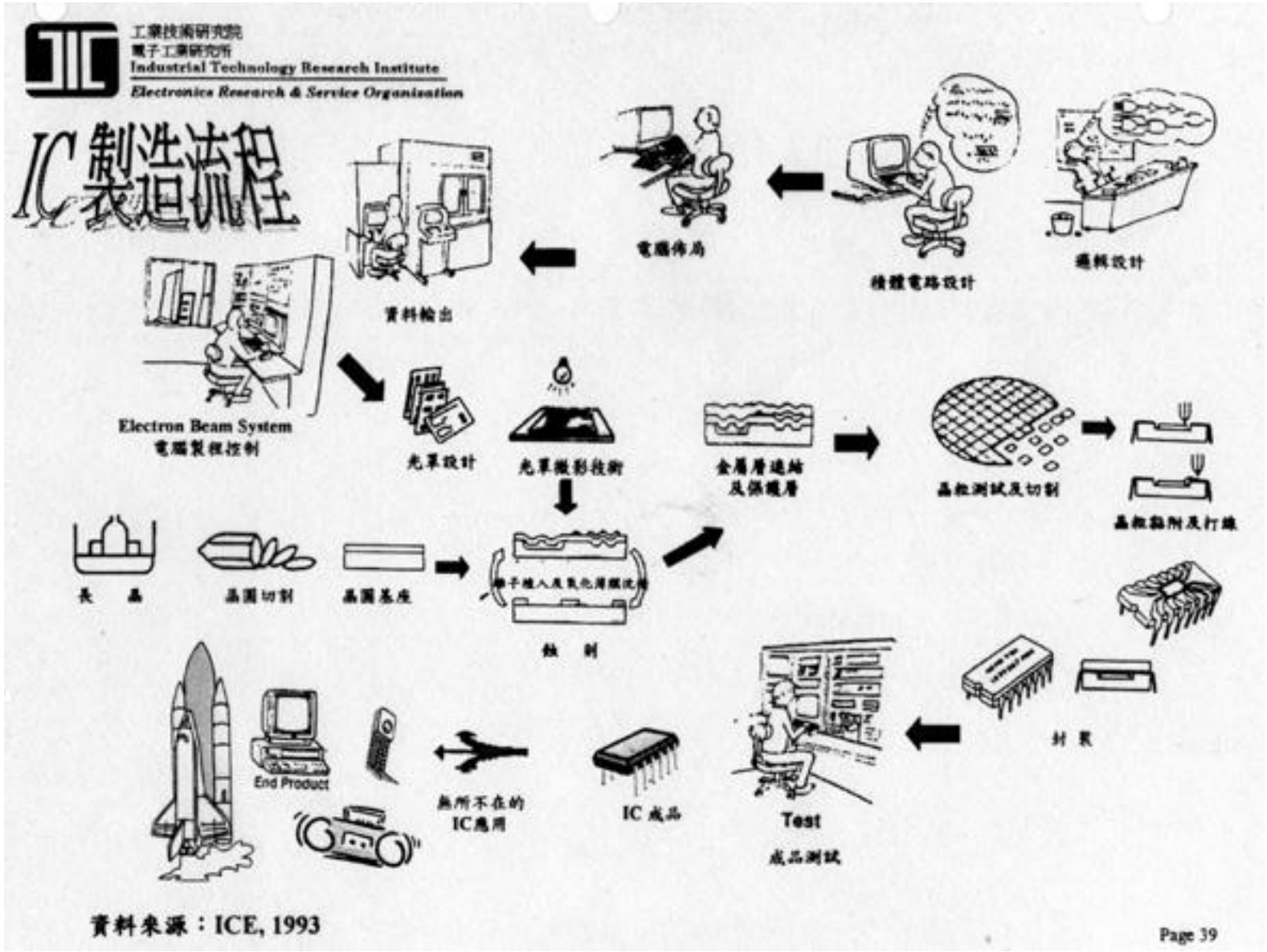
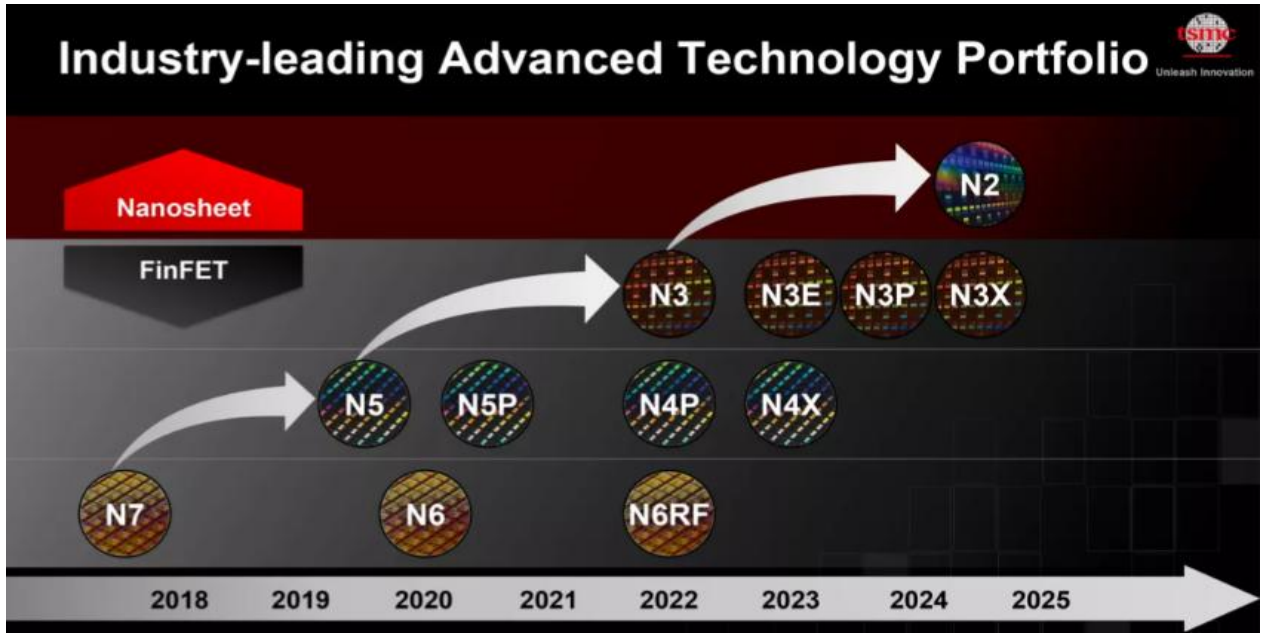


Figure 9-34 Attachment of leads from the Al pads on the periphery of the chip to posts on the package. (Photograph courtesy of Motorola, Inc.)



IC Design & Manufacturing Process





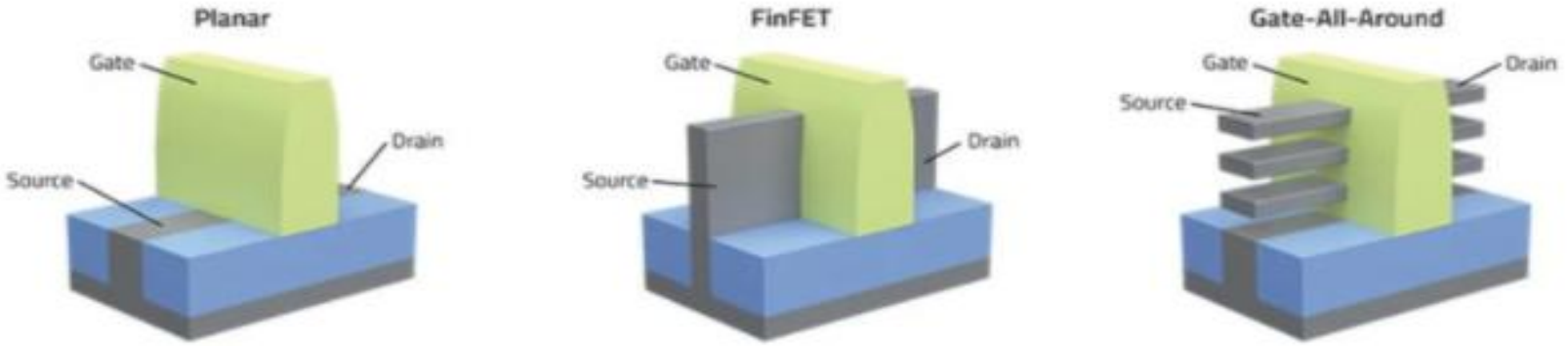
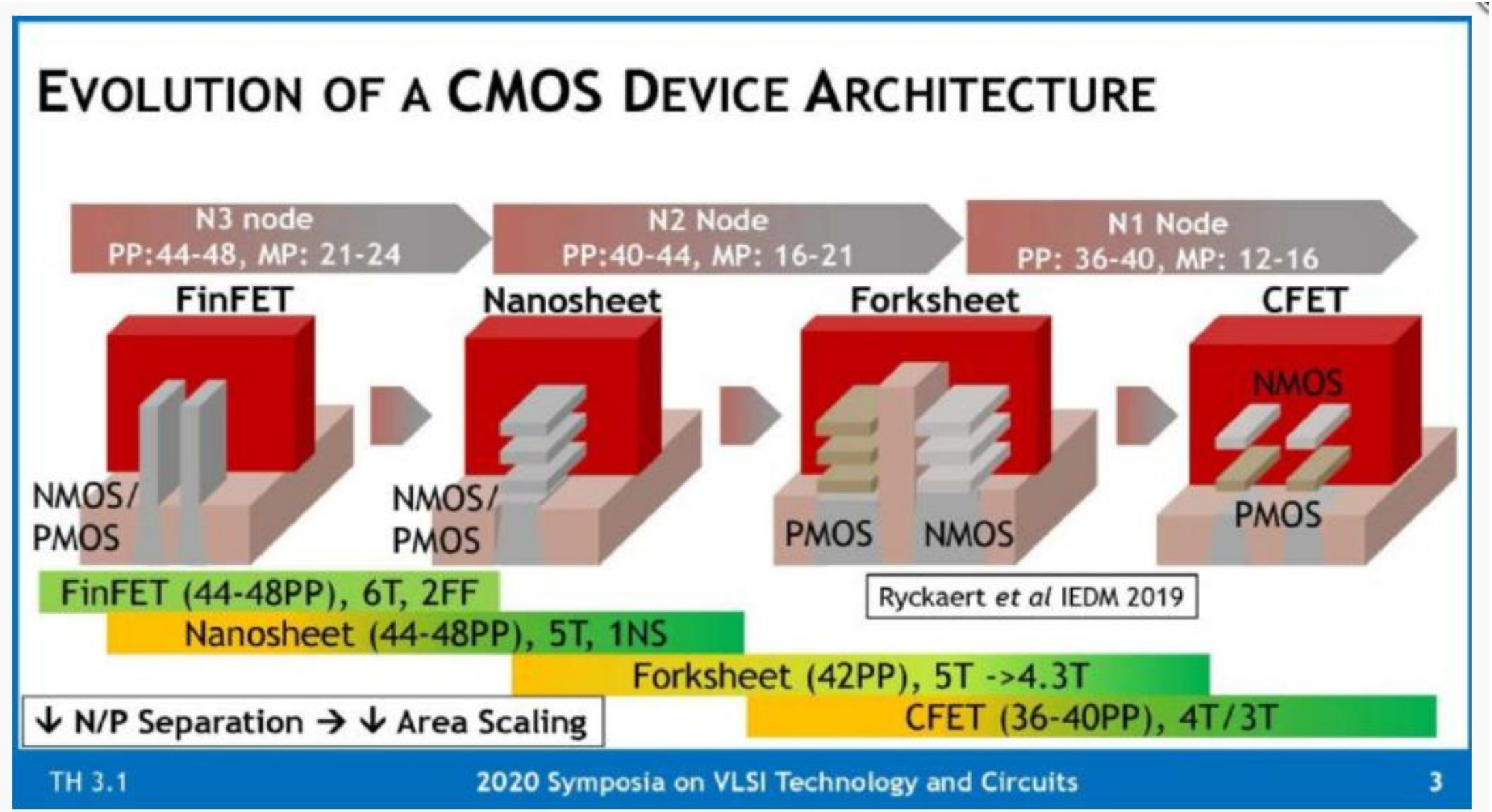


Fig. 1: Planar transistors vs. finFETs vs. gate-all-around

Source: Lam Research

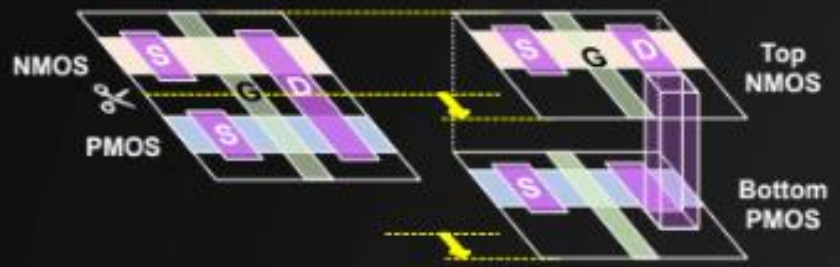
突破2奈米以下瓶頸 **CFET架構接替GAA**
•茅塊 / 綜合外電 2022-6-22

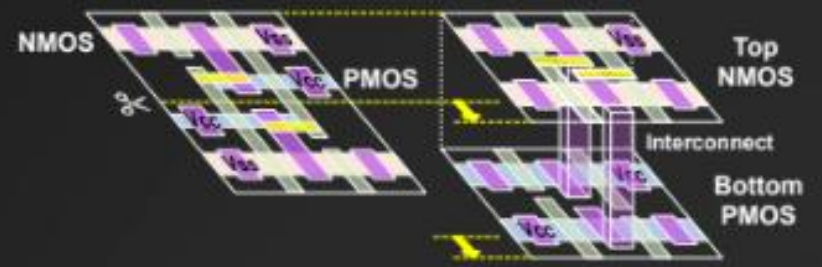
Intel Evolution of Transistor Innovation – SemiWiki
Worthwhile to read it



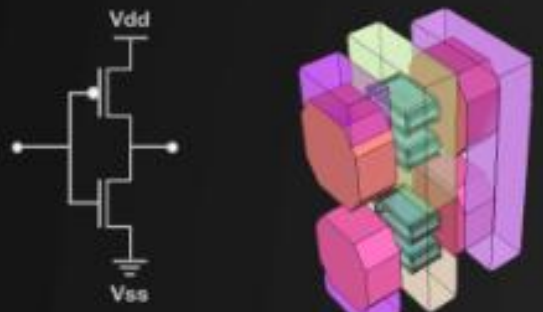
CFET Density Scaling

Density scaling ~ 1.5-2X

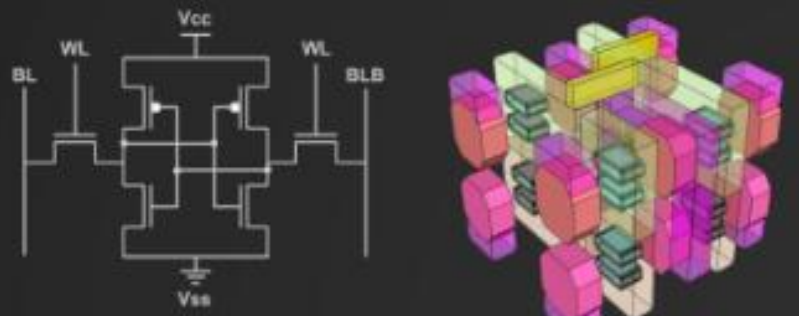




Inverter



SRAM



Lithography

& each unit has a starting price of $119M

IC Manufacturing: Transport of an EUV System

The Center of Gravity Matters!

Summary So Far

- EDA tools have many software components → need lots of software talents
- Such expertise can be applied, expanded to other areas
- Multidisciplinary study is very common
- Artificial intelligence/Machine learning is a big trend
  - But so far not every problem can be improved by ML approaches
  - ML-inside and ML-outside

Before We Close

- Industry trends
- Introduction of chip design flows & methodologies
- Briefly mention the major steps in digital and custom designs

Design Actions – After Specifications

- **Synthesis:** increasing information about the design by providing more detail (e.g., logic synthesis, physical synthesis)
- **Analysis:** collecting information on the quality of the design (e.g., timing analysis)
- **Verification:** checking whether a synthesis step has left the specification intact (e.g., layout verification)
- **Optimization:** increasing the quality of designs by rearrangements in a given description (e.g., logic optimizer, timing optimizer)
- **Design Management:** storage of design data, cooperation between tools, design flow, etc. (e.g., database).

HDL Synthesis

HDL Synthesis = Domain Translation + Optimization

Variable (A,B,C,D,Y) are assigned to registers

Domain translation

Optimization (area, timing, power...)

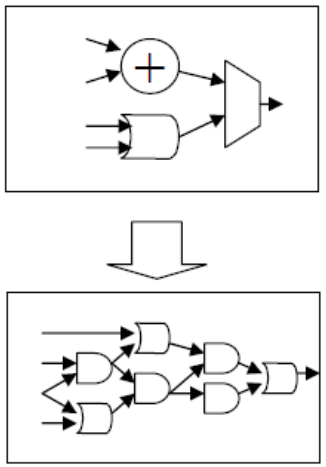
Logic gates can have various sizes for strengths

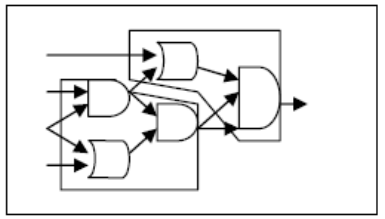
```
--VHDL                          //Verilog
If (A='1') then                 if (A==1)
  Y <= C + D;                     Y = C + D;
elseif  (B='1') then            else if (B==1)
  Y<= C or D;                     Y = C | D;
else Y<= C;                     else Y = C;
endif
```

Behavioral domain

MUX is needed

RTL synthesis

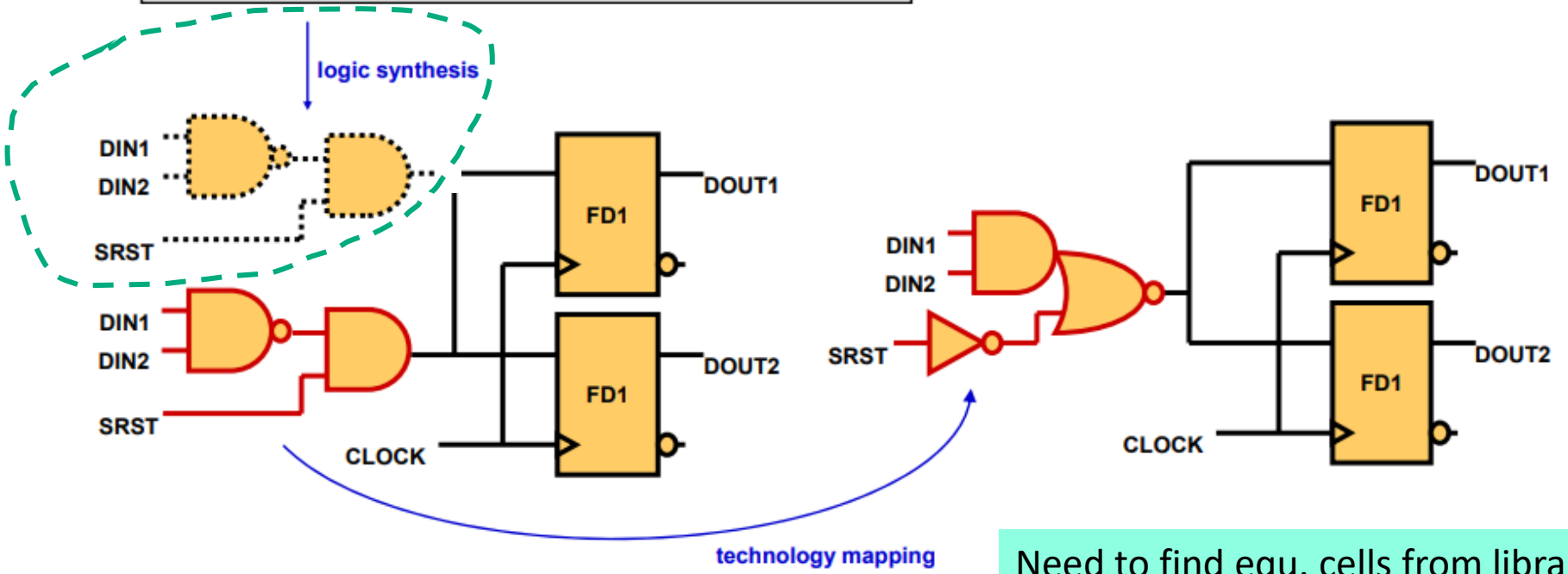




Structural domain

Technology Mapping Problem

```
process
begin
    wait until rising_edge(CLOCK);
    if (SRST='0') then
        DOUT1 <= '0';
    else
        DOUT1 <= DIN1 nand DIN2;
    end if;
    DOUT2 <= SRST and (DIN1 nand DIN2);
end process;
```



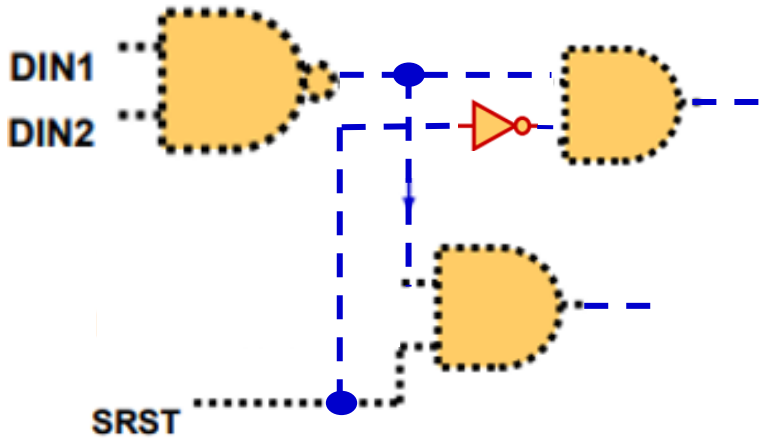
Need to find equ. cells from library

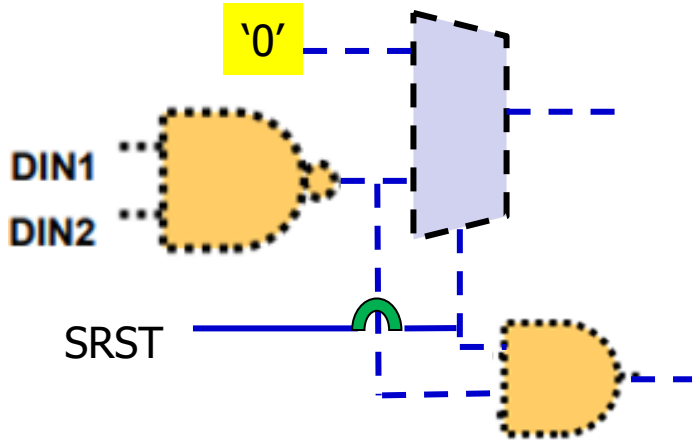
What If We Modify Verilog Statement(s)

```
process
begin
    wait until rising_edge(CLOCK);
    if (SRST= '1' ) then
        DOUT1 <= '0';
    else
        DOUT1 <= DIN1 nand DIN2;
    end if;
    DOUT2 <= SRST and (DIN1 nand DIN2);
end process;
```

How to map statements to combinational logic?

Often we use mux to model if-else





No simplification

5. Digital Standard Cells' descriptions

Inverters: INVX0, INVX1, INVX2, INVX4, INVX8, INVX16, INVX32

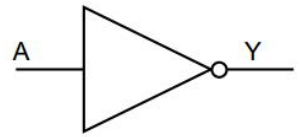


Figure 5.1. Logic Symbol of Inverter

Table 5.1. INV Truth Table

A	Y
0	1
1	0

Table 5.2. Inverter Electrical Parameters and Areas

Cell Name	Operating Conditions: VDD=1.05 V DC, Temp=25 Deg.C, Operating Frequency: Freq=500 MHz, Capacitive Standard Load: Csl=4 fF				Area
	Cload	Prop Delay (Avg)	Power		
			Leakage (VDD=1.05 V DC, Temp=25 Dec.C)	Dynamic	
		ps	nW	nW/MHz	(um2)
INVX0	0.5 x Csl	38	24	0.80	1.27072
INVX1	1 x Csl	30	49	1.30	1.27072
INVX2	2 x Csl	29	100	2.32	1.524864
INVX4	4 x Csl	30	240	4.19	2.033152
INVX8	8 x Csl	28	549	15.80	3.049728
INVX16	16 x Csl	29	1220	15.40	5.08288
INVX32	32 x Csl	29	2580	30.20	9.149184

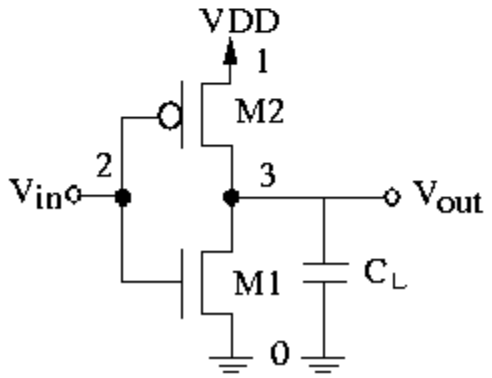
Design Issues and Tools (Cont'd)

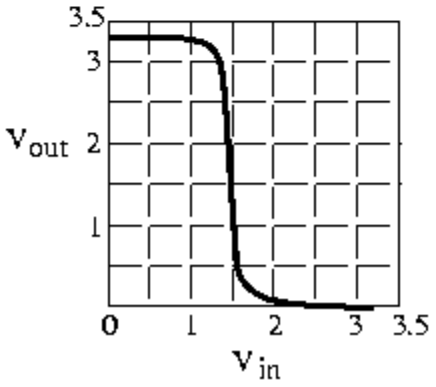
- Transistor-level design (for logic function)
  - Switch-level simulation
  - Circuit simulation
- Physical (layout) design:
  - Partitioning
  - Floorplanning and Placement
  - Routing
  - Layout editing and compaction
  - Design-rule checking
  - Layout extraction
- Design management
  - Data bases, frameworks, etc.
- Silicon compilation: *from algorithm to mask patterns*
  - The *idea* is approached more and more, but still far away from a single *push-button* operations

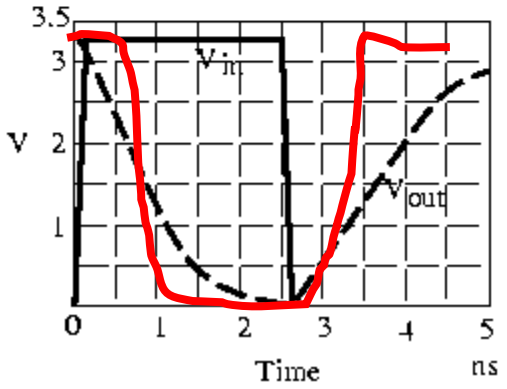
Circuit Simulation of a CMOS Inverter (0.6 μm)

```
M1 3 2 0 0 nch W=1.2u L=0.6u AS=2.16p PS=4.8u AD=2.16p PD=4.8u
M2 3 2 1 1 pch W=1.8u L=0.6u AS=3.24p PS=5.4u AD=3.24p PD=5.4u
CL 3 0 0.2pF

VDD 1 0 3.3
VIN 2 0 DC 0 PULSE (0 3.3 0ns 100ps 100ps 2.4ns 5ns)

.LIB '../mod_06' typical

.OPTION NOMOD POST INGOLD=2 NUMDGT=6 BRIEF
.DC VIN 0V 3.3V 0.001V
.PRINT DC V(3)
.TRAN 0.001N 5N
.PRINT TRAN V(2) V(3)
.END
```

Is this circuit design good?

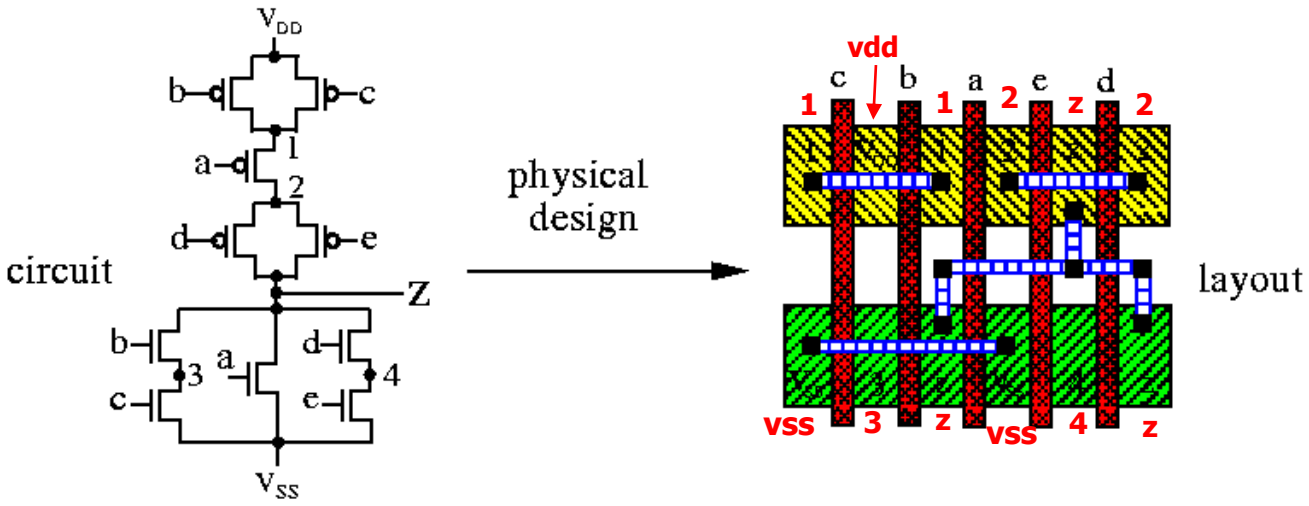




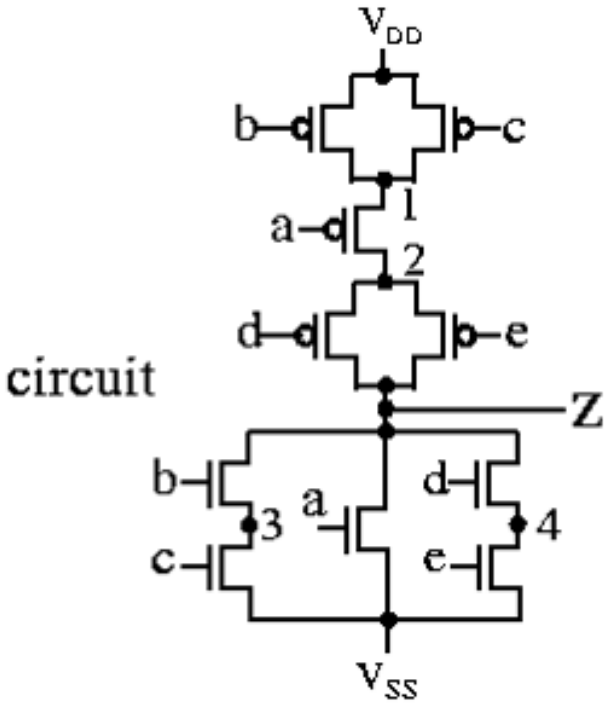


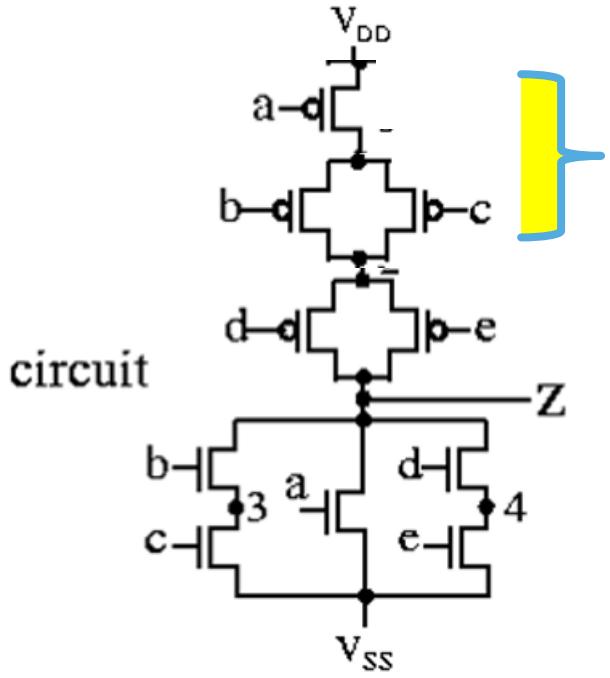
Physical Design

Welcome to change input ordering to, e.g., a,b,c,d,e



- Physical design converts a circuit description into a geometric description
- The description is used to manufacture a chip.
- Physical design cycle:
  1. Logic partitioning
  2. Floorplanning and placement
  3. Routing
  4. DRC fixing, Compaction, …
- Others: circuit extraction, LVS, timing & EM/IR verification, post-layout analyses and debugging, ECO, …


circuit
V_{DD}
b
c
a
1
2
d
e
Z
b
a
d
3
4
c
e
V_{SS}

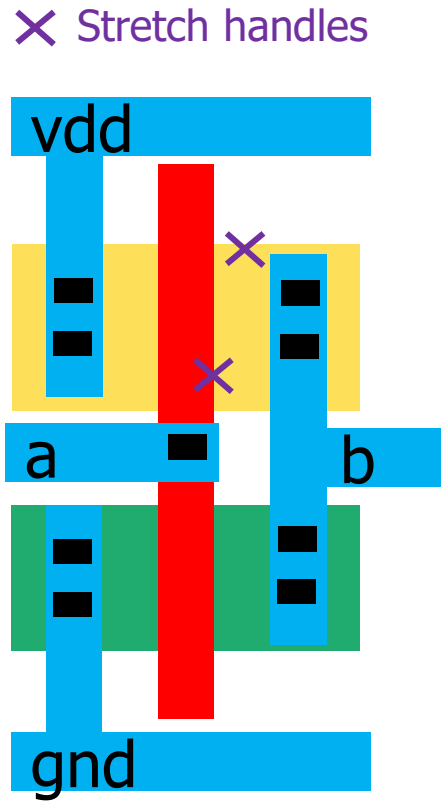

circuit
V_{DD}
a
b
c
d
e
Z
b
a
d
3
4
c
e
V_{SS}

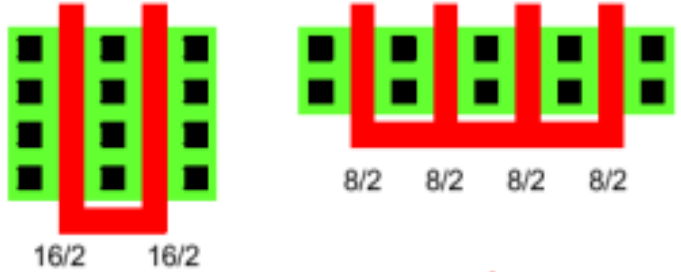
Same or not?

Nowadays Typical Inverter Layout I Saw

- Poly is vertical
- Use fingers to duplicate transistors to get larger width

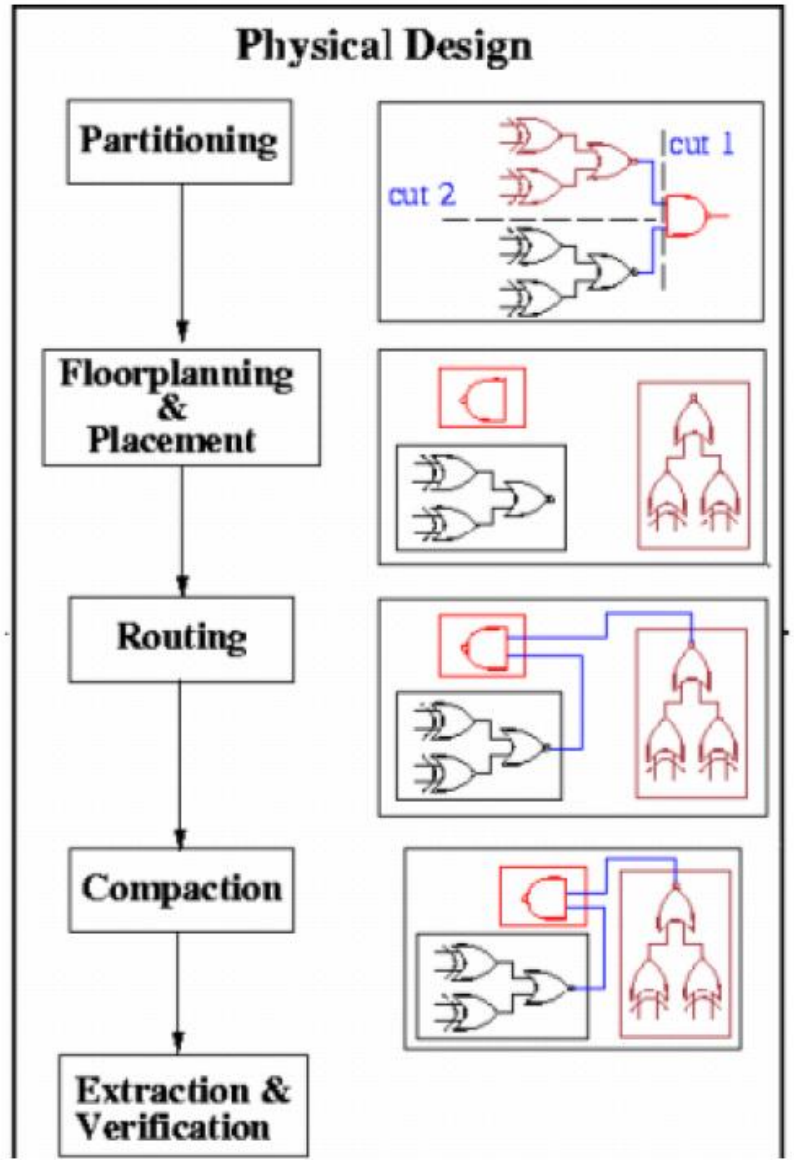
- Graphical Pcell
  - Parameterized cell
- Attach stretch handle to allow easy editing
- PDK - foundation

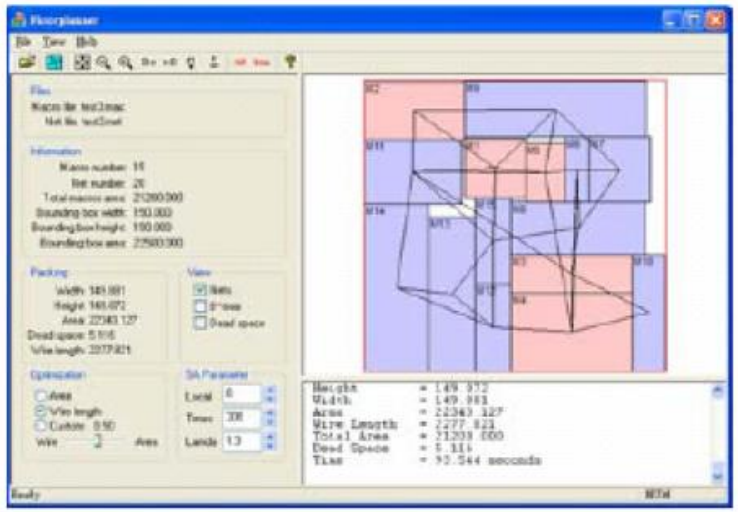




Transistor folding

Physical Design Flow



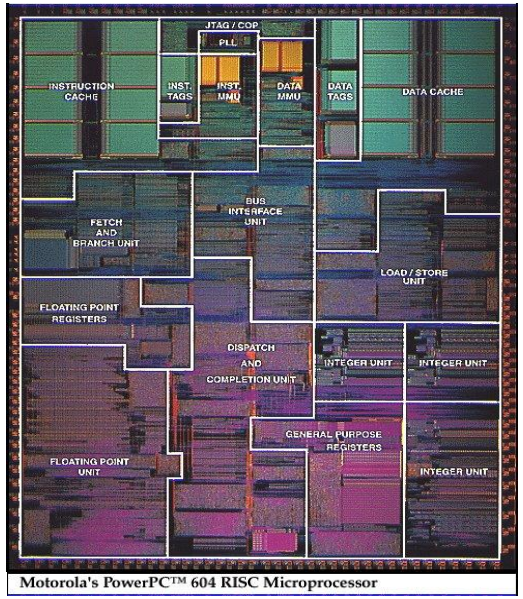


B*-tree based floorplanning system

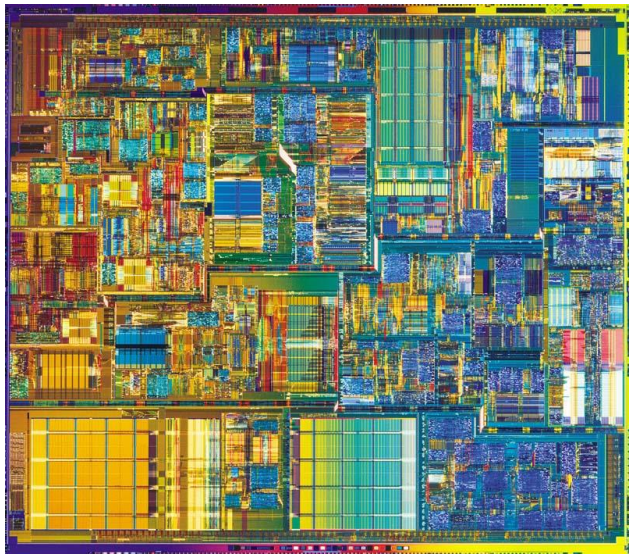
A routing system

Floorplan Examples

PowerPC 604

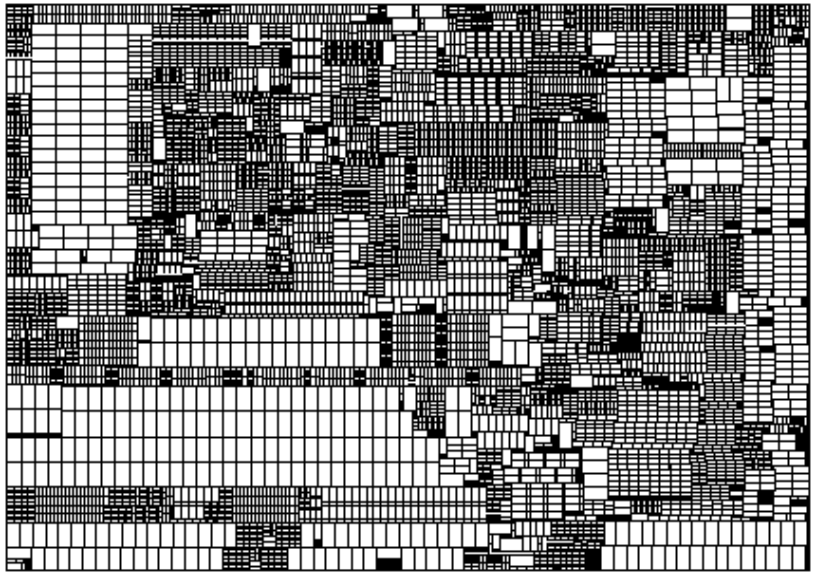


Motorola's PowerPC™ 604 RISC Microprocessor



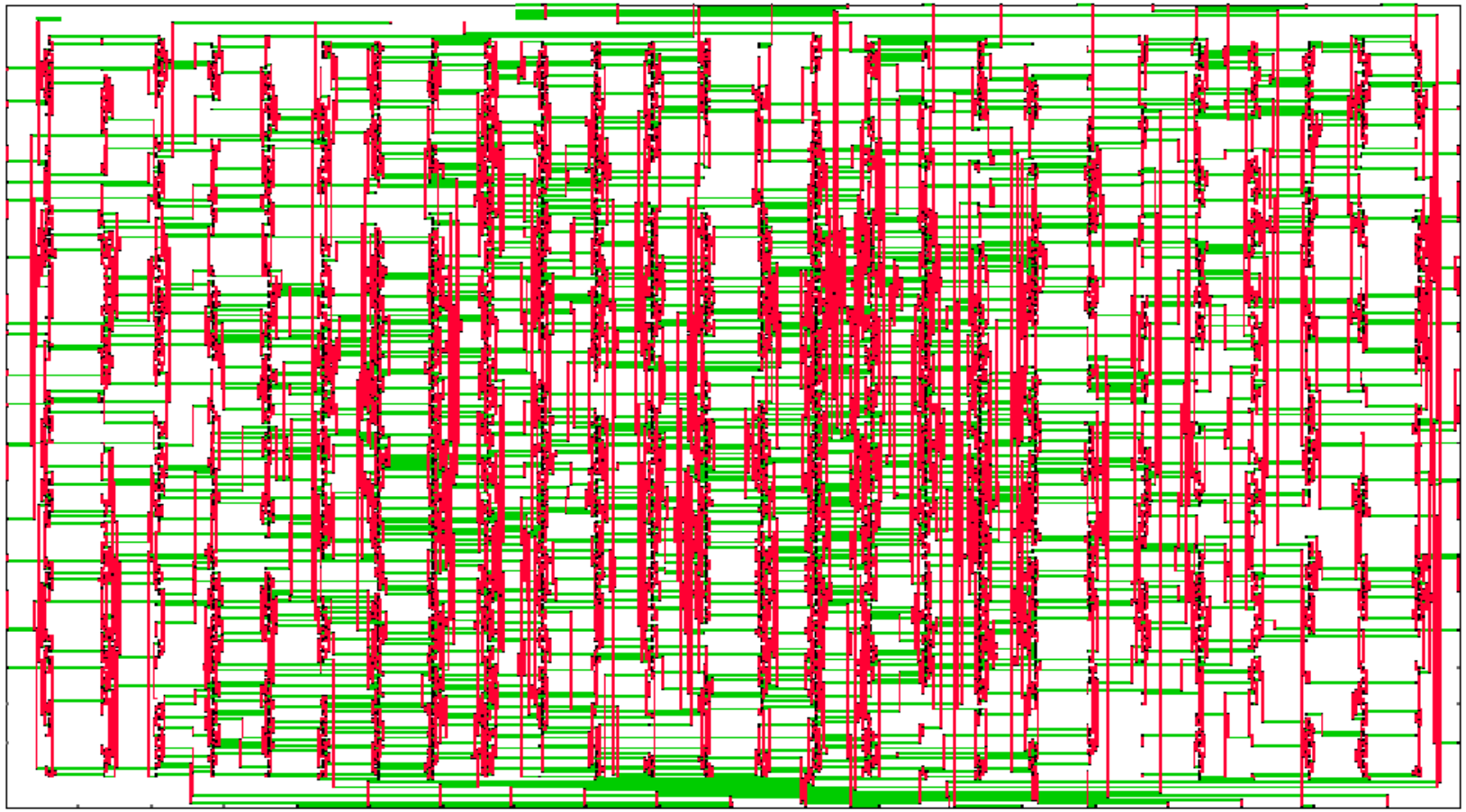
Pentium 4

A floorplan with 9800 blocks



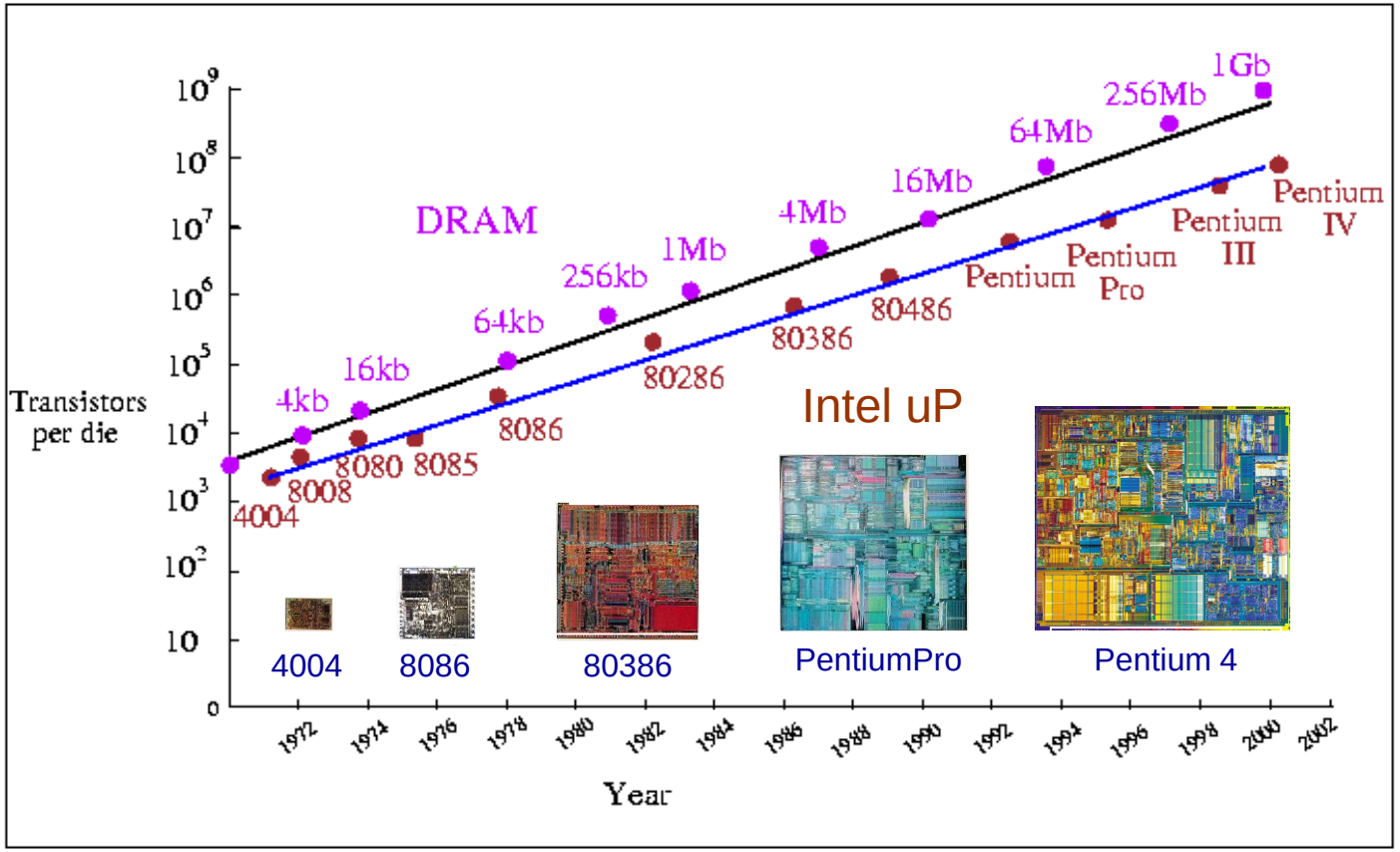
Routing Example

- 0.18um technology, pitch = 1 um, 2774 nets.



“Moore’s” Law: Driving Technology Advances

- Logic capacity doubles per IC at a regular interval.
- Moore: Logic capacity doubles per IC every two years (1975).
- D. House: Computer performance doubles every 18 months (1975)



Technology Roadmap for Semiconductors

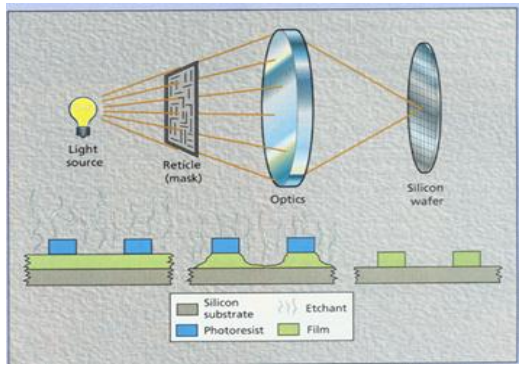
Year	1997	1999	2002	2005	2008	2011	2014
Technology node (nm)	250	180	130	100	70	50	35
On-chip local clock (GHz)	0.75	1.25	2.1	3.5	6.0	10	16.9
Microprocessor chip size (mm^2)	300	340	430	520	620	750	901
Microprocessor transistors/chip	11M	21M	76M	200M	520M	1.40B	3.62B
Microprocessor cost/transistor ($\times 10^{-8}$ USD)	3000	1735	580	255	110	49	22
DRAM bits per chip	256M	1G	4G	16G	64G	256G	1T
Wiring level	6	6–7	7	7–8	8–9	9	10
Supply voltage (V)	1.8–2.5	1.5–1.8	1.2–1.5	0.9–1.2	0.6–0.9	0.5–0.6	0.37–0.42
Power (W)	70	90	130	160	170	175	183

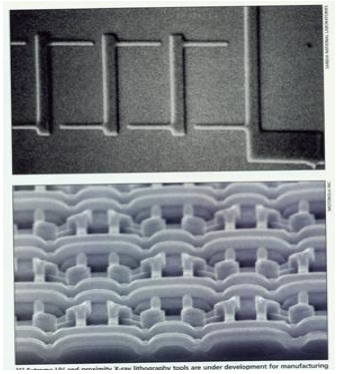
- Source: International Technology Roadmap for Semiconductors, Nov, 2002. http://www.itrs.net/ntrs/publntrs.nsf.
- Deep submicron technology: node (**feature size**) < 0.25 μm.
- Nanometer Technology: node < 0.1 μm.

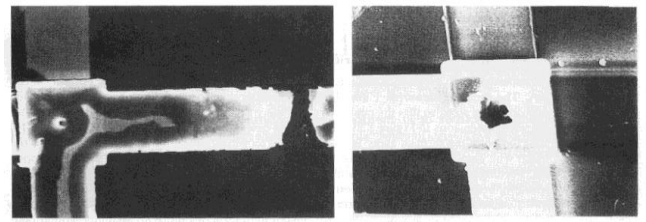
Nanometer Design Challenges (1)

- In 2005,
  - feature size ≈ 0.1 μm,
  - μP frequency ≈ 3.5 GHz,
  - die size ≈ 520 mm^2,
  - μP transistor count per chip ≈ 200M,
  - wiring level ≈ 8 layers,
  - supply voltage ≈ 1 V,
  - power consumption ≈ 160 W.

- Feature size: sub-wavelength lithography (impacts of process variation)? noise? crosstalk? reliability?
- Frequency, dimension: interconnect delay? electromagnetic field effects? timing closure?







(a) Line-open failure. (b) Open failure in contact plug.

Figure 8.30 Electromigration-related failure modes (*Courtesy of N. Cheung and A. Tao, U.C. Berkeley*).

Nanometer Design Challenges (2)

- Chip complexity: large-scale system design methodology?
- Supply voltage: signal integrity (noise, IR drop, etc)?
- Wiring level: manufacturability, yield – (DFM)? 3D layout?
- Power consumption: power, IR drop & thermal issues?

- Design Issues
  - Design space exploration
  - More efficient optimization algorithms

- Verification issues
  - State explosion problem
  - For modern designs, about 60%-80% of the overall design time was spent on verification; 3-to-1 head count ratio between verification engineers and logic designers

https://semiwiki.com/eda/353206-ceo-interview-with-dr-william-wang-of-alpha-ai-design/

That is why some startup is trying: Alpha Design AI

We're solving three key pain points:

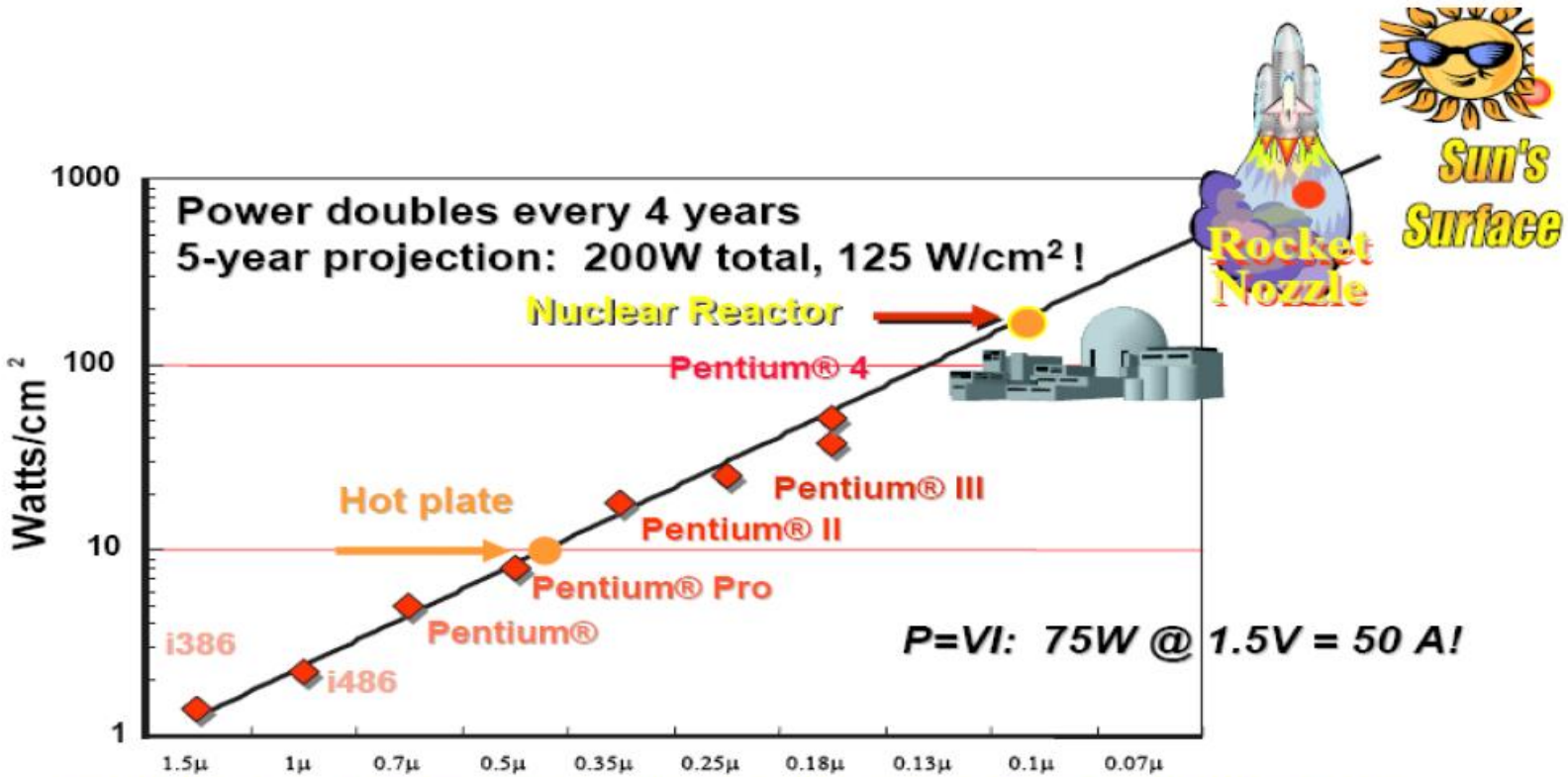
1.Reducing Debug Time: Engineers spend countless hours troubleshooting RTL and testbenches. Our solution helps users understand their design requirements, the RTL code, testbenches, and accelerates bug resolution by working with ChipAgents and automating tedious tasks.

2.Improving Verification Efficiency: Traditional verification relies on manual testbench creation and constraint tuning. ChipAgents can generate, optimize, and validate testbenches, boosting functional coverage.

3.Enhancing Productivity with AI-Driven Insights: Unlike generic coding assistants, ChipAgents understands semiconductor-specific workflows, integrating deeply with EDA tools to provide actionable insights rather than generic code completions.

Power Dissipation Challenges

- Power density increases exponentially!



Fred Pollack, "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies," 1999 Micro32 Conference keynote. Courtesy Avi Mendelson, Intel.

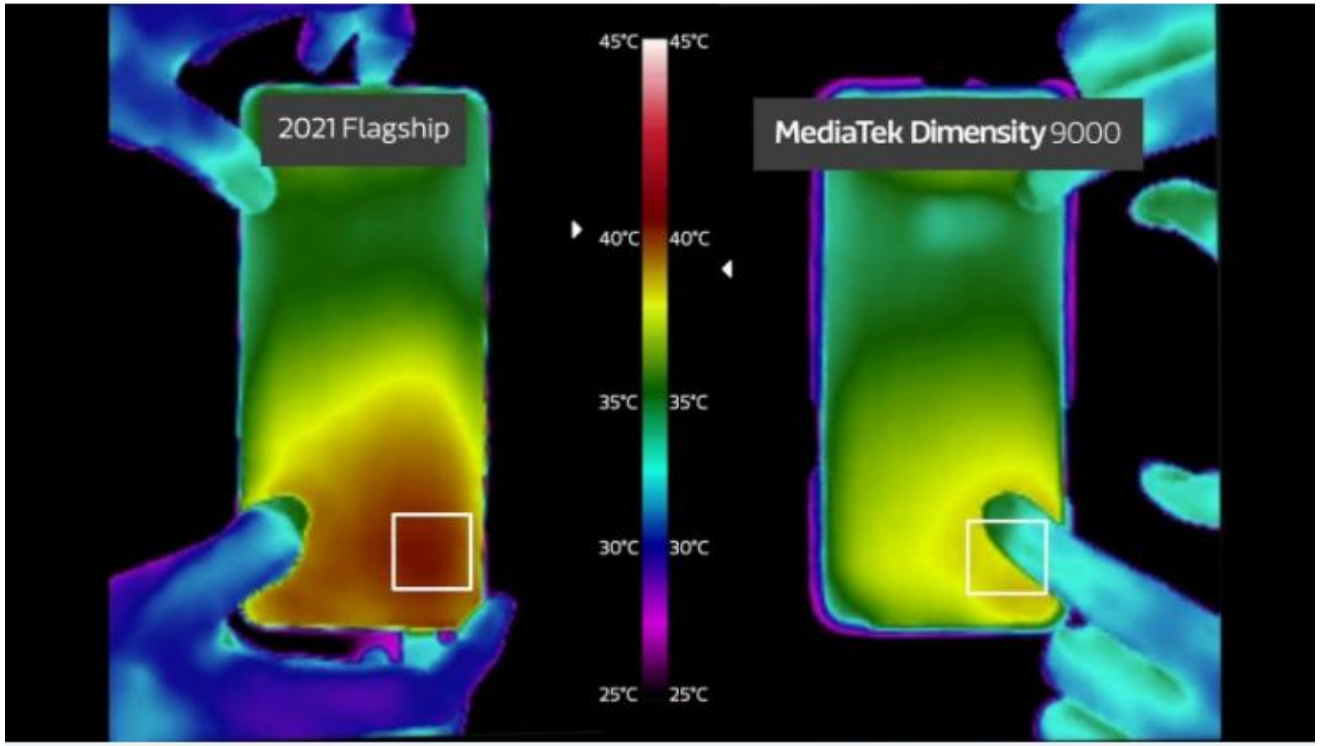
An example showing power consumption minimization

MediaTek
201,054 followers
20h •

+ Follow

Take a look at the superb power efficiency of the Flagship MediaTek Dimensity 9000 during gaming. With its 4nm manufacturing process and our comprehensive power optimizations across CPU, GPU, APU, ISP, and 5G modem, the Dimensity 9000 achieves sustainable flagship-grade performance while keeping cool. https://lnkd.in/dzgCsYf9

#MediaTek5G #MediaTekDimensity9000



MediaTek Dimensity 9000 : Thermal testing demonstrating cooler gaming
youtube.com

Another Example Of Power Optimization

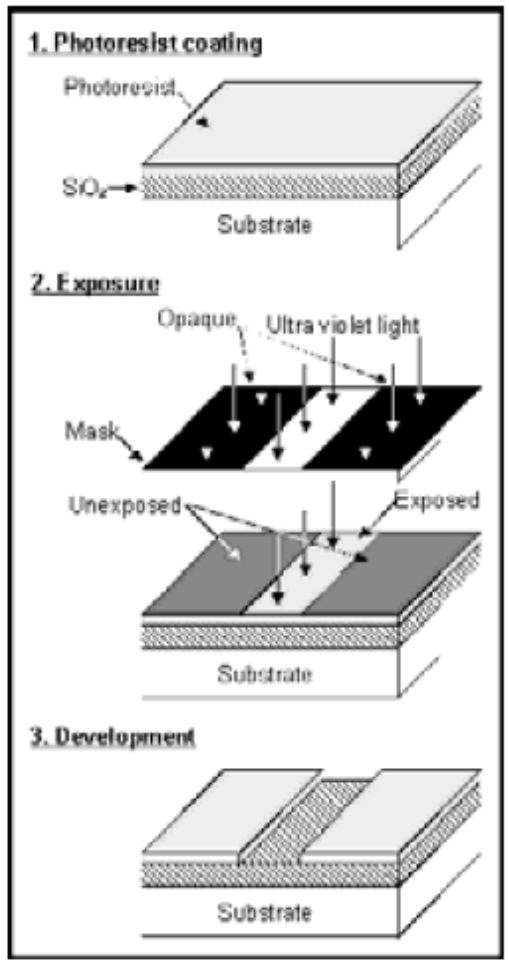
https://www.wsj.com/articles/the-chips-that-rebooted-the-mac-11650081649?mod=e2li

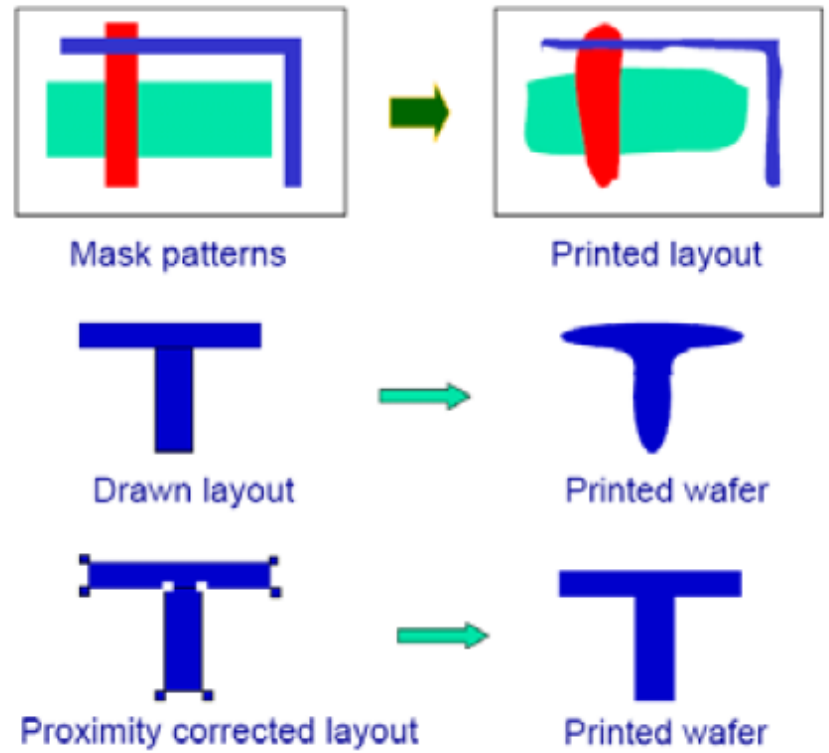
The Chips That Rebooted the Mac

Any thoughts about how to reduce power consumption?

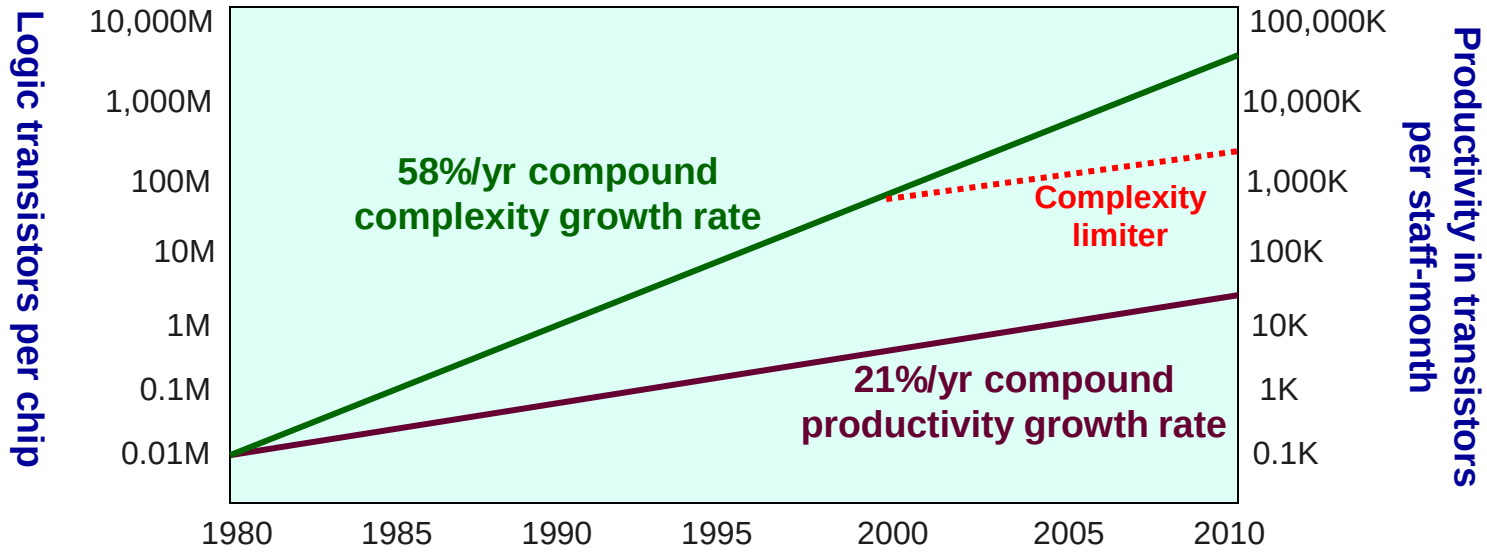
Semiconductor Fabrication Challenges

- Feature-size shrinking approaches physical limitation





Design Productivity Challenges



- Human factors may limit design more than technology.
- Keys to solve the productivity crisis: hierarchical design, abstraction, **CAD (tool & methodology),** IP reuse, etc.

https://ieeexplore.ieee.org/abstract/document/6649058
FinFETs - Technology and circuit design challenges

Agentic AI is coming in

Recap

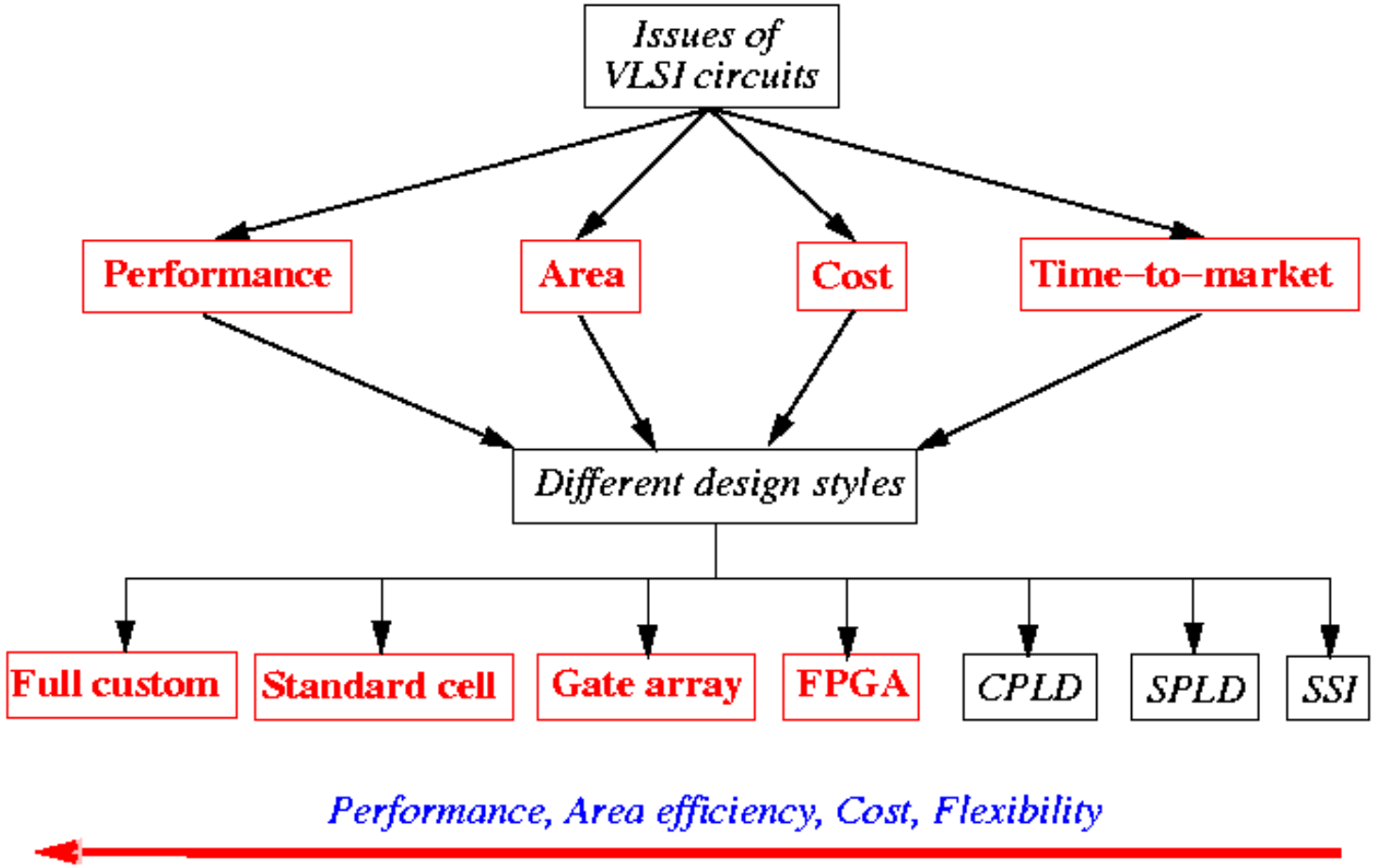
- Explained the importance of EDA
- Covered major components in design flows
  - Digital SoC, custom/analog design, OPC, advanced packaging
- What is needed in developing EDA tools
  - Nowadays need to learn AI/ML technologies; startup opportunity
- Brief introduction of synthesis
  - Using mux a lot in the “domain translation”
- Pictures to show a transistor design and a layout solution
  - Welcome to think about if such a layout is good enough
  - Also explained transistor folding (i.e., multi-fingers)
- Started to show design styles
  - Good to see someone asking about emulation

Design Styles

- There are various design styles:
  - Full custom, standard cell, sea of gates, FPGA, etc.
- Why having different design styles?

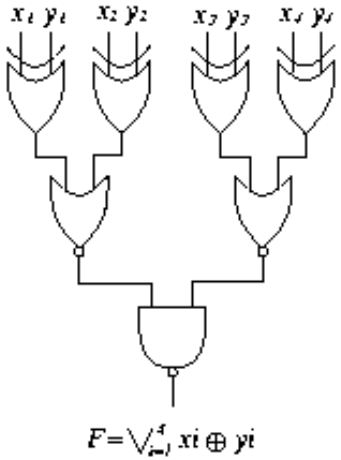
Design Styles

- Specific design styles shall require specific CAD tools

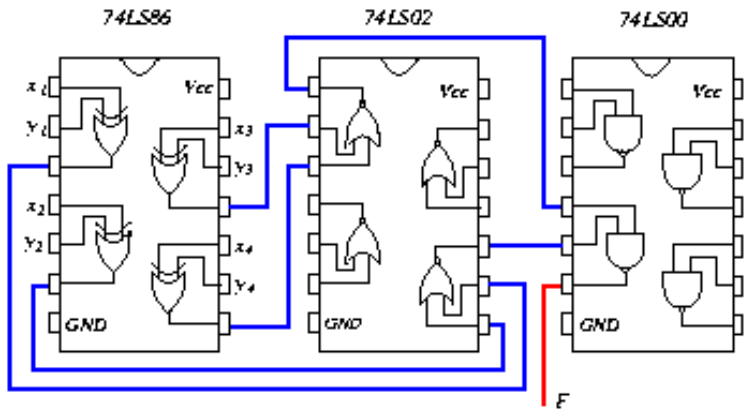


SSI: small scale integration

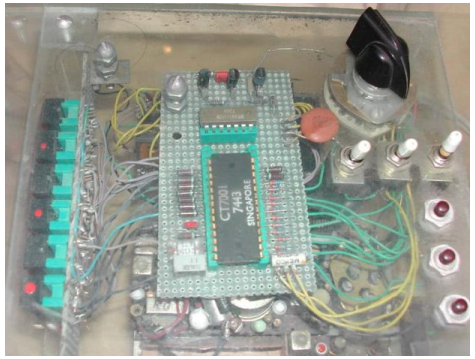
SSI / SPLD Design Style

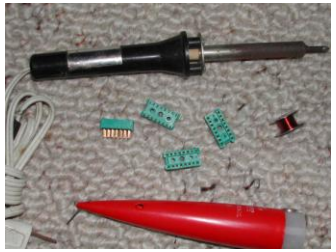


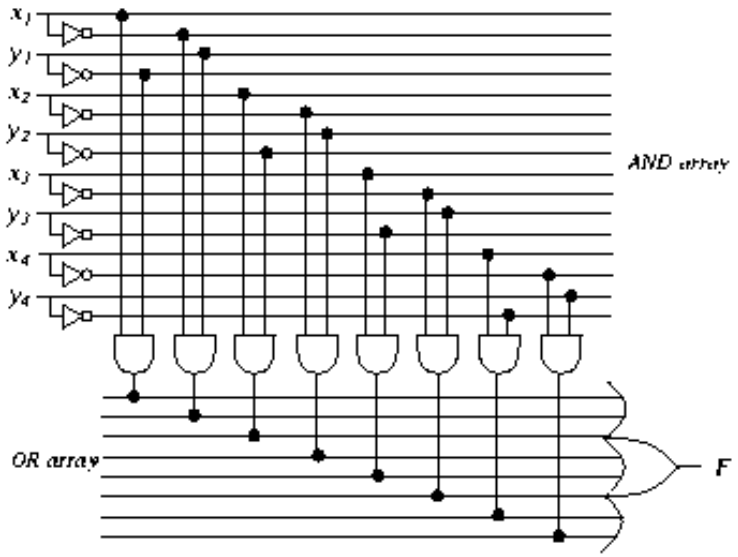
(a) 4–bit comparator.



(b) SSI implementation.





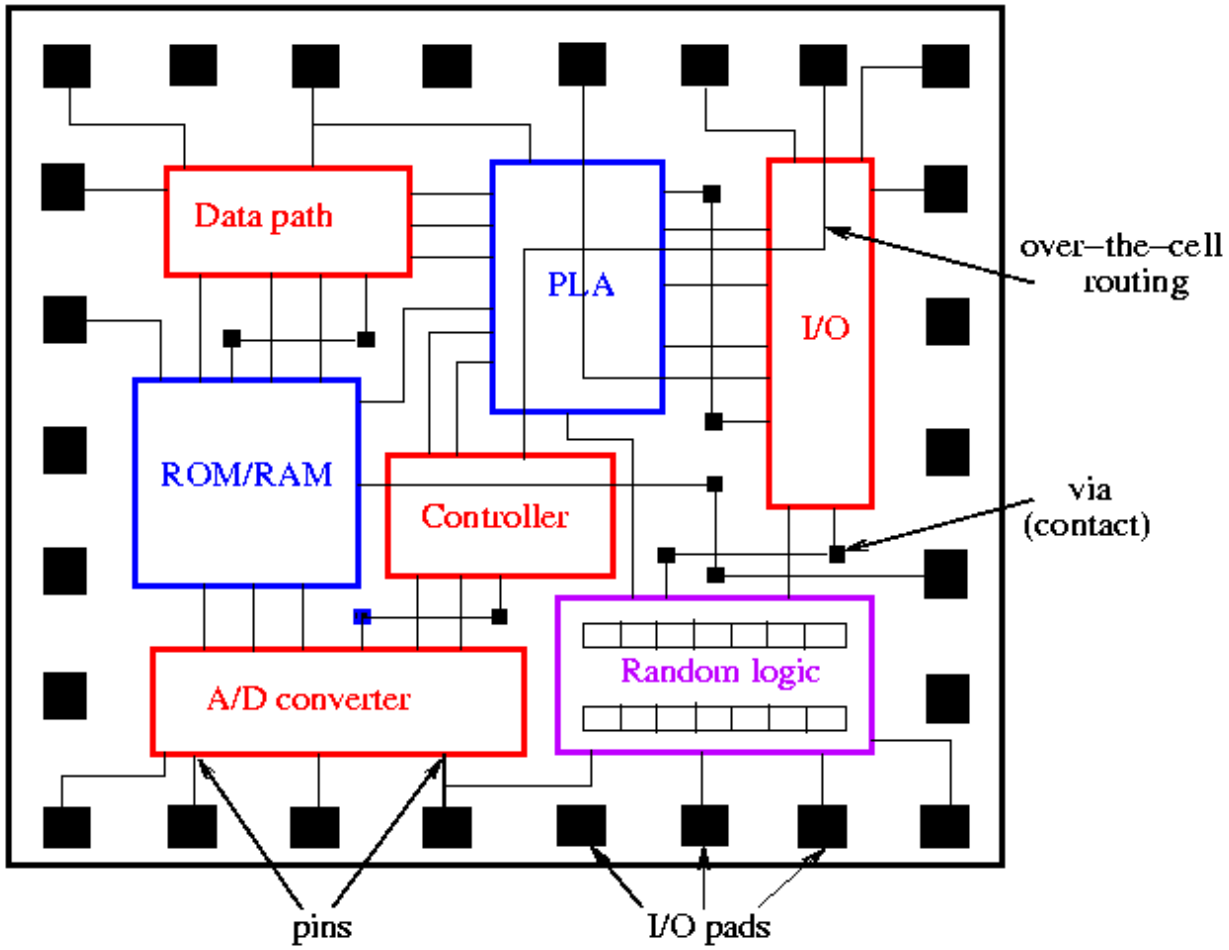


(c) SPLD (PLA) implementation.

Predecessor of FPGA

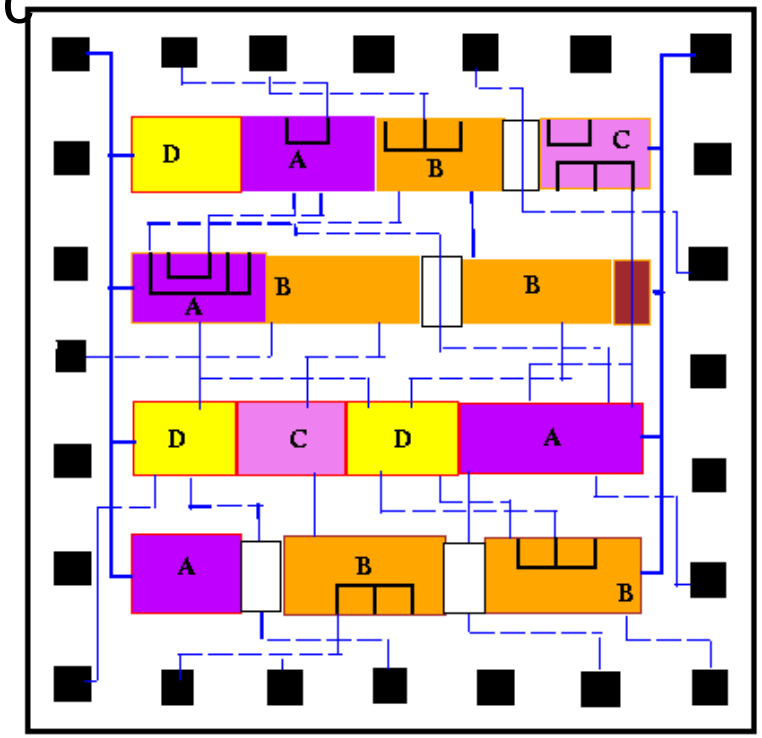
Full Custom Design Style

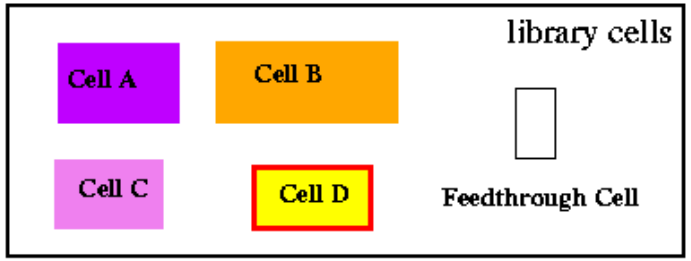
- Designers can control the shape of all mask patterns.
- Designers can specify the design up to the level of individual transistors



Standard Cell Design Style

- Selects pre-designed cells (of same height) to implement logic





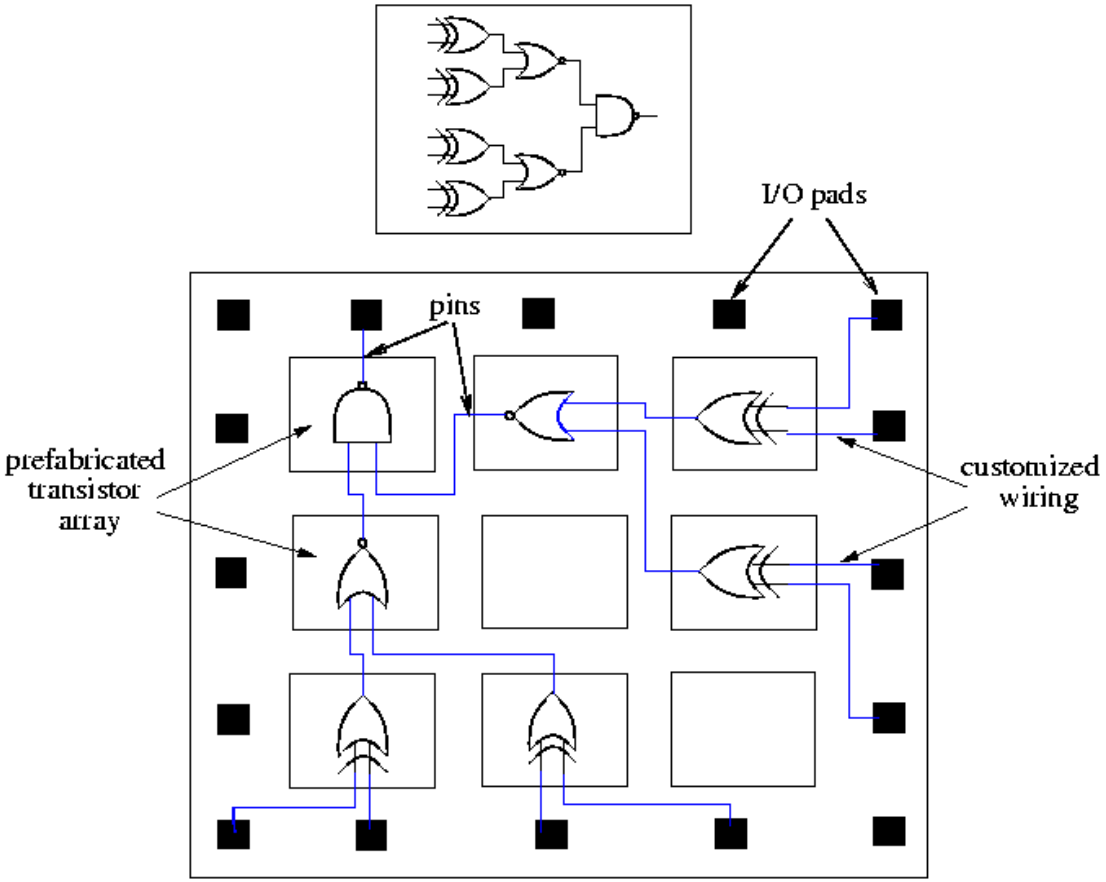
There are cell rows of same height;

Heights of routing channels vary → so chip size is not fixed; (but not today's style)

Standard Cell Example

Gate Array Design Style

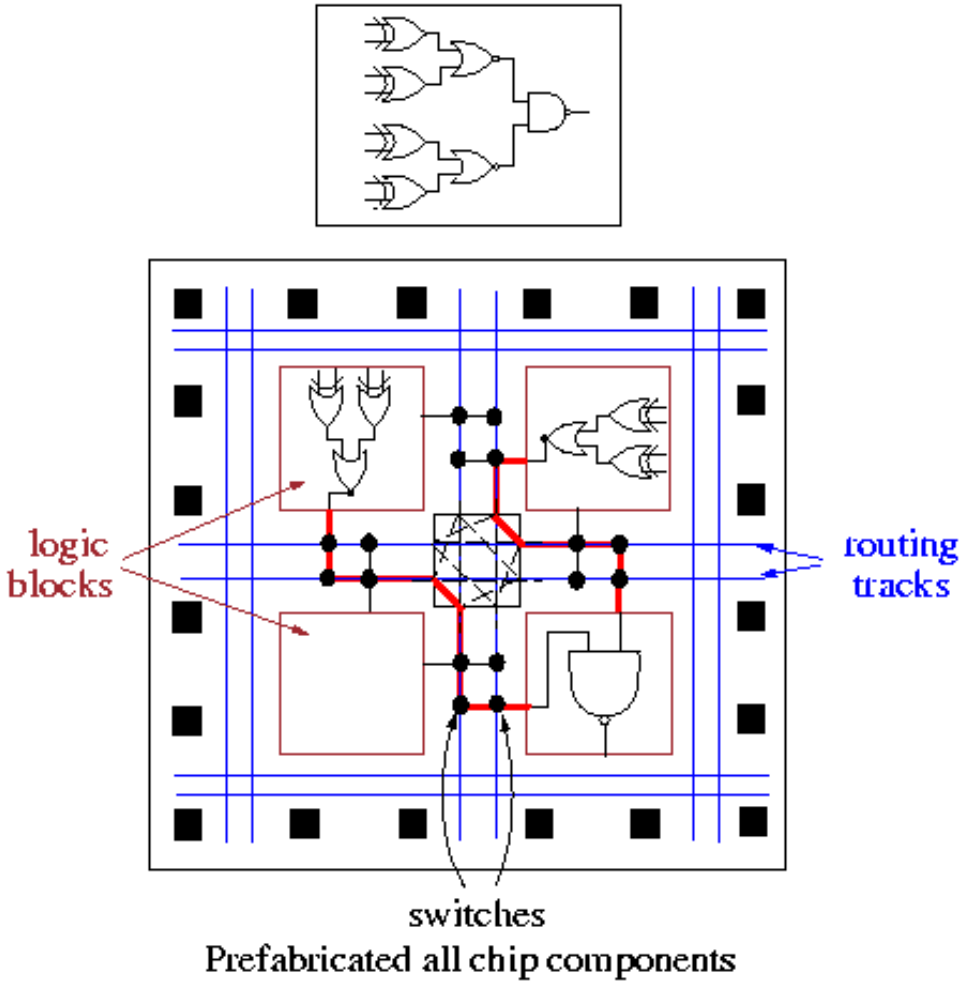
- Prefabricates a transistor array
- Needs wiring customization to implement logic



Cell row's heights are same;
Row spacing is same
→ fixed chip size

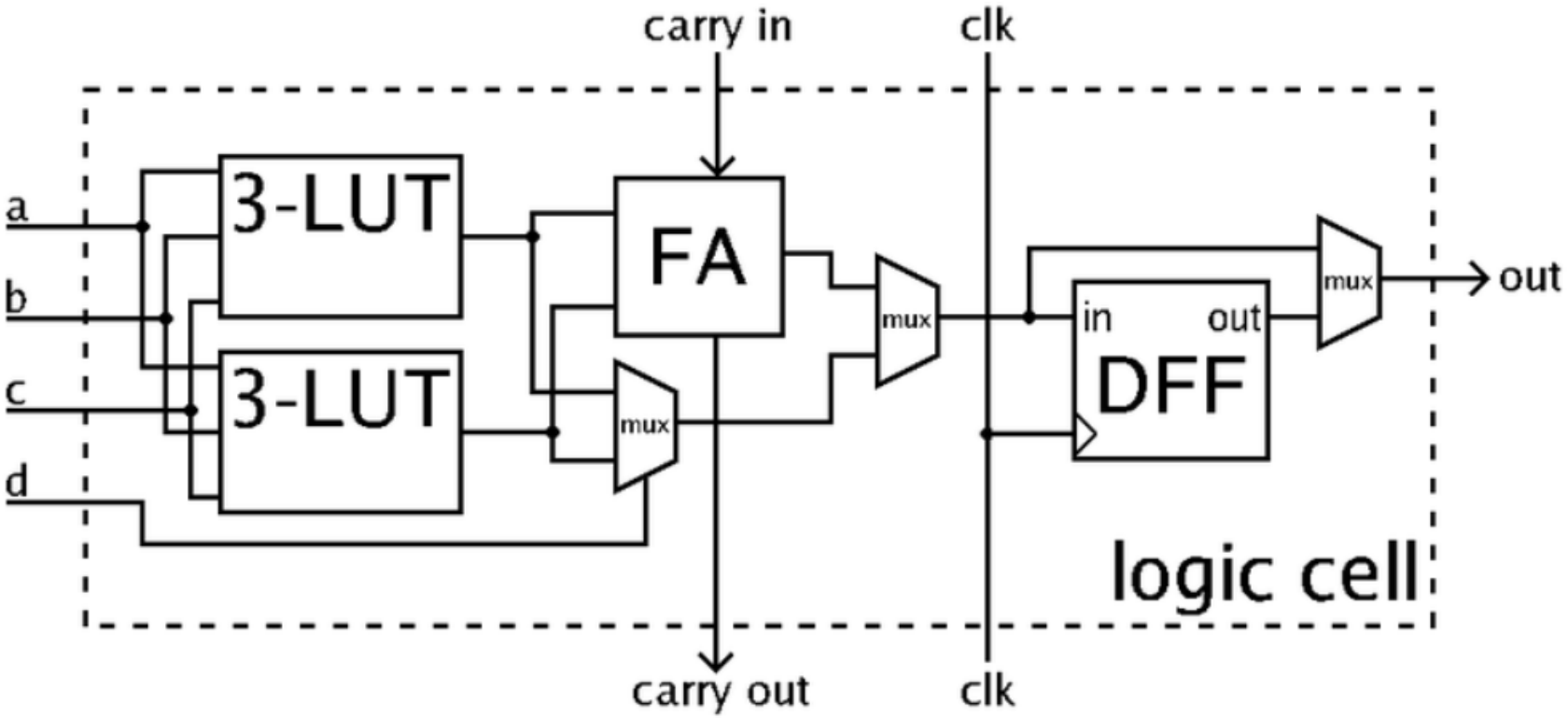
FPGA Design Style

- Logic and interconnects are both prefabricated.
- Illustrated by a symmetric array-based FPGA



By downloading 0101… bit stream, chip function is changed

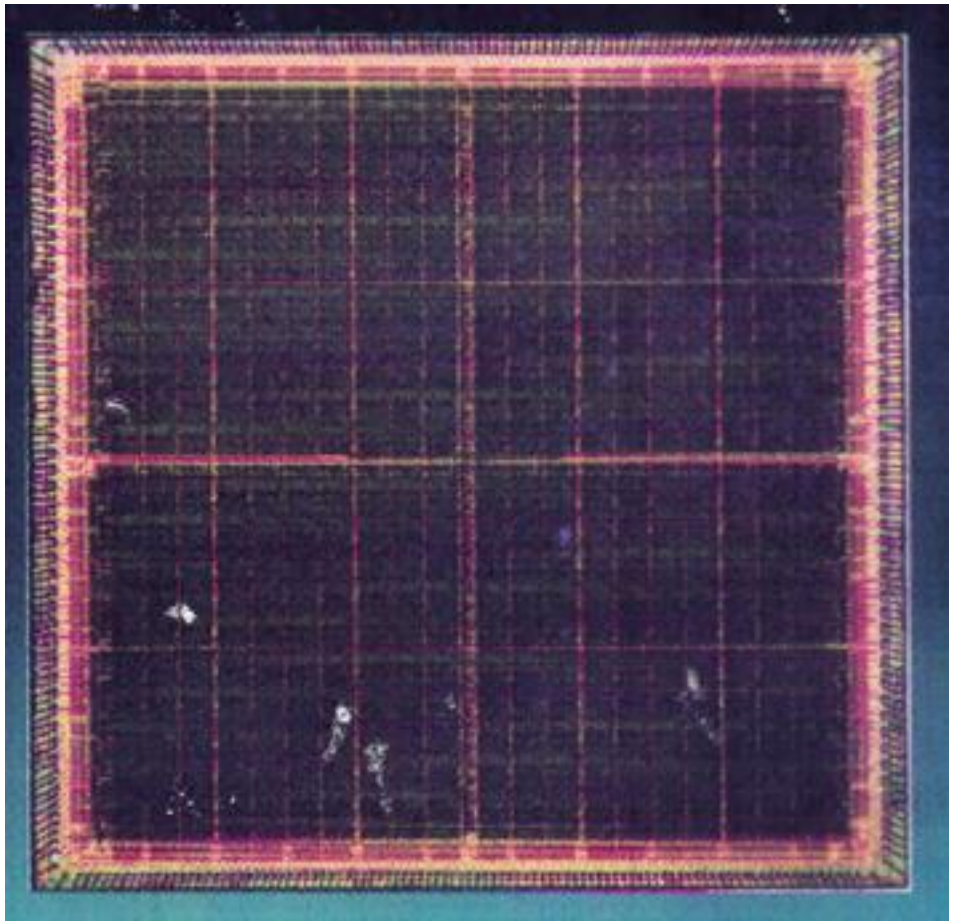
Now it is the main force behind the emulation



Simplified illustration of a logic cell in FPGA

Array-Based FPGA Example

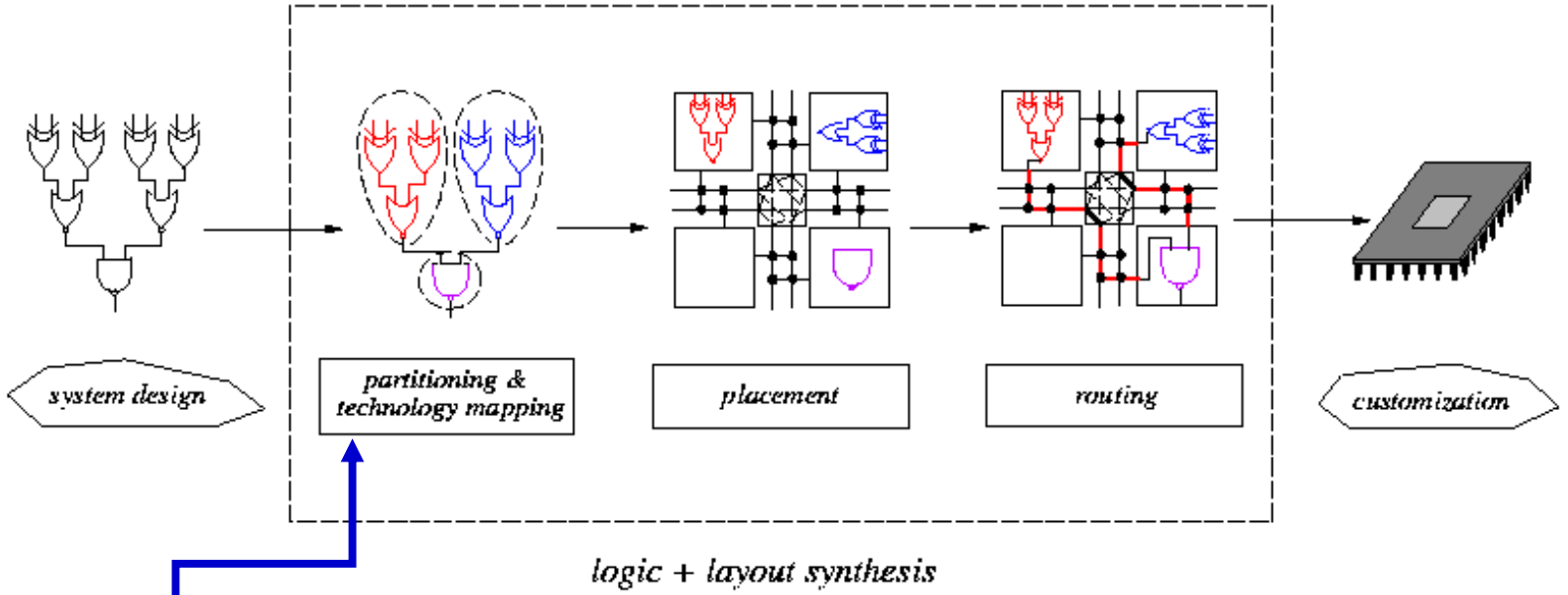
- Lucent Technologies 15K ORCA FPGA



- 0.5 um 3LM CMOS
- 2.45 M Transistors
- 1600 Flip-flops
- 25K bit user RAM
- 320 I/Os

FPGA Design Process

- Illustrated by a symmetric array-based FPGA
- No fabrication is needed



Another design

If a design can be fit into 1 FPGA, … What if can't?

Comparisons of Design Styles

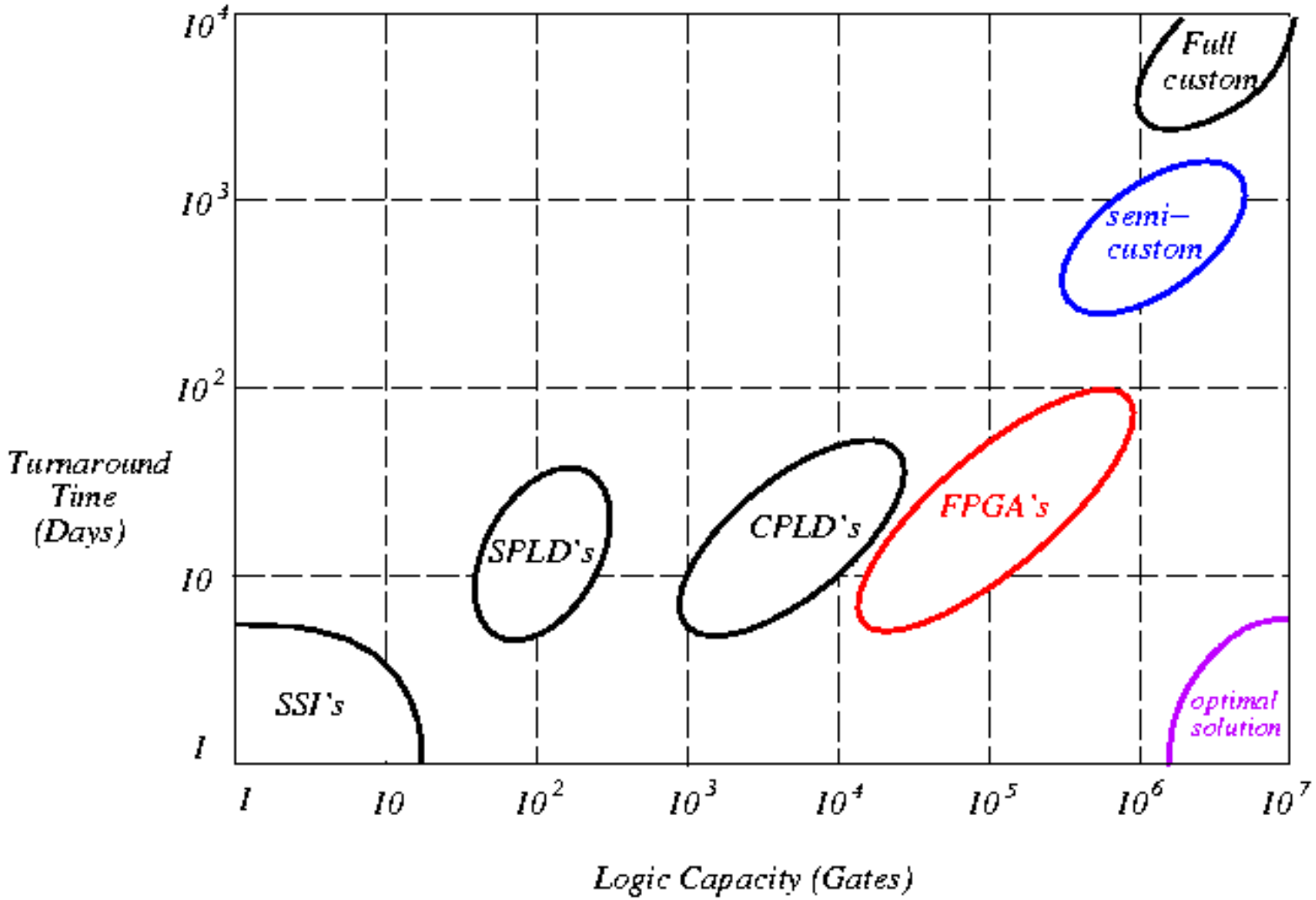
	Full custom	Standard cell	Gate array	FPGA	SPLD
Cell size	variable	fixed height*	fixed	fixed	fixed
Cell type	variable	variable	fixed	programmable	programmable
Cell placement	variable	in row	fixed	fixed	fixed
Interconnections	variable	variable	variable	programmable	programmable

* Uneven height cells are also used.

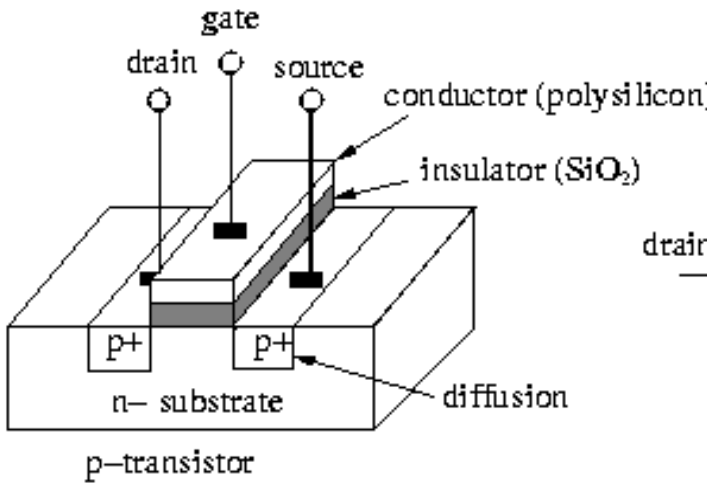
	Full custom	Standard cell	Gate array	FPGA	SPLD
Fabrication time	− − −	− −	+	+++	++
Packing density	+++	++	+	− −	− − −
Unit cost in large quantity	+++	++	+	− −	−
Unit cost in small quantity	− − −	− −	+	+++	++
Easy design and simulation	− − −	− −	−	++	+
Easy design change	− − −	− −	−	++	++
Accuracy of timing simulation	−	−	−	+	++
Chip speed	+++	++	+	−	−−

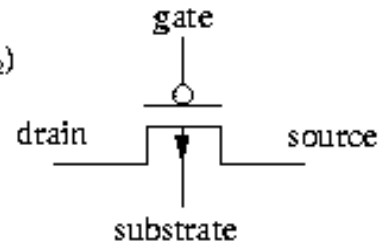
+ desirable; − not desirable

Design Style Trade-offs



MOS Transistors



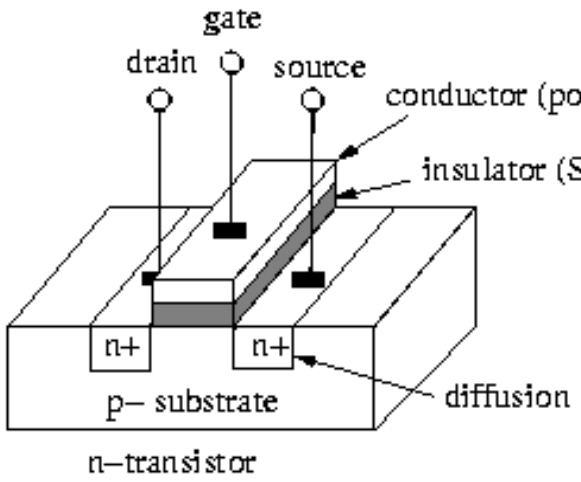


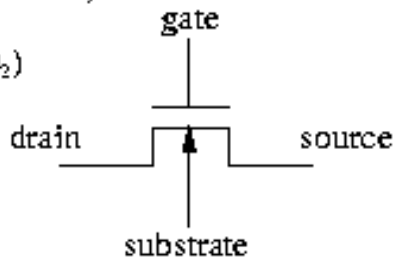
pMOS switch

- control input: gate
- switch terminals: drain, source (physically equivalent)
- apply zero voltage to gate ⟹ switch ON

The pMOS switch passes signal "1" well.

If Gnd is applied to PMOS gate, it will be turned on





nMOS switch

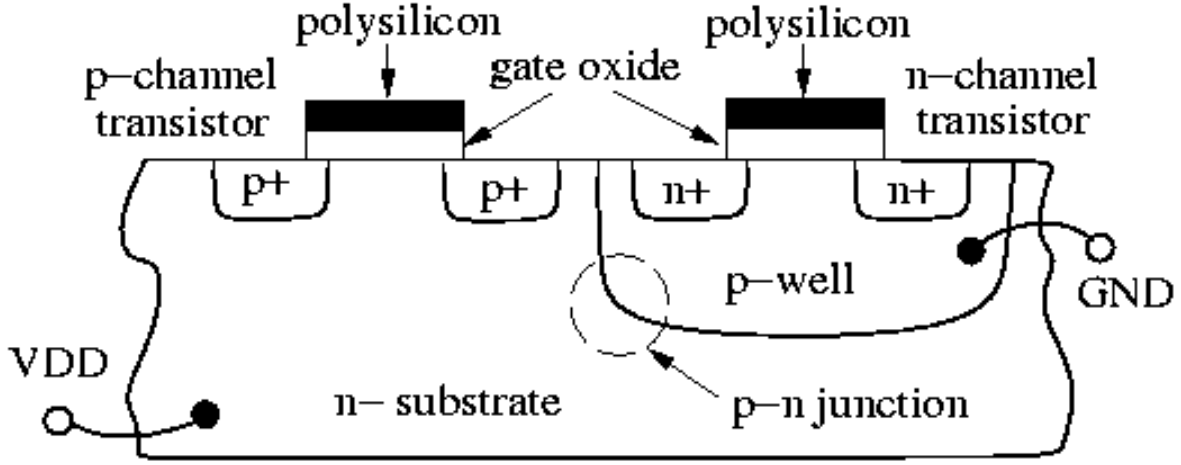
- apply > threshold voltage to gate ⟹ switch ON

The nMOS switch passes signal "0" well.

If Vdd is applied to NMOS gate, it will be turned on

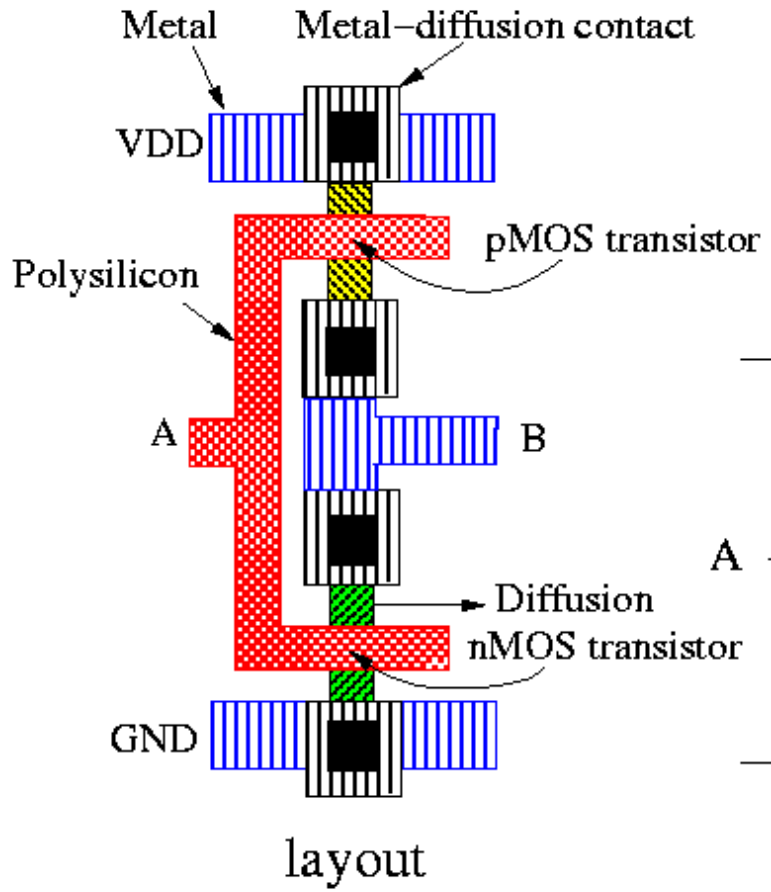
Complementary MOS (CMOS)

- The most popular VLSI technology (v.s. BiCMOS, nMOS).
- CMOS uses both *n*-channel and *p*-channel transistors.
- Advantages: lower power dissipation, higher regularity, more reliable performance, higher noise margin, larger fanout, etc.
- Each type of transistor must sit in a material of the complementary type (the reverse-biased diodes prevent unwanted current flow).



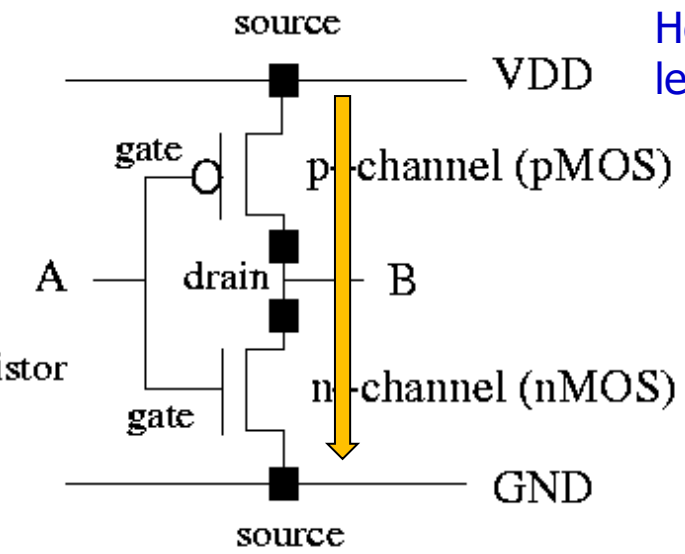
p–well CMOS structure

A CMOS Inverter



A	B
1	0
0	1

B=(A)'



Only exists a conducting path from vdd to gnd for a short while; Hence, less leakage current

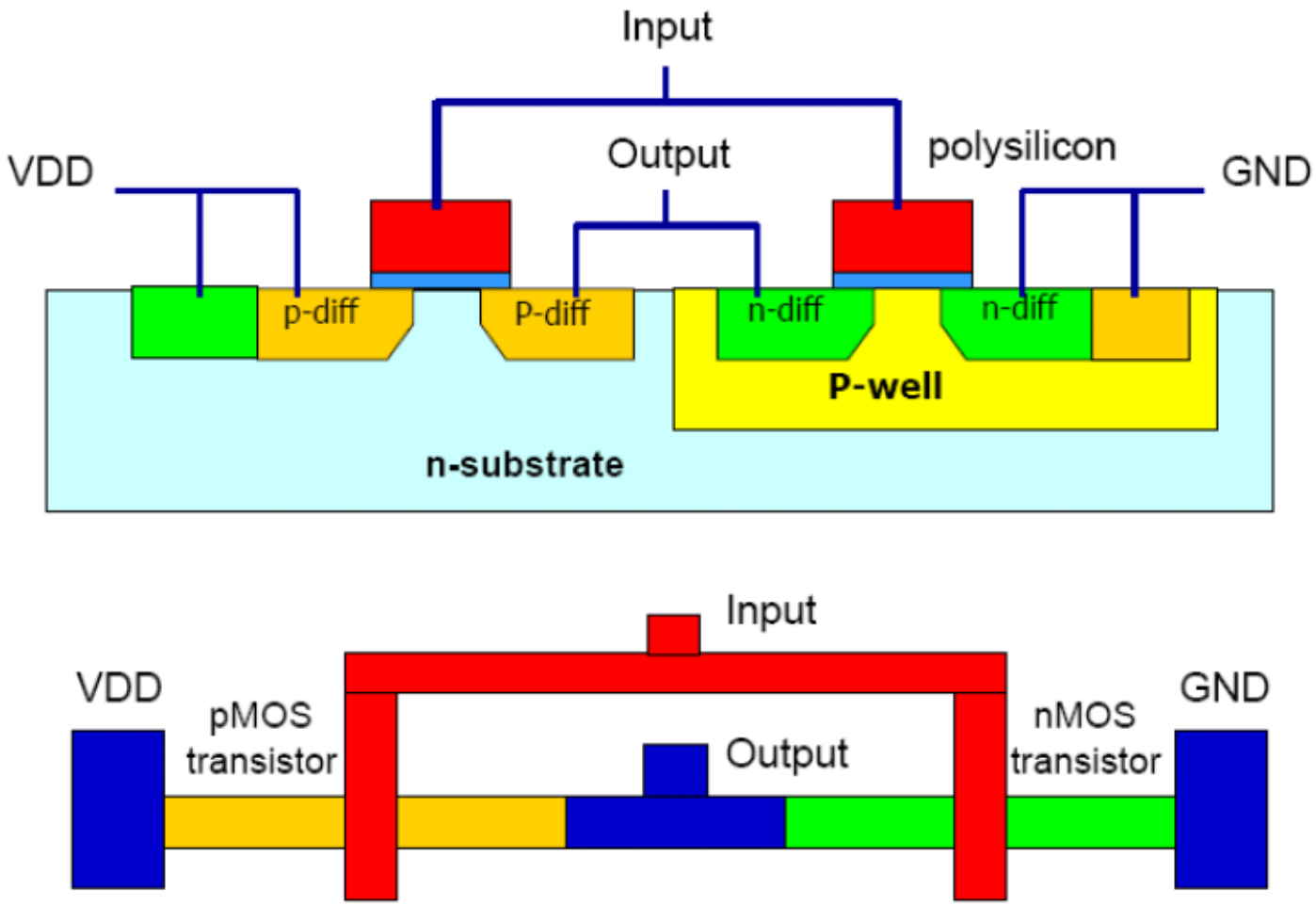
metal 1: blue
polysilicon: red
p–diffusion: yellow (p–well: light yellow)
metal 2: brown
contact/via: black
n–diffusion: green (n–well: light green)

CMOS Properties

- There is always a path from one supply (VDD or GND) to the output.
- There is never a path from one supply to the other. (This is the basis for the low power dissipation in CMOS -- virtually no static power dissipation.)
- There is a momentary drain of current (and thus power consumption) when the gate switches from one state to another.
  - Thus, CMOS circuits have dynamic power dissipation.
  - The amount of power depends on the switching frequency.

$$P = \frac{1}{2} C V_{dd}^2 f$$

CMOS Inverter Cross Section



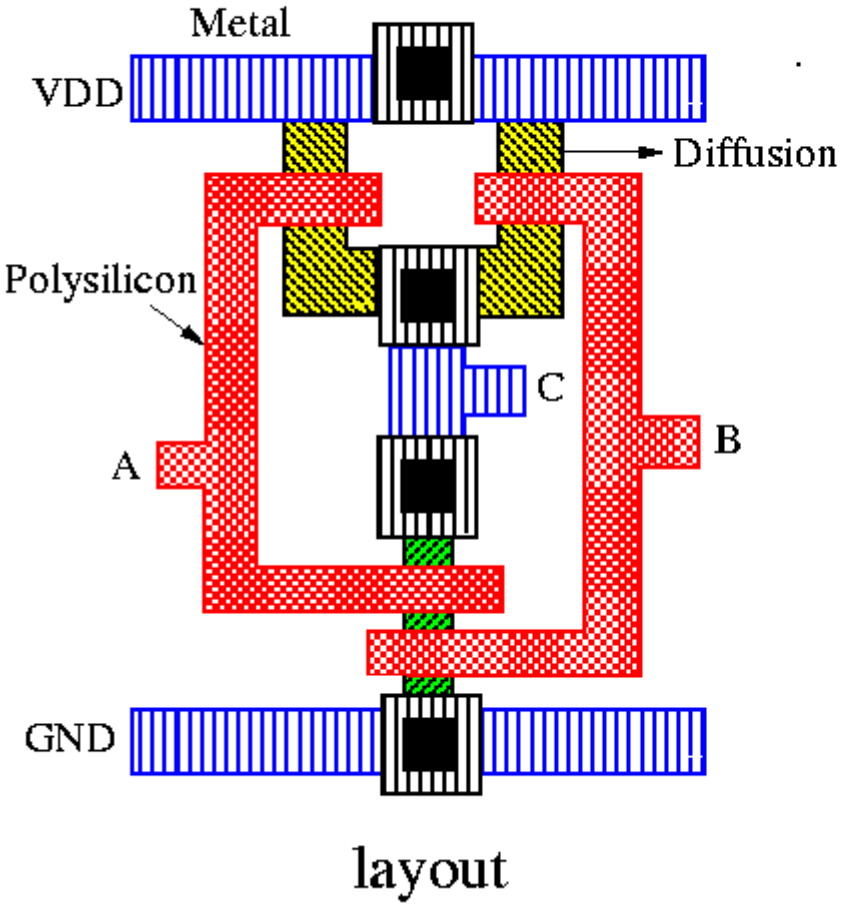
VDD
p-diff
p-diff
n-substrate
Input
Output
n-diff
p-well
polysilicon
n-diff
GND

vdd
a
b
gnd

16/2
16/2

8/2
8/2
8/2
8/2

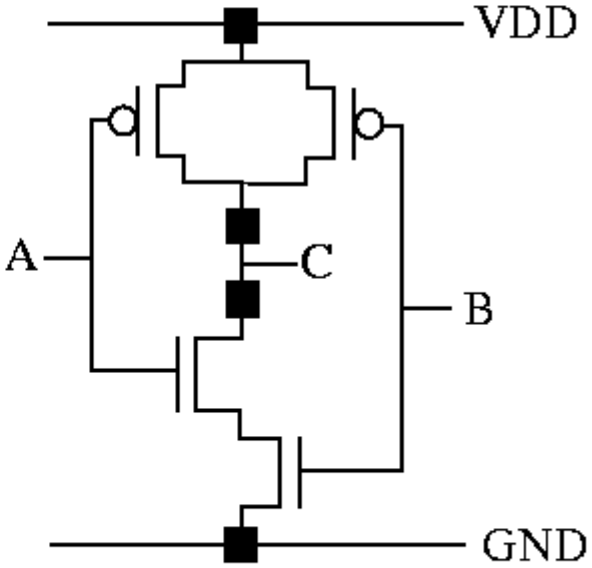
A CMOS NAND Gate



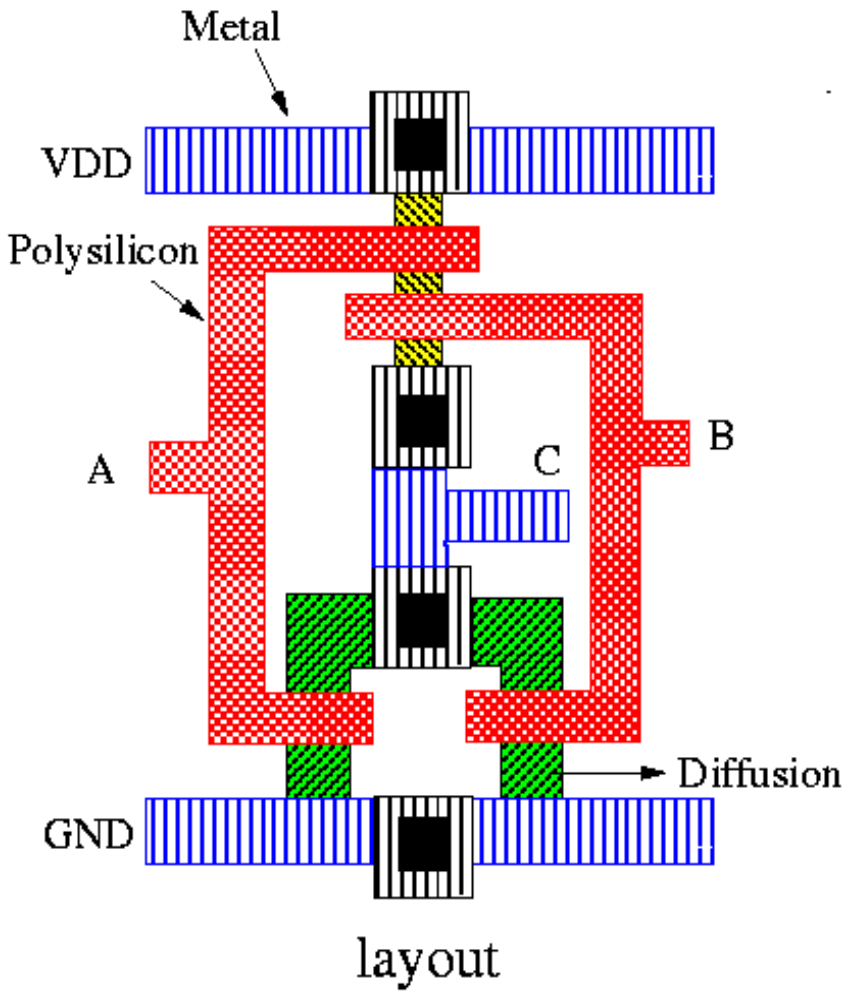
layout

in1	in2	out
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

C=(A * B)'



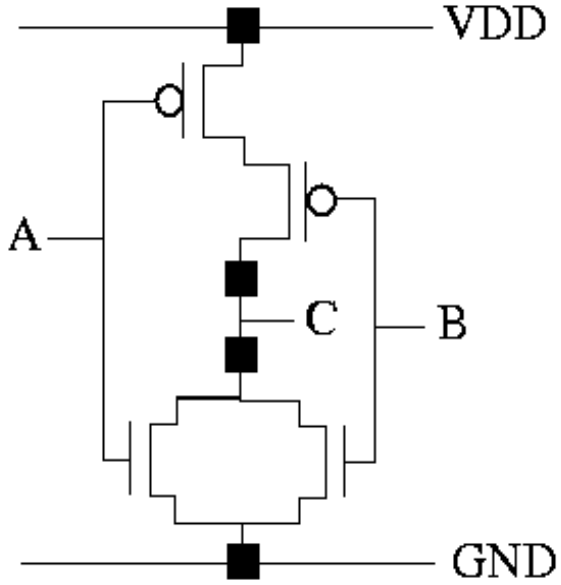
A CMOS NOR Gate



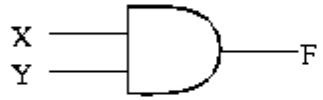
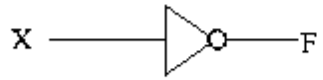
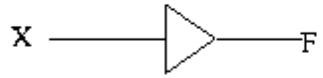
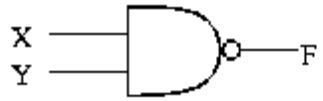
layout

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

C=(A + B)’

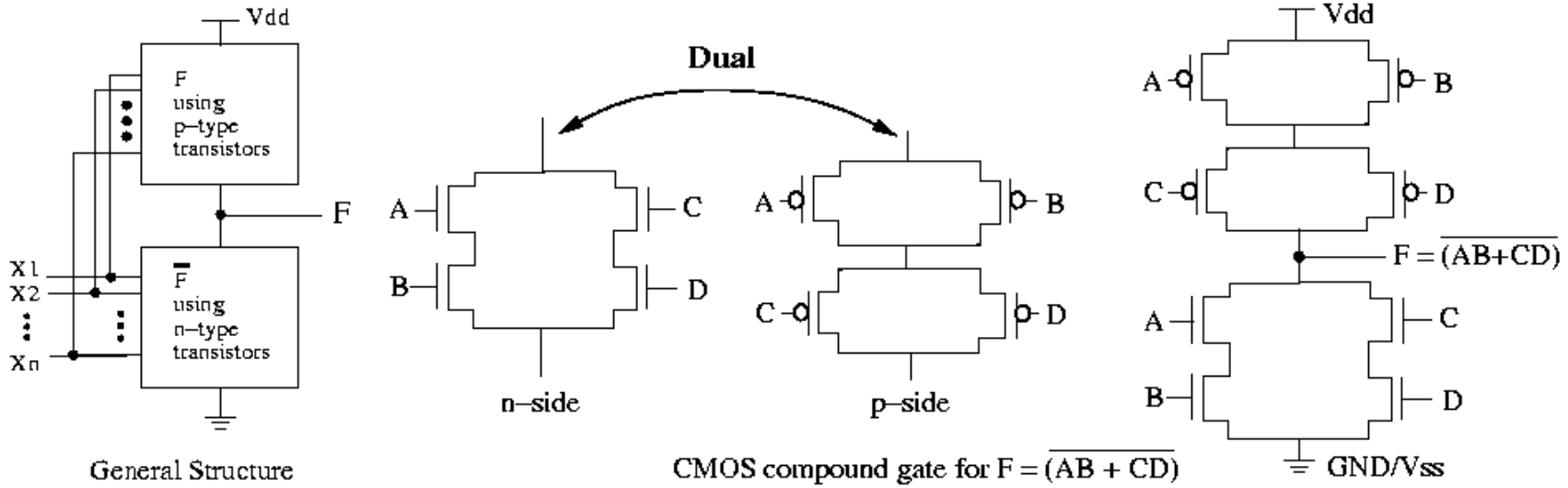


Basic CMOS Logic Library

Name	Distinctive shape	Algebraic equation	Cost (# of transistors)	Scaled gate delay (ps)
AND		$F=XY$	**6**	**24**
OR		$F=X+Y$	**6**	**24**
NOT (inverter/ repeater)		$F=\overline{X}$	**2**	**10**
Buffer (driver/ repeater)		$F=X$	**4**	**20**
NAND		$F=\overline{XY}$	**4**	**14**
NOR		$F=\overline{X+Y}$	**4**	**14**
Exclusive–OR (XOR)		$F=X\overline{Y}+\overline{X}Y$ $=X \oplus Y$	**14**	**42**

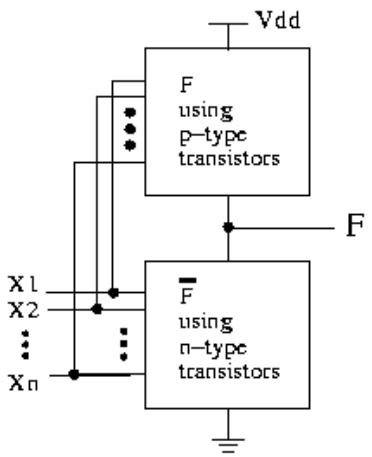
Construction of Compound Gates

- Example: $F = \overline{A \cdot B + C \cdot D}$.
- Step 1 (**n**-network): **Invert** *F* to derive *n*-network ($\overline{F} = A \cdot B + C \cdot D$)
- Step 2 (**n**-network): Make connections of transistors:
  - AND ⇔ Series connection
  - OR ⇔ Parallel connection

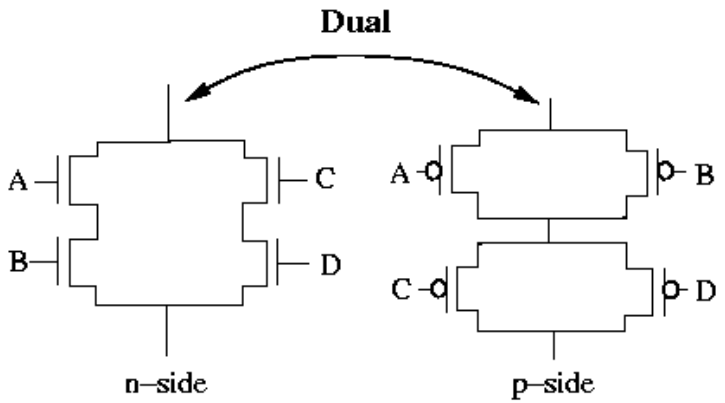


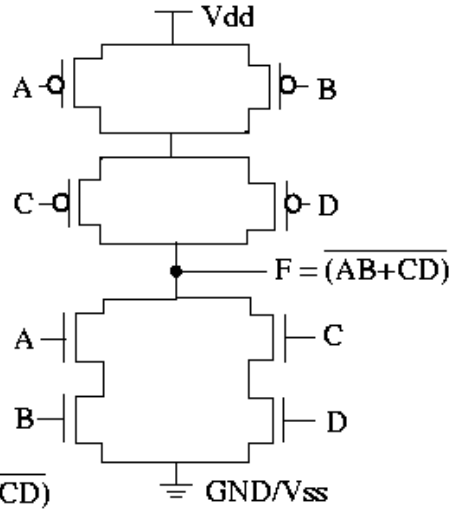
Construction of Compound Gates (cont'd)

- Step 3 (**p**-network): Expand *F* to derive *p*-network
  - $(F = \overline{AB + CD} = \overline{AB} \cdot \overline{CD} = (\overline{A} + \overline{B})' \cdot (\overline{C} + \overline{D}))'$
  - **each input is inverted**
- Step 4 (**p**-network): Make connections of transistors (same as Step 2).
- Step 5: Connect the *n*-network to GND (typically, 0V) and the *p*-network to VDD (5V, 3.3V, or 2.5V, etc).



General Structure





CMOS compound gate for $F = \overline{(AB + CD)}$

Homework Is Coming

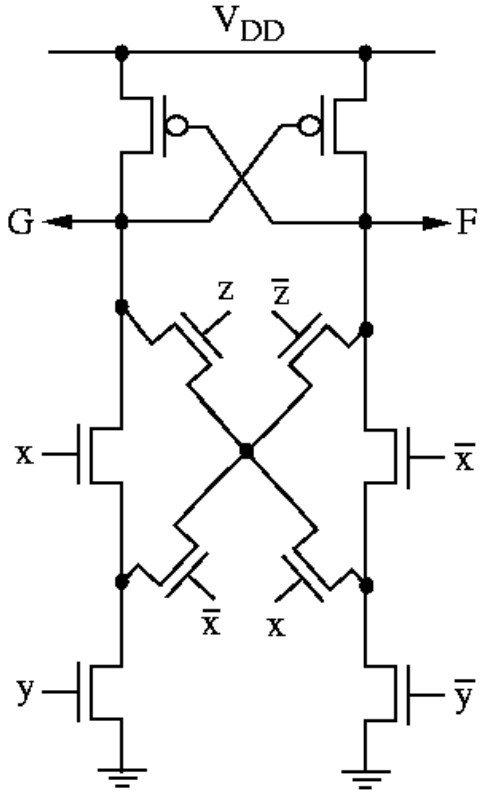
- It is related to using logic gates to realize a function

logic gate F=(A+B)(CD+E)

- Using gates for this logic function
- Using P&N transistors for the same logic

A Complex CMOS Gate

- The functions realized by the *n* and *p* networks must be complementary, and one of the networks must conduct for every input combination.
- Duality is not necessary.



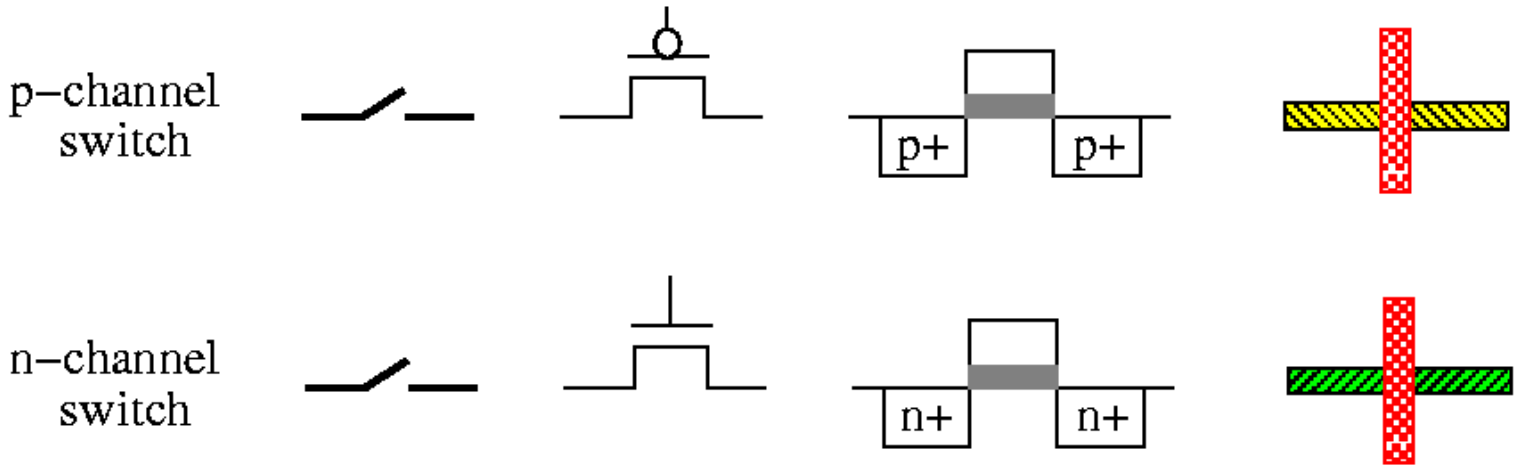
x	y	z	F	G
0	0	0	0	1
0	0	1	0	1
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

F = ?
G = ?

Please complete this truth table and two functions?

Stick Diagram

- Intermediate representation between the transistor level and the mask (layout) level.
- Gives topological information (identifies different layers and their relationship)
- Assumes that wires have no width.
- Possible to translate stick diagram automatically to layout with correct **design rules**.

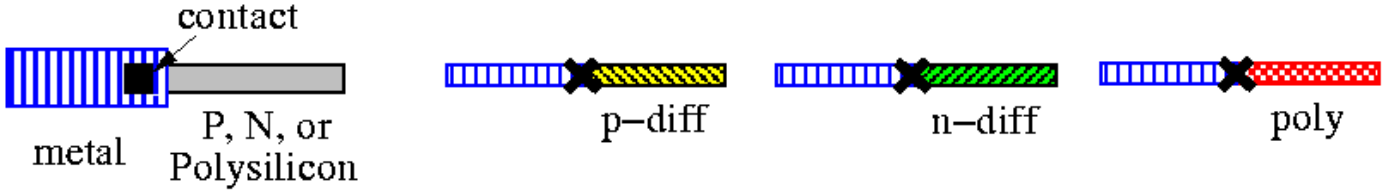


Stick Diagram (cont'd)

- When the same material (on the same layer) touch or cross, they are connected and belong to the same electrical node.

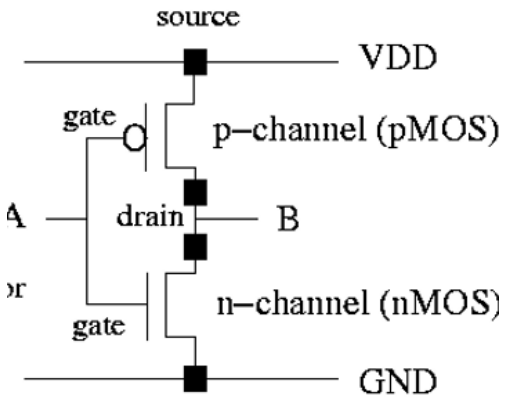
- When **polysilicon** crosses N or P **diffusion**, an N or P transistor is formed.
  - Polysilicon is drawn on top of diffusion.
  - Diffusion must be drawn connecting the source and the drain.
  - Gate is automatically self-aligned during fabrication.

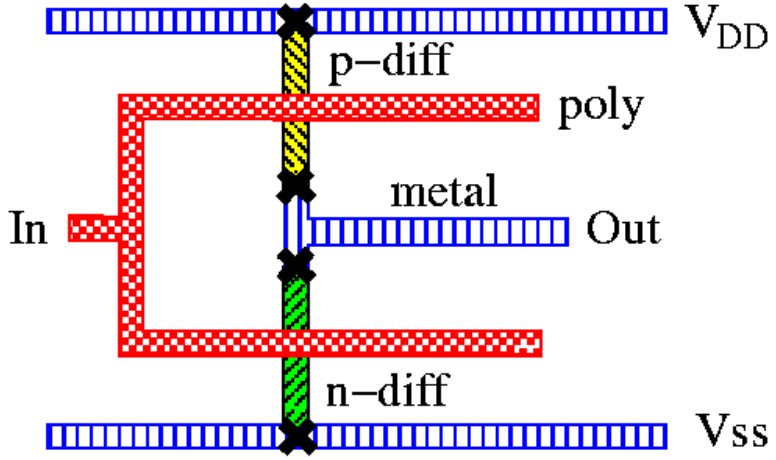
- When a metal line needs to be connected to one of the other three conductors, a **contact** cut (**via**) is required.

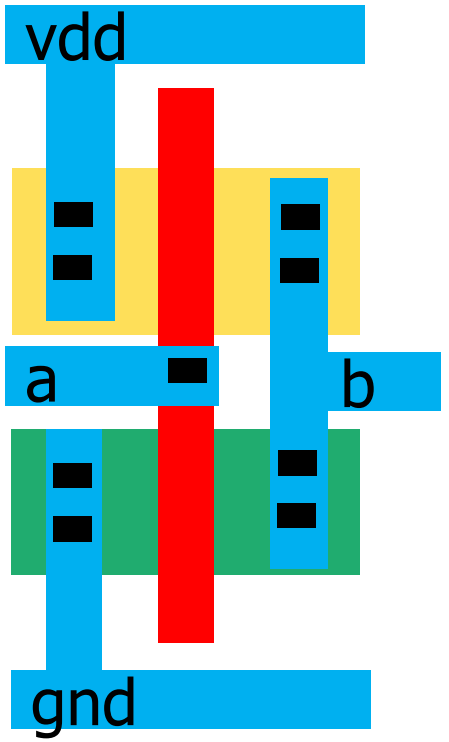


CMOS Inverter Stick Diagrams

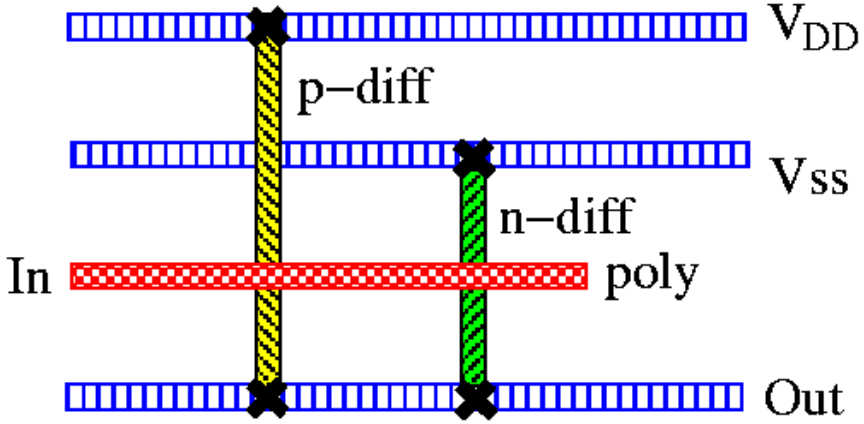
- Basic layout

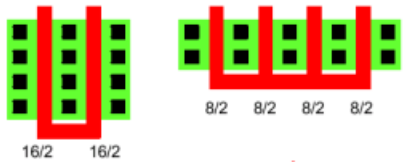






- More area efficient layout





From Planar To FinFet

W_{eff} = (2 * Fin height + Fin width) * N
N = # of fins
Width can become larger, but quantized

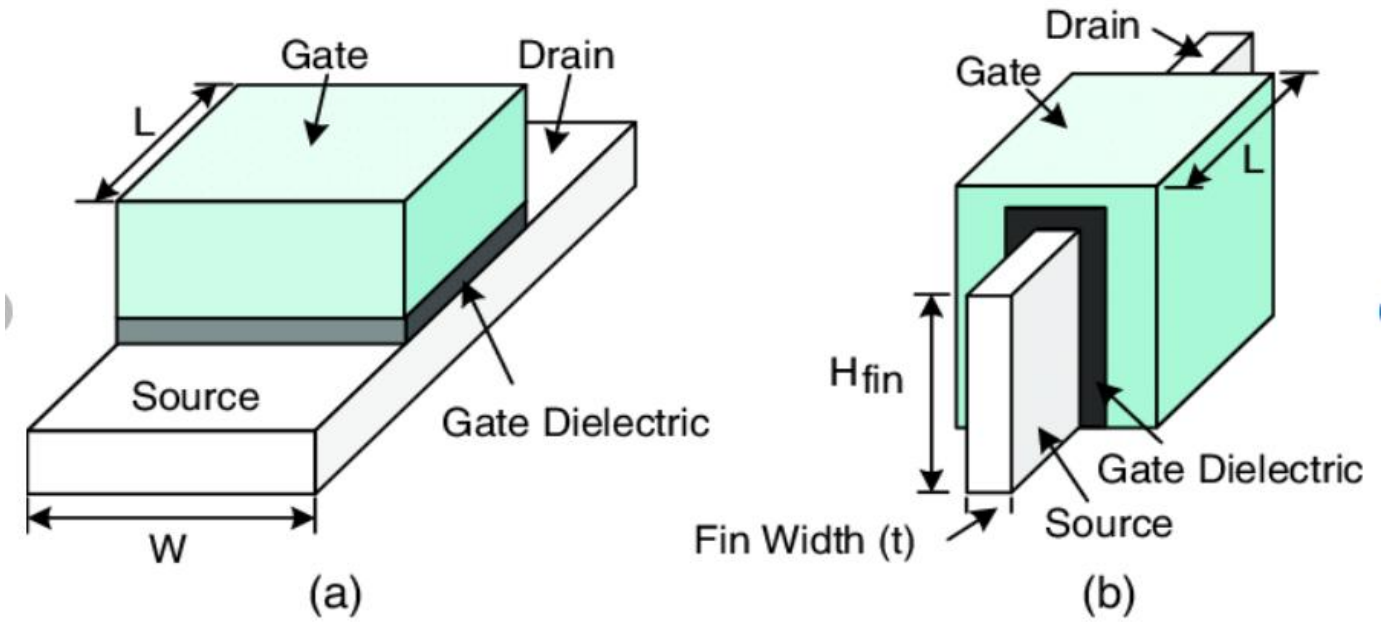


Illustration of structural differences (no substrate): (a) planar device; (b) FinFET device.

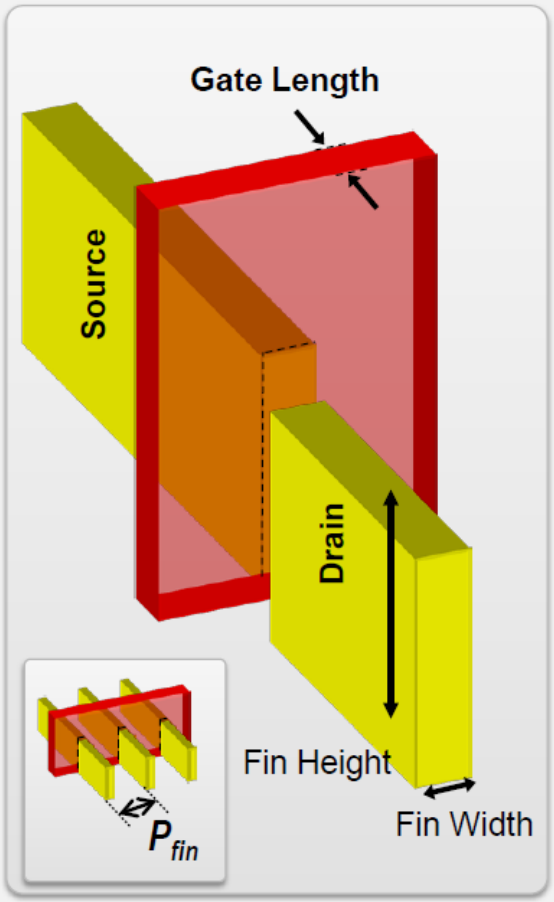
https://ieeexplore.ieee.org/abstract/document/6649058
FinFETs - Technology and circuit design challenges

FinFET Advantages & Considerations

Improved Characteristics; Impacts Circuit Design

Advantages

- Lower leakage
- Less variability
- Lower voltage



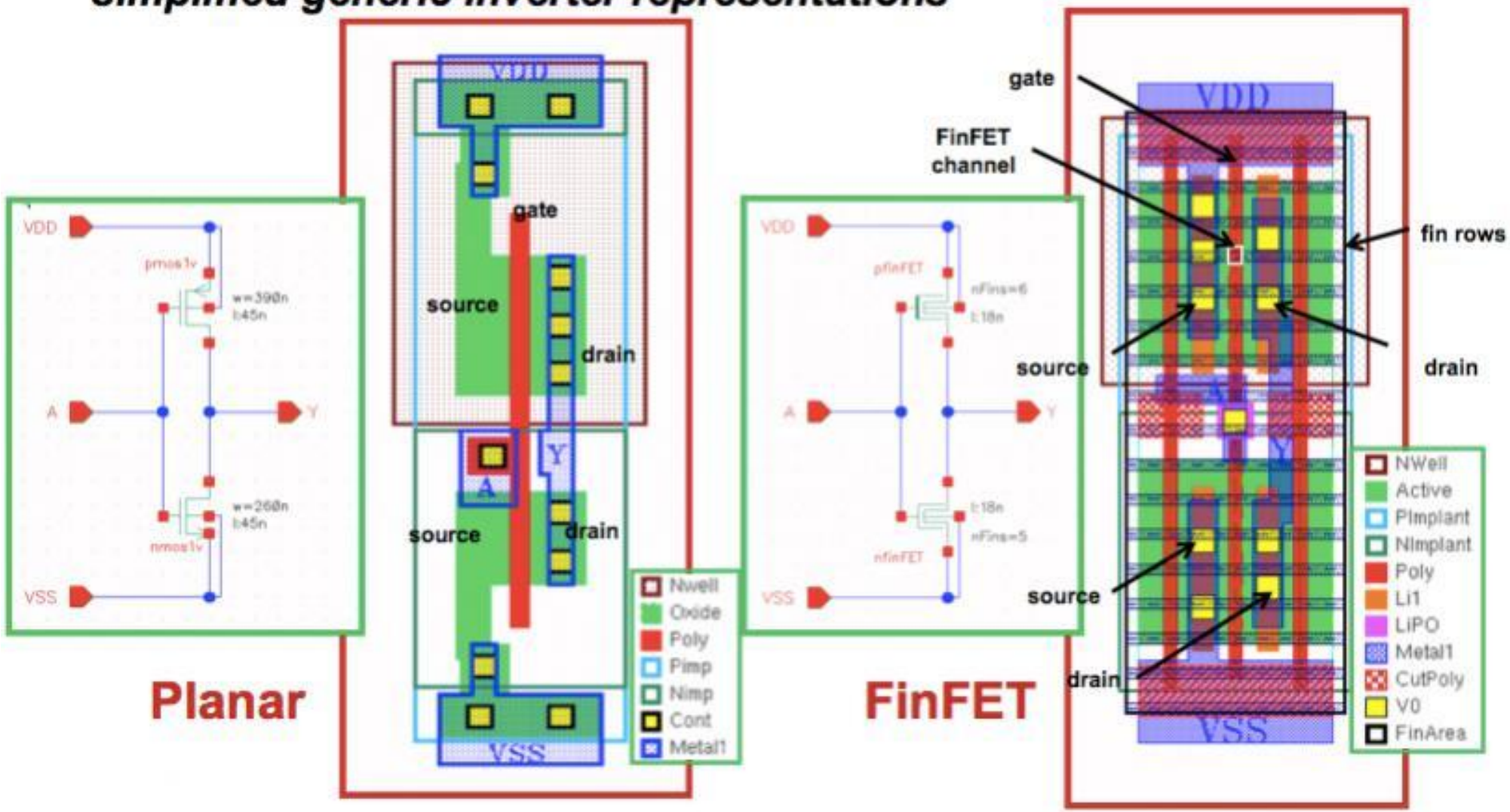
Considerations

- Quantized widths
- No body biasing
- Higher parasitics
- Aging

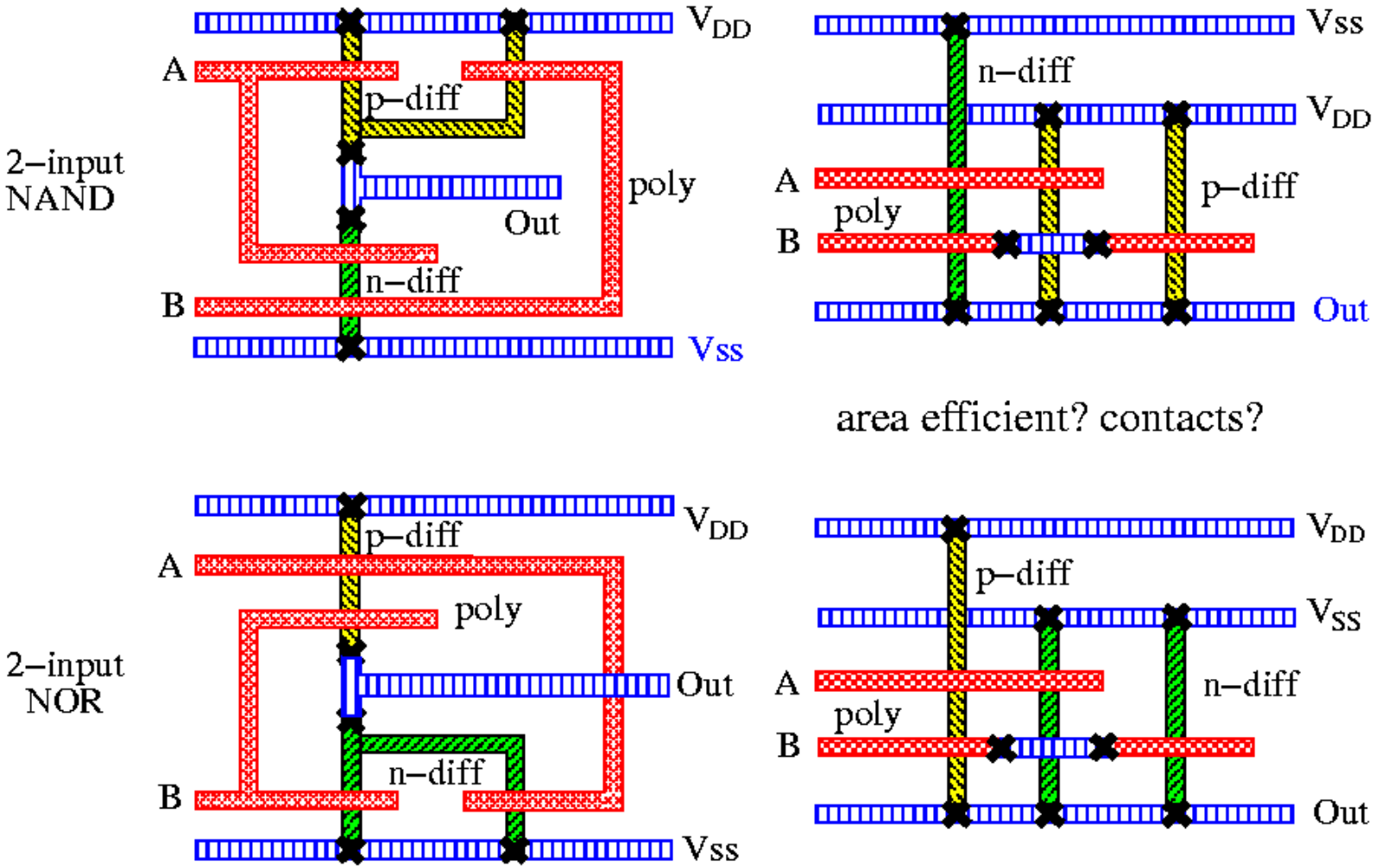
SYNOPSYS

Planar to FinFET Layout Differences

simplified generic inverter representations

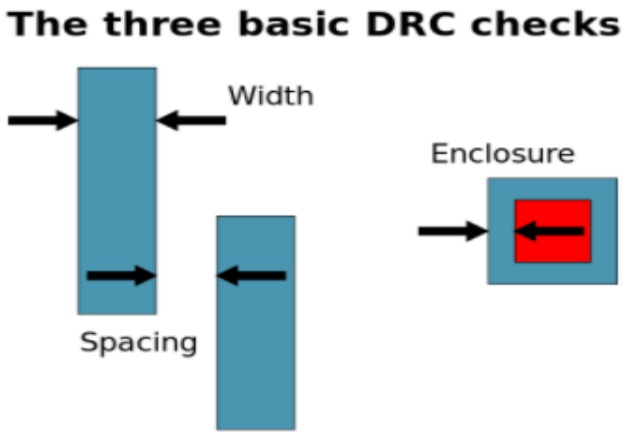


CMOS NAND/NOR Stick Diagrams



Design Rules (1)

- Layout rules are used for preparing the masks for fabrication.
- Fabrication processes have inherent limitations in accuracy.
- Design rules specify geometry of masks to optimize yield and reliability (trade-offs: area, yield, reliability).
- Three major rules:
  - **Wire width:** Minimum dimension associated with a given feature.
  - **Wire separation:** Allowable separation.
  - **Contact:** overlap rules.



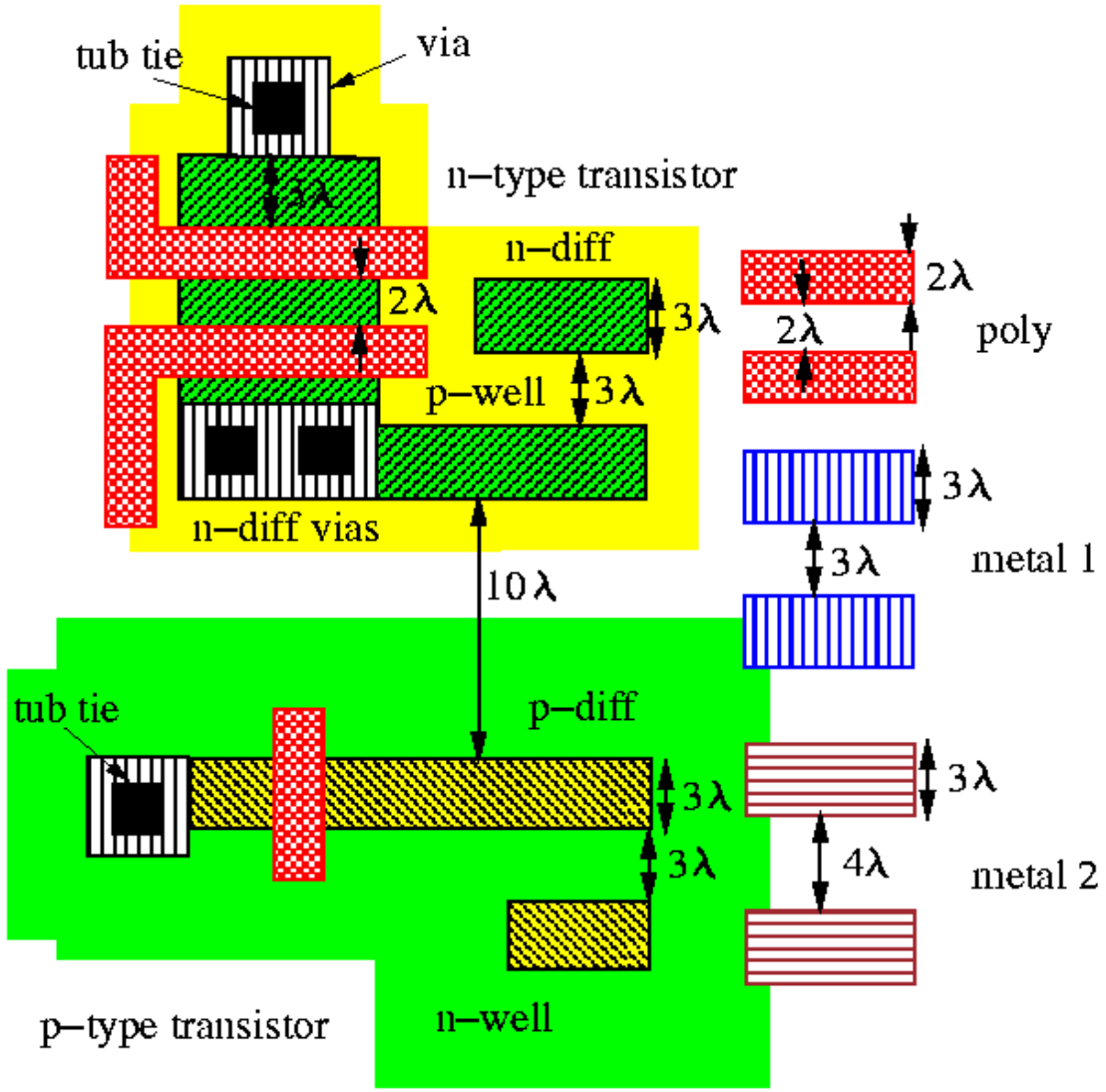
Also need to handle non-Manhattan shapes

Design Rules (2)

- Two major approaches:
  - **"Micron" rules:** stated at micron resolution.
  - **λ rules:** simplified micron rules with limited **scaling** attributes.
- λ may be viewed as the size of minimum feature.
  - Can be viewed as "parameters"

- Design rules represents a tolerance which insures very high probability of correct fabrication (not a hard boundary between correct and incorrect fabrication).
- Design rules are determined by experience/experiments
  - Foundries go through many test patterns … → test chips

SCMOS (Scalable CMOS) Design Rules



There was a layout editor based on lambda rules. When the real value of lambda is given, a new layout is created

MOSIS Layout Design Rules

- MOSIS design rules (SCMOS rules) are available at http://www.mosis.org.
- 3 basic design rules: Wire width, wire separation, contact rule.
- MOSIS design rule examples

R1	Min active area width	3 λ
R3	Min poly width	2 λ
R4	Min poly spacing	2 λ
R5	Min gate extension of poly over active	2 λ
R8	Min metal width	3 λ
R9	Min metal spacing	3 λ
R10	Poly contact size	2 λ
R11	Min poly contact spacing	2 λ

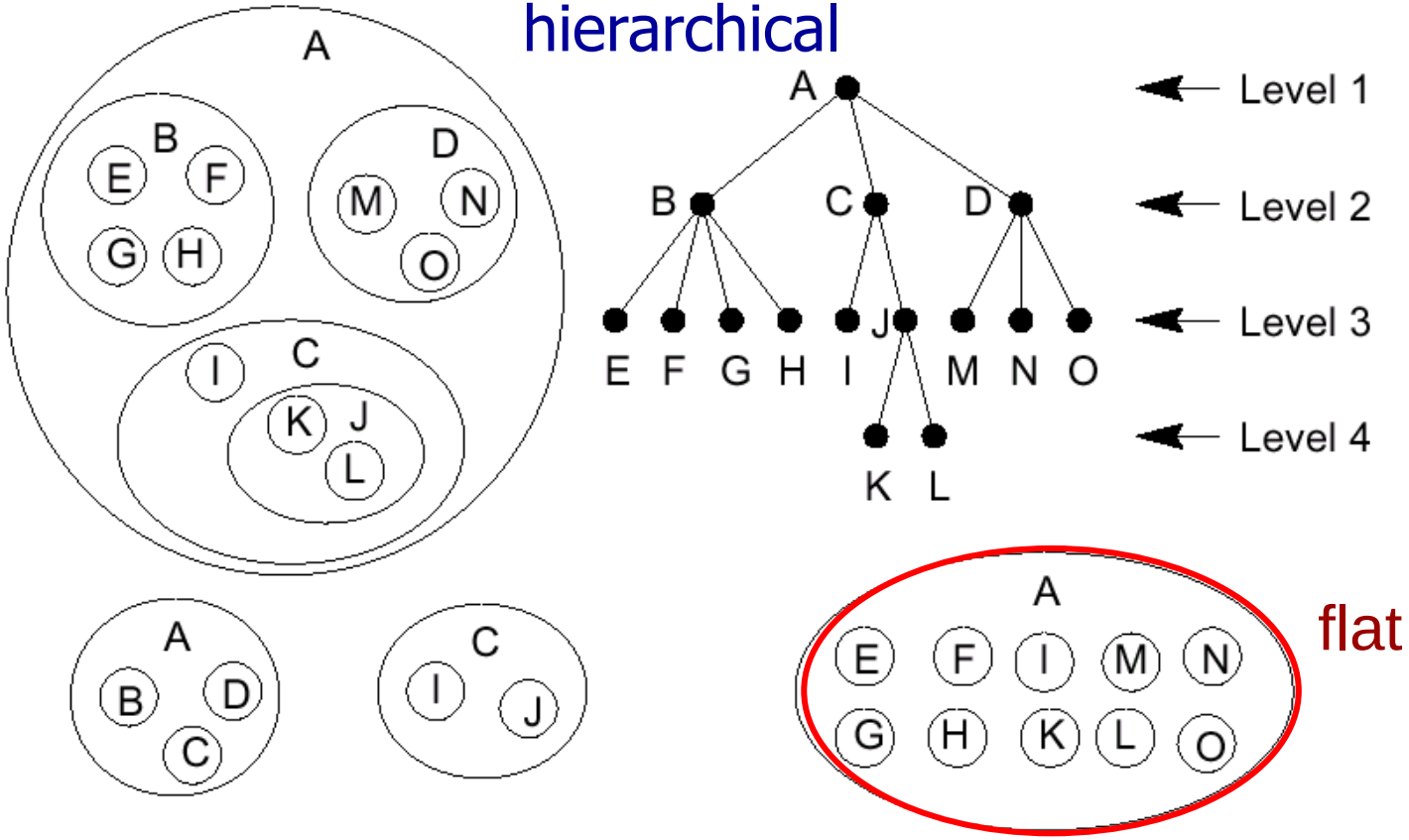
Summary So Far

- Digital design flow - steps
- Boolean functions
- Basic CMOS transistor circuits and their logic gates

How To Address Complexity

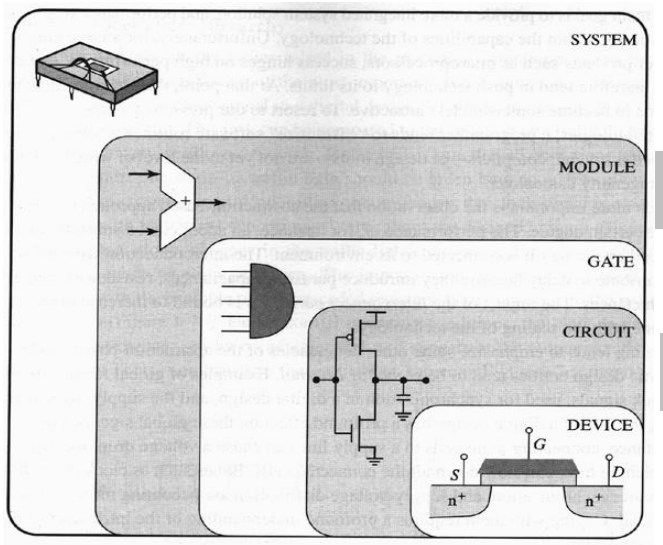
Cope with Complexity - Hierarchical Design

- *Hierarchy:* something is composed of simpler things.
- Design cannot be done in one step ⇒ partition the design hierarchically.



Cope with Complexity – Abstraction & Modeling

- *Abstraction:* when looking at a certain level, you don't need to know all details of the lower levels.



- Design domains:
  - Behavioral: black box view
  - Structural: interconnection of subblocks
  - Physical: layout properties
- Each design domain has its own hierarchy.

Ordering Of The Major Topics

- Digital design flow
- CMOS logic gates and Boolean equations
- Algorithms and complexity
- Compaction
- Partitioning
- Floorplanning
- Placement
- Basic logic synthesis, technology mapping, (High level synthesis)
- Routing