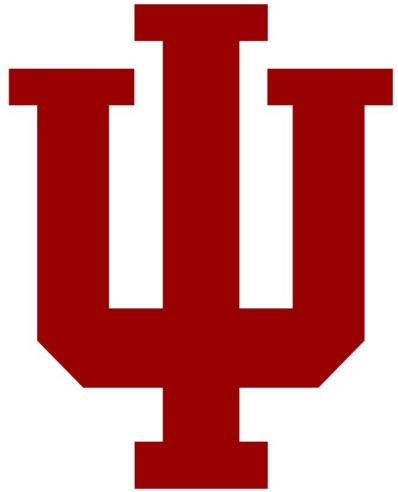
05 Hardware Design I (VLSI)

Engr 399/599: Hardware Security

Andrew Lukefahr
Indiana University



Adapted from: Mark Tehranipoor of University of Florida

Course Website

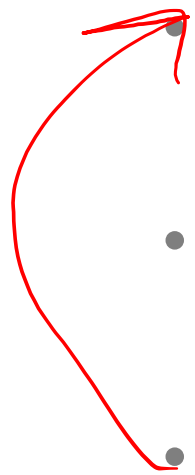
engr599.github.io

Room 2-3-5

Passcode:

Write that down!

Last Time



- Symmetric Key → Same
- Asymmetric Key → Different
- DES

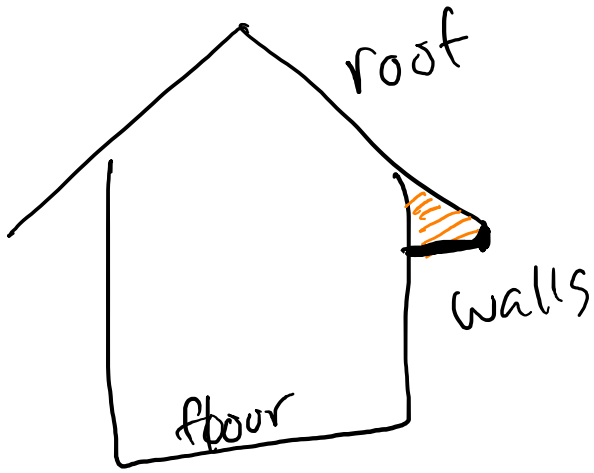
lighter-weight
faster
easier

needs a key

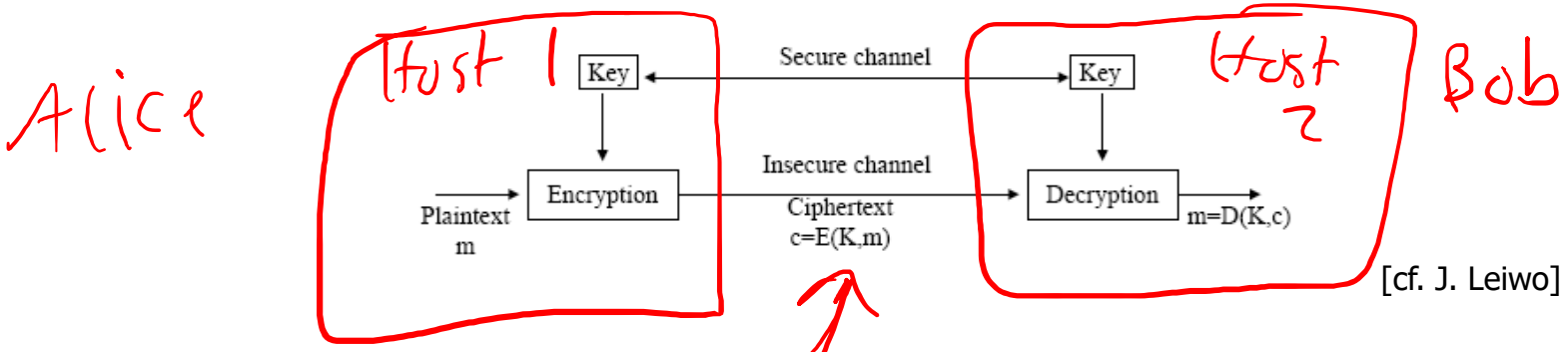
comp. intensive

used to exchange keys

Symmetric and Asymmetric Cryptosystems (1)



- Symmetric encryption = secret key encryption
  - $K_E = K_D$ — called a secret key or a private key
  - Only sender S and receiver R know the key



[cf. J. Leiwo]

  - As long as the key remains secret, it also provides authentication (= proof of sender's identity)

Project 1: Hardware Trojan

Hardware on Monday

- Goal: "Corrupt" a working DES implementation
- We give you DES in HW
- You need to:
  - Deploy DES
  - Corrupt DES

Group Assignments

- Chris Sozio
- Will Fleming
- Clare Barnes

Group #1

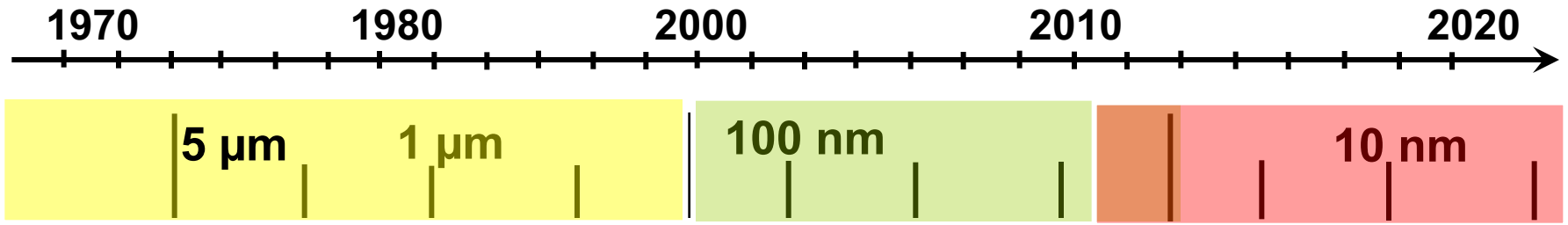
- Austin Parkes
- Max Harms
- Michael Foster

#2

- Trey Peterson
- Yifan Zhang
- Jack Ruocco

#3

Nanoelectronics

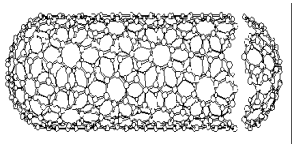


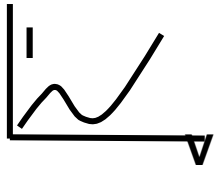
Silicon Micro Electronics

Silicon Nano Electronics

Non-Silicon technology

- New Devices Emerge
  - SOI, FinFET, TriGate

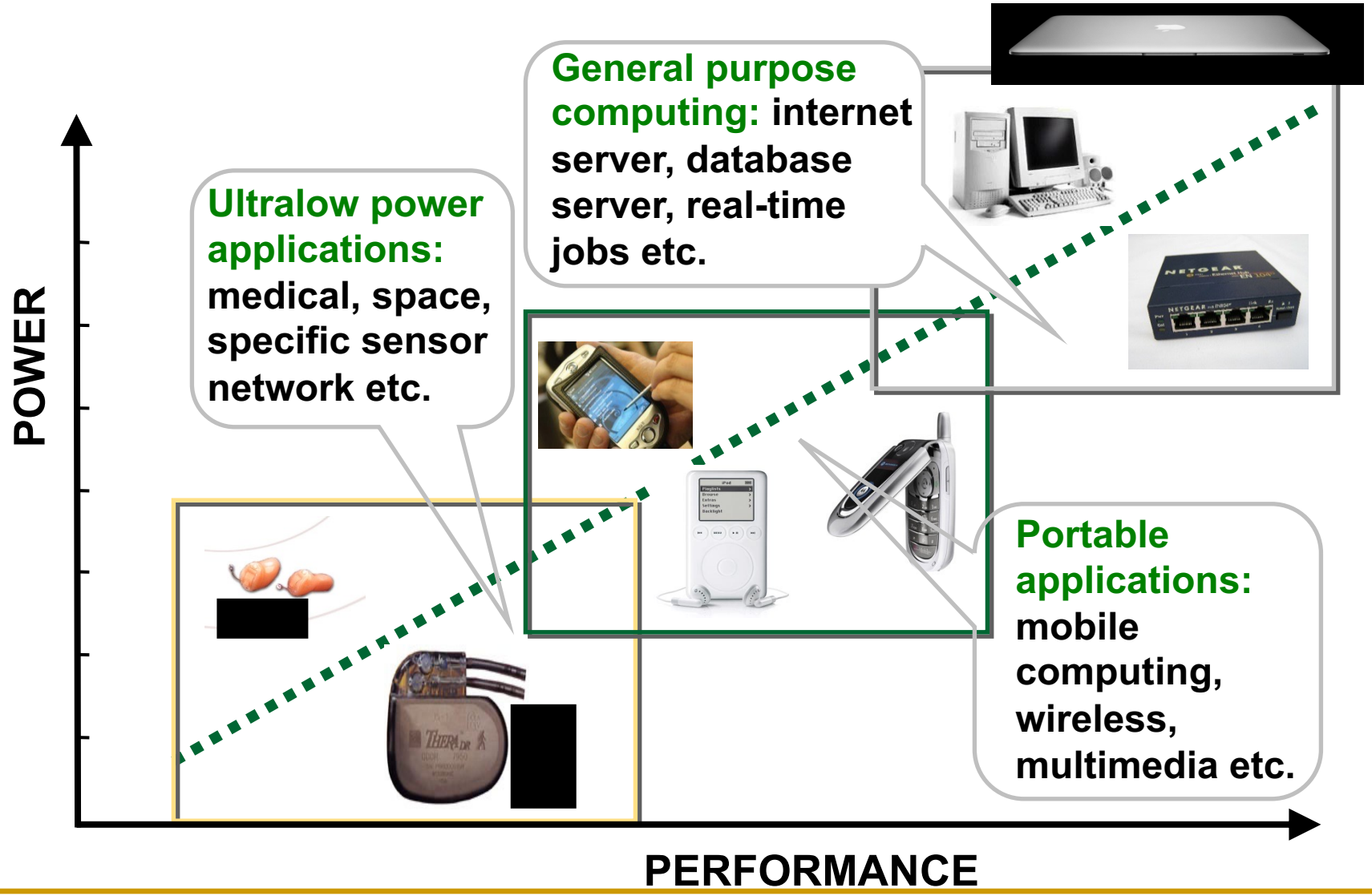




- Power will continue to increase
  - Low-leakage / leakage-tolerance
  - Dynamic power management
- Process variation will be a key issue
  - Design for Manufac. (DFM), Reliability
- Arch. / design methods for new devices

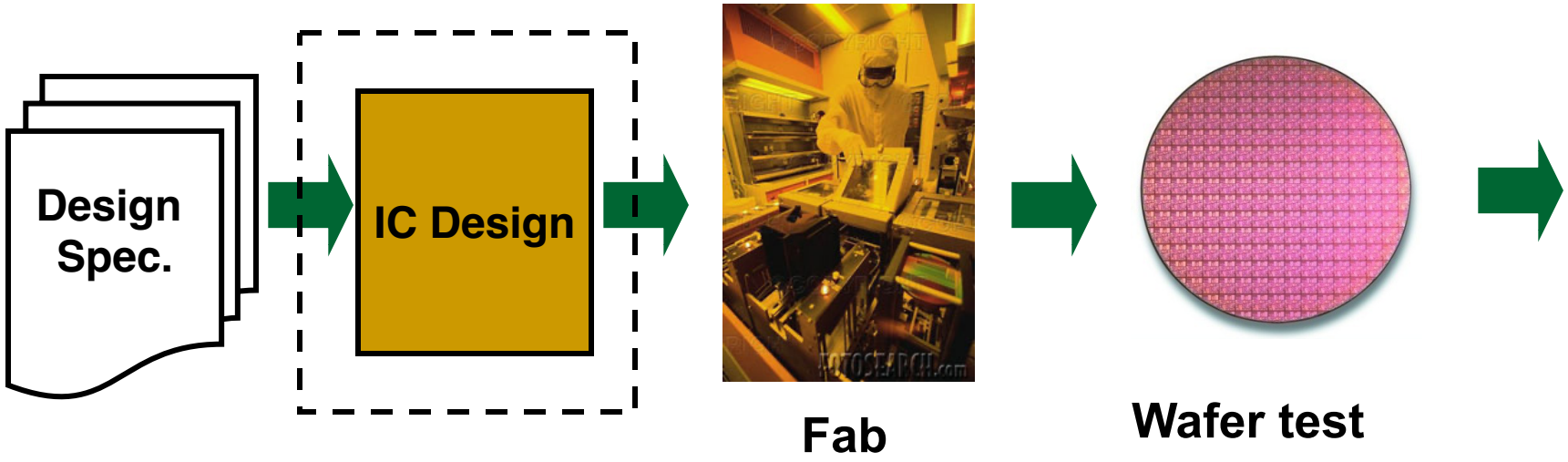
- Carbon Nanotubes
- Quantum computing
- Molecular transistors etc.

VLSI Applications

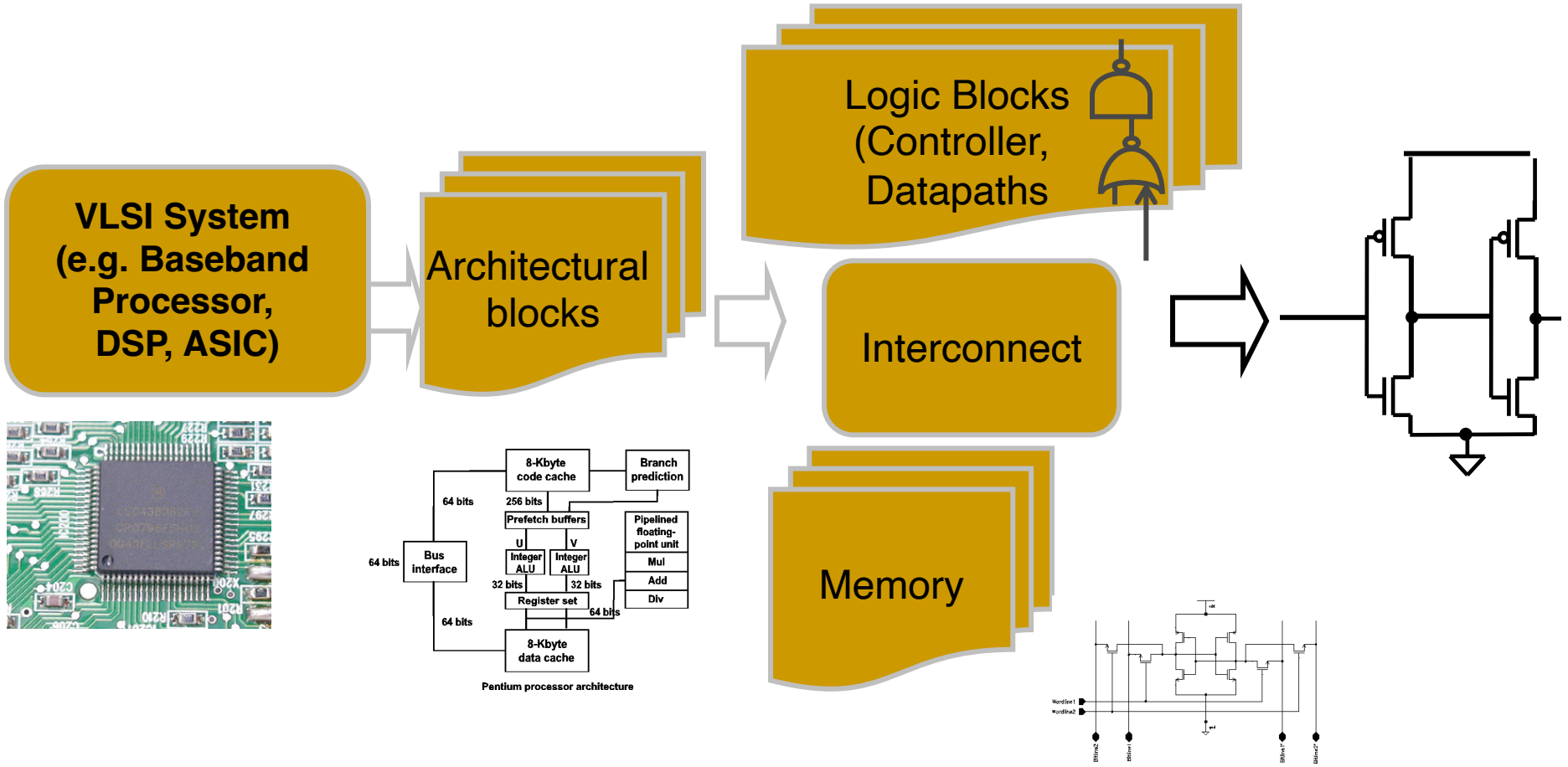


- Different applications have different power-performance demands

IC Design and Test Flow

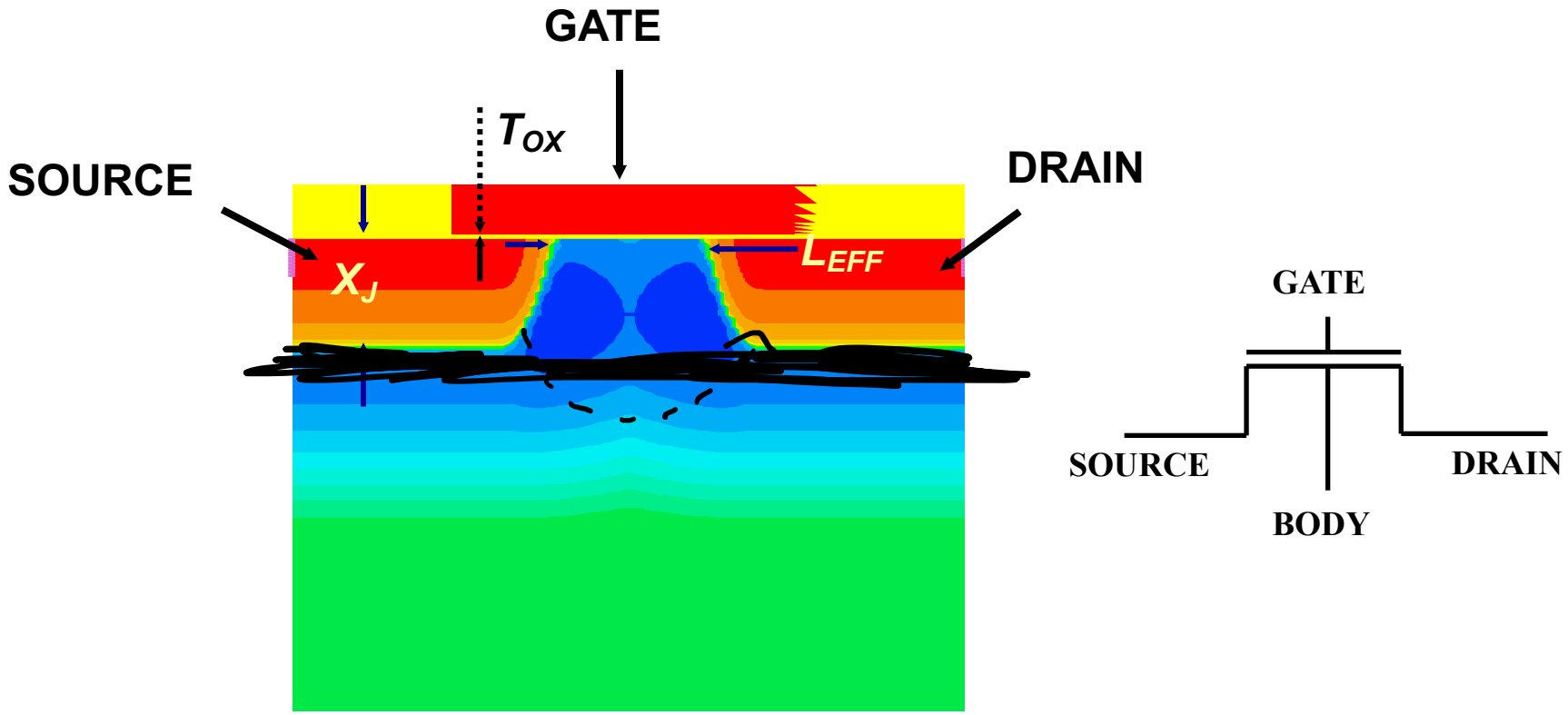


Nanoscale VLSI System



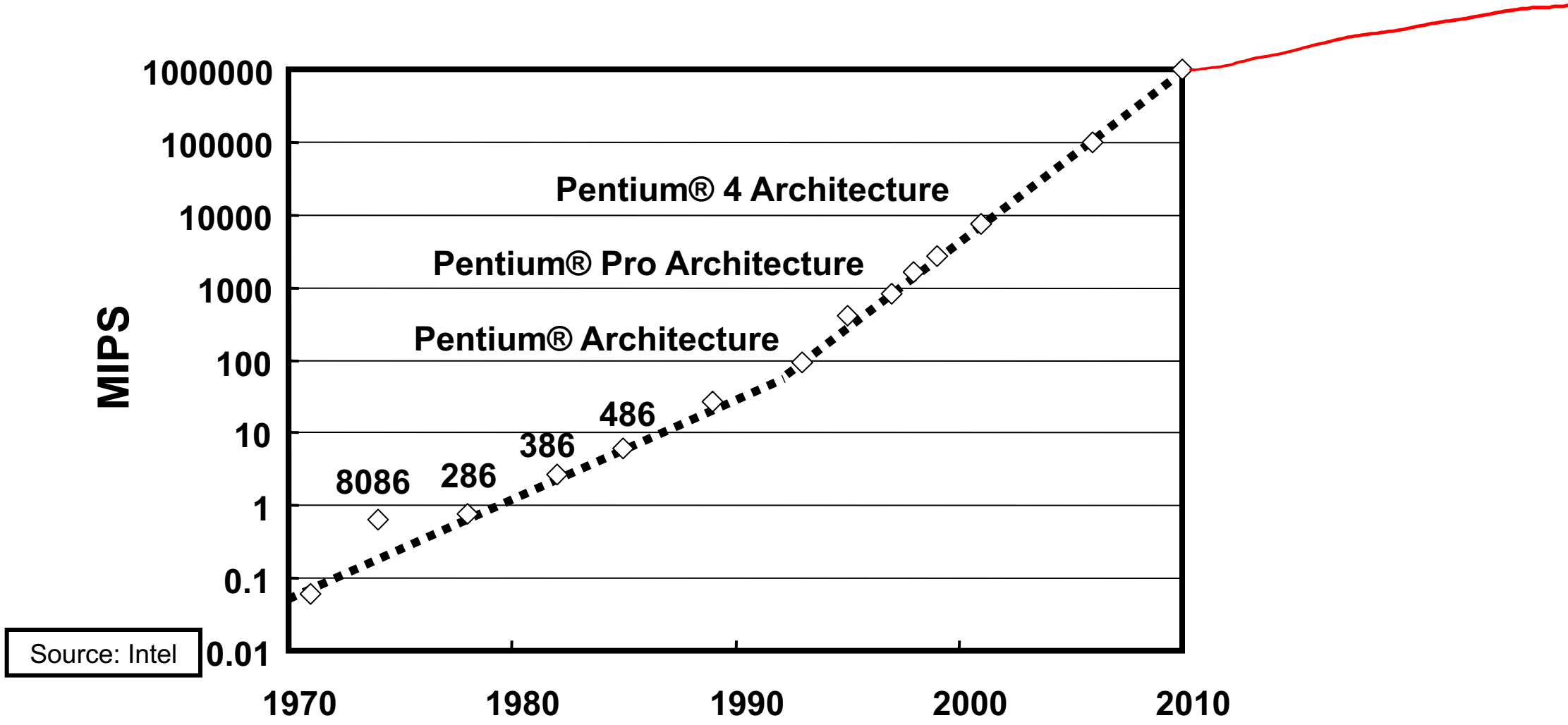
System -> Architecture -> Logic ->Transistor

Nanoscale Transistor



- Basic building block for the integrated circuit: both logic and memory
- Minimum feature size is now well below 100nm

Exponential Growth in Computing Power

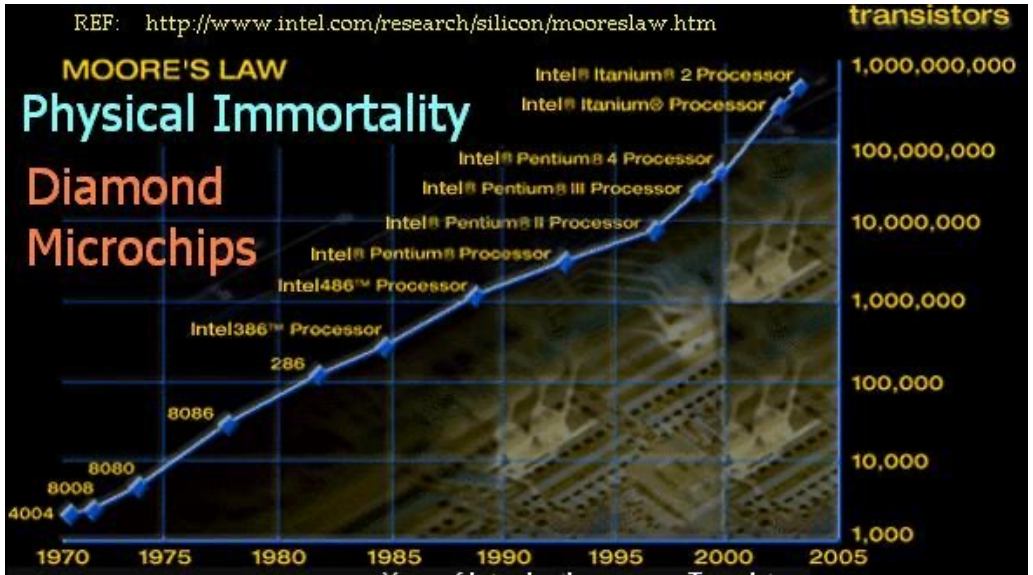


What is the key to the growth in computing power?

Moore's Law

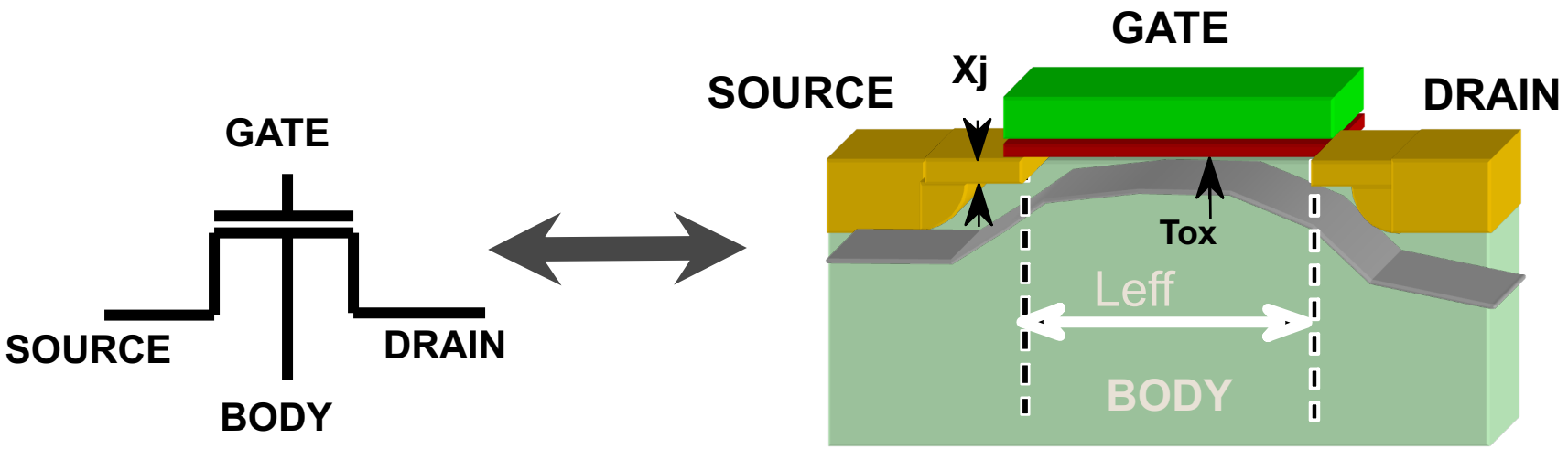
"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. …"

***Electronics Magazine,* 19 April 1965**

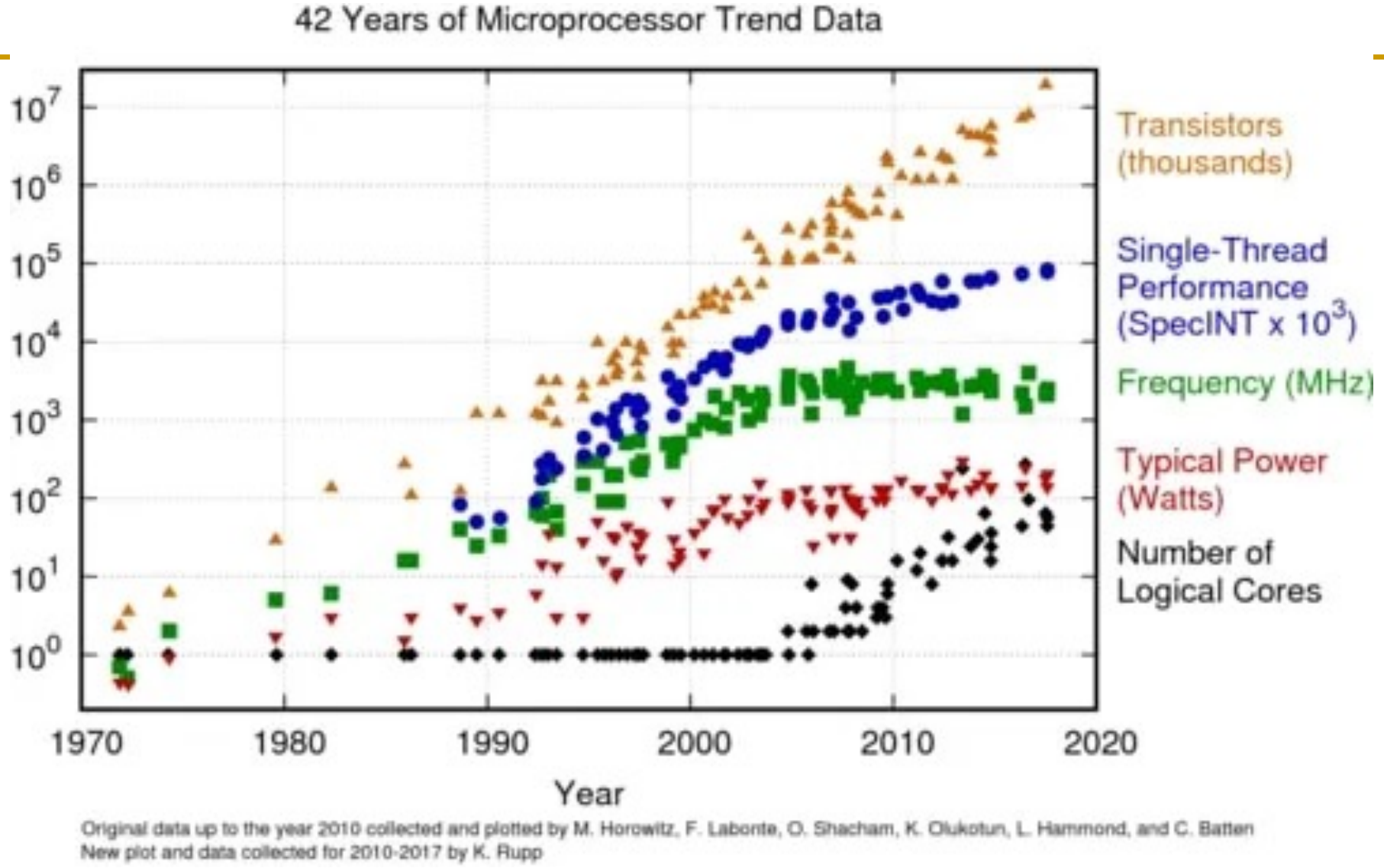


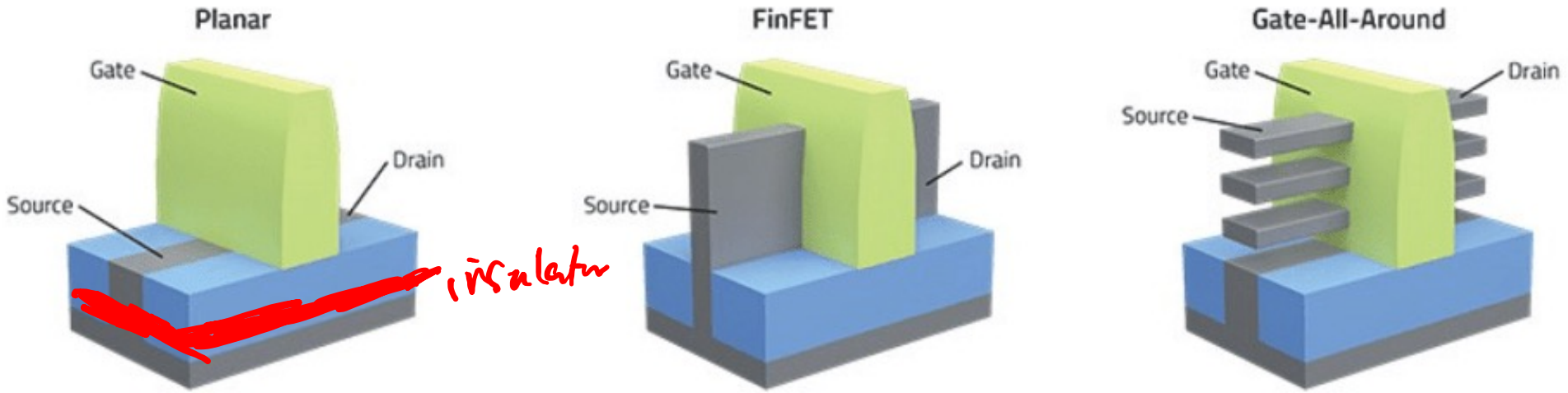
	Year of Introduction	Transistors
4004	1971	2,250
8008	1972	2,500
8080	1974	5,000
8086	1978	29,000
286	1982	120,000
Intel386™ processor	1985	275,000
Intel486™ processor	1989	1,180,000
Intel® Pentium® processor	1993	3,100,000
Intel® Pentium® II processor	1997	7,500,000
Intel® Pentium® III processor	1999	24,000,000
Intel® Pentium® 4 processor	2000	42,000,000
Intel® Itanium® processor	2002	220,000,000
Intel® Itanium® 2 processor	2003	410,000,000

Technology Scaling



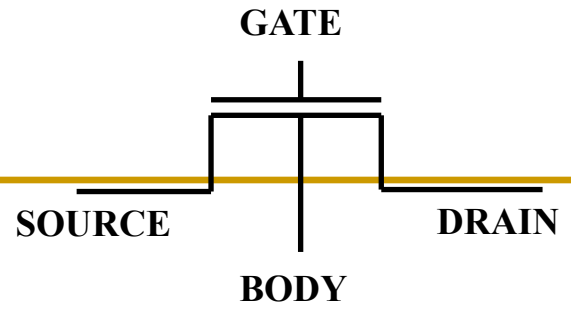
Dimensions scale down by 30%	Doubles transistor density
Oxide thickness scales down	Faster transistor, higher performance
Vdd & Vt scaling	Lower active power


42 Years of Microprocessor Trend Data
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1970
1980
1990
2000
2010
2020
Year
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp



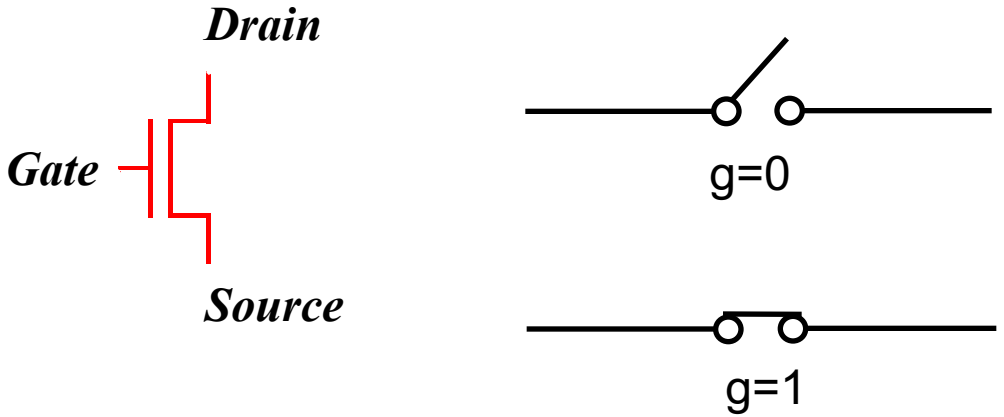
insulator

SOI
silicon on insulators



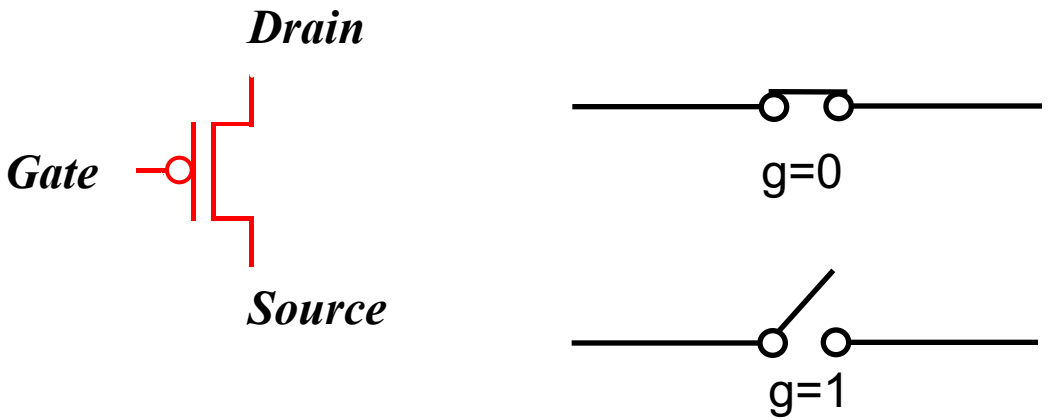
nMOS Transistor

- If the gate is “high”, the switch is on
- If the gate is “low”, the switch is off

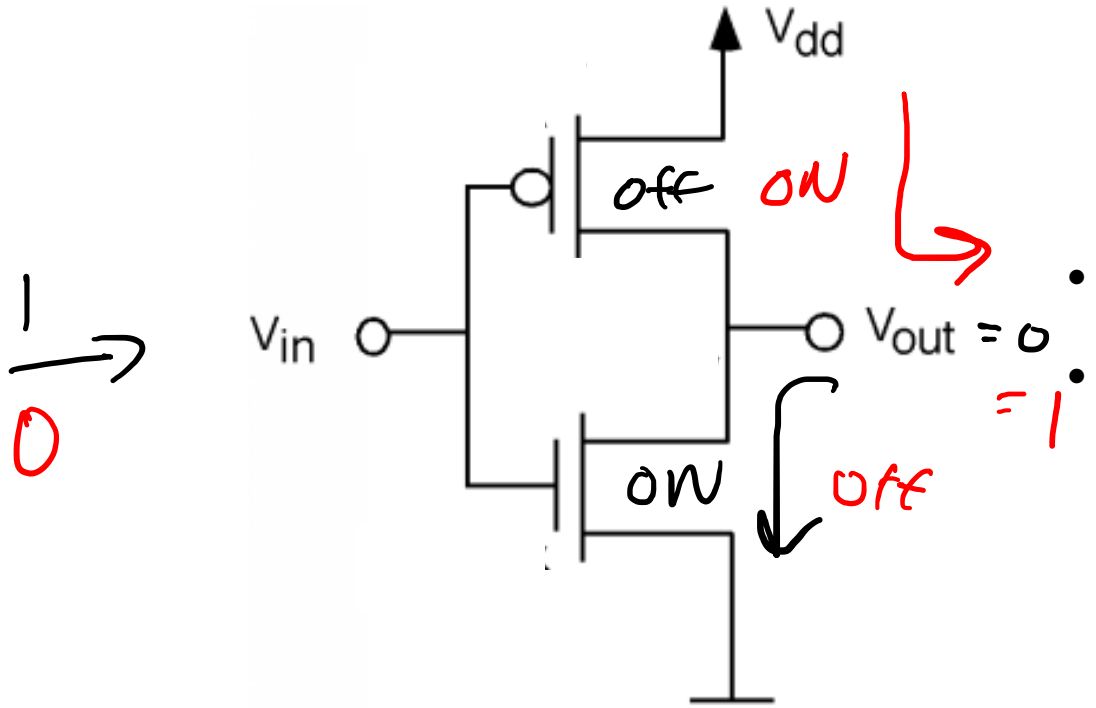


pMOS Transistor

- If the gate is “low”, the switch is on
- If the gate is “high”, the switch is off

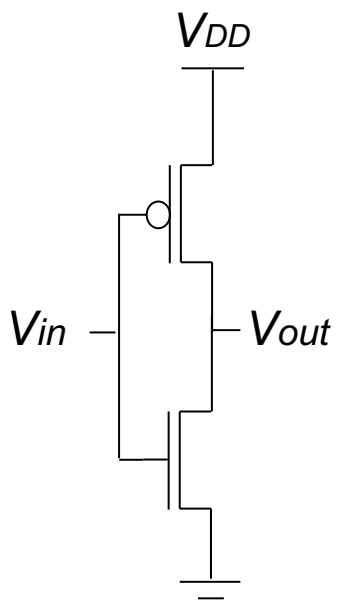


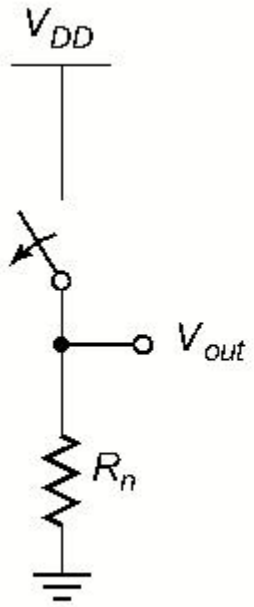
Solution: CMOS



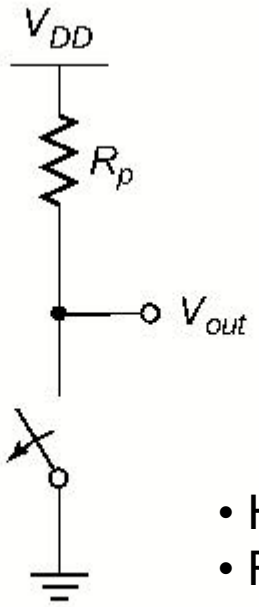
- No static current flow
- Less current means less power

CMOS Inverter First-Order DC Analysis





$V_{in} = V_{DD}$



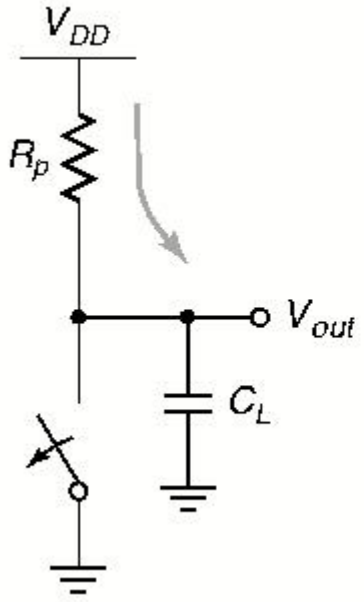
$V_{in} = 0$

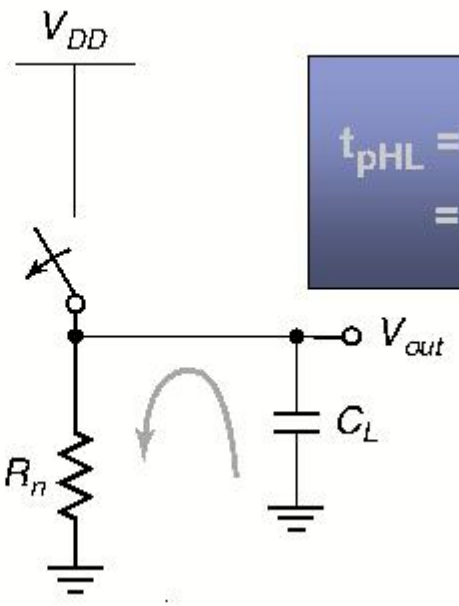
$V_{OL} = 0$

$V_{OH} = V_{DD}$

- High noise margin
- Ratioless
- low output impedance
- extremely high input impedance
- no static power

CMOS Inverter: Transient Response





$t_{pHL} = f(R_{on}.C_L)$
$= 0.69\ R_{on}C_L$

To reduce delay:
- Reduce CL
- Reduce Rp,n
- Increase W/L ratio

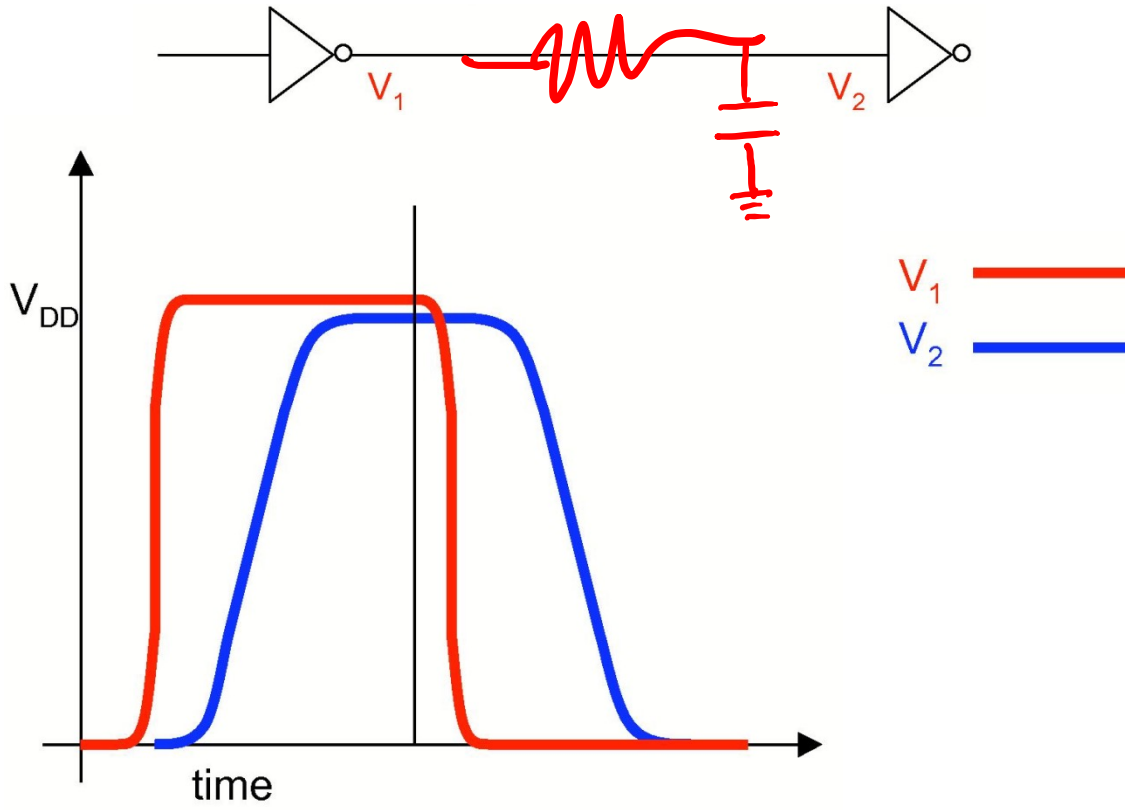
$V_{in} = V_{DD} \rightarrow 0$ $V_{in} = 0 \rightarrow V_{DD}$

Output: Low-to-High High-to-Low

CL is composed of the drain diffusion capacitances of the NMOS and PMOS transistors, the capacitance of connecting wires, and the input capacitance of the fan-out gates

Performance Characterization

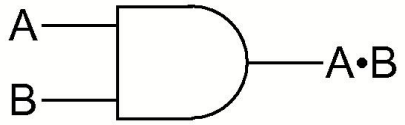
- Interconnect delay



Boolean Algebra

- Basic operators
  - AND

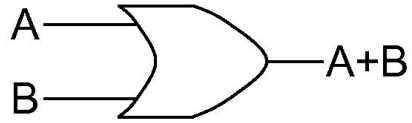
$$f(A,B) = A \cdot B = A \cap B$$



A	B	A•B
0	0	0
0	1	0
1	0	0
1	1	1

  - OR

$$f(A,B) = A + B = A \cup B$$

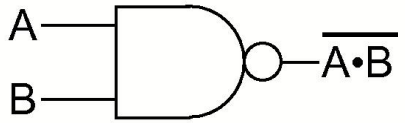


A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Algebra

- Basic operators
  - NAND

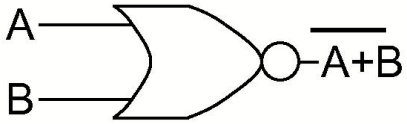
$$f(A,B) = \overline{A \cdot B} = \overline{A \cap B}$$



A	B	$\overline{A \bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0

  - NOR

$$f(A,B) = \overline{A + B} = \overline{A \cup B}$$

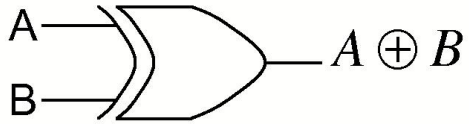


A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Boolean Algebra

- Basic operators
  - XOR

$$f(A,B) = A \oplus B$$



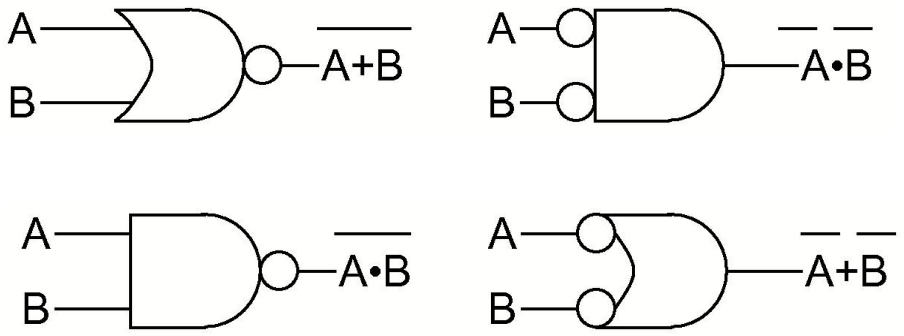
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Boolean Algebra

- DeMorgan's Theorem

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$



Boolean Algebra

- Truth Tables

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

- Representation of the function to be realized
- Sum of Products representation
  - Sum of minterms

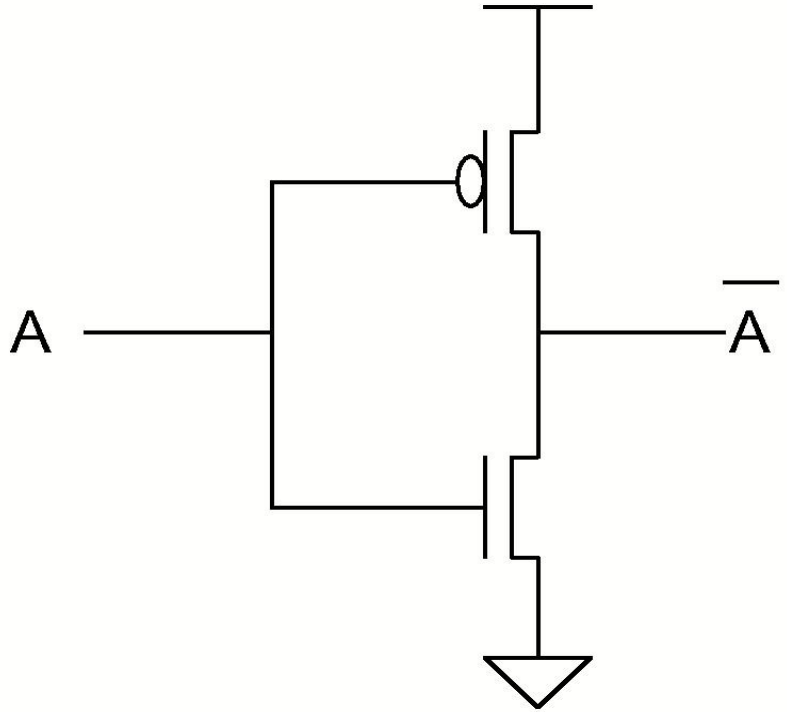
$$F = \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C}$$

- Product of Sums representation
  - Product of maxterms

$$F = (A + B + C) \cdot (A + \overline{B} + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C})$$

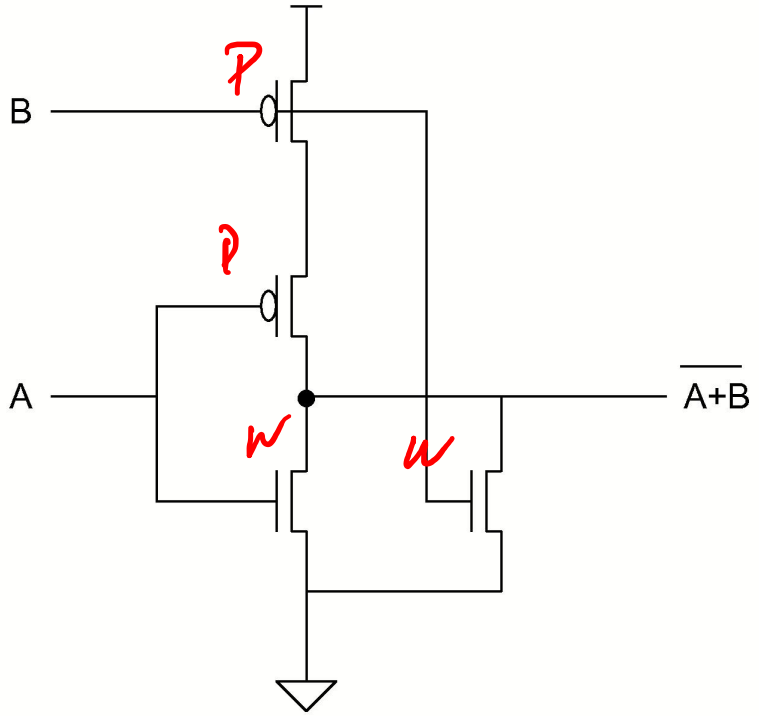
CMOS Logic Implementations

- Inverter



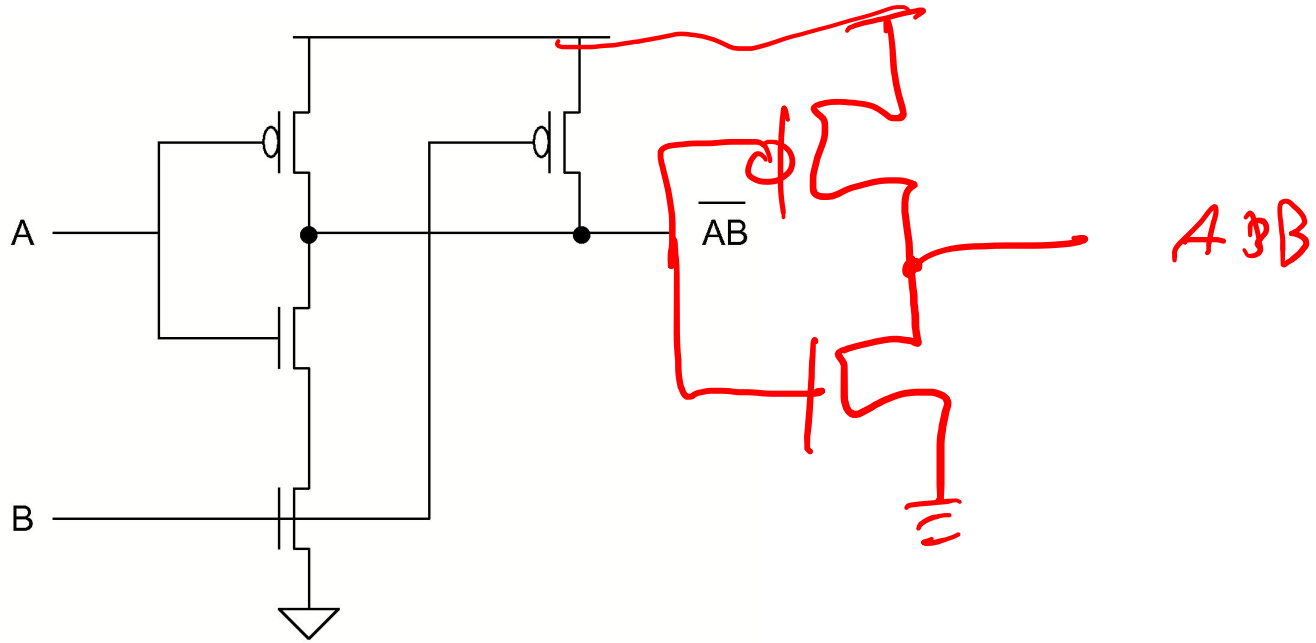
CMOS Logic Implementations

- NOR



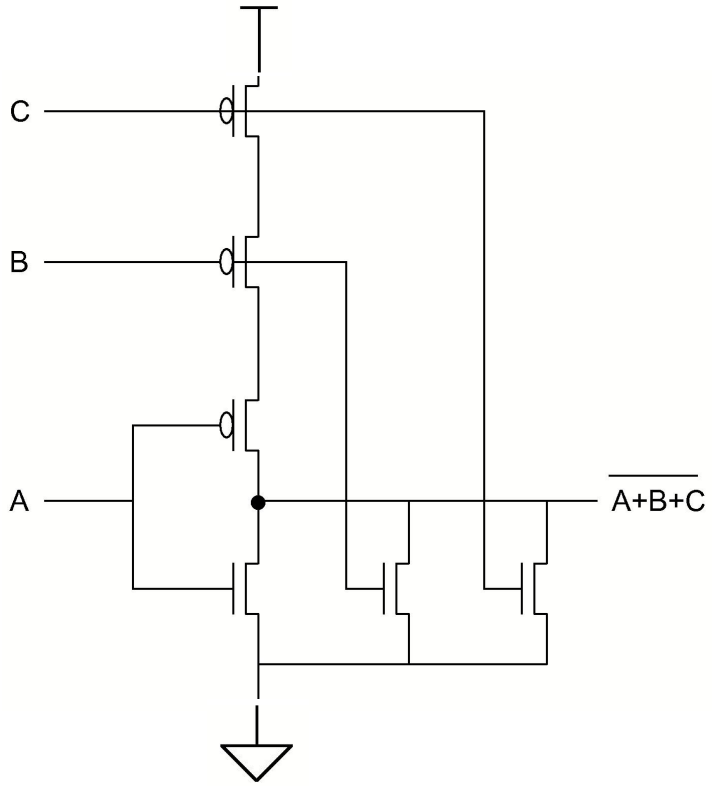
CMOS Logic Implementations

- NAND



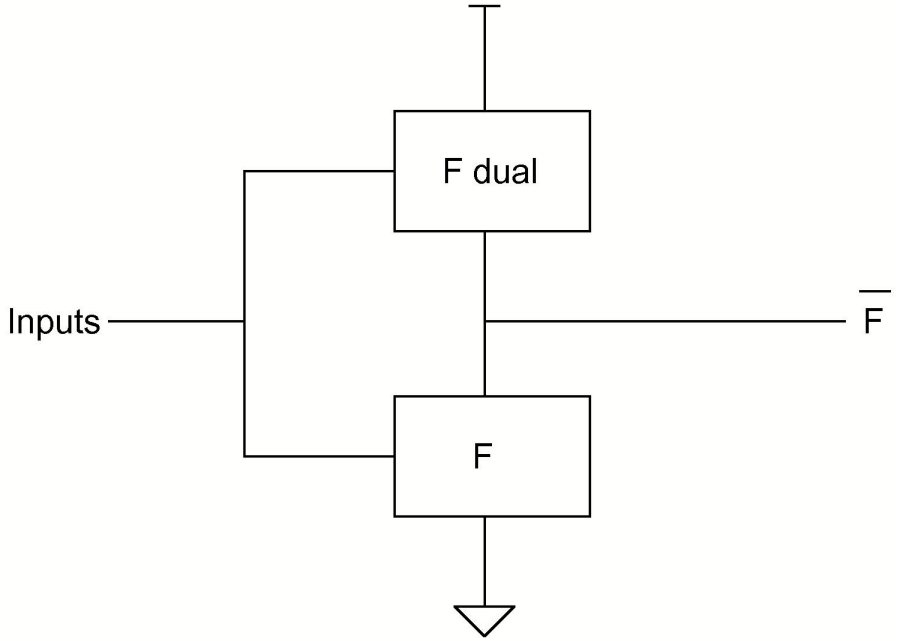
CMOS Logic Implementations

- Multi-input NOR



CMOS Logic Implementations

- General CMOS combinational logic



What is VLSI design?

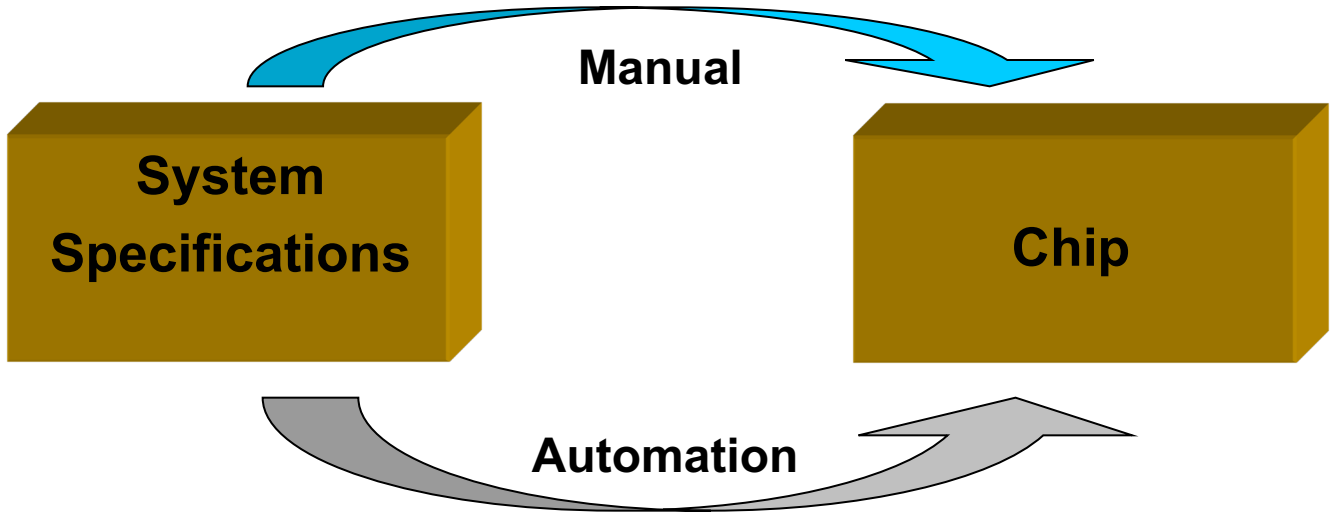
- The process of creating an integrated circuit from specifications to fabrication

What is an integrated circuit?

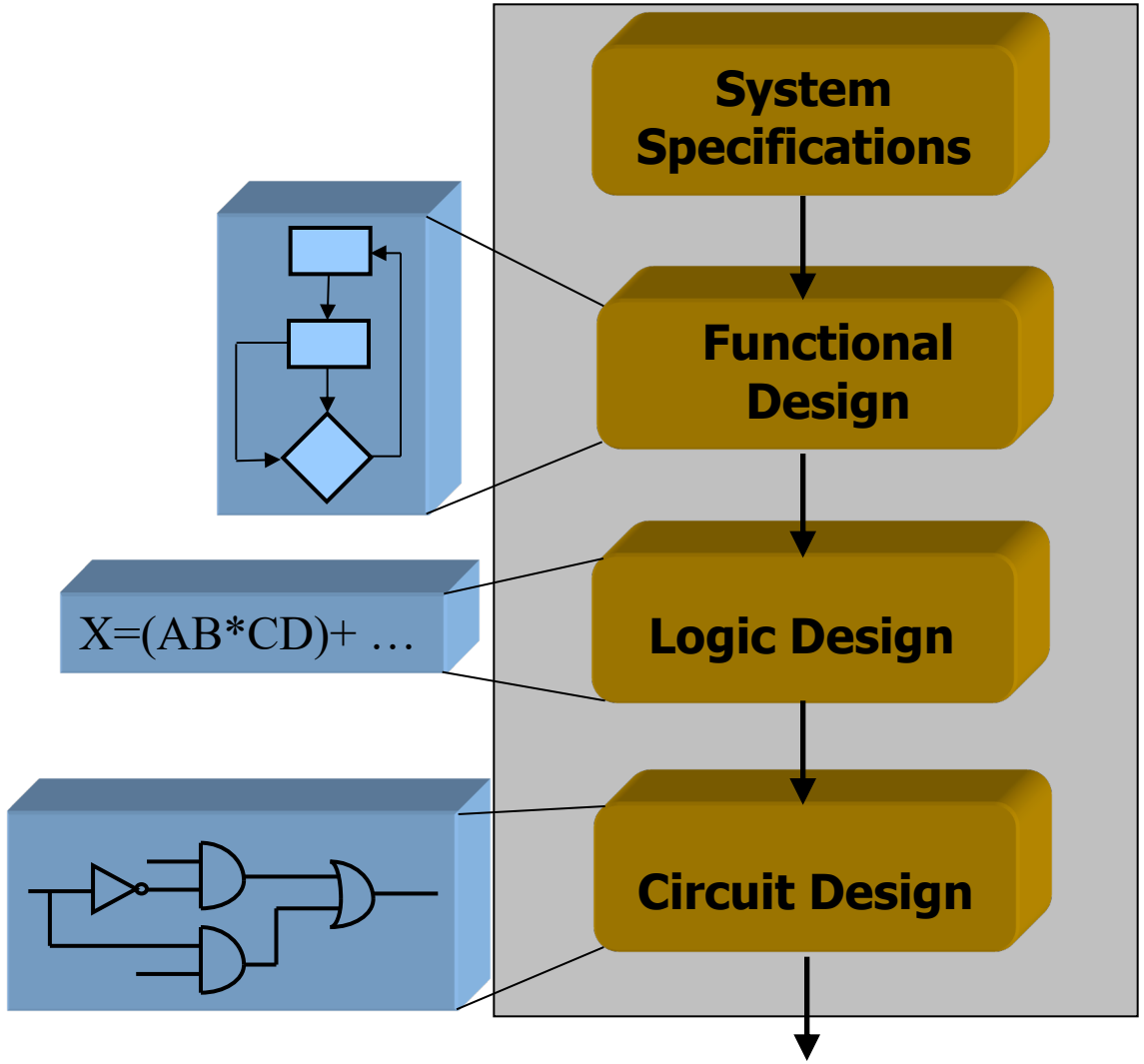
- A single integrated component that contains all the primary elements of an electrical circuit: transistors, wiring, resistors, capacitors, etc.

VLSI Design Automation

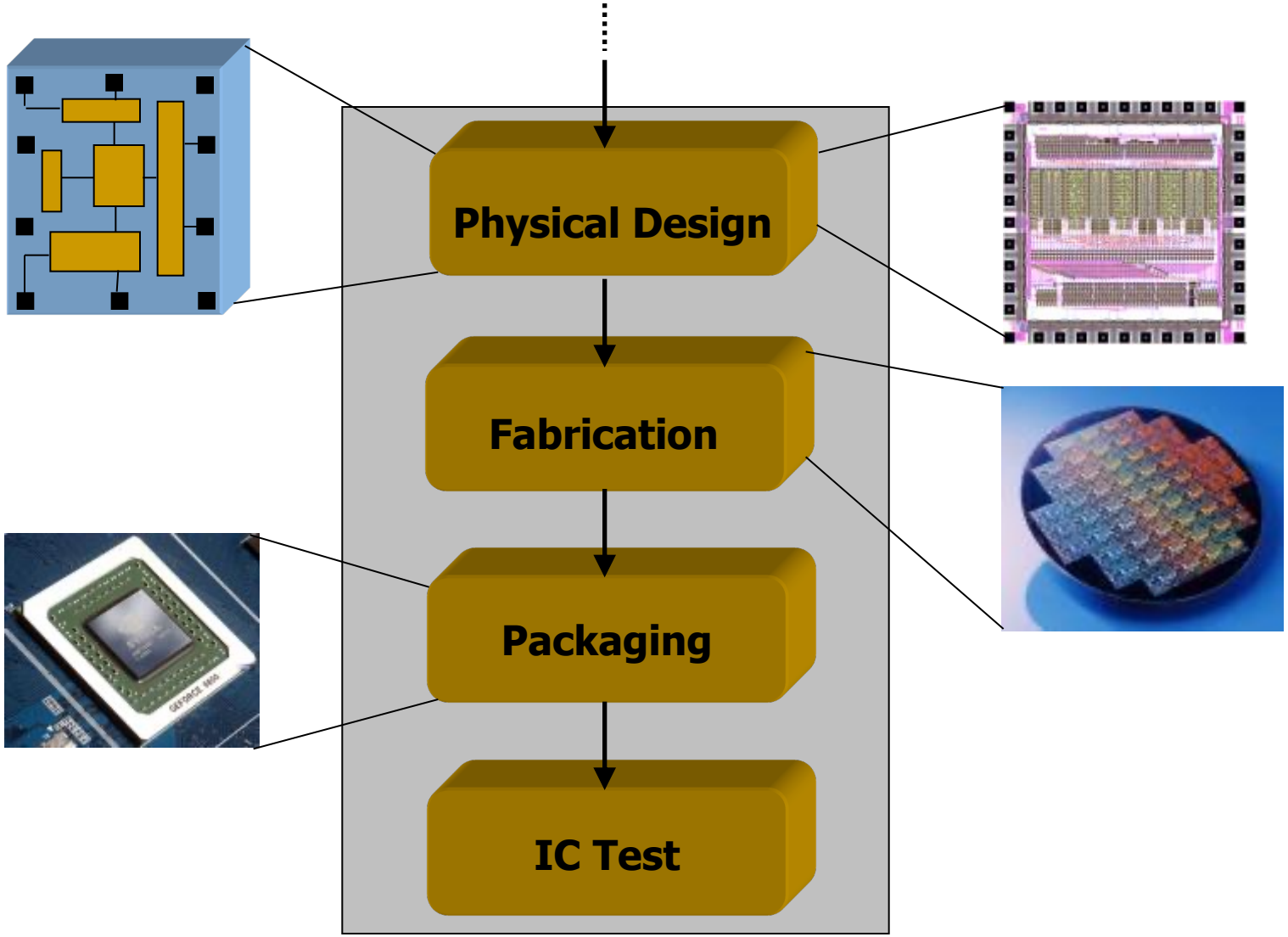
- Large number of components
- Optimize requirements for higher performance
  - Performance relates to speed, power and size.
- Time to market competition
- Cost
  - Using computer makes it cheaper by reducing time-to-market.



VLSI Design Cycle

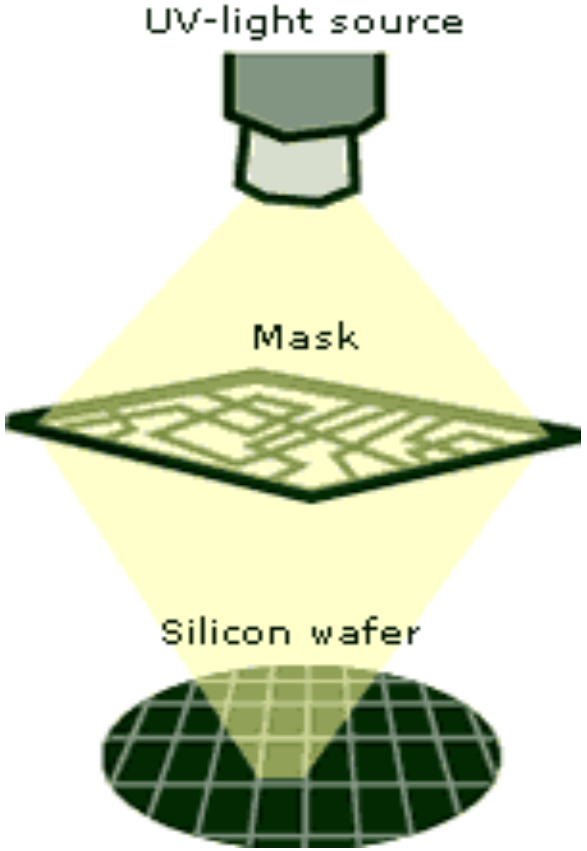


VLSI Design Cycle

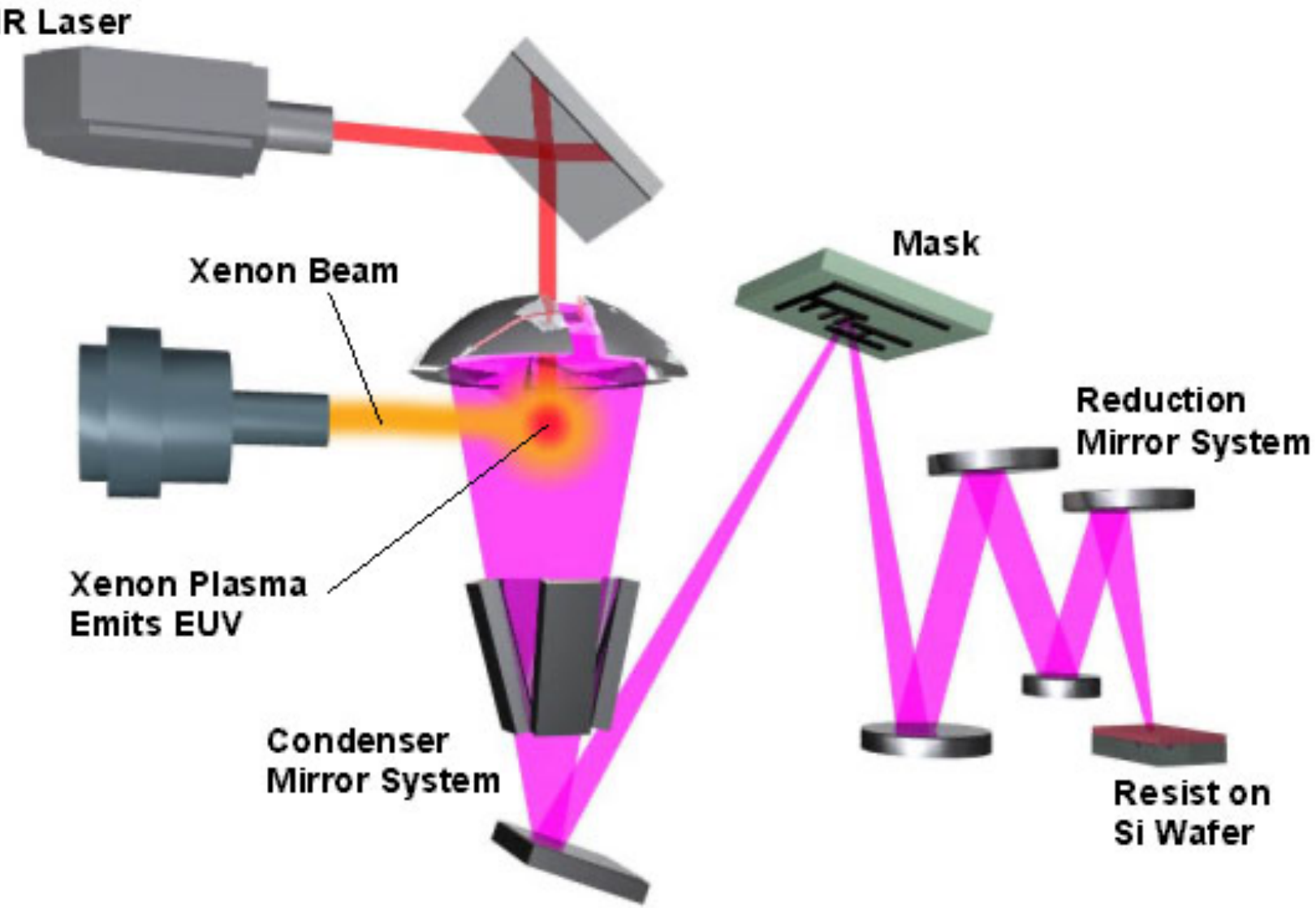


Semiconductor Processing

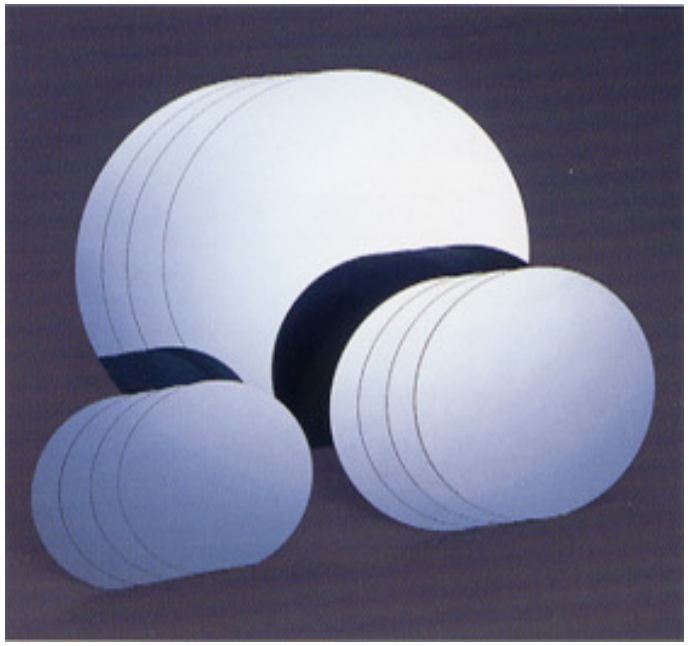
- How do we make a transistor?

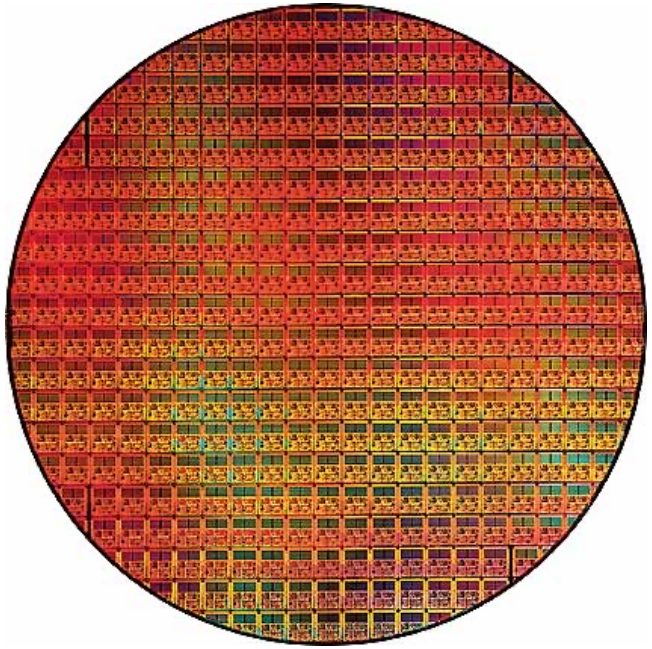


- How do you control where the features get placed?
  - Photo lithography masks

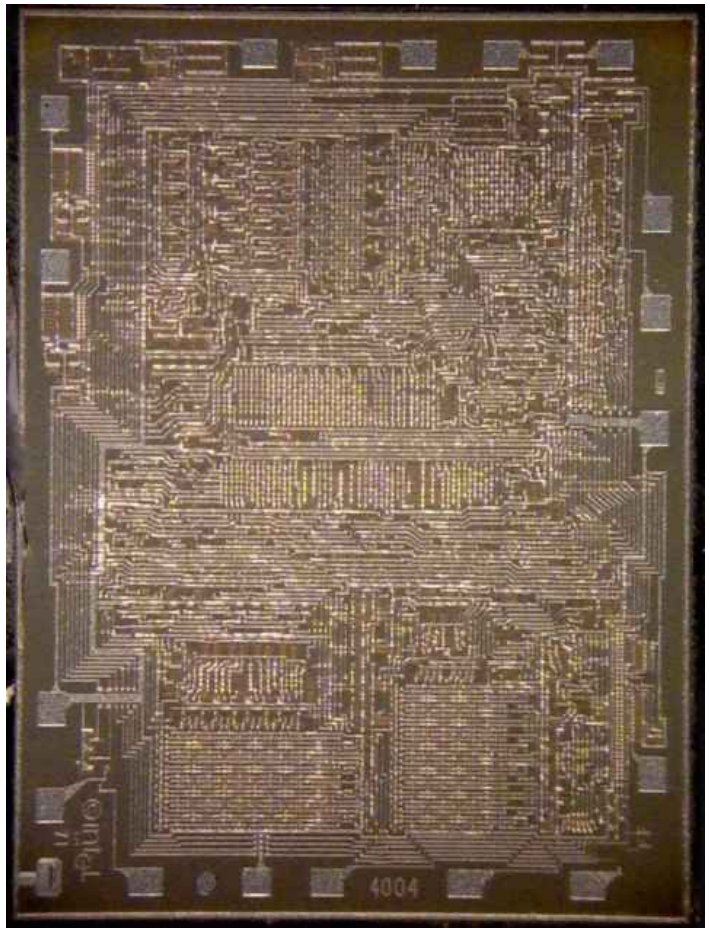

IR Laser
Xenon Beam
Xenon Plasma
Emits EUV
Condenser
Mirror System
Mask
Reduction
Mirror System
Resist on
Si Wafer

Wafer Processing



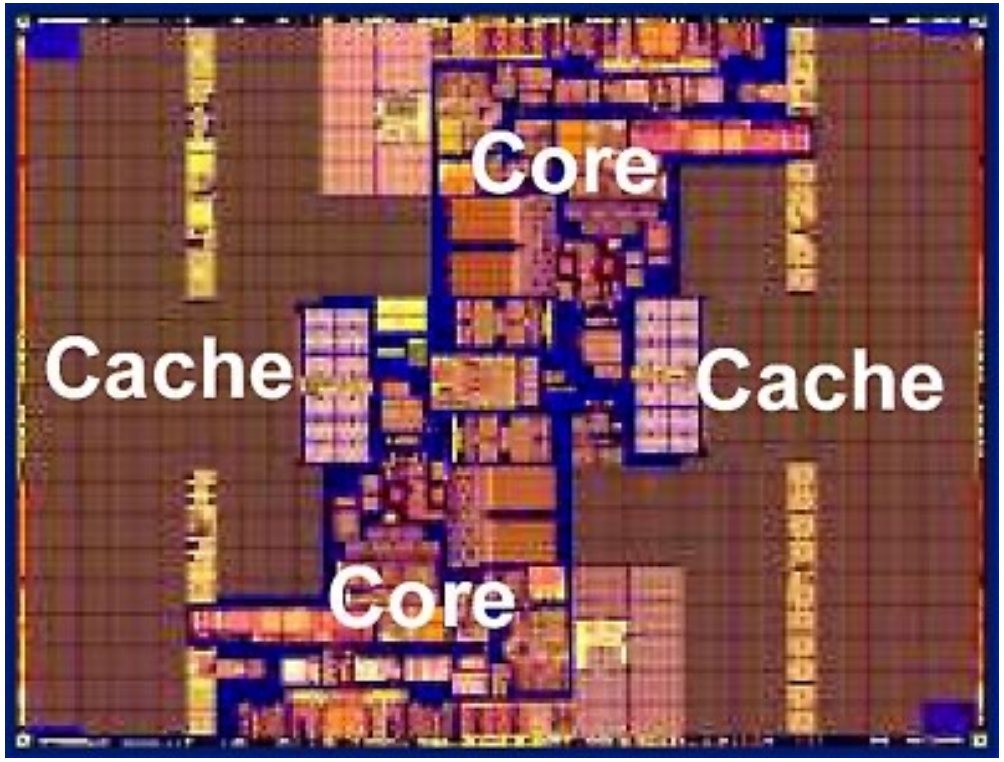


Intel 4004



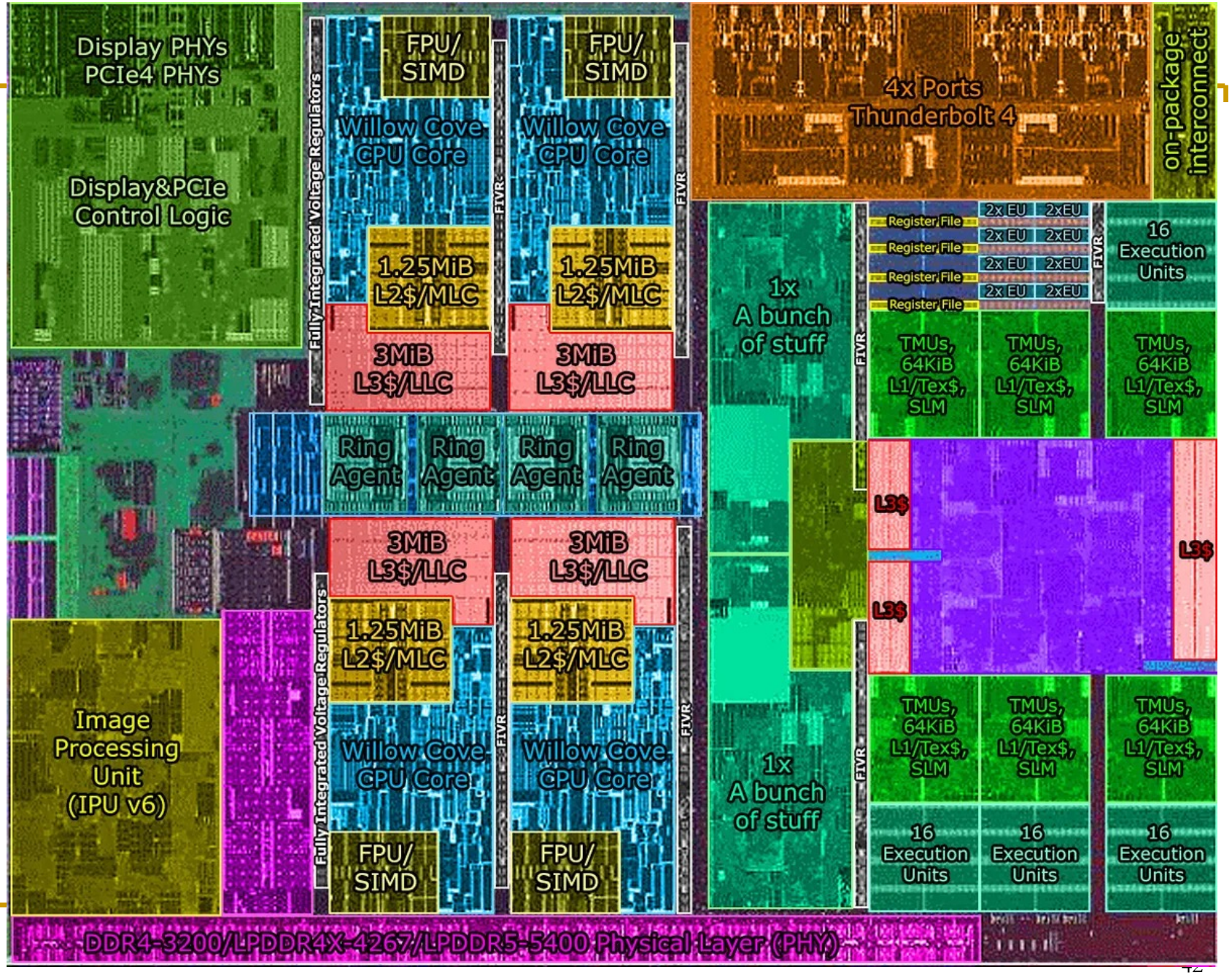
- First microprocessor
- Designed in 1971
- 2300 transistors
- 10-um process
- ~100 KHz

Intel Itanium Processor



- Released in 2005
- 1.72 Billion transistors
- 90-nm process
- 2 GHz

Intel Tiger Lake (10nm)



Design Methodology

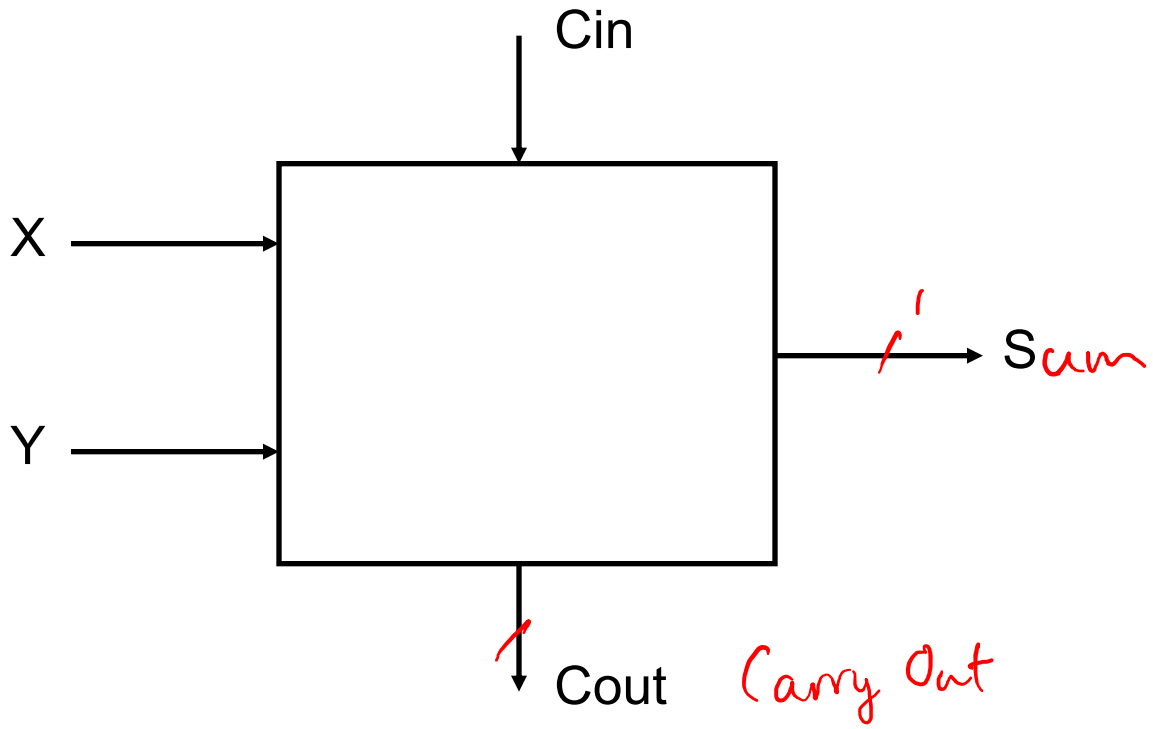
- Functional specification
  - What does the chip do?
- Behavioral specification
  - How does it do it? (abstractly)
- Logic design
  - How does it do it? (logically)
- Layout
  - How does it do it? (physically)

Design Constraints

- Budget
  - Total cost
- Silicon area
- Power requirements
  - Dynamic
  - Static
- Speed
  - Performance
- Schedule
  - Time to market

Functional Specification

- Full adder



Behavioral Specification

- VHDL
- Verilog

```
entity adder is
        -- i0, i1 and the carry-in ci are inputs of the adder.
        -- s is the sum output, co is the carry-out.
        port (i0, i1 : in bit; ci : in bit; s : out bit; co : out bit);
 end adder;
 architecture rtl of adder is
        begin -- This full-adder architecture contains two concurrent assignment.
         -- Compute the sum. s <= i0 xor i1 xor ci;
        -- Compute the carry. co <= (i0 and i1) or (i0 and ci) or (i1 and ci);

end rtl;
```

Behavioral Specification

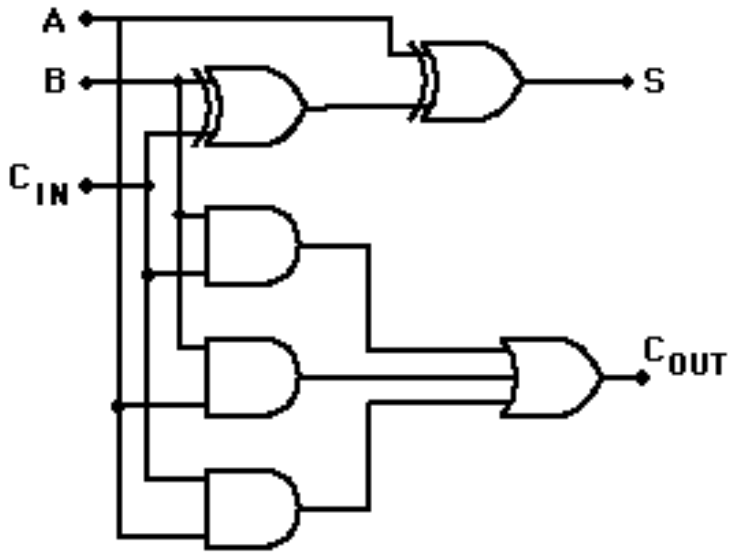
```
module fulladder (a,b,cin,sum,cout);
    input a,b,cin;
    output sum,cout;

    reg sum,cout;
    always @ (a or b or cin)
    begin
       sum <= a ^ b ^ cin;
       cout <= (a & b) | (a & cin) | (b & cin);
    end
endmodule
```

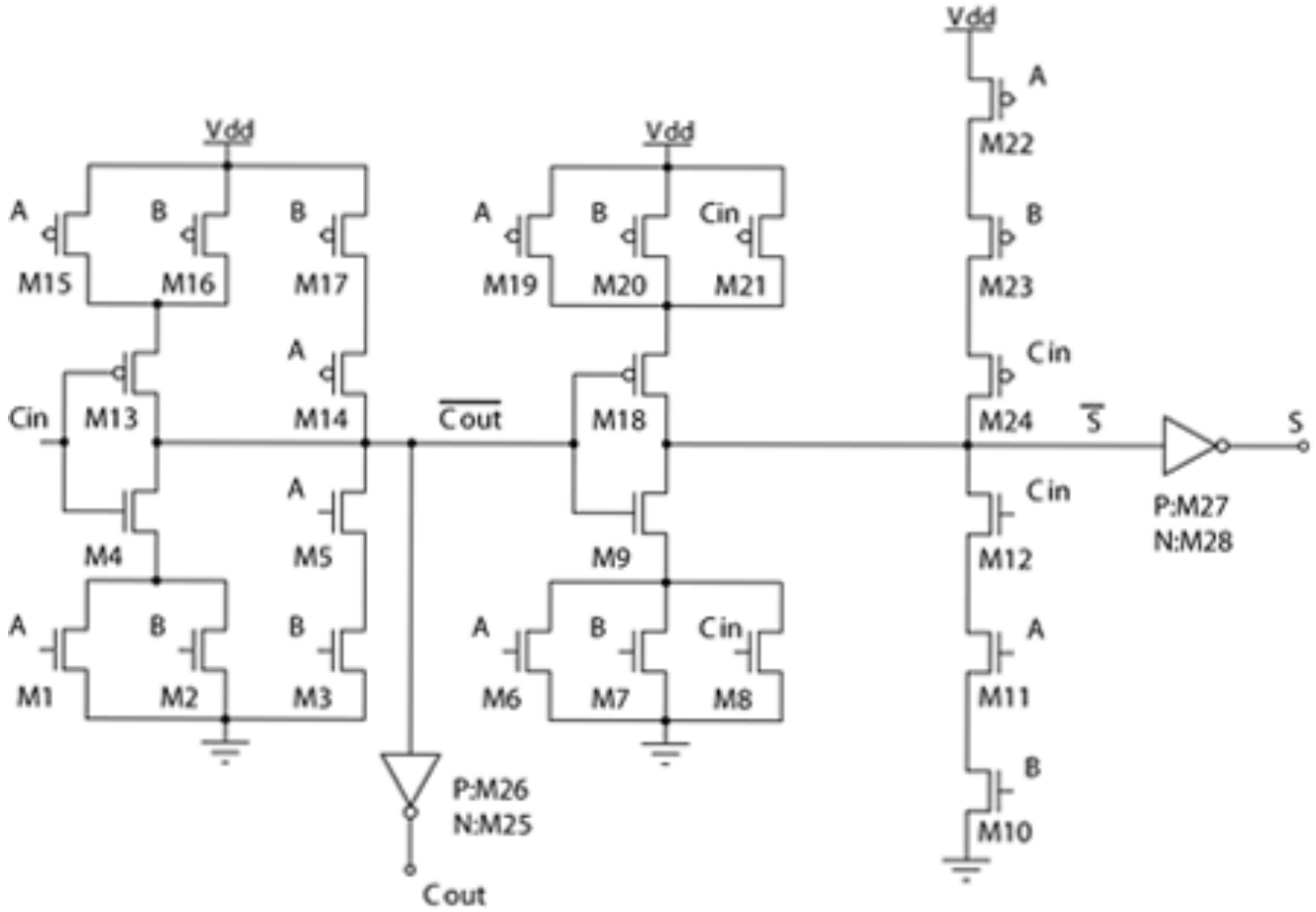
Logic Design

Full Adder Truth Table

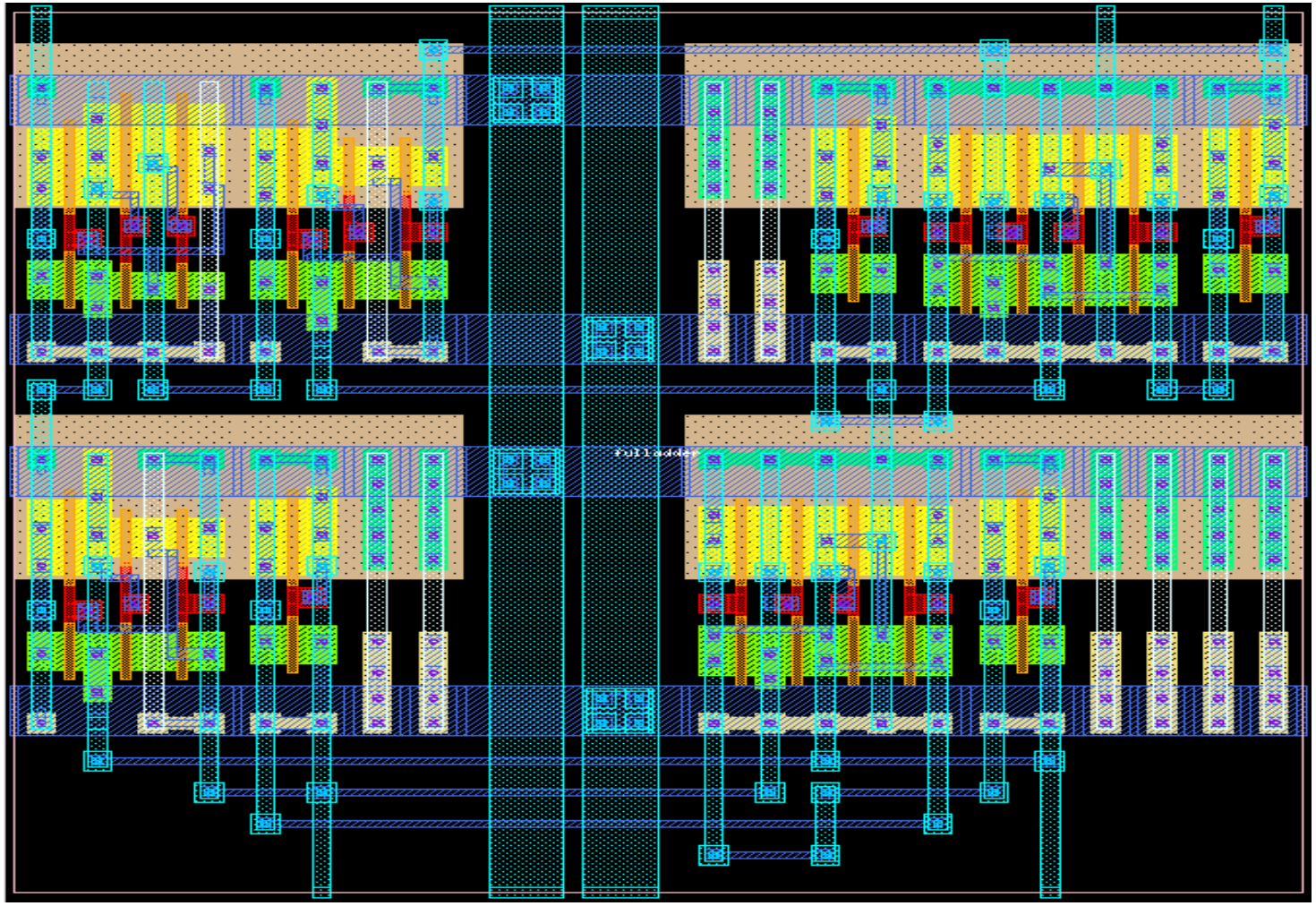
CARRY IN	*input B*	*input A*	*CARRY OUT*	*SUM digit*
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

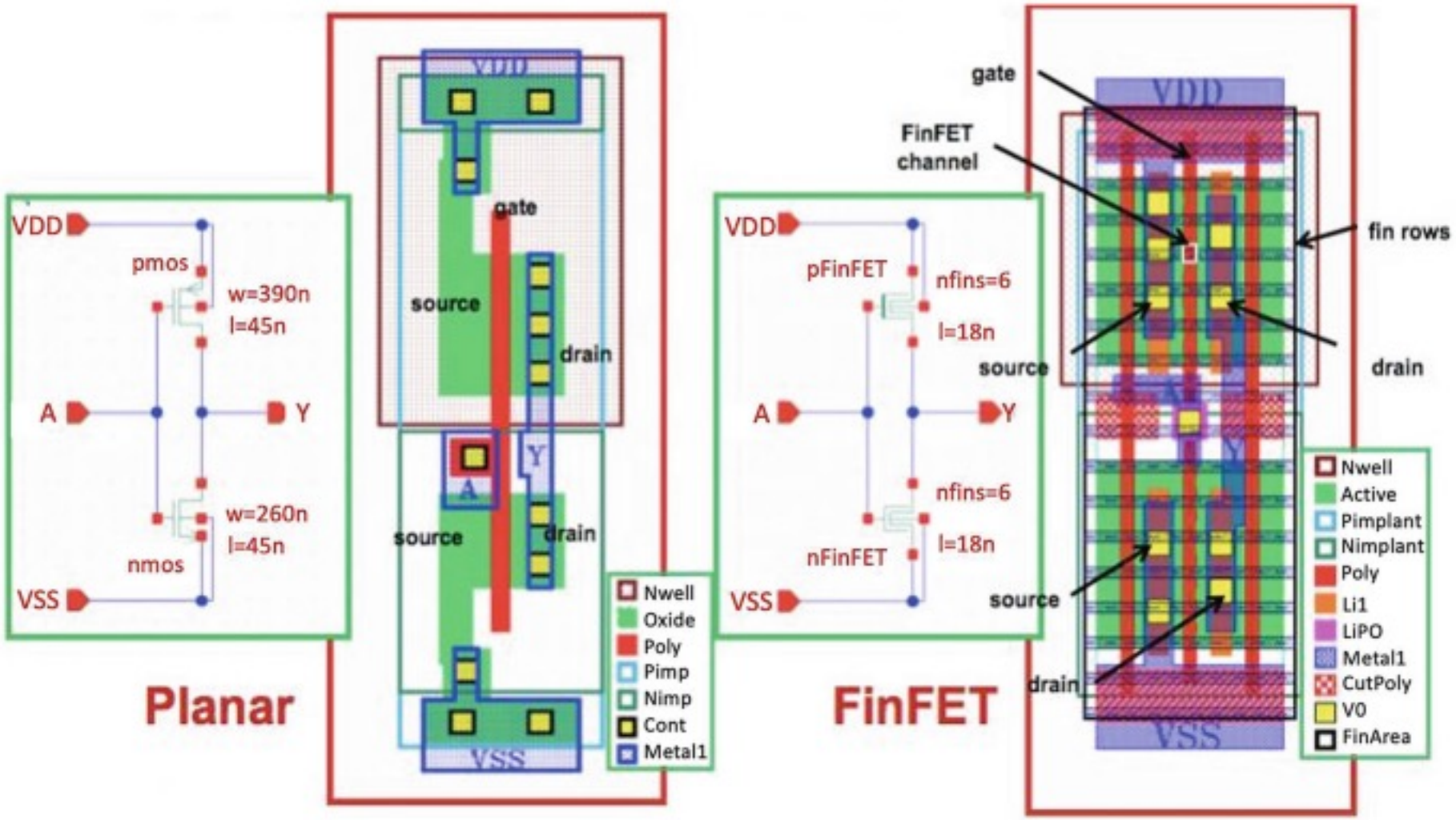


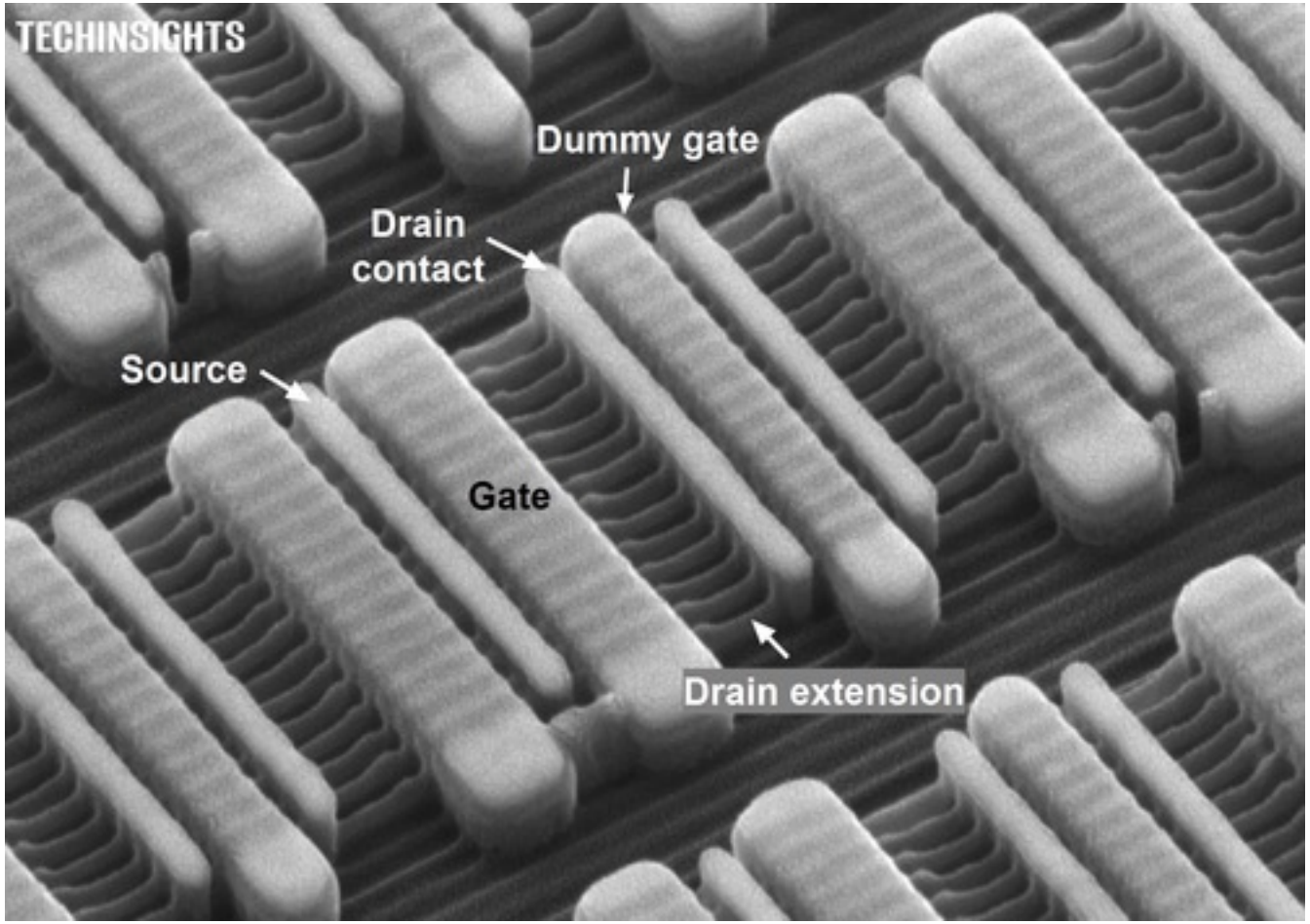
Transistor Schematic



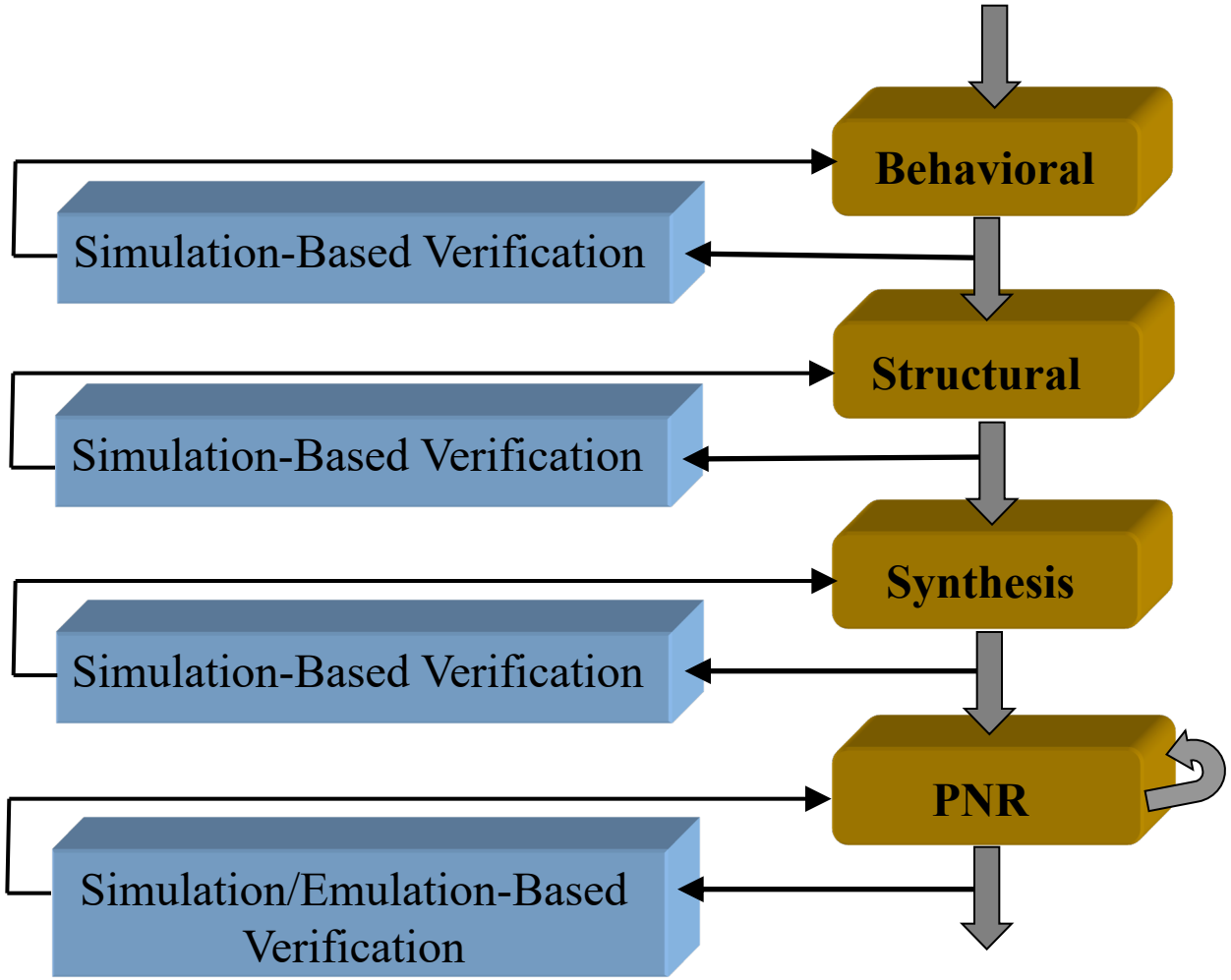
Layout




VDD
pmos
w=390n
l=45n
A
Y
w=260n
l=45n
nmos
VSS
Planar
gate
source
drain
Y
A
source
drain
VDD
VSS
Nwell
Oxide
Poly
Pimp
Nimp
Cont
Metal1
gate
FinFET
channel
VDD
pFinFET
nfins=6
l=18n
source
A
Y
nfins=6
l=18n
nFinFET
VSS
source
drain
fin rows
drain
VDD
VSS
Nwell
Active
Pimplant
Nimplant
Poly
Li1
LiPO
Metal1
CutPoly
V0
FinArea
FinFET


TECHINSIGHTS
Dummy gate
Drain
contact
Source
Gate
Drain extension

Design Process is Iterative



PNR: Placement and routing

VLSI Design Methodologies

- Full custom
  - Design for performance-critical cells
  - Very expensive
- Standard cell
  - Faster
  - Performance is not as good as full custom
- Gate array
- Field Programmable Gate Array

Comparison of Design Styles

	Full Custom	Standard Cell	Gate Array	FPGA
Area	Compact	Moderate	Moderate	Large
Performance	High	Moderate	Moderate	Low

Production Volume:	**Mass Production Volume**	**Medium Production Volume**	**Medium Production Volume**	**Low Production Volume**
Complexity:	**High**			**Low**

VLSI Chip Yield

- A manufacturing defect in the fabrication process causes electrically malfunctioning circuitry.
- A chip with no manufacturing defect is called a good chip.
  - The defective ones are called bad chips.
- Percentage of good chips produced in a manufacturing process is called the *yield*.
- Yield is denoted by symbol *Y*.

$$Y = \frac{\text{\# of good die}}{\text{\# total manufactured die}}$$

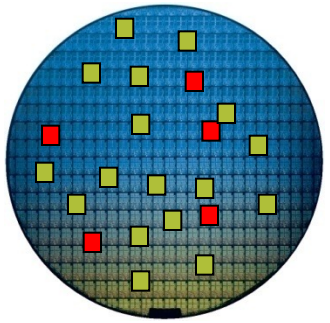
- How to separate bad chips from the good ones?

TEST ALL CHIPS

Why Does Test Matter ?

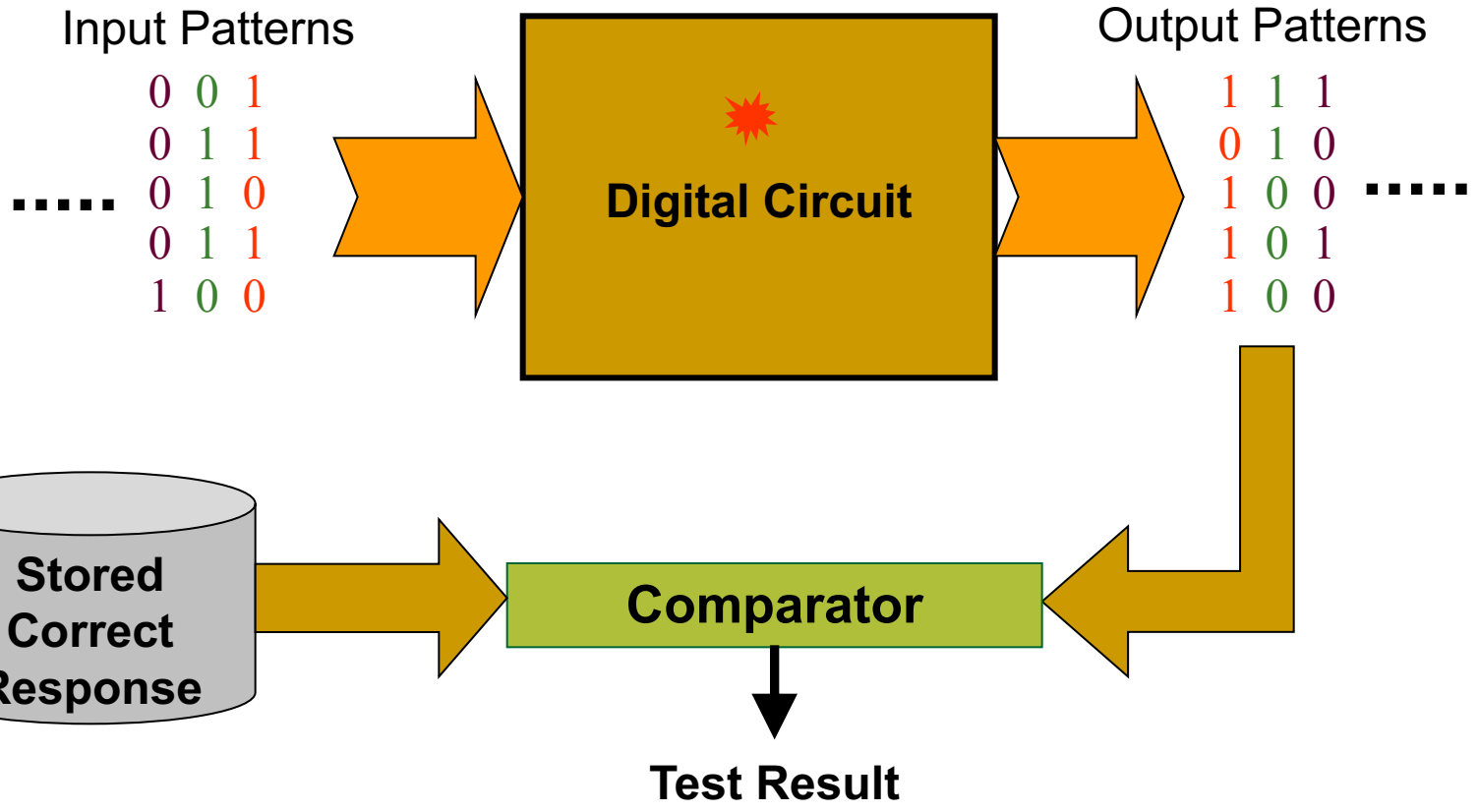
- In simple terms, TEST identifies the defective chips
- Some bad chips (■) are easy to find

Wafer



- Some other are difficult (■)
- Test is associated with
  - Cost
  - Return On Investment (ROI)
    - ¥ € $ - Money

Testing Principle



Functional Test Method – Not very efficient

Contract between design house and fab vendor

- Design is complete and checked (verified)
- Fab vendor: How will you test it?
- Design house: I have checked it and …
- Fab vendor: But, how would you test it?
- Design house: Why is that important?
- *complete the story*

- That is one reason for design-for-testability, test generation etc.

Contract between design …

Hence:

- "Test" must be comprehensive
- It must not be "too long"

Issues:

- Model possible defects in the process
  - Understand the process
- Develop simulator and fault simulator
- Develop test generator
- Methods to quantify the test efficiency
  - Fault coverage

Ideal Tests

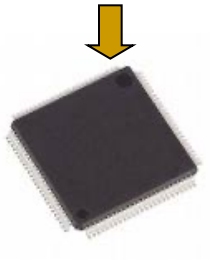
- Ideal tests detect **all** defects produced in the manufacturing process.
- Ideal tests pass all functionally good devices.
- Very large numbers and varieties of possible defects need to be tested.
- Difficult to generate tests for some real defects. *Defect-oriented testing is an open problem.*

Real Tests

- Based on analyzable fault models, which may not map on real defects.
- Incomplete coverage of modeled faults due to high complexity.
- Some good chips are rejected. The fraction (or percentage) of such chips is called the **yield loss**.
- Some bad chips pass tests. The fraction (or percentage) of bad chips among all passing chips is called the **defect level**.

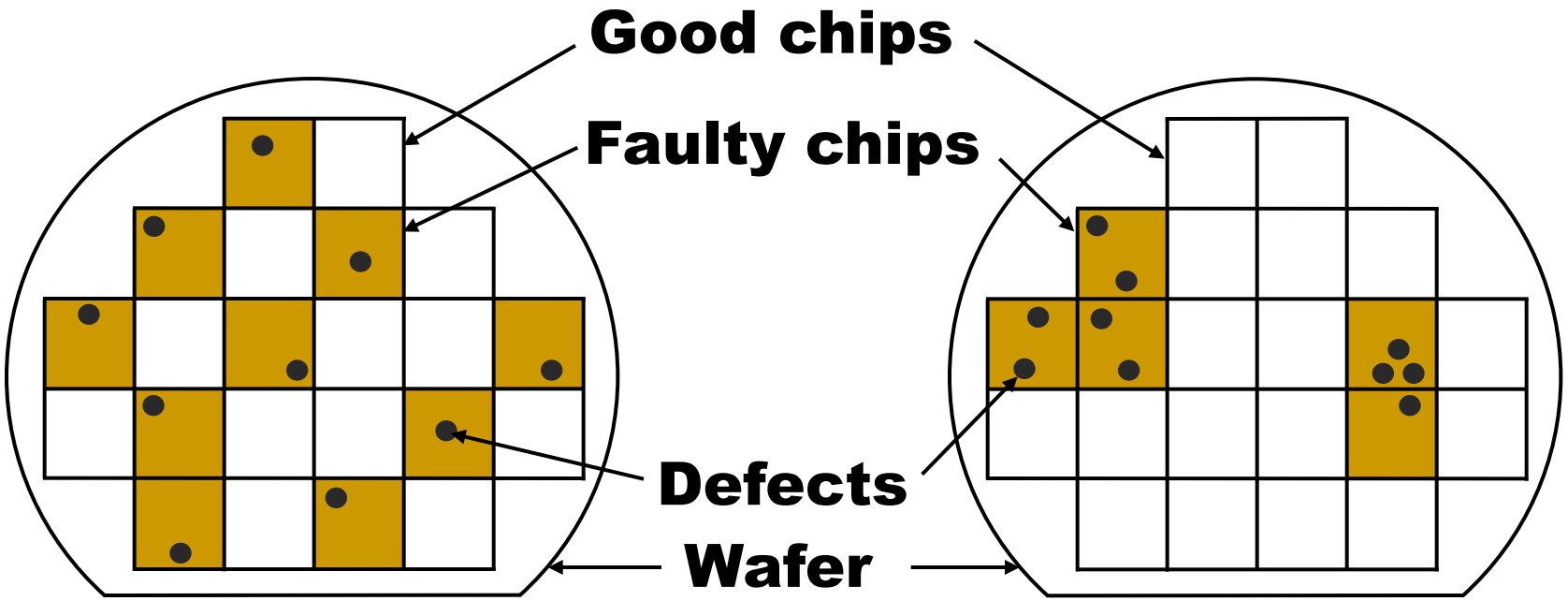
Level of testing (1)

- Levels
  - Chip
  - Board
  - System
    - Boards put together
    - System-on-Chip (SoC)
  - System in field
- Cost – Rule of 10
  - It costs 10 times more to test a device as we move to higher level in the product manufacturing process



VLSI Defects

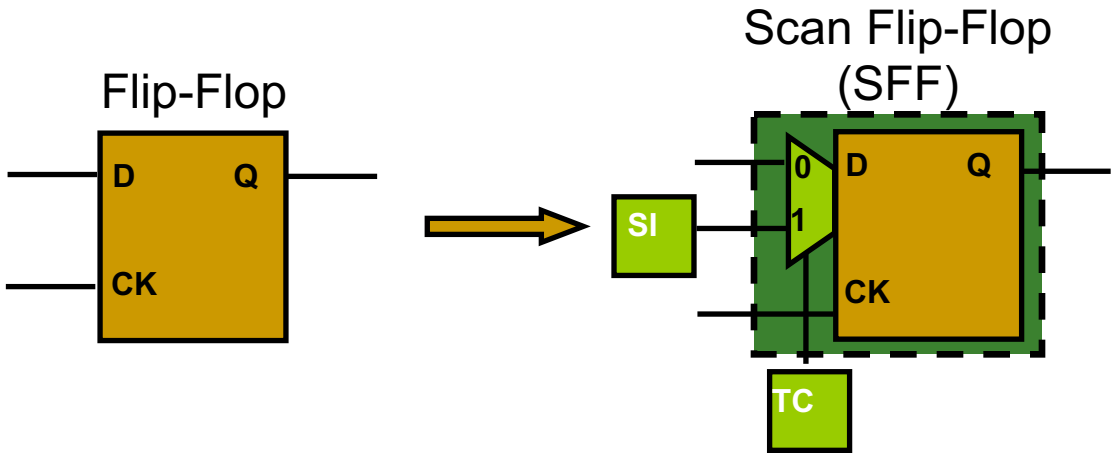


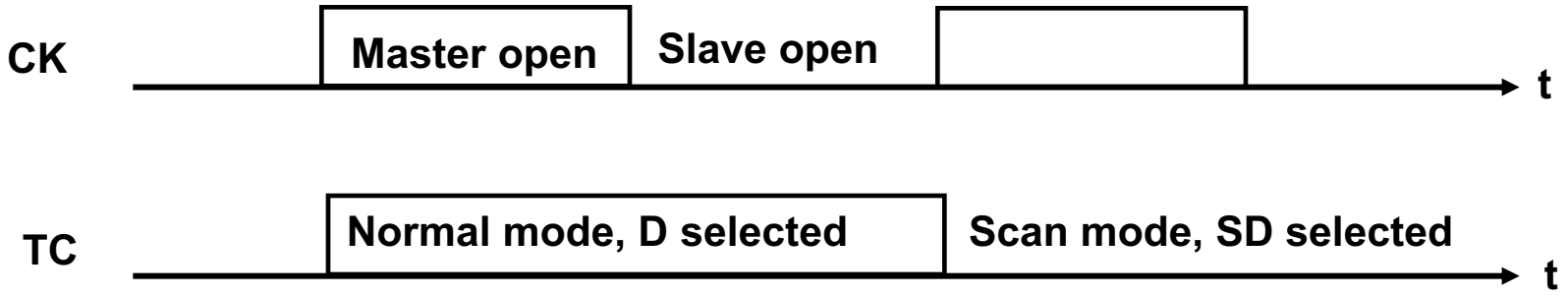
Unclustered defects
Wafer yield = 12/22 = 0.55

Clustered defects (VLSI)
Wafer yield = 17/22 = 0.77

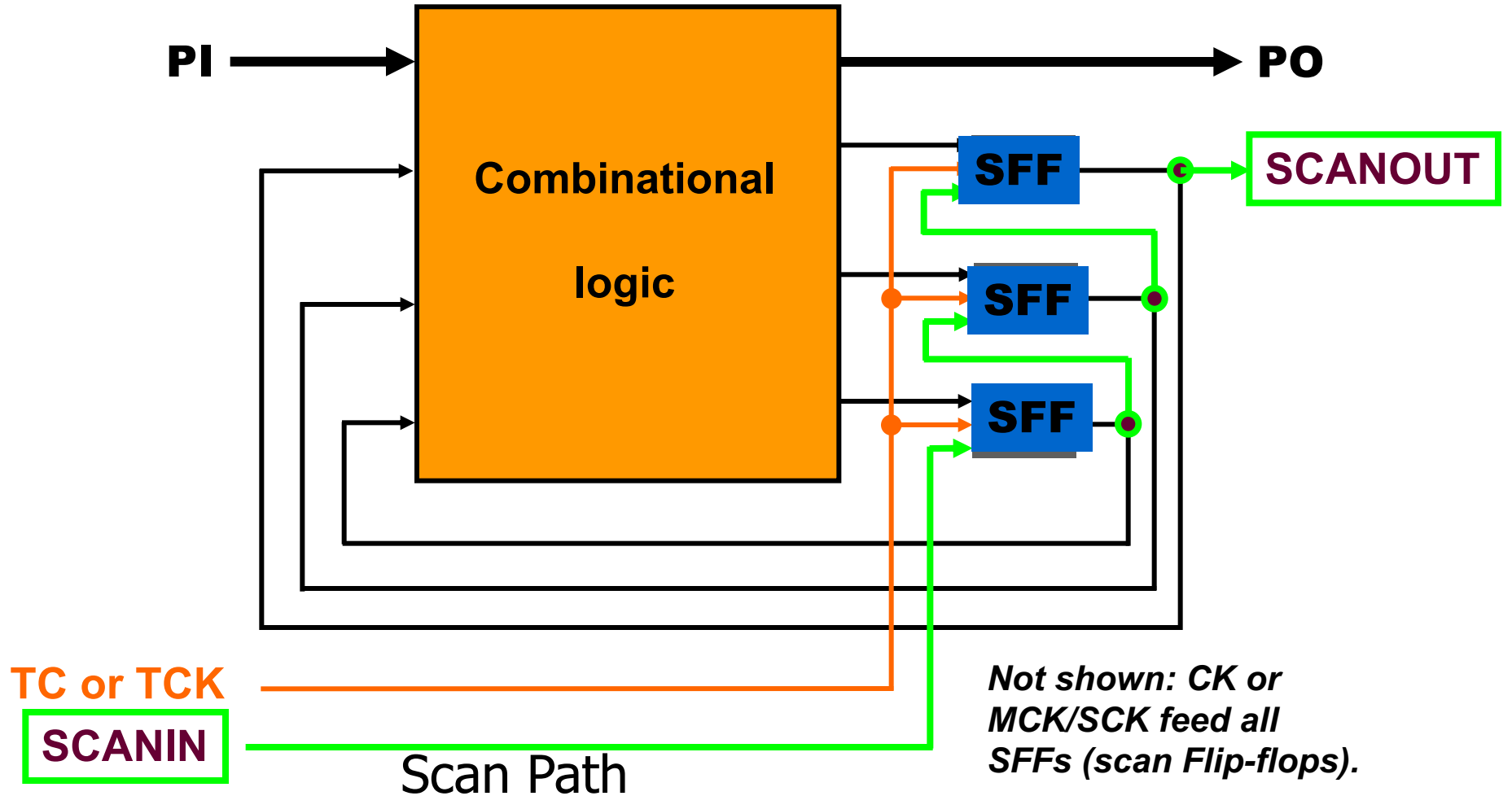
Scan Flip-Flop

Stop here!



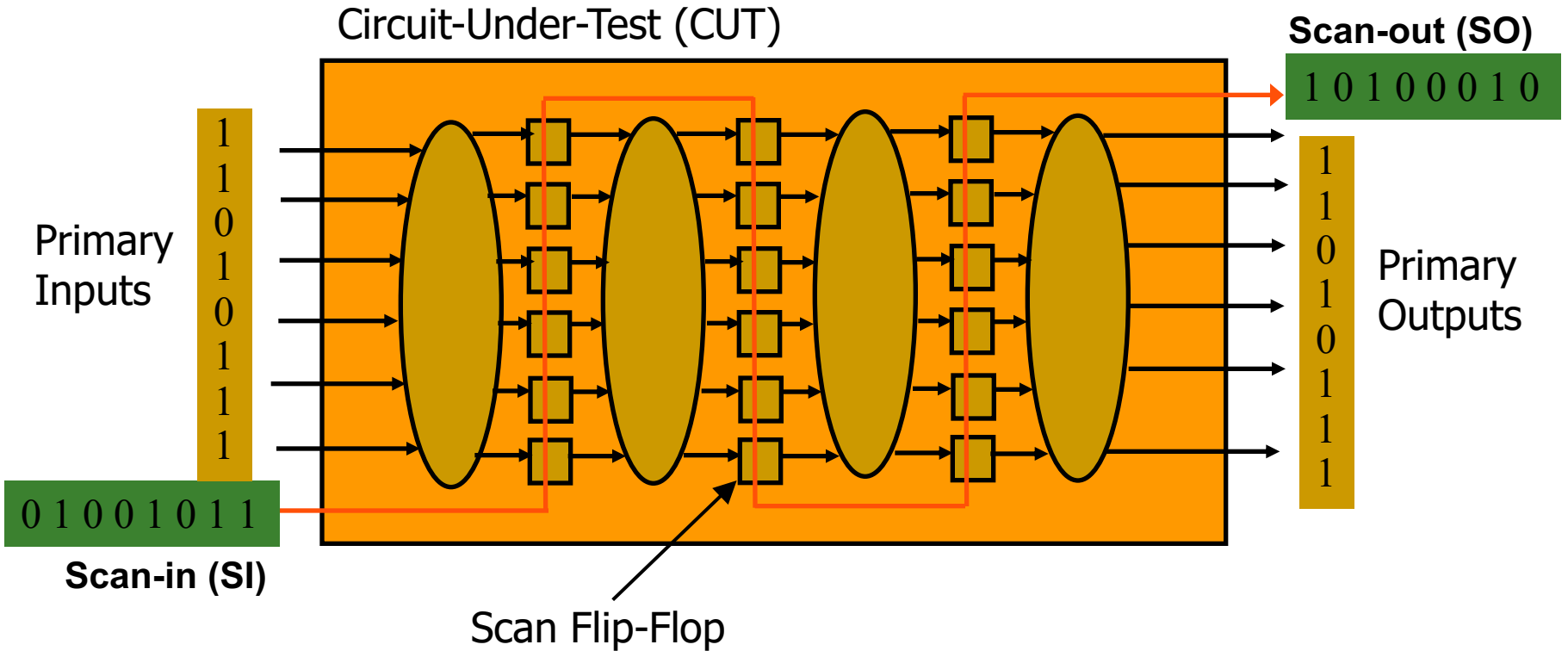


Adding Scan Structure



Scan Path
Also called Scan Chain

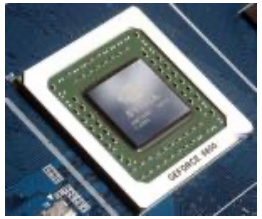
Scan Design



Structural Test Method – Extremely efficient

ADVANTEST Model T6682 ATE

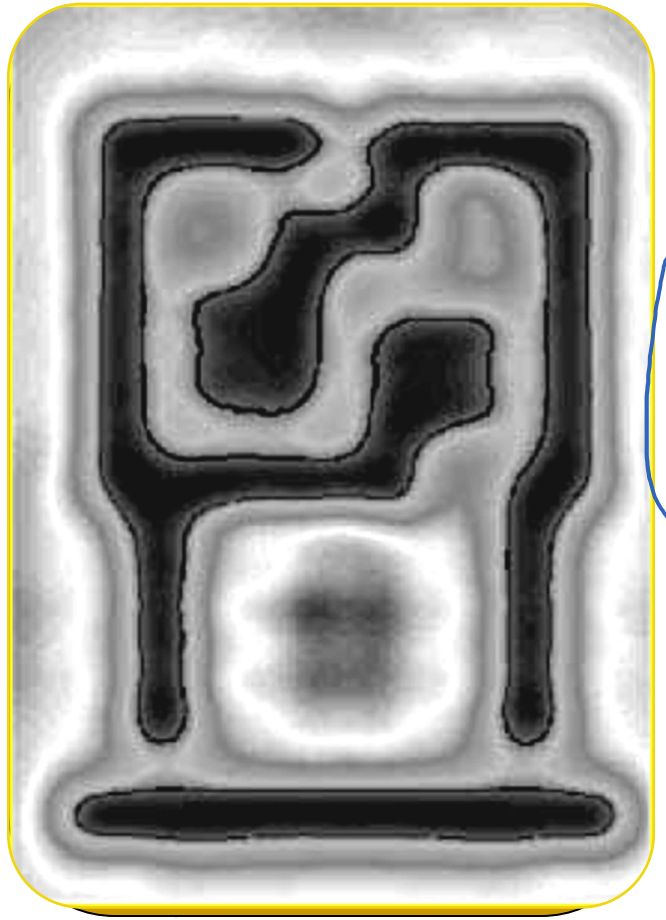
Test Head



Testers are very expensive ($150K – $20M)

WYSINWYG

Sub-Wavelength WYSINWYG



What You See Is Not What You Get

Process variations

No two transistors have the same parameters