



# *Digital System Design*

## **Overview of Digital Design**

**Lecturer:** Prof. An-Yeu (Andy) Wu

**Date:** 2025/02/20



# High-Performance Digital Design in SoC Era

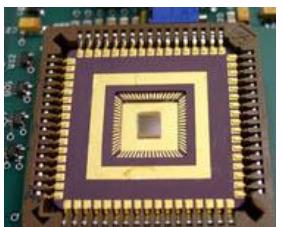
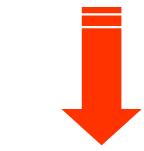
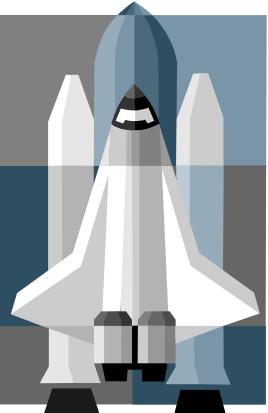
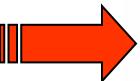
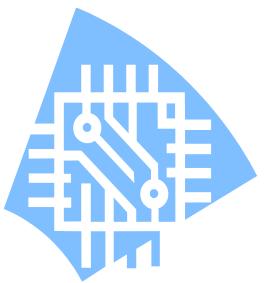


## IC Design and Implementation

Idea



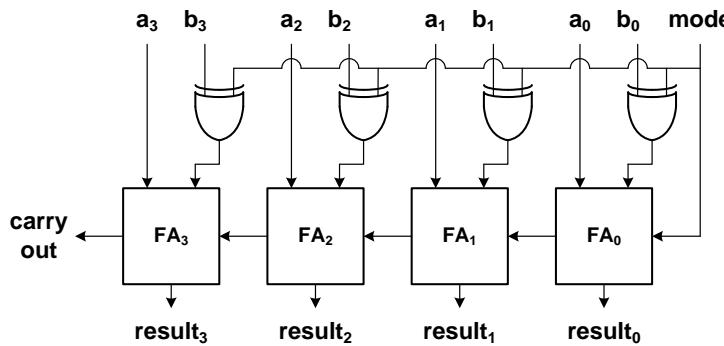
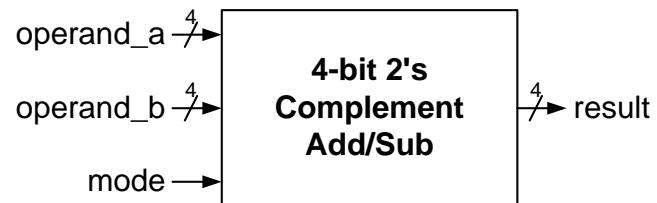
Design





## Digital IC Design Flow

1. Concept/Application
2. Function/Spec. definition
3. Algorithm exploration
4. Architecture design
  1. Divide-and-conquer
  2. Sub-module design
  3. Design verification
5. System prototyping (need training!!)
  1. RTL design
  2. Verilog Coding/Schematic Design
  3. Cell-based IC design flow/FPGA design flow

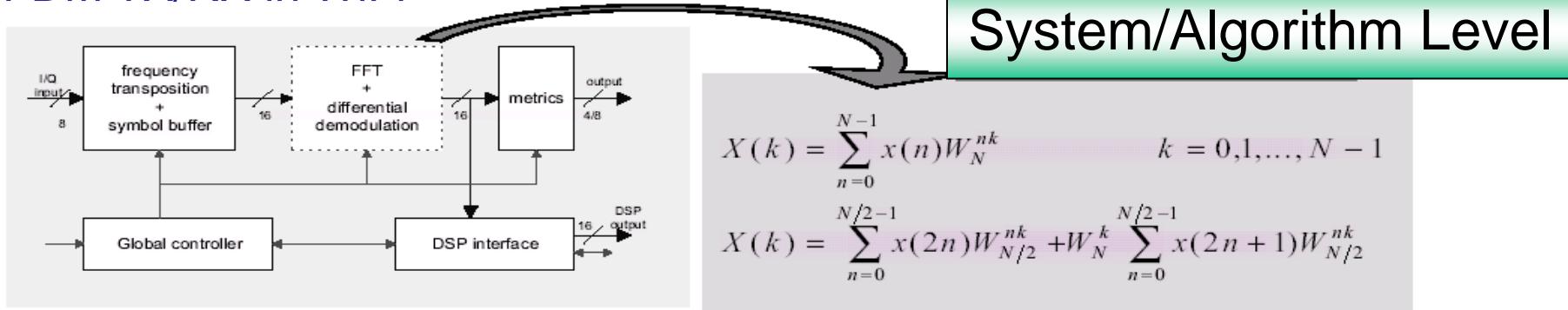


```
module Add_Sub_Unit( result, operand_a, operand_b, mode, detect );
    input [3:0] operand_a, operand_b;
    input mode;
    output [3:0] result;
    output detect; // for question 3
    wire [3:0] xor_b;
    xor g0 ( xor_b[0], operand_b[0], mode );
    
```

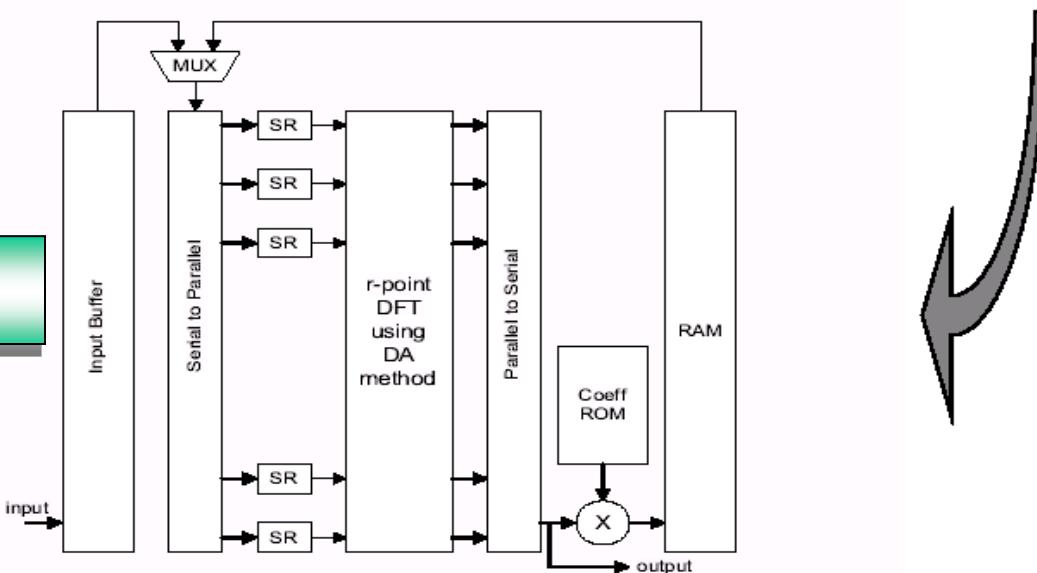


## Algorithm Mapping

OFDM TX/RX in WiFi

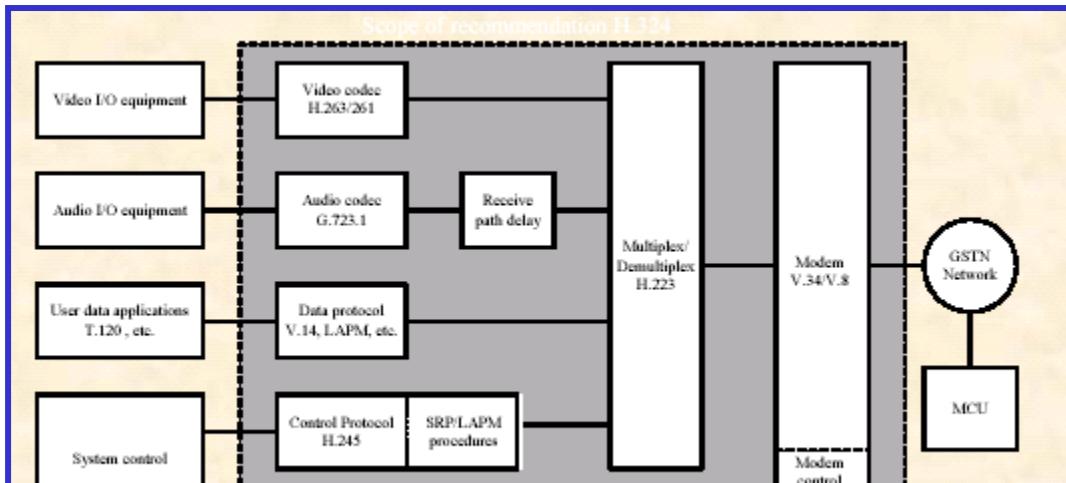


RTL Level





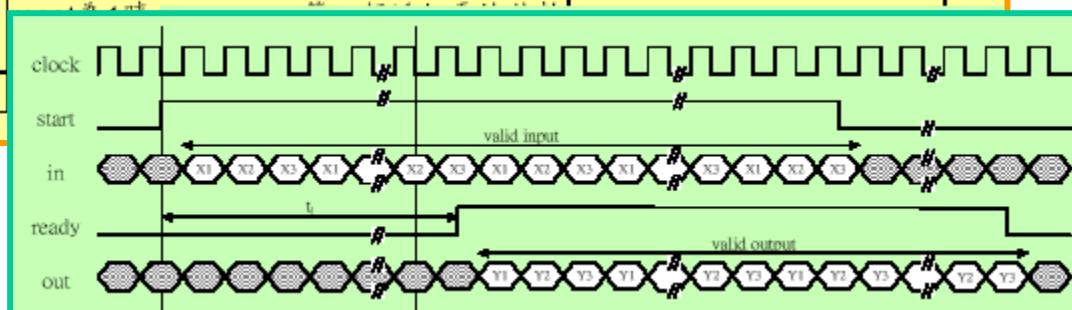
## System Specification



Partition

腳位名稱		描述	Drive Strength/Output Load
clk	輸入	系統時脈	assume infinite
reset	輸入	系統重置訊號，high active	1 ns/pf
din	輸入	每個clock cycle輸入一個16-bit 正整數	1 ns/pf
ready	輸出		
dout	輸出		

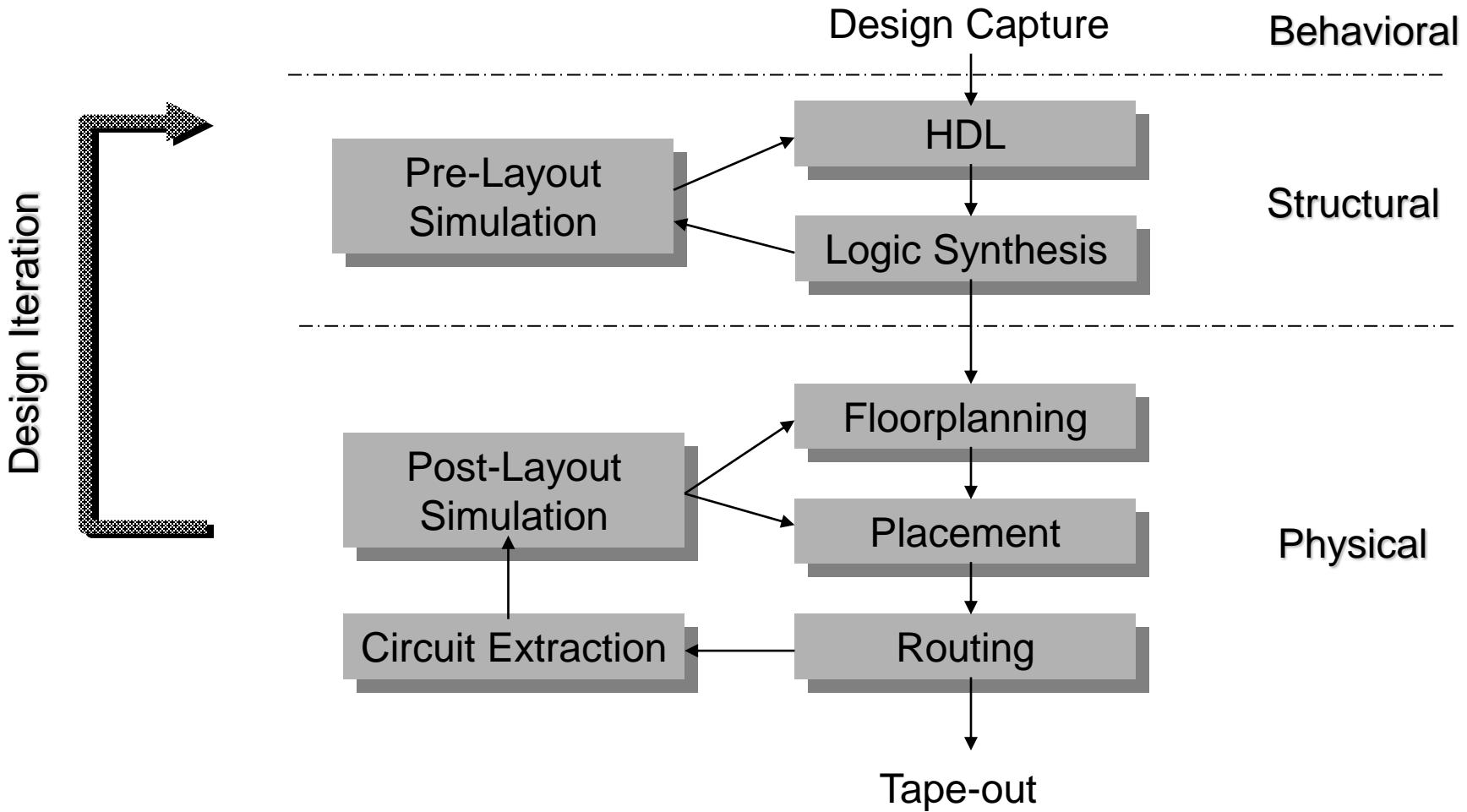
I/O Spec.



I/O Timing Spec.

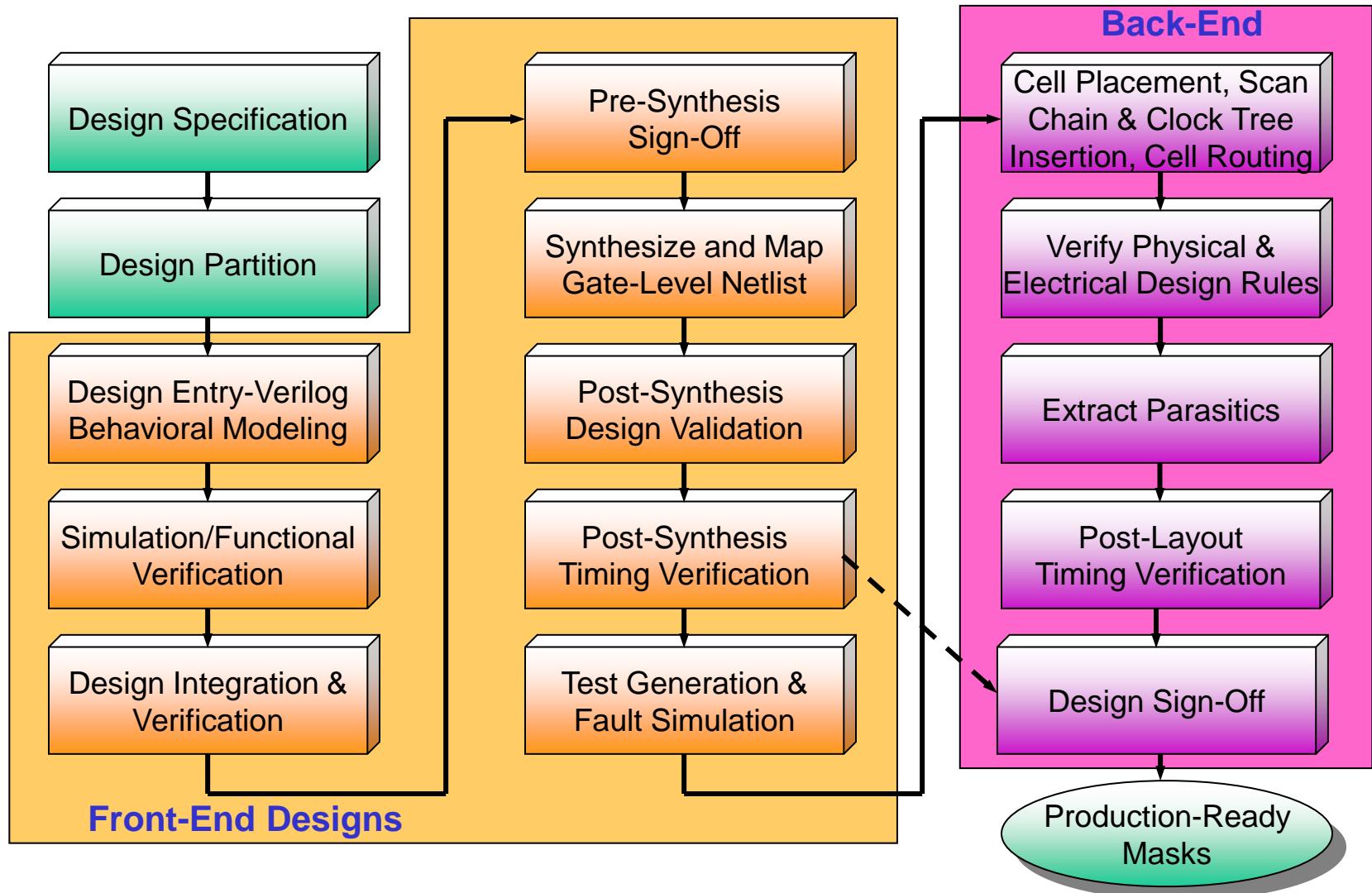


## Semi-Custom Design Flow



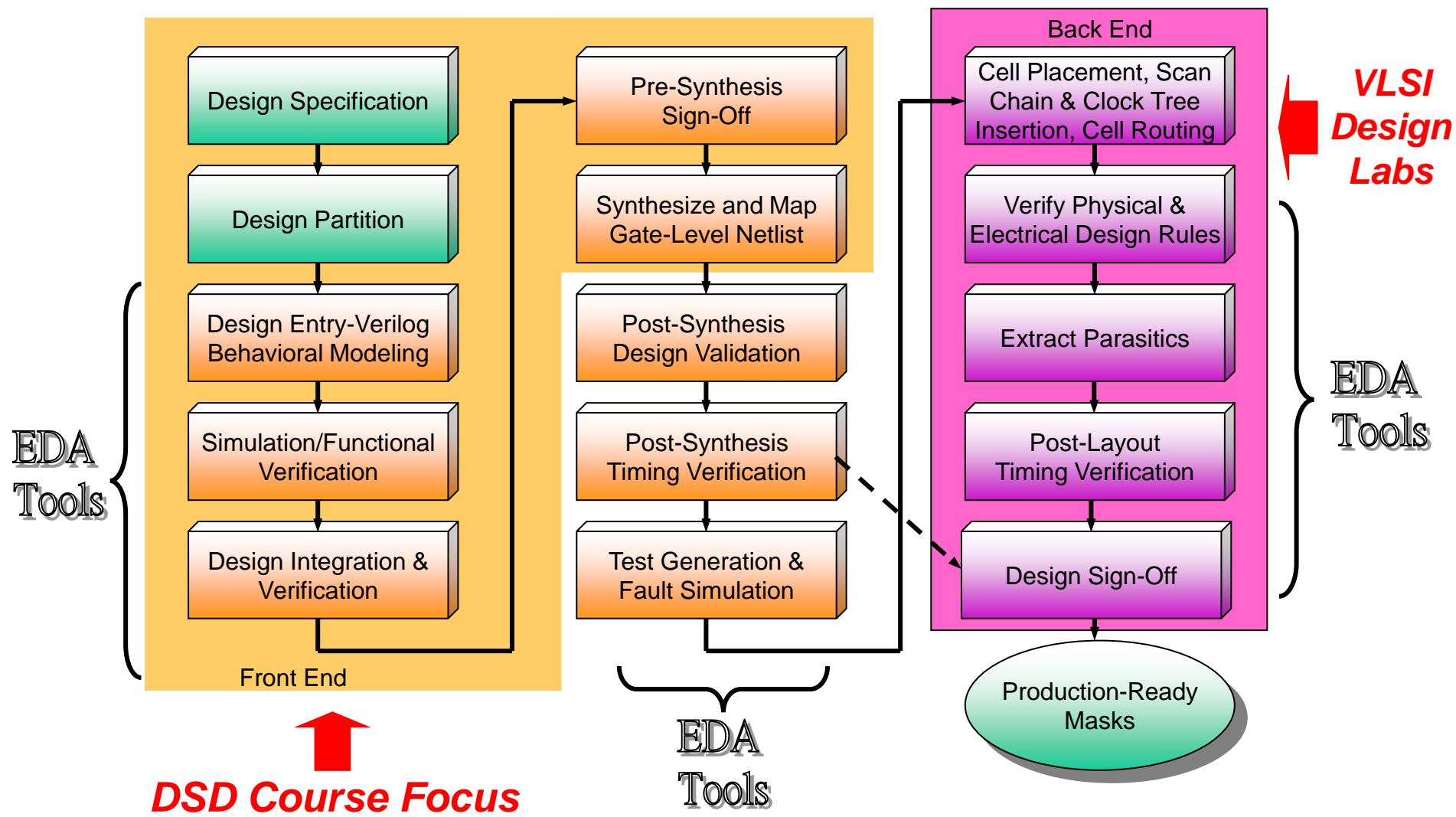


## Digital IC Design Flow





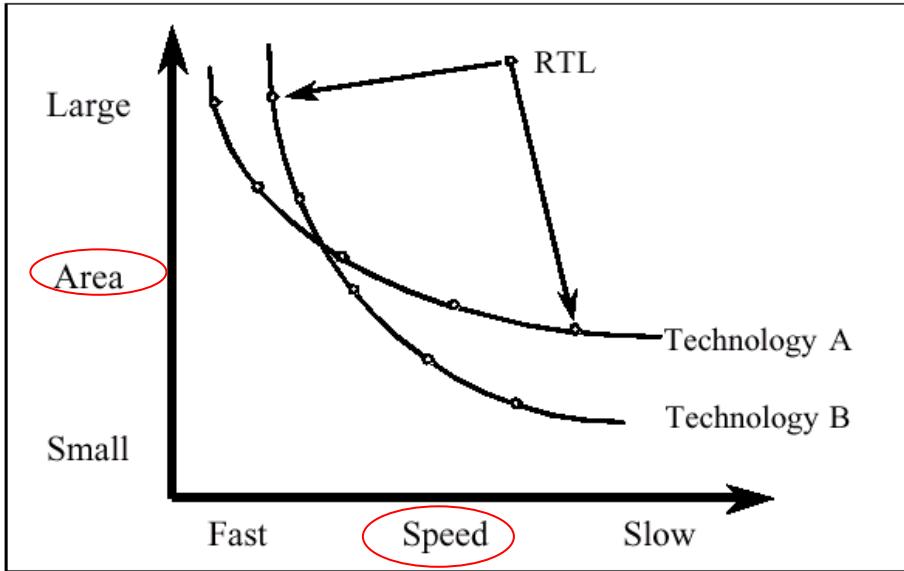
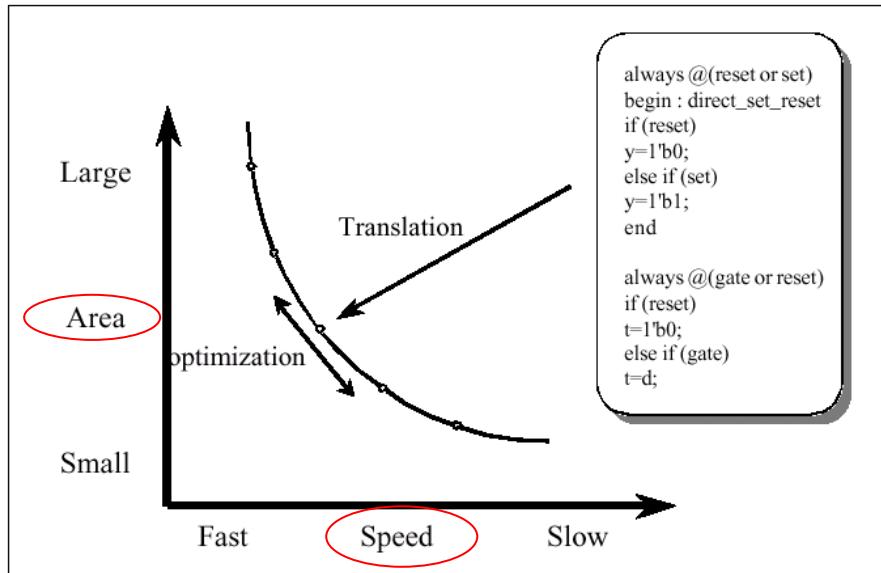
## Cell-based IC Design Flow





## Concept of Synthesis

- ❖ Synthesis is Constraint Driven
- ❖ Technology Independent

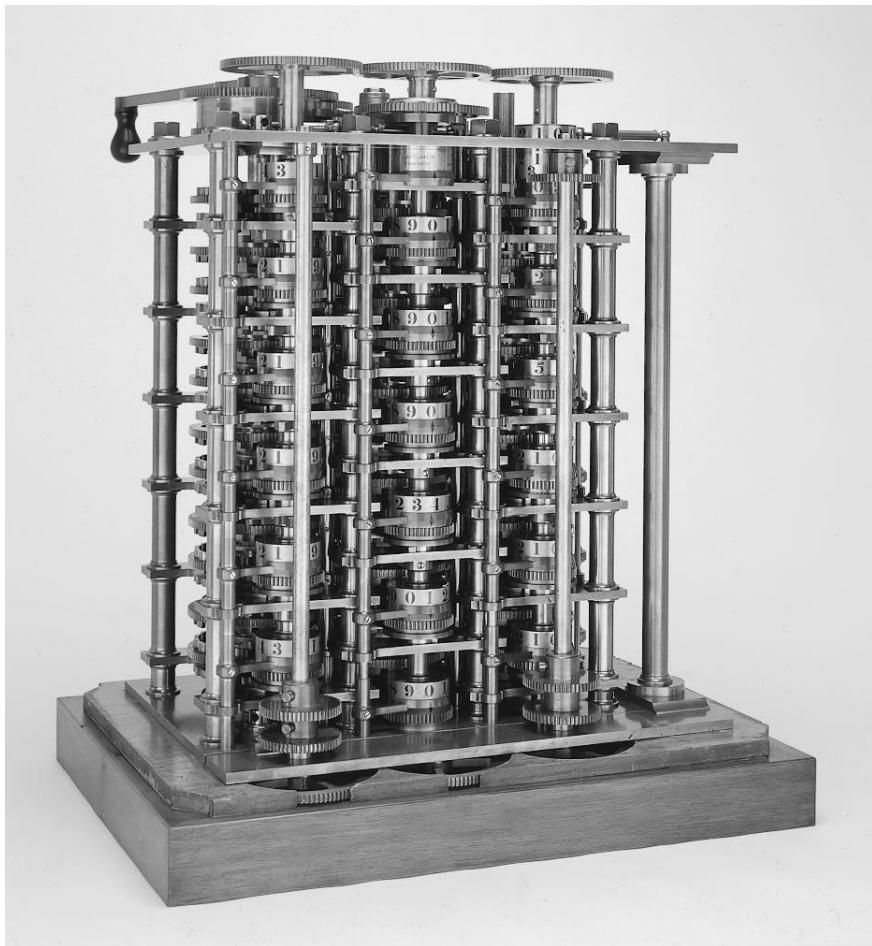




# **Digital Design Using Integrated Circuits (IC) & Very Large-Scale IC (VLSI)**



## The First Computer



**The Babbage  
Difference Engine  
(1832)**

**25,000 parts  
Cost: 17,470 Pounds  
in Year 1832**

*Use Relays as switching components*

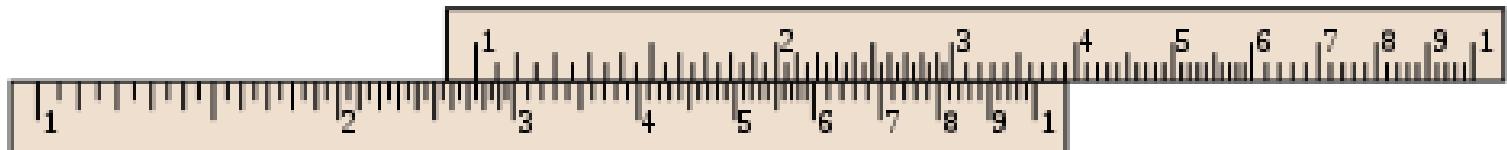


## Slide Rule



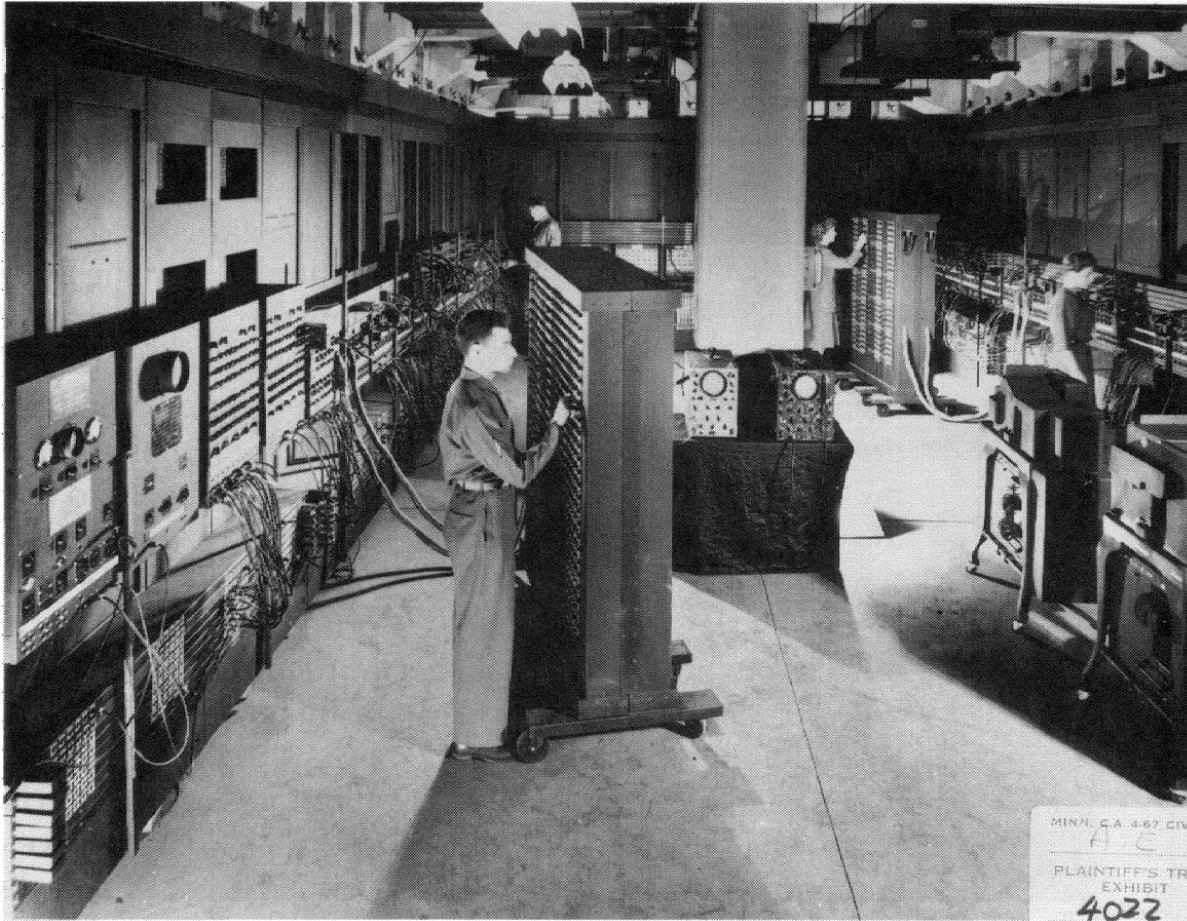
The illustration below demonstrates the computation of  $5.5/2$ . The 2 on the top scale is placed over the 5.5 on the bottom scale. The 1 on the top scale lies above the quotient, 2.75.

[http://en.wikipedia.org/wiki/Slide\\_rule](http://en.wikipedia.org/wiki/Slide_rule)





## ENIAC - The first electronic computer (1946)



*Use Vacuum Tubes as switching components*



## Technologies for Building Processors & Memories

### ❖ Vacuum tube

❖ An electronic component, predecessor of the transistor, that consists of a hollow glass tube about 5 to 10 cm long from which as much air has been removed as possible and which uses an electron beam to transfer data

### ❖ Transistor

❖ An ON/OFF switch controlled by an electric signal

### ❖ Very large scale integrated (VLSI) circuit

❖ A device containing hundreds of thousands to millions of transistors

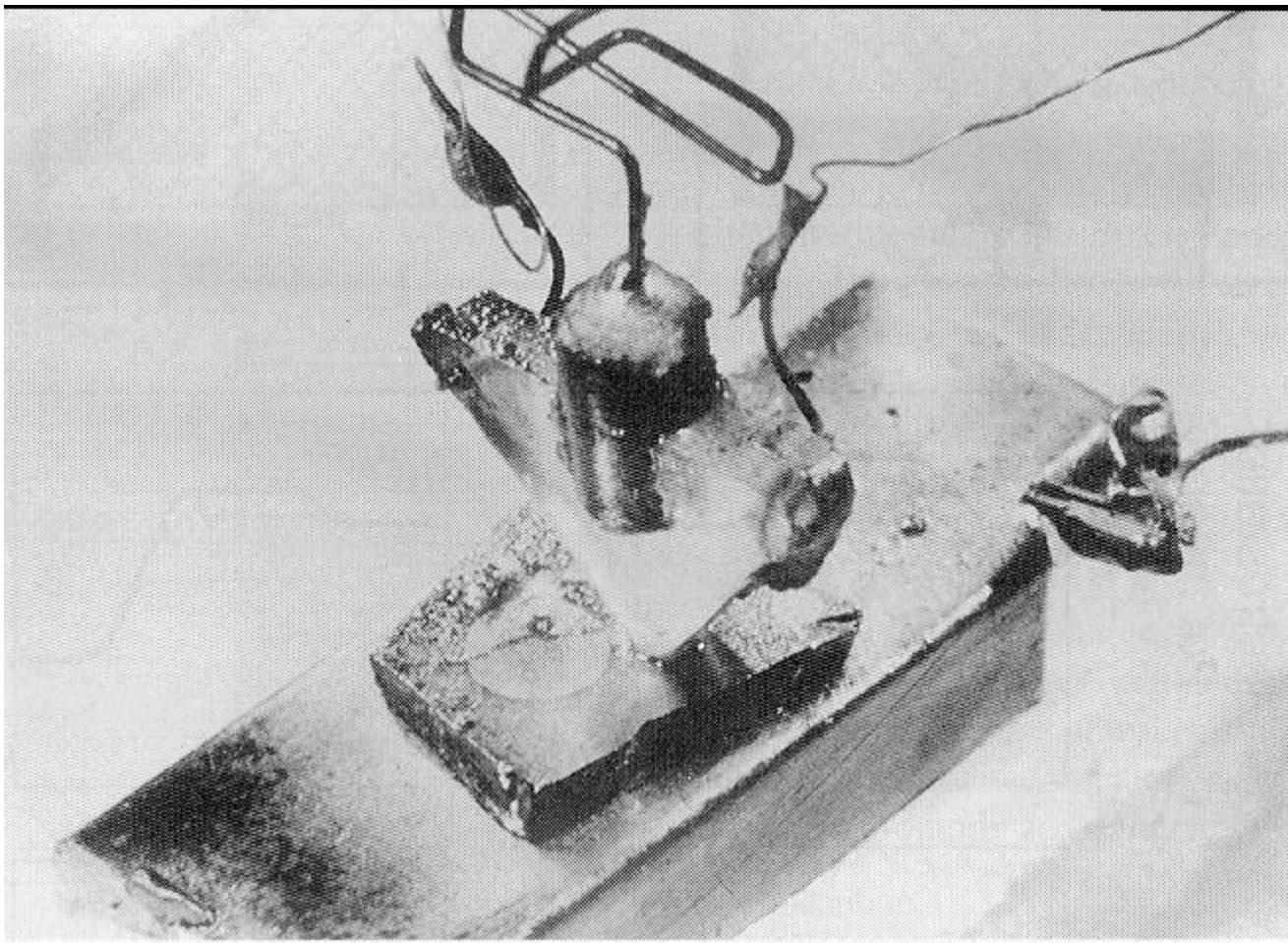


## Vacuum Tube





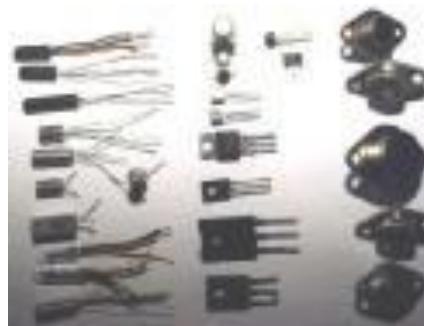
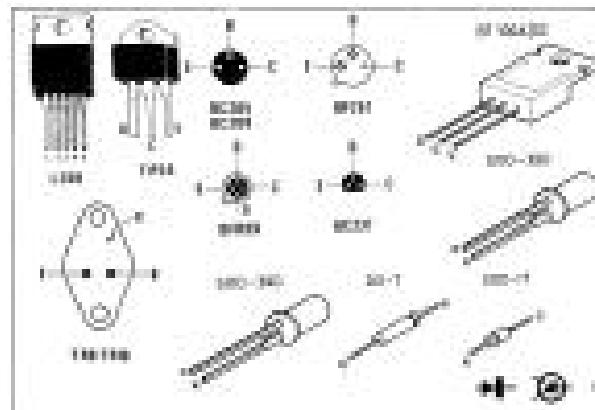
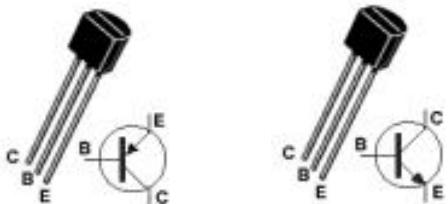
## The Transistor Revolution



First transistor  
Bell Labs, 1948

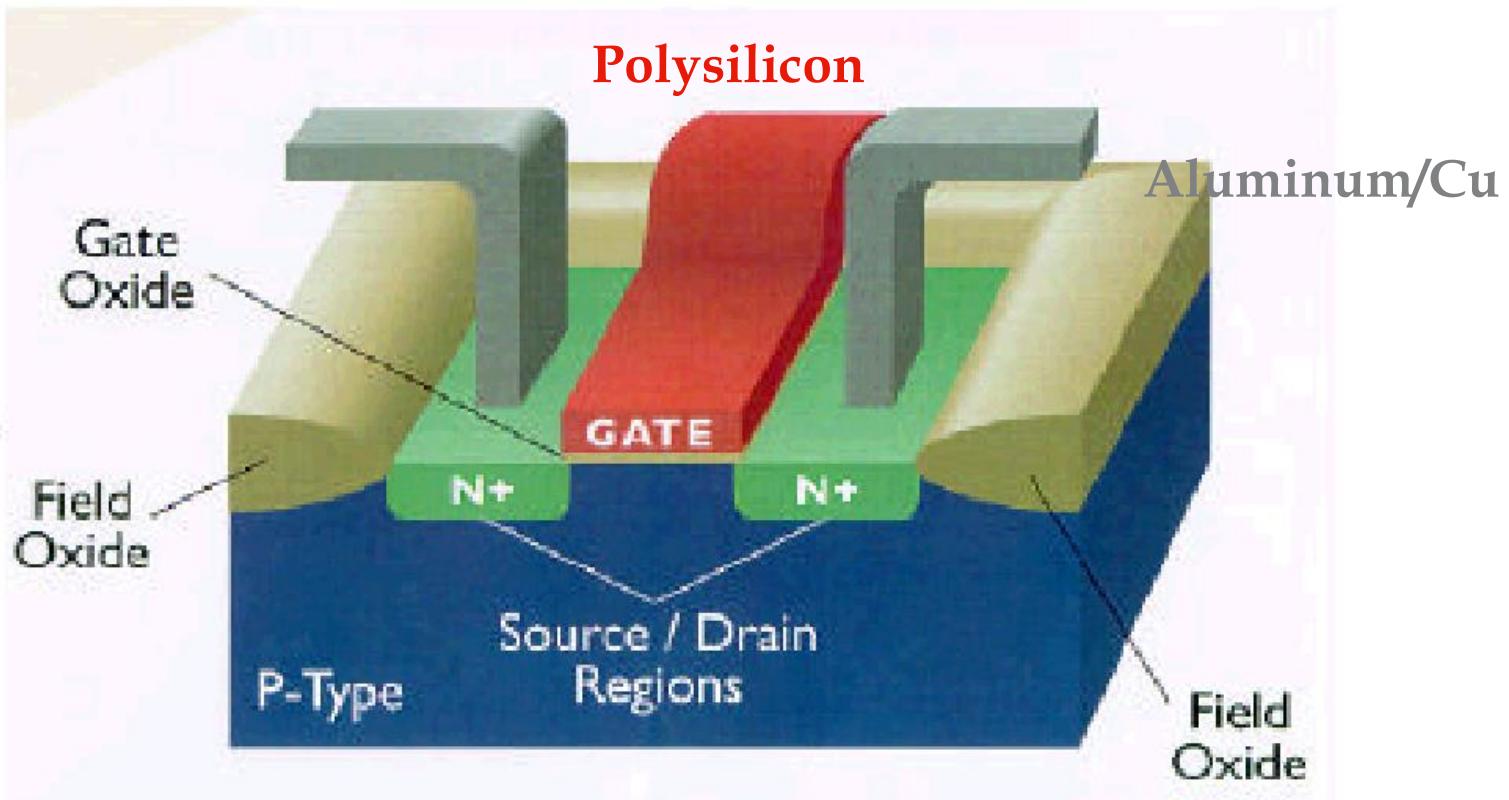


## Discrete Transistors





## The MOS Transistor



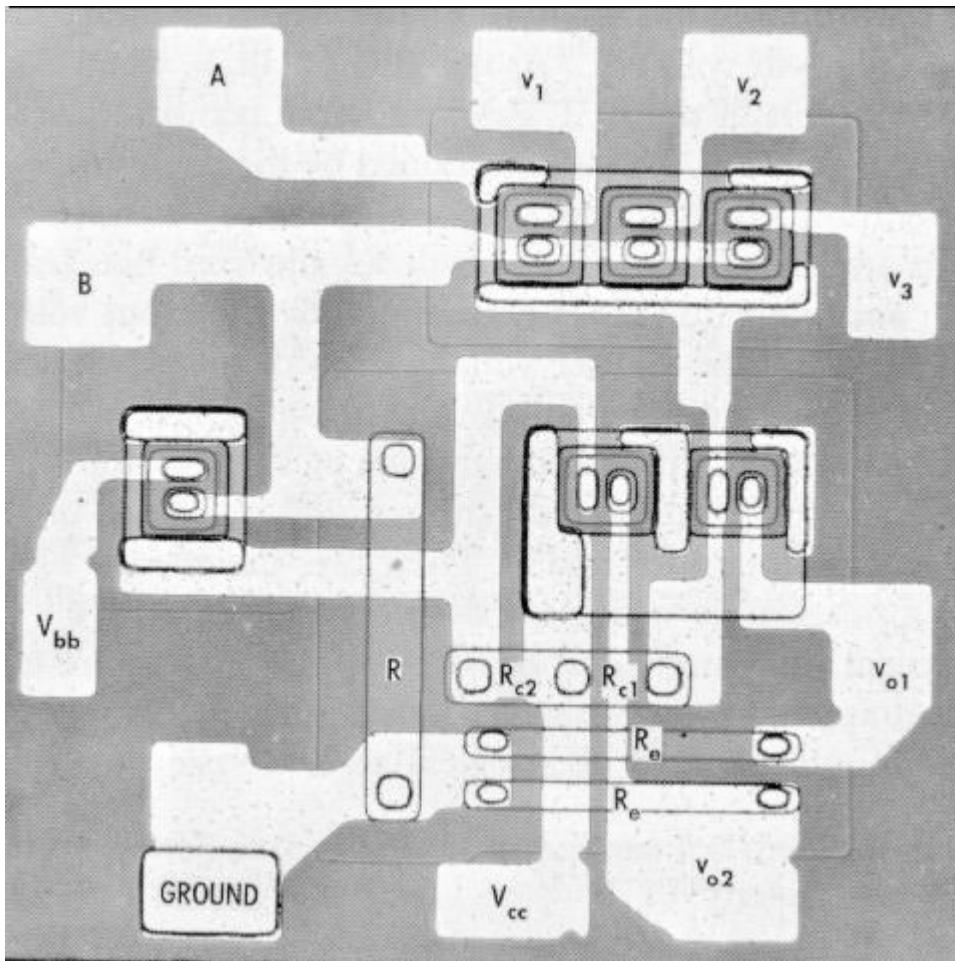
Channel length: The distance between Source and Drain

14/10nm: mainstream in this year

7-5-3 nm: next years



## The First Integrated Circuits

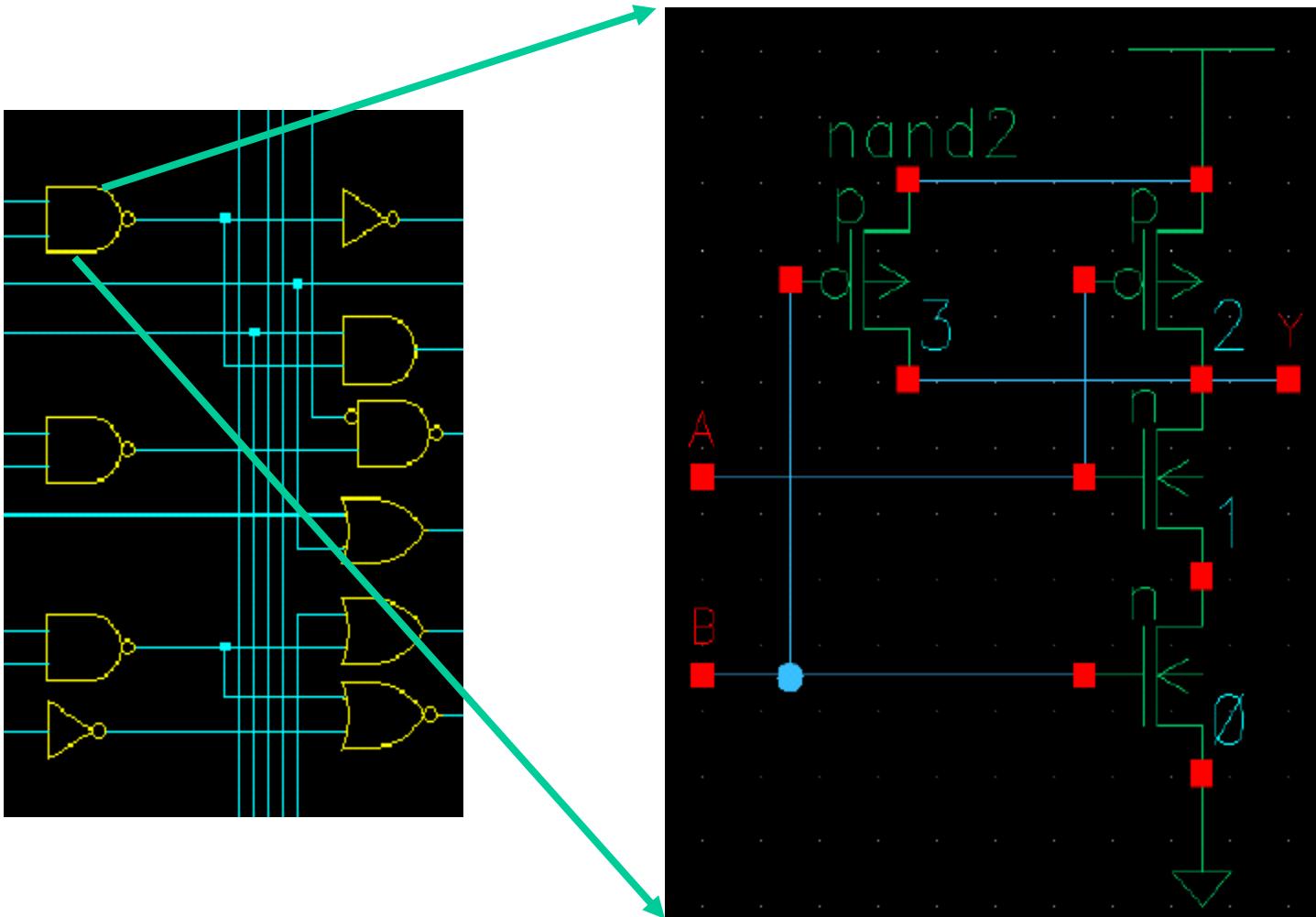


Bipolar logic  
1960's

ECL 3-input Gate  
Motorola 1966

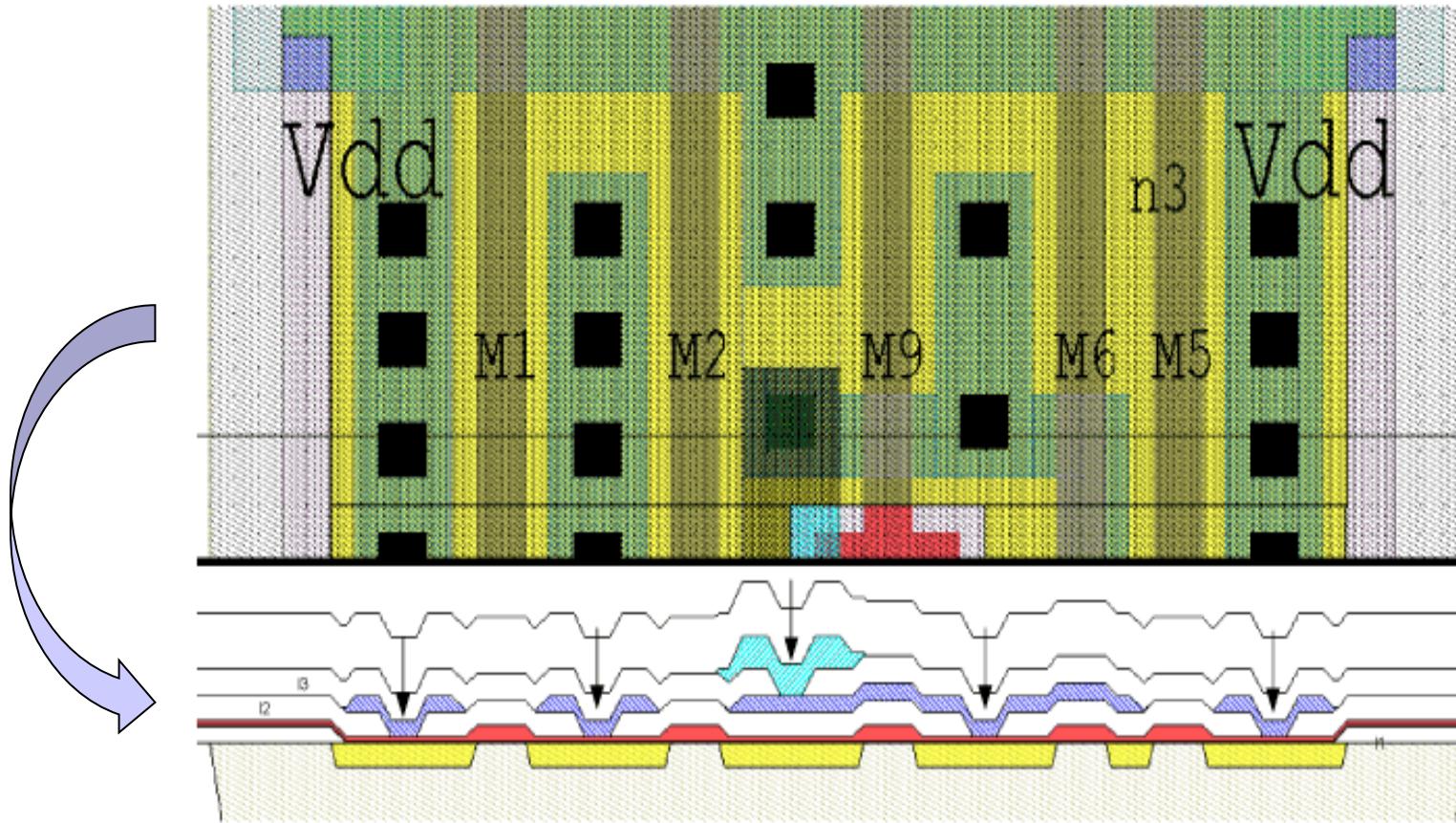


## Gate and Circuit Level Design



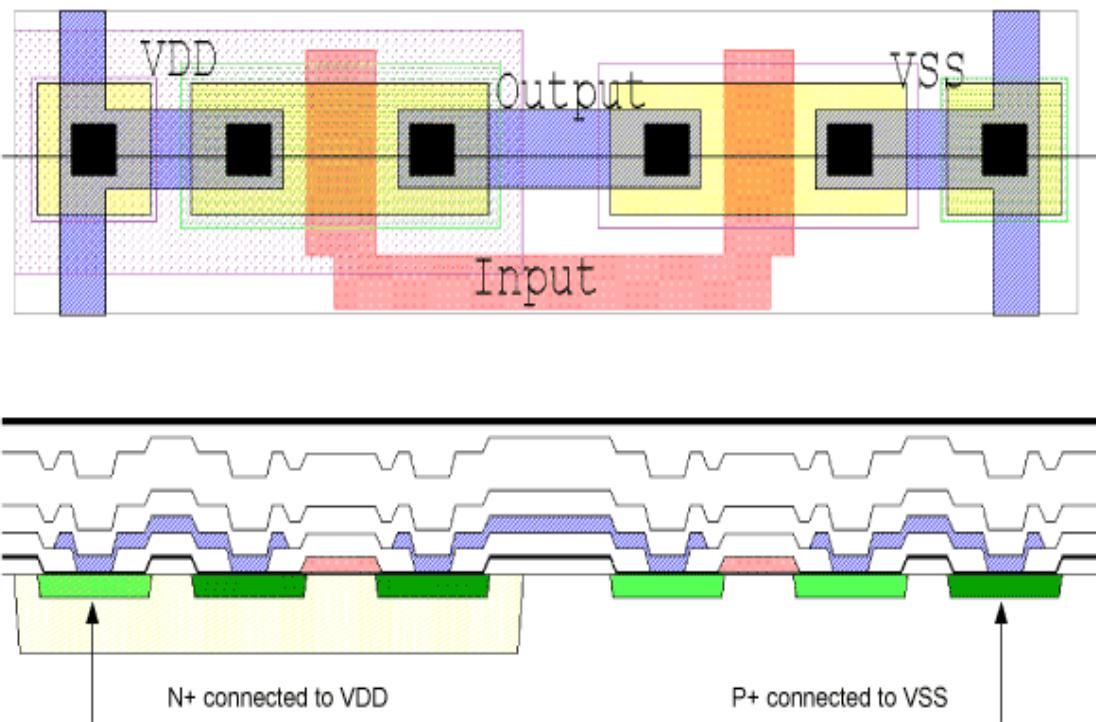
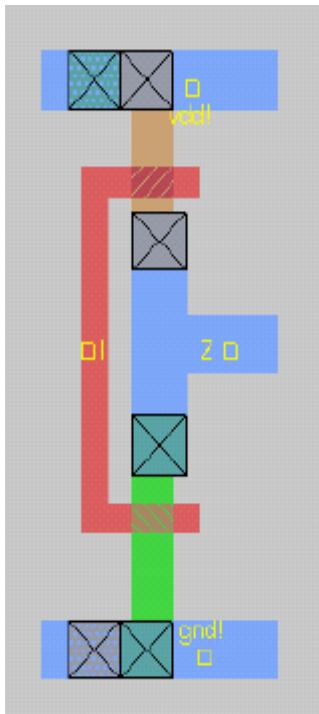


## Mapping of Layout to IC Layers



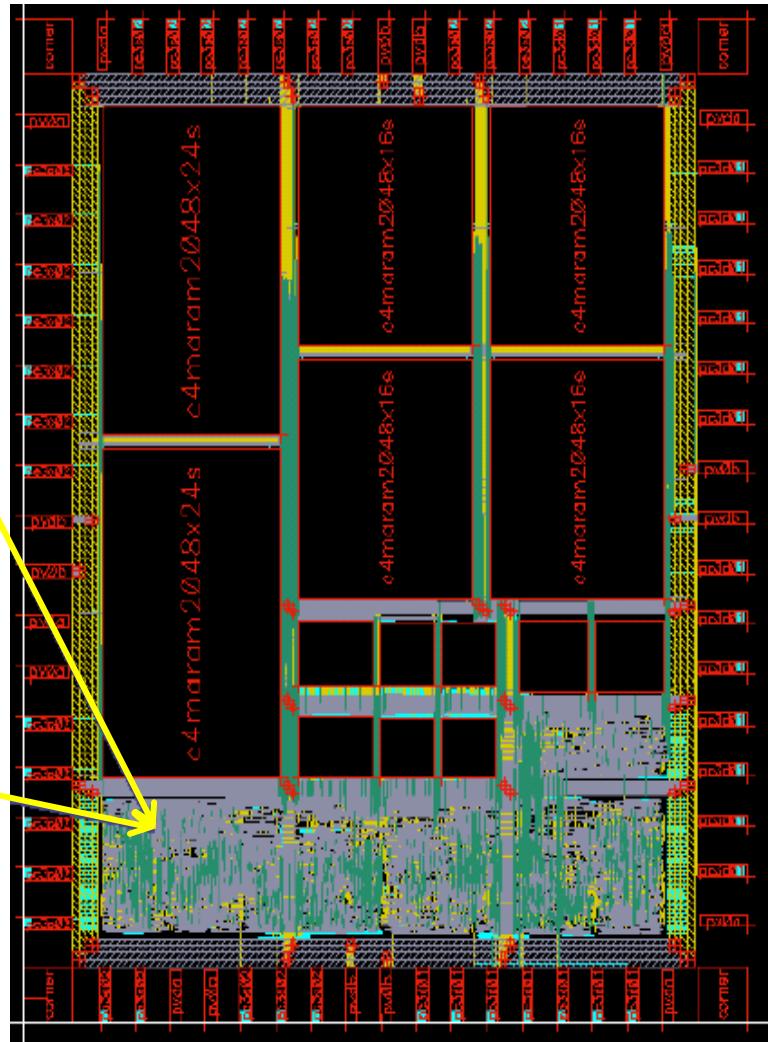
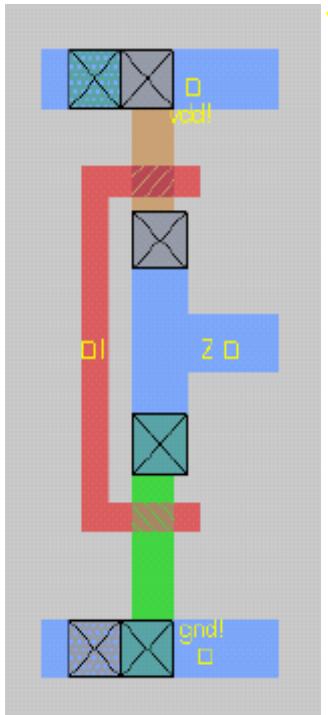


## Layout of an CMOS Inverter



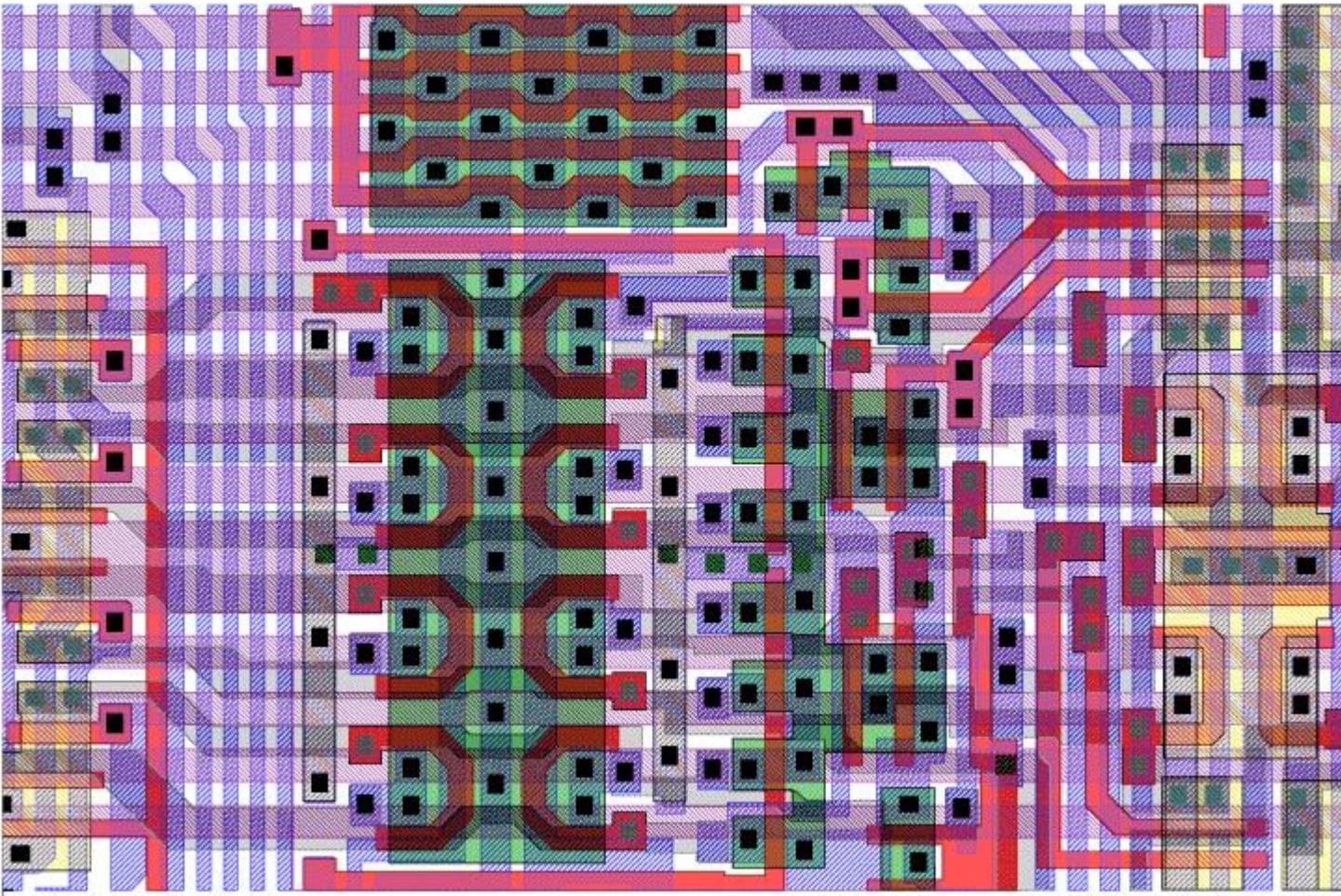


# Physical Design



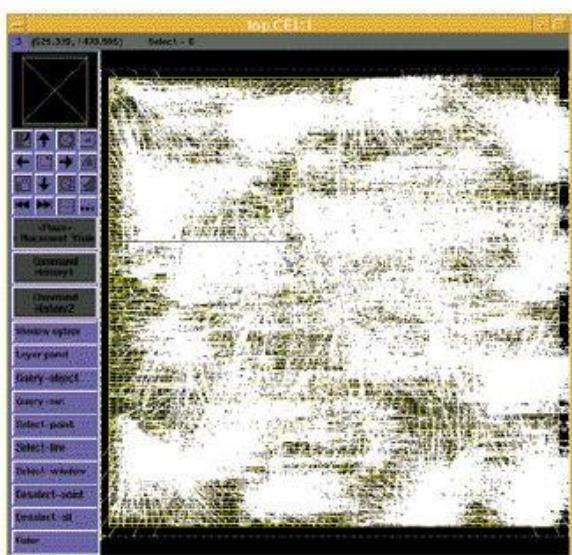


## Physical Layout of your design





# The “Timing Closure” Problem

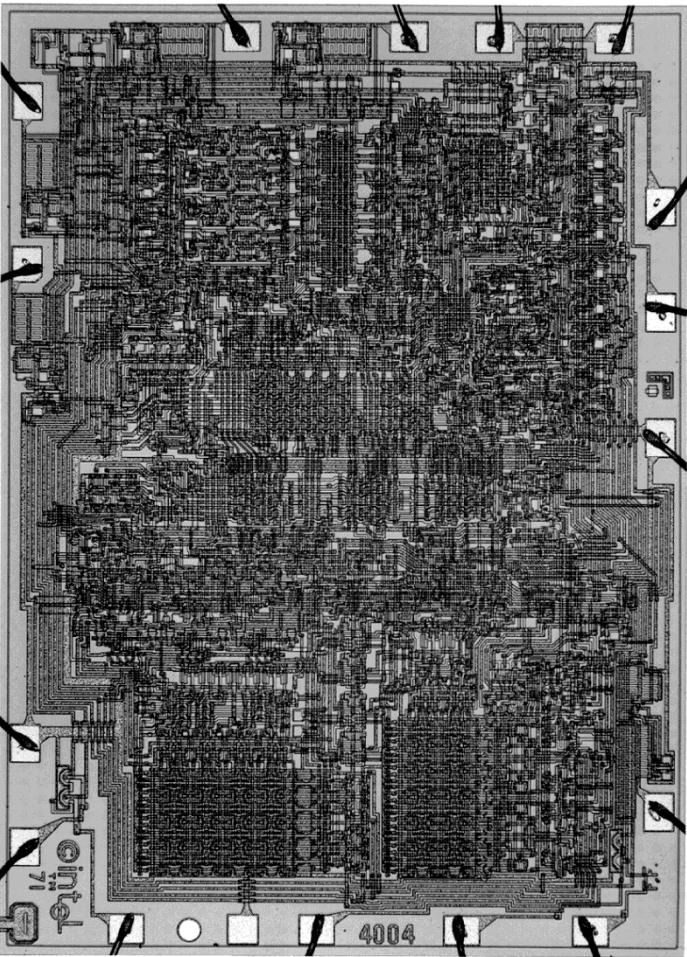


*Iterative Removal of Timing Violations (white lines)*

Courtesy Synopsys



## Example: Intel 4004 Micro-Processor



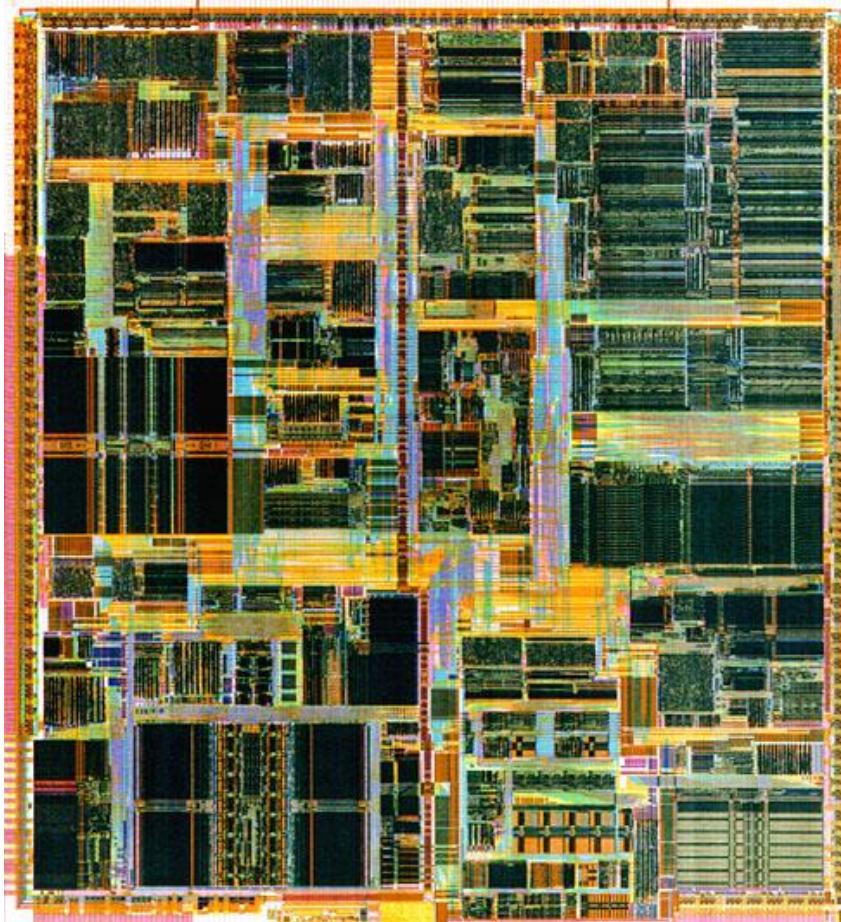
1971  
2,300 transistors  
@ 1 MHz operation

內有2,300個電晶體  
，採用五層設計、10  
微米製程

[https://zh.wikipedia.org/zh-tw/Intel\\_4004](https://zh.wikipedia.org/zh-tw/Intel_4004)



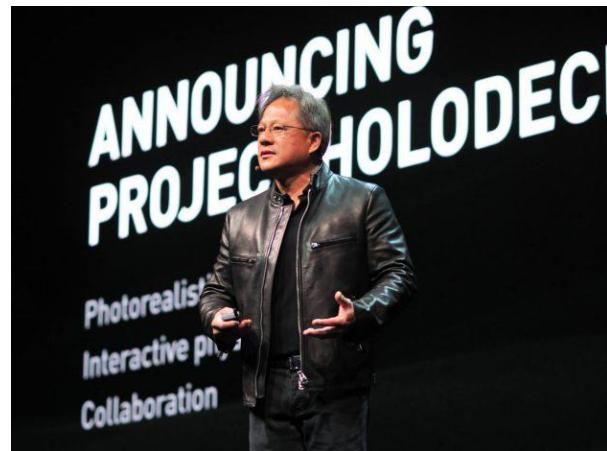
## Intel Pentium (IV) microprocessor





## nVIDIA Tesla V100 (2017-05)

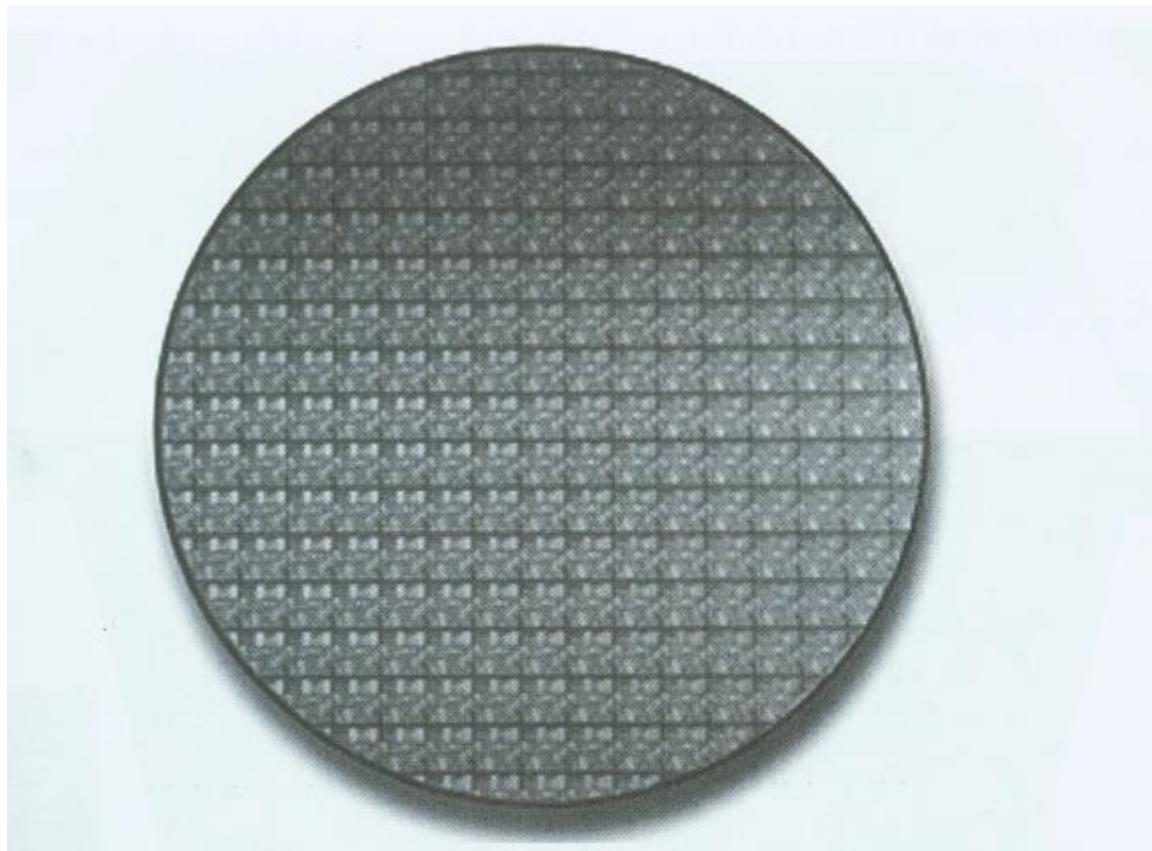
- ❖ 在週三舉行的 GTC 大會上，Nvidia 發表了基於其下一代圖形架構 Volta，針對伺服器市場的 GPU 新品 Tesla V100。該晶片擁有超過 210 億個晶體管 和 5,120 個運算核心。
- ❖ 但是對 AI 來說，最重要的是，Tesla V100 配備了 640 個 Tensor 核心，是專為運行深入學習網路中使用的數學運算設計的。據官方介紹，Tensor 內核為 Tesla V100 提供了高達 120 teraflops、驚人的深度學習能力。



[https://technews.tw/2017/05/12/  
nvidia-give-tesla-v100-to-  
google/](https://technews.tw/2017/05/12/nvidia-give-tesla-v100-to-google/)



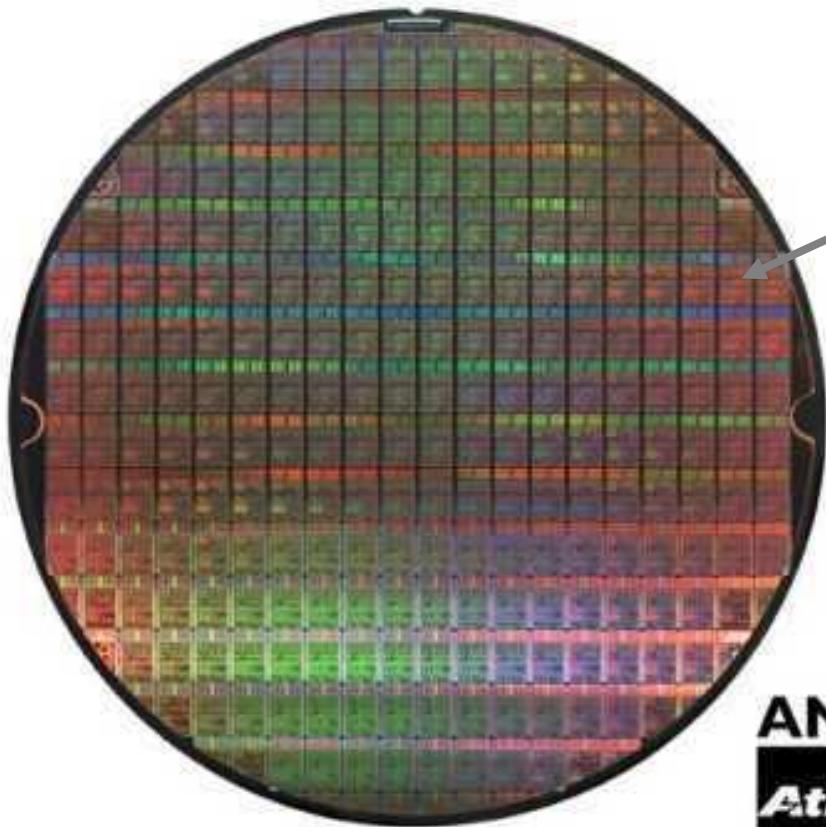
## 8-inch Wafer



An 8-inch (200-mm) diameter wafer containing Intel Pentium 4 processors



## Die Cost



Single die  
Wafer

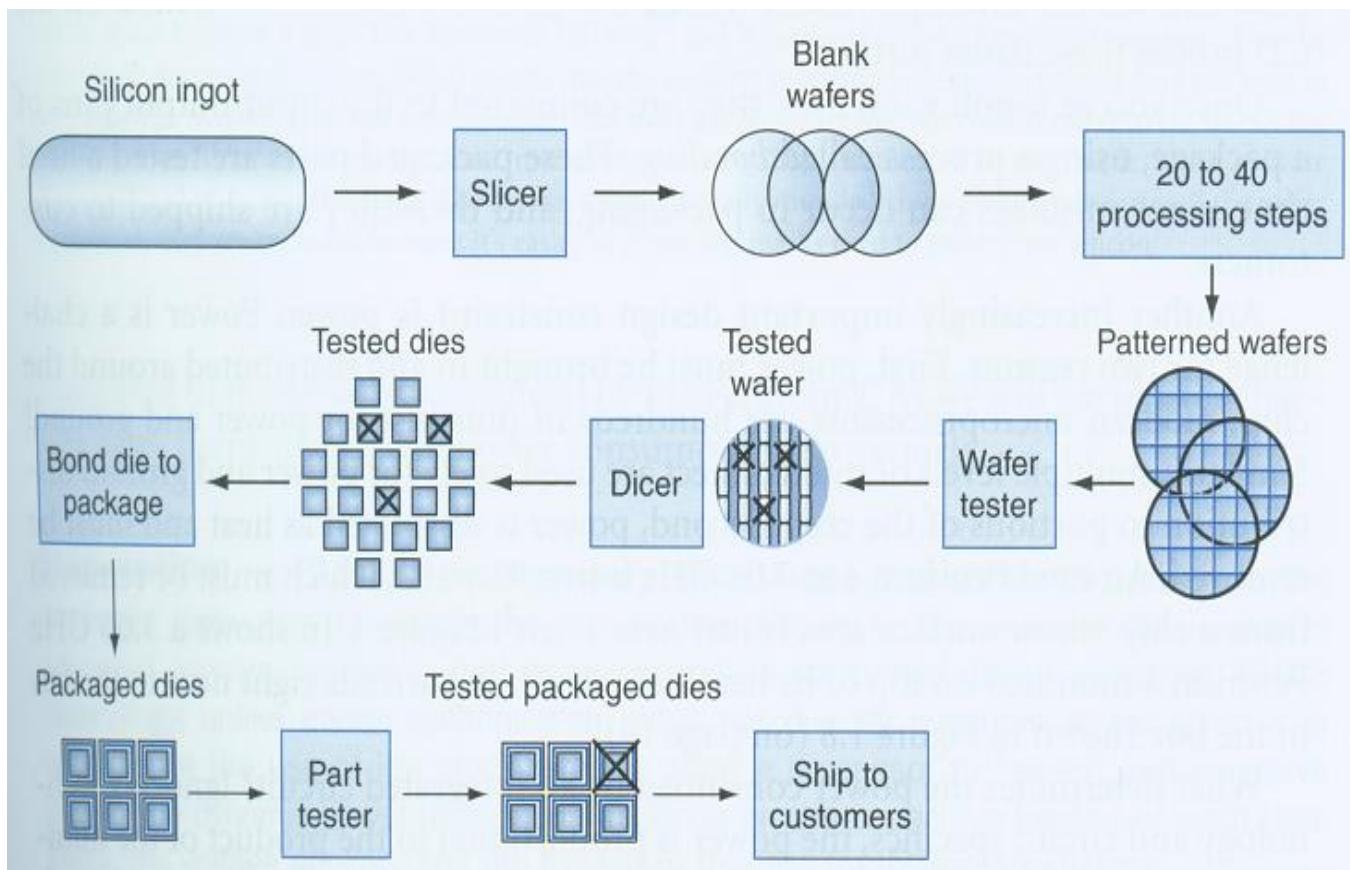


Going up to 12" (30cm)

From <http://www.amd.com>

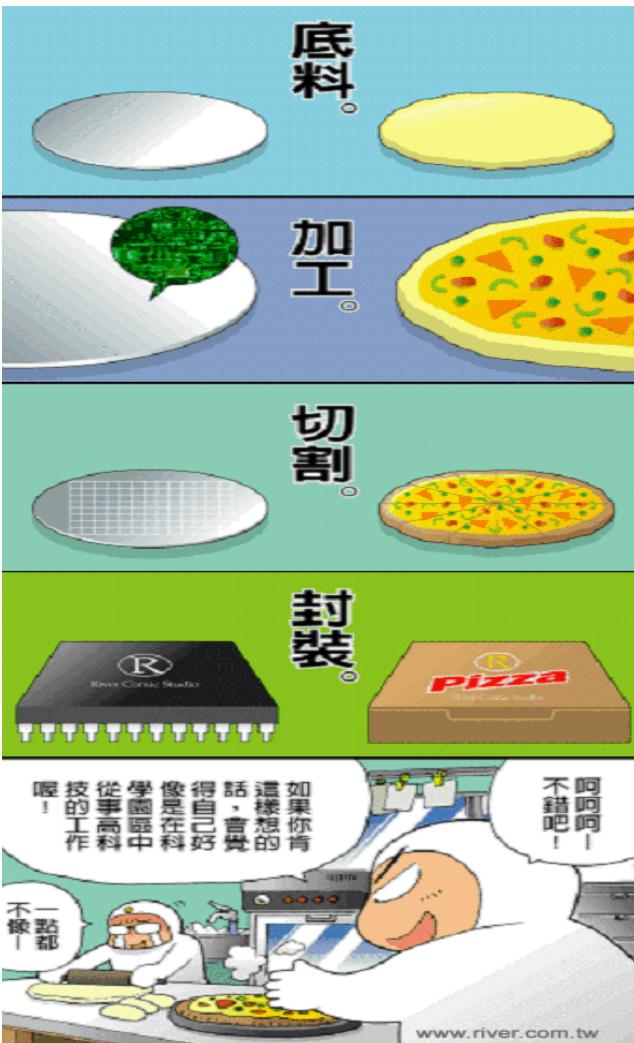


# The Chip Manufacturing Process





## Process v.s. Pizza Making





## Technologies for Building Processors & Memories

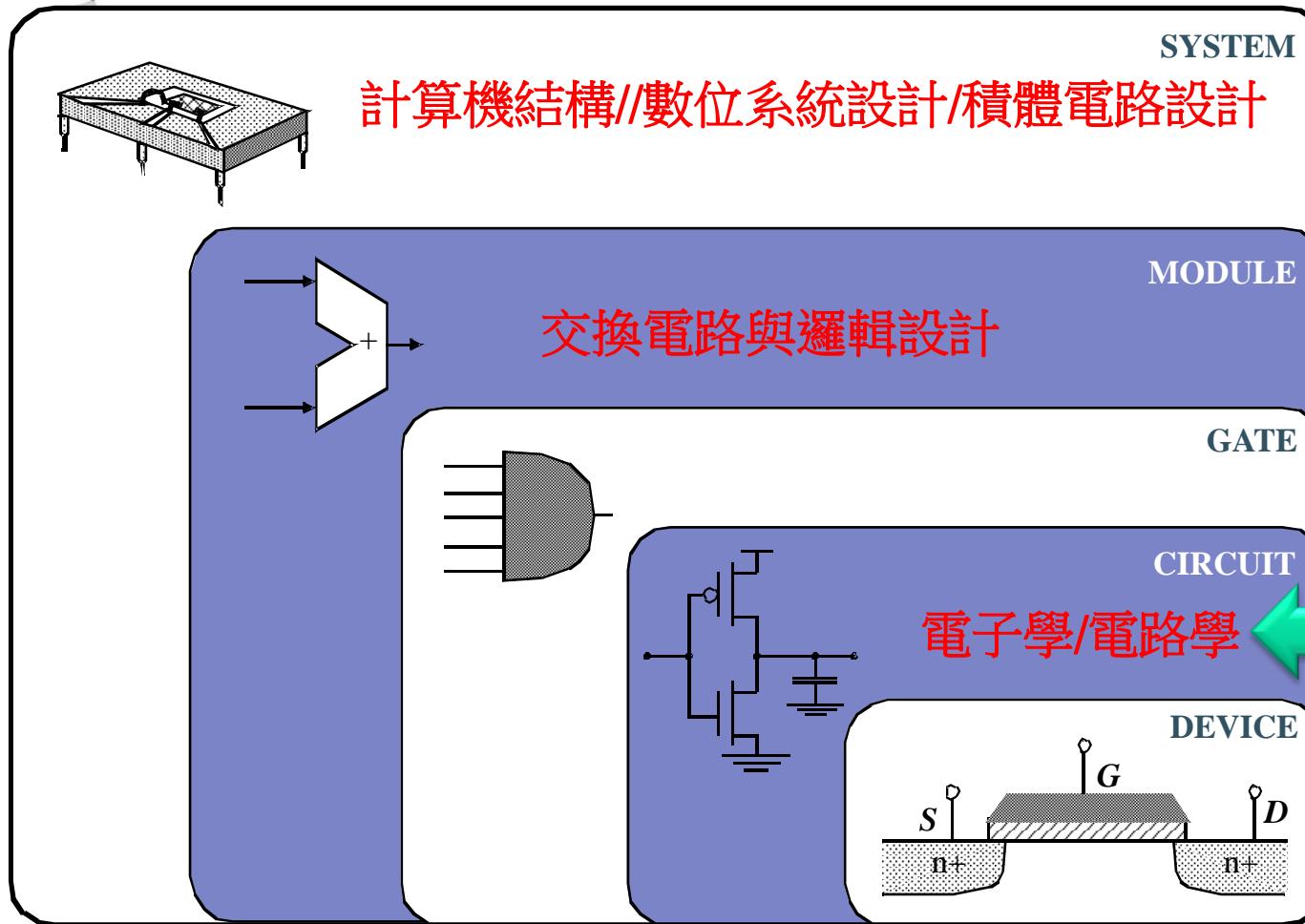
Year	Technology used in computers	Relative performance/unit cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit	900
1995	Very large scale integrated circuit	2,400,000
2005	Ultra large scale integrated circuit	6,200,000,000

Relative performance per unit cost of technologies used in computers over time



數電實驗

## Design Abstraction Levels



電子/電路  
實驗



# Moore's Law v.s. HDL

*Issue : Design Productivity*



## Review of Full-custom Analog Design Flow



## Schematic of 741 Op-amp Circuits

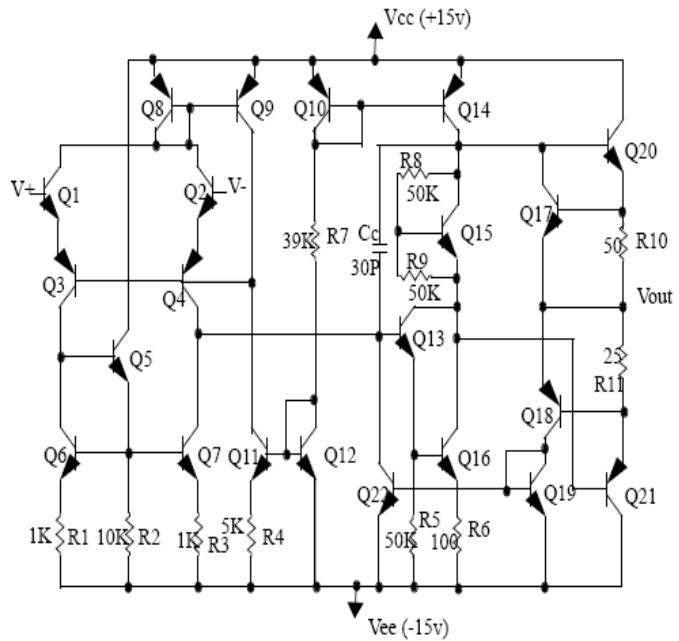
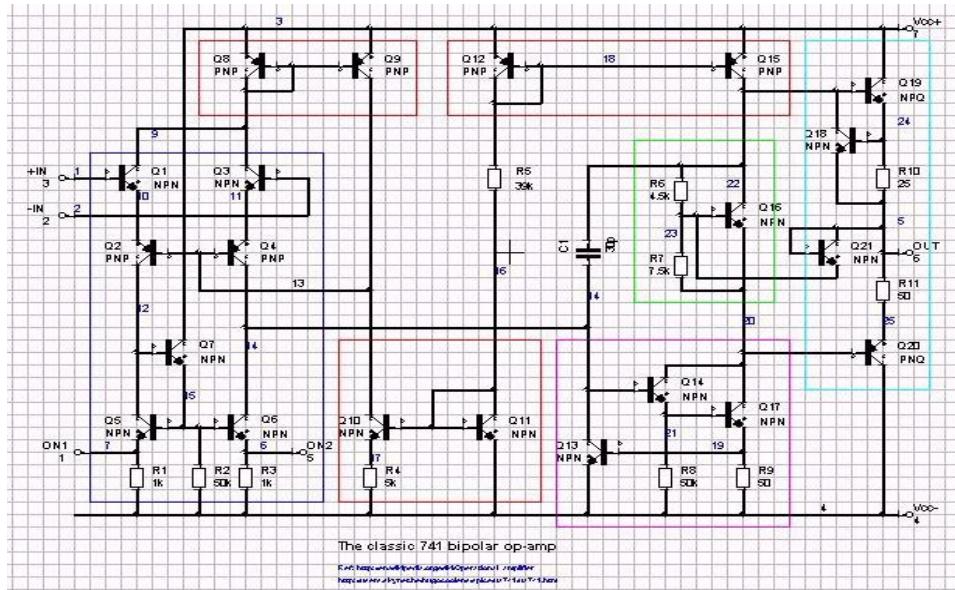


Fig. 1(a) Transistor level circuit diagram for op741

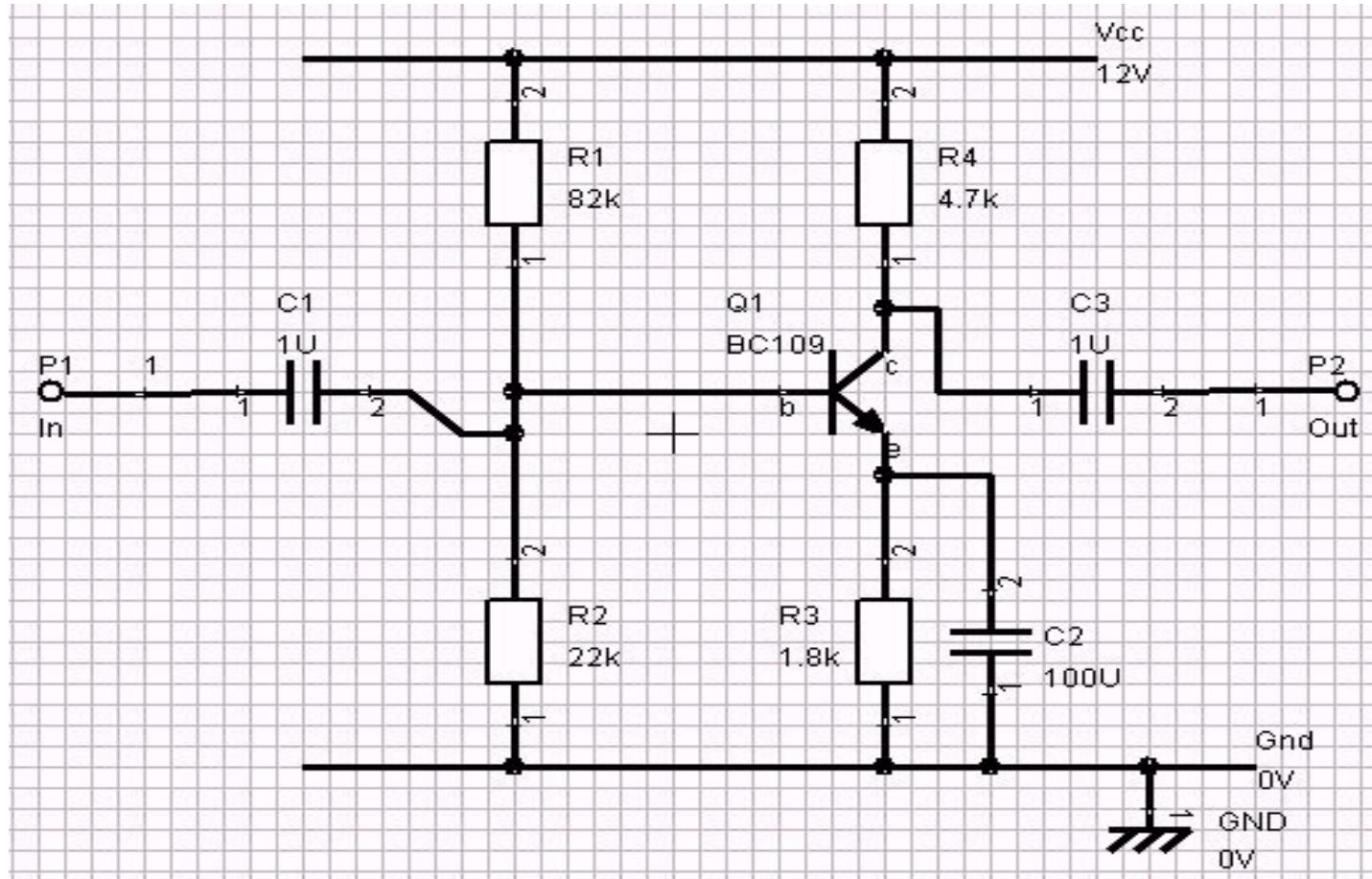


Schematic  
(Transistor level  
circuit diagram)  
for op741

Editing In EDA tools  
(on workstation or your computer)



## SPICE (I)





## SPICE (II) – Manual programming or Export from Schematic Designs

- ❖ When you export the SPICE netlist, you get the following file:

\*SPICE BJT Amplifier

C1 5 1 1U

C3 2 4 1U

C2 0 6 100U

Q1 2 1 6 ZTX109 1.0

R1 1 3 82k

R4 2 3 4.7k

R2 0 1 22k

R3 0 6 1.8k

C4 0 3 100u

.MODEL ZTX109 NPN IS=1.8E-14 ISE=5.0E-14 NF=.9955 BF=400

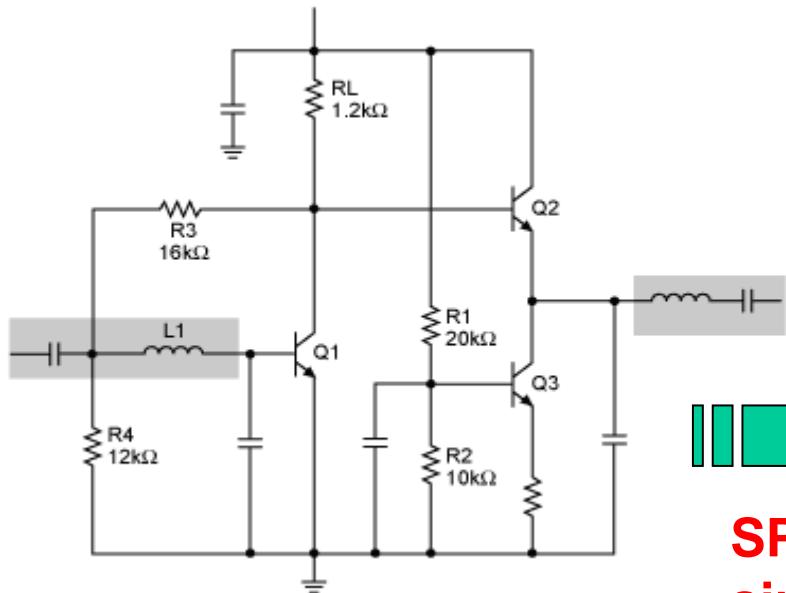
BR=35.5 +IKF=.14 IKR=.03 ISC=1.72E-13 NC=1.27 NR=1.005 RB=.56

RE=.6 RC=.25 +VAF=80 VAR=12.5 CJE=13E-12 TF=.64E-9 CJC=4E-12

TR=50.72E-9 MJC=.33 .END



## Schematic Inputs and Simulations of A two-stage 1.9GHz monolithic low-noise amplifier (LNA)

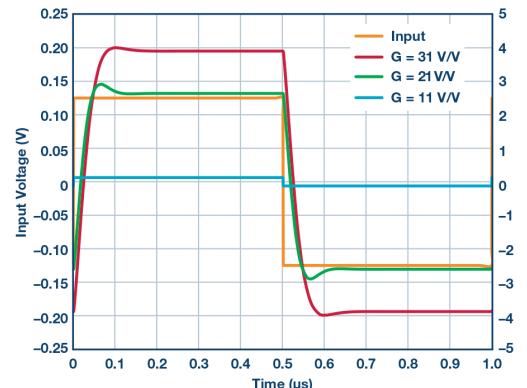


Time domain

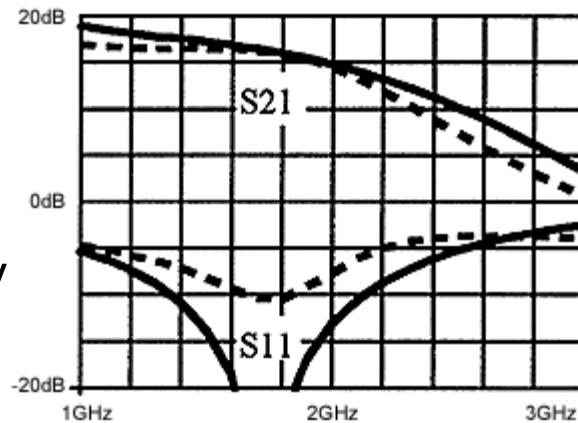


SPICE simulation

Frequency domain



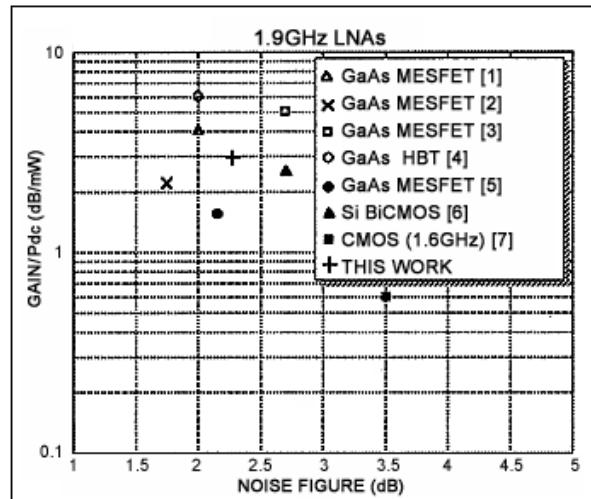
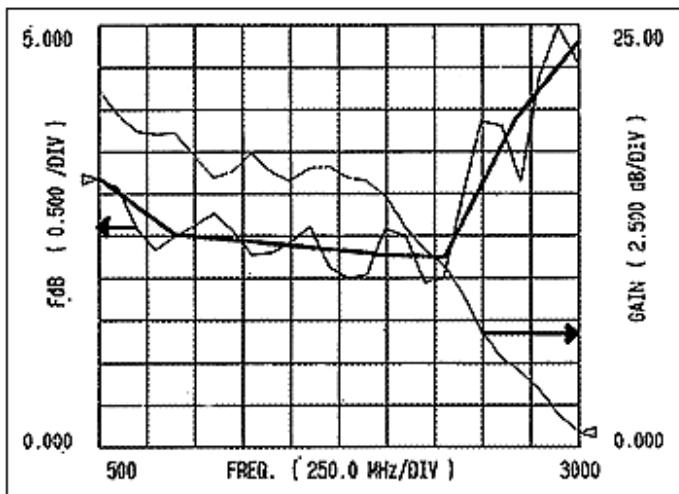
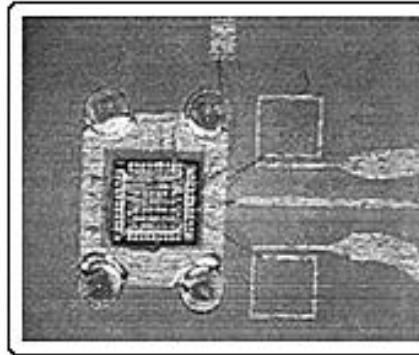
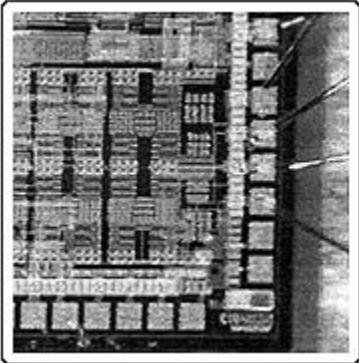
— LINEAR SIMULATION  
- - - MEASUREMENT



Source: <http://www.elecfans.com/article/84/148/2008/2008091712654.html>



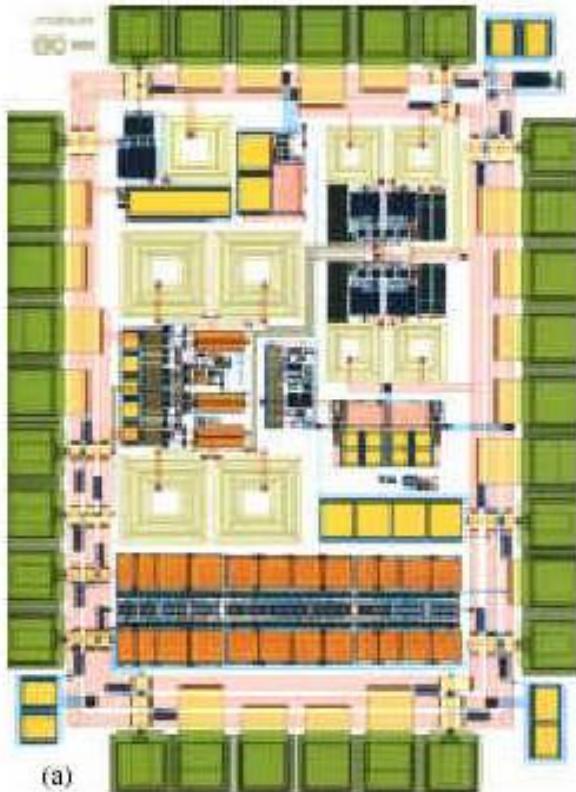
## Fabrication of Chips and Measurement Results



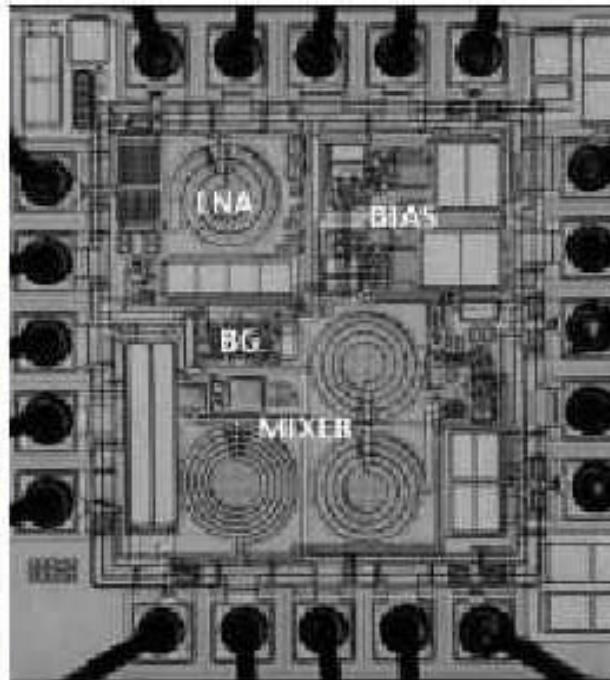
Source: <http://www.elecfans.com/article/84/148/2008/2008091712654.html>



## 7. 2-GHz Single-Chip Radio Developed at Stanford.



Layout view (generated by EDA tools)



Die Photo (chip view) for measurement

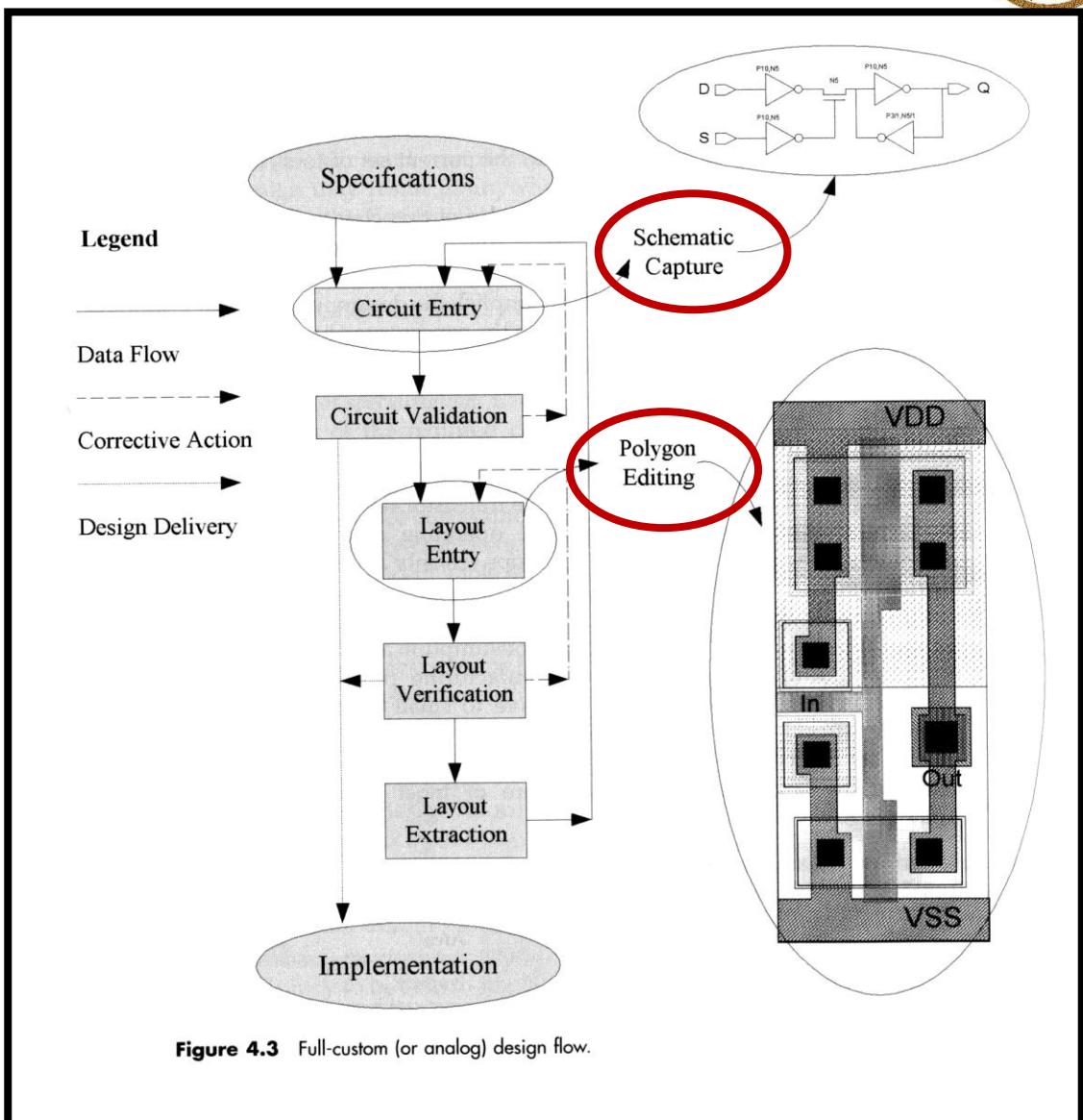


## Summary:

### Full-custom Analog Design Flow



**Low Design  
Productivity**





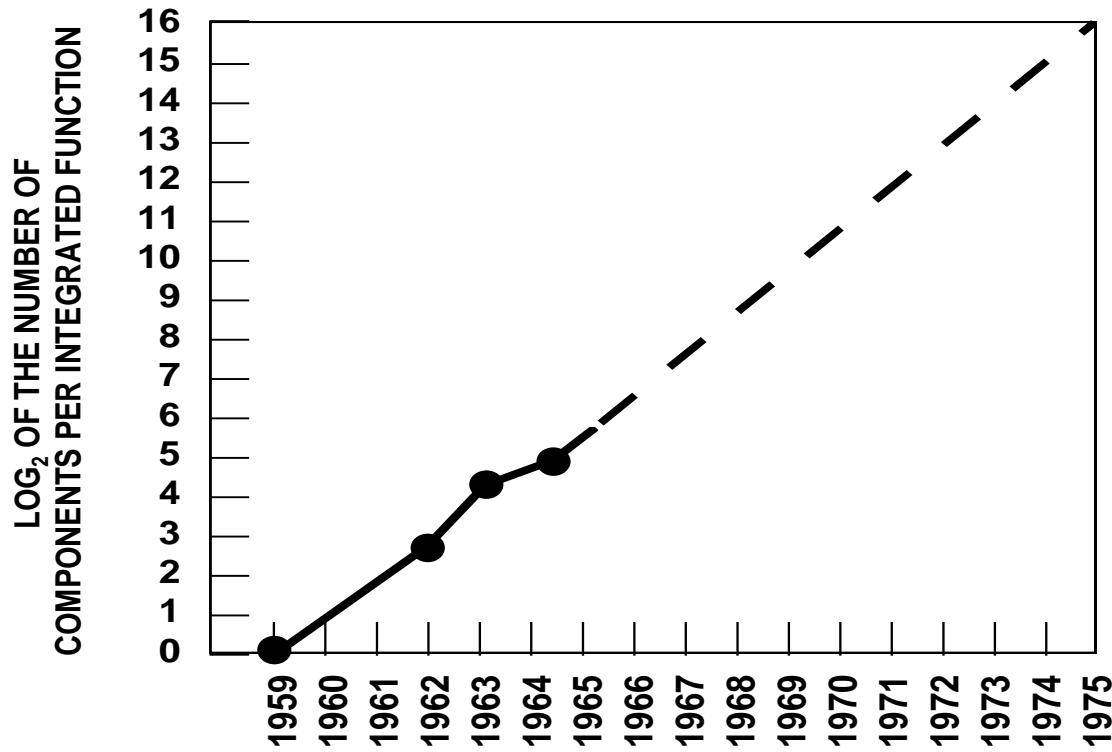
## Moore's Law

In 1965, *Gordon Moore* noted that the number of transistors on a chip doubled every 18 to 24 months.

He made a prediction that semiconductor technology will **double** its effectiveness **every 18~24 months**



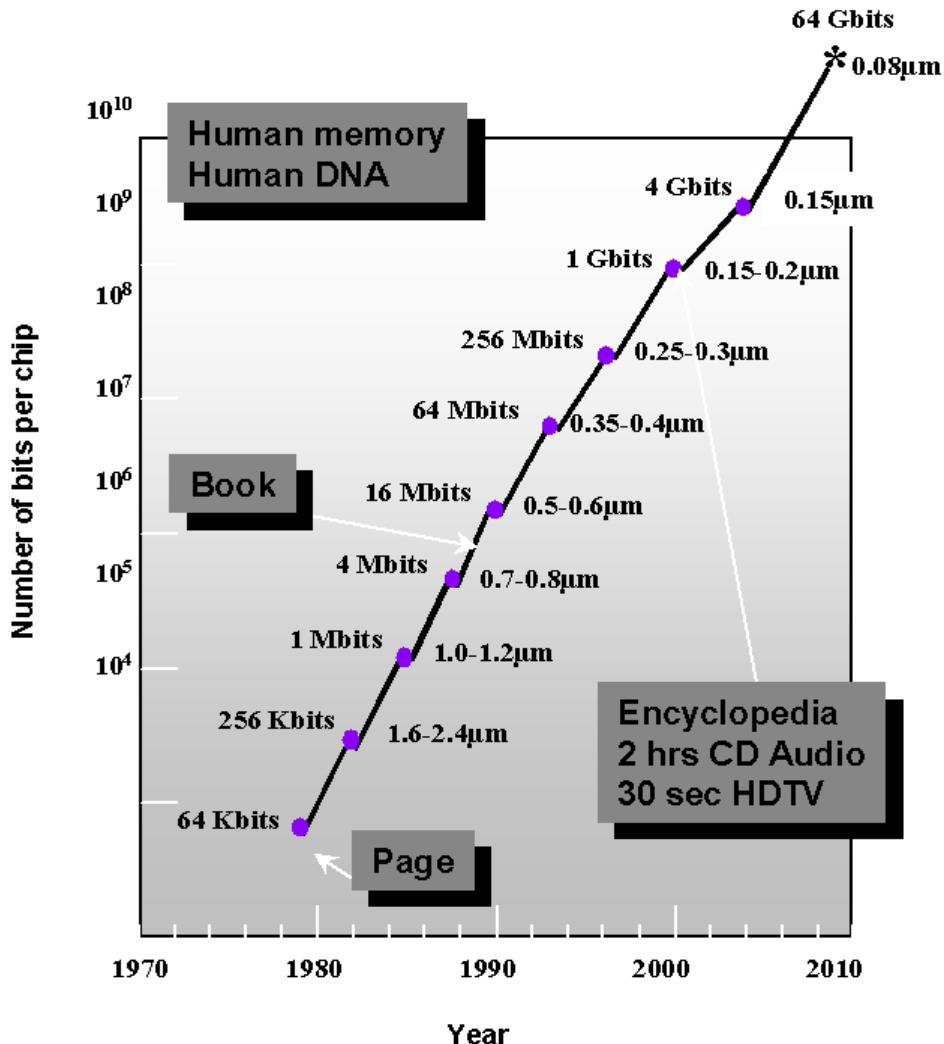
## Moore's Law



*Electronics*, April 19, 1965.



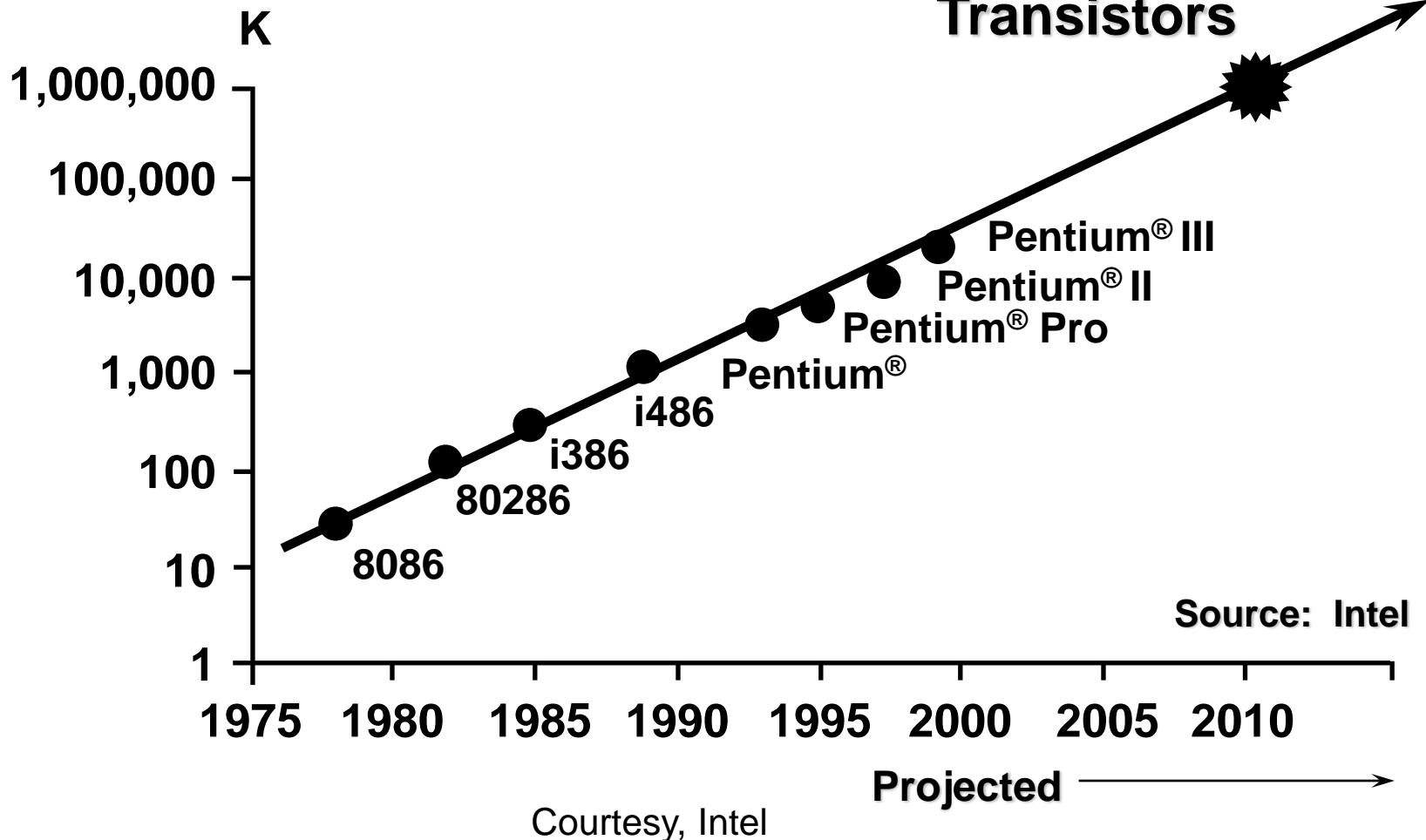
## Evolution in Complexity





## Transistor Counts

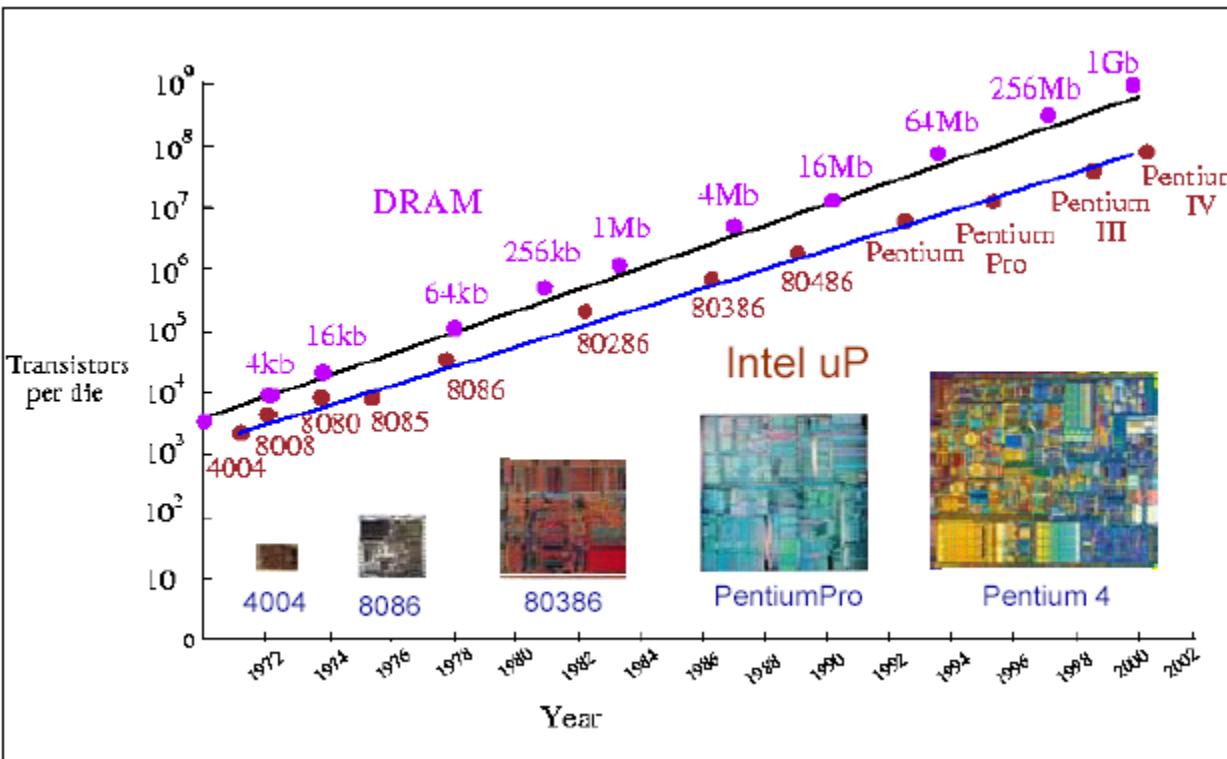
500M~ 1Billion  
Transistors





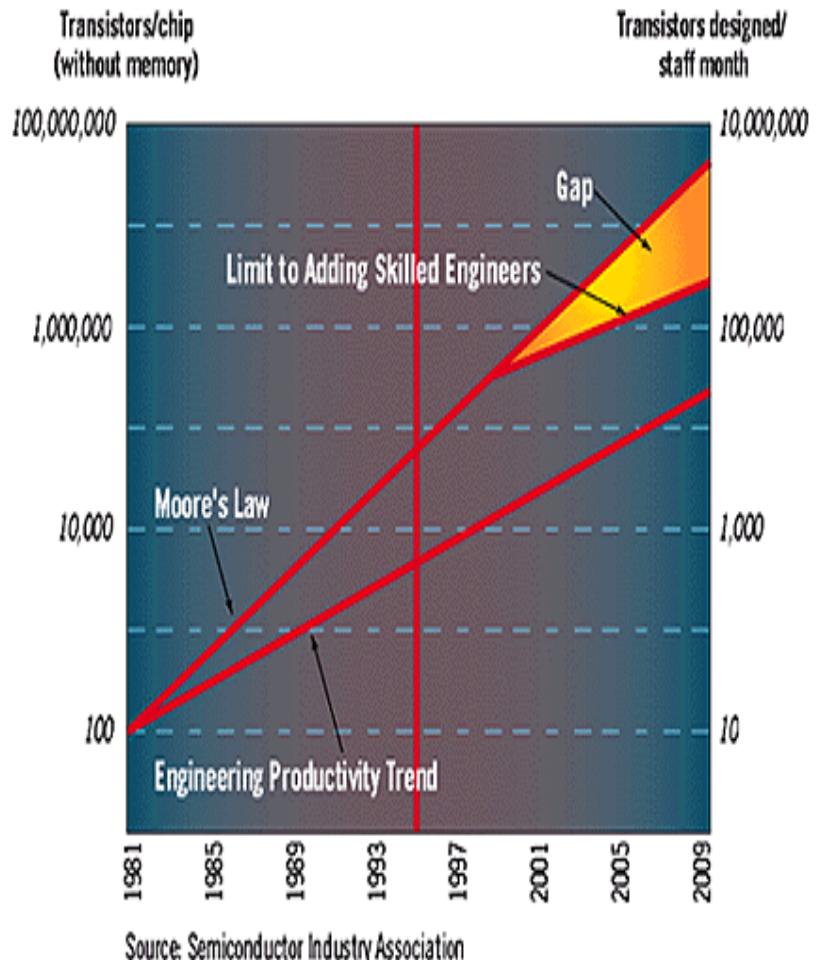
## Moore's Law: Driving Technology Advances

- ❖ Logic capacity doubles per IC at regular intervals (1965).
- ❖ Logic capacity doubles per IC every 18 months (1975).





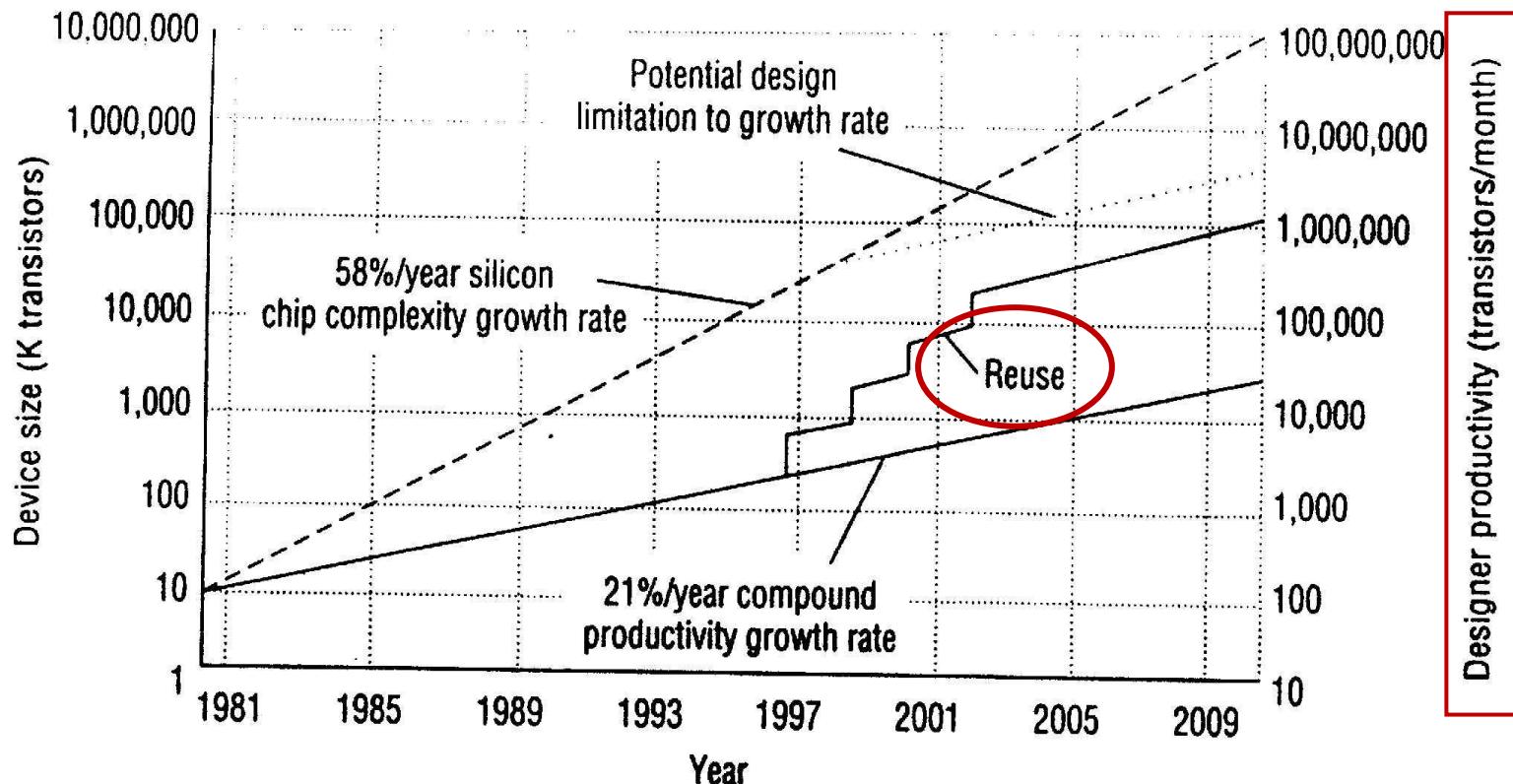
## Engineering Productivity Gap



- Engineering productivity has not been keeping up with silicon gate capacity for several years.
  
- Companies have been using larger design teams, making engineers work longer hours, etc., but clearly the limit is being reached.



## Why Must HDL Tools & IP Reuse?



**Design productivity crisis:**  
Divergence of potential design complexity  
and designer productivity



## HDL and Moore's Law

- ❖ HDL – Hardware Description Language
- ❖ Why use an HDL ?
  - ❖ Unify design entries (for different designs).
  - ❖ Easy for **synthesis**:
    - Hardware is becoming very difficult (and too big!) to design directly
    - HDL is easier and cheaper to explore different design options
    - Reduce time and cost to verify your digital designs in VLSI implementations



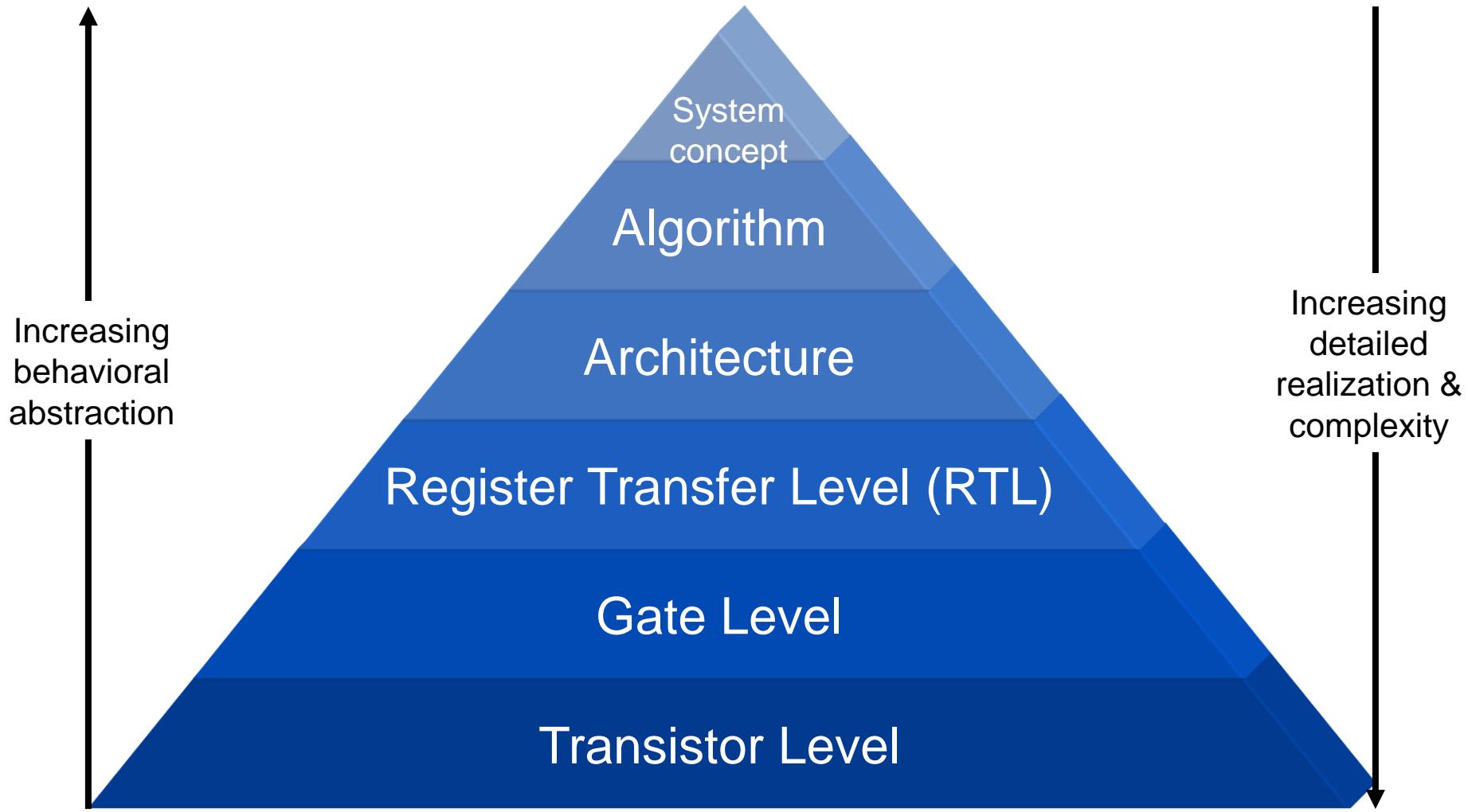
## Verilog HDL

### ❖ Feature

- ❖ HDL has **high-level programming language** constructs to describe the connectivity of your circuit.
- ❖ Ability to mix **different levels** of abstraction freely
- ❖ One language for all aspects of **design, test, and verification**
- ❖ Functionality as well as **timing**
- ❖ **Concurrency** perform target functions
- ❖ Support **timing simulation** for your design

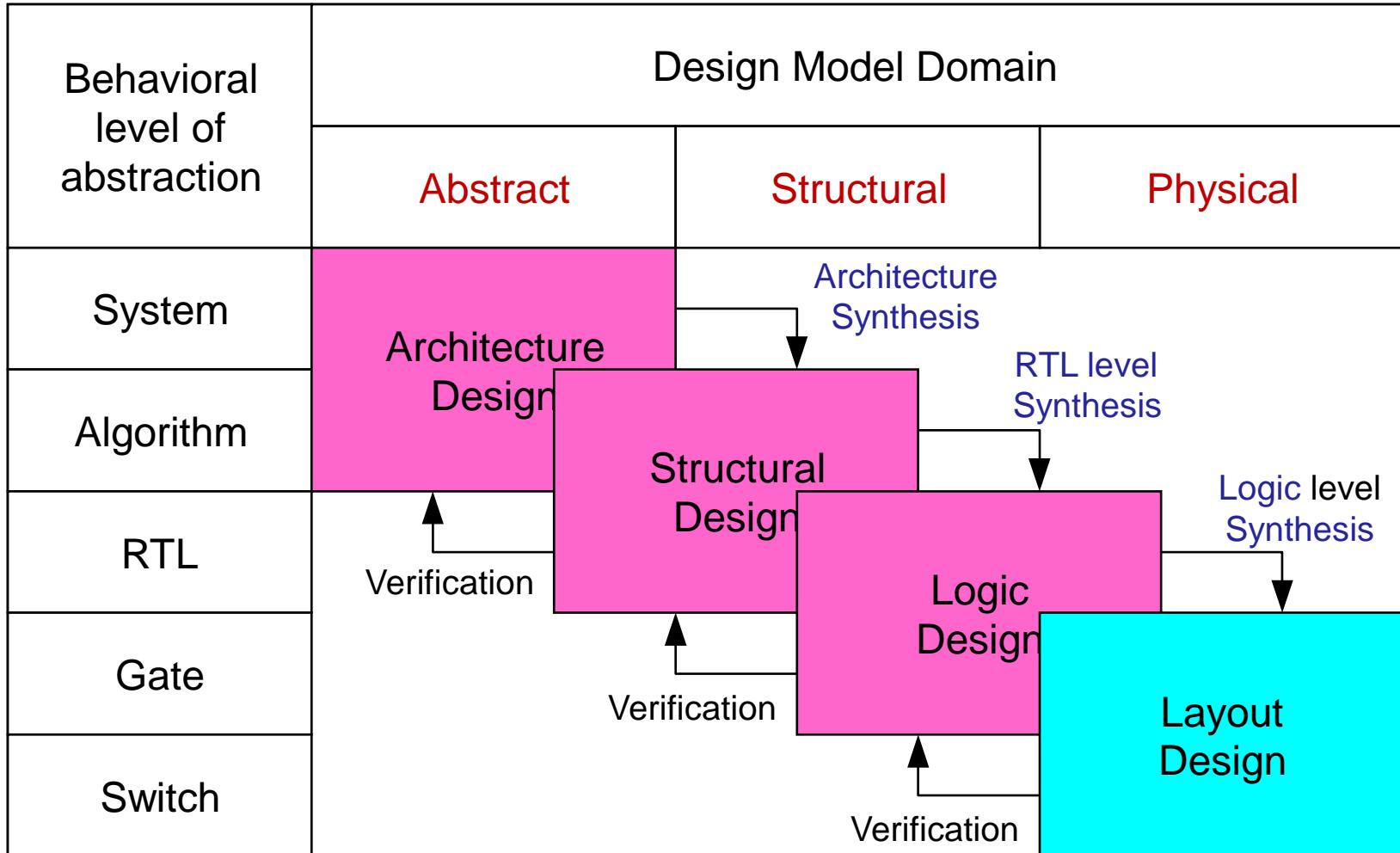


## Behavioral Model



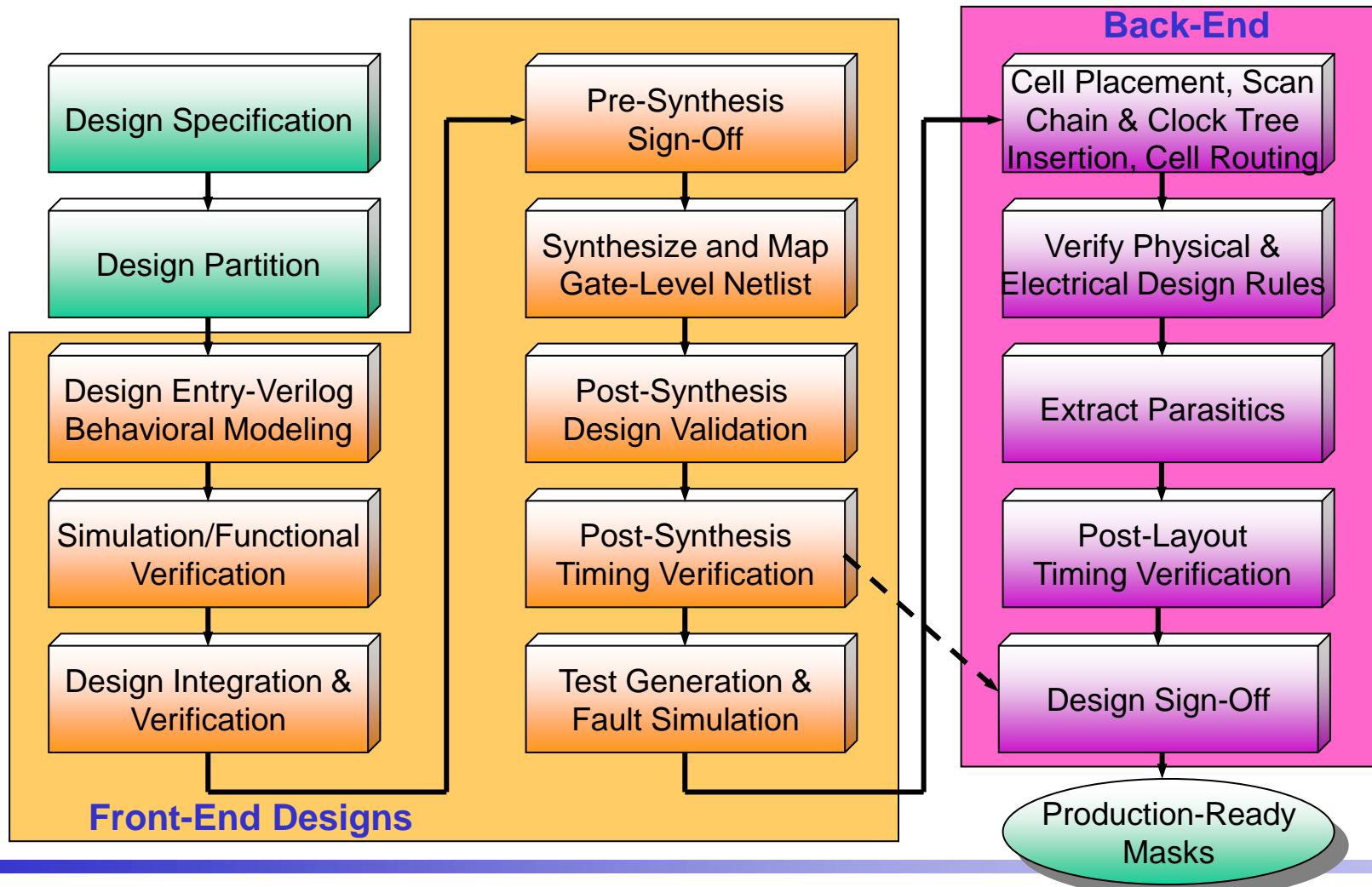


## Verilog HDL in Different Design Domains





## Digital IC Design Flow for Better Design Productivity (EDA tools)





# Case Study:

*A Simple RISC CPU Design*

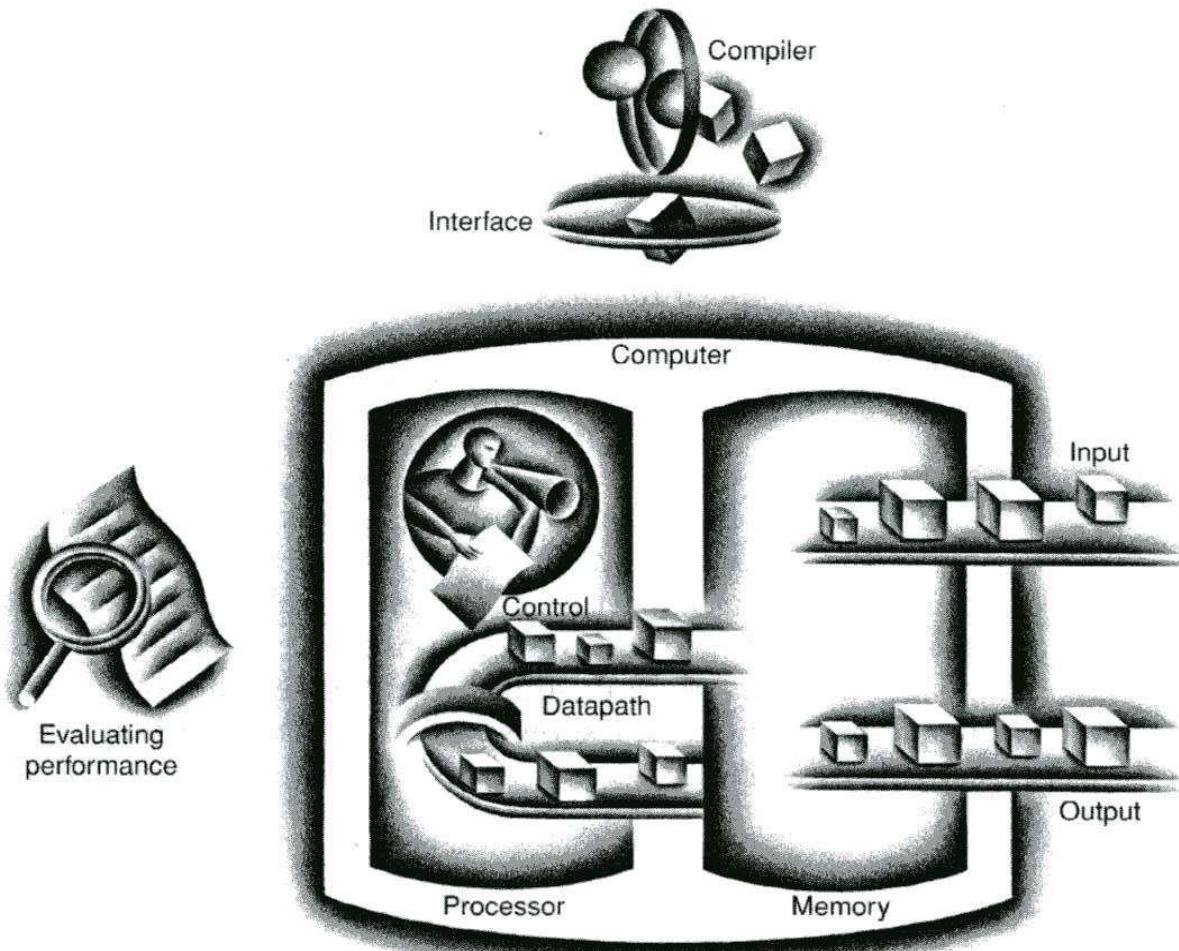


## A Desktop Computer





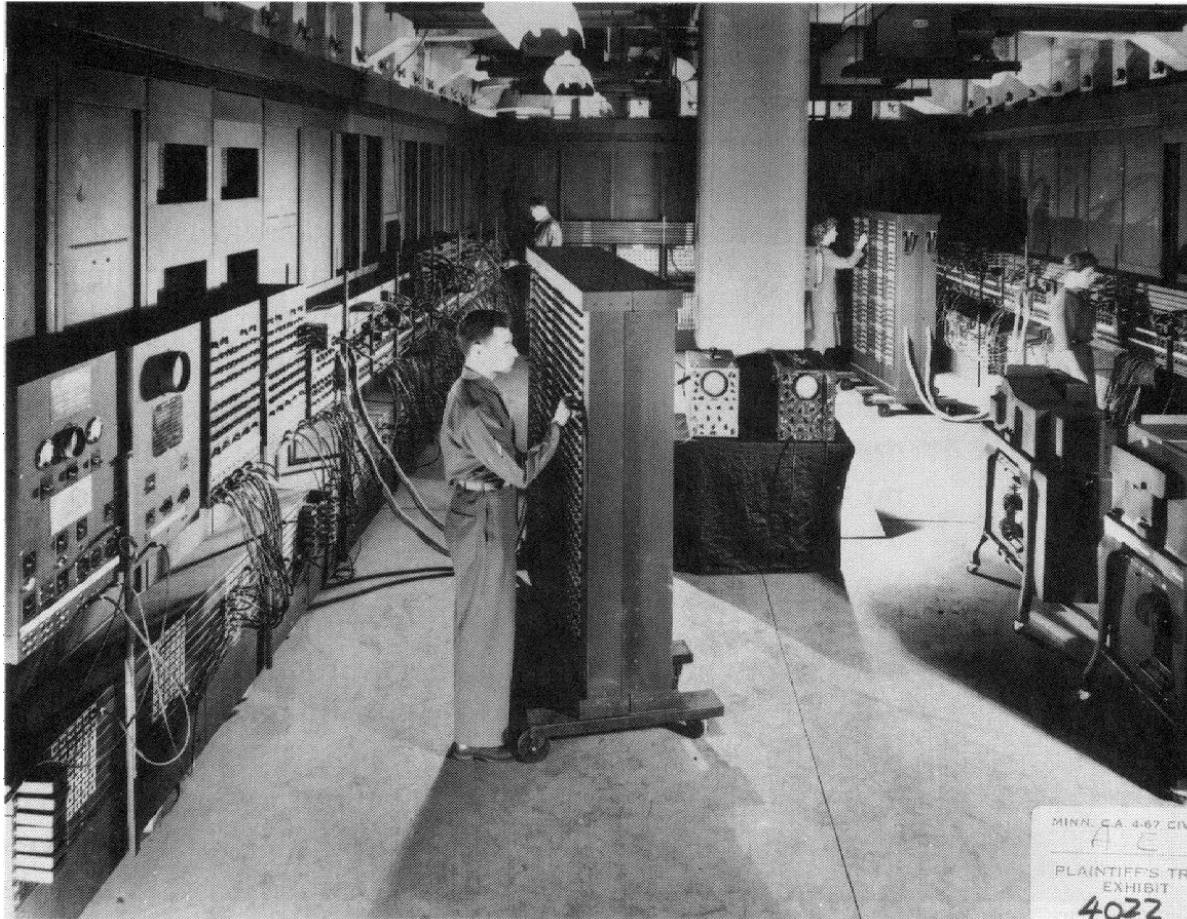
# Organization of a Computer



## Five Classic Components of a Computer



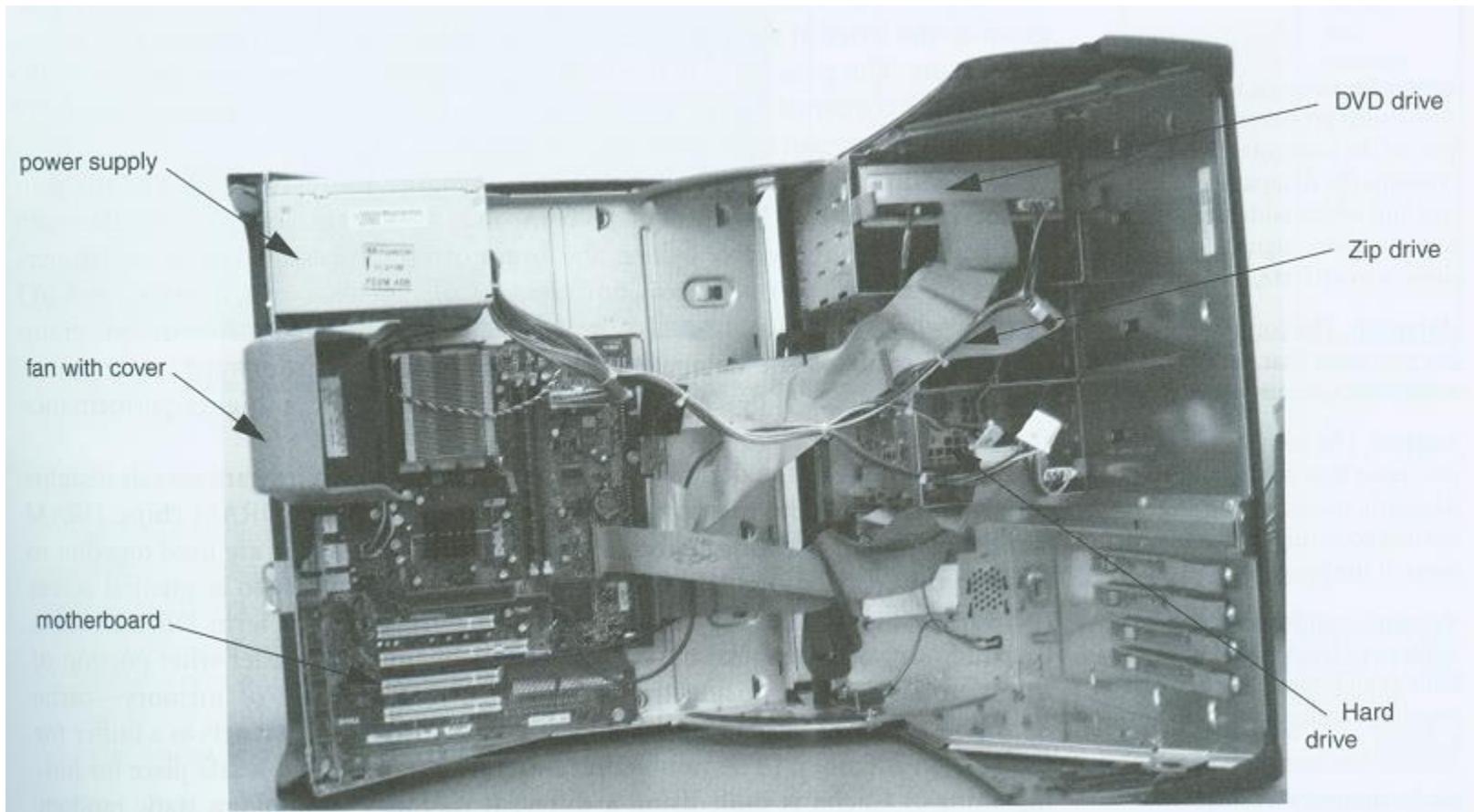
## ENIAC -The First Electronic Computer (1946)



*Use Vacuum Tubes as switching components*



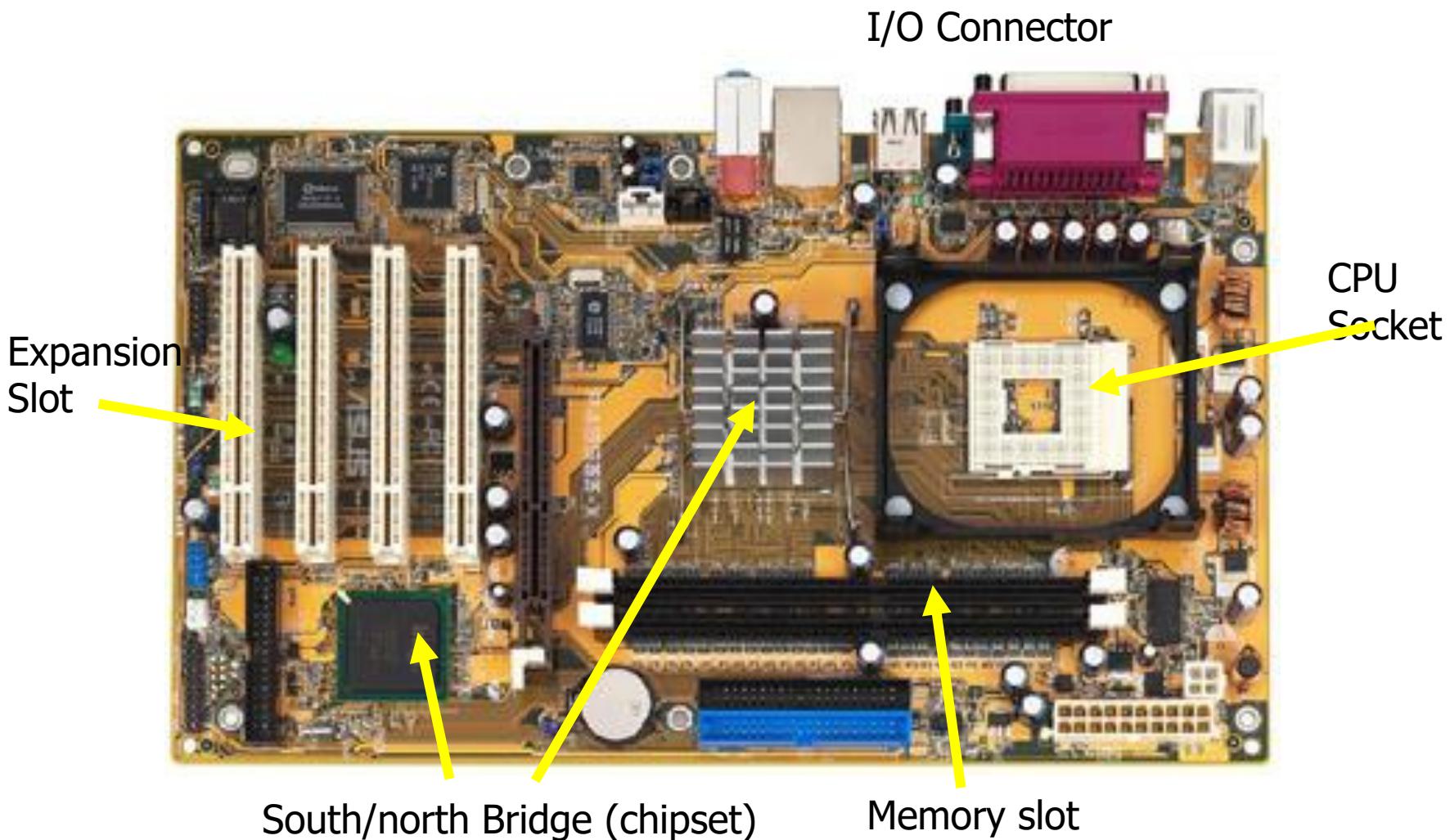
## Opening the Box



Inside the Personal Computer (PC)

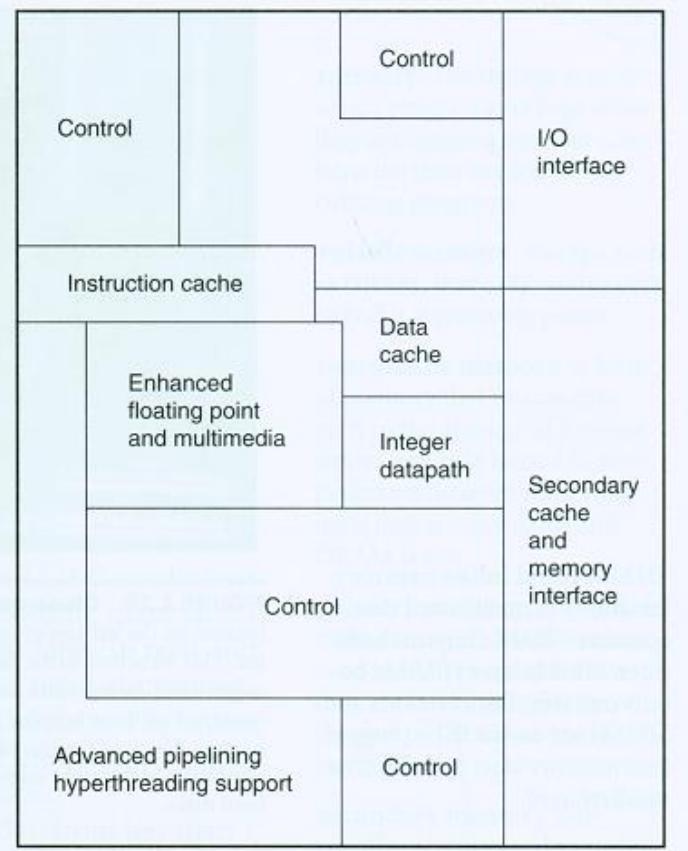
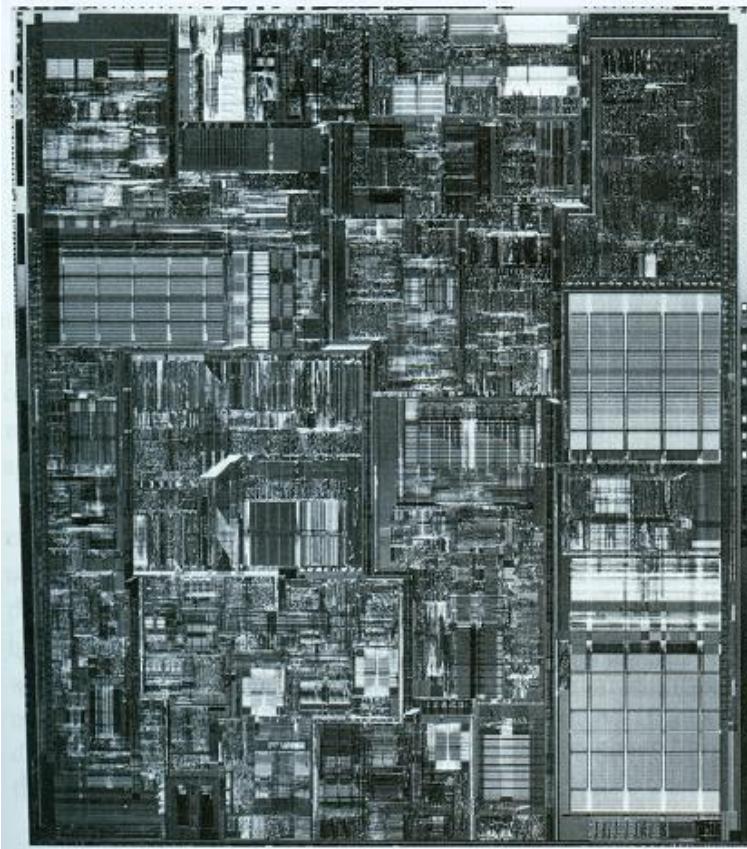


## Close-up of PC Motherboard





## Opening the Box



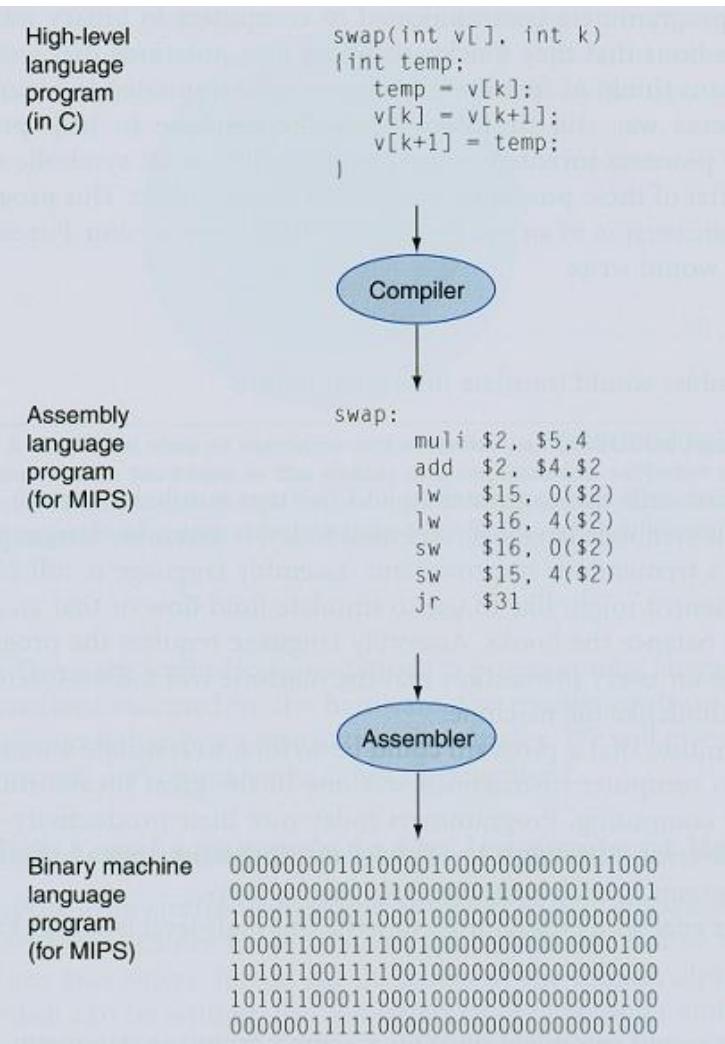
Inside the processor chip used on the board



## How to Make a CPU for Computer?



## From High-Level Language to the Language of Hardware



C program

→ compiled into **assembly language**

→ and then assembled into **binary machine language**



## Representing Instructions in the Computer

- ❖ Example in **C Language**:

Assume  $h = \$s2$  and  $\$t1$  has the base of the array A.

$A[300] = h + A[300];$

- ❖ Compiler → **Assembly language**

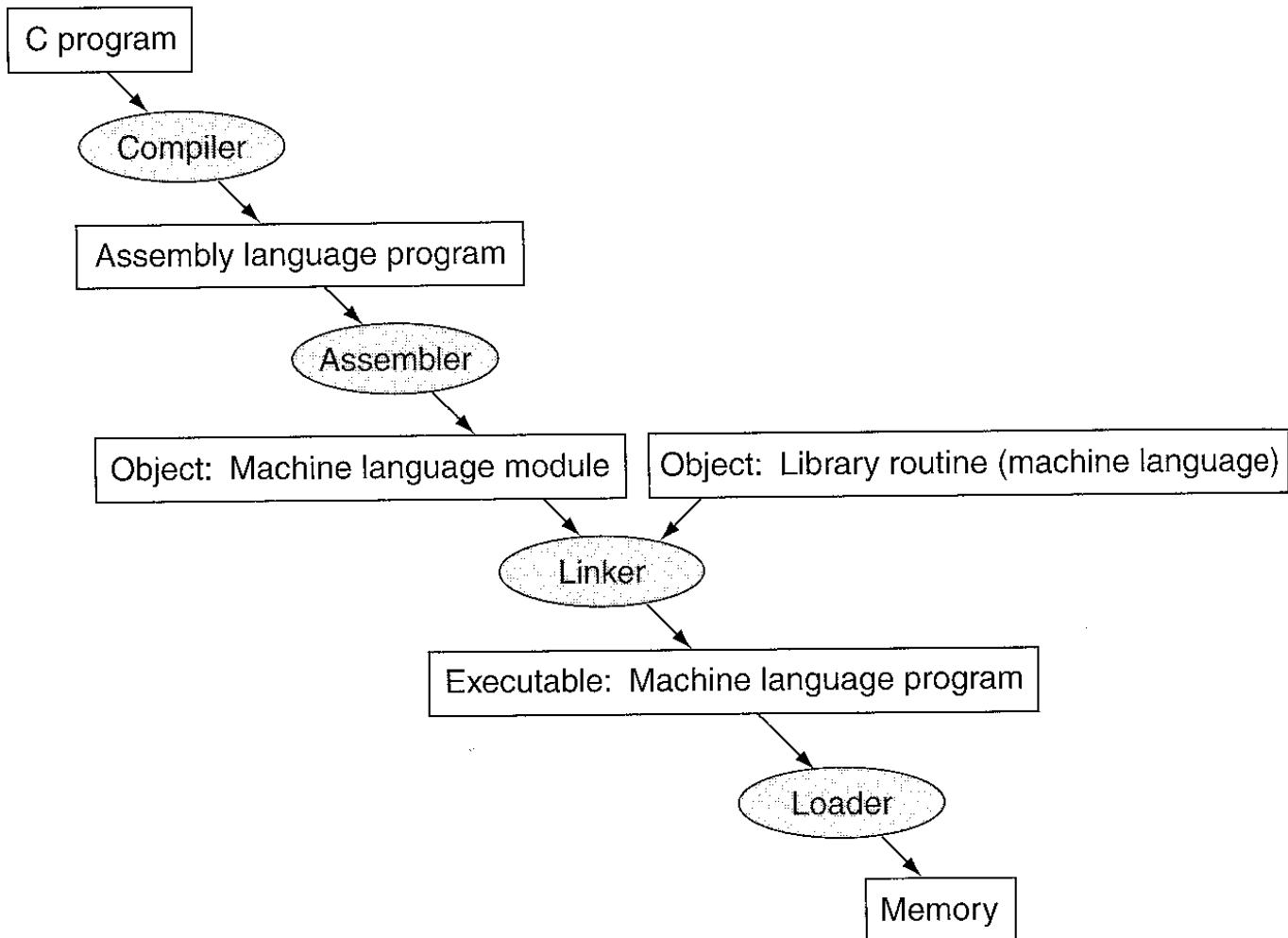
```
lw    $t0, 1200($t1)      # temp $t0 = A[300]
add  $t0, $s2, $t0        # temp $t0= h + A[300]
sw    $t0, 1200($t1)      # A[300] = h +A[300]
```

- ❖ Assembler → **Machine language** (in decimal number)

op	rs	rt	(rd)	(shamt)	address/ (Function)
35	9	8		1200	
0	18	8	8	0	32
43	9	8		1200	



## A Translation Hierarchy for C Program





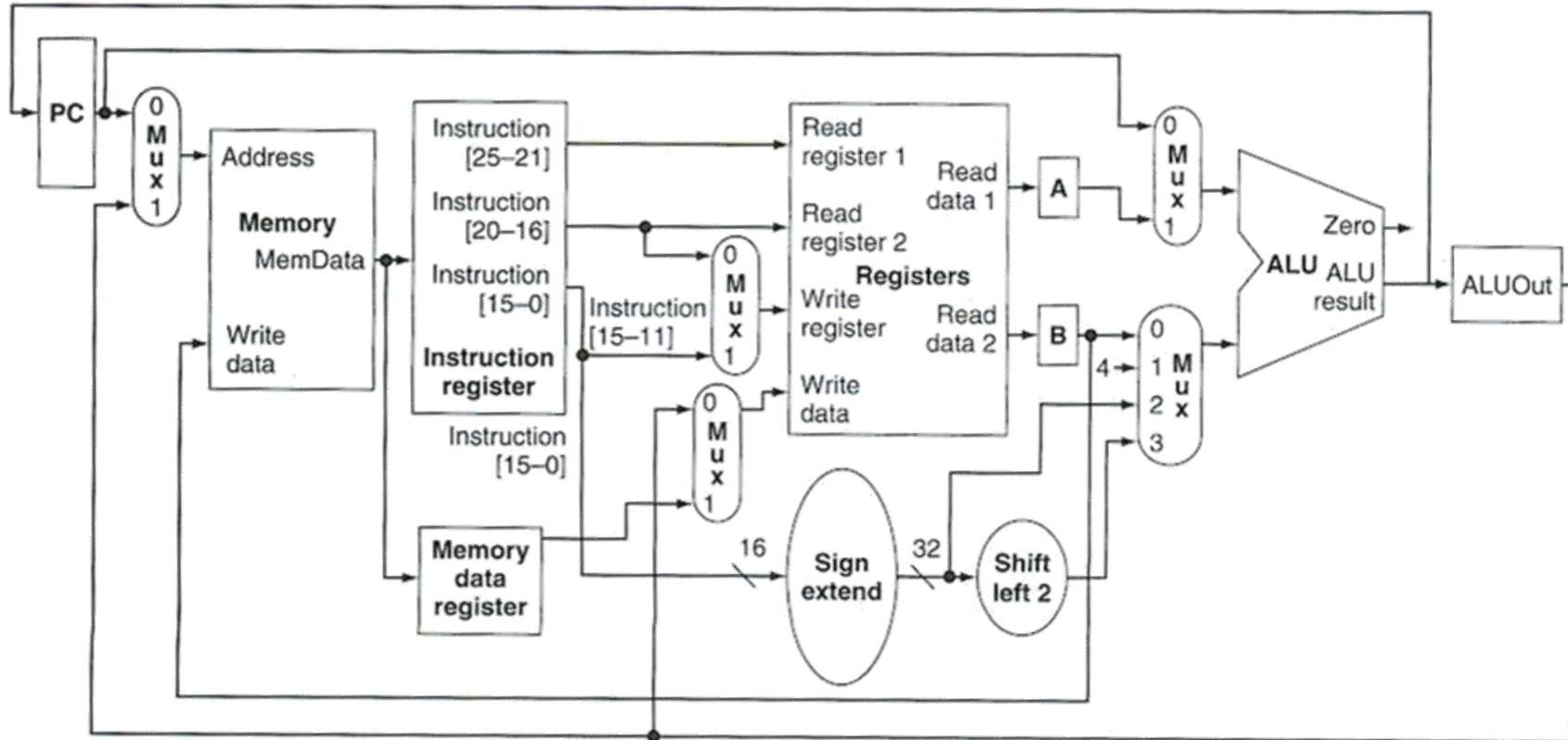
# MIPS Instruction & Machine Language

**MIPS machine language**

Name	Format	Example						Comments
add	R	0	18	19	17	0	32	add \$s1,\$s2,\$s3
sub	R	0	18	19	17	0	34	sub \$s1,\$s2,\$s3
lw	I	35	18	17	100			lw \$s1,100(\$s2)
sw	I	43	18	17	100			sw \$s1,100(\$s2)
and	R	0	18	19	17	0	36	and \$s1,\$s2,\$s3
or	R	0	18	19	17	0	37	or \$s1,\$s2,\$s3
nor	R	0	18	19	17	0	39	nor \$s1,\$s2,\$s3
andi	I	12	18	17	100			andi \$s1,\$s2,100
ori	I	13	18	17	100			ori \$s1,\$s2,100
sll	R	0	0	18	17	10	0	sll \$s1,\$s2,10
srl	R	0	0	18	17	10	2	srl \$s1,\$s2,10
beq	I	4	17	18	25			beq \$s1,\$s2,100
bne	I	5	17	18	25			bne \$s1,\$s2,100
slt	R	0	18	19	17	0	42	slt \$s1,\$s2,\$s3
j	J	2	2500					j 10000 (see Section 2.9)
jr	R	0	31	0	0	0	8	jr \$ra
jal	J	3	2500					jal 10000 (see Section 2.9)
Field size		6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions 32 bits
R-format	R	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
I-format	I	op	rs	rt	address			Data transfer, branch format



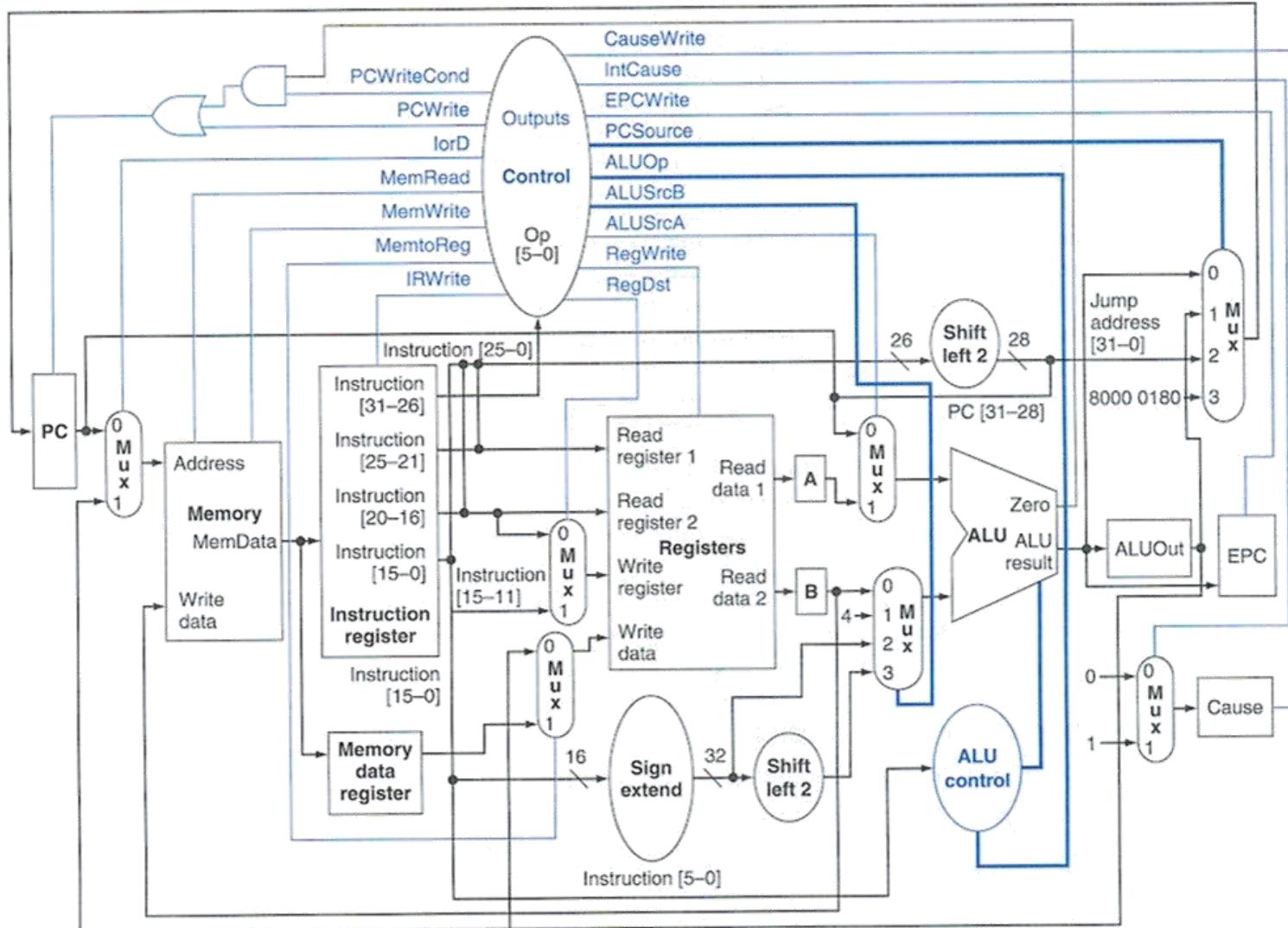
# A Multi-cycle Implementation



**FIGURE 5.26 Multicycle datapath for MIPS handles the basic instructions.** Although this datapath supports normal incrementing of the PC, a few more connections and a multiplexor will be needed for branches and jumps; we will add these shortly. The additions versus the single-clock datapath include several registers (IR, MDR, A, B, ALUOut), a multiplexor for the memory address, a multiplexor for the top ALU input, and expanding the multiplexor on the bottom ALU input into a four-way selector. These small additions allow us to remove two adders and a memory unit.

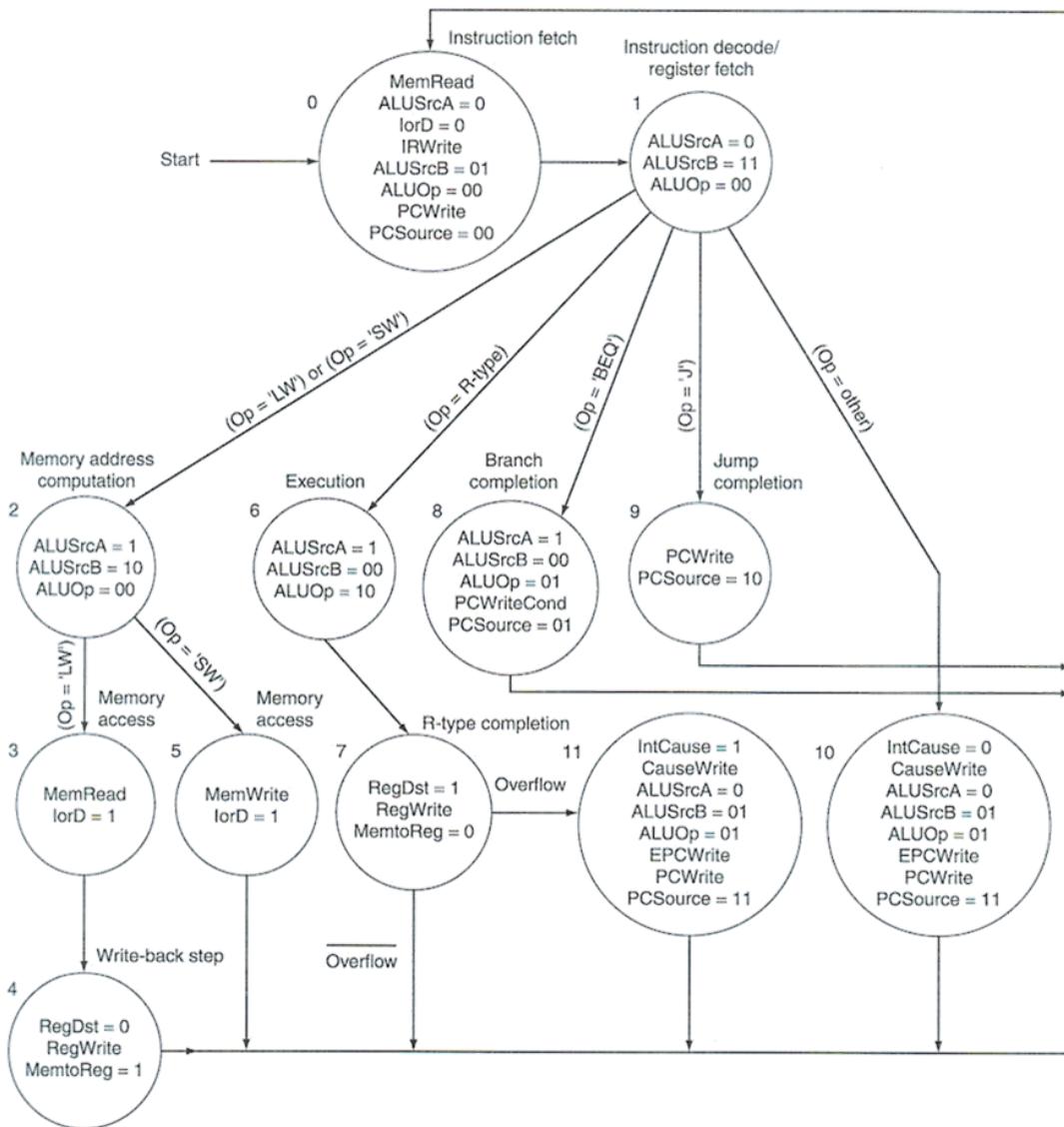


## Multi-cycle Implementation with CU





## Control Unit (CU) in multi-cycle RISC Implementation





## Summary

- ❖ Enhance **Verilog Programming** skills with **hardware sense**
- ❖ Use a complex system to drive your HDL programming skills
- ❖ A project-driven course: 3 people form a team
- ❖ Linking “Computer Organization” with “Digital Circuit Design Labs (數電實驗)” and “VLSI Design Labs (CVSD)”—
  - ❖ Port your design to FPGA board for verification (in Digital Circuit Design lab)
  - ❖ Port your design to VLSI for chip implementation (in VLSI Design lab, or graduate CVSD)
- ❖ Side note: You need more practice of Verilog than C/C++ if you want to be an excellent hardware/IC designer!
- ❖ Good for future parallel & multicore programming



# Backup slides

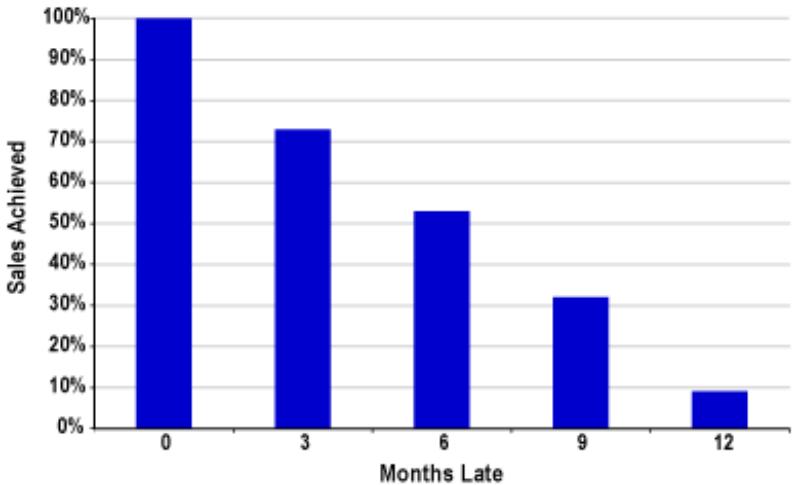


## FPGA Prototyping as Design Verification

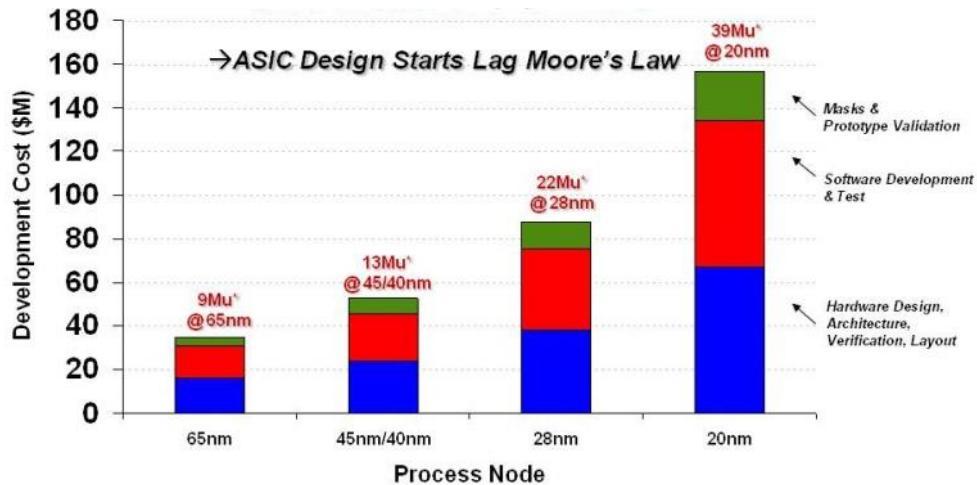


## Cost & Time-to-Market

- ❖ Leading-edge digital system designs are becoming more expensive and time-consuming
  - ❖ Increasing cost of mask sets and the amount of engineering verification required.
  - ❖ Very difficult for a company to react nimbly to competitive pressures or evolving standards.



Declining Product Sales Due to Late-to-Market Designs

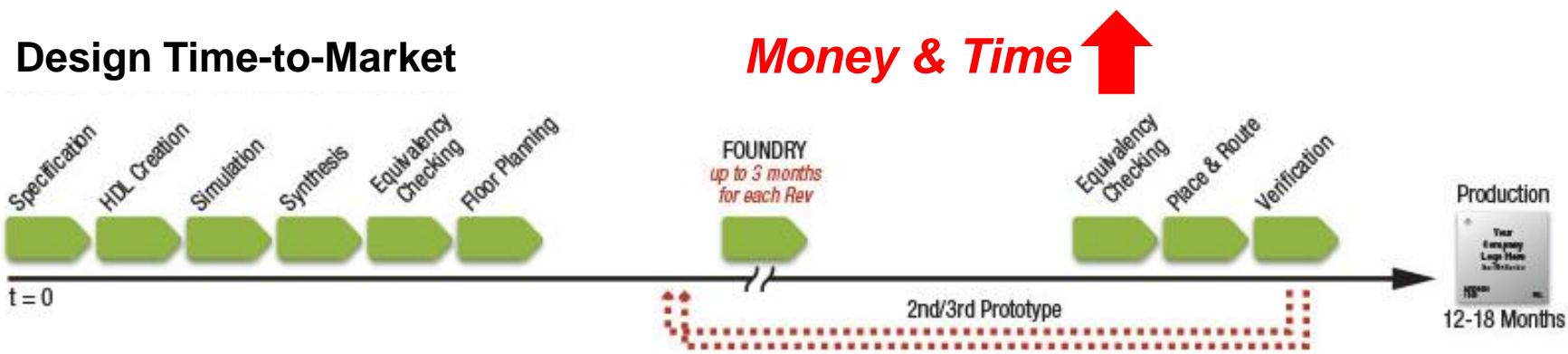


The Cost of Chip Development



## Cost & Time-to-Market

- ❖ Leading-edge digital system designs are becoming more expensive and time-consuming
  - ❖ Increasing cost of mask sets and the amount of engineering verification required.
  - ❖ Very difficult for a company to react nimbly to competitive pressures or evolving standards.



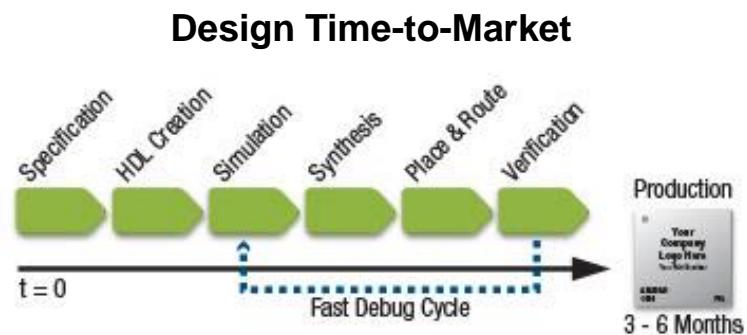
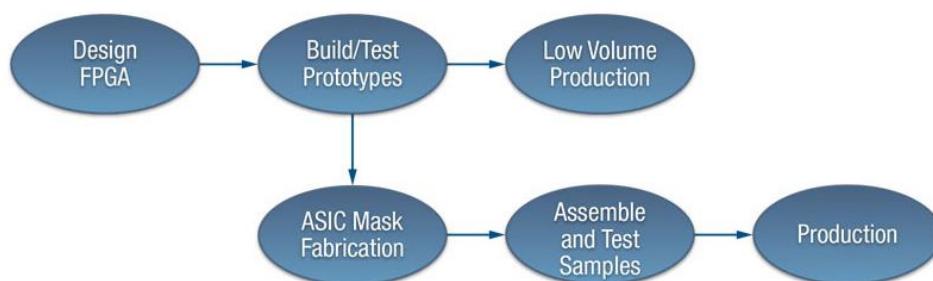
***Getting your design “right the first time” is more and more imperative !!!***



## FPGA Prototyping

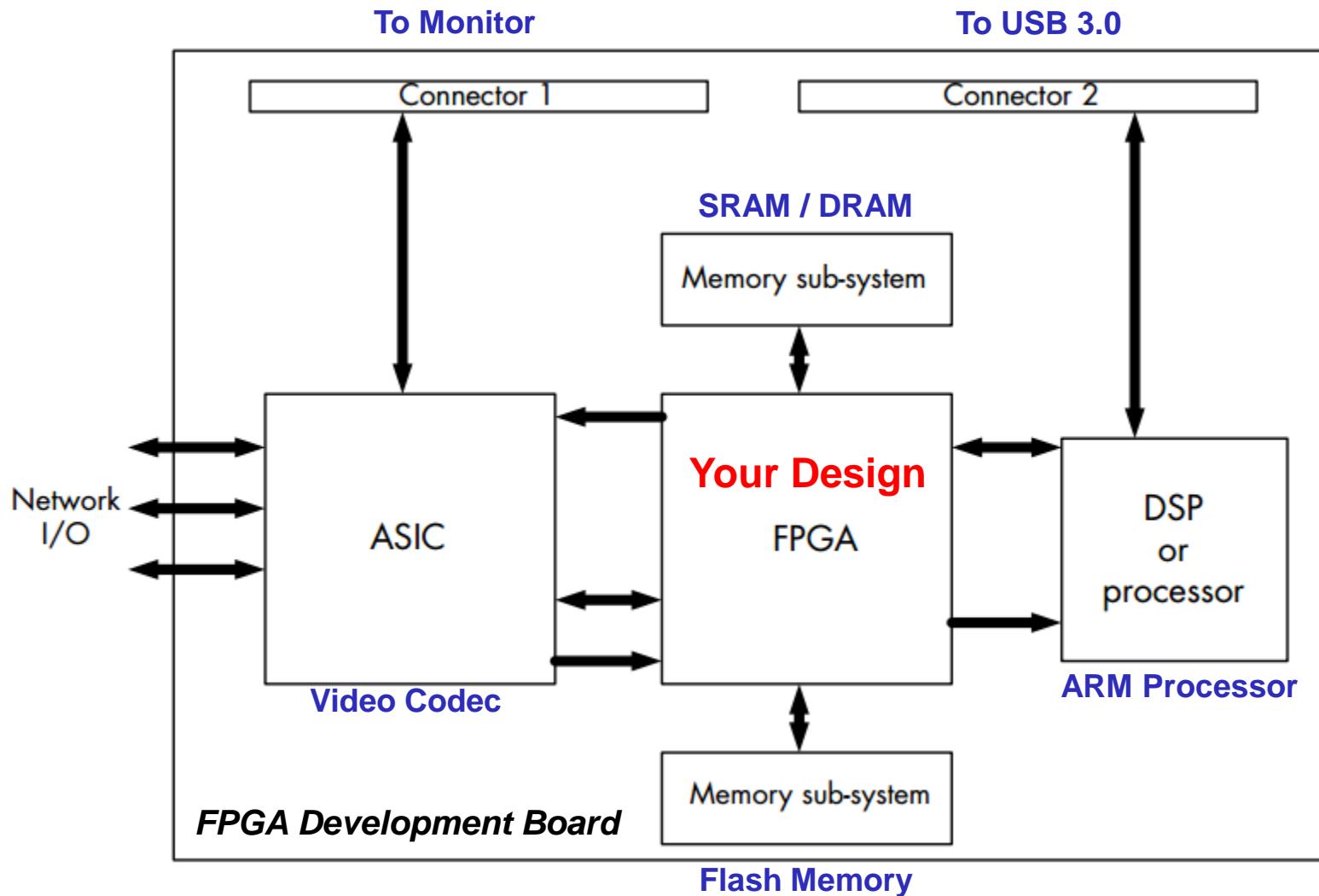
- ❖ Using an FPGA to prototype an digital system for verification has now become standard practice to:
  - ❖ Both decrease development time and reduce the risk of first silicon failure.
  - ❖ Faster “**emulation**” speed
  - ❖ Realistic system environment
  - ❖ System (HW/SW) development platform

Design process for developing a product with an FPGA  
and converting the FPGA to an ASIC for production.





# Verification Platform





## FPGA Design Flow

