《芯片设计自动化与智能优化》
Introduction Part I

Yibo Lin

Peking University

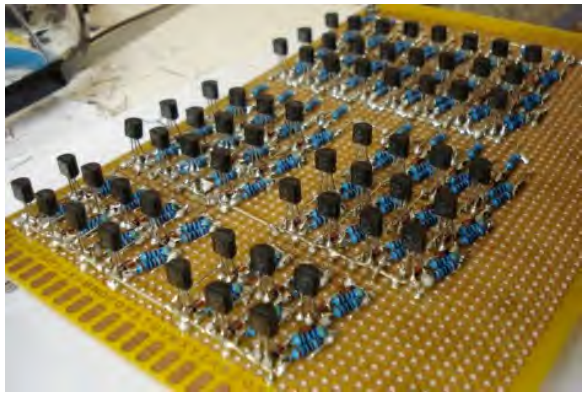
Outline

- Integrated circuits
- Industry and revenue
- From transistor to CPU
- Scaling and Moore's law
- What's this course about
  - Programs and compiler
  - Chip and hardware compile
- ASIC and FPGA design types
- IC design flow
- History of EDA
- What this course covers / course topics
- What to learn

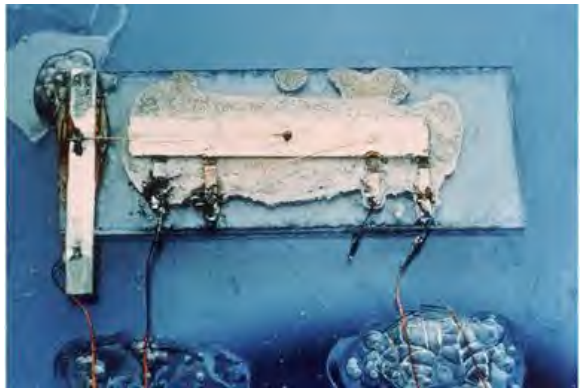
Integrated Circuits

摩尔定律50年下的芯片演变

- 基尔比 美国德州仪器公司工程师
  - 1958年发明集成电路，集成电路近一步缩小计算机体积，将大量晶体管压在一个单独的微型芯片上。
  - 2000年获得了诺贝尔物理学奖
- 1965年开始，**第三代计算机**使用**集成电路**



分立元件组成的电路



Jack Kilby发明的世界上第一个集成电路

杰克.基尔比
Jack Kilby

摩尔定律50年下的芯片演变

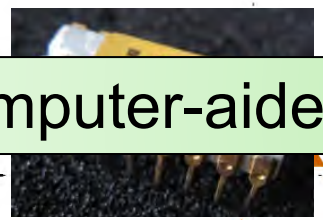
华为麒麟990@2021
103亿晶体管

Tech. Node 10um 3um 1~0.6um 0.25~0.13um 32nm 10nm

Intel奔腾@1993
3.1百万晶体管

Do engineers at Intel put 100 million transistors all manually?

No! Computer-aided design (CAD) tools play an important role.



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

Intel's Change in Development Model

Intel retires "tick-tock" development model, extending the life of each process

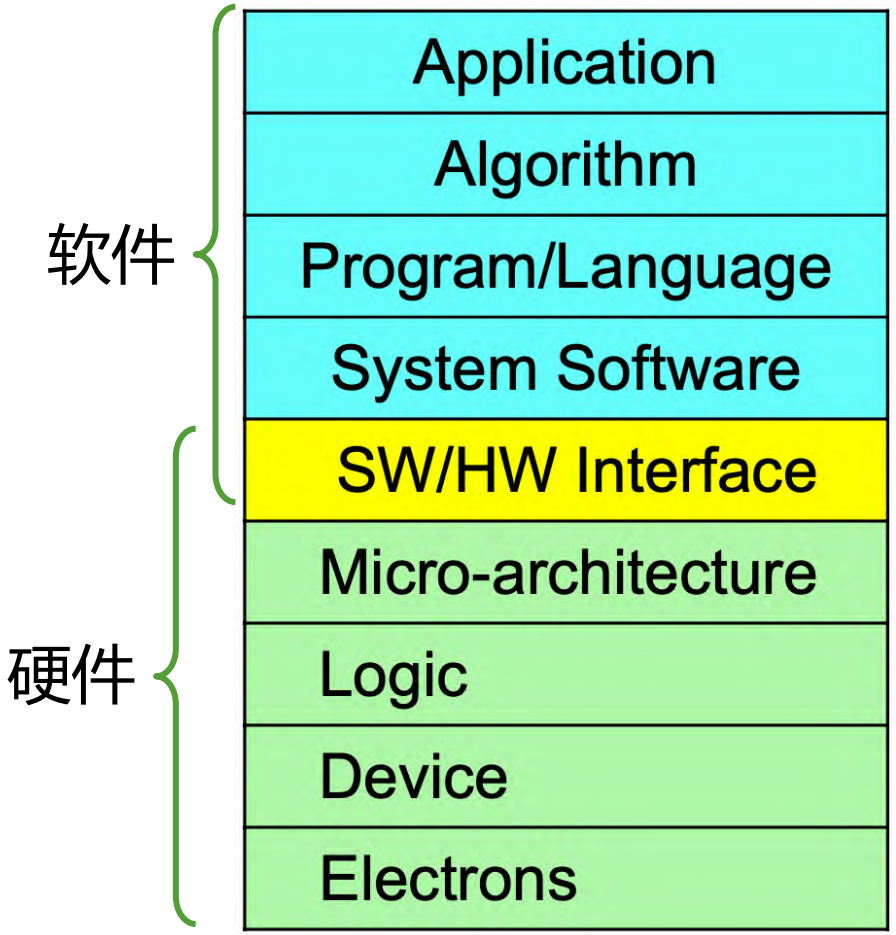
The new pattern is "Process, Architecture, Optimization."

ARS STAFF - 3/24/2016, 12:19 AM

It looks like the Kaby Lake processor isn't a one-off. Intel's latest 10-K filing (spotted at Motley Fool) discloses that the two-phase "tick-tock" development model that the company has been using since 2007 is being replaced with a three-phase model: Process, Architecture, Optimization.

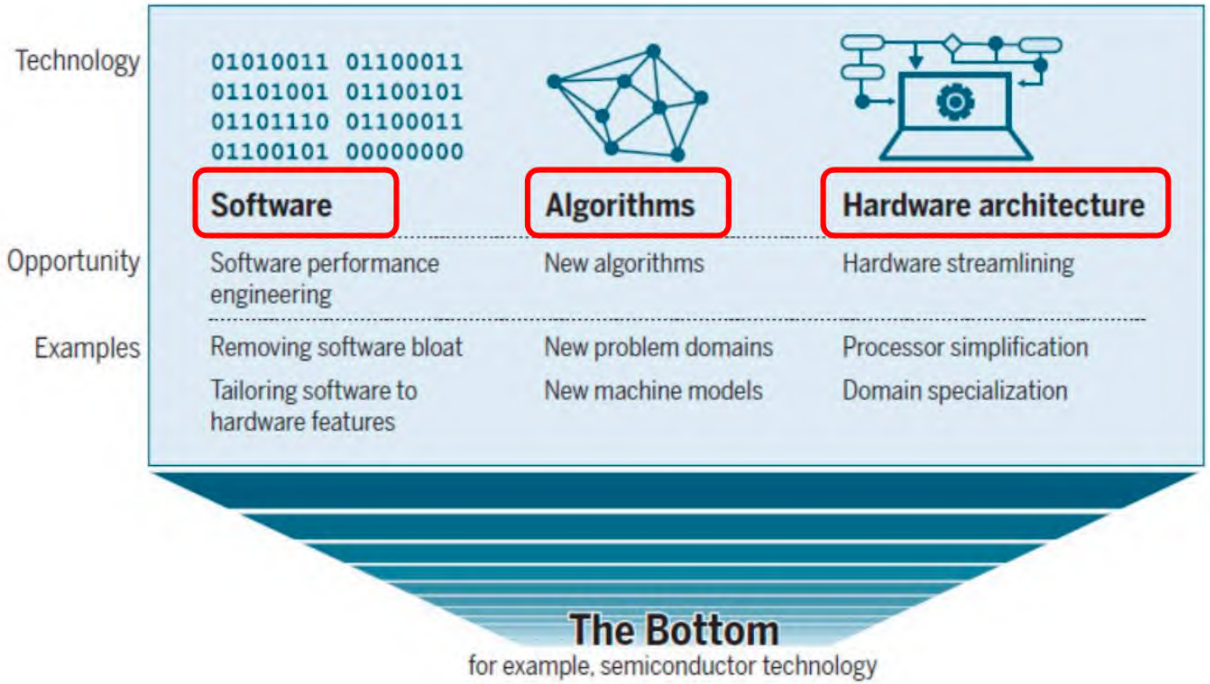
Under tick-tock, development was split into "ticks," where an existing processor design would be migrated to a new manufacturing process, and "tocks," where a new processor design would be released onto an existing process. The process has been used since Intel first introduced its "Core" branded processors, and the model has created a familiar pattern. Each tock introduces new features and improved architectural performance, and each tick has improved power consumption and/or clock speeds.

跨层次设计与优化（Cross-Layer）



There's plenty of room at the Top: What will drive computer performance after Moore's law?

Charles E. Leiserson, Neil C. Thompson*, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez, Tao B. Schardl



[Ct. Yun Liang]

跨层次设计与优化（Cross-Layer）

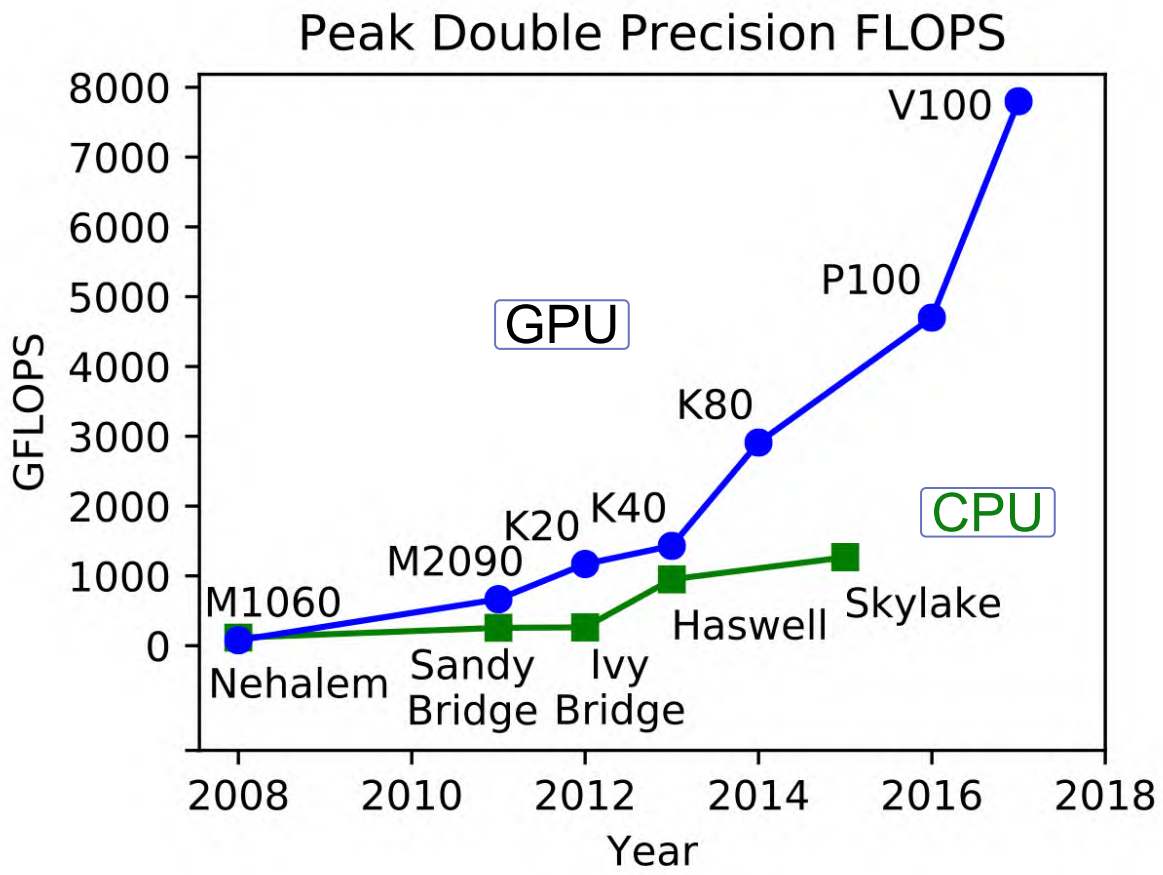
Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the running time of the version. "GFLOPS" is the billions of 64-bit floating-point operations per second that the version executes. "Absolute speedup" is time relative to Python, and "relative speedup," which we show with an additional digit of precision, is time relative to the preceding line. "Fraction of peak" is GFLOPS relative to the computer's peak 835 GFLOPS. See Methods for more details.

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	–	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

[Ct. Yun Liang]

跨层次设计与优化（Cross-Layer）

- 2013年以来，深度学习训练效率提升100倍以上
  - 得益于软硬件协同发展与优化



Two Different Worlds

学计算机/软件的同学

FPGA

Chisel

GPU

ASIC

Memory

Photonic

SSD

Quantum

RTL

Architecture

学硬件的同学

C++

O(nlogn)

Unix

Programming

Algorithm

NP

Python

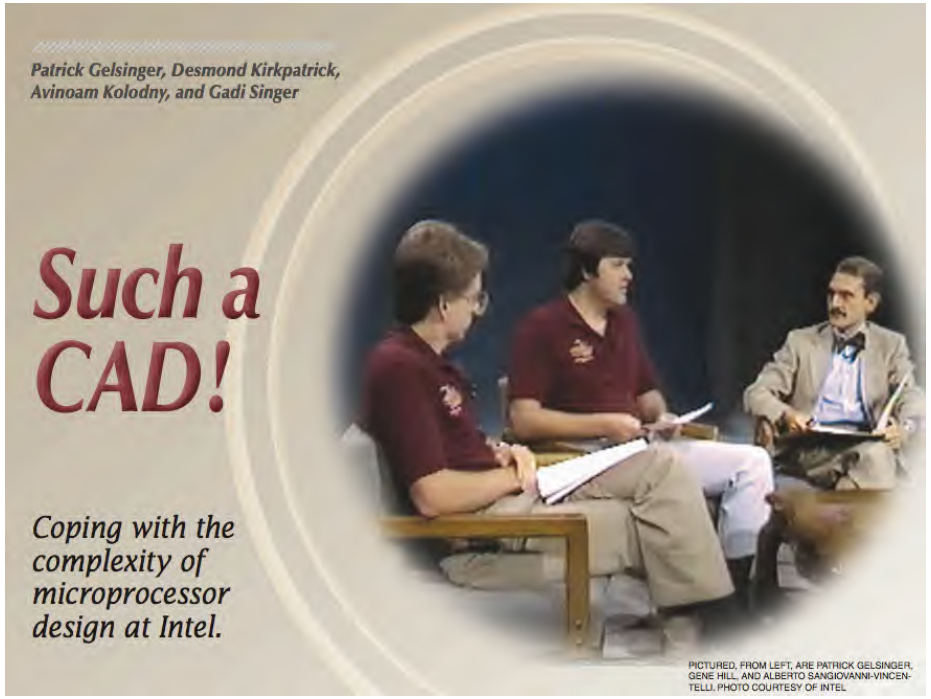
Compiler

Data Structure

Parallel Programming

[Ct. Yun Liang]

Industrial Impact of EDA



[source] Patrick Gelsinger, Desmond Kirkpatrick, Avinoam Kolodny, and Gadi Singer. "Such a CAD!" *IEEE Solid-State Circuits Magazine*, 2010.

Wikipedia, Transistor count

PROCESSOR	INTRO DATE	PROCESS	TRANSISTORS	FREQUENCY
4004	1971	10 μm	2,300	108 KHz
8086	1978	3 μm	29,000	2 MHz
Pentium	1993	0.8 μm	3.1 M	60 MHz
Core 2 Conroe	2006	65 nm	291 M	1.06 GHz
Core i7 Nehalem	2008	45nm	2.3 B	2.6 GHz
Core i7 Haswell	2014	22nm	2.6 B	3.4 GHz
Core i7 Skylake	2015	14 nm	1.75 B	4.5 GHz
Cerebras WSE2	2020	7 nm	2.6 T	3 GHz
Apple M1 Max	2021	5 nm	57 B	3.2 GHz

"This incredible growth rate could not be achieved by hiring an exponentially growing number of design engineers. It was fulfilled by adopting new design methodologies and by introducing innovative design automation software at every processor generation."

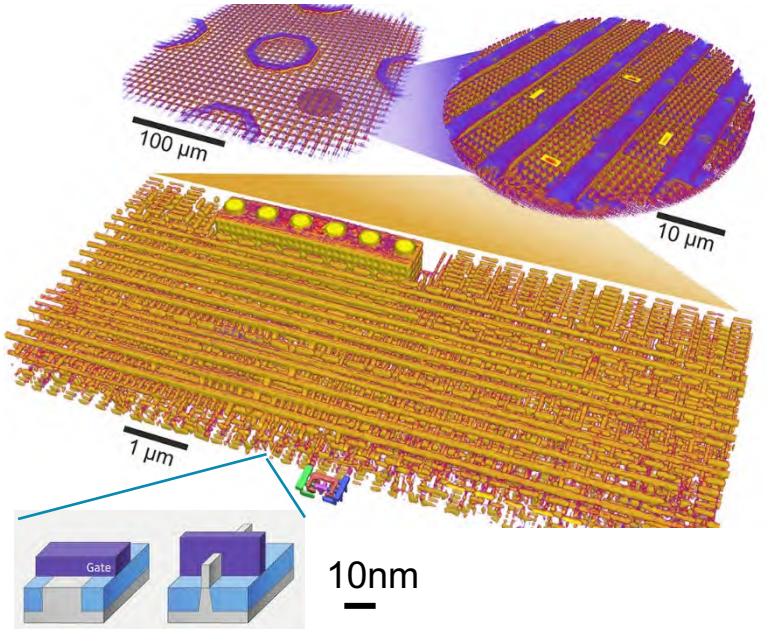
Challenges of EDA – A Rough Analogy

Smart City

Urban Planning : 1 m → 100 km

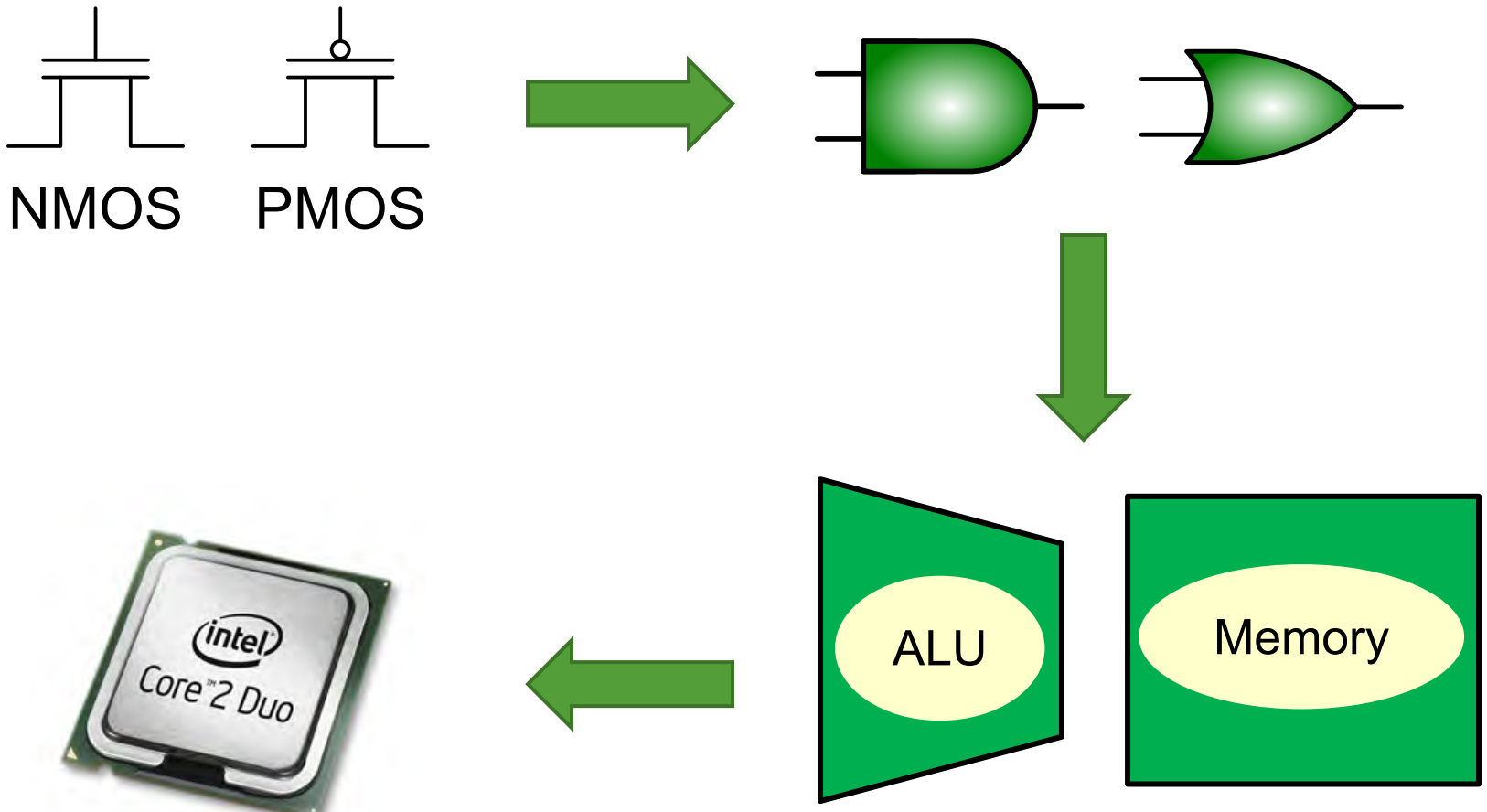
(Area of Beijing : 160 x 176 km^2)

Chip : Nanoscale City



IC Design : 10 nm → 10 mm

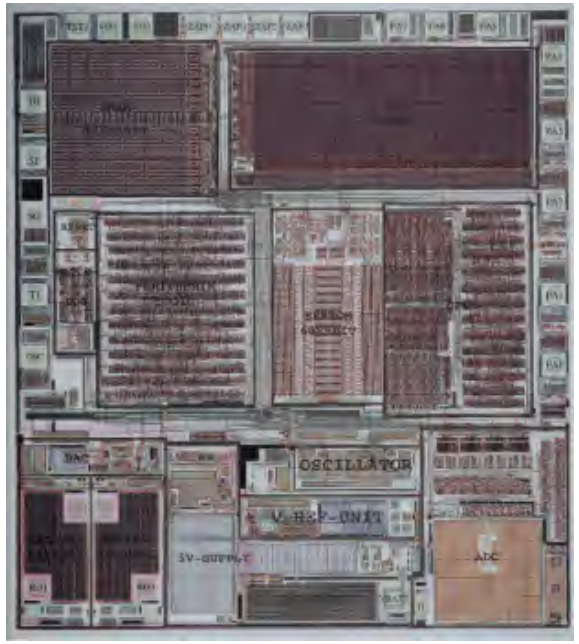
From Transistors to CPU



What's This Course About?

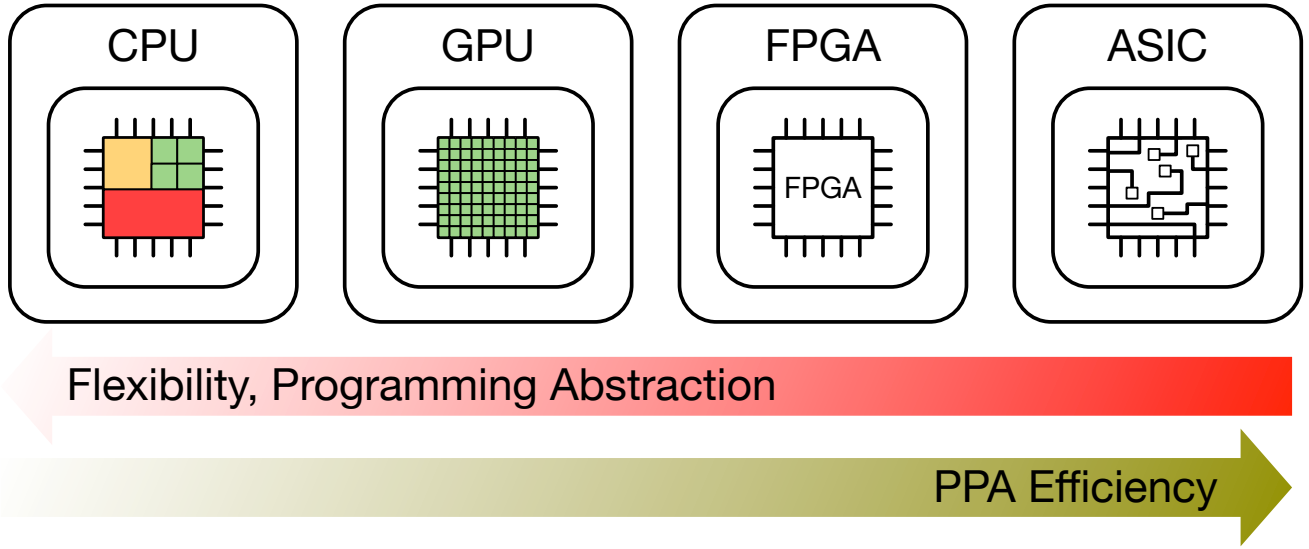
- Computer-aided design (CAD) tools for very large scale integrated (VLSI) circuits.
  - Indeed, we focus on the **algorithms**.
- CAD is also known as electronic design automation (EDA).
  - Analogy: compiler for software

VLSI CAD tools



A Tale of Two Design Types

- ASIC: Application-specific integrated circuit.
  - An integrated circuit customized for a particular use, rather than intended for general-purpose use.
  - Example: microprocessors, memories.
- FPGA: Field-programmable gate array.
  - **Programmable** logic blocks and **programmable** interconnects which allow the same FPGA to be used in many different applications.

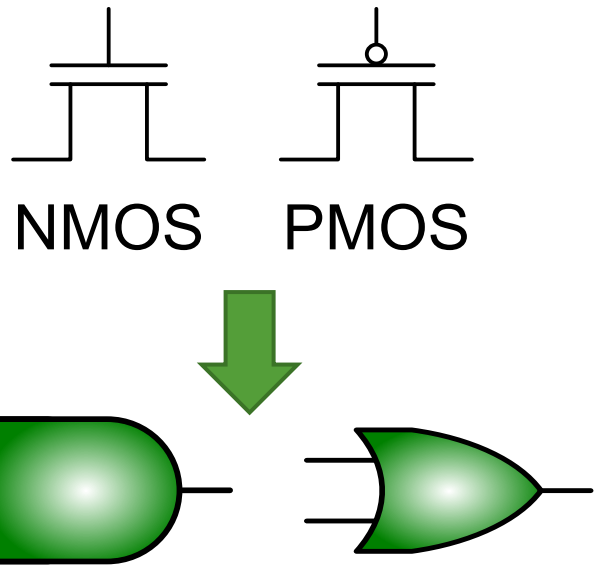


ASIC Design Styles

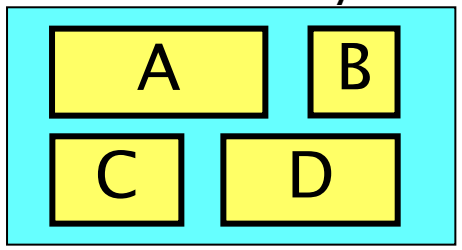
- Full-custom design
  - Transistors are hand-drawn.
  - Best performance (area, speed, etc.).
  - Costly. Only affordable when producing in large volume.
  - Today, only things like microprocessors are full custom.

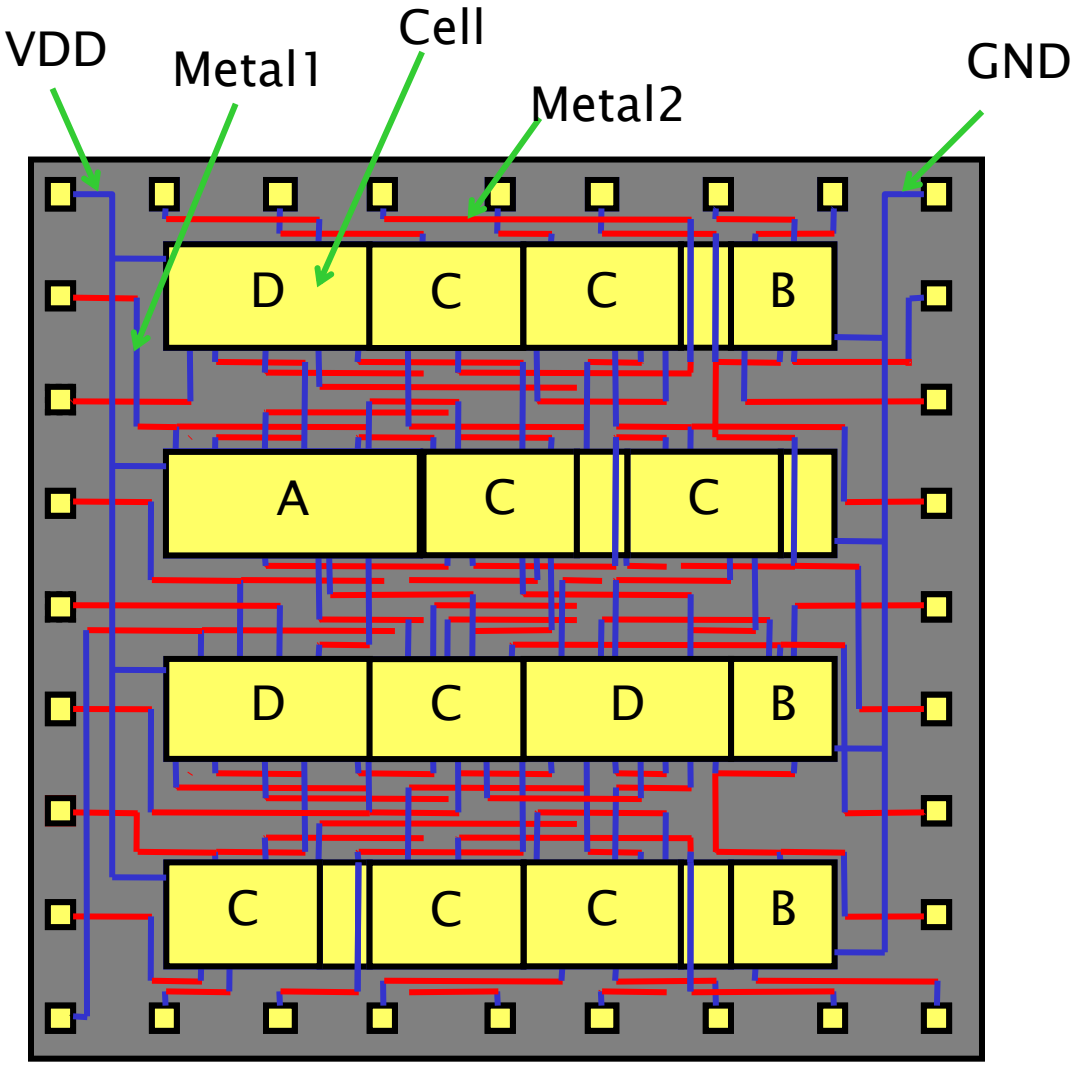
- Semi-custom design
  - Try to design reusing some already designed parts.
  - Not quite as dense (transistors / area) or as fast (GHz) as full custom.
  - One type: **standard-cell design**.

Standard Cell Design



Cell library

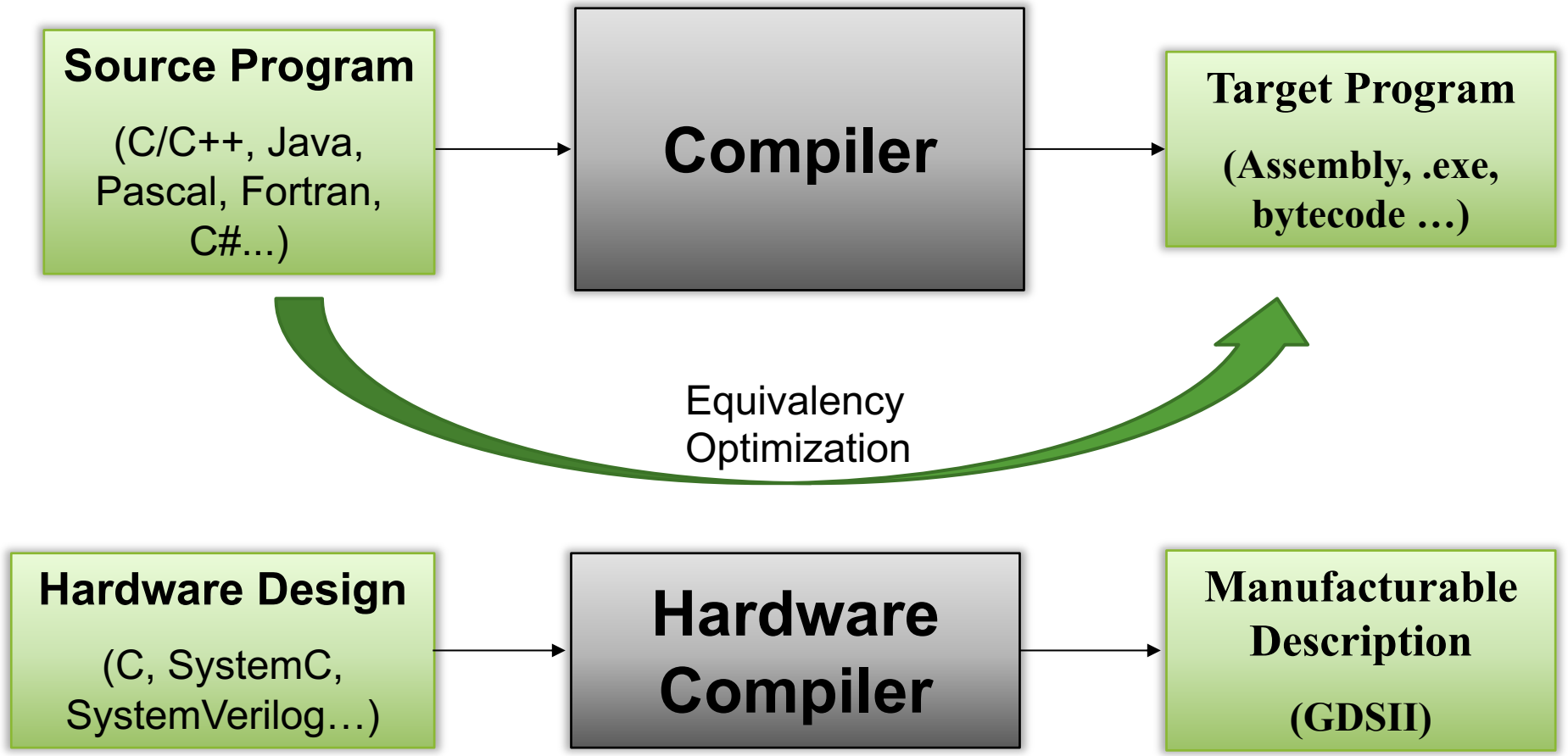




Comparison of ASIC Design Styles

Design Methods	Cost / Development Time	Performance	# Companies Involved
Full-Custom	Large	Best	Few
Semi-Custom	Small	Good	Many

IC Design Flow – Analogy to Software Compiler

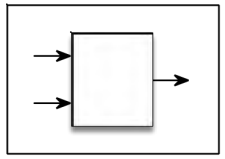


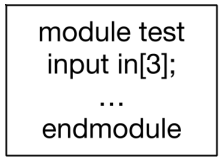
IC Design Flow – Hardware Compiler

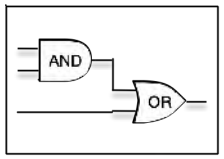
Fabless Design House

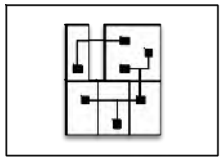
Fab/Foundry

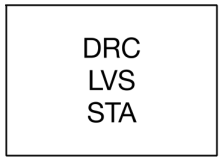
System Design | Logic Design | Backend Design

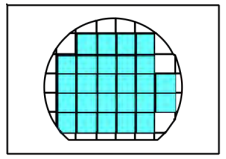


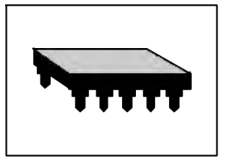












System Level Synthesis | Logic Synthesis | Physical Design | Physical Verification | Fabricate | Package Test

SystemC
SystemVerilog

Verilog
VHDL

```
module mux( input a,
    input b, input sel,
    output z, output zbar);
    assign z = sel ? b : a;
    assign zbar = z;
endmodule
```

Gate-level description

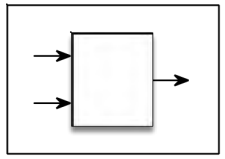
```
module and_gate( input a,
    input b, output z);
    wire zbar;
    NANDx2 inst1(a, b, zbar);
    INVx4 inst2(zbar, z);
endmodule
```

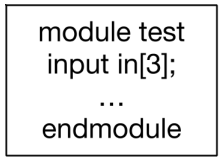
IC Design Flow – Hardware Compiler

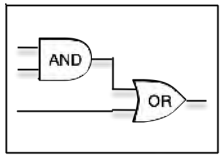
Fabless Design House

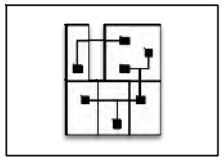
Fab/Foundry

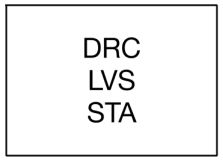
System Design | Logic Design | Backend Design

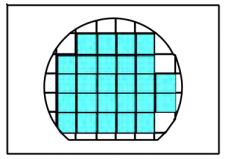


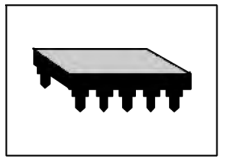




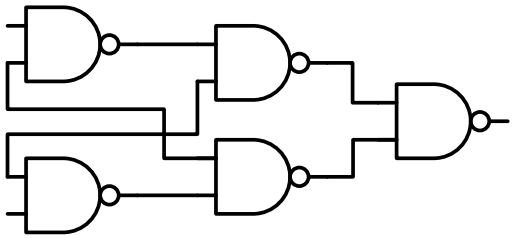


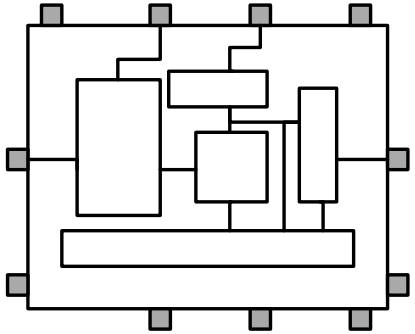


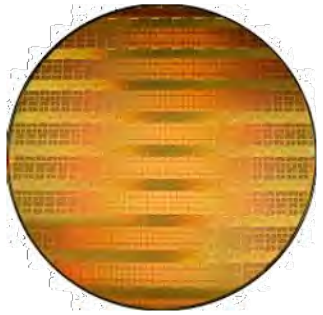




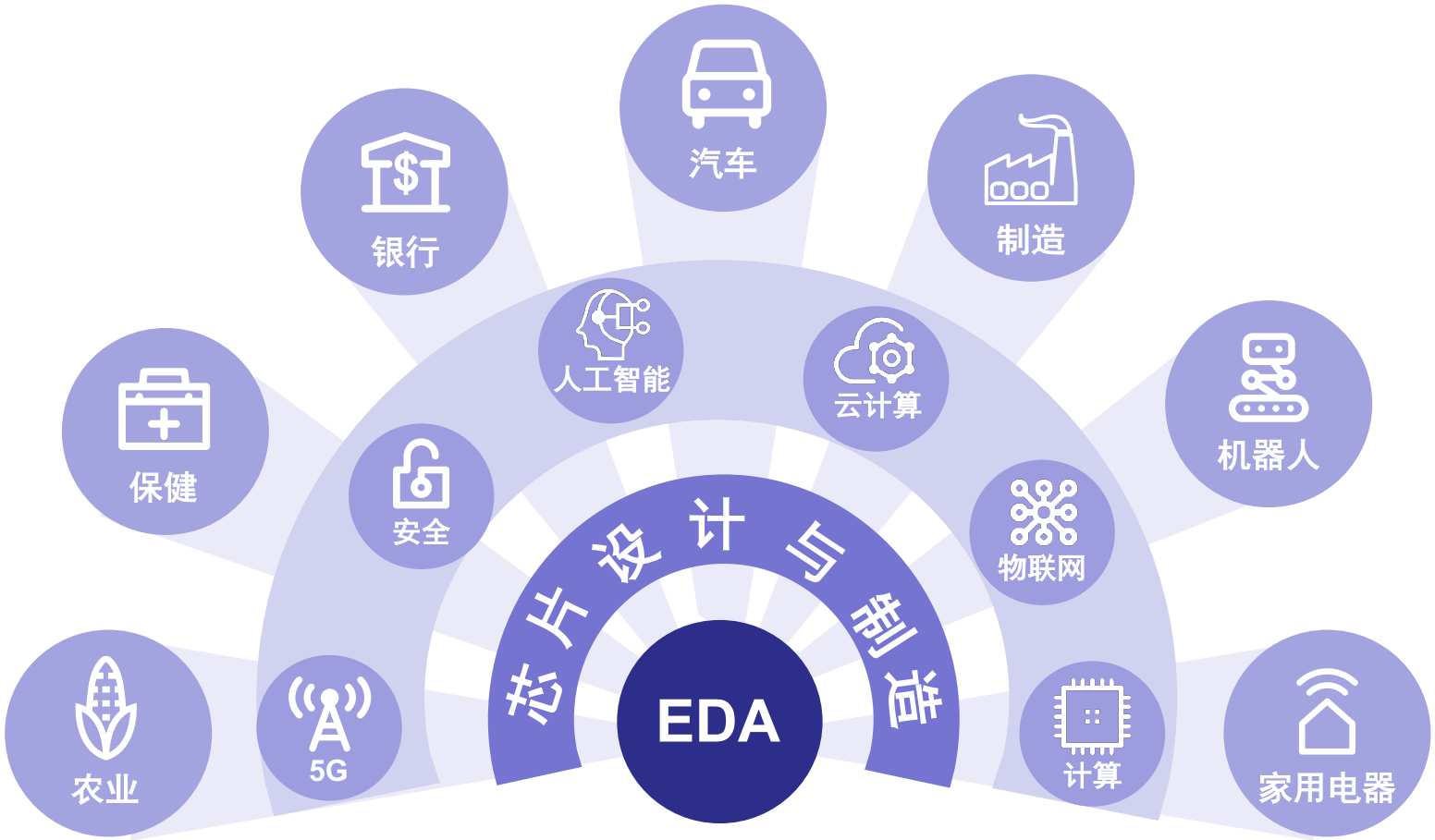
System Level Synthesis | Logic Synthesis | Physical Design | Physical Verification | Fabricate | Package Test





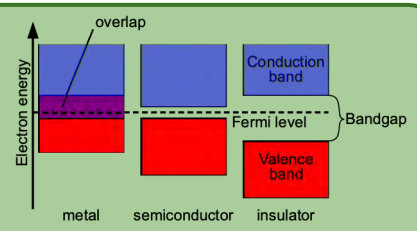


CAD/EDA – Foundation of Semiconductor Industry



History of EDA – Interdisciplinary Fusion

物理学

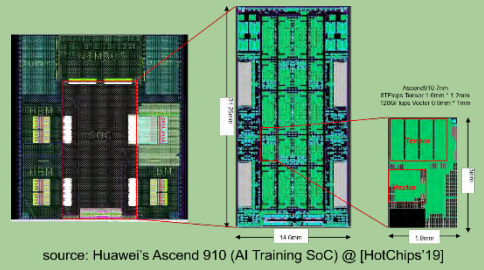


半导体能带理论

微电子学

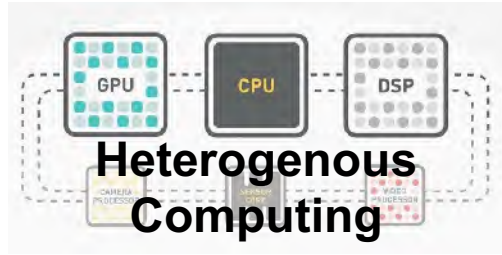
W. Shockley于1947年发明晶体管

J. Kilby于1958年发明集成电路



source: Huawei's Ascend 910 (AI Training SoC) @ [HotChips'19]

现代超大规模集成电路



Heterogenous Computing

Next?

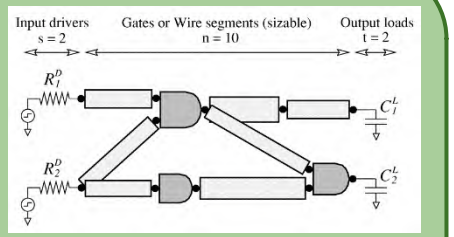
计算机科学

E. Kuh 葛守仁：60年代首批将数值计算用于电路设计

C.L. Liu 刘炯朗：将ad hoc EDA变革为算法驱动的EDA

应用数学

波恩大学离散数学研究院BonnTools被全球芯片设计者使用



凸优化应用于大规模电路优化问题

Fathers of CAD/EDA

Ernest Kuh
1928-2015
Dean@UC Berkeley

Students

- Massoud Pedram@USC
- Sanjit K. Mitra@Berkeley
- C.K. Cheng@UCSD
- Meiling Wang@Berkeley

Chung-Laung Liu
1934-2020
Professor@MIT, 1962-1972
A. Provost@UIUC, 1972-1998
President@NTHU, 1998-2002

Students

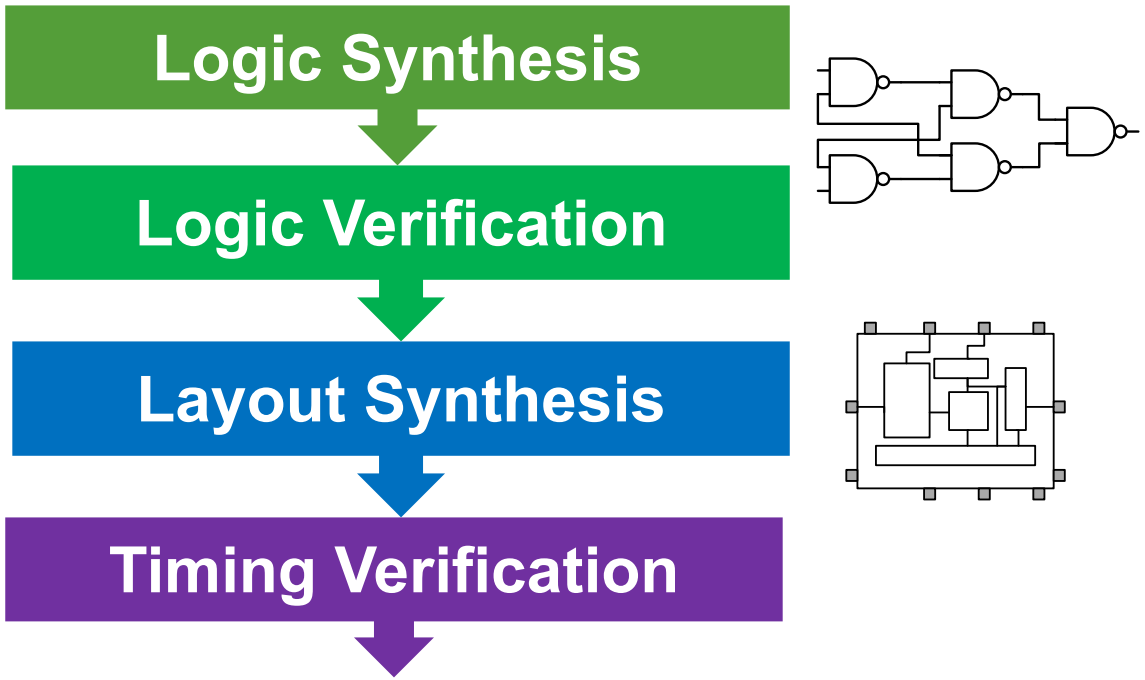
- Martin Wong@UIUC
- Andrew Yao@Princeton
- Shmuel Zaks@UIUC
- Jason Cong@UCLA
- …

Leading ad-hoc EDA to algorithm-driven EDA

https://academictree.org/etree/tree.php?pid=507028
https://academictree.org/evolution/tree.php?pid=826665

Focus of This Course

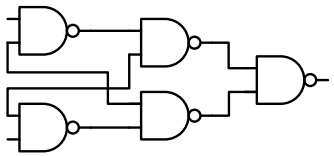
- Start with some Boolean / logic design description …
- …end with gates+wires, located at (x,y) coordinates on chip
- Big goals
  - Explain the critical algorithms, data structures, and modeling assumptions used in each of these big steps



Course Topics (Temporary)

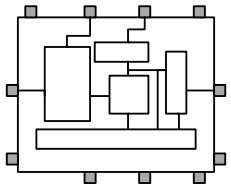
- Logic Synthesis and Verification
  - Computational Boolean Algebra
  - Binary Decision Diagram
  - Boolean Satisfiability
  - Logic Synthesis
  - Technology Mapping

Algebra
SAT
Graph Theory



- Layout Synthesis
  - Partitioning
  - Floorplanning
  - Placement
  - Routing

Numerical
Geometry
Graph Theory



- Timing Analysis
  - Graph-based analysis
  - Path-based analysis

Graph Theory

- Advanced Topics
  - Machine learning for EDA
  - Hardware acceleration
  - Design for manufacturability
  - …

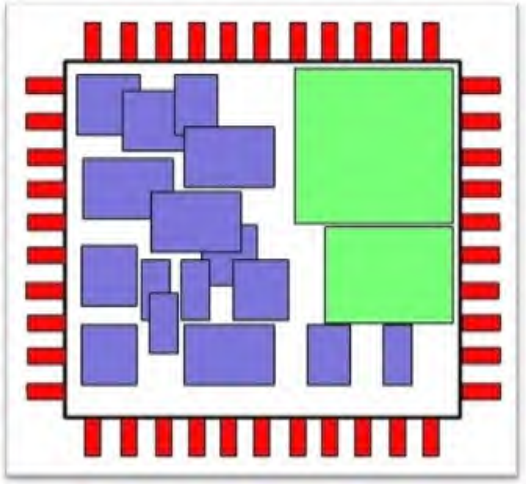
What Skills Will You Learn

- Foundation of IC design flow and methodology
  - Why are there so many design stages and tools
  - Why many existing tools are difficult to use (challenges)
- Formulate problems
  - Formalize a practical problem into a math/CS representation
  - Learn to solve practical problems with theoretical insights

Keep in mind…
In theory, we know everything, but nothing works.
In practice, everything works, but nobody knows why.

- Algorithms and programming skills
  - Typical ones to solve graph and numerical optimization
  - E.g., network flow, partitioning, dynamic programming, basic convex optimization

Widely-Existing Problems – Placement



Circuit Placement

Urban Planning

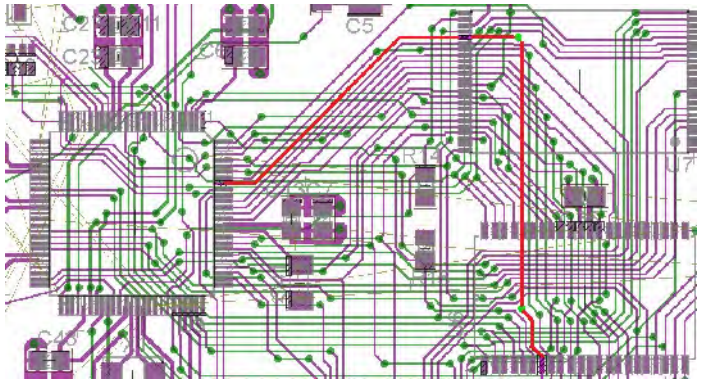
Warehouse Location

Storage Planning

Book Placement

Data Placement on Disk

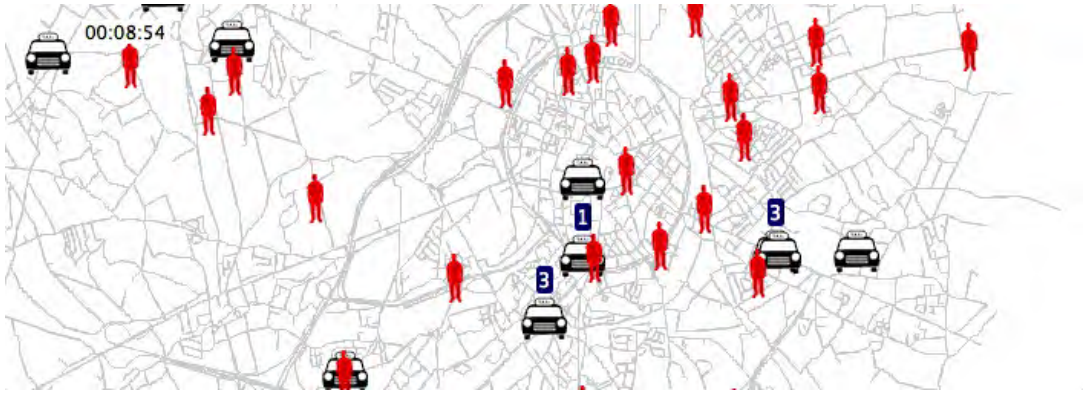
Widely-Existing Problems – Routing



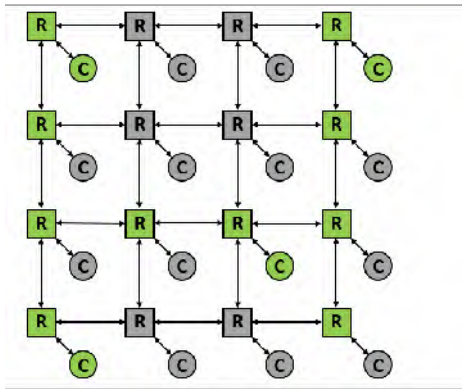
Circuit Routing

Road Planning

GPS Navigation



Vehicle Routing



Network-on-Chip Routing

Any questions?