中国科学技术大学
University of Science and Technology of China

计算系统概论A
Introduction to Computing Systems
(CS1002A.03)

Introduction to Computing Systems
计算系统概论A
CS1002A.03

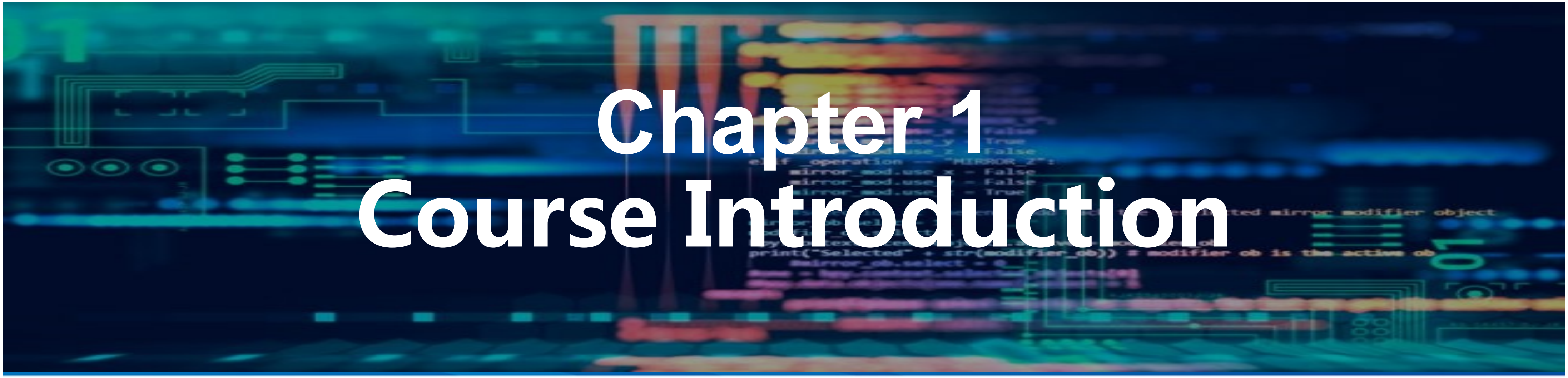
陈俊仕

2024 Fall

计算机科学与技术学院
School of Computer Science and Technology

中国科学技术大学
University of Science and Technology of China

计算系统概论A
Introduction to Computing Systems
(CS1002A.03)



Chapter 1
Course Introduction

计算机科学与技术学院
School of Computer Science and Technology

Outline

0 Course Crew

1 What’ s the difference between “Big” and “Small” ?

2 Why Take This Course?

3 Great Ideas in Computing Systems

4 Course Implementation

Outline

0 **Course Crew**

1 What’ s the difference between “Big” and “Small” ?

2 Why Take This Course?

3 Great Ideas in Computing Systems

4 Course Implementation

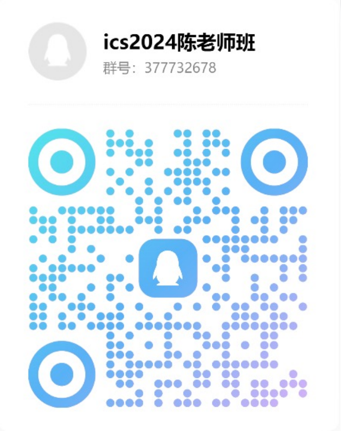
Course Crew of CS1002A.03

	Name	Email	Phone
Instructor	Jun-Shi Chen（陈俊仕）	cjuns@ustc.edu.cn	13721074819
TA	Si-Yue Chen（陈思悦）	daffodils7@mail.ustc.edu.cn	17756001132
TA	Xing-Tang（唐星）	tangxing@mail.ustc.edu.cn	13956281317
TA	Rui-heng Zhang（张锐恒）	zhangruiheng@mail.ustc.edu.cn	13570663434

Office hours and Discussion

- Chaired by 3 TAs
- Face to face help
- See web page for times

Web of course: http://acsa.ustc.edu.cn/ics/



ics2024陈老师班

群号：377732678

Outline

0 Course Crew

1 What’ s the difference between “Big” and “Small” ?

2 Why Take This Course?

3 Great Ideas in Computing Systems

4 Course Implementation

Questions

- How powerful are today's computers? Why are they so powerful?

What can computers do?

- Is an abacus a computer? How to understand Turing's contribution to computers?

How are they done?

- What are the serious flaws in today's computers?

What can't computers do?

Today, Computer is in Everything！

Games

Pad

Refrigerators

Robots

Laptops

Routers

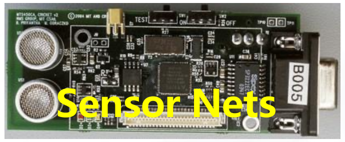
Smart Phones

Media Players

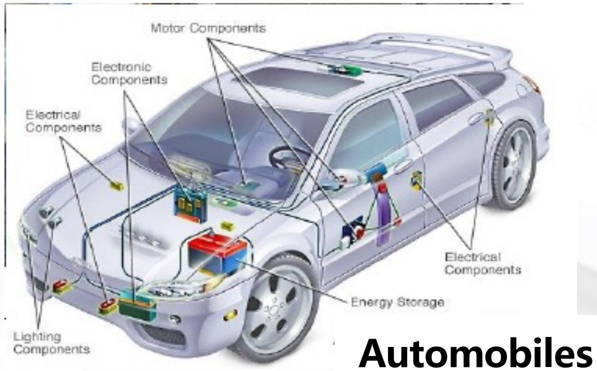
Set-top boxes

Cameras

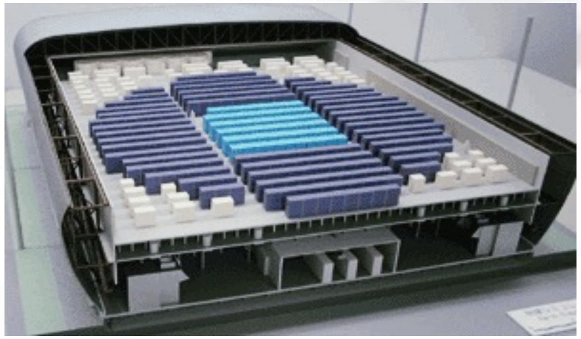
Servers



Sensor Nets



Automobiles



Supercomputers

Today, Computer is in Everything！

Vast infrastructure behind them: from the small to the big

Massive Cluster
Clusters
Gigabit Ethernet
Supercomputers
Laptops
Servers
Routers
Cameras
Games
Cars
Robots
Internet of Things
THE INTERNET OF THINGS
Wi Fi

Scalable, Reliable, Secure Services

Internet Connectivity

Databases
Information Collection
Remote Storage
Online Games
Commerce

MEMS for Sensor Nets

“Big ” vs. “Small” Computer

Personal mobile devices/smart terminal devices

Supercomputing Center (computation) / Data Center (storage)

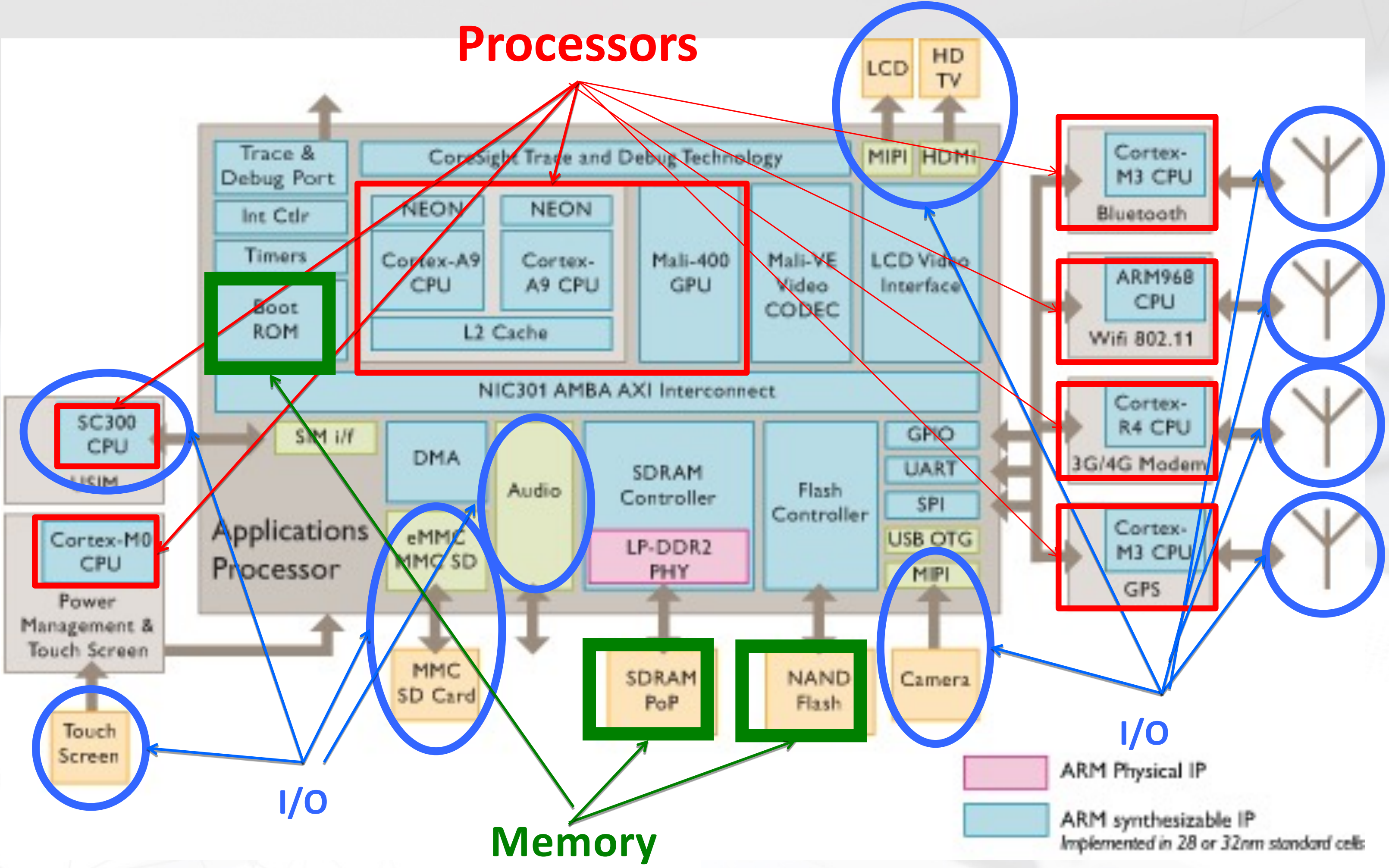
Small

- High performance calculation
- High-performance communications and I/O
- Power consumption constraints
- The volume constraint

Big

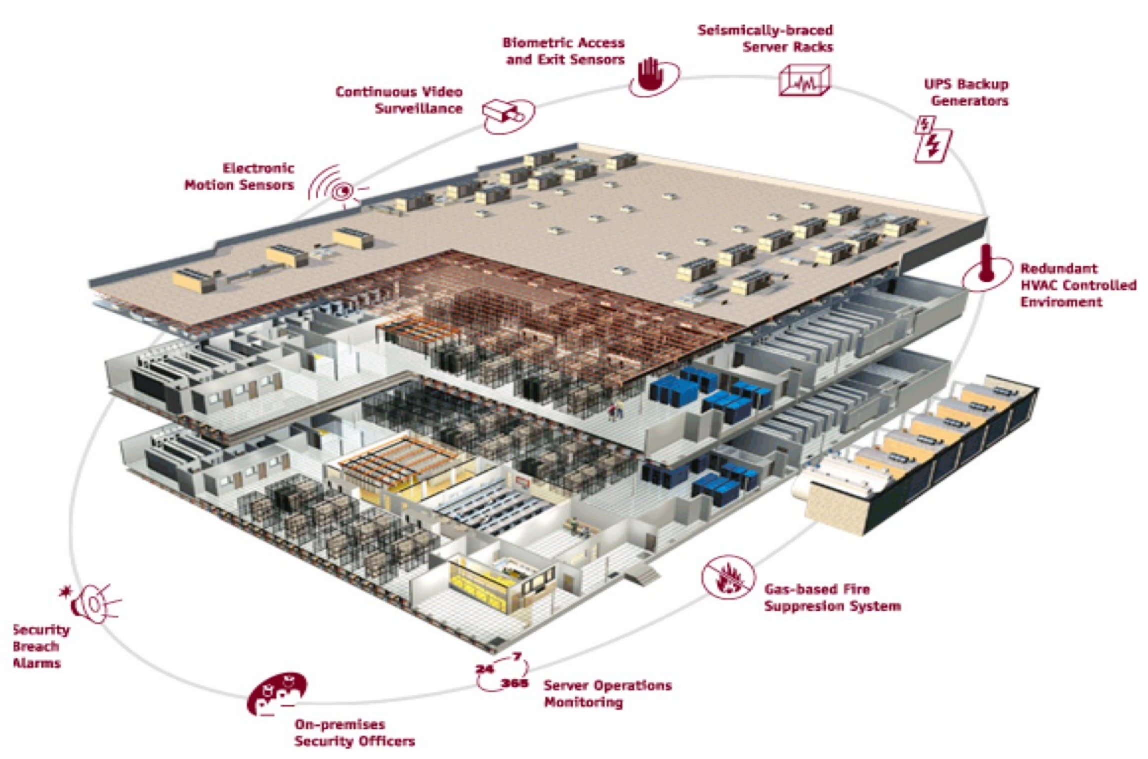
- Capable of high-performance data processing
- More than demand
- Reliability requirement

“Small” Computer Inside



“Big ” Computer Inside

- Data center: a "warehouse" supercomputer


Biometric Access and Exit Sensors
Seismically-braced Server Racks
Continuous Video Surveillance
UPS Backup Generators
Electronic Motion Sensors
Redundant HVAC Controlled Enviroment
Gas-based Fire Suppresion System
Security Breach Alarms
24 7 365 Server Operations Monitoring
On-premises Security Officers

Each data center covers an average of about 45,000 square meters and cost about $600 million to build

The data center has 10,000 square meters

- 占地超过1万平方米的数据中心

俄勒冈州康瑟尔布拉夫斯数据中心

- 悬挂在俄勒冈州康瑟尔布拉夫斯数据中心上方的设备架, google数据中心的规模开始成形。 巨大的钢梁不仅支撑整个结构还负责分配电力。

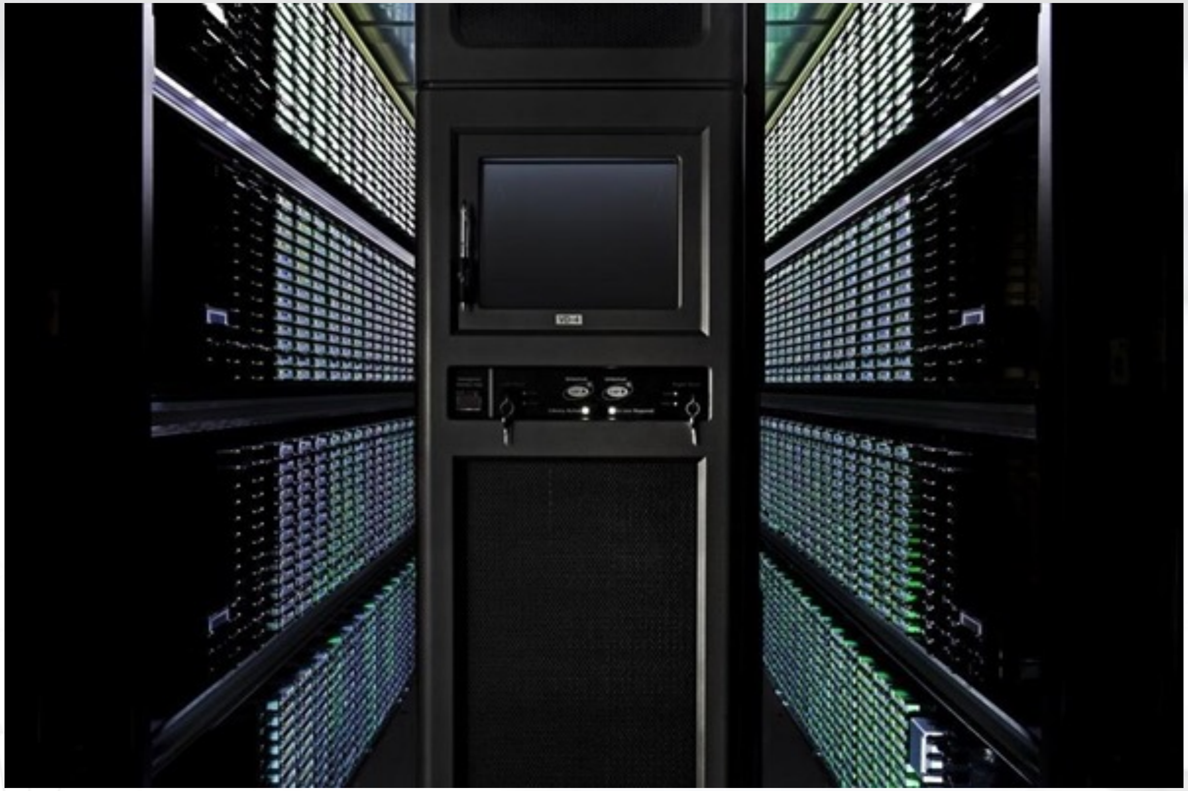
LEDs indicator on the server to monitor the normal operation of the system

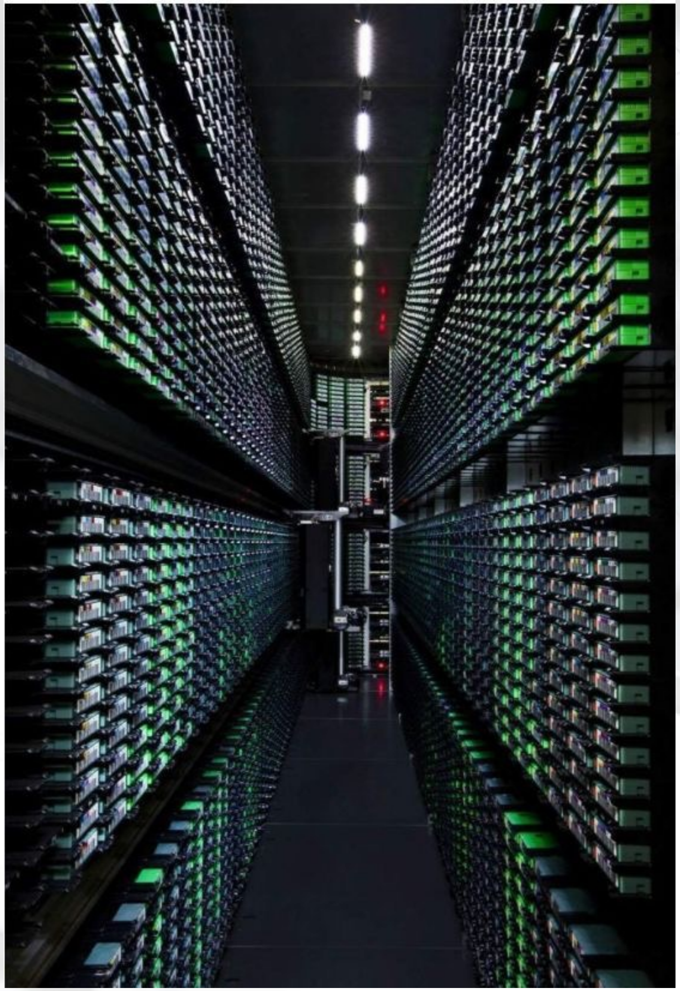
- 蓝色LED灯显示服务器运转良好。使用LED因为它节能、长寿。

High-speed routers and switches built into the data center

- google园区内部的网络机房, 路由器和交换器让各个数据中心可以相互沟通。连接各中心的光纤网络速度比普通家庭的网速快200,000倍。这些光纤电缆沿天花板上的黄色电缆线架安装。

A robotic arm designed to help easily replacing the damaged hard disks





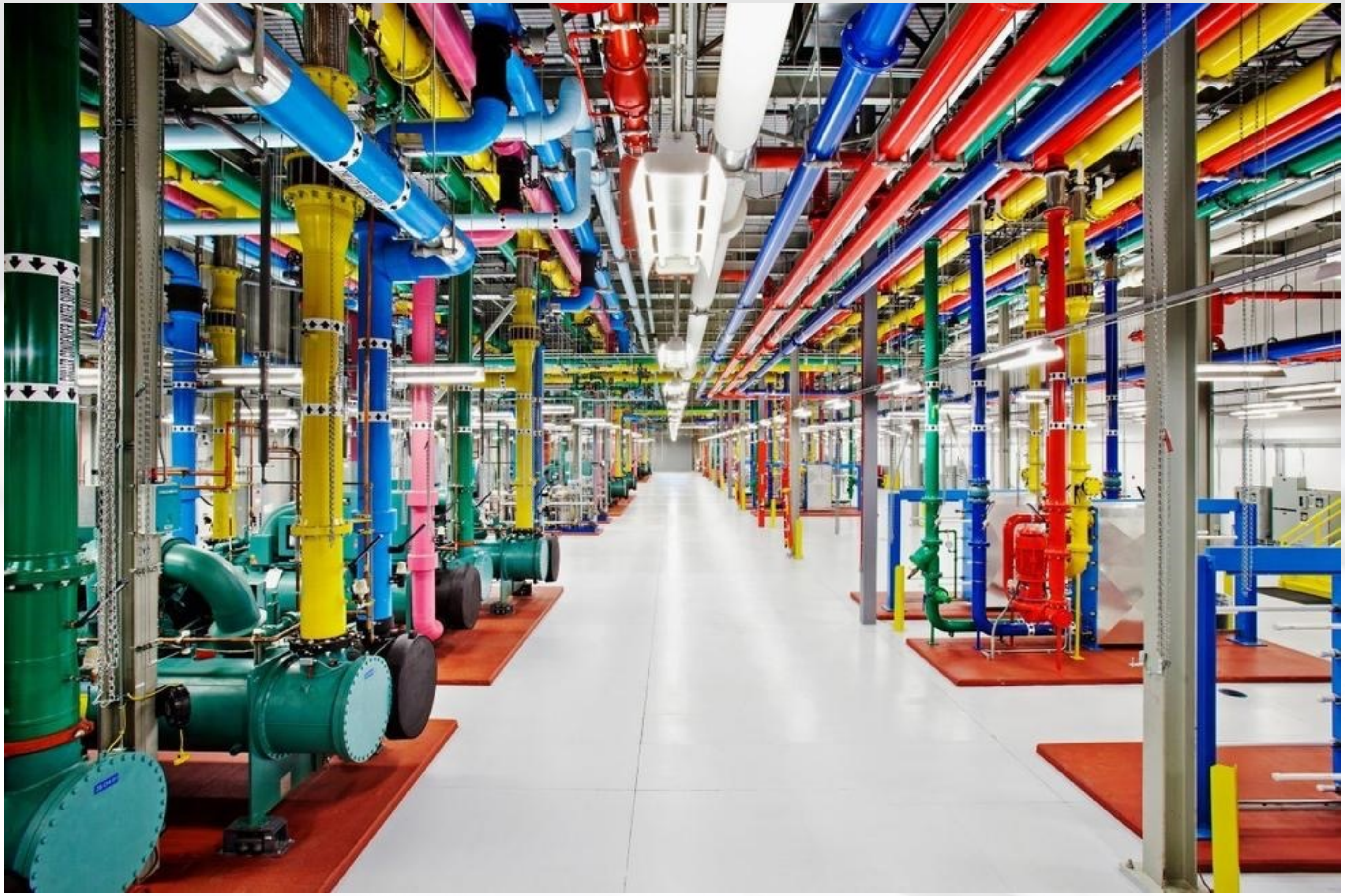
A large air duct used to dissipate heat in a data center

风冷

水冷

油冷

A piping system for dissipating heat in a data center



The TOP500 List (2023, since 1963)

Top 10 supercomputers in the TOP500 List

Rank	System		Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier	US, AMD EPYC, AMD MI250X	8,699,904	1,194.00	1,679.82	22,703
2	Fugaku	JP, A64FX	7,630,848	442.01	537.21	29,899
3	LUMI	Finland, AMD EPYC, AMD MI250X	2,220,288	309.10	428.70	6,016
4	Leonardo	Italy, Intel Xeon, NV A100	1,824,768	238.70	304.47	7,404
5	Summit	US, IBM Power9, NV GV100	2,414,592	148.60	200.79	10,096
6	Sierra	US, IBM Power9, NV GV100	1,572,480	94.64	125.71	7,438
7	Sunway TaihuLight	CN, SW26010	10,649,600	93.01	125.44	15,371
8	Perlmutter	US, AMD EPYC, NV A100	761,856	70.87	93.75	2,589
9	Selene	US, AMD EPYC, NV A100	555,520	63.46	79.22	2,646
10	Tianhe-2A	CN, Intel Xeon, Matrix-2000	4,981,760	61.44	100.68	18,482

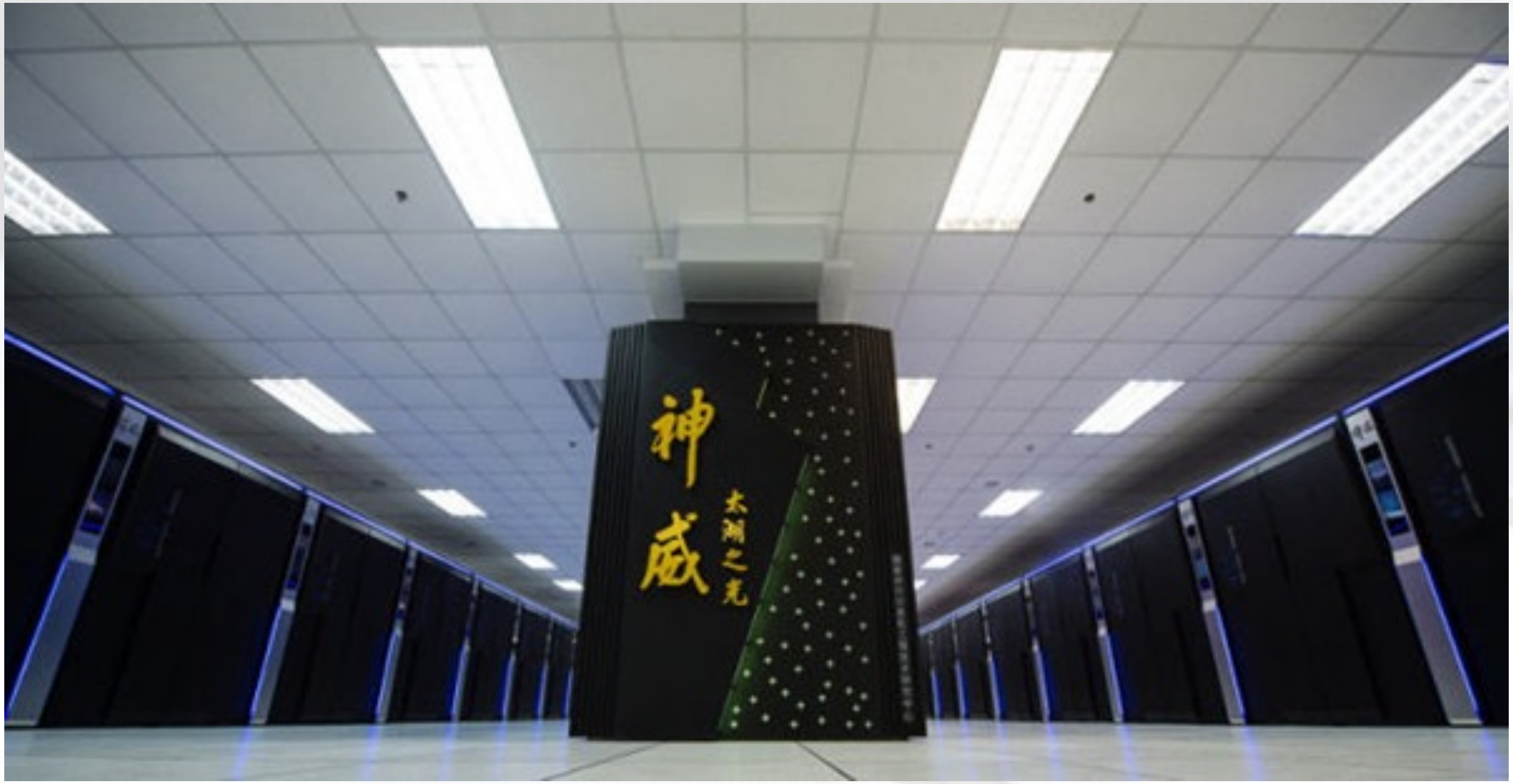
https://www.top500.org/

The TOP500 List (2024, since 1963)

Top 10 supercomputers in the TOP500 List

Rank	System		Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier	US, AMD EPYC, AMD MI250X	8,699,904	1,206.00	1,714.81	22,786
2	Aurora	US, Intel Xeon, Intel GPU	9,264,128	1,012.00	1,980.01	38,698
3	Eagle	US, Intel Xeon, NV H100	2,073,600	561.20	846.84	
4	Fugaku	JP, A64FX	7,630,848	442.01	537.21	29,899
5	LUMI	Finland, AMD EPYC, AMD MI250X	2,752,704	379.70	531.51	7,107
6	Alps	Switzerland, Cray, NV GH200	1,305,600	270.00	353.75	5,194
7	Leonardo	Italy, Intel Xeon, NV A100	1,824,768	241.20	306.31	7,494
8	MareNostrum 5 ACC	Spain, Intel Xeon, NV H100	663,040	175.30	249.44	4,159
9	Summit	US, IBM Power9, NV GV100	2,414,592	148.60	200.79	10,096
10	Eos NVIDIA DGX SuperPOD	US, Intel Xeon, NV DGX H100	485,888	121.40	188.65	

Sunway TaihuLight
- June 2016, National Supercomputing Wuxi Center



Sunway TaihuLight
- June 2016, National Supercomputing Wuxi Center

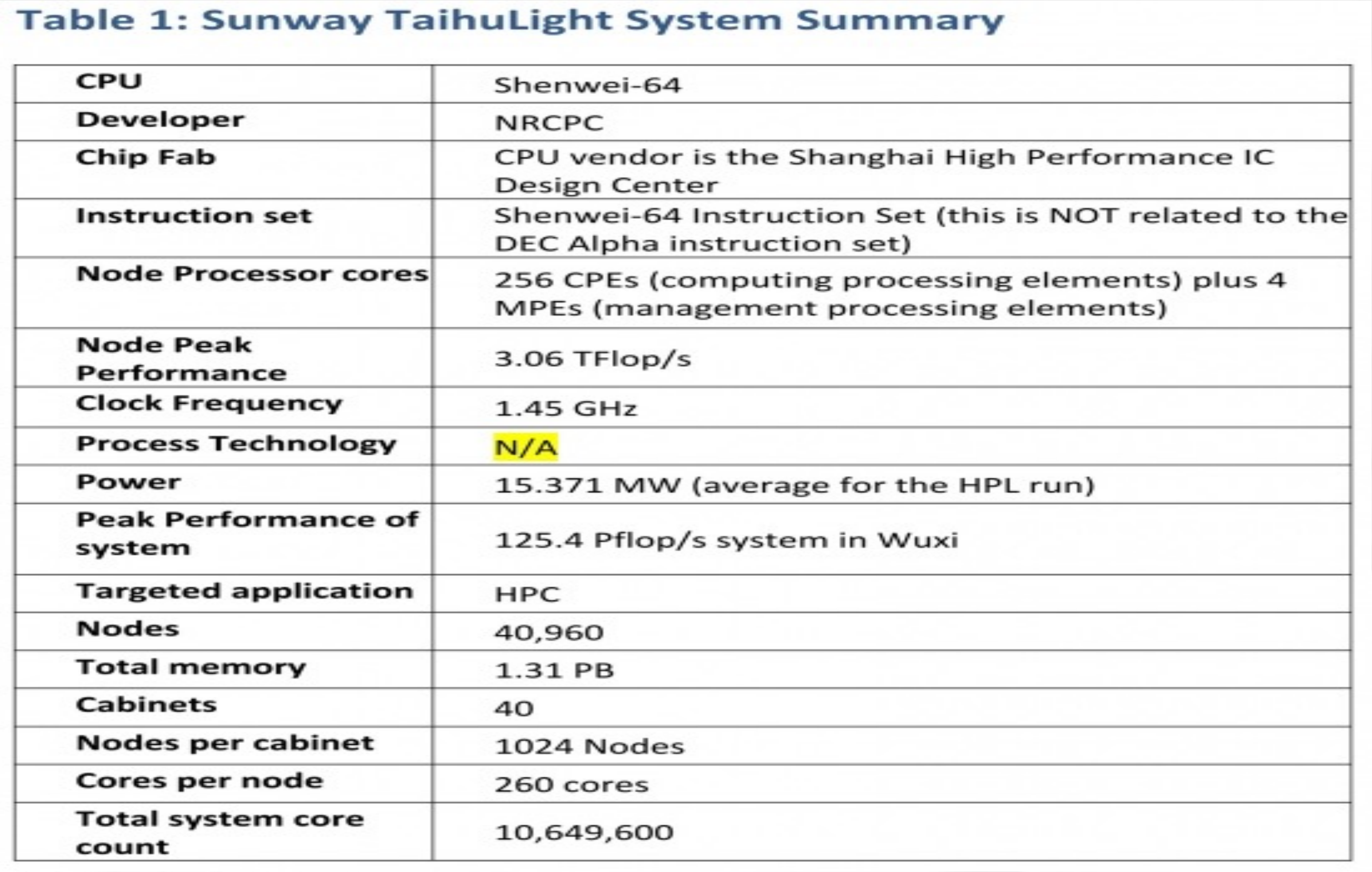
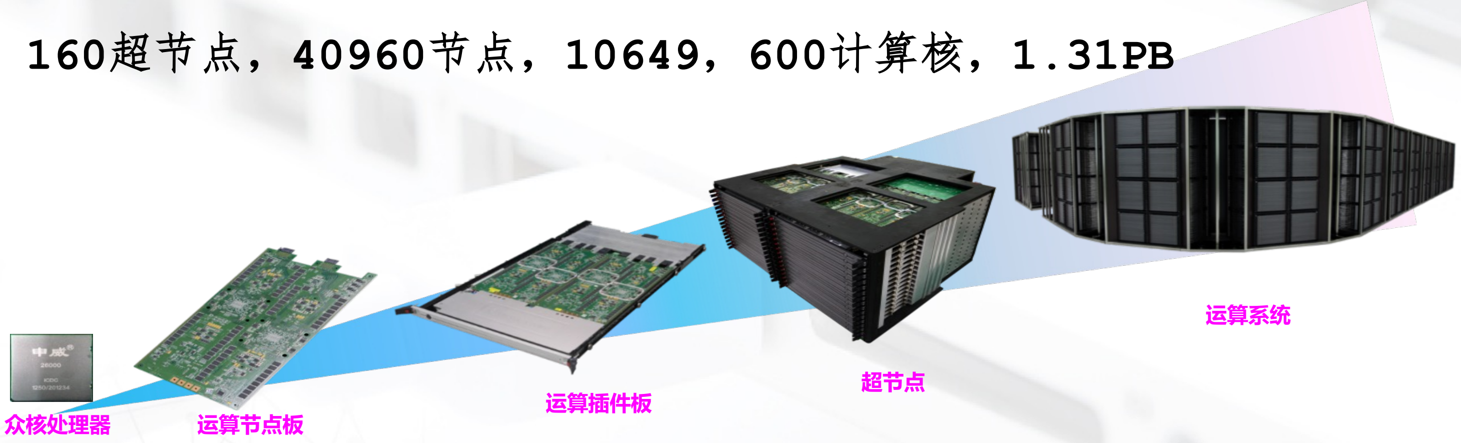


Table 1: Sunway TaihuLight System Summary

CPU	Shenwei-64
Developer	NRCPC
Chip Fab	CPU vendor is the Shanghai High Performance IC Design Center
Instruction set	Shenwei-64 Instruction Set (this is NOT related to the DEC Alpha instruction set)
Node Processor cores	256 CPEs (computing processing elements) plus 4 MPEs (management processing elements)
Node Peak Performance	3.06 TFlop/s
Clock Frequency	1.45 GHz
Process Technology	N/A
Power	15.371 MW (average for the HPL run)
Peak Performance of system	125.4 Pflop/s system in Wuxi
Targeted application	HPC
Nodes	40,960
Total memory	1.31 PB
Cabinets	40
Nodes per cabinet	1024 Nodes
Cores per node	260 cores
Total system core count	10,649,600

How are "very large" computer systems built?
-Sunway TaihuLight

- ■运算节点板
  - ●1CPU，260计算核，系统基本构成单元，众核处理器+存储器
- ■运算插件板
  - ●4 运算节点板高密组装，4CPU。运算节点+网络接口板
- ■运算超节点
  - ●64 运算插件板，256CPU，超节点内部采用紧耦合弹性互连
- ■运算机仓（Cabinet ）
  - ●4运算超节点，256运算插件板，1024运算节点（CPU ）
- ■整机系统
  - ●40运算机仓，160超节点，40960节点，10649，600计算核，1.31PB



Today’ s Dominant Target Systems

- **Mobile (smartphone/tablet)**
  - >1 billion sold/year
  - Market dominated by ARM-ISA-compatible general-purpose processor in system-on-a-chip (SoC)
  - Plus sea of custom accelerators (radio, image, video, graphics, audio, motion, location, security, etc.)
- **Warehouse-Scale Computers (WSCs)**
  - 10,000,000’s cores per warehouse
  - Market dominated by x86-compatible server chips
  - Dedicated apps, plus cloud hosting of virtual machines
  - Now seeing increasing use of GPUs, FPGAs, custom hardware to accelerate workloads
- **Embedded Computing**
  - Wired/wireless network infrastructure, printers
  - Consumer TV/Music/Games/Automotive/Camera/MP3
  - Internet of Things!

Outline

0 Course Crew

1 What’ s the difference between “Big” and “Small” ?

2 **Why Take This Course?**

3 Great Ideas in Computing Systems

4 Course Implementation

Why Take This Course?

- **Foundational Goal**

  Deeply understand Intersects all aspects of computing system

- **Preparatory/Complementary**
  - Algorithm and data structures
  - Programming Language
  - Compilers and Interpreters
  - Operating Systems
  - Digital Systems Organization and Design
  - Mathematical Foundations of CS

- **Fun!!!**
  - Who wouldn't want to understand the magic?

由来已久的问题

■存在问题

- ●缺乏一门独立的能够贯穿整个计算机系统的基础课程
- ●缺少大课时的硬课：无法给课程的深度提供空间
- ●课程之间的衔接和关联不够：无法给课程的宽度提供空间
- ●课程内容比较陈旧
- ●课程体系缺乏对系统设计和应用能力培养的整体考虑

■世界一流计算机专业给我们的启示

- ●用什么来衡量计算机教育做得好的高校？
- ●计算机教育影响力最大的美国高校
- ●院校排名(2019年US News的最新排名)

当代计算机专业学生最重要的核心竞争力是什么？

四大专业基本能力		能力点数
计算思维能力		9
算法设计与分析能力		8
程序设计与实现能力		3
系统能力	系统认知	6
	系统设计	19
	系统开发	23
	系统应用	14

中国科大新一轮计算机专业教学改革：强调“系统观”

■计算机学科的发展趋势

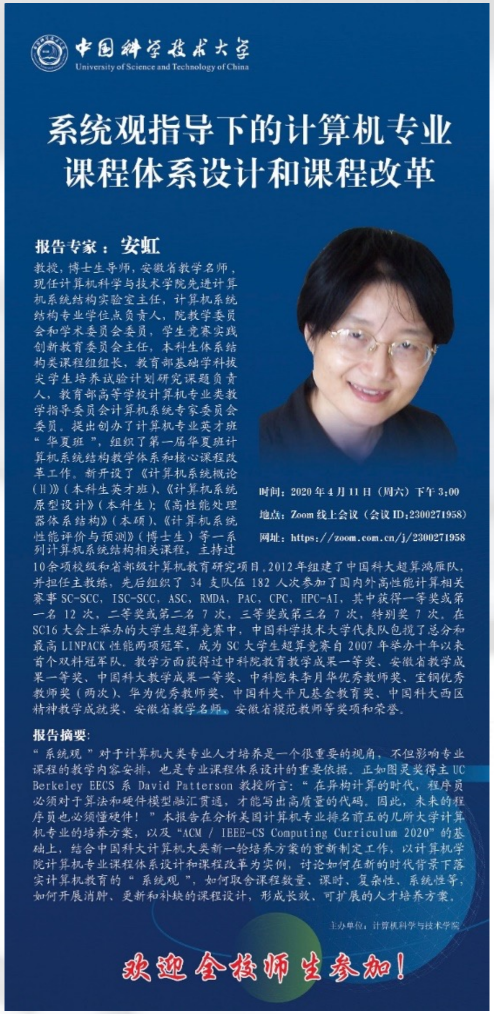
- ●计算X学：越来越多的跨学科应用，并行无处不在
- ●人工智能+：无处不在
- ●后硅时代，量子芯片和量子计算机越来越成为现实

■中国科大计算机教育的目标和理念、特色和优势

- ●计算机系统
- ●高性能计算
- ●人工智能
- ●量子计算

■改革思路

- ●用“系统观”来指导计算机专业培养方案和课程体系的设计
- ●增加一门计算机系统基础课程：《计算系统概论》
- ●重新组合的核心课程，给课程的深度留出空间
- ●调整和更新方向课，拓展课程的宽度



中国科学技术大学
University of Science and Technology of China

系统观指导下的计算机专业
课程体系设计和课程改革

报告专家：安虹

教授，博士生导师，安徽省教学名师，现任计算机科学与技术学院先进计算机系统结构实验室主任，计算机系统结构专业学位点负责人，院教学委员会和学术委员会委员，学生竞赛实践创新教育委员会主任，本科生体系结构类课程组组长，教育部基础学科拔尖学生培养试验计划研究课题负责人，教育部高等学校计算机专业类教学指导委员会计算机系统专家委员会委员。提出创办了计算机专业英才班“华夏班”，组织了第一届华夏班计算机系统结构教学体系和核心课程改革工作。新开设了《计算机系统概论(H)》(本科生英才班)、《计算机系统原型设计》(本科生)；《高性能处理器体系结构》(本硕)、《计算机系统性能评价与预测》(博士生)等一系列计算机系统结构相关课程，主持过10余项校级和省部级计算机教育研究项目.2012年组建了中国科大超算鸿雁队，并担任主教练，先后组织了 34 支队伍 182 人次参加了国内外高性能计算相关赛事 SC-SCC，ISC-SCC，ASC，RMDA，PAC，CPC，HPC-AI，其中获得一等奖或第一名 12 次，二等奖或第二名 7 次，三等奖或第三名 7 次，特别奖 7 次。在 SC16 大会上举办的大学生超算竞赛中，中国科学技术大学代表队包揽了总分和最高 LINPACK 性能两项冠军，成为 SC 大学生超算竞赛自 2007 年举办十年以来首个双料冠军队。教学方面获得过中科院教育教学成果一等奖、安徽省教学成果一等奖、中国科大教学成果一等奖、中科院朱李月华优秀教师奖、宝钢优秀教师奖（两次）、华为优秀教师奖、中国科大平凡基金教育奖、中国科大西区精神教学成就奖、安徽省教学名师、安徽省模范教师等奖项和荣誉。

时间：2020年4月11日（周六）下午3:00

地点：Zoom 线上会议（会议 ID:2300271958）

网址：https://zoom.com.cn/j/2300271958

报告摘要:

“系统观”对于计算机大类专业人才培养是一个很重要的视角，不但影响专业课程的教学内容安排，也是专业课程体系设计的重要依据。正如图灵奖得主 UC Berkeley EECS 系 David Patterson 教授所言：“在异构计算的时代，程序员必须对于算法和硬件模型融汇贯通，才能写出高质量的代码。因此，未来的程序员也必须懂硬件！”本报告在分析美国计算机专业排名前五的几所大学计算机专业的培养方案，以及“ACM / IEEE-CS Computing Curriculum 2020”的基础上，结合中国科大计算机大类新一轮培养方案的重新制定工作，以计算机学院计算机专业课程体系设计和课程改革为实例，讨论如何在新的时代背景下落实计算机教育的“系统观”，如何取舍课程数量、课时、复杂性、系统性等，如何开展消肿、更新和补缺的课程设计，形成长效、可扩展的人才培养方案。

主办单位：计算机科学与技术学院

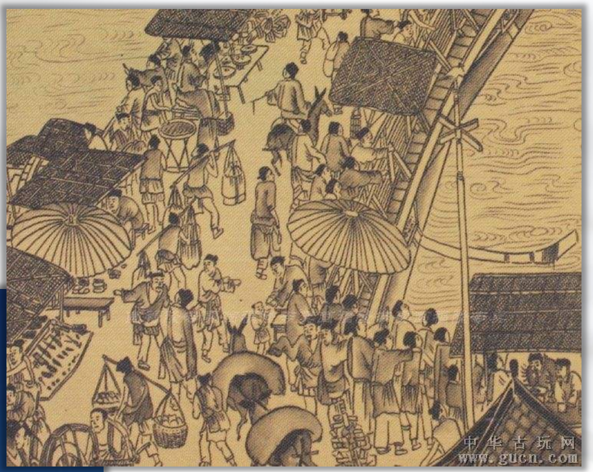
欢迎全校师生参加！

要从本科低年级开始逐步培养学生的**计算思维**能力

计算思维：利用包括网络在内的计算系统进行问题求解（自然问题、技术问题和社会问题）的思维方式

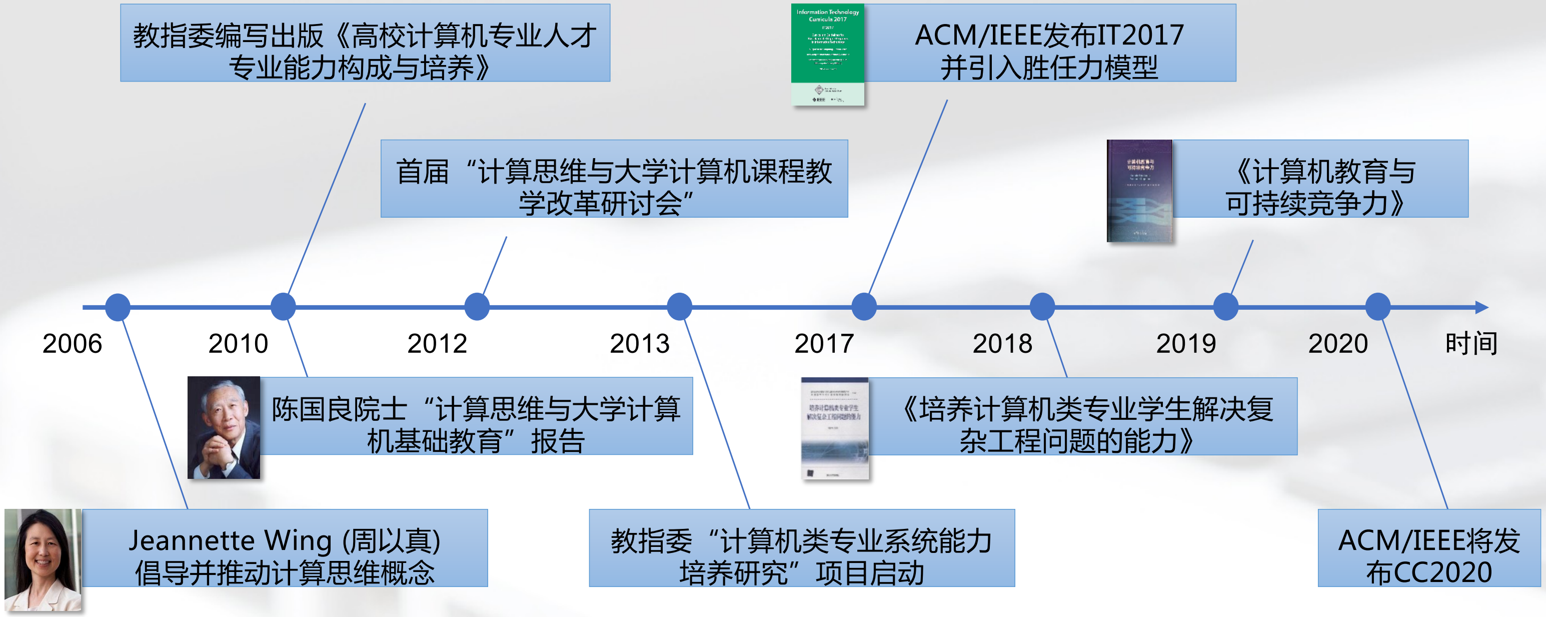
自然

技术



社会

计算思维的发展历程(内涵与外延)



Jeannette Wing (周以真)
倡导并推动计算思维概念

陈国良院士“计算思维与大学计算机基础教育”报告

教指委编写出版《高校计算机专业人才专业能力构成与培养》

首届“计算思维与大学计算机课程教学改革研讨会”

教指委“计算机类专业系统能力培养研究”项目启动

ACM/IEEE发布IT2017
并引入胜任力模型

《培养计算机类专业学生解决复杂工程问题的能力》

《计算机教育与可持续竞争力》

ACM/IEEE将发布CC2020

“在异构计算的时代程序员必须对于算法和硬件模型融汇贯通，才能写出高质量的代码。因此，未来的程序员也必须懂硬件！”

——图灵奖得主David Patterson

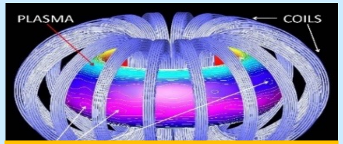
什么是计算思维能力？

不同的历史时期，人类关注的重大问题不同

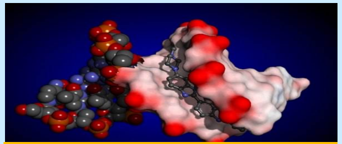
- **20世纪**：**战争、人口增长、饥饿**等
- **21世纪**：自然资源消耗过快、**环境污染**、气候异常、**健康医疗**、人口老龄化、贫富差距过大、城市交通、非传统**安全问题**（非典型传染病爆发、金融危机、恐怖主义、网络攻击等）

科学与工程计算

量子计算模拟



可控核聚变模拟


蛋白质与药物设计

大气环境监测

大数据，人工智能和云计算

健康大数据

人工智能

基因分析

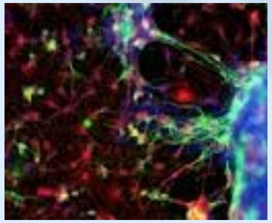
金融分析

Artificial intelligence is everywhere

Supercomputers

Business analytics

AlphaGo



Drug design

Data Centers

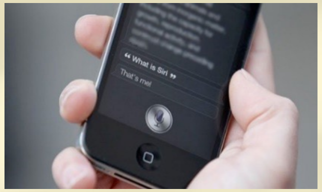
Bai du 百度

Ad prediction

YouTube

Automatic translation

Smartphones

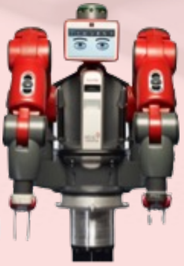


Audio recognition

Image analysis

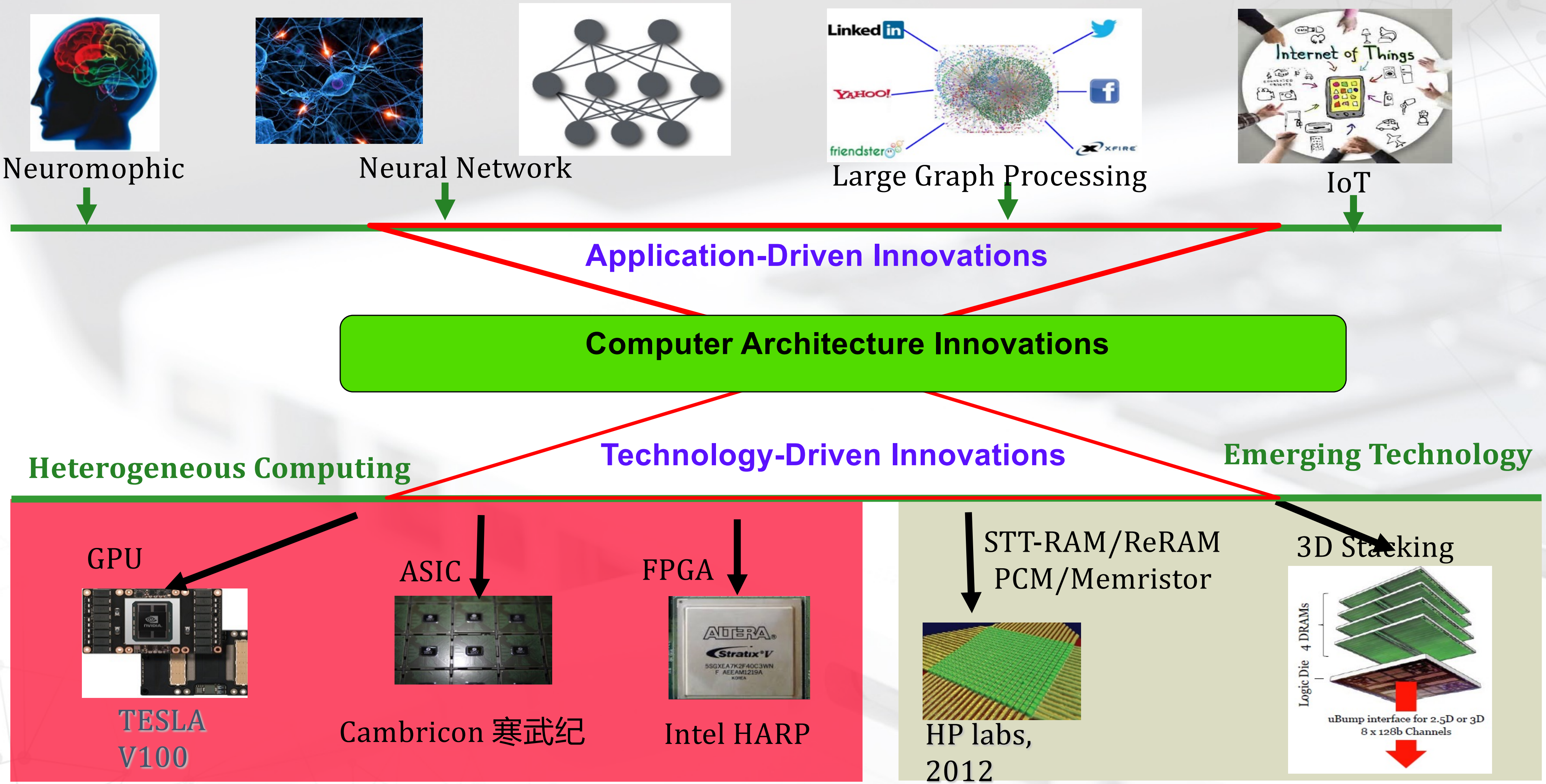
Embedded Devices

Robotics



Consumer electronics

New applications demand new structures all the time



Neuromophic

Neural Network

Large Graph Processing

IoT

Application-Driven Innovations

Computer Architecture Innovations

Heterogeneous Computing

Technology-Driven Innovations

Emerging Technology

GPU

TESLA V100

ASIC

Cambricon 寒武纪

FPGA

Intel HARP

STT-RAM/ReRAM PCM/Memristor

HP labs, 2012

3D Stacking

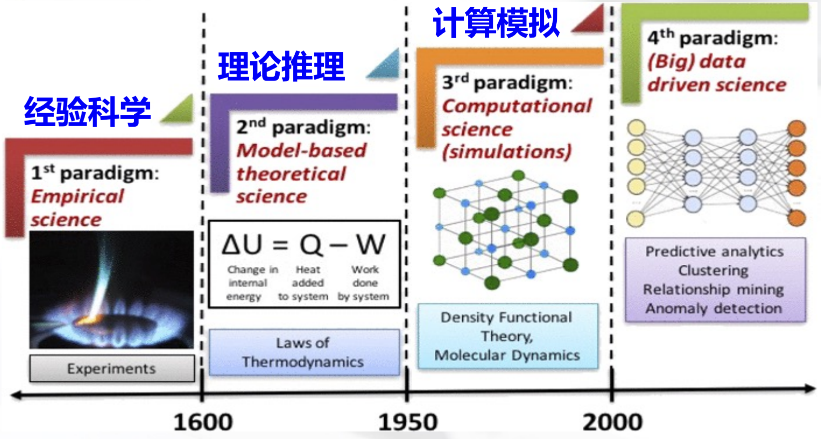
人类如何实现从物理设备到问题求解的？

Application

Physics

差距太大，一步无法跨越！

也有例外，例如罗盘


数据科学
经验科学
1st paradigm: Empirical science
Experiments
理论推理
2nd paradigm: Model-based theoretical science
ΔU = Q – W
Change in internal energy
Heat added to system
Work done by system
Laws of Thermodynamics
计算模拟
3rd paradigm: Computational science (simulations)
Density Functional Theory, Molecular Dynamics
4th paradigm: (Big) data driven science
Predictive analytics
Clustering
Relationship mining
Anomaly detection
1600
1950
2000

Outline

0 Course Crew

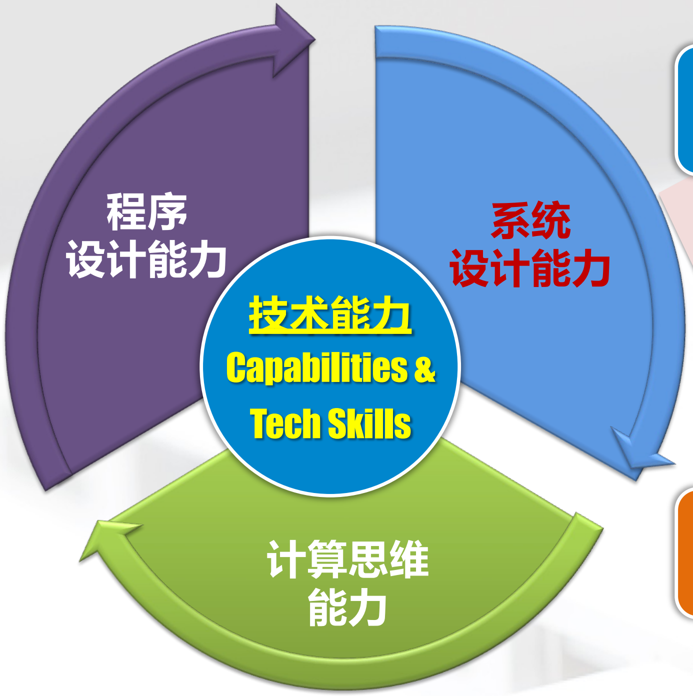
1 What’ s the difference between “Big” and “Small” ?

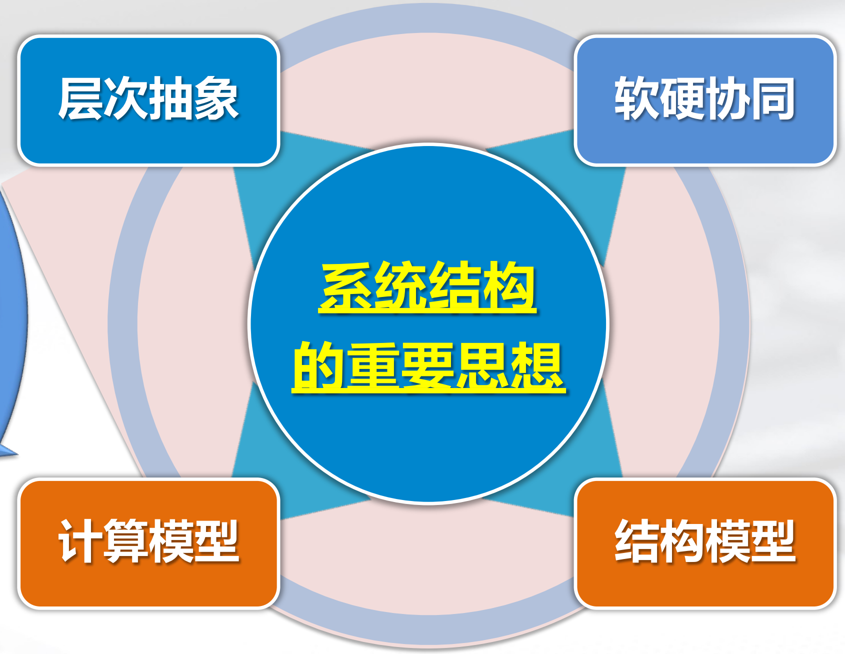
2 Why Take This Course?

3 **Great Ideas in Computing Systems**

4 Course Implementation

课程目标：理解计算机系统结构的重要思想





计算思维能力
（算法）

程序设计能力
（编程）

系统设计能力
（结构）

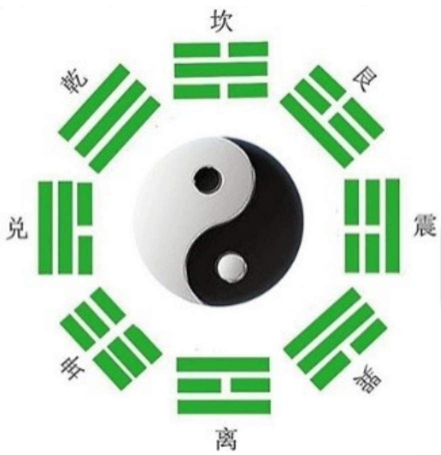
Great Ideas in Computing Systems in This Courses

- **Great Idea #0: Great Idea from Ancient Chinese Philosophy(Bits and Bytes)**
- **Great Idea #1: Computer is an Universal Computing Device(Turing Machine Model)**
- **Great Idea #2: Stored program computer(Von Neumann Model)**
- **Great Idea #3: Abstraction Helps Us Manage Complexity(Layers of Representation/Interpretation)**
- **Great Idea #4: Software and Hardware Co-design**

Great Idea from Ancient Chinese Philosophy

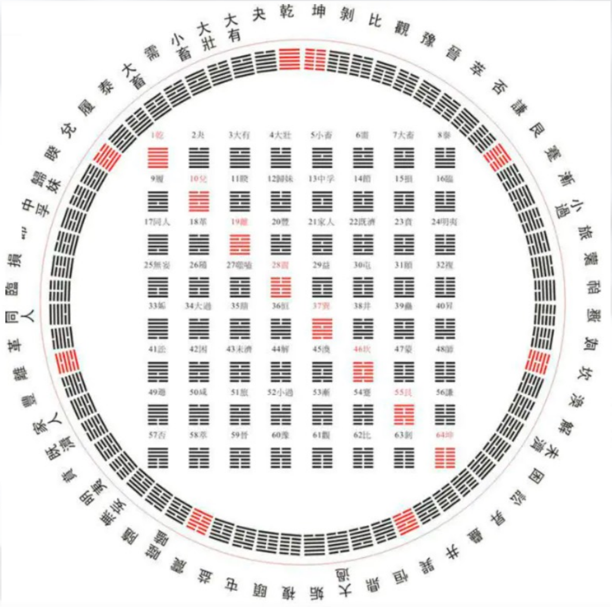
All things come into being, all things come into nothing

天下万物生于有, 有生于无 **《老子•四十章》**

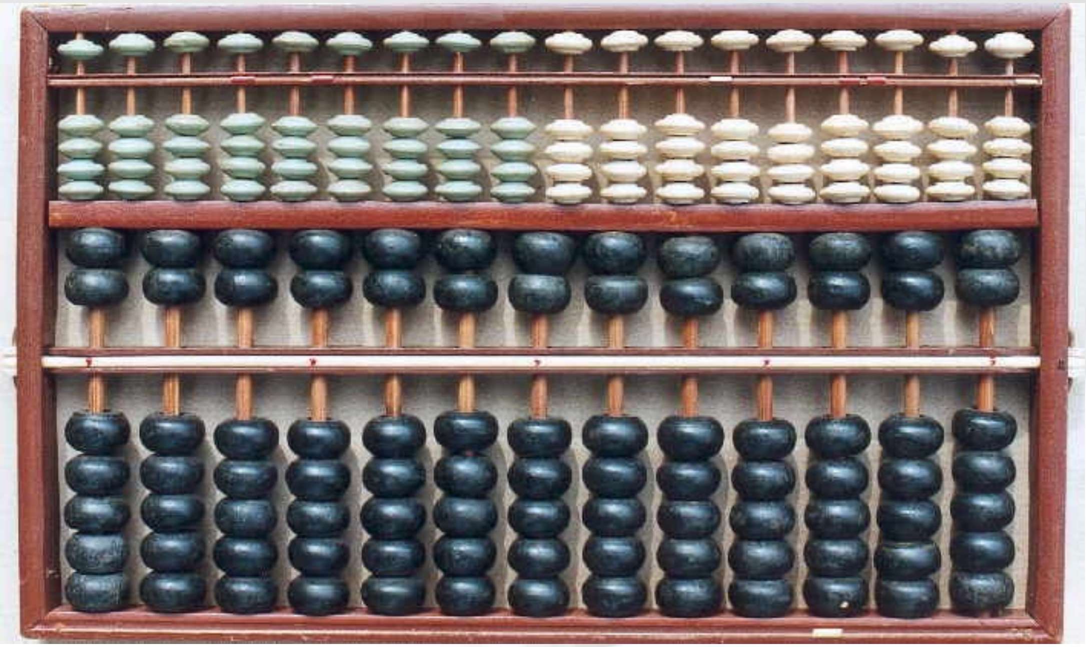


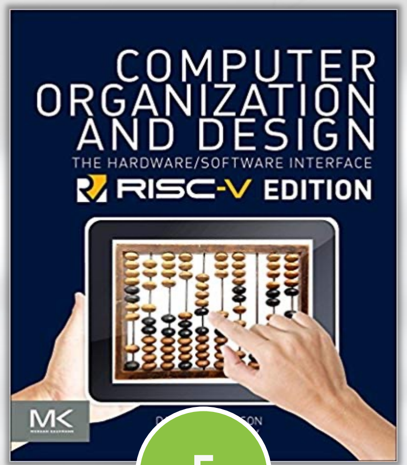
《易经》

太极生两仪，
两仪生四象，
四象生八卦，
八卦演万物。



Great Idea from Ancient Chinese Philosophy





COMPUTER ORGANIZATION AND DESIGN

THE HARDWARE/SOFTWARE INTERFACE

RISC-V EDITION

MK

5

Computer Organization and Design: The Hardware/Software Interface **,**

David A Patterson, John L. Hennessy, 5th edition. Morgan Kaufmann Publishers, Inc. , 2017

A computing tool that does not use electricity

- Abacus (公元前500年，中国)

Abacus
China
c. 1970
Loan of Gwen and Gordon
Bell, B1643.01

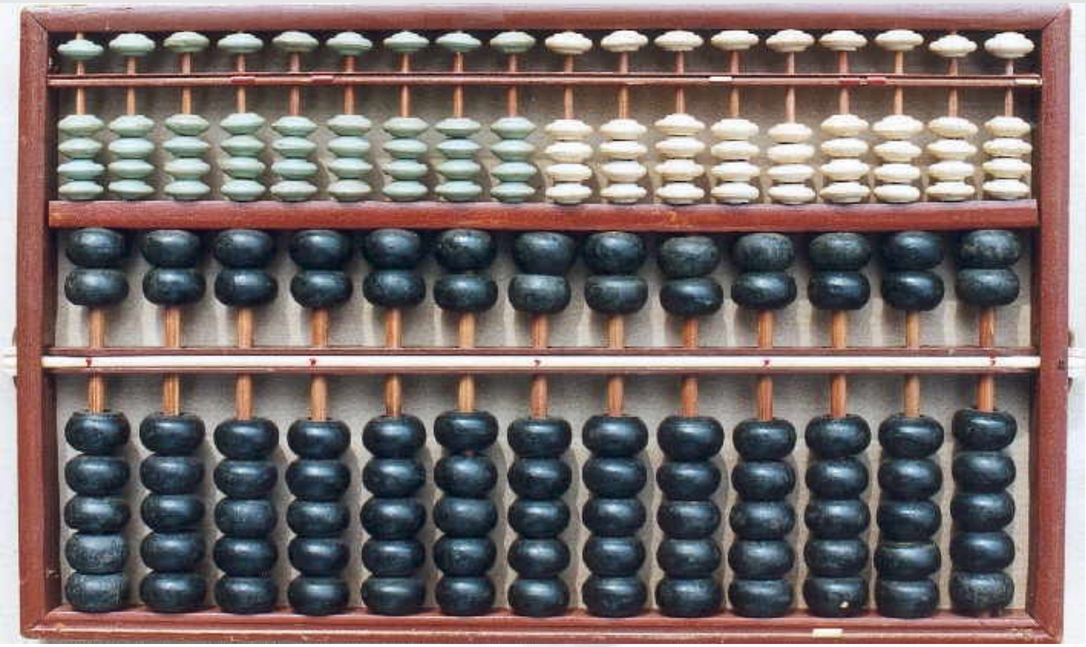
Table abacus
(reproduction) and
jetons
Germany
17th century
Loan of Michael R.
Williams, L2003.3.2

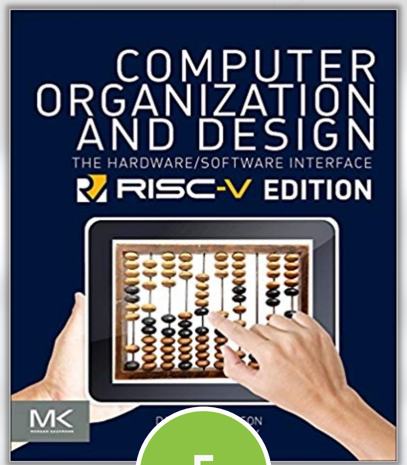
Soroban
Japan
c. 1960
Loan of Gw
Bell, B1659.01

Counting Frame
Early 20th century
Gift of Gwen and
Gordon Bell, B141.80

Schoty
Russia
Early 20th century
Gift of Warren Yogi, 102

Great Idea from Ancient Chinese Philosophy




COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
RISC-V EDITION
MK

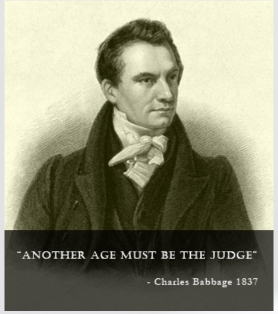
5

Computer Organization and Design: The Hardware/Software Interface ,

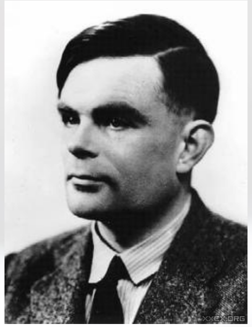
David A Patterson, John L. Hennessy, 5th edition. Morgan Kaufmann Publishers, Inc. , 2017

Is an abacus a computer?

Automatic computing equipment: from mechanical computer to electronic computer



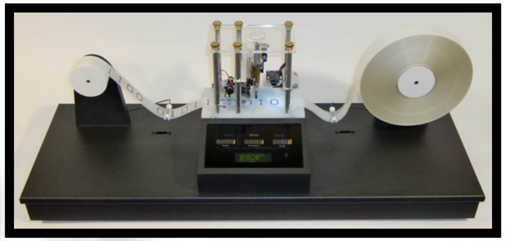
Charles Babbage,
1791 – 1871,England



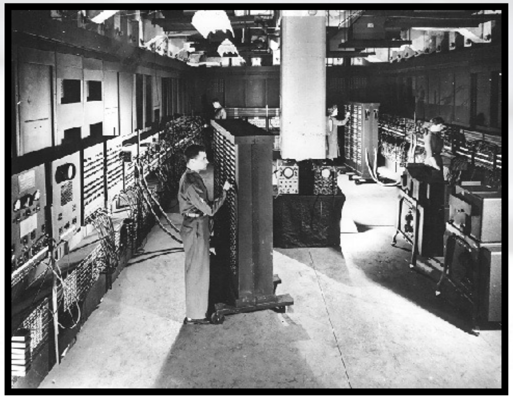
Alan
Turing(24)

Eckert(24) and Mauchly(36)

1832,2002,2008
The Babbage Difference Engine, 17 years, 25,000 parts, 5ton, cost: £17,470



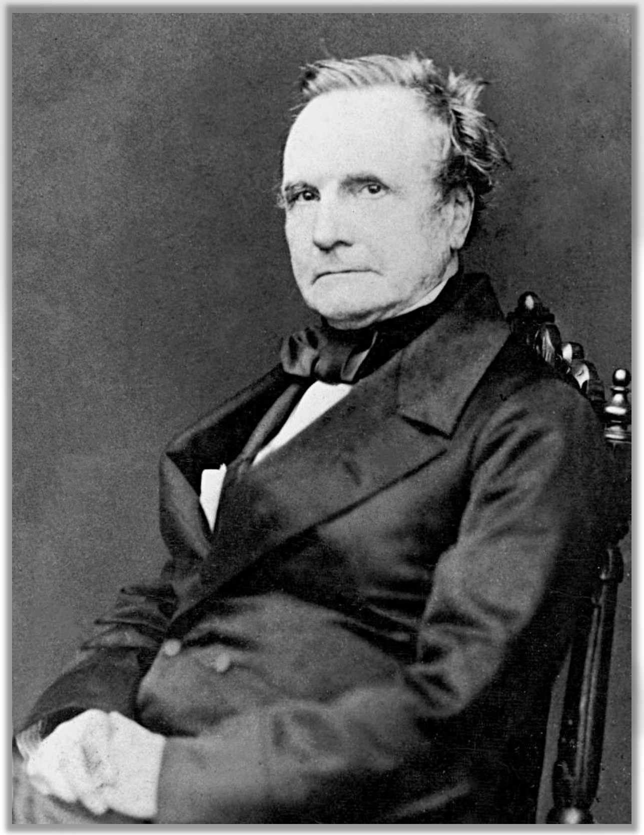
Turing Machine,
1936



ENIAC

1946

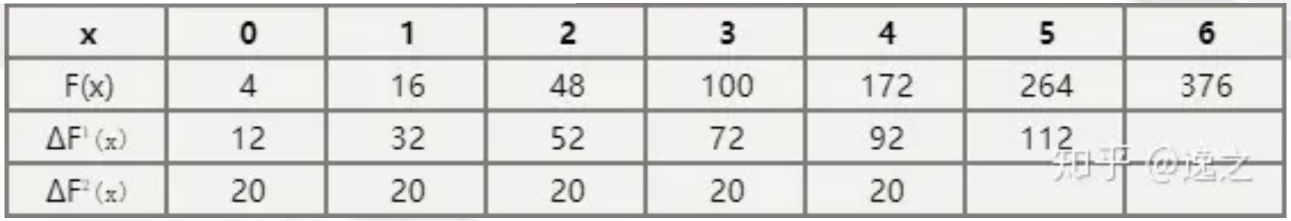
Charles Babbage (1791-1871)： A Fallen Hero!



[Copyright expired and in public domain.
Image obtained from Wikimedia Commons.]

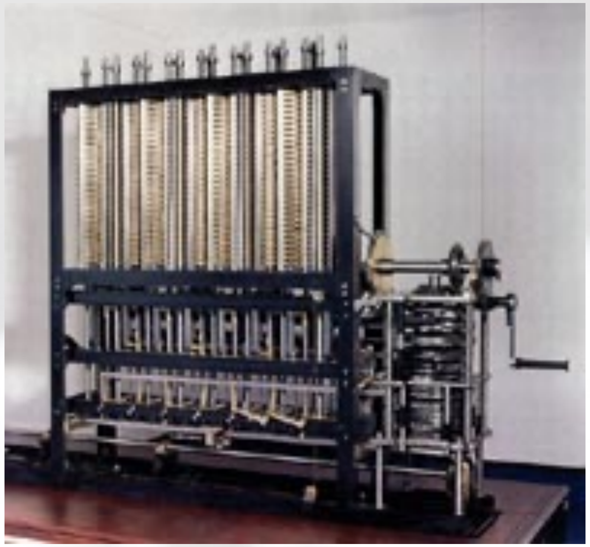
- Lucasian Professor of Mathematics, Cambridge University, 1828-1839
- A true “polymath” with interests in many areas
- Frustrated by errors in printed tables, wanted to build machines to evaluate and print accurate tables
- Inspired by earlier work organizing human “computers” to methodically calculate tables by hand

$$F(x) = 10x^2 + 2x + 4, x \in N$$

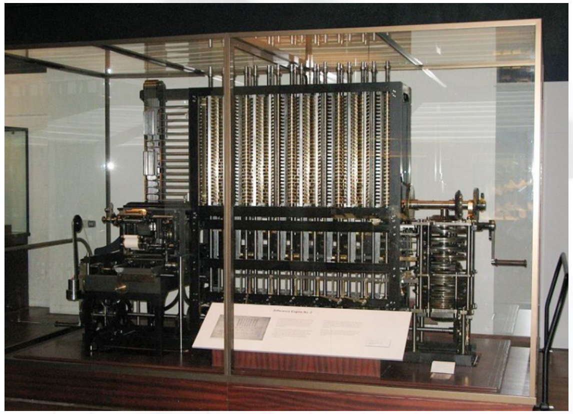


x	0	1	2	3	4	5	6
F(x)	4	16	48	100	172	264	376
ΔF¹(x)	12	32	52	72	92	112	
ΔF²(x)	20	20	20	20	20		

Babbage difference engine: the first mechanical computer (1832)

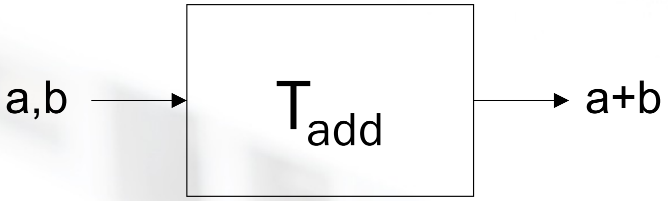


- 2002,2008
- The Babbage
- Difference Engine
- 17 years,
- 25,000 parts, 5ton
- cost: £17,470

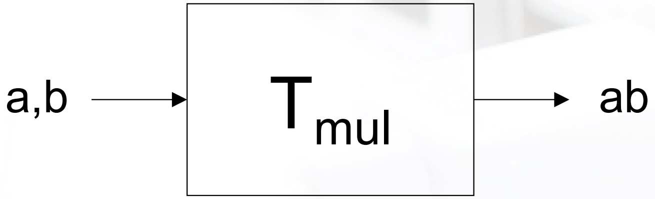


Turing Machine

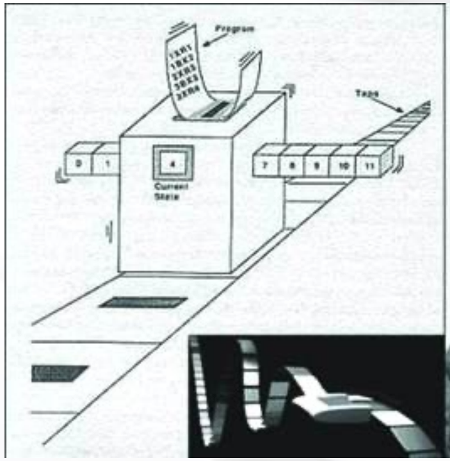
- ■ **Mathematical model of a device that can perform any computation – Alan Turing (1937)**
  - ● `ability to read/write symbols on an infinite “tape”`
  - ● `state transitions, based on current state and symbol`
- ■**Every computation can be performed by some Turing machine.** ***(Turing′ s thesis)***



Turing machine that adds

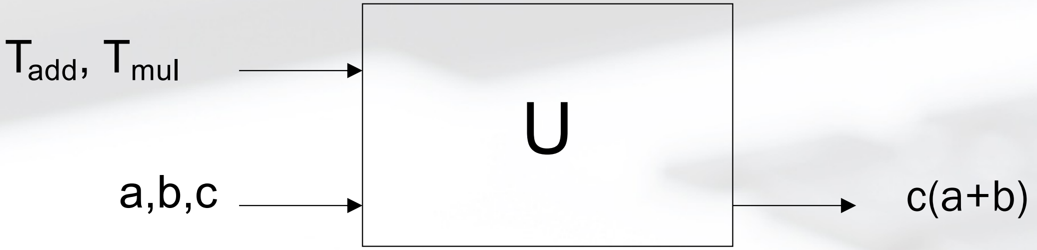


Turing machine that multiplies



Universal Turing Machine

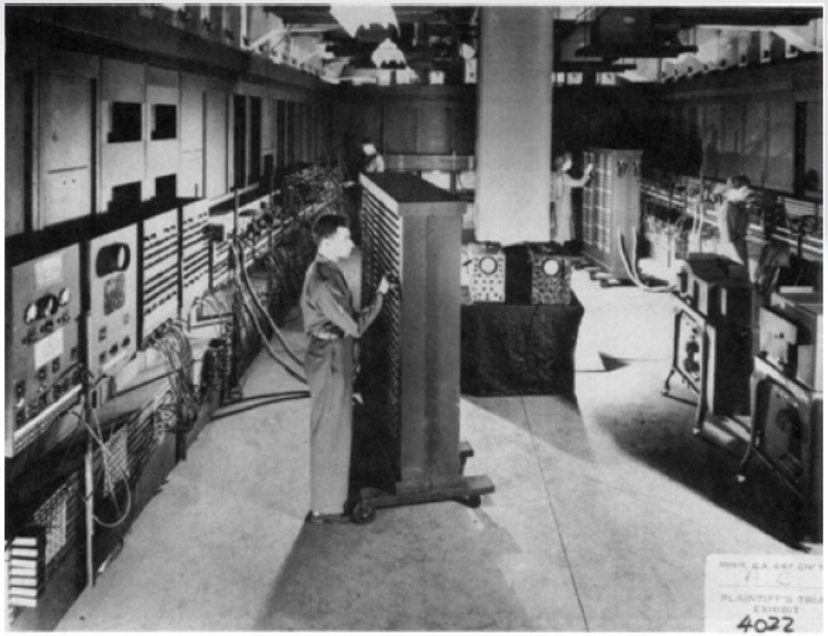
- **Turing described a Turing machine that could implement all other Turing machines.**
  - inputs: data, plus a description of computation (Turing machine)



Universal Turing Machine

- **U is programmable – so is a computer!**
  - instructions are part of the input data
  - a computer can emulate a Universal Turing Machine, and vice versa
- ***Therefore, a computer is a universal computing device!***

ENIAC - The first electronic computer ,1946



1904， The world's first electron tube was born at the hands of the British physicist Fleming

ENIAC(Electrical Numerical Integrator And Calculator)

- 17,468 vacuum tubes
- Power 150kW
- Weighed 30 tons
- Occupied 1800 sq ft
  - —80 feet long
  - —8.5 feet high
- Clock: 100kHz, About 5000 additions per second
- RAM: ~230bytes, Could store 20 numbers Could store 20 numbers in main memory
- IO: punched card
- Cost about $500,000

ENIAC (1946)

- First electronic general-purpose computer
  - Construction started in secret at UPenn Moore School of Electrical Engineering during WWII to calculate firing tables for US Army, designed by Eckert and Mauchly
  - Twelve 10-decimal-digit accumulators
  - Had a conditional branch!
- Programmed by plugboard and switches, time consuming!
- Purely electronic instruction fetch and execution, so fast
  - 10-digit x 10-digit multiply in 2.8ms (2000x faster than Mark-1)
- As a result of speed, it was almost entirely I/O bound
- As a result of large number of tubes, it was often broken (5 days was longest time between failures)

Great Idea #1:
Computer is an Universal Computing Device

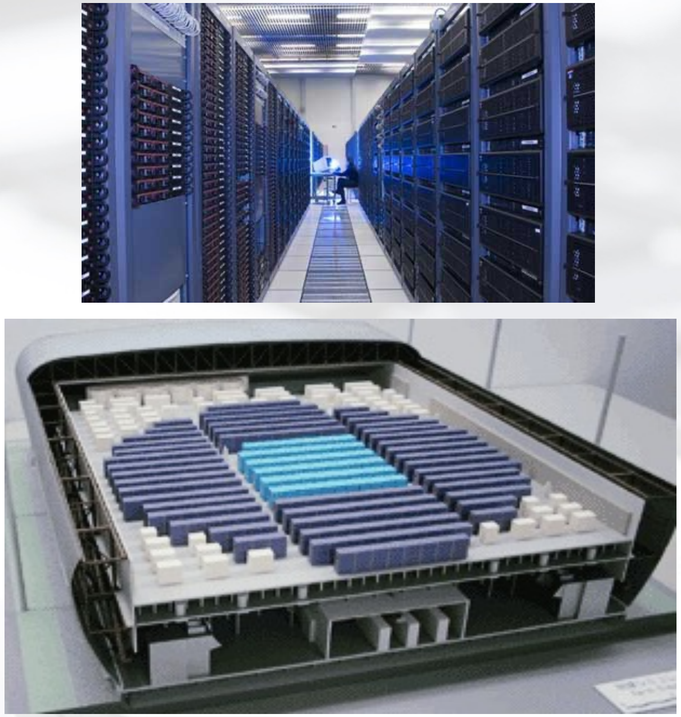
- All computers, given enough time and memory, are capable of computing exactly the same things.

Smart Phones

=

Laptops

=

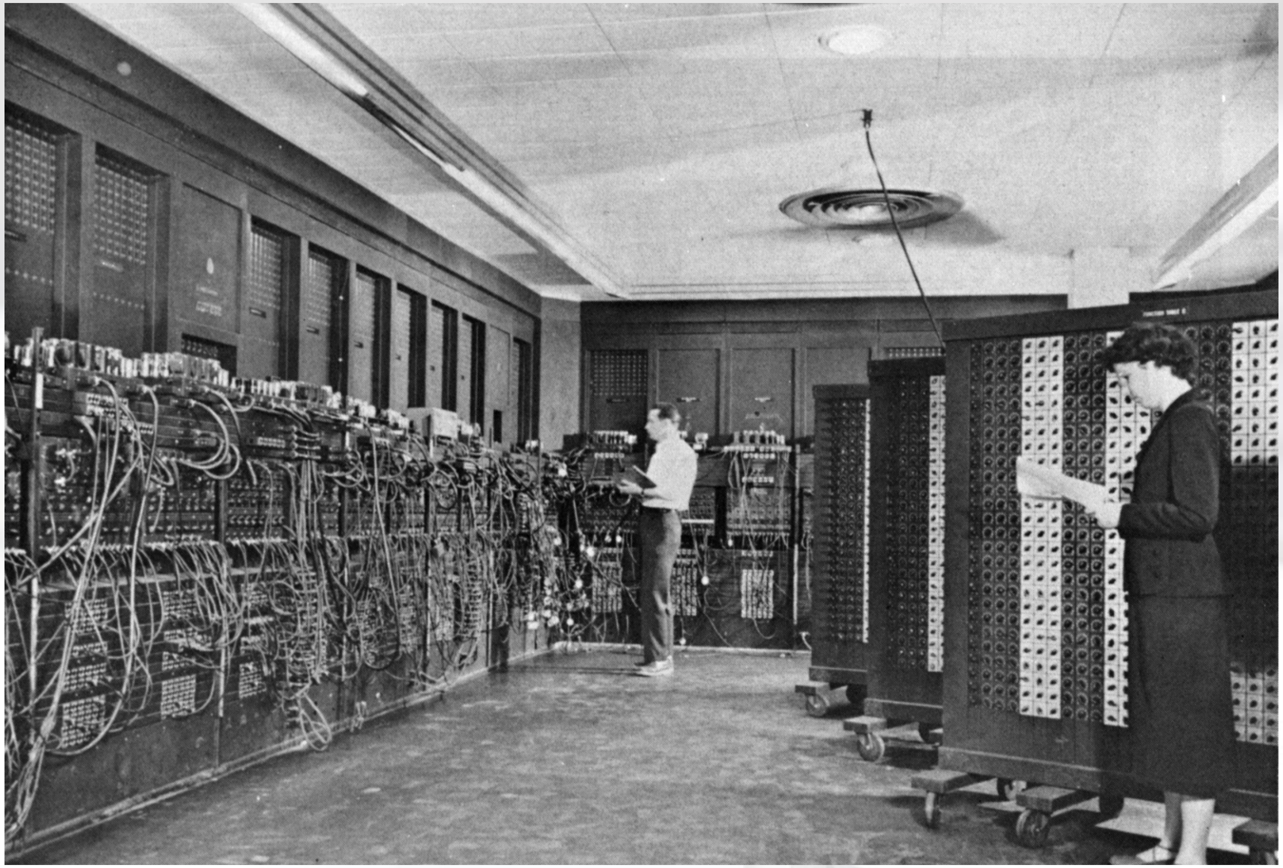


Supercomputers

From Theory to Practice

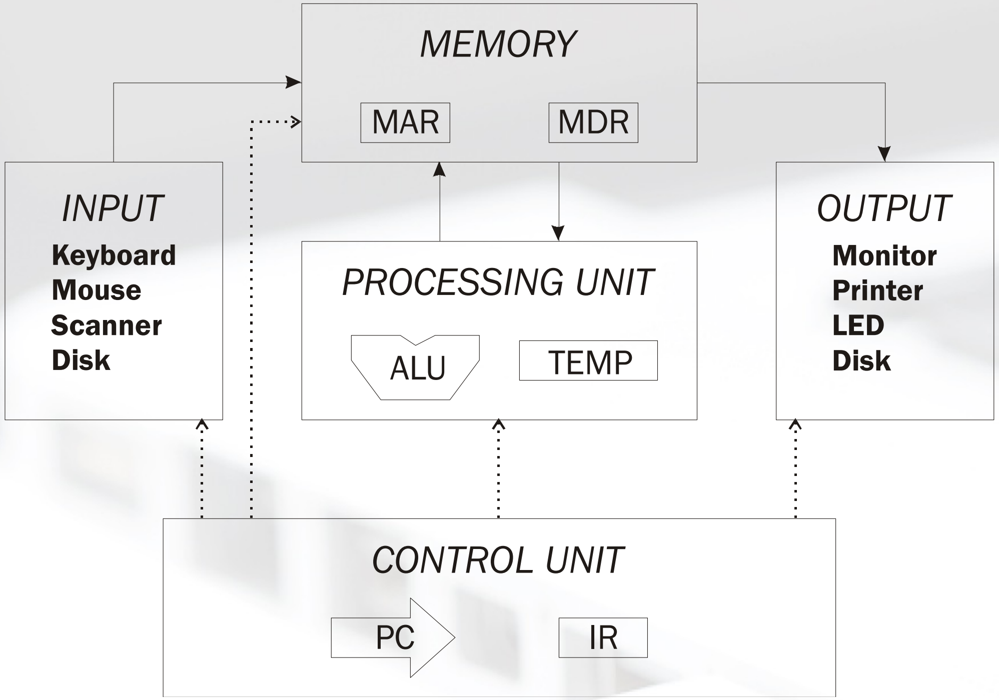
- In theory, computer can *compute* anything
- that’ s possible to compute
  - given enough *memory* and *time*
- In practice, *solving problems* involves computing under constraints.
  - time
    - — weather forecast, next frame of animation, …
  - cost
    - — cell phone, automotive engine controller, …
  - power
    - — cell phone, handheld video game, …

ENIAC



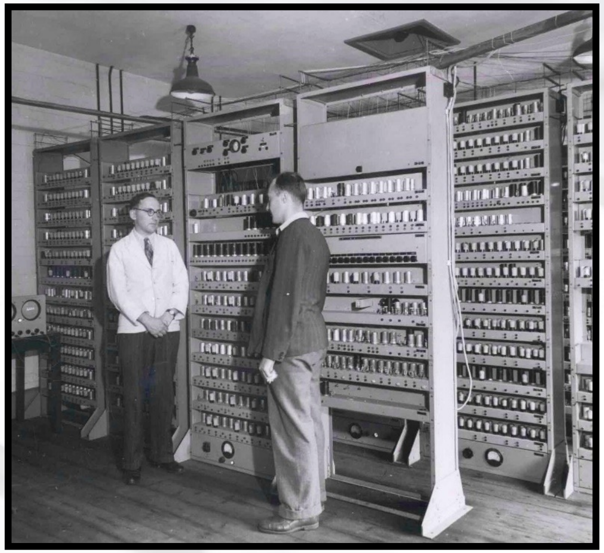
Changing the program could take days!

Great Idea #2
Von Neumann Model(存储程序计算机)



Von Neumann

Maurice
Vincent Wilkes



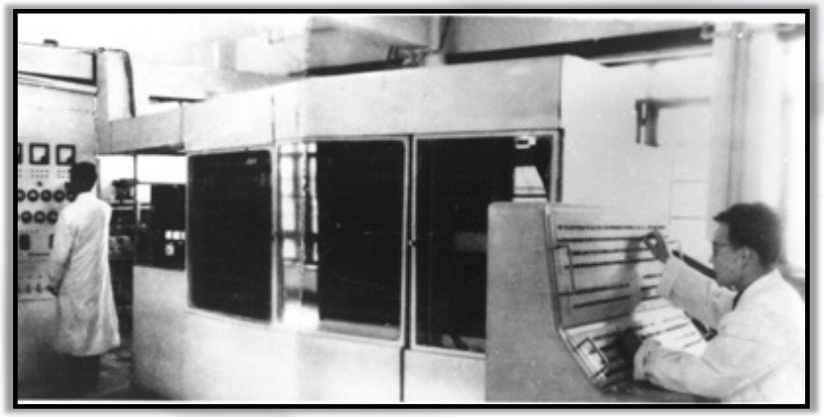
EDSAC, University of Cambridge, UK, 1949

我们的祖上“挺阔的”

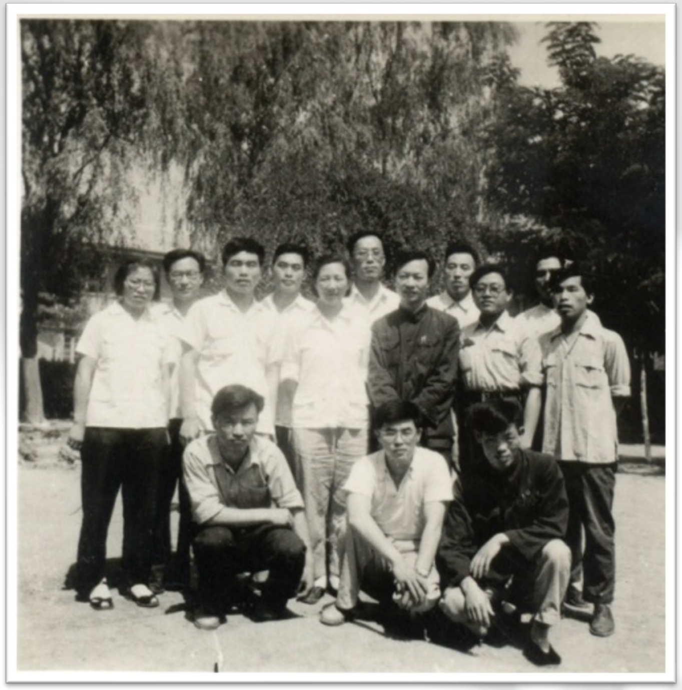
- 中国科大计算机专业创建于1958年, 当时隶属于应用数学和计算技术系
- 应用数学和计算技术系首任系主任：华罗庚
- 计算机专业首任教研室主任：夏培肃
- 中国第一个计算机三人小组
  - 1952年，华罗庚教授会见了三位年轻科学家，讨论的是一个前沿话题：研制中国的计算机。由此中国第一个计算机三人小组成立。他们是夏培肃、闵乃大，王传英。

我校首台计算机：107计算机

- 夏培肃先生主持研制
- 中国第一台自主设计的通用电子计算机
- 中国第一台自主设计的冯•诺依曼结构计算机
  - EDVAC，美国，1945-1952
  - EDSAC，英国，1945-1949
  - 107机，中国，1953-1959
- 1960年在中国科大投入使用（命名为KD-1）
- 1970年随科大下迁至合肥
- 1974年被拆除



首批教师：参与107机研制的教师



夏培肃院士（左六）与我校计算机学科早期建设者、107机研制者、原计算机系统结构教研室主任郑世荣教授、钟津立、周行仁、赵鼎文、杨学良、王武良等在玉泉路校园合影。

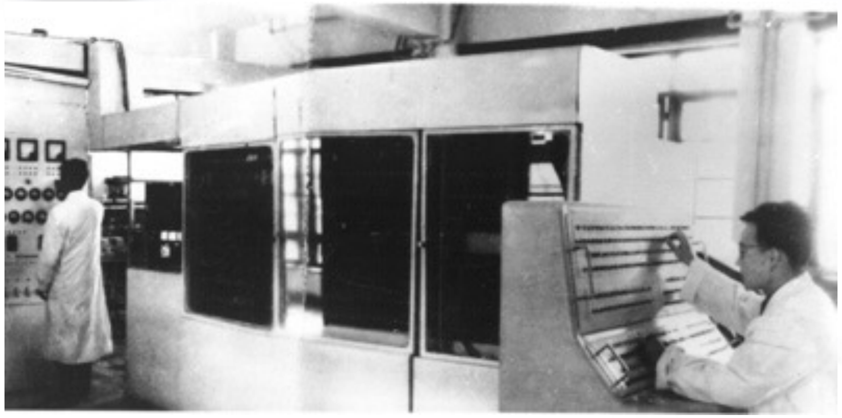
中国科学院
计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY
CHINESE ACADEMY OF SCIENCES
来
50
ICT
195
06
2006/10/25

我校首套计算机教材

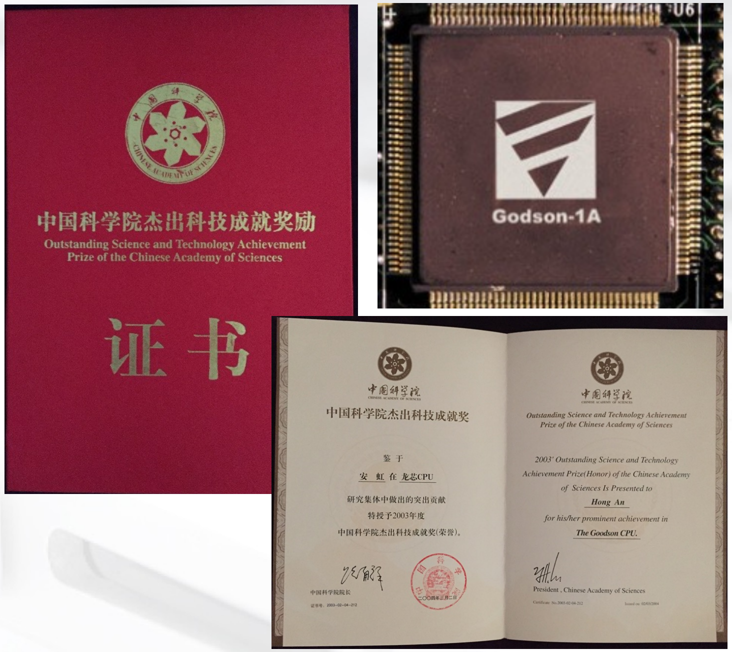
- 夏培肃先生主持编写的我校第一套《计算机原理》教材。也是国内最早的《计算机原理》教材。

The USTCers’ outstanding contributions to the Chinese computer

- 1959年， 107计算机
- 中国第一台自主设计的通用计算机
- 主设计师：夏培肃



- 2002年，龙芯1号
- 中国第一颗自主设计的通用微处理器芯片
- 主设计师：胡伟武(86本)



中国科学院杰出科技成就奖励
Outstanding Science and Technology Achievement
Prize of the Chinese Academy of Sciences

证 书

中国科学院
CHINESE ACADEMY OF SCIENCES

中国科学院杰出科技成就奖

鉴 于

安 虹 在 龙芯CPU

研究集体中做出的突出贡献

特授予2003年度

中国科学院杰出科技成就奖(荣誉)。

中国科学院院长

证书号：2003-02-04-212

二〇〇四年三月二日

中国科学院
CHINESE ACADEMY OF SCIENCES

Outstanding Science and Technology Achievement
Prize of the Chinese Academy of Sciences

2003' Outstanding Science and Technology
Achievement Prize(Honor) of the Chinese Academy
of Sciences Is Presented to

Hong An

for his/her prominent achievement in

The Goodson CPU.

President , Chinese Academy of Sciences

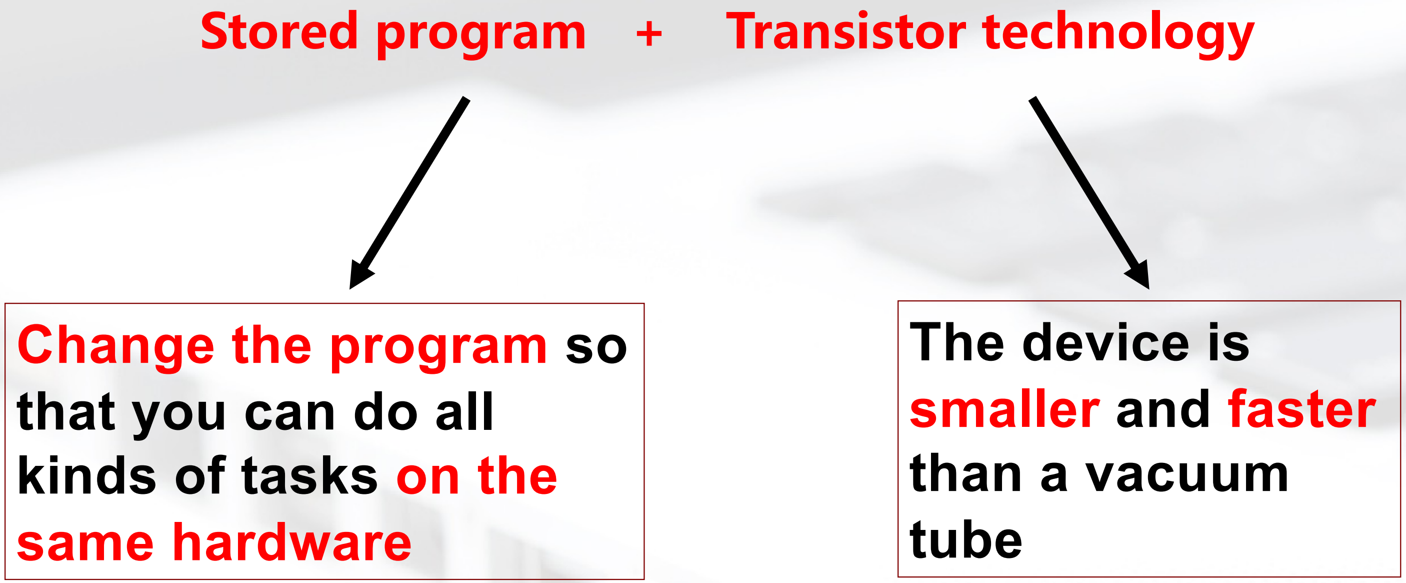
Certificate No.2003-02-04-212 Issued on 02/03/2004

The invention of the transistor

Bell Labs lays the groundwork:

- 1945: Bell sets up lab in the hopes of developing “solid state” components to replace existing electromechanical systems. William Schockley, John Bardeen, Walter Brattain: all solid-state physicists. Focus on Si and Ge.
- 1951: Shockley develops junction transistor which can be manufactured in quantity.
- 1954: The first transistor radio! Also, TI makes first silicon transistor (price $2.50)
- 1956: Bardeen, Shockley, Brattain receive Nobel Prize.

Two major inventions of the microprocessor chip



Stored program + Transistor technology

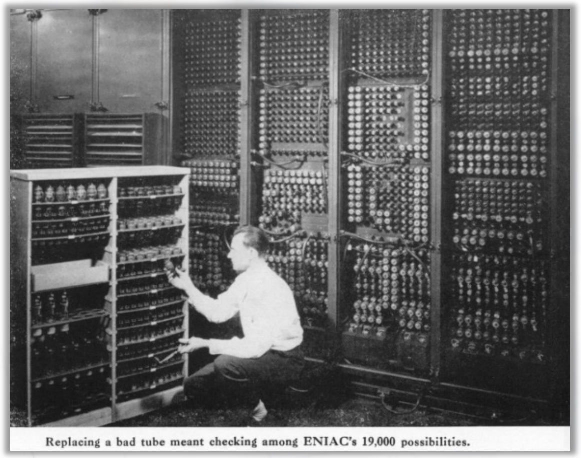
Change the program so that you can do all kinds of tasks on the same hardware

The device is smaller and faster than a vacuum tube

First computer vs. First microprocessor chip

1946 , ENIAC(Electrical Numerical Integrator And Calculator)

- 18000 vacuum tubes
- 1500 relays
- 174 KW
- 30 tons
- 1800 sq. ft. footprint
- Clock: 100kHz
- RAM: ~230bytes
- IO: punched card

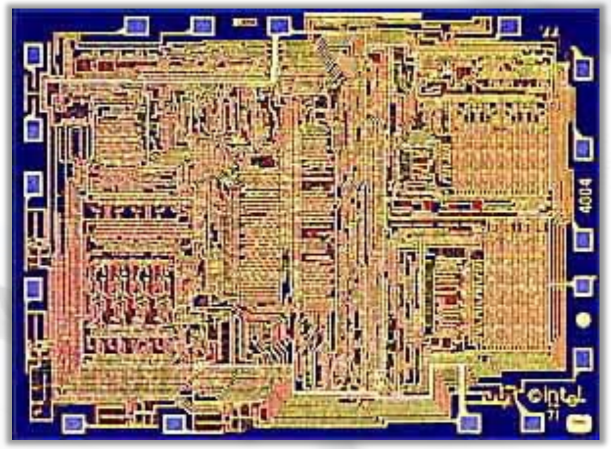


Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

After 25 years

1971, Intel 4004

- 10 micron process，NMOS-Only Logic
- 2,250 transistors
- 3cmx4cm die
- 4-bit bus
- Performance < 0.1 MIPS
- 640 bytes of addressable Memory
- 740 KHz



Thirty years after the first microprocessor chip was born

1971, Intel 4004

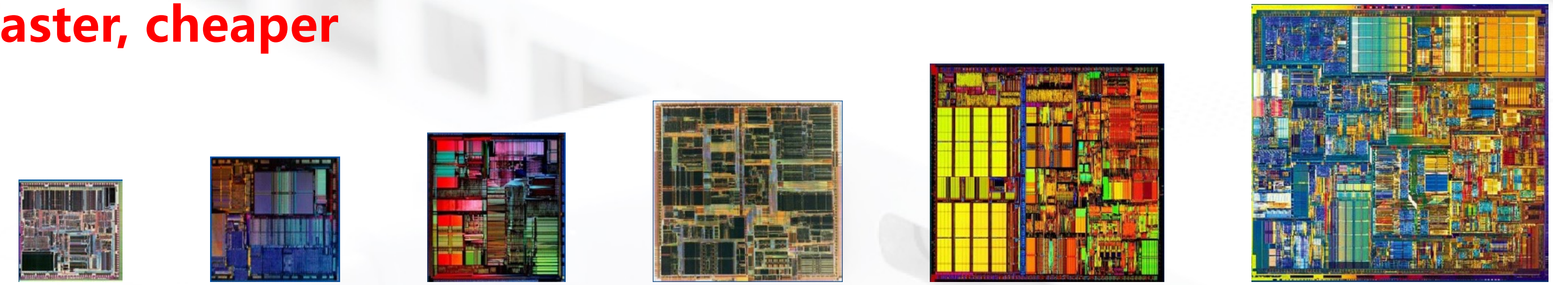
- 10 micron process
- 2,300 transistors
- 3x4 mm die
- 4-bit bus
- 640 bytes of addressable Memory
- 750 KHz

After 30 years

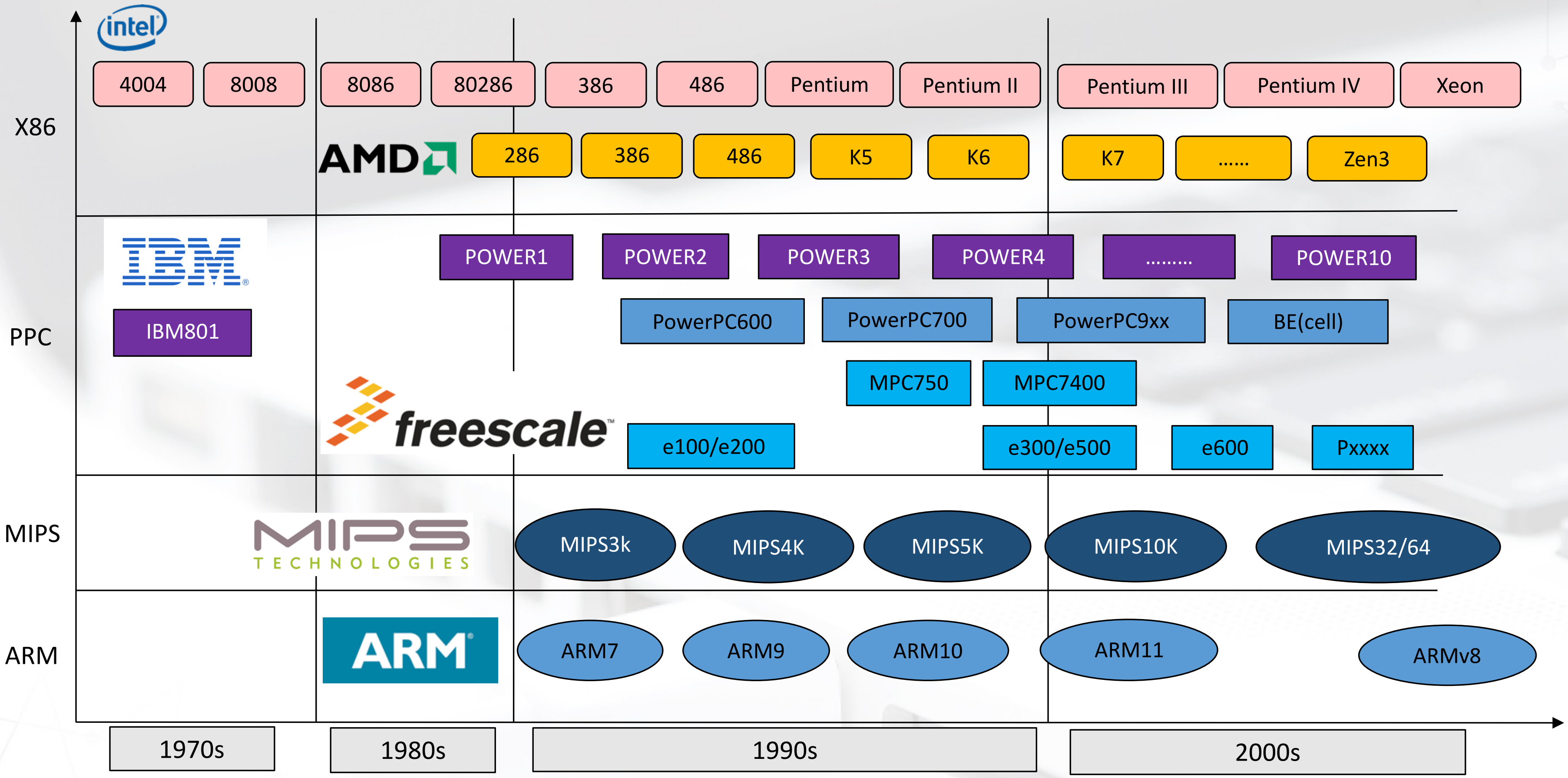
2000, Intel Pentium IV

- Issues up to 5 uOPs per cycle
- MMX, SSE, and SSE2
- 0.18 micron process
- 42 million transistors
- 217 mm die
- 64-bit bus
- 8KB D-cache, 12KB op trace cache (I-cache), 256KB L2 cache
- 1.4 GHz

Performance improved 5000x: smaller, faster, cheaper



Thirty years after the first microprocessor chip was born



X86

intel: 4004, 8008, 8086, 80286, 386, 486, Pentium, Pentium II, Pentium III, Pentium IV, Xeon

AMD: 286, 386, 486, K5, K6, K7, ……, Zen3

PPC

IBM: IBM801, POWER1, POWER2, POWER3, POWER4, ………, POWER10

PowerPC600, PowerPC700, PowerPC9xx, BE(cell)

MPC750, MPC7400

freescale: e100/e200, e300/e500, e600, Pxxxx

MIPS

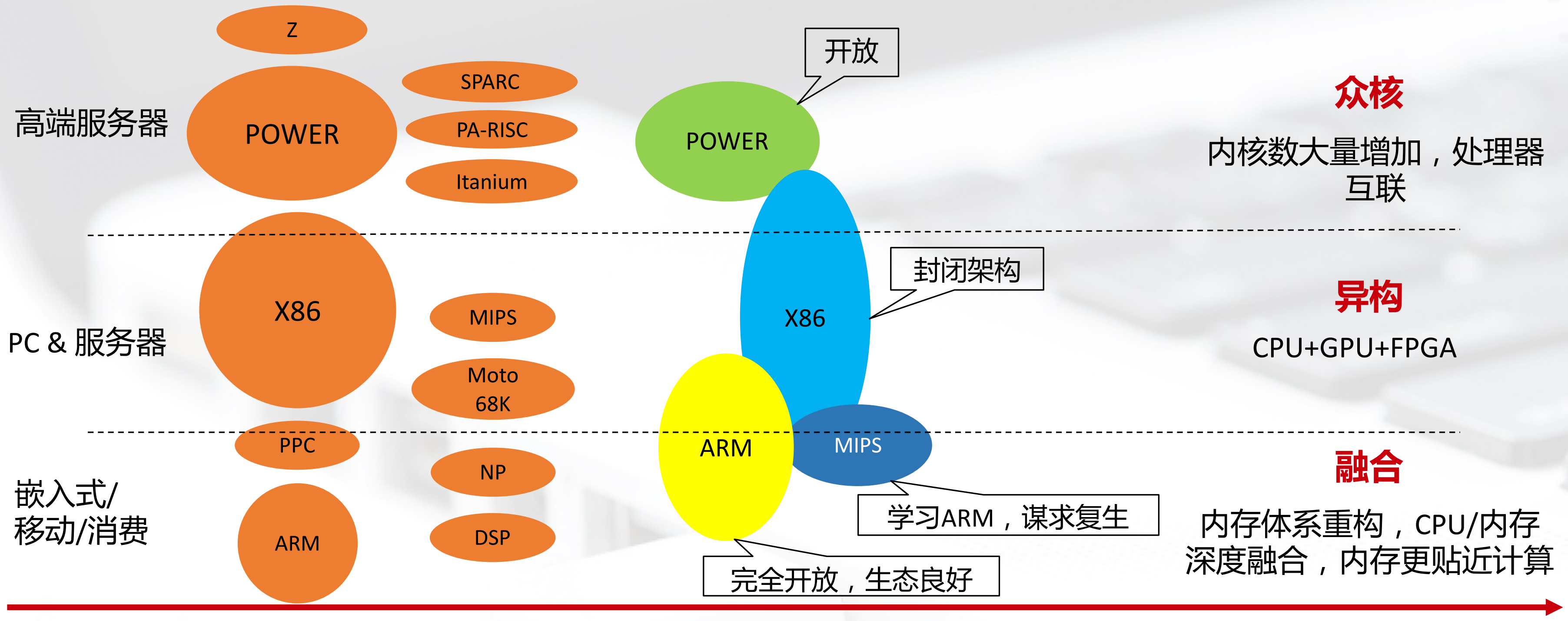
MIPS TECHNOLOGIES: MIPS3k, MIPS4K, MIPS5K, MIPS10K, MIPS32/64

ARM

ARM: ARM7, ARM9, ARM10, ARM11, ARMv8

1970s, 1980s, 1990s, 2000s

Thirty years after the first microprocessor chip was born



过去：架构众多，百花齐放　　**现在：生态成熟，架构垄断**　　**未来：摩尔定律失效，寻求多方向突破**

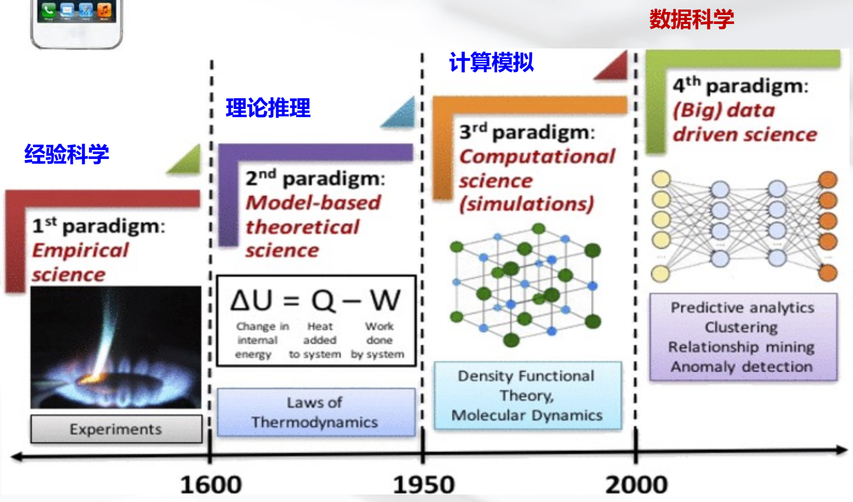
人类如何实现从物理设备到问题求解的？

Application

差距太大，
一步无法跨越！

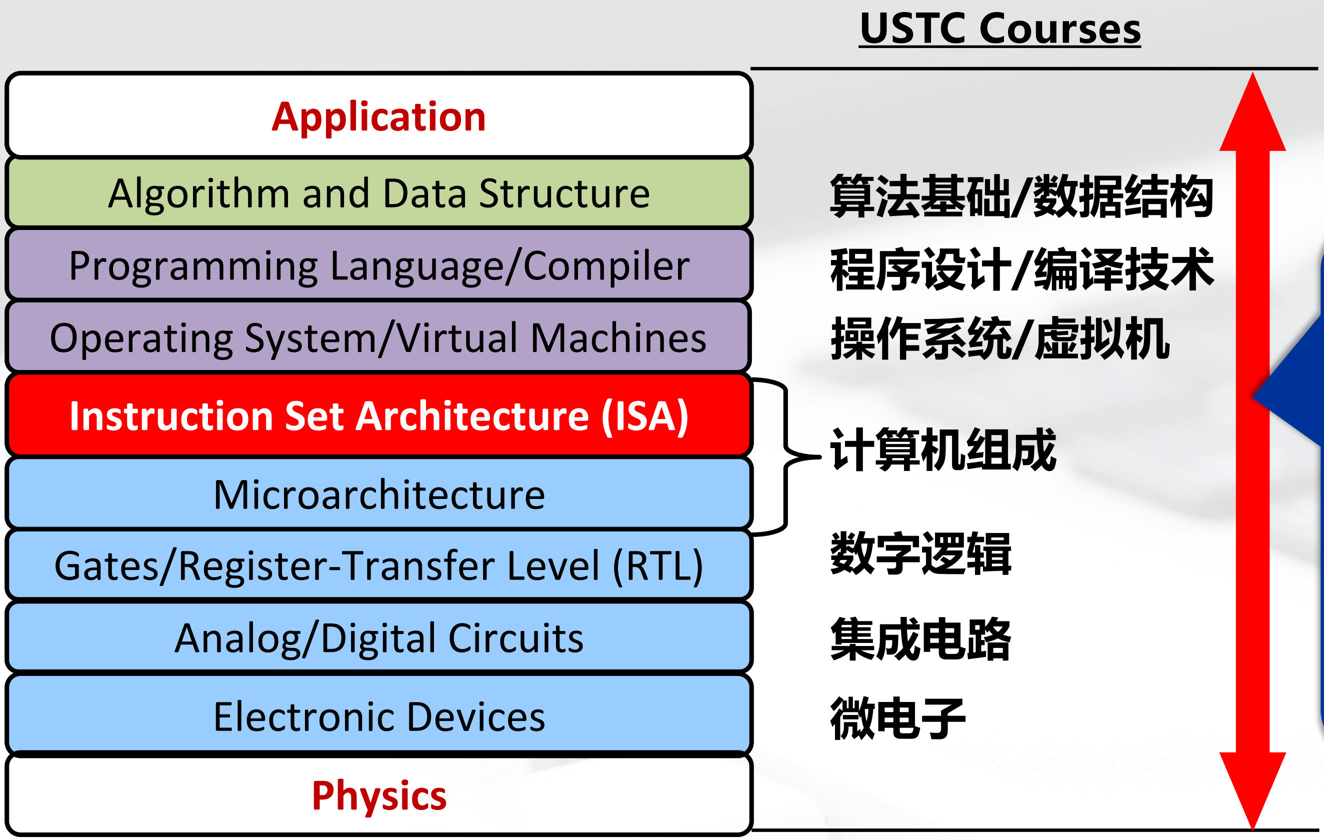
也有例外，
例如罗盘

Physics



经验科学
理论推理
计算模拟
数据科学
1st paradigm: Empirical science
Experiments
2nd paradigm: Model-based theoretical science
ΔU = Q – W
Change in internal energy
Heat added to system
Work done by system
Laws of Thermodynamics
3rd paradigm: Computational science (simulations)
Density Functional Theory, Molecular Dynamics
4th paradigm: (Big) data driven science
Predictive analytics
Clustering
Relationship mining
Anomaly detection
1600
1950
2000

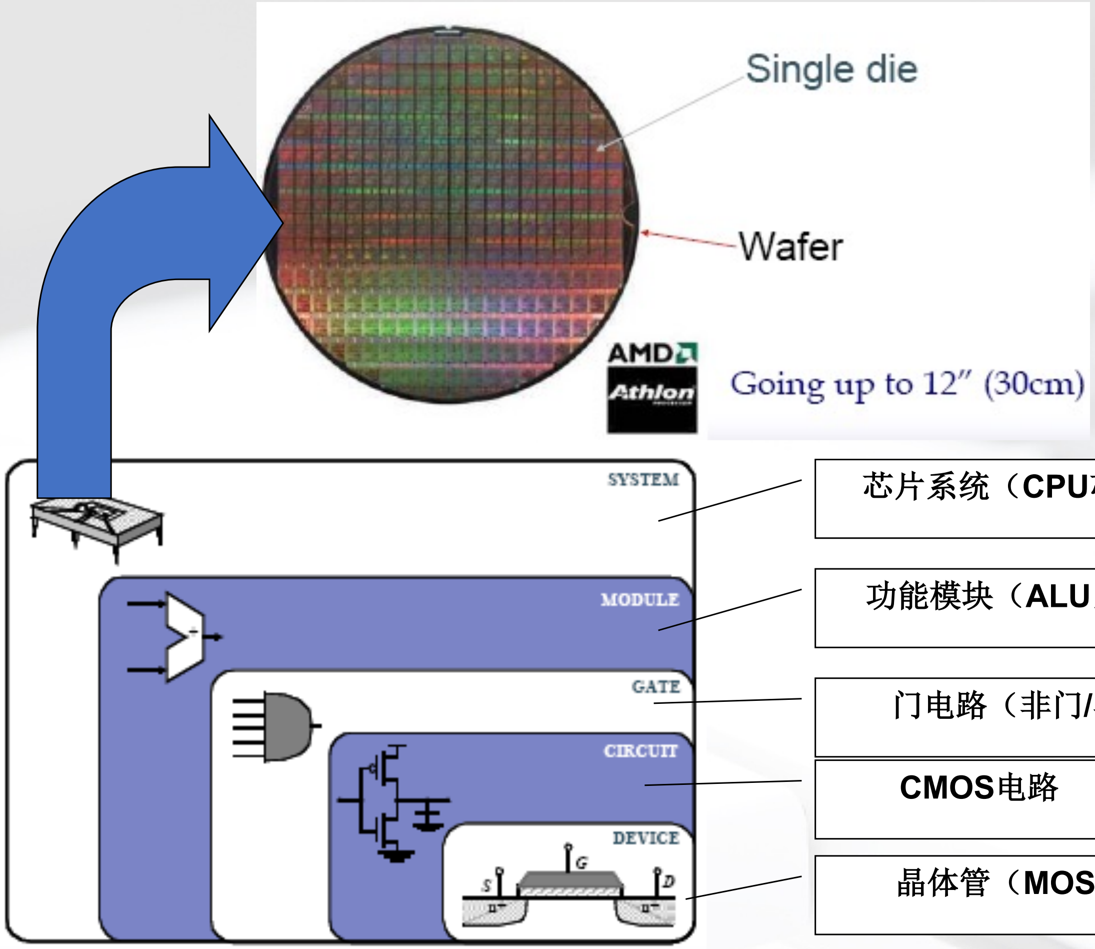
Great Idea #3: Abstraction helps us Manage Complexity



需要一门**贯通课程**，帮助学生从底层物理到高层应用，整体上理解计算机系统。

从广义上讲，**计算机系统结构**是抽象层次的设计，它允许我们使用可用的制造技术有效地实现信息处理**应用程序**。

Abstraction to Simplify Hardware Design



芯片系统（CPU芯片，SOC芯片，。。。）

功能模块（ALU，FPU，寄存器堆。。。）

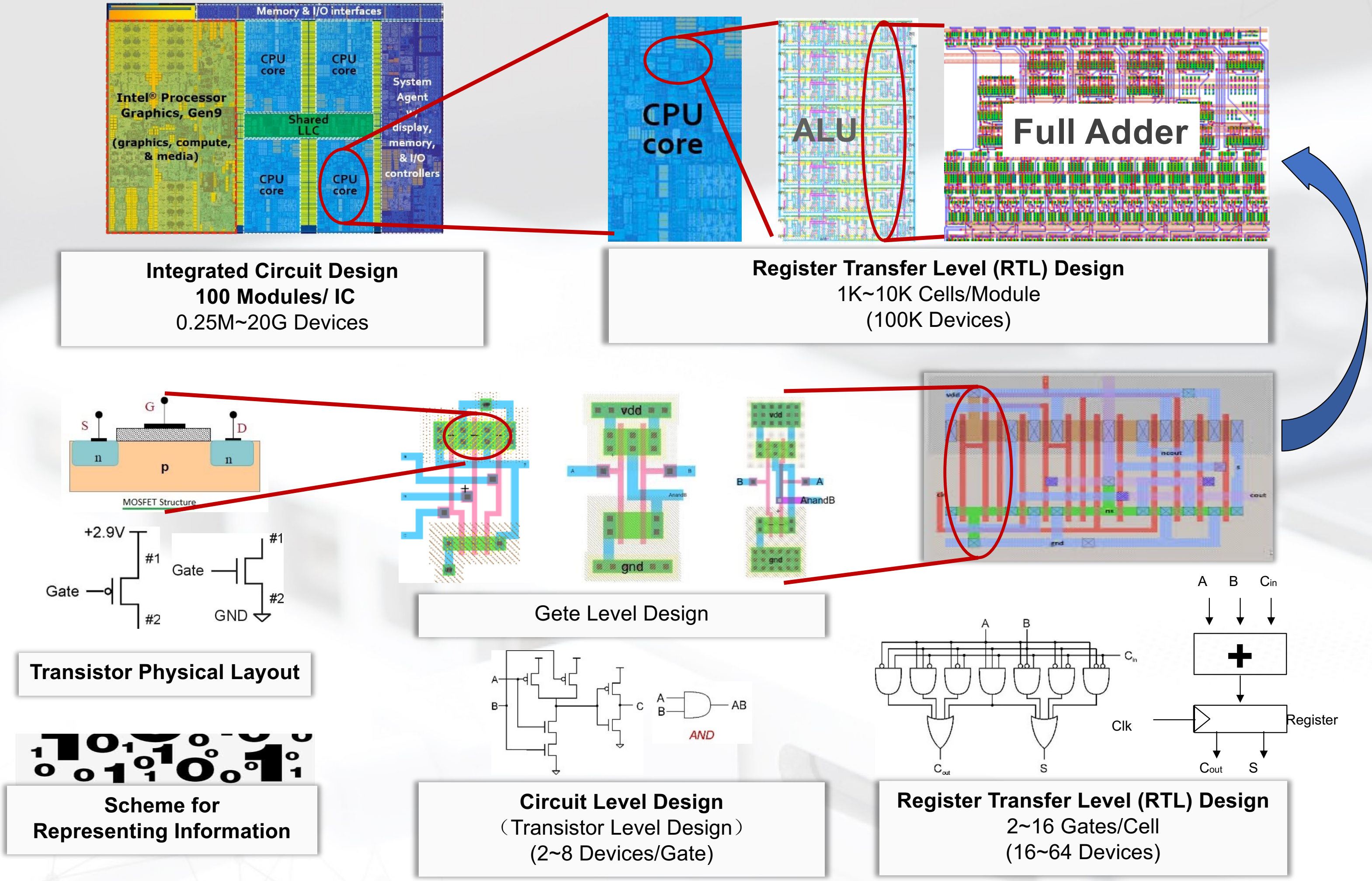
门电路（非门/与非门/或非门。。。）

CMOS电路

晶体管（MOSFET：PMOS/NMOS）

How do we put the devices into system?

Abstraction to Simplify System Design



Integrated Circuit Design
100 Modules/ IC
0.25M~20G Devices

Register Transfer Level (RTL) Design
1K~10K Cells/Module
(100K Devices)

Transistor Physical Layout

Gete Level Design

Scheme for Representing Information

Circuit Level Design
（Transistor Level Design）
(2~8 Devices/Gate)

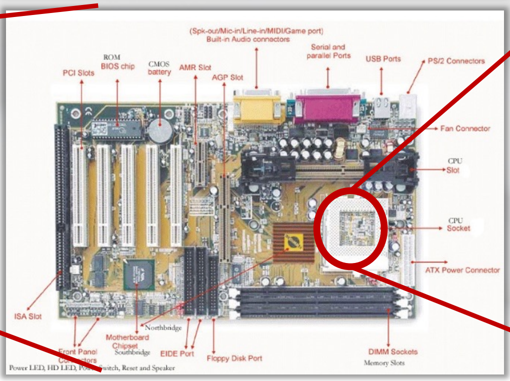
Register Transfer Level (RTL) Design
2~16 Gates/Cell
(16~64 Devices)

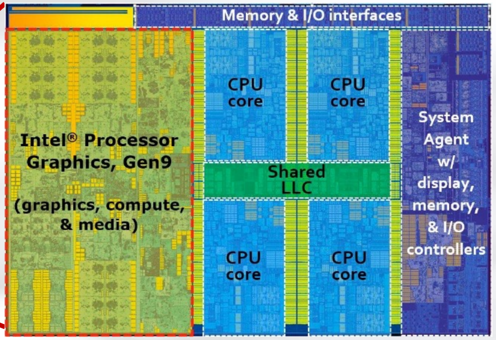
Abstraction to Simplify System Design

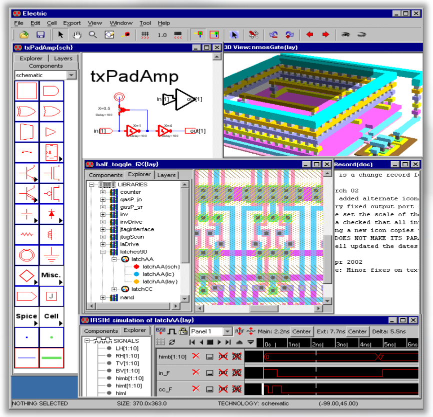
Personal Computer:
Hardware & Software Design
1~10PCBs/System

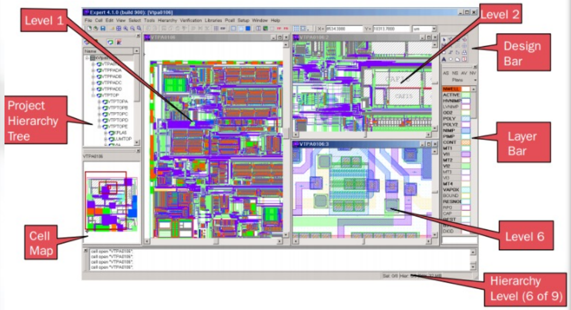
Motherboard Circuit Design
10 ICs/ PCB
1~50G Devices

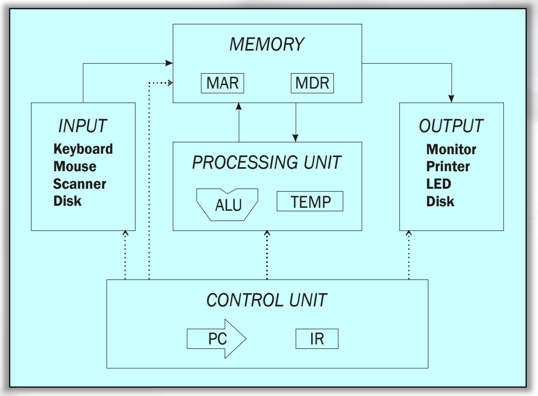
Integrated Circuit Design
100 Modules/ IC
0.25M~20G Devices




Memory & I/O interfaces
CPU core
CPU core
Intel® Processor Graphics, Gen9
(graphics, compute, & media)
Shared LLC
System Agent w/ display, memory, & I/O controllers
CPU core
CPU core

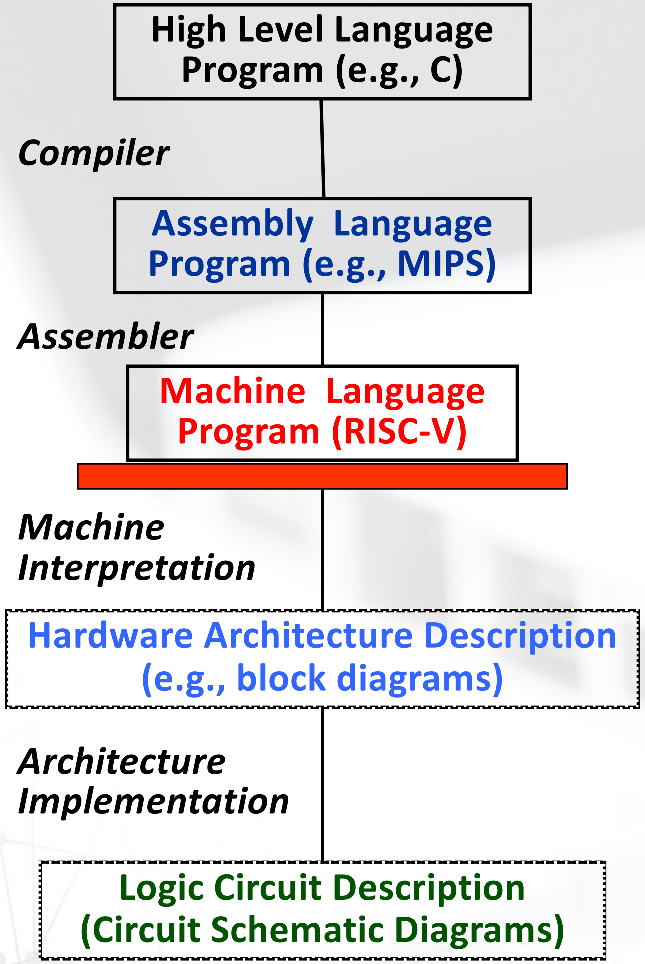






Electronic System Level (ESL)Design

How do we get the electrons to do the work?



High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g., MIPS)

Assembler

Machine Language Program (RISC-V)

Machine Interpretation

Hardware Architecture Description (e.g., block diagrams)

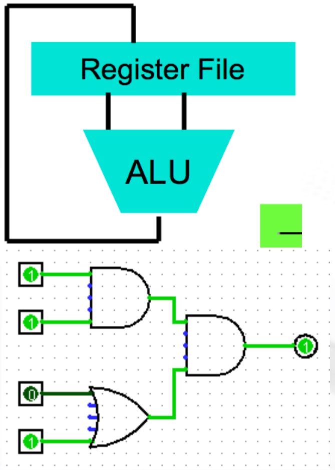
Architecture Implementation

Logic Circuit Description (Circuit Schematic Diagrams)

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

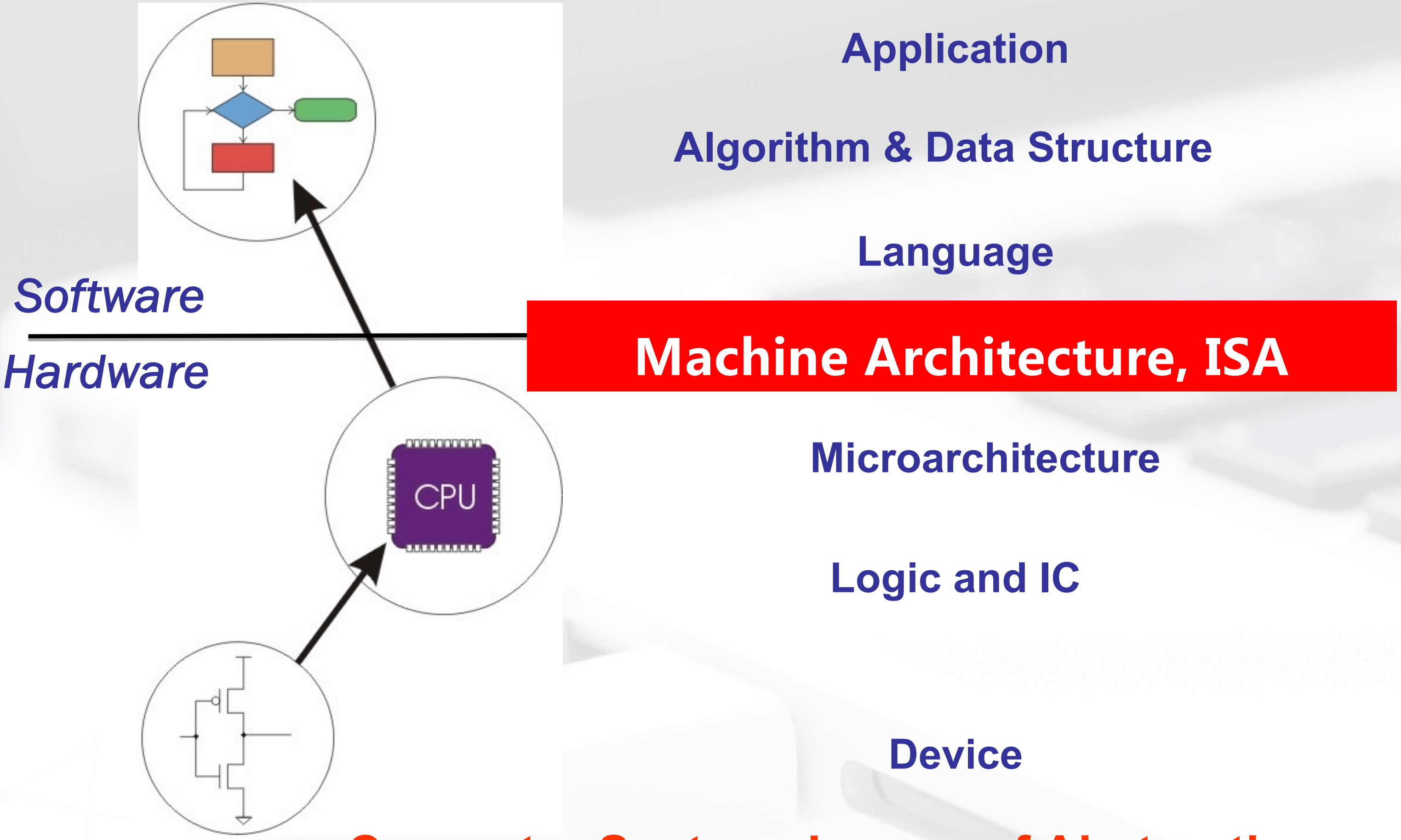
```
lw      $t0, 0($2)
lw      $t1, 4($2)
sw      $t1, 0($2)
sw      $t0, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```



Anything can be represented as a *number*, i.e., data or instructions

Great Idea #4: Software and Hardware Co-design



Software

Hardware

Application

Algorithm & Data Structure

Language

Machine Architecture, ISA

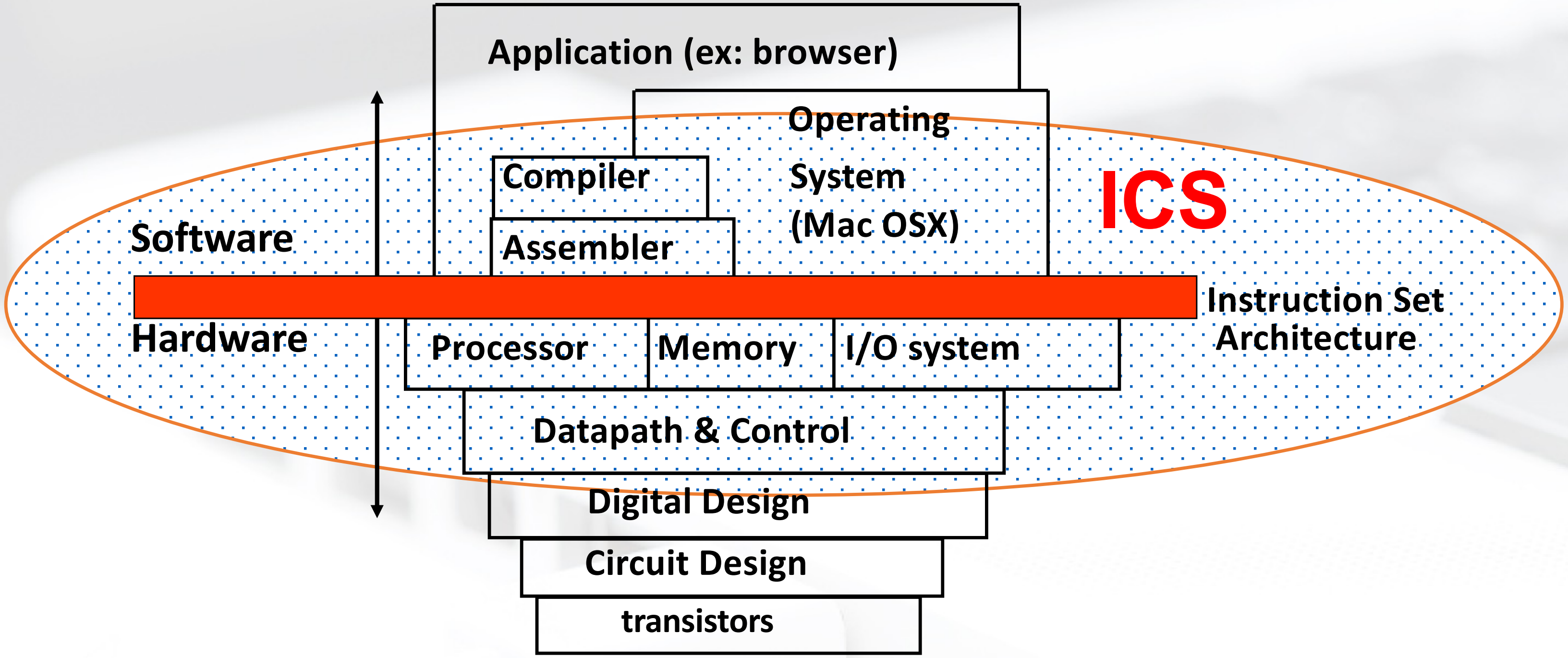
Microarchitecture

Logic and IC

Device

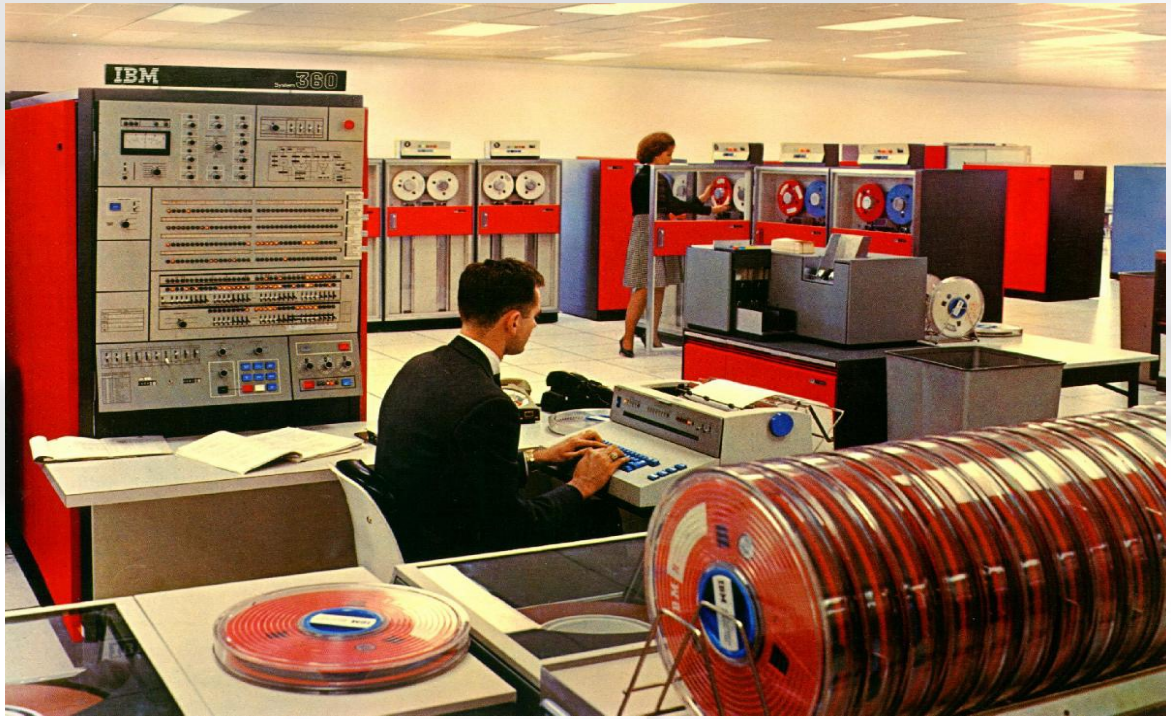
Computer System: Layers of Abstraction

Old Machine Structures



Old Machine Structures

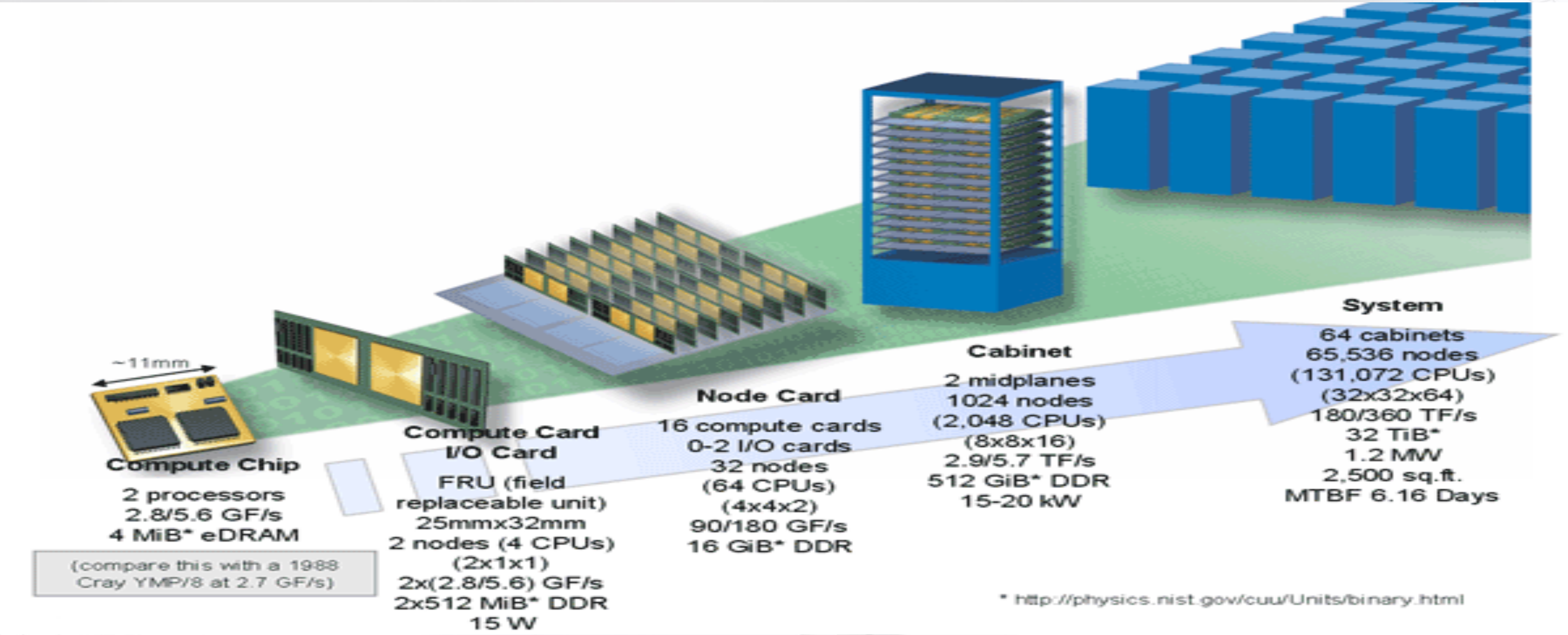
■ Mainframe: IBM System/360



Computer Architecture: the structure of a computer that a machine language programmer must understand to write a correct (timing independent) program for that machine.

New Machine Structures

- **IBM Blue Gene**



Compute Chip

2 processors
2.8/5.6 GF/s
4 MiB* eDRAM

(compare this with a 1988
Cray YMP/8 at 2.7 GF/s)

Compute Card
I/O Card

FRU (field
replaceable unit)
25mmx32mm
2 nodes (4 CPUs)
(2x1x1)
2x(2.8/5.6) GF/s
2x512 MiB* DDR
15 W

Node Card

16 compute cards
0-2 I/O cards
32 nodes
(64 CPUs)
(4x4x2)
90/180 GF/s
16 GiB* DDR

Cabinet

2 midplanes
1024 nodes
(2,048 CPUs)
(8x8x16)
2.9/5.7 TF/s
512 GiB* DDR
15-20 kW

System

64 cabinets
65,536 nodes
(131,072 CPUs)
(32x32x64)
180/360 TF/s
32 TiB*
1.2 MW
2,500 sq.ft.
MTBF 6.16 Days

* http://physics.nist.gov/cuu/Units/binary.html

New Machine Structures: From the Gate to the Cloud

Software *Hardware*

- **Parallel Requests**
  Assigned to computer
  e.g., Search "Katz"
- **Parallel Threads**
  Assigned to core
  e.g., Lookup, Ads
- **Paralle**
  >1 instruct
  e.g., 5 pipe
- **Paralle**
  >1 data ite
  e.g., Add o
- **Hardware Descriptions**
  All gates functioning in parallel at same time
- **Programming Languages**

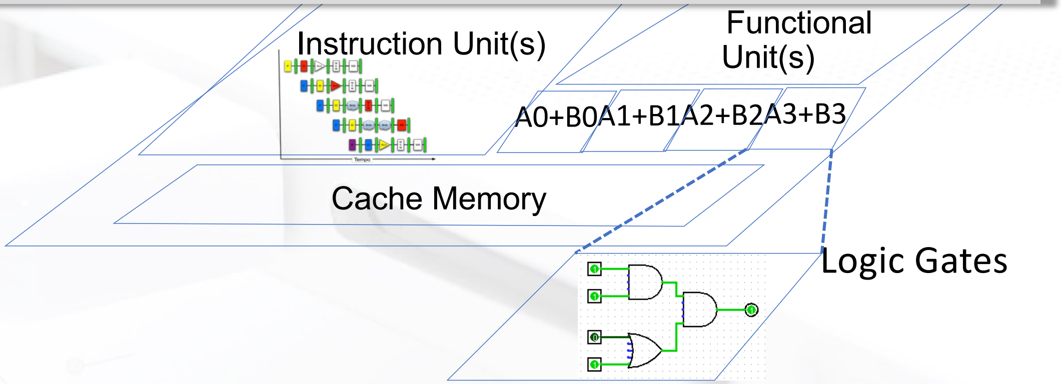
Leverage Parallelism &
Achieve High Performance

Warehouse
Scale Computer

Smart
Phone

All computers are parallel!

All computer engineers and scientists require the knowledge of parallel computation



Course Objectives

- Exposure to. . .
  - Machine organization
  - Assembly language programming
  - C programming
- Understand how to build entire (slow) computing system
  - Hardware and software
  - You’ll get a chance in complementary courses
- Be distinguished from mere programmers
  - E.g. Matrix Multiply

计算机系统（晶体管器件、数字逻辑、组成原理、高级语言的编译与汇编、高级语言的硬件实现、操作系统）核心概念和思想的最小集

Course Objectives

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke, "Profiles of The Future" (Clarke's 3rd law)

- No magic: Computers should not be magic to computer scientists!
- Bottom UP: Start with what they "know"
  - Computing systems from transistors on up
  - The transistor as light switch
  - Not quantum mechanics
- Choose a computer model that is simple
  - Not about "design", but about "insight" into all computers
  - As the genius said: simple, but still rich
  - Continually build on what you know
  - Continually raising the level of abstraction
  - Memorizing as little as absolutely necessary
  - Trying very hard to not introduce magic

You take, You enjoy!!!

Outline

0 Course Crew

1 What’ s the difference between “Big” and “Small” ?

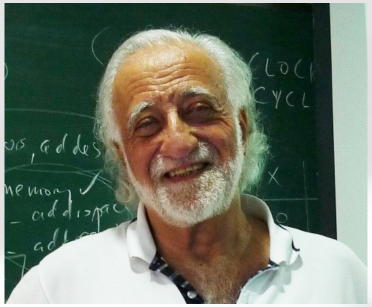
2 Why Take This Course?

3 Great Ideas in Computing Systems

4 **Course Implementation**

Text Book

- Introduction to computer architecture(ISA)
  - How is data represented?
  - What are the pieces of a computer?
  - How do computers work?
- Programming
  - How do I "talk" directly to the machine?-Assembly language
  - How do I program in "C"?-high level language(HLL)
- Computer systems and computation
  - How do simple HW/SW elements come together to realize complex computations?



introduction to computing systems
from bits & gates to C & beyond
yale n. patt
sanjay j. patel
introduction to computing
from bits & gates to C & beyond
yale n. patt
sanjay j. patel
third edition
introduction to computing systems
from bits & gates to C/C++ & beyond
yale n. patt
sanjay j. patel
Mc Graw Hill

Introduction to Computing Systems: from bits and gates to C/C++and beyond(3nd edition),

Yale N. Patt and Sanjay J. Patel， September 2019, McGraw-Hill Higher Education

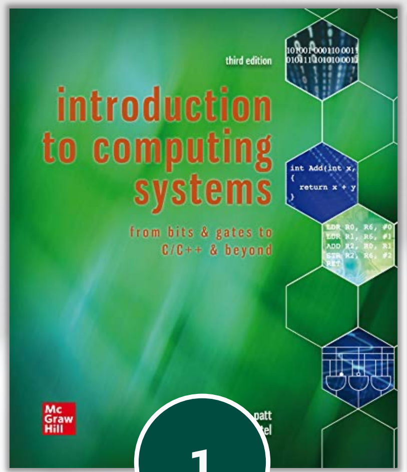
Text Book Components

- **Part 1: Hardware(Chapter 1-4)**
  - Representing data, transistors, gates, digital logic structures
  - von Neumann machine model
- **Part 2: Software: Assembly language(Chapter 5-10)**
  - Instructions, (structured) programming, input/output, *relationship to hardware*
- **Part 3: Software: C/C++ programming(Chapter11-20), selected**
  - Syntax, operators, control structures, functions, *pointers*, recursion, data structures, *relationship to assembly language*
  - Assume already familiar with programming (C)

This Course Focus on

- Chapt 2 Bits, Data Types, and Operations
  - How do we represent information using electrical signals?
- Chapt 3 Digital Logic Structures
  - How do we build circuits to process information?
- Chapt 4, 5 Computer Machine Model, Processor and Instruction Set
  - How do we build a processor out of logic elements?
  - What operations (instructions) will we implement?
- Chapt 6,7 Assembly Language Programming
  - How do we use processor instructions to implement algorithms?
  - How do we write modular, reusable code? (subroutines)
- Chapt 8 Data Structures
- Chapt 9 I/O, Traps, and Interrupts
  - How does processor communicate with outside world?
- Chapt 10 Put It All Together: A Calculator
- Chapt 11, C Implemention Related to Hardware

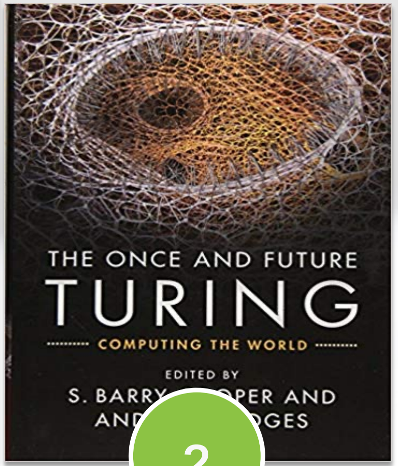
Other Reference Text Books


third edition
introduction to computing systems
from bits & gates to C/C++ & beyond
Mc Graw Hill

1

Introduction to Computing Systems: from bits and gates to C/C++ and beyond(3nd edition),

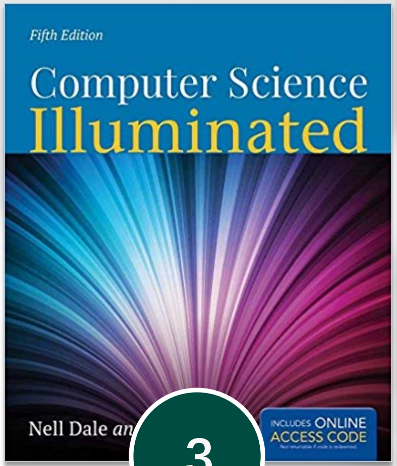
Yale N. Patt and Sanjay J. Patel， June 2019, McGraw-Hill Higher Education



2

***The Once and Future Turing: Computing the World* （1st Edition）,**

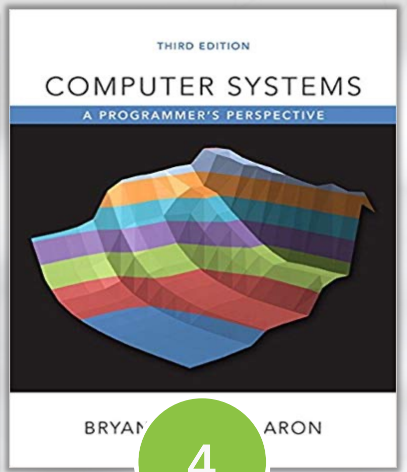
S. Barry Cooper, Andrew Hodges，Cambridge University Press，2016



3

Computer Science Illuminated(5th Edition),

D. M. Harris, S. L. Harris, Morgan Kaufmann, San Francisco, 2007

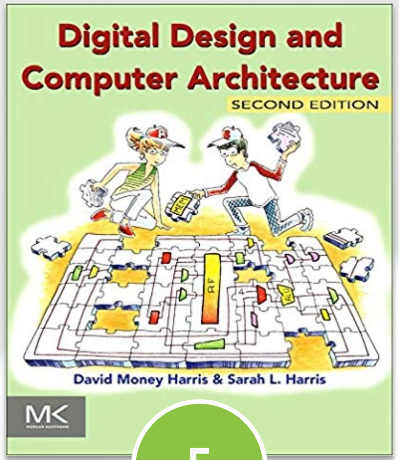


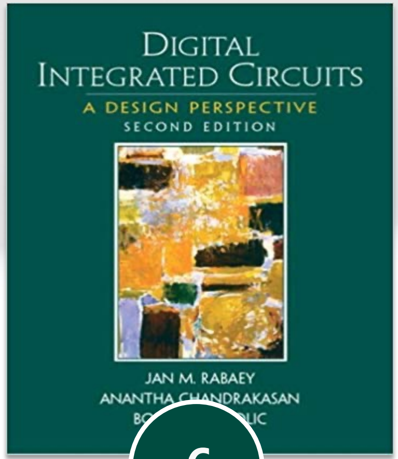
4

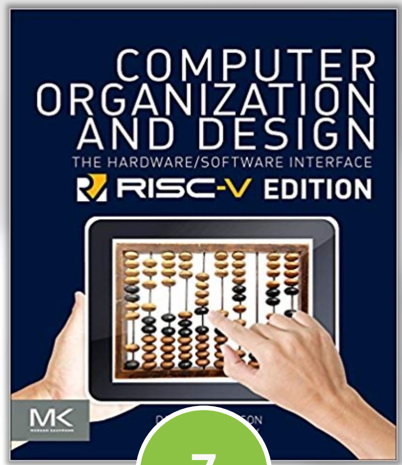
***Computer Systems: A Programmer_‘s Perspective* (3rd Edition)，**

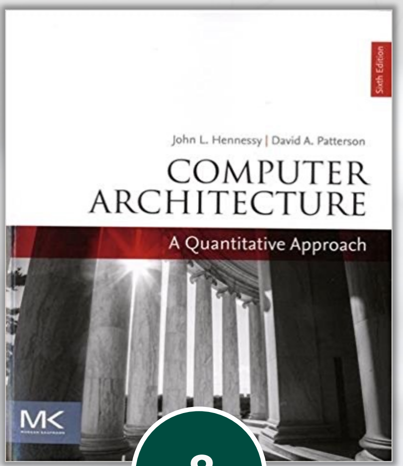
Randal E. Bryant, David R. O’Hallaron，Pearson Education Inc.，2016

Other Reference Text Books


Digital Design and Computer Architecture
SECOND EDITION
David Money Harris & Sarah L. Harris
MK




COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
RISC-V EDITION
MK



5

Digital Design and Computer Architecture (2nd Edition) ，

David Harris Sarah Harris，p712, Morgan Kaufmann，24th July 2012

6

***Digital Integrated Circuits: A Design Perspective* (2nd Edition),**

Jan M. Rabaey，Anantha Chandrakasan, Borivoje, Prentice-Hall, Inc，Nikolic,Jan 3, 2003

7

***Computer Organization and Design: The Hardware/Software Interface*，**

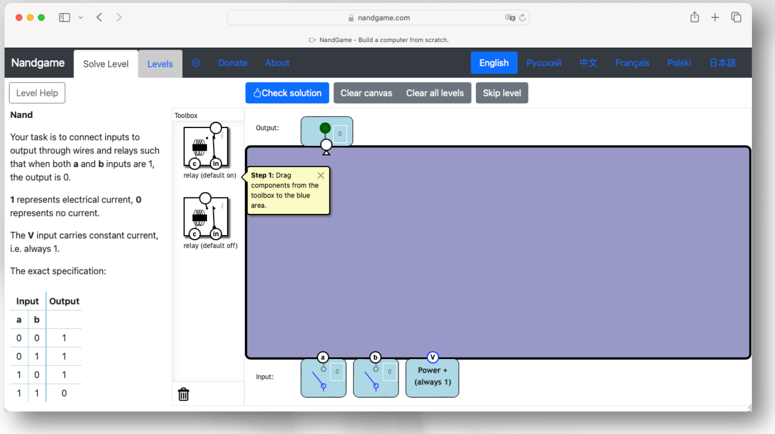
David A Patterson, John L. Hennessy, 5th edition. Morgan Kaufmann Publishers, Inc.，2017

8

***Computer Architecture: A Quantitative Approach*，**

John L. Hennessy and David A. Patterson， The Morgan Kaufmann，Dec 7, 2017

Game



Nand

Your task is to connect inputs to output through wires and relays such that when both **a** and **b** inputs are 1, the output is 0.

1 represents electrical current, **0** represents no current.

The **V** input carries constant current, i.e. always 1.

The exact specification:

Input		Output
a	b	
0	0	1
0	1	1
1	0	1
1	1	0

Step 1: Drag components from the toolbox to the blue area.

NandGame

https://www.nandgame.com

图灵完备

Class Organization

Lectures

- Will not simply "cover" the material
- Will focus on the "hard stuff"
- Will not stand alone, instead build on reading

Discussion sessions

- Encouraged!
- Okay: discuss meaning of problem, discuss approaches
- Not okay: comparing answers, solving questions together

Homework Assignments

- **Problem Sets : 6 sets**
  - Problem solving
  - Complete *before* each due date
  - Can work ahead
  - Great exam preparation!

Labs Assignments

- **Simple Programming Assignments**
  - **Programming LC3 Assignments1:**
    - —Programming In machine language
  - **Programming Assignments2~5:**
    - —Programming In LC3 assembly language
  - **Programming Assignments6:**
    - —Assignments2~5 Programming In C
- **Challenging Course Projects**
  - **LC3 Simulator/Assembler Design (Encouraged! Extra 5 points)**
- **See schedule for each lab due dates**

Exams & Grades

- Middle Exam: 20%
- Final Exam: 30%
- Assignments : 50%
  - Problem Sets for every chapter
  - 6 Programming Assignments

Policy on Assignments and Independent Work

- ALL PROJECTS WILL BE DONE AND SUBMITTED INDIVIDUALLY.
  - With the exception of laboratories and assignments that explicitly permit you to work in groups, all homework and projects are to be YOUR work and your work ALONE.
- You are encouraged to discuss your assignments with other students, and extra credit will be assigned to students who help others, particularly by answering questions on Piazza, but we expect that what you hand in is yours.
- It is NOT acceptable to copy (or even "start with") solutions from other students or the Web

Tips on How to Get a Good Grade

- The lecture material is not the most challenging part of the course. You should be able to understand everything as we go along.
- DO NOT fall behind in lecture and tell yourself you "will figure it out later from the notes or books".
- Notes will be online after the lecture (usually the night). Do assigned reading before the lecture.
- Ask questions in class and stay involved in the class - that will help you understand.
- Come to office hours to check your understanding or to ask questions.
- Complete all the homework problems - even the difficult ones. The exams will test your depth of knowledge.

Tips on How to Get a Good Grade

- You need to understand the material well enough to apply it in new situations. You need to enroll in both the lab and the course.
  - Take the labs very seriously. They are an integral part of the course.
  - Choose your partner carefully. Your best friend may not be the best choice!
  - Most important : Be well organized and neat with homework, labs, project. In lab, add complexity a little bit at a time - always have a working design.

Academic Integrity(学术诚信)

Do not post your work on public repositories like github (private o.k.) --negative points

- The rule is simple
  - Claiming another’s work as your own *will ruin your life*
  - See syllabus for details and examples
- Who will know?
  - We will (inspection, similarity detectors, exams)
  - Your friends will… your parents will…
  - You will
- Remember
  - If you need to cheat now, you’ve got much bigger problems
  - Cheating is like going 150 MPH over speed limit while drunk!

Acknowledgements

- This course is partly inspired by previous MIT 6.823 and Berkeley CS252 computer architecture courses created by my collaborators and colleagues:
  - Yale Patt（Univeristy of Texas at Austin）
  - Arvind (MIT)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
  - Krste Asanovic (UCB)