西安交通大學

XI'AN JIAOTONG UNIVERSITY

第五讲：
如何让程序更快?

师斌

School of Computer Science & Technology

如何让计算机更快?

- 概念
  - The clock speed（时钟速度）measures the number of cycles your CPU executes per second, measured in GHz (gigahertz)
  - 时钟周期（ cycles ）是CPU工作的最小时间单位
- 性能公式

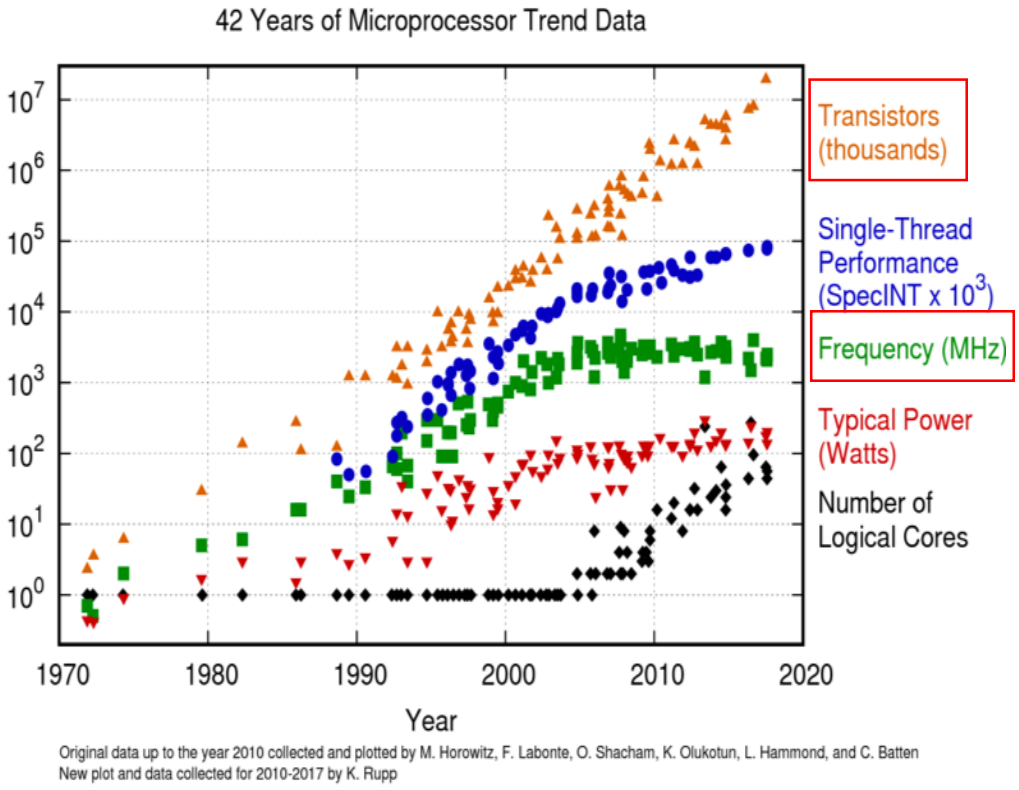
$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

一个时钟周期为多长时间?

在每条指令上花费多少个时钟周期?

执行一项任务需要多少条指令?

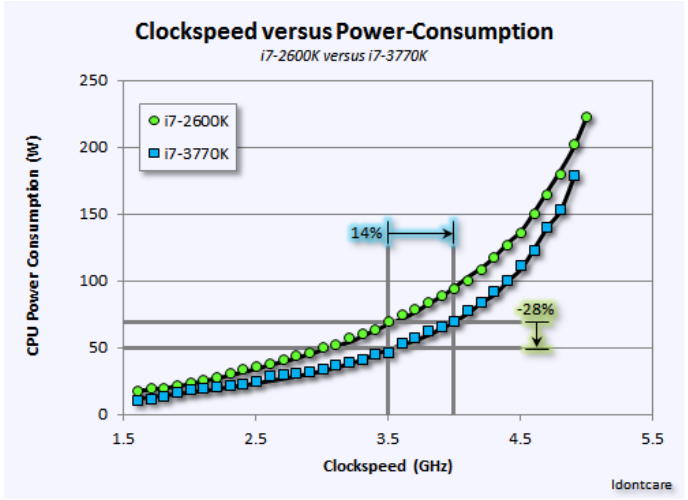
途径1：加速时钟周期



芯片发展趋势数据

为什么过去 10 年 CPU 时钟速度没有明显提高？

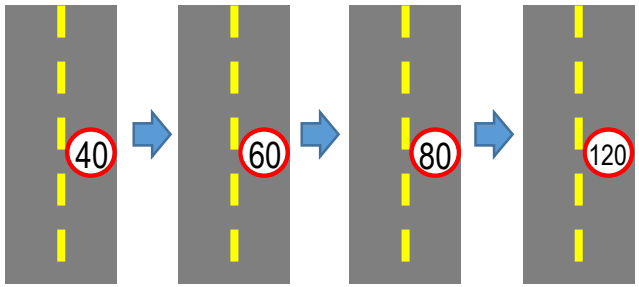
能耗:

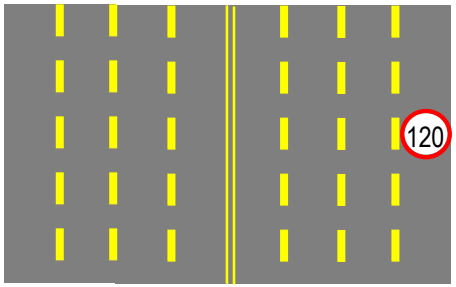


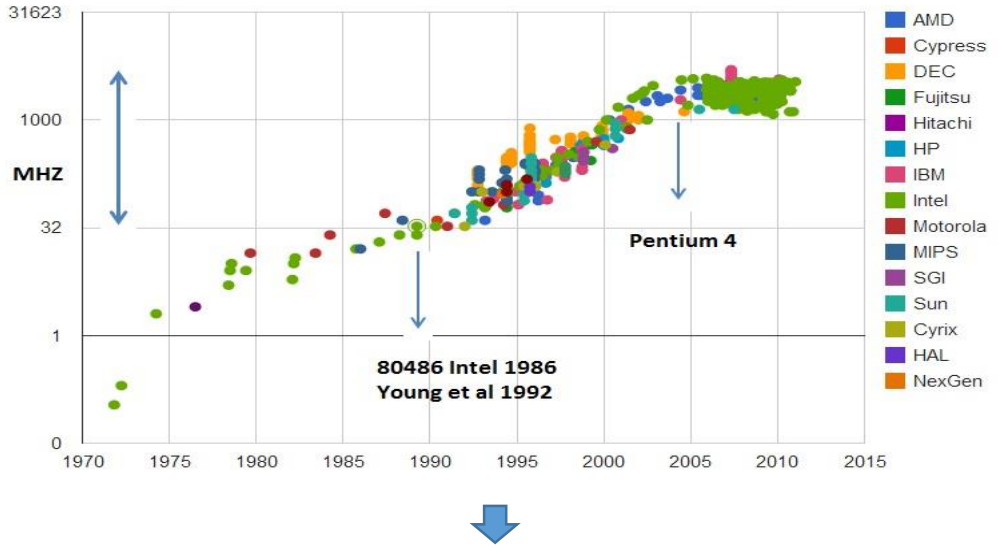
对于处理器来讲，纯高频率的设计在能耗方面并不合理：

- 我们打开和关闭晶体管的速度越快，产生的热量就越多。
- 时钟速度的增加意味着电压的增加，并且这与功率之间存在三次依赖性。

途径2：减少每条指令平均花费的时钟周期数CPI







Instruction-level parallelism

for executing more than one basic instruction one time.

Task-level parallelism

for distributing tasks across different processors.

Data parallelism

for distributing the data across different processors.

途径2：减少每条指令平均花费的时钟周期数CPI

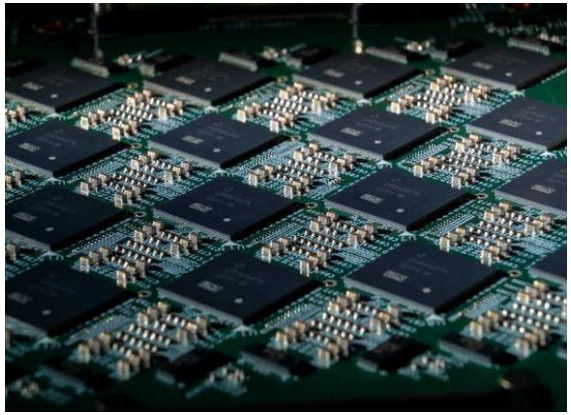
● 途径2-1：Instruction-Level Parallelism

$$\text{Average ILP} = \frac{instructions}{cycles}$$

Instr. No.	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

RISC机器的五层流水线示意图（IF：读取指令，ID：指令解码，EX：运行，MEM：存储器访问，WB：写回寄存器）

IF	ID	EX	MEM	WB
IF	ID	EX	MEM	WB
IF	ID	EX	MEM	WB
IF	ID	EX	MEM	WB
IF	ID	EX	MEM	WB
IF	ID	EX	MEM	WB



1. 流水线

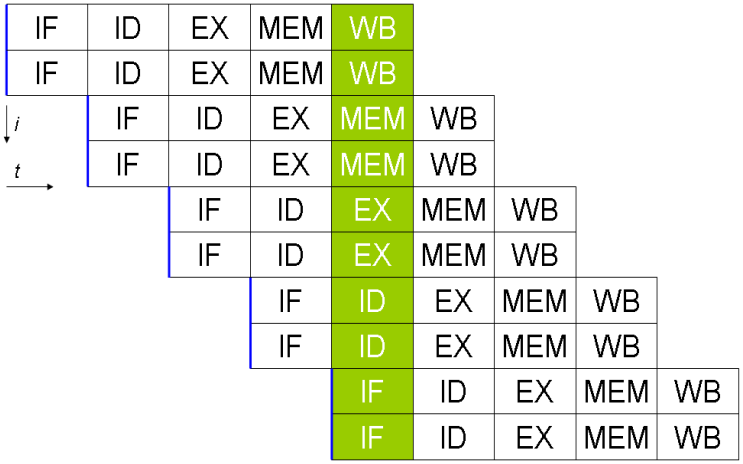
2. 超标量

3. 多核多CPU

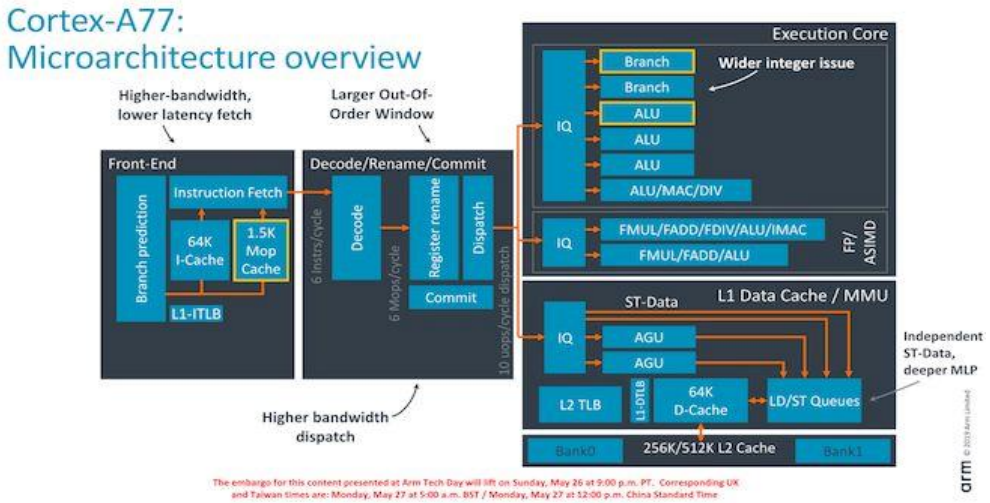
途径2：减少每条指令平均花费的时钟周期数CPI

途径2-1：Instruction-level parallelism

$$\text{Average ILP} = \frac{instructions}{cycles}$$



超标量流水线理想时空图

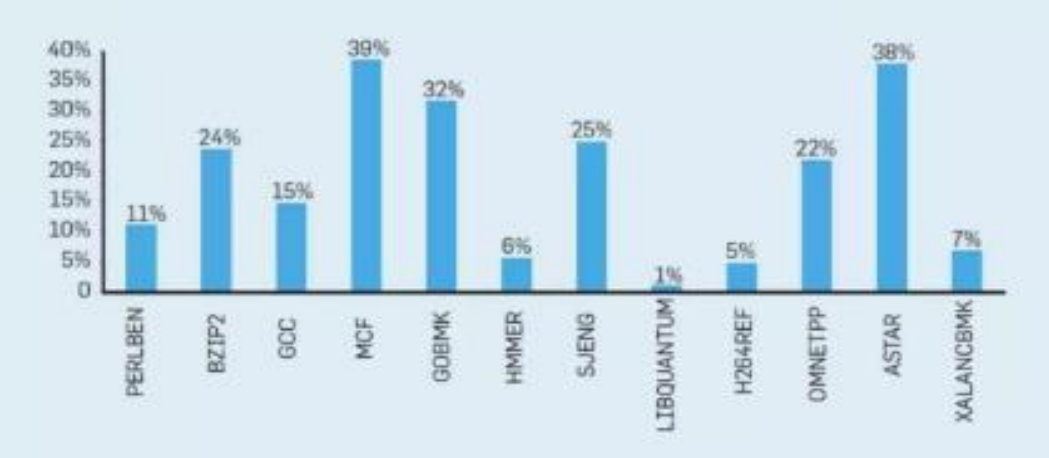


现代计算机

15级流水线，4条指令同时发射，最多60条指令并行，提速60倍

如何让计算机更快?

- 流水线的思路是否有局限性？
  - 有！不知道下一条指令时，无法装载流水线。
    - 例如：条件判断语句下一条指令
  - 于是，预测下一条指令。



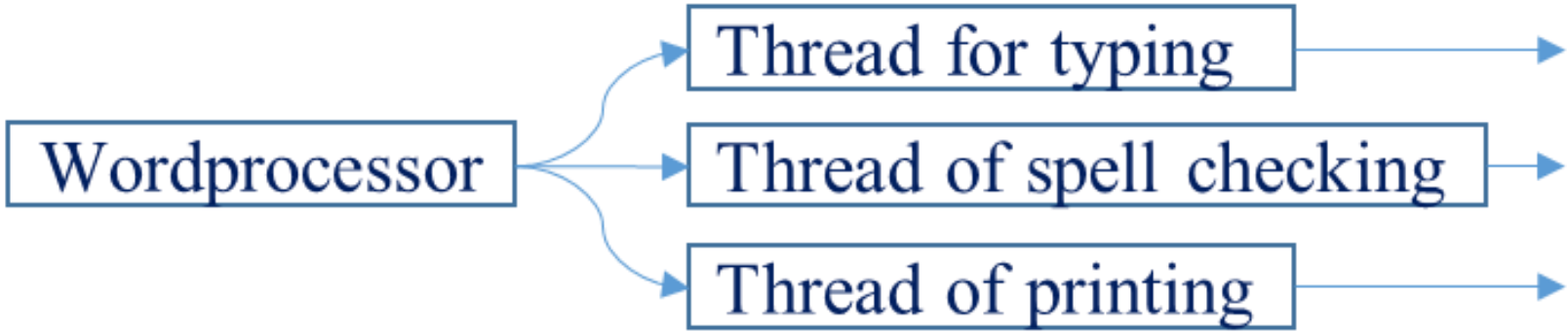
分支预测错误率

- 75%的分支指令预测对了正常执行，预测错误则流水线预取作废。
- 最先进的分支预测技术，平均有20%的预测错误。

途径2：减少每条指令平均花费的时钟周期数CPI

途径2-2：Task-level parallelism

$$\text{Average ILP} = \frac{instructions}{cycles}$$



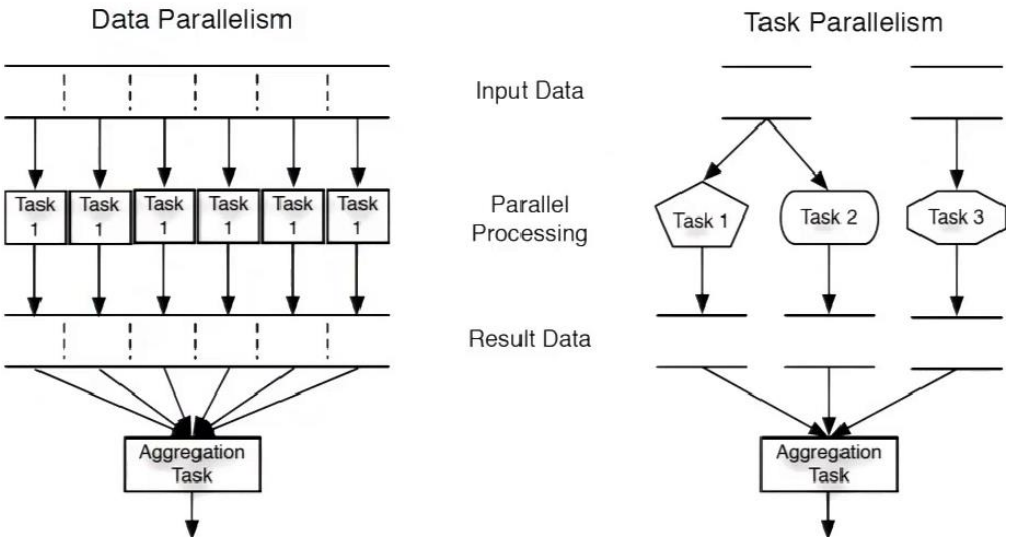
思路：将任务分解成多线程，需要将任务分解成若干子任务。

途径2：减少每条指令平均花费的时钟周期数CPI

途径2-3：Data parallelism

$$\text{Average ILP} = \frac{instructions}{cycles}$$

将要处理的数据分配到不同处理器上

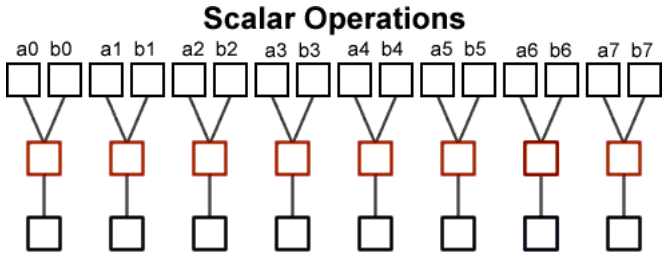


途径2：减少每条指令平均花费的时钟周期数CPI

途径2-4：机器字长的变化：

- 寄存器可以存储一组单一数据类型的数据元素。
- 计算过程可以同时处理多个数据元素

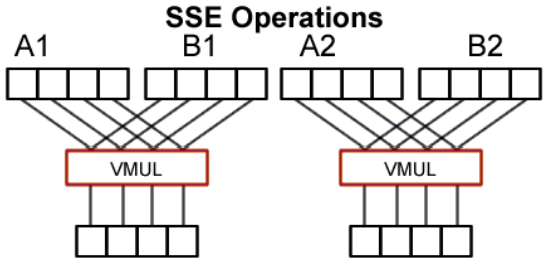
机器字长为1



16 loads,
8 AND operations,
8 stores

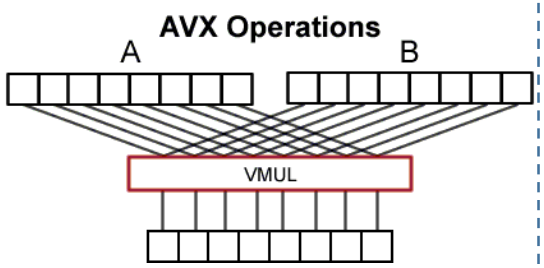
80 cycles

大的机器字长



4 loads,
2 vector AND operations,
2 stores.

20 cycles



2 loads,
1 large vector AND ops,
1 store

10 cycles

Assume: loads and stores cost 3 cycles, all calculation costs 1 cycle

如何让计算机更快?

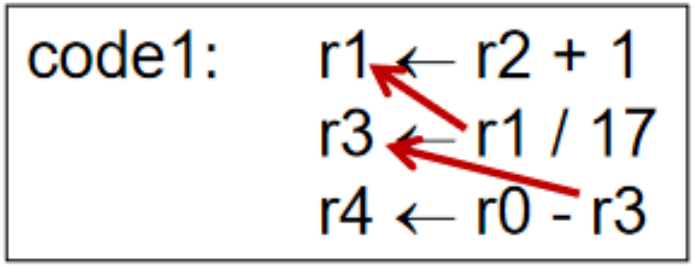
- 并行的思路是否有局限性?
  - 还有！不是所有的程序都能并行

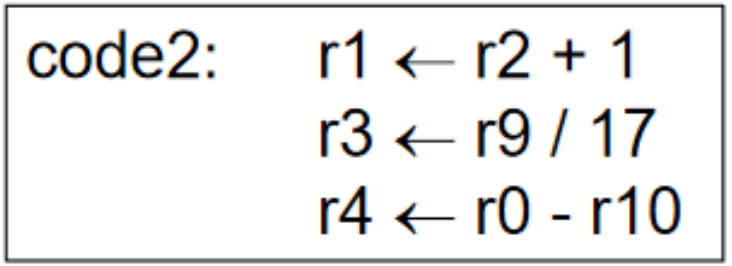
Code1: ILP =1

- **i.e. must execute serially**

Code2: ILP =3

- **i.e. can execute at the same time**





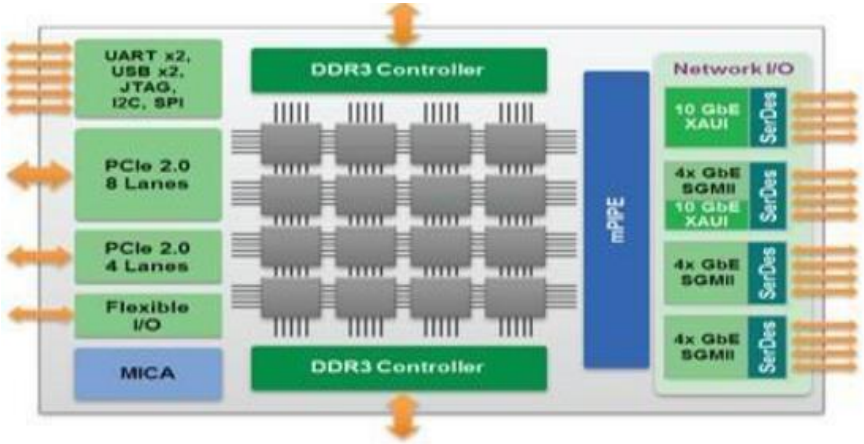
如何让计算机更快?

- 并行的思路是否有局限性？

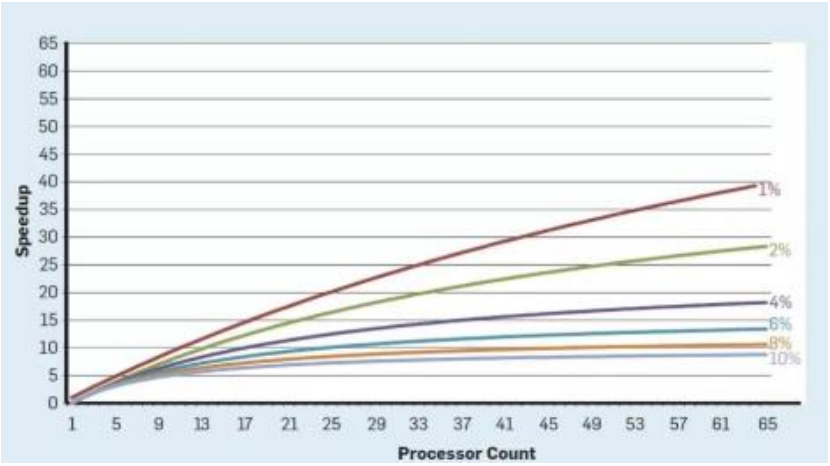
例：50%的程序不能并行执行，这时用64核并行，会加速多少倍？

全局加速比=1/(0.5+0.5/64)=1.97

- 对于50%部分可以并行的程序，64核处理器只加速不到2倍！花费了64倍的功耗！



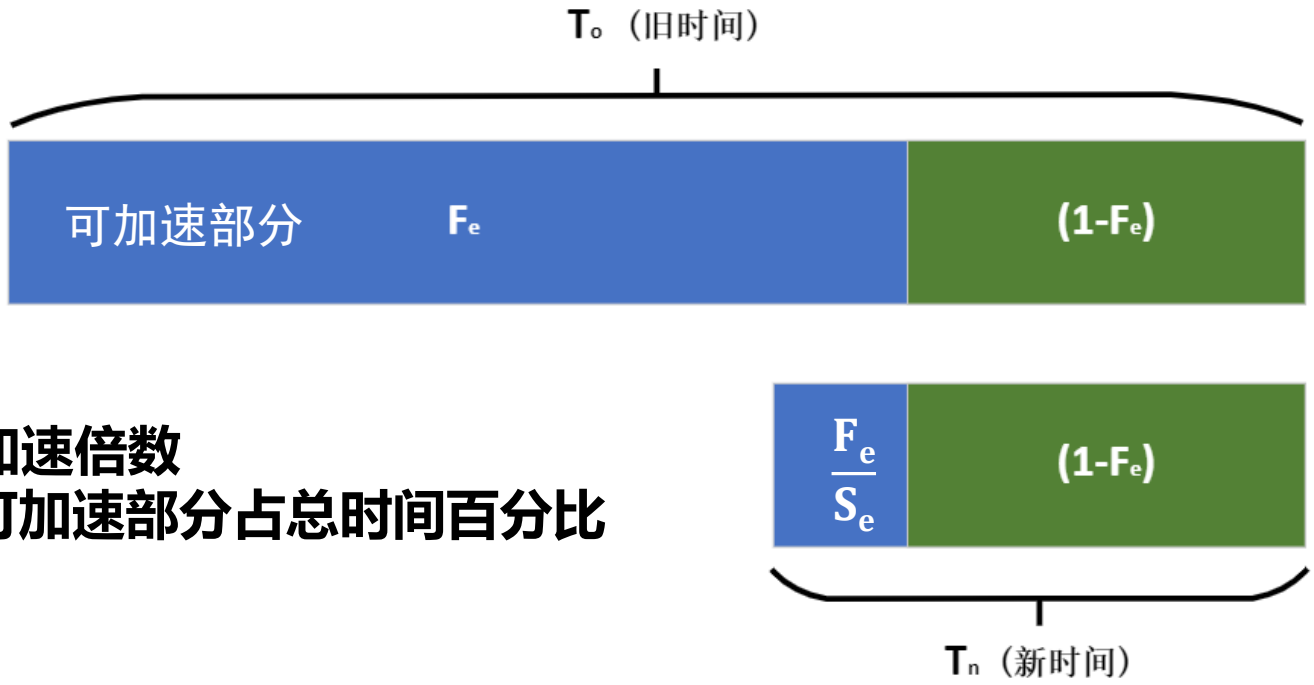
多核处理器



Amdahl定律

- 系统加速比定义：

$$S_n = \frac{\text{加速后的速度}}{\text{原始速度}} = \frac{\text{未加速的耗时}}{\text{加速后的耗时}}$$



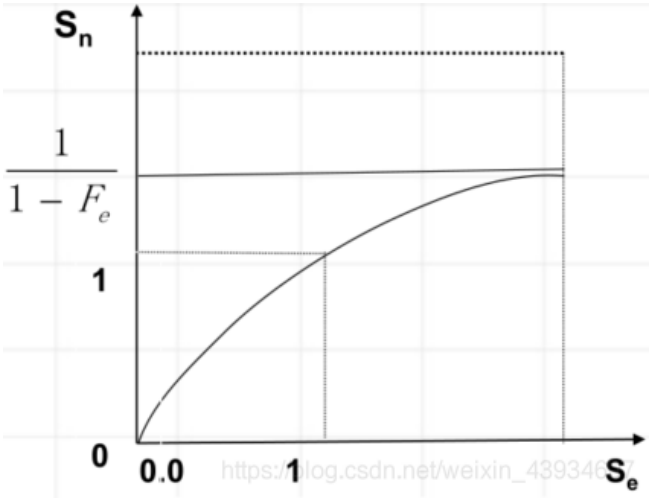
S_e – 加速倍数
F_e – 可加速部分占总时间百分比

Amdahl定律

因此，根据加速比定义有：

$$S_n = \frac{T_o}{T_n} = \frac{1}{(1-Fe) + \frac{F_e}{S_e}}$$

具体来说，Fe不变，随着Se增大，Sn增速越来越慢，且收敛到极限 $\frac{1}{1-F_e}$



这就是Amdahl定律的**性能递减规则**：

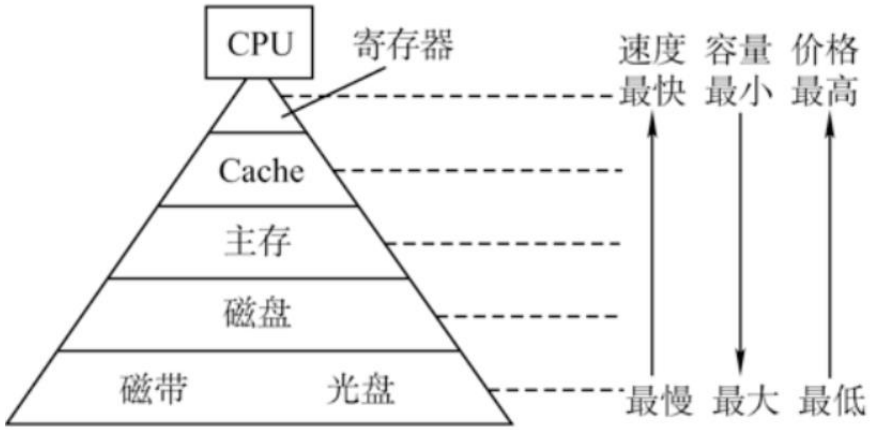
- 若仅对计算机一部分做性能提速，则改进越多，所得到的总性能提升越有限

因此，要想办法增加Fe的比例，即加速经常发生的事件

如何让计算机更快?

- 加速经常发生的事件-示例
  - 存储系统加速

存储器	硬件介质	单位成本（美元/MB）	随机访问延时
L1 Cache	SRAM	7	1ns
L2 Cache	SRAM	7	4ns
Memory	DRAM	0.015	100ns
Disk	SSD（NAND）	0.0004	150μs
Disk	HDD	0.00004	10ms



分级存储体系：

相对不容易被访问的内容，放入便宜、容量大、速度慢的存储

相对容易被访问的内容，放在快速、昂贵的存储

如何让计算机更快?

- 加速经常发生的事件-示例
  - CISC，复杂指令集系统

多媒体功能常用，涉及大量浮点计算

浮点计算子程序 → 协处理器 → 浮点指令

CPU（80286）

减少了一个任务需要的指令数

奔腾芯片：增加浮点指令

如何让计算机更快?

- 复杂指令级的代价！
  - 越来越复杂的电路
  - 指令的时钟周期很难对齐
  - 流水线难管理
- RISC，精简指令级系统
  - 只保留最核心的指令
  - 全部资源优化保留的指令速度

两种对立统一的优化思路，从Amdahl定律看，RISC是更极致的增加Fe（加速事件占比）

如何让计算机更快?

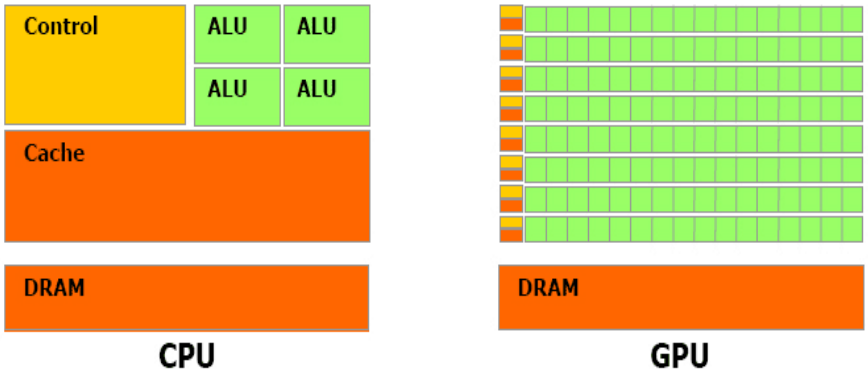
- 进一步的认识与发现
  - 各领域计算模式差异化，采用相同结构通用处理器进行计算，效率低下。
  - 大量指令很少使用，晶体管资源大量浪费。

David A. Patterson

- 新思路：领域特定结构（DSA-Domain Specific Architecture），针对不同领域的计算模式，设计专用的硬件加速器或计算机体系结构。

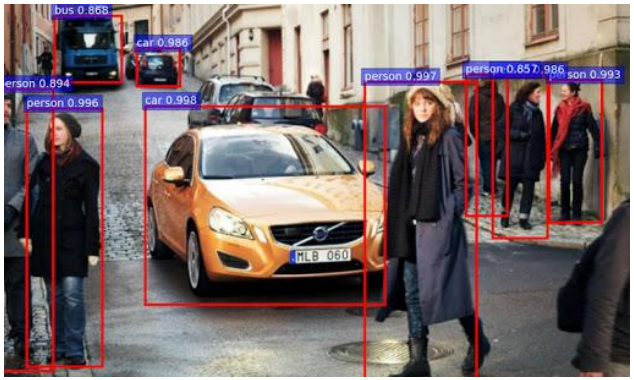
如何让计算机更快?

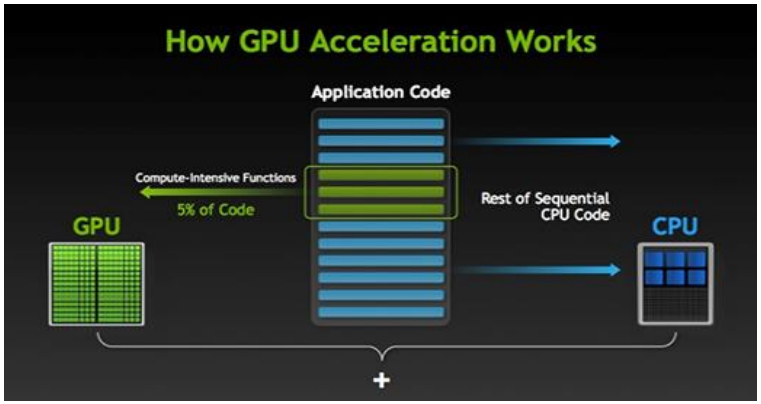
- DSA结构实例-GPU的兴起



相比于CPU

- GPU 拥有数以千计的计算核心，可高效地处理并行任务；
- GPU牺牲了逻辑处理能力，得到更强的大规模计算能力。





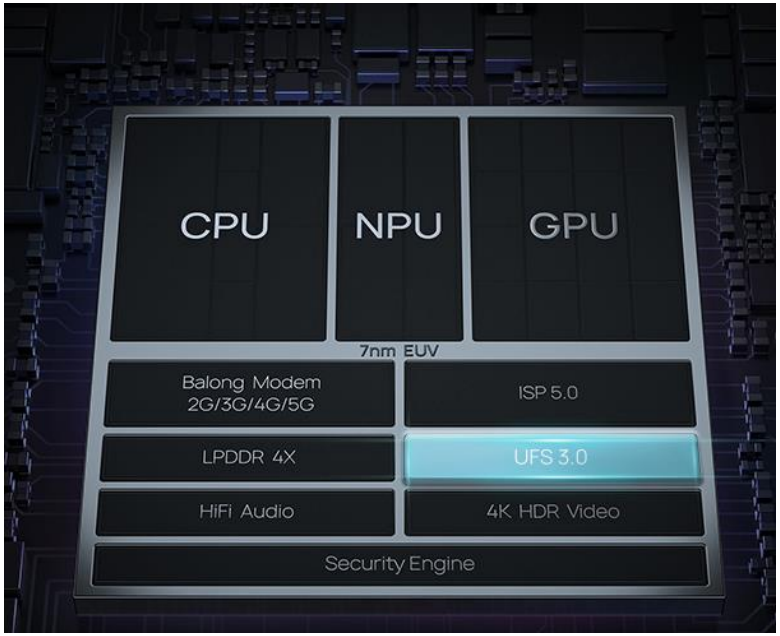
计算密集型任务分配给GPU运算

如何让计算机更快?

- DSA结构实例-GPU的兴起

采用GPU的超级计算机-天河一号

手机芯片麒麟990



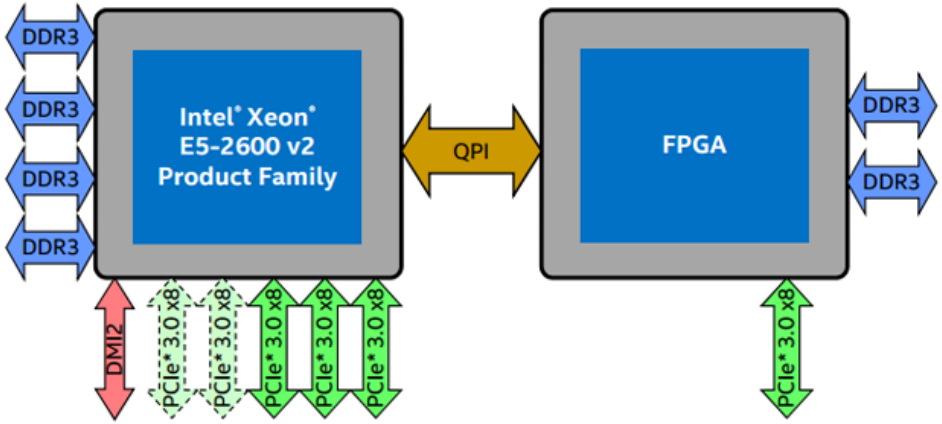
如何让计算机更快?

- DSA结构实例-CPU+FPGA可重构架构

通用CPU

通用CPU速度慢，无法满足特定计算需求

FPGA加速（Field Programable Gate Array)



CPU+FPGA架构

如何让计算机更快?

- DSA结构实例-CPU+FPGA可重构架构

Intel公司收购Altera公司

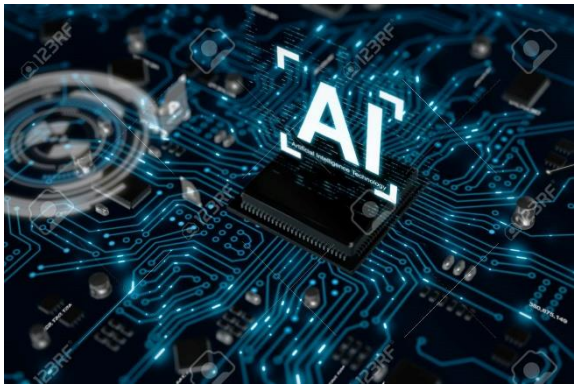
从分离到集成，构建可重构平台

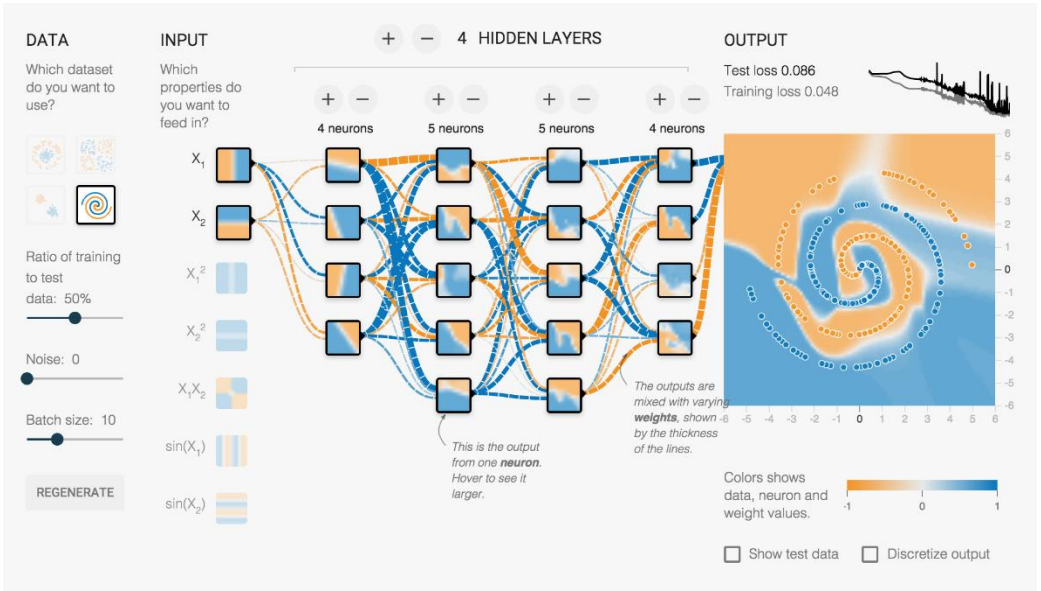
Xilinx公司收购了AutoESL

如何让计算机更快?

- DSA结构实例-CPU+AI加速器

AI加速器是一种针对深度学习算法的硬件处理器



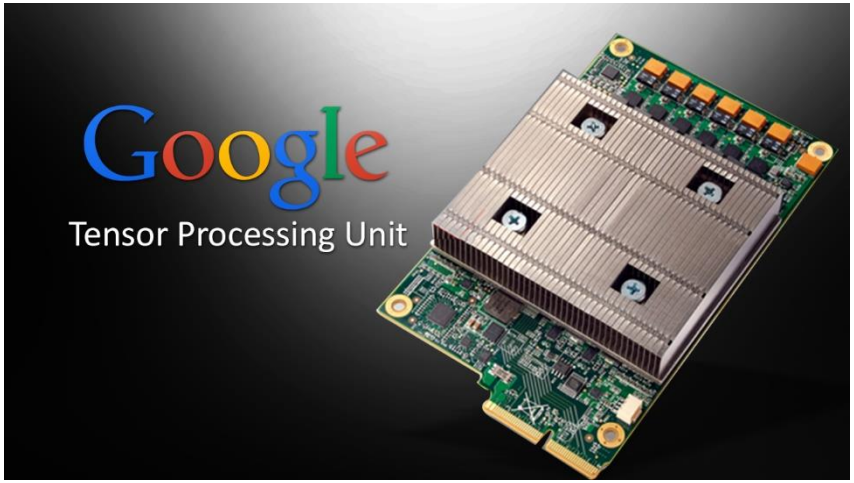


如何让计算机更快?

- DSA结构实例-CPU+AI加速器

中科院寒武纪芯片

GOOGLE的TPU处理器



阅读材料

- https://www.starduster.me/2020/11/05/modern-microprocessors-a-90-minute-guide/

作业

- 这堂课的内容对你将来写程序有什么启示？为了写出更快的程序，你会考虑哪些内容？
- 什么是ISA？为什么ISA与硬件功能密切相关？
- 阅读c语言的fork()函数，尝试写出一个并行程序。输入为100000个数字，让进程1对前50000个数排序，进程2对后50000个数排序，然后对整体100000个数字按顺序输出。

西安交通大學
XI'AN JIAOTONG UNIVERSITY

第五讲（part 2）：
计算机系统结构概述

师斌

School of Computer Science & Technology

Related courses

Computer Hardware Design

- Computer Organization
- Digital Logic Design
- Circuit
- Computer Architecture
- ……

Instruction Set Design

- Assembly Language
- Compile
- Computer Architecture
- ……

Very many people are concerned with computer function; Many people design computer components; Few design instruction sets! Very few people design computers!

Computer Architecture

- 什么是计算机系统结构？

中国砖木建筑

欧洲石材建筑

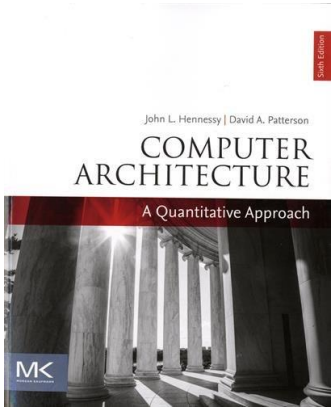
现代钢结构玻璃建筑

- 建筑师用不同的材料设计建筑；计算机架构师用晶体管组成的部件设计计算机

计算机系统结构- 定义

“computer architecture covers three aspects of computer design:

- instruction set architecture,
- computer organization and
- computer hardware. ”

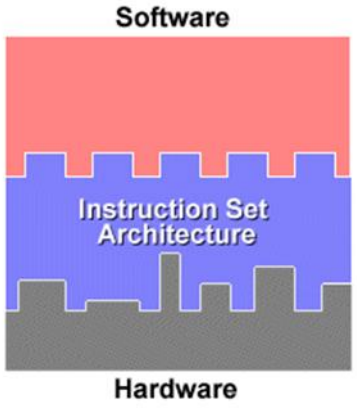


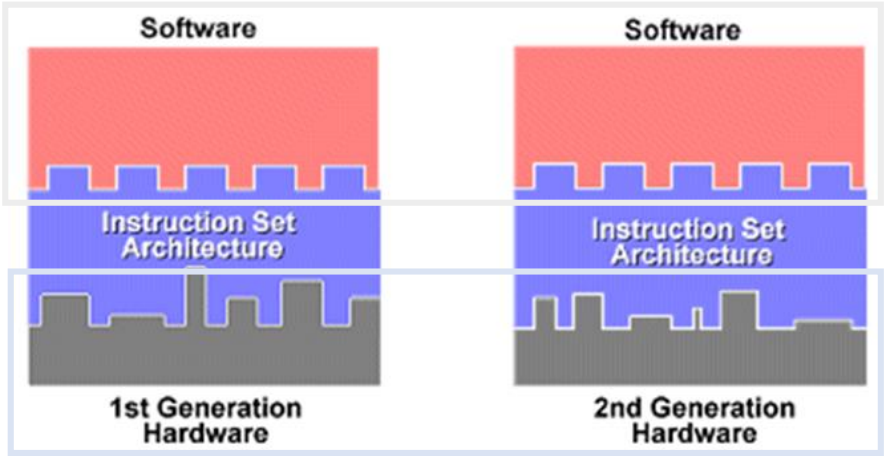
——John L. Hennessy & David A. Patterson, 1990

因计算机体系结构方面做出的突出贡献，获得2017年图灵奖

Instruction Set Architecture（ISA）

- ISA is defined as group of commands and operations used by the software to communicate with the hardware. It acts as an interface between the hardware and the software, specifying a computer “What to do?”
- 可以认为指令集是一个计算机的硬件系统所提供的全部功能





The programs written for a particular IS could run on any machine that implemented that IS.

Manufacturers could innovate and fine-tune that hardware for performance without worrying about breaking the existing software base.

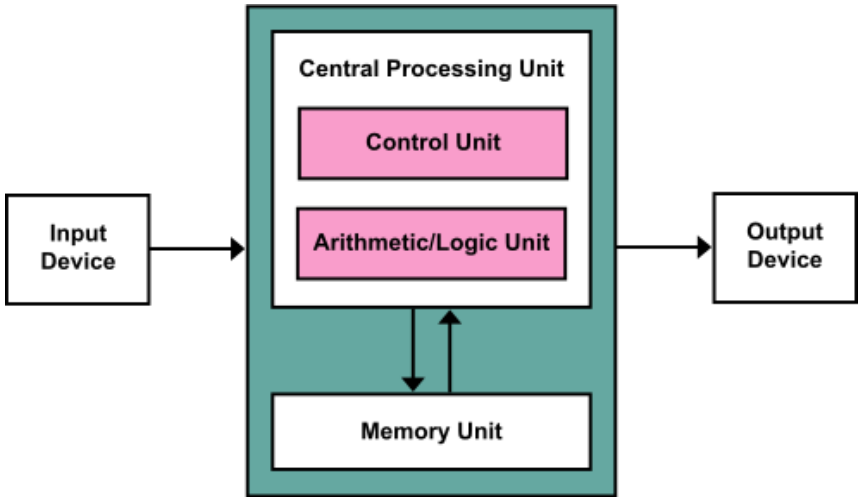
Computer Organization

- Computer organization refers to the way in which the hardware components of a computer system are arranged and interconnected. It implements the provided computer architecture and covers the "How to do?" part.

计算机之父

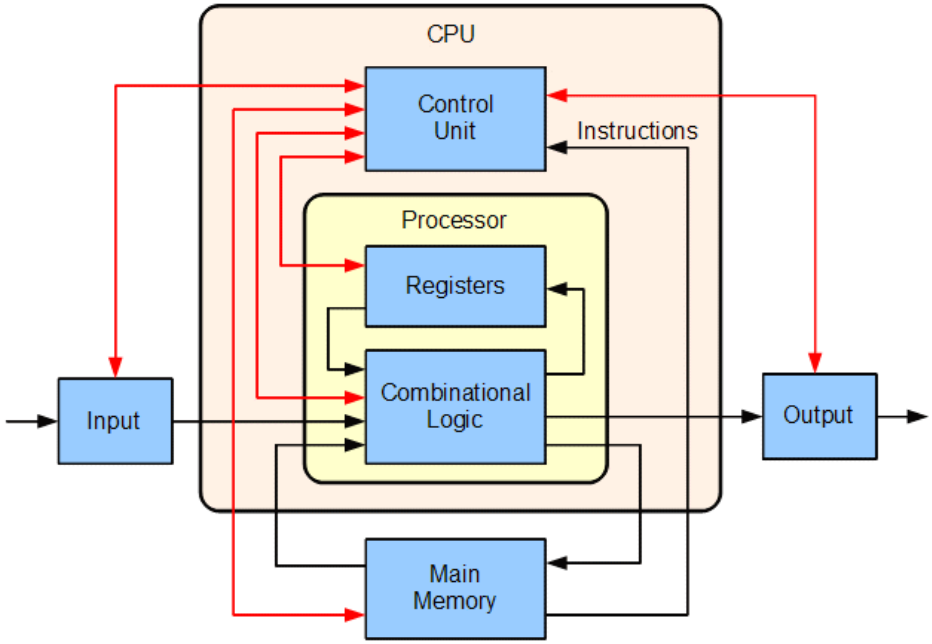
约翰·冯·诺依曼

John von Neumann



Von Neumann architecture scheme

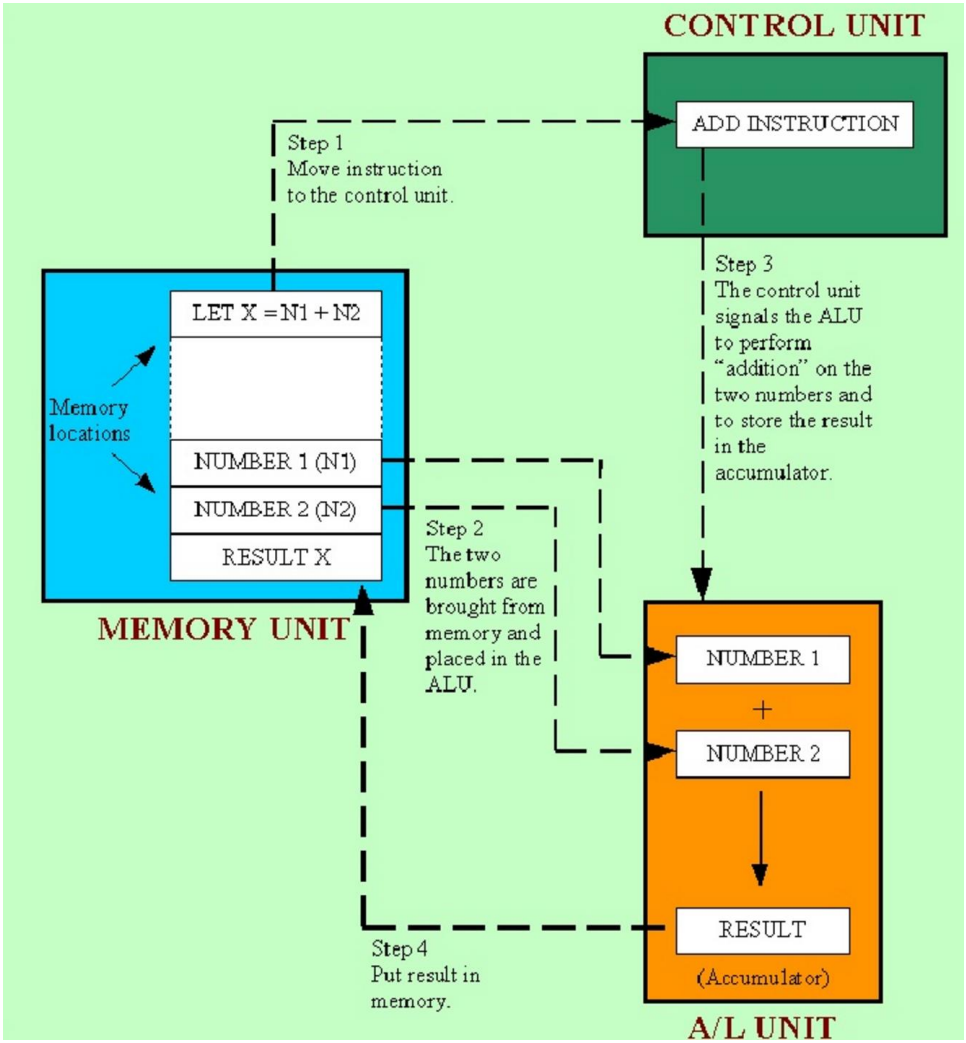
Computer Organization



CPU里有哪些主要部件呢？

- **Arithmetic/Logic Unit (ALU)：执行算术和逻辑运算**;
- **Registers：向 ALU 提供操作数并存储 ALU 运算的结果;**
- **Control Unit (CU)： 从内存中获取指令并通过指导 ALU、寄存器和其他组件的协调操作来“执行”它们。**

Computer Organization



CPU 如何工作?

步骤 1：**CU** 从主存（物理内存，RAM）或寄存器中获取指令。

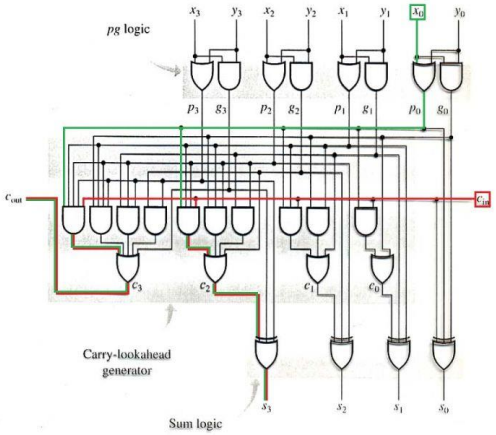
第 2 步：**CU** 确定指令的含义，并将必要的数据从主存或寄存器移动到 **ALU**。

第 3 步：**ALU** 对数据执行实际操作。

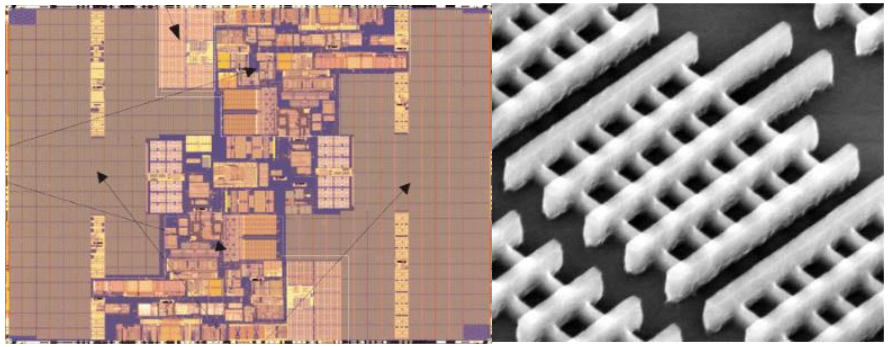
第 4 步：运算结果存储在主存或寄存器中。

Computer hardware

- Computer hardware refers to the specifics of a machine, included the detailed logic design and the packaging technology of the machine.



逻辑设计-四位加法器



芯片电路

Computer organization and computer hardware are two components of the implementation of a machine

计算机系统结构的任务

- 目标：设计计算机，在满足功能要求的条件下，优化成本、功耗和性能目标。

- 所以说……
  - 计算机系统结构是一门工程科学
  - 需要学习如何在满足各类约束条件下提出解决方案

- 最重要的目标是？
  - 性能！快！

Don’t memorize instances; understand why it is that way