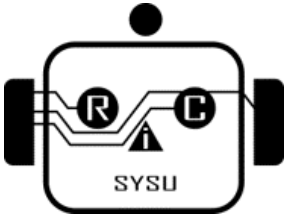
中山大學
SUN YAT-SEN UNIVERSITY



计算机组成原理

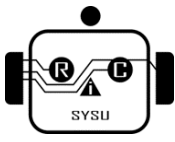
第一章：计算机概要与技术

中山大学计算机学院
陈刚

2022年秋季

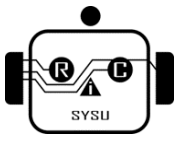
本讲内容

- 计算机系统概述
  - 技术发展历程
  - 国内计算机系统
- 计算机的基本组成
- 计算机拆解
- 计算机性能评价与性能指标



计算机性能评价与性能指标

- 衡量计算机性能的基本指标
  - 响应时间(Response Time)
    - 执行时间(Execution Time)、等待时间(Latency)
  - 吞吐率(Throughput)
    - 单位时间内执行的任务数
  - 指令执行速度(MIPS、MFLOPS)
- 功耗(Power)
- 制造成本(Manufacturing Cost)



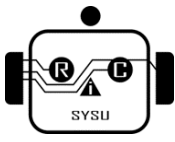
计算机性能的基本指标

“计算机X 比 计算机Y 快”

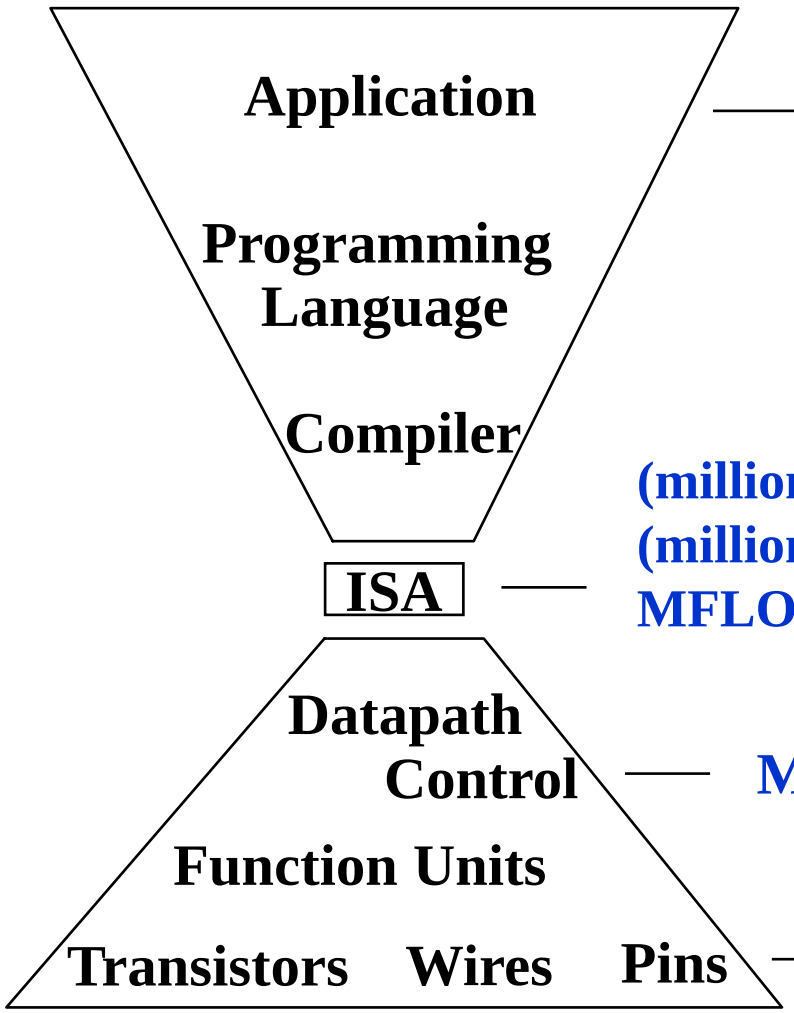
- 计算机有两种不同的性能
  - 完成单个任务的时间
    - 响应时间(response time)：完成单个任务所需的总时间
      - 执行时间(execution time)
      - 等待时间或时延(latency)
  - 单位时间完成的任务量(Tasks per day, hour, sec, ns. ..)
    - 吞吐率(throughput)：单位时间内所完成的任务量
    - 带宽(bandwidth)

计算机性能的基本指标

- 计算机有两种不同的性能
  - 完成单个任务的时间
    - 响应时间(response time)、执行时间(execution time）、等待时间或时延(latency)
  - 单位时间完成的任务量
    - 吞吐率(throughput)、带宽(bandwidth)
- 不同应用场合用户关心的性能不同
  - 吞吐率高的场合——多媒体应用(音/视频播放要流畅)
  - 响应时间短的场合——事务处理系统(存/取款的速度要快)
  - 吞吐率高且响应时间短的场合——ATM、文件服务器、Web服务器等



不同层次上对吞吐率性能的度量



Answers per hour
Operations per second

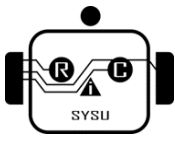
(millions) of Instructions per second – MIPS
(millions) of (F.P.) operations per second – MFLOPS

Megabytes per second(带宽)

Cycles per second (频率)

思考题

- 下面两种改进计算机系统的方式，能否减少其响应时间或增加其吞吐率？
  1. 将计算机中的处理器更换为更高速的型号。
  2. 增加多个处理器来分别处理独立的任务



计算机性能的基本指标

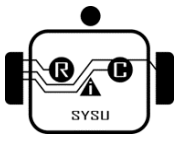
- 计算机的性能：响应时间（执行时间）的倒数

  Performance = 1 / Execution Time

- 计算机的基本性能评价标准是：CPU的执行时间

" X is n times faster than Y" means

$$\frac{\text{Performance(X)}}{\text{Performance(Y)}} = \frac{\text{Execution Time(Y)}}{\text{Execution Time(X)}} = n$$



计算机性能的基本指标

- 计算机的性能：响应时间（执行时间）的倒数

  Performance = 1 / Execution Time

- 计算机的基本性能评价标准是：CPU的执行时间

" X is n times faster than Y" means

$$\frac{\text{Execution Time(Y)}}{\text{Execution Time(X)}} = \frac{\text{Performance(X)}}{\text{Performance(Y)}} = n$$

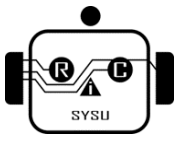
思考题：相对性能计算

如果计算机A运行一个程序需要10秒，计算机B运行同样的程序需要15秒，计算机A比计算机B快多少？

解： A的速度是B的1.5倍。

计算机性能的测量

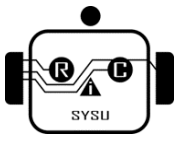
- 比较计算机性能时，如何用执行时间来衡量
  - 完成同样工作量所需时间最短的那台计算机性能最好
  - 处理器时间往往被多个程序共享使用，因此，用户感觉到的程序执行时间并不是程序真正的执行时间
  - 响应时间（用户感觉到的）
    - CPU时间：CPU真正花费在程序执行上的时间：
      - 用户CPU时间：用来运行用户代码的时间
      - 系统CPU时间：为执行用户程序而需运行一些操作系统代码的时间
    - 其他时间：等待I/O操作完成或CPU花在其他用户程序的时间



计算机性能的测量

- 响应时间（用户感觉到的）
  - CPU时间：CPU真正花费在程序执行上的时间：
    - 用户CPU时间：用来运行用户代码的时间
    - 系统CPU时间：为执行用户程序而需运行一些操作系统代码的时间
  - 其他时间：等待I/O操作完成或CPU花在其他用户程序的时间
- 计算机系统性能 ≠ CPU性能
  - 系统性能(System performance)：表示系统响应时间，与CPU之外的其他部分都有关系
  - CPU性能(CPU performance)：表示用户CPU时间

主要讨论CPU性能：CPU真正用在用户程序执行上的时间



CPU执行时间的计算

- 评价CPU性能的最重要指标——CPU执行时间

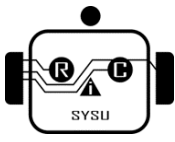
  一个程序的CPU执行时间=

  执行一个程序需要的(计算机)基本时间单元的数量 × 基本时间单元

- **时钟周期（Clock cycle）**
  - 所有计算机都有一个固定频率的硬件时钟，决定各种硬件事件发生和执行的时间和顺序
  - 硬件时钟所产生的离散时间间隔称为时钟周期
- **时钟频率(Clock rate)**
  - 时钟周期的倒数

例：时钟周期为250ps，对应的时钟频率为多少？

时钟频率 = 1/时钟周期 = 1/250ps = 4GHz



CPU执行时间的计算

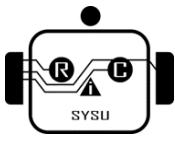
➢CPU执行时间的计算

- 评价CPU性能的最重要指标——CPU执行时间

  一个程序的CPU执行时间

  = 一个程序的CPU时钟周期数×时钟周期时间

  = 一个程序的CPU时钟周期数÷时钟频率



CPU执行时间的计算

➢CPU执行时间的计算

- 评价CPU性能的最重要指标——CPU执行时间

一个程序的CPU执行时间

= 一个程序的CPU时钟周期数 × 时钟周期时间

= 一个程序的CPU时钟周期数 ÷ 时钟频率

一个程序的CPU时钟周期数

=程序的指令数 ×每条指令的平均时钟周期数

CPI(avg. clock Cycles Per Instr):
每条指令的平均时钟周期数

CPU性能的影响因素

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

一个程序的CPU时间 = 指令数/程序 × CPI × 时钟周期时间

CPU性能的影响因素

一个程序的CPU时间 = 指令数/程序 × CPI × 时钟周期时间

	指令数	CPI	时钟周期时间
算法和编程语言	X	X	
编译程序	X	X	
指令集体系结构	X	X	X
计算机组成		X	X
实现技术			X

问题：ISA、计算机组成(Organization)、计算机实现技术(Technology)三者的关系是什么？

体系结构 = 指令集体系结构 + 计算机组成

Computer Design

Instruction Set Design

- **Machine Language**
  - **Compiler View**
  - **Computer Architecture**

"Building Architect"

“建筑设计师”

功能定义与设计

Computer Hardware Design

- Machine Implementation
  - Logic Designer's View
  - Computer Organization

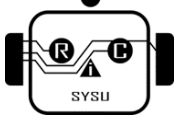
"Construction Engineer"

“建造工程师”

考虑用什么材料，如何布线等

例如：

- 指令集体系结构(ISA)设计考虑：是否提供“乘法指令”？
- 组成(Organization)考虑：如何实现乘法指令(用专门的乘法器还是用一个加法器+移位器实现)
- 计算机实现技术(Technology)考虑：如何布线、用什么材料、工艺？



指令集体系结构——计算机说话的语言及语法

主要内容包括：

- Instruction Formats(指令格式)
- Instruction (or Operation Code) Set (操作码集合：指令功能)
- Organization of Programmable Storage(程序员可见的存储组织)

  如：寄存器个数、名称、长度；内存单元长度、主存地址长度
- Data Types & Data Structures (数据类型和结构)

  Encodings & Representations (编码和表示)
- Modes of Addressing and Accessing Data Items and Instructions

  (寻址方式：数据 / 指令的存取方式)
- Exceptional Conditions and handle (异常条件和处理)

计算机组成

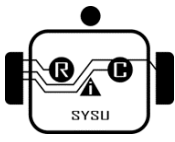
Logic Designer's View → **ISA Level**

FUs & Interconnect

- 为实现ISA，应该如何设计功能部件
- 通常采用寄存器传送级进行描述

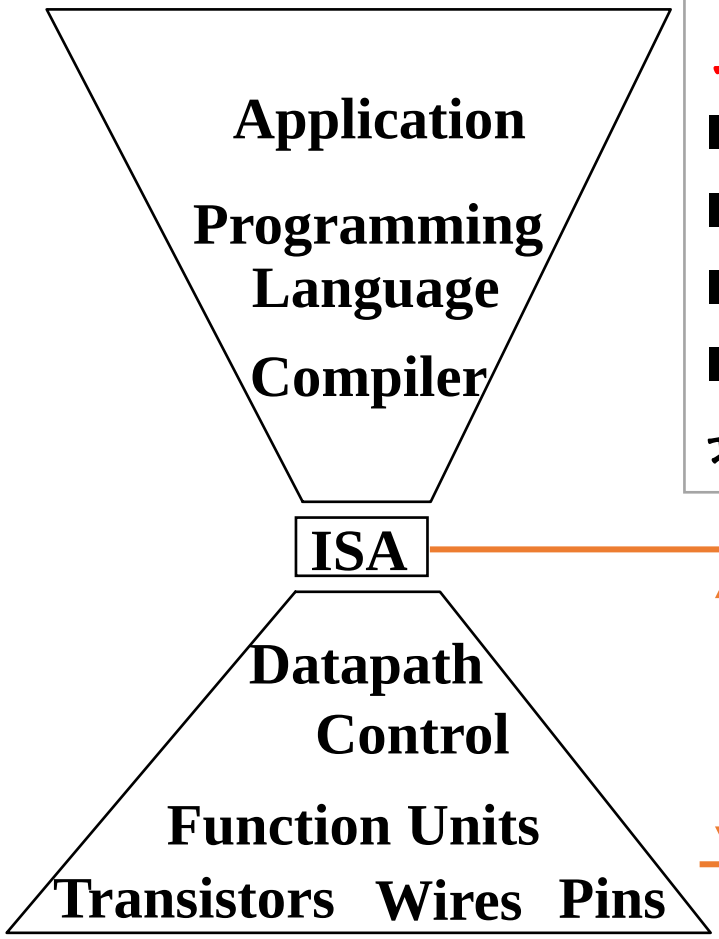
主要内容包括：

- Capabilities & Performance Characteristics of Principal Functional Units
  - 主要功能部件的能力和工作特性
  - e.g., Registers, ALU, Shifters, Memory, Cache, etc.
- Ways in which these components are interconnected
  - 互连方式
- Nature of information flows between components
  - 部件之间的信息流动方式
- Logic and means by which such information flow is controlled
  - 部件之间信息流动的控制逻辑和控制方法



计算机组成的平衡

CPU time = Instruction counts × CPI × Cycle Time



三个因素之间的相互影响？

- CPI的减少可能会增加时钟周期的长度
- 缩短时钟周期可能会增加指令的条数
- 改变IS以减少指令条数会使时钟周期变长
- 即使是在同一台机器上解决同一个问题，最少指令条数的程序不一定执行的最快

Instruction Count

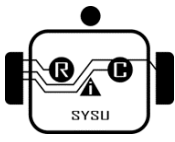
CPI **3 factors：Where are they?**

Cycle Time

评价性能时，必须在各个因素进行权衡！

例：改变一个因素可能影响另一个因素

一个程序在一台时钟频率4GHz的计算机A上运行10秒，现在我们要设计一台时钟频率更高的计算机B，将该程序在计算机B上的执行时间降低为6秒。



例：改变一个因素可能影响另一个因素

一个程序在一台时钟频率4GHz的计算机A上运行10秒，现在我们要设计一台时钟频率更高的计算机B，将该程序在计算机B上的执行时间降低为6秒。

提高时钟频率将影响CPU其他方面的设计，使得在计算机B上运行该程序的时钟周期数增长为1.2倍。请分析计算机B的时钟频率？

	机器A	机器B
CPU时钟频率	4GHz	？？
程序运行的时钟周期数	I	I × 1.2 ↑
执行时间	10秒	6秒 ↓

例：改变一个因素可能影响另一个因素

一个程序在一台时钟频率4GHz的计算机A上运行10秒，现在我们要设计一台时钟频率更高的计算机B，将该程序在计算机B上的执行时间降低为6秒。

提高时钟频率将影响CPU其他方面的设计，使得在计算机B上运行该程序的时钟周期数增长为1.2倍。请分析计算机B的时钟频率？

	机器A	机器B
CPU时钟频率	4GHz	8GHz
运行的时钟周期数	I	I × 1.2
执行时间	10秒	6秒

解答：

$$\text{CPU时间}_A = \text{CPU时钟周期数}_A / \text{时钟频率}_A$$

$$\text{CPU时钟周期数}_A = 10\ \text{sec} \times 4\ \text{GHz}$$

$$\text{时钟频率}_B = \text{CPU时钟周期数}_B / \text{CPU时间}_B$$

$$= 1.2 \times 10\ \text{sec} \times 4\ \text{GHz} / 6\ \text{sec} = 8\text{GHz}$$

机器B的时钟频率是A的两倍，但机器B的速度并不是A的两倍！

如何计算CPI?

对于某一条特定的指令而言，其CPI是一个确定的值。但是，对于某一类指令、或一个程序、或一台机器而言，其CPI是一个平均值，表示该类指令或该程序或该机器中平均每条指令执行所需的时钟周期数。

设CPI_i、F_i是每类指令的CPI和程序中出现的频率，则程序综合CPI:

$$CPI = \sum_{i=1}^{n} CPI_i \times F_i$$

已知CPU时间、时钟频率、总时钟数、指令条数，则程序综合CPI：

CPI = (CPU 时间 × 时钟频率) / 指令条数 = 总时钟周期数 / 指令条数

注意：单靠CPI是不能反映CPU性能的！为什么？

如：单周期处理器CPI = 1，但性能差！

例2：CPU执行时间的计算——性能公式的使用

- 假设有相同指令级的两种不同实现方式：
  计算机A的时钟周期为250ps，对某程序的CPI为2.0
  计算机B的时钟周期为500ps，对同样程序的CPI为1.2
- 请问：对于该程序，请问哪台计算机执行的速度更快？快多少？

理解程序性能

名称	影响因素	如何影响
算法	指令数、CPI	算法决定源程序的指令条数，因此决定了处理器执行的指令条数。算法由于对慢速或快速指令的不同倾向性而同样影响了CPI。
编程语言	指令数、CPI	由于编程语言中的语句被翻译成CPU指令，而后者决定了指令条数，因此编程语言势必会影响指令条数。由于编程语言的自身特点，它可能同样影响CPI。例：强支持数据抽象的语言(如Java)要求间接调用，而这往往会用到具有高CPI的指令
编译器	指令数、CPI	编译器的效率对指令数及每条指令的平均周期数均有影响，这是因为编译器决定了从源语言语句到机器指令的翻译。编译器的作用可以非常复杂，同时通过复杂的方式影响CPI
指令集体系结构	指令数、时钟频率、CPI	指令集的结构对CPU性能的这三个方面均有影响，因为它影响了函数需要的指令、每条指令需要周期及CPU总的时钟频率

例3：比较不同的代码段

一个编译器设计者试图在两个代码序列之间进行选择，硬件设计者给出了如下数据：

	CPI for this instruction class		
	A	B	C
CPI	1	2	3

对于某行高级语言语句的实现，两个代码序列所需的指令数量如下：

Code sequence	Instruction counts for instruction class		
	A	B	C
1	2	1	2
2	4	1	1

哪个代码序列执行的指令数更多？哪个执行速度更快？每个代码序列的CPI是多少？

Marketing Metrics （产品宣称指标）

- **MIPS** = Instruction Counts / Execution Time $\times 10^6$

  = Clock Rate / CPI $\times$ 10^6

  - 一种用来代替执行时间的指标
  - **M**illion **I**nstructions **P**er **S**econds

    因为每条指令执行时间不同，所以MIPS是一个平均值

  - 不同机器的指令集不同
  - 程序由不同的指令混合而成
  - 指令使用的频度动态变化
  - Peak MIPS(不实用)

  用MIPS数表示性能有局限

- **MFLOPS** = FP Operations / Execution Time $\times$ 10^6

  Million **Flo**ating-point **O**perations **P**er **S**econd

讨论题

- 下面是一个程序的性能评测结果：

评测内容	机器A	机器B
指令数	1千万	8百万
时钟频率	4GHz	4GHz
CPI	1.0	1.1

请回答下列问题：

① 哪个机器具有更高的MIPS？

② 哪个机器更快？

$$\text{CPU time} = \text{Instruction counts} \times \text{CPI} \times \text{Cycle Time}$$

$$\text{Cycle Time} = 1 / \text{Clock Rate}$$

$$\text{MIPS} = \text{Instruction Counts} / (\text{Execution Time} \times 10^6)$$

$$= \text{Clock Rate} / (\text{CPI} \times 10^6)$$

性能比较与综合评价

问题：如果用一组基准程序在不同的机器上测出了运行时间，那么如何综合评价机器的性能呢？

例：Program 1: 1 sec on machine A, 10 sec on machine B

Program 2: 1000 sec on A，100 sec

What are your conclusions?

无法比较A和B的好坏，怎么办？

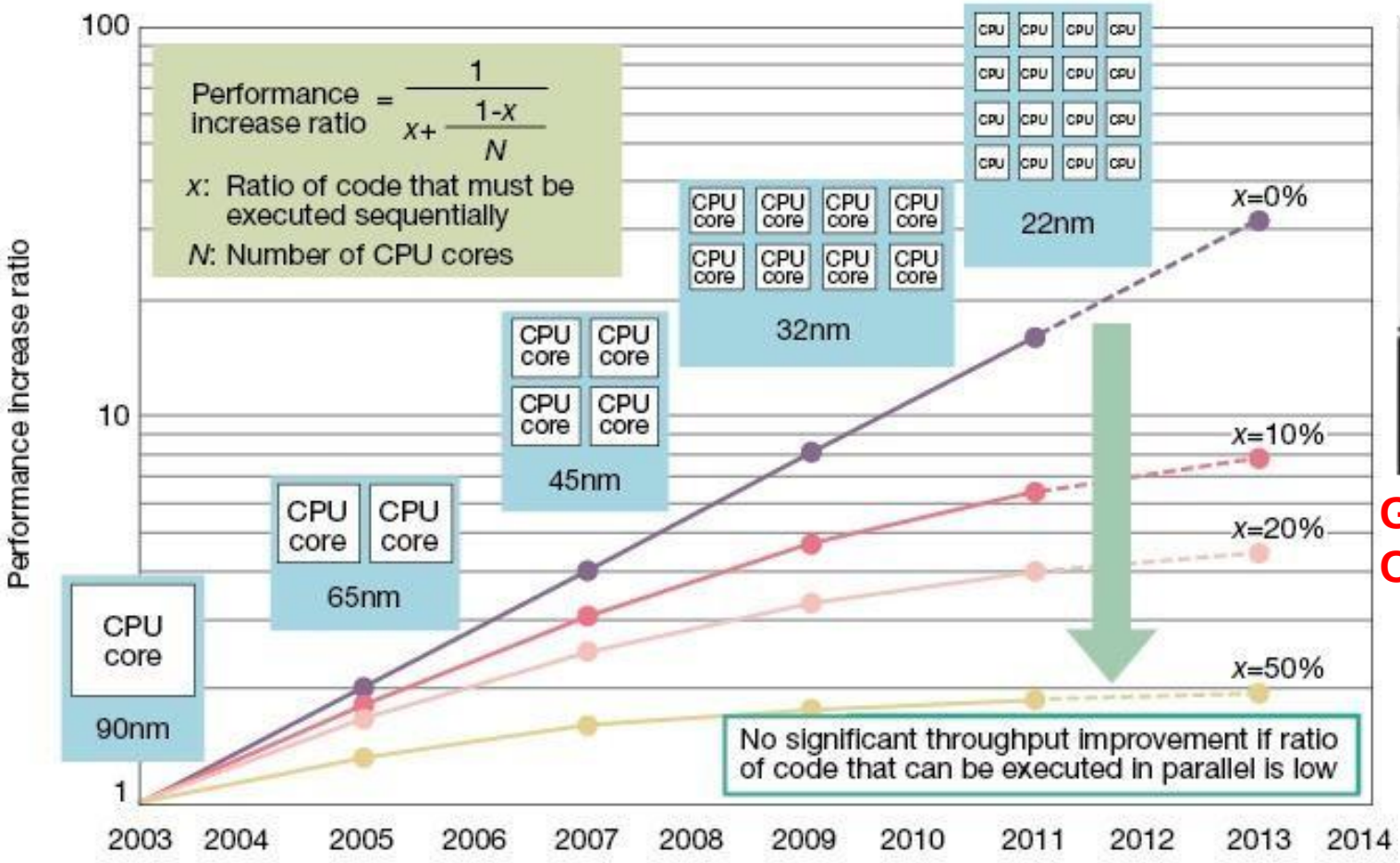
- A is 10 times faster than B for program1
- B is 10 times faster than A for Program2

□一种简单的综合评价指标——Total execution time

总执行时间 = 运行时间$_{程序1}$ × 出现频率$_{程序1}$

+ 运行时间$_{程序2}$ × 出现频率$_{程序2}$ + …

并行加速比：Amdahl's Law 1967

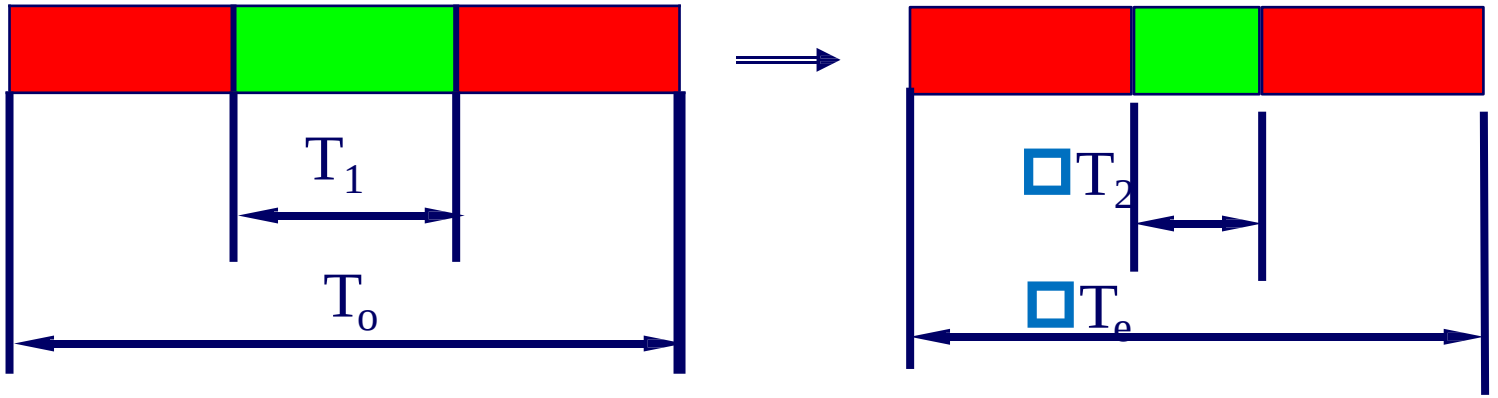


Gene Amdahl
Computer Pioneer

Fig 3 Amdahl's Law an Obstacle to Improved Performance Performance will not rise in the same proportion as the increase in CPU cores. Performance gains are limited by the ratio of software processing that must be executed sequentially. Amdahl's Law is a major obstacle in boosting multicore microprocessor performance. Diagram assumes no overhead in parallel processing. Years shown for design rules based on Intel planned and actual technology. Core count assumed to double for each rule generation.

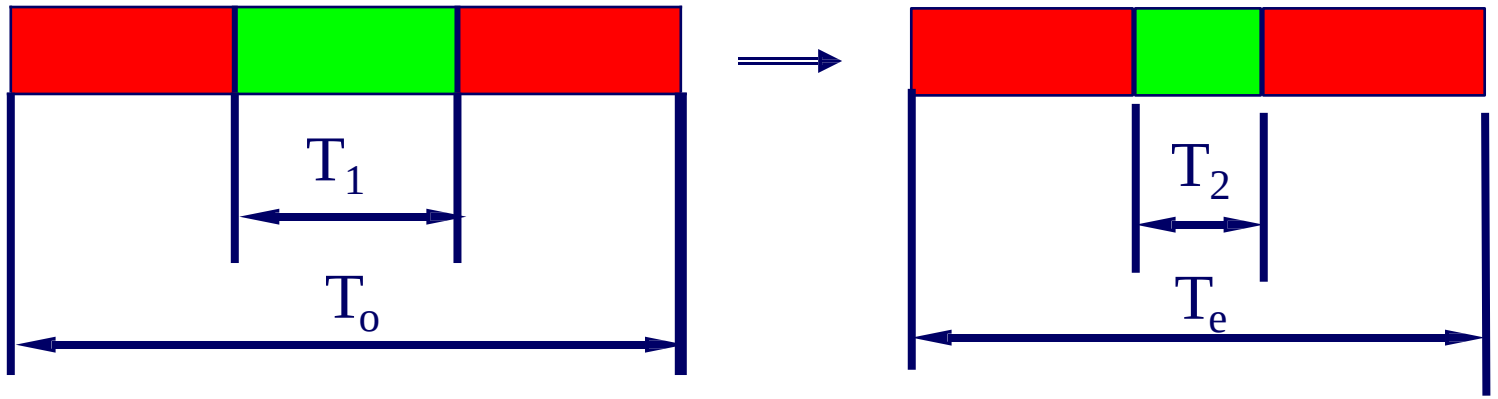
Amdahl定律

- 系统加速比依赖于两个因素：
- “可改进比例”：可改进部分在原系统计算时间中所占的比例，它总是小于等于1的。
- “部件加速比”可改进部分改进以后的性能提高，一般情况下它是大于1的。



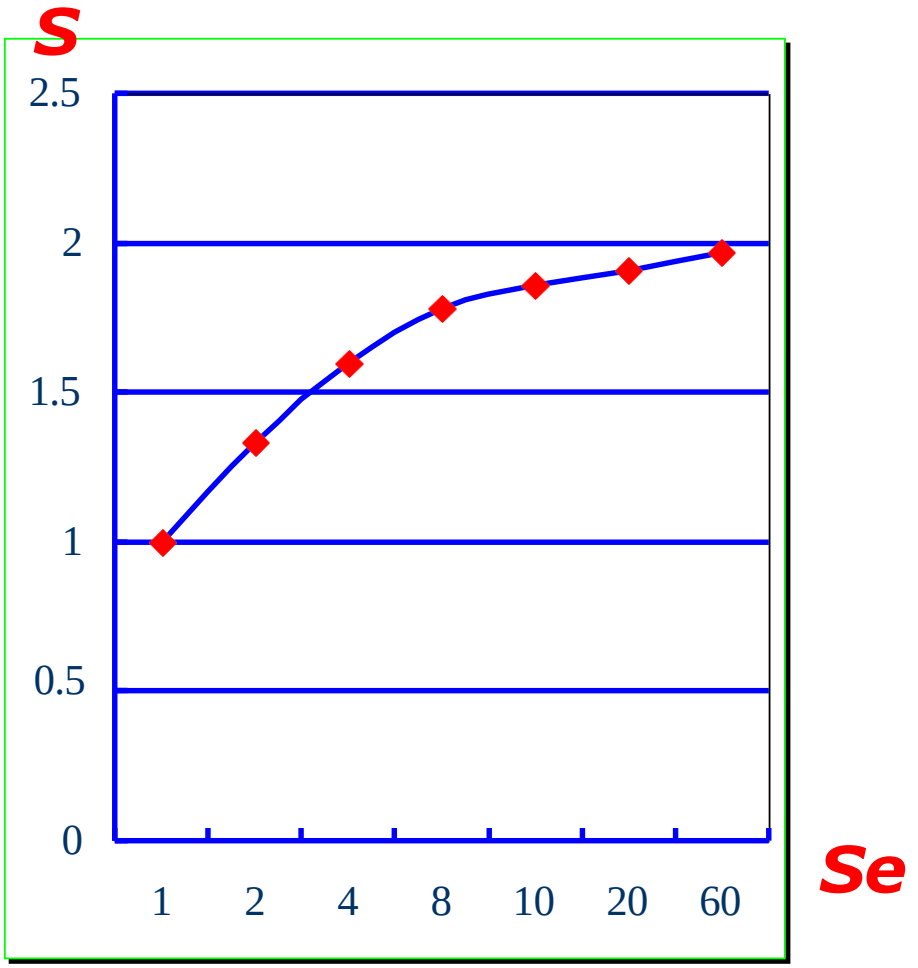
$$f_e = T_1 / T_o \qquad S_e = T_1 / T_2$$

$$T_e = T_o\left[(1-f_e)+\frac{f_e}{S_e}\right] \qquad S = \frac{1}{(1-f_e)+\frac{f_e}{S_e}}$$



Amdahl定律

- 如果仅仅对计算机中的一部分做性能改进，则改进越多，系统获得的效果越小。
- 推论：如果只针对整个任务的一部分进行优化，那么所获得的加速比不大于$1/(1-f_e)$。



计算机某方面改进，系统的性能能成比例提高吗？

例：假设某个程序在某台计算机上运行时所需的时间是100秒，其中80秒是用来执行乘法操作。如果希望使该程序的速度提高到原来的5倍，乘法部件的速度应该是原来的多少倍呢？

计算机某方面改进，系统的性能能成比例提高吗？

例：假设某个程序在某台计算机上运行时所需的时间是100秒，其中80秒是用来执行乘法操作。如果希望使该程序的速度提高到原来的5倍，乘法部件的速度应该是原来的多少倍呢？

Amdahl定律：

$$\text{改进后的执行时间} = \frac{\text{受改进影响部分的执行时间}}{\text{改进提高的倍数}} + \text{不受影响的执行时间}$$

因为程序执行速度提高到原来的5倍，故新的执行时间应该是100/5=20秒

20秒= 80秒/n +(100-80)秒，　　0 = 80秒/n

结论：如果乘法只是占到总计算量的80%的话，无论对乘法部件做何种改进，系统性能都不可能提高到原来的5倍。

计算机某方面改进，系统的性能能成比例提高吗？

例：假设某个程序在某台计算机上运行时所需的时间是100秒，其中80秒是用来执行乘法操作。如果希望使该程序的速度提高到原来的5倍，乘法部件的速度应该是原来的多少倍呢？若其中90秒用来执行乘法操作，结果又会如何？

Amdahl定律：

$$\text{改进后的执行时间} = \frac{\text{受改进影响部分的执行时间}}{\text{改进提高的倍数}} + \text{不受影响的执行时间}$$

因为程序执行速度提高到原来的5倍，故新的执行时间应该是100/5=20秒

20秒= 90秒/n +(100-90)秒， 10 = 90秒/n， n = 9

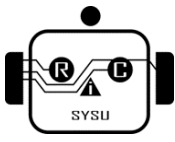
乘法部件的速度需要达到原来的9倍.

硬件设计的基本策略：使最常用的部件执行得更快

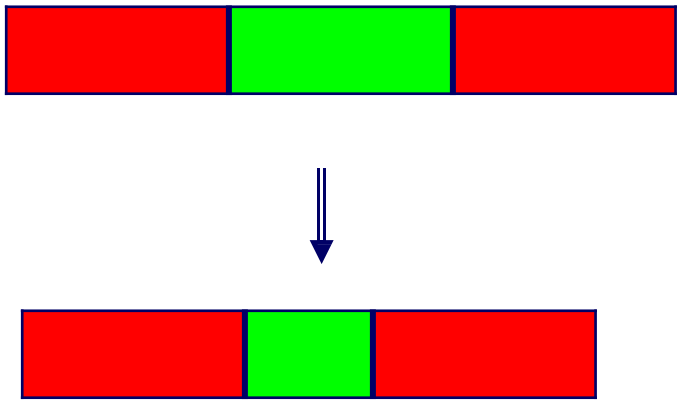
Amdahl定律练习

例1：假设在某程序的执行过程中，浮点操作时间占整个执行时间的10%，现希望对浮点操作加速。

- ✓ 设对浮点操作的加速比为S_f。请画出程序总的加速比S和S_f之间的关系曲线；
- ✓ 请问程序的最大加速比可达多少？



Amdahl定律练习



$$S_{\max} = \lim_{S_f \to \infty} \frac{1}{0.9 + \frac{0.1}{S_f}}$$

$$= 10/9$$

$$S = \frac{1}{(1 - f_f) + \frac{f_f}{S_f}}$$

$$= \frac{1}{(1-10\%) + \frac{10\%}{S_f}}$$

$$= \frac{1}{0.9 + \frac{0.1}{S_f}}$$

Amdahl定律练习

例2 ：求平方根和浮点乘是图形应用中两种常用的转换。假设浮点操作在某机器的一个基准程序中占总执行时间的50%，求平方根操作在总执行时间的20%，现通过两种方法加速操作：

①增加专门的软件加速库处理求平方根，使其执行速度为原来的10倍；

②通过调整流水线把浮点速度提高为原来的1.6倍。

问：分别采用两种方法增强后此基准程序加速比各是多少？

- (1)

$$S = \frac{1}{(1 - f_e) + \frac{f_e}{S_e}}$$

$$= \frac{1}{(1 - 0.2) + \frac{0.2}{10}}$$

$$= 1.22$$

- (2)

$$S = \frac{1}{(1 - f_e) + \frac{f_e}{S_e}}$$

$$= \frac{1}{(1 - 0.5) + \frac{0.5}{1.6}}$$

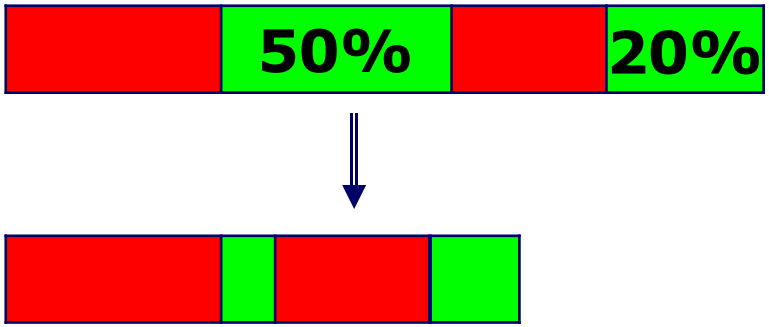
$$= 1.23$$

Amdahl定律练习

例3：求平方根和浮点乘是图形应用中两种常用的转换，假设求平方根操作在某机器的一个基准程序中占总执行时间的20%，浮点乘操作在该程序中占总执行时间50%，现通过两种方法加速两操作：①增加专门的软件函数加速库求平方根，使其执行速度为原来的10倍；②通过调整流水线把浮点乘速度提高为原来的1.6倍。

问：同时采用两种方法增强后此基准程序加速比是多少？

Amdahl定律练习



加速比= 增强前时间/增强后时间

=1/((1-0.5-0.2)+0.2/10+0.5/1.6)

=1/ (0.3+0.2/10+0.5/1.6)

=1/0.6325

=1.58

Amdahl定律练习

例4：求平方根、浮点乘和定点乘是图形应用中三种常用的转换。假设浮点操作在某机器的一个基准程序中占总执行时间的30%，求平方根操作在总执行时间的20%，定点乘15%。现通过三种方法加速操作：

①增加专门的硬件处理求平方根，使其执行速度为原来的10倍；

②通过调整流水线把浮点速度提高为原来的1.6倍。

③通过调整流水线把定点速度提高为原来的3倍。

问：分别采用三种方法增强后此基准程序加速比各是多少？

同时采用三种方法增强后此基准程序加速比各是多少？

Amdahl定律练习

- (1)平方根，fe=0.2, Se=10

$$S = \frac{1}{(1-f_e)+\frac{f_e}{S_e}}$$

$$= \frac{1}{(1-0.2)+\frac{0.2}{10}}$$

$$= 1.22$$

- (2)浮点，fe=0.3, Se=1.6

$$S = \frac{1}{(1-f_e)+\frac{f_e}{S_e}}$$

$$= \frac{1}{(1-0.3)+\frac{0.3}{1.6}}$$

$$= 1.13$$

Amdahl定律练习

- (3) 定点，fe=0.15, Se=3

$$S = \frac{1}{(1-f_e) + \frac{f_e}{S_e}}$$

$$= \frac{1}{(1-0.15) + \frac{0.15}{3}}$$

$$= 1.11$$

- (4) 三种方法同时使用

$$S = \frac{1}{(1-f_e) + \frac{f_e}{S_e}}$$

$$= \frac{1}{(1-0.3-0.2-0.15) + \frac{0.3}{1.6} + \frac{0.2}{10} + \frac{0.15}{3}}$$

$$= 1.65$$

SPEC CPU Benchmark

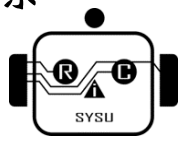
- 基准测试程序是专门进行性能评价的一组程序
  - 目的是为现代计算机系统建立标准的基准测试程序集
  - 用户希望基准测试程序能预测机器运行实际负载时的性能

SPECINTC2006基准程序在AMD Opteron x4 2356(Barcelona)的运行结果：

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^9)	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2,118	0.75	0.4	637	9,770	15.3
Block-sorting compression	bzip2	2,389	0.85	0.4	817	9,650	11.8
GNU C compiler	gcc	1,050	1.72	0.4	724	8,050	11.1
Combinatorial optimization	mcf	336	10.00	0.4	1,345	9,120	6.8
Go game (AI)	go	1,658	1.09	0.4	721	10,490	14.6
Search gene sequence	hmmer	2,783	0.80	0.4	890	9,330	10.5
Chess game (AI)	sjeng	2,176	0.96	0.4	837	12,100	14.5
Quantum computer simulation	libquantum	1,623	1.61	0.4	1,047	20,720	19.8
Video compression	h264avc	3,102	0.80	0.4	993	22,130	22.3
Discrete event simulation library	omnetpp	587	2.94	0.4	690	6,250	9.1
Games/path finding	astar	1,082	1.79	0.4	773	7,020	9.1
XML parsing	xalancbmk	1,058	2.70	0.4	1,143	6,900	6.0
Geometric Mean							11.7

计算机性能小结

- 性能定义：用程序响应时间或系统吞吐率表示系统整体性能
- CPU性能的测量（用户程序的CPU执行时间）
  - 一般把程序的响应时间划分成CPU时间和等待时间，CPU时间又分成用户CPU时间和系统CPU时间
  - 因为操作系统对自己所花费的时间进行测量时，不十分准确，所以对CPU性能的测算一般通过测算用户CPU时间来进行
- 各种性能指标之间的关系
  - CPU执行时间 = CPU时钟周期数 × 时钟周期
  - 时钟周期和时钟频率互为倒数
  - CPU时钟周期数 = 程序指令数×平均每条指令时钟周期数CPI
- MIPS数有些情况下不能说明问题，没有可比性！
- 针对特定指令集体系结构，提高计算机性能的主要途径有：
  - 提高时钟频率（第四章处理器中的流水线技术）
  - 优化处理器中数据通路结构以降低CPI（第四章 处理器）
  - 用编译优化措施来减少指令条数或降低指令复杂度（第二章 指令系统 ）



计算机整体性能评价

- 影响系统性能的硬件技术指标：

1）主频

- ◆定义：CPU的工作节拍是由时钟控制的，时钟不断产生固定频率的时钟脉冲，这个时钟的频率就是CPU的主频。
- ◆主频越高，CPU的工作节拍就越快，运算速度就越高
- ◆主频通常用一秒钟内处理器所能发出电子脉冲数来表示，单位一般为兆赫兹（MHz）
- ◆芯片的功耗与频率成正比

计算机整体性能评价

- 影响系统性能的硬件技术指标：

2）运算速度

◆定义：每秒钟所能执行的指令条数

◆计量单位：

- MIPS(百万条指令每秒)
- MFLOPS(百万次浮点运算每秒)、GFLOPS/PFLOPS

◆几种计算方法：

①吉布森混合法

②计算各种指令的执行速度

③计算典型程序的运算速度

④模型分析和模拟等其他方法

计算机整体性能评价

- 影响系统性能的硬件技术指标：

  3）运算精度

  - ◆计算机能直接处理的二进制位数
  - ◆一般和CPU中存储的数据寄存器的位数是相同。位数越多，精度越高
  - ◆参与运算的操作数的基本位数称之为基本字长。早期的微机字长为8位或16位，现为32位或64位

计算机整体性能评价

- 影响系统性能的硬件技术指标：

4) 存储容量

- 主存越大，处理问题的速度越快
- 与辅存交换次数越少，访存效率越高

主存容量 存储单元个数 × 存储字长

如： 4 GB($4*2^{30}$ Bytes)

辅存容量 字节数 512 GB($512*10^{9}$ Bytes)

计算机性能举例

2013年考研题：某计算机主频为1.2 GHz，其指令分为4类，它们在基准程序中所占比例及CPI如下表所示，该机的MIPS数是（　　）

A. 10　　B. 200　　C. 400　　D. 600

指令类型	所占比例	CPI
A	50%	2
B	20%	3
C	10%	4
D	20%	5

Million Instructions Per Second

百万级的指令条数/秒

计算机性能举例

例：若相同指令集有两种不同的实现I1和I2，在该指令集中有3类指令：A、B和C。I1的时钟频率为6GHz，I2的时钟频率为3GHz。下表给出在I1和I2上运行时所需的每类指令平均时钟周期数(CPI)：

指令类型	I1上的CPI	I2上的CPI	C1编译器	C2编译器	C3编译器
A	2	1	40%	40%	50%
B	3	2	40%	20%	25%
C	5	2	20%	40%	25%

表中还列出了3 个不同编译器生成的程序代码中这3类指令各自所占的百分比。其中，C1是由I1的生产商提供的编译器，C2是由I2的生产商提供的编译器，C3是由第三方生产商提供的。假设对于同一个程序，3个编译器生成的代码中指令数相同，但代码的组合不同，如上表所示。如果在I1与I2上使用C1编译器，那么I1的生产商可以声称其性能是I2的多少倍？如果在I1与I2上使用C2编译器，那么I2的生产商可以声称其性能是I1的多少倍？如果你购买了I1，那么你会选择哪个编译器？如果你购买了I2，那么你又会选择哪个编译器？如果所有其他指标都相同，那么你会买哪台机器和哪个编译器？

计算机性能举例

例：若相同指令集有两种不同的实现I1和I2，在该指令集中有3类指令：A、B和C。I1的时钟频率为6GHz，I2的时钟频率为3GHz。下表给出在I1和I2上运行时所需的每类指令平均时钟周期数(CPI)：

指令类型	I1上的CPI	I2上的CPI	C1编译器	C2编译器	C3编译器
A	2	1	40%	40%	50%
B	3	2	40%	20%	25%
C	5	2	20%	40%	25%

表中还列出了3 个不同编译器生成的程序代码中这3类指令各自所占的百分比。其中，C1是由I1的生产商提供的编译器，C2是由I2的生产商提供的编译器，C3是由第三方生产商提供的。假设对于同一个程序，3个编译器生成的代码中指令数相同，但代码的组合不同，如上表所示。如果在I1与I2上使用C1编译器，那么I1的生产商可以声称其性能是I2的多少倍？如果在I1与I2上使用C2编译器，那么I2的生产商可以声称其性能是I1的多少倍？如果你购买了I1，那么你会选择哪个编译器？如果你购买了I2，那么你又会选择哪个编译器？如果所有其他指标都相同，那么你会买哪台机器和哪个编译器？

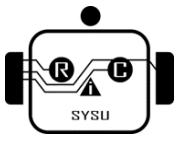
练　习

- 某台计算机只有Load/Store 指令能对存储器进行读/写操作，其它指令只对寄存器进行操作。根据对某程序跟踪实验结果，已知每种指令所占的比例及CPI数如下：

指令类型	指令所占比例	CPI
算逻指令	43%	1
Load指令	21%	2
Store指令	12%	2
转移指令	24%	2

计算机性能的评测

- 利用时间的概念评测一台计算机的性能和测试者所处的角度有关。
  - ✓响应时间（执行时间）：从事件开始到结束之间的时间。
  - ✓吞吐率(Throughput)：在单位时间内所能完成的工作量（任务）。（多用户系统）
- 用户以响应时间为标准，多道程序系统以吞吐率为标准。



联系方式

- Acknowledgements:
- This slides contains materials from following lectures:
  - Computer Architecture (ETH，NUDT，USTC)

- Research Area：
  - 计算机视觉与机器人应用计算加速，
  - 人工智能和深度学习芯片及智能计算机

- Contact：
  - 中山大学计算机学院
  - 管理学院D101（图书馆右侧）
  - 机器人与智能计算实验室
  - cheng83@mail.sysu.edu.cn