



中国科学院大学
University of Chinese Academy of Sciences

B0911006Y-01

2024-2025学年春季学期

计算机组成原理

Principles of Computer Organization

第4讲 计算机系统概论IV

如何使用CPI、MIPS等指标评估计算机性能

主讲教师：石侃
shikan@ict.ac.cn

2025年3月5日

CPU Time & CPI

1.3

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}}$$

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles Per Instruction}$$

$$\begin{aligned}\text{CPU Time} &= \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time} \\ &= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}\end{aligned}$$

- **Instruction Count (IC) for a program**
 - Determined by program, ISA and compiler
- **Cycles Per Instruction (CPI)**
 - Determined by CPU hardware design, not ISA
 - Different instructions have different CPI
 - *Average CPI* affected by instruction mix



CPI Example

1.3

- Computer A: Cycle Time = 250ps, CPI = 2.0
- Computer B: Cycle Time = 500ps, CPI = 1.2
- Same program and same ISA
- Which is faster, and by how much?

$$\begin{aligned}\text{CPU Time}_A &= \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A \\ &= 1 \times 2.0 \times 250\text{ps} = 1 \times 500\text{ps} \leftarrow \text{A is faster...}\end{aligned}$$

$$\begin{aligned}\text{CPU Time}_B &= \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B \\ &= 1 \times 1.2 \times 500\text{ps} = 1 \times 600\text{ps}\end{aligned}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{1 \times 600\text{ps}}{1 \times 500\text{ps}} = 1.2 \leftarrow \text{...by this much}$$



深入理解CPI

1.3

- If different instruction classes take different numbers of cycles

$$\text{Clock Cycles} = \sum_{i=1}^n (\text{CPI}_i \times \text{Instruction Count}_i)$$

■ Weighted average CPI

$$\text{CPI}_{\text{avg}} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^n \left(\text{CPI}_i \times \underbrace{\frac{\text{Instruction Count}_i}{\text{Instruction Count}}}_{\text{Relative frequency of instruction class}} \right)$$

Relative frequency
of instruction class

CPI例子

1.3

- Alternative compiled code sequences using instructions in classes A, B, C

Inst Class	A	B	C
CPI for each class	1	2	3
Inst Cnt in Sequence 1	2	1	2
Inst Cnt in Sequence 2	4	1	1

- CPI of Sequence 1? CPI of Sequence 2?



- Sequence 1: IC = 5

- Clock Cycles
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$
 $= 10$
- Avg. CPI = $10/5 = 2.0$

- Sequence 2: IC = 6

- Clock Cycles
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$
 $= 9$
- Avg. CPI = $9/6 = 1.5$

- Avg. CPI varies between programs on a given CPU*



如何获得CPI中的Instruction Count?

- CPU execution time and Clock cycles are easy to get
- How to measure CPI or instruction count?
 - Static counting for small programs
 - Dynamic recording tools (simulator or hardware performance counter) for large programs
- 位于CPU核或芯片内部的硬件性能计数器(PC), 通常用于针对CPU或计算机系统的底层性能在线统计、分析和调优

CPI小结

对于某一条特定的指令而言，其CPI是一个确定的值。但是，对于某一个程序或一台机器而言，其CPI是一个平均值，表示该程序或该机器指令集中每条指令执行时平均需要多少时钟周期。

假定 CPI_i 和 C_i 分别为第 i 类指令的CPI和指令条数，则程序的总时钟数为：

$$\text{总时钟数} = \sum_{i=1}^n CPI_i \times C_i \quad \text{所以, CPU时间} = \text{时钟周期} \times \sum_{i=1}^n CPI_i \times C_i$$

假定 CPI_i 、 F_i 是各指令CPI和在程序中的出现频率，则程序综合/平均CPI为：

$$CPI_{avg} = \sum_{i=1}^n CPI_i \times F_i \quad \text{where } F_i = \frac{C_i}{\text{Instruction_Count}}$$

已知CPU时间、时钟频率、总时钟数、指令条数，则程序综合CPI为：

$$CPI_{avg} = (\text{CPU 时间} \times \text{时钟频率}) / \text{指令条数} = \text{总时钟周期数} / \text{指令条数}$$

问题：指令的CPI、机器的CPI、程序的CPI各能反映哪方面的性能？

单靠CPI不能反映CPU性能！为什么？

假设有一种CPU，执行每种指令都需要1个时钟周期，即**单周期CPU**

那么单周期CPU的CPI = 1，但单周期CPU的性能不够好！

MIPS指标的概念

1.3

- **MIPS: Million Instructions Per Second**

$$\begin{aligned} \text{MIPS} &= \frac{\text{Instruction count}}{\text{CPU Time} \times 10^6} \\ &= \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6} \end{aligned}$$

- **Doesn't account for**

- **Differences in ISAs between computers**
- **Differences in complexity between instructions**

MIPS architecture: Microprocessor without Interlocked Piped Stages architecture



MIPS与CPI

1.3

- 例：某计算机的主频为1.2GHz，其指令分为4类，他们在某程序中所占的比例及CPI如下表所示。该机器的MIPS数是多少？

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



MIPS与CPI

1.3

- 例：某计算机的主频为1.2GHz，其指令分为4类，他们在某程序中所占的比例及CPI如下表所示。该机器的MIPS数是多少？

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

- 解： $\text{Avg CPI} = 2 \times 0.5 + 3 \times 0.2 + 4 \times 0.1 + 5 \times 0.2 = 3$
 $\text{MIPS} = 1200\text{MHz} / (3 \times 10^6) = 400$



MIPS数不可靠！

仅在软件上优化，没涉及到任何硬件措施。

Assume we build **an optimized compiler** for a load/store machine. The new compiler can discard 50% of the ALU instructions.

1) What is the CPI ?

2) Assuming a 20 ns clock cycle time (50 MHz clock rate). What is the MIPS rating for optimized code versus unoptimized code? Does the MIPS rating agree with the rating of execution time?

Op	Freq	Cycle
ALU	43%	1
Load	21%	2
Store	12%	2
Branch	24%	2

Optimizing use the new compiler

$$21.5 / (21.5 + 21 + 12 + 24) = 27\%$$

$$21 / (21.5 + 21 + 12 + 24) = 27\%$$

$$12 / (21.5 + 21 + 12 + 24) = 15\%$$

$$24 / (21.5 + 21 + 12 + 24) = 31\%$$

New Freq

27%

27%

15%

31%

CPI : 1.57

MIPS : 31.8

$$50M / 1.57 = 31.8MIPS$$

$$50M / 1.73 = 28.9MIPS$$

1.73

28.9



结果：因为优化后减少了ALU指令（其他指令数没变），所以程序执行时间一定减少了，但优化后的MIPS数反而降低了。

发布了想法

2022-03-01 21:53



包云岗

科研八卦人

① 香山处理器南湖架构已经稳定下来。② 香山第一版（雁栖湖）芯片在寒假期间成功点亮并正确运行Linux/Debian操作系统，成功运行SPEC CPU2006，实测评分超过7分@1GHz，CPU/DDR在更高频率下的性能调优正在进行中。③ 香山处理器南湖架构的设计文档正在编写中，敬请关注！

香山开源处理器：【香山双周报】20220228期

zhuanlan.zhihu.com [文章](#)



XIANGSHAN

发布于 2022-03-01 21:53

34

添加评论

分享

★ 收藏

🚩 举报

<https://www.zhihu.com/people/bao-yun-gang>

<https://www.zhihu.com/pin/1482110970641870848>

性能基准评测程序（Benchmarks）

- 用基准程序来评测计算机的性能

- 基准测试程序是专门用来进行性能评价的一组程序
- 不同用户使用的计算机用不同的基准程序
- 基准程序通过运行实际负载来反映计算机的性能
- 最好的基准程序是用户实际使用的程序或典型的简单程序

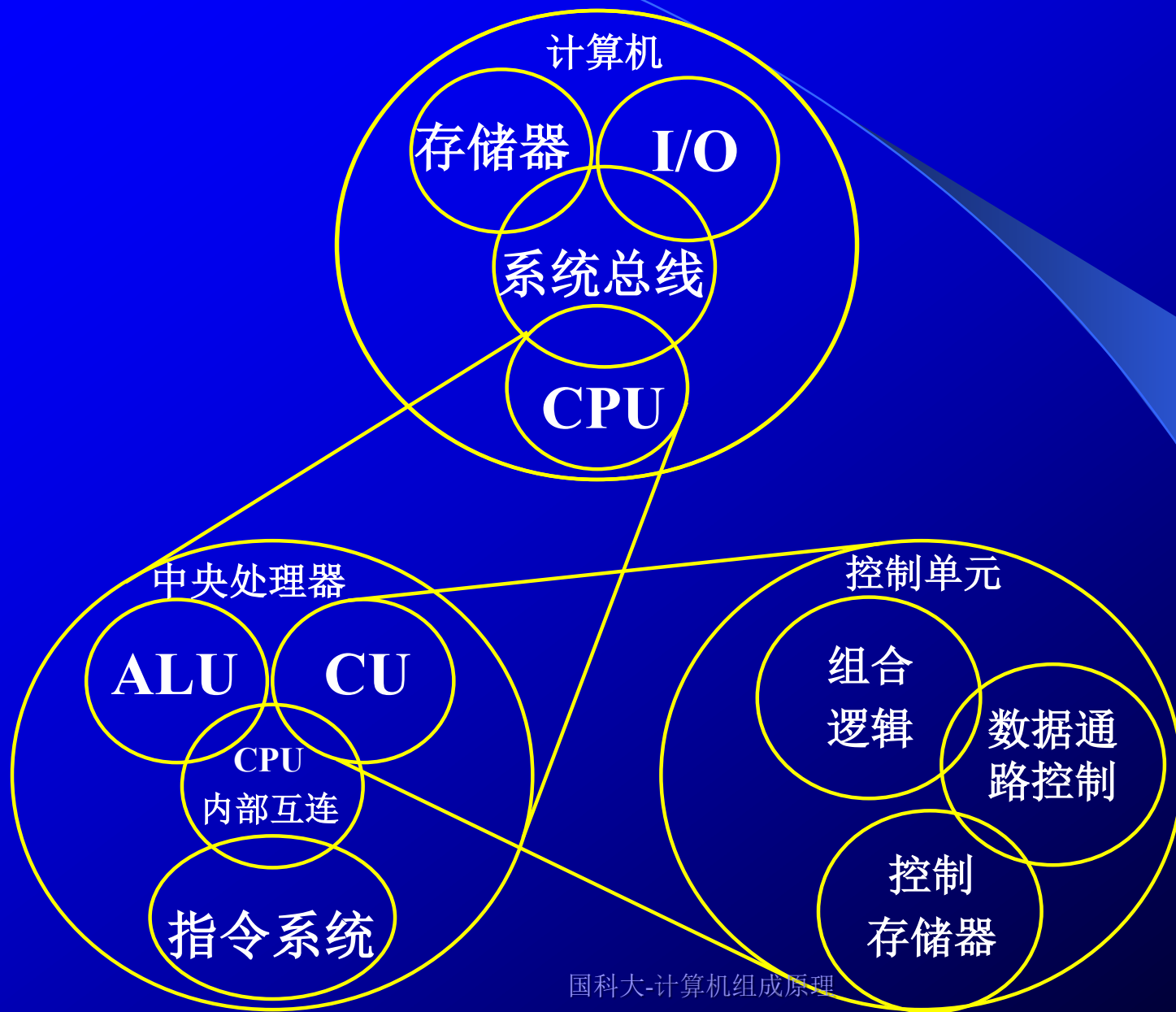
- **Successful Benchmark: SPEC / LINPACK**

- SPEC CPU2017 / SPEC CPU2006, www.spec.org
- LINPACK, www.top500.org

- **基准程序的缺陷**

- 现象：基准程序的性能与某段短代码密切相关时，会被利用以得到不当的性能评测结果
- 手段：硬件系统设计人员或编译器开发者针对这些代码片段进行特殊的优化，使得执行这段代码的速度非常快
 - 例：Intel Pentium处理器运行SPECint时用了公司内部特殊编译器，使其性能极高
 - 例：矩阵乘法程序SPECmatrix300有99%的时间运行在一行语句上，有些厂商用特殊编译器优化该语句，使性能达VAX11/780的729.8倍！

1.4 课程及教材结构：自顶向下



国际教育理念 vs. 计算机组成原理



- ❑ IB（国际文凭组织）课程教育理念
- ❑ 引领孩子“跨学科”探究六大问题
- ❑ 1. Who are we?
 - 我们是谁？
- ❑ 2. Where we are in place and time?
 - 我们所处的时间和空间
- ❑ 3. How we express ourselves?
 - 我们如何表达自己
- ❑ 4. How the world works?
 - 世界如何运作
- ❑ 5. How we organize ourselves?
 - 我们如何管理自己
- ❑ 6. Sharing the planet
 - 人类如何与自然共处

- ❑ 计算机组成原理
- ❑ 引领同学们探究计算机系统的本质
- ❑ 1. Who are ~~we~~ / **Computers**?
 - 第1章：计算机系统概论
- ❑ 2. Where we are in place and time?
 - 第2章：计算机的发展及应用
- ❑ 3. How we express ourselves?
 - 第6/7章：数的表示与指令系统
- ❑ 4. How the world works?
 - 单周期及多周期处理器
 - 第8章：CPU结构功能
- ❑ 5. How we organize ourselves?
 - 第9/10章：CPU控制单元
- ❑ 6. Sharing the planet
 - 第3/4/5章：计算机系统的其他硬件

基于IB国际学校理念，为孩子推荐这25部教育价值极高的电影纪录片！
<https://mp.weixin.qq.com/s/usAoEHP6vCuufOKg1yfEFA>

作业

- **【hw1】**
- 一、课后习题：1.7，1.9，1.10；1.11 (可根据自己之前了解到的计算机系统软硬件相关知识回答1.11题)；9.8。
- 二、结合之前分享的各种阅读材料，自学教材第二章，预习第六章6.1节和6.2节。
- 作业提交时间：3月10日课前，在线提交至SEP。
- 作业提交形式与反馈方式：见下

组成原理理论课课后作业提交流程



- ❑ 教师根据每讲授课内容，在结束后布置相应作业（以教材课后习题为主），并设置作业的提交截止时间
 - 在PPT讲义、班级微信群中发布
 - 在SEP作业区发布（请助教辅助）
- ❑ 作业提交形式
 - 使用计算机编辑或手写作业均可，但需转换为一个单一的PDF文件(单页或多页)
 - 通过扫描或拍照生成的PDF文件，需字体清晰可见，文件大小一般控制在5MB以内
 - **PDF文件命名规则：“学号-hwN.pdf”，其中“-”为英文连字符hyphen，学号中字母大写，hw及pdf为小写，N为作业次数。例如，“2021K8009988710-hw1.pdf”**
 - 建议同学们尽早提交，不要等到截止点之前再提交，这样助教老师也可以提前分批次的批改作业
- ❑ 作业反馈方式
 - 教师或助教对同学们提交的PDF作业进行电子批改（即在PDF文件中进行批注），并于作业提交截止后的一周之内，将批注后的PDF文件反馈给同学们
 - 同学们可在SEP作业区查看本次作业的反馈附件（建议将PDF下载后用软件打开阅读，不要直接使用浏览器查看PDF）
 - 助教仅对PDF作业中的错误进行位置标注，其他问题会在课上进行讲解