



西安电子科技大学  
人工智能学院

# 计算机组成与体系结构

## 第6章 中央处理器 (CPU)

## ➤ CPU性能指标

- ✓ CPU执行时间，简称CPU时间
- ✓ CPI
- ✓ MIPS
- ✓ FLOPS

# 计算机系统性能

## ➤ 存储器容量和速度

- 主存与外存容量以字节为单位度量
- 主存速度用存取周期描述
- 外存速度用数据传输速率描述(受总线速率影响)

## ➤ 外设/总线速度与性能

- 数据传输速率
- 功能与性能指标

## ➤ CPU性能

- 更快的CPU → 减小执行时间
- 多处理器系统 → 提高吞吐量

## ➤ 计算机字长

- 用二进制表示一个数据的位数即为字长
- 字长通常选为字节的整数倍, 如8/16/32/64位
- 字长表示计算机的处理能力和运算精度
- 字长对系统硬件复杂度和价格有影响

# CPU性能指标

## —CPU时间

- CPU工作依据时钟，其速度与时钟密切相关。
- CPU **主频**  $f_{CLK}$ ：CPU使用的时钟的频率，单位为赫兹Hz（每秒时钟脉冲数）。
- CPU**时钟周期**  $T_{CLK}$ ：时钟频率的倒数，计算机系统及CPU工作的时间基准。
- 一般而言，主频高 → 系统/CPU运行速度快

- **CPU时间**：运行一个程序所花费的时间。
- **响应时间**：**CPU时间**与**等待时间**（包括用于磁盘访问、存储器访问、I/O操作、操作系统开销等时间）的总和。
- 假设计算机的时钟周期为  $T_{CLK}$ ，执行某程序时，**CPU**需要使用  $N$  个时钟周期，那么，**CPU**执行该程序所用时间为：

$$\begin{aligned} \text{CPU时间 } T_{CPU} &= \text{CPU时钟周期数 } N \times \text{时钟周期时间 } T_{CLK} \\ &= \frac{\text{CPU时钟周期数 } N}{\text{时钟频率 } f_{CLK}} \end{aligned}$$

- **CPU**执行一段程序所用的时间与该程序所包含的指令数成正比 → 编译器、指令集

**【例6.23】** 计算机A执行某程序用时20s，时钟为1.5GHz，设计者现要构建计算机B，使它以10s的执行时间运行该程序。设计者已确定增加时钟频率是可行的，但会影响CPU设计的其余部分，使得计算机B需要1.2倍于计算机A的时钟数来运行该程序，那么，设计者为B应选择多大的时钟频率？

**【解】**

$$\because T_{CPUA} = \frac{N_A}{f_{CLKA}}$$

$$\therefore N_A = T_{CPUA} \times f_{CLKA} = 20s \times 1.5GHz = 30 \times 10^9 \text{ 周期}$$

$$\because T_{CPUB} = \frac{N_B}{f_{CLKB}} = \frac{1.2 \times N_A}{f_{CLKB}} = 10s$$

$$\therefore f_{CLKB} = \frac{1.2 \times N_A}{T_{CPUB}} = \frac{1.2 \times 30 \times 10^9}{10} = 3.6(GHz)$$

# CPU性能指标

## —CPI



**CPI** (clock cycles per instruction) : 每条指令执行所用的时钟数。平均值。

**IPC** : 每时钟周期执行的指令数。

$N$  = 该程序中的指令数  $I$   $\times$  平均的每条指令所用时钟周期数  $CPI$

$$= \sum_{i=1}^n (CPI_i \times I_i)$$

$$CPI = \frac{1}{I} \sum_{i=1}^n (CPI_i \times I_i) = \sum_{i=1}^n \left( \frac{I_i}{I} \times CPI_i \right)$$

➤ 有三方面的因素使得程序的CPI可能不同于CPU执行的CPI:

- ✓ Cache行为发生变化
- ✓ 指令混合发生变化
- ✓ 分支预测发生变化

➤ 对于某基准程序，用CPI表示的CPU性能为：

$$\text{CPU时间 } T_{CPU} = I \times CPI \times T_{CLK} = \frac{I \times CPI}{f_{CLK}}$$

➤ 影响CPU性能的三个关键因素：

- ✓ **CPI**：依赖存储系统(cache性能)、处理器结构(流水线性能)、指令类型等，随程序、体系结构实现不同而变化
- ✓ **时钟频率**：由硬件确定，受CPU硬件工艺、结构及实现影响
- ✓ **指令数**：依赖指令集的体系结构、编译技术

➤ 有三方面的因素使得程序的CPI可能不同于CPU执行的CPI:

- ✓ Cache行为发生变化
- ✓ 指令混合发生变化
- ✓ 分支预测发生变化

➤ 对于某基准程序，用CPI表示的CPU性能为：

$$\text{CPU时间 } T_{CPU} = I \times CPI \times T_{CLK} = \frac{I \times CPI}{f_{CLK}}$$

➤ 部分软硬件对I、CPI、fCLK的影响：

表6.8 部分软硬件对I、CPI、fCLK的影响

软件或硬件	算法	编程语言	编译器	指令集结构
影响	I、可能CPI	I、CPI	I、CPI	I、CPI、 $f_{CLK}$

【例6.24】假设同一指令集结构有两种实现。计算机A的时钟周期为200ps，执行某程序时CPI=2.0；而计算机B的时钟周期为360ps，执行同一程序时CPI=1.2。执行该程序时，哪个计算机更快？

【解】假设该程序包含I条指令，则

$$T_{CPUA} = I \times CPI_A \times T_{CLKA} = I \times 2.0 \times 200 = 400 \times I(ps)$$

$$T_{CPUB} = I \times CPI_B \times T_{CLKB} = I \times 1.2 \times 360 = 432 \times I(ps)$$

那么，两者的性能比为

$$\frac{P_A}{P_B} = \frac{T_{CPUB}}{T_{CPUA}} = \frac{432 \times I}{400 \times I} = 1.08$$

所以，执行该程序时，计算机A比计算机B快1.08倍。

【例6.25】某个java程序在桌面处理机上运行15秒，一个新的java编译器生成的指令数仅是老编译器的0.6倍，然而CPI增加为1.1倍。利用新的java编译器，我们能够期望这个java程序运行的有多快？

【解】

利用老编译器运行该java程序的时间为  $T_{CPU} = 15s$ ，  
而利用新编译器运行该java程序的时间为：

$$\begin{aligned} T_{CPU} &= I_{\text{新}} \times CPI_{\text{新}} \times T_{CLK} \\ &= 0.6 \times I_{\text{老}} \times 1.1 \times CPI_{\text{老}} \times T_{CLK} \\ &= 0.6 \times 1.1 \times 15 = 9.9(s) \end{aligned}$$

## 习题6.21

【二级Cache系统性能】一个拥有L1和L2 Cache的系统在没有Cache缺失时的CPI为1.2，且平均每条指令访问1.1次主存。考虑有Cache缺失时，有效的CPI是多少？如果L1和L2 Cache被做成单一的Cache模块，那么总的有效命中率和失效惩罚是多少？

级	局部命中率	缺失惩罚
L1	95%	8 cycles
L2	80%	60 cycles

解：  $C_{\text{eff}} = C_{\text{fast}} + (1 - h_1)[C_{\text{medium}} + (1 - h_2)C_{\text{slow}}]$

因为  $C_{\text{fast}}$  被包括在1.2的CPI中，所以有

$$\text{CPI} = 1.2 + 1.1 \times (1 - 0.95)[8 + (1 - 0.8) \times 60] = 2.3$$

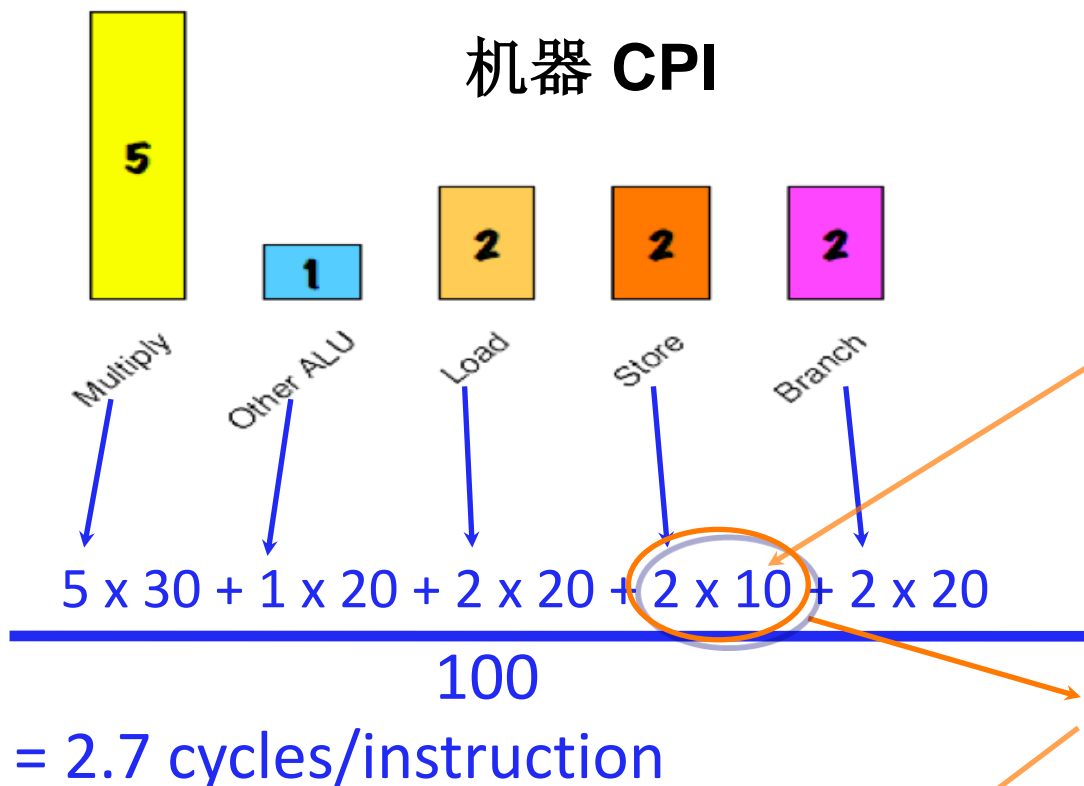
总命中率 =  $(95\% + 5\% \times 80\%) = (1 - 0.05 \times 0.2) \times 100\% = 99\%$

总缺失惩罚 = 60 cycles

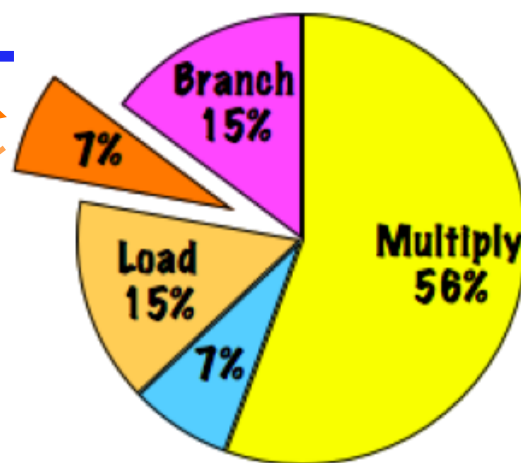
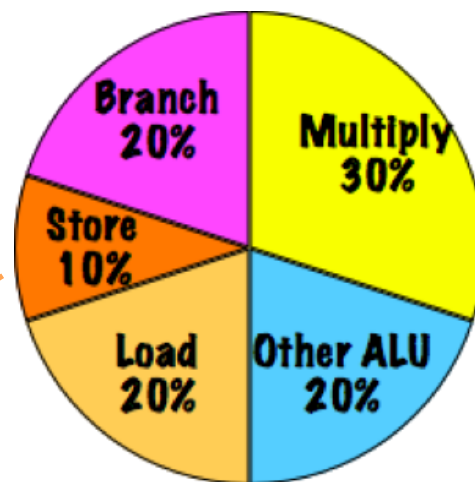
## CPI作为一种指导设计的分析工具

## 程序指令组合情况

## 机器 CPI



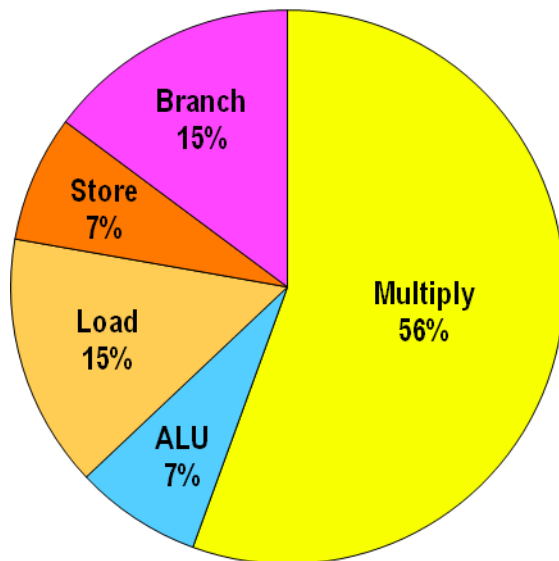
$$\frac{0.2}{2.7} = 7.4\%$$



程序时间花费在哪里？

## Amdahl定律(收益递减律)

程序时间  
花费在哪里?



如果增强“E”使乘法无限地快，而其他指令不改变，最大加速比S是多少？

$$S_{\max} = \frac{1}{\text{un-enhanced \%} / 100\%} = \frac{1}{44\% / 100\%} = 2.27$$

Amdahl定律的启示是什么？

必须以平衡的方式增强计算机！



A black and white photograph of Gene Amdahl, an older man with white hair, wearing a suit and tie. He is sitting on a large, dark computer console. Behind him are several tall, dark computer cabinets filled with electronic components. The floor is made of large, light-colored square tiles. The text "GENE AMDAHL:" is overlaid in large, white, bold letters on the left side of the image.

# GENE AMDAHL:

## COMPUTER PIONEER

ALEXIS DANIELS

Invented the “one ISA, many implementations” business model.

# CPU性能指标

## —MIPS

- **MIPS** (million instructions per second) :  
CPU每秒钟执行的百万指令数。

$$MIPS = \frac{\text{指令数 } I}{\text{执行时间 } T \times 10^6} = \frac{f_{CLK}}{CPI \times 10^6}$$

- **MIPS参数的局限：**

- 不能对指令集不同的计算机使用**MIPS**进行比较：**MIPS**只说明了指令执行速率，而没有考虑指令的能力
- 计算机对所有程序没有单一的**MIPS**值：对于同一个计算机上的不同程序，**MIPS**是变化的
- **MIPS**会与性能反向变化

## 【例6.26】

某计算机具有3类指令，测试每类指令得到的CPI结果如下：

时钟频率 为4GHz	每类指令的CPI		
	A	B	C
CPI	1	2	3

用两种不同的编译器对某个程序进行编译得到的各类指令的数量如下：

时钟频率 为4GHz	每类指令的指令数 ( $\times 10^9$ )		
	A	B	C
编译器1	5	1	1
编译器2	10	1	1

假设计算机的时钟频率为4GHz，问：

- ① 根据MIPS，由哪个编译器生成的代码序列执行得更快？
- ② 根据执行时间，由哪个编译器生成的代码序列执行得更快？

【解】

$$\left\{ \begin{array}{l} CPU\text{时钟周期数}_1 = (5 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 10 \times 10^9 \\ \text{执行时间}_1 = \frac{10 \times 10^9}{4 \times 10^9} = 2.5(s) \\ MIPS_1 = \frac{(5 + 1 + 1) \times 10^9}{2.5 \times 10^6} = 2800 \end{array} \right.$$

编译器1

$$\left\{ \begin{array}{l} CPU\text{时钟周期数}_2 = (10 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 15 \times 10^9 \\ \text{执行时间}_2 = \frac{15 \times 10^9}{4 \times 10^9} = 3.75(s) \\ MIPS_2 = \frac{(10 + 1 + 1) \times 10^9}{3.75 \times 10^6} = 3200 \end{array} \right.$$

编译器2

## 【例6.27】

对某程序进行性能测试，  
结果如右图所示。

① 哪个计算机具有较高的  
**MIPS** 值？

② 哪个计算机更快？

测量	计算机A	计算机B
指令数	$10 \times 10^9$	$8 \times 10^9$
时钟频率	4GHz	4GHz
<i>CPI</i>	1.0	1.1

【解】 
$$T_A = \frac{I_A \times CPI_A}{f_{CLK}} = \frac{10 \times 10^9 \times 1.0}{4 \times 10^9} = 2.5(s)$$

$$MIPS_A = \frac{I_A}{T_A \times 10^6} = \frac{10 \times 10^9}{2.5 \times 10^6} = 4000$$

$$T_B = \frac{I_B \times CPI_B}{f_{CLK}} = \frac{8 \times 10^9 \times 1.1}{4 \times 10^9} = 2.2(s)$$

$$MIPS_B = \frac{I_B}{T_B \times 10^6} = \frac{8 \times 10^9}{2.2 \times 10^6} = 3636$$

对于给定的  
程序，计算  
机**A**具有较  
高的**MIPS**  
值；计算机  
**B**执行速度  
更快。

# CPU性能指标

## —FLOPS

- **FLOPS** (floating point operations per second) : CPU每秒完成的浮点运算次数。

$$FLOPS = \frac{\text{浮点运算次数 } M}{\text{执行时间 } T}$$

- **FLOPS** 可以用于不同计算机之间的比较。
- **FLOPS** 参数有如下几种度量单位：
- ✓ megaflop (MFLOPS, MF,  $10^6$  FLOPS): million ...
  - ✓ gigaflop (GFLOPS, GF,  $10^9$  FLOPS): giga ...
  - ✓ teraflop (TFLOPS, TF,  $10^{12}$  FLOPS): trillion ...
  - ✓ petaflop (PFLOPS, PF,  $10^{15}$  FLOPS): thousand trillion ...
  - ✓ exaflop (EFLOPS, EF,  $10^{18}$  FLOPS): quintillion ...
- 国际超级计算机**500**强排名 (**TOP500**)
- <http://www.top500.org>



# 全球超级计算机2019年6月和11月榜单TOP10

排名	系 统	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	<b>Summit</b> - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
2	<b>Sierra</b> - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
3	<b>Sunway TaihuLight</b> - Sunway MPP, Sunway SW1601 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
4	<b>Tianhe-2A</b> - TH-1V , J-EP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH E-processor 2, Matrix-2000 , NVIDIA National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	<b>Frontera</b> - Dell C6420, Xeon Platinum 8280 18C 2.7GHz, Mellanox InfiniBand HDR , Dell EMC Texas Advanced Computing Center, Univ. of Texas United States	448,448	23,500.4	58,745.9	
6	<b>Piz Daint</b> - Cray XC40, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray MPI Swiss National Supercomputing Centre (SCS) Switzerland	387,872	21,230.0	27,154.3	2,384
7	<b>Trinity</b> - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray MPI DOE/NNSA/LANL/SNL United States	979,072	20,118.0	41,461.2	7,578
8	<b>AI Bridging Cloud Infrastructure (ABCI)</b> - PRIMERGY CX2570 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	32,576.6	1,649
9	<b>SuperMUC-NG</b> - ThinkSystem SD650, Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path , Lenovo Leibniz Rechenzentrum Germany	305,856	19,476.6	26,873.9	
10	<b>Lassen</b> - IBM Power System AC922, IBM POWER9 22C 3.1GHz, Dual-rail Mellanox EDR Infiniband, NVIDIA Tesla V100 , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	288,288	18,200.0	23,047.2	



# 提高CPU速度的策略

## 6.5.2 提高CPU速度的策略

### ➤ 衡量CPU性能的指标

- ✓ 处理能力：指令系统
- ✓ 处理速度
- ✓ 内核数量

### ➤ 提高CPU速度的技术

- ✓ 具体实现的改进
- ✓ 体系结构的改进

(1) 采用更先进的硅加工制造技术

(2) 缩短指令执行路径长度

减少执行指令的时钟周期数：

- ◆ 优化微操作、微程序
- ◆ 双总线、三总线
- ◆ 增加特定硬件，实现并行

(3) 简化组织结构来缩短时钟周期

- ◆ RISC结构

(4) 采用并行处理技术

- ◆ 指令级并行
- ◆ 处理器级并行