Lecture 3: Performance Evaluation

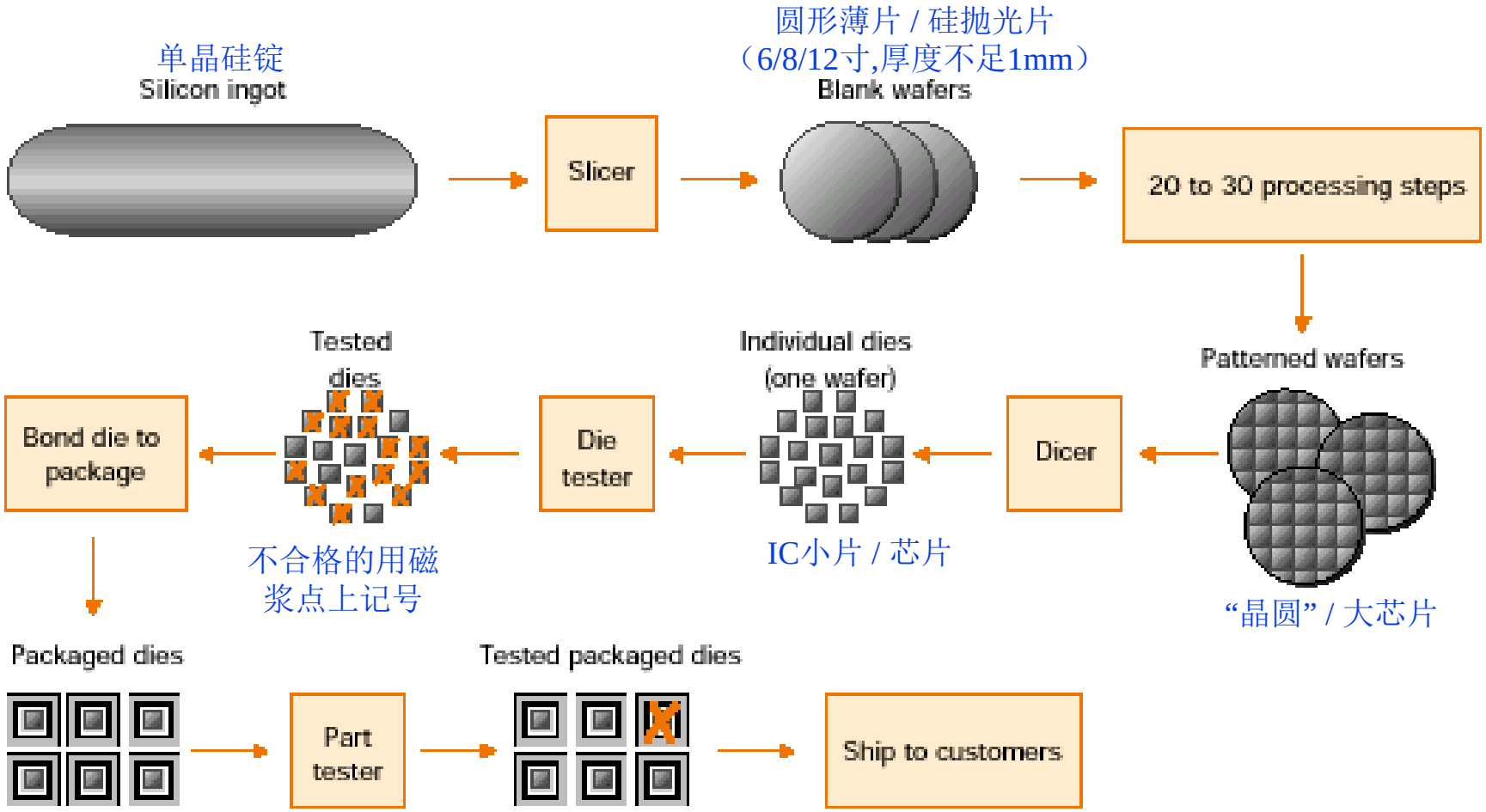
第3讲：计算机性能评价

第三讲 计算机性能评价

- 制造成本（**manufacturing cost**）
- 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**）
  - **throughput**（吞吐量）
    - 带宽（**bandwidth**）
- 计算机性能测量
- 指令执行速度（**MIPS**、**MFLOPS**）
- 基准程序（ **Benchmark**）

回顾: Integrated Circuits Costs --- manufacturing process

在考察性能前，先考察成本！



封装：将芯片固定在塑胶或陶瓷基座上，把芯片上蚀刻出来的引线与基座底部伸出的引脚连接，盖上盖板并封焊成芯片

约需400多道工序！

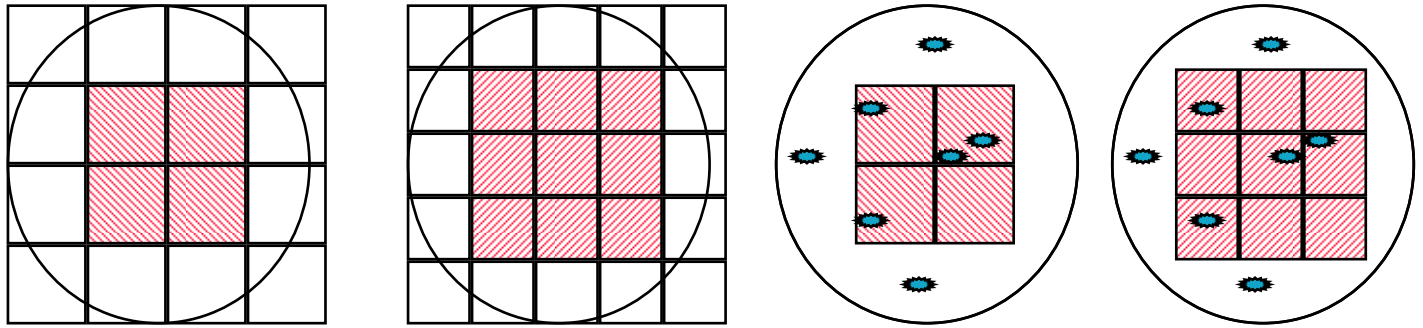
Integrated Circuits Costs 公式

$$\text{Die cost} = \frac{Cost\_per\_wafter}{Die\_per\_wafer \times Yield}$$

$$\text{Dies per wafer} = \frac{wafer\_area}{Die\_area}$$

芯片成本与以下因素有关：

- 圆晶价格
- 圆晶所含小片数
- 小片合格率



小片合格率

$$\text{Die Yield} = \frac{1}{(1+(Defect\_per\_area \times Die\_area))^2}$$

由此看出：每个圆晶片上的小片数、集成电路成本都与芯片面积有关！

计算机性能的基本评价指标

- 计算机有两种不同的性能
  - Time to do the task
    - 响应时间（**response time**）
    - 执行时间（**execution time**）
    - 等待时间或时延（**latency**）
  - Tasks per day, hour, sec, ns. ..
    - 吞吐率（**throughput**）
    - 带宽（**bandwidth**）

不同应用场合用户关心的性能不同：

要求吞吐率高的场合，例如：

多媒体应用（音/视频播放要流畅）

要求响应时间短的场合：例如：

事务处理系统（存/取款速度要快）

要求吞吐率高且响应时间短的场合：

ATM、文件服务器、**Web**服务器等

- 基本的性能评价标准是：**CPU**的执行时间

" X is n times faster than Y" means

$$\frac{\text{ExTime(Y)}}{\text{ExTime(X)}} = \frac{\text{Performance(X)}}{\text{Performance(Y)}}$$

相对性能用执行时间的倒数来表示！

计算机性能的测量

- 比较计算机的性能时，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就是性能最好的
  - 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（从**hello**程序执行过程可知）
  - 通常把用户感觉到的响应时间分成以下两个时间：
    - **CPU**时间：指**CPU**真正花在程序执行上的时间。又包括两部分：
      - 用户**CPU**时间：用来运行用户代码的时间
      - 系统**CPU**时间：为了执行用户程序而需要运行操作系统程序的时间
    - 其他时间：指等待**I/O**操作完成或**CPU**花在其他用户程序的时间
  - 系统性能和**CPU**性能不等价，有一定的区别
    - 系统性能**(System performance)**：系统响应时间，与**CPU**外的其他部分也都有关系
    - **CPU**性能**(CPU performance)**：用户**CPU**时间
  - 本章主要讨论**CPU**性能，即：**CPU**真正用在用户程序执行上的时间

    问题：用户**CPU**时间与系统响应时间哪个更长？

CPU执行时间的计算

CPI：Cycles Per Instruction

CPU 执行时间 = CPU时钟周期数 / 程序 X 时钟周期

= CPU时钟周期数 / 程序 ÷ 时钟频率

= 指令条数 / 程序 X CPI X 时钟周期

CPU时钟周期数 / 程序 = 指令条数 / 程序 X CPI

CPI = CPU时钟周期数 / 程序 ÷指令条数 / 程序

CPI 用来衡量以下各方面的综合结果

- Instruction Set Architecture（ISA）
- Implementation of that architecture
- Program（Compiler、Algorithm）

Aspects of CPU Performance

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	instr. count	CPI	clock rate
Program			
Compiler			
Instr. Set Arch.			
Organization			
Technology			

思考：三个因素与哪些方面有关？

Aspects of CPU Performance

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	instr. count	CPI	clock rate
Program	X	X	
Compiler	X	(X)	
Instr. Set Arch.	X	X	
Organization		X	X
Technology			X

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

Architecture = Instruction Set Arch. + Organization

Instruction Set Design

- Machine Language
  - Compiler View
  - "Computer Architecture"
  - "Instruction Set Processor"

"Building Architect"

Computer Hardware Design

- Machine Implementation
  - Logic Designer's View
  - "Processor Architecture"
  - " Computer Organization"

"Construction Engineer"

例如，是否提供“乘法指令”是ISA设计要考虑的问题； 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）是组成（Organization）考虑的问题； 如何布线、用什么材料、工艺设计等是计算机实现技术（Technology）考虑的问题。

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的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：

$$\text{总时钟数} = \sum_{i=1}^{n} CPI_i \times C_i$$

所以，

$$\text{CPU时间} = \text{时钟周期} \times \sum_{i=1}^{n} CPI_i \times C_i$$

已知CPU时间、时钟频率、总时钟数、指令条数，则程序综合CPI为:

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

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的性能！为什么？如：单周期处理器CPI=1，但性能差！

Example1

程序P在机器A上运行需10 s， 机器A的时钟频率为400MHz。现在要设计一台机器B，希望该程序在B上运行只需6 s.

机器B时钟频率的提高导致了其CPI的增加，使得程序P在机器B上时钟周期数是在机器A上的1.2倍。机器B的时钟频率达到A的多少倍才能使程序P在B上执行速度是A上的10/6=1.67倍？

Answer:

CPU时间A = 时钟周期数A / 时钟频率A

时钟周期数A = 10 sec x 400MHz = 4000M个

时钟频率B = 时钟周期数B / CPU时间B

= 1.2 x 4000M / 6 sec = 800 MHz

机器B的频率是A的两倍，但机器B的速度并不是A的两倍！

Marketing Metrics （产品宣称指标）

MIPS = Instruction Count / Time x10^6

= Clock Rate / CPI x 10^6

Million Instructions Per Second

因为每条指令执行时间不同，所以**MIPS**总是一个平均值。

- 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**数表示性能有没有局限**?**

所以**MIPS**数不能说明性能的好坏（用下页中的例子来说明）

MFLOPS = FP Operations / Time x10^6

Million Floating-point Operations Per Second

- 与机器相关性大
- 并不是程序中花时间的部分

用**MFLOPS**数表示性能也有局限！

Example: MIPS数不可靠！

Assume we build an optimizing compiler for the load/store machine. The compiler discards 50% of the ALU instructions.

1) What is the CPI ?

仅仅在软件上进行优化，没有涉及到任何硬件措施。

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 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	50M/1.57=31.8MIPS	1.73
MIPS	31.8	50M/1.73=28.9MIPS	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。

选择性能评价程序（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
- 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1：**Intel Pentium**处理器运行**SPECint**时用了公司内部使用的特殊编译器，使其性能极高
    - 例2：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！

Successful Benchmark: SPEC

- **1988年，5家公司（ Sun, MIPS, HP, Apollo, DEC ）联合提出了SPEC (Systems Performance Evaluation Committee)**
- **SPEC给出了一组标准的测试程序、标准输入和测试报告。它们是一些实际的程序，包括 OS calls、 I/O等。**
- **版本 89：10 programs = 4 for integer + 6 for FP, 用每个程序的执行时间求出一个综合性能指标**
- **版本92：SPECInt92 (6 integer programs) and SPECfp92 (14 floating point programs)**
  - **整数和浮点数单独提供衡量指标：SPECInt92和SPECfp92**
  - **增加 SPECbase: 禁止使用任何与程序有关的编译优化开关**
- **版本95： 8 int + 10fp**
- **较新版本： include SPEC HPC96, SPEC JVM98, SPEC WEB99, SPEC OMP2001. SPEC CPU2000, SPEC CPU2006,See http://www.spec.org**
  - **“benchmarks useful for 3 years”**
  - **Base machine is changed from VAX-11/780 to Sun SPARC 10/40**

如何给出综合评价结果?

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

先看一个例子：

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

Program 2: 1000 sec on A，100 sec on B

What are your conclusions?

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

这个结论无法比较**A**和**B**的好坏

必须用一个综合的值来表示！

Total exec. time是一个综合度量值，可以据此得出结论：

B is 1001/110=9.1 times faster than A

实际上，可考虑每个程序在作业中的使用频度，即加权平均

综合性能评价的方法

- 可用以下两种平均值来评价：
  - **Arithmetic mean(算术平均)：求和后除n**
  - **Geometric mean(几何平均)：求积后开根号n**
- 根据算术平均执行时间能得到总平均执行时间
- 根据几何平均执行时间不能得到程序总的执行时间
- 执行时间的规格化（测试机器相对于参考机器的性能）：
  - **time on reference machine ÷ time on measured machine**
- 平均规格化执行时间不能用算术平均计算，而应该用几何平均
  - **program A going from 2s to 1s as important as program B going from 2000s to 1000s.**

    （算术平均值不能反映这一点！）

综上所述，算术平均和几何平均各有长处，可灵活使用！

第三讲小结

- 性能的定义：一般用程序的响应时间或系统的吞吐率表示机器或系统整体性能。
- 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 x 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 x 每条指令的平均时钟周期数CPI
- MIPS数在有些情况下不能说明问题，不具有可比性！
- 性能评价程序的选择：
  - 采用一组基准测试程序进行综合（算术（加权）平均/几何平均）评测。
  - 有些制造商会针对评测程序中频繁出现的语句采用专门的编译器，使评测程序的运行效率大幅提高。因此有时基准评测程序也不能说明问题。
- 对于某种特定的指令集体系结构，提高计算机性能的主要途径有：
  - 提高时钟频率（第七章 流水线）
  - 优化处理器中数据通路的结构以降低CPI（单周期/多周期/流水线）
  - 采用编译优化措施来减少指令条数或降低指令复杂度（第五章 指令系统 ）