1.3　程序表象之下

一个典型的应用程序，如字处理程序或大型数据库系统，可以由数百万行代码构成，并依靠软件库来实现异常复杂的功能。众所周知，计算机中的硬件只能执行极为简单的低级指令。从复杂的应用程序到原始的指令涉及若干软件层次来将高层次操作解释或翻译成简单的计算机指令，这可以作为伟大的抽象思想的一个例子。

> 在巴黎，我对当地人讲法语，他们只是瞪着眼看着我；我从来没能让这些白痴理解他们自己的语言。
> *马克·吐温，*
> *异国奇遇，1869*

图 1-3 给出了这些软件的层次结构，外层是应用软件，中心是硬件，**系统软件**（systems software）位于两者之间。

> **系统软件**：提供常用服务的软件，包括操作系统、编译器、加载程序和汇编器等

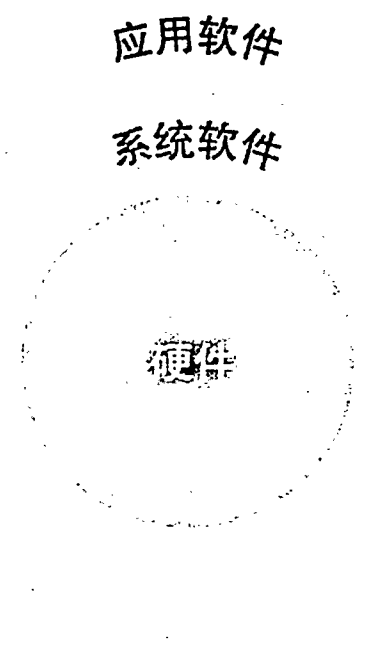

图 1-3　简化的硬件和软件层次图，将硬件作为同心圆的中心，应用软件作为最外层。在复杂的应用中通常存在多层应用软件层。例如，一个数据库系统可运行于系统软件之上，而驻留在该系统软件上的某应用又反过来运行在该数据库之上

系统软件有很多种，其中有两种对于现代计算机系统来说是必需的：操作系统和编译器。**操作系统**（operating system）是用户程序和硬件之间的接口，为用户提供各种服务和监控功能。操作系统最为重要的作用是：

> **操作系统**：为了使程序更好地在计算机上运行而管理计算机资源的监控程序。

- 处理基本的输入和输出操作。
- 分配外存和内存。
- 为多个应用程序提供共享计算机资源的服务。

当前我们使用的操作系统主要有 Linux、iOS 和 Windows。

编译器（compiler）完成另外一项重要功能：把高级语言（如 C、C++、Java 或 Visual Basic 等）编写的程序翻译成硬件能执行的指令。这个翻译过程是相当复杂的，这里仅作简要介绍，第 2 章将作深入介绍。

> **编译器**：将高级语言翻译为计算机所能识别的机器语言的程序。

从高级语言到硬件语言

谈到电子硬件，首先需要谈到电信号的发送。对于计算机来说，最简单的信号是通和断。因此，计算机只用 2 个字母来表示。正如英语 26 个字母写多少不受限制一样，计算机的 2 个字母写多少也不受限制。代表 2 个字母的符号是 0 和 1，我们通常认为计算机语言就是二进制数。每个字母就是二进制数字中的一个**二进制位**（binary digit）或一位（bit）。

> **二进制位**：也称为位。基数为 2 的数字中的 0 或 1，它是信息的基本组成元素。

计算机服从于我们的命令，即计算机术语中的**指令**（instruction）。指令是能被计算机识别并执行的位串，可以将其视为数字。例如，位串

```
1001010100101110
```

告诉计算机将两个数相加。第 2 章将解释为什么数字既表示指令又表示数据。我们不希望在此处涉及第 2 章的具体内容，但是使用数字既表示指令又表示数据是计算机的基础。

指令：计算机硬件能够理解并遵从的命令。

第一代程序员是直接使用二进制数与计算机通信的，这是一项非常乏味的工作。所以他们很快发明了助记符，以符合人类的思维方式。最初助记符是手工翻译成二进制的，其过程显然过于烦琐。随后设计人员开发了一种称为**汇编器**（assembler）的软件，可以将助记符形式的指令自动翻译成对应的二进制。例如，程序员写下

```
add A, B
```

汇编程序会将该符号翻译成

```
1001010100101110
```

该指令告诉计算机将 A 和 B 两个数相加。这种符号语言的名称今天还在用，即**汇编语言**（assembly language）。而机器可以理解的二进制语言是**机器语言**（machine language）。

汇编器：将指令由助记符形式翻译成二进制形式的程序。

汇编语言：以助记符形式表示的机器指令。

虽然这是一个巨大的进步，但汇编语言仍然与科学家用来模拟液体流动或会计师用来结算账目所使用的符号相去甚远。汇编语言需要程序员写出计算机执行的每条指令，要求程序员像计算机一样思考。

机器语言：以二进制形式表示的机器指令。

认识到可以编写一个程序来将更强大的高级语言翻译成计算机指令是计算机早期的一个重大突破。**高级编程语言**及其编译器大大地提高了软件的生产率。图 1-4 给出了这些程序和编程语言之间的关系，这是抽象思想之伟大的另外一个例子。

高级编程语言：如 C、C++、Java、Visual Basic 等可移植的语言，由一些单词和代数符号组成，可以由编译器转换为汇编语言。

编译器使得程序员可以写出高级语言表达式。

```
A + B
```

编译器将其编译为如下的汇编语言语句：

```
add A, B
```

然后，汇编器将此语句翻译为二进制指令，告诉计算机将两个数 A 和 B 相加。

使用高级编程语言有以下几个好处。第一，可以使程序员用更自然的语言来思考，用英文和代数符号来表示，形成的程序看起来更像文字而不是密码表（见图 1-4）。而且，它们可按用途进行设计。例如，Fortran 是为科学计算设计的，Cobol 是为商业数据操作设计的，Lisp 是为符号操作设计的，等等。还有一些特定领域的语言，只为少数专业人群设计，如流体仿真的研究人员等。

第二，高级语言提高了程序员的生产率。如果使用较少行数的编程语言即可表示出设计用意，则可加速程序的开发，这是软件开发方面少有的共识之一。简明性是高级语言相对汇

编语言最为明显的优势。

第三，采用高级语言编写程序提高了程序相对于计算机的独立性，因为编译器和汇编程序能够把高级语言程序翻译成任何计算机的二进制指令。高级编程语言的这些好处，使其直到今天仍应用广泛。

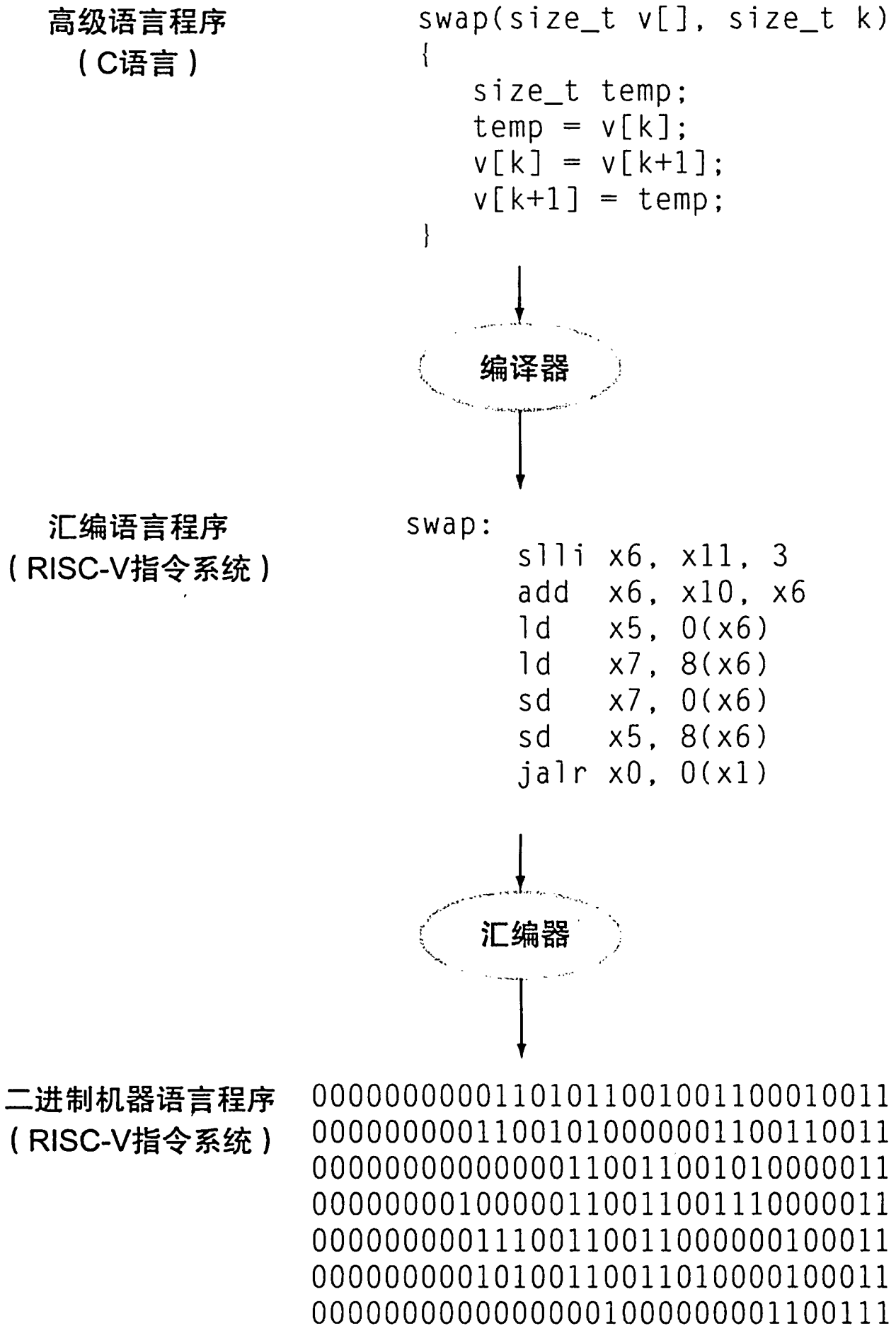

图 1-4　C 程序编译为汇编语言程序，再汇编为二进制机器语言程序。尽管将高级语言翻译成二进制的机器语言仅需要两步，但一些编译器将“中间人”略去并直接产生二进制的机器语言。这些语言和本图中列举的程序将在第 2 章详细介绍

1.4　箱盖后的硬件

我们已经在上节通过程序揭示了计算机软件，在本节中我们将打开机箱盖学习其中的硬件。任何一台计算机的基础硬件都要完成相同的基本功能：输入数据、输出数据、处理数据和存储数据。本书的主题就是描述这些功能是怎样完成的，随后各章将分别讨论这 4 项任务。

本书在遇到重要知识点时，都会用“重点”标题加以强调，希望读者对其重点记忆。全书大致有 10 多个重要知识点，这里是第一个，即计算机是由输入、输出、处理和存储数据任务的 5 个部件构成的。

计算机的两个关键部件是**输入设备**（input device）和**输出设备**（output device），例如麦

克风是输入设备，而扬声器是输出设备。输入为计算机提供数据，输出将计算结果送给用户。像无线网络等设备既是输入设备又是输出设备。

输入设备：为计算机提供信息的装置，如键盘。

输出设备：将计算结果输出给用户（如显示器）或其他计算机的装置。

第5章和第6章将详细介绍I/O设备，这里由外部I/O设备开始先对计算机硬件做一些基本的介绍。

重点 组成计算机的五个经典部件是输入、输出、存储器、数据通路（在计算机中也称运算器）和控制器，其中后两个部件通常合称为处理器。图1-5展示了一台计算机的标准组成部分。该组成与硬件技术无关，你总能够把任何现在或过去的计算机中的任何组件归于这五类组件之一。为了加深读者对这一重点的印象，我们将在每章开始都给出此图，并突出显示该章关注的部分。

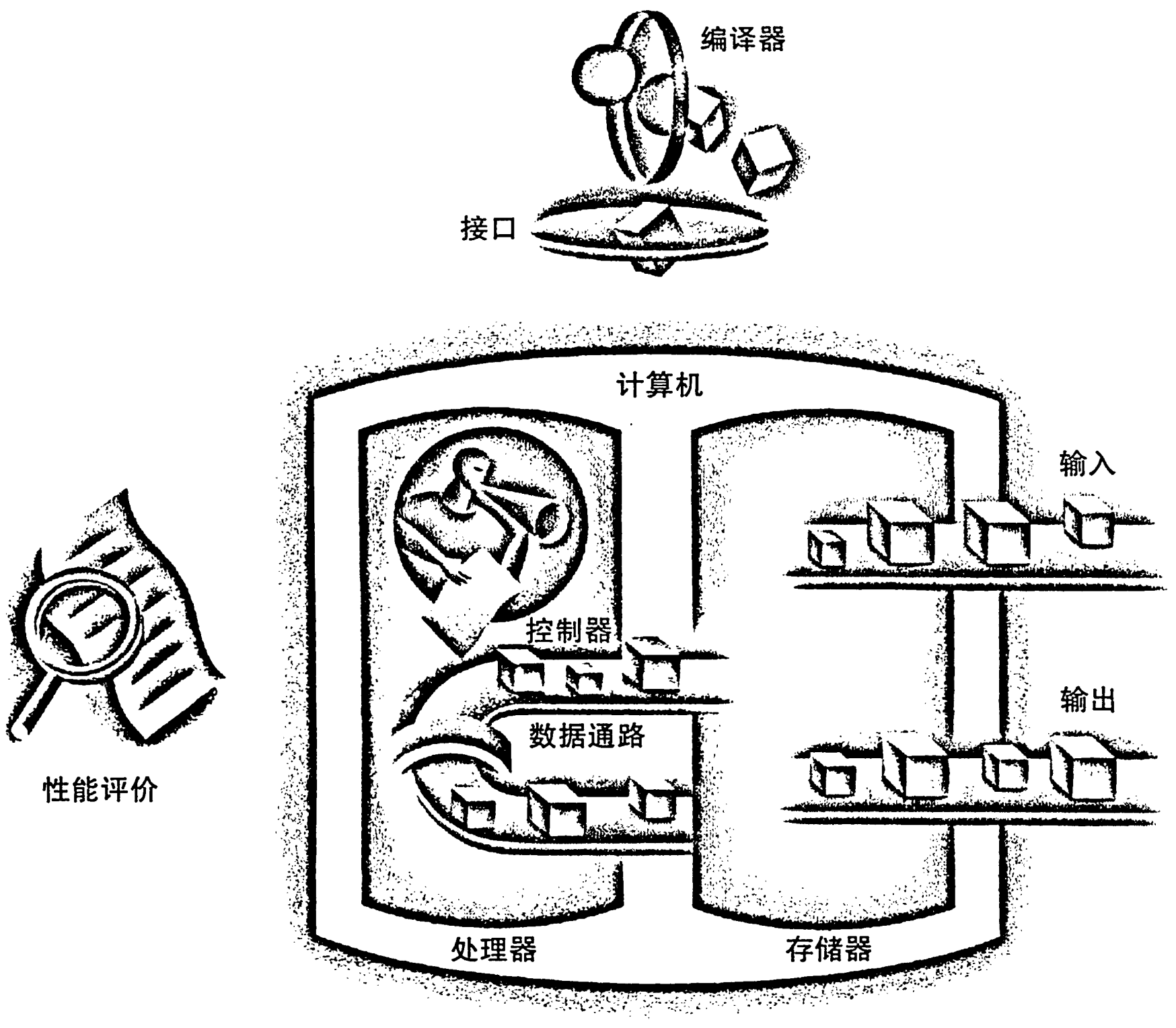

图1-5 组成计算机的五个经典部件。处理器从存储器中得到指令和数据，输入部件将数据写入存储器，输出部件从存储器中读出数据，控制器向数据通路、存储器、输入和输出部件发出命令信号

1.4.1 显示器

最吸引人的I/O设备应该是图形显示器了。大多数个人移动设备都用**液晶显示**（Liquid Crystal Display，LCD）来获得轻巧、低功耗的显示效果。LCD并非光源，而是控制光的传输。典型的LCD内含棒状液态分子团形成的转动螺旋线，用来弯曲来自显示器后方的光线或者少量的反射光线。当电流通过时，液态分子棒不再弯曲，

通过计算机显示器，我将飞机降落在航空母舰的甲板上，观察到一个原子打到势阱中，乘着火箭以接近光的速度飞翔，同时我了解到计算机最深层的工作原理。

Ivan Sutherland，计算机图形学之父，科学美国人，1984

也不再使光线弯曲。由于两层相互垂直的偏光板之间充满液晶材料，如果它不弯曲则光线不能通过。（在不施加任何电压的情况下，液晶处于初始状态，并将入射光的方向扭转 90°，让背光源的入射光能够通过整个结构，在显示屏上呈现白色；而当施加电压时，光线不再弯曲，显示屏呈现黑色。）今天，大多数 LCD 显示器采用**动态矩阵**（active matrix）**显示**技术，其每个像素都由一个晶体管精确地控制电流，使图像更清晰。在彩色动态矩阵 LCD 中，还有一个红 – 绿 – 蓝屏决定三种颜色分量的强度，每个点需要三个晶体管开关。

液晶显示：一种显示技术，用液体聚合物薄层的带电或者不带电来使能或阻止光线的传输。

动态矩阵显示：一种液晶显示技术，使用晶体管控制单个像素上光线的传输。

图像由**像素**矩阵组成，可以表示成二进制位的矩阵，称为位图（bit map）。针对不同的屏幕尺寸及分辨率，典型的屏幕中显示矩阵的大小可以从 1024 × 768 到 2048 × 1536。彩色显示器使用 8 位来表示每个三原色（红、绿和蓝），每个像素用 24 位表示，可以显示百万种不同的颜色。

像素：图像元素的最小单元。屏幕由数百万到数千万像素组成的矩阵构成。

计算机硬件采用*光栅刷新缓冲区*（又称为*帧缓冲区*）来保存位图以支持图像。要显示的图像保存在帧缓冲区中，每个像素的二进制值以刷新频率读出到显示设备。图 1-6 显示了用 4 位表示一个像素的简化设计的帧缓冲区。

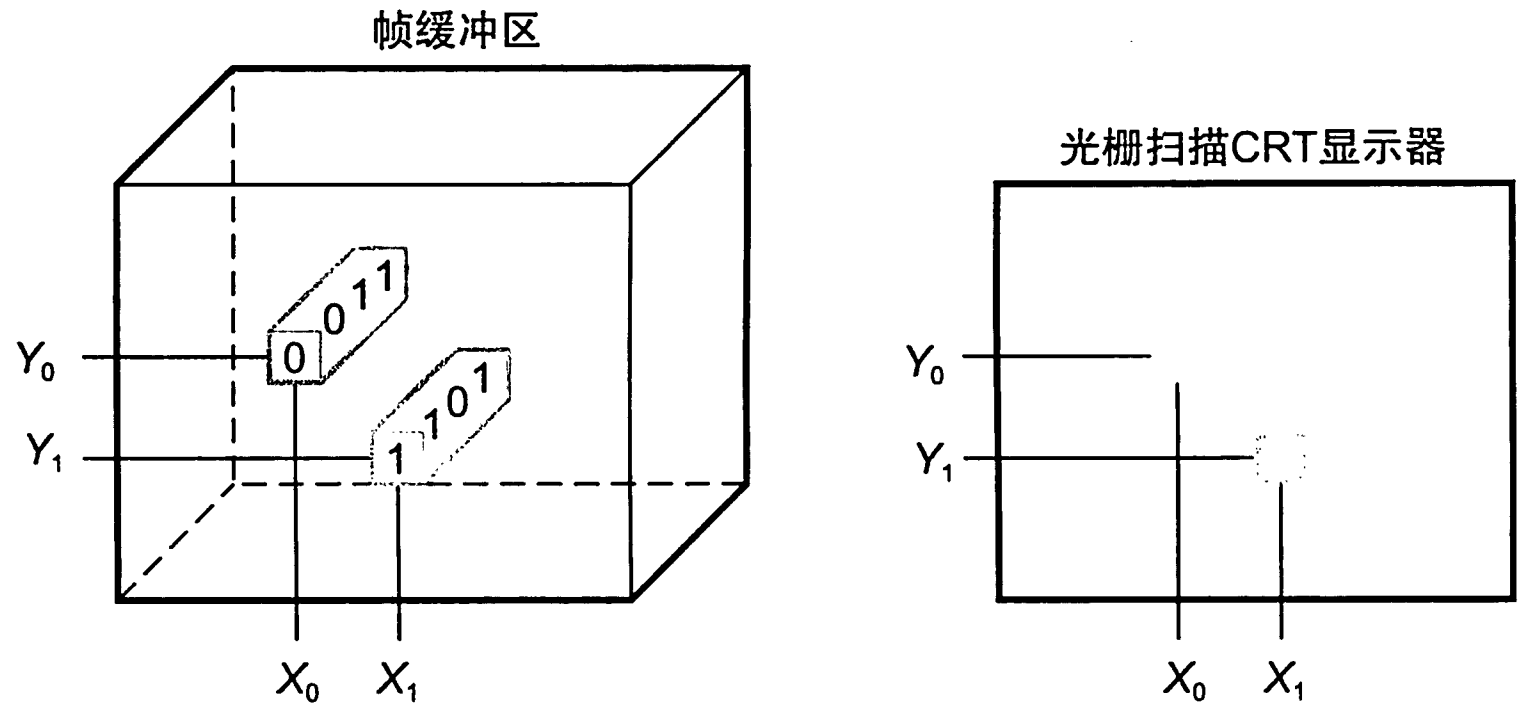

图 1-6　左边帧缓冲区中的每个坐标决定了右边光栅扫描 CRT 显示器中相应坐标的灰度。像素（X_0，Y_0）的灰度值是 0011，小于像素（X_1，Y_1）的灰度值，（X_1，Y_1）的灰度值是 1101

使用位图的目的是如实地在屏幕上进行显示。因为人眼可以分辨出屏幕上的细小变化，所以图形系统仍面临着挑战。

1.4.2　触摸屏

PC 使用 LCD 来进行显示，而后 PC 时代的平板电脑和智能手机使用接触敏感的显示设备替代了键盘和鼠标。这使其拥有良好的用户界面，用户直接指向感兴趣的内容，而不需要使用鼠标。

触摸屏可采用多种方式实现，许多平板电脑采用电容感应实现。如果绝缘玻璃上覆盖一层透明的导体，人的手指接触到屏幕范围时，由于人是导体，将会使屏幕的电场发生变化，进而导致电容的变化。这种技术允许同时接触多个点，可提供非常好的用户界面。

1.4.3　打开机箱

图 1-7 给出了 Apple iPad 2 平板电脑的内部结构。不难看出，计算机五大传统部件中的 I/O 是该设备的主要部分。iPad 2 的 I/O 设备包括一个电容性的多触点 LCD、前置摄像头、后置摄像头、麦克风、耳机插孔、扬声器、加速计、陀螺仪、Wi-Fi 网络和蓝牙网络。其数据通路、控制器和存储器只占很小一部分。

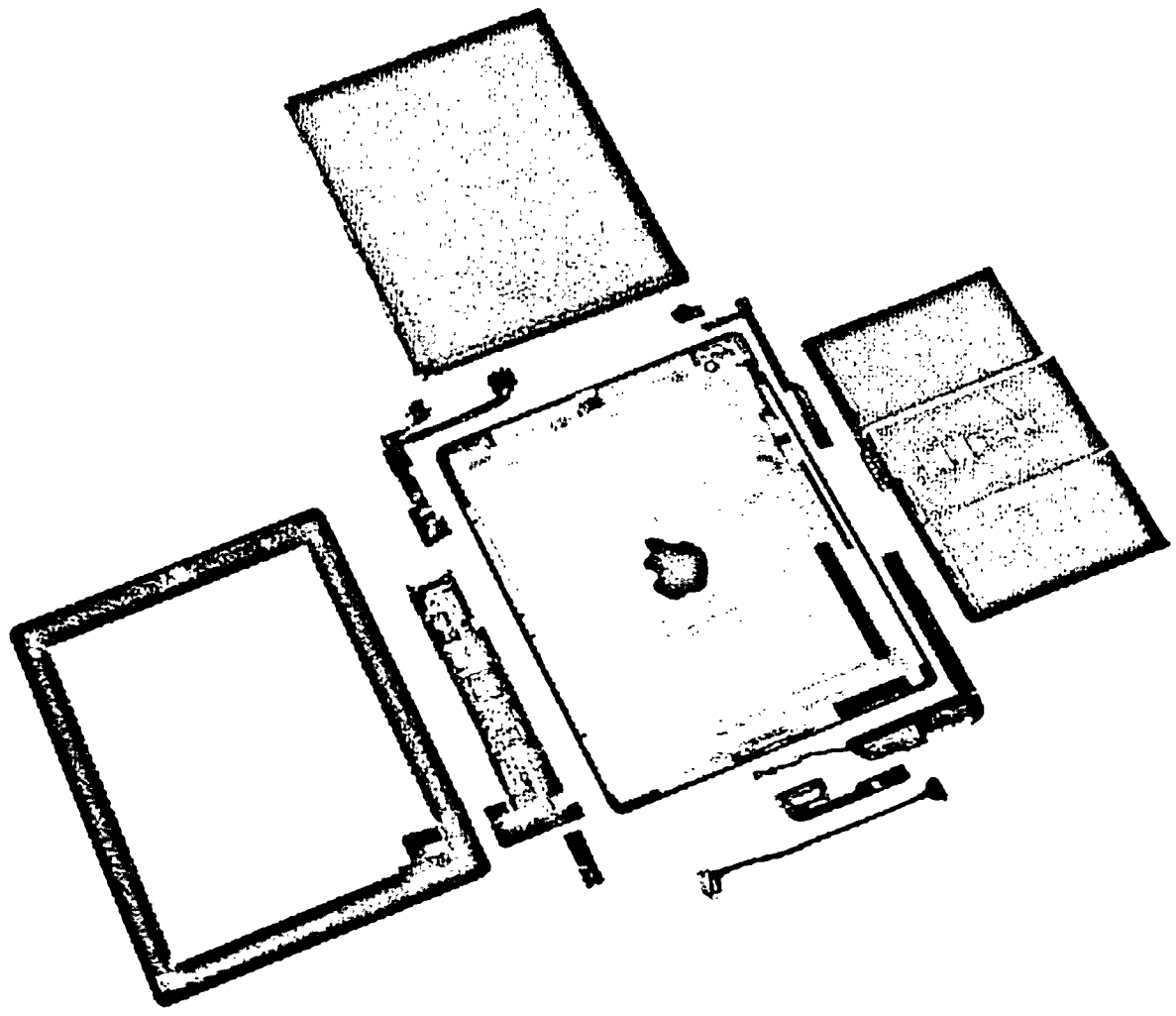

图 1-7　Apple iPad 2 A1395 的组成。中间是 iPad 的金属背板（中心是倒置的 Apple 标志），顶部是电容性触摸屏和 LCD。最右端是 3.8V、25W · h 的聚合物电池，它包含三块锂离子电池芯，可以供电 10 小时。最左端是将 LCD 固定在背板上的金属外壳。金属背板周围的小部件组成了我们熟知的计算机，它们在金属壳内位于电池旁边，呈 L 形排布。图 1-8 显示了靠近金属外壳左下部 L 形的逻辑主板的详细情况，上面有处理器和存储器，其下面的小方块中包含了提供无线通信的芯片，即 Wi-Fi、蓝牙和调频调谐器，它可以插在逻辑主板左下角的插槽中。外壳左上角是另外一个 L 形部件，它是前置摄像组件，包括摄像头、耳机插孔和麦克风。外壳右上角的电路板除了加速计和陀螺仪，还包含了音量控制和静音 / 屏幕旋转锁定按钮。加速计和陀螺仪使得 iPad 可识别六个方向的移动。旁边的小方块是后置摄像头。外壳右下角是 L 形的扬声器组件。底部的电缆连接逻辑主板和摄像 / 声音控制电路板。电缆和扬声器组件之间的电路板是电容性触摸屏的控制器（iFixit 友情提供，www.ifixit.com）

图 1-8 中的小长方形是**集成电路**，俗称**芯片**。其中心标有 A5 的芯片中含有两个运行频率为 1GHz 的 ARM 处理器。处理器是计算机中最活跃的部分。它严格按照程序中的指令运行，完成数据相加、数据测试、按结果发出控制信号使 I/O 设备做出动作等操作。有时候，人们把处理器称为**中央处理单元**（central processor unit），即 CPU。

集成电路：也叫芯片，一种集成了几十个至上亿个晶体管的设备。

中央处理单元：也称为处理器，处理器是计算机中最活跃的部分，它包括数据通路和控制器，能完成数据相加、数据测试、按结果发出控制信号使 I/O 设备做出动作等操作。

为进一步理解硬件，图 1-9 展示了一款微处理器的内部细节。处理器从逻辑上包括两个主要部件：数据通路和控制器，分别相当于处理器的身体和大脑。**数据通路**（datapath）负责完成算术运算，

控制器（control）负责指导数据通路、存储器和I/O设备按照程序的指令正确执行。第4章将进一步详细说明数据通路和控制器。

数据通路：处理器中执行算术操作的部分。

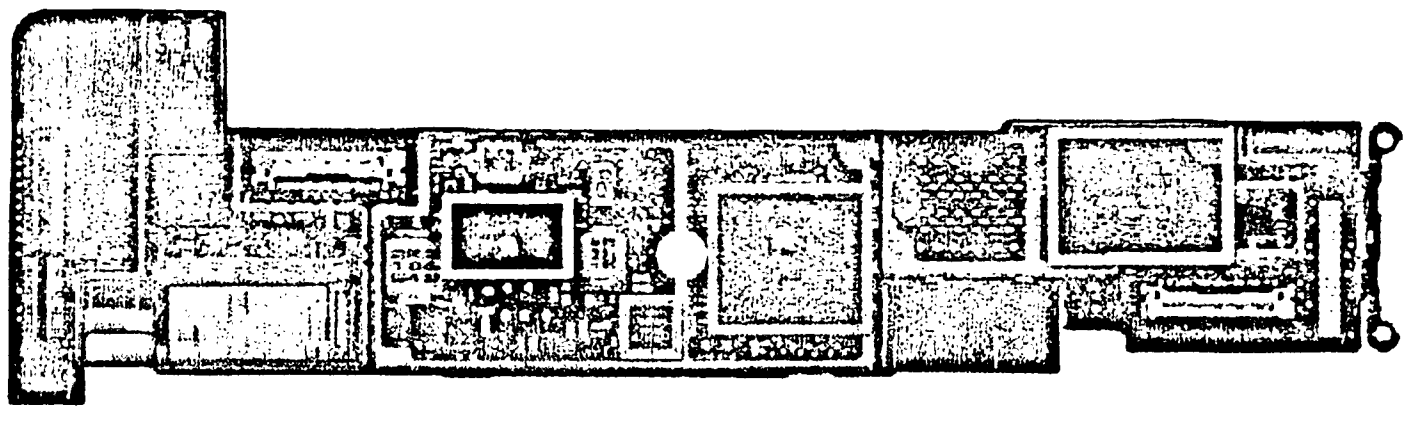

图1-8　图1-7中Apple iPad 2的逻辑主板。图中突出了五块集成电路。中部的大集成电路芯片是Apple A5芯片，包含了一个主频1GHz的双核ARM处理器和512MB的主存。图1-9是A5中处理器芯片的照片。本图右边大小相当的芯片是32GB非易失性的闪存芯片。两块芯片之间的空间可以安装第二块存储器来扩展iPad的存储容量。A5左边的芯片包含了电源控制和I/O控制芯片（iFixit友情提供，www.ifixit.com）

图1-8中的A5芯片中还有两块存储器芯片，每块容量为2Gib，共512MiB。**内存**（memory）是程序运行时的存储空间，它同时也用于保存程序运行时所使用的数据。内存由DRAM芯片组成。DRAM是Dynamic Random Access Memory（**动态随机访问存储器**）的缩写。内存由多片DRAM芯片组成，用来承载程序的指令和数据。与串行访问内存（如磁带）不同的是，无论数据存储在什么位置，DRAM访问内存所需的时间基本相同。

控制器：处理器中根据程序的指令指挥数据通路、存储器和I/O设备的部分。

内存：程序运行时的存储空间，同时还存储程序运行时所需的数据。

进一步深入了解任何一个硬件部件会加深对计算机的理解。在处理器内部使用的是另外一种存储器——高速缓存。**高速缓存**（cache memory）是一种小而快的存储器，一般作为DRAM的缓冲（缓存的一个非技术性定义是：隐藏事物的安全地方）。高速缓存采用的是另一种存储技术，称为**静态随机访问存储器**（Static Random Access Memory，SRAM），其速度更快而且不那么密集，因此价格比DRAM更贵（见第5章）。SRAM和DRAM是存储器层次中的两个层次。

DRAM：动态随机访问存储器，集成电路形式的存储器，可随机访问任何地址的内存。在2012年，其访问时间大约为50ns，每GB的价格为5～10美元。

高速缓存：高速缓存是一种小而快的存储器，一般作为大而慢的存储器的缓冲。

如前所述，改进设计的一个伟大思想是抽象。最重要的抽象之一是硬件和底层软件之间的接口。鉴于其重要性，该抽象被命名为计算机**指令系统体系结构**（instruction set architecture），或简称**体系结构**（architecture）。计算机体系结构包含了程序员正确编写二进制机器语言程序所需的全部信息，如指令、I/O设备等。一般来说，操作系统需要封装I/O操作、存储器分配和其他低级的系统功能细节，以使得应用程序员无须关注这些细节。提供给应用程序员的基本指令系统和操作系统接口合称为**应用二进制接口**（Application Binary Interface，ABI）。

静态随机访问存储器：另一种集成电路形式的存储器，但是比DRAM更快，集成度更低。

指令系统体系结构：也叫体系结构，是低层次软件和硬件之间的抽象接口，包含了需要编写正确运行的机器语言程序所需要的全部信息，包括指令、寄存器、存储器访问和I/O等。

计算机体系结构可以让计算机设计者独立地讨论功能，而不必考虑具体硬件。例如，我们讨论数字时钟的功能（如计时、显示时间、设置闹钟）时，可以不涉及时钟的硬件（如石英晶体、

LED显示、按钮）。计算机设计者将体系结构与体系结构的**实现**（implementation）分开考虑也是沿用同样的思路：硬件的实现方式必须依照体系结构的抽象。这些概念产生了另一个重点。

应用二进制接口：用户部分的指令加上应用程序员调用的操作系统接口，定义了二进制层次可移植的计算机的标准。

实现：遵循体系结构抽象的硬件。

重点 无论硬件还是软件都可以使用抽象分成多个层次，每个较低的层次把细节对上层隐藏起来。抽象层次中的一个关键接口是指令系统体系结构——硬件和底层软件之间的接口。这一抽象接口使得同一软件可以由成本不同、性能也不同的实现方法来完成。

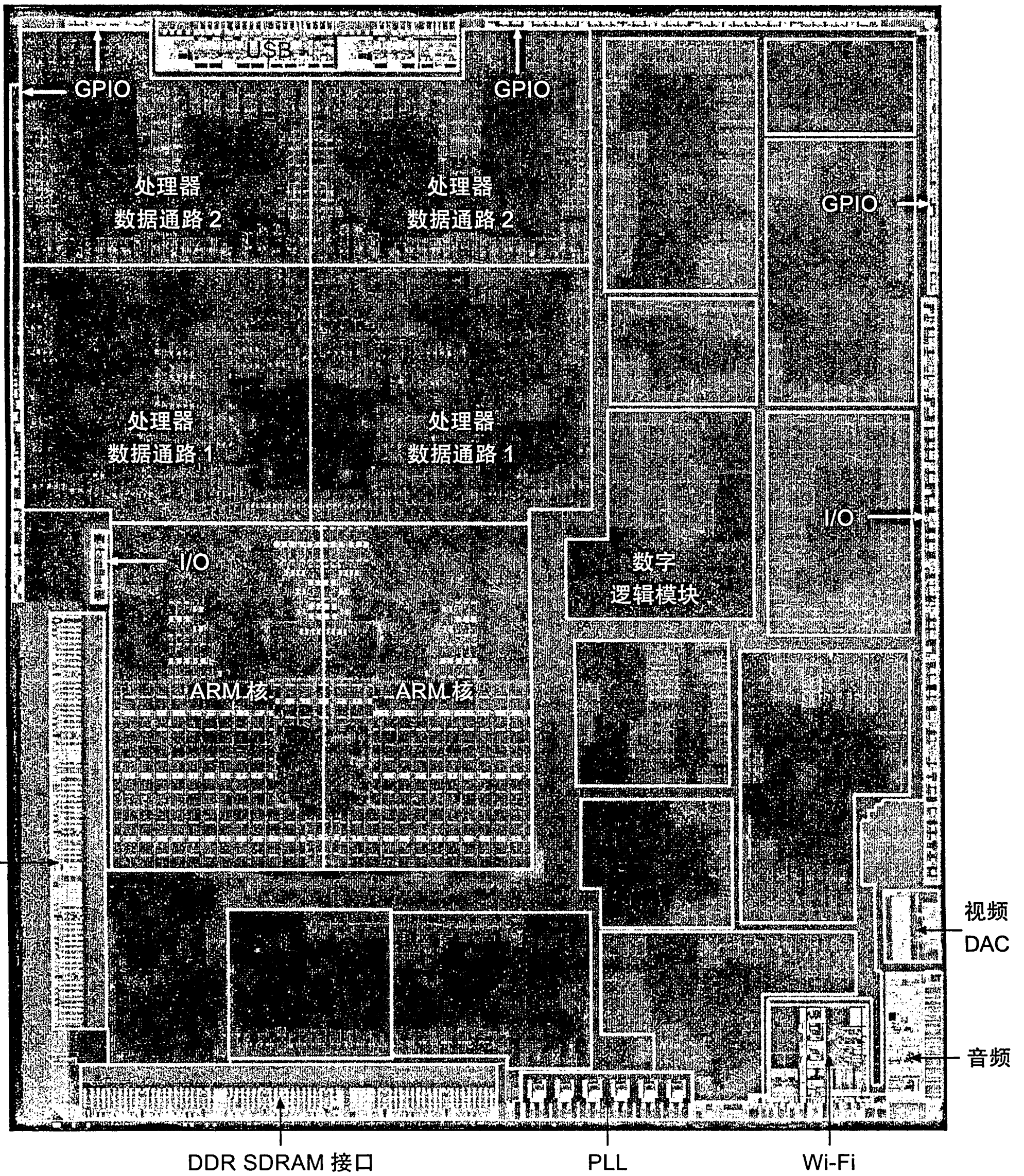

图 1-9 A5 内部的处理器集成电路。芯片尺寸为 12.1mm × 10.1mm，采用 45nm 工艺制造（见 1.5 节）。左半部分靠中间的位置是两个相同的 ARM 处理器，左上角的四分之一部分是具有 4 条数据通路的图形处理单元（Graphic Processor Unit，GPU），左下角和底部是与主存的接口（Chipworks 友情提供，www.chipworks.com）

1.4.4　数据安全

到目前为止，我们已经理解了如何输入数据，如何使用这些数据进行计算，以及如何显示结果。然而，一旦关掉电源，所有数据就丢失了，因为计算机中的内存是**易失性存储**。与之不同的是，如果关掉DVD机的电源，所记录的内容将不会丢失，因为DVD采用的是**非易失性存储**。

易失性存储：类似DRAM的存储器，仅在加电时保存数据

非易失性存储：在掉电时仍可保持数据的存储器，用于存储需运行的程序，例如DVD。

为了区分易失性存储与非易失性存储，我们将前者称为**主存储**（main memory）或**主要存储**（primary memory），将后者称为**辅助存储**（secondary memory）。辅助存储形成了存储层次中更低的一层。DRAM自1975年起在主存储中占主导地位，而**磁盘**在辅助存储中占主导地位的时间更早。由于器件尺寸和前面所述的特点，非易失性半导体存储——**闪存**（flash memory）在个人移动设备中替代了磁盘。图1-8所示的iPad 2中的芯片上包含了闪存。除了非易失性外，闪存比DRAM慢，但却便宜很多。虽然每位的价格高于磁盘，但是闪存在体积、电容、可靠性和能耗方面都优于磁盘。因此闪存是个人移动设备中的标准辅助存储。遗憾的是，与硬盘和DRAM不同的是，闪存在写入100 000～1 000 000次后可能老化或损坏。因此，文件系统必须记录写操作的数目，而且具备类似“移动常用的数据”这种避免存储器损坏的策略。在第5章中将会对磁盘和闪存进行详细介绍。

主存储：也叫主要存储。这种存储用来保持运行中的程序，在现代计算机中一般由DRAM组成。

辅助存储：非易失性存储，用来保存两次运行之间的程序和数据。在个人移动设备中一般由闪存组成，在服务器中由磁盘组成。

1.4.5　与其他计算机通信

我们已经介绍了如何输入、计算、显示和保存数据，但对于今天的计算机来说，还有一项不可缺少的功能：计算机网络。如图1-5所示，处理器与存储器和I/O设备连接。通过网络，一台计算机可以与其他计算机通信，从而扩展计算能力。当今网络已经十分普遍，逐步成为计算机系统的核心组成部分。一台新型个人移动设备或服务器如果没有网络接口将是十分可笑的。联网的计算机具有如下几个主要优点：

- 通信：信息可在计算机之间高速交换。
- 资源共享：I/O设备可以通过网络共享，不必每台计算机都配备。
- 远距离访问：用户无须在要使用的计算机旁边，可远距离连接计算机。

网络的传输距离和性能是多种多样的，根据传输速度以及信息传输的距离，通信代价随之增长。最为普遍的网络类型是以太网。它的传输距离可达到1公里，传输速率可达到40Gbps。根据传输距离和速率特点，以太网可以将一座建筑物中同一层的计算机连接起来，这就形成了通常称为**局域网**（Local Area Network，LAN）的一

磁盘：也叫硬盘，是使用磁介质材料构成的以旋转盘片为基础的非易失性二级存储设备。因为是旋转的机械设备，所以磁盘的访问时间大约是5～20毫秒，2012年每GB的价格大约为0.05～0.1美元。

闪存：一种非易失性半导体内存，单位价格和速度均低于DRAM，但单位价格比磁盘高，速度比磁盘快。其访问时间大约为5～50毫秒，2012年每GB的价格大约为0.75～1美元。

局域网：一种用于在一定地理区域（例如同一栋大楼）内传输数据的网络。

个例子。局域网通过交换机进行连接，可以提供路由与安全服务。**广域网**可跨越大陆，是因特网的骨干构成部分，可支持万维网。它通常以光纤为基础并从通信公司租用。

广域网：一种可以跨越大陆数百公里的网络。

在过去的 30 年间，因为广泛的使用和性能的大幅度提升，网络已经改变了计算的方式。在 20 世纪 70 年代，个人很难接触到电子邮件，网络和 Web 还不存在，物理邮寄的磁带成为两地之间传输大量数据的主要载体。局域网根本不存在，少数几个广域网容量很小且访问受限。

随着网络技术的进步，网络变得越来越便宜，速度越来越快。30 多年前，第一个标准局域网的最大带宽为 10Mbps，只能支持数十台计算机共享工作。今天，局域网技术已能提供从 1Gbps ～ 40Gbps 的带宽。光通信技术已经使广域网有了类似的发展，带宽从几百 Kbps 到 Gbps，支持几百到几百万台计算机与全球网络互连。网络规模的飞速扩大，伴随着带宽的急剧增长，使得网络技术成为最近 30 年来信息革命的中心。

最近 10 年来，新的联网创新变革了计算机通信的方式。推动后 PC 时代的无线技术的广泛应用，加上原本用于无线电的廉价半导体技术（CMOS）被用于存储器和微处理器，使其价格大幅度降低，产量剧增。当前无线通信技术（IEEE 标准 802.11）支持从 1Mbps 到近 100Mbps 的传输速率。无线技术和基于线路的网络相当不同，因为所有用户可以在最近的区域里共享电波。

自我检测 半导体 DRAM 存储器、闪存和磁盘存储器有很大差别。对于任一技术，试从易失性、相对 DRAM 的近似访问时间和相对 DRAM 的近似价格三方面进行比较。

1.5 处理器和存储制造技术

处理器和存储正在以难以置信的速度发展，因为计算机设计者一直采用最新的电子技术进行设计，以期在竞争中取得优势。图 1-10 描述了不断进步的各种新型技术，包括其出现的时间和性价比。这些技术确定了计算机能够做什么，以及以多快的速度发展变化。我们相信，所有计算机专业人员都应该熟悉集成电路的基础知识。

年份	计算机中采用的技术	相对性价比
1951	真空管	1
1965	晶体管	35
1975	集成电路	900
1995	超大规模集成电路	2 400 000
2013	甚大规模集成电路	250 000 000 000

图 1-10 随着时间的推进，不同计算机实现技术的性价比。来源：波士顿计算机博物馆，其中 2013 年的数据由作者推断得到（见 1.12 节）

晶体管仅仅是一种受电流控制的开关。集成电路是由成千上万个晶体管组成的芯片。当戈登·摩尔预测资源持续翻番时，他是在预测单芯片上晶体管数量的增长速度。为了描述这些晶体管从几

晶体管：一种由电信号控制的简单开关。

百个增长到成千上万的情形，形容词超大规模被添加到术语中，即**超大规模集成电路**（Very Large-Scale Integrated Circuit），简写为 VLSI。

超大规模集成电路：由数十万到数百万晶体管组成的电路。

集成度的增长率是相当稳定的。图 1-11 描述了自 1977 年以来 DRAM 容量的发展情况。近 35 年以来，该行业持续发展使得每隔 3 年 DRAM 的容量就翻两番，累积增长已超过 16 000 倍！

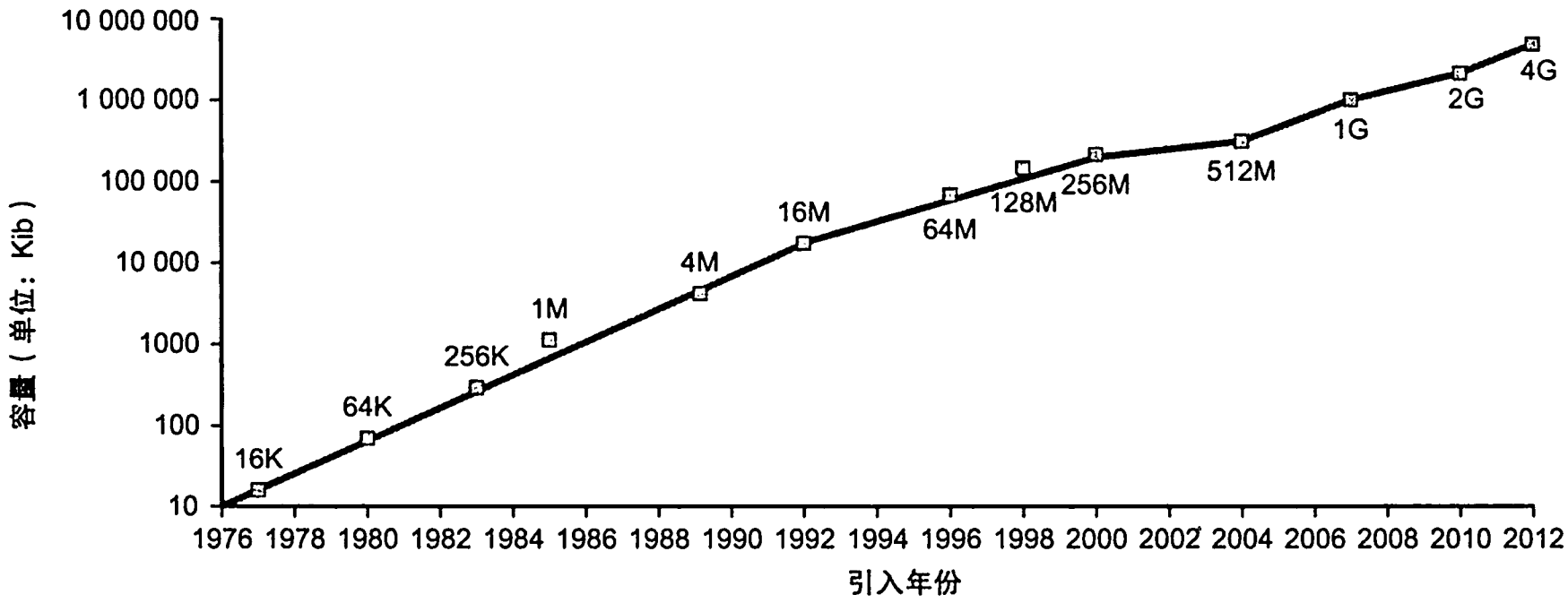

图 1-11　单片 DRAM 容量随时间增长的情况。纵轴单位为 Kib（2^{10} 位）。在过去的 20 多年中，平均每隔 3 年 DRAM 容量翻两番，即每年增长约 60%。在最近几年中，增长速度有所下降，接近每 2 至 3 年翻一番的水平

为了理解集成电路的制造过程，我们从头开始介绍。芯片的制造从**硅**开始，硅是沙子中的一种物质。由于硅的导电能力不强，因此称为**半导体**。使用特殊的化学方法可以对硅添加某些材料，将其细微的区域转变为以下三种类型之一：

硅：一种自然元素，是一种半导体。

半导体：一种导电性能不好的物质。

- 优秀的导电体（细微的铜线或铝线）。
- 优秀的绝缘体（类似于塑料或玻璃膜）。
- 可在特殊条件下导电或绝缘的区域（作为开关）。

晶体管属于第三种类型。VLSI 电路是由数十亿个上述三种类型的材料组合起来并封装在一起制成的。

集成电路的制造过程对芯片的价格非常关键，因此对计算机设计者十分重要。图 1-12 给出了集成电路制造的整个过程。集成电路的制造是从**硅锭**（silicon crystal ingot）开始的，它像一根巨大的香肠。目前使用的硅锭直径约 8 至 12 英寸[⊖]，长度约 12 至 24 英寸。硅锭经切片机切成厚度不超过 0.1 英寸的晶圆。这些晶圆经过一系列化学加工过程最终生成前述的晶体管、导体和绝缘体。如今的集成电路仅包含一层晶体管，但可能具有 2 至 8 层的金属导体，并由绝缘层隔开。

硅锭：一根由单硅晶体构成的圆棒。直径约 8 ～ 12 英寸，长度约 12 ～ 24 英寸。

晶圆：厚度不超过 0.1 英寸的硅锭切片，用于制造芯片。

晶圆或者曝光成像的几十个步骤中出现一个细微的瑕疵就会使其附近的电路失效，这些**缺陷**（defect）使得制成一个完美的晶圆几乎是不可能的。解决这一问题的最简单策略是，将许多独立组件放置在某一晶圆上，然后在曝光成像后切割为**晶片**（die），有时候也不

缺陷：晶圆或者曝光成像过程中的一个微小的瑕疵，晶片可能因为包含这个缺陷而失效。

晶片：从晶圆中切割出来的一个单独的矩形区域，非正式的名称是芯片。

⊖ 1 英寸 = 0.0254 米。——编辑注

正式地称为**芯片**（chip）。图 1-13 的照片所示就是切割之前的微处理器晶圆，而图 1-9 则是单个微处理器晶片的照片。

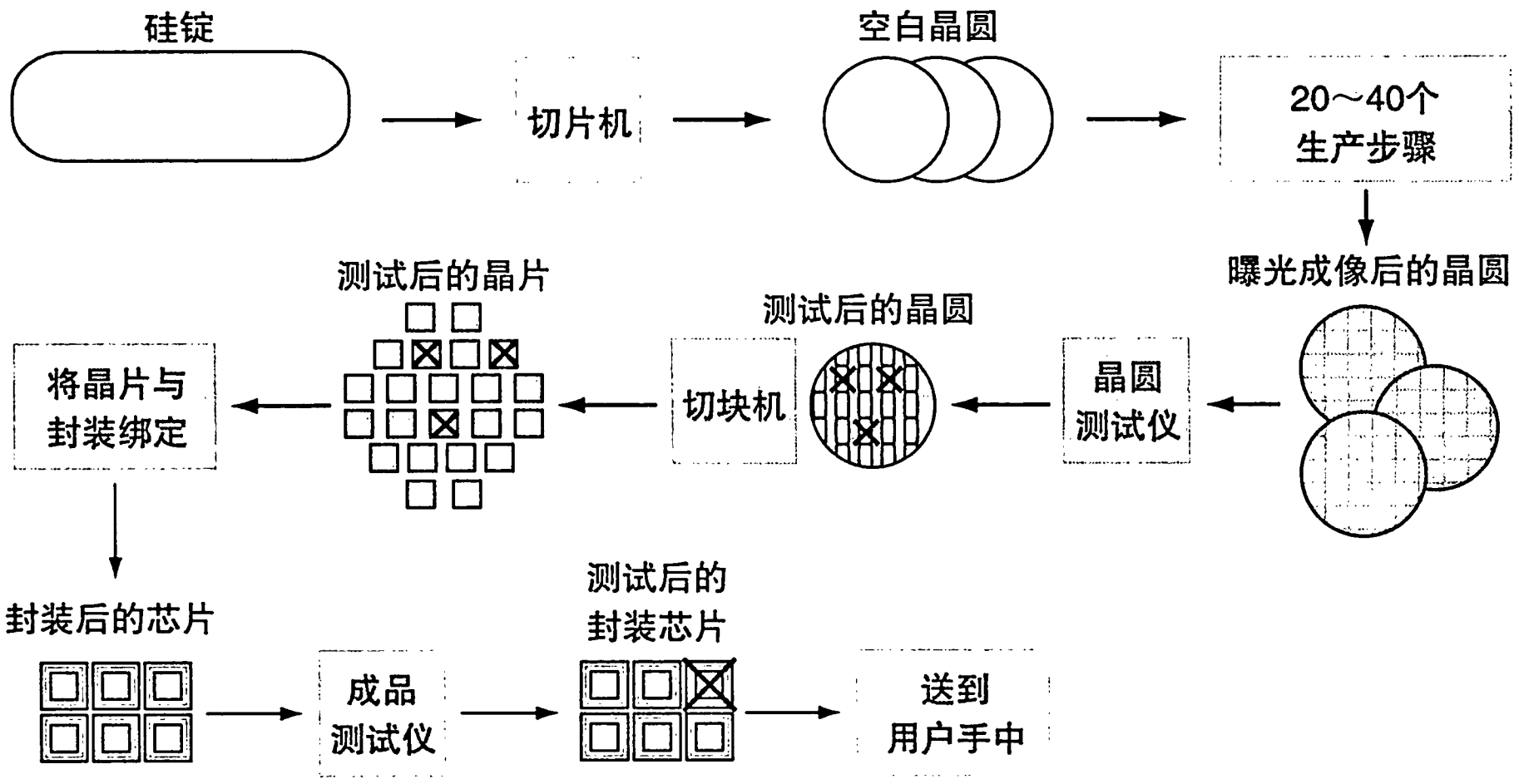

图 1-12　芯片制造的全过程。从硅锭切下来之后，空白的晶圆经过大约 20 ～ 40 步的加工，产生曝光成像后的晶圆（见图 1-13）。这些曝光成像后的晶圆由晶圆测试设备进行测试，测试后生成一张图，表明哪些部分是合格的。之后，这些晶圆被进一步切成晶片（见图 1-9）。在本图中，一个晶圆能生产 20 个晶片，其中有 17 个通过测试（× 表示该晶片存在缺陷）。本例中晶片的良率是 17/20，也就是 85%。这些合格晶片被封装起来并且在发布给用户之前再次测试。不合格的封装会在最终测试中被发现

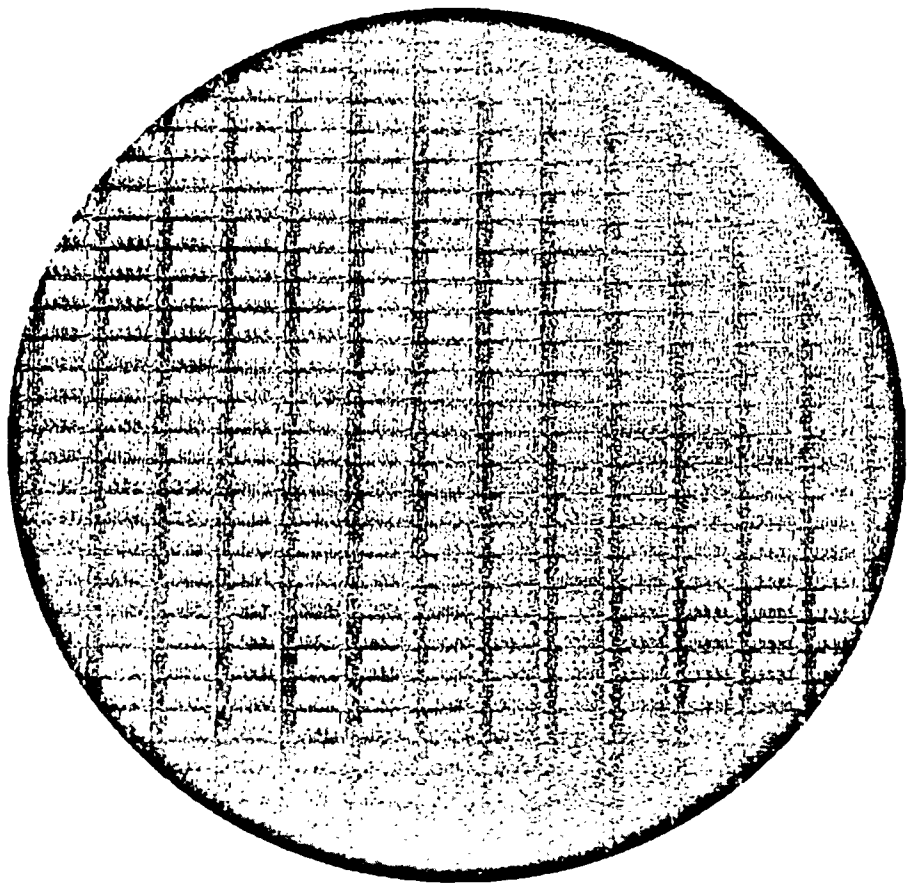

图 1-13　Intel Core i7 芯片的 12 英寸（300mm）晶圆（Intel 提供）。该晶圆在良率为 100% 时，可产生的晶片数目是 280，每个为 20.7mm × 10.5mm。晶圆边缘几十个不完整的芯片是没用的。之所以包含它们，是因为这样给硅片生产用于曝光的掩膜相对容易。该晶片使用 32nm 的工艺，这意味着最小的晶体管的特征尺寸约为 32nm，尽管它们通常比实际的特征尺寸还要小。这个特征尺寸是将晶体管“图纸尺寸”和最终的生产尺寸相比得出的

通过切分，可以只淘汰那些有瑕疵的晶片，而不必淘汰整个晶圆。对这一过程的量化描述可以用工艺**良率**（yield）来表示，其定

良率：合格芯片数占总芯片数的百分比。

义为合格晶片数占总晶片数的百分比。

当晶片尺寸增大时，集成电路的价格会快速上升，因为良率和晶圆中晶片的总数都下降了。为了降低价格，较大的晶片常采用下一代工艺来收缩尺寸，因为它使用了更小的晶体管和导线。这可以改进每晶圆的晶片数和工艺良率。2012 年的典型工艺尺寸为 32nm，这意味着晶片上的最小特征尺寸是 32nm。

合格晶片要连接到 I/O 引脚上，使用压焊工艺形成封装。由于封装过程也可能出错，因此在封装之后必须进行最后一次测试再交付给用户。

详细阐述 集成电路的成本可以用下面 3 个简单公式来表示：

$$\text{每晶片的价格}=\frac{\text{每晶圆的价格}}{\text{每晶元的晶片数}\times\text{良率}}$$

$$\text{每晶圆的晶片数}\approx\frac{\text{晶圆面积}}{\text{晶片面积}}$$

$$\text{工艺良率}=\frac{1}{(1+(\text{单位面积的缺陷数}\times\text{芯片面积}/2))^2}$$

第一个公式是直接导出的。第二个公式是近似的，因为没有减去晶圆边上不满足晶片矩形要求的面积（参见图 1-13）。第三个公式是基于集成电路工厂的良率经验，与重要加工步骤的数量呈指数关系。

因此，晶片的成本取决于工艺良率、晶片和晶圆的面积，与晶片面积之间一般并不是线性关系。

自我检测 产量是决定集成电路价格的关键因素之一。下列哪些理由说明了芯片产量越高成本就越低？

1. 高产量使得在制造过程中能够面向具体设计做适当调节，从而提高良率。
2. 设计高产量芯片的工作量比设计低产量芯片的小。
3. 制造芯片用的掩膜很贵，产量高时每芯片的成本就低。
4. 工程开发的成本高，并且很大程度上与产量无关，故产量高时每芯片的开发成本较低。
5. 产量高时，通常每晶片的面积比产量低时小，因此良率较高。

1.6 性能

对计算机的性能进行评价是富有挑战性的。由于现代软件系统的规模及其复杂性，加上硬件设计者广泛采用了大量先进的性能改进方法，使性能评价变得更加困难。

在不同的计算机中挑选合适的产品，性能是极其重要的因素之一。精确地测量和比较不同计算机之间的性能对于购买者和设计者都很重要。销售计算机的人也需要知道这些。销售人员通常希望用户看到他们的计算机表现最好的一面，无论这一面是否能准确地反映购买者的应用需求。因此，理解怎样才能更合理地测量性能并知晓所选择的计算机的性能限制相当重要。

本节将首先介绍性能评价的不同方法，然后分别从计算机用户和设计者的角度描述性能的度量标准，最后分析这些度量标准之间有什么联系，并提出经典的处理器性能公式，我们在全书中都要使用它进行性能分析。

1.6.1 性能的定义

当我们说一台计算机比另一台计算机具有更好的性能时，意味着什么？虽然这个问题看起来很简单，但如果用客机问题模拟一下，就可以知道其内藏玄机。图 1-14 列出了若干典型客机的型号、载客量、航程、航速等参数。如果要指出表中哪架客机的性能最好，那么我们首先要对性能进行定义。如果考虑不同的性能度量，那么性能最佳的客机是不同的。可以看到，巡航速度最高的是 Concorde（已于 2003 年退出服务序列），航程最远的是 DC-8-50，载客量最大的则是 747。

飞机	载客量	航程（英里）	航速（英里/小时）	乘客吞吐率（载客量×航速）
波音777	375	4630	610	228 750
波音747	470	4150	610	286 700
英国宇航公司/Sud Concorde	132	4000	1350	178 200
道格拉斯 DC-8-50	146	8720	544	79 424

图 1-14 若干商用飞机的载客量、航程和航速。最后一列展示的是飞机运载乘客的速度，它等于载客量乘以航速（忽略距离、起飞和降落次数）

假定用速度来定义性能，这里仍然有两种可能的定义。如果你关心点对点的到达时间，那么可以认为只搭载一名旅客的航速最快的客机是性能最好的。如果你关心的是运输 450 名旅客，那么如图中最后一列所示，747 的性能是最好的。与此类似，我们可以用若干不同的方法来定义计算机性能。

如果你在两台不同的桌面计算机上运行同一个程序，那么可以说首先完成作业的那台计算机更快。如果你运行的是一个数据中心，有好几台服务器供很多用户投放作业，那么应该说在一天之内完成作业最多的那台计算机更快。个人计算机用户会对降低**响应时间**（response time）感兴趣，响应时间是指从开始一个任务到该任务完成的时间，又被称为**执行时间**。而数据中心的管理者感兴趣的常常是提高**吞吐率**或者**带宽**——在给定时间内完成的任务数。因此，在大多数情况下，我们需要对个人移动设备采用不同的应用程序作为评测基准，并采用不同的性能度量标准。个人移动设备更关注响应时间，而服务器则更关注吞吐率。

响应时间：也叫执行时间（execution time），是计算机完成某任务所需的总时间，包括硬盘访问、内存访问、I/O 活动、操作系统开销和 CPU 执行时间等。

吞吐率：也叫作带宽（bandwidth），性能的另一种度量参数，表示单位时间内完成的任务数量。

例题 | 吞吐率和响应时间

下面两种改进计算机系统的方式能否增加其吞吐率或减少其响应时间，或可二者兼得？

1. 将计算机中的处理器更换为更高速的型号。

2. 为系统增加额外的处理器，使用多处理器来分别处理独立的任务，如搜索万维网等。

答案 | 一般来说，降低响应时间几乎总是可以增加吞吐率。因此，方式 1 同时改进了响应时间和吞吐率。方式 2 不会使任务完成得更快，只有吞吐率得到提高。

但是，如果方式 2 对处理任务的需求和吞吐率一样大，系统可能强制后续请求进行排队。在这种情况下，改善吞吐率可同时改进响应时间，因为这会减少队列中的等待时间。所以，在实际的计算机系统中，响应时间和吞吐率往往相互影响。

在讨论计算机性能时，本书前几章将主要考虑响应时间。为了使性能最大化，我们希望任务的响应时间或执行时间最小化。对于某个计算机 X，我们可将性能和执行时间的关系表达为：

$$\text{性能}_X = \frac{1}{\text{执行时间}_X}$$

这意味着如果有两台计算机 X 和 Y，X 比 Y 性能更好，则有

$$\text{性能}_X > \text{性能}_Y$$

$$\frac{1}{\text{执行时间}_X} > \frac{1}{\text{执行时间}_Y}$$

$$\text{执行时间}_Y > \text{执行时间}_X$$

也就是说，如果 Y 的执行时间比 X 长，那么就说 X 比 Y 快。

在讨论计算机设计时，经常要定量地比较两台不同计算机的性能。我们将使用“X 的执行速度是 Y 的 n 倍”的表述方式，即

$$\frac{\text{性能}_X}{\text{性能}_Y} = n$$

如果 X 的执行速度是 Y 的 n 倍，那么在 Y 上的执行时间是在 X 上的执行时间的 n 倍，即

$$\frac{\text{性能}_X}{\text{性能}_Y} = \frac{\text{执行时间}_Y}{\text{执行时间}_X} = n$$

例题｜相对性能

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

答案｜我们知道，如果

$$\frac{\text{性能}_A}{\text{性能}_B} = \frac{\text{执行时间}_B}{\text{执行时间}_A} = n$$

则计算机 A 的执行速度是计算机 B 的 n 倍，故性能之比为

$$\frac{15}{10} = 1.5$$

因此 A 的执行速度是计算机 B 的 1.5 倍。

在以上的例子中，我们可以说，计算机 B 比计算机 A 慢 1/3，因为

$$\frac{\text{性能}_A}{\text{性能}_B} = 1.5$$

意味着

$$\frac{\text{性能}_A}{1.5} = \text{性能}_B$$

简单地说，当我们试图将计算机的比较结果量化时，通常使用术语“和……的性能一样”。因为性能和执行时间是倒数关系，提高性能就需要减少执行时间。为了避免对术语增