

(Industry Standard Architecture, ISA) 总线, 是 IBM 为最初的 PC 设计的。扩展板上有 62 针的连接插头。有 20 个地址信号, 8 个复用的数据输入/输出信号, 6 个中断请求信号及 3 个直接存储器访问 (Direct Memory Access, DMA) 请求信号。DMA 可以使存储设备 (本章最后我们会讲到) 快速地执行存储操作, 这比采用其他方法快得多。通常情况下, 所有读/写内存的操作都是由微处理器来完成的, 但采用了 DMA 后, 其他设备可以不通过微处理器而获得总线的控制权, 进而直接对内存进行读写。

在 S-100 系统中, 所有的部件都安放在扩展板上。就拿 IBM PC 来说, 微处理器、支持芯片及一些 RAM 都安装在一块系统板上, 系统板 (system board) 是 IBM 的“内部称呼”, 但它常常也被称为母板或主板。

1984 年, IBM 推出了个人计算机 AT, 采用的是 16 位的 Intel 80286 微处理器, 这个微处理器可以寻址 16 MB 的存储单元。IBM 保留了原有的总线, 但添加了一个 36 针的插槽。这个插槽包括 7 个地址信号 (其实只需 4 个)、8 个数据输入/输出信号、5 个中断请求信号以及 4 个 DMA 请求信号, 这些信号都是新增的。

微处理器所使用的数据宽度 (从 8 位到 16 位再到 32 位) 和输出的地址信号的数目在不断增长, 当这些超出总线的承受能力时, 总线就需要升级换代了。如果微处理器的处理速度很快时, 也会出现这种情况。早期的总线是为当时的微处理器而设计的, 它们的时钟频率一般是几兆赫兹而不是几百兆赫兹。如果设计出来的总线不适合高速传输的话, 就会出现射频干扰 (RFI), 这会使附近的收音机和电视机产生静电或其他噪声。

1987 年, IBM 推出了微通道体系结构 (Micro Channel Architecture, MCA) 总线。这种总线的某些部分已经成为 IBM 的专利, 如果其他公司使用这种总线, IBM 就会从中收取授权费用。也许就正是由于这个原因, MCA 才没能成为一种工业标准。然而就在 1988 年, 9 家公司 (并不包括 IBM) 联合推出的 32 位 EISA (Extended Industry Standard Architecture) 总线取代了 MAC, 成为了工业标准。近几年, Intel 公司设计的外围部件互连 (PCI) 总线已普遍使用在 PC 兼容机上。

计算机上的各种不同的部件是如何工作的呢? 为了能更好地理解, 让我们再次回到 20 世纪 70 年代中期去看一看。想象一下, 我们正在为 Altair 设计电路板, 或者是在为自己设计的 8080 或 6800 计算机做这样的事情。我们不仅要考虑为计算机设计一些存储器, 用键盘作为输入, 用电视机作为输出; 还要考虑关上计算机时, 如何把存储器中的内容

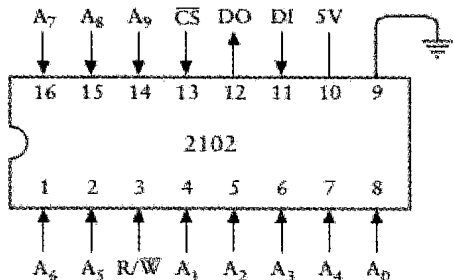
保存下来。如何把这些部件添加到计算机中呢？下面就来看看能实现这个功能的各种接口（Interface）。

现在回想一下第 16 章所讲的内容，RAM 阵列有地址输入、数据输入，以及数据输出信号，另外还有一个用来把数据写入存储器的控制信号。RAM 阵列能存放的数据的数量是和地址输入信号的个数有关的，它们之间有着如下的关系：

$$\text{RAM 阵列中数字的个数} = 2^{\text{地址信号的个数}}$$

讲到这里你可能会问，数据输入、输出信号又有怎样的作用呢？其实它们决定着所存储的数值的大小（位数）。

20 世纪 70 年代中期，2102 是用于家用计算机的一款流行的存储器芯片。其管脚分布如下图所示。



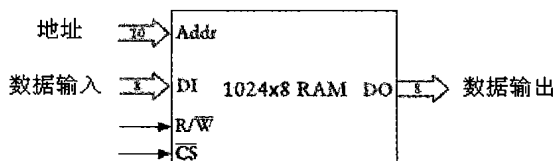
它也是 MOS（metal-oxide semiconductor）家族中的一员，与 8080 和 6800 微处理器所采用的技术相同。MOS 半导体管很容易与 TTL 芯片连接起来；通常情况下，其内部晶体管密度要比 TTL 高，但速度却不如 TTL 快。

这个芯片存储容量可以达到 1024 位，这个数值可以根据地址信号（A0~A9）、数据输出（DO）和数据输入（DI）信号（输入和输出复用一个信号线）的数目计算出来。你所使用 2102 芯片型号不同，访问时间（read access time，指从芯片接收到地址信息到输出有效数据所需的时间）也是各有差异，从 350 ns~1000 ns 不等。当需要从存储器中读取数据时， $\overline{R/\overline{W}}$ （读/写）信号置 1；当向芯片中写入数据的时候，这个信号要置 0，而且至少要持续 170~550 ns 的时间，也是由所使用的 2102 芯片的型号决定的。

这里我们不得不提到的一个信号就是 \overline{CS} 信号，也称片选信号。该信号置 1 时，芯片不被选中，意思就是说，不会响应 $\overline{R/\overline{W}}$ 信号。其实， \overline{CS} 信号的作用不止这些，对芯片

还有其他重要的作用，下面我们将简单描述一下。

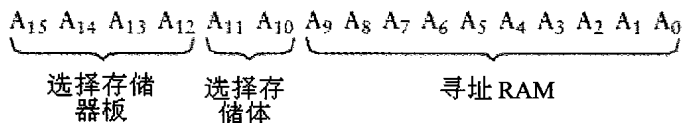
想想看，若让你为 8 位的微处理器组织存储器的话，你会怎么做呢？是选择按 8 位存储形式，还是 1 位存储形式？你肯定会选择前者。如果想存储整个字节，则至少需要 8 个这样的 2102 芯片。具体的做法就是，把 8 个芯片对应的地址信号、 R/\overline{W} 及 \overline{CS} 信号连接起来，如下图所示。



实际上，这是一个 1024×8 位的 RAM 阵列，或者说是容量为 1 KB 的 RAM。

把存储器芯片安装在一块电路板上，这是很符合实际的做法。那么，到底一块电路板上能安装多少块这样的芯片呢？如果是紧紧排列在一起的话，一块 S-100 板就能容纳 64 个。这样一来，就提供了一个 8 KB 的存储空间。一般我们不这样做，更合适的方法是，用 32 个芯片组成一个 4 KB 的存储器。为了存储完整的字节，而连接在一起的芯片的集合，称为存储体 (bank)。例如，一个 4 KB 大小的存储器板就由 4 个存储体组成，而每个存储体又包含 8 个芯片。

8 位微处理器，例如 8080、6800，有 16 位地址，可用来寻址 64 KB 的存储空间。如果你制作了一个包含 4 个存储体、大小为 4 KB 的存储器板，则存储器板上的 16 位地址信号就有如下所示的功能。



下面详细解释一下这 16 位地址信号。A0~A9 直接与 RAM 芯片相连接；A10 和 A11 用来选择 4 个存储体中要被寻址的那一个；A12~A15 确定哪些地址申请用这块存储器板，换言之，就是这块存储器板响应哪些地址。微处理器整个存储空间的大小是 64 KB，被划分成 16 个不同的区域，每个区域的大小是 4 KB，我们设计的 4KB 存储器板占用了其中一个区域。这 16 个区域划分情况如下。

0000h ~ 0FFFh

1000h ~ 1FFFh

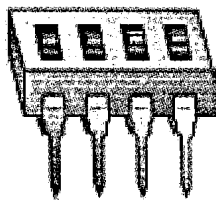
2000h ~ 2FFFh

.....

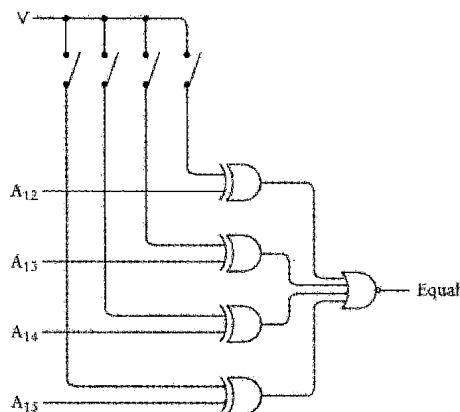
F000h ~ FFFFh

举例说明，假定 4 KB 存储器板使用了 A000h ~ AFFFh 地址区域。这就意味着，第一个存储体占用了地址 A000h ~ A3FFh，第二个占用了地址 A400h ~ A700h，第三个占用了地址 A800h ~ ABFFh，剩下的 AC00h ~ AFFFh 地址空间分给了第四个存储体。

你完全可以制作一块 4 KB 存储器板，在用到它的时候再灵活确定其地址范围。要获得这样的灵活性，可以使用一种名为双列直插式封装（dual inline package，DIP）开关的器件。在 DIP 中，有一系列极小的开关（从 2 到 12 个不等）。DIP 是可以插在标准的 IC 插槽中的，如下图所示。

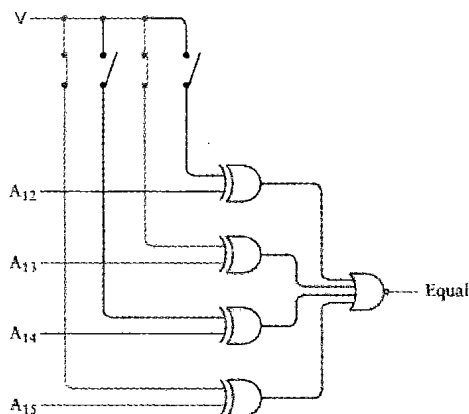


在一种称为比较器（comparator）的电路中，你可以把这个开关和总线上地址信号的高 4 位连接起来，就像下面这样。

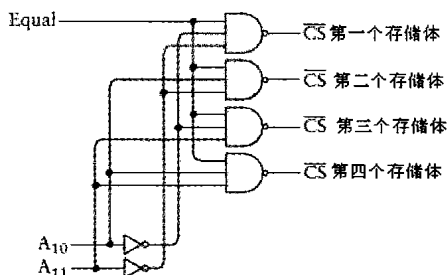


回想一下前面讲过的内容，异或（XOR）门电路在两个输入端中只有一个是高电平时，输出才为高电平；当两个输入端同时为低电平或高电平时，输出是低电平。

例如，如果把 A_{13} 和 A_{15} 对应的开关闭合，就意味着让存储器板能响应存储器空间 $A000h \sim AFFFh$ 。若总线上的地址信号 A_{12} 、 A_{13} 、 A_{14} 和 A_{15} 与开关上设置的值相同的话，四个异或（XOR）门的输出都是 0，或非（NOR）门的输出为 1，如下图所示。



接下来我们把 Equal 信号和一个 2-4 译码器联合起来使用，就能为四个存储体中的每一个都产生一个 \overline{CS} 信号，便于对存储体进行选择。具体连接图如下图所示。



例如，若想选择第三个存储体，把 A_{10} 、 A_{11} 分别置 0 和 1 就可以了。

现在回想一下在第 16 章中阐述过的如何组织 RAM 阵列，这一过程的细节是十分繁琐的，你可能会认为我们还需要 8 个 4-1 选择器，用来从 4 个存储体中选择正确的数据输出信号。但我们并没有这么做，下面来讨论下原因。

通常情况下，TTL 兼容集成电路的输出信号要么大于 2.2V（逻辑 1）要么小于 0.4V（逻辑 0）。试想一下，如果把输出信号连接起来会发生什么呢？一个集成电路的输出为 1，

另一个集成电路的输出为 0，若把这两个输出连接在一起，结果又是什么呢？恐怕谁也无法回答。就是由于这种不确定性，一般不会把集成电路的输出信号连接在一起。

2102 芯片的数据输出信号是三态 (tri-state) 的，也就是说，除了逻辑 0 和逻辑 1 之外，数据输出信号还有第三种状态。我们必须清楚地认识这种状态——它其实是一种“真空”态，就像芯片的引脚上什么也没连一样。当片选信号 (\overline{CS}) 为 1 的时，2102 芯片的数据输出信号就会进入这种状态。这样一来，我们可以把 4 个存储体相应的数据输出信号连接在一起，并且可以把 8 个输出复用作为总线的 8 个数据输入信号。

之所以强调三态输出的概念，是因为它对总线的操作是至关重要的。几乎所有连接在总线上的器件都使用由总线传递而来的数据输入信号。但不管何时，连接在总线上的电路板中只有一个能确定总线数据输入信号的类型，其他电路板处于三种状态中的无效状态。

或许大家听说过，2102 是一款静态随机访问存储器芯片 (Static Random Access Memory, SRAM)，它与动态访问存储器 (Dynamic Random Access Memory, DRAM) 是不同的。通常对于每 1 位存储空间，SRAM 需要用 4 个晶体管 (在第 16 章中讲过将触发器作为存储器用，其用到的晶体管更多)，而 DRAM 只需要 1 个晶体管，但 DRAM 需要较复杂的外围支持电路，这正是它的缺点。

SRAM 芯片，例如 2102，在电源持续供电的情况下，其内容就能保留下来；一旦掉电，其内容就会丢失。在这方面，DRAM 和 SRAM 很类似。但不同的是，DRAM 芯片在使用时需要定期访问其存储器中的内容，尽管有时并不需要这些内容。这一过程称之为更新 (refresh) 周期，每秒钟都必须进行几百次。这种做法就好像为不让某人入睡而每隔一段时间就用手肘轻推他一样。

尽管业界在使用 DRAM 上有些争论，但近年来，DRAM 芯片的容量日益增加，使得 DRAM 最终成为标准。1975 年，英特尔公司推出了一款 DRAM 芯片，容量为 16,384 位。其实，DRAM 芯片在容量上基本每三年翻两番，符合摩尔定律。如今，计算机主板一般都配备内存插槽，这些内存插槽可以容纳几块小存储器板，分为单列直插内存模块 (single inline memory modules, SIMM) 和双列直插内存模块 (dual inline memory module, DIMM) 两种，里面包含好几个 DRAM 芯片。如今，花费不到 300 美元就可以买到 128 MB 的 DIMM 了。

既然已经知道如何制作存储器板了，应该没有人会把微处理器的整个存储空间都分配给存储器，必须留些空间给输出设备。

电子射线管 (cathode-ray tube, CRT)——20 世纪上半个世纪，在家庭中常见的物件，它从外观上看就像电视机一样——已经成为最常见的计算机输出设备了。我们称连接到计算机上的 CRT 为视频显示器 (video display) 或监视器 (monitor)，而称可以为视频显示器提供信号的电子元件为视频适配器 (video display adapter)。通常在计算机中，视频适配器是独立存在的，它们拥有自己的电路板，也就是我们常说的显卡 (video board)。

表面上看来，视频显示器或电视机的二维图像很复杂，但实际上它是由一束连续的光束射线迅速扫描屏幕而形成的。射线从屏幕左上角开始，从左到右进行扫描，到达屏幕边缘后又折回向左，进行第二行扫描。我们称每一个水平行为扫描行 (scan line)，称射线回到每个扫描行的开始位置为水平回归 (horizontal retrace)。当完成了对最后一行的扫描时，射线不会停下来，它会从屏幕的右下角返回到屏幕的左上角 (垂直回归, vertical retrace)，并重复上一过程。就拿美国的电视信号来说，每秒钟要进行 60 次 (称为场频, field rate) 这样的扫描。由于扫描的速度很快，所以不会看到图像出现闪烁的现象。

电视机采用的是隔行 (interlaced) 扫描技术，情况要复杂些。我们先来看一下帧 (frame) 的概念，帧是一个完整的静态视频图像，两个场 (field) 才能形成一个单独的帧。整个帧的扫描线分由两个场来完成——偶数扫描线属于第一个场，奇数扫描线属于第二个场。这里要说明一下水平扫描频率 (horizontal scan rate) 的概念，即扫描每个水平行的速率，例如 15,750 Hz。把这个数除以 60 Hz，结果是 262.5 行，这正是每个场所包含的扫描线的数目，整个帧的扫描线的数目是场的两倍，也即 525 行。

不管隔行扫描技术是怎样实现的，组成视频图像的连续射线都是由一个连续的信号所控制。虽然一套电视节目的声音和图像部分是一起播出的，但若想把它们广播出去或者通过有线电视系统传送出去，就不得不分开进行。这里所说的视频信号其实与 VCR、录像机、摄像机及一些电视机上的视频输入或输出信号是一样的。

黑白电视机的视频信号十分简单且易于理解 (彩色电视机要稍微复杂些)。每秒钟扫描 60 次，扫描信号包含一个垂直同步脉冲 (vertical sync pulse)，用来指示一个场的开始。这个脉冲为 0 V，宽度约为 400 ms。相比较而言，水平同步脉冲 (horizontal sync pulse) 则用来指示每个扫描行的开始：视频信号为 0 V，宽度为 5 ms，每秒钟出现 15,750 次。

在两个水平同步脉冲之间，信号的电压是在 $0.5 \sim 2.0 \text{ V}$ 范围内变化的，其中 0.5 V 表示黑色， 2.0 V 表示是白色，处于两者之间的电压则表示一定的灰度。

正是由于上述原因，电视才会出现部分是数字图像、部分却是模拟图像的情况。虽然在垂直方向上，图像被分为 525 行，但每个扫描行的电压却是连续变化的——用来模拟图像的可视强度。这并不等于说，电压可以随意地变化。事实上，电视机能响应的信号变化频率是有上限的，我们称这一上限为电视机带宽 (bandwidth)。

在通信领域中，带宽是极其重要的概念，某个特定的传输媒介能够传输的信息量都是受带宽限制的。以电视机为例，带宽限制了视频信号从黑到白然后又回到黑这一变化的速率。对于美国的广播电视来说，带宽大约为 4.2 MHz 。

一旦我们把视频显示器连接到计算机上，就不该把它作为模拟和数字的混合设备来对待，把它看做是完完全全的数字设备更合适一些。从计算机的角度来说，我们可以很方便地把视频图像想象成由离散点组成的矩形网格，这些离散点称为像素（这一术语来自 picture element）。

水平扫描行上像素的个数是受带宽严格限制的。在这里，我把带宽定义为视频信号从黑到白然后又回到黑的变化速率。如果电视机的带宽为 4.2 MHz ，它就允许 2 个像素每秒 420 万次的变化，或者——用 $2 \times 4,200,000$ 除以水平扫描速率 15,750——每个水平扫描行有 533 个像素。但并不是所有的像素都可用，约 $1/3$ 的像素被隐藏了起来——处于图像的远端或射线的水平回归中。这样算来，水平扫描行上可用的像素约为 320 个。

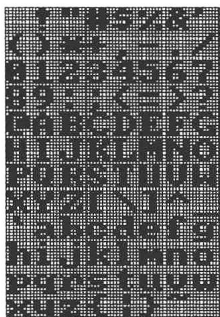
与水平方向类似，垂直方向上 525 个像素也不是都可用。原因是，像素在屏幕的顶部、底部以及垂直回归期间都会有所损失。当计算机采用电视机作为显示器时，就不依赖于隔行扫描技术了，垂直方向上有着合理的像素数目 200。

因此我们可以说，早期普通电视机上的视频适配器的分辨率为 320×200 ，即水平方向上有 320 个像素、垂直方向上有 200 个像素。



如何确定上面网格中像素的总数呢？你可以去数一下，也可以简单地用 320 乘以 200 得出结果 64,000。每个像素可以是黑色或白色，或者为某一特定的颜色，这要取决于视频适配器的配置。

现在我们要在显示器上显示出一些文本字符，那么到底能显示出多少呢？很明显，这取决于每个字符所用的像素数。下面是一种可行的方法，每个字符使用 8×8 的网格（64 个像素）。



上图中显示出的字符对应的 ASCII 码在 20h ~ 7Fh 区间（在 00h ~ 1Fh 的 ASCII 码字符都是无法显示的字符）。

每个字符都被定义为一个 7 位的 ASCII 码，但每个字符也与 64 比特（位）相关，这 64 比特决定了字符会显示为什么样子。当然，你也可以把这 64 位信息当做代码看待。

上面我们对字符进行了定义，使用这些定义，分辨率为 320×200 视频显示器的每一屏就能显示 25 行、每行 40 个字符，足够把艾米·洛威尔（Amy Lowell, 1874-1925）的一首短诗显示出来，看看下面的图。



视频适配器中必须配置一些 RAM，用以存储所显示的内容；微处理器也必须能够向此 RAM 中写入数据以改变显示器上显示的内容。更方便的是，这个 RAM 也是微处理器存储空间的一部分。那么，上面描述的显示适配器需要多大的 RAM 呢？

这个问题并不太好回答！我们只能说，结果可能处于 1 KB ~ 192 KB 之间。

我们从最简单的情况去考虑。怎样减少显示适配器的内存需求呢？一种方法是限制适配器的功能，让其只显示文本。我们已经明确地知道，视频显示器的每屏幕能显示 25 行、每行 40 个字符，也可以说，总共能显示 1000 个字符。这样一来，视频卡上的 RAM 只需存储这 1000 个字符的 7 位 ASCII 码。 $1000 \times 7\text{bit}$ ，大小约为 1024 字节，即 1 KB。

字符生成器 (character generator) 也是视频适配器板上的一部分，包含了所有 ASCII 码字符的像素图，这点前面已经讲过。通常，它是只读存储器 (read-only memory)，即 ROM。它是一种集成电路，在生产时里面已经填入了数据，固定的地址输出的数据是不变的。ROM 中并没有数据输入信号，这点与 RAM 不同。

你可以把 ROM 看成是可以进行代码转换的电路。每片 ROM 都有 7 个地址信号（用来表示 ASCII 码）及 64 个数据输出信号，里面存储了 128 个 ASCII 码字符的 8×8 像素图。因此，ROM 可以实现 7 位 ASCII 码到 64 位码（定义了字符显示的外观）的转换。但是你有没有想过，64 个数据输出信号会使芯片变得很大。更合适的做法是，用 10 个地址信号和 8 个输出信号。其中 7 个地址信号是用来确定 ASCII 码字符的（这 7 个地址位来自视频板上 RAM 的数据输出）。其他三个地址信号则用来表示行。举个例子来说，最高行用 000 表示，最低行用 111 表示。8 个输出位就是每行的 8 个像素。

我们来做个假设，ASCII 码为 41h，就是大写的字母 A。总共有 8 行，每行 8 位。下表给出了字母 A 的 10 位地址（ASCII 码和行代码之间用空格分开）和数据输出信号。

地址	数据输出
1000001 000	00110000
1000001 001	01111000
1000001 010	11001100
1000001 011	11001100
1000001 100	11111100
1000001 101	11001100
1000001 110	11001100
1000001 111	00000000

从上表中，你能看出以 0 为背景、用 1 表示的字母 A 吗？

只显示文本的视频显示适配器还必须支持光标（Cursor）功能。光标是一个小小的下画线，用来表明从键盘上输入的下一字符会在屏幕的什么位置显示出来。光标所在的行和列常被存储在两个 8 位的寄存器中，这两个寄存器也是视频板的一部分，而且微处理器可以对其进行写操作。

有的显示适配器不仅仅只显示文本，还可以显示其他数据，我们称这样的显示适配器为图形适配器（图形显卡）。通过向图形显卡上的 RAM 写入数据，微处理器就可以画出图形了，当然能显示各种大小和样式的文本。相比较而言，图形显卡要比只显示文本的显卡所需的存储空间更大。320×200 的图形显卡有 64,000 个像素，如果每个像素需要 1 位 RAM，那么这样的图形显卡就需要 64,000 位的 RAM，即 8000 字节。然而，这只是最低的要求。1 位是和 1 个像素相对应的，只能用来表示两种颜色——例如黑白两色。0 可能对应于黑色像素，1 可能对应于白色像素。

让我们仔细观察一下黑白电视机，很快会发现，它们不仅仅只显示黑色和白色，还能显示不同灰度的色彩。为了让图形显卡拥有这种功能，通常每个像素对应于 RAM 中的一整个字节，其中 00h 表示的是黑色，FFh 表示的是白色，介于两者之间的数值对应不同的灰度。一个 320×200 的视频板若能显示 256 种灰度，就需要 64,000 字节的 RAM。这与一直在讨论的某个 8 位微处理器的整个地址空间非常接近。

如果想显示出丰富多彩的颜色，每个像素就需要至少 3 个字节。如果现在你手头有放大镜的话，不妨用它观察一下彩色电视机或计算机视频显示器，你会发现，每种颜色都是由红、绿、蓝三原色的不同组合而形成的。为了获取所有的颜色，三原色中每种颜色的强度都需要用一个字节来表示。这么算来，就需要 192,000 字节的 RAM（更多有关彩色图形的内容将在本书最后一章介绍）。

图形显卡到底能显示出多少种不同的颜色呢？这与每个像素所赋予的比特数是有关的。对于这种关系，你可能会感到很熟悉，因为本书中讲到的很多编码都与之类似，它们都涉及 2 的幂，它们之间的关系如下：

$$\text{颜色数量} = 2^{\text{每个像素赋予的比特数}}$$

在标准的电视机上，320×200 的分辨率是所能达到的最高分辨率。正是由于这样的原

因，我们要为计算机特制显示器，以使其具有比电视机更高的带宽。1981 年，第一台显示器随 IBM PC 一起销售，它可以显示 25 行，每行 80 个字符。这正是使用在 IBM 大型机上的 CRT 显示器能显示的字符数目。对于 IBM 来说，80 个字符具有特殊的意义，为什么这样说呢？因为它和 IBM 的打孔卡片（punch card）上的字符数目一样。的确，早期连接到主机上的 CRT 显示器常被用来显示打孔卡片上的内容。偶尔，你会听到有人称只显示字符的视频显示器为卡片，当然这是一种过时的叫法。

这么多年以来，视频适配器的分辨率以及能显示的颜色不断增加，这两者也成为了视频显示适配器的重要参数。到了 1987 年，水平 640 像素、垂直 480 像素的视频适配器被 IBM 的 PS/2 个人计算机和苹果公司的 Macintosh II 机采用，这种适配器的出现起到了里程碑的作用，因为从那时起 640×480 就是视频分辨率的最低标准了。

640×480 的分辨率具有很重要的意义。也许你可能无法相信，它之所以那么重要，是因为它和托马斯·爱迪生（Thomas Edison, 1847–1931）有关。大概在 1889 年，爱迪生及他的工程师威廉·肯尼迪·劳里·迪克生（William Kennedy Laurie Dickson）正在进行活动电影摄影机和活动电影放映机的研究，他们决定：让电影图像的宽比高多出 1/3。图像的宽和高之比，称为屏幕长宽比（aspect ratio）。通常，我们把爱迪生和迪克生所确定的这个比表示成 1.33:1，或者不想使用小数点的话，就表示成 4:3。60 多年了，大多数电影一直采用这个比例，电视机也是如此。但在 20 世纪 50 年代早期，好莱坞引入宽屏（widescreen）技术，与电视展开竞争，并最终打破了这个比例。

多数计算机的显示器的长宽比也是 4:3，如果你不信的话，可以用尺子实际量一下，就可以证明我所说非虚。640×480 分辨率也是这个比例。这就说明（打个比方）100 个像素的水平线和 100 个像素的垂直线有着相同的物理长度。对于计算机图形学来说，这是个非常重要的特性，我们称为正方形像素（square pixel）。

如今，视频适配器和显示器都支持 640×480 的分辨率，但同样也支持多种其他的视频模式，包括 800×600、1024×768、1280×960、1600×1200。

我们常常认为计算机的显示器和键盘之间存在某种联系——在键盘上输入什么，显示器就会显示什么——但物理上它们是分开的。

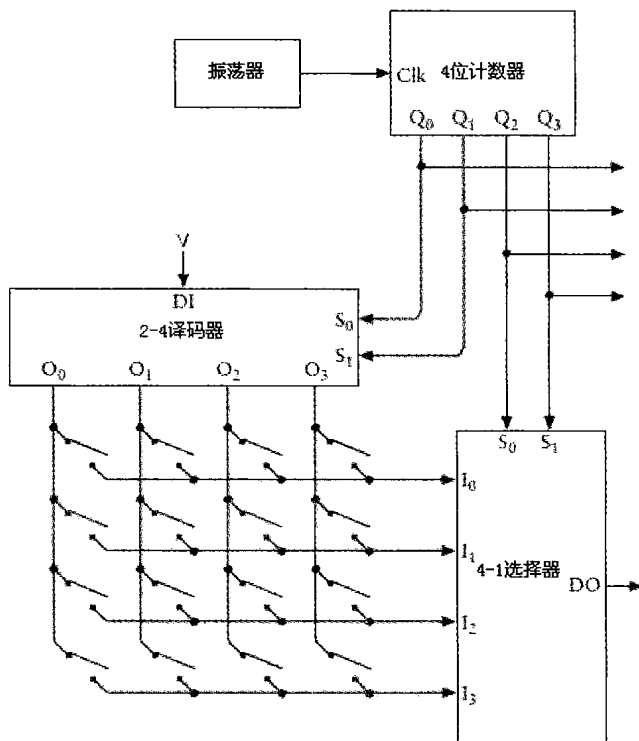
其实键盘上的每个按键就是一个简单的开关。按键按下，开关就会闭合。现在个人

计算机的键盘有 100 多个按键，但类似于打字机的键盘可能只有 48 个按键。

如果要让连接到计算机上的键盘能正常工作的话，就需要配备一些硬件来为每个按键提供唯一的代码，以便区分哪一个按键被按下了。假定这个代码就是按键的 ASCII 码，这样可行吗？你的答案或许是肯定的，但要设计出能识别 ASCII 码的硬件却是不切实际的。举例来说，键盘上的按键 A 对应的 ASCII 码可能是 41h，也可能是 61h，具体是哪个，还取决于用户是否按下了 Shift 键；另外，现在计算机键盘上有很多的按键并没有 ASCII 码与之对应。我们称键盘硬件提供的代码为扫描码（scan code）。当按下键盘上的某个按键时，一小段计算机程序就会计算出这个按键对应的 ASCII 码（如果有的话）。

这里为了避免键盘硬件的电路图太复杂，假设键盘上只有 16 个按键。任何一个按键被按下，键盘硬件就会产生一个 4 位的代码，二进制数值范围是 0000 到 1111。

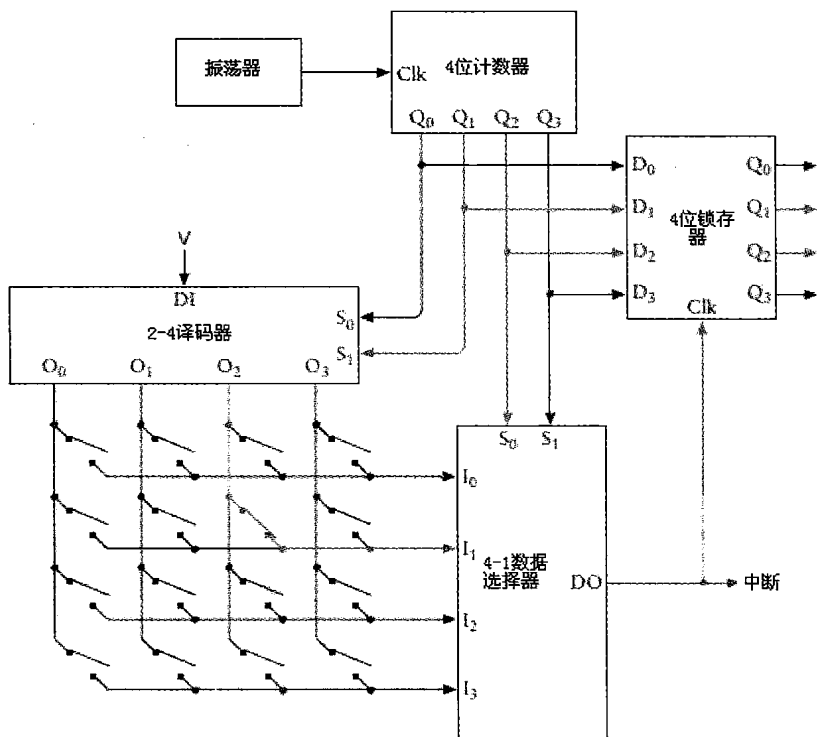
键盘硬件包含了一些前面曾讲过的部件，如下图所示。



上图左下部分所示的是键盘的 16 个按键，简单地用开关表示。4 位的计数器在按键

对应的 16 个编码间快速且重复地循环着，循环的速度必须足够快，以保证在按下并松开一个按键之前循环已经结束。

4 位计数器的输出同样也是 2-4 译码器和 4-1 数据选择器的输入。在没有按键按下的情况下，选择器的输入全都不为 1，因此，其输出也不为 1；一旦有某个按键被按下，而且与 4 位计数器某一特定输出相对应，那么选择器的输出就为 1。例如，如果右上角对角线方向的第二个开关被按下，且计数器的输出是 0110，选择器就会输出 1，如下图所示。



0110 就是这个按键的代码。在这个按键按下的情况下，计数器的其他输出都不会使选择器的输出为 1，也就是说每个按键都代码都是唯一的。

扫描码的位数是由键盘上按键的数目确定的。如果键盘上有 64 个键，就需要 6 位的扫描码，也就需要一个 6 位的计数器。用一个 3-8 译码器和一个 8-1 选择器就可以把这些按键组成一个 8×8 的阵列。如果键盘上的按键数目为 65~128 个，就需要 7 位的扫描码。你就可以用一个 4-16 译码器和一个 8-1 的选择器（或者一个 3-8 译码器和一个 16-1 选择

器)把这些按键组成一个 8×16 的阵列。

在这个电路中,接下来将会发生什么事情呢?这取决于键盘接口。每个按键都应该在 RAM 中拥有 1 位的存储空间,这是设计键盘硬件时该考虑的事情。而且这些 RAM 是由计数器寻址的, RAM 的内容为 0 或 1,具体是什么值取决于按键按下(RAM 为 1)与否(RAM 为 0)。微处理器是可以读取 RAM 中的内容的,并通过内容判断每个按键的状态。

中断信号是键盘接口一个很有用的信号。回想一下前面讲过的内容,我们知道 8080 有一个输入信号允许外部设备中断当前微处理器正在进行的工作。微处理器是通过从内存中读取一条指令来响应中断的,通常是一条 RST 指令。这条指令使微处理器跳转到内存中一个特定的区域并执行其中的中断处理程序。

最后,我们要介绍一下能够长期存储信息的外围设备。前面曾提到,无论是用继电器、电子管,还是用晶体管作为介质构成随机访问存储器,一旦掉电,它存储的内容就会丢失。正因如此,能够在掉电时长期保存信息的存储器,是一台完整的计算机不可或缺的组成部分。长期以来,人们通过在纸上或卡片上打孔来保存永久信息,IBM 打孔卡片是其中典型的代表。在早期小型计算机中,为了能够长久保存程序和数据,通常在滚动的纸带上打孔,而在需要时,这些程序和数据可以从纸带加载到内存。

打孔卡片和纸带的使用也不是尽善尽美的,它也存在一些问题。首先是介质的不可重用性,一旦打孔卡片或纸带被打孔后就很难还原为原来的状态。其次是效率很低,假如你有机会看到当时纸带上保留的某一位信息,就会发现这种做法实在太浪费纸带了。

正是由于这些原因,打孔卡片和纸带慢慢退出了历史的舞台。磁介质存储器(magnetic storage)逐渐发展成目前最为流行的长期存储器。磁介质存储器的起源要追溯到 1878 年,这一年美国工程师奥柏林·史密斯(Oberlin Smith, 1840-1926)描述了它的工作原理。1898 年,即工作原理被提出 20 年后,第一块可用的磁介质存储器问世,它由丹麦发明家巴尔德马尔·波尔森(Valdemar Poulsen, 1869-1942)制造。波尔森后来发明了录音电话机,当家里没人时,通过它可以记录收到的电话信息。声音通过电磁铁和可变长度的金属丝来记录,其中电磁铁是电报机里很常见的部件,它根据声音的高低来磁化金属丝。当磁化的金属丝切割电磁线圈运动的时候,产生的电流强度与其磁化程度有关。不论使用何种磁化介质,记录和读取信息都是利用电磁铁的磁头(head)来完成的。