



13 | 存储电路：计算机存储芯片的电路结构是怎样的？

2021-11-24 海纳

《编程高手必学的内存知识》

[课程介绍 >](#)



讲述：海纳

时长 23:41 大小 21.69M



你好，我是海纳。

之前的课程，我们从软件的角度学习了内存管理的基本知识。从这一节课开始，我们把注意力转向内存的硬件实现。掌握硬件篇的知识，是你学习计算机组成原理和体系结构的基础。而且，计算机体系结构中最常用的手段就是合理地使用各种器件，通过体系手段来使得它们扬长避短，形成有机的整体。

可以说，深刻地掌握计算机的体系结构，就是你写出高性能代码的关键。那么，这么重要且基础的部分，为什么我会放到现在才讲呢？这是因为，程序员日常打交道的是软件接口，硬件的感知度不高。所以在有了前面软件篇的知识后，我们才能更好地理解硬件上的各种晦涩的概念。



整个硬件篇的内容主要就是聚焦于，各种不同的存储器和它们的器件是如何组成高效、大容量、低成本的存储体系结构的。而各类存储器的基本原理是存储体系结构的基础。

我们把用于存储数据的电路叫做存储器，按照到 CPU 距离的远近，存储器主要分为寄存器、缓存和主存。今天这节课，我们就来重点分析这三种存储器的特点、原理，以及应用场景。

存储器是由基本的存储单元组成的，要想搞清楚存储器原理，我们还要先搞明白基本的存储单元是什么，它又是怎么工作的，我们先按寄存器、缓存和主存的顺序，逐个分析。

首先，我们来看寄存器的存储单元是什么样的。

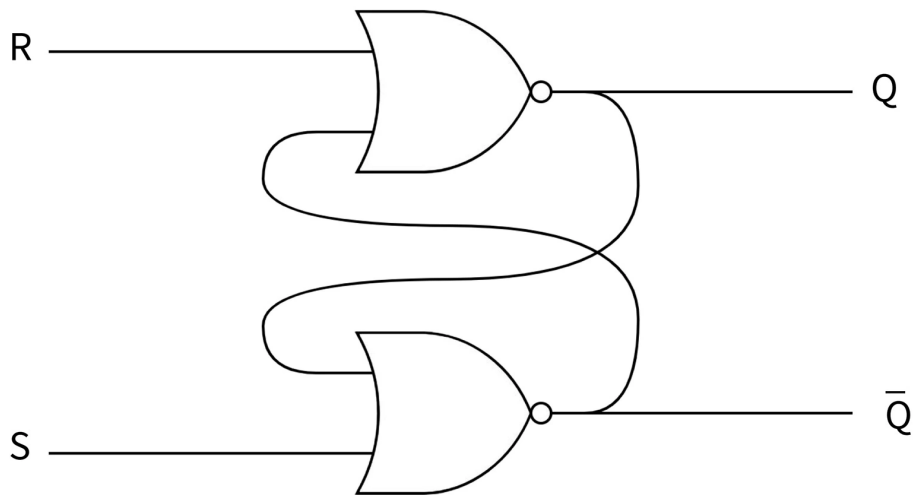
寄存器存储单元

在 [导学（一）](#) 里，我曾经讲过组合逻辑电路和时序逻辑电路的区别。**组合逻辑电路是指，输出仅由输入信号的状态决定的电路。而时序逻辑电路是指，电路的输出值同时依赖于电路过去的状态和现在的输入，所以时序逻辑电路中含有用于记忆电路状态的存储单元。**

接下来，我们就从最简单的具有存储功能的电路开始，逐步将它扩展成相对复杂的存储电路，以此来深度拆解存储电路的运行原理。

RS 锁存器

我们把具有存储信息能力的电路，称为存储器。其中，RS 锁存器 (Latch) 是最简单的一种存储器电路，它可以存储一个比特，如下图所示：



上图中的电路由两个或非门组成，它的特点是，图上方的或非门的输出作为图下方的或非门的输入，反过来，图下方的或非门的输出，也是图上方的或非门的输入。我来分析一下这个电路的特点，你就能理解这个电路是如何完成一个比特的存储的。

一开始，输入端 R 和 S 都是低电压，代表 0，上方的或非门输出为 0，下方的或非门输出为 1，这是一种合法状态。或者下方的或非门的输出为 0，上方的或非门的输出为 1，这也是一种合法状态。也就是说这个电路在 R 和 S 都为 0 的时候，有两种合法的稳定状态。

如果此时，S 变成高电压，也就是 1，那么下方或非门的输出就变成 0，进而导致上方或非门的输出变成 1，也就是 Q 变成 1。这个时候，如果 S 又变成低电压的话，因为上方或非门的输出为 1，所以下方或非门的输出仍然保持为 0。而输入端 R 仍然为 0，这就使得上方或非门的两个输入端都为 0，进而 Q 的高电压可以得到保持。

于是，我们就看到了神奇的一幕：输入端 S 变为 1 以后，可以将输出端 Q 变成 1，但是当 S 变为 0 以后，输出端仍然保持 1。这就说明这个电路可以存储 1。

当 R 和 S 都是 0、Q 为 1 时，如果此时 R 变成 1，由于电路是对称的，可以很容易分析得到 Q 将变成 0，Q 反（表示 \bar{Q} ，下同）变成了 1。同样地，如果此时 R 再变成 0，Q 为 0，Q 反为 1 的状态仍然可以保持。这就实现了一个可以保持一个比特的存储器。

经过上面的分析，我们总结出它的真值表，如下所示：

Reset(R)	Set(S)	Q	\bar{Q}
0	0	Q	\bar{Q}
1	0	0	1
0	1	1	0
1	1	-	-



简单说，就是当 S 为 1，R 为 0 时，Q 为 1；当 S 为 0，R 为 1 时，Q 为 0。当 S 和 R 都为 0 时，Q 保持原来的状态，若原来 Q 的状态是 0，则 Q 保持为 0，如果原来是 1，则保持为 1。Q 和 \bar{Q} 一直保持相反的状态。

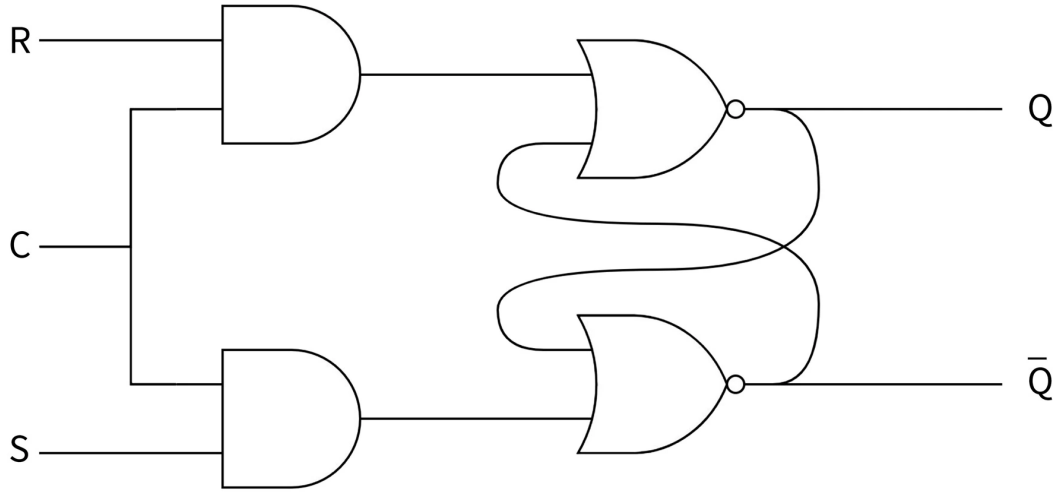
R 和 S 都为 1 是非法状态，这并不是说这种电路真的不允许 R 和 S 都为 1，而是当它们都为 1 时，Q 和 \bar{Q} 都为 0，这不符合 Q 和 \bar{Q} 保持相反状态的假设，从而失去了存储信息的功能。所以我们对于 S 和 R 都为 1 的情况就不再讨论了。

除了或非门组成的锁存器，还可以使用与门和或门构成锁存器，也可以使用与非门来构成锁存器。锁存器的特点是输入一旦发生变化，输出端立即就能反应出这种变化。我们知道，RS 锁存器的输入信号可能会不稳定，那当输入端的毛刺影响输出端时，该怎么办呢？

D 锁存器

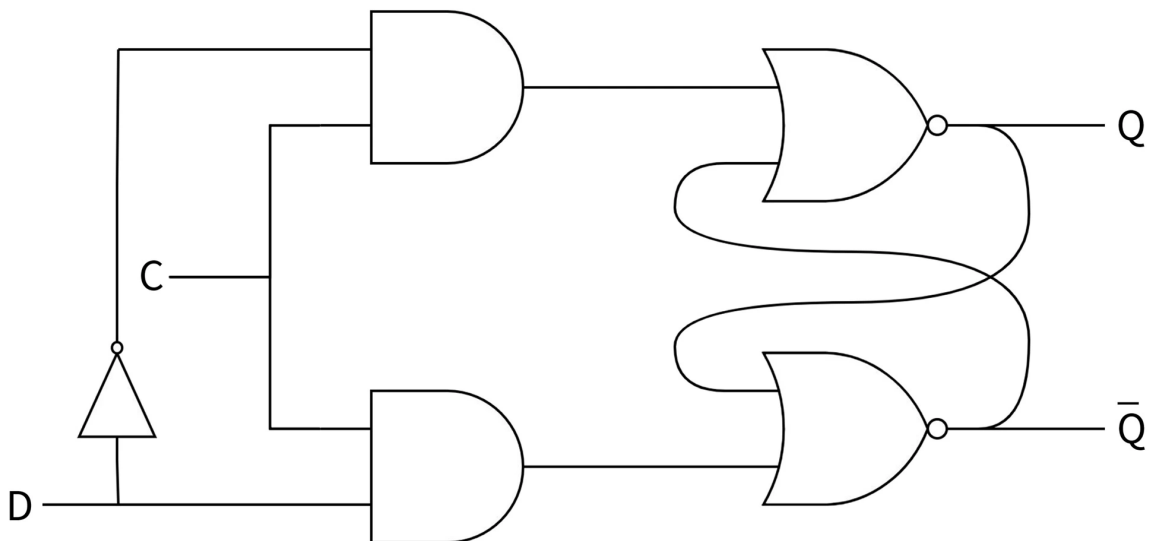
为了让存储电路可以保持记忆的能力，我们可以考虑为这种电路引入保持位，当保持位为 1 时，数据可以被存储进电路，当存储完成以后，保持位变为 0，输入就不能再影响输出了。

如下图所示，通过增加两个与门来实现保持位 C。当保持位为 1 时，RS 锁存器的功能与不加保持位时完全相同，当保持位为 0 时，则 R 端和 S 端不论取什么值，都不再起作用了。



极客时间

我们再对上图进行更深一步的分析，R 和 S 同时为 0 的情况，可以使用保持位为 0 来代替。R 和 S 同时为 1 的情况是非法情况，所以我们可以使用一个非门，把 R 和 S 合并成一个信号 D，这样的话，电路的输入端就可以进一步化简，如下图所示：



极客时间

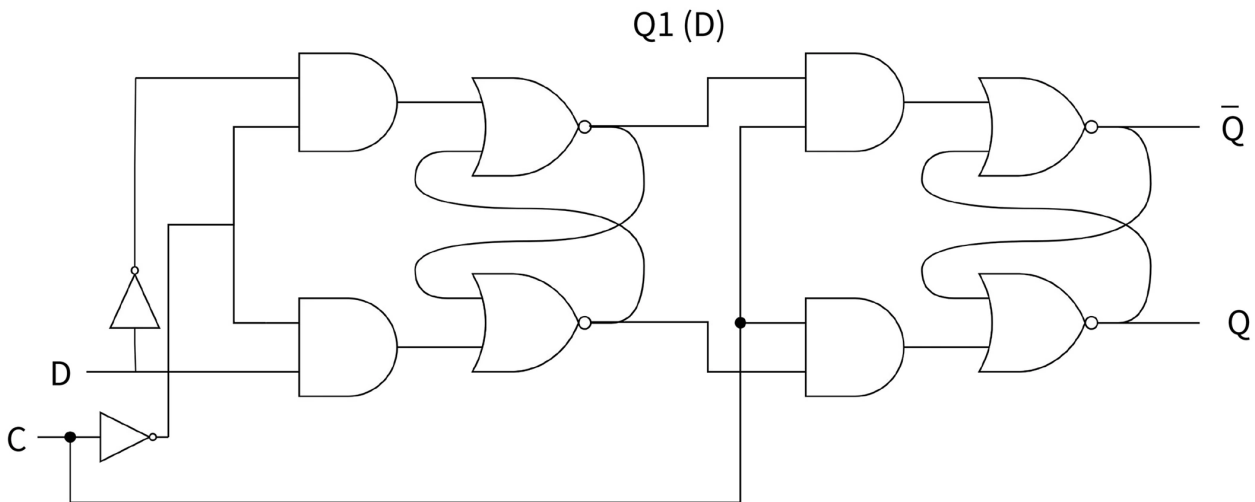
上面电路的保持位为 1 时，输出 Q 可以反应输入 D 的变化，当保持位为 0 时，输出 Q 保持原来的值不变。这种电路被称为电平触发的 D 型锁存器。其中的 D 代表数据 Data。

我们知道信号在传输的过程中容易产生毛刺等不稳定的现象，所以在保持位为 1 的期间，如果输入信号还是不能稳定的话，那么输出将随着输入的改变而发生相应的变化。这种情况下，锁存器的状态就难以稳定。我们把在保持位为 1 期间，锁存器发生多次翻转的情况称为空翻。

在实际应用中，人们还是希望存储电路有良好的稳定性，从这个角度上看，电平触发的锁存器的抗干扰能力相对较差。

D 触发器

为了解决锁存器稳定性不好的问题，人们在实现存储电路时，常用的方法是使用两个 D 锁存器来实现一个触发器。如下图所示：



D 触发器有 D(Data) 和 C(Clock) 两个输入信号，Q 和 Q 反两个输出信号。D 触发器由两个 D 锁存器组成。当时钟信号 C 为 0 时，前端锁存器输出信号 D 的值，后端锁存器保持之前的数据。当 C 为 1 时，前端锁存器保持之前的数据，后端锁存器将前端锁存器保持的数据直接通过 Q 输出。

可以看出来，对于 D 触发器而言，只有在 C 从 0 变成 1 的瞬间，输入数据 D 的值才会反应到输出 Q 上。所以这种触发器就被称为上升边沿触发的触发器。

虽然触发器可以在一个 CPU 时钟周期内完成读写，是最快的一类存储单元，但是它的造价十分高昂，用于存储一个比特的触发器就要使用几十个晶体管。所以它往往用来实现 CPU 内部的寄存器，一个寄存器的位宽不过 64 比特，它的电路面积的消耗还是可以接受的。

但如果我们需要大规模，比如几十 K，甚至几 M 的存储时，就不得不考虑更加节约的方案了，这就要用到大容量的存储器。接下来，我们再来看一下大容量存储器的工作原理。

大容量存储器

普通的大容量存储器大致可以分为两类，**一类是只能读，不能写，这种存储器称为只读存储器**（Read Only Memory，ROM），在生产的时候，厂家将内容写入存储器之后，用户就只能从其中读取数据，不能再更改其中的内容了。

另一类是可以支持读和写操作的存储器，它们被称为随机访问存储器(Random Access Memory, RAM)，随机访问这个名字是与顺序访问相对应的。早期的存储设备，例如纸带，磁带等等，只能顺序访问，如果想要跳到某一个特定位置进行播放，只能将磁带快进到我们所需要的那个位置，然后再顺序地访问该位置的数据。而支持随机访问的存储器则可以用同样的速度访问存储器内任何位置上的数据。

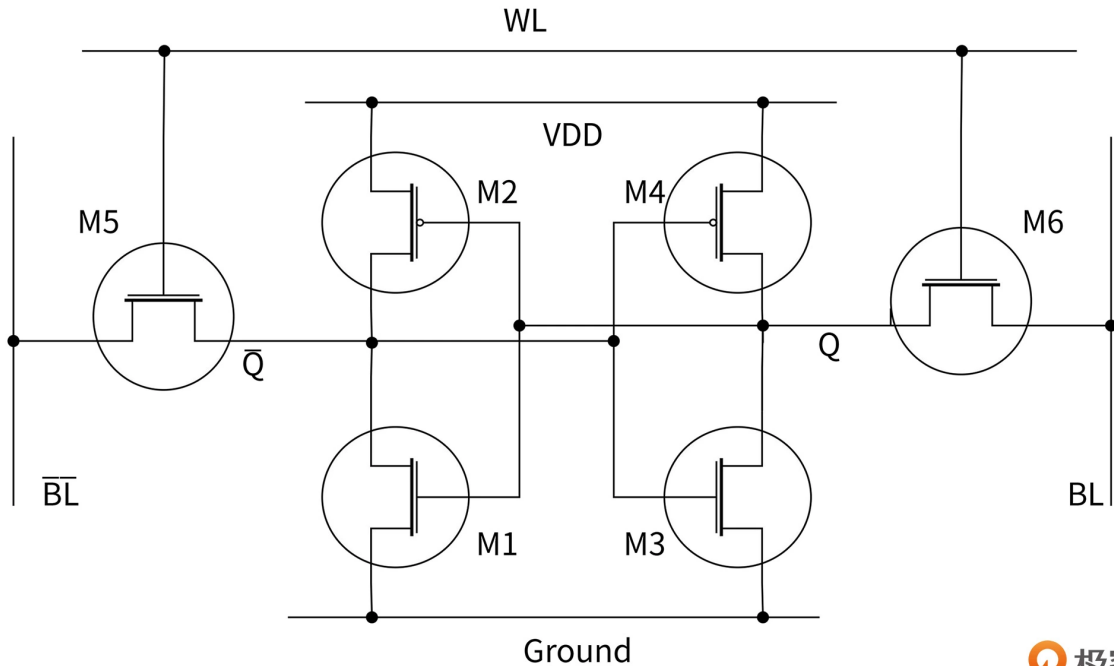
由于 ROM 大多用在一次烧制，永远不需要更新的地方，这往往不是软件程序员所关心的话题。所以，我们这个专栏课里所讲的内存，都是指可读写的 RAM，所以我们下面就来详细地分析 RAM 的工作原理。

RAM 大体上分为两类：静态随机访问存储器（Static RAM, SRAM）和动态随机访问存储器（Dynamic RAM, DRAM）。SRAM 的特点是速度快、造价高，往往用来制作高速缓存，集成在 CPU 里，它的容量一般不会超过几兆字节。DRAM 的特点是速度慢、造价低，计算机中的主存就是 DRAM，单根内存条就可以有十几，甚至几十 G 的容量。

接下来，我们就分别考察 SRAM 和 DRAM 的电路结构，来搞清楚它们的原理。

SRAM 的电路结构

为了节约电路面积，SRAM 采用了 6 管式的存储电路。如图所示：



 极客时间

在上图中，M1 和 M3 两个 MOS 管，是 N 沟道场效应管，在高电压时导通；而 M2 和 M4 这两个 MOS，则是 P 沟道场效应管，在低电压时导通。本质上，M1 和 M2 一起组成了一个非门，M3 和 M4 一起组成了另一个非门，这两个非门的输出互为对方的输入，这样，两个非门就组成了一种可以存储比特值的电路。

访问 SRAM 时，字线 (Word Line, WL) 加高电平，使得每个基本单元的两个控制开关：晶体管 M5 与 M6 开通，把存储单元与位线 (Bit Line, BL) 连通。位线用于读取或写入基本单元的保存状态。

我们假定储存的内容为 1，即在 Q 处的电平为高。读取周期开始时，两条位线预先设成高电平，随后字线 WL 变成高电平，使得两个访问控制晶体管 M5 与 M6 导通。Q 的高电平使得晶体管 M1 导通，而 Q 反与 BL 反的预充值不同，使得 BL 反经由 M1 与 M5 放电而变成逻辑 0。在位线 BL 一侧，Q 反的低电平使得 M4 导通，再加上 M6 通路，位线就连接到 VDD 的高电压。

如果储存的内容为 0，相反的电路状态将会使 BL 反为 1，而 BL 为 0。这时，只需要 BL 与 BL 反有一个很小的电位差，读取的放大电路就会辨识出哪条位线是 1，哪条是 0。也就是说，当敏感度越高时，读取的速度就越快。

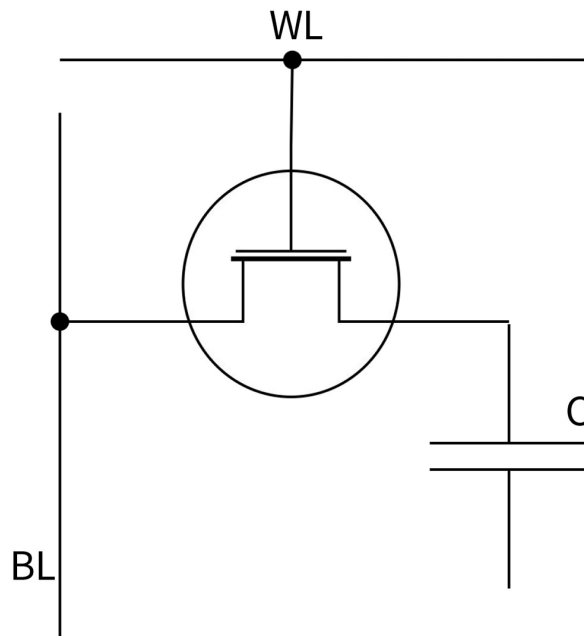
在写入周期开始时，把要写入的状态加载到位线。如果要写入 0，则设置 BL 反为 1 且 BL 为 0。随后字线 WL 加载为高电平，位线的状态被加载进 SRAM 的基本单元。

简单说，SRAM 存储单元的特点是使用 6 个晶体管来实现。其中两个 P 型 MOS 管和两个 N 型 MOS 管组成两个反相器用于存储信息。还有两个用于控制存储单元是否选通。6 管 SRAM 的结构比触发器简单，速度也比较快。

讲到这里，你就清楚 SRAM 的电路结构了，下面我们来看看 DRAM 的电路结构。

DRAM 的电路结构

DRAM 相比起 SRAM，它的电路结构更简单，它是由一个 CMOS 开关和一个电容组成的，如图所示：



 极客时间

当 WL 为高电压时，开关打开，存储单元选通，如果此时数据线 BL 为高电压，则向电容中充电，这就是 DRAM 存储单元中写 1，如果数据线为低电压，电容放电，这就是往存储单元中写 0。

在读取的时候，同样要打开开关，如果电容放电，那么就表示这个单元里原来存的值是 1，如果电容不放电，则表示原来的值是 0。可见，DRAM 的读取是破坏性的，它会使得原来为 1 的存储单元变成 0。

为了解决这个问题，在读取 DRAM 的数据的时候，人们要想办法给原来为 1 的比特再进行一次充电。另外，电容本身也会缓慢漏电，所以存储器也要每隔一段时间就为电容补充电荷。这也是 Dynamic 这个名称的由来。

DRAM 的存储单元使用一个 MOS 管和一个电容实现，其特点是电路相比 SRAM 更加简单，也就更容易大规模集成，成本也更低，但是它的读取速度比较慢。

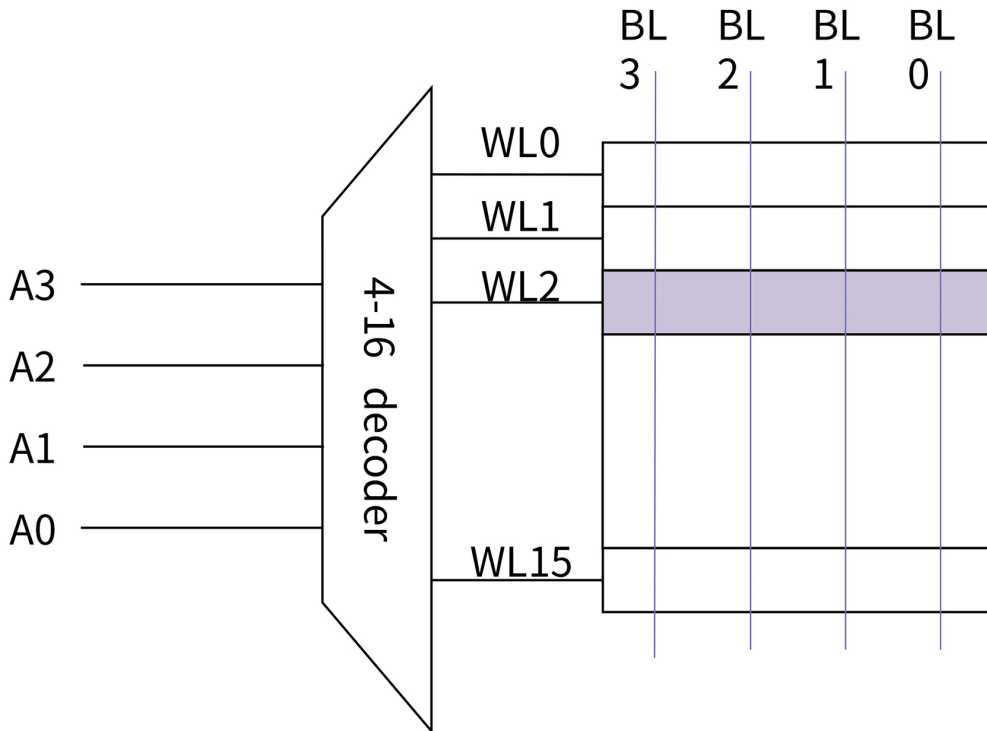
到这里，我们就把用于存储一个比特的常见电路介绍完了。但存储器的容量是十分巨大的，这又是怎么做到的呢？换句话说，一个字节是由 8 个比特组成，而一根容量为 4G 的内存条，它包含的比特是 $4 \times 2^{32} \times 8 = 2^{37}$ 个。存储器是如何知道 CPU 要读写哪个比特的呢？这就要进一步了解存储器是如何对字节和比特进行地址编码的。

存储器的地址编码

存储器在存储数据时，一定要区分数据是要存在哪个地方的，也就是说，我们要想办法对存储器里的各个单元进行编码，并且能将地址总线的数据转换成相应存储单元的使能信号，然后进一步区分控制总线的读写信号。**如果是读操作，存储控制器就要将存储单元内的值读入数据总线，如果是写操作，控制器就应该把数据总线的数据写入存储单元。**

我们假设存储器是按字节进行编码，一次读写最少也要操作一个字节，数据总线的宽度是 8 位。这种存储器在制造时，要将每一个字节的第 n 比特的位线都连接到数据总线的第 n 位，也就是 8 个比特的位线分别与数据总线的 8 位相连。而存储控制器则根据地址总线的值将对应的字节的字线设成高电平，以选通目标字节。

它的简单示意图如下所示：



从这个角度，我们也就能理解，存储单元的字线和位线为什么要这样命名了。这是因为字线可以对存储单元所在的机器字进行选通，而位线是真的接在了数据总线的某一位上。

现在的核心问题在于，**控制器如何对地址信号进行转换，使得目标字节的字线变成高电压？**

其实，这个工作需要译码器来完成。我们以 4 位地址总线来举例。首先，4 位地址总线可以编码的地址是 0x0~0xf，共计 16 个。如上图所示。

上图中的电路有 4 个输入，16 个输出，16 个输出分别对应了存储器的 16 个存储单元的字线。当地址总线上的数据是 0000 时，我们希望第 0 个字节的字线是高电压，当地址总线上的数据是 0001 时，第 1 个字节的字线就是高电压，依次类推。我们会发现，地址总线上的数据刚好就是存储单元地址编码的二进制数。这里也就说明了为什么地址总线的宽度会影响存储编码范围了。

我们可以把这个电路的真值表总结出来：

A0	A1	A2	A3	选通输出端
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
...
1	1	1	1	Y15

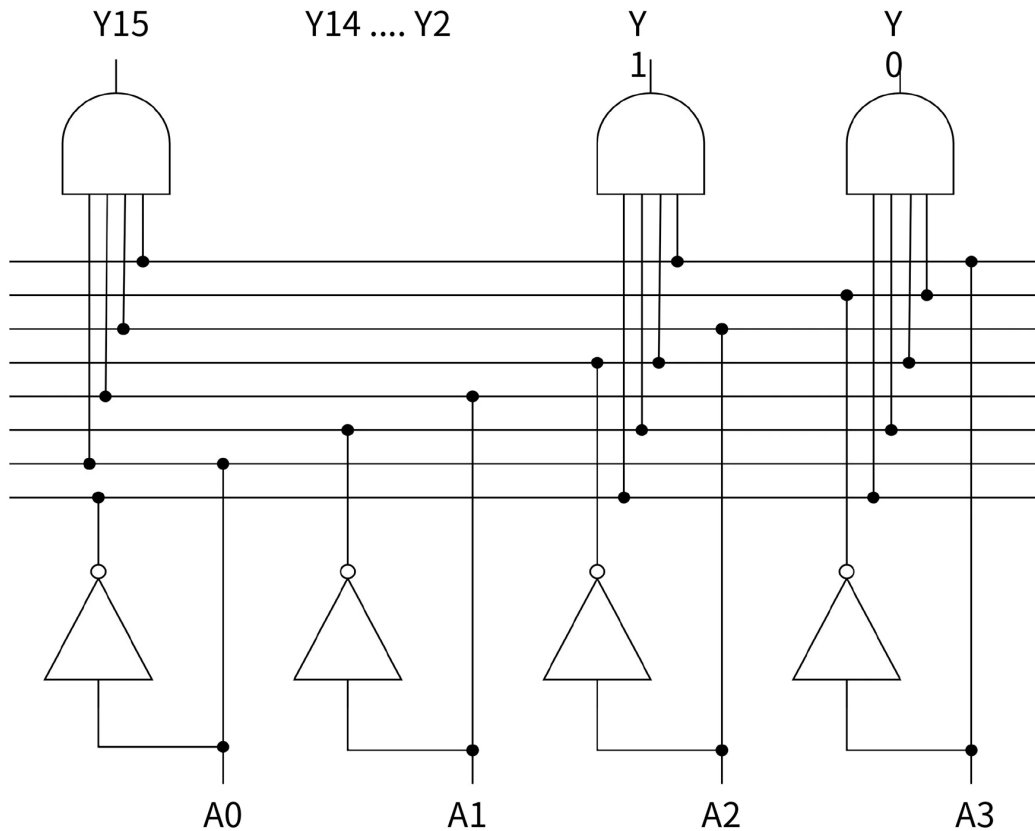


接下来，我们由真值表写出 16 个输出端的表达式：

$$Y0 = (\sim A0) \& (\sim A1) \& (\sim A2) \& (\sim A3), Y2 = (\sim A0) \& (\sim A1) \& (\sim A2) \& A3 \dots, Y15 = A0 \& A1 \& A2 \& A3$$

这个表达式再翻译成组合逻辑电路是简单的，只需要将 A0、A1、A2、A3 都取反，然后将这个 4 个信号使用与门连接，它的输出就是 Y0 了。同样地，把 A0、A1、A2 取反，再与 A3 一起使用与门连接，这个输出就是 Y1。

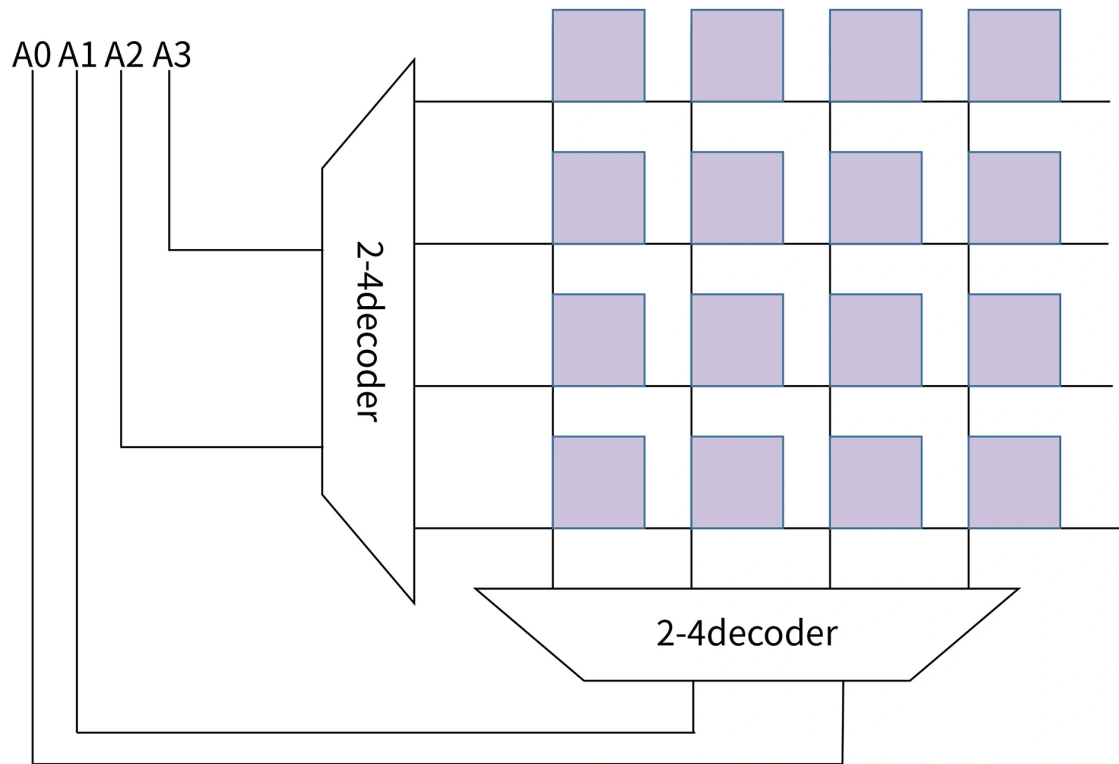
作为示意，我画出了 Y0、Y1 和 Y15 的电路图，其余的作为练习，希望你可以模仿我的例子自己补充。



这种电路就是译码器，我们把 4 输入，16 输出的译码器称为 4-16 线译码器。使用译码器就可以解决字线选通的问题。

但这样做会带来一个新的问题，那就是随着存储器地址范围变大，译码器的输出端口会变得非常多，比如 32 位地址总线会有 4G 个输出端。为了解决这个问题，人们把地址平分成高低两个部分，并且把存储单元做成矩阵式排列，使用高地址决定存储单元所在的行，使用低地址决定它所在的列。

为了方便，我们仍然以 16 个字节为例。它排成存储矩阵以后是这样子的：



我们将 4 位地址分成两组，分别送入两个译码器。高地址送入行译码器，用于选通行线，低地址送入列译码器，用于选通列线。只有行线和列线同时选通的存储单元才是有效的。在这种设计里，32 位地址总线的字线由原来的 4G 下降为 $64K+64K=128K$ 。这样就大大减少了字线的数量。

使行线和列线同时生效也有两种做法，一种是在每个存储单元引入一个与门，这需要为每个字节多增加两个 MOS 管，会降低芯片的集成度。

另一种是在存储芯片里再增加一行的缓存，读取时分两步进行，第一步先使行线生效，将目标存储单元所在行，整行读入到缓存中；第二步再使列线生效，从缓存中读取目标单元的值。这种做法需要两次读取，性能会差一些。

存储器的编码主要依赖译码器这种电路结构。它的特点是每一个输出端与一个存储单元的字线相连。译码器将地址总线发送过来的地址作为输入，然后只有一个输出成为高电平，这样就选通了一个存储单元，从而实现了对多个存储单元的地址编码。

在不同场景根据相应的约束，选择可能就会不一样。从上面的分析中，我们可以看到，没有哪一种存储电路是十全十美的，每一种都有自己的优点和缺点，如何把它们用在合适的场景，从而以系统化的方法做到扬长避短，才是存储系统设计的首要目标。

只读存储器简介

虽然我们说只读存储器往往被用在固件中或者闪存中，断电后，存储的数据也不会消失。它不是内存，但作为现在存储体系的必要组成，对它有所了解还是必要的。所以，这里我就对 ROM 做一个简单的介绍。

存储的本质就是把 0 和 1 存到特定的地方，然后在需要恢复的时候，再把数据恢复出来。RAM 的特点是存储数据时需要加电，一旦断电，则存储的数据消失。而 ROM 中的信息断电也不会消失，它非常像纸笔这种记录信息的手段，不易更改，保存时间长。

早期的 ROM，是基于熔丝制作的，写入操作是通过烧断熔丝进行的，这种类型的 ROM 只能在初始化时写入一次，之后就不能再更改了。当前，这种类型的 ROM 已经不存在了。

人们在 MOS 管上增加浮置栅，这个特殊的浮栅中可以存储电荷。当有电荷时代表 1，没有电荷时代表 0。在一定的条件下可以为浮栅充电，也可以为它放电。所以，这就是可编程 ROM (Programmable ROM, PROM)。

早期的 PROM，是可以使用紫外线进行擦除的，这就是紫外线可擦除可编程 ROM (Ultraviolet Erasable Programmable ROM, UV-EPROM)。使用光进行擦除，有一个缺点就是擦除速度比较慢，而且可擦除次数有限。

为了克服上面的两个缺点，人们又发明了电可擦除 ROM (Electronic Erasable ROM, EEPROM)。这就是当今使用得最广泛的一种存储器件，它有个专门的名字叫做闪存。

到这里，几种存储器件是如何存储一个比特的，这个问题，我们就已经搞清楚了。

总结

这节课，我们分别学习了 CPU 中寄存器的存储单元，SRAM 和 DRAM 的存储单元，以及 ROM 的存储单元的基本原理。从中，我们能了解到，CPU 中的寄存器使用触发器存储一个比特，读写速度最快，但所占电路面积最大。

在这节课开头，我带你从最简单的锁存器电路开始，先改进成电平触发的 D 锁存器，然后通过两个 D 锁存器搭建了上升沿触发的 D 触发器。D 触发器配合时钟工作，提供稳定的，可控的，快速的存储能力。

D 触发器的优点很多，但是缺点也很明显，那就是电路比较复杂、成本高、难以高度集成。所以我们使用 6 管存储单元来替换 D 触发器，6 管存储单元所组成的存储器就是静态随机读写存储器（SRAM）。

SRAM 的读写速度比较快，造价也更低，但集成度仍然不如单 MOS 管的动态随机存储器（DRAM）。DRAM 的原理是使用电容的充放电状态来代表 0 和 1，电容的特点是读数据时，它的状态会被破坏，所以需要以一定的频率对电容进行充电刷新。

最后，我们简单了解了 ROM 和闪存的发展历史和分类，从中也可以看到 ROM 这个名字其实已经名不副实了。我们常把闪存归到外存储器分类中，所以我们这个专栏就没有使用大量的篇幅对它进行介绍，但是闪存和当前非常流行的非易失性内存是存储系统的重要部分。如果你对这部分内容比较感兴趣，可以继续阅读 [《大话存储》](#) 等资料进行学习。

思考题

我们在设计各种存储器件的时候，要不断地平衡它们的优缺点。请你思考一下，在存储电路设计中人们要平衡哪些因素呢？举个例子，既然 D 触发器的速度最快，那我使用 D 触发器来制作主存可以吗？为什么？欢迎你在留言区分享你的想法和收获，我在留言区等你。

吊打面试官

- 你对存储器件有哪些了解。


存储器大体上可以分为只读存储器ROM和随机存储器RAM两大类。

早期的只读存储器往往是只能写一次，不能更改的。随着技术的进步，逐渐出现了紫外线可擦除ROM和电可擦除ROM，从而使得ROM可以多次写入。现代的闪存就是EEPROM，它是一种电可擦除的可编程ROM。

ROM断电后信息不会丢失，RAM断电后存储的信息就会消失。

RAM可以进一步分为静态存储器SRAM和动态存储器DRAM。SRAM的存储单元多采用6管实现，DRAM则采用一个开关加一个电容的方式实现。DRAM的存储单元占用面积小，更容易集成，所以容量也更大，但是DRAM的电容会缓慢漏电，因此需要定期刷新。SRAM相比DRAM有更好的读写速度，多用来制作CPU的高速缓存。

高频面试真题

 极客时间

好啦，这节课到这就结束啦。欢迎你把这节课分享给更多对计算机内存感兴趣的朋友。我是海纳，我们下节课再见！

分享给需要的人，Ta订阅后你可得 **20** 元现金奖励

 生成海报并分享

 赞 0  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 12 | 内存虚拟化：云原生时代的奠基者

下一篇 14 | CPU Cache：访存速度是如何大幅提升的？

训练营推荐

Java 学习包免费领 NEW

面试题答案均由大厂工程师整理

阿里、美团等
大厂真题18 大知识点
专项练习大厂面试
流程解析可复用的
面试方法面试前
要做的准备

精选留言 (8)

写留言

**费城的二鹏**

2021-11-24

老师的课程进入硬件篇章啦，感觉这门课程太划算了，老师讲的内容深入浅出，非常透彻。虽然底层软件知识与硬件知识在工作中不能直接用到，但是它加深了我对计算机的理解，对于我的编程理解有了很大的提升。

谢谢老师与小编，带来如此优秀的内容！

展开 ∨



👍 2

**Roy Liang**

2021-12-02

老师会讲持久内存吗？未来可能派上用场

展开 ∨

作者回复：不会。在这个文章里简单提了一下Flash和非易失性内存。非易失性内存确实有机会改变很多软件运行形态。可能是未来的一个热点，值得关注和学习。



通过快速的成长



达达快进的蜗牛
2021-12-01

不懂硬件的表示，看了这门课等于看了这门课

展开 ∨

作者回复: 哈哈。没关系的。等有需要的时候再回来看看吧。其实也不必被这个文章里的图吓到，它无非就是与门，或门，非门的组合而已。规则非常简单。



AIMD

2021-11-26

老师，最开始的驱动程序是怎么安装存储电路中？

展开 ∨

作者回复: 这个问题太大了，三言两语说不清。建议你阅读一下《计算机组成原理》之类的书，有了一个比较系统的认识以后再思考。



相逢是缘

2021-11-25

老师，想问一下在课程中没有发现D触发器（制作寄存器）从电路原理上为什么比SRAM读取速度快呢？

作者回复: D触发器的特点是在一个时钟周期内必然可以完成读写。但是SRAM的读写原理不和时钟相关。它的速度与读取时的BL电压探测的速度有关，而这个速度现在最快情况下要1~2个时钟周期。这一点我文中没有刻意强调，这是因为随着技术的进步，未来SRAM的速度能赶上寄存器的速度也不一定。所以这里不能简单地说，SRAM 6 管单元就一定慢。现在已经有一些芯片寄存器（主要是那种寄存器特别多的芯片）是用SRAM来实现的了。所以未来是怎么样的，我们还要再观察。



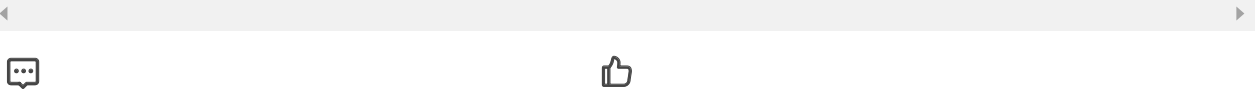
shenglin

2021-11-24

D触发器电路复杂，难以高度集成，是不是说制作相同容量的主存，用D触发器制作的内存条体积会更大？

展开 ∨

作者回复: 会大太多，而且功耗，散热等等也会指数级增加。不会有人拿触发器做内存条的，技术上就做不到。



独孤

2021-11-24

问2个问题，电路可以存储比特，但软件里面的最小处理单位大都是字节，如何做到读或写一个字节（8个比特，而不是小于8）这个原子性保证的？

软件最后都会转换为比特，电路也可以存储比特，问题是软件中的比特怎么转换为硬件中的信号的？或者说软件最终怎么控制硬件的？

展开 ∨



费城的二鹏

2021-11-24

思考题

要平衡性能与价格。如果价格可以接受，完全可以使用 D触发器做主存，我猜测有壕这么做 😊

展开 ∨

作者回复: 哈哈，说得很有趣。但壕也不能这么做，因为电路面积不光关系到成本，还有功耗和散热。最关键的是，学习完下节课以后你就明白了，没必要这么干。

