

18-建立数据通路（中）：指令more运算=CPU

上一讲，我们看到，要能够实现一个完整的CPU功能，除了加法器这样的电路之外，我们还需要实现其他功能的电路。其中有一些电路，和我们实现过的加法器一样，只需要给定输入，就能得到固定的输出。这样的电路，我们称之为**组合逻辑电路**（Combinational Logic Circuit）。

但是，光有组合逻辑电路是不够的。你可以想一下，如果只有组合逻辑电路，我们的CPU会是什么样的？电路输入是确定的，对应的输出自然也就确定了。那么，我们要进行不同的计算，就要去手动拨动各种开关，来改变电路的开闭状态。这样的计算机，不像我们现在每天用的功能强大的电子计算机，反倒更像古老的计算尺或者机械计算机，干不了太复杂的工作，只能协助我们完成一些计算工作。

这样，我们就需要引入第二类的电路，也就是**时序逻辑电路**（Sequential Logic Circuit）。时序逻辑电路可以帮我们解决这样几个问题。

第一个就是**自动运行**的问题。时序电路接通之后可以不停地开启和关闭开关，进入一个自动运行的状态。这个使得我们上一讲说的，控制器不停地让PC寄存器自增读取下一条指令成为可能。

第二个是**存储**的问题。通过时序电路实现的触发器，能把计算结果存储在特定的电路里面，而不是像组合逻辑电路那样，一旦输入有任何改变，对应的输出也会改变。

第三个本质上解决了各个功能按照**时序协调**的问题。无论是程序实现的软件指令，还是到硬件层面，各种指令的操作都有先后的顺序要求。时序电路使得不同的事件按照时间顺序发生。

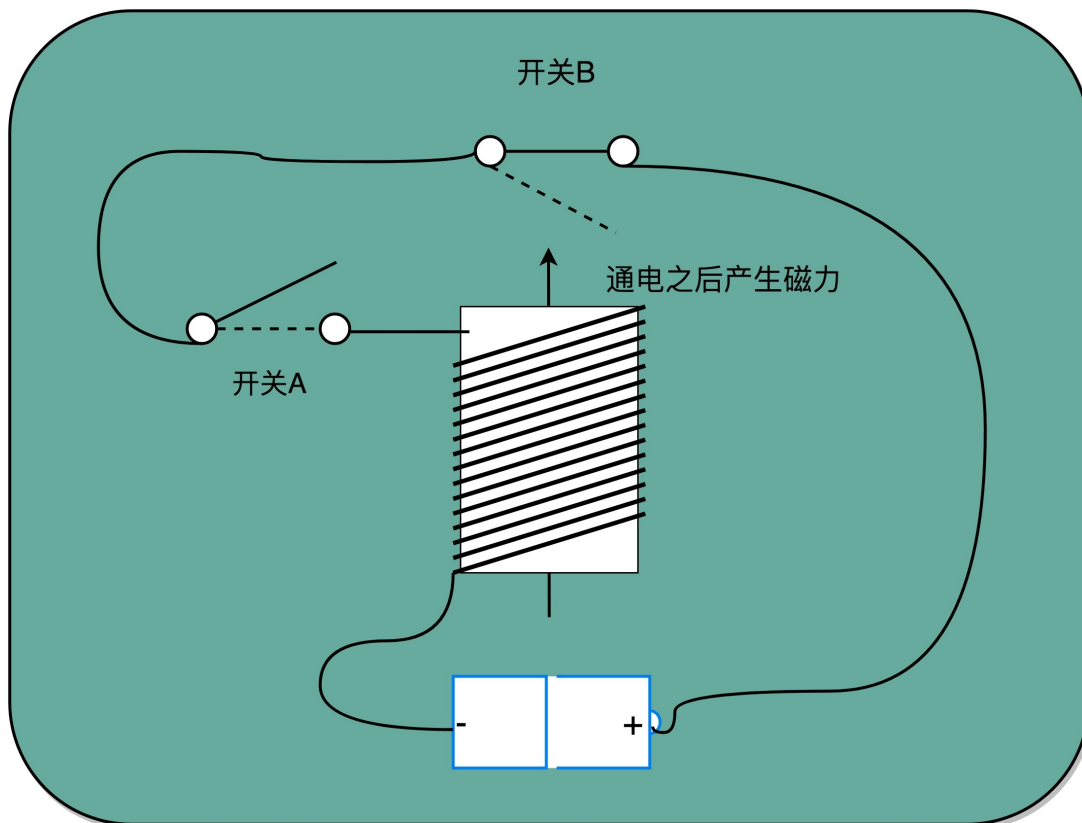
时钟信号的硬件实现

想要实现时序逻辑电路，第一步我们需要的就是一个**时钟**。我在[第3讲](#)说过，CPU的主频是由一个晶体振荡器来实现的，而这个晶体振荡器生成的电路信号，就是我们的时钟信号。

实现这样一个电路，和我们之前讲的，通过电的磁效应产生开关信号的方法是一样的。只不过，这里的磁性开关，打开的不再是后续的线路，而是当前的线路。

在下面这张图里你可以看到，我们在原先一般只放一个开关的信号输入端，放上了两个开关。一个开关A，一开始是断开的，由我们手工控制；另外一个开关B，一开始是合上的，磁性线圈对准一开始就合上的开关B。

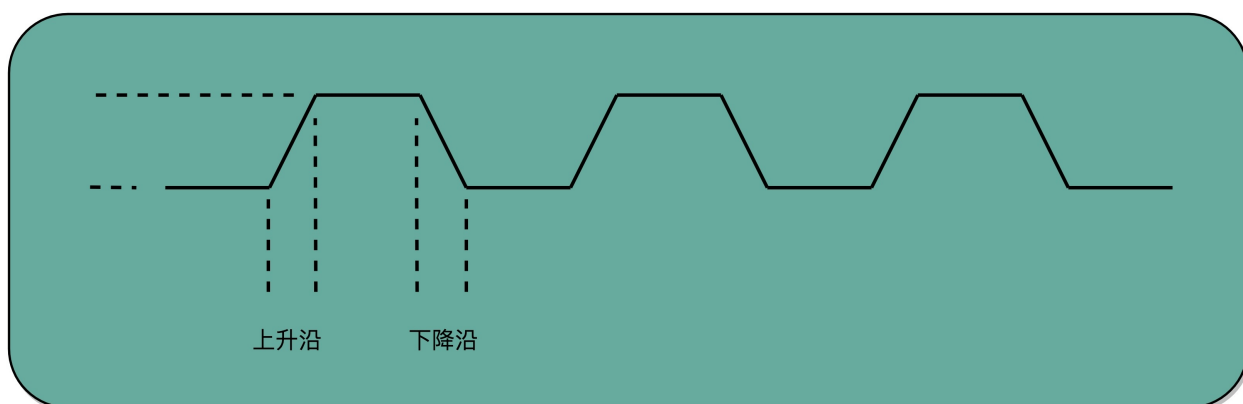
于是，一旦我们合上开关A，磁性线圈就会通电，产生磁性，开关B就会从合上变成断开。一旦这个开关断开了，电路就中断了，磁性线圈就失去了磁性。于是，开关B又会弹回到合上的状态。这样一来，电路接通，线圈又有了磁性。我们的电路就会来回不断地在开启、关闭这两个状态中切换。



开关A闭合（也就是相当于接通电路之后），开关B就会不停地在开和关之间切换，生成对应的时钟信号

这个不断切换的过程，对于下游电路来说，就是不断地产生新的0和1这样的信号。如果你在下游的电路接上一个灯泡，就会发现这个灯泡在亮和暗之间不停切换。这个按照固定的周期不断在0和1之间切换的信号，就是我们的**时钟信号**（Clock Signal）。

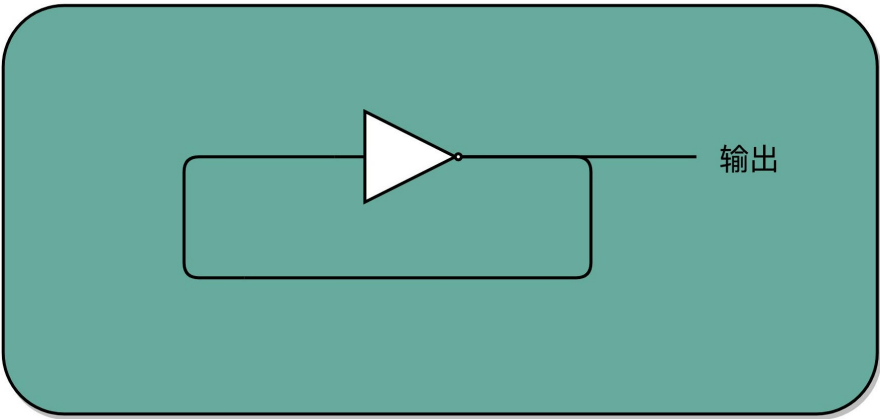
一般这样产生的时钟信号，就像你在各种教科书图例中看到的一样，是一个振荡产生的0、1信号。



时钟信号示意图

这种电路，其实就相当于把电路的输出信号作为输入信号，再回到当前电路。这样的电路构造方式呢，我们叫作**反馈电路**（Feedback Circuit）。

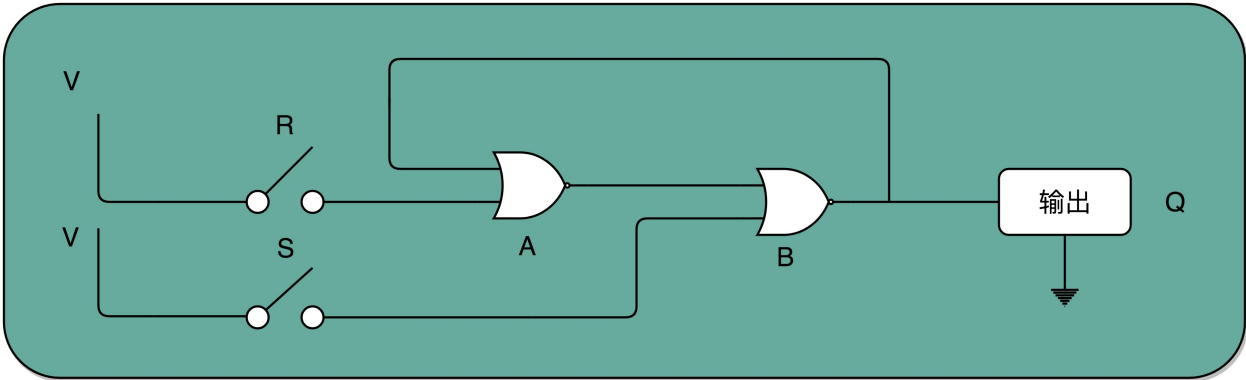
接下来，我们还会看到更多的反馈电路。上面这个反馈电路一般可以用下面这个示意图来表示，其实就是一个输出结果接回输入的反相器（Inverter），也就是我们之前讲过的非门。



通过一个反相器实现时钟信号

通过D触发器实现存储功能

有了时钟信号，我们的系统里就有了一个像“自动门”一样的开关。利用这个开关和相同的反馈电路，我们就可以构造出一个有“记忆”功能的电路。这个有记忆功能的电路，可以实现在CPU中用来存储计算结果的寄存器，也可以用来实现计算机五大组成部分之一的存储器。



我们先来看下面这个RS触发器电路。这个电路由两个或非门电路组成。我在图里面，把它标成了A和B。

NOR	0	1
0	1	0
1	0	0

或非门的真值表

- 1. 在这个电路一开始，输入开关都是关闭的，所以或非门（NOR）A的输入是0和0。对应到我列的这个真值表，输出就是1。而或非门B的输入是0和A的输出1，对应输出就是0。B的输出0反馈到A，和之前的输入

没有变化，A的输出仍然是1。而整个电路的**输出Q**，也就是0。

2. 当我们把A前面的开关R合上的时候，A的输入变成了1和0，输出就变成了0，对应B的输入变成0和0，输出就变成了1。B的输出1反馈给到了A，A的输入变成了1和1，输出仍然是0。所以把A的开关合上之后，电路仍然是稳定的，不会像晶振那样振荡，但是整个电路的**输出Q**变成了1。
3. 这个时候，如果我们再把A前面的开关R打开，A的输入变成和1和0，输出还是0，对应的B的输入没有变化，输出也还是1。B的输出1反馈给到了A，A的输入变成了1和0，输出仍然是0。这个时候，电路仍然稳定。**开关R和S的状态和上面的第一步是一样的，但是最终的输出Q仍然是1**，和第1步里Q状态是相反的。我们的输入和刚才第二步的开关状态不一样，但是输出结果仍然保留在了第2步时的输出没有发生变化。
4. 这个时候，只有我们再去关闭下面的开关S，才可以看到，这个时候，B有一个输入必然是1，所以B的输出必然是0，也就是电路的最终**输出Q**必然是0。

这样一个电路，我们称之为触发器（Flip-Flop）。接通开关R，输出变为1，即使断开开关，输出还是1不变。接通开关S，输出变为0，即使断开开关，输出也还是0。也就是，**当两个开关都断开的时候，最终的输出结果，取决于之前动作的输出结果，这个也就是我们说的记忆功能。**

这里的这个电路是最简单的RS触发器，也就是所谓的复位置位触发器（Reset-Set Flip Flop）。对应的输出结果的真值表，你可以看下面这个表格。可以看到，当两个开关都是0的时候，对应的输出不是1或者0，而是和Q的上一个状态一致。

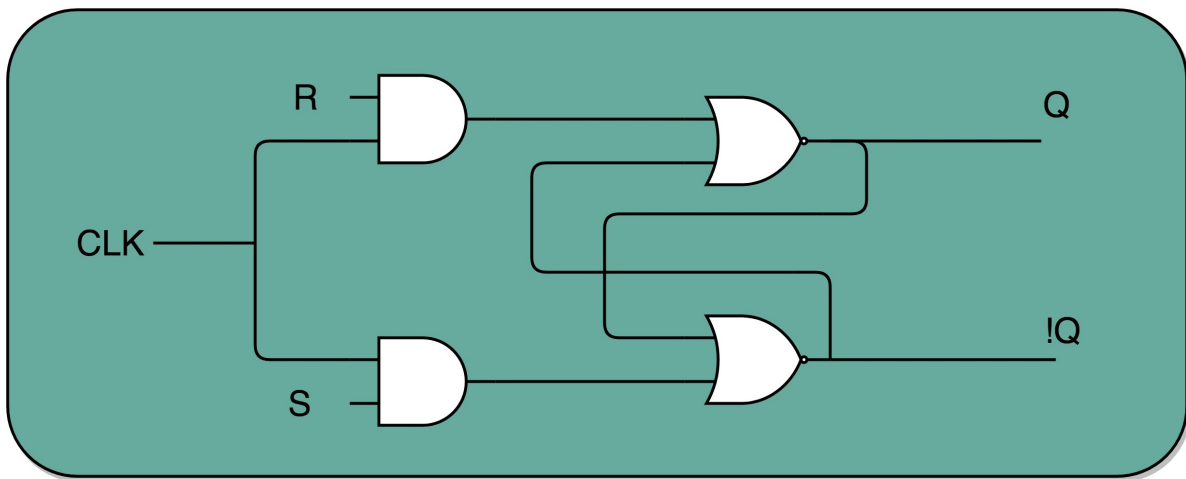
输入		输出
S	R	Q
1	0	1
0	1	0
0	0	Q
1	1	NA

再往这个电路里加两个与门和一个小小的时钟信号，我们就可以实现一个利用时钟信号来操作一个电路了。这个电路可以帮我们实现什么时候可以往Q里写入数据。

我们看看下面这个电路，这个在我们的上面的R-S触发器基础之上，在R和S开关之后，加入了两个与门，同时给这两个与门加入了一个**时钟信号CLK**作为电路输入。

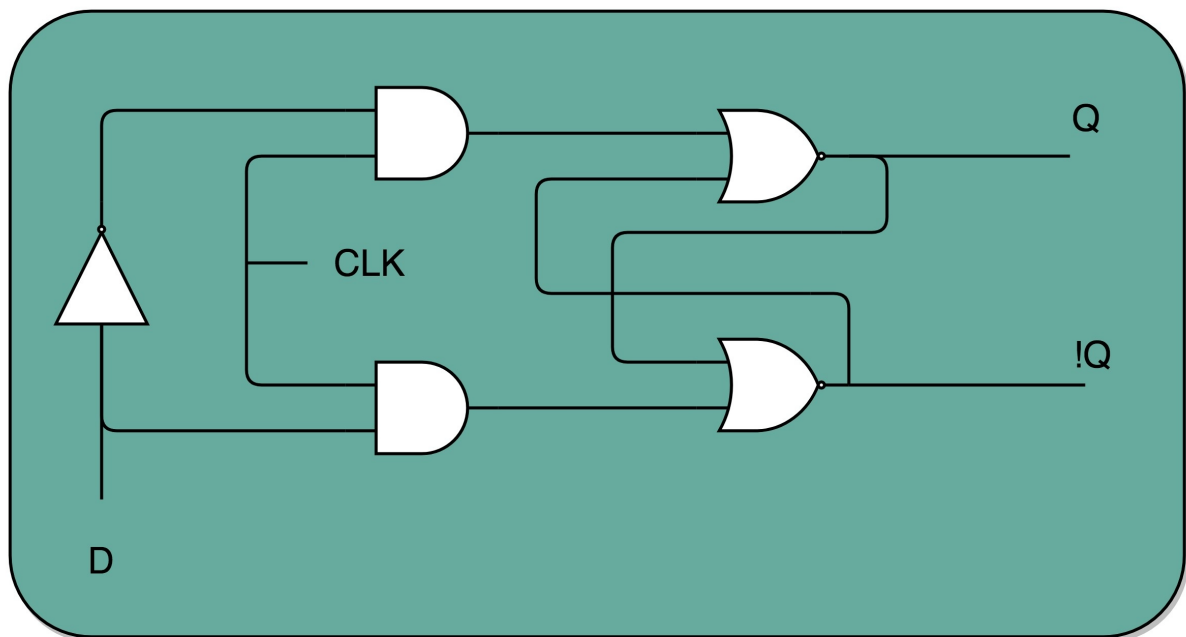
这样，当时钟信号CLK在低电平的时候，与门的输入里有一个0，两个实际的R和S后的与门的输出必然是0。也就是说，无论我们怎么按R和S的开关，根据R-S触发器的真值表，对应的Q的输出都不会发生变化。

只有当时钟信号CLK在高电平的时候，与门的一个输入是1，输出结果完全取决于R和S的开关。我们可以在这个时候，通过开关R和S，来决定对应Q的输出。



通过一个时钟信号，我们可以在特定的时间对输出的Q进行写入操作

如果这个时候，我们让R和S的开关，也用同一个反相器连起来，也就是通过同一个开关控制R和S。只要CLK信号是1，R和S就可以设置输出Q。而当CLK信号是0的时候，无论R和S怎么设置，输出信号Q是不变的。这样，这个电路就成了我们最常用的D型触发器。用来控制R和S这两个开关的信号呢，我们视作一个输入的数据信号D，也就是Data，这就是D型触发器的由来。



把R和S两个信号通过一个反相器合并，我们可以通过一个数据信号D进行Q的写入操作

一个D型触发器，只能控制1个比特的读写，但是如果我们同时拿出多个D型触发器并列在一起，并且把用同一个CLK信号控制作为所有D型触发器的开关，这就变成了一个N位的D型触发器，也就可以同时控制N位的读写。

CPU里面的寄存器可以直接通过D型触发器来构造。我们可以在D型触发器的基础上，加上更多的开关，来实现清0或者全部置为1这样的快捷操作。

总结延伸

好了，到了这里，我们可以顺一顺思路了。通过引入了时序电路，我们终于可以把数据“存储”下来了。我们通过反馈电路，创建了时钟信号，然后再利用这个时钟信号和门电路组合，实现了“状态记忆”的功能。

电路的输出信号不单单取决于当前的输入信号，还要取决于输出信号之前的状态。最常见的这个电路就是我们的D触发器，它也是我们实际在CPU内实现存储功能的寄存器的实现方式。

这也是现代计算机体系结构中的“冯·诺伊曼”机的一个关键，就是程序需要可以“存储”，而不是靠固定的线路连接或者手工拨动开关，来实现计算机的可存储和可编程的功能。

有了时钟信号和触发器之后，我们还差一个“自动”需求没有实现。我们的计算机还不能做到自动地不停地从内存里面读取指令去执行。这一部分，我们留在下一讲。下一讲里，我们看看怎么让程序自动运转起来。

推荐阅读

想要深入了解计算机里面的各种功能组件，是怎么通过电路来实现的，推荐你去阅读《编码：隐匿在计算机软硬件背后的语言》这本书的第14章和16章。

如果对于数字电路和数字逻辑特别感兴趣，想要彻底弄清楚数字电路、时序逻辑电路，也可以看一看计算机学科的一本专业的教科书《数字逻辑应用与设计》。

课后思考

现在我们的CPU主频非常高了，通常在几GHz了，但是实际上我们的晶振并不能提供这么高的频率，而是通过“外频+倍频”的方式来实现高频率的时钟信号。请你研究一下，倍频和分频的信号是通过什么样的电路实现的？

欢迎留言和我分享你的疑惑和见解，也欢迎你把今天的内容分享给你的朋友，和他一起学习和进步。




深入浅出计算机组成原理

带你掌握计算机体系全貌

徐文浩 bothub 创始人



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

精选留言：

- 鱼向北游 2019-06-05 09:49:07
分频用计数器就可以吧 从0开始计数模n 归零就来个输入 应该根据此原理用d触发器+计数器有优化的方法
倍频数学的方法叫时域信号傅里叶级数展开到频域 然后滤波器滤专门那个频滤
电路的方法我编不下去了。。。老师解答吧 [1赞]
- LDxy 2019-06-05 22:27:56
n分之一分频器可以使用n进制计数器实现，n进制计数器的进位输出端的频率即为输入时钟信号频率的n分之一。
n倍频器可以由锁相环加n分之一分频器实现。锁相环是一个反馈环路，这个环路里面有一个叫鉴相器的部件，外部输入信号f0会进入鉴相器，同时锁相环输出端的信号f1也会反馈到鉴相器的另一个输入端，鉴相器会比较f0与f1的相位差，输出一个变化的电压信号去控制锁相环的其他部件，从而调整输出信号f1的频率。总的来说，锁相环的基本功能就是使得输出端的信号f1与输入端信号f0的相位差维持恒定，所以名为「锁相环」。如上所述，如果f1直接反馈回到鉴相器，为了维持相位差恒定，会有输出信号f1=f0。如果将f1经过n分之一分频器得到信号f2再反馈回到鉴相器，即将f2=f1/n与f0输入鉴相器比较相位。此时，为了维持f2与f0的相位差恒定，鉴相器会输出一个电压信号控制锁相环的其他部件，调整输出信号f1，使得f0=f2。此时，锁相环的输出信号f1=n*f2=n*f0。从而实现n倍频。
- 易儿易 2019-06-05 21:30:05
当两个开关都断开的时候，最终的输出结果，取决于之前动作的输出结果，这个也就是我们说的记忆功能
老师，暂存实现原理我能明白，但是有疑虑，暂存的结果有时效性是多少？
时钟信号 CLK再次从0变成1之后，暂存的结果就会被冲掉了吧？时效性是不是短了？
有些迷糊，请指教~
- Ant 2019-06-05 21:22:36
囫囵吞枣的看了，晶体振荡器的实现原理
- 易儿易 2019-06-05 21:21:50
老师，最后一张图，加入反向器之后，不太明白如何用信号D同时控制R-S两个开关……看了之前反向器的介绍，还是没理解……是D输入0之后，R收到1，S收到0，D输入1，R收到0，S收到1吗？