




计算机组成原理学习-实验二 寄存器实验（详细、系统）

原创

小王在努力  于 2020-06-11 16:42:59 发布  7829  收藏 36

分类专栏：[计算机组成原理](#) 文章标签：[嵌入式](#) [其他](#) [经验分享](#) [后端](#) [程序人生](#)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/vangoudan/article/details/106685452>

版权



[计算机组成原理](#) 专栏收录该内容

9 篇文章 34 订阅

订阅专栏

如果你对其他计算机组成原理知识感兴趣，请考虑阅读我的专栏：

[计算机组成原理【专栏】](#)

须知

本文仅作学习笔记使用，仅在CSDN网站发布，如果在其他网站发现，均为侵权行为，请举报。作者：小王在努力。

寄存器实验

须知

实验要求

基本要求

扩展要求

思考内容

实验步骤

- 一、理论分析
- 二、实现方法
- 三、实验结果分析
- 四、思考问题
- 五、问题和解决办法
- 六、结论

后话

实验要求

基本要求

理解CPU运算器中寄存器的作用

设计并验证寄存器组（至少四个寄存器）

利用寄存器或组合逻辑电路实现移位运算功能（至少含左移、右移、循环左移、循环右移四种运算功能）

扩展要求

(扩展)实现更多的寄存器

(扩展)实现多总线结构寄存器访问

思考内容

思考随着寄存器的增多，电路设计的复杂度是什么比例增大

实验步骤

一、理论分析

本次实验需要验证寄存器组，我们选择74LS373来充当寄存器，在此处我们选择用299芯片来实现移位运算功能，包含了左移、右移、循环左移、循环右移，299的功能图（图片摘自电子发烧友网站）如下：

方式	输 入				输入/输出								输 出					
	清零	功能选择		输出控制		时钟	串行		A/Q _A	B/Q _B	C/Q _C	D/Q _D	E/Q _E	F/Q _F	G/Q _G	H/Q _H	Q _A '	Q _H '
	CLR	S ₁	S ₀	$\overline{G1}$	$\overline{G2}$	CLK	SL	SR										
清零	L	X	L	L	L	X	X	X	L	L	L	L	L	L	L	L	L	L
	L	L	X	L	L	X	X	X	L	L	L	L	L	L	L	L	L	L
保持	H	L	L	L	L	X	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}	Q _{EO}	Q _{FO}	Q _{GO}	Q _{HO}	Q _{AO}	Q _{HO}
	H	X	X	L	L	L	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}	Q _{EO}	Q _{FO}	Q _{GO}	Q _{HO}	Q _{AO}	Q _{HO}
右移	H	L	H	L	L	↑	X	H	H	Q _{An}	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	H	Q _{Gn}
	H	L	H	L	L	↑	X	L	L	Q _{An}	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	L	Q _{Gn}
左移	H	H	L	L	L	↑	H	X	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	Q _{Hn}	H	Q _{Bn}	H
	H	H	L	L	L	↑	L	X	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	Q _{Hn}	H	Q _{Bn}	H
置数	H	H	H	X	X	↑	X	X	a	b	c	d	e	f	g	h	a'	h'

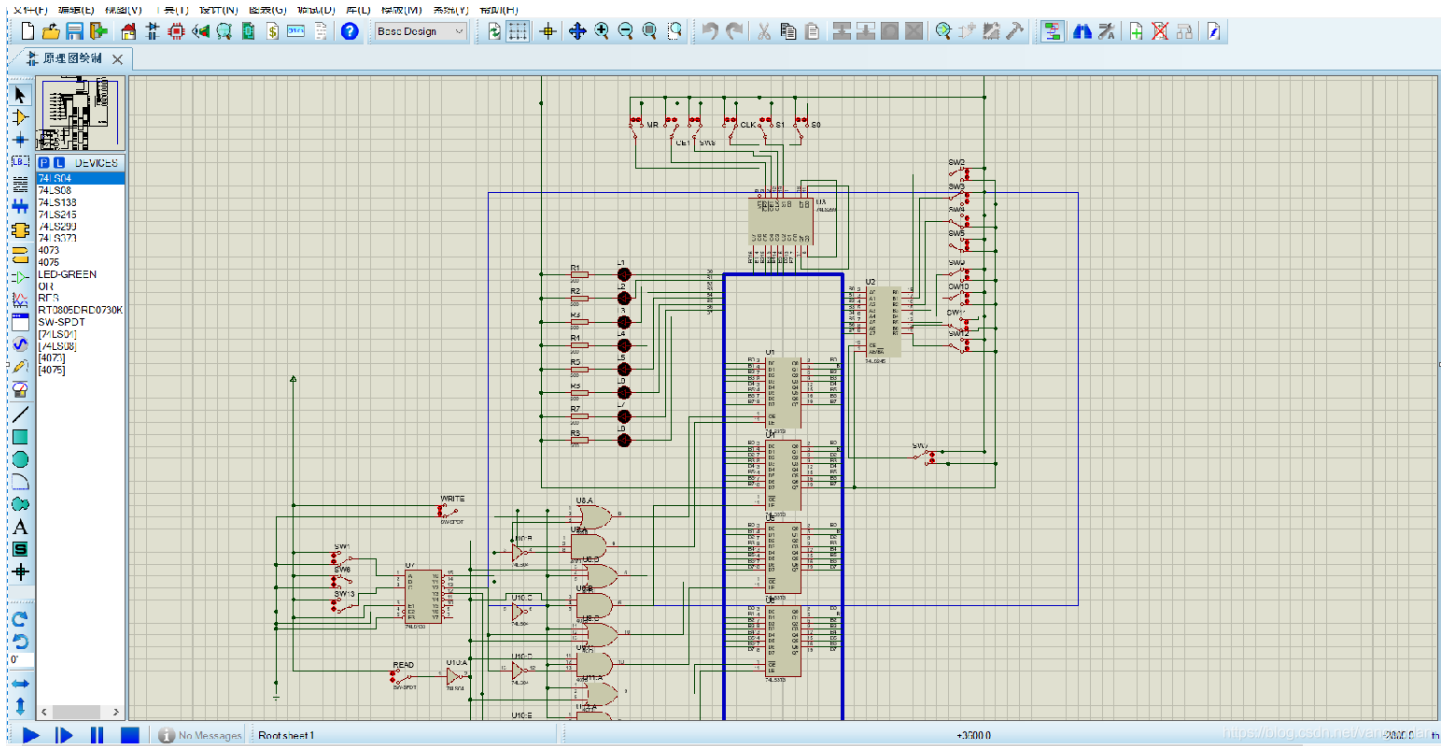
由于扩展要求：用74LS138配合一个读开关和写开关实现选中8个寄存器中的一个并对其进行读写控制，所以我们用到了一些与或运算，因此，我们采用74LS04来实现非运算，4073来实现3位与运算、4075来实现3位或运算。

二、实现方法

基础要求：

四位寄存器的设计：

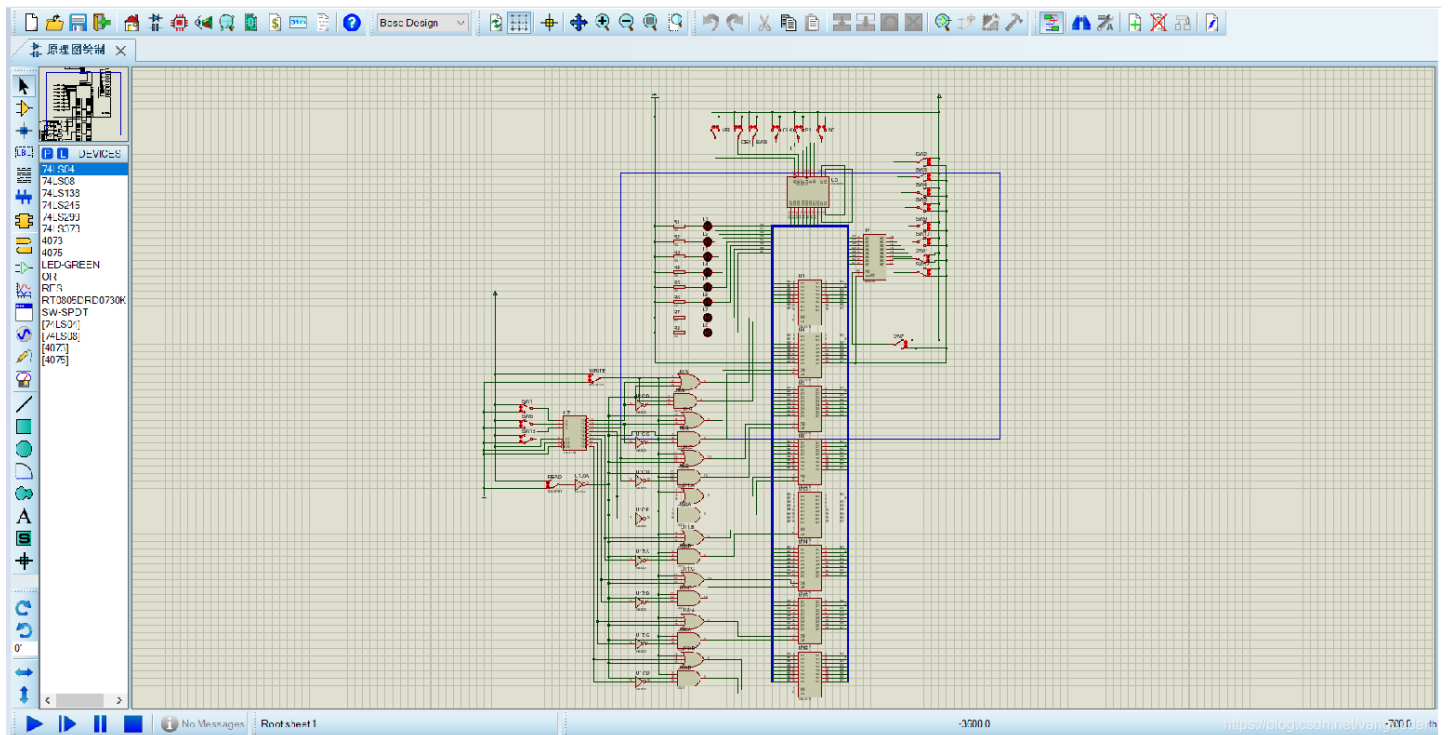
通过使用74LS245将数据输入到总线上，然后数据在总线上流动。如果需要移位操作，则进入74LS299芯片，并将数据保存在74LS299芯片中，同时关闭74LS245,进行移位操作，最后将数据放回到总线上，数据流动到寄存器输入端位置，通过74LS138和两个开关进行与或运算，实现选中并控制寄存器的操作，用74LS138选中寄存器之后，打开写开关，将数据写入到选中寄存器中，然后打开读开关，将数据读出到总线上。



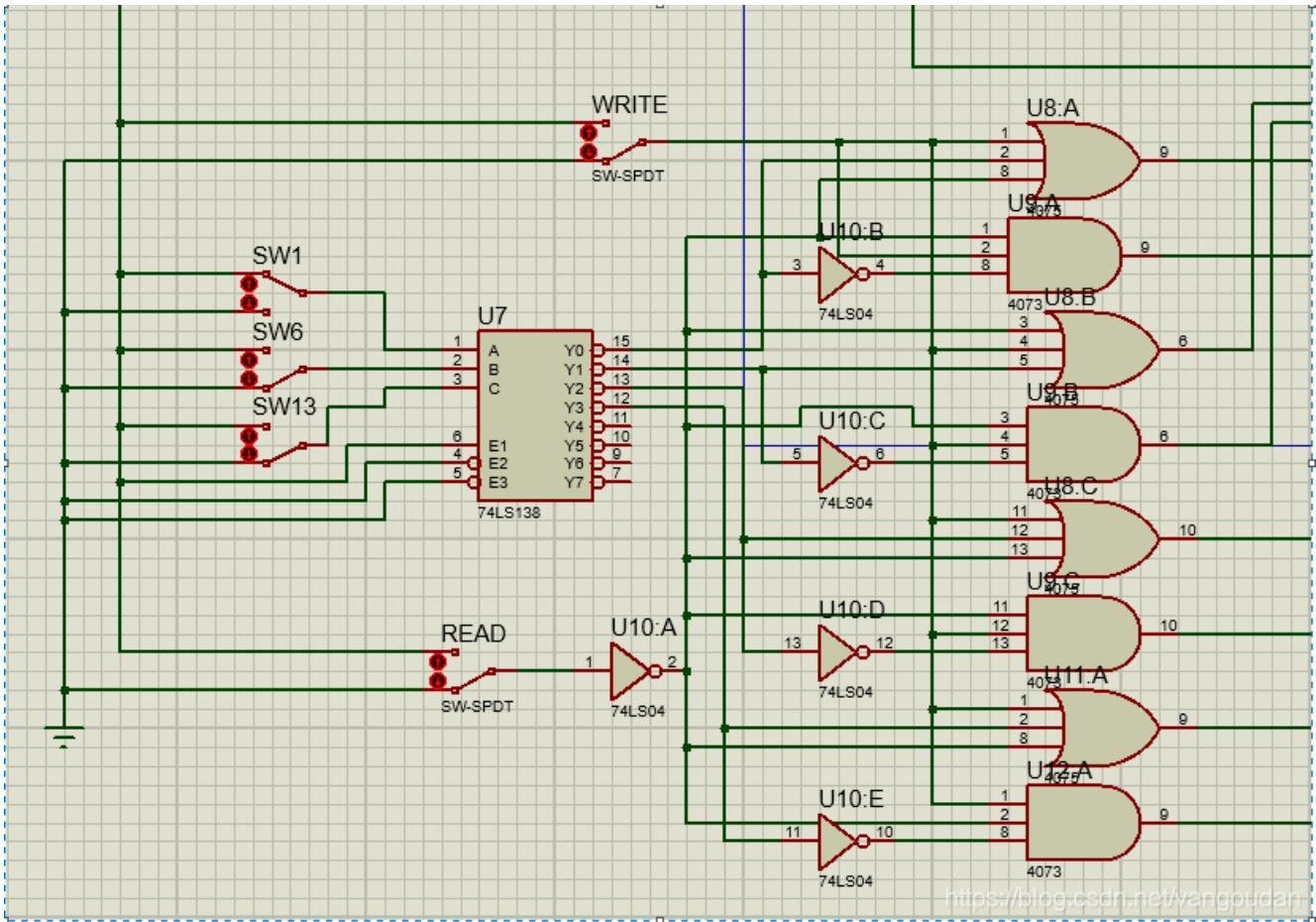
扩展要求:

八个寄存器的设计:

在四个的基础上纵向延伸4个寄存器，其重要功能与四个的一样。



扩展要求: 实现138和读、写开关与或运算实现选中读、选中写功能。



介绍一下读、写开关和74LS138进行的与或运算。

输入			输出	
A (读)	B (写)	C (138 输出)	OE 非	LE
1	0	0	0	0
0	1	0	1	1
X	X	X	1	0

通过真值表可得：

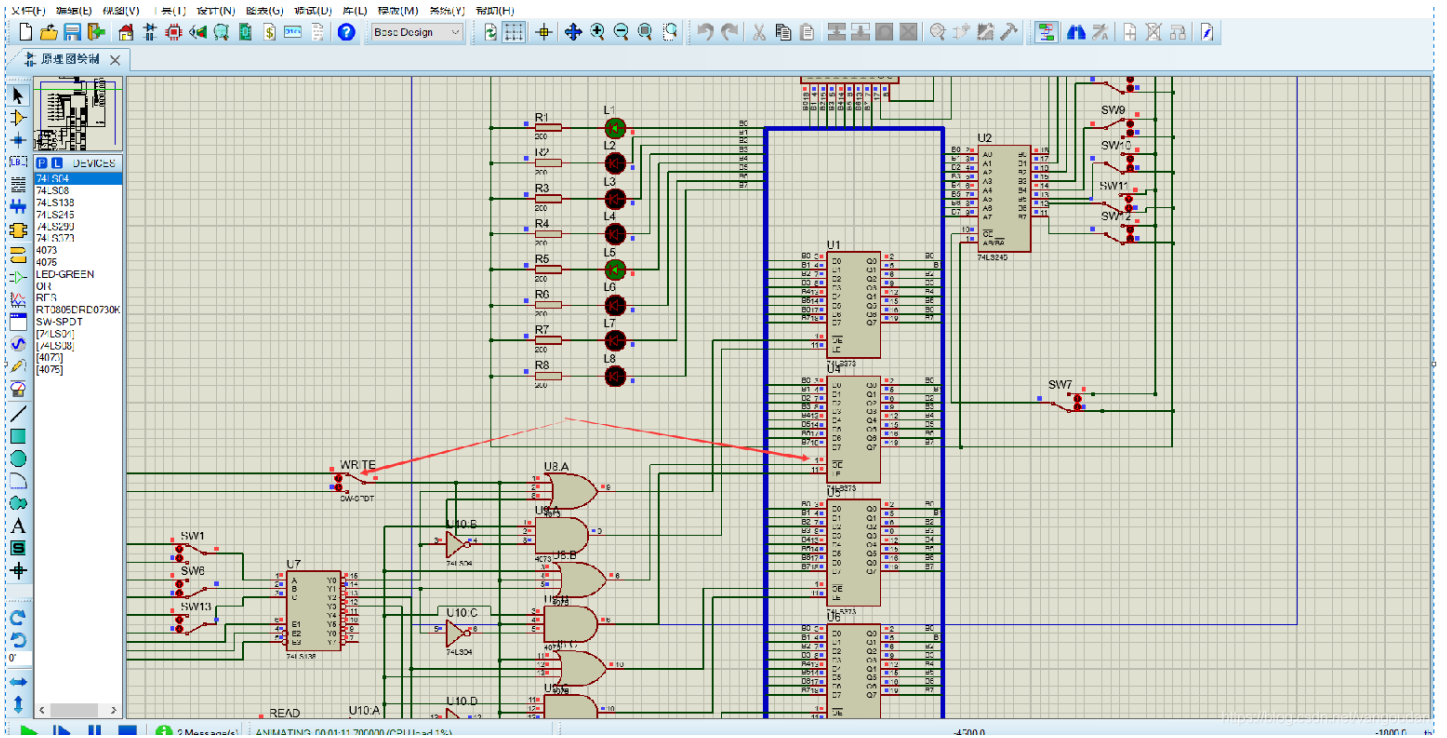
$OE_{非} = A_{非} + B + C$

$LE = (A_{非}) \text{ 与 } B \text{ 与 } (C_{非})$

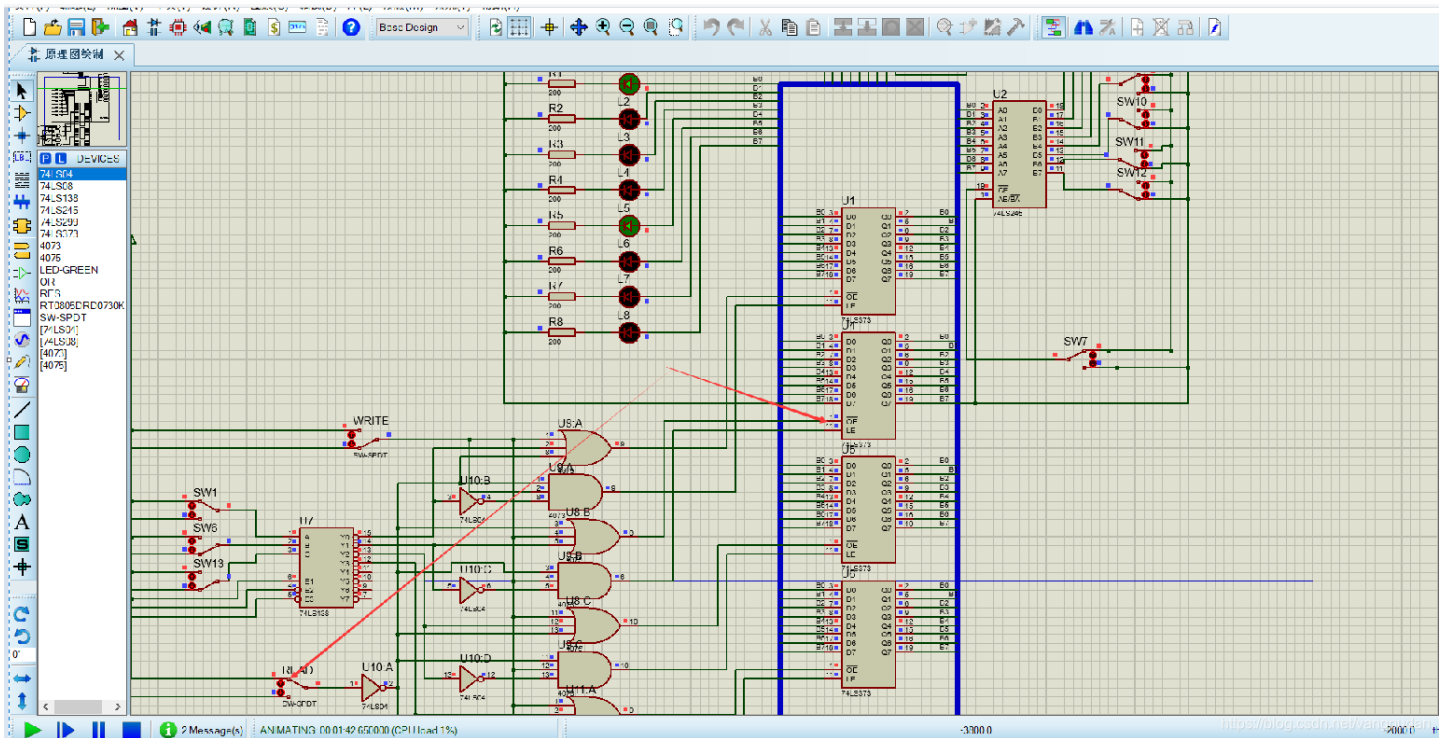
三、实验结果分析

四个八位寄存器读写：

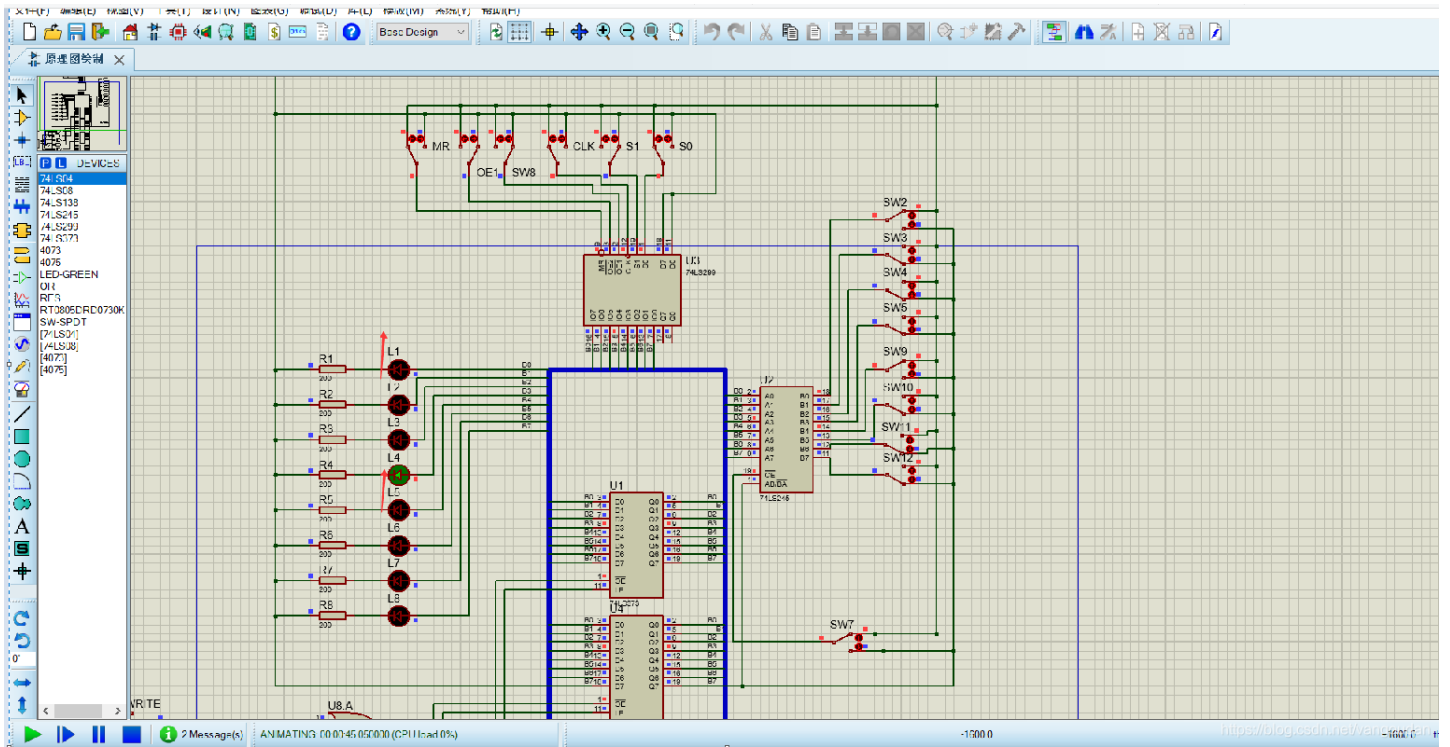
让编号为010的寄存器写入0001 0001，输出结果：



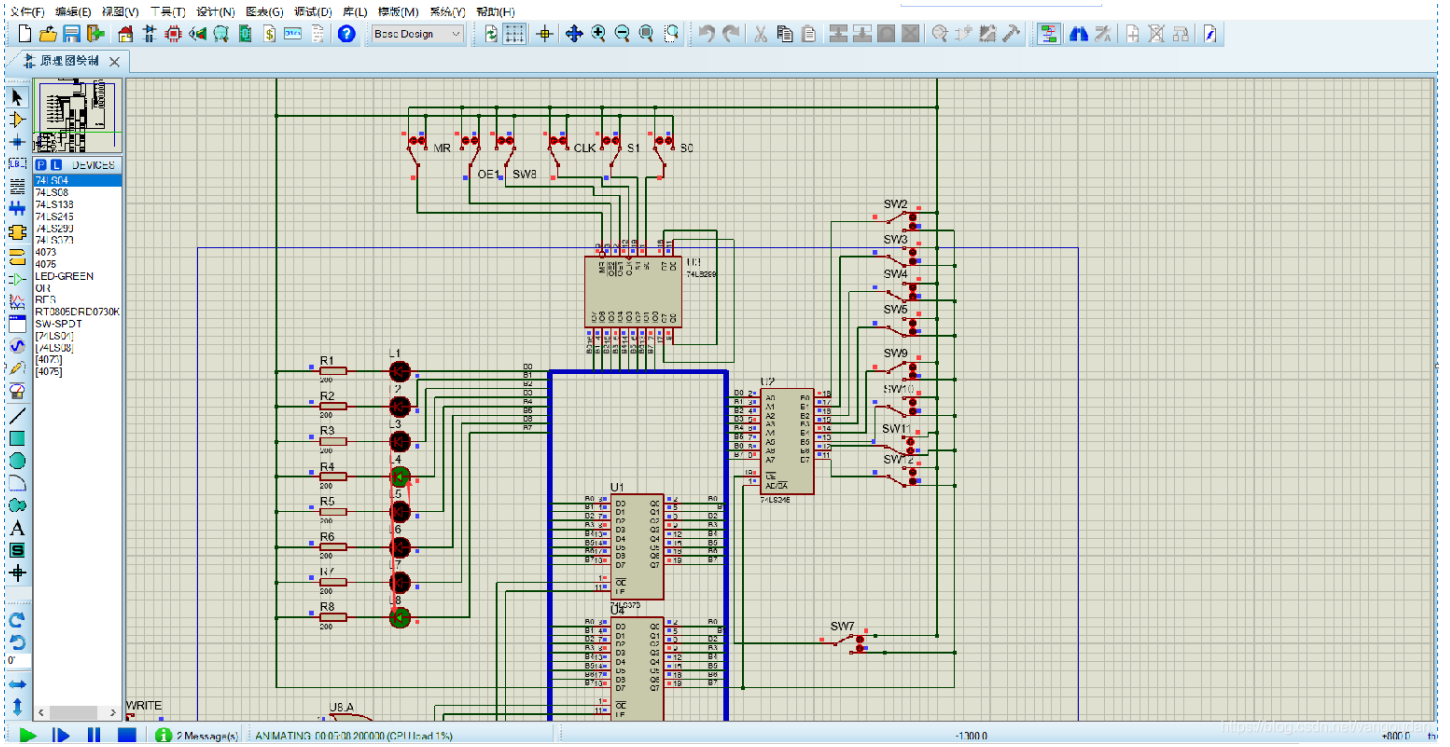
读出结果：



四个寄存器右移（由于篇幅原因，左移右移选择一个实现）：

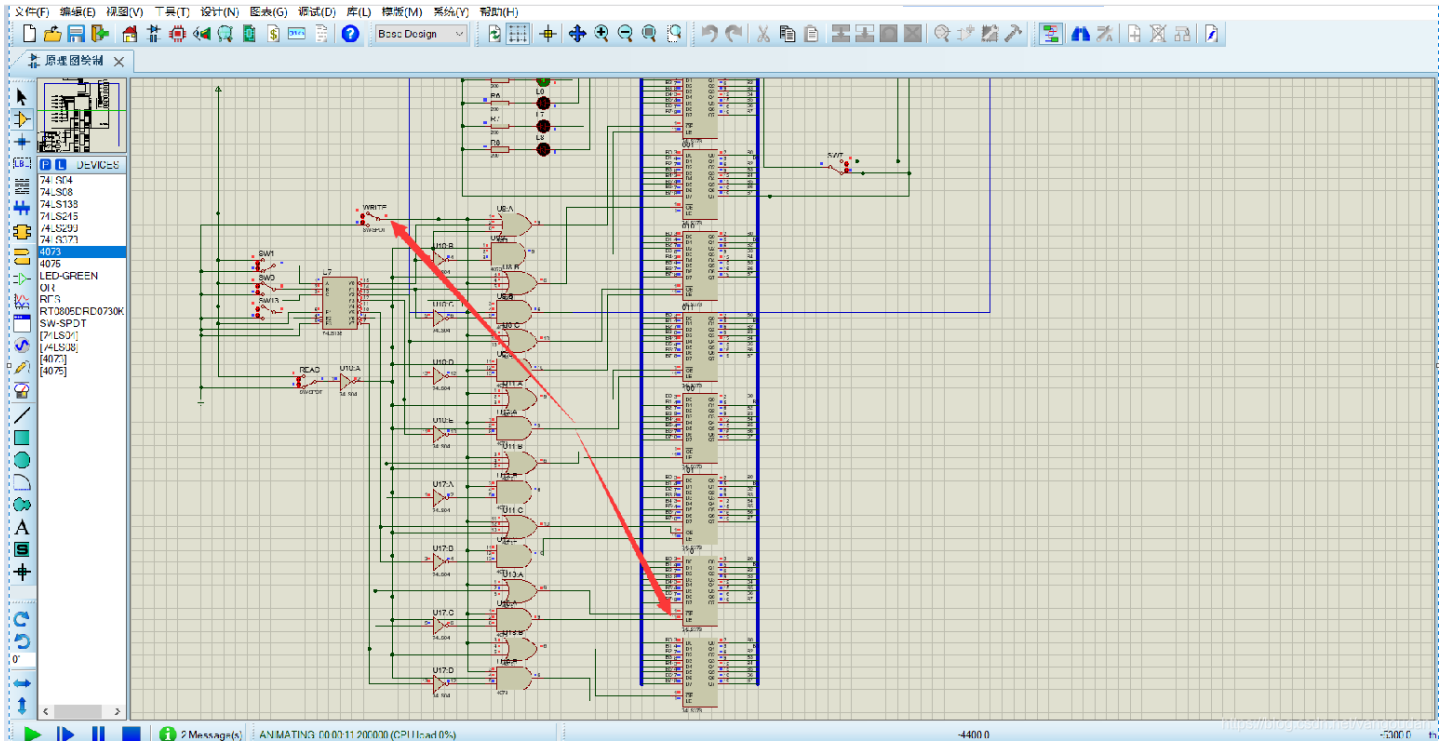


四个寄存器循环右移（由于篇幅原因，循环左移循环右移选择一个实现）：

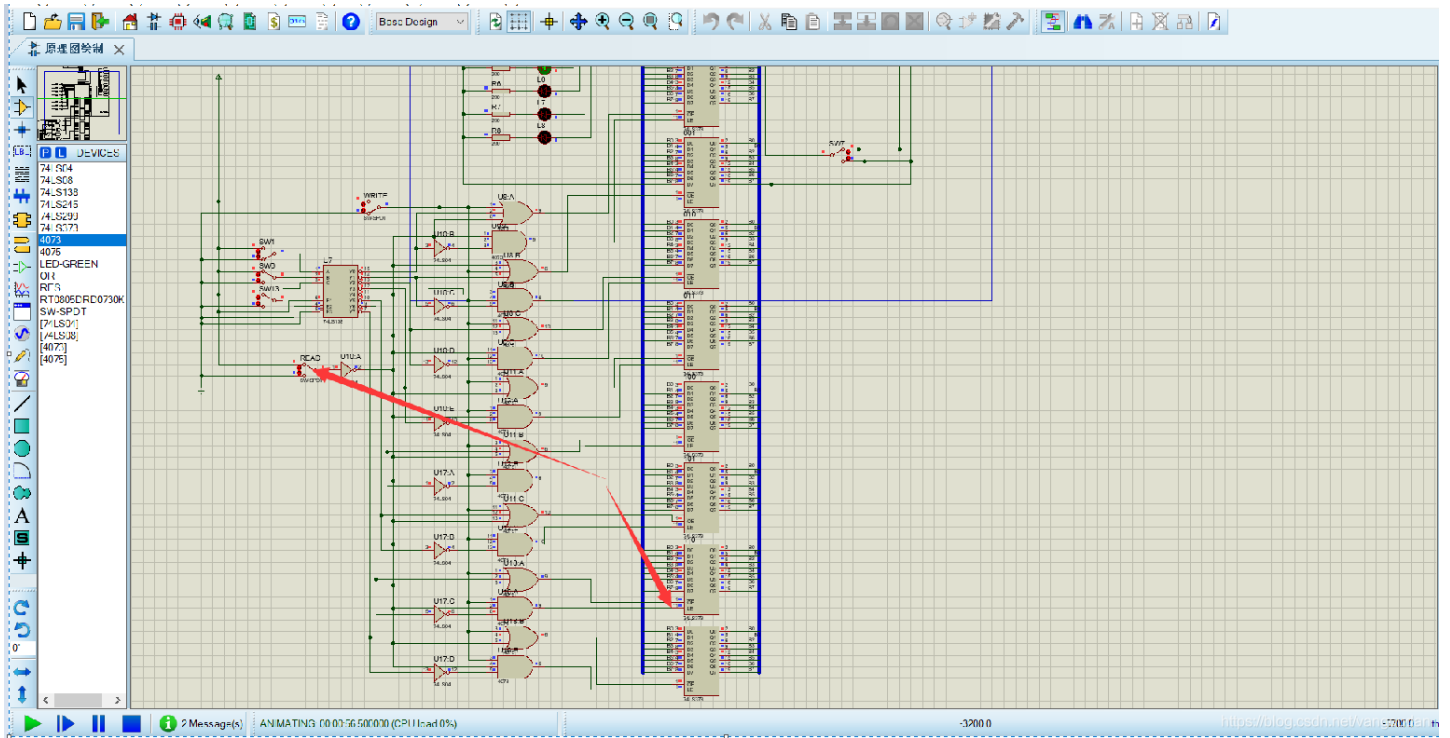


八个八位寄存器读写：

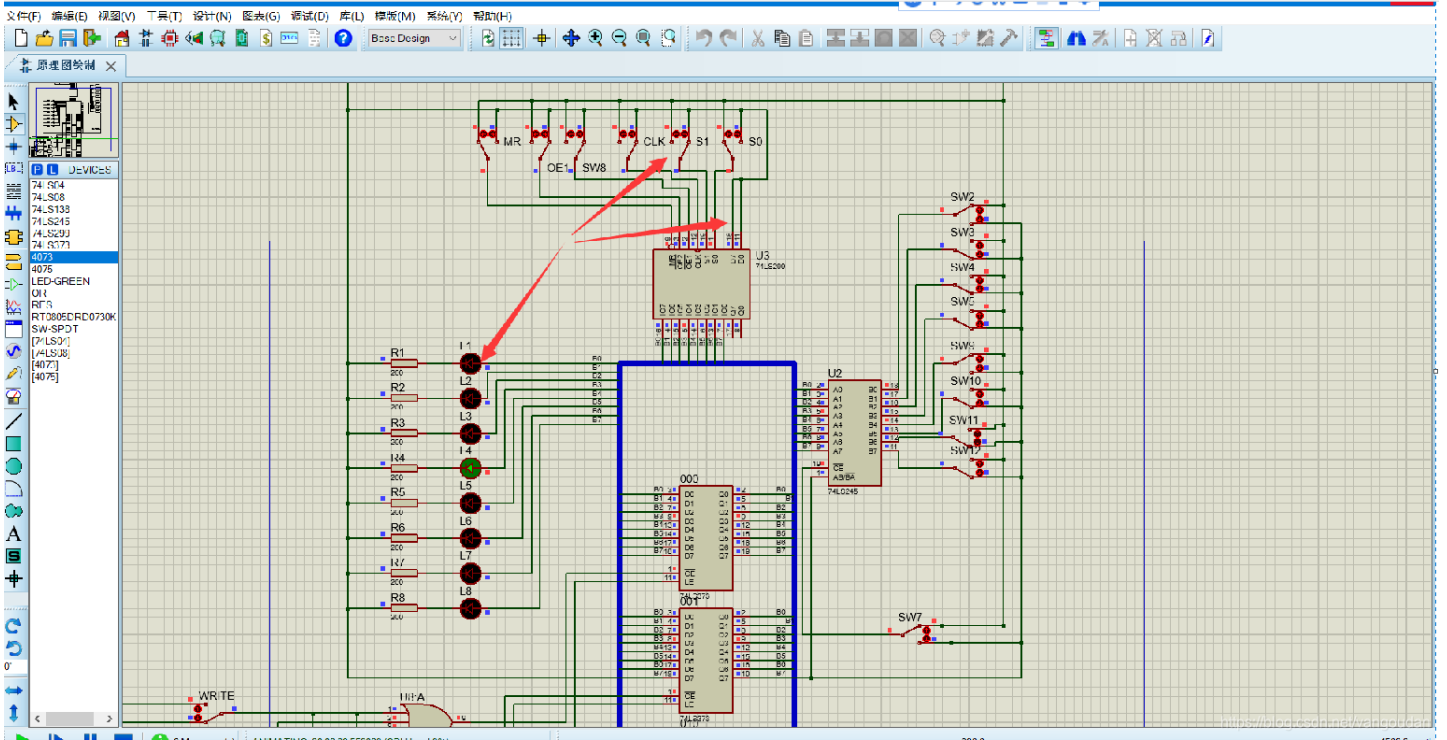
让编号为110的寄存器写入0001 0001，输出结果：



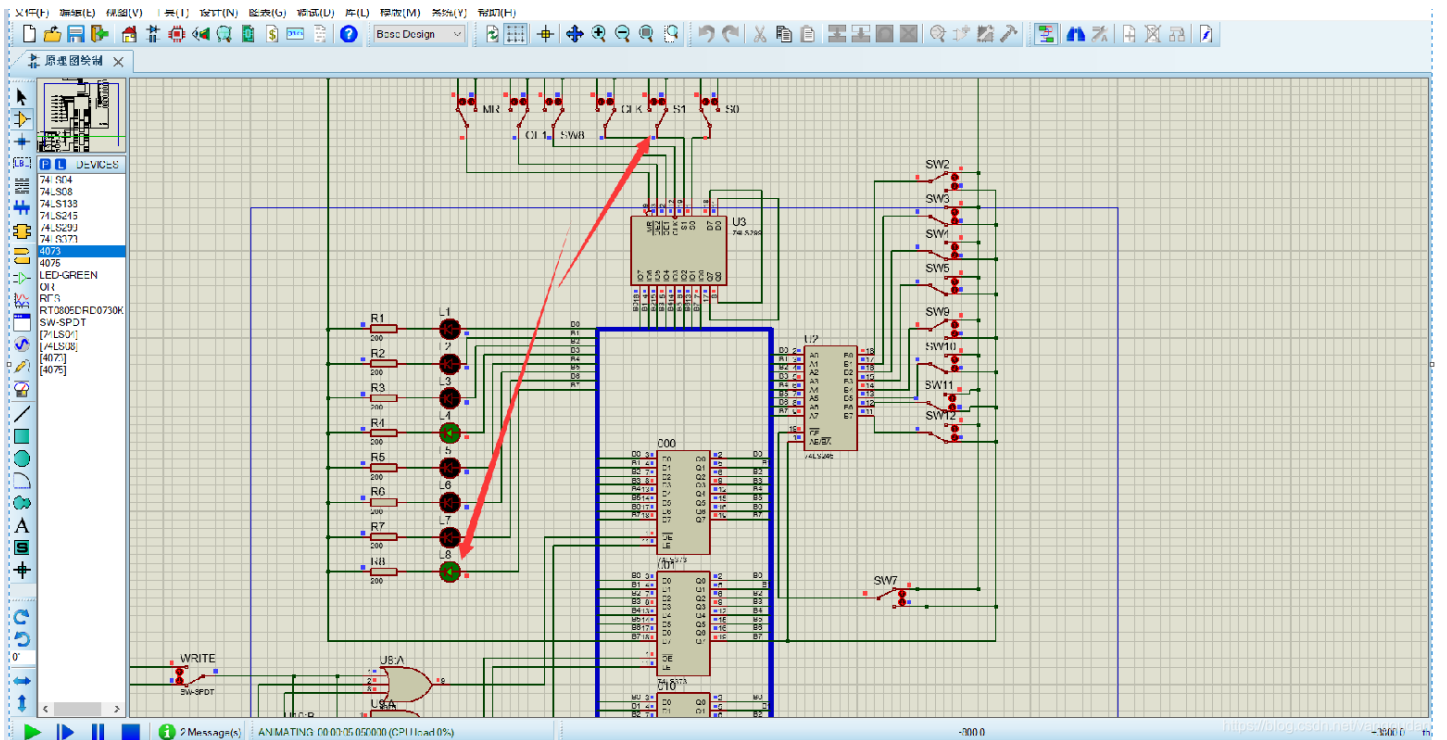
读出结果：



八个寄存器右移（由于篇幅原因，左移右移选择一个实现）：



八个寄存器循环右移（由于篇幅原因，循环左移循环右移选择一个实现）：



四、思考问题

我在增加寄存器的时候，并没有感到电路设计复杂难度有明显的提高(不是博主装X，应该是博主没有理解题目意思吧！)。

五、问题和解决办法

一、299的使用:

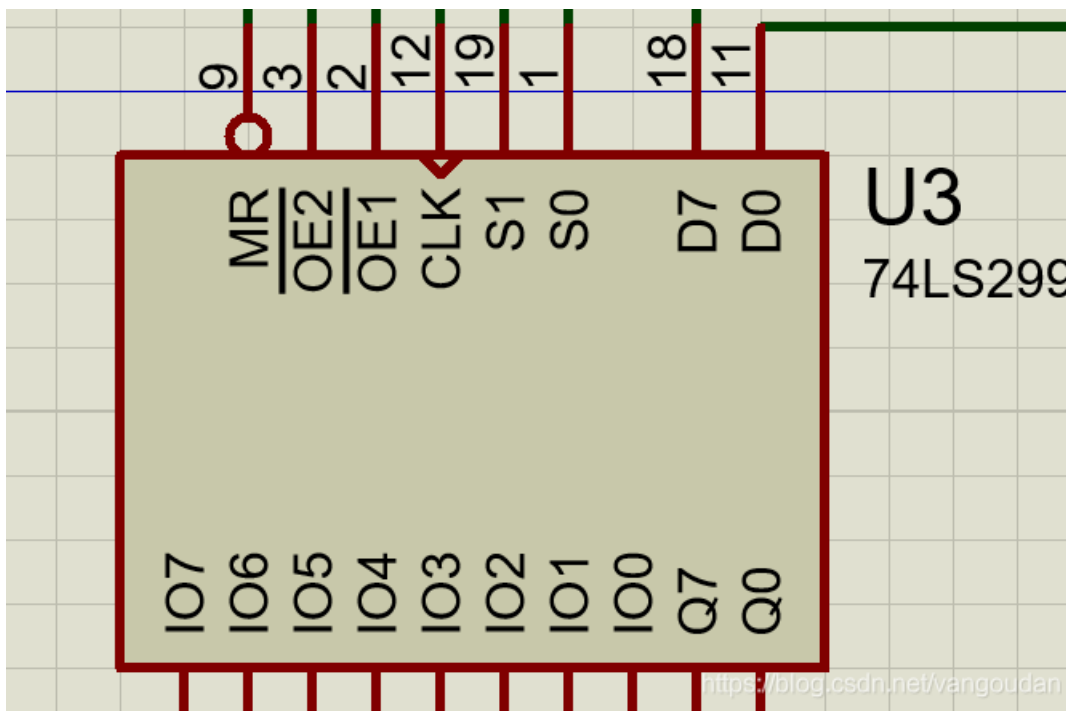
在接触到这个299的芯片的时候，我就纳闷了，你说你弄成个输入输出分开的不好吗？非要输入输出在一起，不吐槽了，不如正题。

首先我们先看功能图：

方式	输				入			输入/输出								输 出		
	清零 CLR	功能选择 S ₁ S ₀		输出控制 G ₁ , G ₂		时钟 CLK	串行 SL SR		A/Q _A	B/Q _B	C/Q _C	D/Q _D	E/Q _E	F/Q _F	G/Q _G	H/Q _H	Q _A '	Q _H '
清零	L	X	L	L	L	X	X	X	L	L	L	L	L	L	L	L	L	L
	L	L	X	L	L	X	X	X	L	L	L	L	L	L	L	L	L	L
保持	H	L	L	L	L	X	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}	Q _{EO}	Q _{FO}	Q _{GO}	Q _{HO}	Q _{AO} Q _{HO}	
	H	X	X	L	L	L	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}	Q _{EO}	Q _{FO}	Q _{GO}	Q _{HO}	Q _{AO} Q _{HO}	
右移	H	L	H	L	L	↑	X	H	H	Q _{An}	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	H Q _{Gn}	
	H	L	H	L	L	↑	X	L	L	Q _{An}	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	L Q _{Gn}	
左移	H	H	L	L	L	↑	H	X	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	Q _{Hn}	H	Q _{Bn} H	
	H	H	L	L	L	↑	L	X	Q _{Bn}	Q _{Cn}	Q _{Dn}	Q _{En}	Q _{Fn}	Q _{Gn}	Q _{Hn}	H	Q _{Bn} H	
置数	H	H	H	X	X	↑	X	X	a	b	c	d	e	f	g	h	a' h'	

从这张图上，我们得出，299的功能有：清零、保持、左移、右移和置数功能。但是我们只需要用到他的三个功能：左移、右移和置数功能。

proteus里面的74LS299:



简单说一下上面的端口对应：

proteus	真实端口
MR	CLR非
OE2非	G2非
OE1非	G1非
CLK	CLK
S1	S1

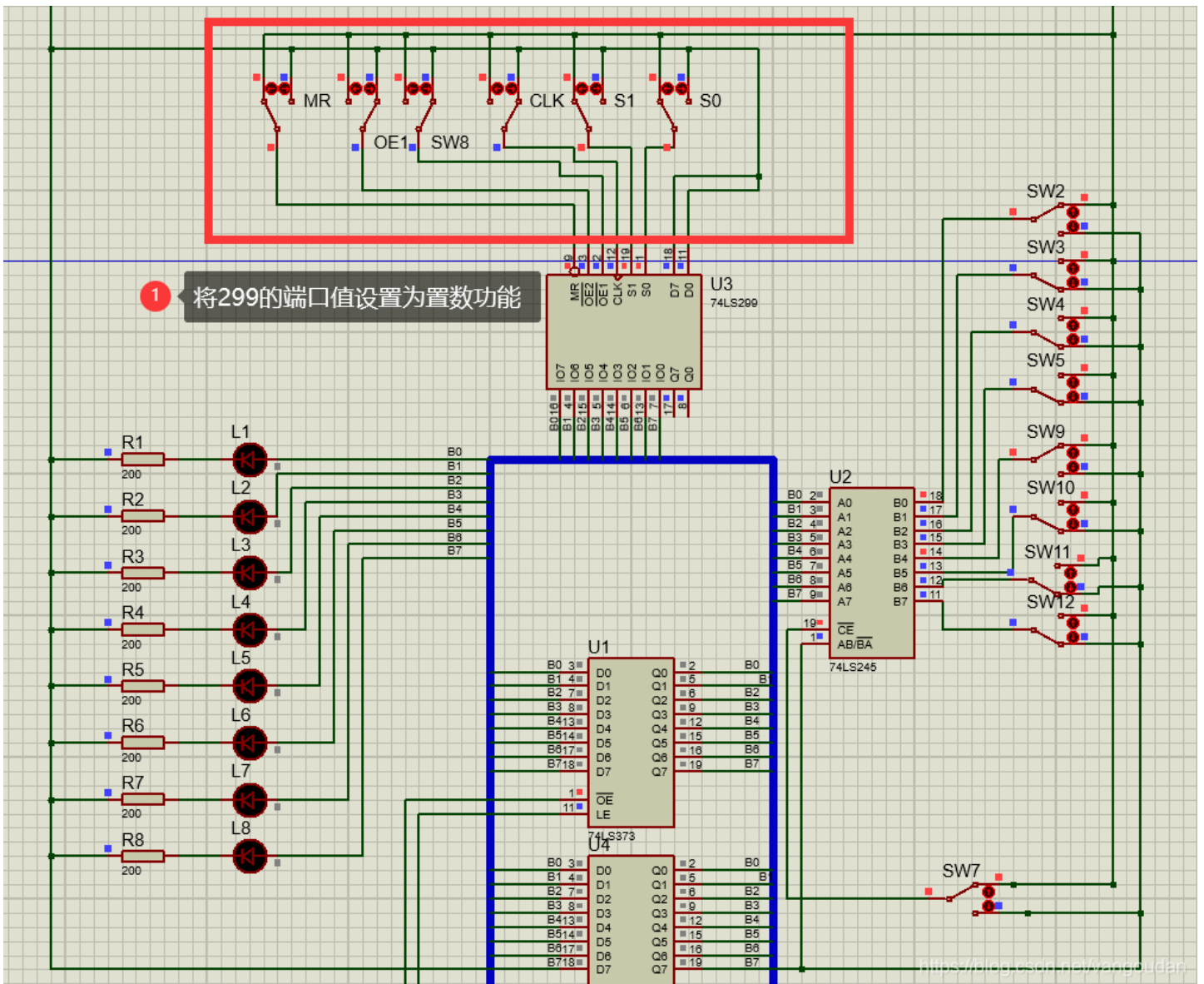
proteus	真实端口
S0	S0
D7	SL
D0	SR

看完功能表和端口对应，想必大家对299芯片有了一个初步的认识了，接下来我就来演示如何实现左移功能：

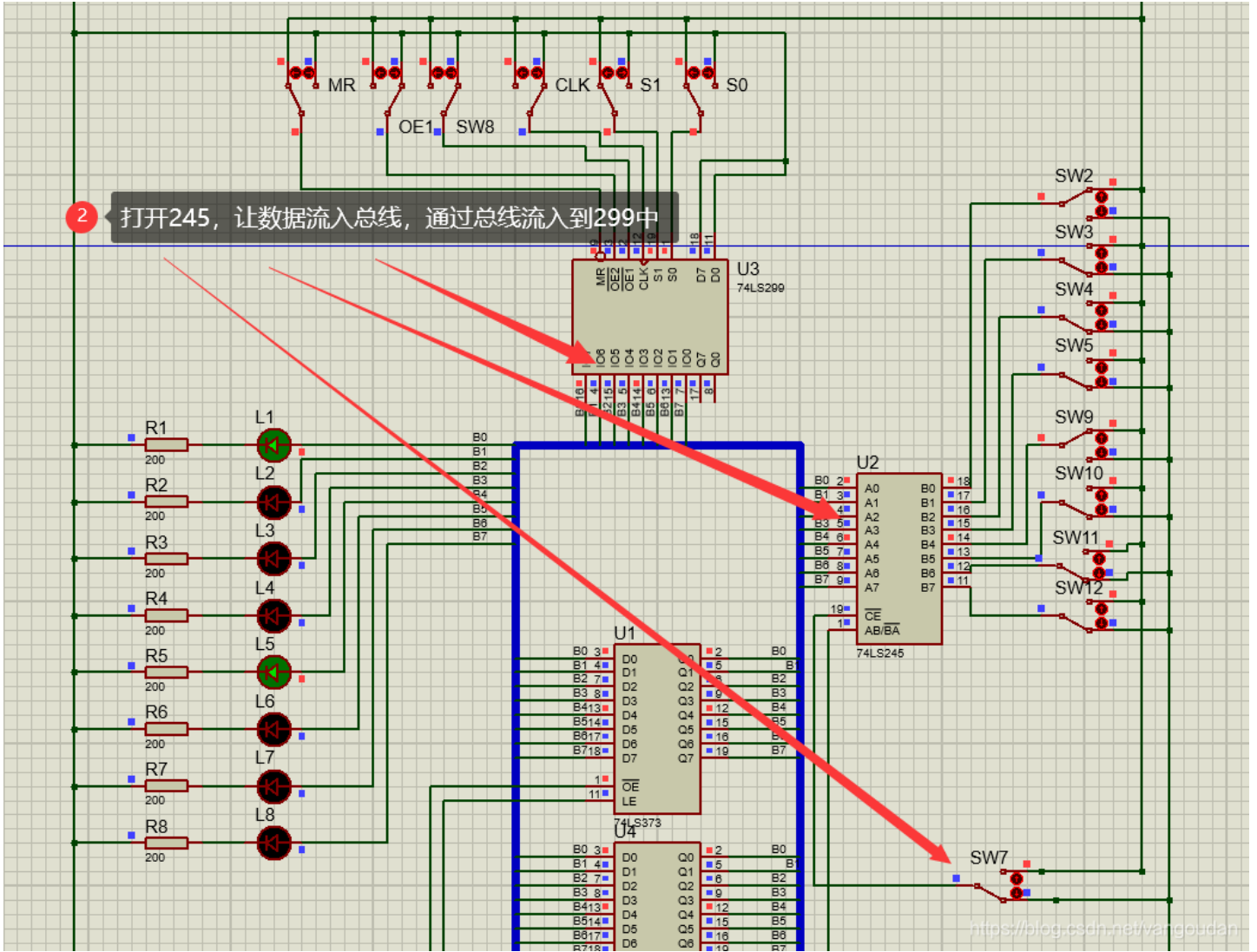
总体来说，想要让其实现移位，得让要移位的数据存储进299芯片内部，也就是第一步，置数，通过查看功能表，只需要将：MR置为1，OE2非和OE1非置为0，S1和S0置为1，D7和D0置为0。然后让数据流入，此时拨动CLK，实施置数功能即可，然后改变S1或S0通过拨动CLK来实现自己想要的功能。

具体步骤：

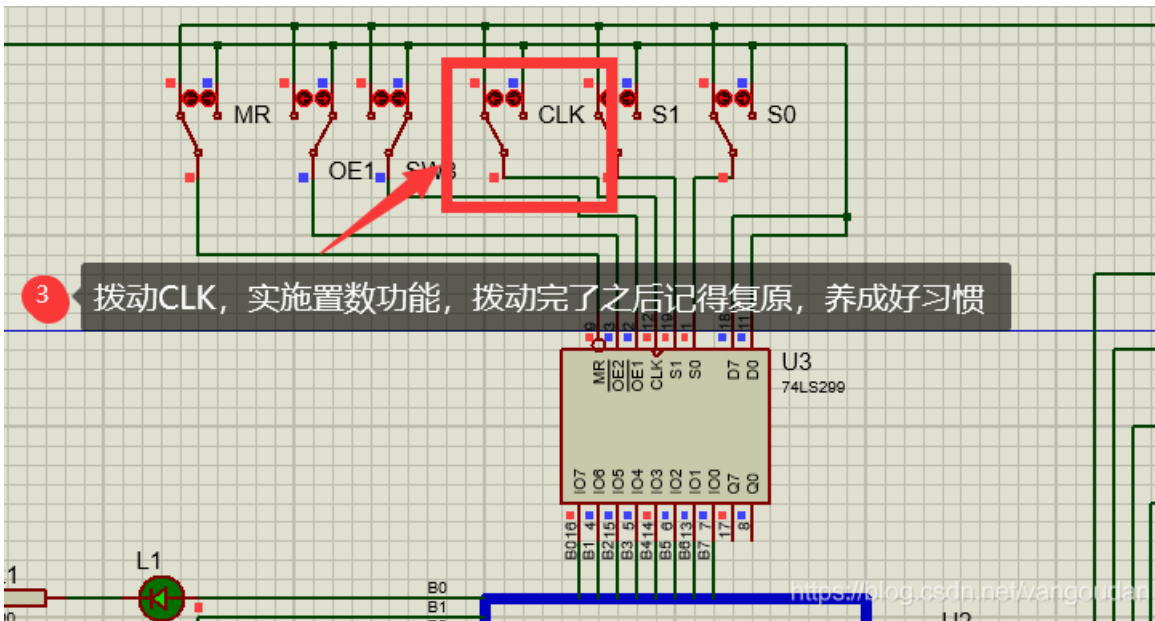
(1) .将299的端口值设置为置数功能（参考功能表进行设置）



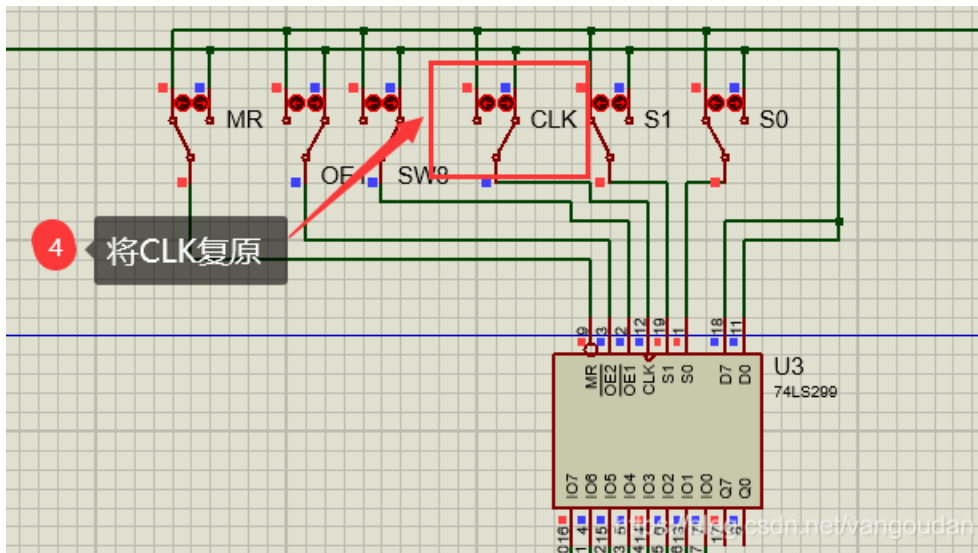
(2) .打开245，让245内保存的数据流入总线，同时流经299。



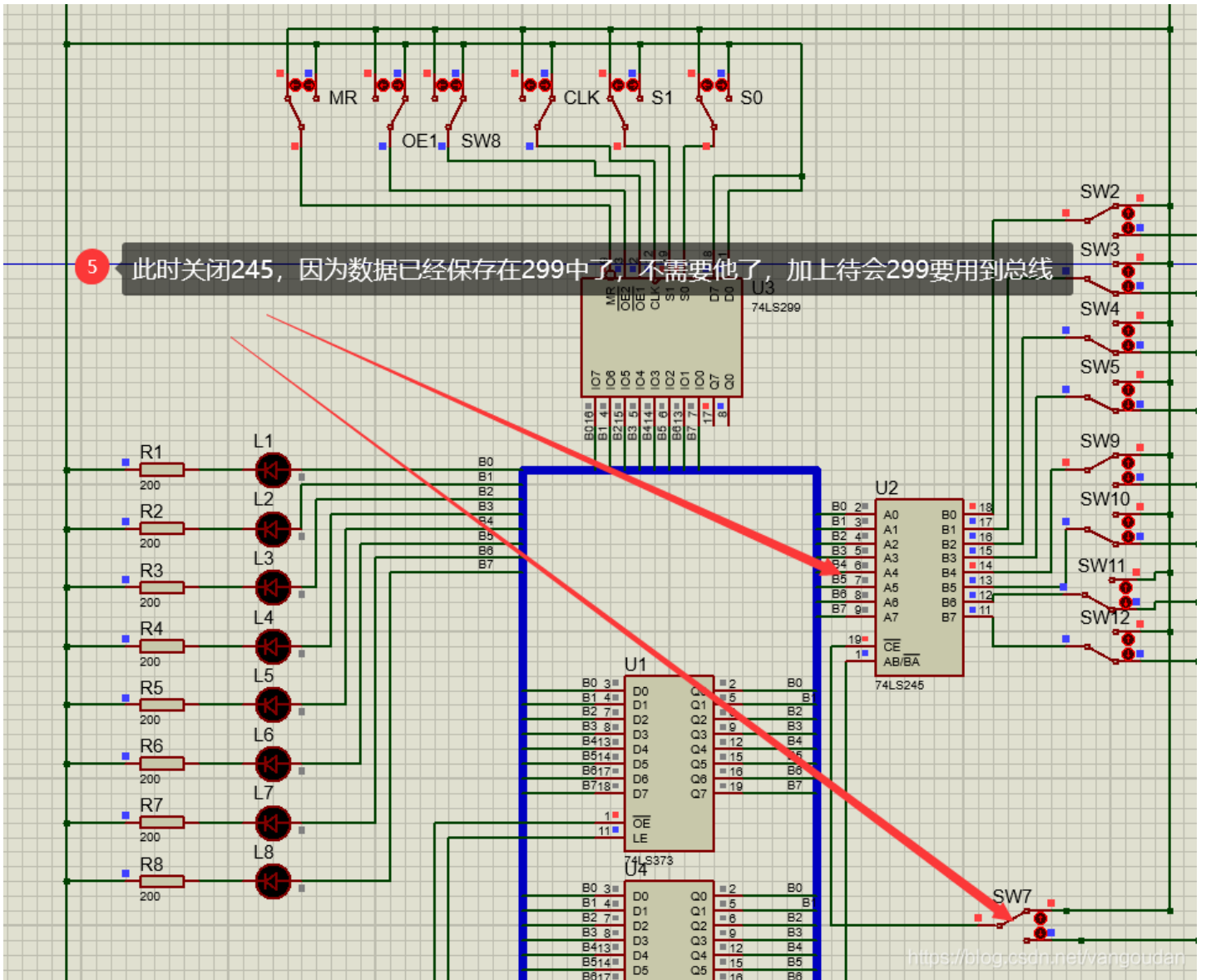
(3) .此时总线上有了数据, 而且299还设置为置数功能, 此时只需要拨动两下下CLK, 实现这个功能即可。



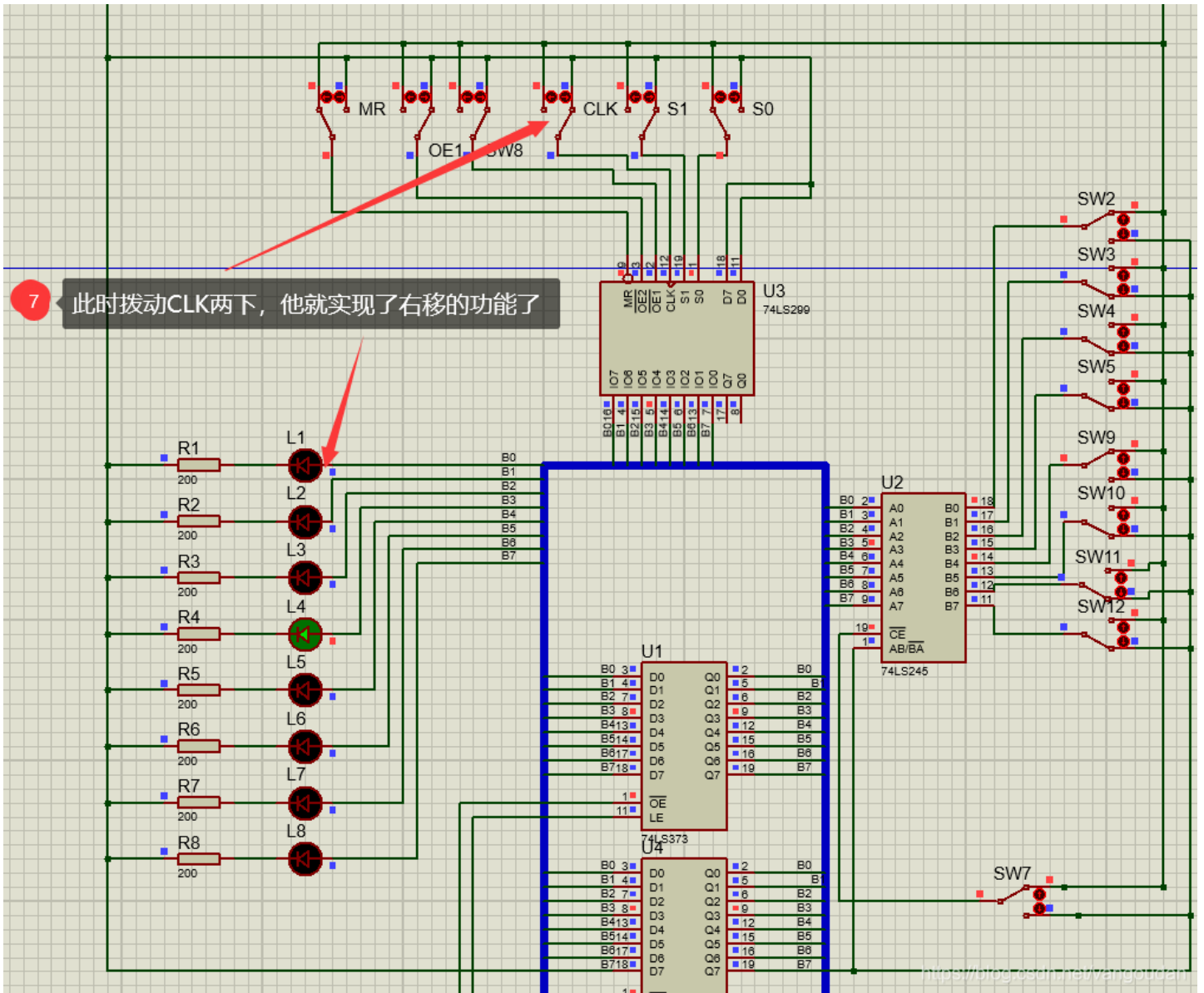
(4) .拨动完一次CLK之后，记得将其复原为低电平，以便形成下一次上升沿。



(5) .此时关闭245，因为299已经实现了置数功能，也就是将数据保存在了299芯片中了，我们就不需要245的数据了，而且待会299也需要往总线上输出数据，所以现在需要将245关闭。



(6) 此时, 我们的299内部已经保存了要移位的数据了, 我们只需要设置S0和S1来实现我们想要的功能, 此处我将S1置为0, S0置为1来实现右移的功能, 我们看到, 在我设置S1和S0之后, 299内部保存的数据就放出来了。



二、74LS373的使用：

我一直以为74LS373就是一个简单的锁存器，其接口OE非是类似“总闸”的一个芯片开关，LE是让数据流过的开关，结果打错特错！

OE非是74LS373的读开关，此处的读是讲74LS373芯片内锁存的数据读出芯片，而LE则是74LS373的写开关，同理，此处的写是将外部数据写入到74LS373中。

六、结论

通过本次实验，我深刻了解了74LS373的功能，并学会了74LS299芯片的使用，重要的是了解了运算器内部寄存器的工作方式，并运用了数电知识来解决自己遇到的问题。

后话

1. 首先给大家说一下，博主经常在线，如果有什么问题或者想法，可以在下方评论，我会积极反馈的。
2. 其次还是要请大家能够多多指出问题，我也会在评论区等候大家！



想battle吗



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)