



实验三

191098328 计算机科学与技术系 张世茂

191098328@smail.nju.edu.cn



2021-11-15

目录

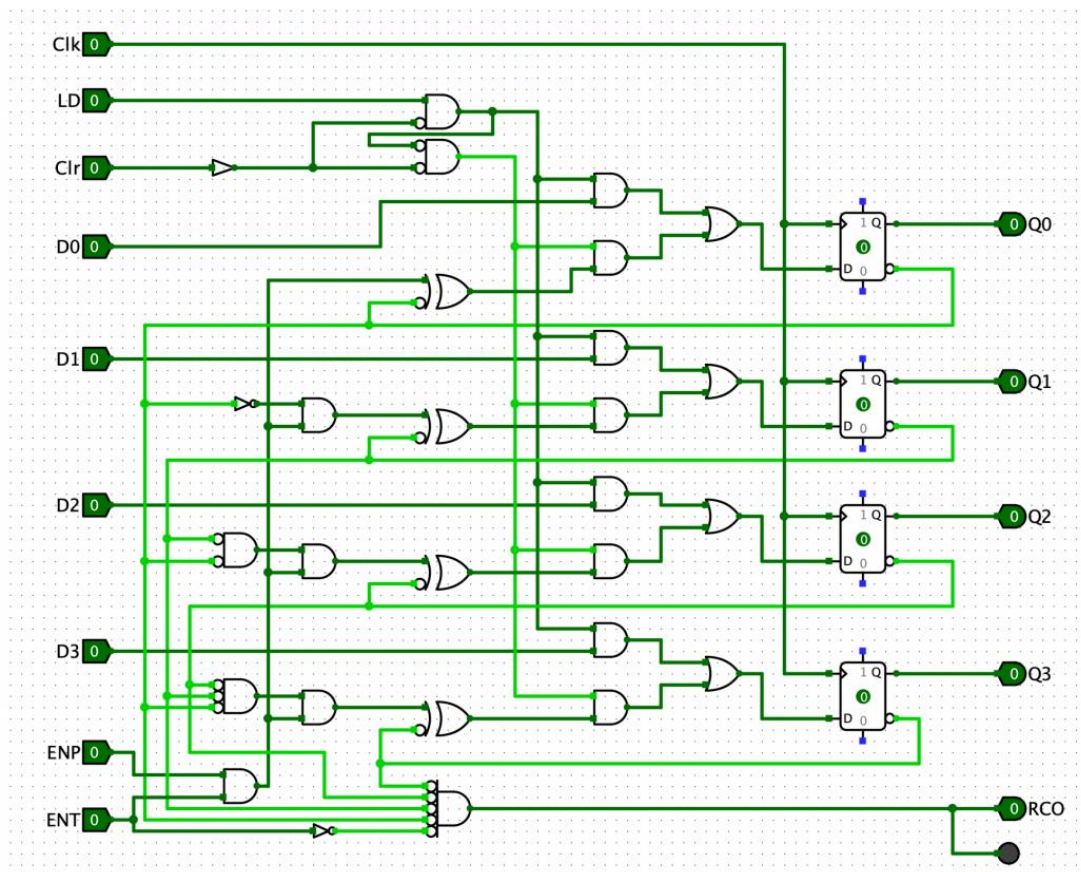
1. 十进制计数器（清零计数&置位计数）	2
1.1 实验操作流程	2
1.2 实验结果	4
1.3 实验错误与原因分析	8
2. 生成重复二进制序列	8
2.1 实验操作流程	8
2.2 实验结果	10
2.3 实验错误与原因分析	12
3. 含有 8 个 32 位寄存器堆 Regfile 的读写电路	12
3.1 实验操作流程	12
3.2 实验结果	13
3.3 实验错误与原因分析	15
4. 思考题	18
5. 实验总结	21

1. 十进制计数器（清零计数&置位计数）

1.1 实验操作流程

设计十进制计数器，首先要设计出一个 4 位同步二进制计数器 CNTR4U 子电路。

4 位同步二进制计数器的电路原理图如下所示：

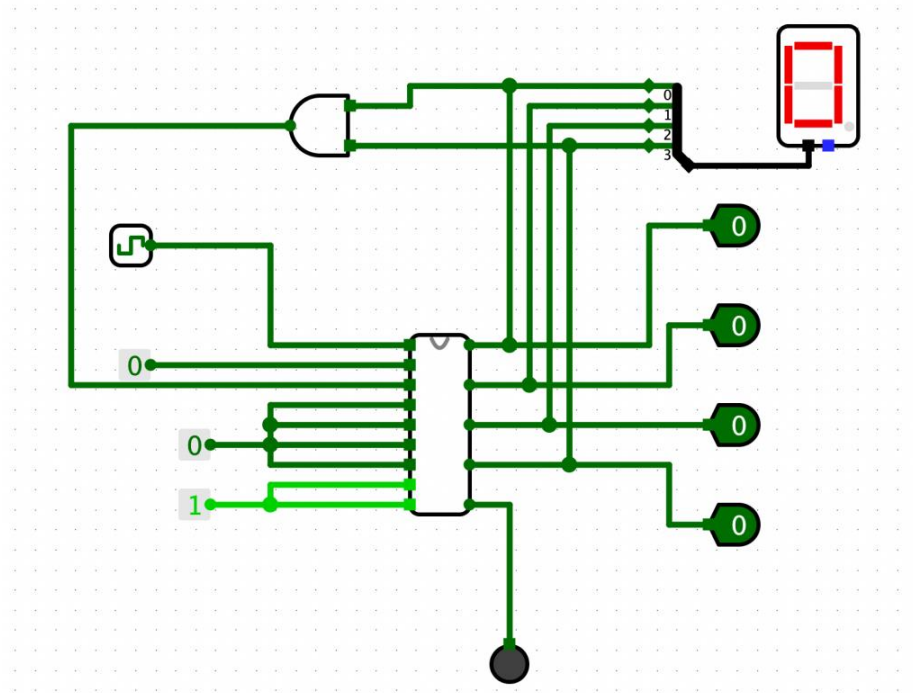


根据目标要求可以知道，CLR 为清零端，被置为有效时四个数值输出端在下一个上升沿被置为全 0；LD 为置位端，当 CLR 不为有效且 LD 被置为有效时四个数值输出端在下一个上升沿被置为 Q3Q2Q1Q0；当上述两者都被置为 0 时，若 ENP 和 ENT 中有任何一个被置为 0，则输出端的输出无论何时都保持不变；当 CLR、LD 全为 0，且 ENP 和 ENT 都被置为 1 时，在每一个时钟的上升沿 Q3Q2Q1Q0 都会加 1，直至为全 1 后在下一个时钟上升沿返回 0000 重新开始计数。

图中还有一个指示位输出端 RCO 以及与之相连的 LED 灯，在 Q3Q2Q1Q0 为全 1 时 RCO 输出为 1，LED 灯被点亮。

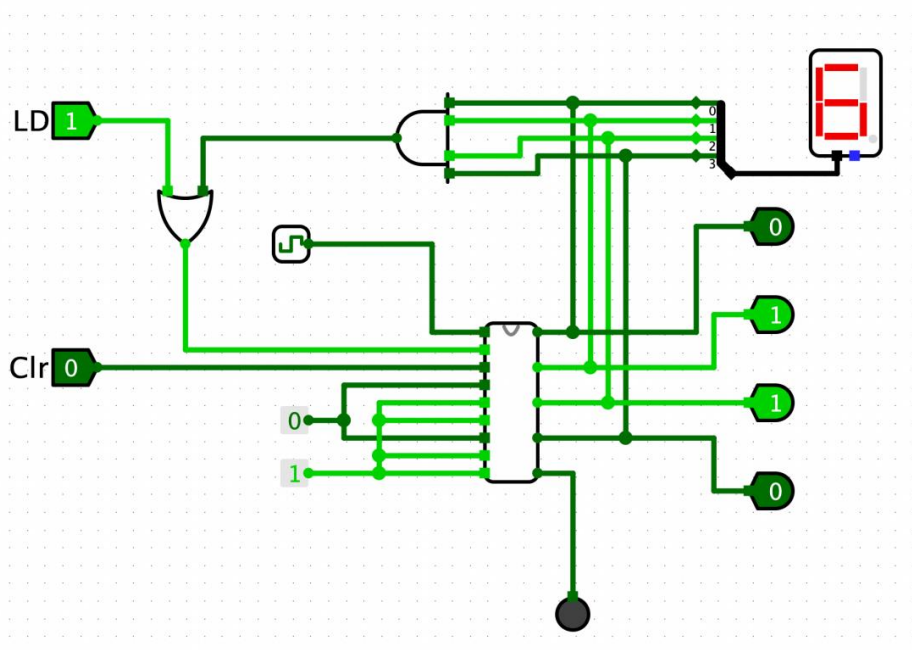
有了上面所示的 4 位同步二进制计数器，就可以实现下面的十进制计数器，其具体实现方法有清零计数和置位计数两种。

清零计数的电路原理图如下所示：



可以看到，由于在循环计数的过程中 ENP、ENT 无需变化，因此使用常量将 ENT、ENP 置为恒 1。又因为采用清零计数，因此置位使能端 LD 被置为恒 0，D3D2D1D0 被置为常量。将四位数值输出 Q3Q2Q1Q0 通过分线器连接至十六进制数码管，同时由于 Q3Q2Q1Q0 为 1001B 的下一个上升沿需要被清零，因此将对应的两位 Q3、Q0 通过二输入与门连接至清零端 CLR，这样就实现了如上图所示的符合要求的十进制清零计数器。

置位计数的电路原理图如下所示：



可以看到，在置位计数电路中，ENP 和 ENT 仍然被置为恒 1，而 D3D2D1D0 的值此时为 0110，为之后的置数做准备。输入端 Clr 连至清零端，时钟信号连接至时钟输入端。将四位数值输出 Q3Q2Q1Q0 通过分线器连接至十六进制数码管，同时由于 Q3Q2Q1Q0 为 1111B 时下一个上升沿需要被置数为 0110B，故将四位数值输入端通过与门，并将输出的信号和 LD 通过或门输入到置位端，这样在 LD 为 1 或 Q3Q2Q1Q0 为 1111B 时，下一个上升沿来临后计数器都可以被置为 0110B。这样就实现了如上图所示的符合要求的十进制置位计数器。

在实现所有要求的电路后，对电路进行进一步的仿真检验。

1.2 实验结果

首先分析相关电路的输入输出对应表，4 位同步二进制计数器子电路输入输出对应表如下：

CLR	LD	ENT	ENP	Q3	Q2	Q1	Q0	Q3*	Q2*	Q1*	Q0*
1	x	x	x	x	x	x	x	0	0	0	0
0	1	x	x	x	x	x	x	D3	D2	D1	D0
0	0	0	x	x	x	x	x	Q3	Q2	Q1	Q0
0	0	x	0	x	x	x	x	Q3	Q2	Q1	Q0
0	0	1	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	1	0	1	0	0
0	0	1	1	0	1	0	0	0	1	0	1
0	0	1	1	0	1	0	1	0	1	1	0
0	0	1	1	0	1	1	0	0	1	1	1
0	0	1	1	0	1	1	1	1	0	0	0
0	0	1	1	1	0	0	0	1	0	0	1
0	0	1	1	1	0	0	1	1	0	1	0
0	0	1	1	1	0	1	0	1	0	1	1
0	0	1	1	1	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0	1	1	0	1
0	0	1	1	1	1	0	1	1	1	1	0
0	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	1	1	1	1	0	0	0	0

然后分别列出十进制清零计数器和十进制置位计数器的输入输出对应表。

十进制置位计数器：

CLR	LD	Q3	Q2	Q1	Q0	Q3*	Q2*	Q1*	Q0*
1	x	x	x	x	x	0	0	0	0
0	1	x	x	x	x	0	1	1	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	1	0
0	0	0	0	1	0	0	0	1	1
0	0	0	0	1	1	0	1	0	0
0	0	0	1	0	0	0	1	0	1
0	0	0	1	0	1	0	1	1	0
0	0	0	1	1	0	0	1	1	1
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	1	0	0	1
0	0	1	0	0	1	1	0	1	0
0	0	1	0	1	0	1	0	1	1
0	0	1	0	1	1	1	1	0	0
0	0	1	1	0	0	1	1	0	1
0	0	1	1	0	1	1	1	1	0
0	0	1	1	1	0	1	1	1	1
0	0	1	1	1	1	0	1	1	0

十进制清零计数器：

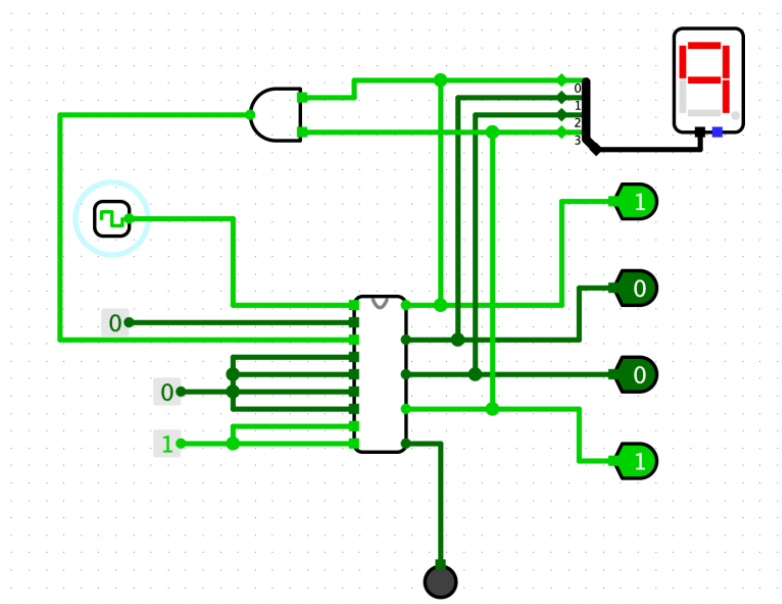
Q3	Q2	Q1	Q0	Q3*	Q2*	Q1*	Q0*
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0

1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

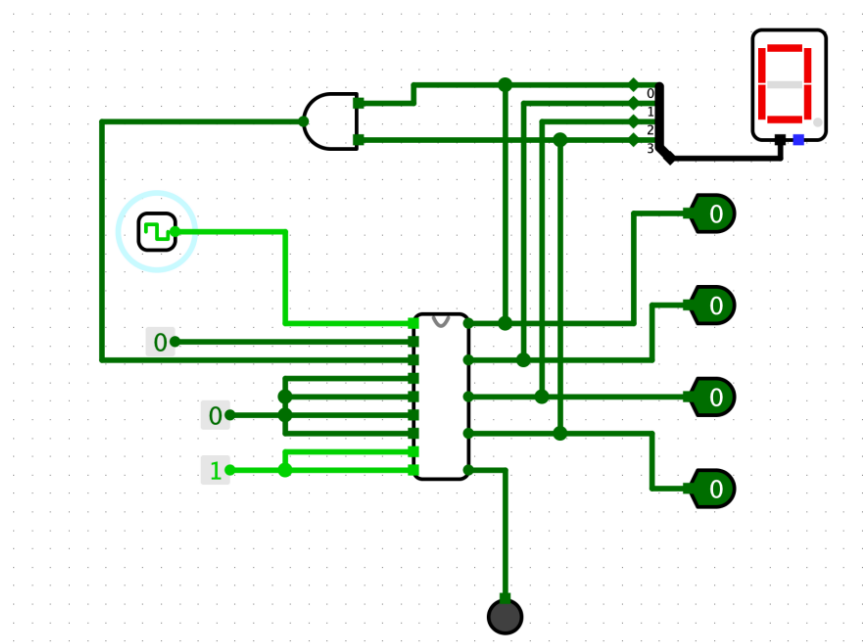
可以看到，在进入稳定状态后，十进制清零计数器可以在 0-9 之间进行周期性计数，十进制置位计数器可以在 6-15 之间进行周期性计数。

然后对电路进行仿真检验，即使电路在多个时钟周期中不断进行计数，重点观察临界点输出数值的变化。

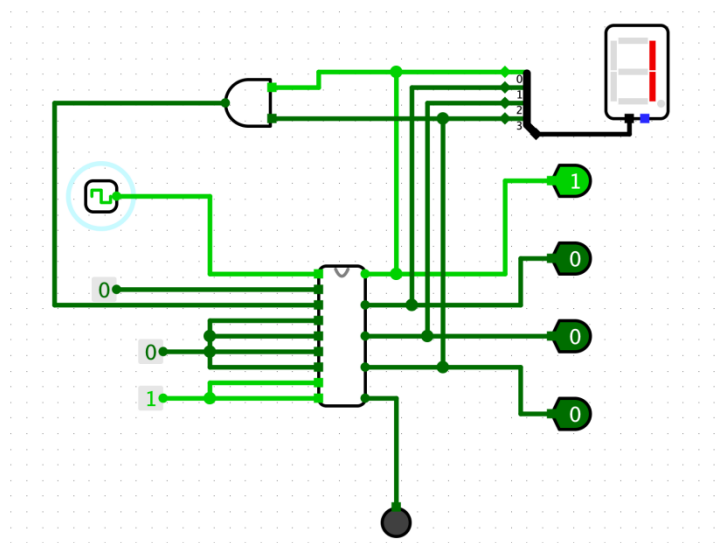
十进制清零计数器：



此时在时钟的上升沿计数器输出数值变为了 9。

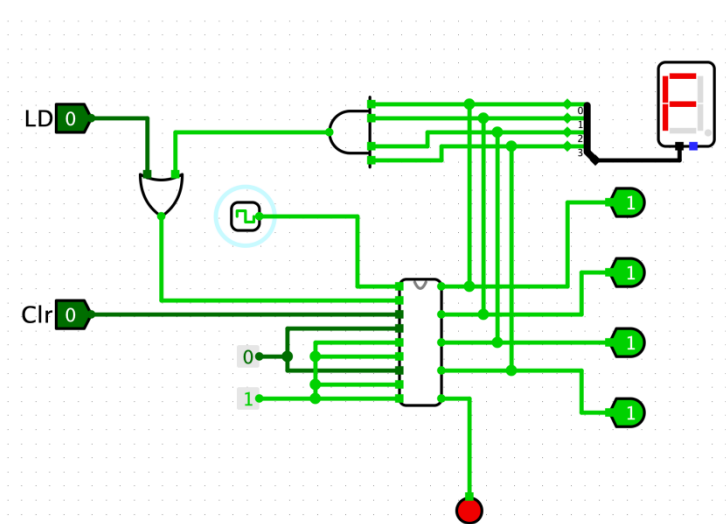


下一个时钟信号上升沿到来后，输出数值变为了 0。

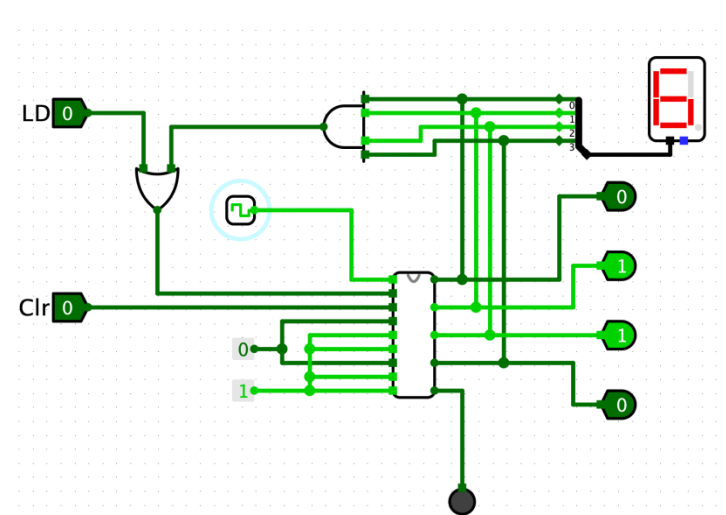


来到下一个上升沿，可以看到计数器又继续逐位向上计数了。

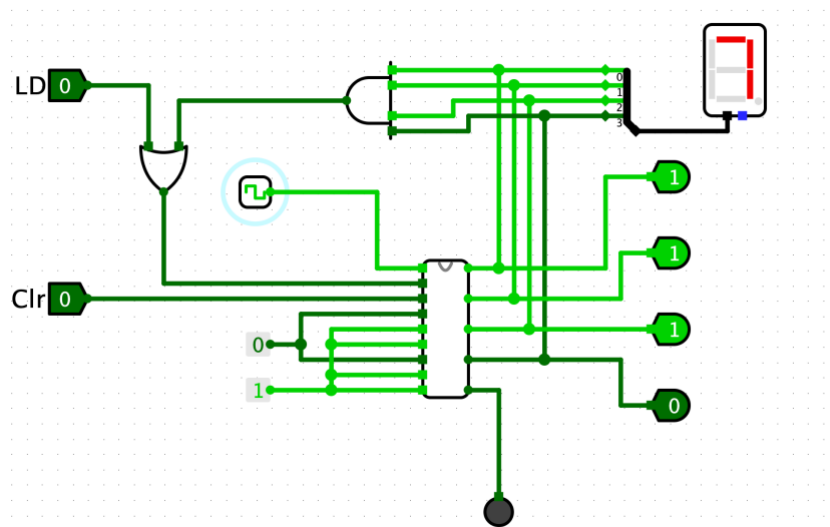
十进制置位计数器：



此时经过时钟的上升沿计数器输出数值变为了 0xF，且 LED 灯亮。



下一个时钟信号上升沿到来后，输出数值变为了 0x6，LED 灯熄灭。



来到下一个时钟周期，可以看到计数器继续向上计数。

可以发现，所设计的电路可以按照所预想的功能运行，一定程度上验证了所实现电路的正确性。

1.3 实验错误与原因分析

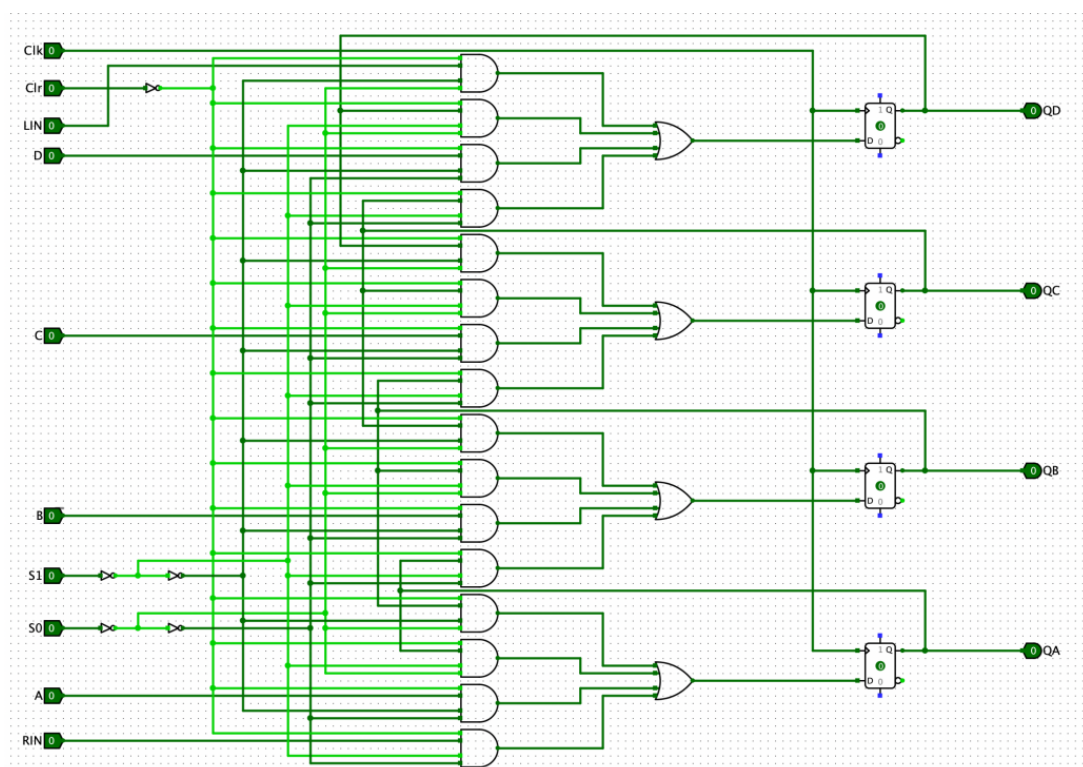
本实验按照课程所学知识和实验手册所提供的信息即可实现要求的电路功能，没有遇到什么特殊的错误。

2. 生成重复二进制序列

2.1 实验操作流程

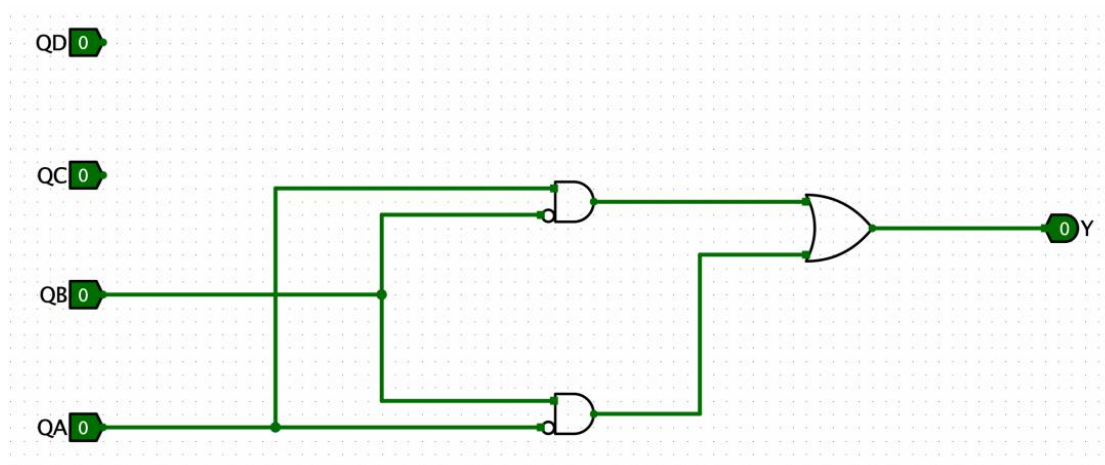
要实现生成重复二进制序列“000100110101111”的电路，为简化实现，首先要实现一系列子电路，包括 4 位通用移位寄存器和所要求的二进制序列对应的反馈电路，然后在此基础上实现所期望的电路。

4 位通用移位寄存器的电路原理图如下图所示：



该电路的输入端有 A、B、C、D 四个置位输入，一个 Clk 时钟信号输入端，一个清零端 Clr，一个左移数据输入端 LIN，一个右移数据输入端 RIN，以及最关键的两位功能选择端 S1S0。而四位输出 QA、QB、QC、QD 就代表着此时寄存器中的值。借助一系列门电路和 4 个上升沿触发的 D 触发器可以实现一个如上图所示的 4 位移位寄存器。

由于在最终的总电路中需要将反馈电路的输出作为数据输入端的输入，因此反馈电路的逻辑需要根据所要生成的二进制串来确定，其最终化简出的电路原理图如下：

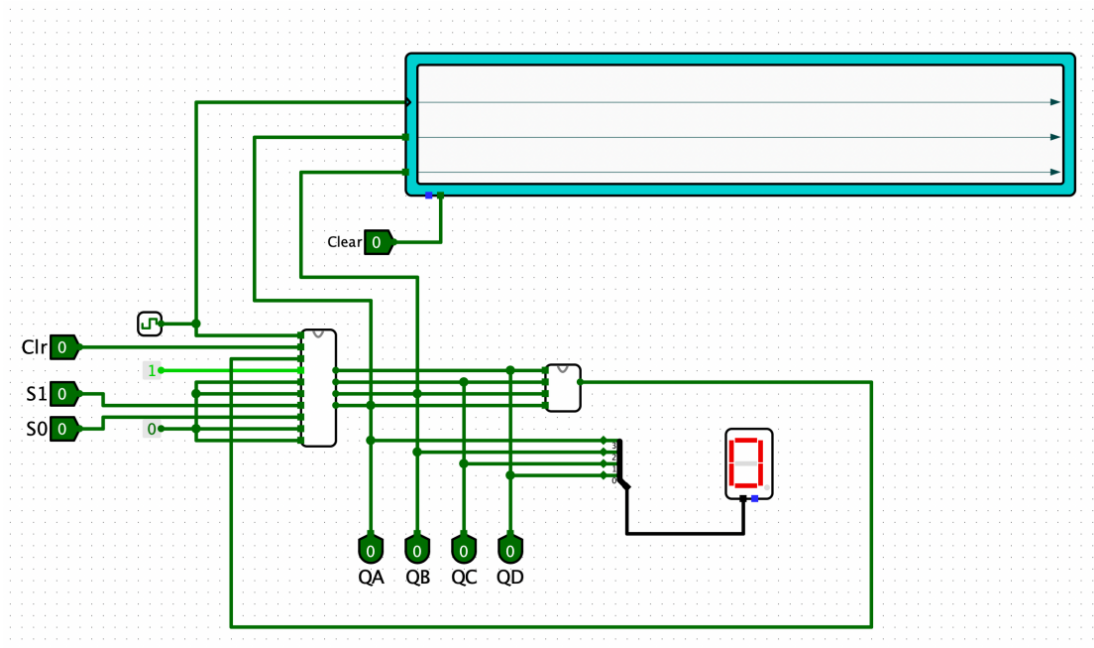


在确定反馈电路逻辑时，可以发现下一步移位寄存器期望得到的输入，即反馈电路的输出，实际上是将上一个时钟周期寄存器中的值作为输入通过反馈电路产生的，

因此只需要观察二进制串中的某一位及其前面紧邻的四位，遍历所有情况后借助卡诺图进行化简，就可以得到最终的逻辑表达式。

经过遍历可以得出反馈电路输出 Y 与输入（此处将 QA 、 QB 、 QC 、 QD 简写为 A 、 B 、 C 、 D ）之间的关系为 $Y = \sim AB \sim C \sim D + A \sim B \sim CD + \sim ABC \sim D + A \sim BC \sim D + \sim AB \sim CD + A \sim BCD + \sim ABCD + A \sim B \sim C \sim D$ ，通过卡诺图化简得到简式 $Y = \sim AB + A \sim B$ ，即上图所实现的电路。

最终的总体电路电路原理图如下所示：



该部分实验的实现为了方便最终对于所生成二进制串的观察，将 $QA \sim QD$ 俺从左到右的顺序依次排列，并采用左移的方式。为方便电路运行，将 A 、 B 、 C 、 D 输入端分别以常量 0、0、0、1 进行输入， RIN 由于不使用所以也直接置为常量 0。此外将清零信号 Clr 连接至寄存器的清零端，清零信号 $Clear$ 连接至数字示波器的清零端，选择端 $S1S0$ 连接至功能选择端，时钟信号 Clk 分别连接至寄存器和数字示波器。为方便观察和检验，将 QA 、 QB 连接到数字示波器。最后，同时将 4 位输出信号通过分线器连接到一个十六进制数码管，以观察输出的伪随机数。

实现电路后，对电路进行进一步的观察和检验。

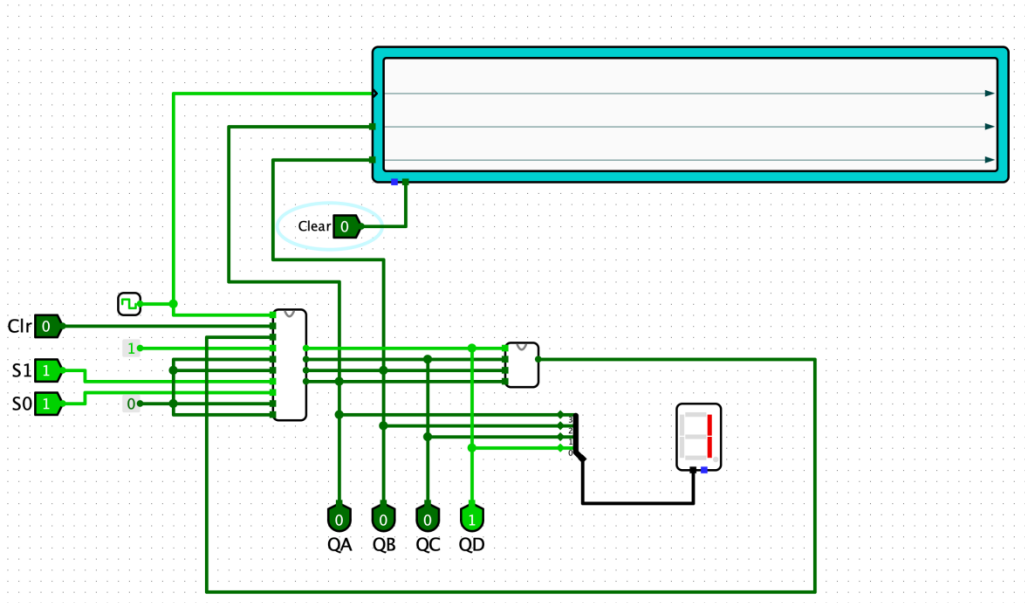
2.2 实验结果

该电路的输出变化对应表如下所示如下：

Function	CLR	S1	S0	QA*	QB*	QC*	QD*
Clear	1	x	x	0	0	0	0

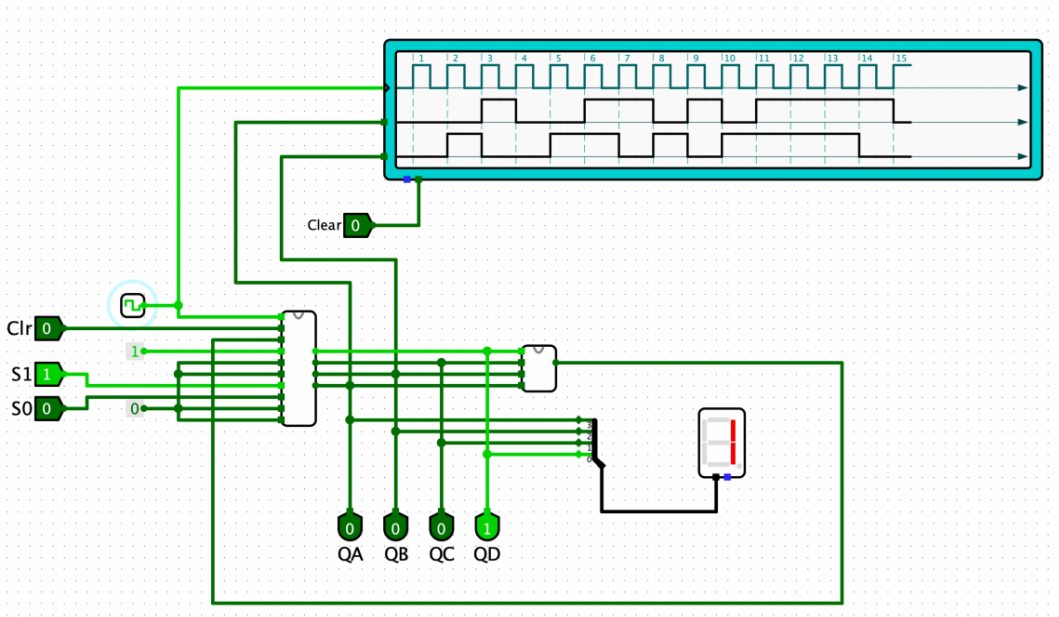
Hold	0	0	0	QA	QB	QC	QD
Shift right	0	0	1	RIN	QA	QB	QC
Shift left	0	1	0	QB	QC	QD	LIN
Load	0	1	1	A	B	C	D

接着对电路进行仿真检验，通过调整置位端将电路初始化为起始状态，然后在调整好电路后让时钟走过多个时钟周期，观察 QAQBQCQD 所显示的二进制串是否正确，并记录所生成的伪随机数。



先通过将 S1S0 置为 11 使电路的值被置为初始状态，然后通过将 Clear 置为有效清除这一步数字示波器上生成的波形。

然后将 S1S0 的值置 10 使电路进入左移状态，在多个时钟周期内观察输出结果。



观察上面的生成的波形中 QA 段，发现生成了 000100110101111 的序列，并且在
该序列结束后又回到了开始的 0 进行重复生成。生成的十六进制伪随机数序列记录为
124936DA5B7FEC8。仿真检验得到的结果符合功能预期。

通过上述的仿真检验，可以在一定程度上验证所实现电路的正确性。

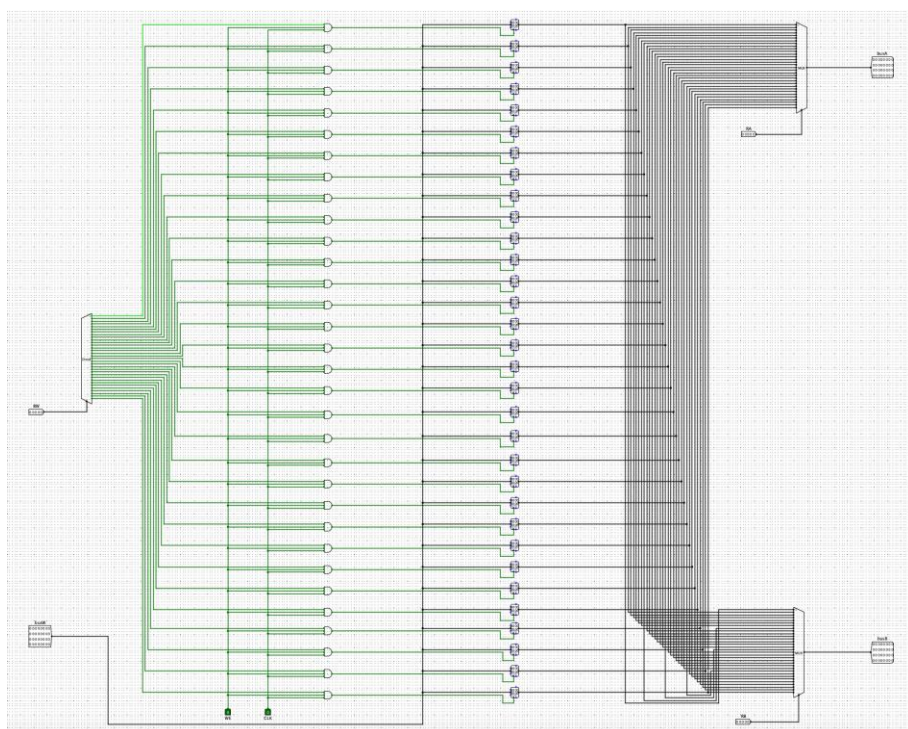
2.3 实验错误与原因分析

第一次尝试实现电路时由于误将 LIN 当作是“从左边移入”，将 RIN 当作是“从
右边移入”而导致判断错误，仔细观察实验手册中给出的信息可以发现，LIN 实际上
是指“电路进行左移时所移入的输入”，与之前所想的恰恰相反，此时的 LIN 在电路
图象上是由右边移入的！实际上也很好理解，左或右的移入方向根据电路摆放的不同
本身就是一个不确定的因素，而左移或者右移操作根据 S1S0 的取值是确定的。导致该
错误的主要原因是没有认真考虑清楚实验手册所提供的信息就盲目根据主观臆断进行
实现，实际上应当在充分了解信息和要求的情况下再谨慎思考实现方法。

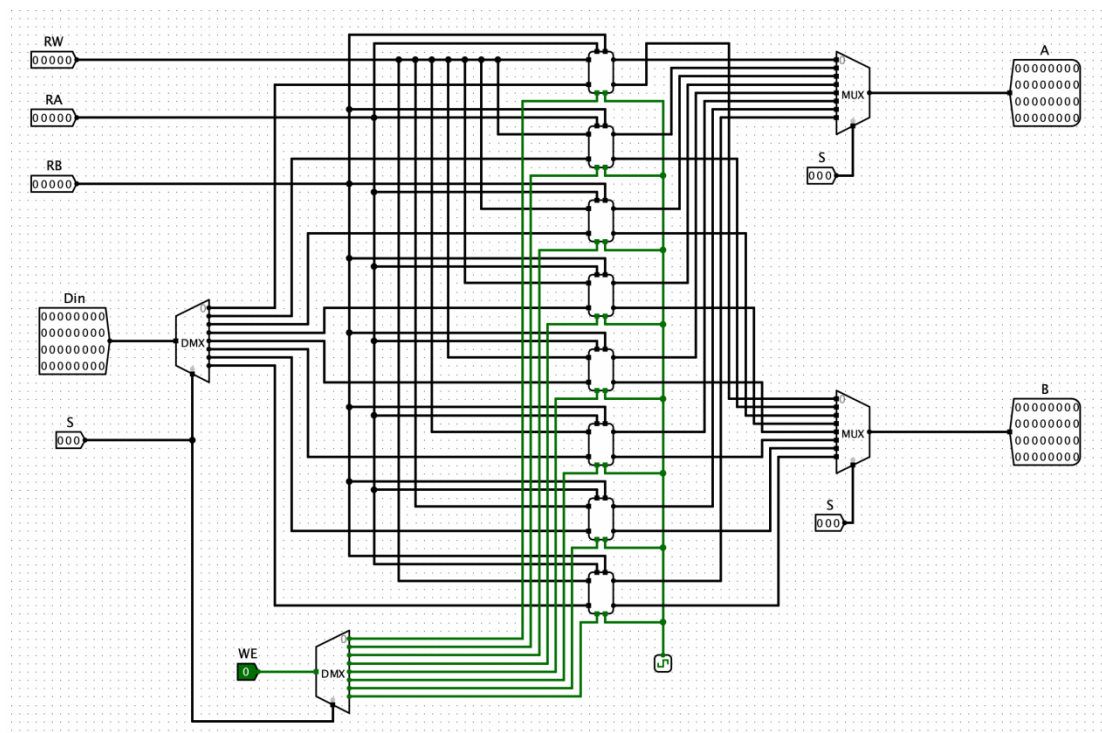
3. 含有 8 个 32 位寄存器堆 Regfile 的读写电路

3.1 实验操作流程

要构建一个含有 8 个 32 位寄存器堆 Regfile 的读写电路，按照实验要求，首
先要构造出 32 位的寄存器堆 Regfile 作为子电路，具体的电路原理图如下：



然后以寄存器堆作为封装好的子电路，可以实现最终的功能电路：



在电路实现完成后查看实验结果，对电路进行进一步的观察与检验。

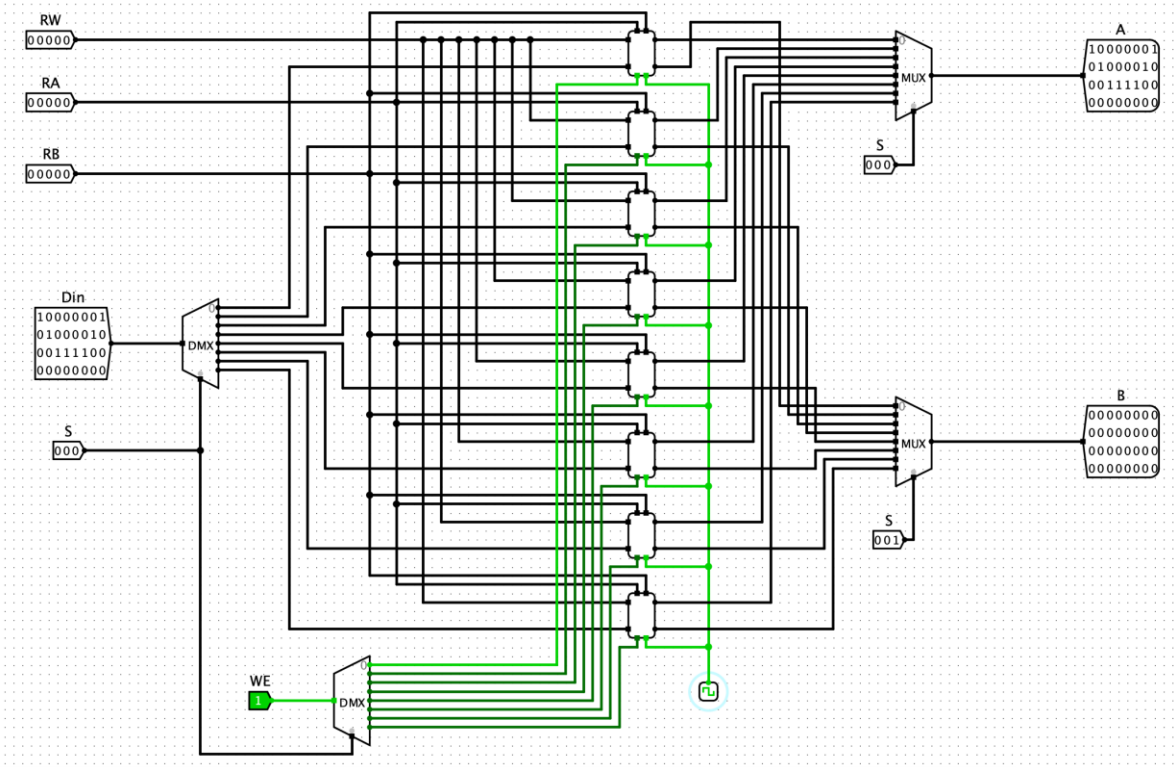
3.2 实验结果

下面列出所实现的读写电路的输入输出对应表:

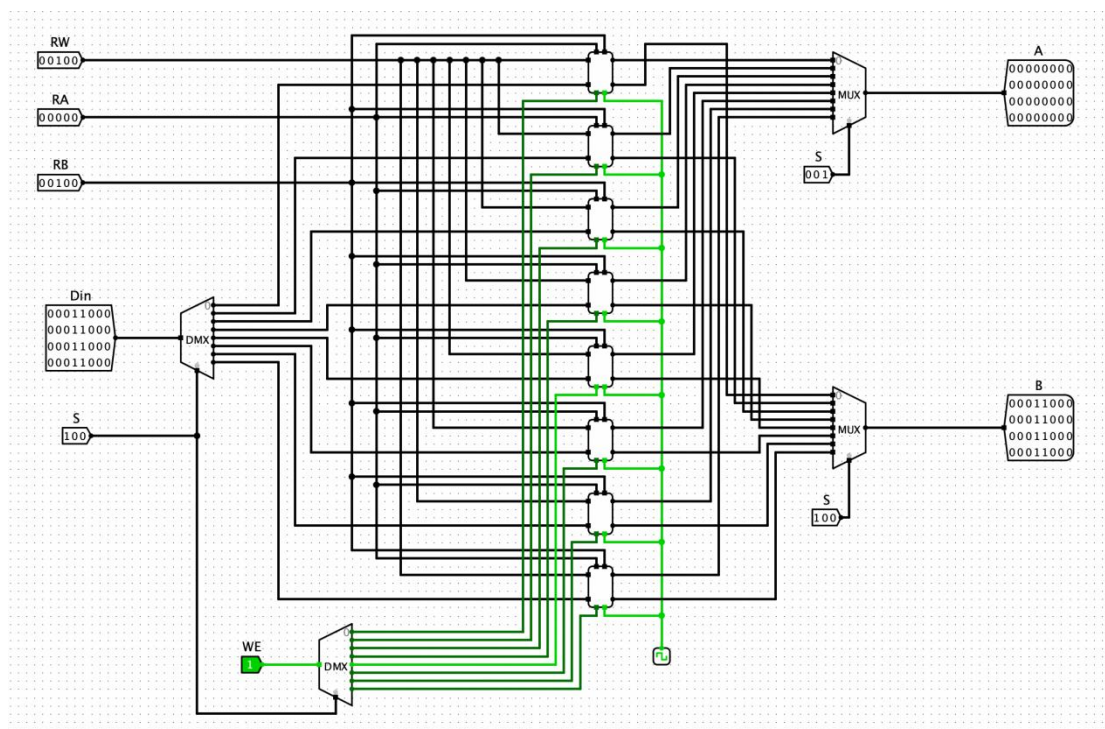
S(W)	WE	RW	Din	S(A)	RA	S(B)	RB	A	B	寄存器
x	0	x	x	sa	ra	sb	rb	第 sa 个 寄存器堆	第 sb 个 寄存器堆	不发生变 化

								的第 ra 个寄存器 中的值	的第 rb 个寄存器 中的值	
sw	1	rw	din	sa	ra	sb	rb	第 sa 个 寄存器堆 的第 ra 号寄存器 中的值	第 sb 个 寄存器堆 的第 rb 号寄存器 中的值	第 sw 个 寄存器堆 中的第 rw 号寄存器 值 变 为 din

然后挑取合适的输入样例对所实现的电路进行检验。



WE 为 1 时，此时在时钟的上升沿输入端的值被赋给 0 号寄存器堆的 0 号寄存器，此时 A 输出 0 号寄存器堆的 0 号寄存器的值，可以看到与刚刚的输入相同，而 B 输出 1 号寄存器堆的 0 号寄存器的值，可以看到由于没被赋值，其输出值还为 0。

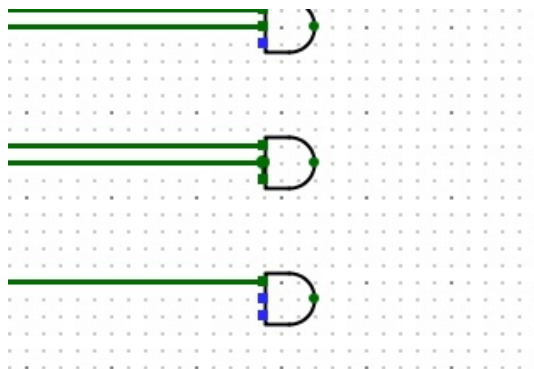


WE 为 1 时，此时在时钟的上升沿输入端的值被赋给 4 号寄存器堆的 4 号寄存器，此时 A 输出 1 号寄存器堆的 0 号寄存器的值，可以看到由于没被赋值，其输出值为 0，而 B 输出 4 号寄存器堆的 4 号寄存器的值，可以看到其输出值已经和输入保持一致，即对应寄存器中此时的值。

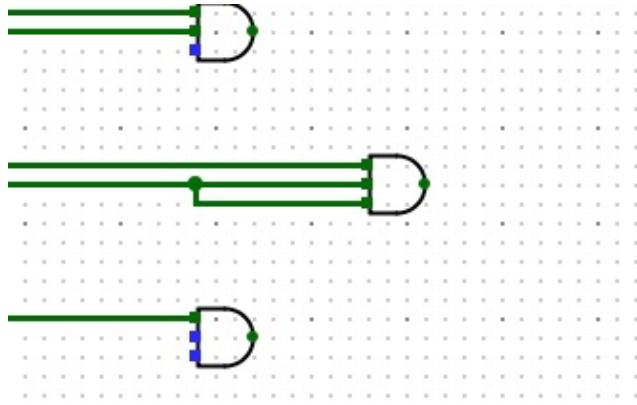
经过上述检验，任意挑选样例进行仿真检验所得出的结果都符合预期，可以在一定程度上验证所实现电路的正确性。

3.3 实验错误与原因分析

由于这部分实验涉及到的元件数量多，排布的方式较为密集，因此在连线时就会非常容易出错，而且较难发现。



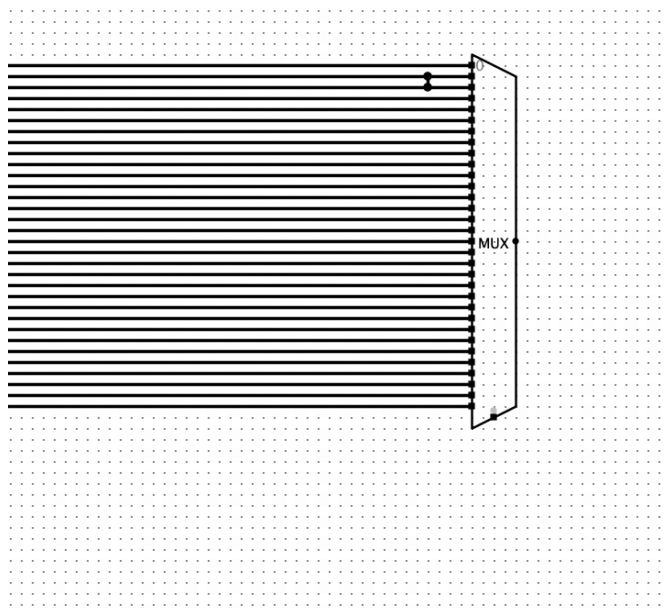
譬如上图，虽然此时能从颜色上发现有些许不同，但实际上放在全图视角来看这一部分元件将变得非常小，这一错误就是难以发现的了。



将元件向右拉，此时可以清楚地看到连线上发生了错误。

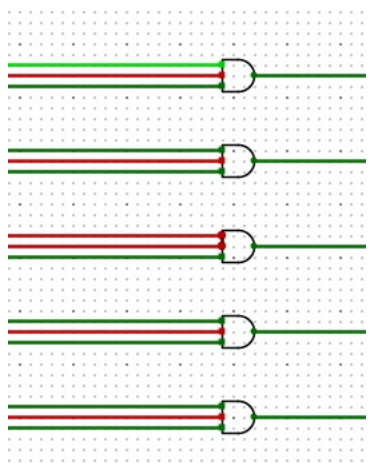
实际上可以延伸的是，这一方法也可以作为一种预先排查的手段，来检查是否有类似的连线错误。因为单独两条线之间出现类似的错误即使在较小的尺度下也可以发现，但很难发现的就是这种元件接口处有线条重合的地方。

下面是出现的一些类似错误，一不小心就会犯错：





将上面这张图放大来看：



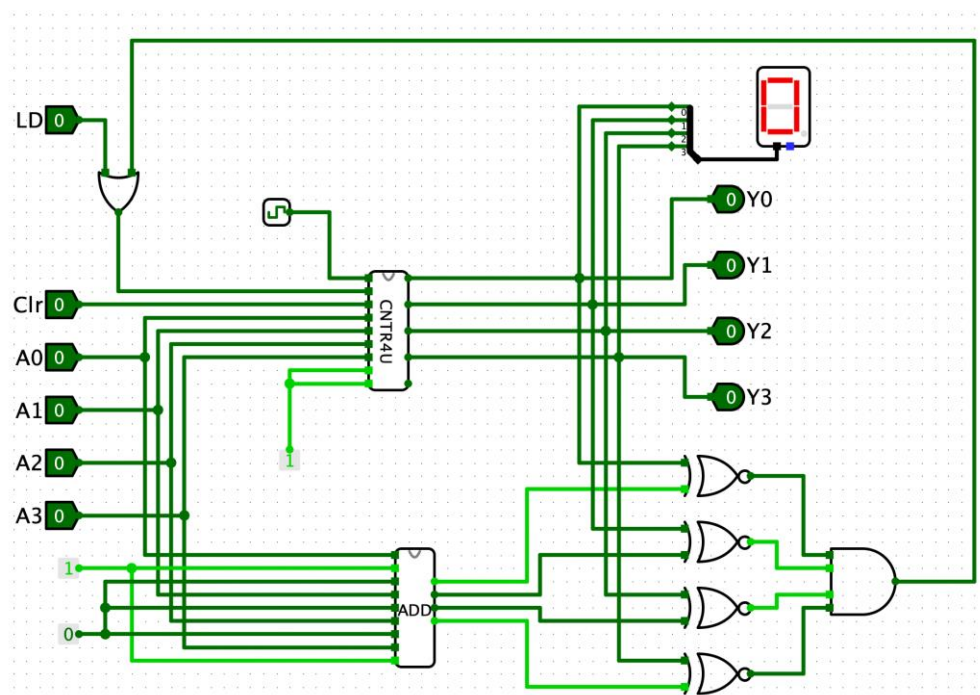
然后能够定位错误，仍然是类似的连线错误，可以看到在实验进行过程中是经常发生的，积累了丰富的错误素材。

4. 思考题

1. 如何利用 CNTR4U 实现从任意初始值开始的 10 进制计数器？

对于 0x0-0xF 之间的任意一个初始值 a，都可以通过 4 位置位值输入端将其作为初始值，并通过将 LD 置为 1 在时钟信号的上升沿使计数器的计数值被置为初始值。然后将 LD 置为 0 后计数器就可以进行十进制循环计数了。若想调整循环计数的初始值（循环范围），则按照与上面相同的步骤进行置位后开始计数即可。

实现的具体电路图如下：

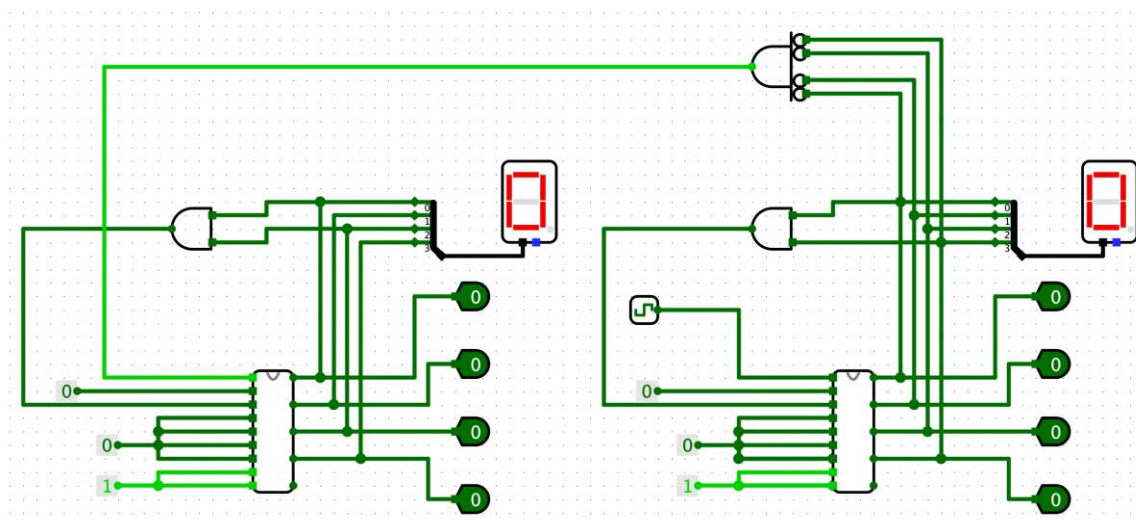


可以看到，为实现该电路还构造了子电路 CNTR4U 和 4 位串行加法器。通过加法器可以计算出当前初始值的情况下应当发生置位的情况下输出端的值，而右边连接到一个 4 输入与门的 4 个异或非门即是判断当前输出是否与上面计算出的值相同，若相同则与门输出为 1，反之则为 0。将与门的输出与 LD 信号通过一个二输入或门连接到 CNTR4U 的置位使能端。数码管上显示的即为计数器当前的值。

2. 如何用两片 CNTR4U 子电路设计一个 60 进制计数器？

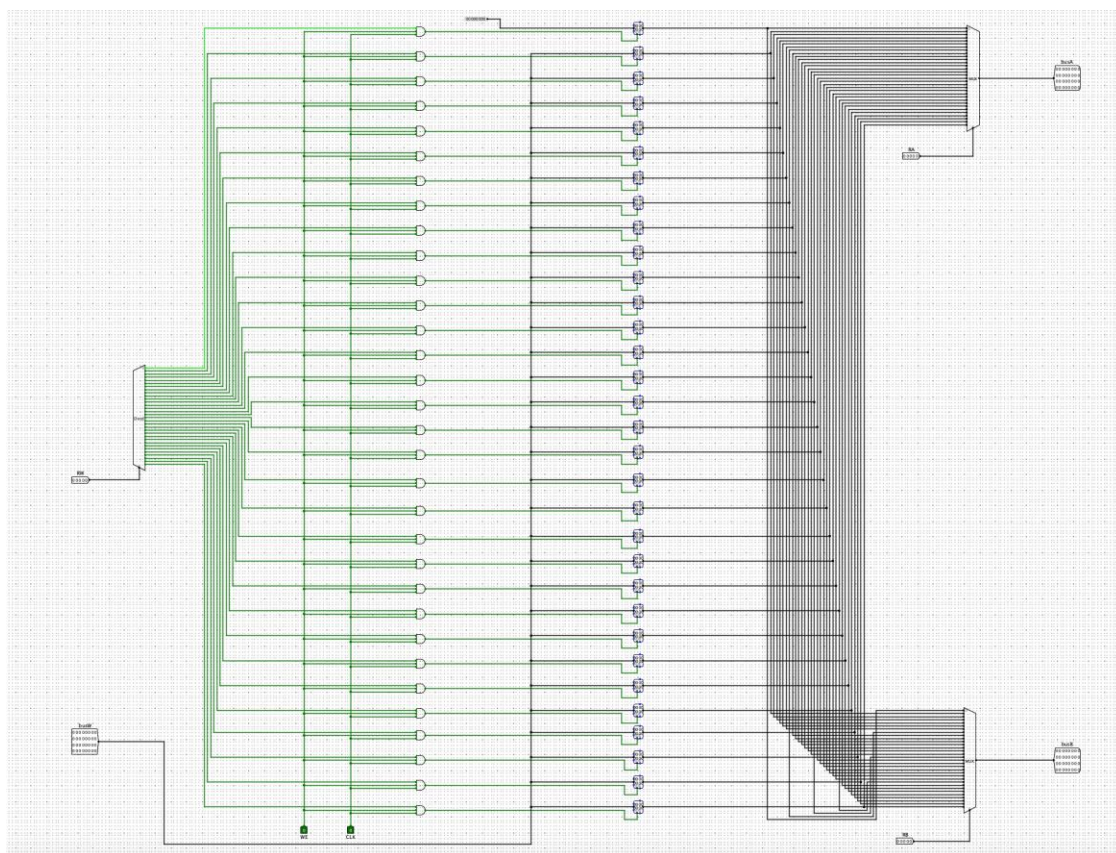
将两片 CNTR4U 一个用作十位的计数，一个用作个位的计数。两个计数器初值都为 0，个位在 0~10 之间进行循环计数，十位进行 0~5 之间的循环计数。每当时钟信号上升沿来到时，若此时个位要被清零，则用作十位计数的计数器也会迎来一个时钟信号的上升沿，完成向十位进位的模拟。

实现的具体电路原理图如下：

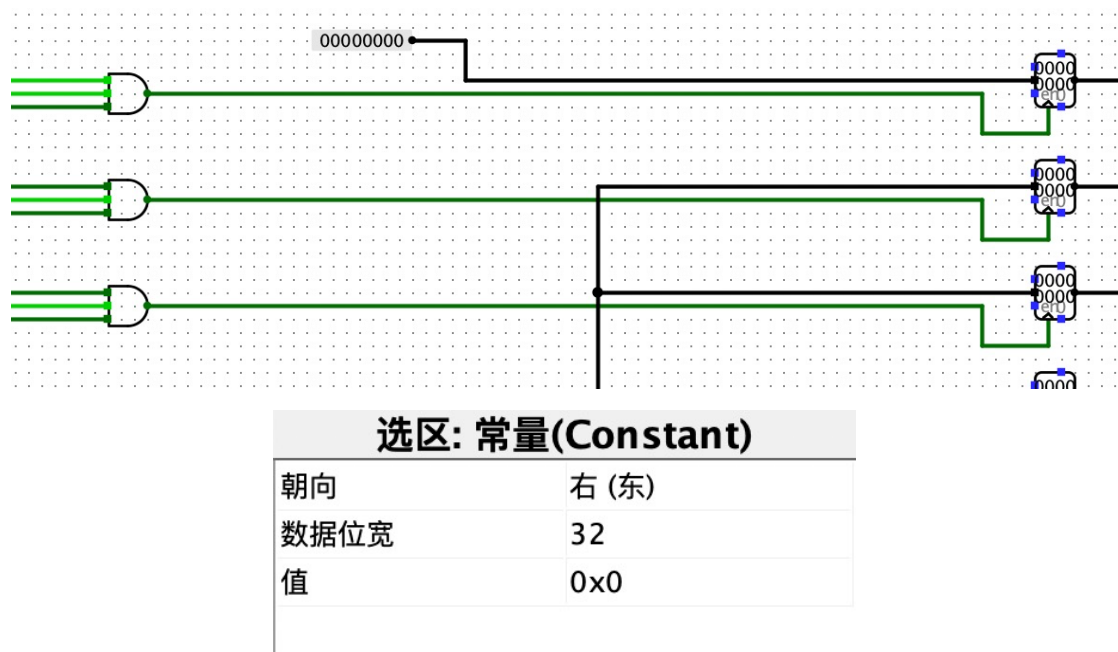


3.在寄存器堆中，如何实现 0 号寄存器始终存储数值 0？

在前面实验中实现的寄存器堆电路上进行调整，要想在无论何种操作下 0 号寄存器中的值都为 0，只需将寄存器的数值输入端输入值由原先的 busW 改为常量 0x00000000。这样不仅该寄存器初始值为 0，且无论怎样操作其被赋的值也只能是 0，调整后的电路原理图如下：



修改后的局部细节放大如下图：



4.如何用组合电路实现 4 位移位寄存器？

要用组合电路实现 4 位移位寄存器，参考移位寄存器的主要功能模式，由于对称性，不妨假设此时要实现的是左移移位寄存器，也就是要实现在当前存储的一个 4 位数据的基础上将其左移 1 位，并将要输入的 1 位数据填充到最低位上。

但在实现该电路的过程中我碰到了一些困难，经过总结和思考我分析了我碰到的主要困难，列举说明如下：

1.因为要实现移位寄存器的功能，因此下一步的输出就不仅与输入有关，还与当前寄存器中所存储的数据（现在所处的状态）有关，而这又恰好是符合书上对时序逻辑电路的定义的，即：所谓时序逻辑电路，其输出结果不仅取决于当前的外部输入，而且取决于系统所处的内部状态。因此，针对如何使寄存器电路不依赖当前的值而能准确获知下一步移位后应该得到的结果给我造成了困难。

2.另外，在时序逻辑电路中可以实现保持原状或接受当前的输入，但在组合逻辑电路中一端的输入无论变或不变都会直接反映到输出中，这就为输入的处理带来了困难，如何使寄存器电路能够实现单次接收的输入而不会再输入保持的时间内使输出无法保持而不断的变化也使我对该问题的实现变得困难。这次在 `circ` 文件中展示的之前一个难以进行下去的电路设计也是因为这个地方在最后一步不太容易走下去。

经过查阅相关资料，我了解到桶状寄存器是一种组合逻辑电路，是通过采用大量多路选择器实现的。

5. 实验总结

本次实验针对一系列时序逻辑电路进行了探究和设计，加深了对一系列典型时序逻辑器件如触发器、寄存器等，以及一系列复杂时序逻辑电路的理解。一方面，根据具体需求设计符合要求的时序逻辑电路的能力有所提升，另一方面，实验内容也是对课堂内容充分有效的补充。另外，在本次实验中还遇到了一些我难以解决的问题，虽然最终还是有一些困惑的地方暂时没能解决，但在尝试解决问题的过程中我仍然经过了反复的思考和尝试，查阅和了解到了许多我以前没有了解到的知识，作为思考题目它确实极大地提升了我在本次实验中的思考量，给我带来了许多收获，也让我可以以更全面的视角看待时序逻辑与组合逻辑间的区别和联系。