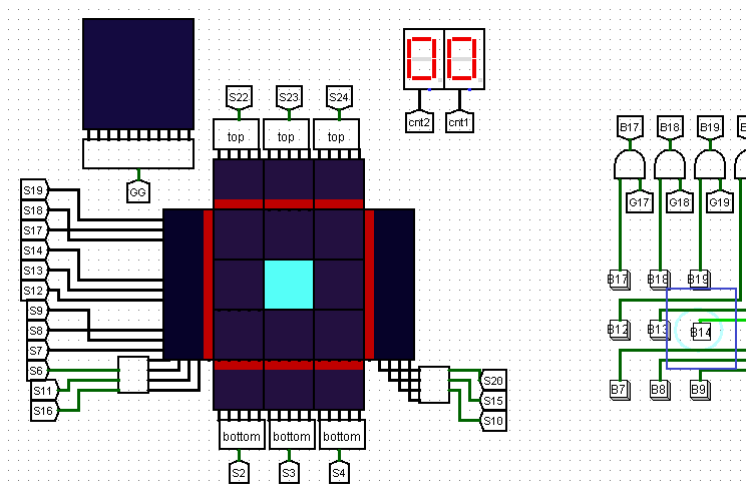
自主设计实验——消方块 PB19111661 柯景瀚

规则设计

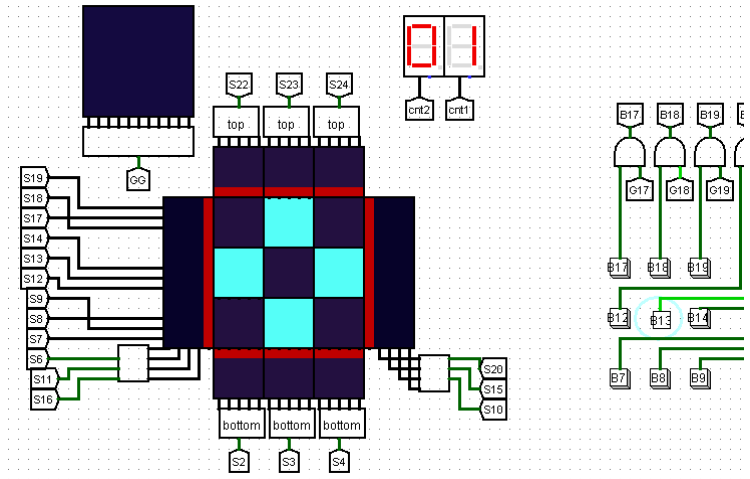
用Logisim电路构建一个9*9消方块的游戏，规则如下：

1. 只可以点亮的部分，且每点一次记一次数。点亮块时，亮块会灭，上下左右也会受到影响，亮的会灭，灭的会亮。如下演示，初始化后只可以点中间的button13(B13)。如果改变的信号碰到红线，则信号破碎，会有特效。

a.

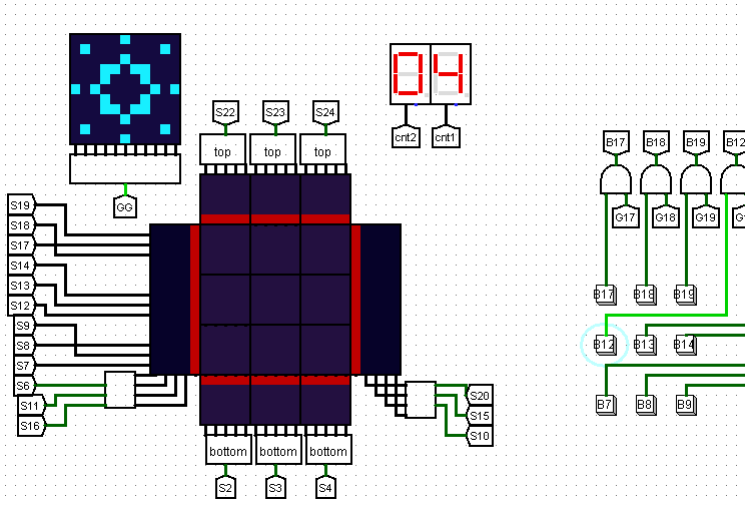


b.



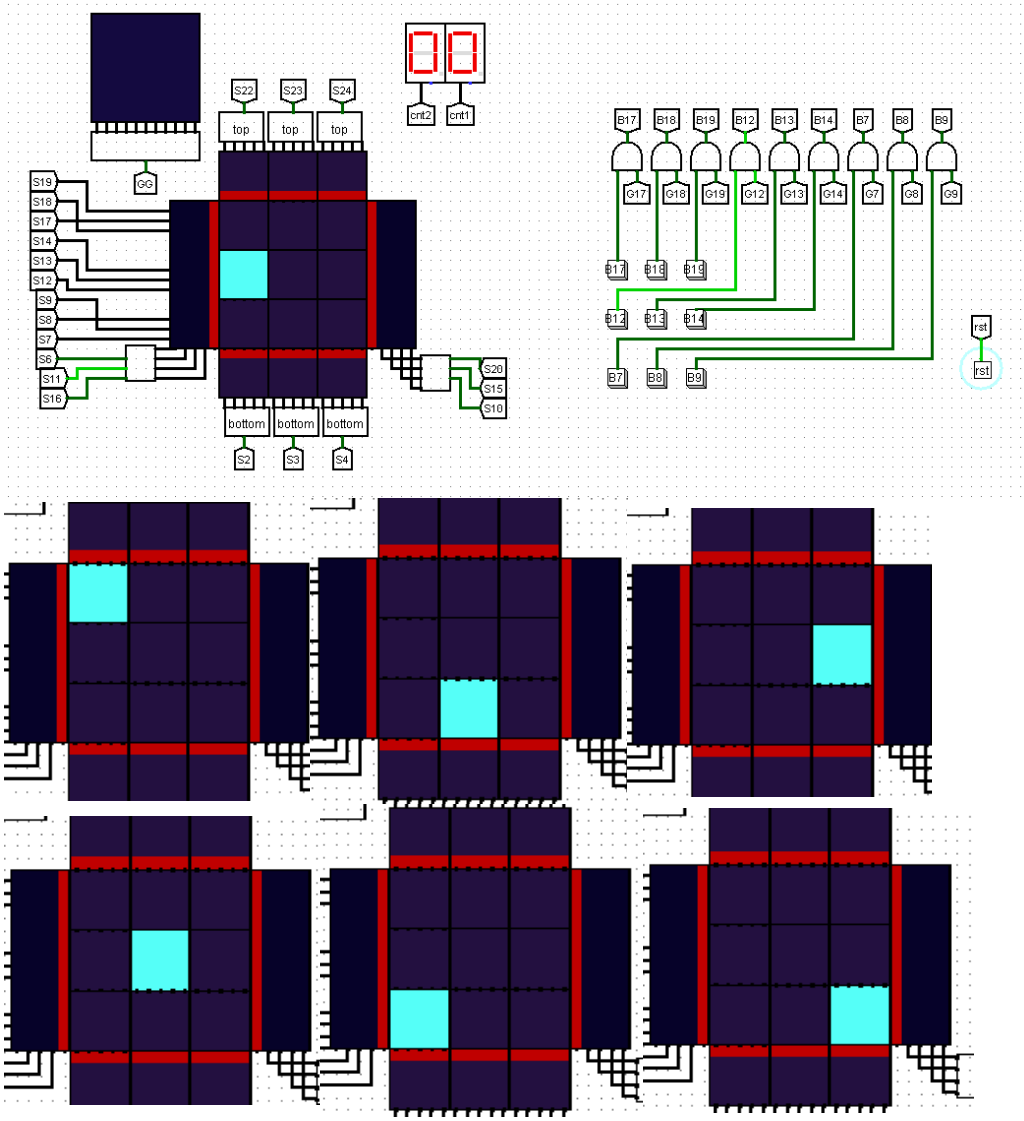
2. 使所有亮块都灭掉后游戏结束。

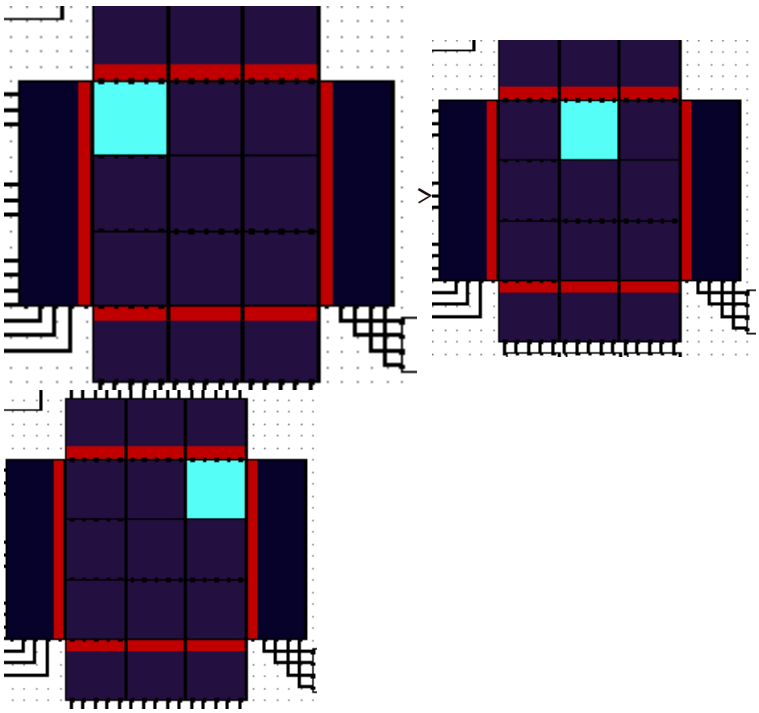
a.



3. 再按button reset(rst)，会随机出现初始方块，计数器归零，开始下一次游戏（当然进行的过程中也可以随意reset）。

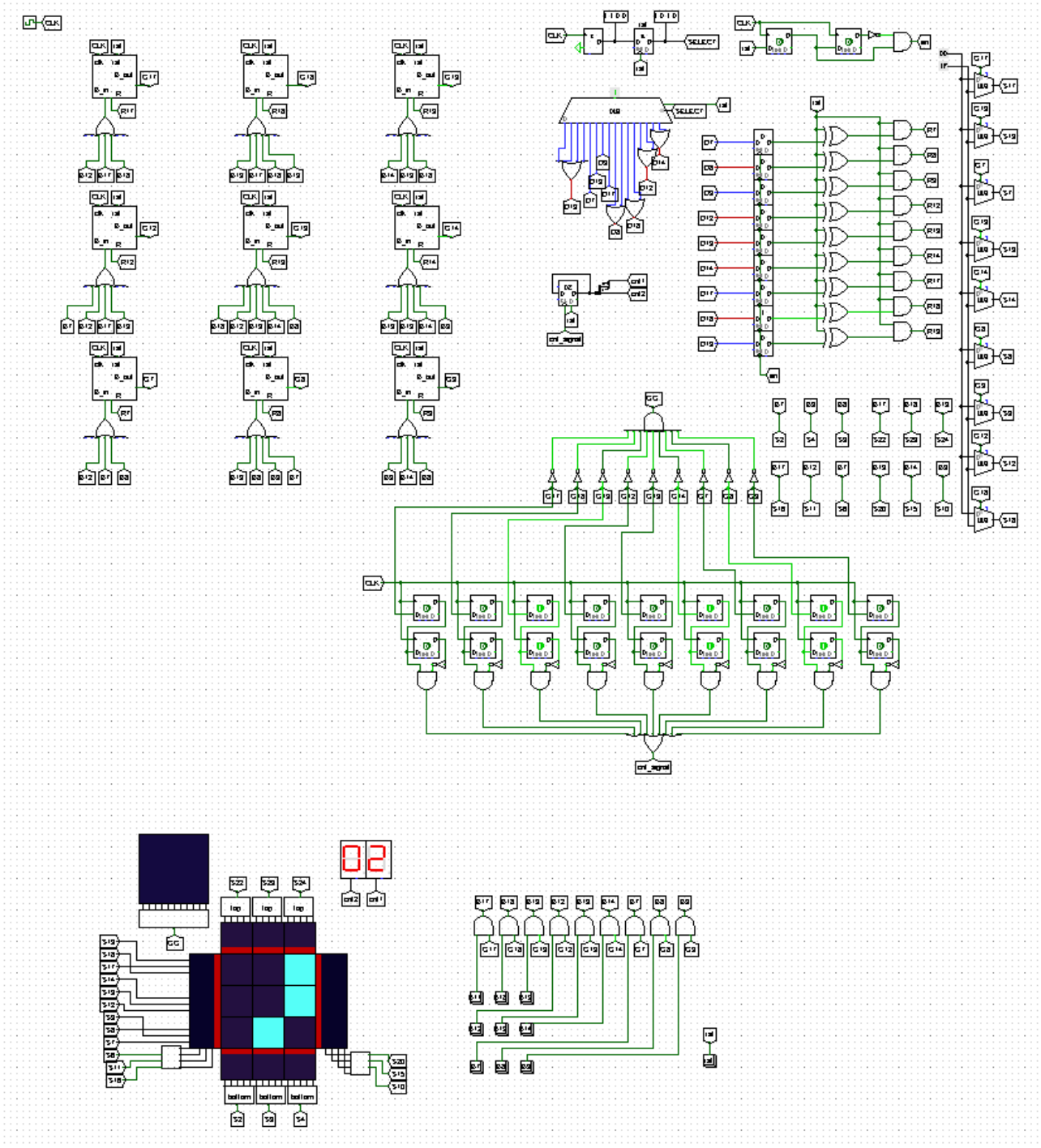
a.

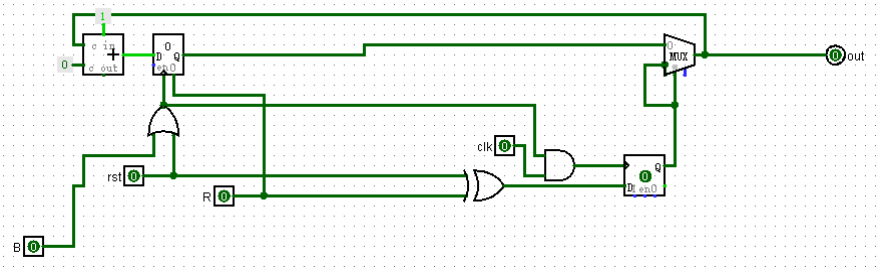




规则设计的实现（9个方块状态改变、随机复位、计数器）

该部分将描述下图所示电路的实现细节

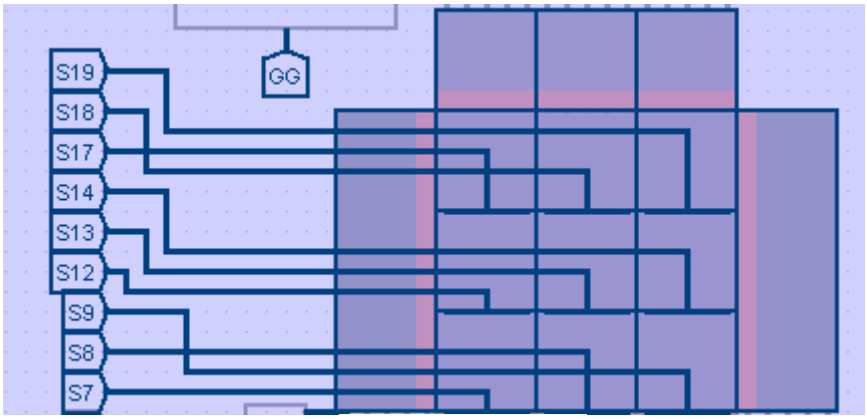




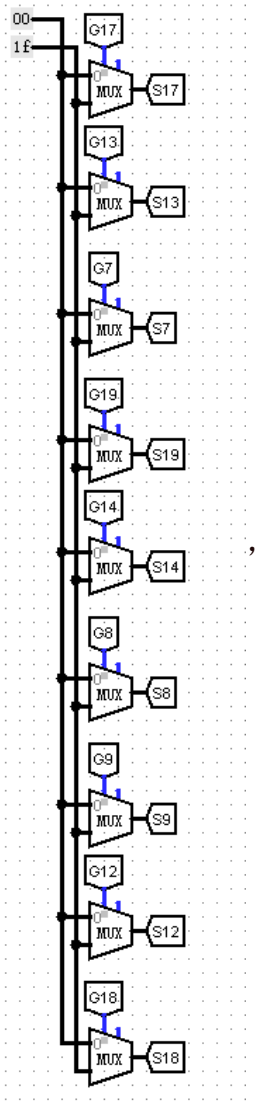
1. 按键与显示

a. 显示方面用5*5的LED Matrix表示一个方块，用S标签表示show的状态，并连接LED的五个接口，当有信号时，五个接口都ON，便得到亮块。

i. 线路连接如图所示



ii. S标签输出内容如图所示

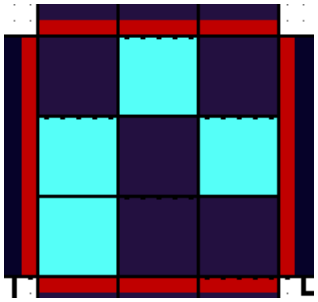


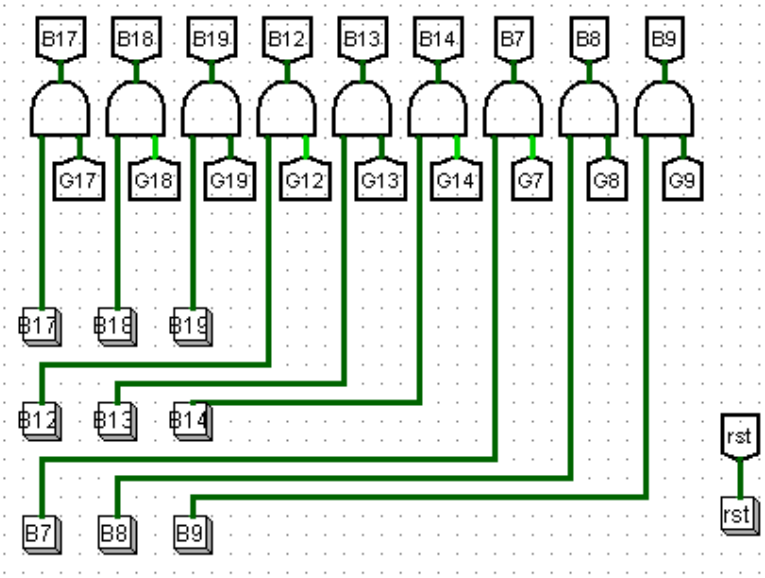
，如果G标签有信号就输出11111，没有就输出0。

b. G标签表示gate，它只有两个状态1和0，每一个方块都有一个gate，gate只有在按键有效或复位时才会改变状态。

i. 作用1：描述亮块。如上所述，当亮块亮的时候，是G标签使S标签选到11111，所以哪个方块是亮的，该方块的gate状态是1。如下图，G18、G12、G14、G7是亮的

i.

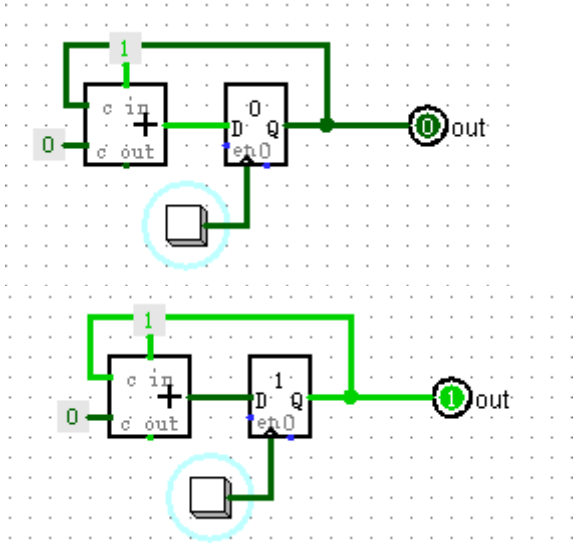




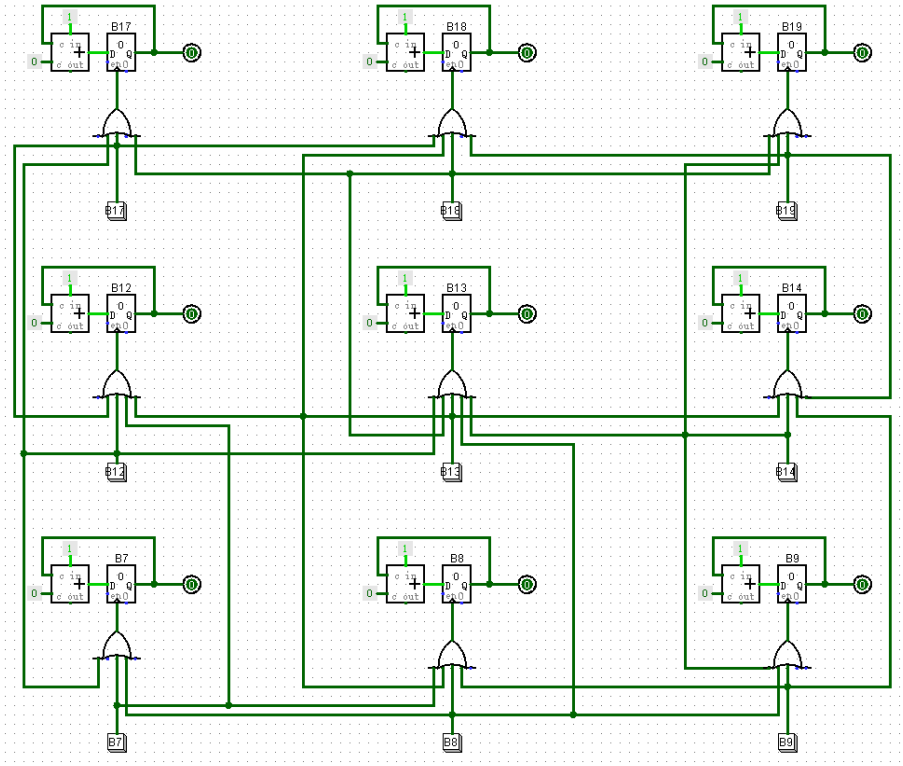
ii. 作用2：用于判断按键是否有效。如上图，除了G18、G12、G14、G7，其余按键按了是不起作用的，所以B标签表示的是button的有效信号。

c. B标签得到有效信号后，会导致上下左右和自身的亮灭。在不考虑随机复位的状况下，可以用一位加法器和寄存器描述状态的改变，如下所示

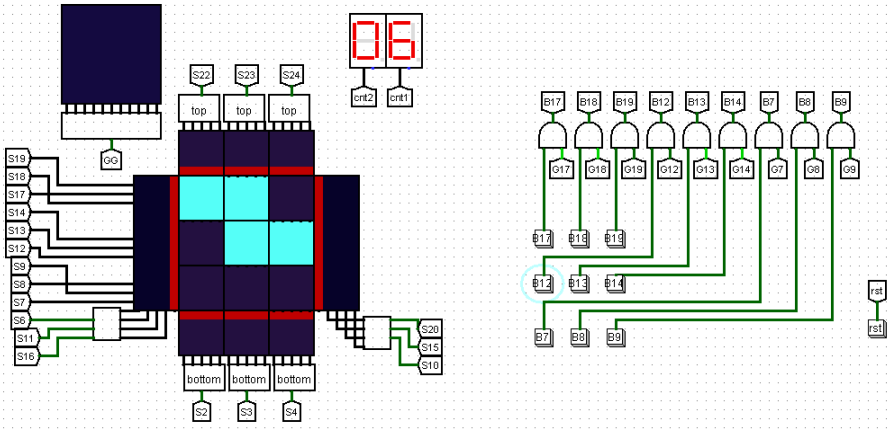
i.



ii.

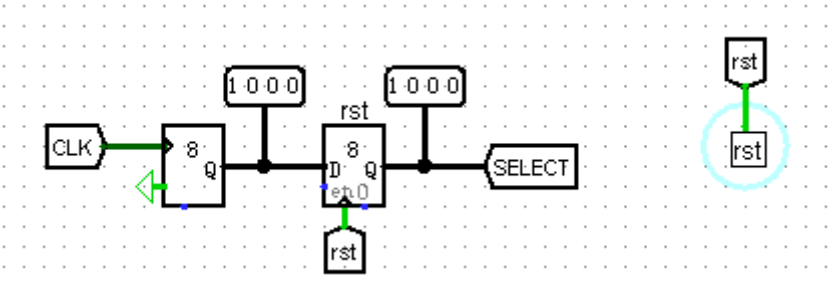


iii. 于是当按下B12后，B17和B13会亮(寄存器输出1，又存入0)，B12、B7会灭(寄存器输出0，存入1)。

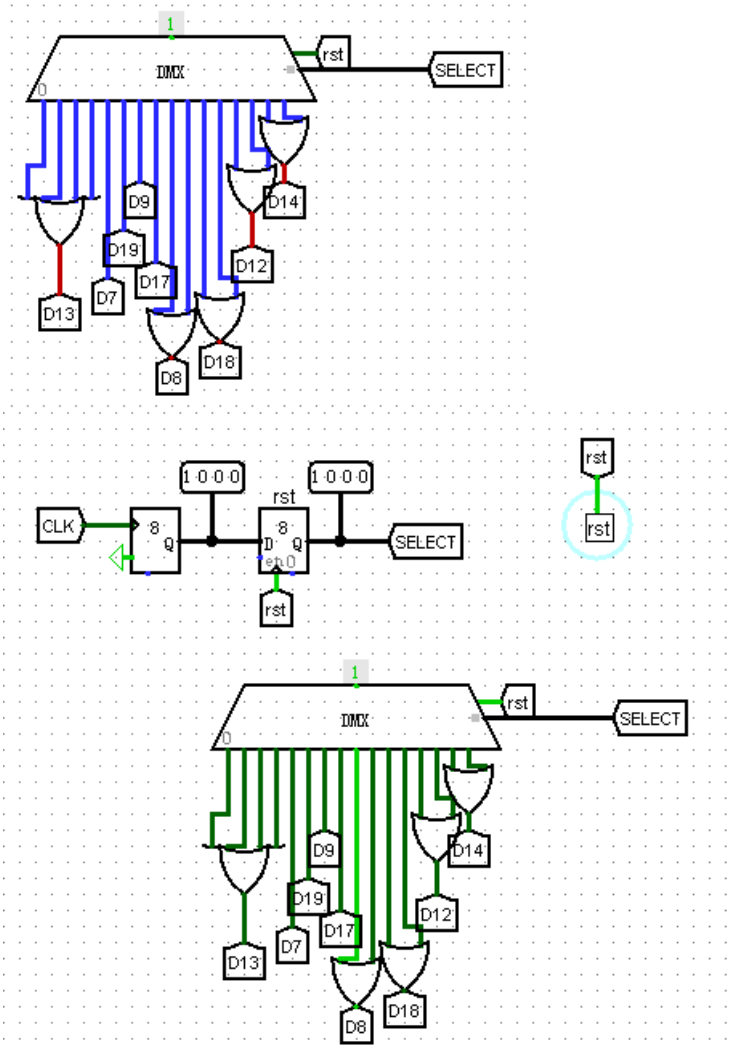


d. 随机复位的构建

i. 得到一个随机数用于选片，如图所示，用logisim自带的Random Generator在每个时钟下降沿沿产生一个随机数，当按下rst按键时，寄存器存入一个随机数并输出给select。

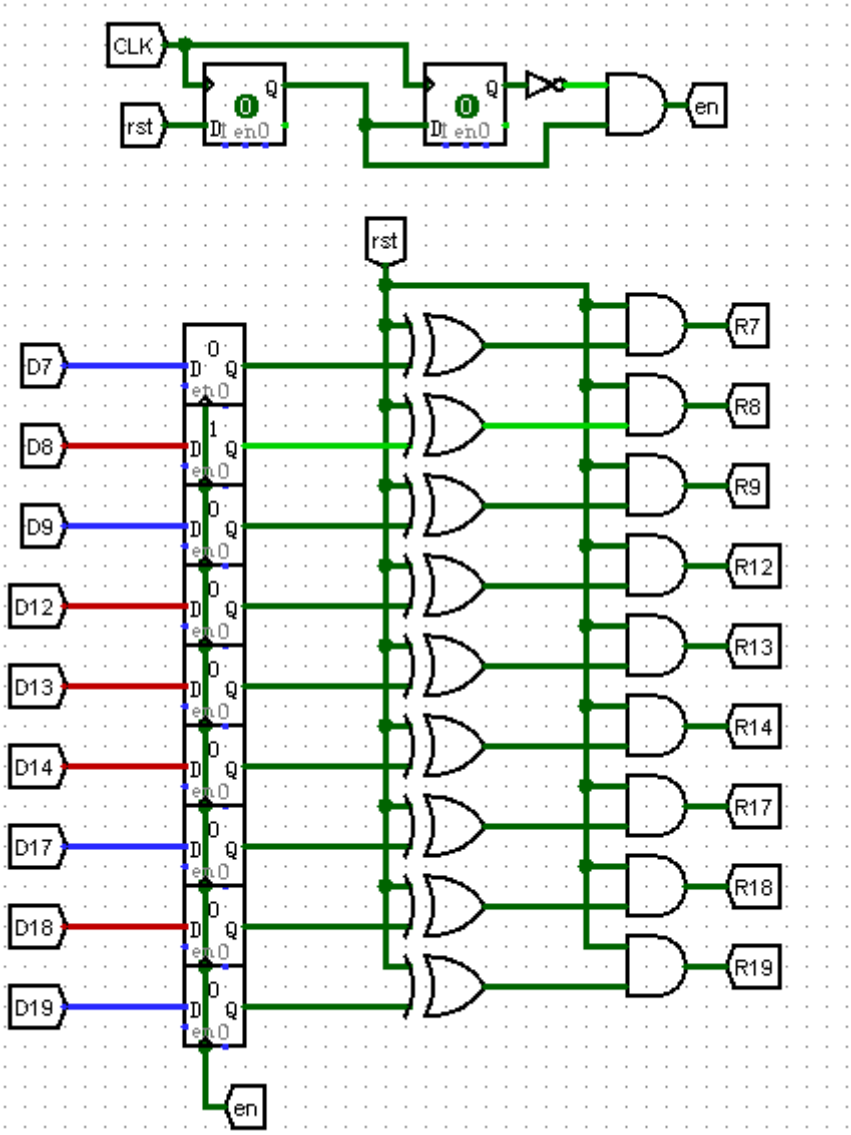


ii. 选片。用rst作为使能输入信号，即只有按着rst，才可以按照select信号进行选片。由于只有9个块，但可以产生0到15共16个随机数，所以可以调节下概率，使容易完成游戏的初始化方块多出现，如正中间编号13的方块。

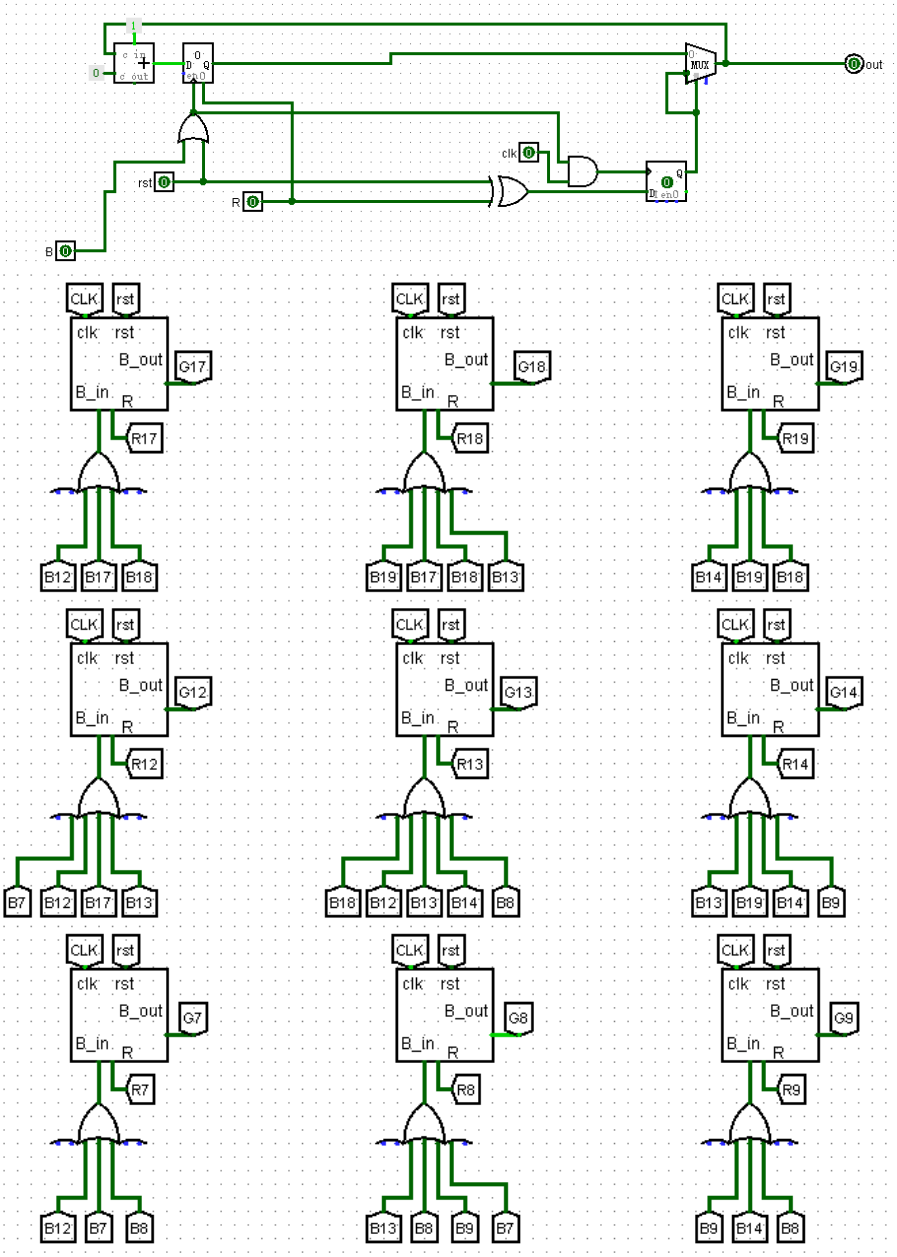


iii. 实现一个reset为1，其余8个reset为0。如图，当按下rst时，用一个去抖动操作的到en信号，en信号与rst相比，有效时长更短，用该信号可以使承载选片信息的D标签在rst有效时间内存入寄存器中，在下一个rst有效时间内输出，通过如下电路设计，可以得到R标签信号(reset)。可以看到，下一次rst有效时，R8为

0，其余为1。



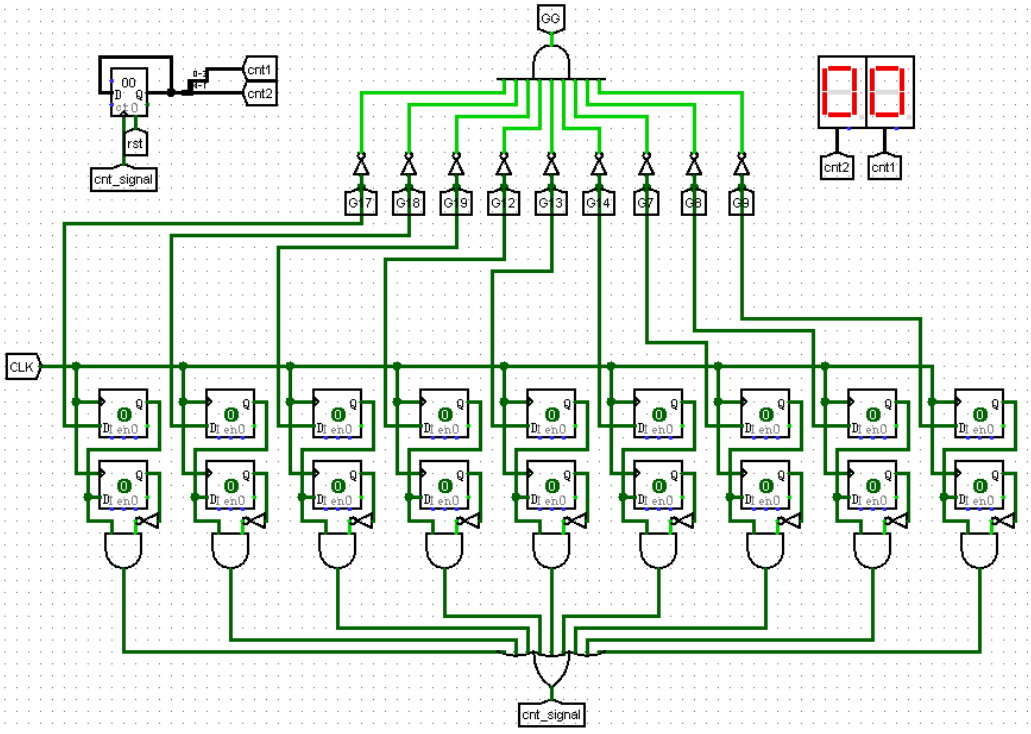
iv. 现在得到了每个片reset信号，根据信号判断：无论之前是什么状态R=1则灭(G=0)，R=0则亮(G=1)，且在松开rst键时，还要保持该状态。于是在之前的电路上稍微改动亿点，得到如下电路，并封装使用。



2. 计数器

a. 如图，取G标签的上升沿(去抖动)作为计数判断，cnt_signal触发寄存器循环，再用7 Segment Display展示信息。当所有的G标签都是0时，游戏结

束，GG标签(good game)将触发一段结束动画。

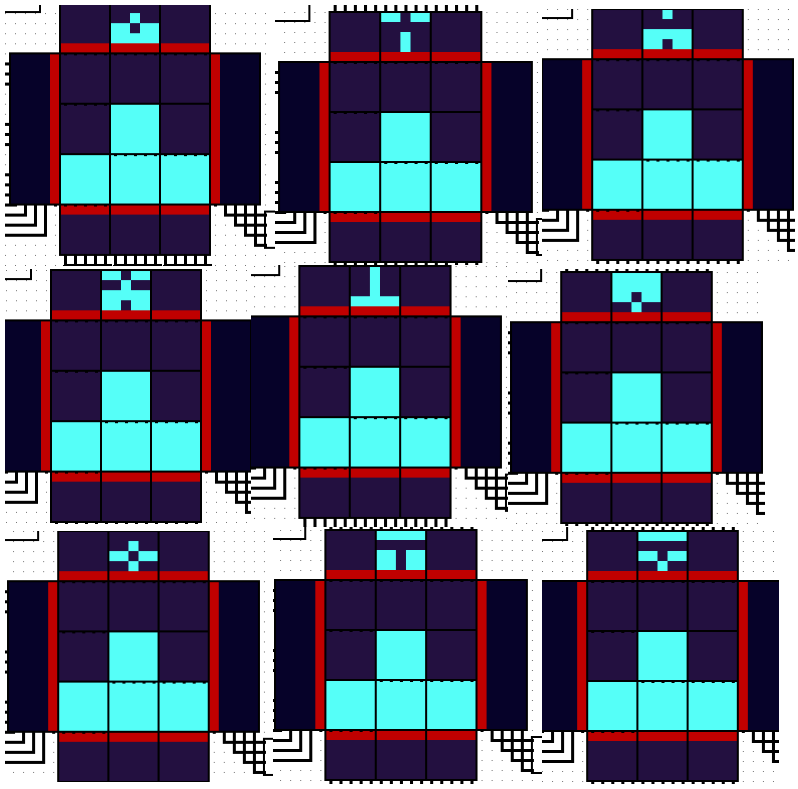


特效设计

1. 改变的信号碰到红线会触发破碎效果。

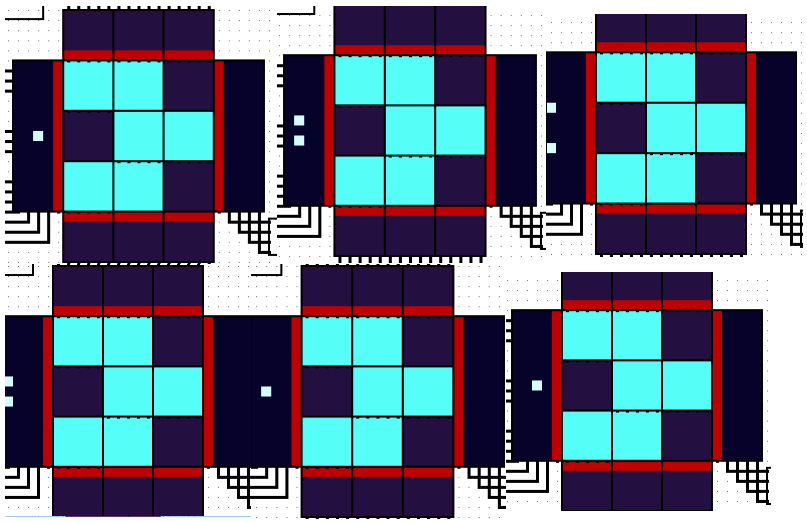
a. 随机数破碎效果，在上下红线设置。

i.



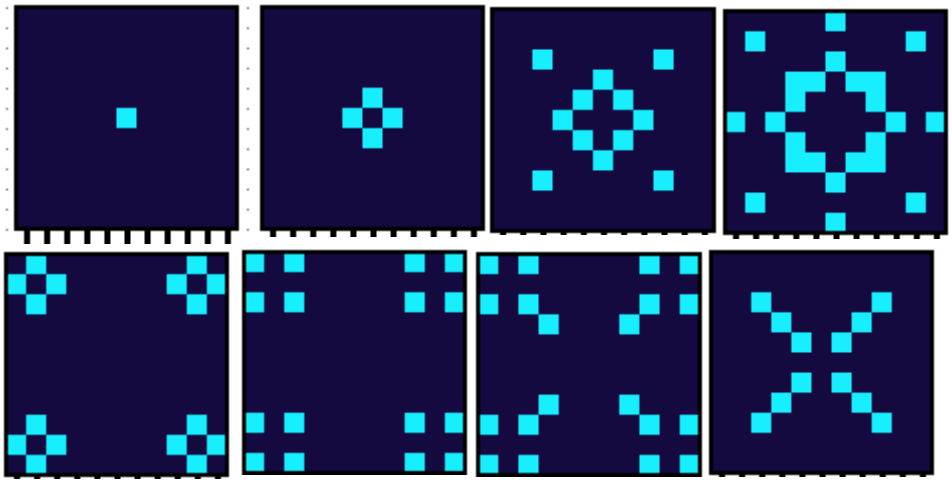
b. 自定义粒子破碎效果，在左右红线设置。

i.



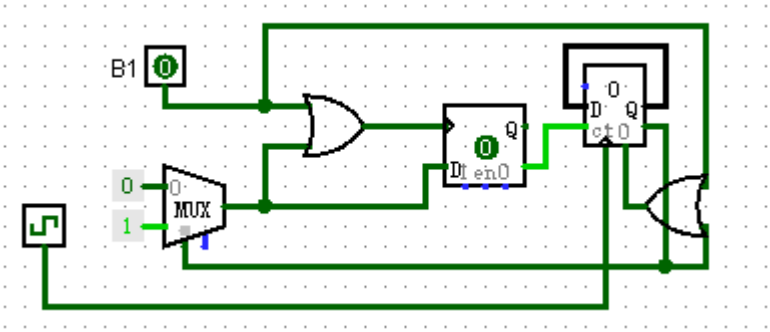
2. 结束时烟花动画（原本想画只猫舔爪的，但做完烟花后才发觉像素没分配够）。

a.



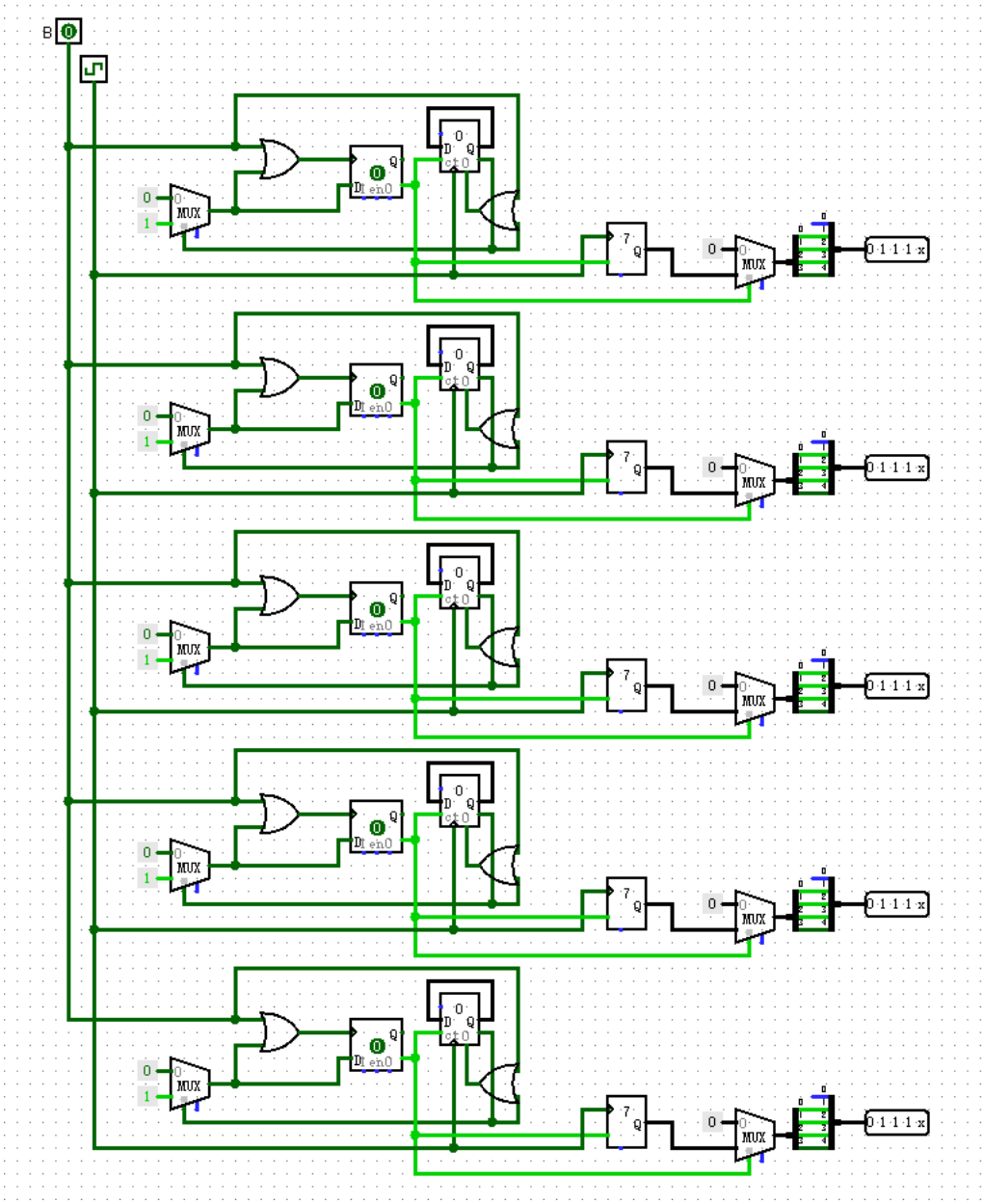
特效实现

所有特效实现都基于一个基本模块的实现。如图，该电路用了选择器、D触发器、计数器，实现了：B1上升沿时会触发计数，计数完一轮后，该模块回到初态并自动锁住不再改变状态。 为了保证特效的可观赏性质，每个特效的CLK有调频率。

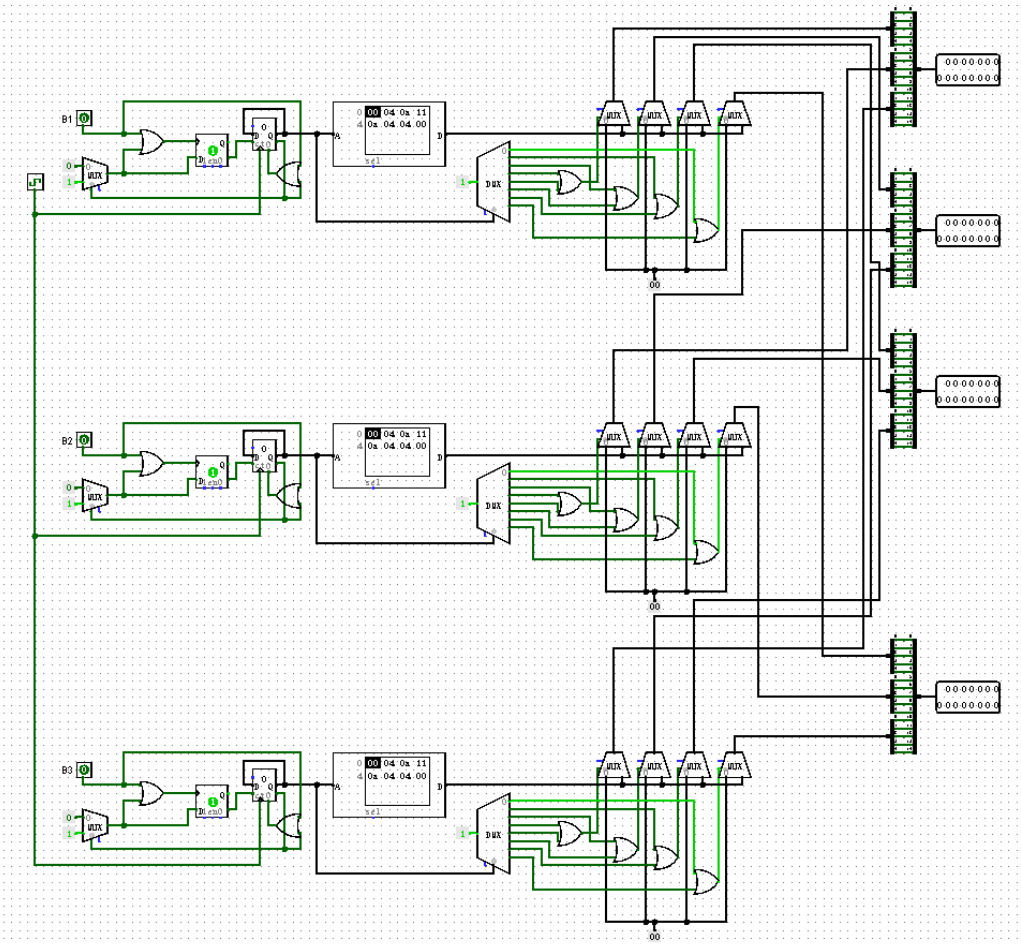


1. 破碎特效

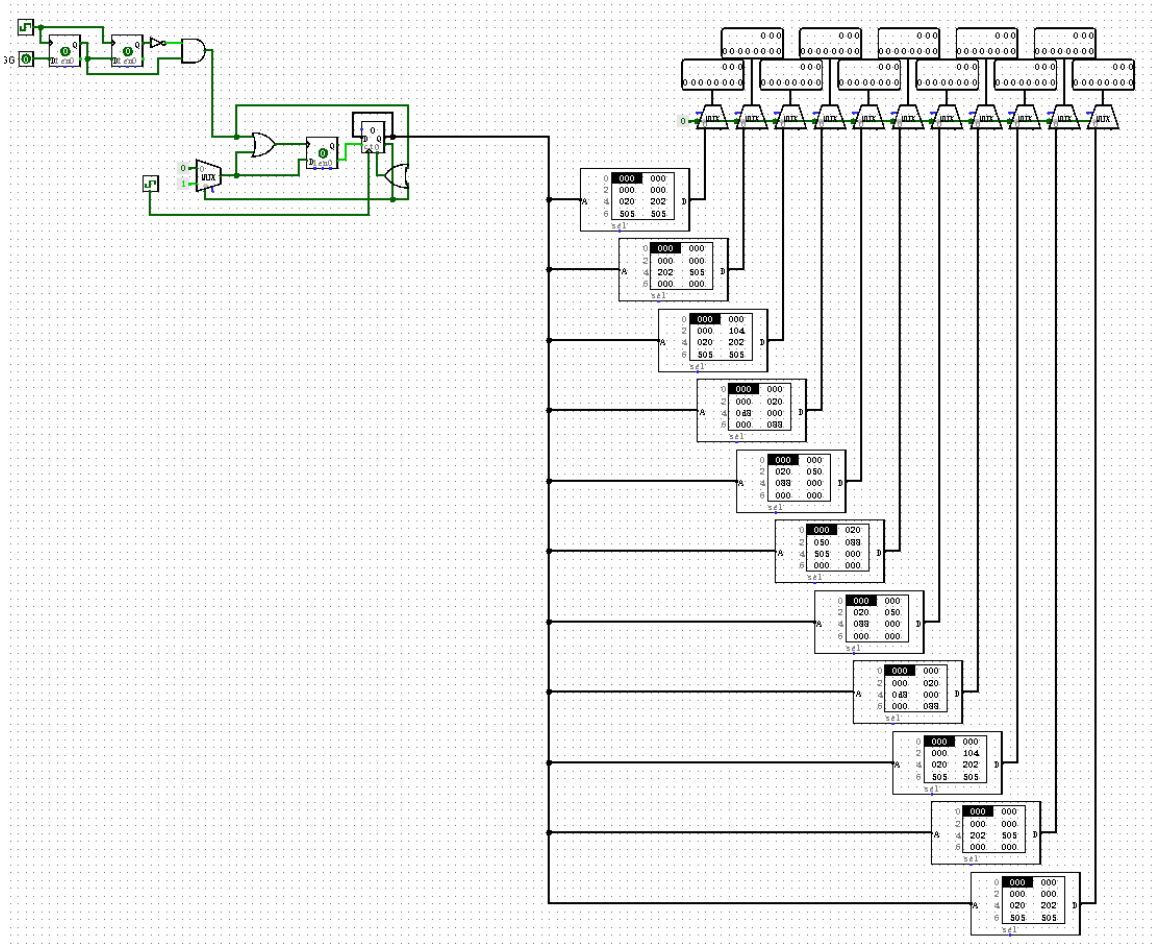
a. 随机数



b. 粒子



2. 结束动画



该游戏特色

1. 耐玩。别看只有9格，不记住套路，真的能玩很久，这也是我为什么喜欢这个游戏的原因。
2. 极好的拓展性。
    a. 一是游戏的大小：不将9个5×5的格子设为15×15的原因，就是为了保证良好的空间可拓展性，该游戏想要多少个块的都可以自设，且基本只在原本的电路进行复制粘贴、更换标签即可，因为就功能而言，各部分都是独立设计的，相当于调用函数；
    b. 二是游戏的玩法：只要改变输入时的组合电路，和贴B标签的方式，就可以改变游戏规则，于是，我发现只要是方块消除类游戏(各种消消乐)，我的游戏模型都可以进行模拟实现。当然，如果游戏中存在单个方块有多种状态的改变，还需要用寄存器、选择器和rom搭个状态变化（其实这个基本电路在特效的搭建中已经实现了，复制粘贴改改位数就能用）。

其它

- 在搭出动画电路后，我发现
    - 一部分同学的用Verybug写的小游戏我都可以做出来了；
    - 我完全可以制作GIF进行恶搞、或小型定格动画（以后有npy后我会搭个作为礼物，希望不会等到30岁）。
- Logisim让我找回了小时候搭积木的快乐。