

# 基于 Logisim 的跑马灯设计与仿真

李成奥 卫朝霞

四川大学锦城学院 四川 成都 611731

**【摘要】**在进行数字电路设计时，设计方案多数都是围绕着单片机进行电路设计和仿真的。对于数字电路初学者来说，刚刚认识基本逻辑门和一些常用小规模集成芯片，尚未建立计算机完整的系统概念，如若直接跳跃到用单片机设计电路，知识跨度太大，使理解 and 设计都变得不友好。针对该问题，提出一种初级入门的跑马灯设计方法，该方法利用多种不同的小规模集成器件，构建多个不同结构的跑马灯电路，使电路设计初学者能够从多维角度充分理解电路实现的灵活性。利用 Logisim 仿真软件模拟不同结构的跑马灯电路，仿真结果展示了电路实现的真实效果，友好的仿真界面给初学者带来很好的设计体验。

**【关键词】**跑马灯；模拟仿真；集成器件

## 引言

数字电路设计用到的集成器件种类繁多，功能差异大。译码器作为小型组合逻辑集成器件之一，在数字电路中有着广泛的用途，不仅可以用于代码的转换、终端设备的数字显示、信息的译码翻译，还可用于存储器的地址译码分配、数据分配，以及实现具有控制和判别功能的组合电路等<sup>[1]</sup>。译码器丰富的应用场景和各式各样的使用方法，给数字电路设计者留下了深刻的印象，尤其是 74LS138 译码器。此外，具有信息存储功能的触发器，是构成时序电路的最基本的逻辑单元，也是数字逻辑电路中一种重要的单元电路，在计算机和数字系统中有着广泛的应用<sup>[2]</sup>。本文主要围绕 74LS138 译码器和常用触发器集成器件，实现几种不同结构的跑马灯电路，并用仿真软件 Logisim 对每种电路进行模拟，实现电路工作起来的真实效果。

## 1 Logisim 仿真软件

Logisim 软件为数字逻辑电路设计提供了友好的仿真界面环境。通过环境内的工具箱接口和电路原件调度，能够方便快速地构建逻辑电路，设计出不同功能的数字电路。各种电路通过模块化，又可以在模块的基础上构建更加复杂的电路，甚至可以用于设计 CPU 并模拟整个计算机系统<sup>[3]</sup>，使用起来十分方便。

## 2 跑马灯设计

### 2.1 设计方案

本文以 8 个输出灯的流水效果为例，来设计跑马灯控制电路。系统上电工作后，在连续的 1Hz 时钟脉冲（CP）控制下，8 个灯中，始终要求是 1 灭 7 亮，且灭的灯，能够随着时钟脉冲的节拍自动循环右移（或左移）。1Hz 的时钟脉冲（CP）是控制跑马灯电路流水跳动的唯一控制输入。

从数字电路设计初学者的角度出发，考虑采用常规逻辑门电路和小规模集成芯片来实现电路设计，不考虑 51 单片机和 ARM 单片机等高集成度芯片。电路启动后，从 8 个灯的亮灭效果来分析，需要使用 8 个 LED 灯来输出流水效果。

首先设计的电路需要定义有 8 个输出端，用于挂接 8 个灯。电路启动后始终要保持 1 灭 7 亮，假设灯灭对应逻辑“0”，灯亮对应逻辑“1”，则所设计的电路输出端也应始终保持 1 个输出端为低电平逻辑“0”，另外 7 个输出端保持高电平逻辑“1”。其次，所熄灭的 LED 灯会随着时钟节拍顺序移动而非随机熄灭，这个条件可以通过改变线路排列来完成设计。再次，考虑引入一个输入端，用于接入连续脉冲 CP，用于控制电路自动运行。

在小规模数字电路设计中，常用的集成器件有：加法器、数值比较器、编码器、译码器、数据选择器、数据分配器等。结合电路分析的输出特点，采用 74LS138 译码器，74LS138 译码器有 8 个输出端和 3 个编码输入端，芯片启动后，8 个输出中每次只有 1 个输出有效低电平，剩余 7 个输出无效高电平。哪个输出端为有效低电信号取决于 3 个编码输入端的值。因此，可以考虑选用 74LS138 译码器，将其 8 个输出来连接 8 个灯，通过控制其 3 个输入端编码的顺序变化，达到使 8 个灯中 1 灭 7 亮，且灭的灯能够规律性的顺序移动。

当 74LS138 的 3 个输入端顺序从 000~111 时，8 个输出端  $y_0 \sim y_7$  依次输出低电平信号。因此，通过控制 3 个输入端的信号从 000 变化到 111，就可以使挂接在 74LS138 上面的 8 个输出灯显示为 1 个灯灭 7 个灯亮，并且灭的灯，会随着输入端的顺序变化而规律性移动。

电路的设计方案如图 1 所示。根据逻辑电路的设计方法，通过“分析逻辑问题→列真值表→写出逻辑函数表达式→简化表达式，确定相关配件→画逻辑电路图”的流程，可以详细实现电路的具体设计过程。

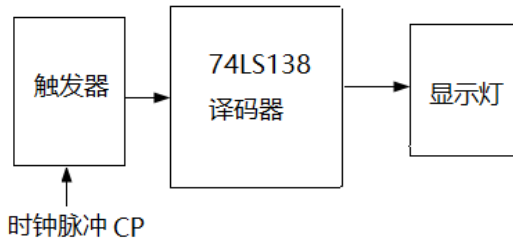


图 1 设计方案框图

## 2.2 设计原理

74LS138 为 3 线译码器，当输入 3 个任意信号时，使得其输出端总是保持七个高电平一个低电平（即 7 个 1，1 个 0），符合 8 个跑马灯的设计需求。因此采用 74LS138 译码器配合三种不同触发器（即：T 触发器、D 触发器、J—K 触发器）来完成跑马灯的设计。（假设高电平为 1，低电平为 0）

跑马灯设计需要有 1 个 74LS138 译码器，1 个时钟，高电平（即常量 1）和 3 个 T 触发器（或 3 个 D 触发器或 3 个 J—K 触发器）。由于 Logisim 中没有集成器件，所以我们需要先将 74LS138 译码器通过基本逻辑电路配置出来再使用。完成配置后，在器件库中选择所需的触发器与时钟和常量按照下列方法进行连接即可完成跑马灯设计。

## 2.3 设计实现

### 2.3.1 T 触发器与 74LS138 译码器的实现

利用 3 个 T 触发器相连接，实现具有三个输出端的组合异步时序逻辑电路。假设 3 个 T 触发器的输入端和输出端分别为：CP0、T0、Q0；CP1 T1 Q1；CP2 T2 Q2。将 T0、T1 T2 始终接入同一高电平 1，CP0 接入时钟控制器 CP，CP1 接入输出端 Q0，CP2 接入输出端 Q1。然后将 3 个输出端分别接 74LS138 译码器的 3 个输入端，使得启动电路时输入端按 000 → 001 → 010 → 011 → 100 → 101 → 110 → 111 → 001 的顺序进行循环，用于保证输出端也可按顺序进行循环，使得跑马灯按要求点亮或熄灭。

### 2.3.2 D 触发器与 74LS138 译码器的实现

以 T 触发器为基础，将 D 触发器改进为 T 触发器的形式后再与 74LS138 译码器进行连接，使得该逻辑电路能够按照一定顺序循环运行。假设 3 个 D 触发器的输入端和输出端分别为：D0、CP0、Q' 0、Q0；D1 CP1 Q' 1 Q1，D2 CP2 Q' 2 Q2。改进方法为：

1. 将 D 触发器自身的输入端 D 与输出端 Q 相连接。
2. 将 Q' 0 同时接入 74LS138 译码器的输入端，和 CP1，Q' 1 同时接入 74LS138 译码器的输入端，以及 CP2，Q' 2 接入 74LS138 译码器的输入端。且 Q' 0、Q' 1 Q' 2 按照一定顺序接入 74LS138 译码器，使输入端按照 00 0 → 001 → 010 → 011 → 100 → 101 → 110 → 111 → 001 的顺序进行循环，以此符合跑马灯的相关要求。

### 2.3.3 J—K 触发器与 74LS138 译码器的实现







以 T 触发器为基础，改进 J—K 触发器，利用 3 个改进后的 J—K 触发器相互连接，实现具有三个输出端的组合异步时序逻辑电路。假设 3 个 J—K 触发器的输入端和输出端分别为：CP0、J0、K0、Q0、Q' 0；CP1 J1 K1 Q1 Q' 1；CP2 J2 K2 Q2 Q' 2。分别将 3 个 J—K 触发器自身的 J、K 输入端相连接（即 J0 与 K0 相连，J1 与 K1 相连，J2 与 K2 相连）并接入同一高电平常量 1；CP0 接入时钟控制器 CP，Q' 0 接入 CP1 与 74LS138 中的一个输入端，Q' 1 接入 CP2 与 74LS138 中的一个输入端，Q' 2 接入 74LS138 中的一个输入端，其中 3 个 74LS138 的输入端按照一定顺序进行接入，使得其能够按照 000 → 001 → 010 → 011 → 100 → 101 → 110 → 111 → 0

01 的循环运行，以此完成跑马灯的相关要求。

## 3 电路设计与仿真

### 3.1 74LS138 译码器的电路设计与仿真

74LS138 译码器的电路设计，是基于真值表的唯一性来完成的。具体制作过程如下。

打开 Logisim，在菜单栏中选择“项目”→“添加电路”，给该电路命名为 74LS138。由于 74LS138 有 3 个输入端和 8 个输出端，在这里我们选择两种不同形状的引脚表示。在页面上方有  和  两种不同形状引脚，这里选用  来定义译码器的 3 个输入端，选用  来定义译码器的 8 个输出端。从菜单栏单击并将引脚移入到设计框内，加入 3 个输入端  和 8 个输出端 。点击“项目”→“分析电路”→“真值表”，单击 X 将其真值表更改为 74LS138 的真值表，在表中每一行相对应的数值处，点击“建立电路”→“OK”，74LS138 的电路设计就完成了，如图 2 所示。

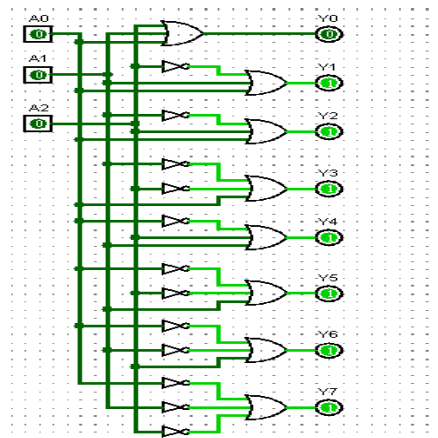
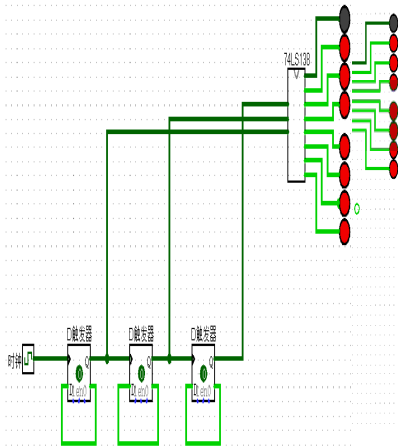


图 2 74LS138 电路设计图

这种设计电路的方法，适合于常规小中规模集成电路，通过真值表可以快速建立其逻辑电路图。将图 2 所示的电路，定义为一个模块，方便在此模块基础上，构建功能强大集成度更高的复杂电路。

### 3.2 跑马灯电路的设计与仿真

跑马灯电路的设计制作方法，可以按照前面设计原理的分析，将三种不同触发器（D 触发器、T 触发器、JK 触发器），结合前面已经设计好的译码器电路模块，分别进行线路连接。在 Logisim 左侧分类栏中，选重所需器件和译码器电路模块，将其移入设计框内，通过连线建立电路。当电路连接好后，点击菜单栏中的“模拟”→“启用时钟模拟”，调节时钟频率并启动电路，测试并观察输出端 8 个灯在连续时钟脉冲控制下的变化规律。



用 T 触发器和 74LS138 译码器设计的跑马灯电路图如图 3 所示。

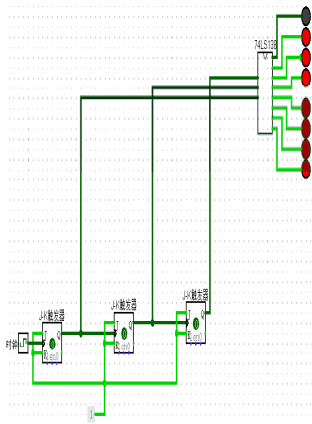


图 3 T 触发器和译码器制作的跑马灯电路

用 D 触发器和 74LS138 译码器设计的跑马灯电路如图 4 所示。

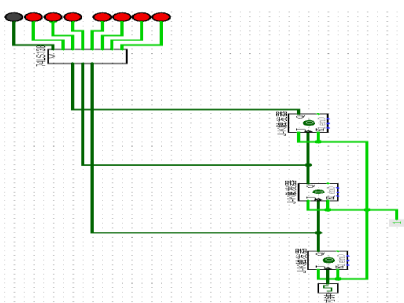


图 4 D 触发器和译码器制作的跑马灯电路

用 JK 触发器和 74LS138 译码器设计的跑马灯电路如图 5 所示。

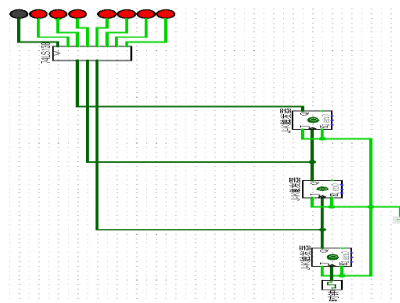


图 5 JK 触发器和译码器制作的跑马灯电路

三种跑马灯电路及其在 Logisim 中的仿真结果如图 3~5 所示，8 个 LED 灯在时钟脉冲 CP 的控制下，按照一定规律进行定向交替熄灭，从而完成跑马灯的流水设计。

### 总结

本文针对数字电路设计初学者，提出了用常规逻辑门和小规模集成电路设计跑马灯的方法，避免了用单片机设计存在知识跨度大带来的设计障碍。本文通过对跑马灯设计方案和设计原理的分析，分别用三种不同的触发器（D 触发器、T 触发器、JK 触发器）和 74LS138 译码器，设计出三种不同结构的跑马灯电路，使初学者能够从多维角度充分理解电路实现的灵活性。利用 Logisim 仿真软件，对每种电路进行模拟仿真。仿真结果展示了电路实现的真实效果，友好的仿真界面给初学者带来很好的设计体验。

### 【参考文献】

- [1] 史萍；余艳．基于 STM32 单片机的实验开发板设计 [J]．电子世界，2020，(03)：128-129.
- [2] 陆璐．基于单片机的 LED 跑马灯设计 [J]．电子制作，2018，(09)：83-84+62.
- [3] 马惠兰．基于 Proteus 的跑马灯设计及仿真 [J]．电子技术与软件工程，2016(04)：12.