

实验二

**译码器电路仿真实验**

**实验报告**

学期：2023-2024 第一学期

编制日期：2023 年 10 月 31 日

编制人：江家玮

学号：22281188

班级：计科2204

1. 实验目的

1、熟悉Multisim14.0仿真软件环境；

2、掌握译码器的工作原理；

3、常见译码器的应用。

1. 实验内容

**任务A：**完成实验指导书中的实验二第四部分：“利用两块3线-8线译码器74LS138D设计一个4线-16线译码电路”。

**任务B：**把你学号转换为16进制数，将转换后的十六进制数做为最小项编号构建**四变量**输入的逻辑函数。例如，你本人学号为21225678，转换16进制后为143E0CE，则你需要实现的逻辑函数为：F(A,B,C,D)=∑m(0,1,3,4,13,15)（重复出现的数字仅计算1次），用任务A中实现的4-16译码器实现该逻辑函数，并测试验证。

1. 实验原理

所谓编码是指在选定的一系列二进制数码中，赋予每个二进制数码以某一固定含义。译码器是编码的逆过程，将输入的每个二进制代码赋予的含义“翻译”过来，给出相应的输出信号，能够完成译码功能的电路叫做译码器。74LS138D属于3线 - 8线译码器，该译码器有三个使能端，三路输入端,输入高电平有效，输出低电平有效。

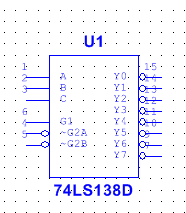


图2-1译码器74LS138D

1. 实验过程

**任务A：**完成实验指导书中的实验二第四部分：“利用两块3线-8线译码器74LS138D设计一个4线-16线译码电路”。利用两块3线-8线译码器74LS138D设计一个4线-16线优先编码电路，具体要求如下：

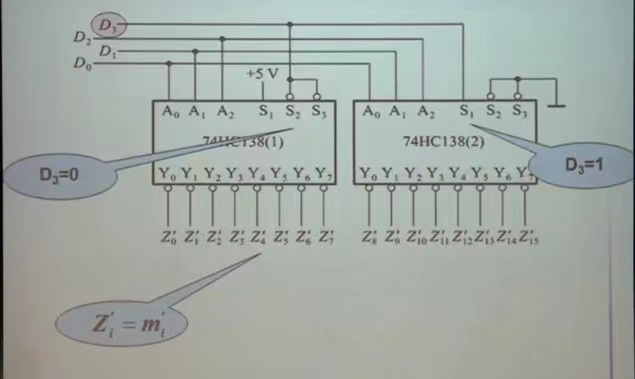
1. 简要说明设计思路，画出实验电路图。
2. 对4线-16线译码器优先编码电路进行仿真。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入端 | | | | | | | 输出端 | | | | | | | | | | | | | | | |
| G1 | G2A | G2B | D | C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 | Y8 | Y9 | Y10 | Y11 | Y12 | Y13 | Y14 | Y15 |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

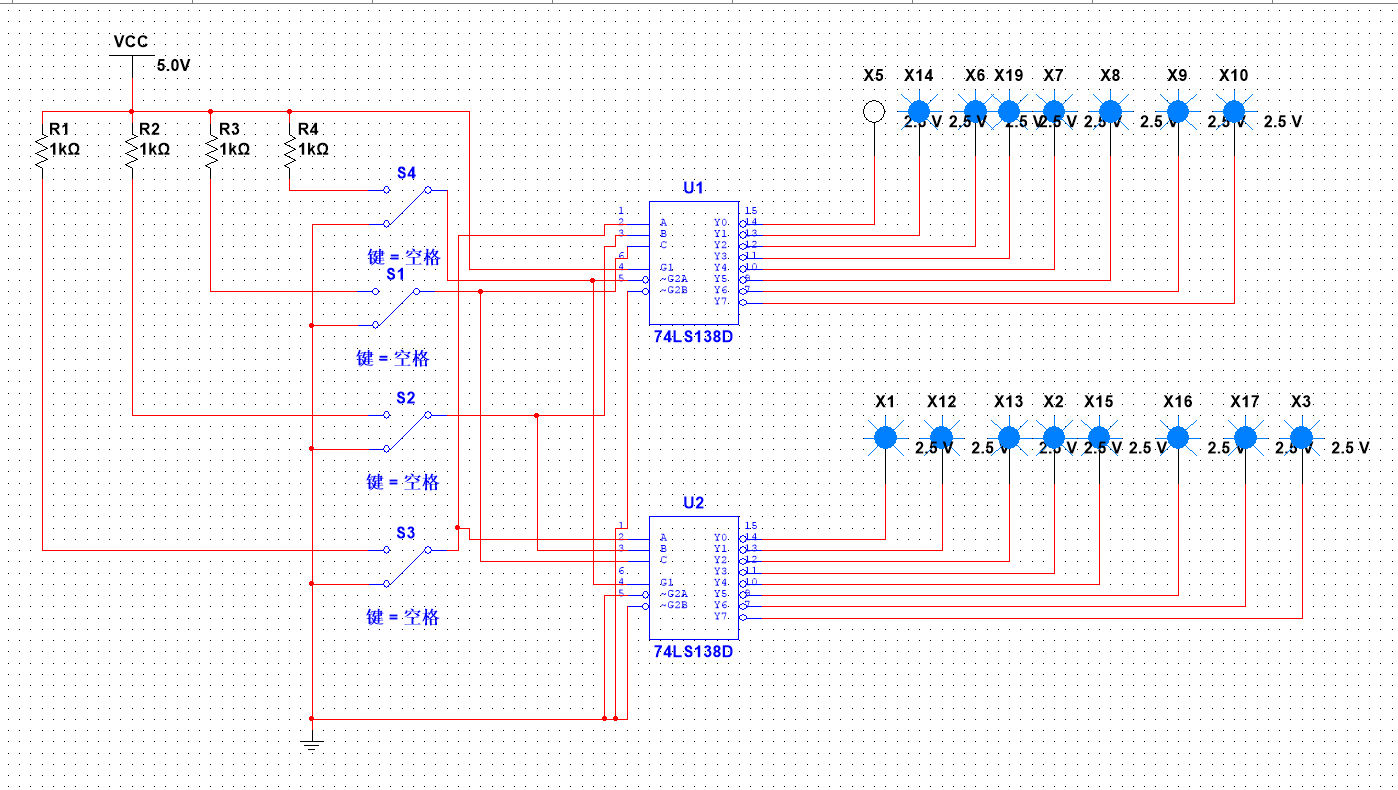
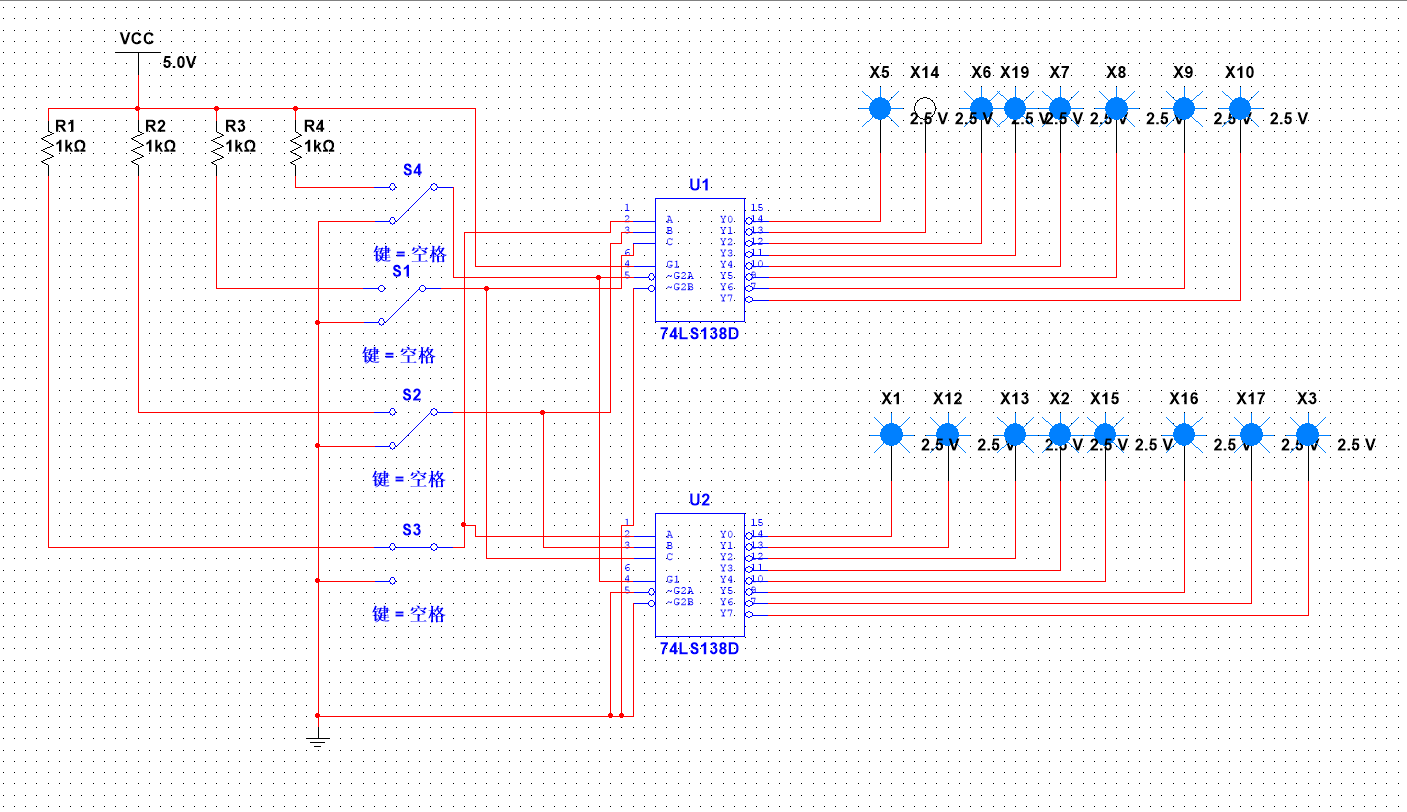
1. 将实验结果记录在下表中，验证4线-16线译码器的逻辑功能。
2. 设计思路：

由8输出变16输出，两个位片的A、B、C相同按之前的连接，用最高位D控制两个片，D=0时低位片工作，D=1时高位片工作,已知G1=1，G2A=0，G2B=0时位片工作，故连接使D = G1(高位片) = -G2A(低位片) =G2B(低位片) ；再固定：G1(低位片) = -G2A(高位片) =G2B(高位片) = 1即可。

实验电路图：



2、仿真结果如图所示：



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入端 | | | | | | | | | | | | 输出端 | | | | | | | | | | | |
| G1 | G2A | G2B | D | C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | | Y5 | Y6 | Y7 | Y8 | Y9 | Y10 | Y11 | Y12 | Y13 | Y14 | Y15 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

3、

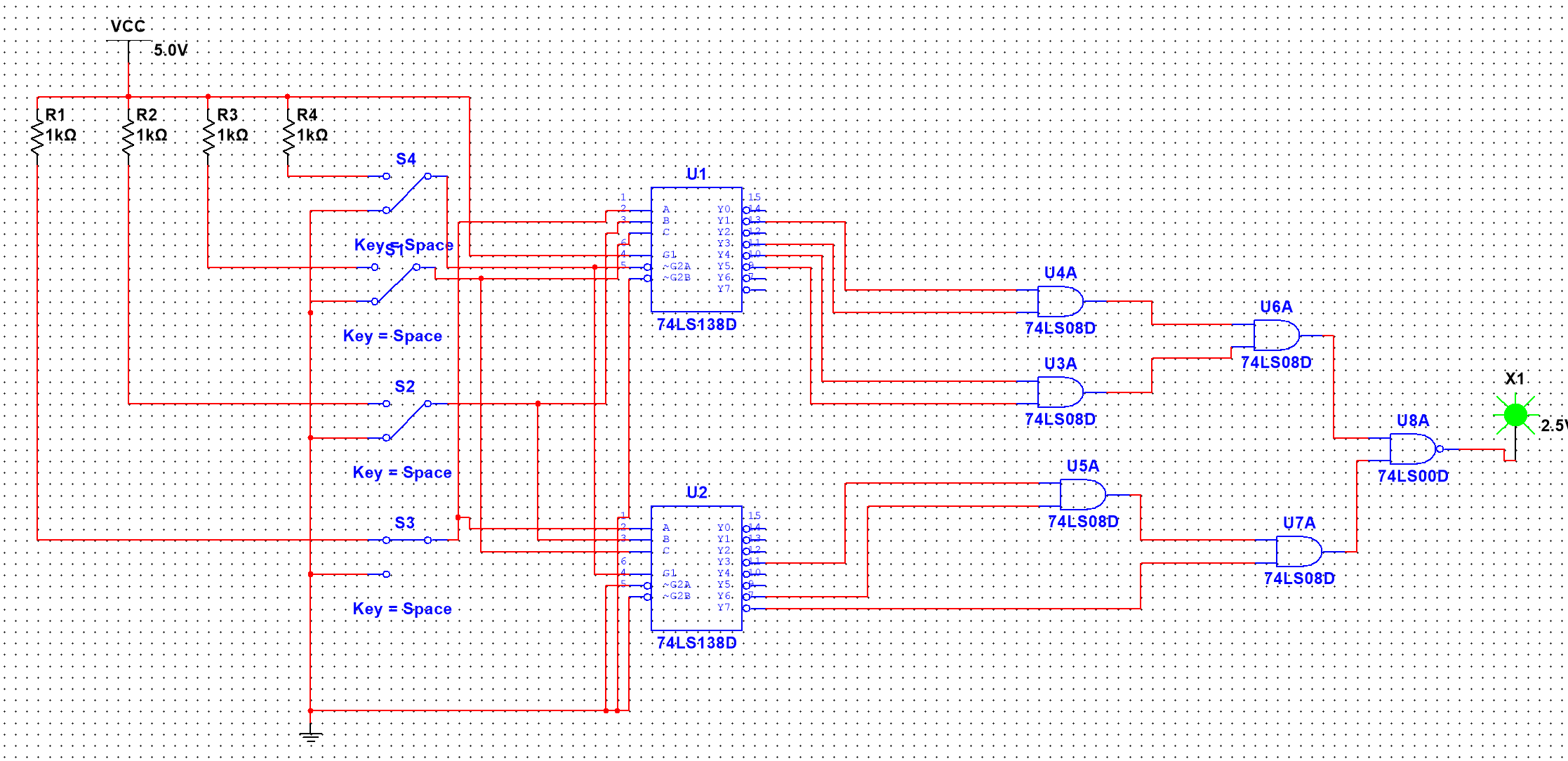
**任务B：**把你学号转换为16进制数，将转换后的十六进制数做为最小项编号构建**四变量**输入的逻辑函数。例如，你本人学号为21225678，转换16进制后为143E0CE，则你需要实现的逻辑函数为：F(A,B,C,D)=∑m(0,1,3,4,13,15)（重复出现的数字仅计算1次），用任务A中实现的4-16译码器实现该逻辑函数，并测试验证。

学号：22281188

转换后的16进制数：153FBE4

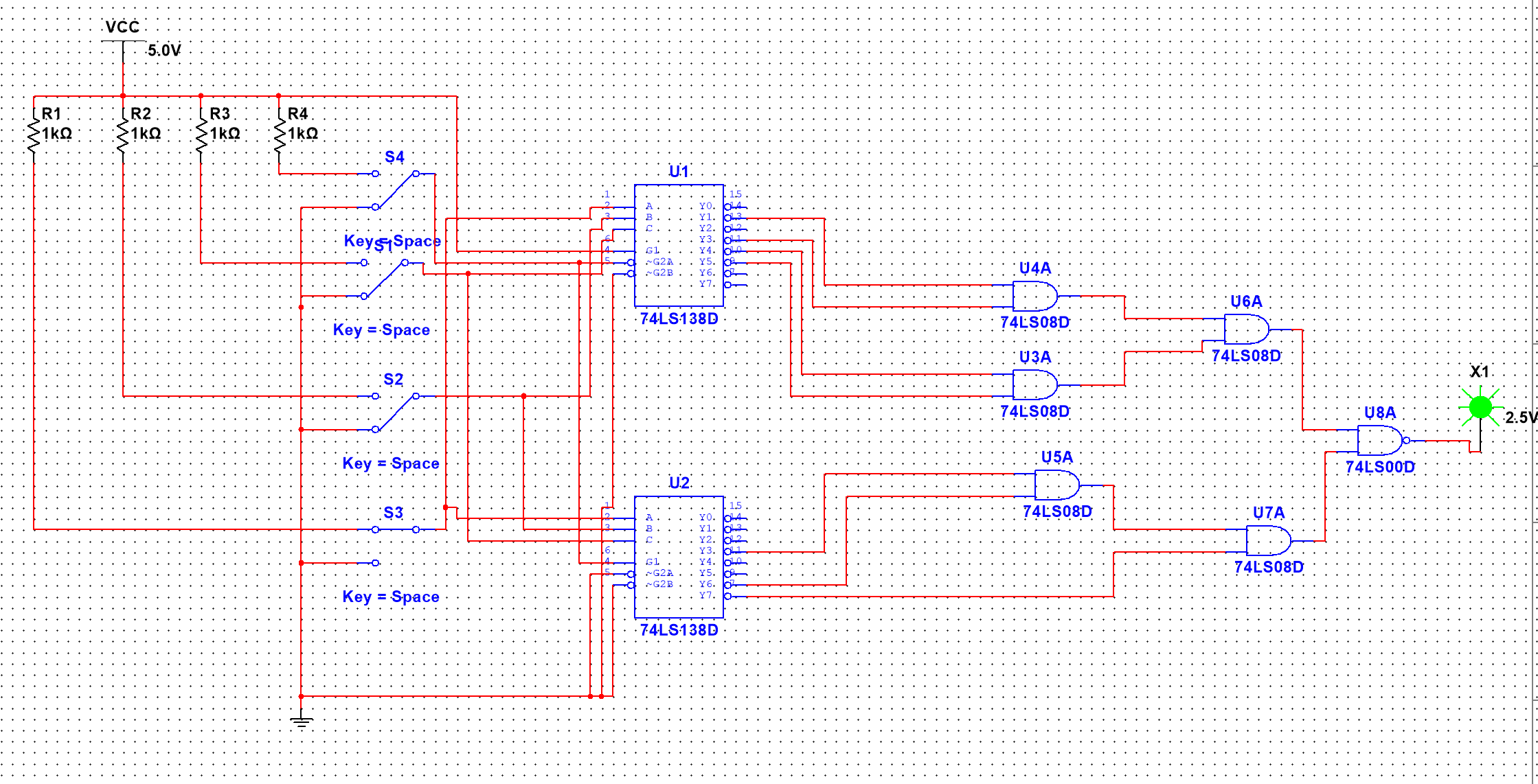
需要实现的逻辑函数：F(A,B,C,D)=∑m(1，3，4，5，11，14，15)

实验电路：

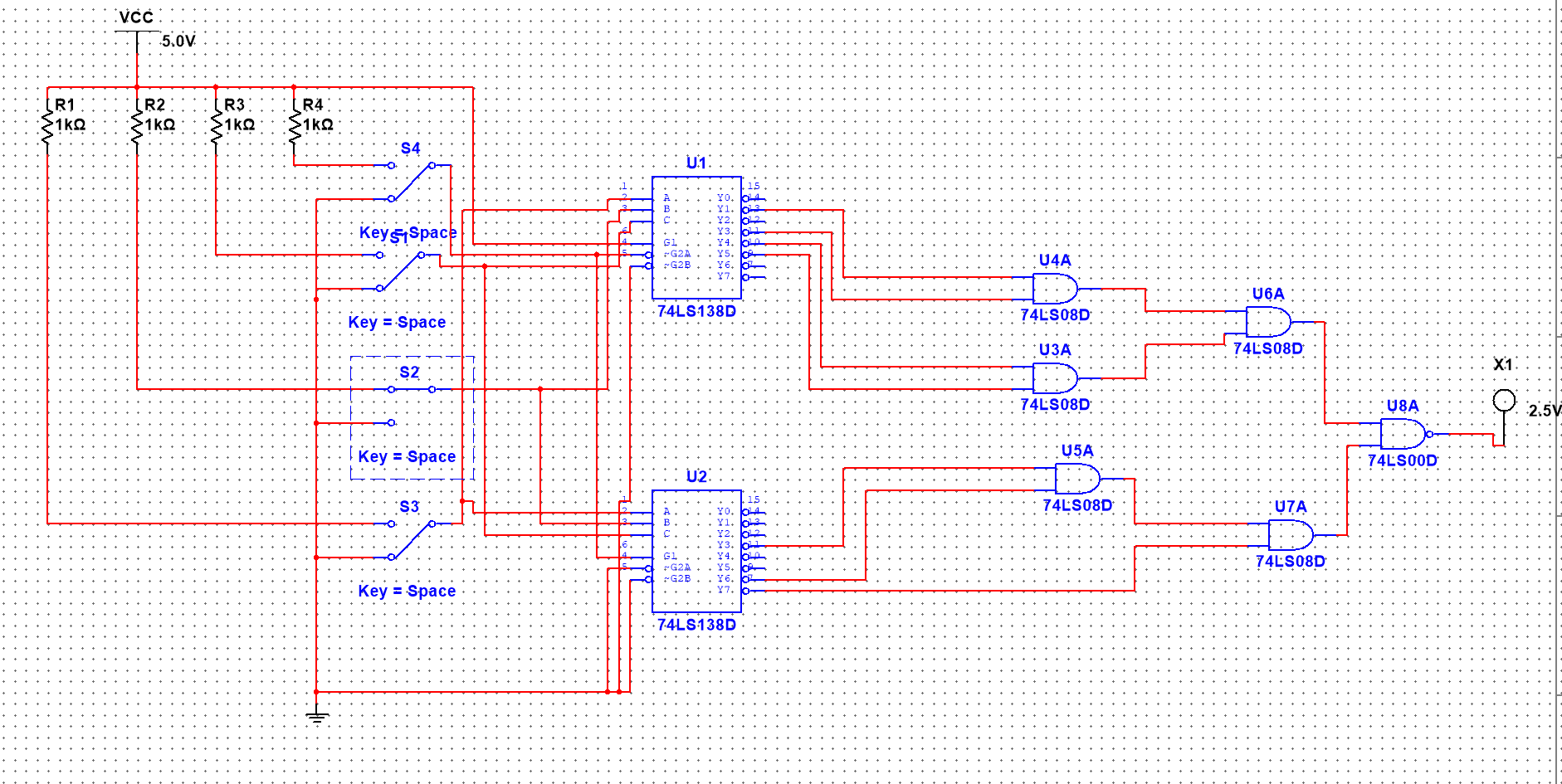


测试验证结果：

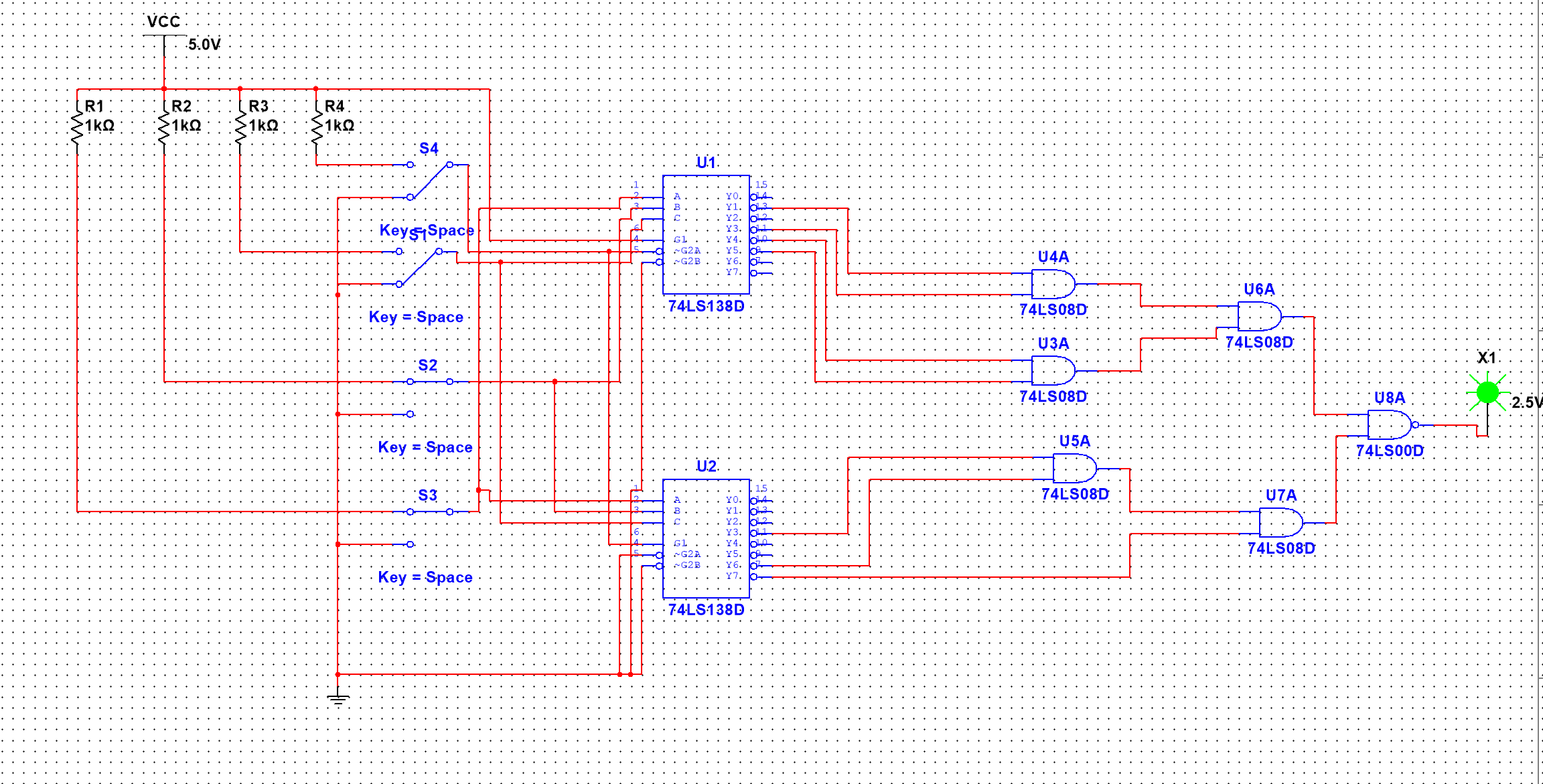
当为m1时：(亮)



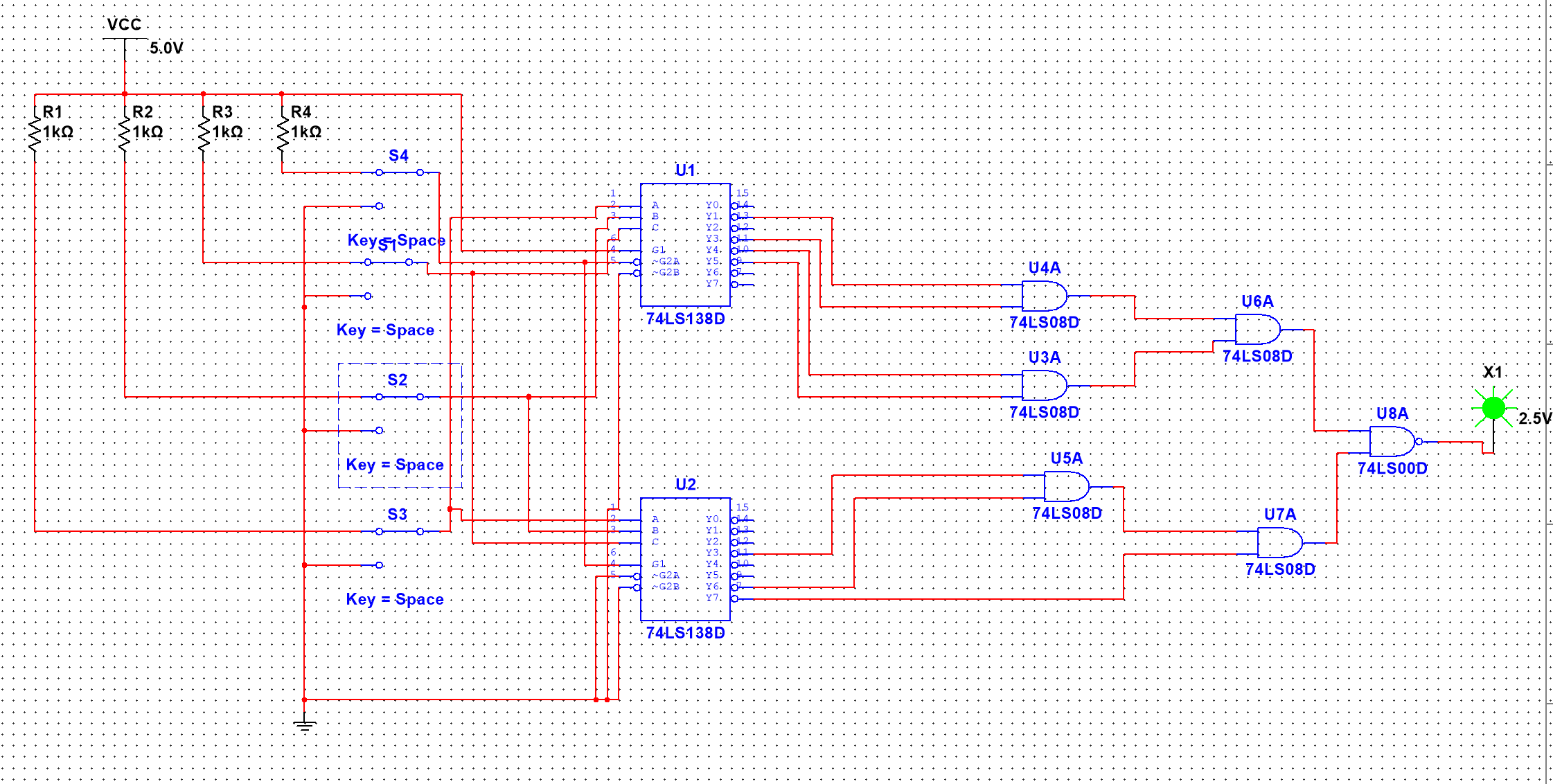
当为m2时：(不亮)



当为m3时：(亮)



当为m15：(亮)



1. 实验结果与分析

1、为什么138采用低电平有效的输入/输出方式？

答：主要是为了降低功率。选中信号输出时，地址译码器输出端为低电平，此时译码器不输出功率；选中信号没有输出（不选中）时，译码器输出端为高阻状态，输出消耗功率也为0。因此芯片的CS信号采用低电平有效可以最大程度减小片选控制的功率消耗。 此外，低电平有效也可以最大程度地减小干扰和保证控制的可靠性。低电平有效时，外部的任何干扰都不能进入被控制的芯片，因而保证芯片的可靠工作。这样在干扰信号能够进入芯片时是在芯片不工作时（片选无效），也就是说，干扰信号对芯片的工作没有影响。而若采用高电平有效，在芯片工作时不要说外部干扰信号能够很容易地进入芯片干扰，造成各种误动作，就连电源的任何波动都可能影响芯片的正常工作。

2、调研译码器电路的应用场景，并举例说明。

答：（1）数码显示器驱动：

应用场景：七段数码显示器通常使用BCD码（二进制编码的十进制）来显示数字。译码器可将BCD输入转换为七段数码显示器的相应段选信号，从而实现数字的显示。

示例：将BCD编码的数字输入（例如0000到1001）传递给译码器，以便在七段数码显示器上显示相应的数字。

（2）存储器芯片选择：

应用场景：在计算机系统中，存储器芯片（如RAM或ROM）的选择通常使用地址译码器进行。地址译码器将输入的地址信号转换为特定的芯片选通信号，以访问存储器中的数据。

示例：通过地址译码器，将特定的内存地址映射到RAM芯片上，以便读取或写入数据。

（3）键盘扫描器：

应用场景：在计算机键盘中，键盘扫描器使用译码器来识别用户按下的键。每个键都有一个唯一的编码，译码器将按下的键的编码转换为可供计算机识别的信号。

示例：用户按下键盘上的特定键，译码器将键的编码转换为计算机可理解的信号，以便在屏幕上显示相应的字符。

（4）遥控器解码：

应用场景：遥控器用于控制各种设备，如电视、音响系统等。通常，遥控器上的按键编码需要被解码器识别，以执行相应的操作。

示例：在遥控器上按下音量加大按钮，译码器将该按钮的编码转换为音响系统的音量增加指令。

（5）电梯控制系统：

应用场景：电梯控制系统使用译码器来解释来自不同楼层按钮的输入，以确定电梯的移动方向和目标楼层。

示例：乘客在电梯大厅按下上行或下行按钮，译码器将这些输入转换为电梯的运行指令。

总的来说，译码器电路在数字电子设备中起到了关键的作用，用于将输入信号转化为可理解或可操作的信号。它们在多种应用中广泛使用，包括显示、存储器选择、用户界面、控制系统等。