湖南科技大学计算机科学与工程学院

计算机组成原理 课程设计报告

**专业班级：** 23计科六班

**姓 名：** 肖永财

**学 号：** 2314020106

**指导教师：** 李志刚

**时 间**： 2024.12.27-2025.1.2

**地 点**： 逸夫楼328

|  |
| --- |
| **指导教师评语：**  **成绩： 等级：**  **签名：**  **年 月 日** |

**实验一：ROM仿真**

1. **实验题目**

ROM仿真

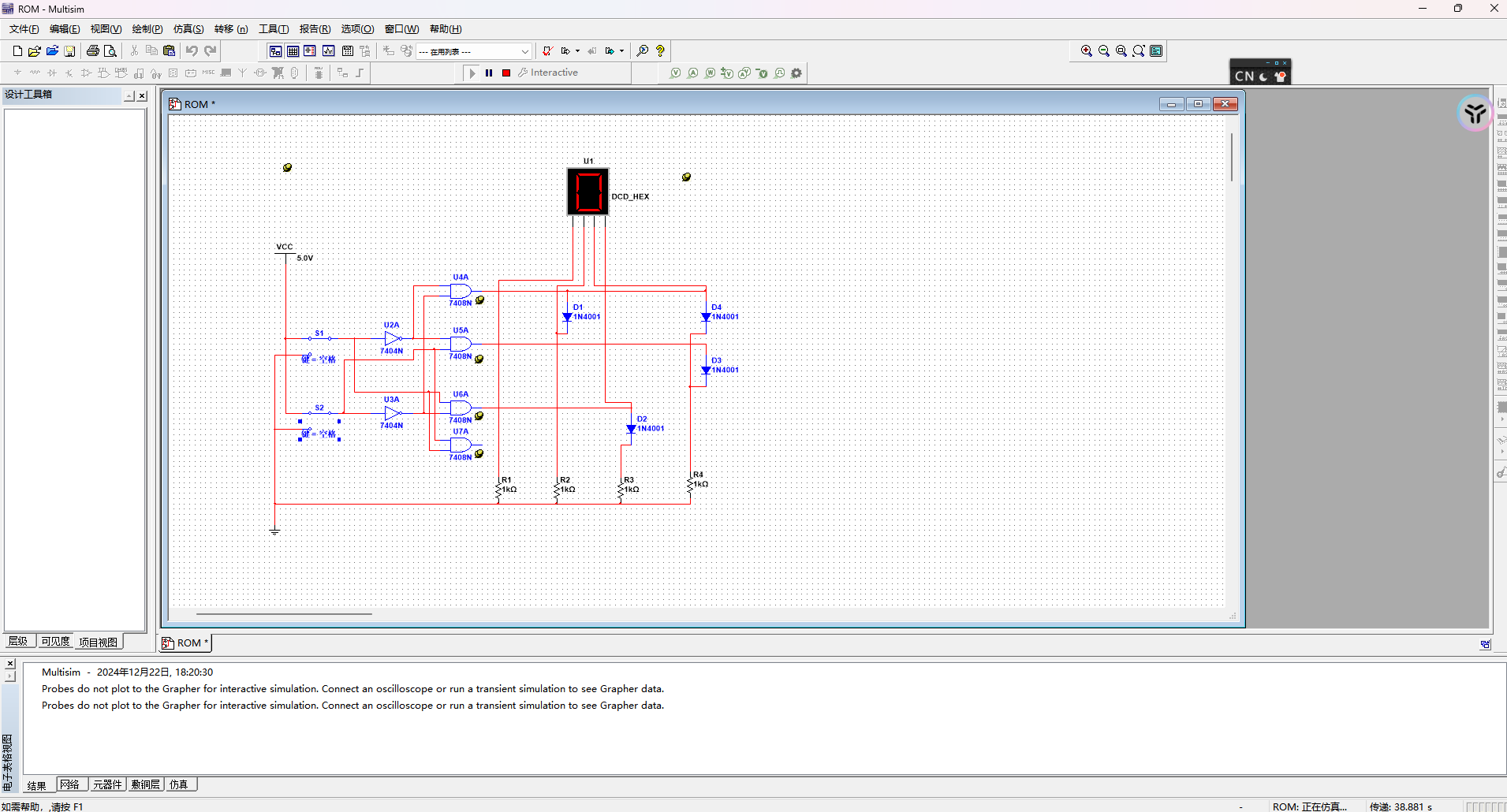
1. **实验目的**
2. 了解ROM的工作原理
3. 需要画出ROM的布线图
4. 掌握ROM读出和写入原理
5. **总体设计**
6. **实验原理**

存储元是否有有二极管组成，若有，由于二极管导通那么数据线上可以读出1；如果没有二极管，数据线上输出接地电平，读出0。

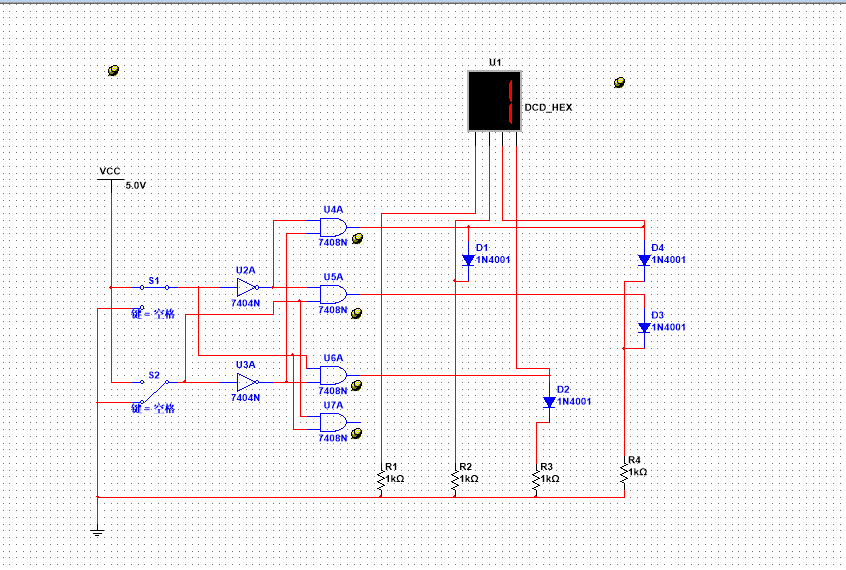
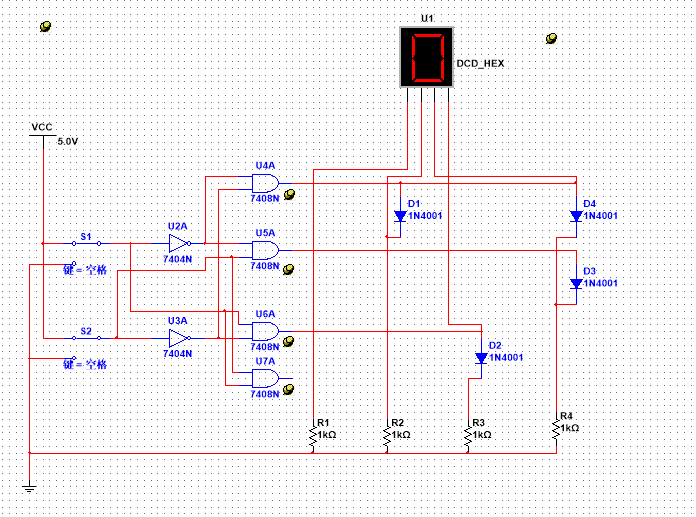
1. **实验内容**

当S1和S2都为0时，输出0。当S1为0，S1接高电平时，值为1。当s1和s2都接高电平时，值为6。数码管下方有四个引脚，从右到左分别是代表1,2，4，8。我们想要输出6，本质是在第2个引脚和第3个引脚处输出高电平，这样2+4的结果就是6。当我们想输出0时候，只需要让4个引脚输入的值都是0即可。当我们想输出1时，只需要让最右侧的引脚为高电平，其他引脚为低电平。

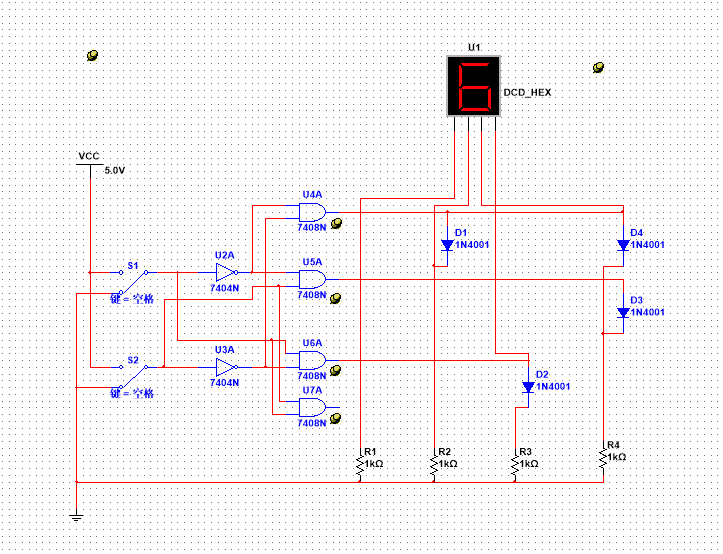
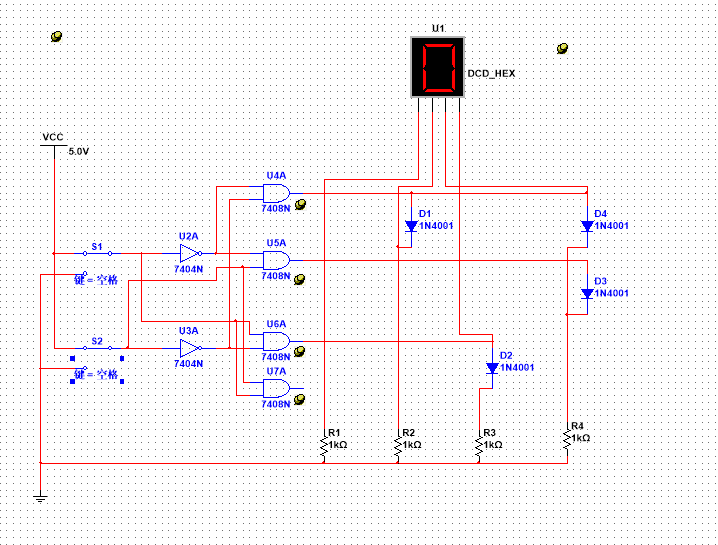
1. **详细设计**



1. **实验结果与分析**



数码管输出0 数码管输出1



数码管输出0 数码管输出6

在该电路中，数码管的右侧表示低位，左侧表示高位，且高电平输入为有效信号。为了实现数字显示，只需将数字转换为对应的BCD码，并根据BCD码中为1的位，将高电平信号通过二极管连接至LED灯的相应引脚。学号的后四位为0106，通过两个开关可以控制LED灯的不同状态，从而显示学号的特定位。

1. **小结与心得体会**
   1. **要注意数码管的四个引脚代表的二进制含义与十进制含义的关联**
   2. **要注意加下拉电阻，实现保护作用**

**实验二：验证74LS181运算和逻辑功能**

1. 实验目的
2. 掌握算术逻辑单元（ALU）的工作原理；
3. 熟悉简单运算器的数据传送通路；
4. 画出逻辑电路图及布出美观整齐的接线图；
5. 验证4位运算功能发生器(**74LS181**)组合功能。
6. 实验原理

ALU能进行多种算术运算和逻辑运算。4位ALU-**74LS181**能进行16种算术运算和逻辑运算。功能表如下：

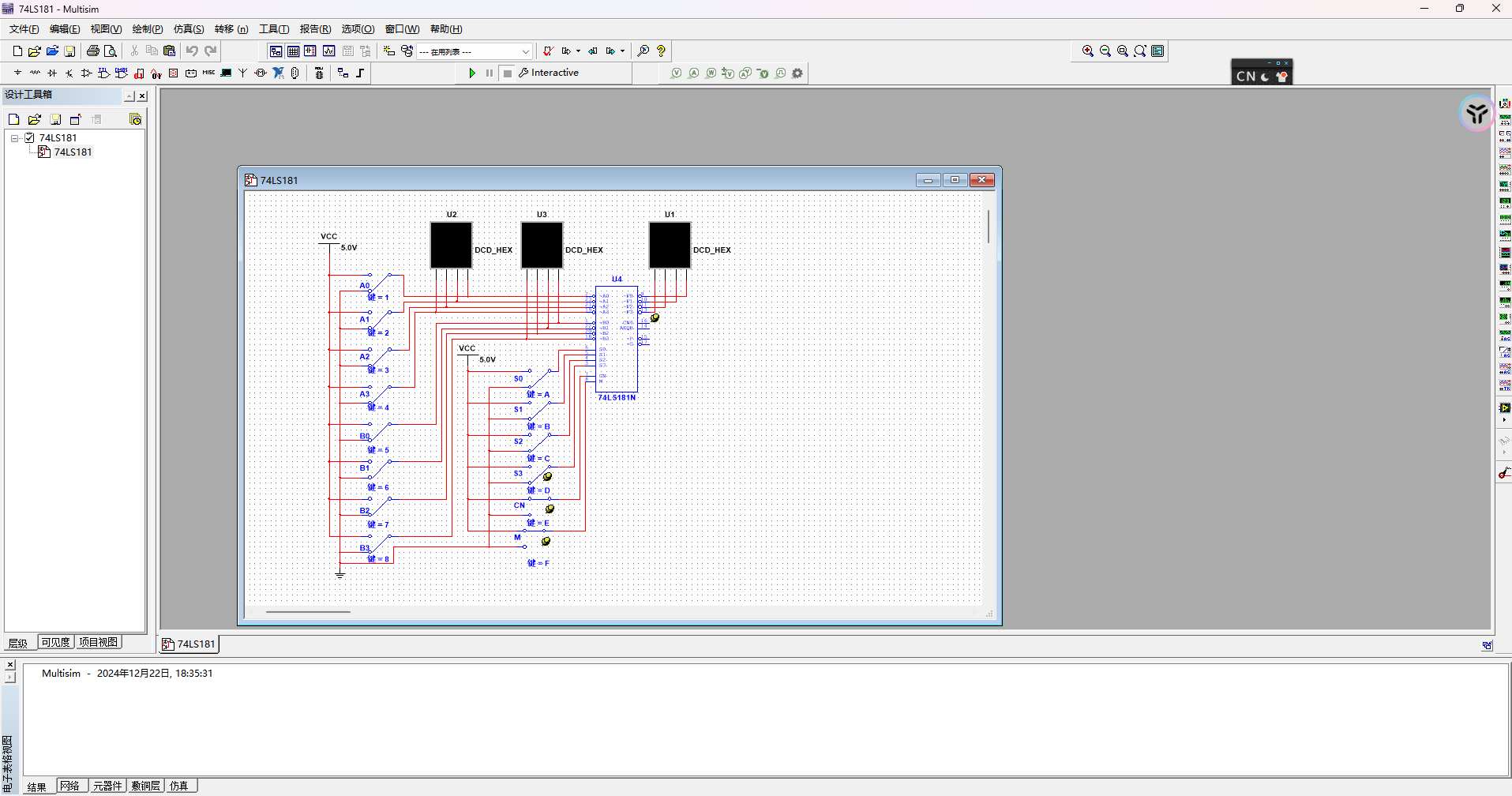
|  |  |  |  |
| --- | --- | --- | --- |
| 方式 | **M** = 1 逻辑运算 | **M** = 0算术运算 | |
| **S3 S2 S1 S0** | 逻辑运算 | **CN=1 (无进位)** | **CN =0 (有进位)** |
| **0 0 0 0** | F=/A | **F=A** | **F=A加1** |
| **0 0 0 1** | **F=/(A + B)** | **F=A + B** | **F=(A + B) 加1** |
| **0 0 1 0** | **F=(/A ) B** | **F=A + /B** | **F=( A + /B )加1** |
| **0 0 1 1** | **F=0** | **F=负1（补码形式）** | **F=0** |
| **0 1 0 0** | **F=/(A B)** | **F=A加A ( / B)** | **F=A加A / B加1** |
| **0 1 0 1** | **F=/B** | **F=(A + B) 加A / B** | **F=(A + B)加A / B加1** |
| **0 1 1 0** | **F=A ⊕ B** | **F=A减B减1** | **F=A减B** |
| **0 1 1 1** | **F=A/B** | **F=A (/ B)减1** | **F=A (/ B)** |
| **1 0 0 0** | **F=/A +B** | **F=A加A B** | **F=A加A B加1** |
| **1 0 0 1** | **F=/( A ⊕ B)** | **F=A加 B** | **F=A加B加1** |
| **1 0 1 0** | **F=B** | **F=( A + /B )加A B** | **F=( A + /B )加A B加1** |
| **1 0 1 1** | **F=AB** | **F=AB减1** | **F=AB** |
| **1 1 0 0** | **F=1** | **F=A加 A** | **F=A加 A加1** |
| **1 1 0 1** | **F=A + /B** | **F=(A + B) 加 A** | **F=(A + B) 加 A加1** |
| **1 1 1 0** | **F=A + B** | **F=(A + / B) 加 A** | **F=(A + / B) 加 A加1** |
| **1 1 1 1** | **F=A** | **F=A 减1** | **F=A** |

(**上表中的“/”表示求反**)

ALU-**74LS181**引脚说明：M=1 逻辑运算，M=0算术运算。

|  |  |
| --- | --- |
| 引 脚 | 说 明 |
| M 状态控制端 | M=1 逻辑运算；M=0算术运算。 |
| **S3S3 S1 S1**运算选择控制 | **S3S3 S1 S1**决定电路执行哪一种算术 |
| **A3A2 A1 A1** | 运算数1，引脚3为最高位 |
| **B3B2 B1 B0** | 运算数2，引脚3为最高位 |
| **Cn** 最低位进位输入 | **Cn** =0 有进位；**Cn** =1 无进位； |
| **Cn+4**本片产生的进位信号 | **Cn+4**=0 有进位；**Cn+4**=1 无进位； |
| **F3F2 F1 F0** | **F3F2 F1 F0**运算结果，**F3**为最高位 |

1. 实验内容
2. **详细设计**



**1. 电源供应**

**VCC连接到5V电源，为整个电路供电。**

**2. 74LS181N 芯片**

**功能：74LS181N是一个4位数值比较器，用于执行两个4位数之间的比较操作。**

**输入端口：**

**A0, A1, A2, A3: 比较数据A的四位二进制数输入。**

**B0, B1, B2, B3: 比较数据B的四位二进制数输入。**

**S0, S1, S2, S3: 控制信号输入，决定比较器的具体操作模式。**

**输出端口：**

**M: 输出结果之一，表示A<B。**

**CN: 输出结果之一，表示A=B。**

**G: 输出结果之一，表示A>B。**

**3. 七段数码管**

**U1、U2和U3是共阳极七段数码管，分别用来显示不同的状态或信息。**

**连接方式：**

**各个数码管的a至g段通过电阻与74LS181N的不同引脚相连，以控制各个段的亮灭，从而显示出相应的字符。**

1. **实验结果与分析**

验证74LS181型4位ALU的逻辑算术功能，填写下表：

表3 验证记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S3 S2 S1 S0 | 数据1 | 数据2 | M=1逻辑运算 | M=0算术运算 | |
| 逻辑运算 | CN=1 (无进位) | CN =0 (有进位) |
| 0 0 0 0 | AH | 5H | F=5 | F=A | F=B |
| 0 0 0 1 | AH | 5H | F=0 | F=F | F=0 |
| 0 0 1 0 | AH | 5H | F=5 | F=A | F=B |
| 0 0 1 1 | AH | 5H | F=0 | F=F | F=0 |
| 0 1 0 0 | FH | 1H | F=E | F=D | F=E |
| 0 1 0 1 | FH | 1H | F=E | F=D | F=E |
| 0 1 1 0 | FH | 1H | F=E | F=D | F=E |
| 0 1 1 1 | FH | 1H | F=E | F=D | F=E |
| 1 0 0 0 | FH | FH | F=F | F=E | F=F |
| 1 0 0 1 | FH | FH | F=F | F=E | F=F |
| 1 0 1 0 | FH | FH | F=F | F=E | F=F |
| 1 0 1 1 | FH | FH | F=F | F=E | F=F |
| 1 1 0 0 | 5H | 5H | F=F | F=A | F=B |
| 1 1 0 1 | 5H | 5H | F=F | F=A | F=B |
| 1 1 1 0 | 5H | 5H | F=5 | F=4 | F=5 |
| 1 1 1 1 | 5H | 5H | F=5 | F=4 | F=5 |

1. **小结与心得体会**
2. **通过老师提供的实验指导书，我完成了连线工作，在这个过程里，我体悟了数码管的工作原理**
3. **同时，这部分工作也为后面实现运算器打下了基础。让我更好的理解CPU内组件的实现机制**

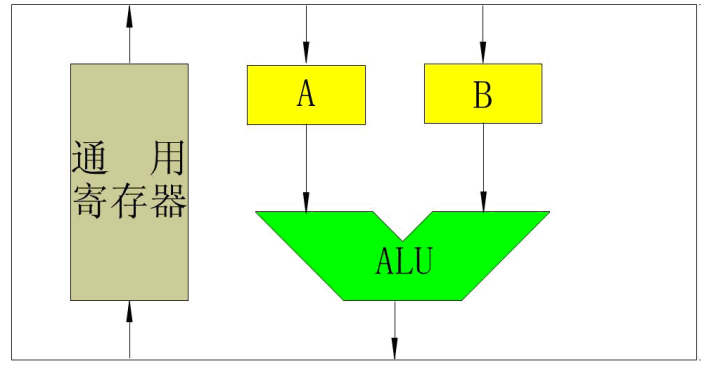
**实验三：运算器**

1. **实验题目**

运算器

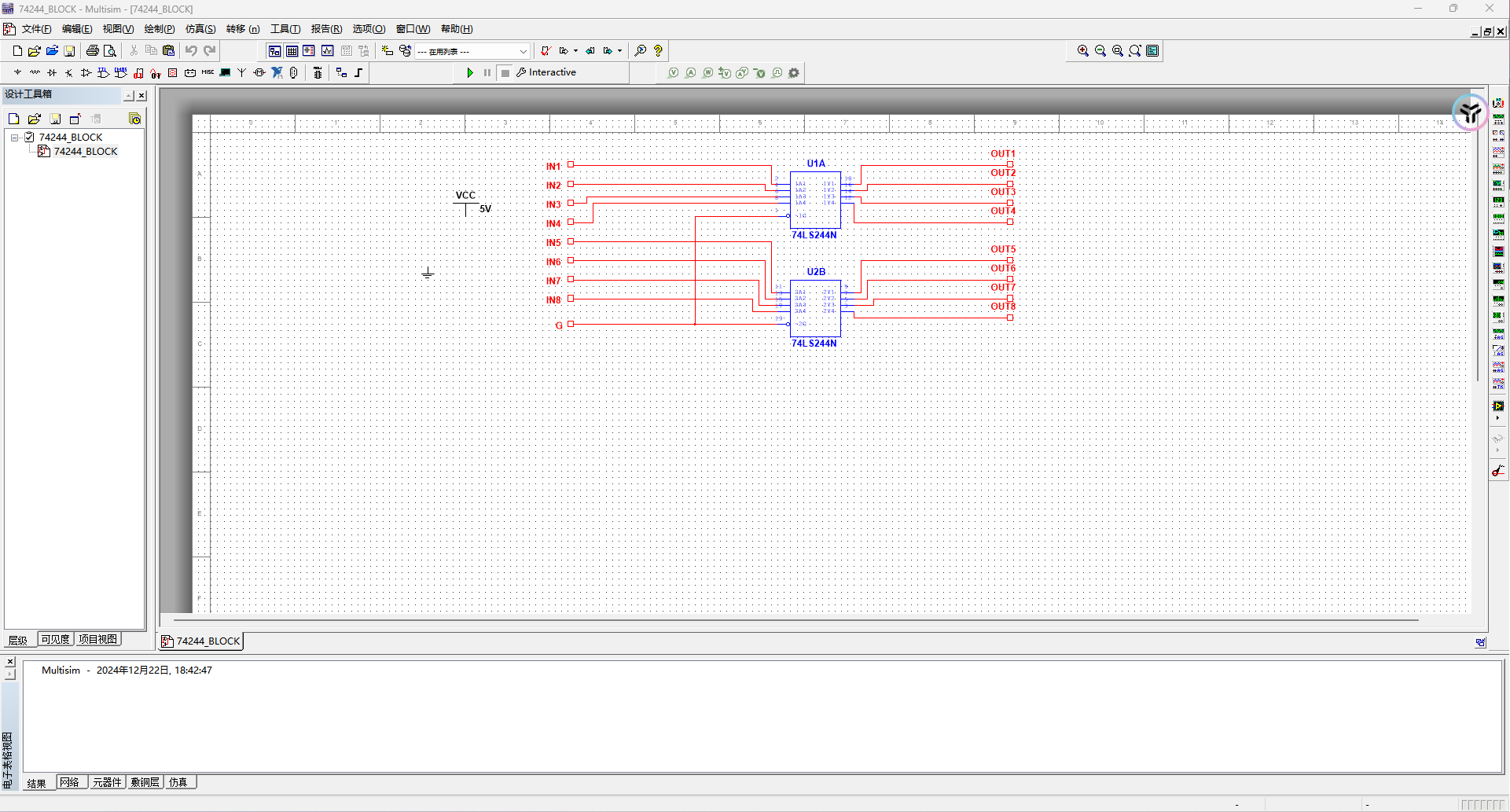
1. **实验目的**
2. 熟练掌握算术逻辑单元（ALU）的应用方法；
3. 进一步熟悉简单运算器的数据传送原理；
4. 画出逻辑电路图及布出美观整齐的接线图；
5. 熟练掌握有关数字元件的功能和使用方法。
6. 熟练掌握子电路的创建及使用。
7. **总体设计**
8. **实验原理**

本实验仿真单总线结构的运算器，原理如下图所示。电路图中，上右下三方的8条线模拟8位数据总线；K8产生所需数据；74244层次块为三态门电路，将部件与总线连接或断开，总线上只能有一个输入；两个74273层次块作为暂存工作寄存器DR1和DR2；两个74374层次块作为通用寄存器组；众多的开关作为控制电平或打入脉冲；众多的8段代码管显示相应位置的数据信息；核心为8位ALU层次块。



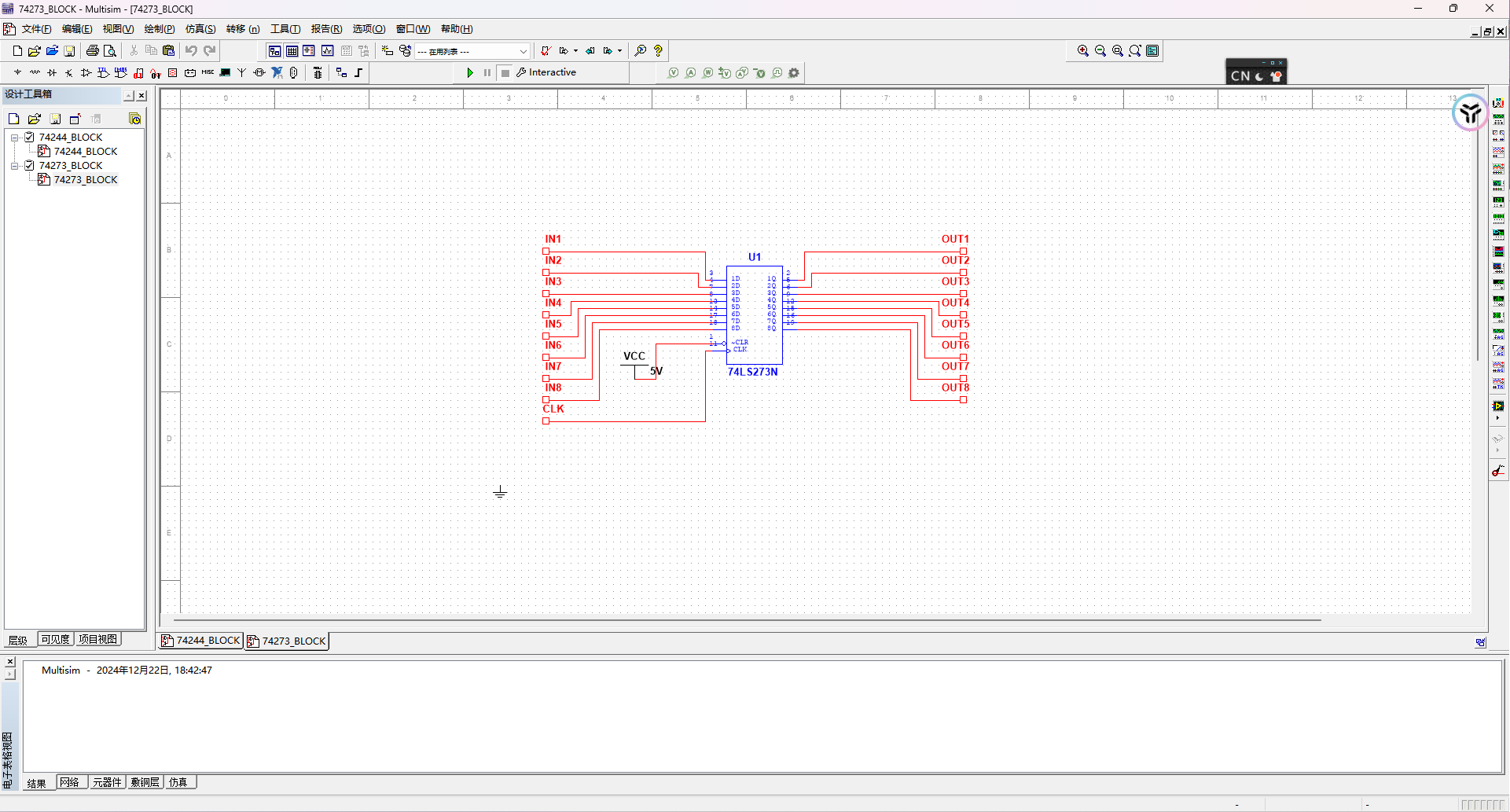
单总线结构的运算器示意图

1. **模块介绍**
2. 74244模块



此模块用于将两个普通的74LS244三态门芯片进行级联，两个芯片的G端统一连接，两个芯片的输入输出端分别负责低四位和高四位，实现控制八位总线通断。

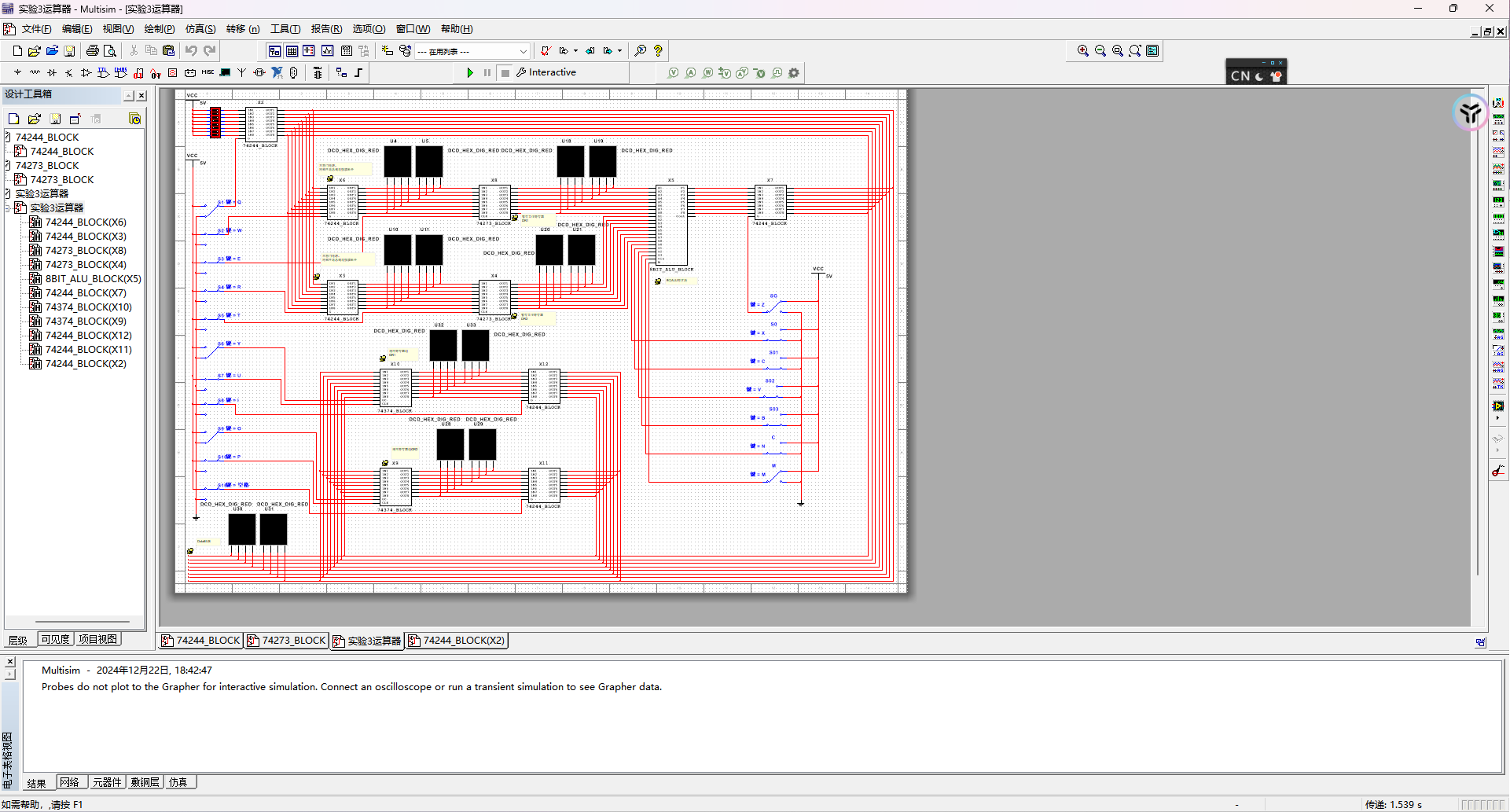
1. 74273模块



此模块用于将74LS273的清零端始终置为高电平，避免运行时出错。

1. **详细设计**

线路图如下



1. **实验结果与分析**
2. 说明整个电路工作原理。

答：整个电路仿真单总线结构的运算器完成算术和逻辑运算。具体实现为总线为8位，用一个八路开关控制总线上的数据输入；

1. 说明74LS244N的功能及其在电路中作用，及输入信号G有何作用；

答：74LS244N是三态门,控制电平G=1时隔绝电路。控制电平G=0时，74LS244N元件开始工作，选通电路。

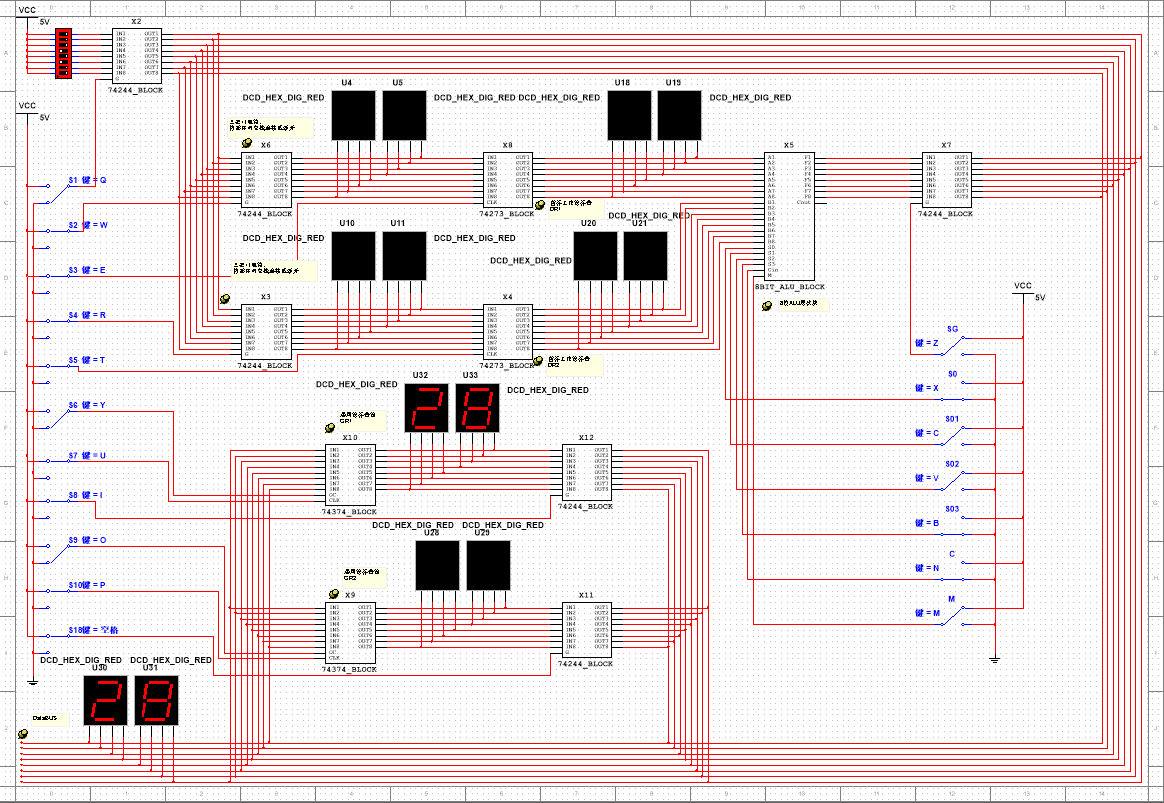
1. 说明74LS273N的功能及其在电路中作用，及输入信号CLK有何作用；

答：74LS273N为工作寄存器，输出直接接到ALU的输入端，CLK为接入的脉冲，CLK上升沿有效并且输出寄存的内容，下降沿锁存数据。

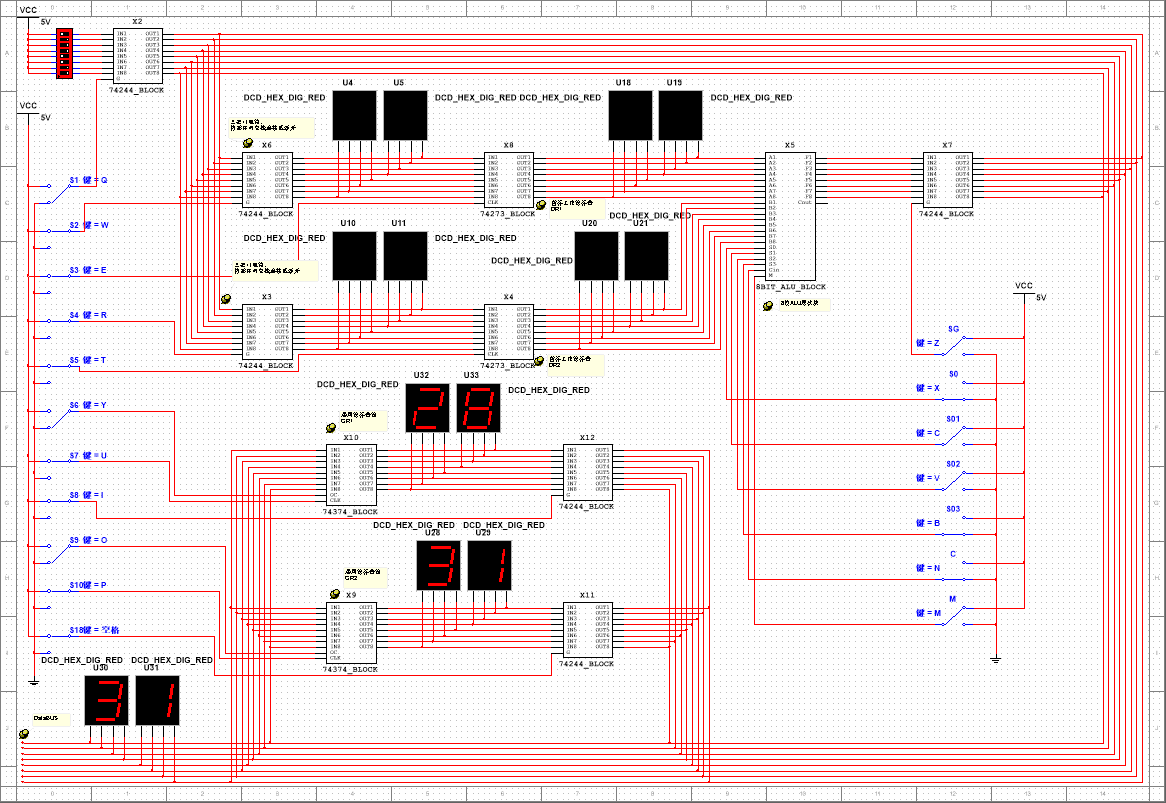
1. 说明74LS374N的功能及其在电路中作用，及CLK和OC有何作用；

答：74LS374N作为通用寄存器，同时具有三态门的作用。CLK上升沿有效，OC高电平为高阻态，逻辑操作不受影响，低电平工作。

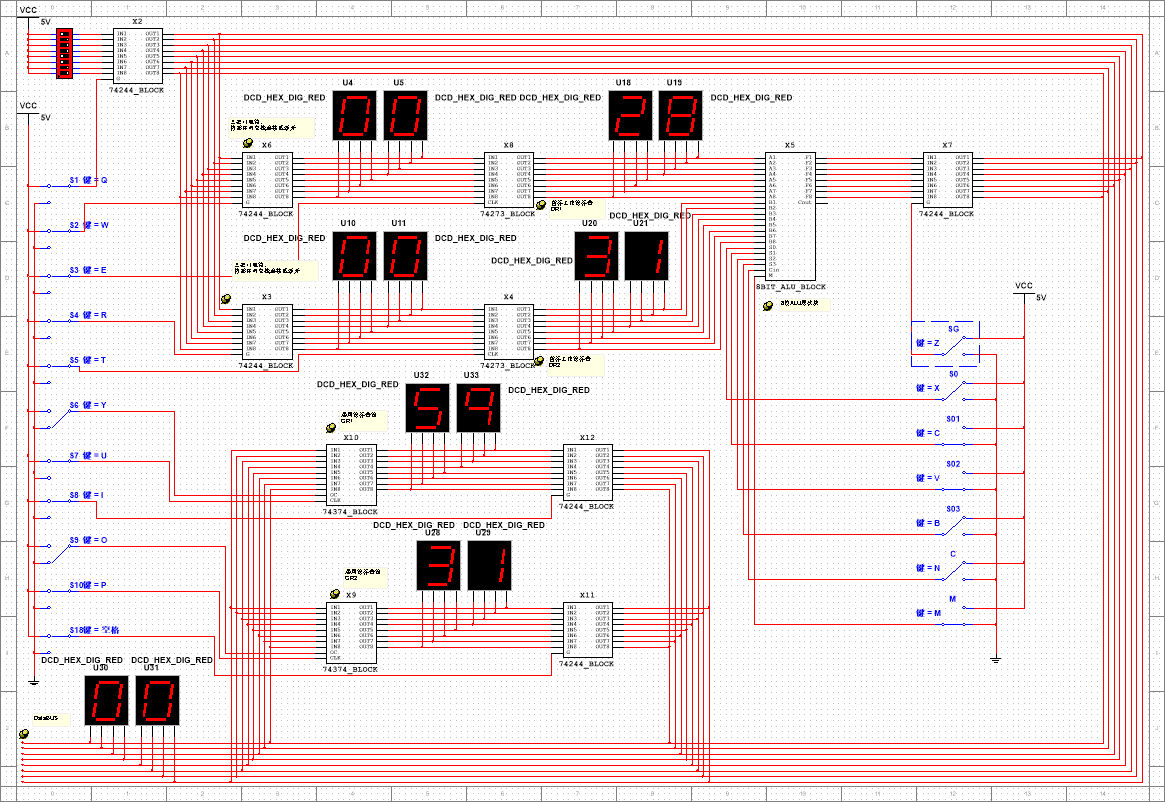
1. K8产生任意数据存入通用寄存器GR1。
2. 多路开关产生数据，S1接低电平，74244三态门模块X2选通将数据投放总线。
3. S6接低电平，74374芯片X10开始工作，把S7开关从低电平拨到高电平，数据存入GR1。同时保证74244模块X12的G端口置为高电平，避免投放总线。
4. 将X2的G端口置为高电平，封锁。



1. K8产生任意数据存入通用寄存器GR2，步骤和5、类似。



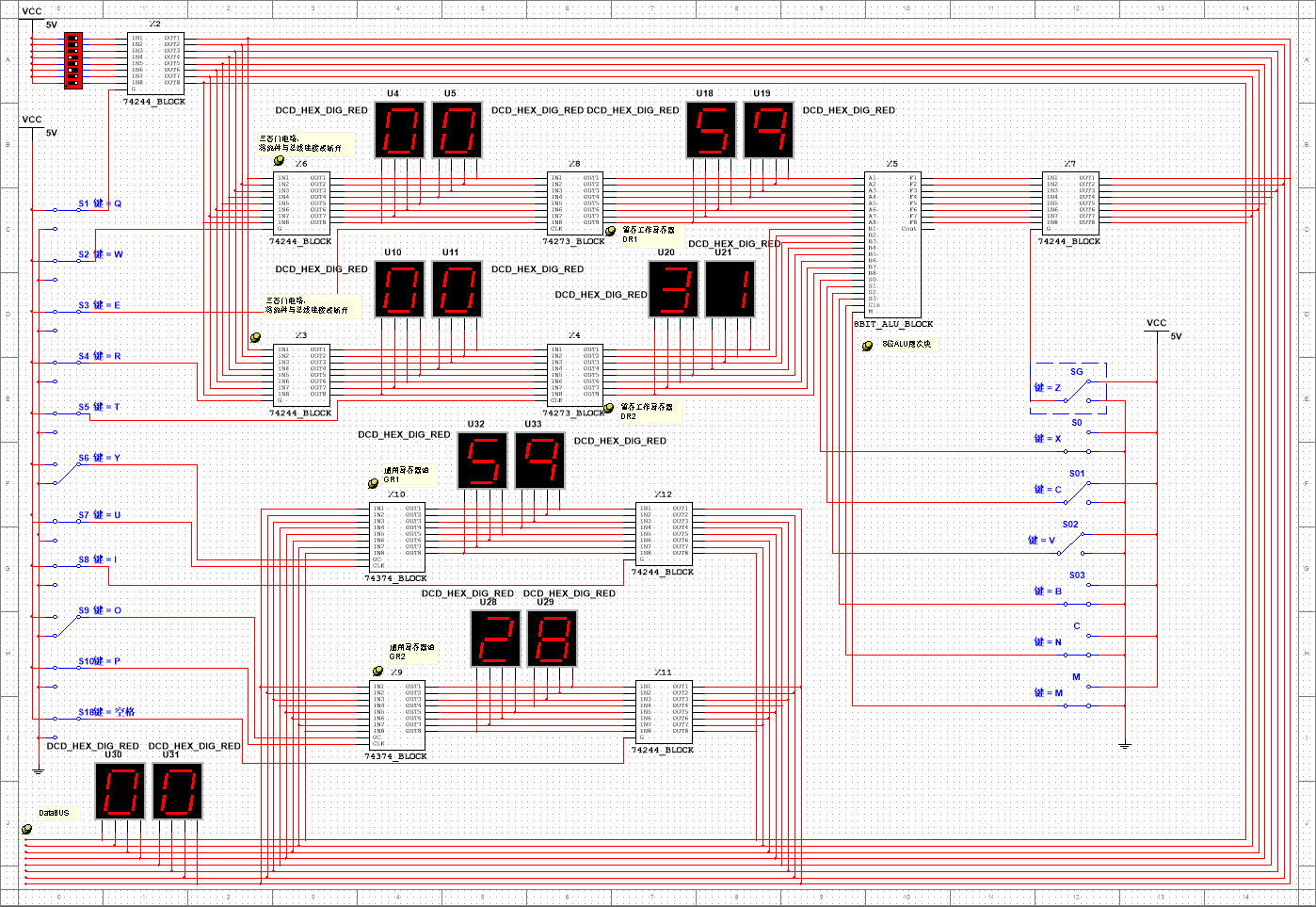
1. 完成GR1+GR2→GR1。
2. 将X12的G端口置为低电平，使GR1寄存器的内容投放总线。
3. 将X6的G端口置为低电平，并把X8的CLK时钟脉冲从低电平变成高电平，使总线数据存入寄存器DR1。
4. 将X6和X12的G端口置为高电平。
5. 同理将GR2的数据存入DR2。
6. 将八位算术逻辑单元的运算选择控制端分别置为S0=1、S1=0、S2=0、S3=1，进位标志Cin置为1表示低位无进位，同时将M状态控制端置为0，选择算术运算。
7. X7三态门的G端置为0，将运算结果投放总线。
8. 将X10的OC端置为低电平，并且将CLK从低电平变为高电平，将总线的计算结果存入GR1。
9. 关闭X7的G端口。



结果分析：29H+31H=59H，结果和GR1中存储的结果一致。

1. 完成GR1-GR2→GR2。

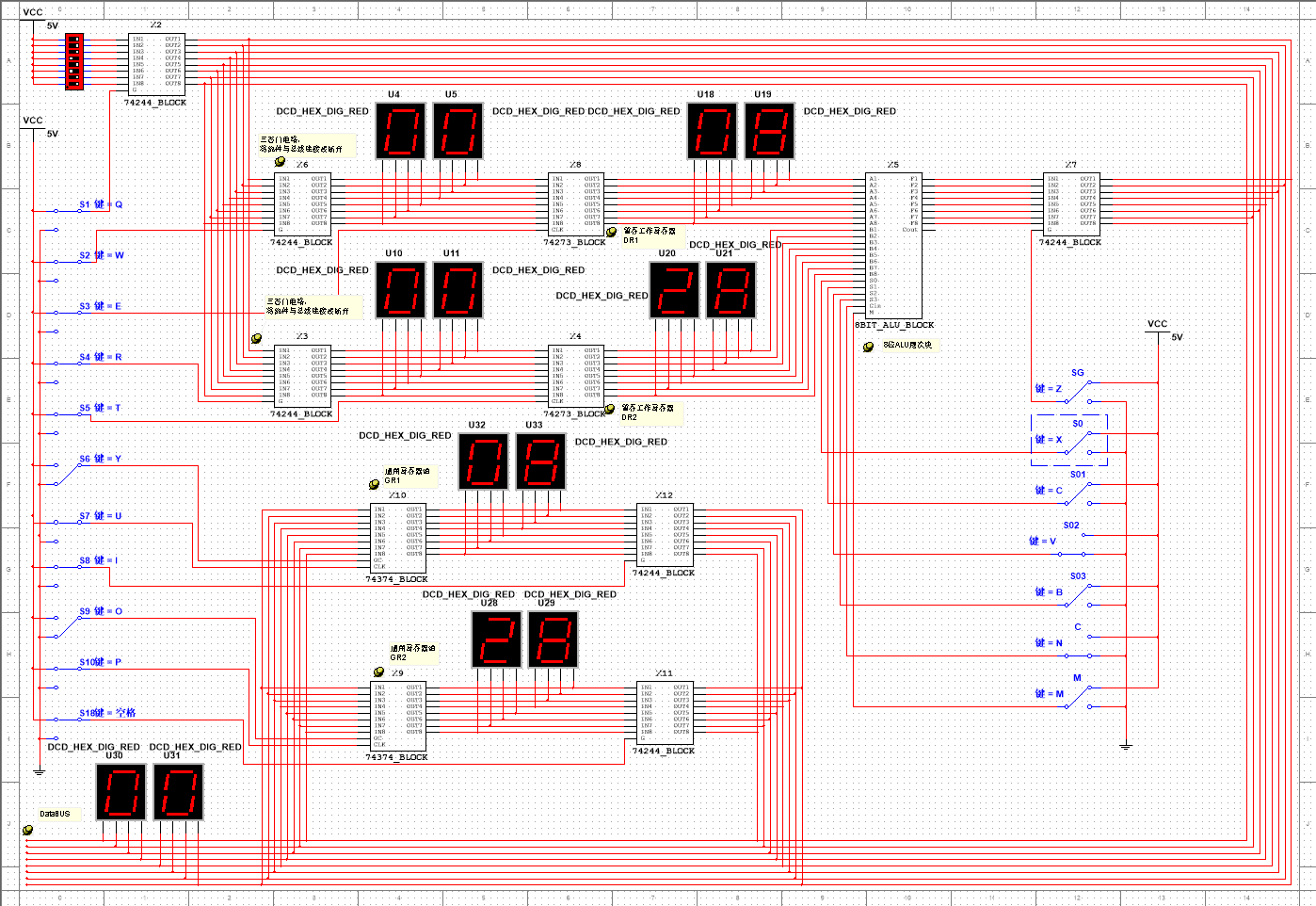
步骤与7、大致相似，不同之处在于算术逻辑单元的运算选择控制端分别置为S0=0、S1=1、S2=1、S3=0，进位标志Cin置为0。



结果分析：59H-31H=28H，结果和GR2中存储的结果一致。

1. 完成GR1∧GR2→GR1。

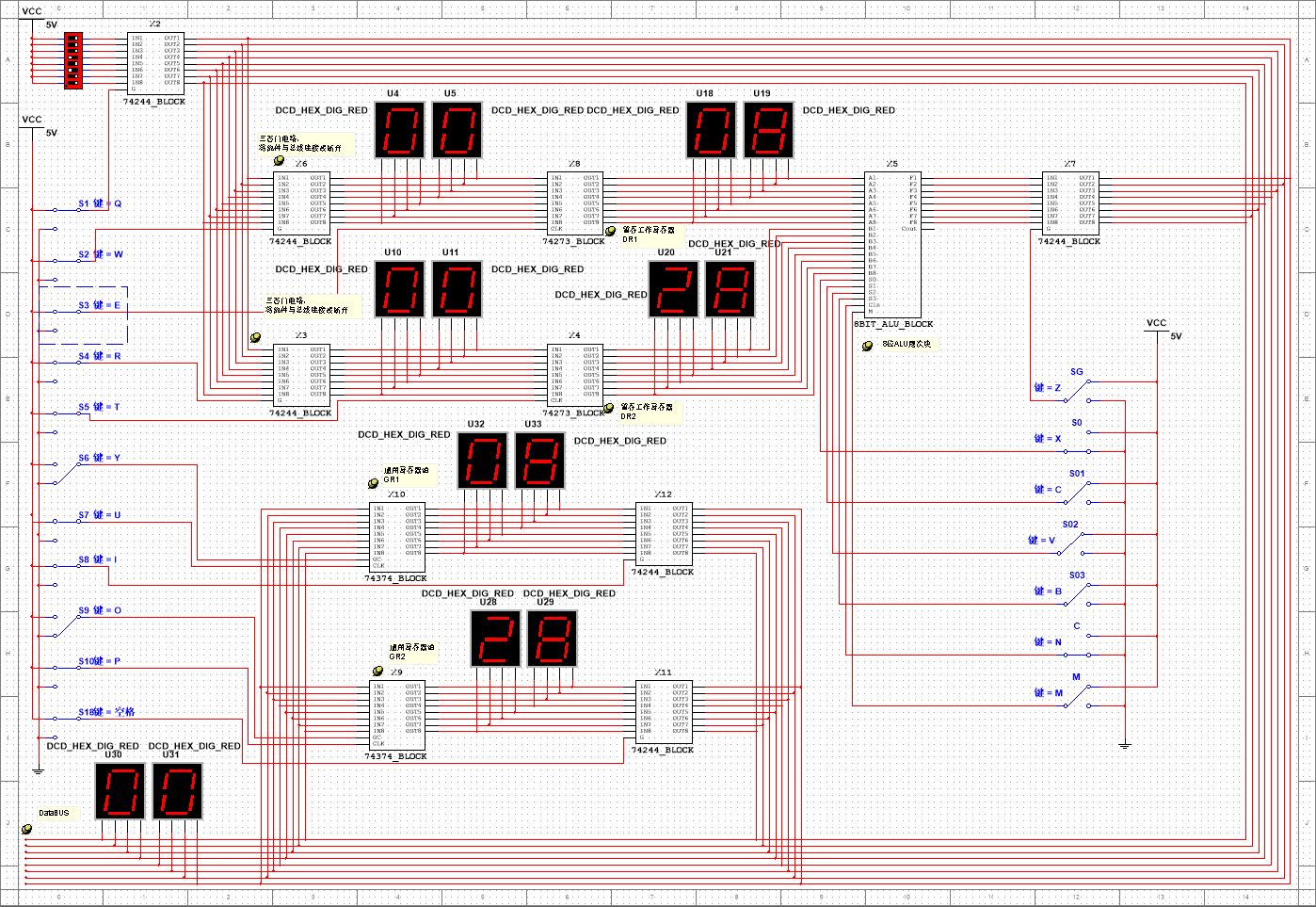
步骤与7、大致相似，不同之处在于算术逻辑单元的运算选择控制端分别置为S0=1、S1=1、S2=0、S3=1，同时将M状态控制端置为1，选择逻辑运算。



结果分析：59H=01011001B，28H=00101000B，59H∧28H=00001000B=08H，结果和GR1中存储的结果一致。

1. 完成GR1∨GR2→GR2。

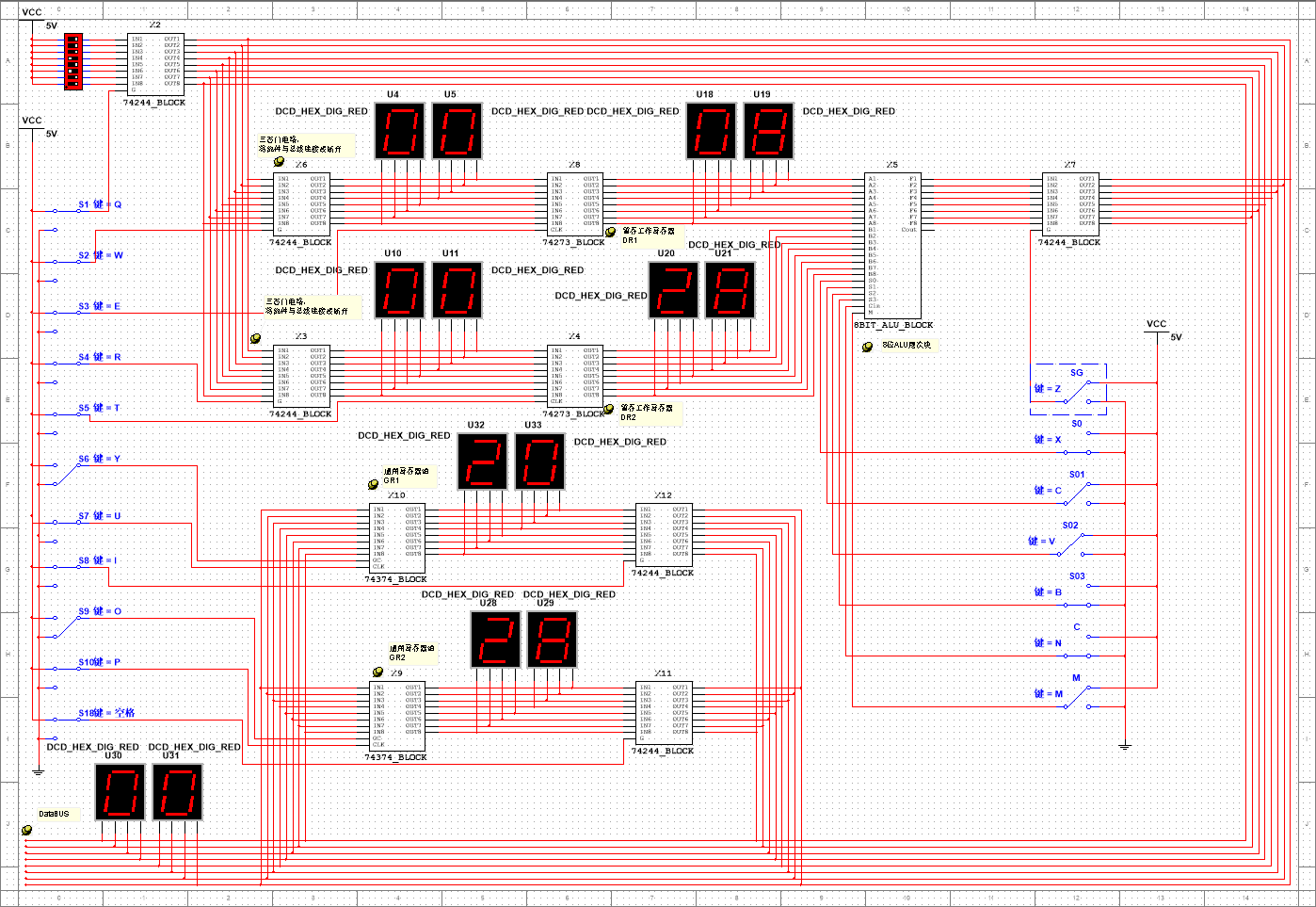
步骤与7、大致相似，不同之处在于算术逻辑单元的运算选择控制端分别置为S0=0、S1=1、S2=1、S3=1，同时将M状态控制端置为1，选择逻辑运算。



结果分析：08H=00001000B，28H=00101000B，08H∨28H=00101001B=28H，结果和GR2中存储的结果一致。

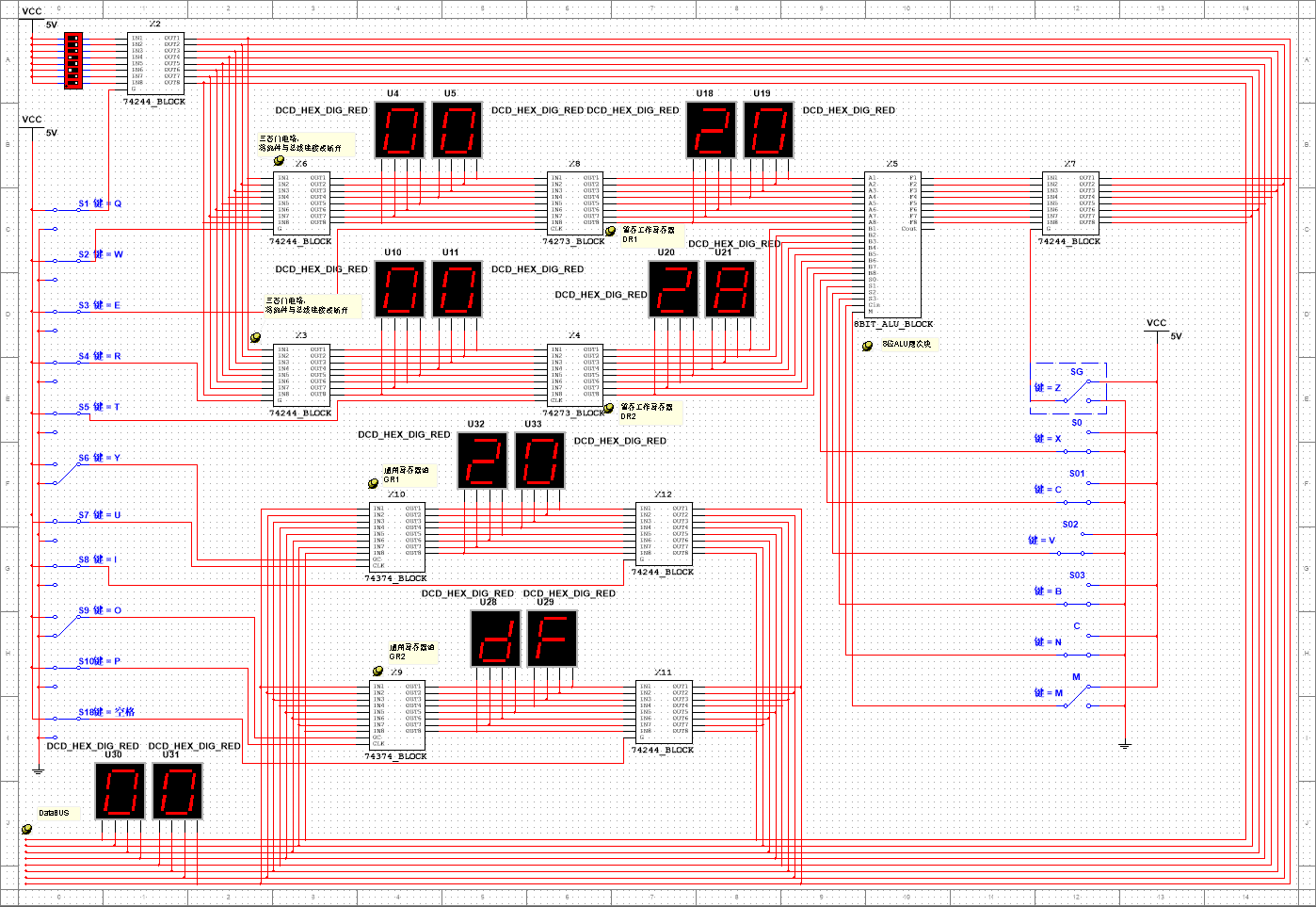
1. 完成GR1⊕GR2→GR1。

步骤与7、大致相似，不同之处在于算术逻辑单元的运算选择控制端分别置为S0=0、S1=1、S2=1、S3=0，同时将M状态控制端置为1，选择逻辑运算。



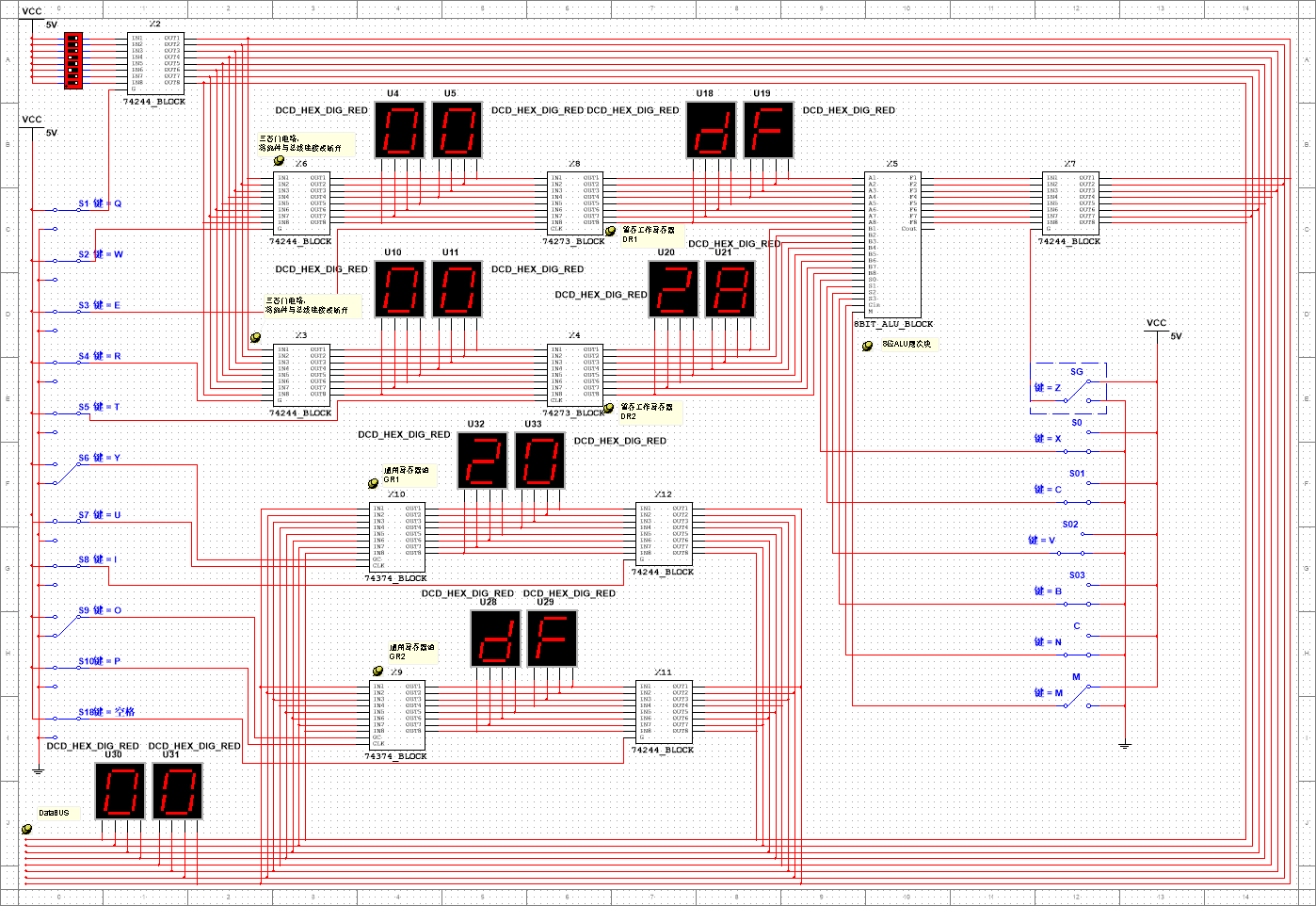
结果分析：08H=00001000B，28H=00101000B，08H⊕28H=00100000B=20H，结果和GR1中存储的结果一致。

1. ~GR1→GR2。（“~”表示逻辑非运算）
2. 将X12的G端口置为低电平，使GR1寄存器的内容投放总线。
3. 将X6的G端口置为低电平，并把X8的CLK时钟脉冲从低电平变成高电平，使总线数据存入寄存器DR1。
4. 将X6和X12的G端口置为高电平。
5. 将八位算术逻辑单元的运算选择控制端分别置为S0=0、S1=0、S2=0、S3=0，同时将M状态控制端置为1，选择逻辑运算。
6. X7三态门的G端置为0，将运算结果投放总线。
7. 将X9的OC端置为低电平，并且将CLK从低电平变为高电平，将总线的计算结果存入GR2。
8. 关闭X7的G端口。



结果分析：20H=00100000B，~20H=11011111B=DFH，结果和GR2中存储的结果一致。

1. ~GR2→GR1。
2. 将X11的G端口置为低电平，使GR2寄存器的内容投放总线。
3. 将X6的G端口置为低电平，并把X8的CLK时钟脉冲从低电平变成高电平，使总线数据存入寄存器DR1。
4. 将X6和X12的G端口置为高电平。
5. 将八位算术逻辑单元的运算选择控制端分别置为S0=0、S1=0、S2=0、S3=0，同时将M状态控制端置为1，选择逻辑运算。
6. X7三态门的G端置为0，将运算结果投放总线。
7. 将X10的OC端置为低电平，并且将CLK从低电平变为高电平，将总线的计算结果存入GR1。
8. 关闭X7的G端口。



结果分析：20H=00100000B，~20H=11011111B=DFH，结果和GR1中存储的结果一致。

1. **小结与心得体会**
2. 刚开始看实验要求和线路图感觉很迷茫，但是在沉下心了解各个模块的功能，结合实验二74181芯片的功能分析单总线系统运行过程后，实验也就迎刃而解了。
3. 由于系统比较复杂，一开始我很难记住各个模块的作用，于是我在每个模块旁都放置了注释，这样模块的功能就一目了然了。
4. 在完成实验指导书上的实验步骤时，由于没有看清题意，我每次计算忘记更新暂存寄存器DR1的数值，导致结果出错，今后在做实验之前一定要看清楚题目要求，不要做无用功。
5. 一开始连线时，由于对模块的了解不清晰，并没有意识到有一些功能要通过模块来实现。因此导致连线文件十分复杂，并且芯片之间的关系也不明确。后来经过同学指点，我采用了模块化的设计，大大简化了设计图的复杂度。
6. 在实验中由于粗心，在投放总线数据时没有将ALU结果封锁，导致总线上同时有ALU和八路开关的数据，总线上收到的数据出错。以后的实验中一定要注意单总线系统中总线上不能同时接受多个数据。
7. 通过本实验，我学会了74181芯片的用法，学会了向总线投放数据，寄存器存储数据，运算逻辑单元处理数据的方法，培养了绘制和分析电路图的能力。
8. 本题虽然给了电路图，但是仅仅照着葫芦画瓢是远远不够的。本题涉及了数据存储，转运和运算等复杂操作，如果不理解各个模块之间的关系，完全不知道该如何下手。

**实验四：字发生器及跑马灯**

1. **实验题目**

字发生器及跑马灯

1. **实验目的**

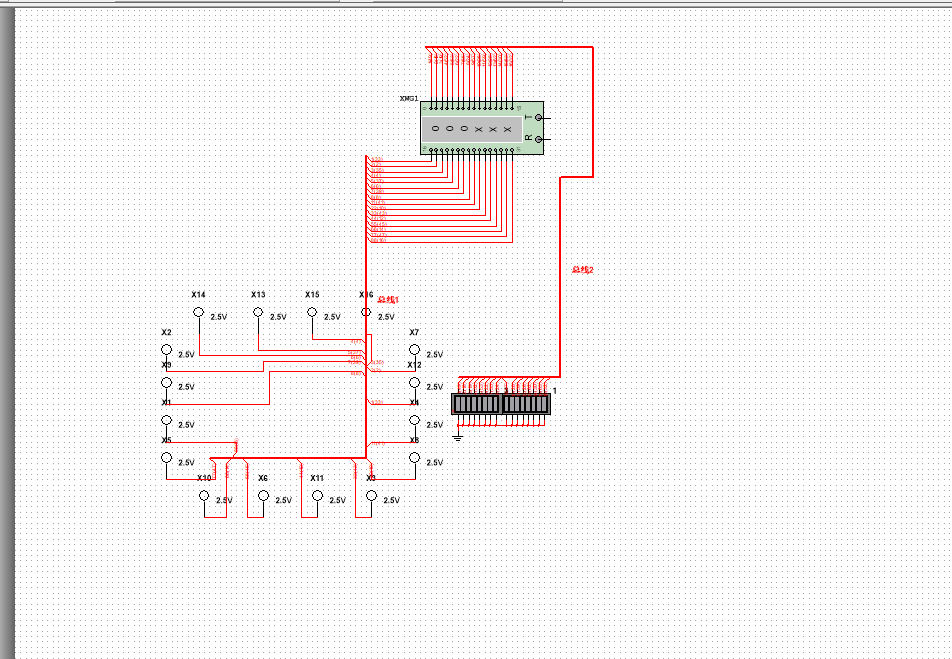
了解字发生器的使用方法

1. **总体设计**
2. **实验原理**

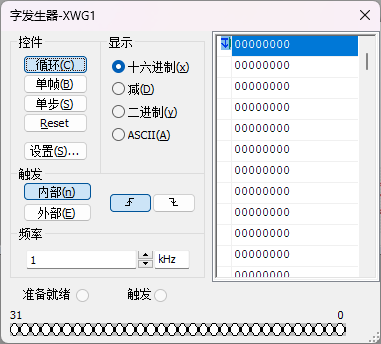
通过设定字发生器的数据，使探针依次点亮，形成跑马灯。

1. **设计步骤**
2. 将各个灯的引脚依次连到字发生器引脚上。
3. 根据设计的灯点亮的顺序和形状设计字发生器产生字的序列，其中每一位代表一个引脚的电平。
4. 开始模拟，记录结果。
5. **详细设计**

线路图如下



1. **实验结果与分析**



分析：字发生器有32位同步逻辑信号，每一位分别对应0-31号引脚，当对应的位为高电平1时，对应的灯随即亮，通过依次0-15位对应的灯亮并且使字发生器循环输出逻辑信号实现了跑马灯的效果。

1. **小结与心得体会**
2. 实验四相对其它的实验最简单，只需要根据指导书上的电路图，按图索骥即可实现跑马灯功能。
3. 虽然易于实现，但是我也明白了字发生器的功能其实类似于开关接电源或者接地，给电路输入一个高电平或者低电平，从而达到了控制电路元件的作用。
4. 通过本题，我学会了字发生器的用法，为实验五将字发生器和实验三运算器相结合打下了基础。

**实验五：模拟微程序实现指令**

1. **实验题目**

模拟微程序实现指令

1. **实验目的**

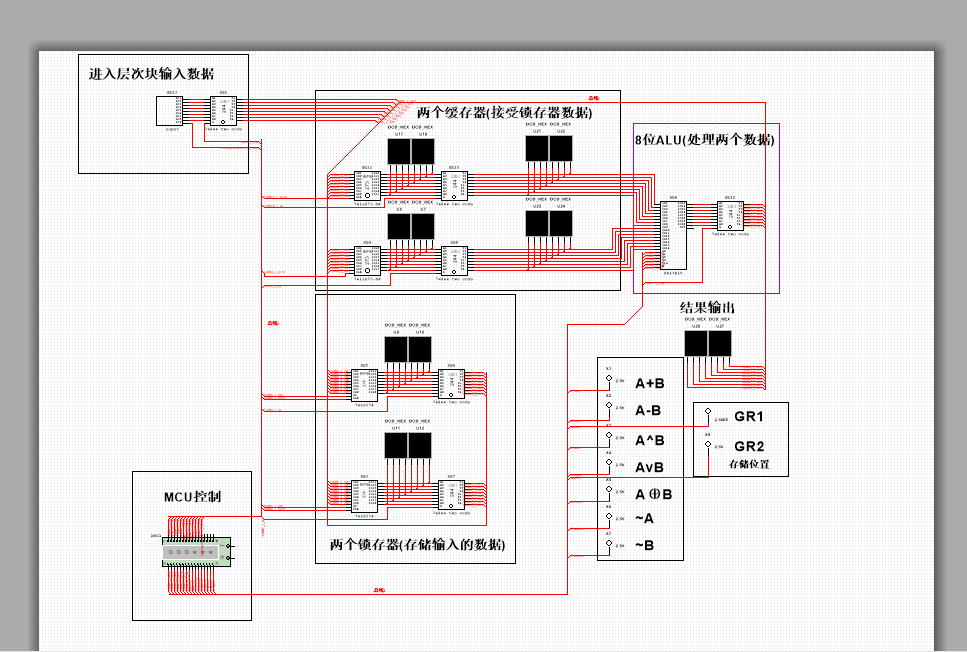
模拟微程序实现机器语言指令

1. **总体设计**
2. **实验原理**

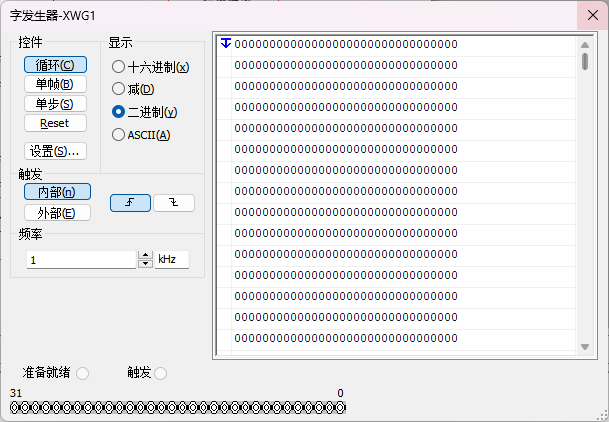
字发生器的一行输出数据可以作为一条微指令，一条机器语言指令由若干条微指令组成。用字发生器的输出取代实验三线路图中开关，模拟微指令自动执行实现一条机器语言指令。

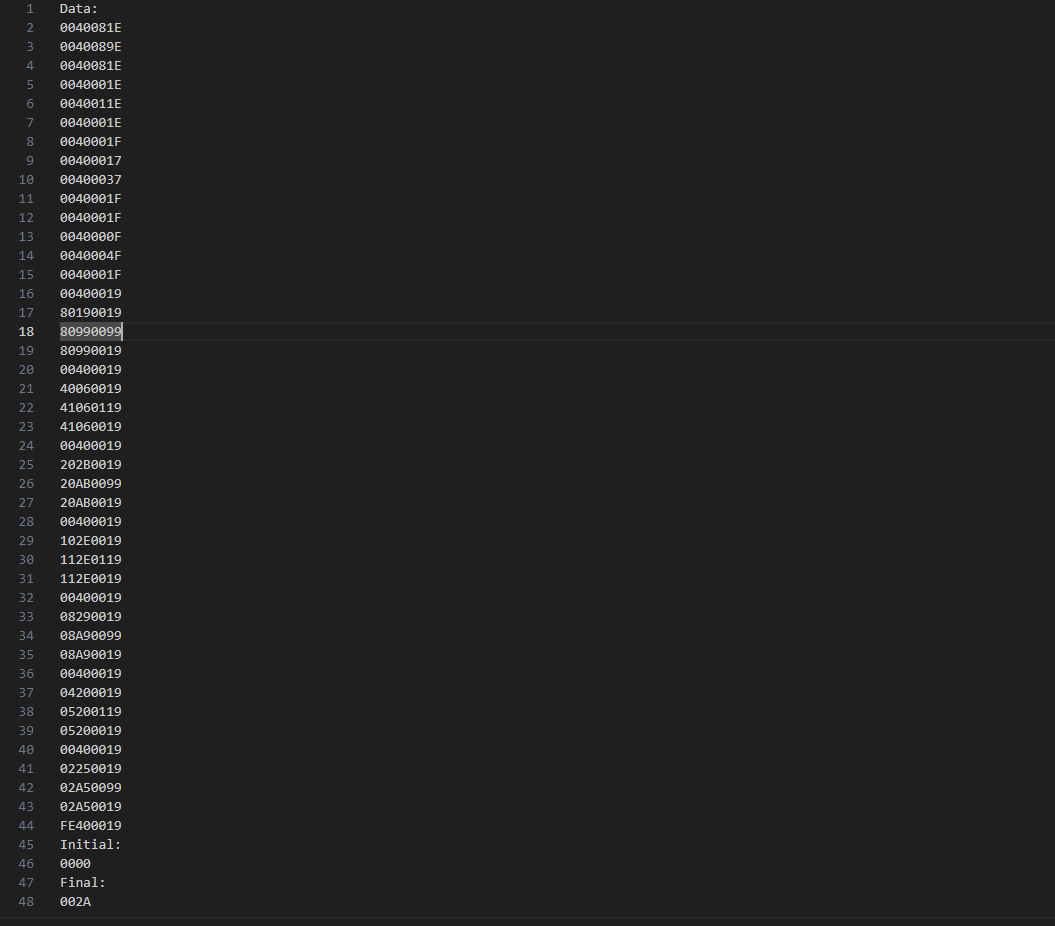
1. **实验步骤**
2. 以实验三线路图为基础，除去所有的开关，并用字发生器的输出信号控制。
3. 根据各操作涉及到的开关，设计每步操作字发生器的字序列。
4. 开始模拟，记录每一步操作LED灯的输出结果以及数据流动。
5. **详细设计**

线路图如下

本模块：

我们在字发生器中填入若干二进制序列





**然后观察右侧的亮灯情况，以及查看相应的数码管值的变化。**

1. **小结与心得体会**
2. 由于实验中需要将数据分时投放到总线上，且所有操作必须自动完成，因此使用了两个八路开关来存储待投放的数据。
3. 字发生器的序列可以保存为.dp文件，这样在每次运行时无需重新输入字序列，而是可以直接读取配置文件，从而节省时间。
4. 为了使字发生器的序列更加易于理解和记忆，控制位会尽量按操作时间的先后顺序进行分组，以便于操作时的快速识别。
5. 尽管借鉴了实验三的经验，然而每步操作的控制项仍然庞大繁杂，因此在输入字序列时必须格外细致，并且要保持足够的耐心。
6. 为了简化字序列的编写过程，我将每一位控制开关及其功能整理成表格，并将需要实现的操作列成清单。这样，在生成最终的字序列时，操作变得更加清晰。
7. 虽然确定字序列的过程费时费力，但一旦完成某个序列的设定，后续工作会变得轻松许多。许多运算操作相似，只需找出其中的差异，如运算选择控制端S0~S3、状态控制端M及存储的通用寄存器等，然后修改对应位即可。
8. 在测试字序列的正确性时，有时会遇到在执行过程中需要重新开始的情况，往往是因为忘记关闭并重新打开模拟，导致寄存器内容未被清空，进而产生误差。通过多次尝试，最终找出了问题的根源。这个问题的本质还是因为对各模块和芯片功能的不熟悉。
9. 尽管在完成这个任务过程中耗费了大量的时间和精力，并且出现了许多错误，但在最终完成时，我内心充满了喜悦与骄傲。同时，我也意识到，面对复杂的任务时，只有静下心来、一步一步推进，才能最终成功。