**实验0 ROM仿真**

1. **实验目的**

（1） 掌握ROM的工作原理；

（2） 画出逻辑电路图及布出美观整齐的接线图；

（3） 掌握ROM读出原理。

2、**总体设计（含背景知识或基本原理与算法、或模块介绍、设  
计步骤等）**

1. **实验原理与背景知识**

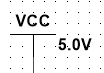
在设计时，使用了DCD\_HEX数码管以及8421BCD码的基本原理。

DCD\_HEX数码管是内部自带译码的七段数码显示器，他有四个输入端，可以直接接到编码器的输出端。DCD-HEX数码管规定4条引脚从左到右对应BCD码位的左高右低。

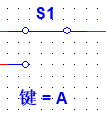
8421BCD码是最基本和最常用的BCD码，它和四位自然二进制码相似，各位的权值为8、4、2、1，故称为有权BCD码。和四位自然二进制码不同的是，它只选用了四位二进制码中前10组代码，即用0000~1001分别代表它所对应的十进制数，余下的六组代码不用。

相关存储元由二极管组成，若有二极管，那么数据线上可以读出1，若无则读0；通过二极管的合理排列则可以显示相应的输出。

1. **模块介绍**



VCC电源：提供高电平



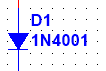
SPDT：双刀单掷开关



7404N：非门



7408N：与门

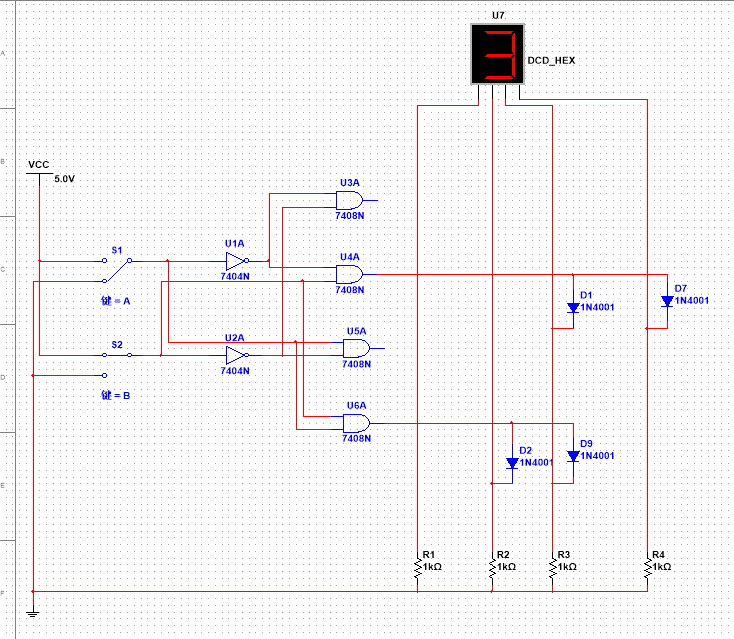


1N4001：二极管



电阻丝（1k）

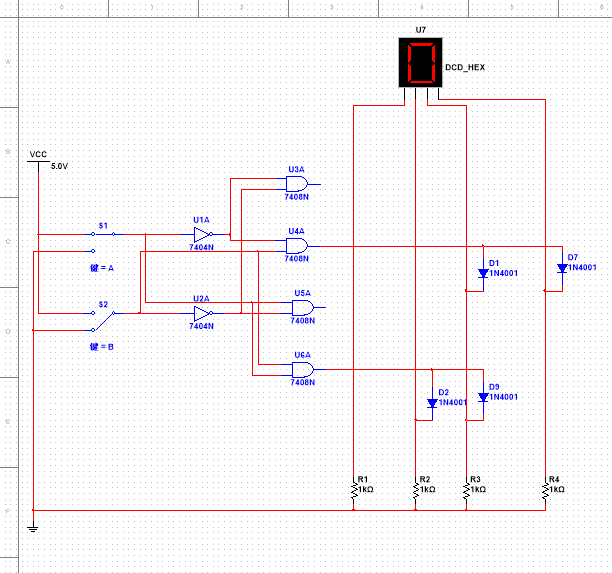
1. **设计步骤**
2. 通过阅读实验指导书了解multisim软件的使用与实验的基本要求；
3. 根据实验指导书选取所需要的元器件，并且依次连线
4. 将自己学号的后四位作为ROM的存储信息并读出
5. **详细设计（含主要的数据结构、程序流程图、关键代码等）**
6. **整体电路图**



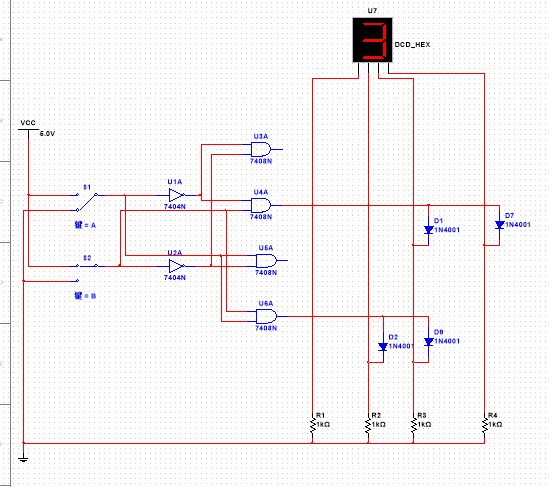
1. **思路**

单刀双掷开关A,B：接通高低电平，通过非门和与门相结合，U3=A’B’，U4=A’B,U5=AB’,U6=A’B’， 在与门输出端形成相应的高低电平，在输出端为高电平的传输线上若遇到二极管，则最终所对应的DCD-HEX数码管的相应位置输入为1，否则输入为0，最终数码管可根据输入的四位二进制转换成十六进制数输出显示。

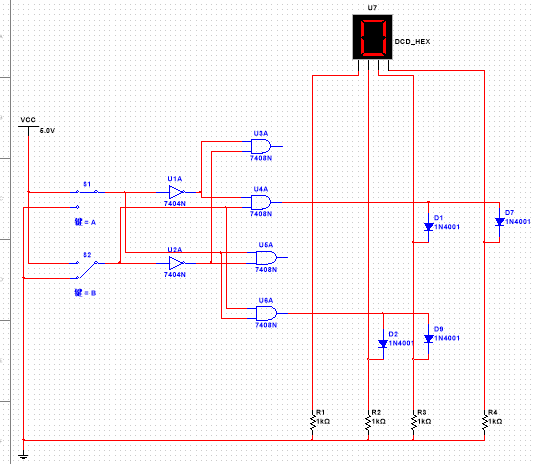
1. **实验结果与分析**
   1. **第一位数字0**：BCD码0000，全部输入为0即可，这里采用的方法是不连接与门实现接地操作。



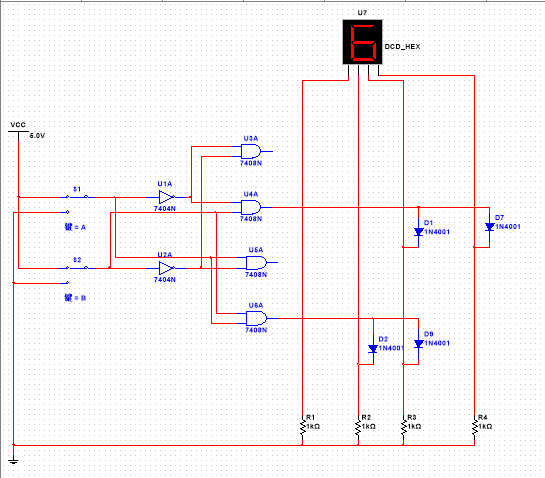
1. **第二位数字3：**BCD码为0011。

****

1. **第二位数字0：**BCD码为0000。

****

1. **第二位数字6：**BCD码为0110 。



1. **小结与心得体会**

实验0总体来说难度不大，但是由于不熟悉mutisim软件的运用与元器件的基本功能，所以耗费时间较长。

通过本次实验的学习，我更加深刻地认识了mutisim的实验，以及仿真实验板的搭建，也通过上网查找资料，学习到了BCD码与DCD\_HEX数码管的使用方式。

仿真实验板搭建完成后，我第一时间不知道如何去改变数码管的显示数字，通过不断的试错，最终才发现数码管输入的逻辑表达。在以后的实验中，我将更加重视由物理电路层向逻辑表达的转换。

**实验一 验证74LS181运算和逻辑功能**

**1、实验目的**

（1） 掌握算术逻辑单元（ALU）的工作原理；

（2） 熟悉简单运算器的数据传送通路；

（3） 画出逻辑电路图及布出美观整齐的接线图；

（4） 验证4位运算功能发生器(74LS181)组合功能。

**2、总体设计（含背景知识或基本原理与算法、或模块介绍、设  
计步骤等）**

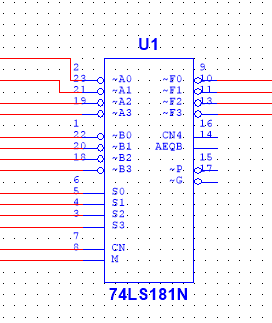
1. **实验原理**

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)

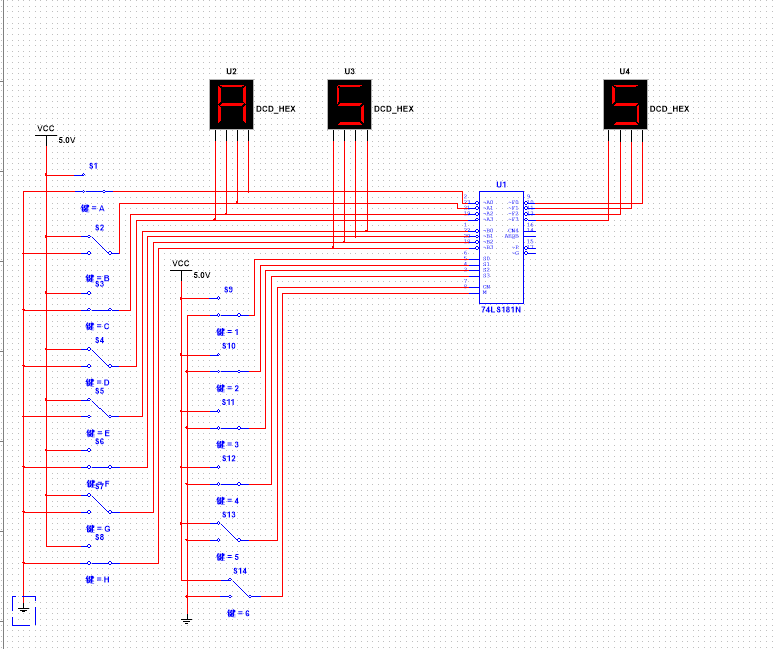
1. **模块介绍**



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. **设计步骤**
2. 阅读实验指导书，了解74138的逻辑功能；
3. 设计电路图；
4. 运行仿真实验板，填写实验数据。
5. **详细设计（含主要的数据结构、程序流程图、关键代码等）**
6. **4位ALU验证电路示意图**

****

1. **原理介绍**

74181的输入端A3A2A1A0，B3B2B1B0分别由开关ABCDEFGH控制，接VCC高电平时输入为1，接地时输入为0，此外U2,U3两个数码显示管分别接在DCBA,HGFE上，以显示两个运算数。

开关1、2、3、4控制操作输入S0S1S2S3,以确定操作。开关5接进位输入CN，开关6接逻辑运算输入。数码显示管U4接输入信号为74138的输出信号F3F2F1F0，显示的是运算结果。

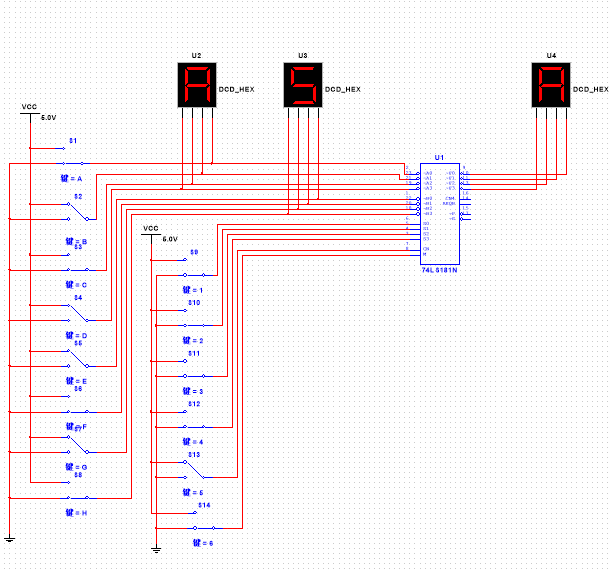
1. **实验结果与分析**

验证74LS181型4位ALU的逻辑算术功能，填写下表：

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

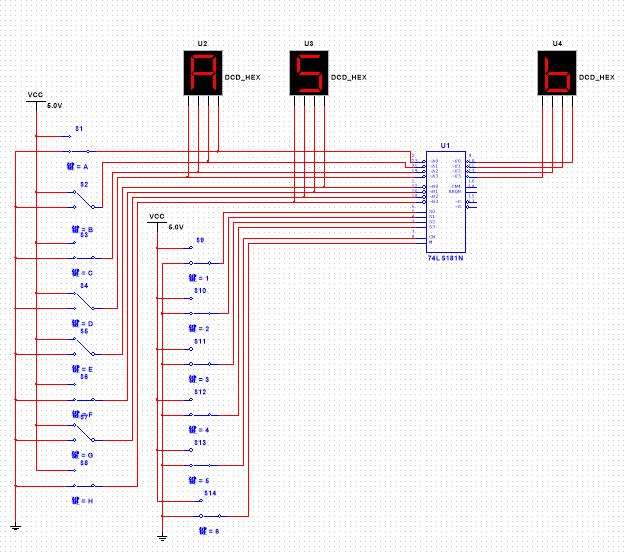
1. **第一组数据**

当S3 S2 S1 S0=0000，数据1=AH，数据2=5H，M=0，CN=1时的测试，此时无进位，且两数进行算术运算，F=A,可以看到U4数码管显示结果为AH，仿真结果如图所示。（其中H表示为16进制数）



1. **第二组数据**

S3 S2 S1 S0=0000，数据1=AH，数据2=5H， CN=0，M=0时的情况测试，此时有进位，两数进行算术运算，F=A+1， U4数码管显示结果为bH，仿真结果如图所示。



1. **第三组数据**

S3 S2 S1 S0=0000，数据1=AH，数据2=5H CN=0，M=1时的情况测试，此时无进位，两数进行逻辑运算，F=A’，可以看到U4数码管显示结果为5H（这是数据1的四位二进制形式取反后的结果），仿真结果如图所示。

1. **小结与心得体会**

本实验难度不大，只是需要调试的运算变多了，导致消耗了比较多的时间。同时由于数据量的增加，需要有耐心地验证每一组数据，确保结果不会出错。同时，在本实验中进一步了解了74138的逻辑功能以及每个引脚的作用。

**实验二 运算器（2）**

**1、实验目的**

（1） 熟练掌握算术逻辑单元（ALU）的应用方法；

（2） 进一步熟悉简单运算器的数据传送原理；

（3） 画出逻辑电路图及布出美观整齐的接线图；

（4） 熟练掌握有关数字元件的功能和使用方法；

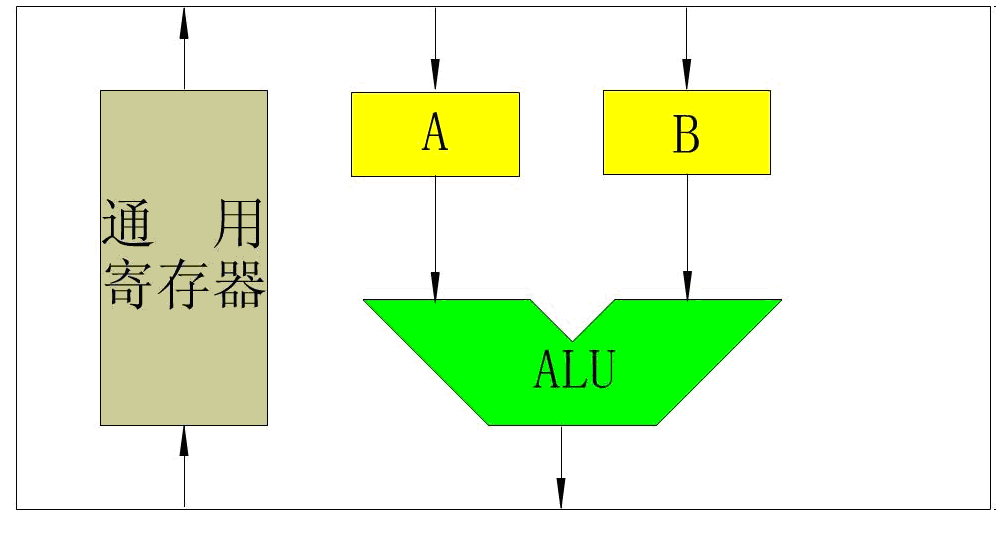
（5） 熟练掌握子电路的创建及使用。

**2、总体设计（含背景知识或基本原理与算法、或模块介绍、设  
计步骤等）**

1. **实验原理**

本实验仿真单总线结构的运算器，原理如图2-2所示。相应的电路如图2-3所示。

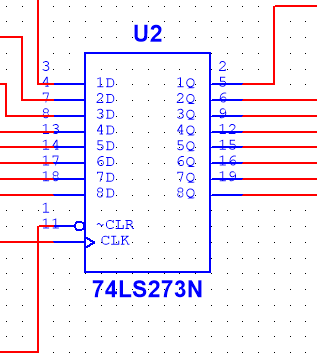
电路图中，上右下三方的8条线模拟8位数据总线；K8产生所需数据；74244层次块为三态门电路，将部件与总线连接或断开，切记总线上只能有一个输入；两个74273层次块作为暂存工作寄存器DR1和DR2；两个74374层次块作为通用寄存器组（鉴于电路排列情况，只画出两个通用寄存器GR1和GR2，如果可能的话可设计4个或8个通用寄存器）；众多的开关作为控制电平或打入脉冲；众多的8段代码管显示相应位置的数据信息；核心为8位ALU层次块。



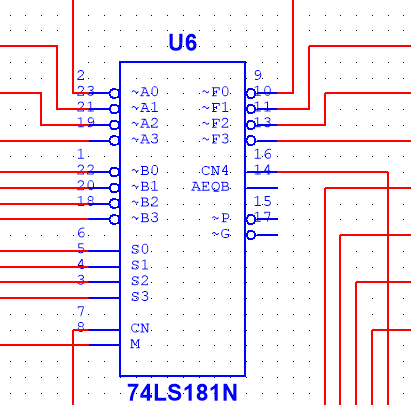
1. **背景知识**
2. **74LS244N芯片介绍：**74LS244N芯片是一种4路数据总线缓冲器，当门控信号G为低电平时，芯片工作，将A1–A4输入端的高低电平传输到Y1~Y4输出端；当门控信号G为高电平时，芯片成高阻态。



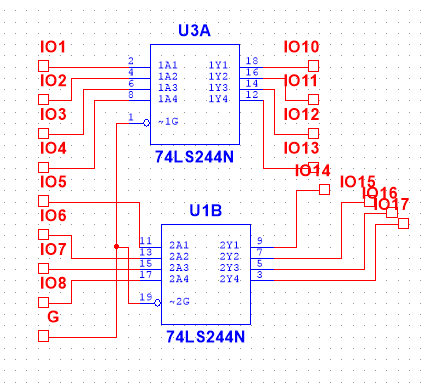
1. **74LS273N芯片介绍：**74LS273芯片的作用是缓冲时钟和直接清除输入，数据独立输入到各触发器。其中1D–8D为数据输入端，1Q–8Q为数据输出端。CLR是复位脚，当CLR脚位低电平时，无论有无脉冲还是输入端是高或低电平，输出端始终位低电平；当且仅当CLR脚为高电平时，输入端1D~8D的数据在脉冲上升期间被传送到输出端1Q–8Q。



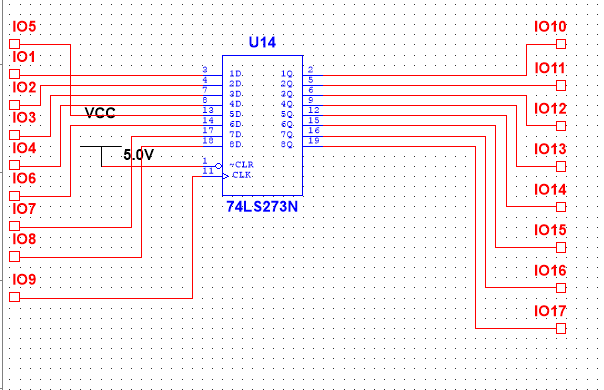
1. **74LS181N芯片介绍：**74LS181N芯片是主要进行算术和逻辑运算的芯片，可以作为处理器进行运算的核心部件。它对两个4位操作数进行逻辑或者算术运算等。



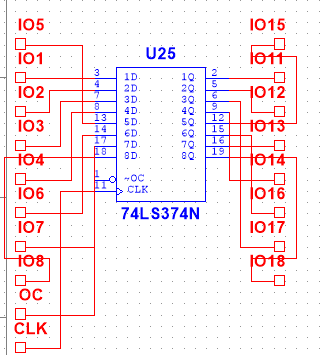
1. **详细设计（含主要的数据结构、程序流程图、关键代码等）**
2. **模块设计**
3. **74LS244N\_BLOCK：**该层次块中有2片74LS244N芯片，将1G和2G输入端接入同一条片选信号，保证两片芯片在低电平有效时同时工作，高电平时同时成高阻态，两片芯片工作时，可将8个输入端的数据传输到输出端。

****

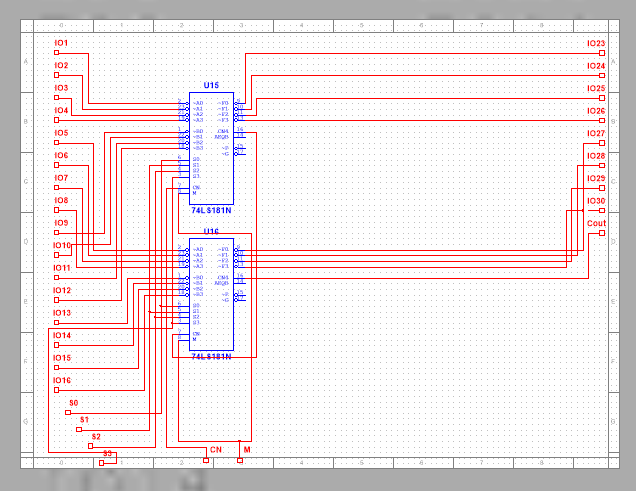
1. **74LS273\_BLOCK：**该层次块选取了一片74LS273N芯片，不同的是，在74LS273芯片CLR输入端接入了VCC高电平，这是为了保证该层次块只要有上升沿时钟脉冲传入就会将输入端值传入到输出端。

****

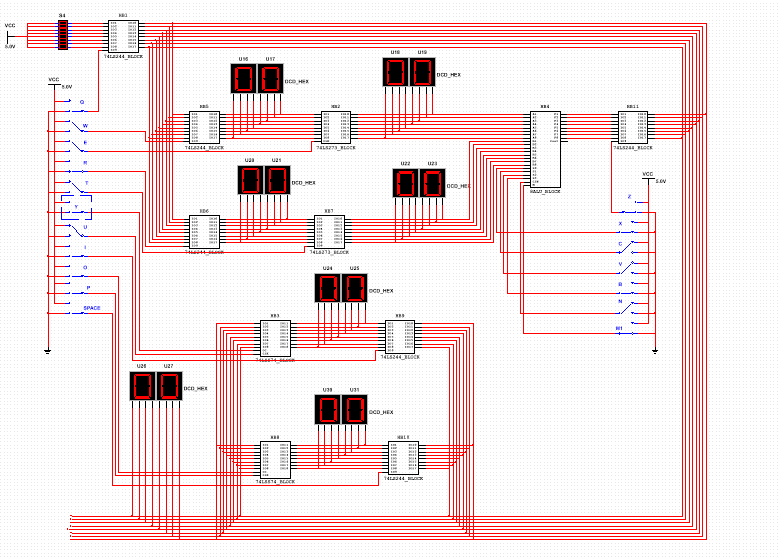
1. **74LS374\_BLOCK：**该层次块选取了一片74LS374N芯片，其实该层次块实现的就是74LS374芯片的功能，理解芯片作用后自然也会了解层次块。



1. **8bit\_AUL\_BLOCK：**该层次块选取了两篇74LS181N芯片，1A0–1A3，2A0–2A3的8个输入端口是数据1的输入，1B0–1B3，2B0–2B3的8个输入是数据2的输入，由于输入的数据位限制，只能选用两片芯片并保证其同时以相同的计算方式工作，运算出结果以输出端1F0–1F3，2F0~2F3输出，要注意的点是CN输入端口和CN4输出端口的连接，这是一种串行进位的思想。



1. **整体电路**

****

1. **实验结果与分析**

**（1） 说明整个电路工作原理。**

答：ALU 的工作方式可通过设置两个74181芯片的控制信号（S0、S1、S2、S3、M、CN）来实现, 其74LS181逻辑功能表由表1-1给出，运算器 ALU 的输出经过三态门（两片74LS244或一片74LS245）和数据总线 BUS 相连。当二进制控制开关CBA=010状态时，通过138译码选通输出ALU\_BUS运算器运行结果。运算器 ALU 的两个数据输入端分别由两个数据暂存器（74LS273）DR1、DR2 锁存， 74LS181 将 DR1、DR2 内的数据作为上述表1.1中参与运算的数 A 和 B。由于 DR1、DR2 已经把数据锁存，只要74LS181的控制信号不变，那么74LS181 的输出数据也不会发生改变。数据缓冲寄存器 DR1、DR2 的输入端D0~D7连至8位数据总线 BUS，在 DR1\_CLK和DR2\_CLK 端出现上升沿跳变的时候，总线BUS的数据分别打入DR1、DR2锁存。

**（2） 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，起到隔绝电路的作用。

（**3） 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：工作寄存器，其输出直接接到ALU的输入端，LCK为打入脉冲。

**（4） 说明74LS374N的功能及其在电路中作用，及输入信号CLK和OC有何作用；**

答：74ls374n作为通用寄存器（转送，暂存，计算），另具有三态门的作用。CLK作为打入脉冲的作用，而OC高电平为高阻态，逻辑操作不受影响。

**（5） K8产生任意数据存入通用寄存器GR1。**

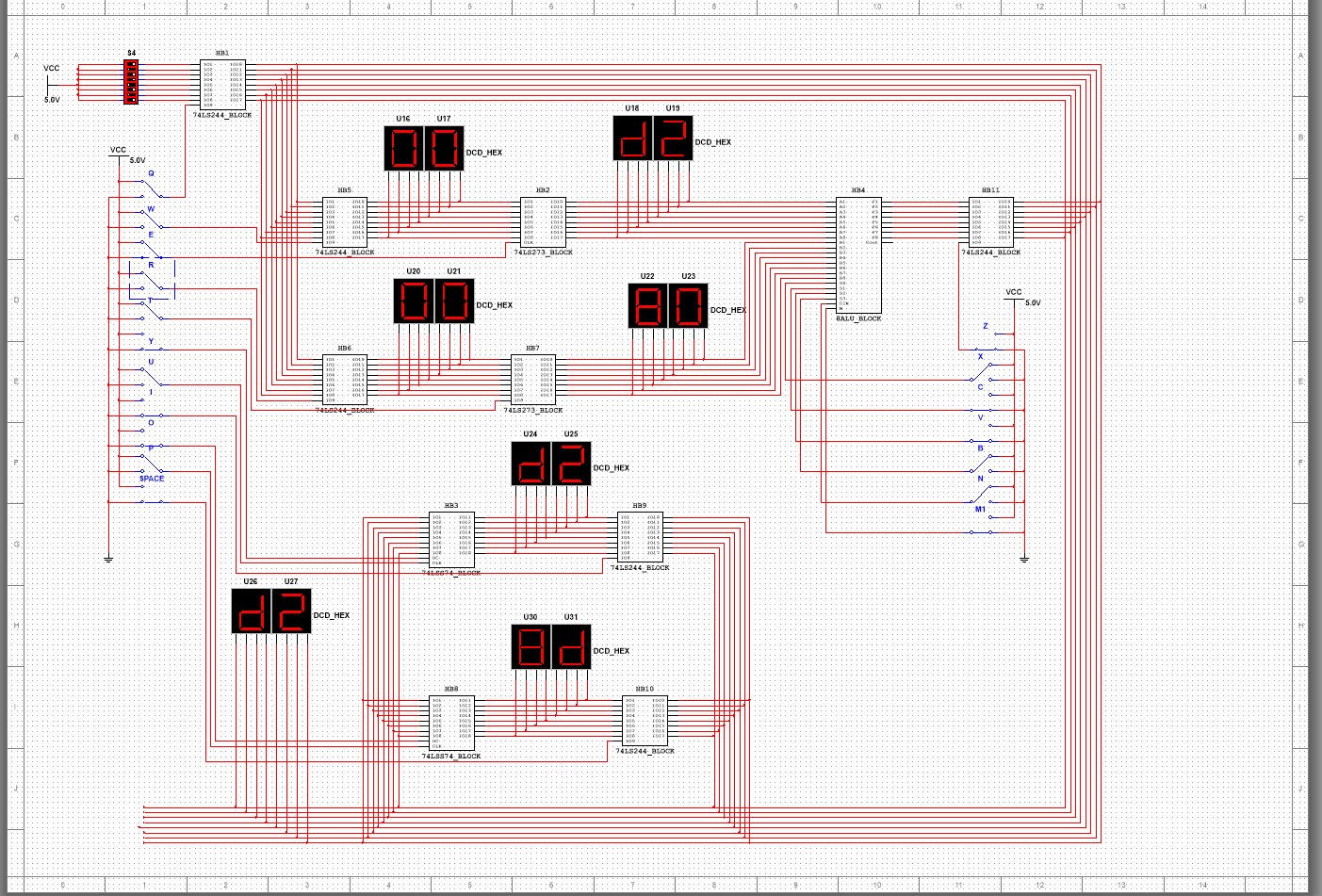
答：（1）K8确定任意数（测试时可指定一个2位16进制数）；（2）Q=0（低电平），数据送总线；（3）U↑（正跳变）且Y=0.该数存入GR1.

**（6） K8产生任意数据存入通用寄存器GR2。**

答：前2步同上，第三步为：P↑且O=0（左边为字母O，右边数字零）。该数送GR2.

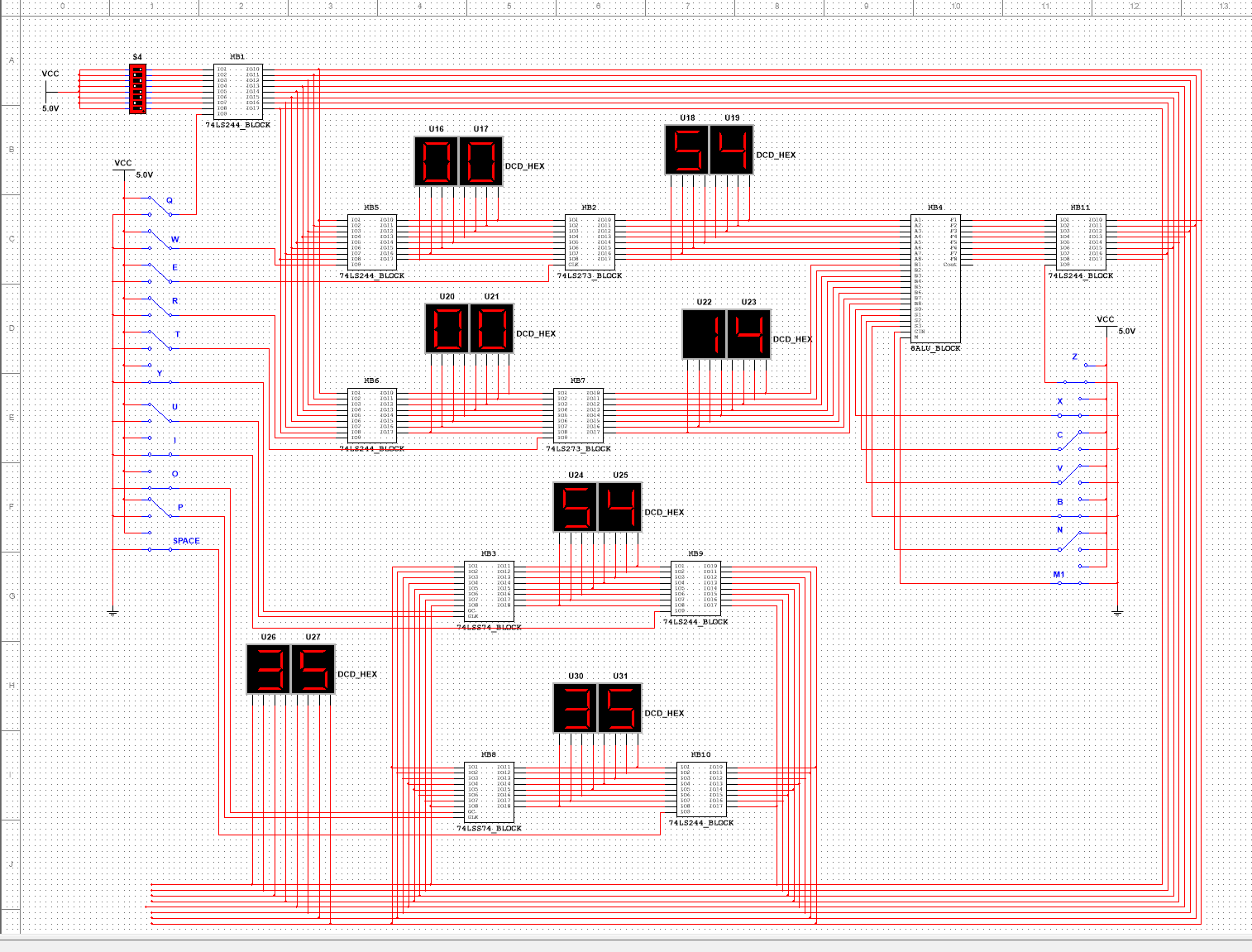
**（7） 完成GR1+GR2→GR1。**

答：GR1+GR2→GR1.（1）Q=1（屏蔽掉K8）、I=0、W=0、E↑，GR1→DR1；（2）Space=0、R=0、T↑，GR2→DR2；（3）加法：S3~S0=1001B（对应XCVB四键）、C=1（对应N键）、M=0（对应M1键），此时结果在ALU的输出端，令Z=0，结果送总线，U↑结果存入GR1.



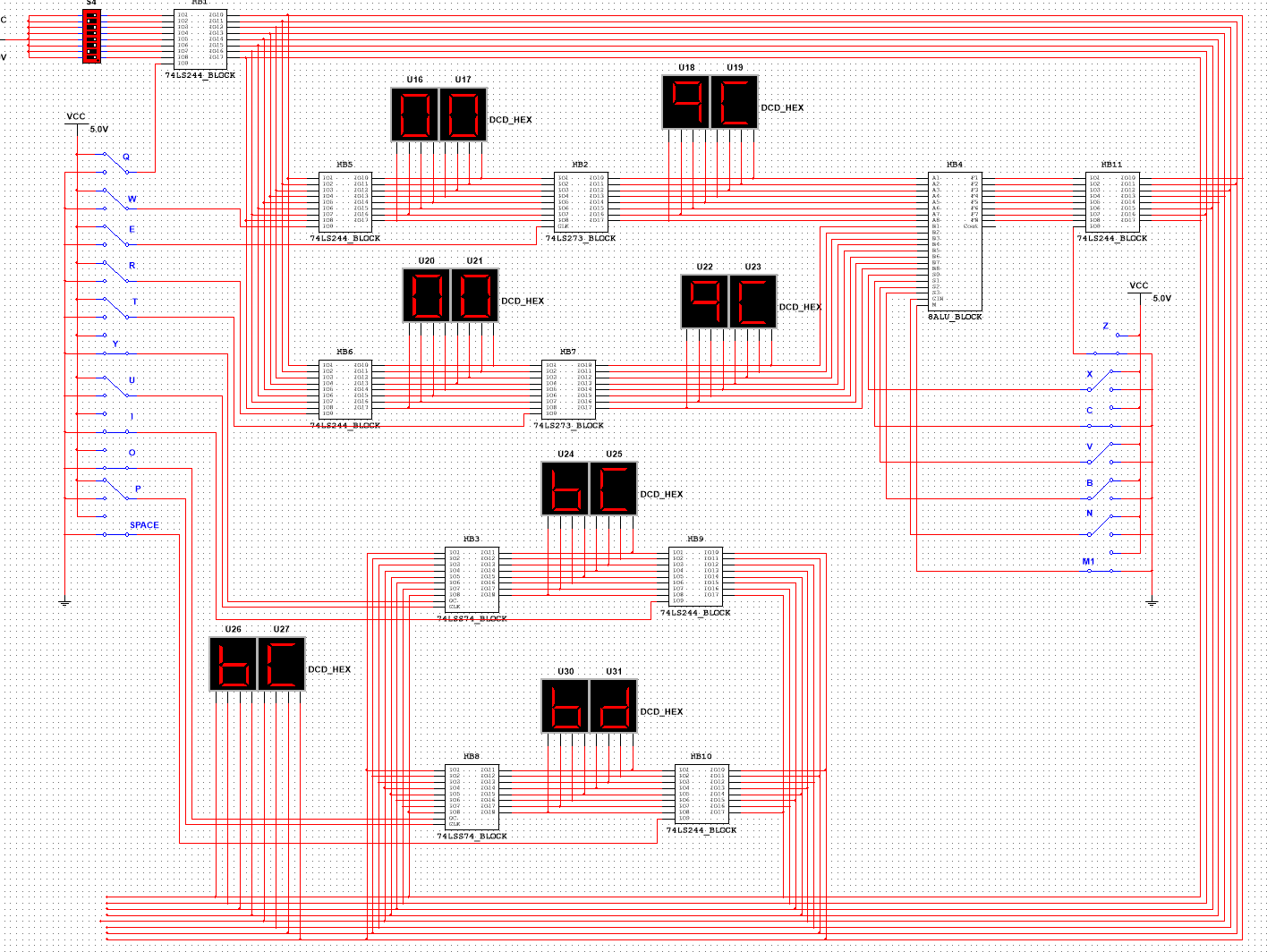
**（8） 完成GR1-GR2→GR2。**

答：GR1-GR2→GR2.前2步同上；S3~S0=0110B（对应XCVB四键）、C=1（对应N键）、M=0（对应M键），Z=0，P↑。



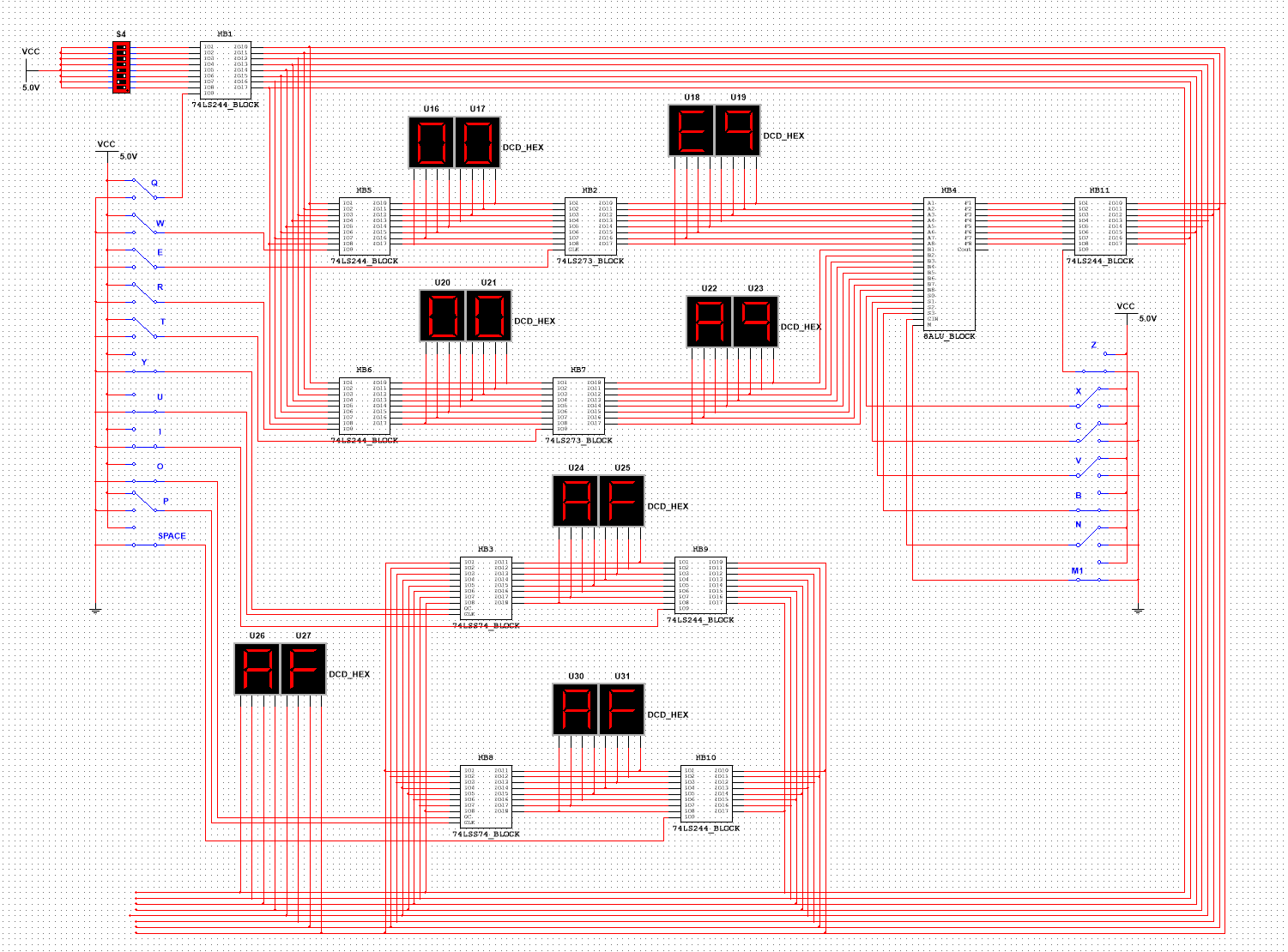
**（9） 完成GR1∧GR2→GR1。**

答：GR1ΛGR2→GR1. 前2步同前；S3~S0=1011B（对应XCVB四键）、M=1（对应M键），Z=0，U↑。



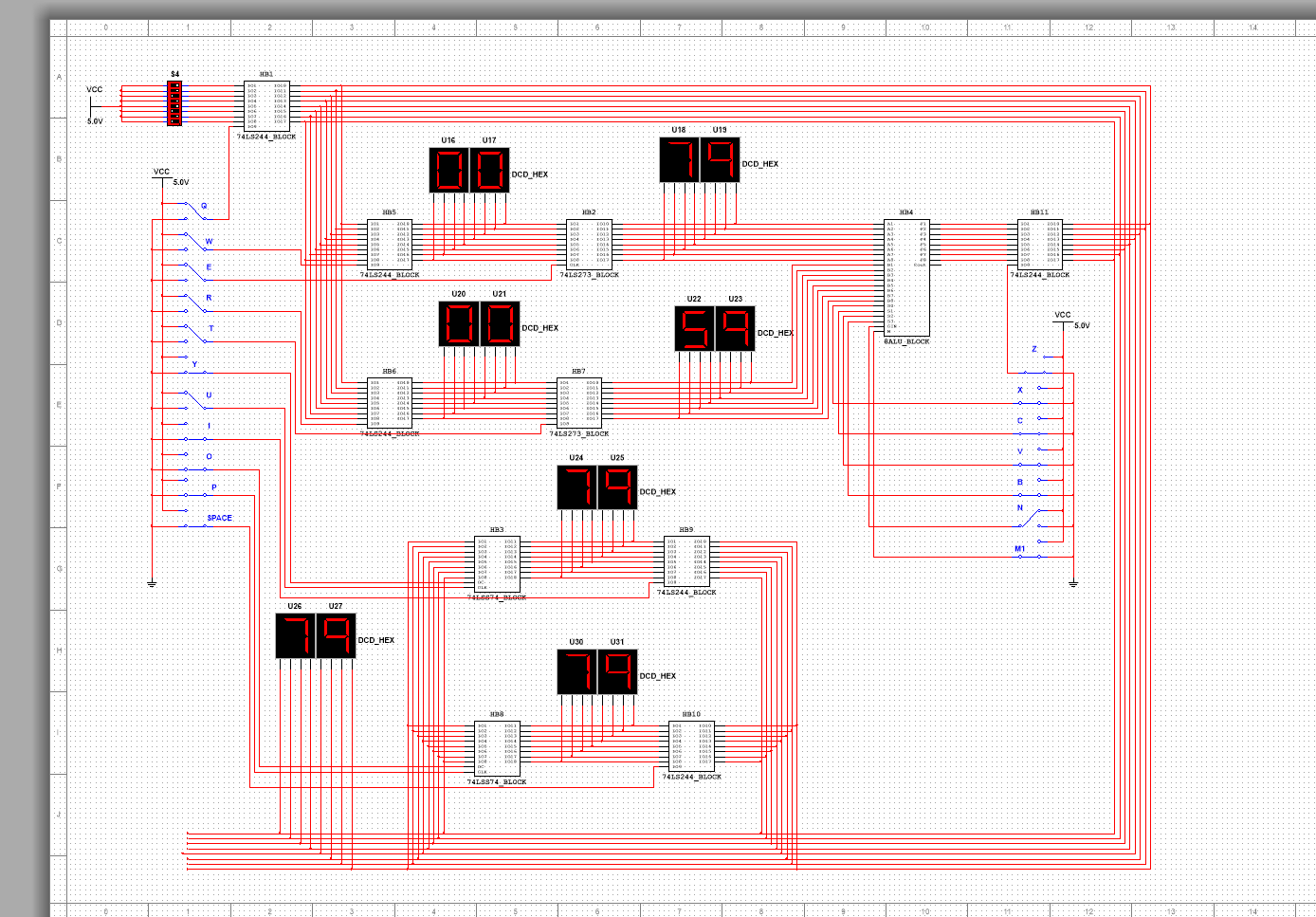
**（10） 完成GR1∨GR2→GR2。**

答：GR1VGR2→GR2. 前2步同前；S3~S0=1110B（对应XCVB四键）、M=1（对应M键），Z=0，P↑。



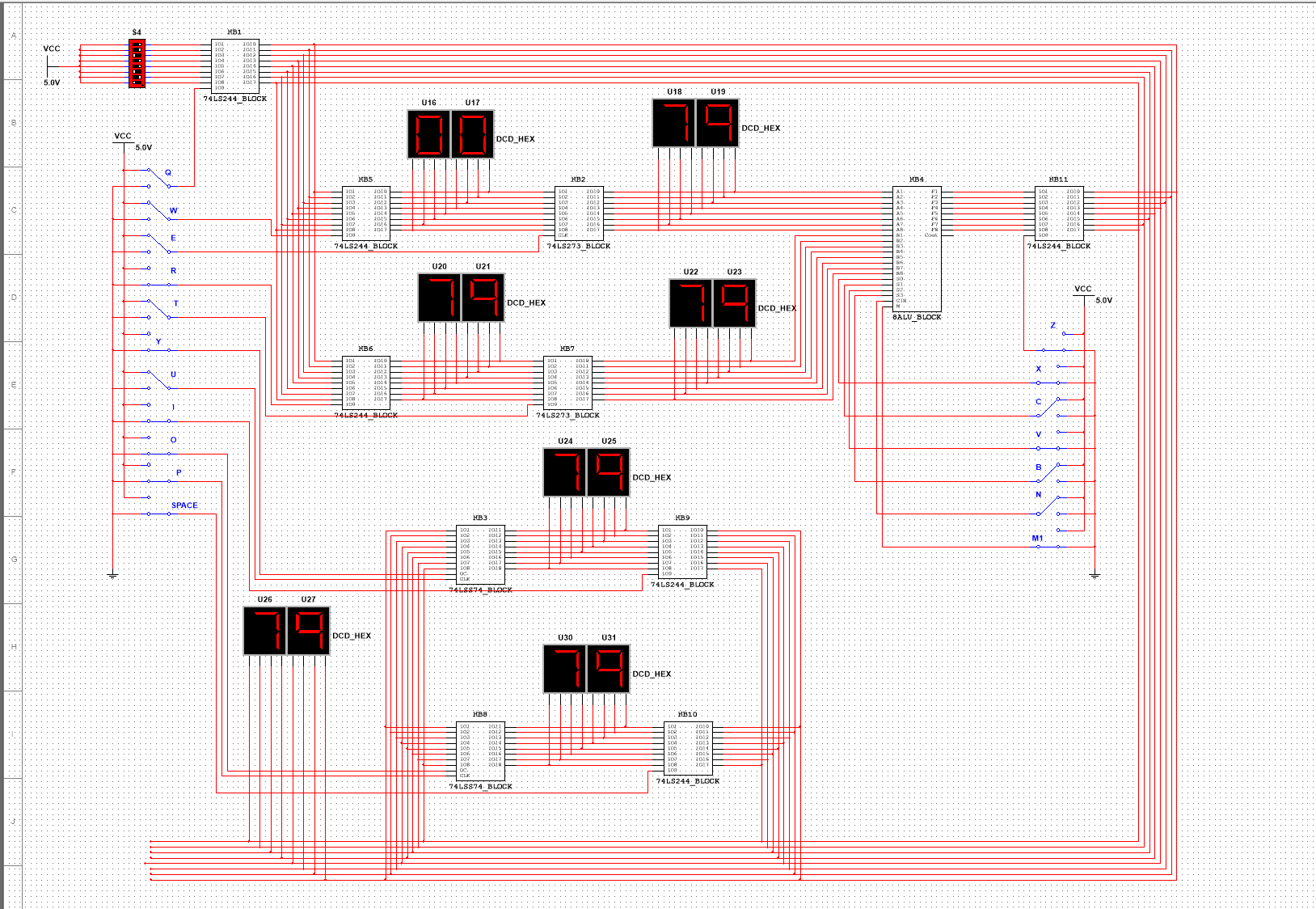
**（11） 完成GR1⊕GR2→GR1。**

答：NOT（GR1）→GR2.（1）I=0、W=0、E↑，GR1→DR1；（2）S3~S0=0000B（对应XCVB四键）、M=1（对应M键），Z=0，P↑。



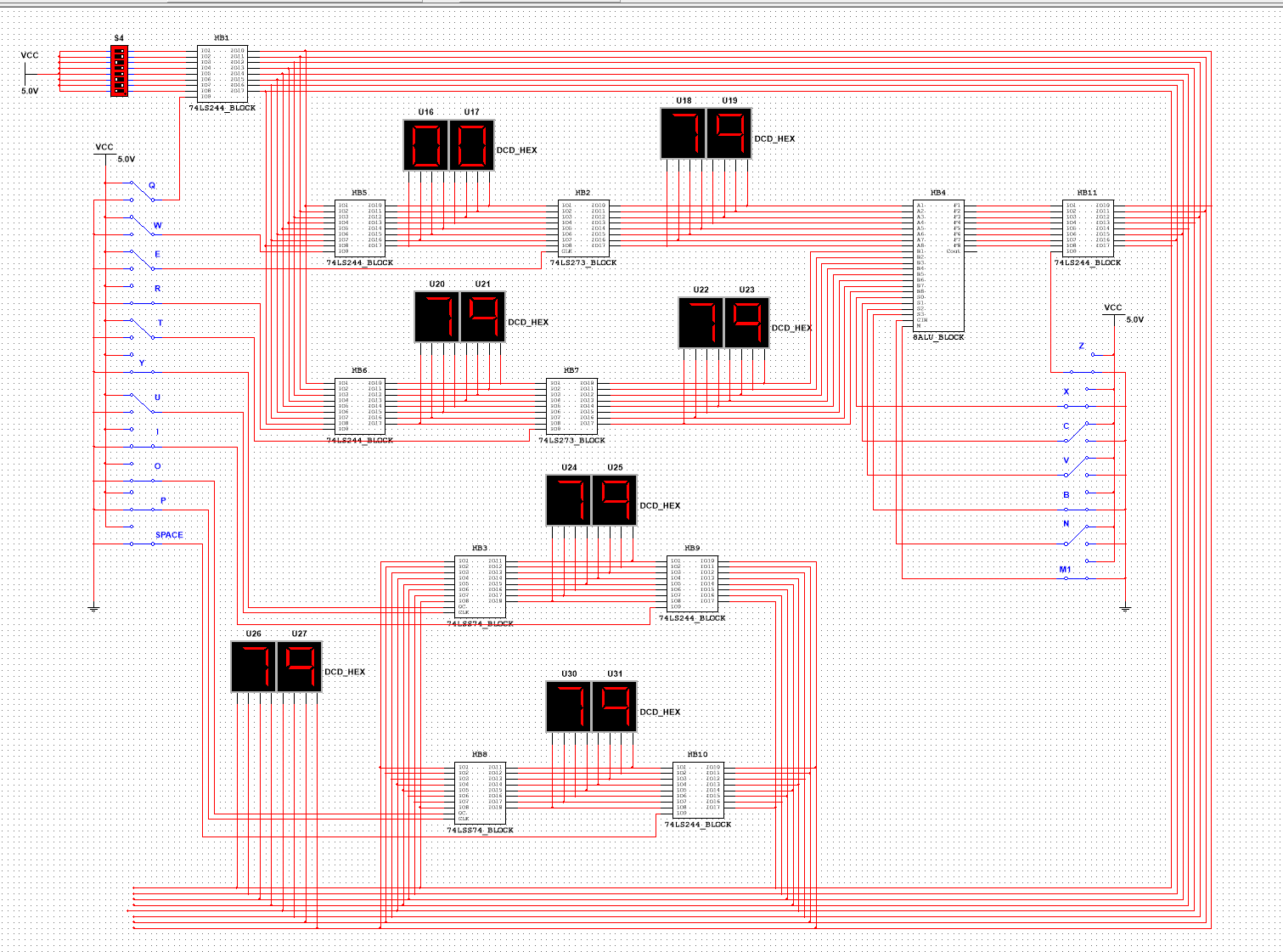
**（12） ~GR1→GR2。（“~”表示逻辑非运算）**

答：NOT（GR2）→GR1. （1）Space=0、R=0、T↑，GR2→DR2；（2）S3~S0=0101B（对应XCVB四键）、M=1（对应M键），Z=0，U↑。（实际上通过第11的功能实现也可以）



**（13） ~GR2→GR1。**

答：GR1⊕GR2→GR1.前2步同前；S3~S0=0110B（对应XCVB四键）、M=1（对应M键），Z=0，U↑。



1. **小结与心得体会**

本次实验消耗了较长时间，首先是对实验具体内容不够了解，其次是对实验器材及其相关原理认识不够深入，特别是对于ALU逻辑运算器的认识有待提高，不清楚某些步骤的含义，实际上每一次信号的输入都需要调整开关，以将数据保存在对应的寄存器中。

**实验三 字发生器及跑马灯**

* + 1. **实验目的**

通过设定字发生器的数据，使探针依次点亮，形成跑马灯。

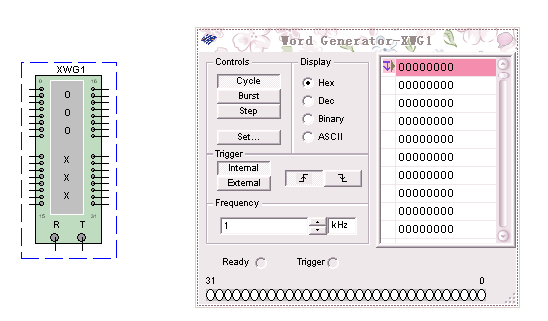
* + 1. **总体设计（含背景知识或基本原理与算法、或模块介绍、设计步骤等）**

1. **实验原理**

字信号发生器（Word Generator）可以采用多种方式产生32位同步逻辑信号，用于对数字电路进行测试，是一个通用的数字输入编辑器。

单击Simulate/Instruments/Word Generator,得到如图2-4（a）所示的字信号发生器的图标。在字信号发生器的左右两侧各有16个端口，分别为0～15和16～31的数字信号输出端，下面的R表示输出端，用以输出与字信号同步的时钟脉冲；T表示输入端，用来接外部触发信号。

双击图2-4（a）中的字信号发生器图标，便可以得到图2-4（b）所示的字信号发生器内部参数设置控制面板。该控制面板大致分为5个部分。



（a） （b）

图2-4字信号发生器内部参数设置控制面板

1. Control区：输出字符控制，用来设置字信号发生器的最右侧的字符编辑显示区字符信号的输出方式，有下列3种模式。

* Cycle 区：在已经设置好的初始值和终止值之间循环输出字符。
* Burst:每单击一次，字信号发生器将从初始值开始到终止值之间的逻辑字符输出一次，即单页模式。
* Step：每单击一次，输出一条字信号。即单步模式。

单击Set按钮，弹出如图2-5所示的对话框。该对话框主要用来设置字符信号的变化规律。其中各参数含义如下所述。

No Change：保持原有的设置。

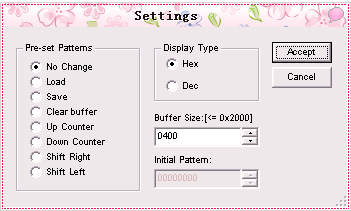


图2-5 字符信号的变化规律

Load：装载以前的字符信号的变化规律的文件。

Save：保存当前的字符信号的变化规律的文件。

Clear buffer:将字信号发生器的最右侧的字符编辑显示区的字信号清零。

Up Count：字符编辑显示区的字信号以加1的形式计数。

Down Count：字符编辑显示区的字信号以减1的形式计数。

Shift Right：字符编辑显示区的字信号右移。

Shift Left：字符编辑显示区的字信号左移。

Display Type选项区：用来设置字符编辑显示区的字信号的显示格式：Hex（十六进制），Dec（十进制）。

Buffer Size：字符编辑显示区的缓冲区的长度。

Initial Patterns：采用某种编码的初始值。

（2）Display区：用于设置字信号发生器的最右侧的字符编辑显示区的字符显示格式，有Hex、Dec、Binary、ASCII等几种计数格式。

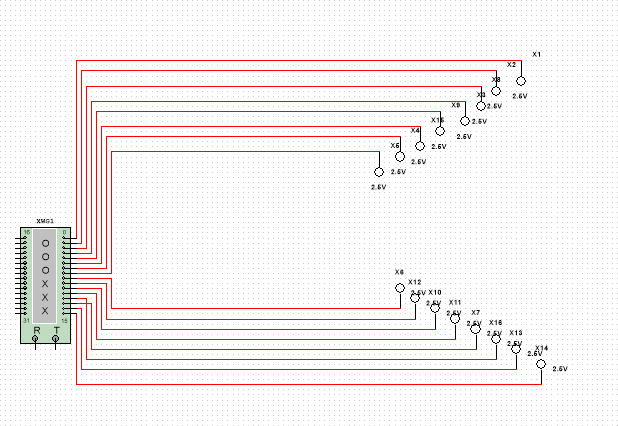
（3）Trigger区：用于设置触发方式。

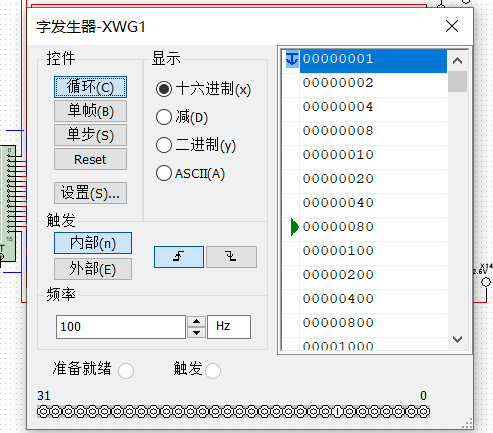
* Internal：内部触发方式，字符信号的输出由Control区的3种输出方式中的某一种来控制。
* External：外部触发方式，此时，需要接入外部触发信号。右侧的两个按钮用于外部触发脉冲的上升或下降沿的选择。

（4）Frequency区：用于设置字符信号飞输出时钟频率。

(5)字符编辑显示区：字信号发生器的最右侧的空白显示区，用来显示字符。

1. **设计步骤**
2. 了解字信号发生器的基本功能与工作原理；
3. 绘制仿真电路；
4. 编入字符信号，运行验证。
   * 1. **详细设计（含主要的数据结构、程序流程图、关键代码等）**
5. **电路设计图**

****

****

* + 1. **实验结果与分析**

16个探针被依次点亮，，并且循环点亮。

* + 1. **小结与心得体会**

本实验难度不大，依次连线同时设定好参数即可。

**实验四 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（含背景知识或基本原理与算法、或模块介绍、设  
   计步骤等）**
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图2-3中开关，模拟微指令自动执行实现一条机器语言指令。

1. **实验内容**

在Multisim画出电路图并仿真，完成如下操作。

（1）产生任意数据存入通用寄存器GR1。

（2）产生任意数据存入通用寄存器GR2。

（3）完成GR1+GR2→GR1。

（4）完成GR1-GR2→GR2。

（5）完成GR1∧GR2→GR1。

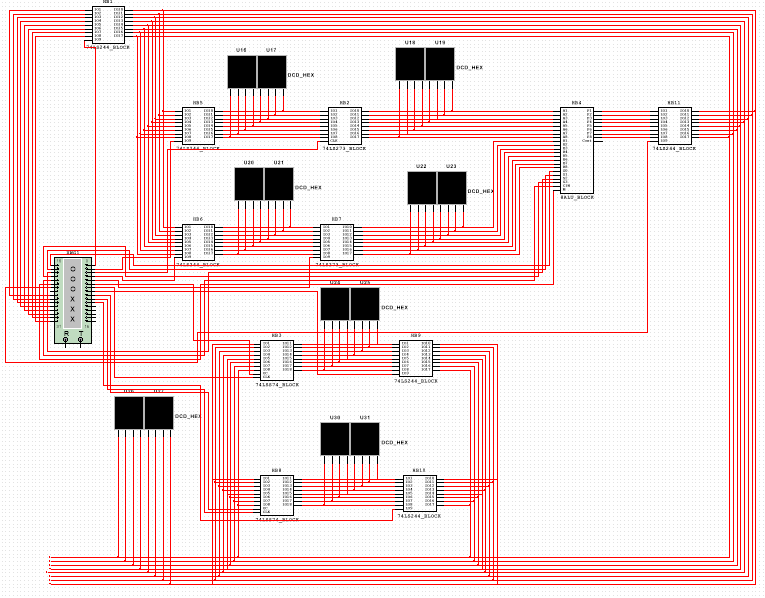
（6）完成GR1∨GR2→GR2。

（7）完成GR1⊕GR2→GR1。

（8）~GR1→GR2。（“~”表示逻辑非运算）

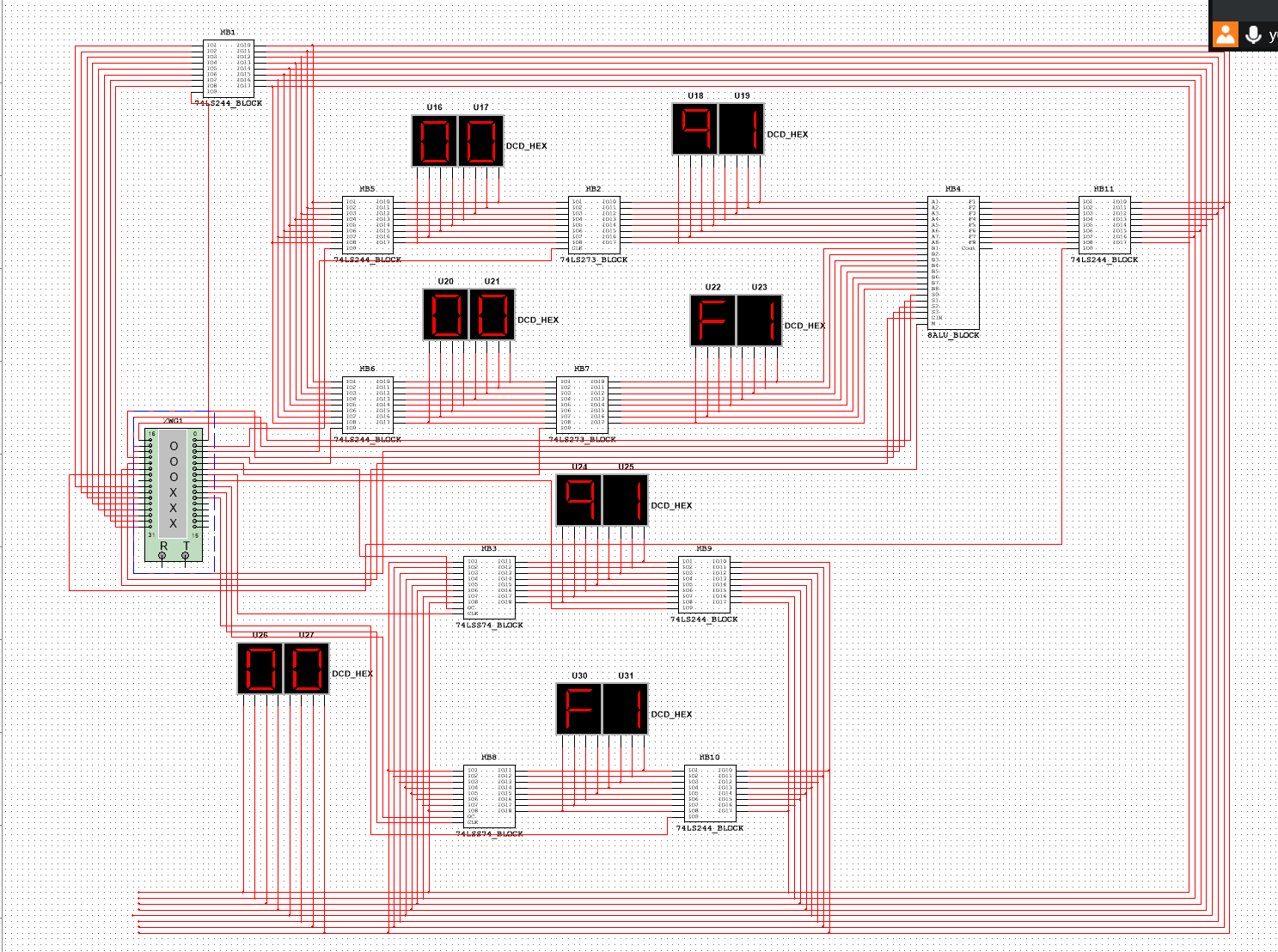
（9）~GR2→GR1。

1. **设计步骤**
2. 在实验2的基础上对电路图进行修改，用字信号发生器替代实验2电路图中的开关；
3. 参照实验2的每一个步骤，将操作编写为符号信号；
4. 运行验证。
5. **详细设计（含主要的数据结构、程序流程图、关键代码等）**
6. **电路设计图**

****

1. **实验结果与分析**

（1）产生任意数据存入通用寄存器GR1。  
（2）产生任意数据存入通用寄存器GR2完成GR1+GR2→GR1



字发生器的内指令如下：

10010001010000000000010010001011

10010001010000000000010010000000

10010001010000000000010011000100

10010001010000000000010010000000

11110001010000000000010010000000

11110001010000000000011010010000

11110001010000000000010010000000

11110001010000000000010010001011

11110001010110010000010010001011

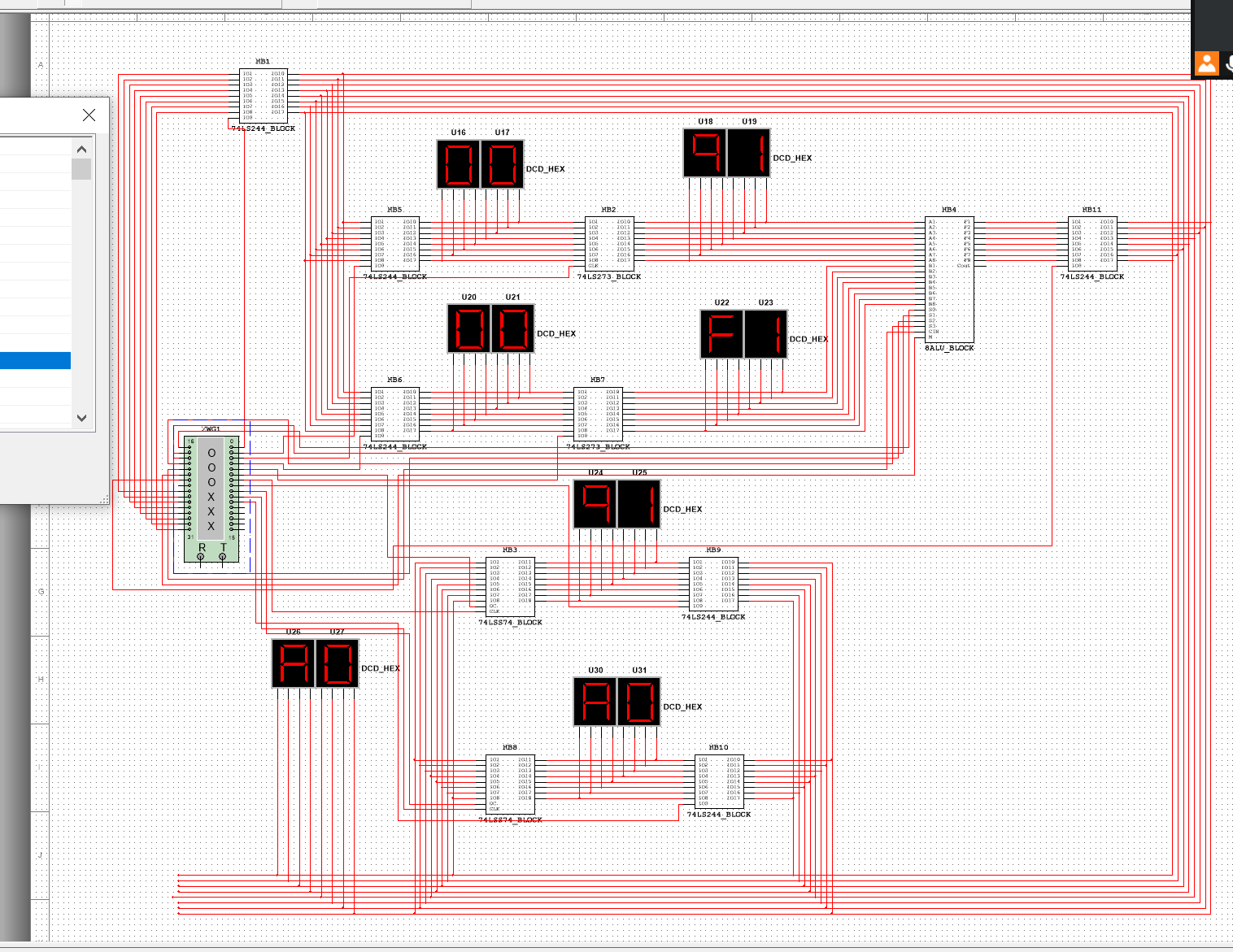
11110001000110010000010010001011

11110001000110010000010011001011

11110001000110010000010010001011

00000000010110010000010010001011

（3）完成GR1-GR2→GR2。



字发生器内部指令如下：

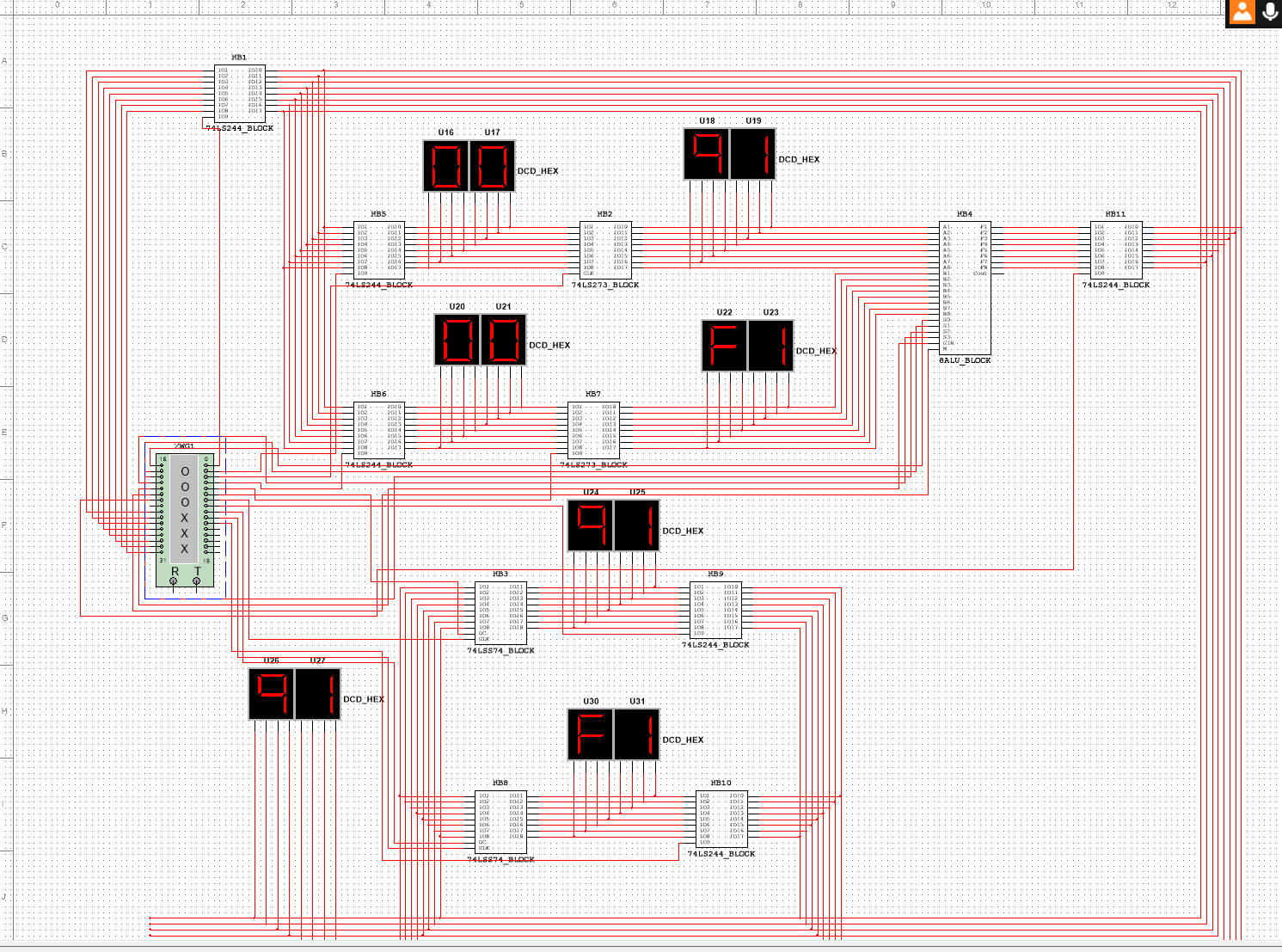
11110001010001100000010010001011

11110001000001100000010010001011

11110001000001100000011010001011

11110001000001100000010010001011

（4）完成GR1∧GR2→GR1。



字发生器内部指令如下：

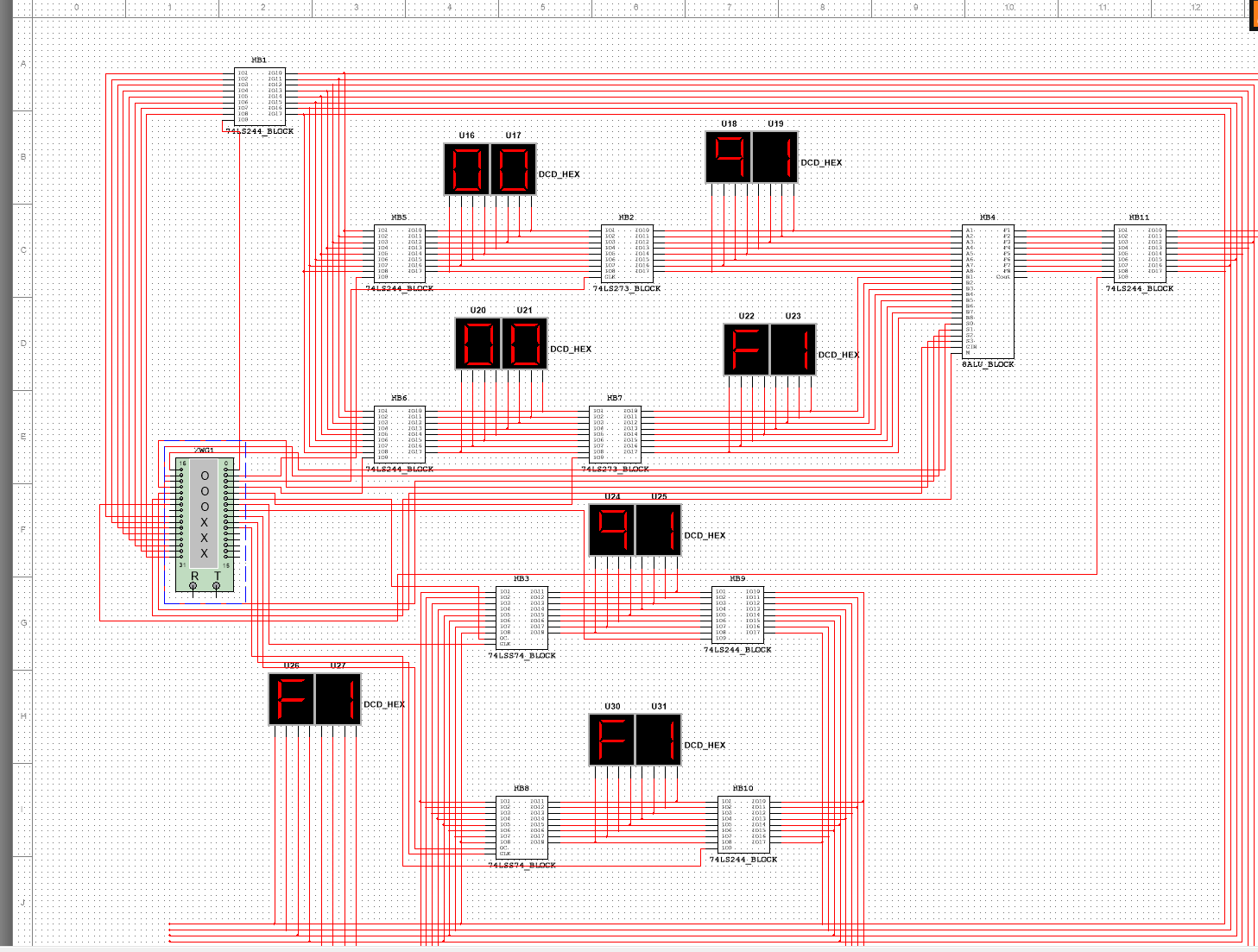
11110001010010110000010010001011

11110001000010110000010010001011

11110001000010110000010011001011

11110001000010110000010010001011

（5）完成GR1∨GR2→GR2。



字发生器内部指令如下：

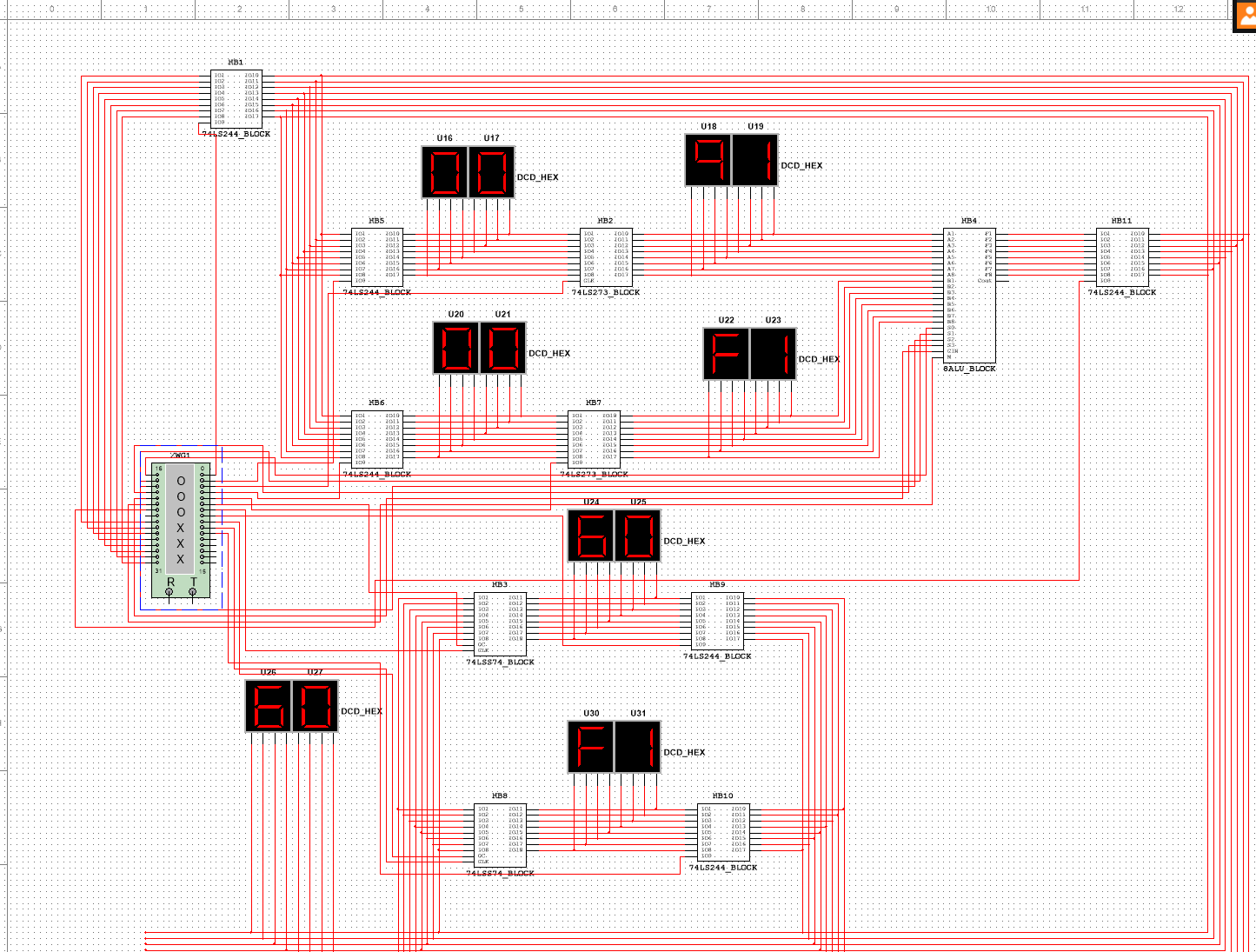
11110001010100010000010010001011

11110001000100010000010010001011

11110001000100010000011010001011

11110001000100010000010010001011

（6）完成GR1⊕GR2→GR1。



字发生器内部指令如下：

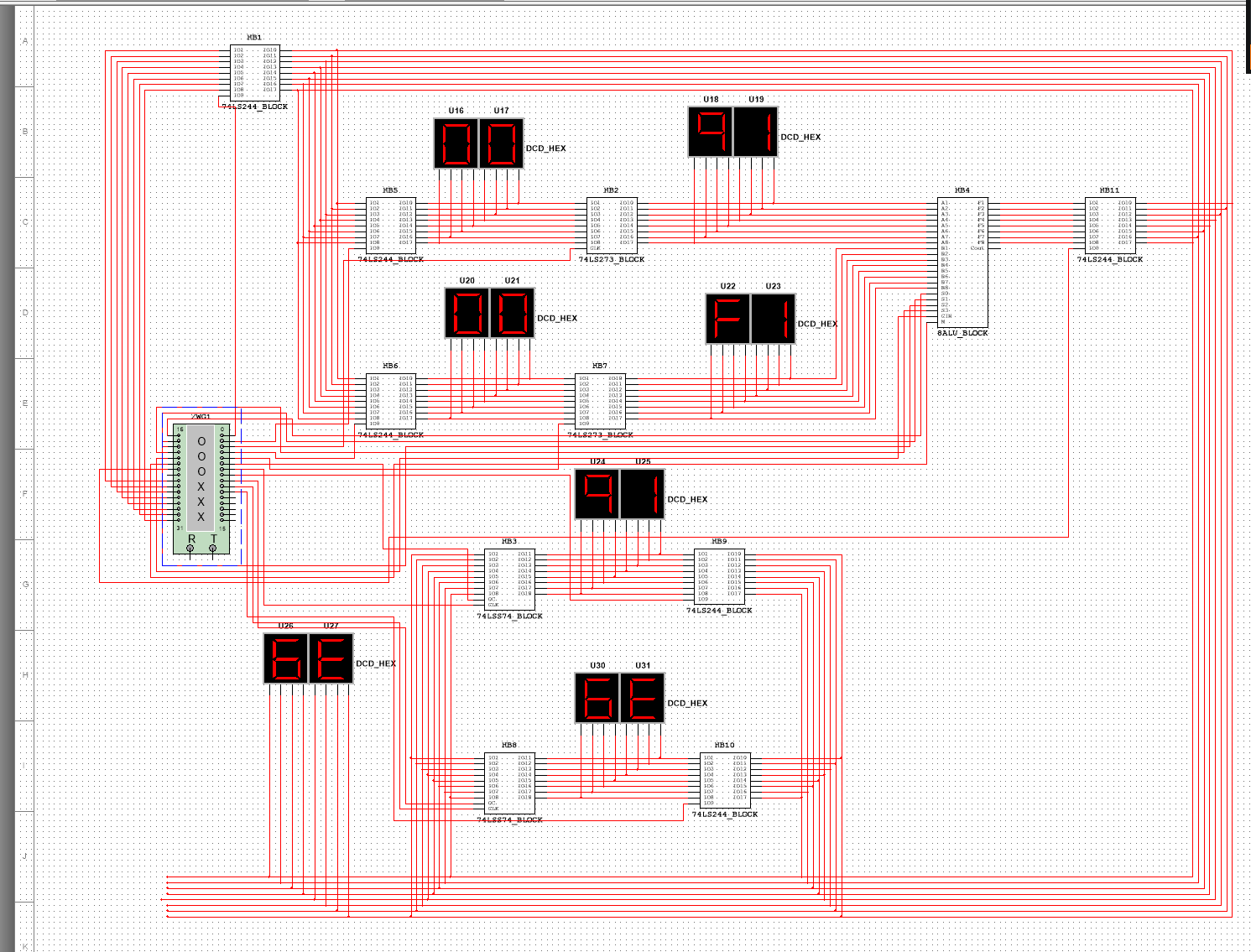
11110001011001100000010010001011

11110001001001100000010010001011

11110001001001100000010011001011

11110001001001100000010010001011

（7）~GR1→GR2。



字发生器内部指令如下：

11110001011000000000010010001011

11110001001000000000010010001011

11110001001000000000011010001011

11110001001000000000010010001011

（8）~GR2→GR1。



字发生器内部指令如下：

11110001011001010000010010001011

11110001001001010000010010001011

11110001001001010000010011001011

11110001001001010000010010001011

1. **小结与心得体会**

实际上，本实验在了解实验2和3后并不难，只需要将相关模块替换，同时进行组合即可，本次实验中，用字信号器代替了实验2中的开关，只需要输入相关的操作，便可以代替实验2中的手动调试。

仔细对比本次实验和实验2，发现同实验2一样，前面几步的操作都是相同的，所以前面的代码无需修改，只用改动最后四行，此外，可以通过字信号器的load功能将操作代码保存为db文件，便于后续调试。