***2018***



**计算机组成原理 ·实验报告·**

|  |  |
| --- | --- |
| 实验内容： | **计算机组成原理实验** |
| 专 业： | 软件工程 |
| 班 级： | 1603 |
| 学 号： | U201617047 |
| 姓 名： | 刘俊傲 |
| 电 话： | 15827563592 |
| 邮 件： | 1713507920@qq.com |
| 报告日期： | 2018-6-29 |
| 指导教师： | 肖亮 黄浩 |

目录

[1 实验一 数据表示实验 3](#_Toc7508)

[1.1 实验目的与内容 3](#_Toc2987)

[1.1.1 实验目的 3](#_Toc18815)

[1.1.2 实验内容 3](#_Toc15851)

[1.2 实验原理与方案 7](#_Toc15770)

[1.2.1 实验原理 7](#_Toc13222)

[1.2.2 实验方案 7](#_Toc3686)

[1.3 实验测试与结果 10](#_Toc13036)

[1.3.1 汉字编码测试 10](#_Toc9406)

[1.3.2 偶校验测试 11](#_Toc30728)

[1.3.3 海明校验测试 11](#_Toc28438)

[1.4 总结与心得 12](#_Toc13717)

[1.4.1 实验总结 12](#_Toc9164)

[1.4.2 实验心得 12](#_Toc1533)

[2 实验二 运算器组成实验 13](#_Toc17180)

[2.1 实验目的与内容 13](#_Toc25113)

[2.1.1 实验目的 13](#_Toc15810)

[2.1.2 实验内容 13](#_Toc3546)

[2.2 实验原理与方案 16](#_Toc3462)

[2.2.1 实验原理 16](#_Toc13081)

[2.2.2 实验方案 16](#_Toc14149)

[2.3 实验测试与结果 20](#_Toc25602)

[2.3.1 ALU综合测试 20](#_Toc10811)

[2.4 总结与心得 20](#_Toc18867)

[2.4.1 实验总结 20](#_Toc20115)

[2.4.2 实验心得 21](#_Toc10506)

[3 实验三 补码乘法器实验 22](#_Toc25998)

[3.1 实验目的与实验内容 22](#_Toc30395)

[3.1.1 实验目的 22](#_Toc12217)

[3.1.2 实验内容 22](#_Toc15643)

[3.2 实验原理及方案 23](#_Toc12818)

[3.2.1 实验原理 23](#_Toc26952)

[3.2.2 实验方案 23](#_Toc30982)

[3.3 总结与心得 24](#_Toc664)

[3.3.1 实验总结 24](#_Toc28847)

[3.3.2 实验心得 24](#_Toc9843)

[4 实验四 存储系统综合实验 25](#_Toc25535)

[4.1 实验目的与内容 25](#_Toc13313)

[4.1.1 实验目的 25](#_Toc32501)

[4.1.2 实验内容 25](#_Toc11867)

[4.2 实验原理与方案 28](#_Toc25876)

[4.2.1 实验原理 28](#_Toc20760)

[4.2.2 实验方案 28](#_Toc32689)

[4.3 实验测试与结果 30](#_Toc29293)

[4.4 总结与心得 31](#_Toc23379)

[4.4.1 实验总结 31](#_Toc26434)

[4.4.2 实验心得 31](#_Toc4542)

[5 实验总结与评价 32](#_Toc226)

# 1 实验一 数据表示实验

## 实验目的与内容

### 1.1.1 实验目的

1）掌握GB2312与区位码转换方法;

2）了解字模码显示的原理;

3）掌握奇偶校验校验位的生成方法；

4）掌握奇偶校验检错基本原理；

5）掌握海明码编解码电路基本原理；

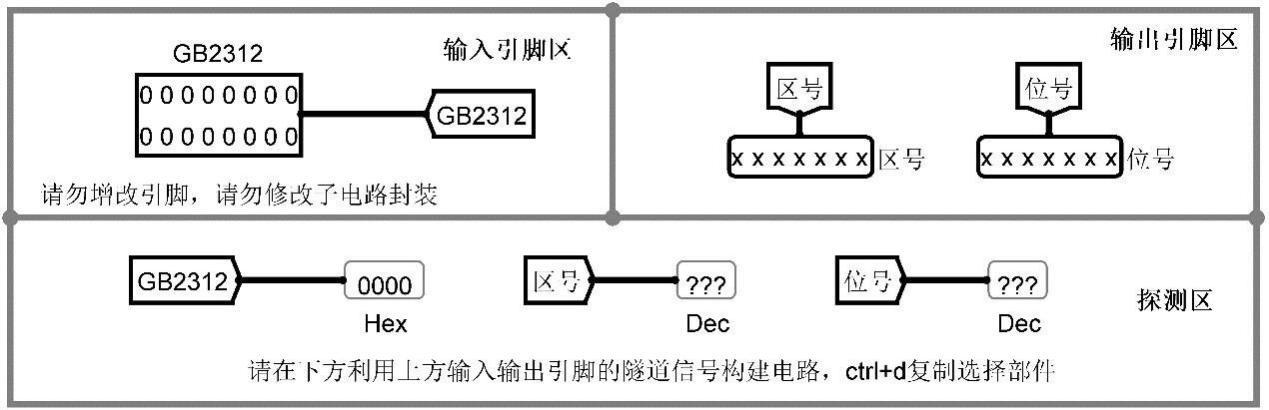
6）熟悉流水数据传输机制，流水暂停原理；

### 1.1.2 实验内容

**1)** 汉字编码

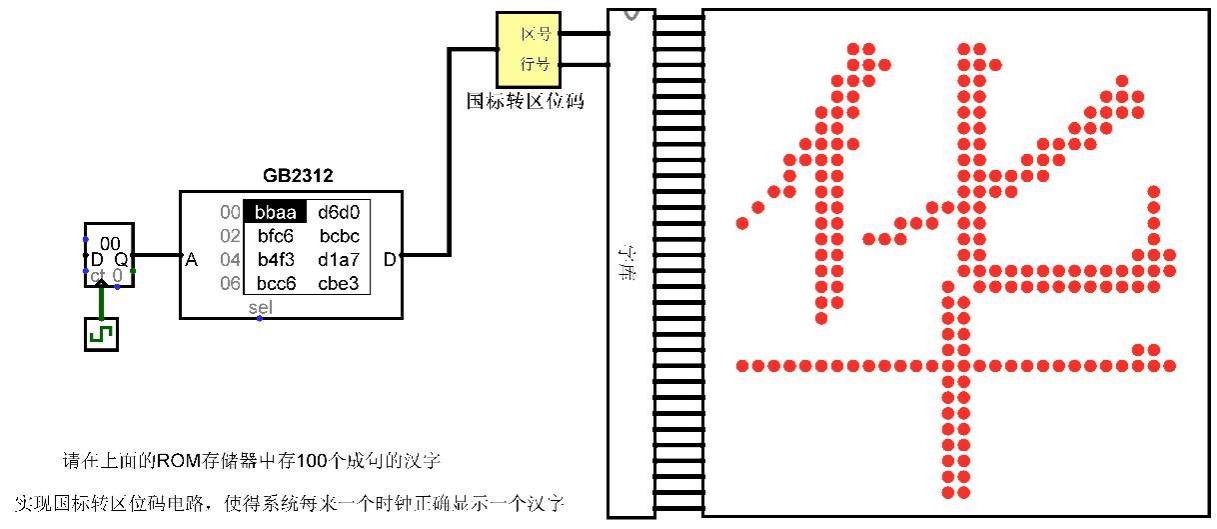
设计国标码转区位码电路

输入：GB2312 16位国标码；输出：区号，行号（区号行号均从1开始计数），下图为电路引脚定义，请在电路中复制隧道连接信号，注意不要增改引脚，不要修改子电路封装，以免影响该子电路在其他电路模块中的调用。

 图 1 汉字国标码转区位码电路引脚定义

汉字GB2312编码实验 完成国标码到区位码的转换电路后，可以在汉字显示电路中进行测试，尝试在下面电路中的ROM存储器中存入25个成句的汉字（要求与原始数据不同），ROM存储器使用方法见Logisim参考手册。

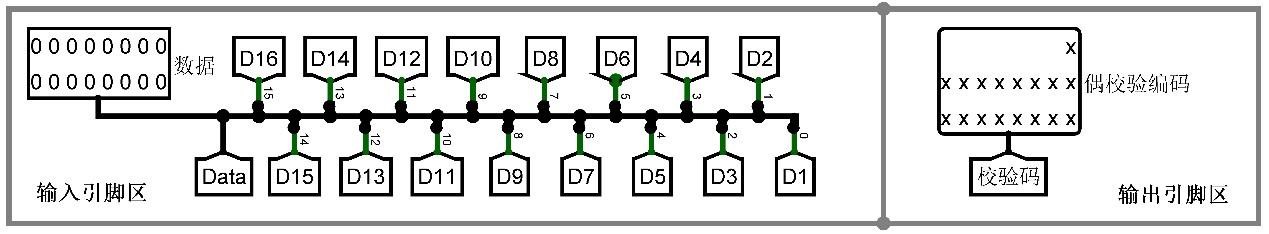
该电路启用时钟仿真后可以依次显示ROM中的事先预存的所有汉字（时钟仿真快捷键：Ctrl+k）。

 图 2 汉字字模码显示电路

**2)** 偶校验

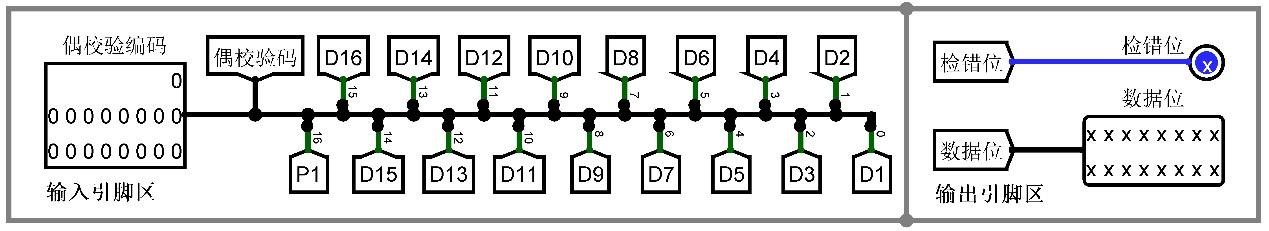
设计16位数据编码的偶校验编码电路

输入：16 位原始数据；输出：17 位校验码（16 位数据位+1 位校验位），输入输出引脚定义如下：

 图 3 偶校验编码电路引脚定义

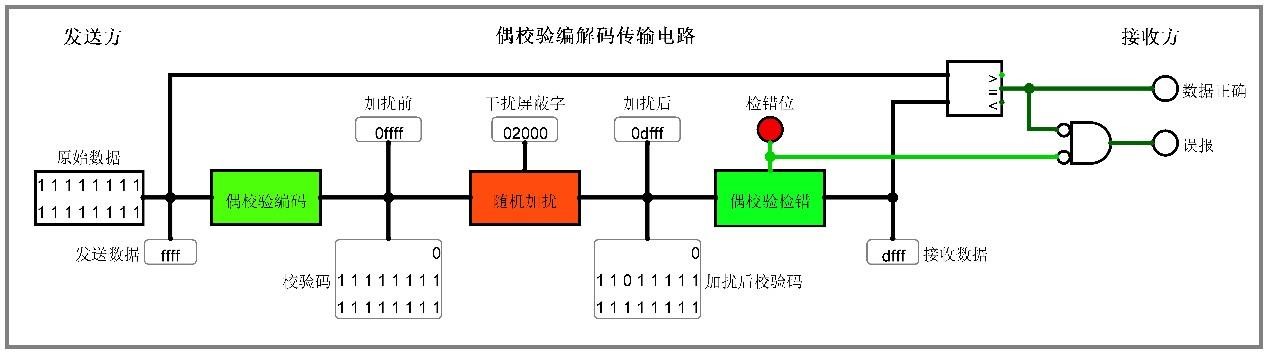
设计17位偶校验编码的检错电路

输入：17 位校验码；输出：16 位原始数据，1 位检错位；输入输出引脚定义如下：

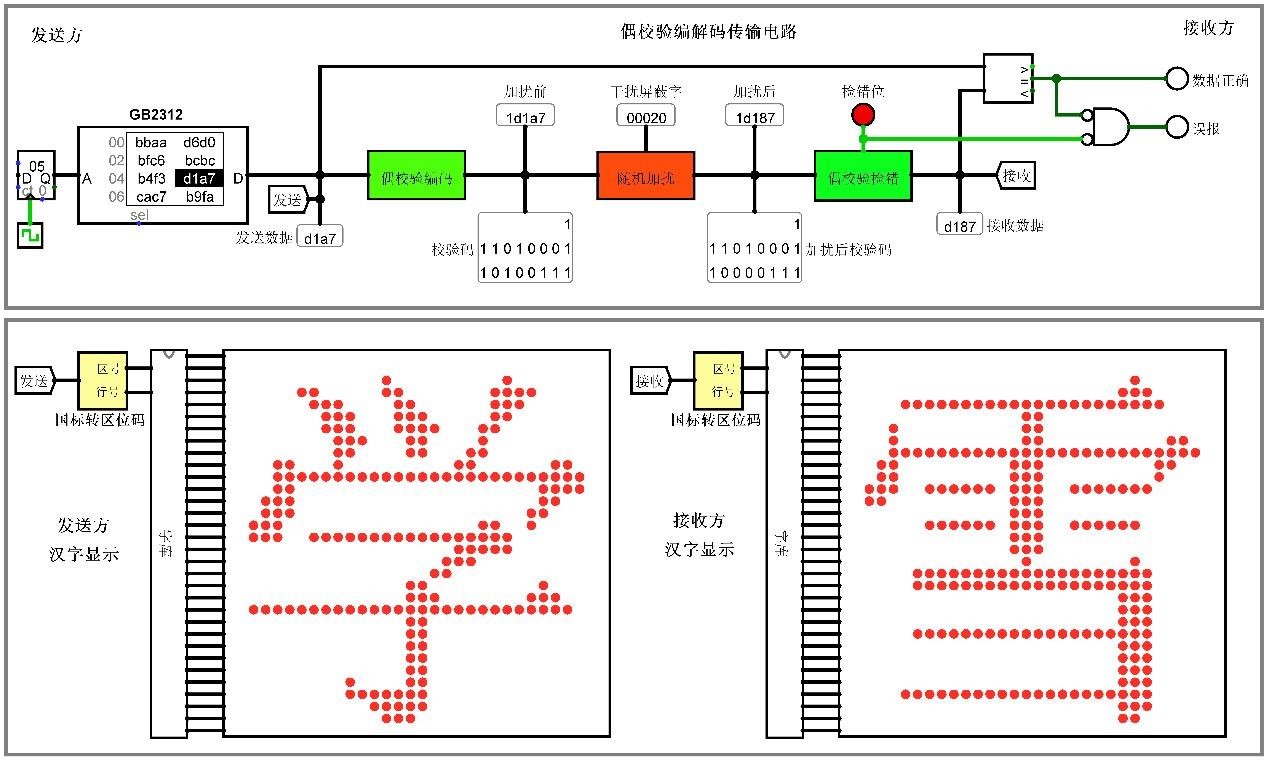
 图 4 偶校验解码电路引脚定义

偶校验传输测试

在偶校验传输测试 1 电路中测试偶校验编解码电路的正确性，并观察数据传输过程中何时会出现误报情况，分析奇偶校验传输的性能，如果已经实现汉字显示模块，可直接使用偶校验传输测试 2 电路。

 图 5 偶校验传输测试电路 1

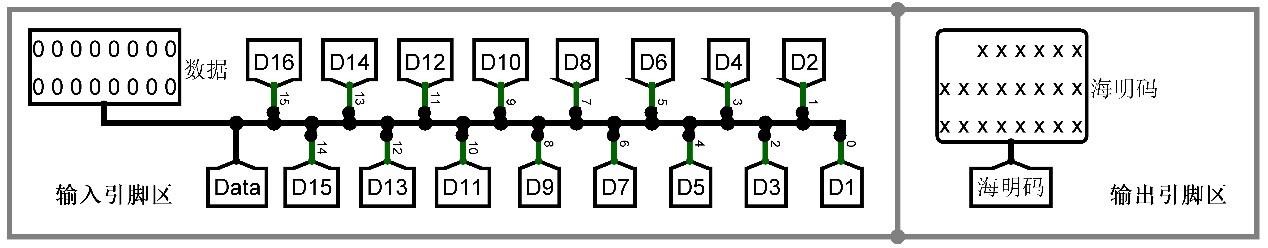
在偶校验传输测试 2 电路中测试偶校验编解码电路实现是否正确，测该电路引入了汉字显示模块，可以直接显示接收端和发送端的编码的汉字，通过汉字显示可以很直观观察传输是否发生错误，从而观察采用偶校验进行数据传输时传输的可靠性，用户可以使用 ctrl+t 快捷键开启时钟自动仿真测试，具体测试电路如下图所示：

 图 6 偶校验传输测试电路 2

**3)** 海明校验

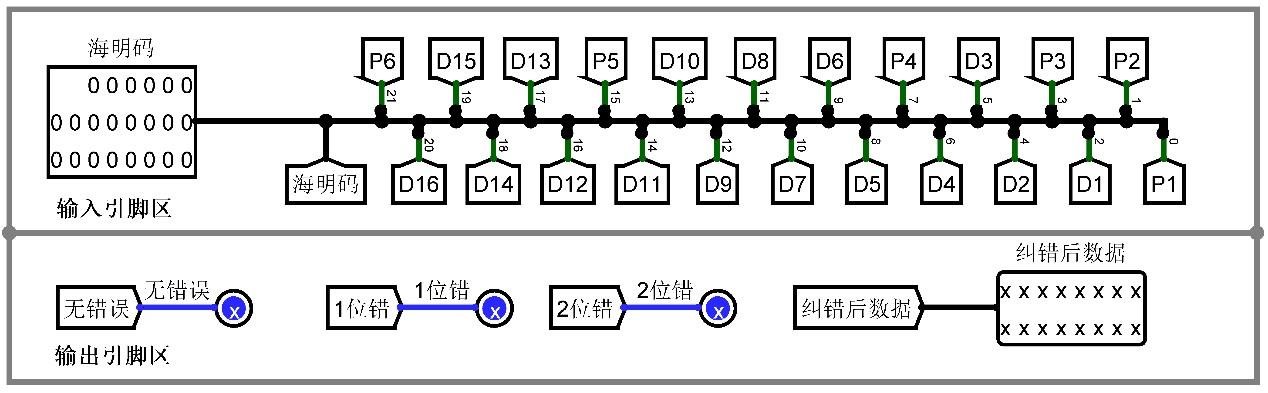
设计16位数据编码的海明校验编码电路

输入：16 位原始数据；输出：22 位校验码（16 位数据位+5 位海明校验位+1位奇校验位），输入输出引脚定义如下：

 图 7 海明编码电路引脚定义

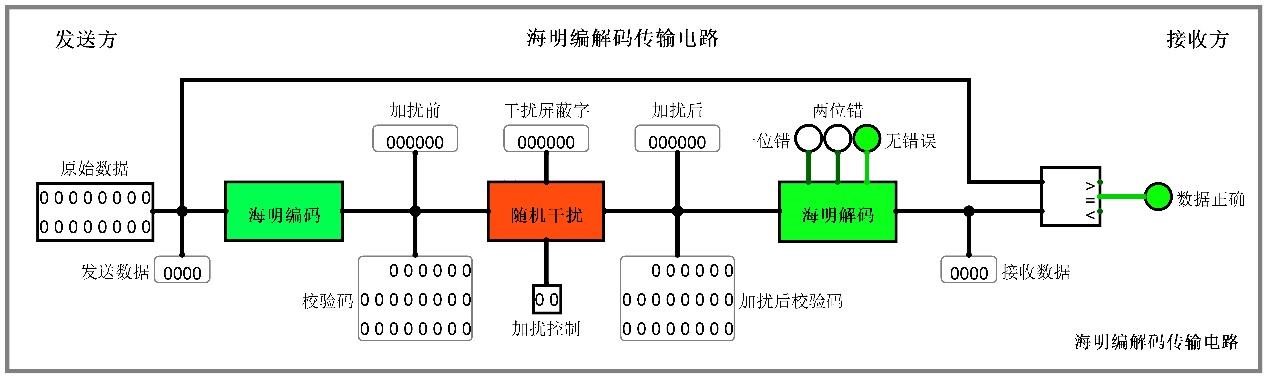
设计22位海明校验码的解码电路

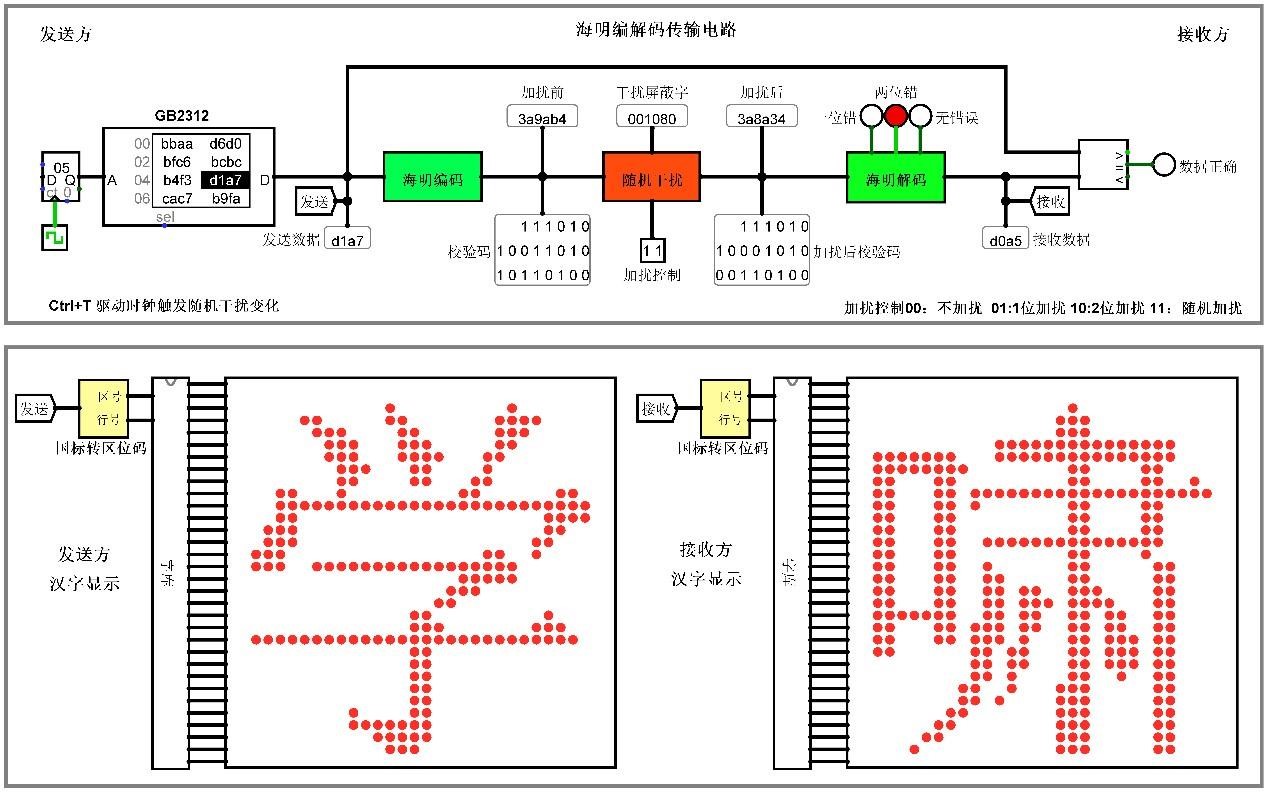
输入：22 位校验码；输出：16 位原始数据，1 位检错位；2 位检错位；无错误状态位；输入输出引脚定义如下：

 图 8 海明解码电路引脚定义

海明校验传输测试

在海明校验传输测试 1 子电路中测试海明校验编解码电路的正确性，注意图中随机干扰电路只能产生最多 2 位错误，具体测试电路如下图所示：

图 9 海明编码传输电路 1



在海明校验传输测试 2 子电路中测试海明校验编解码电路的正确性，该电路引入了汉字显示模块，可以直接显示接收端和发送端的编码的汉字，通过汉字显示可以很直观的看出海明纠错的效果，用户可以使用 ctrl+t 快捷键开启时钟自动仿真测试。

## 实验原理与方案

### 实验原理

1. 国标码转区位码：

使用国标码减去2020h即可得到区位码。

1. 偶校验：

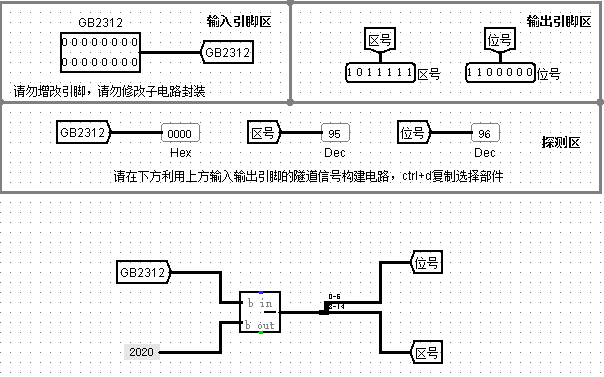
偶校验是一种校验代码传输正确性的方法。专门设置一个偶校验位，用它使这组代码中“1”的个数为偶数。当接收端收到这组代码时，校验“1”的个数是否为偶数，从而确定传输代码的正确性。

1. 海明校验：

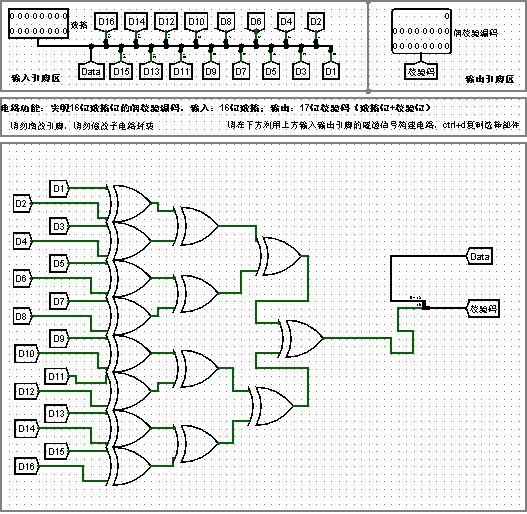
将有效信息按某种规律分成若干组，每组安排一个校验位，做奇偶测试，就能提供多位检错信息，以指出最大可能是哪位出错，从而将其纠正。实质上，海明校验是一种多重校验。本实验中G1-G5形成指错字，P6位总的奇偶校验，P6可指出一位错，若P6无错且指错字为0，则无错，若指错字不为零且P6有错，则为一位错，可以依据指错字进行纠错。若P6无错且指错字不为零，则为两位错，两位错无法纠正。

### 实验方案

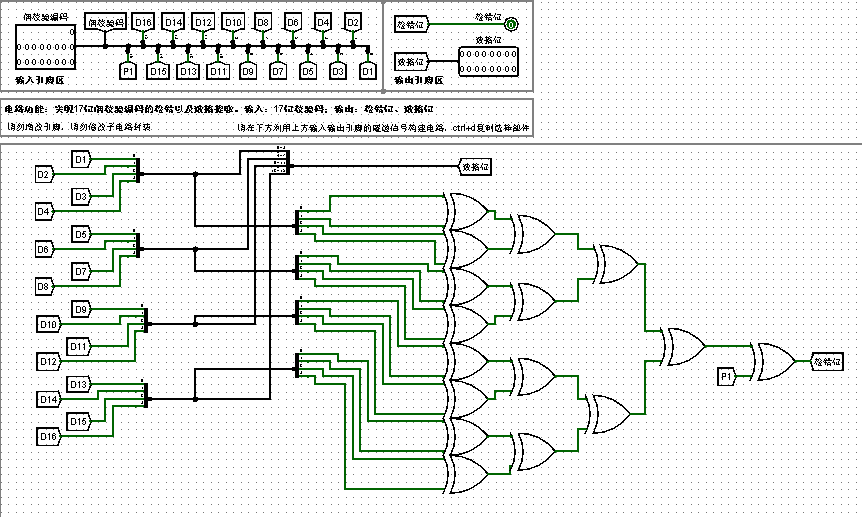
1. 国标码转区位码：



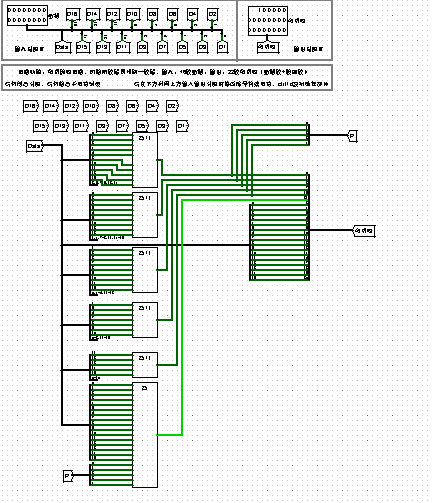
1. 偶校验：
2. 偶校验编码



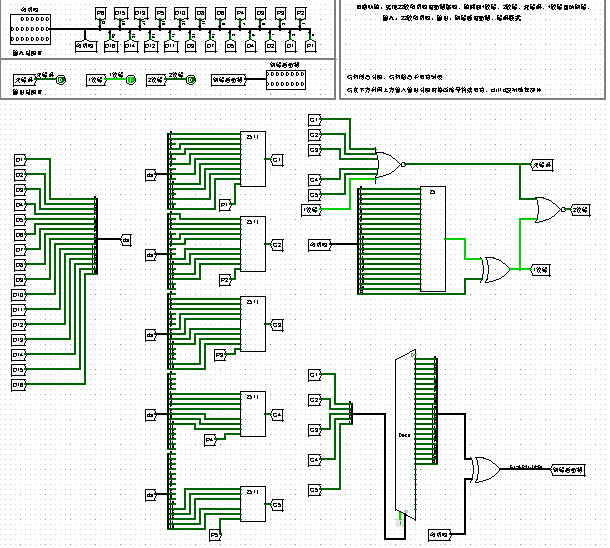
1. 偶校验检错



1. 海明校验：
2. 海明编码

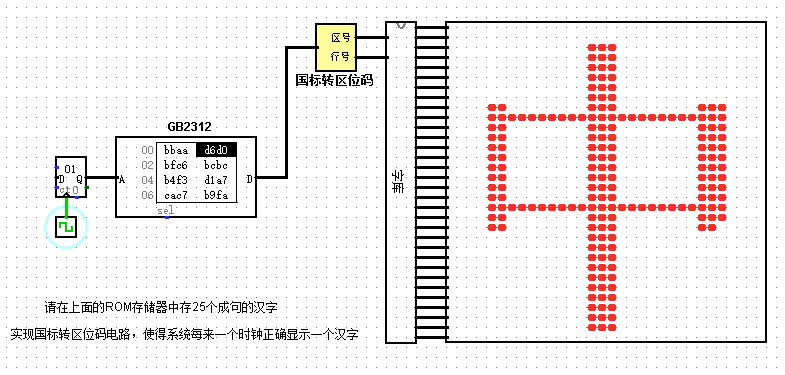


1. 海明解码：

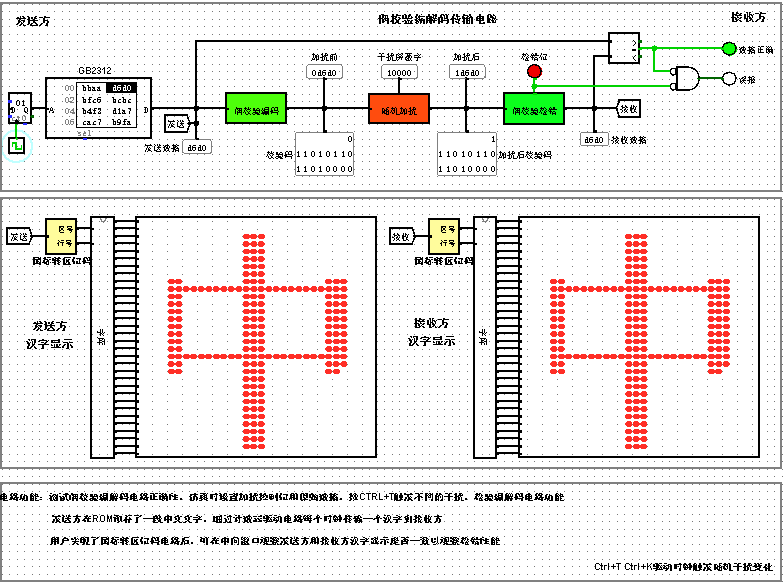


## 实验测试与结果

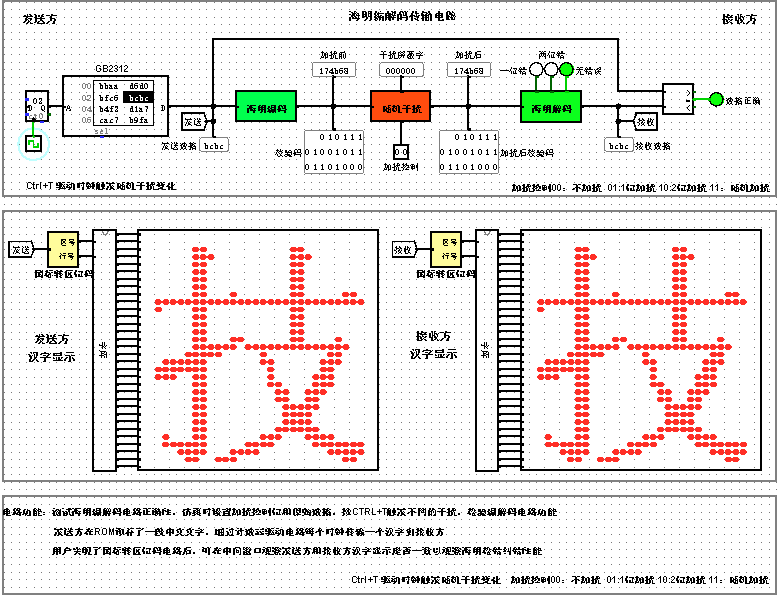
### 汉字编码测试



### 偶校验测试



### 海明校验测试



# 

# **1.4 总结与心得**

## **1.4.1 实验总结**

本次实验主要作了如下几点工作：

a.掌握GB2312以及区位码转换方法

b.了解字模码的原理

c.学习奇偶校验校验位

d.掌握奇偶校验检错基本原理

e.学习海明码编解码电路基本原理

f.熟悉流水数据传输机制

## **1.4.2 实验心得**

在这次实验中，我们动手探究验证书上的知识，让我们对系统的内部结构都有了更深的了解，并且对计算机组成原理也有了更深层次的理解，同时这次课程设计还锻炼了我的实验动手能力，也培养了我的认真负责的科学态度和团队合作的精神。

# 实验二 运算器组成实验

## 实验目的与内容

### 实验目的

1. 熟悉 Logisim 软件平台。
2. 掌握运算器基本工作原理
3. 掌握运算溢出检测的原理和实现方法；
4. 理解有符号数和无符号数运算的区别；
5. 理解基于补码的加/减运算实现原理；
6. 熟悉运算器的数据传输通路。

### 2.1.2 实验内容

1) 八位串行可控加减法电路设计

利用已经封装好的全加器（封装 1）设计 8 位串行可控加减法电路，其引脚电路如下图所示。

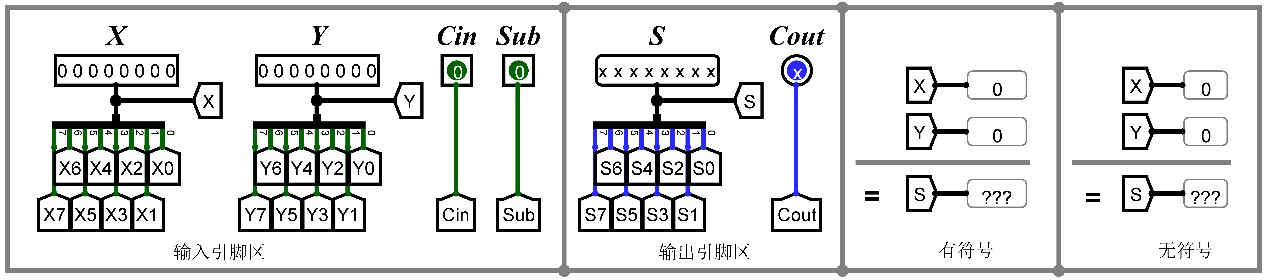
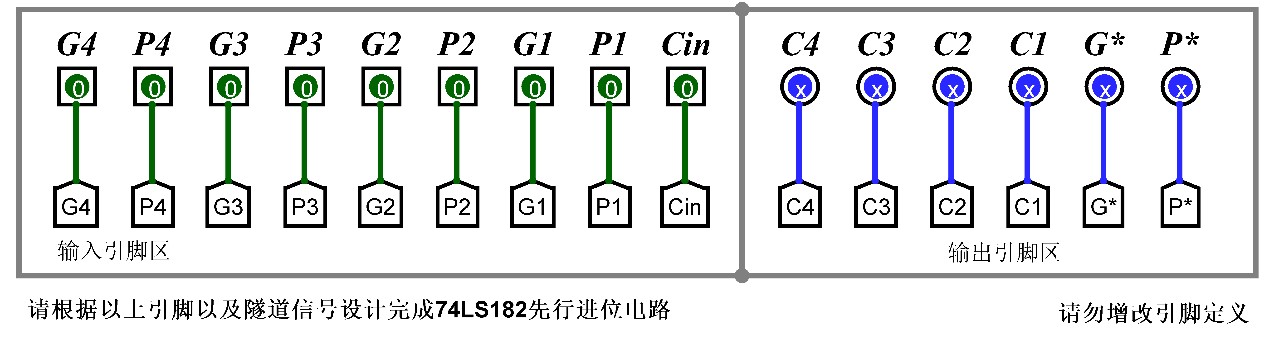


图 **1** 八位串行可控加减法电路引脚定义

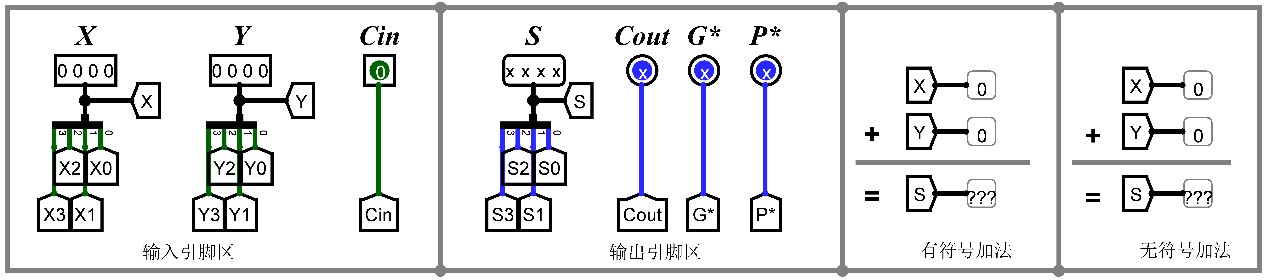
1. 位先行进位电路

根据下图定义的输入输出引脚完成 4 位先行进位电路。

图 **2** 四位先行进位电路引脚定义

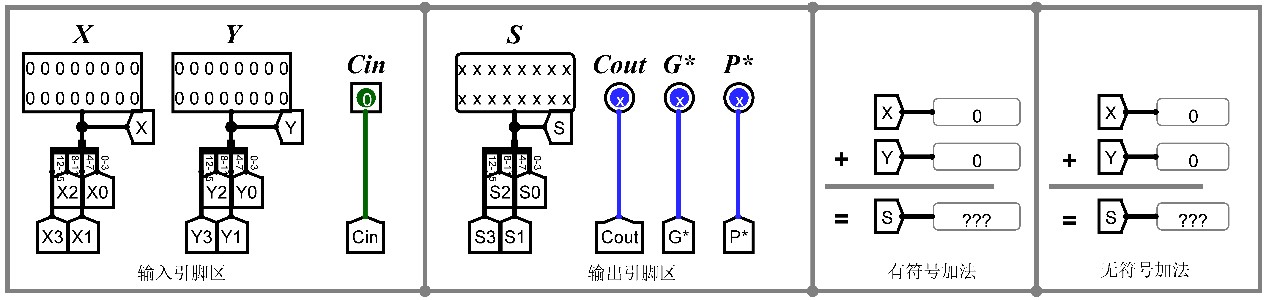
1. ① 四位快速加法器设计

利用已经前一步设计好的四位先行进位电路构造四位快速加法器，其引脚定义如下图。

图 **3** 四位快速加法器引脚定义

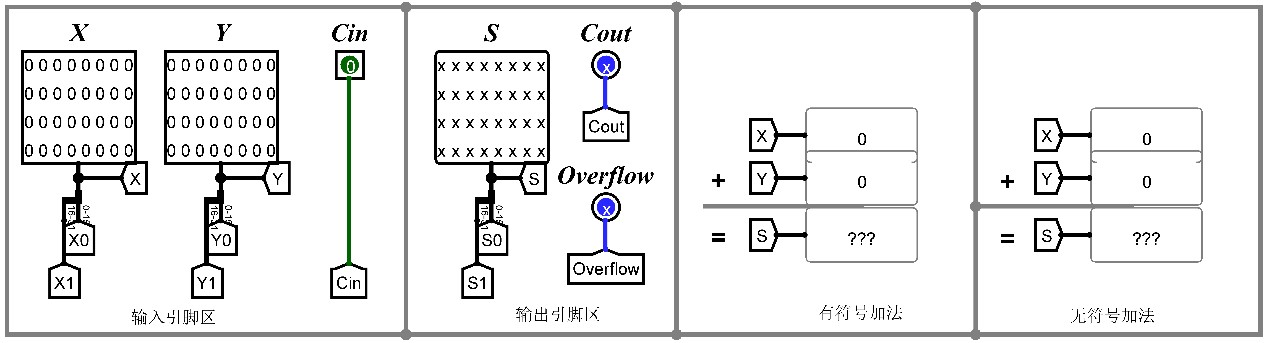
**②** 十六位快速加法器设计

利用四位先行进位电路和四位快速加法器构造十六位组间先行进位，组内先行进位快速加法器，其引脚定义如下图。

图 **4** 十六位快速加法器引脚定义

**③** 32位快速加法器设计

利用前面构建的部件完成 32 位快速加法器，并分析其时间延迟

 图 **5 32** 位快速加法器引脚定义

1. 32 位 MIPS 运算器设计

构建 **32** 位运算器。利用封装好的 32 位加法器以及 logisim 平台中现有运算部件（禁用系统自带的加法器，减法器）构建一个 32 位运算器，可支持算术加、减、乘、除，逻辑与、或、非、异或运算、逻辑左移、逻辑右移，算术右移运算，支持常用程序状态标志（有符号溢出 OF、无符号溢出 CF，结果相等 Equal），运算器功能以及输入输出引脚见下表，在主电路中详细测试自己封装的运算器，在报告中分析该运算器的优缺点。

表 **1.** 芯片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| X | 输入 | 32 | 操作数 X |
| Y | 输入 | 32 | 操作数 Y |
| ALU\_OP | 输入 | 4 | 运算器功能码，具体功能见下表 |
| Result | 输出 | 32 | ALU 运算结果 |
| Result2 | 输出 | 32 | ALU 结果第二部分，用于乘法指令结果高位或除法指令的余数位，其他操作为零 |
| OF | 输出 | 1 | 有符号加减溢出标记，其他操作为零 |
| UOF | 输出 | 1 | 无符号加减溢出标记，其他操作为零  溢出条件（加法和小于加数，减法差大于被减数） |
| Equal | 输出 | 1 | Equal=(x==y)?1:0, 对所有操作有效 |

表 **2.** 运算符功能

|  |  |  |
| --- | --- | --- |
| ALU OP | 十进制 | 运算功能 |
| 0000 | 0 | Result = X << Y 逻辑左移 （Y 取低五位）  Result2=0 |
| 0001 | 1 | Result = X >>>Y 算术右移 （Y 取低五位）  Result2=0 |
| 0010 | 2 | Result = X >> Y 逻辑右移 （Y 取低五位）  Result2=0 |
| 0011 | 3 | Result = (X \* Y)[31:0]; Result2 = (X \* Y)[63:32] 有符号 |
| 0100 | 4 | Result = X/Y; Result2 = X%Y 无符号 |
| 0101 | 5 | Result = X + Y (Set OF/UOF) |
| 0110 | 6 | Result = X - Y (Set OF/UOF) |
| 0111 | 7 | Result = X & Y 按位与 |
|  | |  |  |
| 1000 | | 8 | Result = X | Y 按位或 |
| 1001 | | 9 | Result = X⊕Y 按位异或 |
| 1010 | | 10 | Result = ~(X |Y) 按位或非 |
| 1011 | | 11 | Result = (X < Y) ? 1 : 0 符号比较 |
| 1100 | | 12 | Result = (X < Y) ? 1 : 0 无符号比较 |

## 实验原理与方案

### 2.2.1 实验原理

1. 串行可控加减法设计

该电路采用全加器实现，每一位的全加器的cin连接后一位的cout端，进位串行传递，同时有一控制信号sub，当sub为1是，将输入Y的各位取反加一组成其补码形式，从而达到相减的目的。溢出采用双符号位判断，若符号位为01或10时表示结果数据溢出。

1. 并行快速加法器

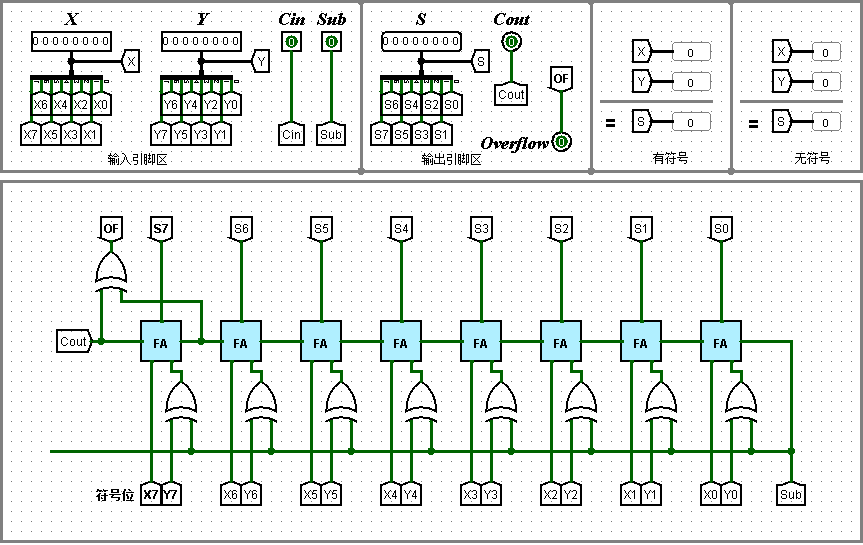
采用公式先行算出各位的进位，再一次性并行输入各个加法器的输入端，一次并行得出结果，从而达到更快的运行速度。

1. ALU设计

详见上述实验内容中的描述。

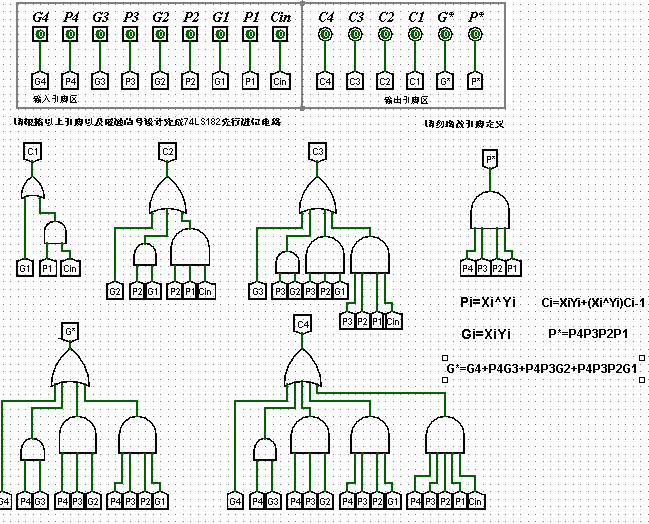
### 实验方案

1. 串行可控加减法设计

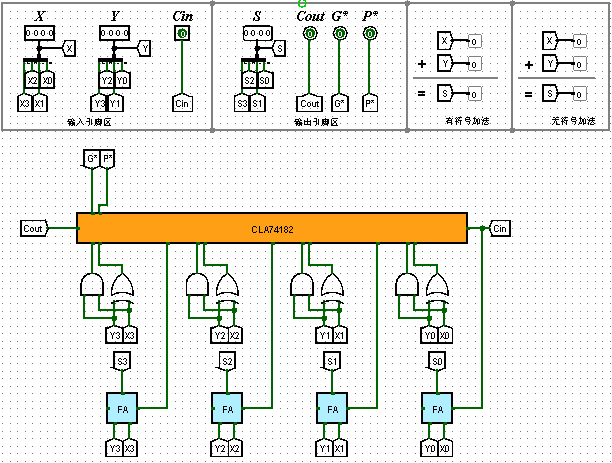


2) 并行快速加法器

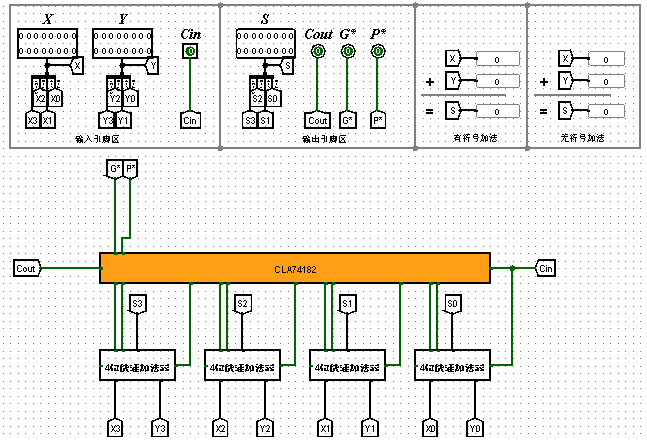
① CLA74182



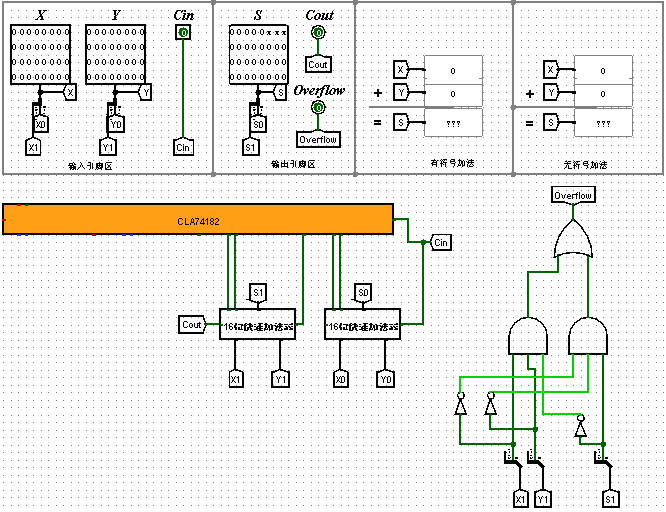
② 4位并行快速加法器



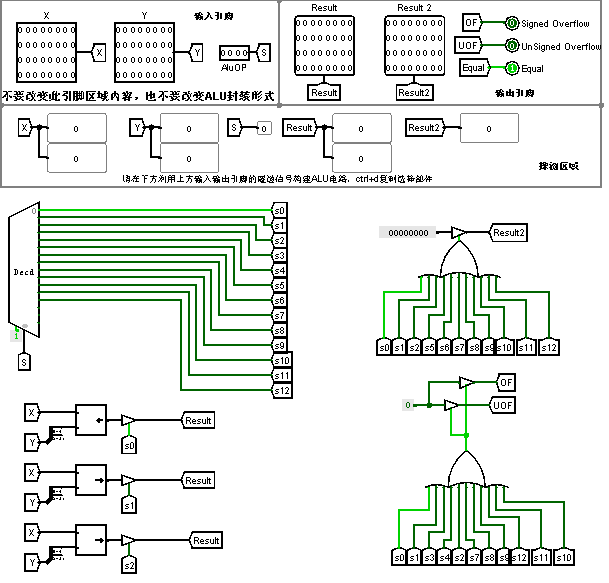
③ 16位并行快速加法器

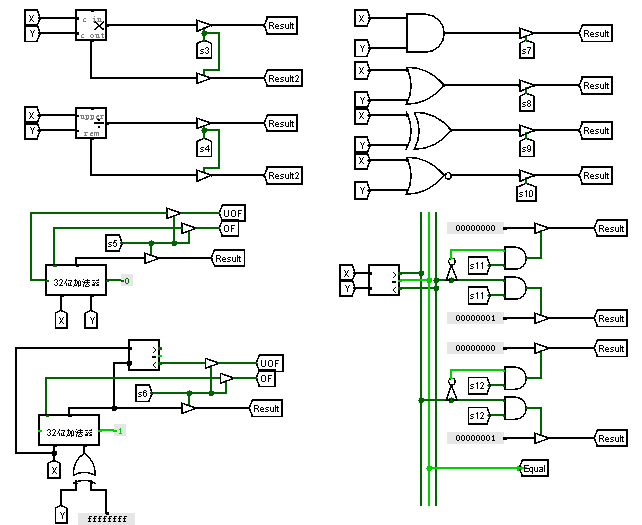


④ 32位并行快速加法器



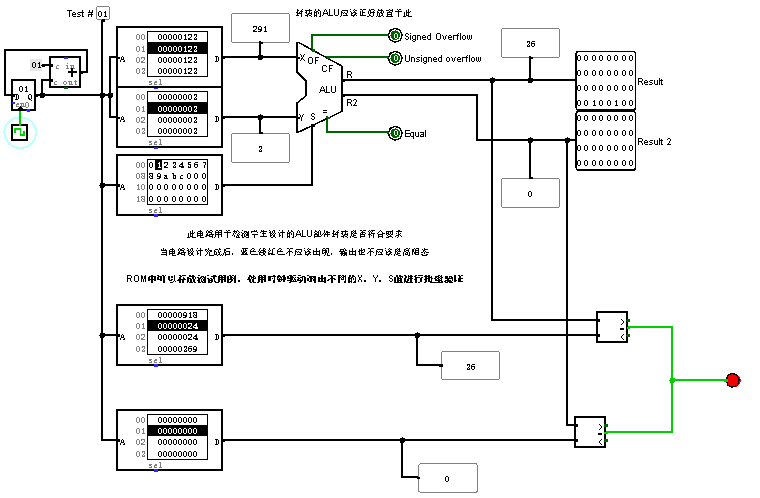
1. ALU设计





## 实验测试与结果

### 2.3.1 ALU综合测试



# **2.4 总结与心得**

## **2.7.1** 实验总结

本次实验主要作了如下几点工作：

a.熟悉 Logisim 软件平台。

b.掌握运算器基本工作原理

c.掌握运算溢出检测的原理和实现方法；

d.理解有符号数和无符号数运算的区别；

e.理解基于补码的加/减运算实现原理；

f.熟悉运算器的数据传输通路。

## 

## **2.4.2 实验心得**

实验过程中，一个小小的失误可能导致花费很多的时间来检测电路，但即使如此也有可能查不出电路出错，而导致不得不重新进行电路的连接。这次实验也让我渐渐体会到细致认真的重要性，以及电路成功后的喜悦。

# 

# 实验三 补码乘法器实验

## 3.1 实验目的与实验内容

### 3.1.1 实验目的

1）熟悉 Logisim 软件平台。

2）掌握运算器基本工作原理

3）掌握运算溢出检测的原理和实现方法；

4）理解一阶补码的乘法运算实现原理；

5）熟悉运算器的数据传输通路

### 3.1.2 实验内容

1) Logism 实验

1. 学习使用 Logism 工具栏上的功能
2. 学会使用子电路，并能将子电路放到 main 电路中使用
3. 学习使用时钟，并能使用时钟单步或自动运行
4. 学会使用分线器，理解线宽的概念
5. 学会使用隧道，学习使用探测器，了解 logisim 数据监测方法。
6. 熟悉按键、LED，数码管等基本输出设备

注（此部分要求可在作中学，相应部分在后续实验中均有要求，简单熟悉平台后可直接跳越到实验 2）

1. 一位补码乘法器设计

下图给出了一位补码乘法器的引脚定义以及主要部件，请增加控制电路和数据通路使得该电路能自动完成 8 位补码一位乘法运算，设置引脚值，然后驱动时钟仿真，电路自动完成运算，运算结束结果传输到输出引脚。

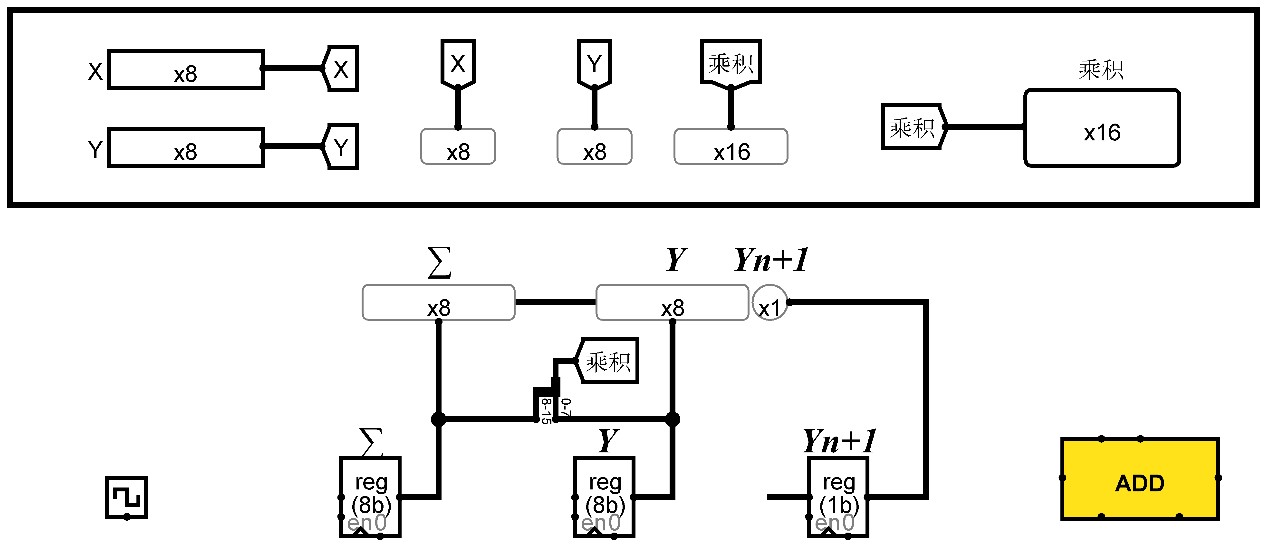


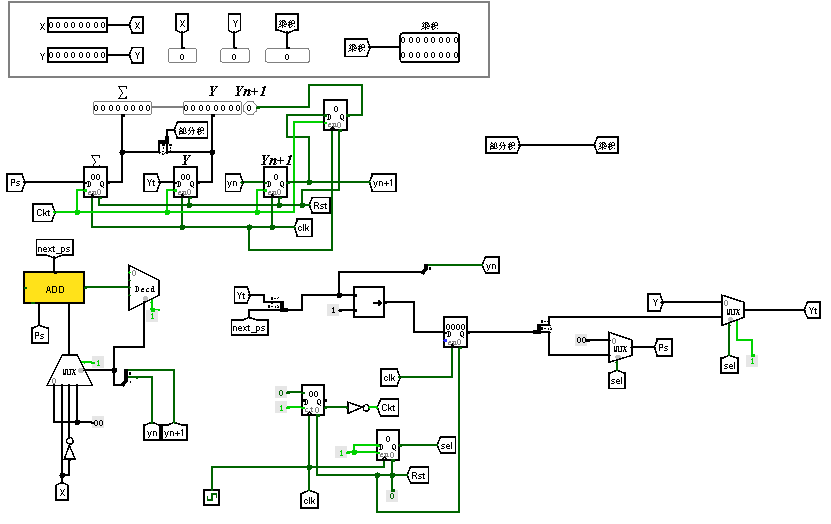
图 **8** 一位补码乘法器引脚定义及电路主要部件

## 实验原理及方案

**3.2.1 实验原理**

采用booth算法的补码乘法运算。Booth算法对乘数从低位开始判断，根据两个数据位的情况决定进行加法、减法还是仅仅移位操作。判断的两个数据位为当前位及其右边的位(初始时需要增加一个辅助位0)，移位操作是向右移动。

**3.2.2 实验方案**



# **3.3 总结与心得**

## **3.3.1 实验总结**

本次实验主要作了如下几点工作：

a.熟悉 Logisim 软件平台

b.学习并掌握运算器基本工作原理

c.学习运算溢出检测的原理和实现方法

d.体会一阶补码的乘法运算实现原理

e.熟悉运算器的数据传输通路

## **3.3.2 实验心得**

这次实验可能较前两次实验的工作量小了许多，但是难度却有所提高，而且渐渐的使我体会到了其中的乐趣。同时，老师给出的电路的巧妙性也让我佩服不已，让我进一步感受到实践的乐趣与重要性，课本知识的复现也是非常重要的。

# 

# 实验四 存储系统综合实验

## 4.1 实验目的与内容

### 实验目的

1) 熟悉 Logisim 软件平台

2) 熟悉 ROM、RAM 存储器的使用

3) 掌握存储器字扩展，位扩展的基本原理

4) MIPS CPU 设计功能部件---寄存器文件

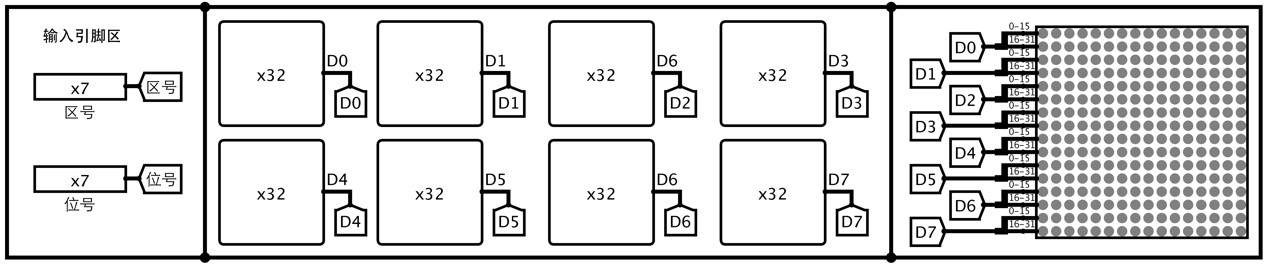
### 4.1.2 实验内容

**1**) 存储扩展实验

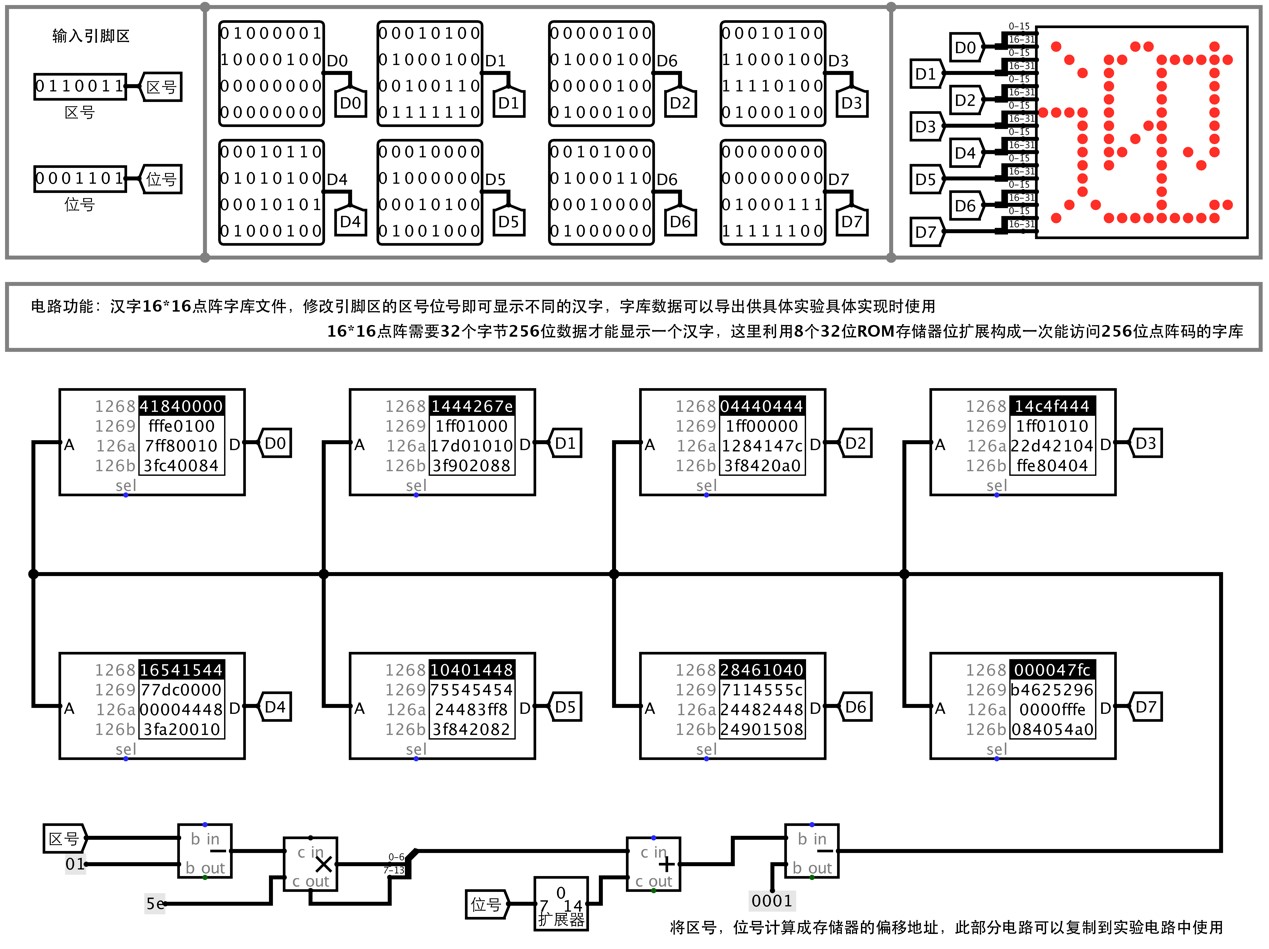
实验目的：掌握存储扩展基本原理。

实验内容：设计字库文件，利用指定规格存储器进行存储器字扩展和位扩展。

实验要求：现有如下 ROM 部件，2个4K\*16位ROM，3 个 4K\*32 位 ROM（由于数据已经导入到ROM中，所以顺序固定为从左到右地址增加），7 个 16K\*32 位 ROM，请构建 GB2312 16\*16 点阵字库存储器电路，电路输入为汉字区号和位号，由于 16\*16 点阵的字模码需要256 位点阵信息才能显示一个汉字，所以电路输出为 8\*32 位（256 位点阵信息），实验电路输入输出引脚如下图：

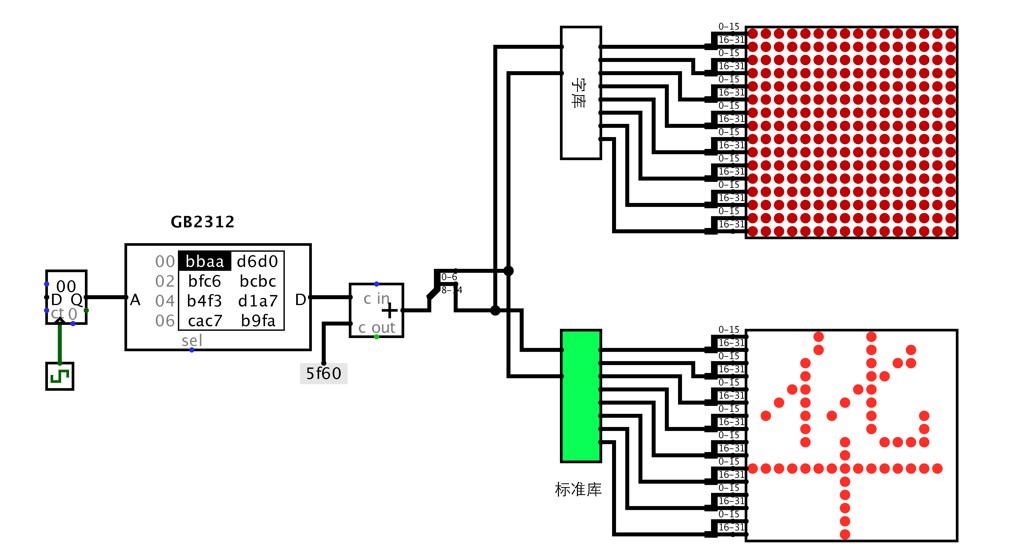
 图1 16\*16点阵字库电路输入输出引脚

本实验的主要目的是进行存储器字扩展和位扩展，故实验工程文件中已经提供了一个参考实现，完成实验所需的点阵信息均可以通过该电路直接导出后载入，也可直接复制拷贝，区位码转存储器地址的电路也可一并参考使用。

 图2 6\*16点阵字库参考实现

功能测试：

设计实现待测字库后，可以在如下字库测试电路进行功能测试，测试时按下 ctrl+T(command+T MAC)键启动时钟自动仿真即可，通过对比上下两个显示区显示内容是否一致即可验证字库功能正确性。

图3 16\*16点阵字库功能测试电路

1. **MIPS** 寄存器文件设计

实验目的：为 MIPS CPU 构造核心功能部件，进一步熟悉多路选择器，译码器，解复用器等 Logisim 部件的使用。

实验内容：设计完成满足如下规格要求的 MIPS 通用寄存器组。

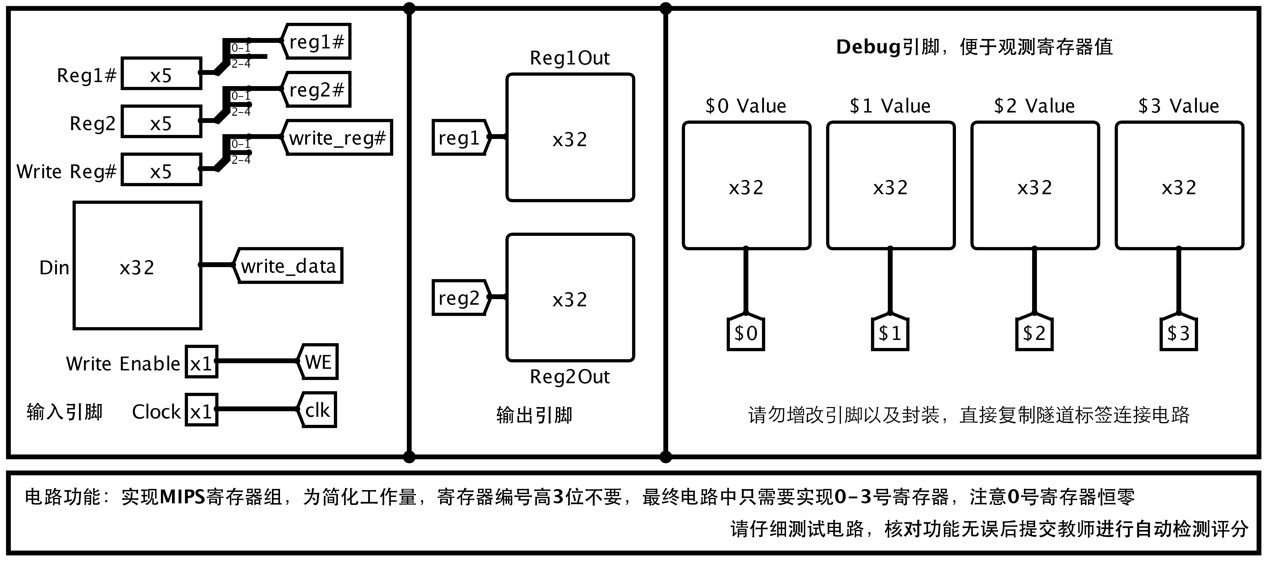
① 利用 logisim 平台构建一个 MIPS 寄存器组，内部包含 32 个 32 位寄存器，其具体功能如下，具体封装文件为 regfile.circ.

表 **1.** 芯片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| R1# | 输入 | 5 | 读寄存器 1 编号 |
| R2# | 输入 | 5 | 读寄存器 2 编号 |
| W# | 输入 | 5 | 写入寄存器编号 |
| Din | 输入 | 32 | 写入数据 |
| WE | 输入 | 1 | 写入使能信号，为 1 时，CLK 上跳沿将  Din 数据写入 W#寄存器 |
| CLK | 输入 | 1 | 时钟信号，上跳沿有效 |
| R1 | 输出 | 32 | R1#寄存器的值 |
| R2 | 输出 | 32 | R2#寄存器的值 |
| $s0 | 输出 | 32 | 编号为 16 的寄存器的值 |
| $s1 | 输出 | 32 | 编号为 17 的寄存器的值 |
| $s2 | 输出 | 32 | 编号为 18 的寄存器的值 |
| $ra | 输出 | 32 | 编号为 31 的寄存器的值 |

注意零号寄存器值应该恒零！

实验电路输入输出引脚如下图：

图3 MIPS Regfile输入输出引脚

* 1. 为减少实验中画图工作量，实验工程文件中对 5 位寄存器地址进行了简化，具体见引脚示意图，最终只需实现 4 个寄存器，0 号寄存器功能仍
  2. 然是恒零。后续实验中如需要使用 32 个寄存器的 MIPS 寄存器文件组，将提供标准组件。
  3. 注意时钟信号和电平信号不要混连，时钟仅仅触发状态改变。

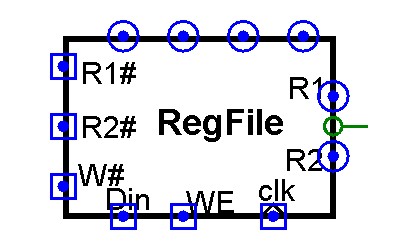


图4. MIPS Regfile封装形式

## 实验原理与方案

### 实验原理

1. 存储拓展

采用两片12位地址，16位字宽的存储芯片进行位扩展成一片12位地址的32位字宽的存储芯片。然后再将四片12位地址的32位字宽的存储芯片进行字扩展成一片14位地址的32位存储芯片。

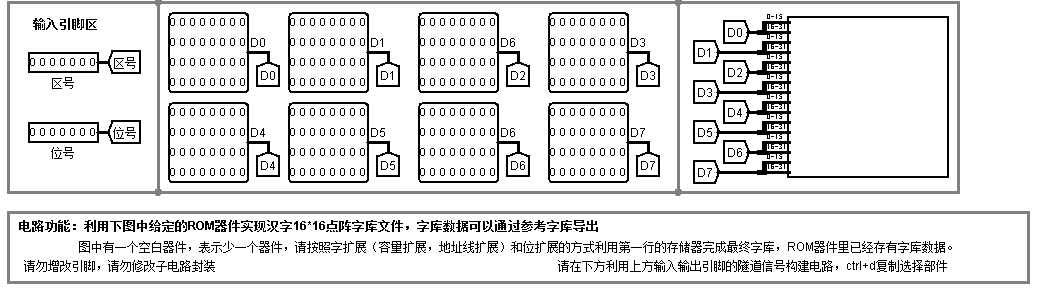
1. 寄存器堆

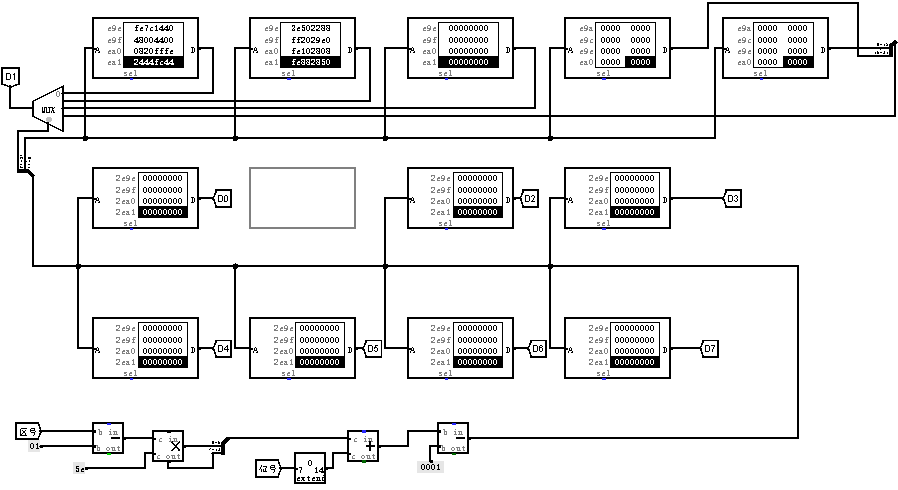
根据reg1#和reg2#的输入选择指定寄存器将内容从reg1和reg2中输出。

根据write\_reg#和输入使能信号WE将din的内容写入指定的寄存器中。

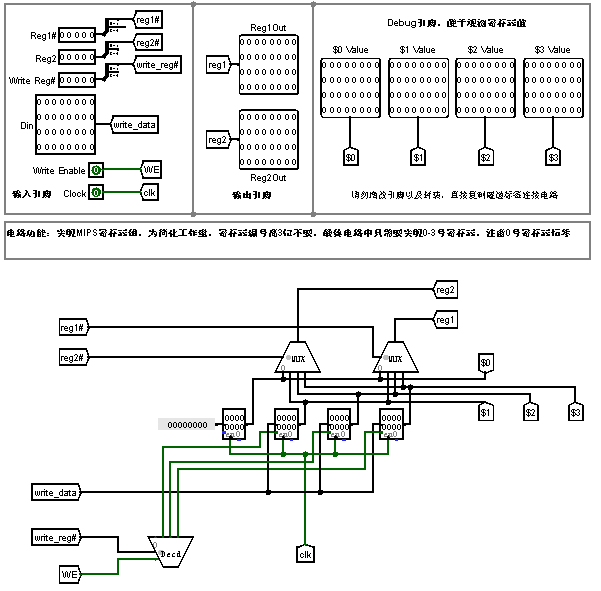
### 实验方案

1. 存储扩展





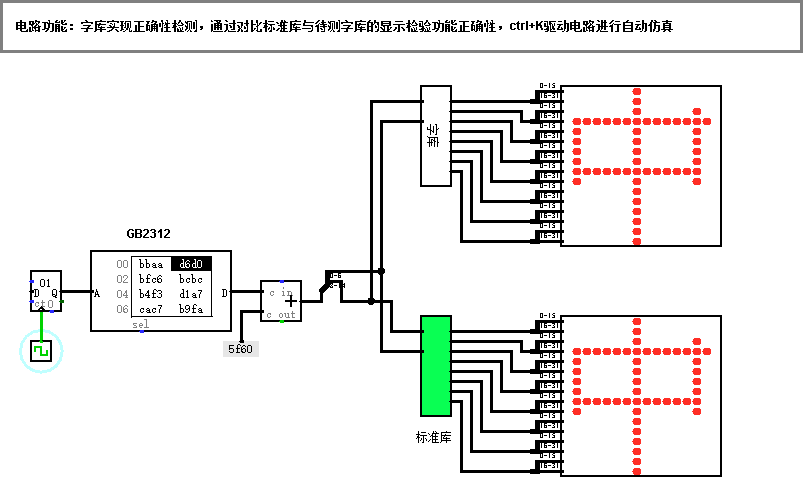
1. mips寄存器堆



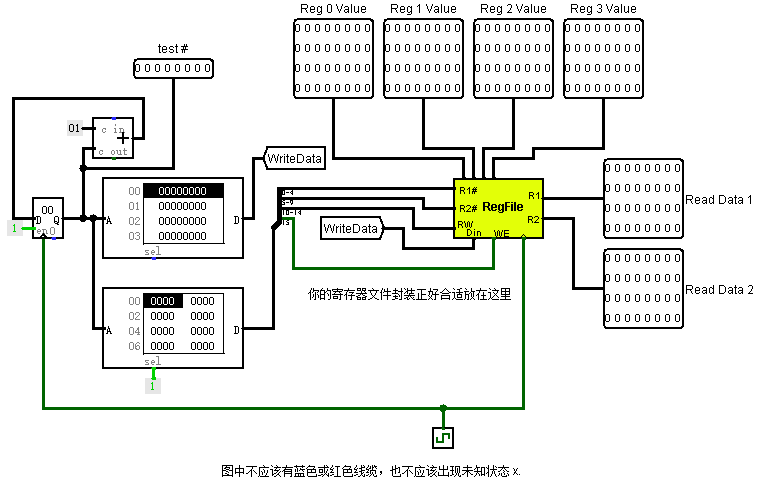
## 

## 实验测试与结果

1. 存储扩展



2）寄存器堆



# 

# **4.4 总结与心得**

## **4.7.1 实验总结**

本次实验主要作了如下几点工作：

a.熟悉 Logisim 软件平台

b.熟悉 ROM、RAM 存储器的使用

c.掌握存储器字扩展，位扩展的基本原理

d.MIPS CPU 设计功能部件---寄存器文件

## **4.4.2 实验心得**

最后一次实验，有了前几次的经验教训，自然变得更加有信心了，但是最后一次的内容是关于存储系统的，和我们前几次做的实验有些差别，还是需要仔细研究。因此，我首先仔细研究课本将RAM和ROM的详细知识点搞清楚，确定没有问题以后才在软件上画电路图，这次我比前几次更加仔细有耐心，最后我只出现了一个小错误就完成了测试。在验收的时候，因为我将书本看的很明白了，所以也能更清楚的回答助教的问题，解释我画的电路图的流程和原理等等

# 实验总结与评价

通过这么多次的计组实验，让我通过实际的动手操作来验证书本上的理论知识，让我被运算器，存储器，控制器以及输入输出等各种计算机部件的魅力所吸引，这种收获不仅仅是从书本上能体会到的。

除此之外，这几次实验也对我的认真有了巨大的测验，因为每当连错一根线时，就有可能导致整个实验受阻，很多时候不得不重新进行连线，因为由于线路太多根本找不到错误。所以慢慢的，自己学会了将电路的各个功能组建分开实现，而不是全部放在同一个电路中。这样，效率不仅提高了许多，而且差错也更快速一点。

最后，感谢老师的给我们安排的实践，让我进一步体会到了计算机的魅力，或许只有自己动手实现才能真正体会到其中的乐趣，而这恰恰的书中所不具有的。

**参考文献**

[1]秦磊华 王小兰 计算机组成原理实验指导及课程设计指导书(基于EDA平台)武汉：华中科技大学出版社，2010年.

[2]秦磊华，吴非，莫正坤.计算机组成原理. 北京：清华大学出版社，2011年.

[3]袁春风编著. 计算机组成与系统结构. 北京：清华大学出版社，2011年.

|  |
| --- |
| 一、原创性声明 |
| 本人郑重声明本实验的实验报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在抄袭行为。  特此声明！  **作者签字:** |
| 二、对实验的学术评语 |
|  |
| 三、对实验的评分 |
| |  |  |  |  | | --- | --- | --- | --- | | 评分项目  （分值） | 报告撰写  （30分） | 实验过程  （70分） | 最终评定  （100分） | | 得分 |  |  |  | |
| **指导教师签字: 年 月 日** |