

计算机组成原理课程设计总体报告

**学 院： 计 算 机 工 程 学 院**

**班 级： 计算2114**

**姓 名: 庄佳强**

**学 号： 202121331104**

二〇二二 年 12 月1 日

目录

[一 题目要求 3](#_Toc10386)

[二 设计思路 3](#_Toc11393)

[实验效果： 3](#_Toc16326)

[目标： 4](#_Toc12144)

[设计思路： 4](#_Toc4082)

[三 核心模块的实现方法 5](#_Toc7193)

[多数字滚动： 6](#_Toc31249)

[分数的判断和记录： 6](#_Toc30357)

[数码管的直接显示： 7](#_Toc9394)

[中断和返回： 8](#_Toc14991)

[游戏进行部分模块： 9](#_Toc14027)

[四 成果展现 10](#_Toc27987)

[界面： 10](#_Toc23030)

[测试方法： 10](#_Toc28038)

[测试实例： 11](#_Toc18395)

[五 优化策略 13](#_Toc21889)

[六 实验总结 14](#_Toc909)

# 一 题目要求

结合 MIPS 处理器和 MIPS 汇编程序开发一个小型游戏机。游戏的主题和方式 不限，可以由你自由设计。这个游戏机至少应该使用到 Logisim 提供的三种输入外 设之一（按钮/摇杆/键盘）作为输入，至少应该使用到 Logisim 提供的三种输出外 设之一（数码管/点阵屏/终端）作为输出。你的硬件系统将以自己设计的 MIPS 处 理器为核心，通过内存映射的方式挂载外设；你的软件系统将以 MIPS 汇编程序为 核心，通过 SW/LW 访问外设，并通过其他算术指令完成数据的计算、控制流等。进一步使用流水线并行和中断机制对系统进行优化。

本来只想完成基于 MIPS 处理器的应用系统开发-1这个课题，但在不断完善中，超出了预期，完成了基于 MIPS 处理器的应用系统开发-2这一扩展选题。

# 二 设计思路

## 实验效果：

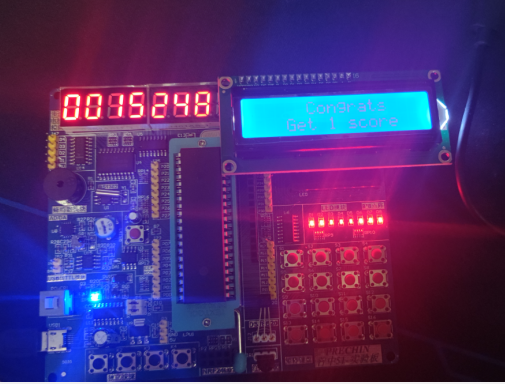


图1 实验效果图

## 目标：

本次设计实践的游戏为滚动数字抽奖机。

核心模块：多数字的滚动，分数的判断和记录，数码管的直接显示，游戏进行部分模块。

拓展创新模块：硬币添加系统，中断部分。

## 设计思路：

通过Mar编码，完成数字的变化，数字是否相同的判断，再编译成指令后写入到logisim的指令存储器中。

通过logisim编程，完成数码管访问存储映射，按钮控制Mar指令中断跳转到判断是否相同的子程序。按钮操作lw指令存储硬币数到数据存储器中。

# 三 核心模块的实现方法

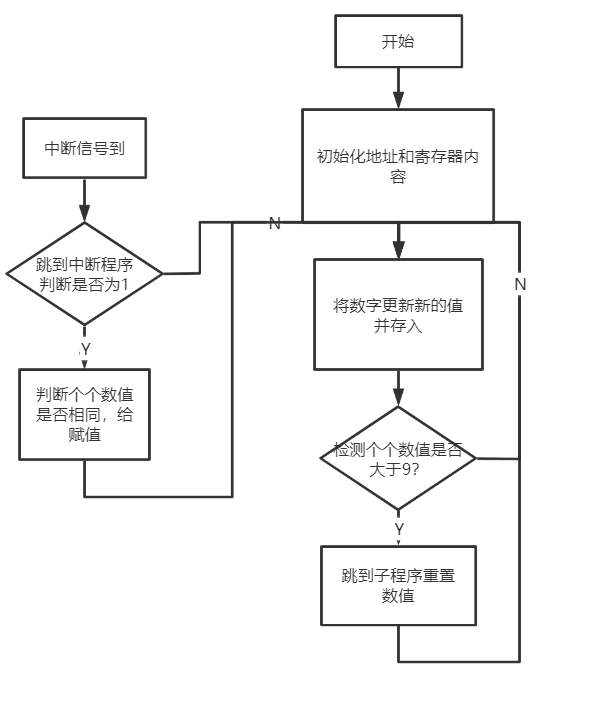


图2 流程图

## 多数字滚动：

首先在Mar中编辑好要使用的代码 ，在把写好的代码编译，输出到logisim中的指令存储器中。

在代码中，我使用的是给三个寄存器不同的频率不断的赋值加1，再把更新好的寄存器依次存储到字节地址中。从而实现伪随机行性，也实现了数字的更新。

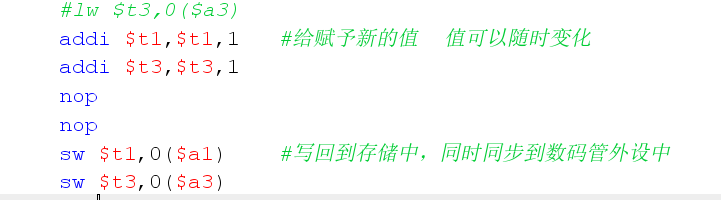


图3 滚动代码图

## 分数的判断和记录：

通过调用判断数据存储器中的000地址是否为1。当按下button后，要用中断程序，跳到指定的代码，通过读取000地址值判断是否要跳转到判断子程序，在判断程序中会把000清空，如果有两个相同 就把008地址值写入到寄存器，再加一后重新存回到008地址中。如果有三个相同 就把008地址值写入到寄存器，再加二后重新存回到008地址中。之后退出子程序，返回到判断开头正常进行。



图4 判断信号清除图

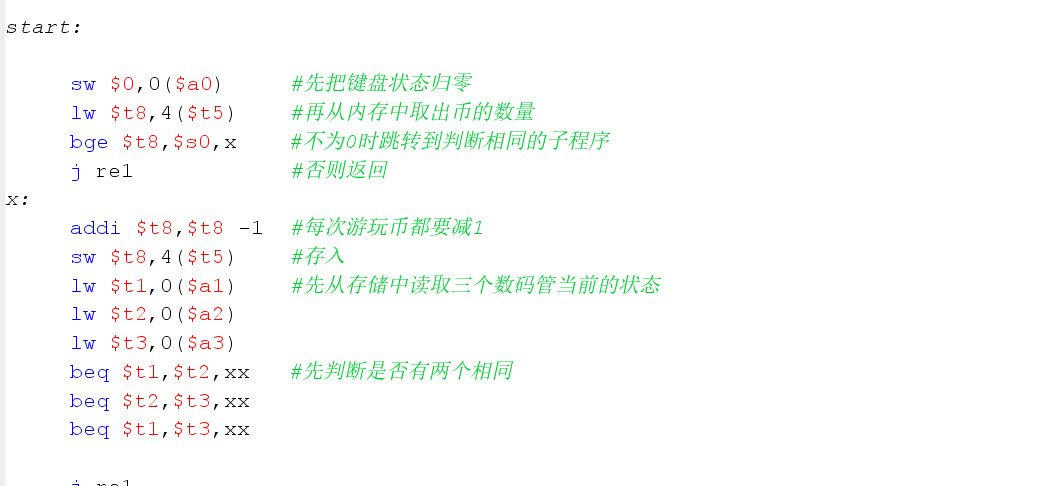


图5 判断相同图

## 数码管的直接显示：

要实现数码管的显示，首先要把数字更新存储到数据存储器中。前两个模块已经实现了。通过比较当前要sw的字节地址，如果对应了三个寄存器存储的字节地址，就将当前的要存入的内容存储对应的数码管的寄存器中。得分也同理。要存入数据只前把数据存到分数寄存器中。

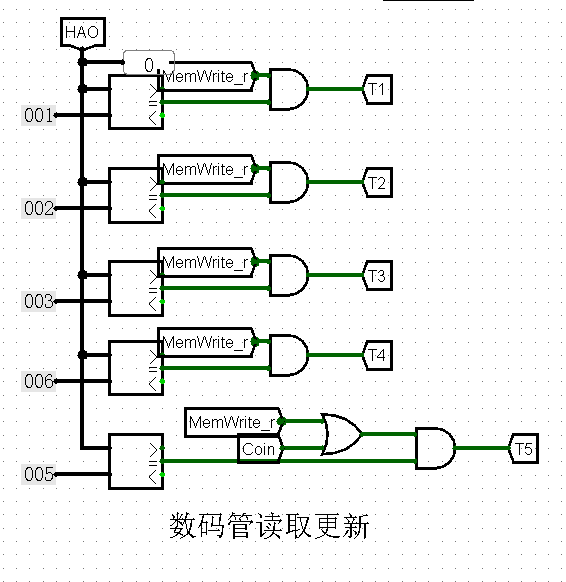


图6 数码管更新图

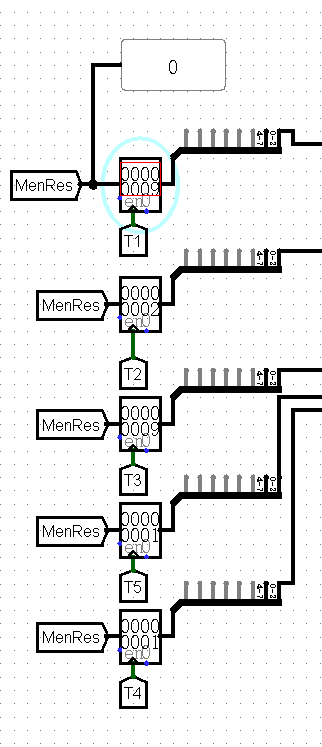


图7 存储器读取到数码管图

## 中断和返回：

一开始当按下游戏按钮后，中断服务后同时记录该指令的下一条指令的地址，存储到寄存器，当判断部分进行完后，回跳转调用之前保存好的指令地址，赋给PC,完成一次中断。

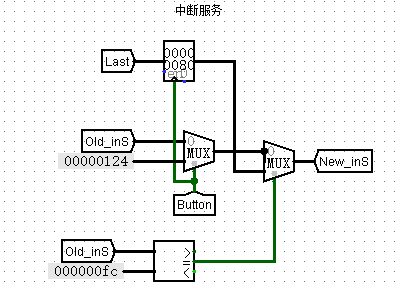


图8 中断服务图

## 游戏进行部分模块：

当按下游戏的游玩button时，游戏会中断跳转到对应的判断指令，同时为000地址赋予1，让程序可以进入判断子程序。

当按下游戏的加币buton时，数据存储器会强制跳到币随对应的地址，并把加一后的币数加入到数据存储器中，当程序进行游玩时，会把币对应的地址lw到寄存器，减一后再存入。当币为0时无法游玩。

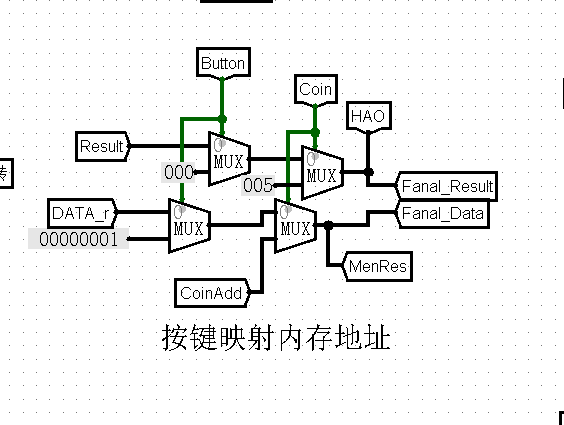


图9 按键映射内存图

# 四 成果展现

## 界面：

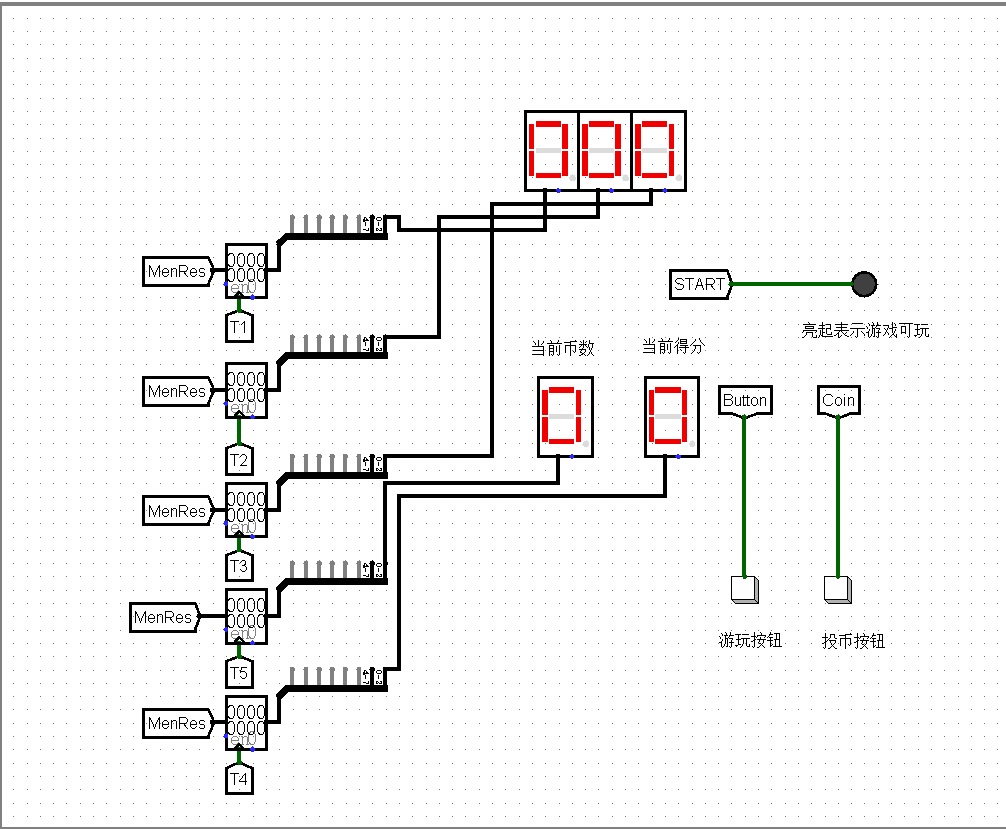


图10 界面图

## 测试方法：

通过查看数码管中数字和数据寄存器中是否相同看查看数码管显示是否正确。

通过按下按键后查看是否有正确加上数字，来查看分数系统是否正常执行。

通过记录每次bug出现的时机，在通过查看个个电信号和指令的执行，通过单步执行clk，在不断的尝试中寻找bug。

此上操作反复进行多次，直到完全不出现bug。

## 测试实例：

测试加币系统 —正常

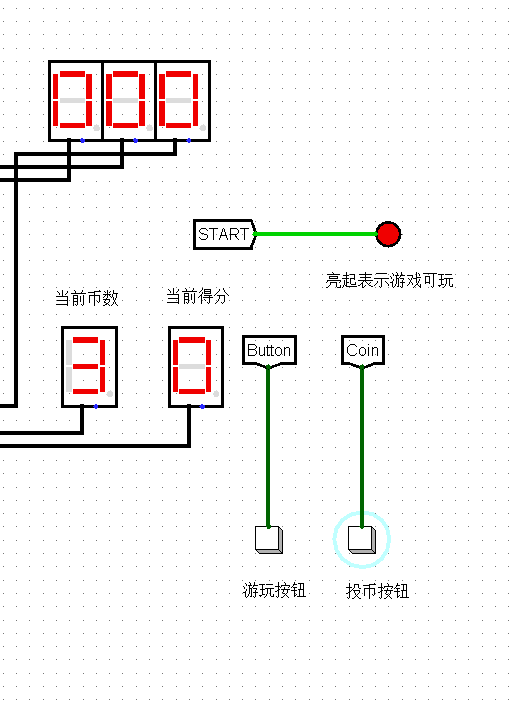


图11 初始界面图

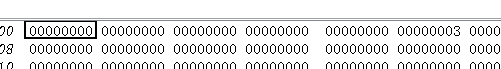


图12 数据存储器状况图

测试数码管显示系统 —正常

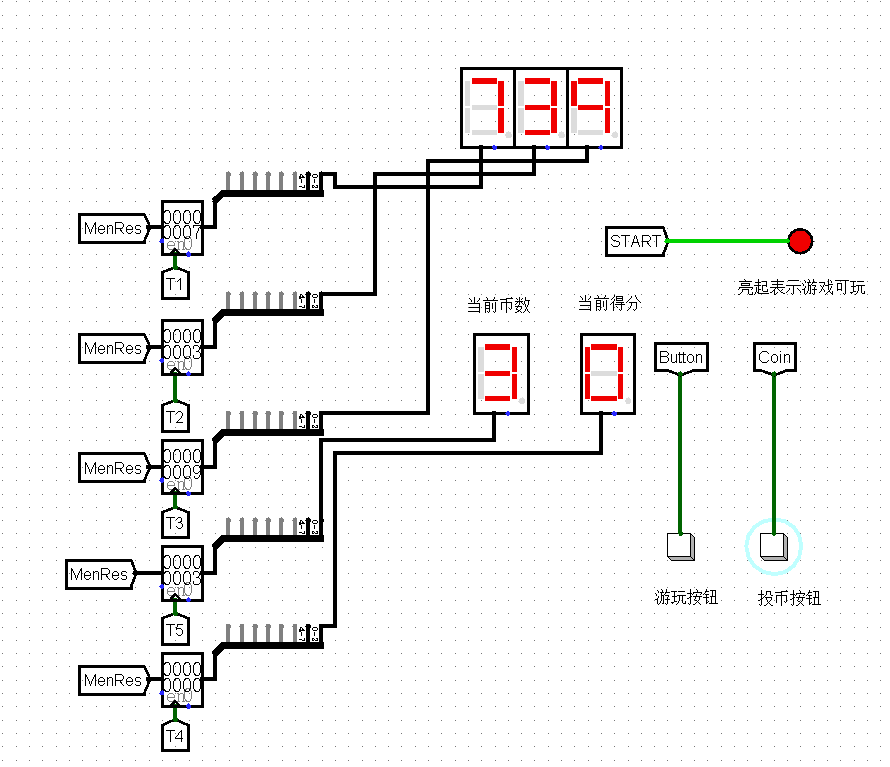


图13 开始变化图

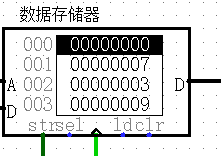


图14 数据存储器对应变化图

分数阶段 —正常

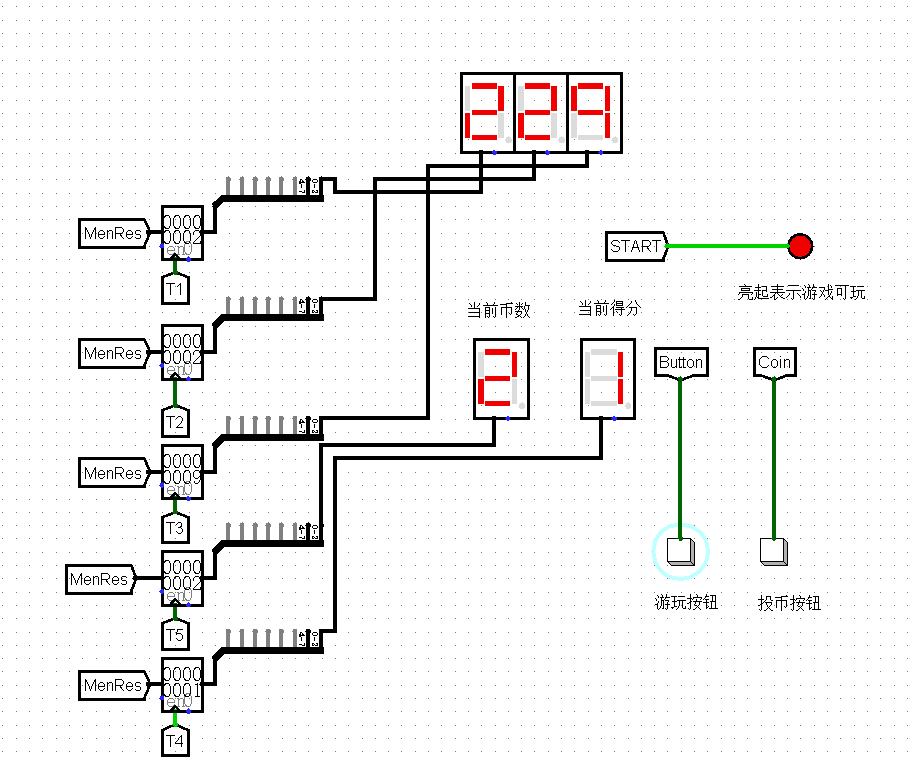


图15 得分图

# 五 优化策略

一开始采用为二级流水线，但在这个情况下，当你要使用到游玩button时，程序有可能会因为上个指令占用了ALU或者RegFile 导致本次的000地址未被读取到，从而没有跳转指令。但采用三级流水线后就解决了问题。000地址可以保证被读取，和跳转到回应的位置上。

一开始编程中判断000地址是否为1没有用到中断跳转的办法，而是使用轮询的方法，这样导致这次按下的按钮不一样对应本次的数码管显示的数字，这样出错的机会很大，而使用中断后，就可以保证每次跳转到判断子程序时，都是当前数码管所对应的数值被比较。

# 六 实验总结

1. 通过本次课设，我巩固了这学期学到的有关知识点。特别是复习了一下MIPS指令集的使用。
2. 本次实验是在断断续续中完成，期间有时已经完成了，但过一两天有想到了新的点子，又不断优化。一个好的项目，是通过不断的补丁下建成的。
3. 一开始测试的时候怎么也跑不通，bug也众多，一度让我想放弃这个课题，但是心里不甘，觉得应该坚持下来，在不断的修改和试错后，bug变少了程序也可以简单的跑动，让我非常的开心。坚持到最后后，课题基本完成，也没有什么重大的bug出现。这时是十分的自豪，坚持就一定有成果。