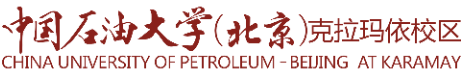
实 验 报 告





课 程 ：计算机组成原理

实验名称：1.MIPS寄存器文件设计实验

2.MIPS RAM设计实验（选做）

学 院 ：石油学院

专 业 ：数据科学与大数据技术

学 号 ：2023015509

姓 名 ：胡林森

2024年秋季学期

目录

[一、实验目的 3](#_Toc182234895)

[ 1.MIPS寄存器文件设计实验 3](#_Toc182234896)

[ 2.MIPS RAM设计实验（选做） 3](#_Toc182234897)

[二、实验内容 3](#_Toc182234898)

[ 1.MIPS寄存器文件设计实验 3](#_Toc182234899)

[ 2.MIPS RAM设计实验（选做） 4](#_Toc182234900)

[三、实验步骤 4](#_Toc182234901)

[ 1.MIPS寄存器文件设计实验 4](#_Toc182234902)

[ 2.MIPS RAM设计实验（选做） 5](#_Toc182234903)

[四、实验结果 5](#_Toc182234904)

[ 1.MIPS寄存器文件设计实验 5](#_Toc182234905)

[初始电路 5](#_Toc182234906)

[设置Din=1，W#=1，WE=1 5](#_Toc182234907)

[重新将CLK设置为0，再设置Din=2，W#=2 6](#_Toc182234908)

[设置R1#=1，R2#=2 6](#_Toc182234909)

[ 2.MIPS RAM设计实验（选做） 6](#_Toc182234910)

[初始时，设置Addr=0，Din=0，CLK=0，WE=0，Mode=00； 6](#_Toc182234911)

[设置Din=12345678H，WE=1，经过CLK上升沿（将CLK从原为0的状态设置为1），检查：Dout的值是否已被设置为12345678H？ 6](#_Toc182234912)

[重新将WE和CLK设置为0，再设置MODE=01，检查：Dout的值是否已被设置为00000012H？ 6](#_Toc182234913)

[再设置MODE=10，检查：Dout的值是否已被设置00001234H？ 7](#_Toc182234914)

[五、问题和感想 7](#_Toc182234915)

# 一、实验目的

## 1.MIPS寄存器文件设计实验

- 了解MIPS寄存器文件基本概念，进一步熟悉多路选择器、译码器、解复用器等Logisim组件的使用，并利用相关组件设计实现MIPS寄存器文件。

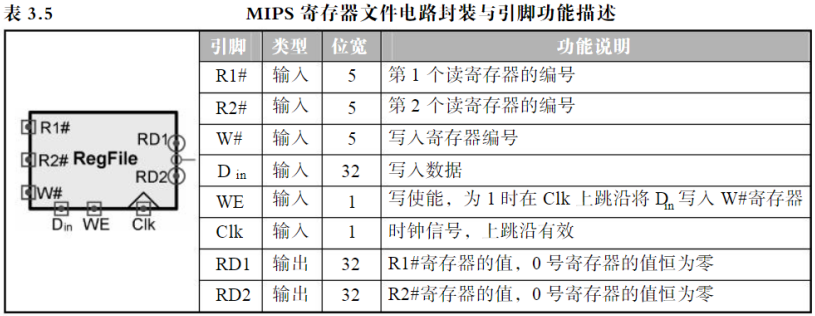
## 2.MIPS RAM设计实验（选做）

- 理解主存地址基本概念和存储位扩展基本思想，并能利用相关原理构建同时支持字节、半字、字访问的存储子系统。

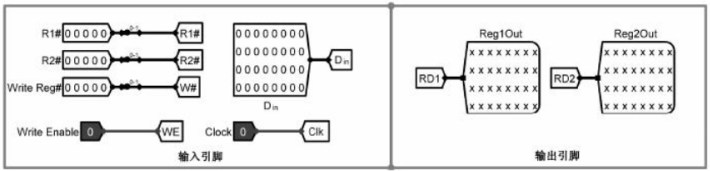
# 二、实验内容

## 1.MIPS寄存器文件设计实验

- 利用多路选择器、译码器等组件设计实现一个MIPS寄存器文件，内部包含32个32位寄存器，其电路封装与引脚功能描述如表3.5所示。



- 具体电路详见storage.circ的MIPS寄存器文件子电路，电路框架如图3.8所示。

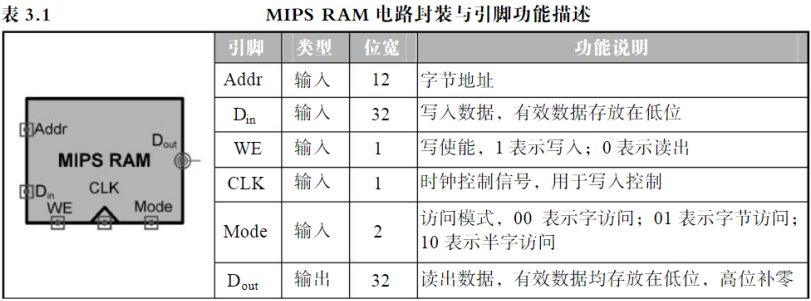




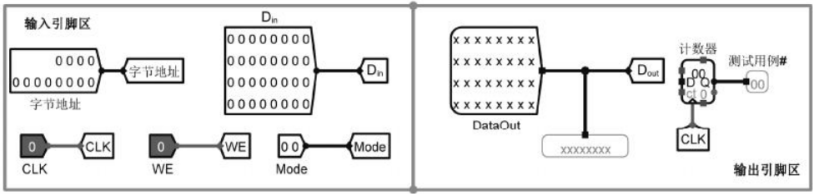
- 为减少绘图工作量，实验工程文件中对5位寄存器地址进行了简化，具体见图3.8左上角的引脚示意图。图中采用分线器将5位寄存器编号的低2位引出，实际只使用了2位编号，所以最终只需实现4个寄存器，其中0号寄存器的值仍然是恒零。

## 2.MIPS RAM设计实验（选做）

- Logisim中RAM组件只能提供固定的地址位宽，数据输出也只能提供固定的数据位宽，访问时无法同时支持字节、半字、字3种访问模式，实验要求利用4个4KB×8位的RAM组件进行扩展，设计完成既能按8位也能按16位和32位进行读写访问的32位存储器，最终存储器电路封装与引脚功能描述如表3.1所示。



- 电路框架如图3.6所示，请使用相应的RAM组件实现MIPS RAM。注意本实验中对数据总线的控制可以考虑使用三态门组件，也可以使用多路选择器组件。





# 三、实验步骤

## 1.MIPS寄存器文件设计实验

- 完成设计后，按如下步骤对所设计电路进行验证：

• 初始时，设置Din=0，W#=0，CLK=0，WE=0，R1#=0，R2#=0；

• 设置Din=1，W#=1，WE=1，经过CLK上升沿（将CLK从原为0的状态设置为1），检查：寄存器1的值是否已被设置为1？

• 重新将CLK设置为0，再设置Din=2，W#=2，经过CLK上升沿，检查：寄存器2的值是否已被设置为2？

• 设置R1#=1，R2#=2，检查：RD1和RD2的值是否分别为1和2？

## 2.MIPS RAM设计实验（选做）

• 初始时，设置Addr=0，Din=0，CLK=0，WE=0，Mode=00；

• 设置Din=12345678H，WE=1，经过CLK上升沿（将CLK从原为0的状态设置为1），检查：Dout的值是否已被设置为12345678H？

• 重新将WE和CLK设置为0，再设置MODE=01，检查：Dout的值是否已被设置为00000012H？

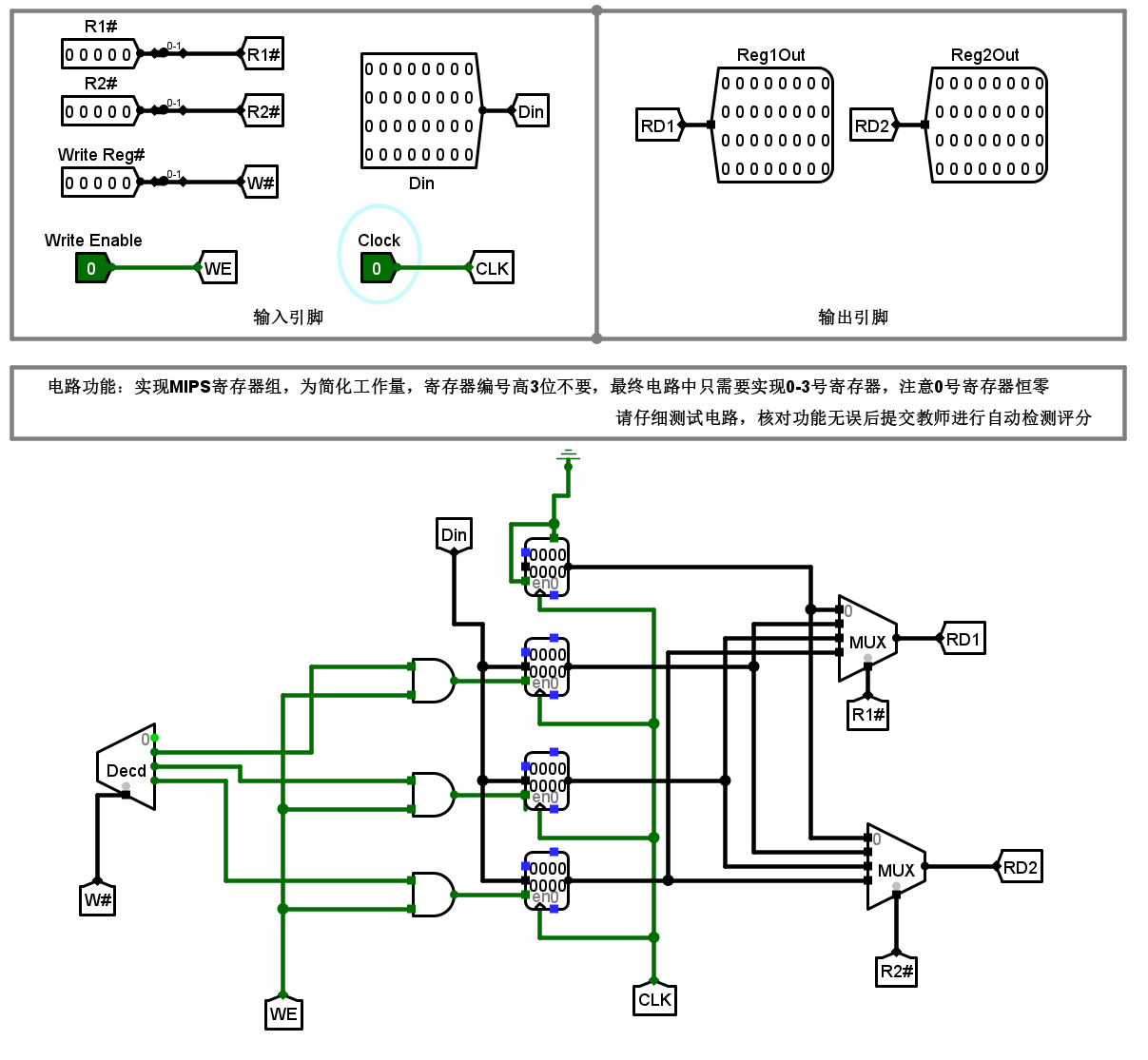
• 再设置MODE=10，检查：Dout的值是否已被设置00001234H？

# 四、实验结果

## 1.MIPS寄存器文件设计实验

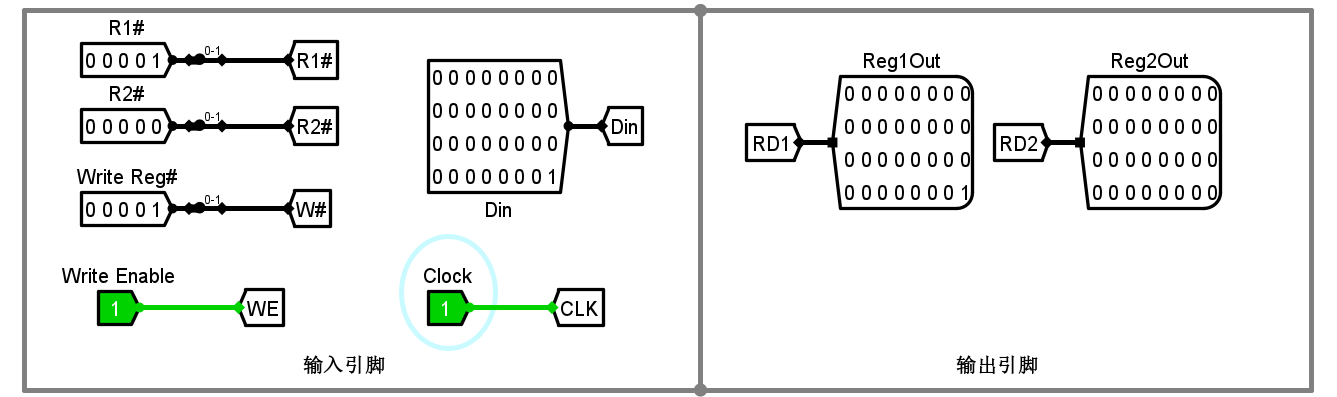
### 初始电路

设置Din=0，W#=0，CLK=0，WE=0，R1#=0，R2#=0；



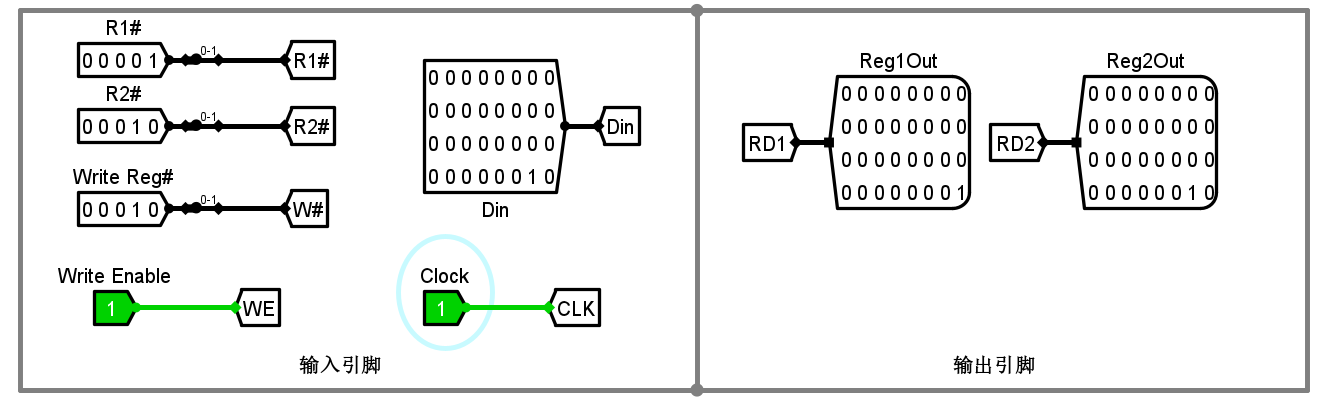
### 设置Din=1，W#=1，WE=1

经过CLK上升沿（将CLK从原为0的状态设置为1），检查：寄存器1的值为1，正确



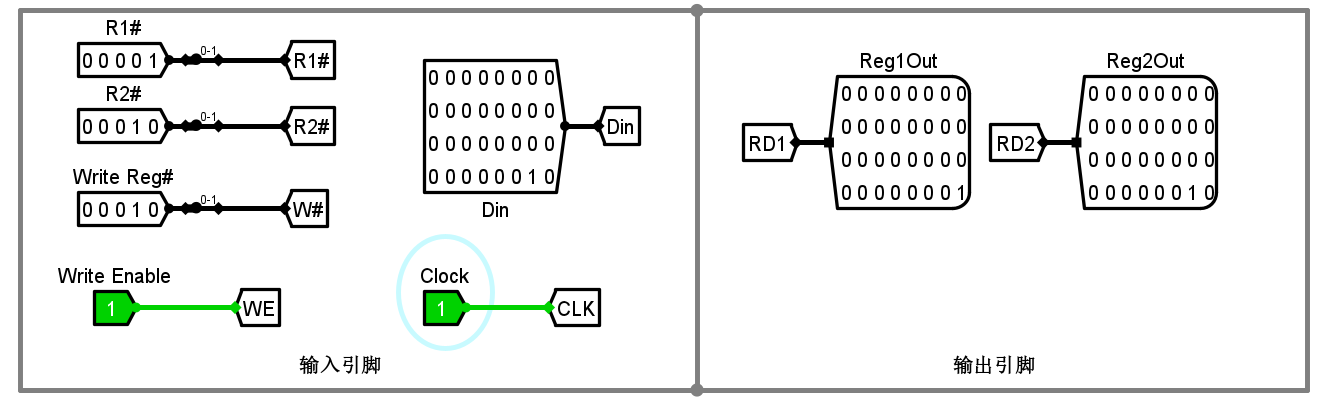
### 重新将**CLK设置为0，再设置Din=2，W#=2**

经过CLK上升沿，检查：寄存器2的值已被设置为2，正确。



### 设置R1#=1，R2#=2

检查：RD1和RD2的值分别为1和2，正确。



## 2.MIPS RAM设计实验（选做）

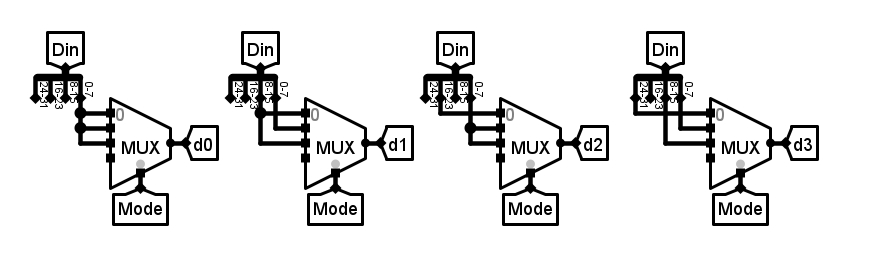
### 初始时，设置Addr=0，Din=0，CLK=0，WE=0，Mode=00；

电路主要分为三部分：写入、读出、模式控制。

**写入：**

要实现不同模式的访问，最小单位为字节，所以RAM位宽为8位，最大单位为字32位，所以一共需要4片8位RAM组件。把这四片RAM的数据对应标记为byte0，byte1，byte2，byte3。

Addr为12位字节地址，RAM的地址位宽为10位，所以可用分线器将字节地址低2位作为片选地址，00、01、10、11对应4个RAM，使每个RAM可输出一个字节，高10位为RAM的块内地址。

****

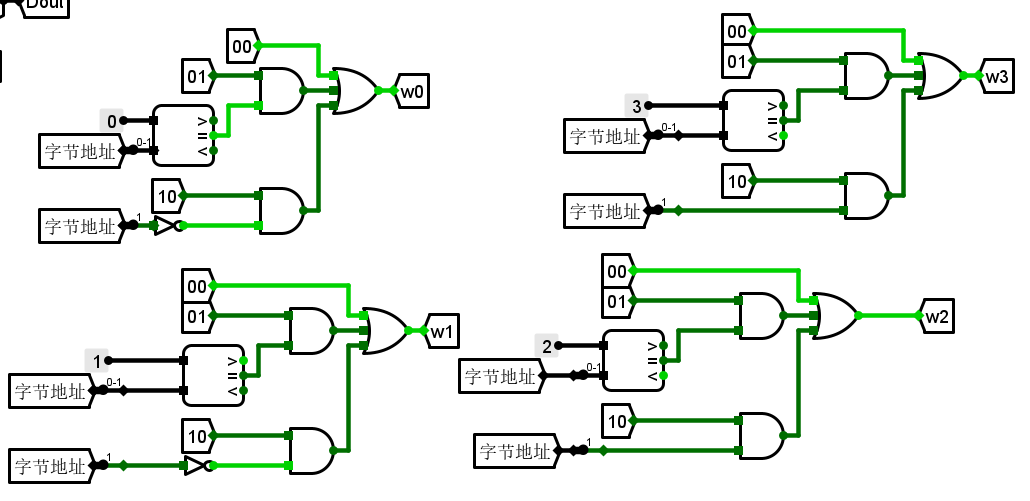
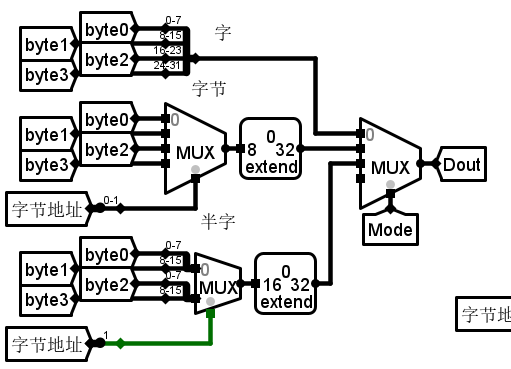
当mode = 00时，所有片选信号都要为真，进行写入

当mode = 01时，字节地址最低两位和寄存器编号相对应的片选信号为真，进行写入。

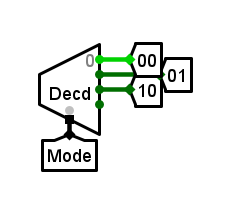
当mode = 10时，字节地址的倒数第二低的一位来决定选择那两片。当这一位等于0，应该选择01号组合，否则选择23号组合。

**读出：**

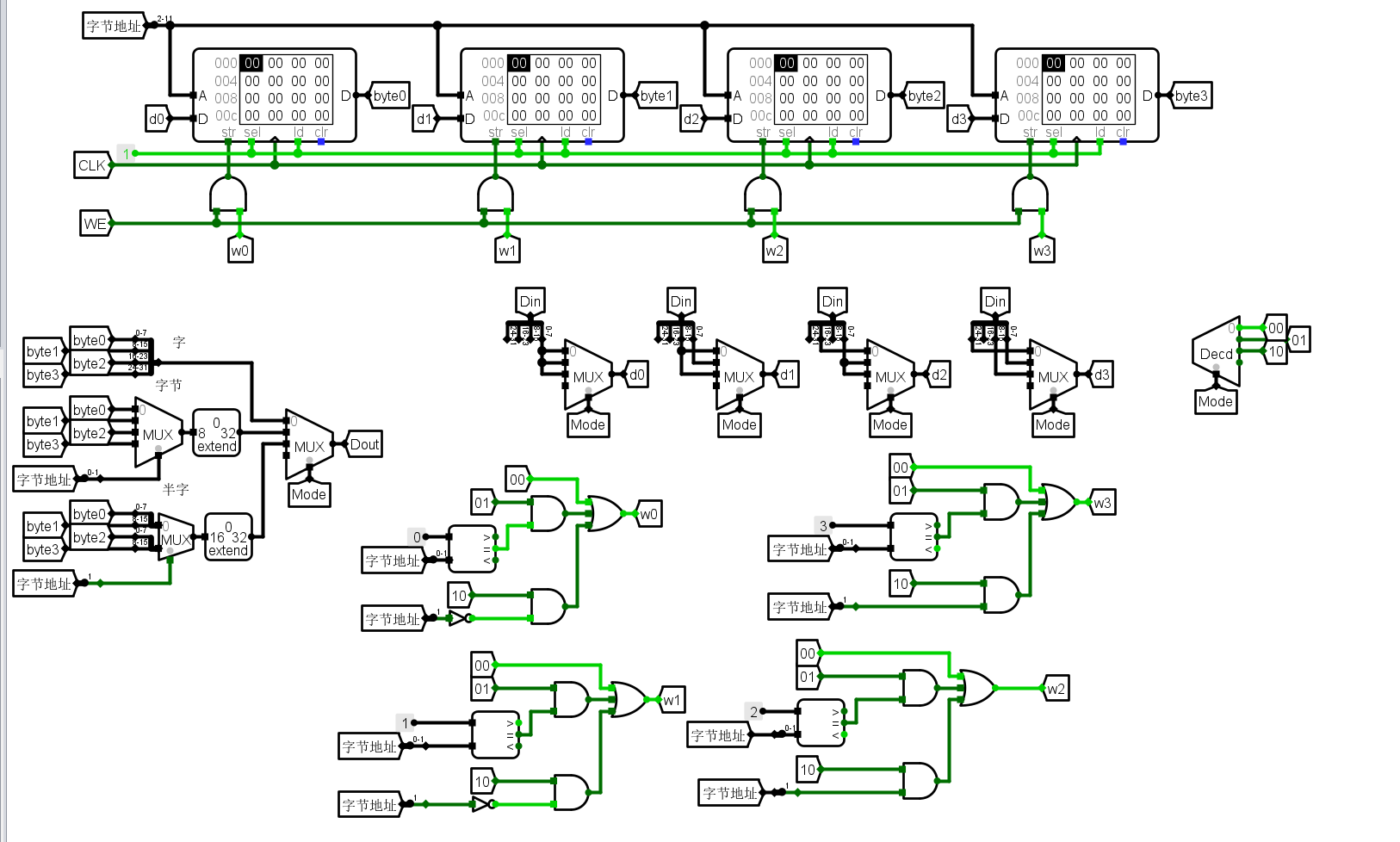
字读出时，直接将四个RAM的数据全部输出半字读出时，根据字节地址倒数第二位决定输出哪个半字(此时忽略字节地址最低位，倒数第二位实现片选)字节读出时，根据字节地址低两位决定输出哪个字节(字节地址低两位实现片选)



**模式控制**

****

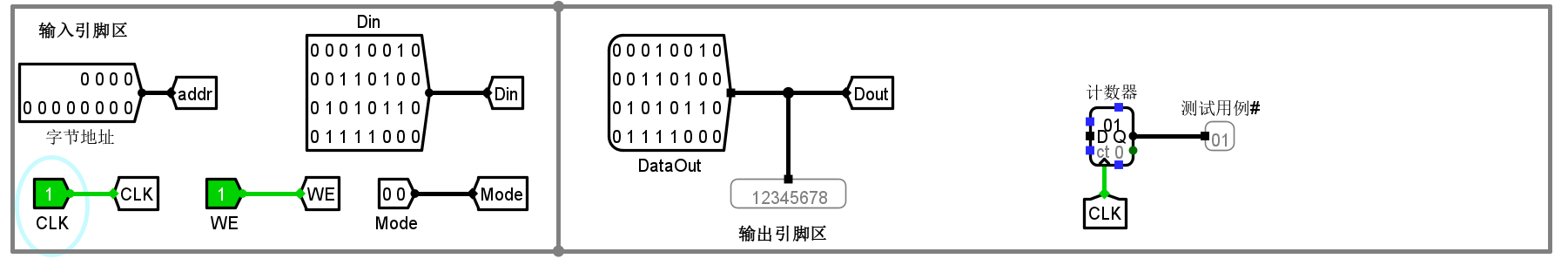
**整体电路**



### 设置Din=12345678H，WE=1，经过CLK上升沿（将CLK从原为0的状态设置为1），Dout的值为12345678H

地址后两位为00，所有片选信号都要为真，进行写入

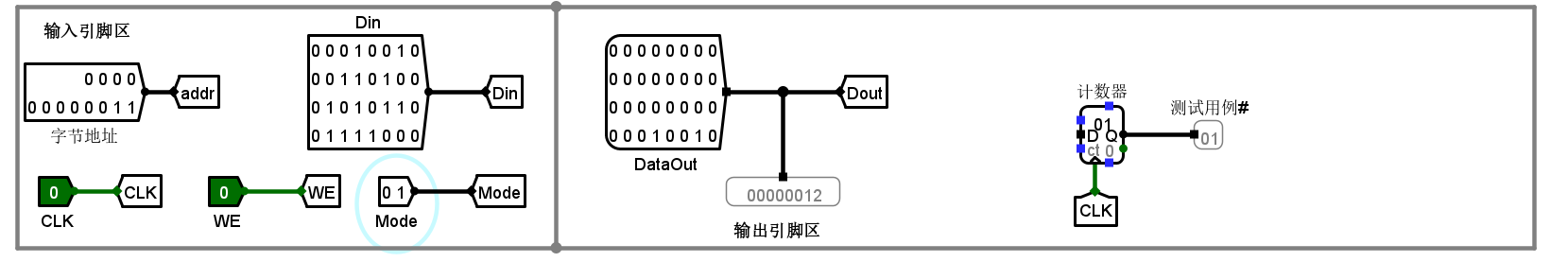
Mode值为00，字读写模式，从d0-d3分别输入0-7,8-15,16-23,24-31位数据



### 重新将WE和CLK设置为0，再设置MODE=01，Dout的值为00000012H

地址后两位为11，寄存器编号相对应的片选（3号）信号为真进行读取

Mode值为01，字节读写模式，读取8位2进制数：00010010，转化位16进制位00000012H

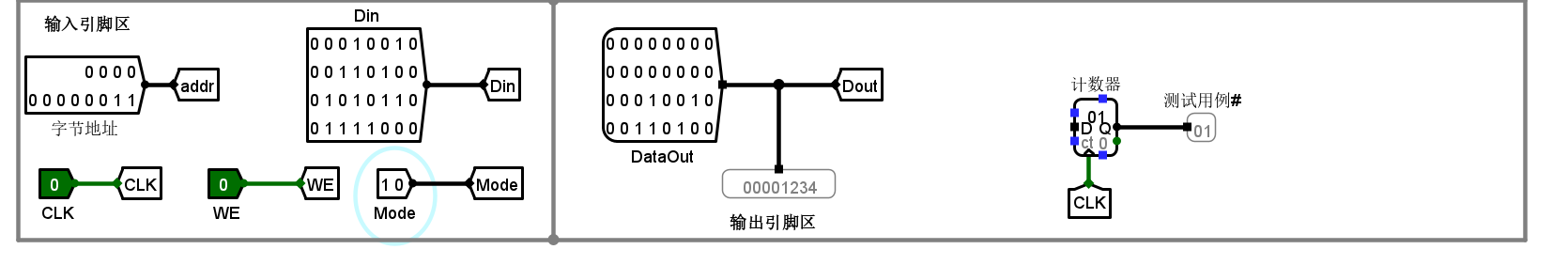


### 再设置MODE=10，Dout的值为00001234H

Mode = 10时，字节地址的倒数第二低的一位来决定选择那两片。当这一位等于0，应该选择0、1号组合，否则选择2、3号组合。

设置地址后两位为1\*，进行读取（\*代表任取）

读取2、3号，半字节读取，2进制数：0001，0010，0011，00100，转化为16进制数：00001234H



# 五、问题和感想

**问题：**

1.对于MIPS RAM设计，较为复杂，控制输入输出位数时多用分离器（splitter），将多位数分组，实验中常常将分离后的位数链接错误，需要理清逻辑，仔细地检查电路连接，并确保每个部分都按照设计逻辑正确连接。

2.对RAM部件不熟悉，通过网络资源进行学习。通过网络资源和参考书籍进行了额外的学习。

**感想：**

1.对于复杂电路，可以使用tunnel，将各部分组合起来，使逻辑更清晰。

2.加深了我对MIPS寄存器文件和RAM设计的理解。