

存储器设计实验

实验整体框架

MIPS寄存器文件设计

4位可级联先行进位电路设计

□ 4位快速加法器设计

□ 16位快速加法器

□ 32位快速加法器

MIPS 的RAM设计

5位阵列乘法器

□ 5位乘法流水线

算数逻辑运算单元



打开资源包中的“storage.circ”，可以看到已经建立的电路（白色星为待编辑，黑色星为已完成待调用）

一、MIPS寄存器文件设计

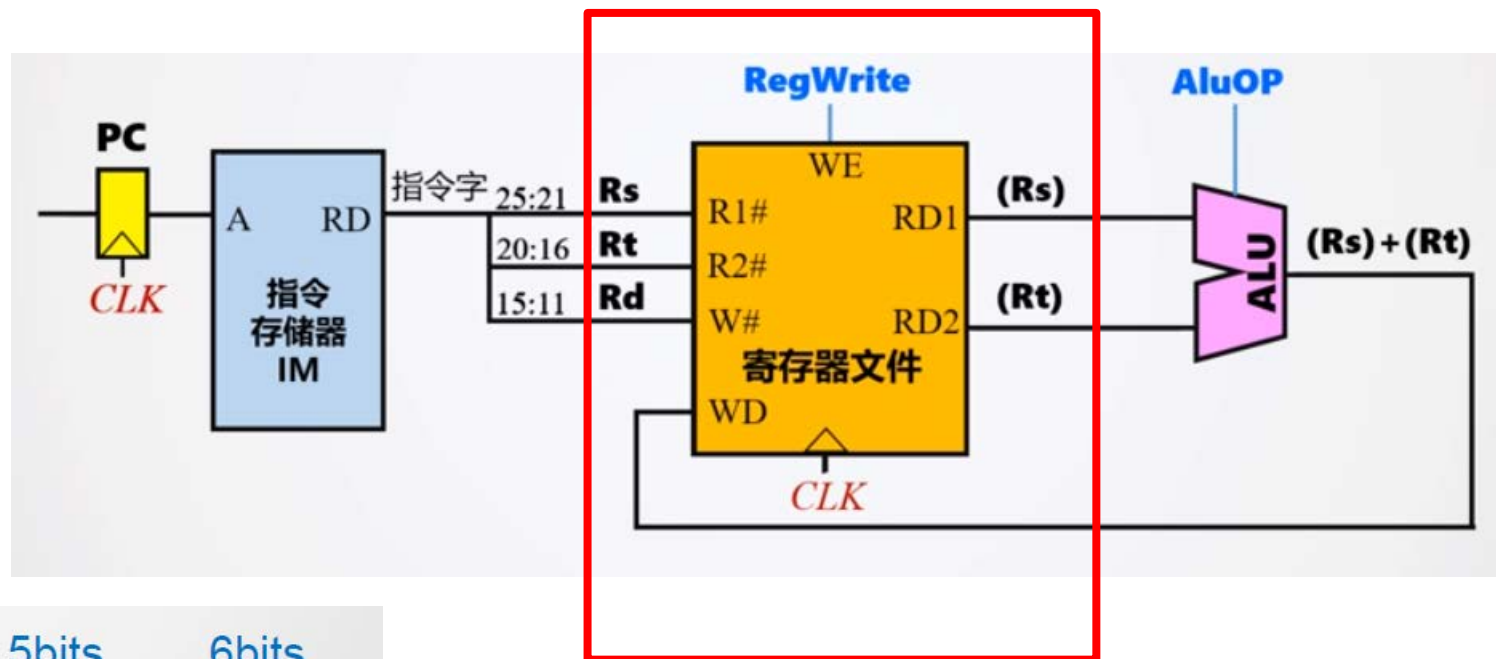
基础任务

实验目标

- 熟悉多路选择器、译码器、解复用器
- 在Logisim中实现MIPS寄存器组（寄存器文件）

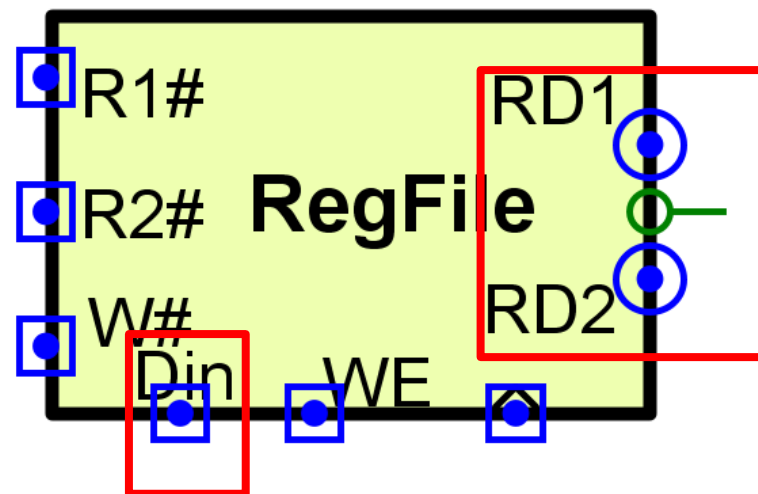
MIPS中典型的R型指令

```
add $s0,$s1,$s2
```



MIPS寄存器组的外部特性

- 包含32个寄存器
- 1个写入端口，2个读端口

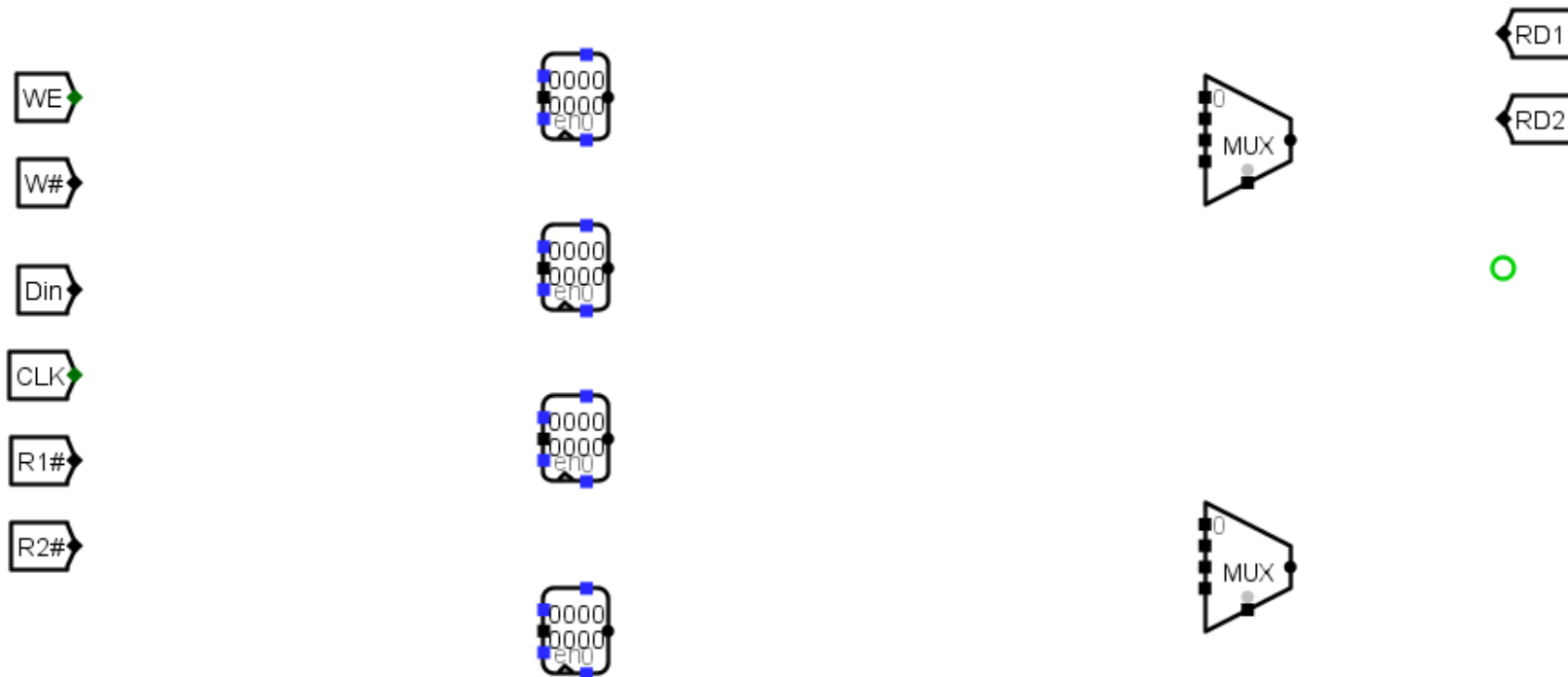
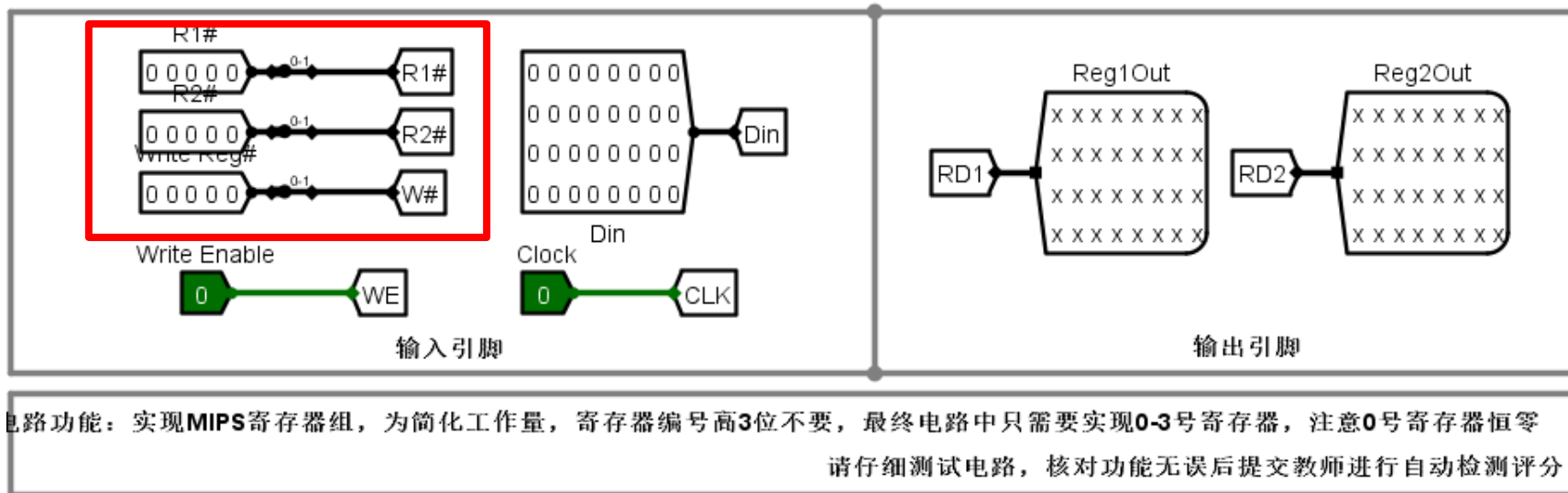


引脚	输入/输出	位宽	功能描述
R1#	输入	5	第1个读寄存器的编号
R2#	输入	5	第2个读寄存器的编号
W#	输入	5	写入寄存器编号
Din	输入	32	写入数据
WE	输入	1	写使能信号，为1时在CLK上跳沿将Din数据写入W#寄存器
CLK	输入	1	时钟信号，上跳沿有效
RD1	输出	32	R1#寄存器的值， 0号寄存器值恒零
RD2	输出	32	R2#寄存器的值， 0号寄存器值恒零

设计方法

- 简化设计，只需要实现寄存器组中4个寄存器
- 根据R1#、R2#、W#来控制读出、写入的对象
- 0号寄存器恒0，设计时如何变通？

运用多路选择器、译码器、解复用器等完成寄存器组的内部逻辑



打开资源包中的“storage.circ”，进行选择，可以进入电路编辑

MIPS寄存器组的自动测试

测试逻辑伪代码:

For (i=0;i<N;i++)

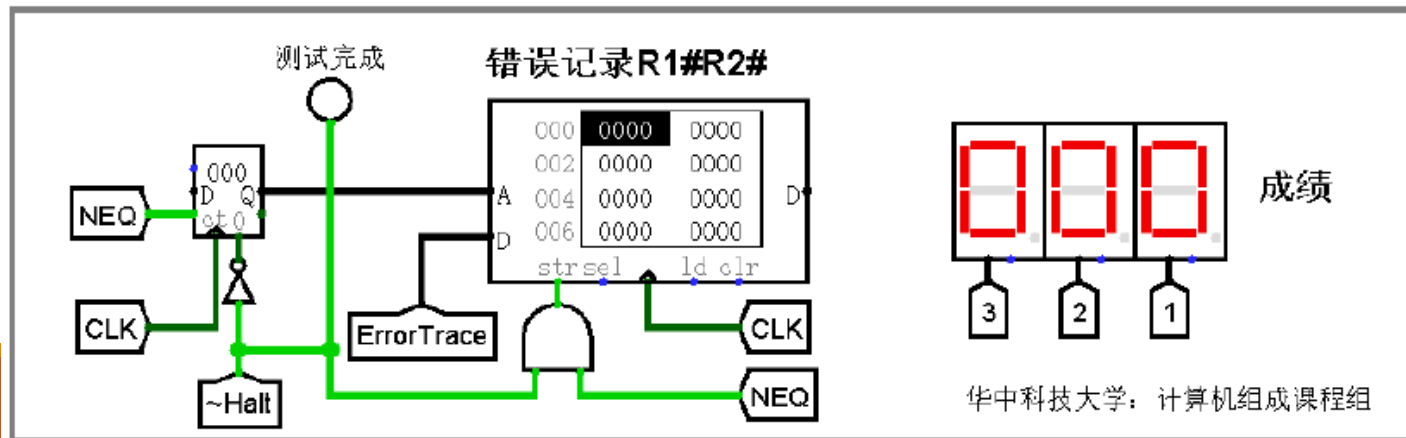
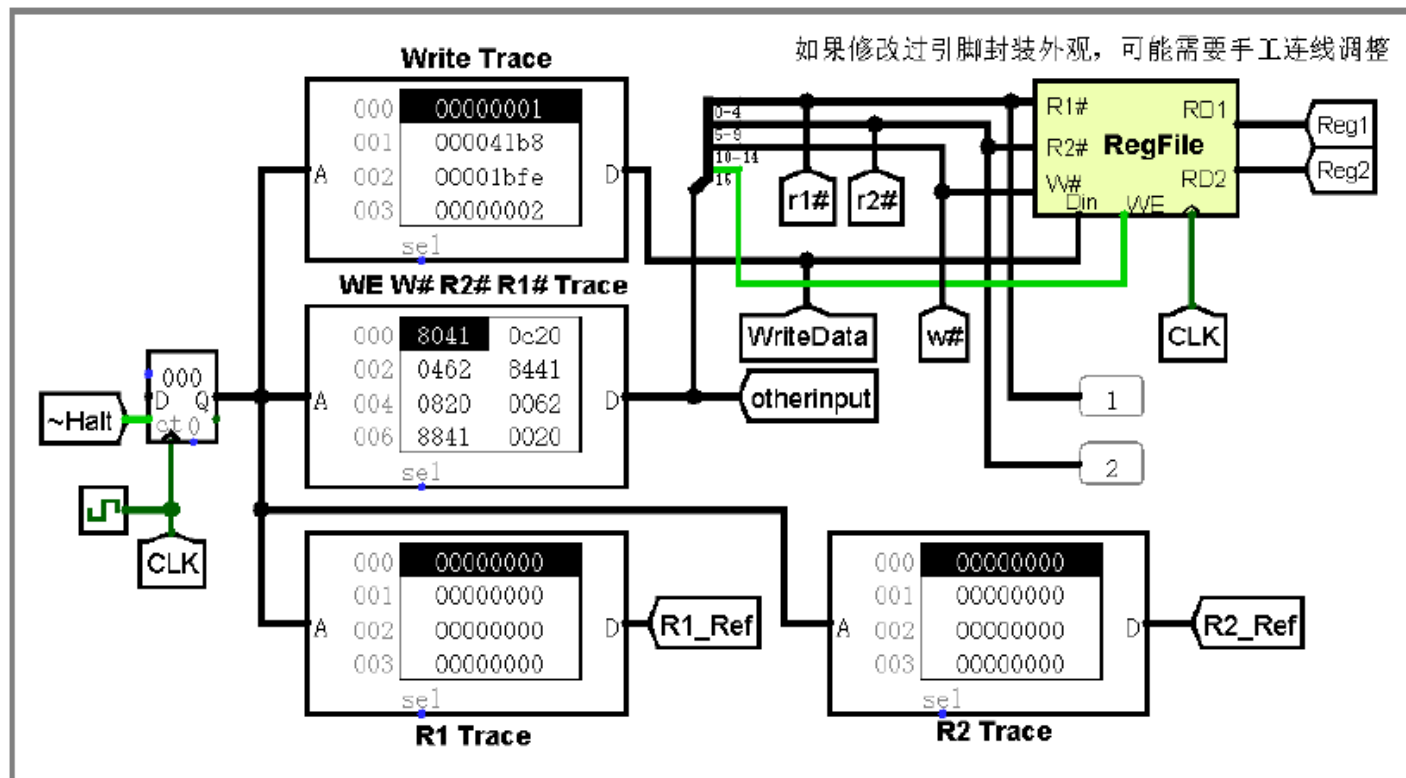
{

 写寄存器, 使 $\text{reg}[i] = i+1$;

 读出所有寄存器的值与标准值进行比较, 若发生异常则记录错误寄存器编号;

}

常见问题: 写入某个寄存器, 影响了其他寄存器的值



二、MIPS 的RAM设计

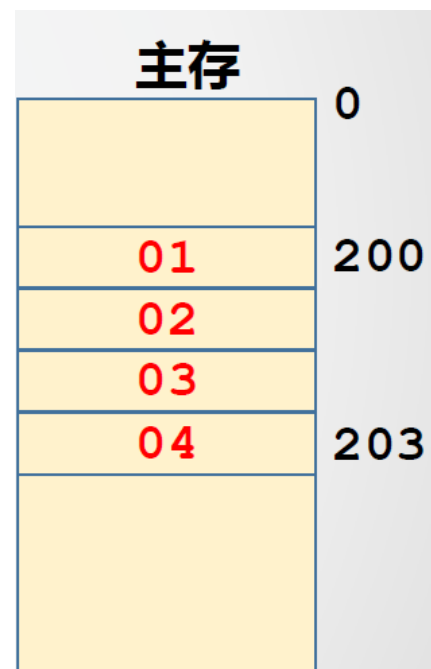
实验目标

- 理解主存地址的基本概念
- 掌握存储系统位扩充的基本思想
- 在Logisim中实现MIPS的RAM

MIPS RAM的特点:

- 同时支持字节、半字、字访问的存储子系统
- 数据位宽可变

```
mov ah, [200]    #按字节访存 ah=0x01
mov ax, [200]    #按半字访存 ax=0x0201
mov eax, [200]   #按字访存  eax=0x04030201
mov ax, [201]    #未对齐
mov eax, [201]   #硬件强制对齐到200
```

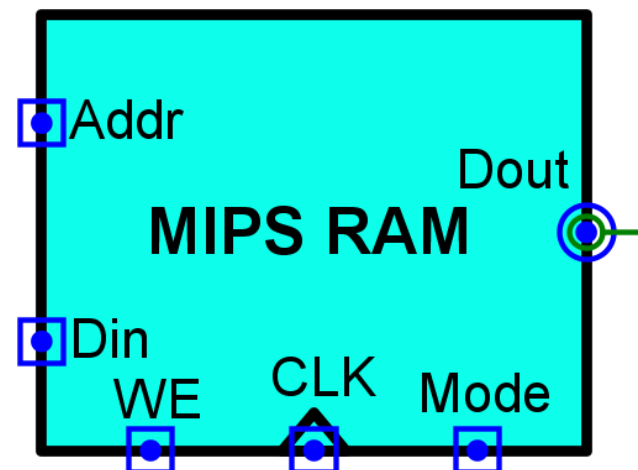


提示：硬件强制对齐地址，可以靠截断地址实现

MIPS RAM的外部特性

- 存储容量： 4KB (1K * 4B)
- 能支持MIPS中 **Load/Store**指令的存储器

□ lb lh lw sb sh sw



引脚	输入/输出	位宽	功能描述
Addr	输入	12	字节地址（字访问半字访问时应硬件强制对齐）
Din	输入	32	写入数据，不同访问模式有效数据均存放在最低位，高位忽略
WE	输入	1	写使能，高电平有效
CLK	输入	1	时钟信号，上跳沿有效
Mode	输入	2	访问模式 00：字访问，01：字节访问，10：2字节访问
Dout	输出	32	输出数据，不同访问模式有效数据均存放在最低位，高位忽略

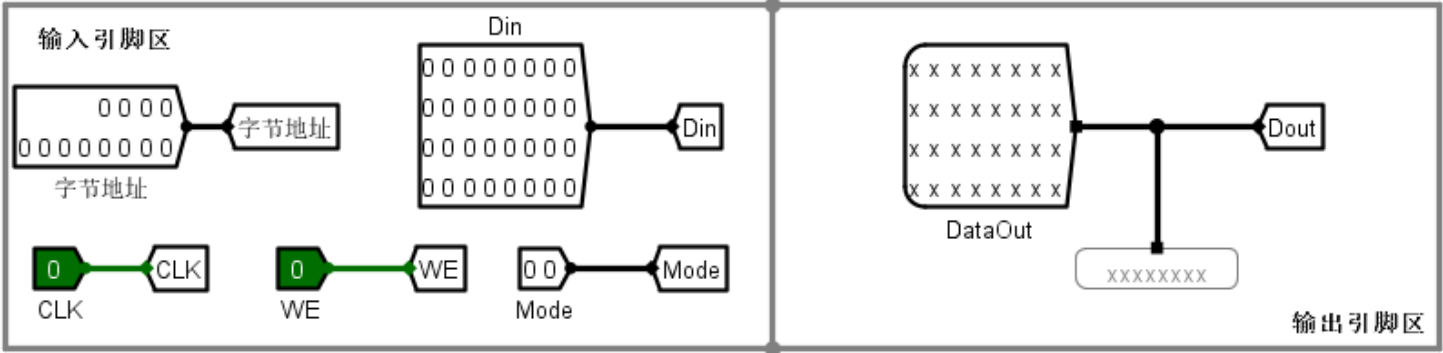
MIPS RAM的电路设计

- MIPS RAM内部，由4片数据宽度为1KB的RAM通过位扩充构成
- 4片RAM依次提供了 Byte3、Byte2、Byte1、Byte0

1024*8 bit的RAM



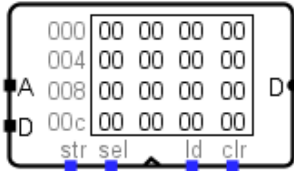
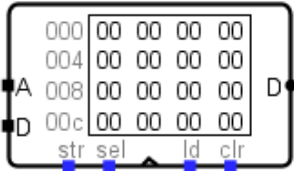
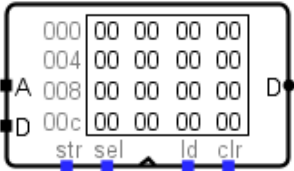
10位地址A; 8位数据D
写使能Str; 读使能Id;
片选信号sel;



电路功能：利用下图中给定的RAM器件实现既既可以按照字节，也可以按照半字，也可以按照32位字访问的MIPS存储器
请勿增减引脚，请勿修改子电路封装 请在下方利用上方输入输出引脚的隧道信号构建电路，ctrl+d复制选择部件

外部输入地址共12位

- ◆ 高10位：选中某一个字（32位）
- ◆ 低2位：区分不同字节（8位）



重点在于访问模式控制，
输入/输出的位宽变化，地
址的使用

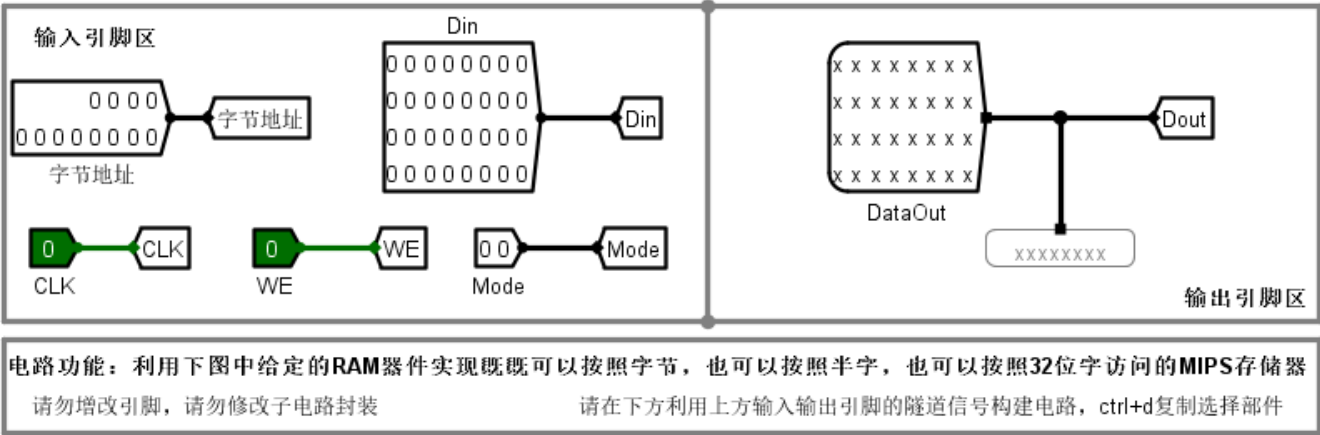
不同访问方式下输入输出的总线控制

有效数据总是出现在数据总线的低位

- **Mode** 决定了数据总线上**有效数据的位宽**（字：32位， 半字：低16位， 字节：低8位）
- **低位地址** 决定了数据总线上**有效数据的构成**（与哪个字节/存储体有关？）

Mode	Din组成				Dout组成					
字访问	字写入	Byte3	Byte2	Byte1	Byte0	字读出	Byte3	Byte2	Byte1	Byte0
半字访问	高半字写入			Byte3	Byte2	高半字读出			Byte3	Byte2
	低半字写入			Byte1	Byte0	低半字读出			Byte1	Byte0
字节访问	最高字节写入				Byte3	最高字节读出				Byte3
	次高字节写入				Byte2	次高字节读出				Byte2
	次低字节写入				Byte1	次低字节读出				Byte1
	最低字节写入				Byte0	最低字节读出				Byte0

MIPS RAM的设计要点



1、外部输入地址（字节地址）如何使用？

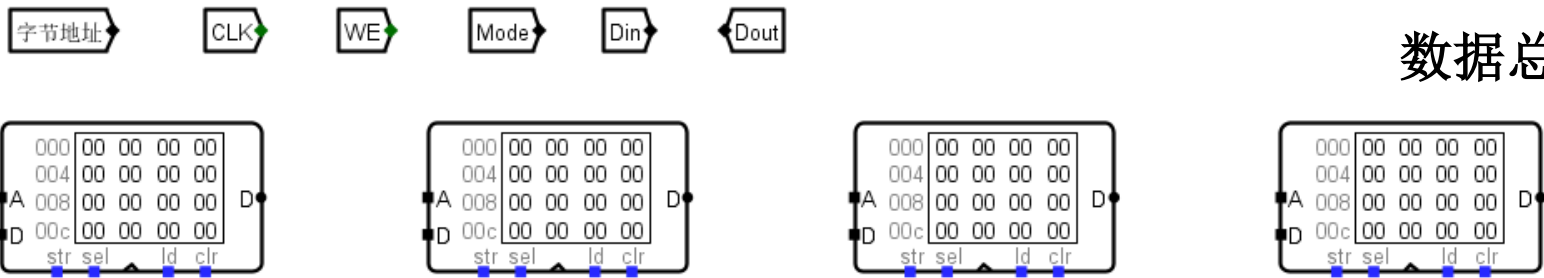
A11-A2 字地址；
A11-A1 半字地址，A1表示高/低半字；
A11-A0 字节地址，A1 A0 表示不同字节

2、访问模式（Mode）如何起作用？

3、他们与4个存储体的片选信号有何关系？

例如：Sel3 在字访问、高半字访问、最高字节访问时均需有效

4、不同模式、不同地址下，输入/输出数据总线的构成



半字、字节访问时，输出数据总线的无关高位需设为0，否则会影响后续自动测试验证。

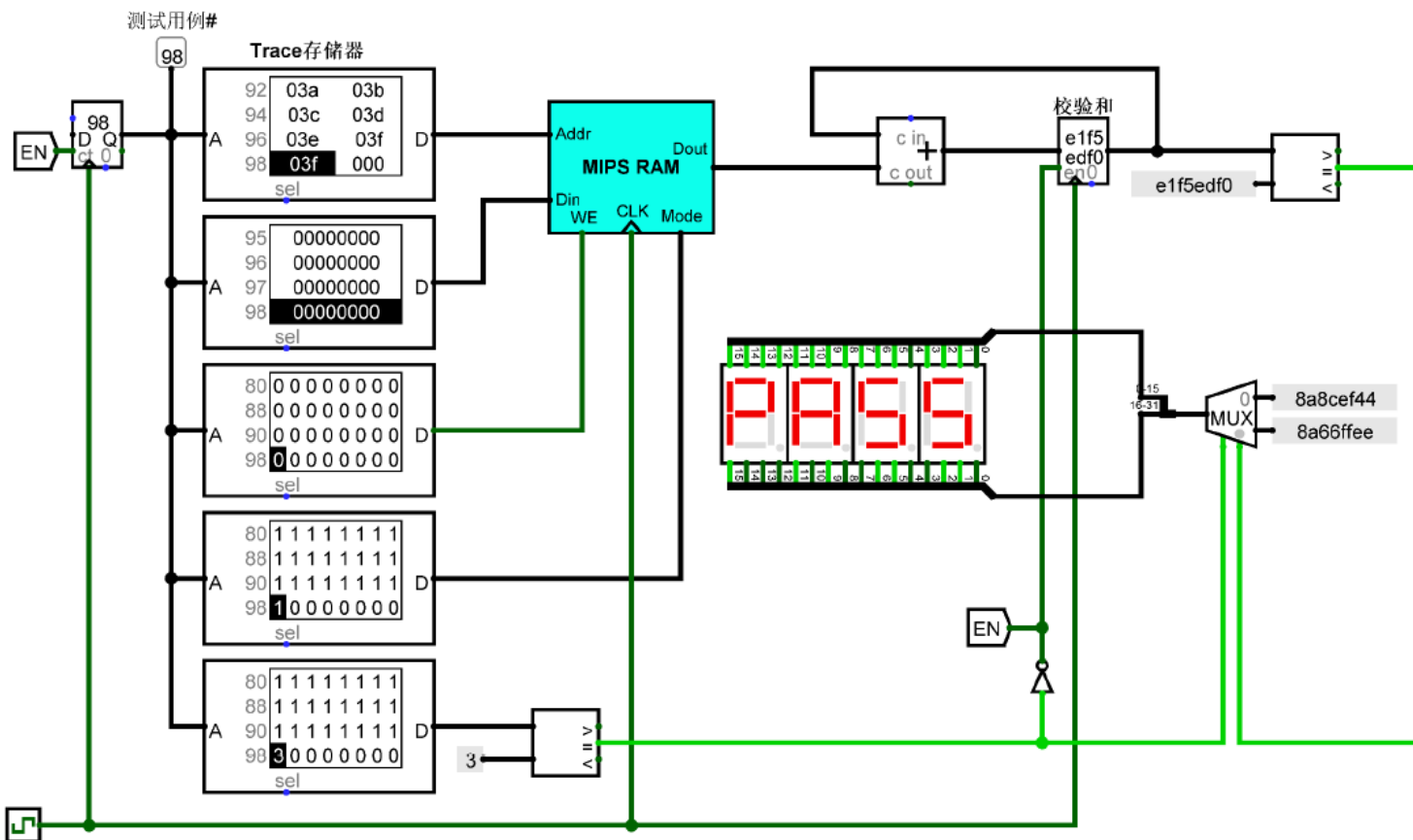
MIPS RAM的测试

1、手动测试

- 对连续空间，依次进行字写入、半字写入、字节写入
- 测试非对齐写入、重复覆盖写入
- 对连续空间进行字读出、半字读出、字节读出

2、自动测试

依次执行设计好的操作序列，完成读写动作。读出的数据进行累加求和，与标准值对比



Ctrl + K 连续自动测试, Ctrl+T单步自动测试

实验报告要求——存储器设计

- 1、撰写实验报告。报告需将2个实验子任务的分析、设计思路与过程清晰描述，并相应贴出测试截图。
- 2、实验完成后，将最终的“Storage.circ”文件以“学号_存储器.circ”命名另存。
- 3、在福大课程中心平台中，本门课程的“作业”中，相应提交实验报告与电路文件。