



廈門大學

《计算机组成原理》 课程实验报告

姓名： 苏一涵

学院：信息学院

系：软件工程

专业：软件工程

学号：36720232204041

2025 年 4 月 23 日

第 5 次实验 CPU 设计 (I)

1. 实验目的

- (1) 掌握单总线结构 MIPS 处理器（硬布线控制器、微程序控制器）、单周期 MIPS 处理器（硬布线控制器）、单周期 RISC-V 处理器（硬布线控制器）设计的基本原理。
- (2) 对已有的 5 条指令单总线结构 MIPS 处理器（微程序控制器）进行修改，增加 1 条 add 指令（共 6 条指令）。
- (3) 鼓励同学们对已有的 5 条指令单总线结构 MIPS 处理器（硬布线控制器）进行修改，增加 1 条 add 指令（共 6 条指令）。

2. 实验环境

- (1) Windows 系统下运行 Logisim 软件（需安装 JDK）。
- (2) MARS 4.5、RARS 1.5 汇编工具。

3. 实验内容

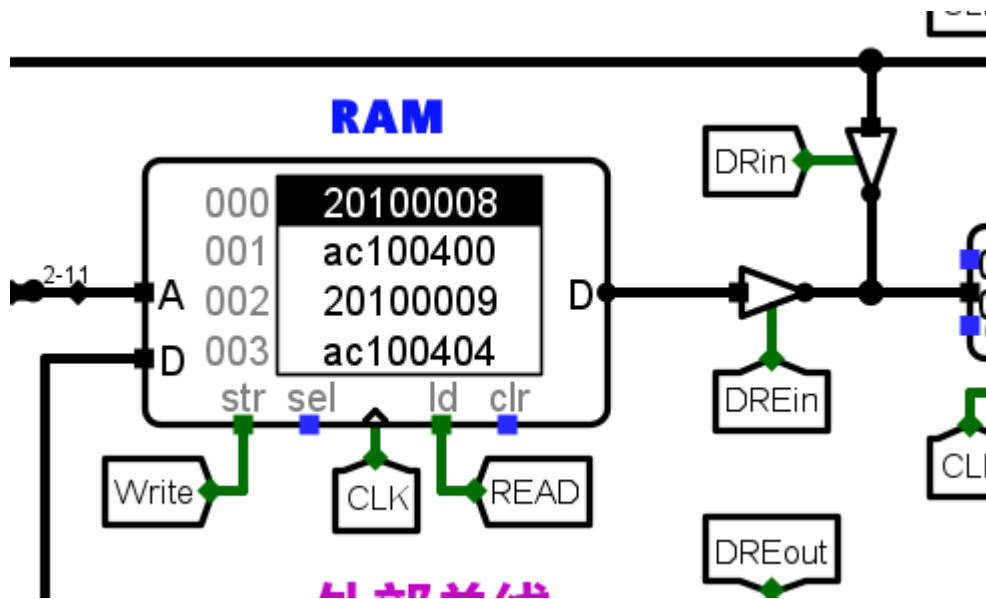
3.1 验证实验

- (1) 单总线结构 MIPS 处理器（硬布线控制器）（5 条指令）

第一步：先打开 mars 仿真，测试 test1.asm.

0x00000008
0x00000009
0x00000001

第二步：在单总线结构 MIPS 处理器（硬布线控制器）的数据通路上运行测试程序（test1.hex）



运行程序，结果如下

```
0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
100 00000008 00000009 00000001 00000000 00000000 00000000 00000000
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

第三步：运行 5 个数的排序程序 sort3_mips_bus.hex 和 sort4_mips_bus.hex

Sort3

```
0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
100 00000005 00000004 00000003 00000002 00000001 00000000 00000000
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000
-----
```

Sort4

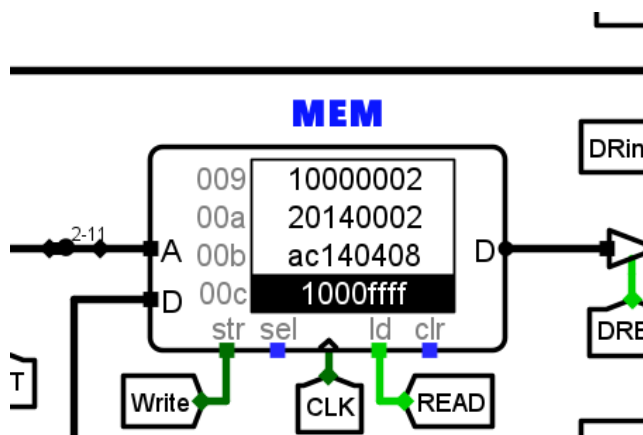
```
0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00
100 00000001 00000002 00000003 00000004 00000005 00000000 00000000 00
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00
-----
```

(2) 单总线结构 MIPS 处理器（微程序控制器）（5 条指令）

在单总线结构 MIPS 处理器（微程序控制器）的数据通路上运行程序：test1.hex、sort3_mips_bus.hex、sort4_mips_bus.hex

步骤与（1）相同

Test1



```

0e0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
100 00000008 00000009 00000001 00000000 00000000 00000000 00000000 00000000
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

Sort3

```

工程 电路仿真 窗口 帮助
000 20100003 ac100400 20100001 ac100404 20100005 ac100408 20100002 ac10040c 20100004 ac100410 20100400 20110410 8e130000 8e340000 0274402a 11000002
010 ae330000 ae140000 2231fff 12110001 1000fff7 22100004 20110410 12110001 1000fff3 1000fff 00000000 00000000 00000000 00000000 00000000 00000000

0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000
100 00000005 00000004 00000003 00000002 00000001 00000000 00000000 00000000 000
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000

```

Sort4

```

工程 电路仿真 窗口 帮助
000 20100003 ac100400 20100001 ac100404 20100005 ac100408 20100002 ac10040c 20100004 ac100410 20100400 20110410 8e130000 8e340000 0293402a 11000002
010 ae330000 ae140000 2231fff 12110001 1000fff7 22100004 20110410 12110001 1000fff3 1000fff 00000000 00000000 00000000 00000000 00000000 00000000

0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000
100 00000001 00000002 00000003 00000004 00000005 00000000 00000000 00000000 000

```

(3) 单周期 MIPS 处理器（硬布线控制器）(24 条指令)

测试程序：包括求累加和程序、计算费波那契数列程序、排序程序（降序、升序）

求累加和程序

```

工程 电路仿真 窗口 帮助
000 2010000a 20110001 20120001 20130000 02719820 12300002 02328820 08000c04 ac130000 2002000a 0000000c 00000000 000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000

```

```

000 00000037 00000000 00000000
010 00000000 00000000 00000000
020 00000000 00000000 00000000

```

计算费波那契数列程序

```

000 2012000a 20110000 ac110000 20110001 ac110004 ac110008 20130002 20140001 20150001 0295b020 22b40000 22d50000 22730001 0273b820 02f78020 ae160000
010 12530001 08000c09 2002000a 0000000c 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

编辑 工程 电路仿真 窗口 帮助
000 00000000 00000001 00000001 00000002 00000003 00000005 00000008 0000000d 00000015 00000022 00000037 000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

排序程序（降序、升序）

降序：

```

编辑 工程 电路仿真 窗口 帮助
000 20100003 ac100000 20100001 ac100004 20100005 ac100008 20100002 ac10000c 20100004 ac100010 20100000 20110010 8e130000 8e340000 0274402a 11000002
010 ae330000 ae140000 2231fff 12110001 1000fff7 22100004 20110010 12110001 1000fff3 1000fff 00000000 00000000 00000000 00000000 00000000 00000000
020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

000 00000005 00000004 00000003 00000002 00000001 00000000 00000000 00000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

升序：

```

000 20100003 ac100000 20100001 ac100004 20100005 ac100008 20100002 ac10000c 20100004 ac100010 20100000 20110010 8e130000 8e340000 0293402a 11000002
010 ae330000 ae140000 2231fff 12110001 1000fff7 22100004 20110010 12110001 1000fff3 1000fff 00000000 00000000 00000000 00000000 00000000 00000000

000 00000001 00000002 00000003 00000004 00000005 00000000 00000000 00000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

（4）单周期 RISC-V 处理器（硬布线控制器）（9 条指令）

测试程序：包括求累加和程序、计算费波那契数列程序、排序程序（降序、升序），步骤同（3）

求累加和程序

```

编辑 工程 电路仿真 窗口 帮助
000 00a06693 00106793 00106613 00006713 00f70733 00d78663 00c787b3 f5fff06f 00e02023 0000006f 00000000 00000000 00000000 00000000 00000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

000000 00000037 00000000 00000000
000010 00000000 00000000 00000000 00000000

```

计算费波那契数列程序

工程 电路仿真 窗口 帮助

```

000 00a06613 00006593 00b02023 00106593 00b02223 00b02423 00206693 00106713 00106793 0070833 0007e713 00086793 00b686b3 00d688b3 01188533 01052023
010 00d60463 fe1ff06f 0000006f 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
-----
000000 00000000 00000001 00000001 00000002 00000003 00000005 00000008 0000000d 00000015 00000022 00000037 00000000 00000000 00000000
-----

```

排序程序（降序、升序）

降序：

```

000 00506613 00306593 00b02023 00106593 00b02223 00506593 00b02423 00206593 00b02623 00406593 00b02823 00106713 00406793 ff06813 04e60663 00106693
010 00006893 00406c13 00c6bdb3 020d8063 0008ac83 000c2d03 019d3db3 00ed8c63 01a8a023 019c2023 00c0006f 01060633 fc9ff06f 00e686b3 00f888b3 00fc0c33
020 fc9ff06f 0000006f 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
-----
000000 00000005 00000004 00000003 00000002 00000001 000
-----

```

升序：

```

000 00506613 00306593 00b02023 00106593 00b02223 00506593 00b02423 00206593 00b02623 00406593 00b02823 00106713 00406793 ff06813 04e60663 00106693
010 00006893 00406c13 00c6bdb3 020d8063 0008ac83 000c2d03 019d3db3 00ed8c63 01a8a023 019c2023 00c0006f 01060633 fc9ff06f 00e686b3 00f888b3 00fc0c33
020 fc9ff06f 0000006f 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
-----
000000 00000001 00000002 00000003 00000004 00000005 0000
-----

```

除完成实验课件要求内容外（实验环节基本要求），需回答下述问题（实验报告采分点）：

（1）针对 test1.asm 程序段

```

loop1:
    addi $s4,$zero,2      #s4=2
    sw $s4,1032($zero)    #s4=2 保存到(1032, 即102单元)

```

这条指令不会执行到
这条指令不会执行到

为何上述两条指令不会执行到？

答：因为 $s1=8$, $s2=9$, $s1 \neq s2$, 故跳过跳转到 loop1 的指令，然后直接进入 loop2 死循环了，所以 loop1 下面的两条指令不会被执行

```

main:
    addi $s0,$zero,8          #s0=8
    sw $s0,1024($zero)        #s0=8 保存到(1024, 即100单元)

    addi $s0,$zero,9          #s0=9
    sw $s0,1028($zero)        #s0=9 保存到(1028, 即101单元)

    lw $s1,1024($zero)         #$s1=8
    lw $s2,1028($zero)         #$s2=9

    slt $s3,$s1,$s2           # s1 < s2    s3=1

    beq $s1,$s2,loop1         #s1不等于s2  执行下一条指令
    sw $s3,1032($zero)         #s3=1 保存到(1032, 即102单元)
    beq $zero,$zero,loop2      #转loop2

loop1:
    addi $s4,$zero,2           #s4=2
    sw $s4,1032($zero)         #s4=2 保存到(1032, 即102单元)    这条指令不会执行到
                                                                    这条指令不会执行到

loop2:
    beq $zero,$zero,loop2      #转loop2    死循环

```

(2) 为何安排死循环？若不采用死循环，针对此种情况，MIPS 架构还可以采用什么方式？（课外扩展）

```

loop2:
    beq $zero,$zero,loop2      #转loop2    死循环

```

答：安排死循环 loop2 的目的是：防止程序执行完测试指令后继续运行后续可能改变测试结果的代码，确保 RAM 存储器中 100、101、102 单元的测试结果（8、9、1）稳定保持，避免因程序继续执行导致结果被意外修改。

若不采用死循环，在 MIPS 架构中可通过系统调用来实现程序正常终止，避免干扰测试结果。具体方式为：在程序最后添加代码 `li $v0, 10` 调用终止程序，再执行 `syscall`（触发系统调用）。这样程序会主动结束运行，维持测试结果的稳定性，无需通过死循环阻塞后续执行。

(3) 在该单总线结构 MIPS 处理器（硬布线控制器）的数据通路上，能够运行求累加和程序、计算费波那契数列程序吗？为什么？

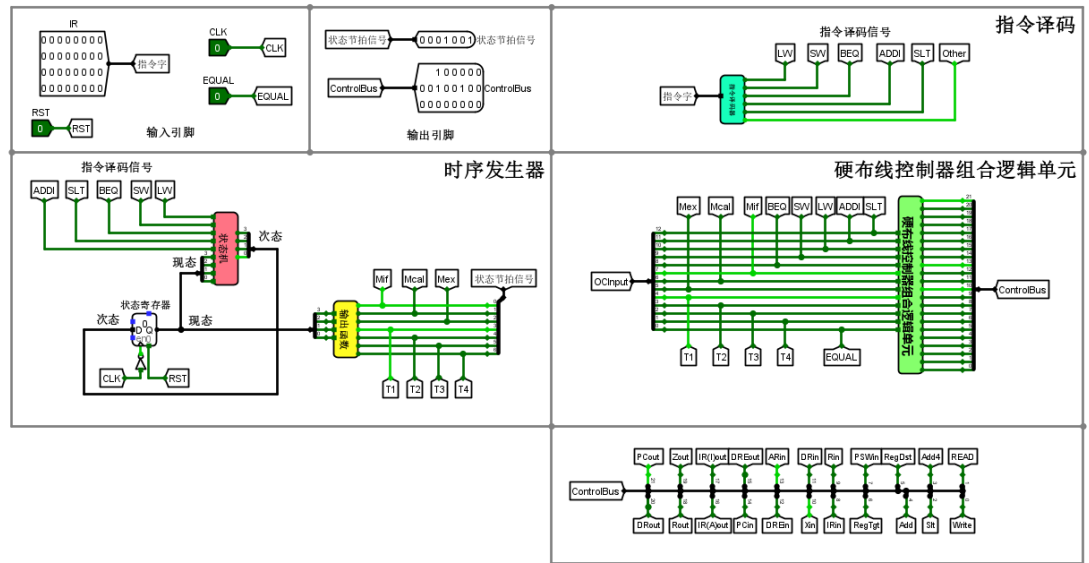
答：不能运行累加和程序、计算费波那契数列程序，因为该电路只支持 5 条指令，缺少 ADD 指令。

(4) 针对单总线结构 MIPS 处理器（硬布线控制器）的下述电路进行分析。

a) 硬布线控制器

答：

硬布线控制器

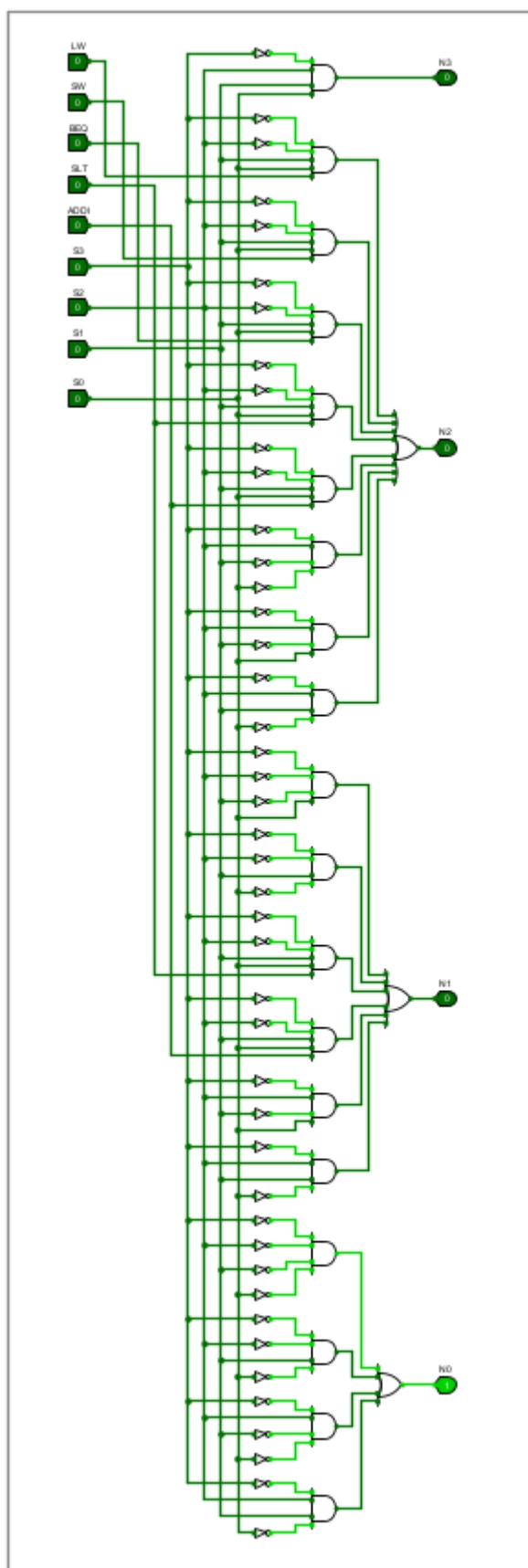


是单总线结构 MIPS 处理器的主要部分之一，包含指令译码器，状态机，输出函数，硬布线控制器组合逻辑单元，算术逻辑单元 ALU 是通过组合逻辑电路直接产生控制信号的部件。根据指令译码结果、状态信号等，利用组合逻辑生成控制寄存器读写、ALU 运算、总线数据传输等操作的信号。

b) 状态机

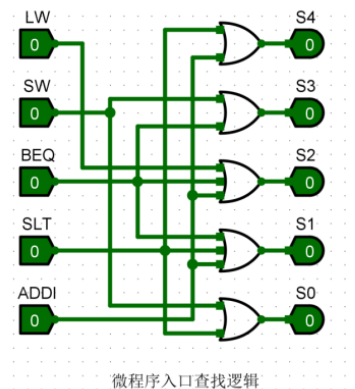
答：用于根据指令的类型和执行状态生成后续地址的多路选择信号。作为硬布线控制器的核心，状态机通过定义一系列状态（如取指、译码、执行等）及状态转换条件，控制指令执行流程。每个状态对应特定操作，输出相应控制信号，确保各部件在单总线上按序工作。

状态机



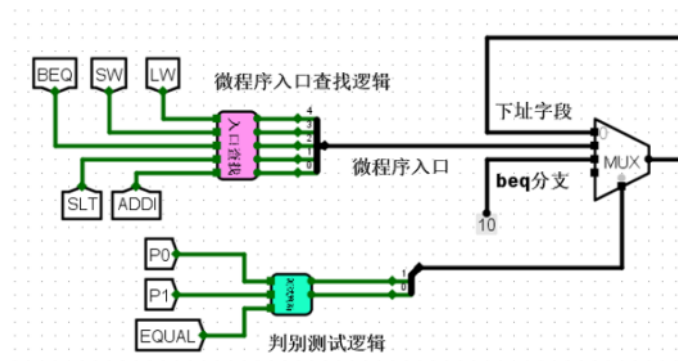
(5) 分析单总线结构 MIPS 处理器（微程序控制器）的下述电路原理

a) 微程序入口查找逻辑



答：微程序入口查找逻辑根据不同指令生成对应的微程序入口地址。图中输入为 LW、SW、BEQ、SLT、ADDI 等指令信号，通过或门等逻辑电路组合生成 S4~S0 信号。每个指令对应唯一的 S4~S0 组合，以此精准定位该指令的微程序入口。例如，当 LW 指令有效时，逻辑门组合使 S4~S0 产生特定值，指向 LW 指令的微程序起始位置，实现指令到微程序入口的映射。

b) 判别测试逻辑



答：判别测试逻辑用于确定微程序的下一条执行地址，支持流程控制。图中指令信号（如 BEQ、SW 等）参与入口逻辑，同时 P0、P1 和 EQUAL 等信号输入。以 BEQ 指令为例，当 EQUAL 有效（表示相等），判别测试逻辑控制多路选择器（MUX）选择特定下址字段，实现分支转移。

(6) 结合工作原理，比较下述微程序控制器和硬布线控制器引脚上的不同



答：微程序控制器：

有 **RST** 引脚，用于复位，初始化微程序执行起点，确保控制器从初始状态开始工作。

输出 **微地址 / 状态**，因微程序控制器需通过微地址访问控制存储器中的微指令，同时可能输出当前状态信息。

硬布线控制器：

无 **RST** 引脚（或依赖其他复位方式），其逻辑直接由组合电路实现，复位机制可能集成于整体系统而非单独引脚。

输出 **状态节拍信号**，硬布线控制器通过组合逻辑生成控制信号，需状态节拍信号同步各操作步骤，确保控制信号按正确时序发出。

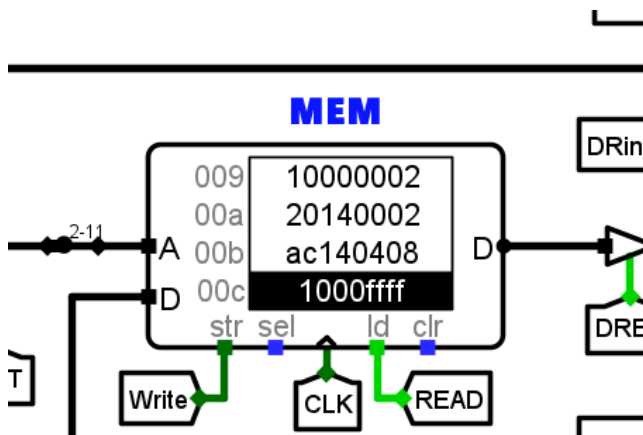
(7) 请在单总线结构 MIPS 处理器（微程序控制器）的数据通路上运行程序：

test1.hex、 sort3_mips_bus.hex、 sort4_mips_bus.hex

观看 RAM 存储器中的结果是否正确？

答：

Test1



```
0e0 00000000 00000000 00000000 00000000 0
0f0 00000000 00000000 00000000 00000000 0
100 00000008 00000009 00000001 00000000 0
110 00000000 00000000 00000000 00000000 0
```

Sort3

```
000 20100003 ac100400 20100001 ac100404 20100005 ac100408 20100002 ac10040c 20100004 ac100410 20100400 20110410 8e130000 8e340000 0274402a 11000002
010 ae330000 ae140000 2231fff0 12110001 1000fff7 22100004 20110410 12110001 1000fff3 1000fff7 00000000 00000000 00000000 00000000 00000000
```

```

0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000
100 00000005 00000004 00000003 00000002 00000001 00000000 00000000 000
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000
-----

```

Sort4

```

000 20100003 ac100400 20100001 ac100404 20100005 ac100408 20100002 ac10040c 20100004 ac100410 20100400 20110410 8e130000 8e340000 0293402a 11000002
010 ae330000 ae140000 2231fff 12110001 1000fff 22100004 20110410 12110001 1000fff 1000fff 00000000 00000000 00000000 00000000 00000000 00000000

```

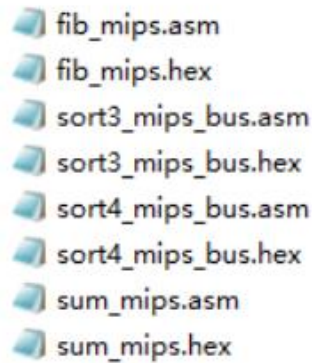
```

0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
100 00000001 00000002 00000003 00000004 00000005 00000006 00000007

```

结果正确。

(8) 针对单周期 MIPS 处理器，测试程序，观察结果（拷屏+文字解释）



答：求累加和程序

```

编辑 工程 电路仿真 窗口 帮助
000 2010000a 20110001 20120001 20130000 02719820 12300002 02328820 08000c04 ac130000 2002000a 0000000c 00000000 000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000

```

```

000 00000037 00000000 00000000
010 00000000 00000000 00000000
020 00000000 00000000 00000000

```

计算费波那契数列程序

```

000 2012000a 20110000 ac110000 20110001 ac110004 ac110008 20130002 20140001 20150001 0295b020 22b40000 22d50000 22730001 0273b820 02f78020 ae160000
010 12530001 08000c09 2002000a 0000000c 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

```

编辑 工程 电路仿真 窗口 帮助
000 00000000 00000001 00000001 00000002 00000003 00000005 00000008 0000000d 00000015 00000022 00000037 000000
010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

排序程序（降序、升序）

降序:

编辑 工程 电路仿真 窗口 帮助															
000	20100003	ac100000	20100001	ac100004	20100005	ac100008	20100002	ac10000c	20100004	ac100010	20100000	20110010	8e130000	8e340000	0274402a 11000002
010	ae330000	ae140000	2231fff	12110001	1000fff7	22100004	20110010	12110001	1000fff3	1000fff	00000000	00000000	00000000	00000000	00000000 00000000
020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000 00000000
000	00000005	00000004	00000003	00000002	00000001	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
010	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

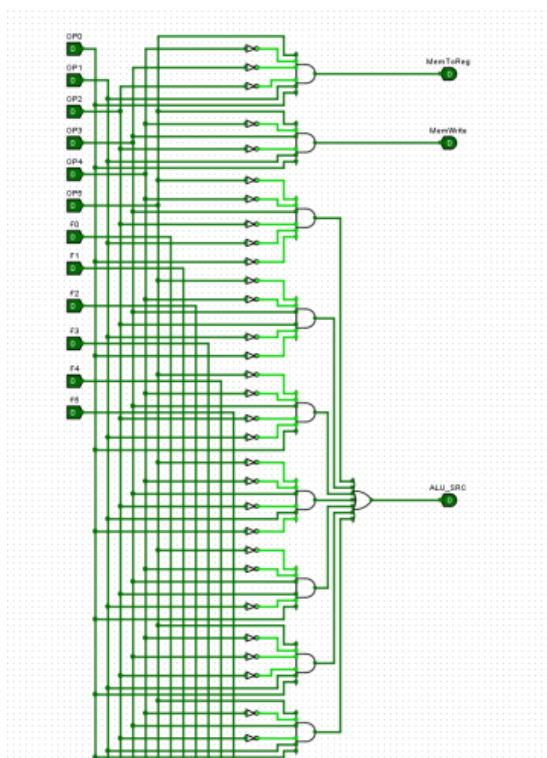
升序:

000	20100003	ac100000	20100001	ac100004	20100005	ac100008	20100002	ac10000c	20100004	ac100010	20100000	20110010	8e130000	8e340000	0293402a 11000002
010	ae330000	ae140000	2231fff	12110001	1000fff7	22100004	20110010	12110001	1000fff3	1000fff	00000000	00000000	00000000	00000000	00000000 00000000
000	00000001	00000002	00000003	00000004	00000005	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
010	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

(9) 该单周期 MIPS 处理器的控制器能不能采用微程序控制器的方法设计? 为什么?

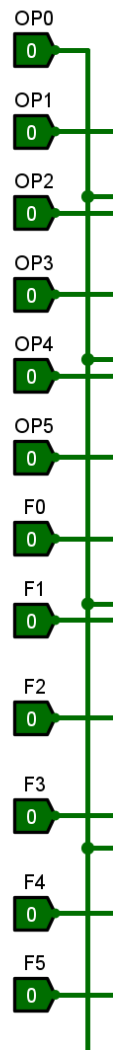
答: 可以, 微程序控制器是一种用于控制计算机硬件操作的方法, 它通过将指令执行划分为一系列微操作, 并使用存储器中的微指令序列来实现控制。对于支持的指令数, 微程序控制器能够提供更高的控制精度和可编程性。

(10) 分析单周期 MIPS 处理器控制信号的生成电路原理(输入输出的逻辑, 拷屏+文字解释)

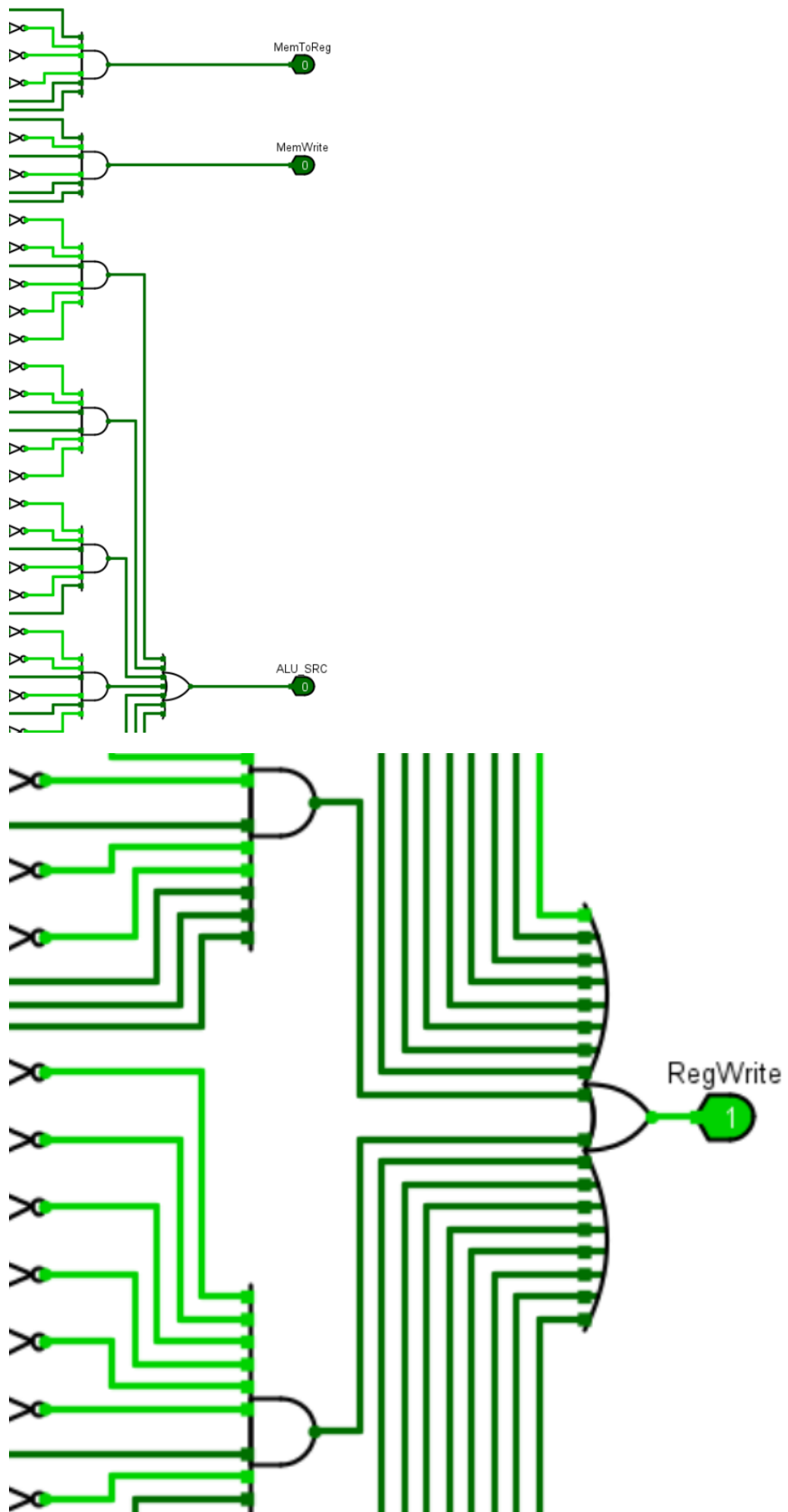


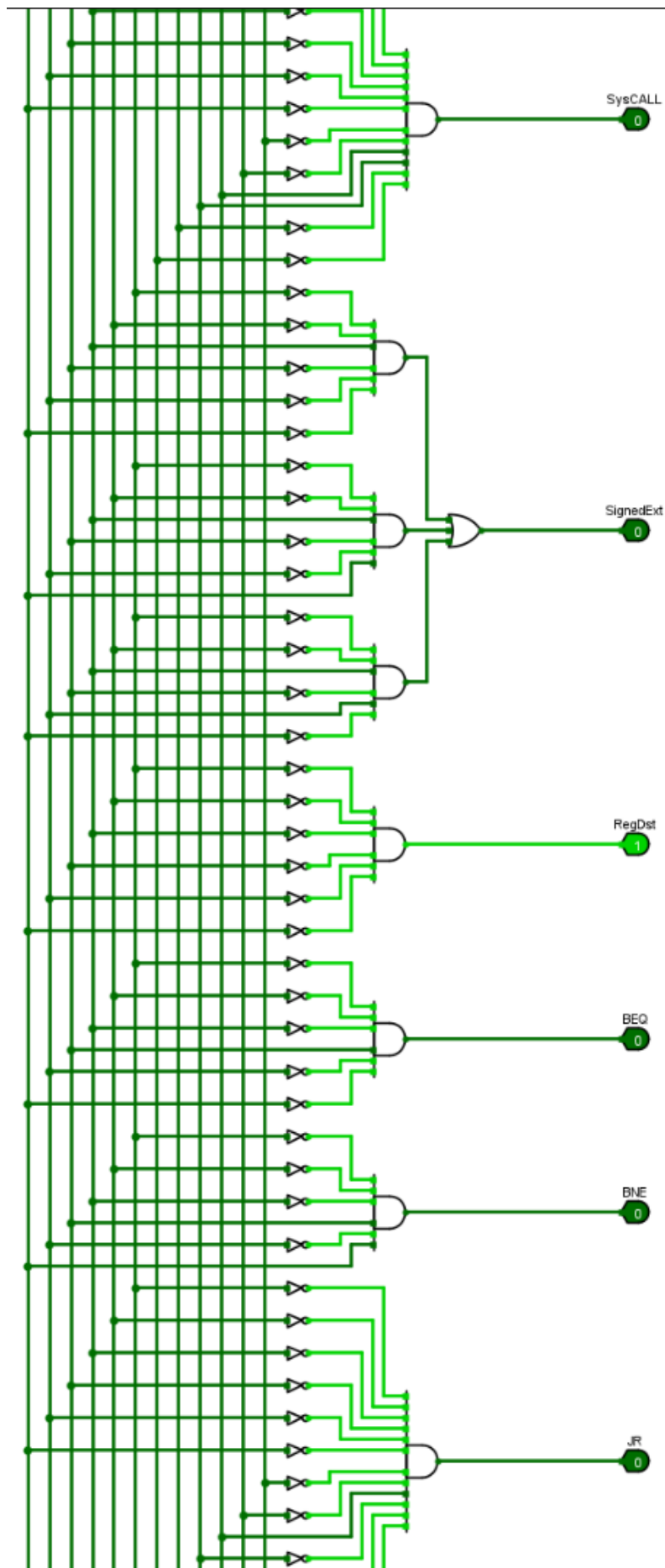
答：单周期 MIPS 处理器控制信号生成电路通过组合逻辑对输入的指令操作码（OP0 - OP5）和功能码（F0 - F5）进行译码，生成相应控制信号，协调处理器各部件工作。以下是关键输入输出逻辑分析：

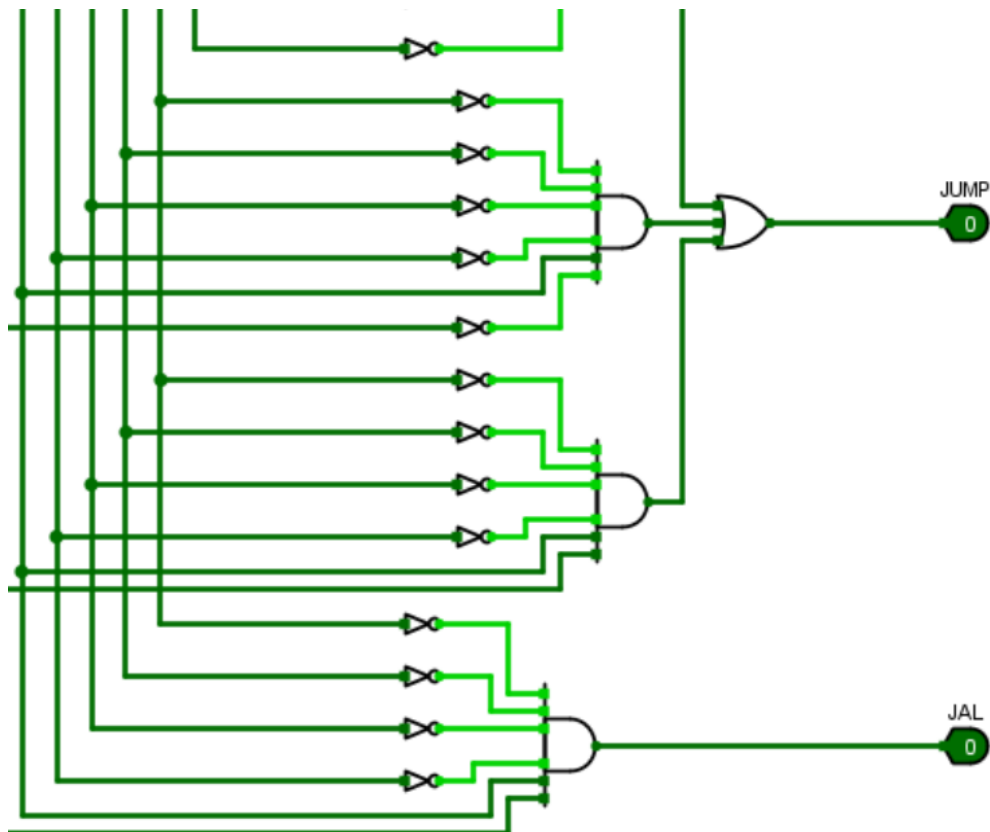
输入：主要为指令操作码（OP0 - OP5）和功能码（F0 - F5），用于识别指令类型（如 R 型、LW、SW、BEQ 等）。



输出：包括 MemToReg（存储器到寄存器数据选择）、MemWrite（存储器写使能）、ALU_SRC（ALU 第二操作数选择）、RegWrite（寄存器写使能）、BEQ（相等则跳转）等控制信号





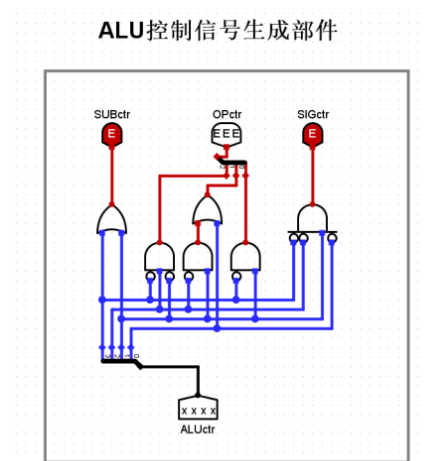


(11) 单周期 RISC-V 处理器的控制器能不能采用微程序控制器的方法设计？为什么？

答：可以，相比于硬布线控制器中的硬件时序，微程序控制器是一种软件时序，它可以使复杂的控制逻辑得到简化，但微程序控制器因为需要频繁访问控制存储器，其性能相比于硬布线控制器较差。RISC 系列计算机通常使用硬布线控制器。

(12) 请分析单周期 RISC-V 处理器（硬布线控制器）的下述电路原理：

a) ALU 控制信号生成部件

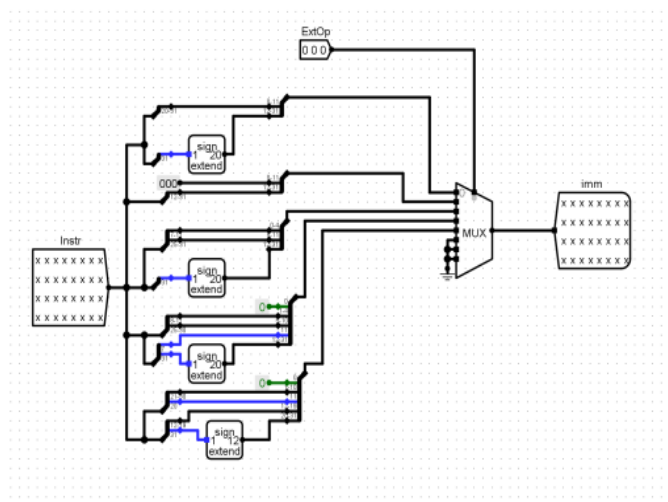


答：ALU 控制信号生成部件通过组合逻辑对输入信号（如 SUBctr、OPctr、

SIGctr) 进行处理, 产生控制 ALU 操作的信号 (ALUctr)。其原理是利用逻辑门 (与门、或门等) 对输入的控制信号进行译码与组合:

- ① 输入信号来自指令译码结果, 表征指令对 ALU 操作的需求 (如算术运算、逻辑运算等)。
- ② 逻辑门电路对这些信号进行判别与组合, 例如: 当 SUBctr 有效时, 可能通过逻辑门组合使 ALUctr 指示 ALU 执行减法操作; 通过不同输入组合, 生成对应 ALU 操作 (加、减、逻辑与、或等) 的控制信号, 确保 ALU 按指令要求完成运算

b) 立即数扩展部件



答: 立即数扩展部件的作用是根据指令类型, 将指令中的短立即数扩展为合适长度 (如 32 位), 以满足运算需求。其原理如下:

- ① 输入为指令中的立即数 (instr) 和控制信号 (EnOp)。
- ② 内部包含多种扩展逻辑 (如符号扩展 sign_extend):
 - 1) 符号扩展: 将立即数的符号位扩展至高位, 适用于需要保留符号的运算 (如加法、减法)。
 - 2) 不同扩展路径通过多路选择器 (MUX) 选择, EnOp 控制选择哪种扩展方式。例如, 对于某类指令, EnOp 选择符号扩展路径, 将 12 位立即数扩展为 32 位, 使数据在运算中保持正确语义。
- ③ 最终输出扩展后的立即数 (imm), 供 ALU 或其他部件使用, 确保指令中立即数在运算时具有正确的数值表示。

建议：针对电路分析题目，可以给出组合逻辑表达式或结合具体实例分析输入输出关系，从而支持其功能实现。

3.2 设计实验

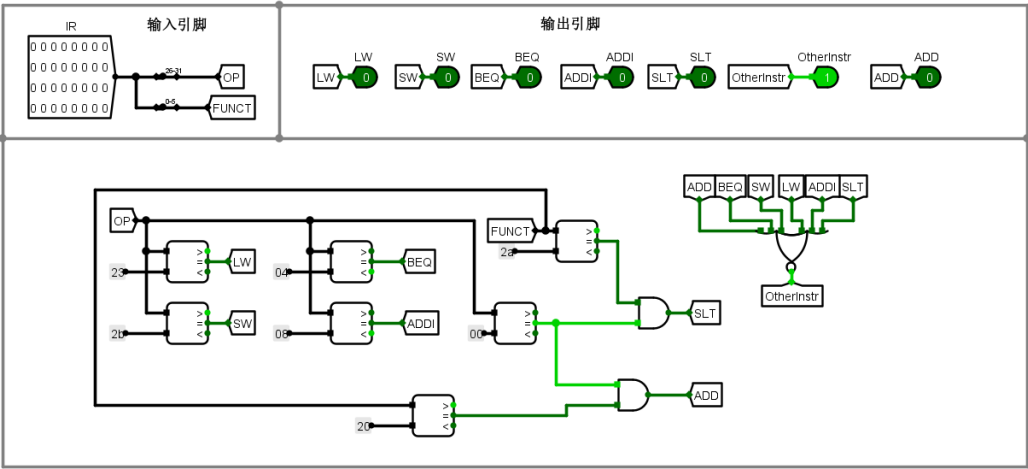
单总线结构 MIPS 处理器（微程序控制器）（增加 1 条 add 指令，共 6 条指令）

(1) 按照实验课件的要求，完成设计任务，提交设计文件单总线 MIPS 处理器-微程序.circ;

设计步骤：

根据课本，当 OP 码为 000000 且 ADD 的 FUNCT 码为 100000 时执行 add 指令，所以修改指令译码器如下：增加了 add 相关指令

指令译码器 (ID)



再根据微程序自动生成表格得到 add 指令：

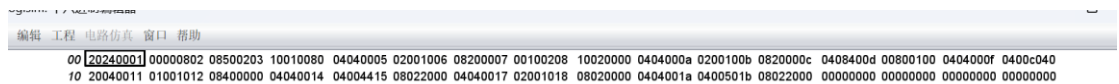
指令功能	PCout	DRout	Zout	Rout	PCin	ARin	DRin	DRin	Xin	Rin	Rin	PSW	Rs/Rt	Rs/Rt	Add	Add	Sl	Sl	READ	WRITE	P0	P1	P2	下址DEC	微指令	微指令十六进制
ADD	25		1						1						1									26	000100000001000000000000011010	404001A
ADD	26		1										1		1									27	00010000000000000101000000011011	400501B
ADD	27		1							1					1									0	001000000000100010000000000000	8022000

将得到的三条指令添加到微程序（5 条指令）.hex 的后面

```

v2.0 raw
20240001
802
8500203
10010080
4040005
2001006
8200007
100208
10020000
404000A
200100B
820000C
408400D
800100
404000F
400C040
20040011
1001012
8400000
4040014
4004415
8022000
4040017
2001018
8020000
404001A
400501B
8022000

```



最后修改微程序的入口查找逻辑

根据表格自动生成入口地址：

	A	B	C	D	E	I	J	K	L	M	N	O
1	机器指令译码信号						微程序入口地址					
2	LW	SW	BEQ	SLT	ADDI	ADD	入口地址 10进制	S4	S3	S2	S1	S0
3	1						4	0	0	1	0	0
4		1					9	0	1	0	0	1
5			1				14	0	1	1	1	0
6				1			19	1	0	0	1	1
7					1		22	1	0	1	1	0
8						1	25	1	1	0	0	1
9	次态10进制											
10												

	A	B	C	D	E	I	J	K	L	M	N	O
1	LW	SW	BEQ	SLT	ADDI	ADD	最小项表达式	S4	S3	S2	S1	S0
2	LW&						LW			LW+		
3		SW&					SW		SW+			SW+
4			BEQ&				BEQ		BEQ+	BEQ+	BEQ+	
5				SLT&			SLT	SLT+			SLT+	SLT+
6					ADDI&		ADDI	ADDI+		ADDI+	ADDI+	
7						ADD&	ADD	ADD+	ADD+			ADD+
8												
9												
10												
11												
12												
13												
14												
31								SLT+ADDI+ADD	SW+BEQ+ADD	LW+BEQ+ADDI	BEQ+SLT+ADDI	SW+SLT+ADD

输入表达式生成电路：

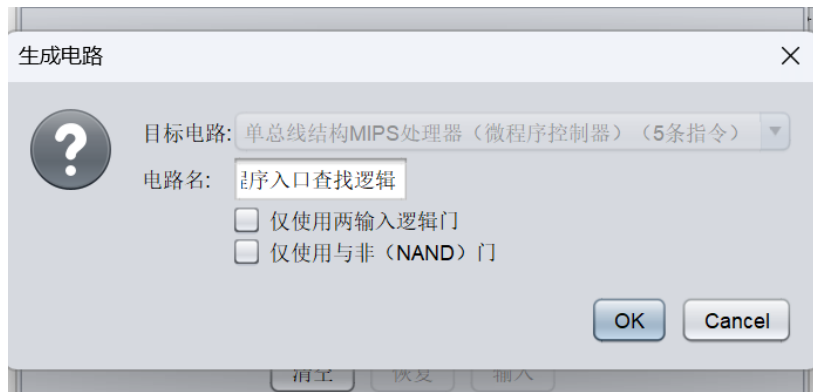
组合逻辑电路分析

文件 编辑 工程 电路仿真 窗口 帮助

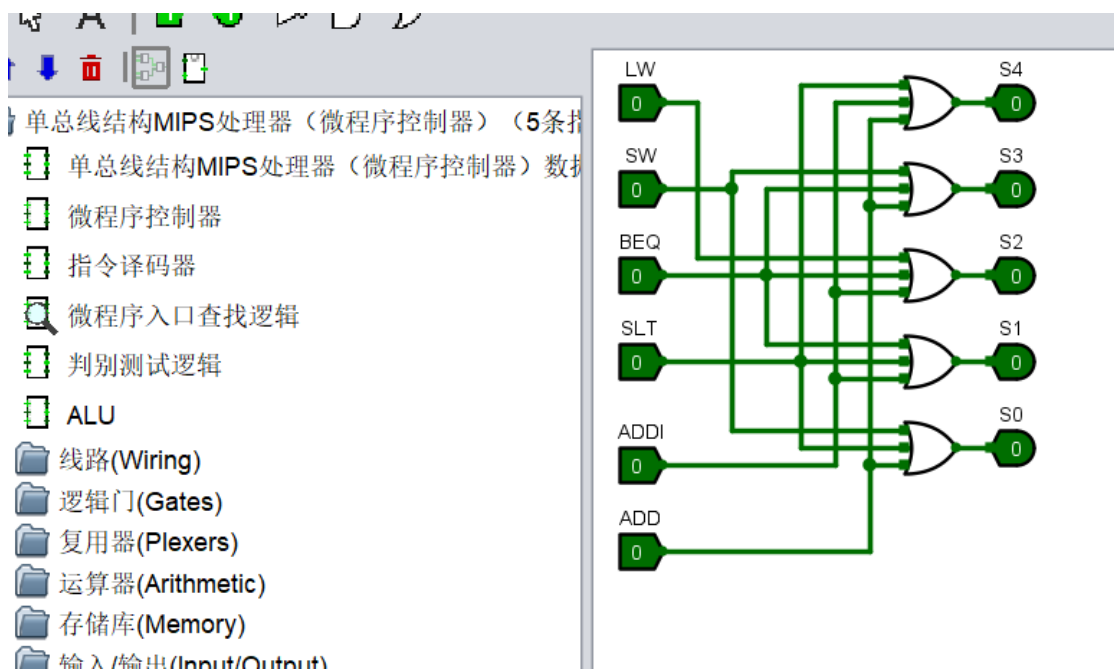
输出: S0

$SW + SLT + ADD$

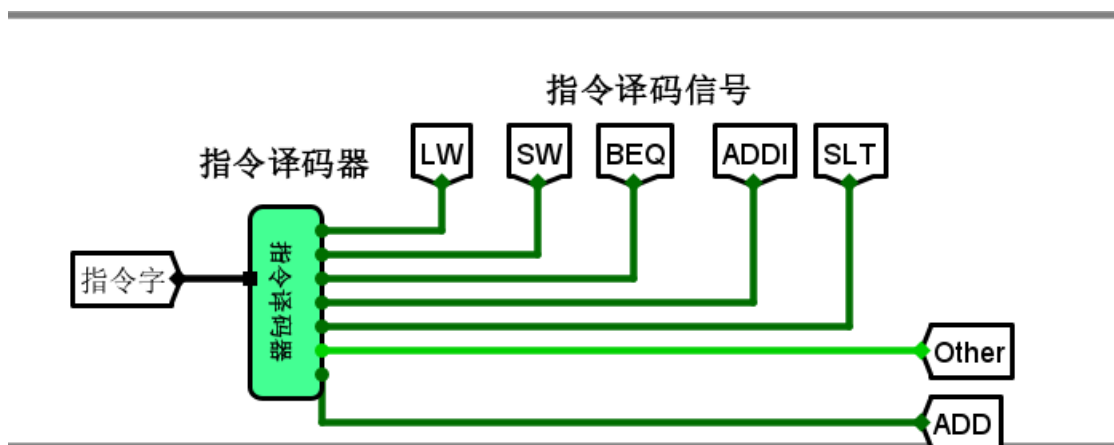
SW+SLT+ADD

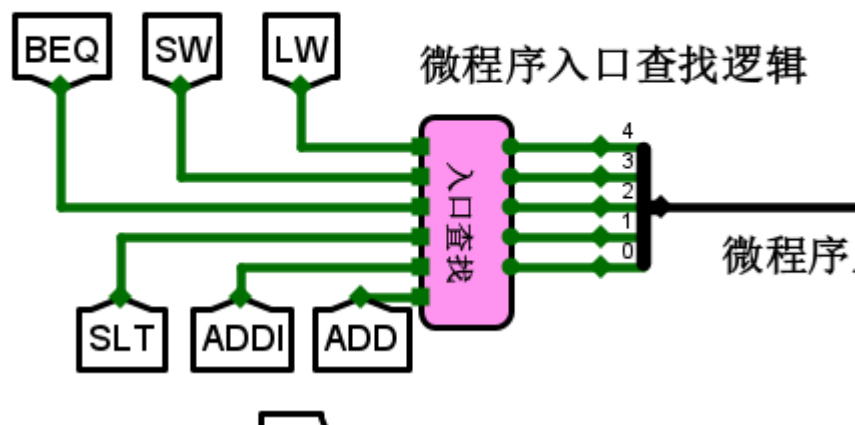


生成的微程序入口查找逻辑电路如下图：



最后再修改微程序控制器：添加 add





电路完成

- (2) 通过测试程序验证设计电路的正确性，将运行结果 copy 到实验报告中，并给出相关文字说明。

测试：

- 1) 请使用测试程序 test2.hex，对设计好的单总线 MIPS 处理器（微程序控制器，支持 6 条指令）进行测试。

```
0e0 00000000 00000000 00000000 00000000
0f0 00000000 00000000 00000000 00000000
100 00000008 00000009 0000000a 00000000
110 00000000 00000000 00000000 00000000
...
```

可以看到结果为 8 9 a 正确

- 2) 在单总线 MIPS 处理器（微程序控制器，支持 6 条指令）的数据通路上运行求累加和程序、计算费波那契数列程序、排序程序（降序排序、升序排序）。

求累加和程序

```
0f0 00000000 00000000 00000000 000
100 00000037 00000000 00000000 000
110 00000000 00000000 00000000 000
```


计算费波那契数列程序

```
.....  
100 00000000 00000001 00000001 00000002 00000003 00000005 00000008 0000000d 00000015 00000022 00000037 00000000  
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

排序程序（降序排序、升序排序）

降序：

```
0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00  
100 00000005 00000004 00000003 00000002 00000001 00000000 00000000 00  
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00
```

升序：

```
0f0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00  
100 00000001 00000002 00000003 00000004 00000005 00000000 00000000 00  
110 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00
```

验证结果显示正确。

3.3 挑战性实验

单总线结构 MIPS 处理器（硬布线控制器）（增加 1 条 add 指令，共 6 条指令）

- (1) 按照实验课件的要求，完成设计任务，提交设计文件单总线 MIPS 处理器-硬布线.circ;
- (2) 通过测试程序验证设计电路的正确性，将运行结果 copy 到实验报告中，并给出相关文字说明。

4. 实验报告提交

- (1) 实验报告命名为：学号+姓名+第 5 次实验报告.pdf。
- (2) 将实验报告和设计文件打包为 1 个压缩文件，压缩文件命名为：学号+姓名+第 5 次实验.zip。
- (3) 将压缩文件上传到数字化教学平台，第 5 次实验报告提交截止时间：2025 年 5 月 7 日晚上 24 点。