厦门大学

《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：3672023204041

2025 年 5 月 7 日

第 6 次实验　CPU 设计（II）

1. 实验目的

（1） 掌握多周期 MIPS 处理器（硬布线控制器、微程序控制器）设计的基本原理。

（2） 对已有的 6 条指令单总线结构 MIPS 处理器（微程序控制器）进行修改，再增加 1 条 j 指令（共 7 条指令）。

（3） 鼓励同学们对已有的 6 条指令单总线结构 MIPS 处理器（硬布线控制器）进行修改，再增加 1 条 j 指令（共 7 条指令）。

（4） 鼓励同学们对已有的 8 条指令多周期 MIPS 处理器(微程序控制器)进行修改，增加 1 条 j 指令（共 9 条指令）。

（5） 鼓励同学们对已有的 8 条指令多周期 MIPS 处理器(硬布线控制器)进行修改，增加 1 条 j 指令（共 9 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

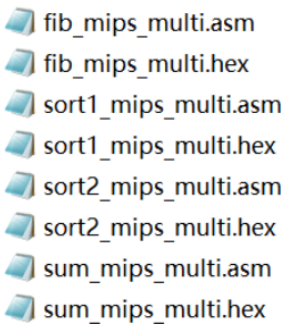
3.1 验证实验

（1）多周期 MIPS 处理器（硬布线控制器）（8 条指令）

（2）多周期 MIPS 处理器（微程序控制器）（8 条指令）

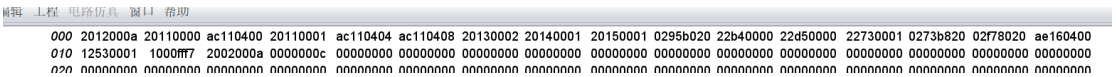
具体完成：

（1） 在多周期 MIPS 处理器（硬布线控制器）数据通路运行下述程序，结果截屏 copy 到实验报告中，给出相关的文字说明

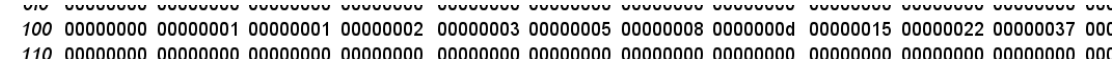


运行斐波那契数列：

加载程序

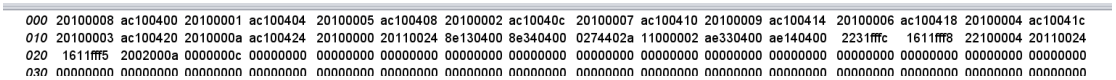


运行结果

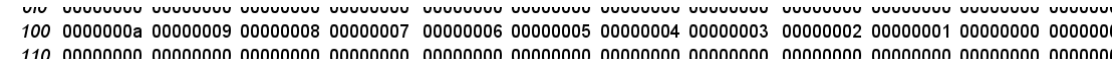


降序排序 **sort1**：

加载程序

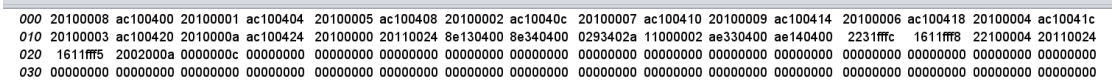


运行结果如下

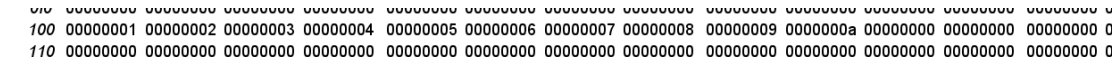


升序排序 **sort2**：

加载程序



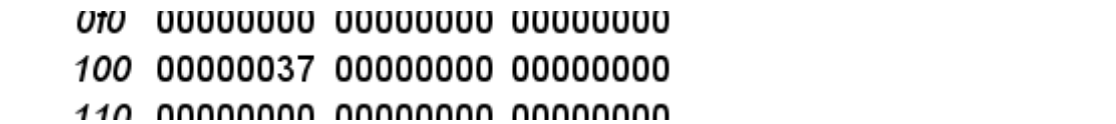
运行结果如下



求和程序 **sum**：

加载程序

运行结果如下



（2） 分析多周期 MIPS 处理器数据通路（硬布线或者微程序）中的电路原理：

a) 存在的可以并行的数据通路，（文字符号描述或图示法）

在多周期 MIPS 处理器数据通路中，以下操作可并行：

1）寄存器读与立即数扩展并行：读取寄存器堆（RegFile）中 rs、rt 数据的同时，对指令中的立即数（Imm16）进行符号扩展（16 - 32 位扩展）。两者无冲突，分别由不同部件处理，可在同一周期内并行执行，提高效率。

2）ALU 计算与寄存器写准备并行：例如，在 LW 指令中，ALU 计算数据地址（如 PC + offset）时，可同时准备寄存器写操作（如确定写寄存器号、写数据来源等），只要不涉及同一资源的冲突，即可并行。

b) 控制器状态机的输入输出之间的逻辑关系（表达式+实例）

输入有：S3 , S2 , S1 , S0 , LW , SW , BEQ , BNE , RTYPE , ADDI , SYSCALL

输出有：N3 , N2 , N1 , N0

表达式如下：

N3= ~S2 ~S1 S0 SYSCALL + ~S2 ~S1 S0 ADDI + ~S2 ~S1 S0 RTYPE + ~S2 ~S1 S0 BNE + S3 ~S2 S0 + S3 ~S1 S0

输出: N3

$\overline{S2}\ \overline{S1}\ S0\ SYSCALL + \overline{S2}\ \overline{S1}\ S0\ ADDI + \overline{S2}\ \overline{S1}\ S0\ RTYPE + \overline{S2}\ \overline{S1}\ S0\ BNE + S3\ \overline{S2}\ S0 + S3\ \overline{S1}\ S0$

N2= ~S3 ~S1 S0 SYSCALL + ~S3 ~S2 S0 BEQ + ~S3 ~S1 S0 SW + ~S2 S1 S0 + S2 ~S1 S0

输出: N2

$\overline{S3}\ \overline{S1}\ S0\ SYSCALL + \overline{S3}\ \overline{S2}\ S0\ BEQ + \overline{S3}\ \overline{S1}\ S0\ SW + \overline{S2}\ S1\ S0 + S2\ \overline{S1}\ S0$

N1= ~S2 ~S1 S0 ADDI + ~S2 ~S1 S0 BEQ + ~S2 ~S1 S0 LW + ~S3 ~S2 S1 ~S0 + ~S3 S2 ~S1 S0 + S3 ~S2 ~S1 S0

输出: N1

$\overline{S2}\ \overline{S1}\ S0\ ADDI + \overline{S2}\ \overline{S1}\ S0\ BEQ + \overline{S2}\ \overline{S1}\ S0\ LW + \overline{S3}\ \overline{S2}\ S1\ \overline{S0} + \overline{S3}\ S2\ \overline{S1}\ S0 + S3\ \overline{S2}\ \overline{S1}\ S0$

N0= ~S3 ~S2 ~S0 + ~S3 ~S2 ~S1 SYSCALL + ~S3 ~S2 ~S1 ADDI + ~S3 ~S2 ~S1 RTYPE + ~S3 ~S2 ~S1 BEQ + ~S3 ~S2 ~S1 SW + S3 S2 ~S1 S0

输出: N0

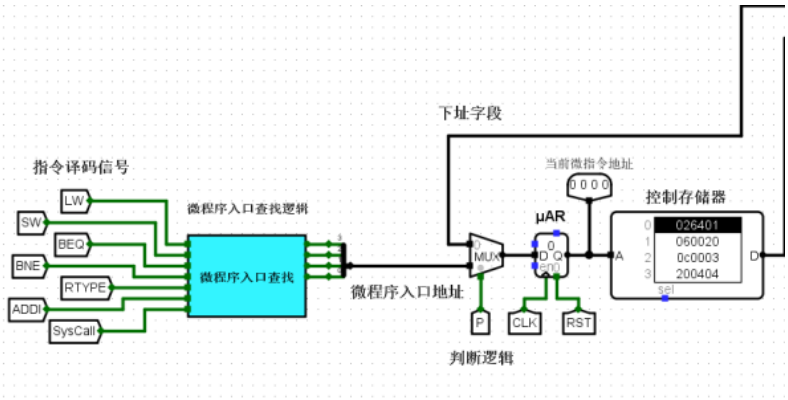
$\overline{S3}\ \overline{S2}\ \overline{S0} + \overline{S3}\ \overline{S2}\ \overline{S1}\ SYSCALL + \overline{S3}\ \overline{S2}\ \overline{S1}\ ADDI + \overline{S3}\ \overline{S2}\ \overline{S1}\ RTYPE + \overline{S3}\ \overline{S2}\ \overline{S1}\ BEQ + \overline{S3}\ \overline{S2}\ \overline{S1}\ SW + S3\ S2\ \overline{S1}\ S0$

实例：

当 S3=0，S2=1，S1=0，S0=1，LW=1，SW=1，BEQ=1，BNE=0，RTYPE=1，ADDI=0，SYSCALL=1 时

N3=0，N2=1，N1=1，N0=0

（3）分析多周期 MIPS 处理器（微程序控制器）下址字段法电路中的地址转移逻辑（即针对不同情况，如何得到后续微指令的微地址）

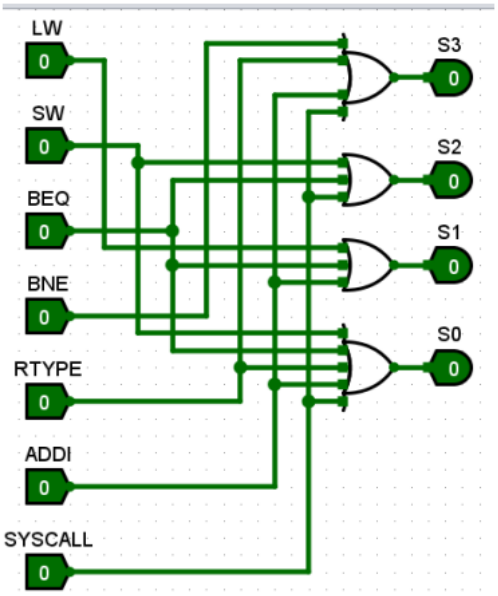


顺序执行时：微指令的下址字段直接给出下一条微指令的地址，无需额外条件判断。例如，当一条微指令执行完毕且无分支需求时，直接依据下址字段的值，将其送入微地址寄存器（μAR），作为下一条微指令的地址，实现顺序执行。

条件转移时：结合判断逻辑与下址字段，同时利用微程序入口查找逻辑处理不同指令的起始地址，实现灵活准确的微指令地址转移，确保各指令的微程序按正确流程执行。

微程序入口地址生成：指令译码信号（如 LW、SW、RTYPE 等）通过 “微程序入口查找逻辑” 生成对应指令的微程序入口地址。例如，LW 指令经译码后，微程序入口查找逻辑输出其专属入口地址，送入 μAR，启动该指令的微程序执行流程。

（4）分析多周期 MIPS 处理器（微程序控制器）下址字段法电路中的微程序入口查找逻辑，即：如何得到不同指令对应的微程序入口地址的



输入：7个指令的译码信号	输出：微程序入口地址
LW=1	微程序入口地址=2
SW=1	微程序入口地址=5
BEQ=1	微程序入口地址=7
BNE=1	微程序入口地址=8
RTYPE=1	微程序入口地址=9
ADDI=1	微程序入口地址=11
SYSCALL=1	微程序入口地址=13

多周期 MIPS 处理器（微程序控制器）下址字段法电路中的微程序入口查找逻辑，通过指令译码信号与逻辑门电路组合生成微程序入口地址。输入为 7 个指令的译码信号（LW、SW、BEQ、BNE、RTYPE、ADDI、SYSCALL），输出为微程序入口地址（由 S3、S2、S1、S0 组合表示）。具体逻辑如下：

每个指令对应唯一的 S3S2S1S0 组合：

LW=1：逻辑门组合使 S3S2S1S0 为 0010（二进制），对应十进制地址 2。

SW=1：S3S2S1S0 为 0101，对应地址 5。

BEQ=1：S3S2S1S0 为 0111，对应地址 7。

BNE=1：S3S2S1S0 为 1000，对应地址 8。

RTYPE=1：S3S2S1S0 为 1001，对应地址 9。

ADDI=1：S3S2S1S0 为 1011，对应地址 11。

SYSCALL=1：S3S2S1S0 为 1101，对应地址 13。

通过指令译码信号触发相应逻辑门，输出特定的 S3S2S1S0 组合，形成对应指令的微程序入口地址，实现不同指令到微程序入口的映射。

3.2 设计实验

单总线结构 MIPS 处理器（微程序控制器）（再增加 1 条 j 指令，共 7 条指令）

请按照实验课件的要求，完成设计任务，提交电路文件：单总线 MIPS（微程序）-7 条指令.circ。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。

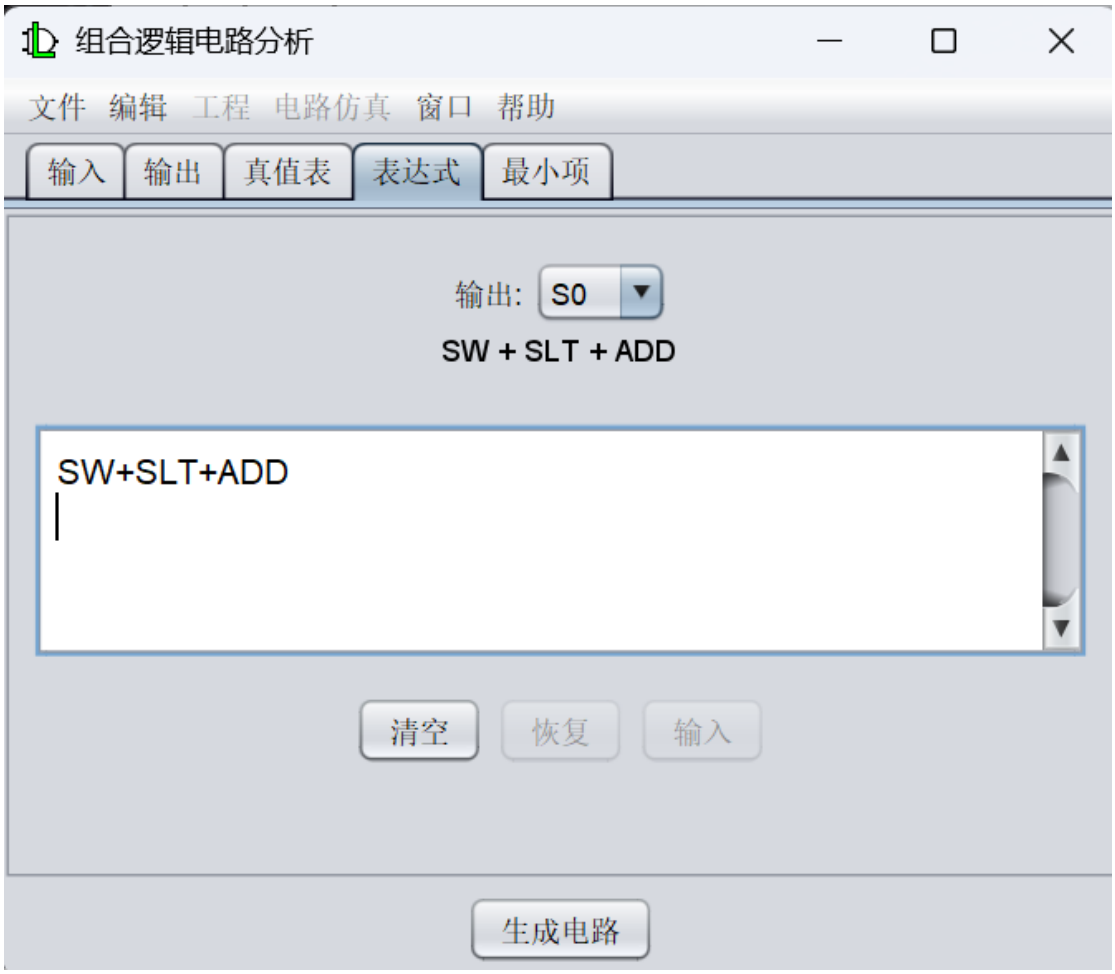
填写“微程序入口查找逻辑自动生成（7 条指令）（空表）.xlsx”，生成逻辑公式

机器指令译码信号							微程序入口地址					
LW	SW	BEQ	SLT	ADDI	ADD	J	入口地址10进制	S4	S3	S2	S1	S0
1							4	0	0	1	0	0
	1						9	0	1	0	0	1
		1					14	0	1	1	1	0
			1				19	1	0	0	1	1
				1			22	1	0	1	1	0
					1		25	1	1	0	0	1
						1	28	1	1	1	0	0

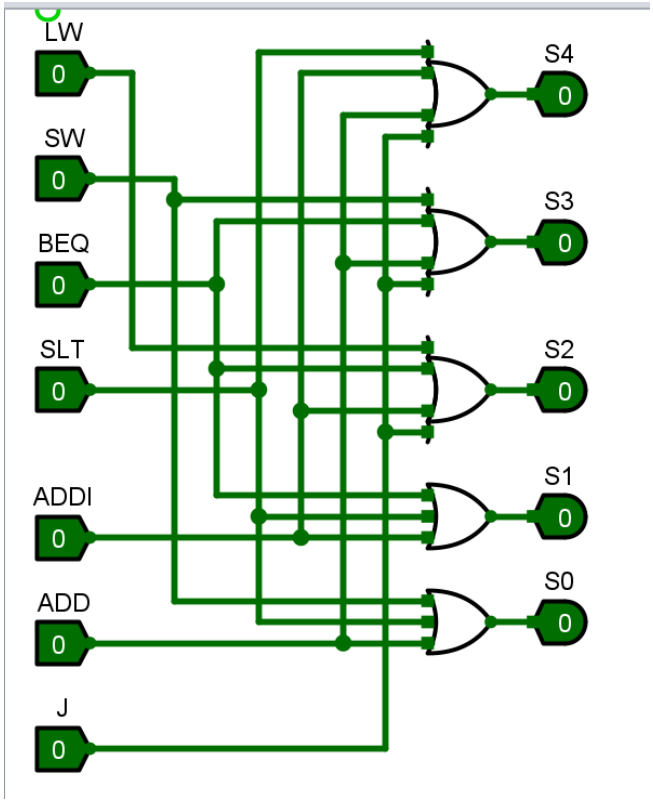
LW	SW	BEQ	SLT	ADDI	ADD	J	最小项表达式	S4	S3	S2	S1	S0
LW&							LW			LW+		
	SW&						SW		SW+			SW+
		BEQ&					BEQ		BEQ+	BEQ+	BEQ+	
			SLT&				SLT	SLT+			SLT+	SLT+
				ADDI&			ADDI	ADDI+		ADDI+	ADDI+	
					ADD&		ADD	ADD+	ADD+			ADD+
						J&	J	J+	J+	J+		
								SLT+ADDI+ADD+J	SW+BEQ+ADD+J	LW+BEQ+ADDI+J	BEQ+SLT+ADDI	SW+SLT+ADD

在 Logisim 上生成“微程序入口查找逻辑”电路：

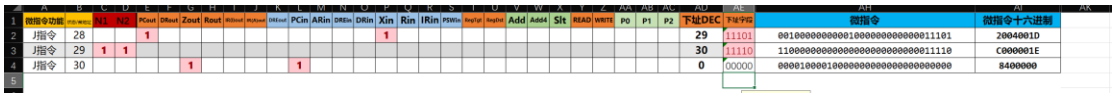
将上面得到的表达式输入



生成微程序入口查找逻辑电路如下：



填写“微程序自动生成（j 指令）（空表）.xlsx”，生成 j 指令的微程序：



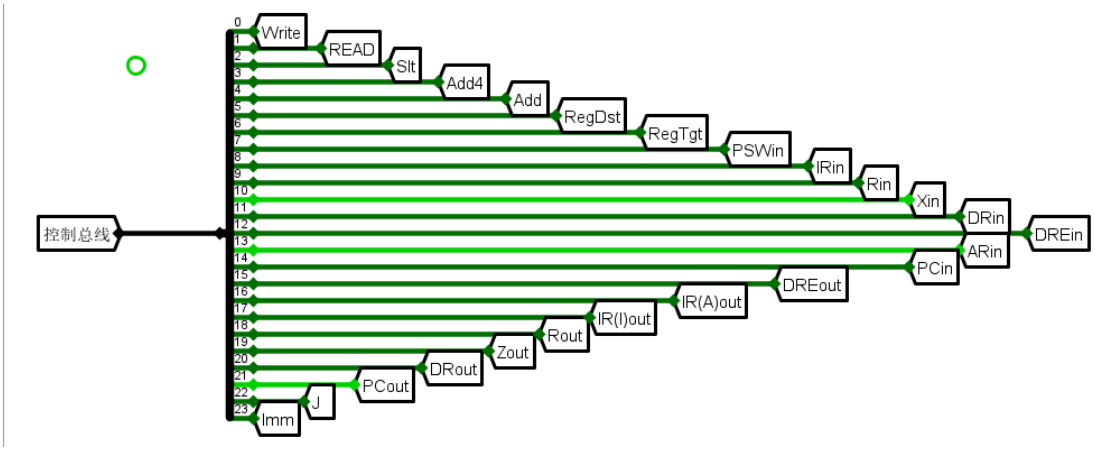
微指令功能	微地址	N1	N2	PCout	DRout	Zout	Rout	IRout	PUSHout	DREin	PCin	ARin	DREin	DRin	Xin	Rin	IRin	PSWin	Reg1in	Reg2in	Add	Add4	Slt	READ	WRITE	P0	P1	P2	下址DEC	下址字段	微指令	微指令十六进制
J指令	28			1											1														29	11101	00100000000001000000000000011101	2004001D
J指令	29	1	1																										30	11110	11000000000000000000000000011110	C000001E
J指令	30					1					1																		0	00000	00001000010000000000000000000000	8400000

微指令十六进制
2004001D
C000001E
8400000

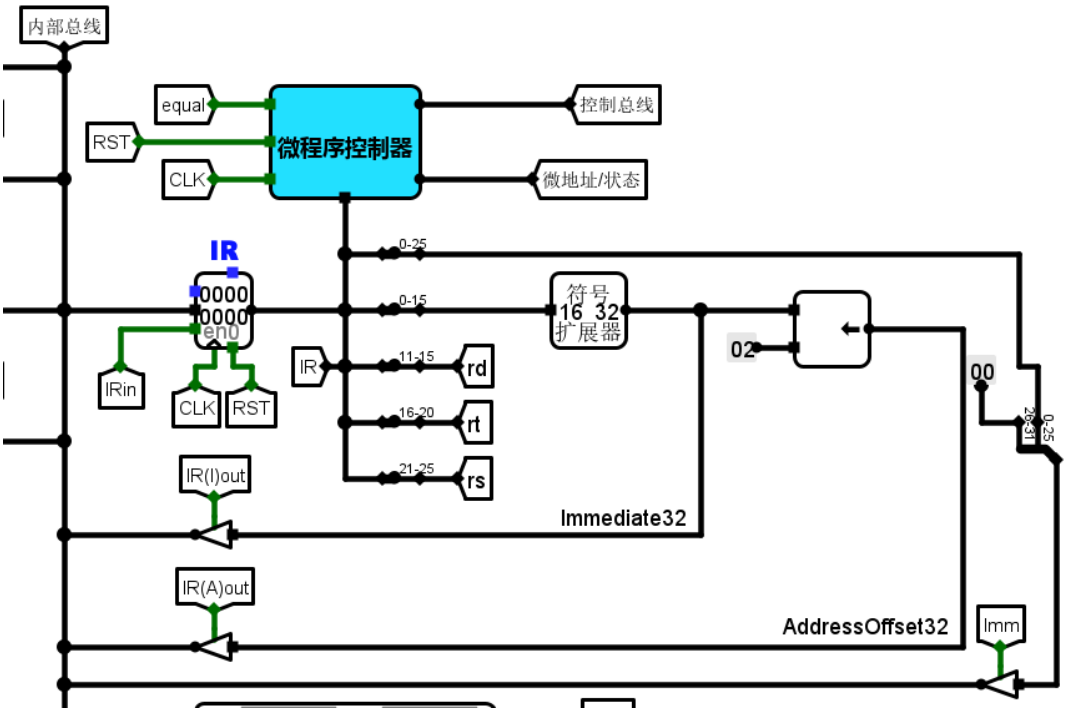
添加到 6 条指令的微程序最后面

```
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
2004001D
C000001E
8400000
```

修改数据通路，将 26 位立即数 imm26 从指令寄存器 IR 中引出，新增 2 个控制信号 Imm 和 J

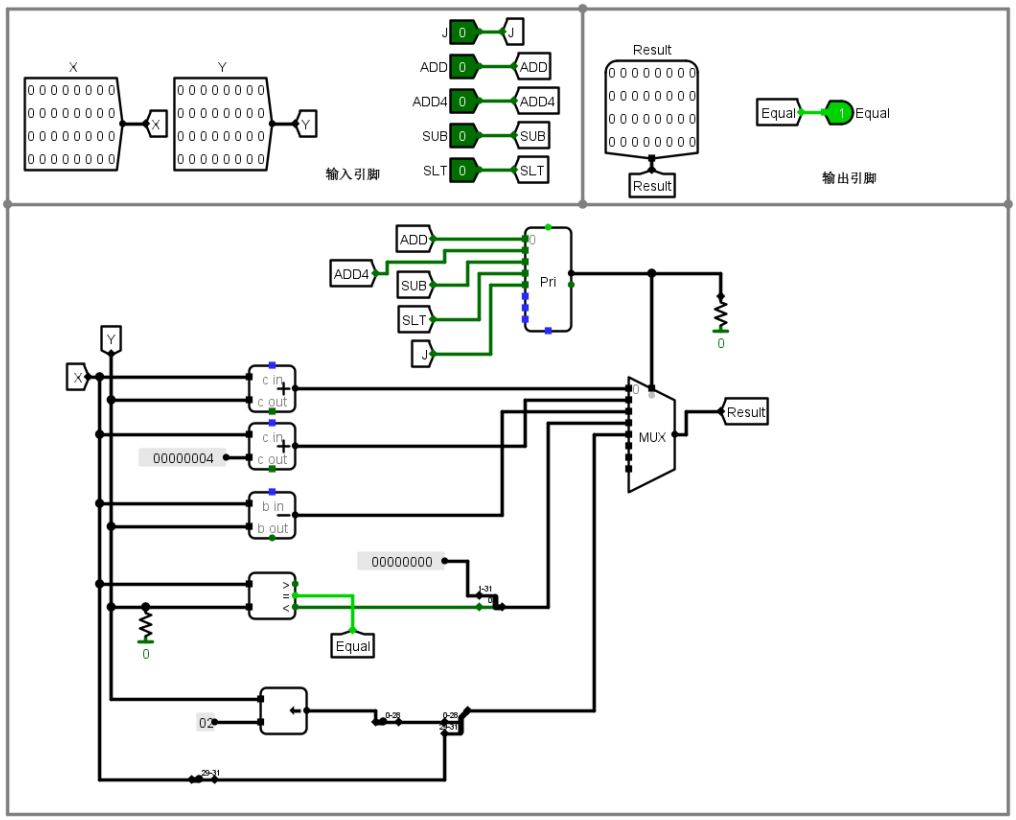


获取指令字的低 26 位，如果 Imm 为 1，三态门打开，则将 26 位立即数传入总线



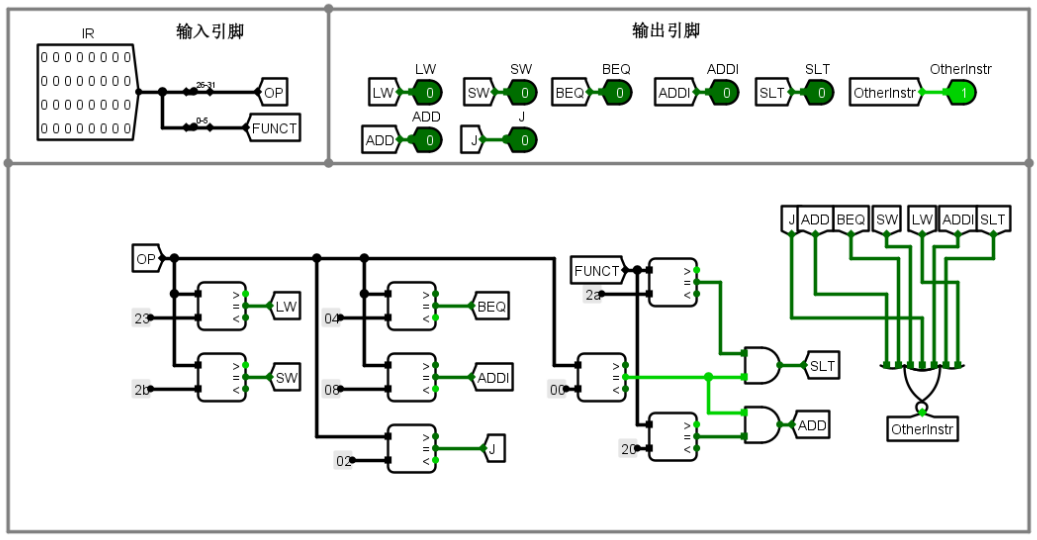
修改 Alu：增加 J 的输入引脚和隧道，优先编码器中新增的 J 隧道控制 MUX 的第 4 路，第 4 路用于拼接运算：具体原理是：获取 Y 的整体数据左移 2 位后得到的低 28 位（立即数），获取 X（存储 PC 值）的数据的高 4 位，再利用分线器将低 28 位和高 4 位进行拼接，成为 32 位的地址

算术逻辑运算单元ALU

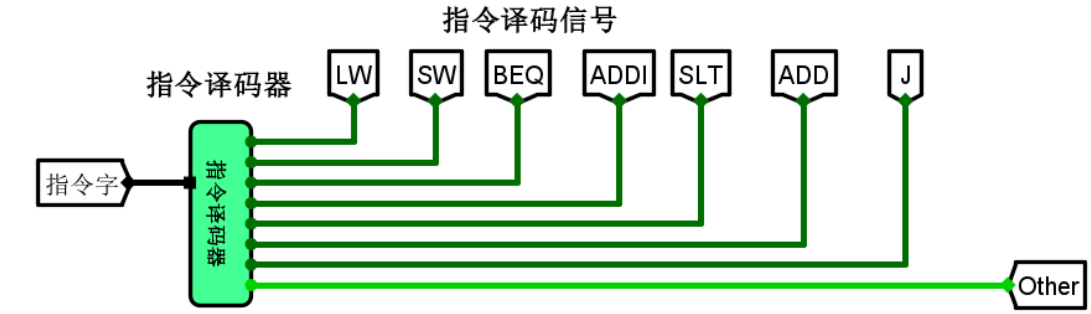


J 型指令 OP 码为 000010，十六进制 02，在指令译码器（ID）的输出引脚增加 J 指令

指令译码器（ID）



修改微程序控制器，新增 J 型指令的指令译码信号，如下图：

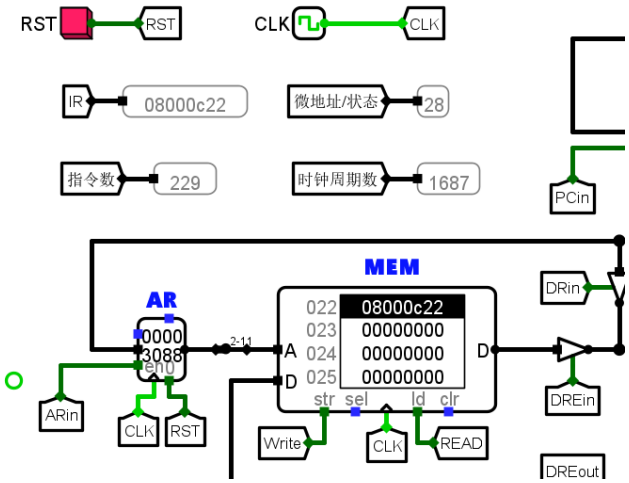


验证如下：

（1） 请使用测试程序 test3.hex，对设计好的单总线 MIPS 处理器（微程序控制器，支持 7 条指令）进行测试。

```
0f0  00000000 00000000 00000000 00000000
100  00000008 00000009 00000014 00000000
110  00000000 00000000 00000000 00000000
```

（2） 请使用 sort_sum_mips_bus.hex（排序+求累加和程序），对设计好的单总线 MIPS 处理器（微程序控制器，支持 7 条指令）进行测试。




```
0f0  00000000 00000000 00000000 00000000 00000000 0000
100  00000005 00000004 00000003 00000002 00000001 0000
110  00000037 00000000 00000000 00000000 00000000 0000
120  00000000 00000000 00000000 00000000 00000000 0000
```

3.3 挑战性实验

（1）单总线结构 MIPS 处理器（微程序控制器）（再增加 1 条 j 指令，共 7 条指令）

（2）多周期 MIPS 处理器（微程序控制器）（增加 1 条 j 指令，共 9 条指令）

（3）多周期 MIPS 处理器（硬布线控制器）（增加 1 条 j 指令，共 9 条指令）

请按照实验课件的要求，完成设计任务，提交电路文件：单总线 MIPS（硬布线）-7 条指令.circ，多周期 MIPS（微程序）-9 条.circ，多周期 MIPS（硬布线）-9 条.circ 。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 6 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 6 次实验.zip。

(3) 将压缩文件上传到平台上，第 6 次实验报告提交截止时间（2 周内）：2025 年 5 月 21 日晚上 24 点。