

华中科技大学

2024

计算机组成原理

· 实验报告 ·

专 业: 计算机科学与技术

班 级: CS2201

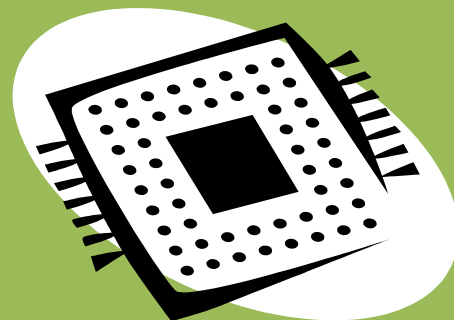
学 号: U202215365

姓 名: 叶俊江

电 话: 15767792195

邮 件: 1993783308@qq.com

完成日期: 2024-06-20



计算机科学与技术学院

1 CPU 设计实验

1.1 设计要求

利用 logisim 平台完成 MIPS 现代时序中断机制实现，任务由如下几部分组成：

- MIPS 指令译码器设计
- 支持中断的微程序入口查找逻辑
- 支持中断的微程序条件判别测试逻辑
- 支持中断的微程序控制器设计
- 支持中断的微程序单总线 CPU 设计
- 支持中断的现代时序硬布线控制器状态机设计
- 支持中断的现代时序硬布线控制器设计

在 Logisim 中打开 MipsOnBusCpu-1.circ 文件完成电路连接后，再将文件使用文本编辑打开并将代码复制到头歌平台中评测，七个关卡都应评测通过。

1.2 方案设计

1.2.1 MIPS 指令译码器设计

子电路引脚功能如下图表 1-1 所示。

表 1-1 指令译码器电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
IR	输入	32	MIPS 指令字
LW	输出	1	当前指令为 lw 指令时输出为 1
SW	输出	1	当前指令为 sw 指令时输出为 1
BEQ	输出	1	当前指令为 beq 指令时输出为 1
ADDI	输出	1	当前指令为 addi 指令时输出为 1

子电路要求根据输入的指令字 IR 进行译码 (IR 已分为 OP 和 FUNC 两部分)，判

断是否为 LW/SW/BEQ/ADDI/SLT 指令或者是其他的指令，查阅 MIPS32 指令手册可知各个指令对应的 OP 与 FUNC 的值如下表 1-2 所示。

表 1-2 各指令对应 OP 与 FUNC 的值

指令	OP	FUNC
SLT	000000	101010
BEQ	000100	
ADDI	001000	
LW	100011	
SW	101011	

那么我们只需要将 OP 和 FUNC 分别与对应的常数值比较即可，若不是 LW/SW/BEQ/ADDI/SLT 这五种指令时 OtherInstr 输出为 1。电路图如下图 1-1 所示。

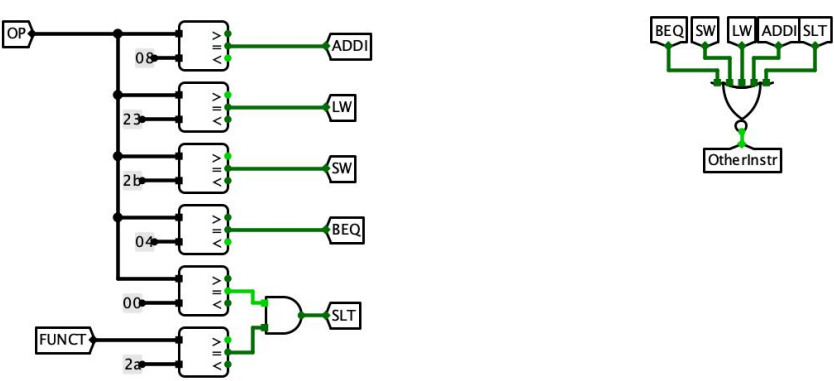


图 1-1 指令译码器电路图

1.2.2 支持中断的微程序入口查找逻辑

子电路引脚功能如下表 1-3 所示。

表 1-3 支持中断的微程序入口查找逻辑电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
LW	输入	1	为 1 时表示为 lw 指令
SW	输入	1	为 1 时表示为 sw 指令
BEQ	输入	1	为 1 时表示为 beq 指令
ADDI	输入	1	为 1 时表示为 addi 指令

华中科技大学课程实验报告

SLT	输入	1	为 1 时表示为 slt 指令
ERET	输入	1	为 1 时表示为 lw 指令
S4 ~ S0	输出	1	5 位微程序入口地址

根据教材中微指令格式和 6 条指令的微程序设计原理，由于我们加入了中断功能，那么微指令操作的控制字段就需要添加相应的内容，设计完成后可得输入指令为 LW/SW/BEQ/ADDI/SLT/ERET 的时候，微程序入口地址为 4/9/14/19/22/25，相应的 Excel 表格填写如下图 1-2 所示。

机器指令译码信号						微程序入口地址					
LW	SW	BEQ	SLT	ADDI	ERET	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

图 1-2 微程序入口地址表

电路图如下图 1-3 所示。

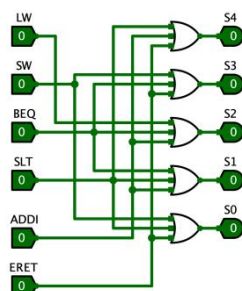


图 1-3 微程序入口查找逻辑电路图

1.2.3 支持中断的微程序条件判别测试逻辑

子电路引脚功能如下表 1-4 所示。

表 1-4 支持中断的微程序条件判别测试逻辑电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
P0	输入	1	条件判断位，用于指令译码
P1	输入	1	条件判断位，用于 beq 分支
P2	输入	1	条件判断位，用于中断检测
equal	输入	1	运算操作数相等标志位

华中科技大学课程实验报告

IntR	输入	1	中断请求信号
MuxSel	输出	2	MUX 选择控制信号

根据教材中支持中断微程序控制器原理图，再结合判别位测试位的定义，设计条件判别测试组合逻辑。当 P0 ~ P2 中出现 1 时，表示需要进行分支判断，随后再根据 equal 和 IntR 的值，判断是否进入入口地址/beq 分支/中断响应入口/取值微程序入口。相应的 Excel 文件填写如下图 1-4 所示。

输入 (填1或0, 不填为无关项x)							
P0	P1	P2	equal	IntR	S2	S1	S0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	1	0	0	1	0
0	1	1	1	1	0	1	0
0	1	1	0	1	0	1	1
0	1	1	0	0	1	0	0
0	0	1	0	1	0	1	1
0	0	1	1	1	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	0
0	1	0	0	1	1	0	0

图 1-4 判别测试逻辑真值表

由于电路图是使用 Logisim 的自动生成功能生成的，较为复杂，故在此处不再列出电路图。

1.2.4 支持中断的微程序控制器设计

子电路引脚功能如下表 1-5 所示。

表 1-5 支持中断的微程序控制器设计电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
IR	输入	32	MIPS 指令字
equal	输入	1	条件反馈信号，表示运算相等
IntR	输入	1	中断请求信号，高电平有效
mAddr	输出	1	当前微指令地址
ControlBus	输出	1	微指令字的控制字段

华中科技大学课程实验报告

表 1-6 支持中断的微程序单总线 CPU 设计电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
ControlBus	输出	22	控制总线，用于 EduCoder 系统自动测试
中断控制信号	输出	5	中断控制信号

本关主要是要在单总线数据通路中添加与中断相关的硬件模块，主要是包括异常程序设计计数器 EPC，中断使能寄存器 IE，中断控制器等模块，需要在主电路中将这
些模块进行有效连接，并在本馆进行最终的联调，测试 CPU 能否正常响应两个按键所
对应的中断服务程序。（注意不同按键的中断服务程序入口地址可以利用 MARS 汇编
器汇编源程序查看 label 地址得到）电路图如下图 1-7 所示。

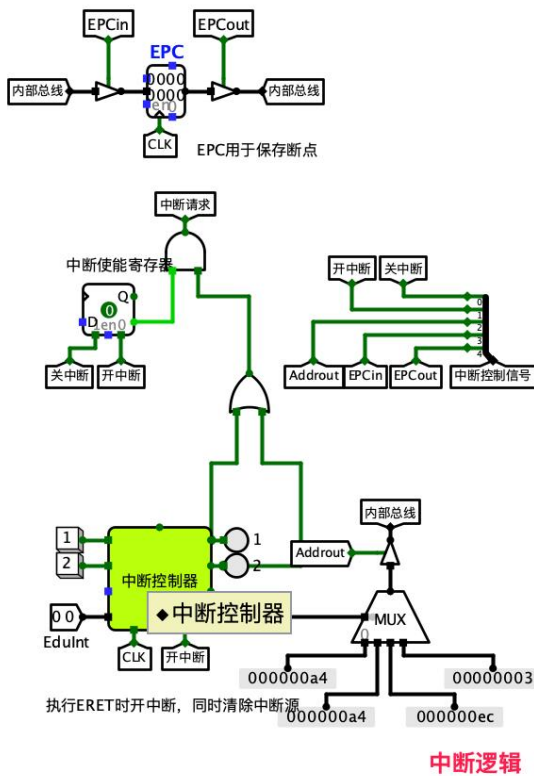


图 1-7 中断逻辑电路图

1.2.6 支持中断的现代时序硬布线控制器状态机设计

子电路引脚功能如下表 1-7 所示。

表 1-6 支持中断的微程序单总线 CPU 设计电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
LW	输入	1	为 1 表示为 lw 指令

华中科技大学课程实验报告

SW	输入	1	为 1 表示为 sw 指令
BEQ	输入	1	为 1 表示为 beq 指令
SLT	输入	1	为 1 表示为 slt 指令
ADDI	输入	1	为 1 表示为 addi 指令
ERET	输出	1	为 1 表示为 eret 指令
IntR	输入	1	中断请求信号
equal	输入	1	运算操作数相等标志位
现态 S0~S4	输入	5	FSM 状态机现态
次态 N0~N4	输出	5	FSM 状态机次态

要实现的状态图如图 1-8 所示。

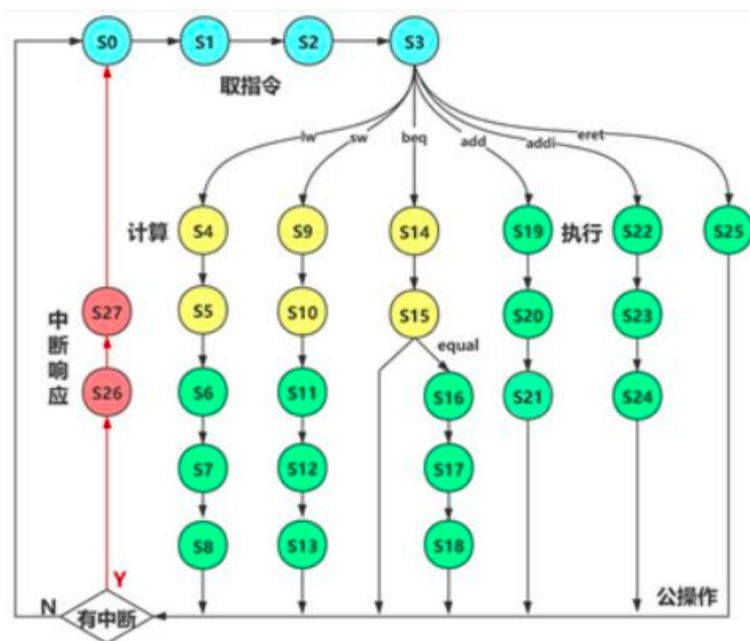


图 1-8 支持中断的现代时序状态机

根据上图可直接填写相应的 Excel 文件，自动生成组合逻辑，填写文件如图 1-9 所示。

当前状态(现态)						输入信号								下一状态(次态)					
S4	S3	S2	S1	S0	现态 10进制	LW	SW	BEQ	SLT	ADDI	ERET	IR	EQUAL	次态 10进制	N4	N3	N2	N1	N0
0	0	0	0	0	0									1	0	0	0	0	1
0	0	0	0	1	1									2	0	0	0	1	0
0	0	0	1	0	2									3	0	0	0	1	1
0	0	0	1	1	3	1								4	0	0	1	0	0
0	0	0	1	1	3		1							9	0	1	0	0	1
0	0	0	1	1	3			1						14	0	1	1	1	0
0	0	0	1	1	3				1					19	1	0	0	1	1
0	0	0	1	1	3					1				22	1	0	1	1	0
0	0	0	1	1	3						1			25	1	1	0	0	1
0	0	1	0	0	4							1		5	0	0	1	0	1
0	0	1	0	1	5									6	0	0	1	1	0
0	0	1	1	0	6									7	0	0	1	1	1
0	0	1	1	1	7									8	0	1	0	0	0
0	1	0	0	0	8							1		26	1	1	0	1	0
0	1	0	0	0	8							0		0	0	0	0	0	0
0	1	0	0	1	9									10	0	1	0	1	0
0	1	0	1	0	10									11	0	1	0	1	1
0	1	0	1	1	11									12	0	1	1	0	0
0	1	1	0	0	12									13	0	1	1	0	1
0	1	1	0	1	13							1		26	1	1	0	1	0
0	1	1	0	1	13							0		0	0	0	0	0	0
0	1	1	1	0	14									15	0	1	1	1	1
0	1	1	1	1	15							1	0	26	1	1	0	1	0
0	1	1	1	1	15							0	0	0	0	0	0	0	0
0	1	1	1	1	15								1	16	1	0	0	0	0
1	0	0	0	0	16									17	1	0	0	0	1
1	0	0	0	1	17									18	1	0	0	1	0
1	0	0	1	0	18							1		26	1	1	0	1	0
1	0	0	1	0	18							0		0	0	0	0	0	0
1	0	0	1	1	19									20	1	0	1	0	0
1	0	1	0	0	20									21	1	0	1	0	1
1	0	1	0	1	21							1		26	1	1	0	1	0
1	0	1	0	1	21							0		0	0	0	0	0	0
1	0	1	1	0	22									23	1	0	1	1	1
1	0	1	1	1	23									24	1	1	0	0	0
1	1	0	0	0	24							1		26	1	1	0	1	0
1	1	0	0	0	24							0		0	0	0	0	0	0
1	1	0	0	1	25							1		26	1	1	0	1	0
1	1	0	0	1	25							0		0	0	0	0	0	0
1	1	0	1	0	26									27	1	1	0	1	1
1	1	0	1	1	27									0	0	0	0	0	0

图 1-9 硬布线控制器状态机状态转换表

由于电路图是使用 Logisim 的自动生成功能生成的，较为复杂，故在此处不再列出电路图。

1.2.7 支持中断的现代时序硬布线控制器设计

子电路引脚功能如下表 1-8 所示。

表 1-8 支持中断的现代时序硬布线控制器设计电路引脚与功能描述

引脚	输入/输出	位宽	功能描述
IR	输入	32	MIPS 指令字
CLK	输入	1	时钟输入
equal	输入	1	条件状态信号，相等标志
控制总线	输出	22	所有控制信号汇总而成的控制总线
当前状态值	输出	5	状态字的值

华中科技大学课程实验报告

在实现指令译码、现代时序状态机模块后，最终完成硬布线控制器的封装。如下图 1-10 中完成了硬布线控制器框架连接，此时需注意硬布线控制器组合逻辑不需要实现直接采用微程序控制器的控制存储器代替即可，完成测试后用硬布线控制器替换 CPU 中的微程序控制器进行程序测试。

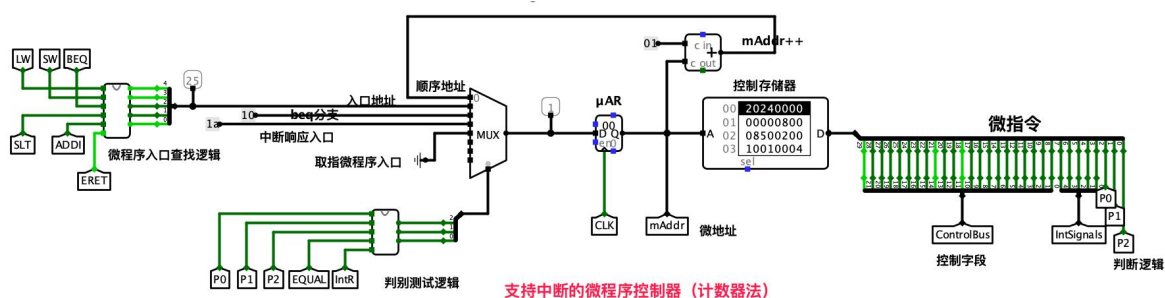


图 1-10 硬布线控制器电路图

1.3 实验步骤

- (1) 仔细阅读头歌平台各关卡中的实验提示，并了解各部分子电路的功能。
- (2) 根据提示及上课所学知识完成子电路的设计，对于部分关卡需要填写实验包里面提供的 Excel 文件来自动生成表达式，再使用 Logisim 的自动生成功能生成组合逻辑电路。
- (3) 提交代码文件到头歌平台进行评测。

1.4 故障与调试

1.4.1 状态转移存在问题

故障现象：在“支持中断的微程序控制器设计”关卡中，后续近一半指令全部错误。

原因分析：在 Excel 文件中设计时，在 SW 微程序最后一条指令 P3=1 未设置，导致 SW 微程序执行后无法返回取指微程序。

解决方案：修改对应 Excel 文件中出错的跳转逻辑。

1.5 测试与分析

在“支持中断的微程序单总线 CPU 设计”关卡中可对设计的电路进行测试。在 RAM 中加载 sort-5-int.hex 数据镜像，使用快捷键⌘command+k 即可运行，若无其他按键行为，程序会在 1992 节拍数停下，指令计数为 252。测试结果如下图 1-11 所示。

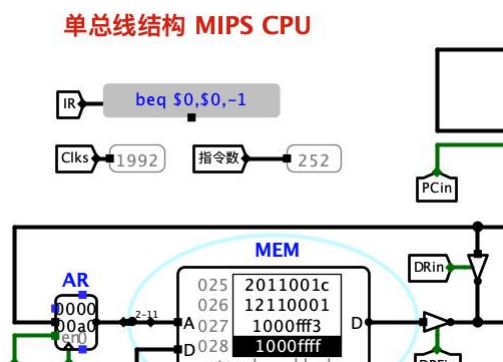


图 1-11 CPU 测试结果

从测试结果可见符合我们的预期结果，故而电路设计正确。

内存数据如下图 1-12 所示。

```
000 23bd0400 2010ffff 20110000 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200
010 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001
020 ae140200 2231ffff 12110001 1000fff7 22100004 2011001c 12110001 1000fff3 1000ffff 23bd0008 afb00000 afb10004 20310240 8e300000 22100001 ae300000
030 ae300004 ae300008 ae30000c ae300010 ae300014 ae300018 ae30001c 8fb10004 8fb00000 23bdfff8 42000018 23bd0008 afb00000 afb10004 20310280 8e300000
040 2210ffff ae300000 ae300004 ae300008 ae30000c ae300010 ae300014 ae300018 ae30001c 8fb10004 8fb00000 23bdfff8 42000018 00000000 00000000 00000000
050 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
070 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
080 00000006 00000005 00000004 00000003 00000002 00000001 00000000 ffffffff 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

图 1-12 内存数据

1.6 实验总结

本次实验主要完成了如下几点工作：

- 1) 完成方案总结：本次实验实现了 MIPS 指令译码器设计、支持中断的微程序入口查找逻辑、支持中断的微程序条件判别测试逻辑、支持中断的微程序控制器设计、支持中断的微程序单总线 CPU 设计、支持中断的现代硬布线控制器状态机设计、支持中断的现代时序硬布线控制器设计。本次试验是在“单总线 CPU 设计（现代时序）”的基础上完成的，添加了中断机制。
- 2) 功能总结：本次实验实现的 CPU 可以按照微程序或硬布线的方式来处理数据，并且支持中断。

1.7 实验心得

- 1) 对 logisim 的使用进一步熟练，如利用表达式生成组合逻辑电路，一些常用的快捷键等功能，使设计电路的效率有所提升。
- 2) 通过本次实验，我对课本知识的理解更加深刻透彻，对 CPU 的结构和运行原理更加清晰，并且对硬布线和微程序两者的优势不足和不同之处认识更加充分。总的来说本次实验帮助我巩固了课本知识，激发了我对计算机硬件设计的兴趣，为日后的学习打下基础。

一、原创性声明

本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。

特此声明!

作者签字:

叶俊红

二、对课程实验的学术评语（教师填写）

三、对课程实验的评分（教师填写）

评分项目 (分值)	课程目标 1 工具应用 (10 分)	课程目标 2 设计实现 (70 分)	课程目标 3 验收与报告 (20 分)	最终评定 (100 分)
得分				

指导教师签字: _____