

华中科技大学

2022

计算机组成原理

· 实验报告 ·

专 业： 计算机科学与技术

班 级： CS2003 班

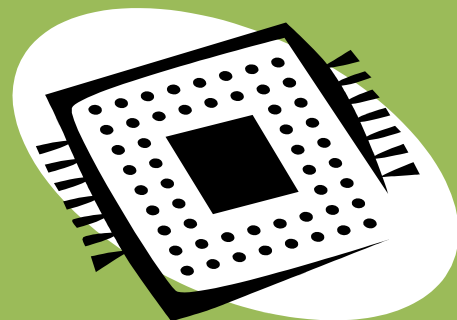
学 号： U202015375

姓 名： 汪宇飞

电 话： 15926575495

邮 件： 2453253921@qq.com

完成日期： 2022-07-05



计算机科学与技术学院

目 录

1 CPU 设计实验	2
1.1 设计要求	2
1.2 方案设计	2
1.2.1 MIPS 指令译码器设计	2
1.2.2 支持中断的微程序入口查找逻辑	3
1.2.3 支持中断的微程序条件判别测试逻辑	4
1.2.4 支持中断的微程序控制器设计	5
1.2.5 支持中断的微程序单总线 CPU 设计	6
1.2.6 支持中断的现代时序硬布线控制器状态机设计	7
1.2.7 现代时序硬布线控制器设计	8
1.3 实验步骤	8
1.4 故障与调试	8
1.4.1 硬布线状态机问题	8
1.5 测试与分析	9
1.6 实验总结	10
1.7 实验心得	10

1 CPU 设计实验

1.1 设计要求

根据实验要求、使用所提供的 circ 文件和 excel 表格、利用 logisim 平台中现有运算部件构建一个支持中断的微程序控制器，要求支持 lw、sw、beq、slt 和 addi 指令，并且能够支持中断响应，指令功能如表 1-1 所示。

表 1-1 需要实现的指令功能

名称	MIPS 指令	RTL 功能描述	说明
lw	lw rt,imm(rs)	$R[rt] \leftarrow M[R[rs] + \text{SignExt}(imm)]$	
sw	sw rt,imm(rs)	$M[R[rs] + \text{SignExt}(imm)] \leftarrow R[rt]$	
beq	beq rs,rt,imm	if($R[rs] = R[rt]$) $PC \leftarrow PC + \text{SignExt}(imm) \ll 2$	
slt	slt rd,rs,rt	$R[rd] \leftarrow R[rs] < R[rt]$	有符号比较
addi	addi rt,rs,imm	$R[rt] \leftarrow R[rs] + \text{SignExt}(imm)$	不考虑溢出
eret	eret	$PC \leftarrow EPC, IE \leftarrow 1$	

基于 educoder 上的实验指引，设计硬布线控制器内部逻辑，实现支持中断的现代时序单总线结构的 MIPS CPU，并且能够最终调试运行标准测试程序 sort-5-int.hex。

1.2 方案设计

1.2.1 MIPS 指令译码器设计

利用比较器等功能模块将 32 位 MIPS 指令字译码生成 LW、SW、BEQ、SLT、ADDI、OtherInstr 信号，即使用多个比较器与输入信号 MIPS 指令字 IR 中的 OP 字段与各个指令所对应的 OP 相比较，相同则输出对应的输出信号。其中对于 slt 指令的判断需要在比较 OP 的同时也比较 FUNCT 字段。

根据分析并结合各个指令的 OP，所设计得出的电路如图 1-1 所示。

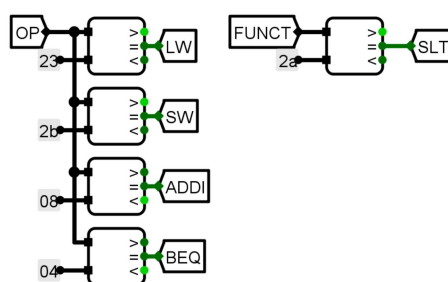


图 1-1 MIPS 指令译码器

1.2.2 支持中断的微程序入口查找逻辑

设计如图 1-2 所示的电路，根据指令译码信号生成 5 位的微程序入口地址。

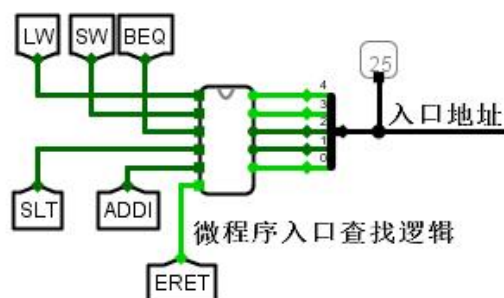


图 1-2 微程序入口查找逻辑电路

地址转移逻辑如图 1-3 所示。

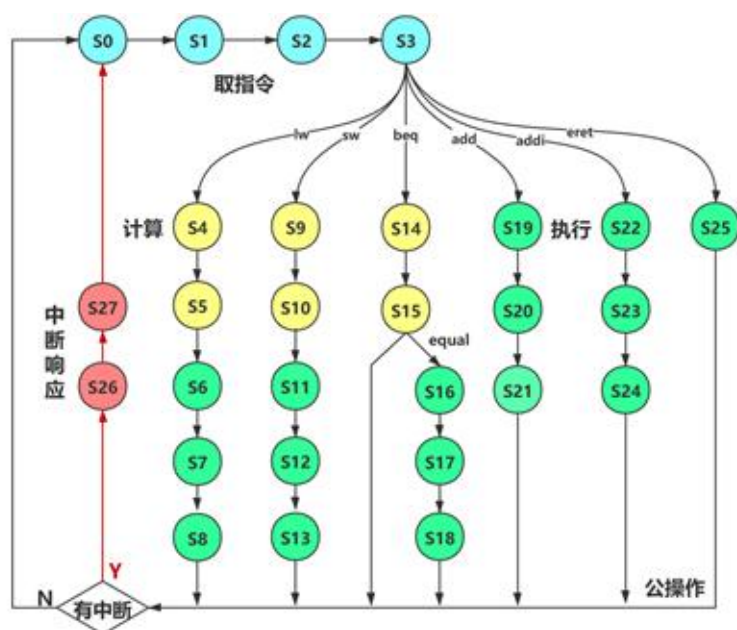


图 1-3 地址转移逻辑

华中科技大学课程实验报告

填写所提供的 6 号表格中的微程序入口地址表格，由图 1-3 中可以得出 lw、sw、beq、slt、addi 和 eret 指令的入口地址分别为 4、9、14、19、22 和 25，填入表格如图 1-4 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 1-4 微程序入口地址表格

将表格自动生成的表达式使用 logisim 的分析组合逻辑电路功能即可自动生成电路。

1.2.3 支持中断的微程序条件判别测试逻辑

根据微指令字中的判别测试字段和条件反馈信息生成后续地址的多路选择信号，实现对应组合逻辑。

P0、P1 和 P2 为输入判别测试位。P0 为 1 时需要根据指令功能进行微程序分支，P1 为 1 时需要根据 equal 标志进行微程序分支，P2 为 1 时表示是当前微程序的最后一条微指令。

equal 为输入条件状态位，为 1 表示运算相等。IntR 为 1 时表示存在中断请求，需要转中断响应微程序执行。

根据上述分析填写所提供的 4 号表格中的组合逻辑真值表，如图 1-5 所示。

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

图 1-5 组合逻辑真值表

华中科技大学课程实验报告

将表格自动生成的表达式使用 logisim 的分析组合逻辑电路功能自动生成电路。

1.2.4 支持中断的微程序控制器设计

根据指令操作以及实验指引，填写 6 号表格中的微程序自动生成表，得出所对应的 16 进制微指令，如图 1-6 所示。

微指令功能	PCout	Dout	Zout	Rout	RtZero	RtOverflow	DRZero	PCin	ARin	DRin	Xin	Rin	IRin	PSWin	Rz/Rt	Reg/Rt	Add	Add4	Stt	READ	WRITE	PCOut	EPCin	Addout	STI	CLI	P1	P2	P3	微指令	微指令十六进制
取指令	0	1							1			1							1										10000000100100000000000000000000	20240000	
取指令	1																												00000000000000000000000000000000	800	
取指令	2			1					1	1										1									00100001010000000000100000000000	8500200	
取指令	3			1									1														1		0100000000000100000000000000100	10010004	
lw	4				1						1																		00010000000100000000000000000000	4040000	
	5					1												1											00001000000000000000000000000000	2001000	
	6			1						1																			00100000100000000000000000000000	8200000	
	7										1										1								00000000010000000000010000000000	100200	
	8			1									1															1	01000000000100000000000000000001	10020001	
sw	9				1						1																		00010000000100000000000000000000	4040000	
	10					1												1											00001000000000000000000000000000	2001000	
	11			1						1																			00100000100000000000000000000000	8200000	
	12				1						1					1													00010000001000010000000000000000	4084000	
	13						1														1						1		00000010000000000000000000000001	800101	
beq	14				1						1																		00010000000100000000000000000000	4040000	
	15				1								1	1														1	1	00010000000000011000000000000011	400C003
	16	1									1																		10000000000100000000000000000000	20040000	
	17					1													1										00000100000000000000000000000000	1001000	
	18			1						1																	1		00100001000000000000000000000001	8400001	
slt	19				1						1																		00010000000100000000000000000000	4040000	
	20				1										1					1									00010000000000010001000000000000	4004400	
	21			1								1					1											1	00100000000100010000000000000001	8022001	
addi	22				1						1																		00010000000100000000000000000000	4040000	
	23					1												1											00001000000000000000000000000000	2001000	
	24			1									1														1		00100000000100000000000000000001	8020001	
eret	25								1														1			1			00000001000000000000000000000001	400091	
	26	1																						1		1			10000000000000000000000000000001	20000048	
	27								1																	1		1	00000001000000000000000000000001	400021	

图 1-6 微程序自动生成表

将所得到的微指令存入到对应的控制存储器中，如图 1-7 所示。

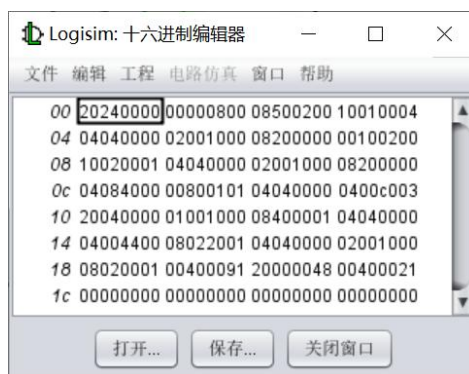


图 1-7 控制存储器

将前面所设计的微程序入口查找逻辑、判别测试逻辑、控制存储器等部件进行适当连接，实现微程序控制器的主要数据通路。根据功能，判别测试逻辑所控制的多路选择器的前四个输入端从 0 至 3 分别为：顺序地址即下一状态、微程序入口查找逻辑的输出即微程序入口、beq 分支 equal 为 1 时的 0x10、中断响应入口的 0x1a。

对于寄存器，应设置为下降沿有效。

华中科技大学课程实验报告

根据分析所得电路如图 1-8 所示。

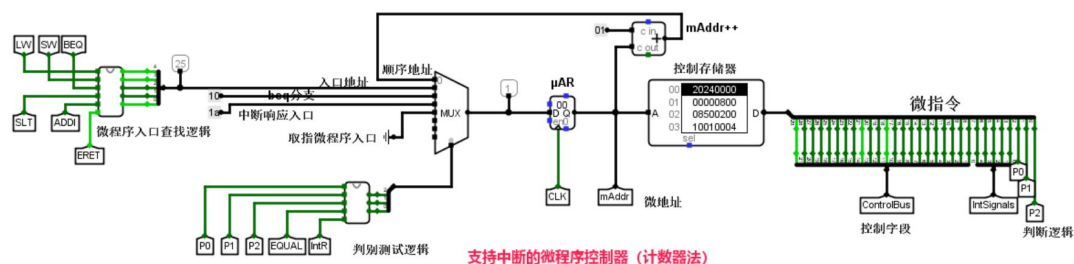


图 1-8 支持中断的微程序控制器

1.2.5 支持中断的微程序单总线 CPU 设计

根据实验指引，打开 Mars4_5.jar 文件，打开 sort-5-int.asm 文件，将 Memory Configuration 更改为 Compact,data at address 0 后进行 Run 中的 Assemble, 点击 Execute 查看。

在 Text Segment 部分查找中断程序入口指令 `addi $sp,$sp,8` 所对应的地址，一共有两个，分别为 0x000030a4 和 0x000030ec，分别如图 1-9 和图 1-10 所示。

0x000030a0	0x1000ffff	beq \$0,\$0,0xffffffff	54: beq \$0,\$0,ProgramEnd #死循环
0x000030a4	0x23bd0008	addi \$29,\$29,0x00000008	57: addi \$sp,\$sp,8 #push registers 需要保留哪些寄存器? 中断程序用到的寄存器
0x000030a8	0xafb00000	sw \$16,0x00000000(\$29)	58: sw \$s0,0(\$sp)
0x000030ac	0xafb10004	sw \$17,0x00000004(\$29)	59: sw \$s1,4(\$sp)
0x000030b0	0x20110240	addi \$17,\$0,0x00000240	61: addi \$s1,\$0,0x240
0x000030b4	0x8e300000	lw \$16,0x00000000(\$17)	62: lw \$s0,(\$s1)
0x000030b8	0x22100001	addi \$16,\$16,0x00000001	63: addi \$s0,\$s0,1

图 1-9 0x000030a4

0x000030e8	0x42000018	eret	75: eret
0x000030ec	0x23bd0008	addi \$29,\$29,0x00000008	77: addi \$sp,\$sp,8 #push registers 需要保留哪些寄存器? 中断程序用到的寄存器
0x000030f0	0xafb00000	sw \$16,0x00000000(\$29)	78: sw \$s0,0(\$sp)
0x000030f4	0xafb10004	sw \$17,0x00000004(\$29)	79: sw \$s1,4(\$sp)
0x000030f8	0x20110280	addi \$17,\$0,0x00000280	81: addi \$s1,\$0,0x280
0x000030fc	0x8e300000	lw \$16,0x00000000(\$17)	82: lw \$s0,(\$s1)
0x00003100	0x2210ffff	addi \$16,\$16,0xffffffff	83: addi \$s0,\$s0,-1

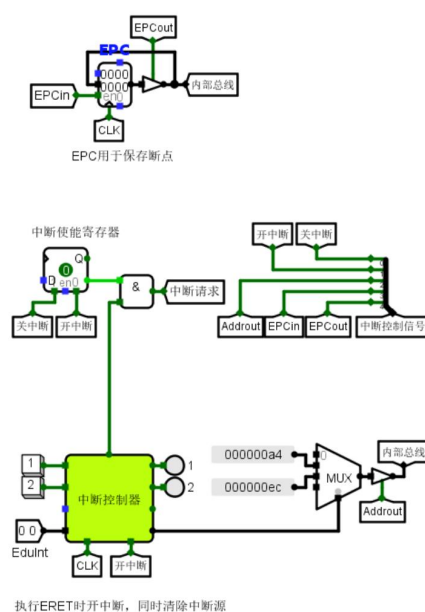
图 1-10 0x000030ec

得出地址后可以设计中断逻辑，根据中断控制器的输出来选择相对应的地址。

对于保存断点的 EPC，寄存器的输入和输出都连接内部总线，输入使用 EPCin 控制中断地址的输入，输出使用 EPCout 控制的三态门实现中断地址的恢复。

根据上述分析实现中断逻辑，电路图如图 1-11 所示。

华中科技大学课程实验报告



中断逻辑

图 1-11 中断逻辑

1.2.6 支持中断的现代时序硬布线控制器状态机设计

按状态图填写 5 号 excel 表, 在 IR 为 1 时次态应为 26 即进入中断。对于状态 15, 应先判断 equal 是否为 1, 是则进入状态 16, 否则判断 IR。填写完成的表格如图 1-12 所示, 自动生成次态逻辑表达式后, 即可在 logisim 中自动生成该电路。

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	LR	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	1	0	0	4									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							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							0		0	0	0	0	0	0	
0	1	1	1	0	14									15	0	0	1	1	1	
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							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							0		0	0	0	0	0	0	
1	0	1	1	0	22									23	1	0	1	1	0	
1	0	1	1	1	23									24	1	1	0	0	0	
1	1	0	0	0	24							0		0	0	0	0	0	0	
0	0	0	1	1	3						1			25	1	1	0	0	1	
1	1	0	1	0	26									27	1	1	0	1	1	
1	1	0	1	1	27									0	0	0	0	0	0	
0	1	0	0	0	8							1		26	1	1	0	1	0	
0	1	1	0	1	13								1	26	1	1	0	1	0	
0	1	1	1	1	15							1	0	26	1	1	0	1	0	
1	0	0	1	0	18								1	26	1	1	0	1	0	
1	0	1	0	1	21								1	26	1	1	0	1	0	
1	1	0	0	0	24								1	26	1	1	0	1	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	

图 1-12 状态机逻辑生成表

1.2.7 现代时序硬布线控制器设计

在实现指令译码、现代时序状态机模块后，最终实现硬布线控制器的集成，根据功能设计硬布线控制器。硬布线控制器状态机输出次态给状态寄存器，状态寄存器输出现态给状态机。

将 1.2.4 中所得的 16 进制微指令存入控制存储器。对于状态寄存器，应设置为下降沿有效。最终电路如图 1-13 所示。

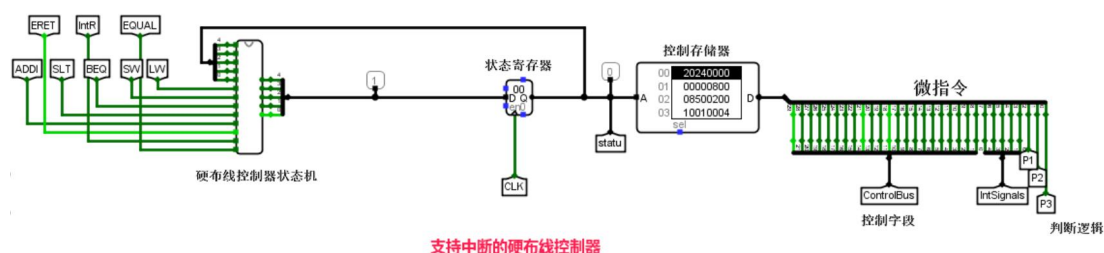


图 1-13 支持中断的现代时序硬布线控制器状态机

1.3 实验步骤

- (1) 根据实验指引以及课程所学知识，设计上述解决方案，使用所提供的 excel 表格和 circ 文件，利用 logisim 工具实现所设计的解决方案。
- (2) 对照 educoder 所提供的测试用例及输出结果说明，对所设计并实现的电路进行测试与验证并作出相应调整。
- (3) 使用记事本打开设计好的 circ 文件，复制粘贴至 educoder 上进行测试，根据测试结果作出分析并进行进一步的调整与修改。

1.4 故障与调试

1.4.1 硬布线状态机问题

故障现象：硬布线状态机出现错误，无法通过 educoder 测试。

原因分析：在填写状态机逻辑生成表时，最初未注意到 IR 和 EQUAL 信号出现情况的不同组合，为在表格中填入 0 而只填入了 1，导致在 logisim 中所生成的硬布线状态机出现问题，影响了测试。

华中科技大学课程实验报告

解决方案：根据 IR 和 EQUAL 所出现的不同组合，按照状态图填入对应的 0 和 1 组合以及对应的次态，使状态机能够顺利按照预期工作。

1.5 测试与分析

根据实验要求，在带有中断功能的微程序单总线 CPU 的 MEM 中载入镜像文件 sort-5-int.hex，使用 ctrl+k 开始运行，最终结果如图 1-14 所示。



图 1-14 最终运行结果

程序运行至 1992 节拍停下，指令计数为 252，最后一条指令是一条 beq 分支指令，会跳回当前指令继续执行，是死循环。完成了有符号数的降序排序，MEM 中排序结果如图 1-15 所示。

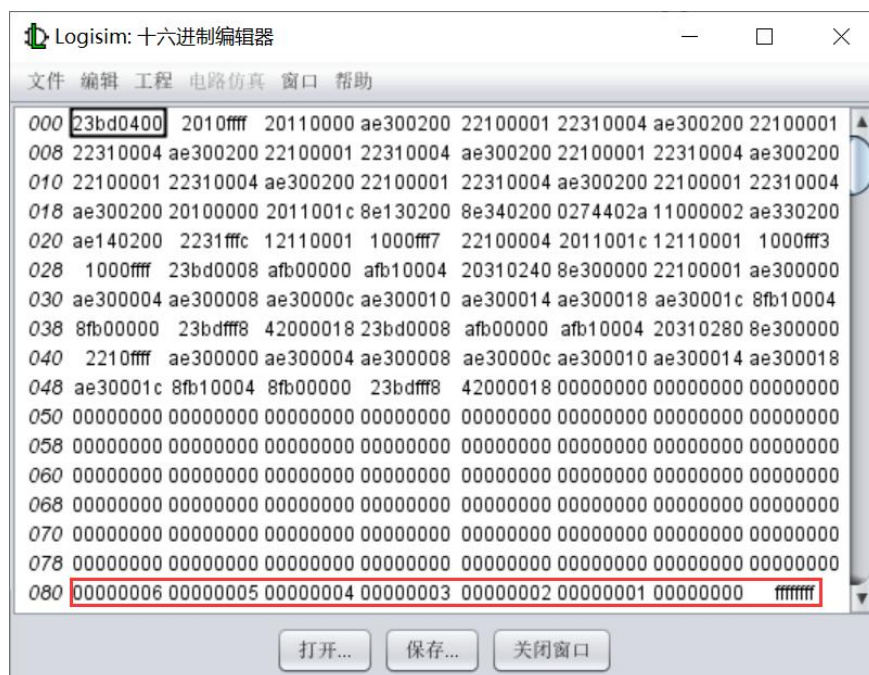


图 1-15 内存 MEM 中有符号数降序排序结果

1.6 实验总结

本次实验主要完成了如下几点工作：

- 1) 设计了支持中断的微程序控制器，实现了支持中断的现代时序单总线结构的 MIPS CPU。
- 2) 设计了现代时序的微程序指令，添加了中断处理的部分，实现了 CPU 中断功能的添加，完成了对标准测试程序 `sort-5-int.hex` 的调试和运行，达到了所预期的目标。

1.7 实验心得

- 1) 本次实验是 MIPS 现代时序中断机制实现，是在已经完成了现代时序的单总线 CPU 设计后的进阶，添加了中断的功能。通过本次的实验，我对 CPU 的组成和工作原理有了更加清晰的了解与认识，诸如 CPU 的各个内部部件的功能及其设计与实现、各个部件之间的相互联系与连接、CPU 在这些部件的基础上运行的过程等等一系列知识。
- 2) 对于 CPU 的中断功能的实现也有了进一步的认识。对于处理单级中断的流程有了更深的认知，对于关中断、保存断点、执行中断以及开中断这一系列操作的具体流程有了更深刻的认知。
- 3) 进一步熟悉了使用 logisim 进行设计的操作，了解并使用了如译码器等各个部件，了解了使用 logisim 进行调试运行 hex 文件的操作。
- 4) 认识了 Mars4_5.jar 工具的使用，了解了使用该工具对 asm 文件的操作。
- 5) 进一步熟悉了 lw 等数条指令的操作，深入理解了这些指令各个操作的具体在 CPU 中的流程。
- 6) 本次实验的指导清晰明确、简明扼要，能够让人快速上手实验操作，对于使用 excel 表格生成表达式和微指令也为实验带来了极大的便利，同时也能使学生在填写表格的同时能够进一步体会与熟悉相关的知识点。
- 7) 略微让人迷惑的是对于 Mars4_5.jar 的使用以及对于 `sort-5-int.hex` 文件的调试运行的说明与指引较少，可以对这两方面稍微增加一些指导。

• 指导教师评定意见 •

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：汪宇飞 汪宇飞

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: _____