

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

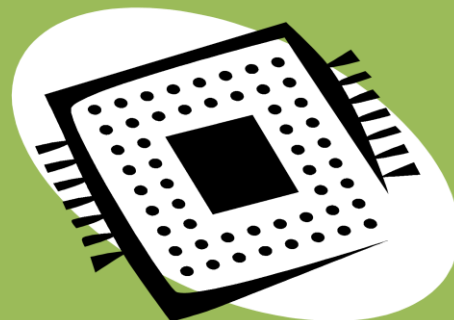
学 号： U202215357

姓 名： 王文涛

电 话： 13607252896

邮 件： 2380169004@qq.com

完成日期： 2024-06-22



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 平台中现有运算部件设计支持中断的微程序控制器的单总线 CPU，使得 MIPS 程序能在单总线结构上运行，最终能运行简单的排序程序 sort-5-int.hex。运算器功能以及输入输出引脚见下表，在主电路中详细测试自己封装的运算器。

1.2 方案设计

1.2.1 MIPS 指令译码器设计

指令译码器将指令字翻译成一根根的指令译码信号，将 32 位 MIPS 指令字译码生成 LW、SW、BEQ、SLT、ADDI、OtherInstr 信号。

对于 LW、SW、BEQ、ADDI 等 I 型指令，可以通过操作码 OP 区分。

指令	格式	OP	rs	rt	rd	shamt	funct
addi	I	8	Reg	Reg	16bits 立即数		
lw	I	35	Reg	Reg	16bits 立即数		
sw	I	43	Reg	Reg	16bits 立即数		
beq	I	4	Reg	Reg	16bits 立即数 (相对寻址)		

图 1-1 I 型指令机器码

SLT 为 R 型指令，OP 皆为 0，可以根据扩展操作码 funct 区分。

指令	格式	OP	rs	rt	rd	shamt	funct
slt	R	0	Reg	Reg	Reg	0	$2a_{16}$

图 1-2 R 型指令机器码

使用比较器即可根据指令的 OP 和 funct 生成对应的指令译码信号，若不是 LW、SW、BEQ、SLT、ADDI 信号则为 OtherInstr 信号。

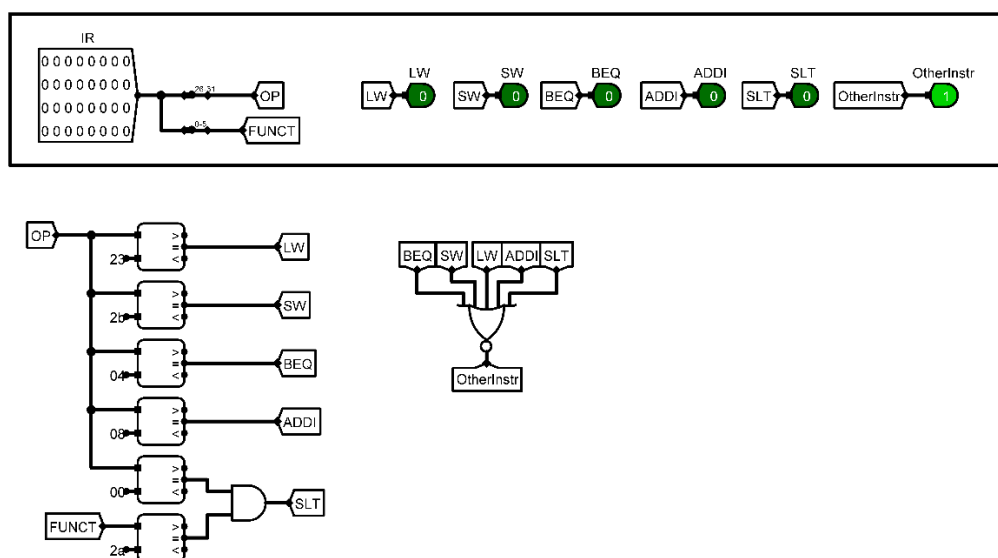


图 1-3 指令译码器电路图

1.2.2 支持中断的微程序入口查找逻辑

微程序按照取指令，LW、SW、BEQ、SLT、ADDI、ERET 的顺序放置在控制存储器中。由于取指令需要 4 个节拍，LW 指令地址从 4 开始。LW、SW、BEQ 需要 5 个节拍，SLT、ADDI 需要 3 个节拍，所以入口地址分别为 9、14、19、22、25。填写 6 号 EXCEL 表格中的微程序入口地址表格如下图所示。

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

将生成的逻辑表达式复制到 logisim 中。得到电路图如下。

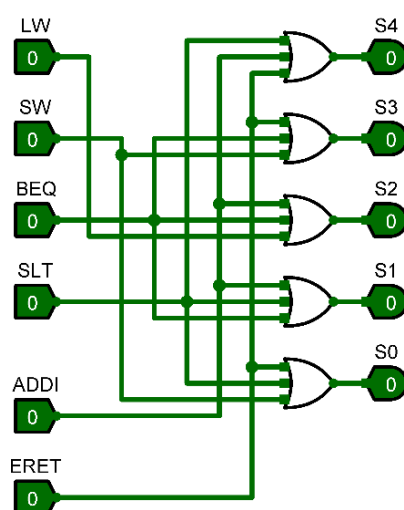


图 1-5 微程序入口查找逻辑电路图

1.2.3 支持中断的微程序条件判别测试逻辑

表 1-1 引脚功能

引脚	输入/输出	位宽	功能描述
P0	输入	1	条件测试位，根据指令功能进行微程序分支 P_{IR}
P1	输入	1	条件测试位,根据 PSW.equal 标记进行分支 P_{equal}
P2	输入	1	条件测试位，表示是最后一条微指令，需要判断中断请求 IntR
equal	输入	1	相等标记
intR	输入	1	中断请求信号
S3~S0	输出	3	后续地址多路选择控制信号

若 $P0=0$, $P1=0$, $P2=0$, 表示按顺序取地址，地址加一，多路选择控制信号为 0。

若 $P0=1$, $P1=0$, $P2=0$, 表示取址周期结束，下一步取微指令的入口地址，多路选择控制信号为 1。

若 $P0=0$, $P1=1$, $P2=1$, 表示 beq 指令计算周期结束，需要判断 equal 标记和 intR 信号。若 equal=1,无论 intR 为何值，都需要先跳转到 beq 分支。多路选择控制信号为 2。若 equal=0, intR=0, 则跳转到取址微指令入口重新取址，多路选择控制信号为 4。若 equal=0, intR=1, 则跳转到中断处理程序入口，多路选择控制信号为 3。

若 $P0=0$, $P1=0$, $P2=1$, 表示指令执行周期结束，需要判断 intR 信号。若 intR=0,

华中科技大学课程实验报告

则重新进入取址周期，多路选择控制信号为 4。若 $\text{intR}=1$ ，则跳转到中断处理程序入口，多路选择控制信号为 3。

根据上述分析填写 4 号 EXCEL 表格中的组合逻辑真值表如下图。

P0	P1	P2	equal	IntR	S2	S1	S0
0	0	0					
1	0	0					1
0	1	0	1			1	
0	1	0	0		1		
0	1	1	1			1	
0	1	1	0	1		1	1
0	1	1	0	0	1		
0	0	1		1		1	1
0	0	1		0	1		

图 1-6 判别测试逻辑

将得到的逻辑表达式导入电路，得到如下电路图。

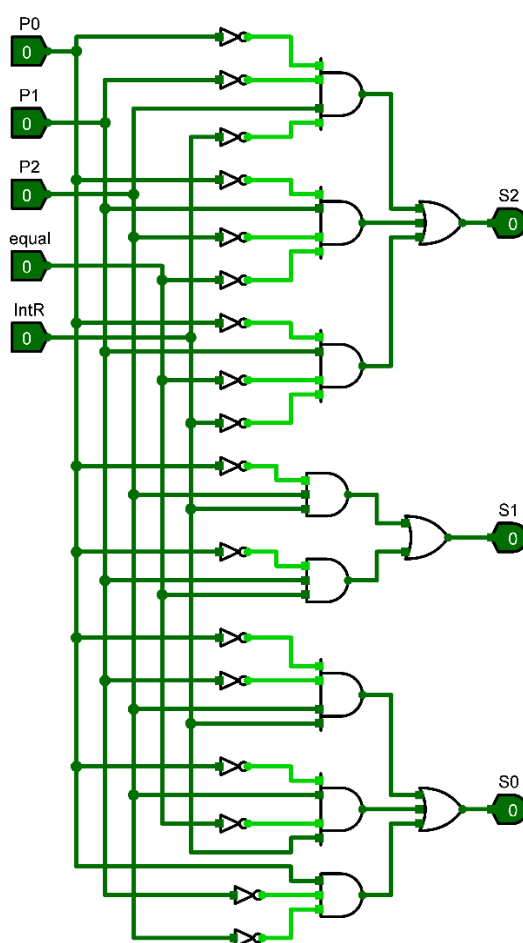


图 1-7 微程序条件判别测试逻辑电路图

华中科技大学课程实验报告

1.2.4 支持中断的微程序控制器设计

首先填写 6 号 EXCEL 文件中的微程序自动生成表。

根据下图填写取指令周期控制信号，其中 T4 取址周期结束，下一节拍取微指令的入口地址，P1=1。

节拍	控制信号(4 cycles)
T1	PC _{out} , AR _{in} , X _{in}
T2	+4
T3	Z _{out} , PC _{in} , DRE _{in} , Read
T4	DR _{out} , IR _{in}

图 1-8 取指令周期控制信号

根据下图填写取 Lw、Sw、Beq、Slr、Addi 控制信号，在每个指令的最后一个节拍 P3=1，表示指令结束。

其中在 Beq 的第二节拍计算周期结束时需要根据 equal 的值判断是否跳转到 Beq 执行周期分支，所以 P2=1，P3=1。

节拍	lw (5 cycles)	sw (5 cycles)	beq (5 cycles)	add (3 cycles)	addi (3 cycles)
T1	R _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}		
T2	IR(I) _{out} , ADD	IR(I) _{out} , ADD	R _{out} , Rs/Rt, SUB, PSW _{in}		
T1	Z _{out} , AR _{in}	Z _{out} , AR _{in}	PC _{out} , X _{in}	R _{out} , X _{in}	R _{out} , X _{in}
T2	DRE _{in} , Read	R _{out} , Rs/Rt, DR _{in}	IR(A) _{out} , ADD	Rs/Rt, R _{out} , ADD	IR(I) _{out} , ADD
T3	DR _{out} , R _{in}	DRE _{out} , Write	Z _{out} , PC _{in} =PSW.equal	Z _{out} , R _{in} , RegDst	Z _{out} , R _{in}

图 1-9 指令控制信号

Eret 指令只有一个节拍。根据下图填写，其中 STI、EPC_{out}为中断控制信号。P3=1。

节拍	控制信号(控制流)
M _{ex} , T1	STI, EPC _{out} , PC _{in}

图 1-10 eret 指令控制信号

华中科技大学课程实验报告

接下来两个微地址存放中断响应微指令。根据下图填写。P 字段均为 0。

节拍	控制信号(控制流)
$M_{int}, T1$	CLI, PC_{out}, EPC_{in}
$M_{int}, T2$	$IntA_{out}, PC_{in}$

图 1-11 中断响应控制信号

填写 6 号 EXCEL 文件中的微程序自动生成表（未截取生成的微指令）如下。

微指令功能	PCout	DRout	Zout	Rout	WZout	WZinout	DRout	PCin	ARin	DREin	DRin	Xin	Rin	IRin	PSWin	Rs/Rt	RegOut	Add	Add4	Slt	READ	WRITE	EPCout	EPCin	Adinout	STI	CLI	P1	P2	P3
取指令	0	1							1			1																		
取指令	1																	1												
取指令	2			1					1		1										1									
取指令	3		1											1														1		
Lw	4				1							1																		
	5					1												1												
	6			1					1																					
	7									1											1									
	8		1										1																	
Sw	9				1							1																	1	
	10					1												1												
	11			1					1																					
	12				1						1					1							1							
	13						1																							1
Beq	14				1							1																		
	15				1										1	1													1	1
	16		1									1																		
	17					1												1												
	18			1				1																						1
Slt	19				1							1																		
	20				1											1			1											
	21			1									1				1													1
Addi	22				1							1						1												
	23					1													1											
	24			1									1																	1
Eret	25							1															1			1				1
	26		1																					1			1			
	27							1																	1					

图 1-12 微程序自动生成表

将微指令十六进制编码直接复制粘贴到电路图的 ROM 中。

接着完善电路图。将加法器的输出（mAddr++）连到顺序地址处，表示指令地址加一。将微程序入口查找逻辑输出连接到入口地址。根据微程序可知 beq 分支地址为 0x10，中断响应入口为 0x1a，取址微程序入口为 0。

将判别测试逻辑的输出使用分线器（注意分线器从上到下端口为 2，1，0）连接到 MUX 的选择端。

最终电路图如下。

[illegible]

1.2.5 支持中断的微程序单总线 CPU 设计

Label	Address ▲
sort-5-int.asm	
sort_init	0x00000000
sort_loop	0x0000006d
sort_next	0x00000084
sort_next1	0x00000090
ProgramEnd	0x000000a0
IntProgram1	0x000000a4
IntProgram2	0x000000e0

电路图只需根据中断异常处理数据通路，补充中断逻辑部分即可。

EPC 寄存器输入端口直接与内部总线相连，输出端口通过三态门与内部总线相连，由 EPC_{out} 信号控制。使能端由 EPC_{in} 控制。

中断使能寄存器只由开中断和关中断信号直接控制，状态的非值与中断控制器的输出（intR）通过与门连接，输出中断请求。关中断后，中断请求始终为 1，起到了屏蔽中断请求的效果。

中断控制器的中断号作为选择，将终端号对应的中断处理程序入口地址输出到内部总线。

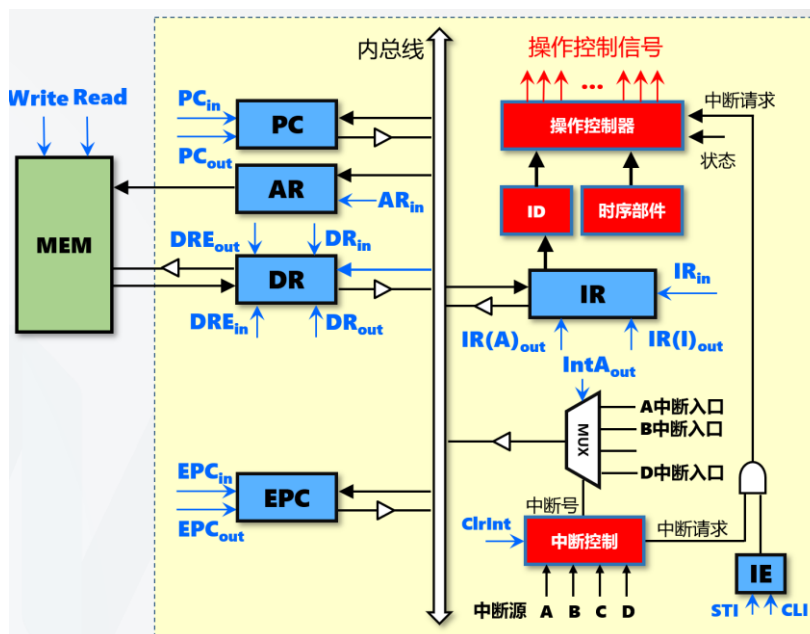


图 1-15 中断异常处理数据通路

最终电路图如下图所示。

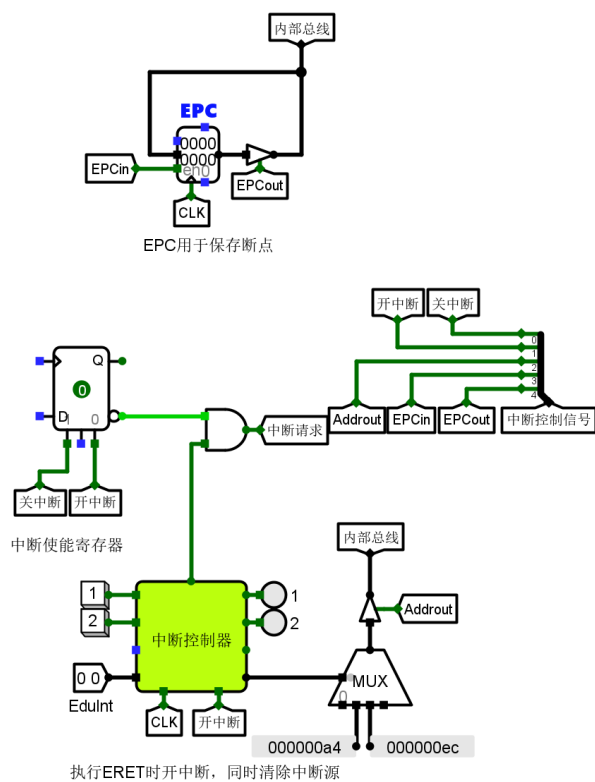


图 1-16 单总线 CPU 电路图

1.2.6 支持中断的现代时序硬布线控制器状态机设计

根据下图所示的现代时序状态机可知，一共存在 27 个状态。

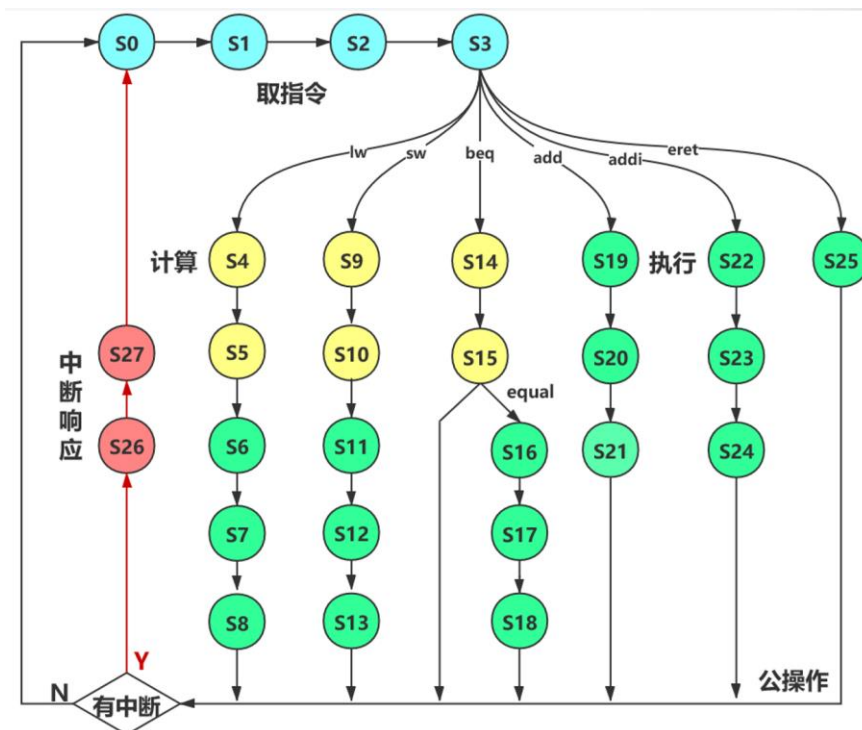


图 1-17 现代时序状态机

状态在取指令周期，计算周期，执行周期和中断响应周期中分别自然递增，无需信号。

在取指令周期的最后一个状态 S3 时，根据指令信号 lw、sw、beq、add、addi、eret 的值决定进入 S4、S9、S14、S19、S22、S25 中的一个。

在执行周期的最后一个状态 S8、S13、S15、S18、S21、S24、S25 时。若中断信号 IR 的值为 1 则进入中断响应 S26，否则直接进入取指令周期 S0。

特别的，进入 beq 分支的 S15 状态时，需要同时根据 equal 信号和 IR 信号决定下一状态。若 equal=1，则进入执行周期中的 S16。若 equal=0，IR=1，则进入中断响应周期 S26。若 equal=0，IR=0，则直接进入取指令周期 S0。

根据上述分析。填写 5 号 excel 表如下。

华中科技大学课程实验报告

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									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	0	8							1		26	1	1	0	1	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	0	1	13							1		26	1	1	0	1	0
0	1	1	1	0	14									15	0	1	1	1	1
0	1	1	1	1	15								1	16	1	0	0	0	0
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
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	0	18							1		26	1	1	0	1	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	0	1	21							1		26	1	1	0	1	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							0		0	0	0	0	0	0
1	1	0	0	0	24							1		26	1	1	0	1	0
1	1	0	0	1	25							0		0	0	0	0	0	0
1	1	0	0	1	25							1		26	1	1	0	1	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 1-18 单总线 MIPS 硬布线控制器状态机逻辑自动生成表

将表格自动生成的次态逻辑表达式导入电路图。生成最终生成电路图。由于电路图过大，不在此处贴出。

1.2.7 支持中断的现代时序硬布线控制器设计

本实验完成硬布线控制器框架连接即可。

由上一个实验可知，硬布线控制器状态机的前四个引脚为现态 S4~S0，所以将状态寄存器的输出通过分线器连接到前四个引脚。

硬布线控制器状态机后面的引脚依次为 LW、SW、BEQ、SLT、ADDI、ERET、IR、EQUAL，将对应引脚连接即可。

硬布线控制器状态机的输出端通过分线器连接到状态寄存器的输入端。

最终的电路图如下。

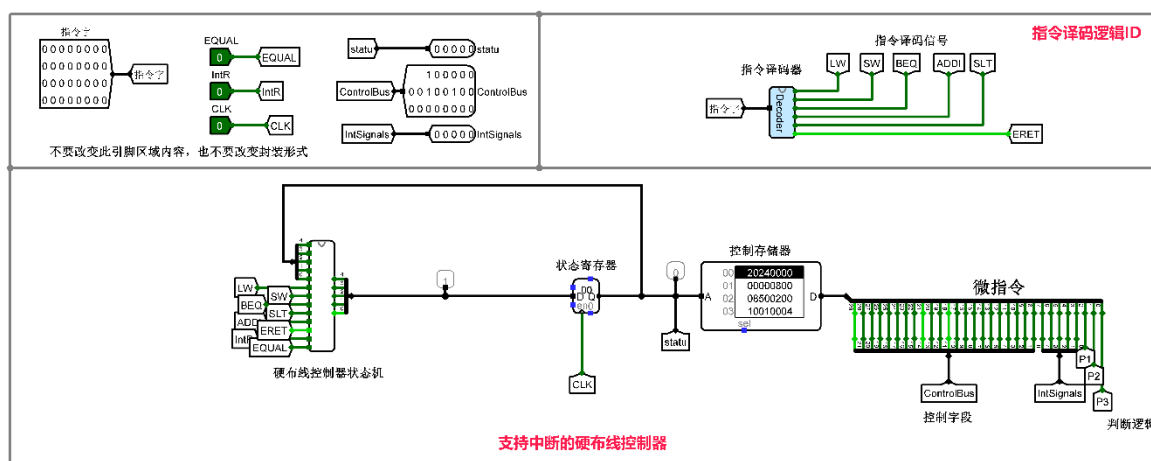


图 1-19 硬布线控制器电路图

1.3 实验步骤

- (1) 下载资料包到本地，使用 logisim 打开 MipsOnBusCpu-1.circ 文件。
- (2) 根据头歌要求，在计算机组成原理 PPT 上查找对应知识。
- (3) 在一些关卡使用提供的 excel 表格，填写完成后将表达式导入对应电路图。
- (4) 在 logisim 上实现最初的版本。
- (5) 在头歌上进行测评，如果没通过，则研究测试集，同时在 logisim 中进行测试，修改电路。
- (6) 如果遇到了无法解决的问题则向同学和老师请教。

1.4 故障与调试

1.4.1 微程序控制器出错

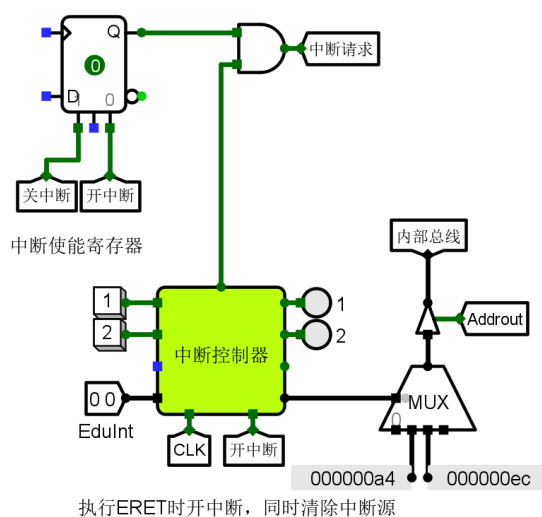
故障现象：在头歌上提交代码时报错。

原因分析：寄存器为上升沿触发，导致经过一个时钟脉冲后，寄存器中的数据仍然没有改变，在下一个时钟脉冲时才改变，导致测试结果出错。

解决方案：将寄存器改为下降沿触发。

1.4.2 接口处数据传输问题

故障现象：进入中断处理程序开中断后，中断请求无法输出。



原因分析：中断使能寄存器应输出低电平以实现在中断处理中屏蔽其他中断。

解决方案：将触发器非值连接到与门。

1.5 测试与分析

测试用例为 sort-5-int.hex 程序，将其载入电路的 RAM 后，RAM 储存内容如下图所示。

华中科技大学课程实验报告

```
000 23bd0400 2010fff 20110000 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200
010 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 20100000 2011001c 8e130200 8e340200 0274402a 11000002 ae330200
020 ae140200 2231fff 12110001 1000fff7 22100004 2011001c 12110001 1000fff3 1000fff 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 2210fff 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 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
090 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0a0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0b0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

图 1-21 CPU 测试用例

1.6 实验总结

设计了基于 MIPS 支持中断的现代时序单总线 CPU，实现了 MIPS 指令译码器，支持中断的微程序入口查找逻辑电路，支持中断的微程序条件判断电路，支持中断的微程序控制器和单总线 CPU，支持中断的现代时序硬布线状态机和控制器。补充了中断功能部件，完成了中断优先级处理和中断识别的逻辑。

1.7 实验心得

- 1) 在完成了计算机组成原理的所有课程设计之后，我对从书本中学习到的知识有了更加深刻的理解，其中最让我印象深刻也对我启发最大的就是 CPU 实验。在刚开始学习这门课时，老师说我们最后会自己设计 CPU，当时我感觉这是一件完全超出自己能力的事情，但是在经过学习计算机组成原理并完成了一系列课程设计之后感觉设计 CPU 也没有那么难。实验与 PPT 是高度相关的，在做实验的过程中，我把 PPT 又认真看了一遍，之前有一些没有完全理解的地方也都搞懂了。由于这次实验可以看到测试集，降低了很大一部分难度，在写微程序自动生成表时，可以直接从测试集中得到各个字段的值。
- 2) 最后是一点建议。如果我没记错的话，课设的资料包是在我们的第一次实验课已经开始一段时间之后才提供给我们的，并且里面有好几个版本的资料。我一开始还因为下错了资料包浪费了很多时间。希望之后能在实验开始前将这些东西说明清楚。并且 CPU 中断实验有一关存在寄存器的 bug 需要更换 logisim 的版本，这个问题在头歌上有说明，但是我在做到这一关才看到，最后我用老版本的 logisim 将前面的实验全部重做一遍才解决（也许这并不是必须的），希望之后老师能提前将这些问题说明一下。

• 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：王文涛 王文涛

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：_____