

# 华中科技大学

## 2021

### 计算机组成原理

### · 实验报告 ·

专    业：        计算机科学与技术

班    级：        ZY1901

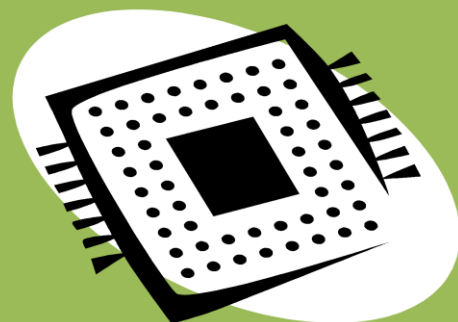
学    号：        U201914869

姓    名：        李宇

电    话：        17720338099

邮    件：        cavalryly@gmail.com

完成日期：        2021-12-14



计算机科学与技术学院

# 华中科技大学课程实验报告

---

## 目 录

<b>1</b>	<b>CPU 设计实验 .....</b>	<b>1</b>
1.1	设计要求 .....	1
1.2	具体方案 .....	2
1.3	实验步骤 .....	11
1.4	故障与调试 .....	12
1.5	测试与分析 .....	13
<b>2</b>	<b>总结与心得 .....</b>	<b>15</b>
2.1	实验总结 .....	15
2.2	实验心得 .....	15
	<b>参考文献 .....</b>	<b>17</b>

## 1 CPU 设计实验

### 1.1 设计要求

#### 1.1.1 单总线 RISC-V CPU 设计（变长指令周期 3 级时序）

要求理解变长指令周期三级时序系统的设计，能利用该时序构造硬布线控制器，支持 5 条典型 RISC-V 指令在单总线 CPU 上运行，最终 CPU 能运行内存冒泡排序。

需要完成的部件有：

- ✧ 指令译码器设计
- ✧ 时序发生器 FSM 设计
- ✧ 时序发生器输出函数设计
- ✧ 硬布线控制器组合逻辑单元
- ✧ 硬布线控制器设计
- ✧ 单总线 CPU 设计

#### 1.1.2 RISC-V 现代时序中断机制实现

要求理解现代时序控制器中断机制的实现原理，能为采用现代时序单总线结构的 RISC-V CPU 增加中断处理机制，可实现多个外部按键中断事件的随机处理，本实验需要完成现代时序微程序控制器的基础上完成，需要增加硬件数据通路，增加中断返回指令 `meret` 的支持，需要中断服务程序配合。

需要完成的部件有：

- ✧ 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

## 1.2 具体方案

### 1.2.1 单总线 RISC-V CPU 设计（变长指令周期 3 级时序）

#### ✧ 指令译码器设计

设计如图 1-1 所示，将输入的 32 位 RISC-V 指令字译码成不同的指令译码信号，每条指令由操作码和操作码扩展唯一确定。因此可以利用比较器比较输入指令的指定位和指令操作码以及其扩展来确定是否输出该指令。

指令扩展码如表 1-1 指令扩展码对应表所示

指令	funct3	opcode
LW	2	0
SW	2	0x08
BEQ	0	0x18
SLT	2	0xc
ADDI	0	0x4

表 1-1 指令扩展码对应表

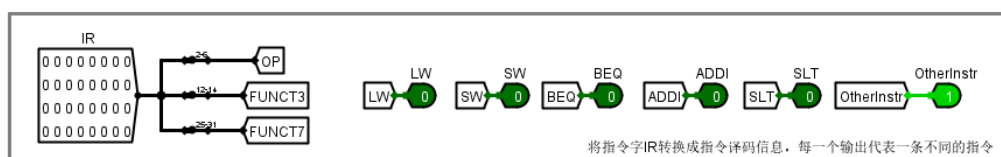


图 1-1 指令译码器设计

# 华中科技大学课程实验报告

## ◇ 时序发生器 FSM 设计

时序发生器主要包括状态寄存器，状态机组合逻辑，输出函数组合逻辑三部分。其中状态机负责现态与次态的转换，输出函数根据当前状态生成状态周期电位和节拍周期电位。

实验中采用变长指令周期，不同指令机器周期数不同，每个机器周期节拍数也是可变化的，通过分析每条指令的周期填写相应 excel 表即可完成所要求。

填写的 excel 如图 1-2 所示，在取指周期的最后一个节拍，根据输入信号的不同进行次态的转移。

当前状态(现态)					输入信号							下一状态 (次态)				
S3	S2	S1	S0	现态 10进制	LW	SW	BEQ	SLT	ADDI	ERET	IntR	次态 10进制	N3	N2	N1	N0
0	0	0	0	0								1	0	0	0	1
0	0	0	1	1								2	0	0	1	0
0	0	1	0	2								3	0	0	1	1
0	1	0	0	4								5	0	1	0	1
0	1	0	1	5								6	0	1	1	0
0	1	1	0	6								7	0	1	1	1
0	1	1	1	7								8	1	0	0	0
1	0	0	0	8								0	0	0	0	0
0	0	1	1	3	1							4	0	1	0	0
0	0	1	1	3		1						4	0	1	0	0
0	0	1	1	3			1					4	0	1	0	0
0	0	1	1	3				1				6	0	1	1	0
0	0	1	1	3					1			6	0	1	1	0

图 1-2 时序 FSM 状态机组合逻辑设计

## ◇ 时序发生器输出函数设计

上述部件已经完成了 FSM 状态机逻辑，要完成完整的时序发生器，还需要完成时序发生器的输出函数设计，根据书上指导以及状态寄存器的现态输入，填写 excel 如图 1-3 所示，得到状态周期点位和节拍电位信号。Mif 为取指周期占 4 个节拍，Mcal 为计算周期占 2 个节拍，Mex 为执行周期占 3 个节拍，不同指令机器周期数不同，其中 beq 指令较为特殊，可能没有执行周期。

# 华中科技大学课程实验报告

当前状态(现态)					输出							
S3	S2	S1	S0	现态 10进制	Mif	Mcal	Mex	Mint	T1	T2	T3	T4
0	0	0	0	0	1				1			
0	0	0	1	1	1					1		
0	0	1	0	2	1						1	
0	0	1	1	3	1							1
0	1	0	0	4		1			1			
0	1	0	1	5		1				1		
0	1	1	0	6			1		1			
0	1	1	1	7			1			1		
1	0	0	0	8			1				1	

图 1-3 时序 FSM 输出函数设计

## ◇ 硬布线控制器组合逻辑单元

微操作控制信号都是反馈信号，指令译码信号，状态周期电位，节拍电位的组合逻辑函数，根据数据通路列出所有微操作信号的产生条件，并根据其填写 excel，如图 1-4 硬布线控制器组合逻辑单元所示。

微指令功能	PCout	DRout	Zout	Rout	DRin	PCin	ARin	DRin	Xin	Rin	IRin	PSW	vs1/2	Add	Alt	Stt	READ	WRITE	P0	P1	P2	下址DEC	微指令	微指令十六进制
取指令 0	1					1			1													1	10000000010010000000000000000001	20120001
取指令 1														1								2	0000000000000000000000000000010	802
取指令 2			1			1	1									1						3	0010000001010000000001000000011	8280203
取指令 3		1								1										1		4	01000000000001000000010000000	10008080
LW 4				1					1													5	0001000000001000000000000000101	4020005
LW 5					1									1								6	000010000000000001000000000110	2001006
LW 6			1				1															7	0010000001000000000000000000111	8100007
LW 7								1									1					8	00000000010000000001000001000	80208
LW 8		1								1												9	0100000000001000000000000000000	10010000
SW 9				1					1													10	0001000000001000000000000001010	402000A
SW 10																1						11	00000100000000000100000001011	100100B
SW 11			1				1															12	00100000010000000000000000001100	810000C
SW 12				1					1					1								13	0001000000100001000000000001101	404200D
SW 13						1												1				14	0000000100000000000001000000000	400100
BEQ 14				1					1													15	0001000000010000000000000001111	402000F
BEQ 15				1								1	1							1		16	0001000000000001100000010000000	4006040
BEQ 16	1								1													17	100000000000100000000000010001	20020011
BEQ 17														1								18	00000010000000000100000010010	801012
BEQ 18			1				1															19	0010000010000000000000000000000	8200000
SLT 19				1					1													20	000100000000100000000000010100	4020014
SLT 20				1									1			1						21	0001000000000000100100000010101	4002415
SLT 21			1							1												22	0010000000001000000000000000000	8010000
ADDI 22				1					1													23	000100000000100000000000010111	4020017
ADDI 23					1									1								24	00001000000000000100000011000	2001018
ADDI 24			1							1												25	0010000000001000000000000000000	8010000

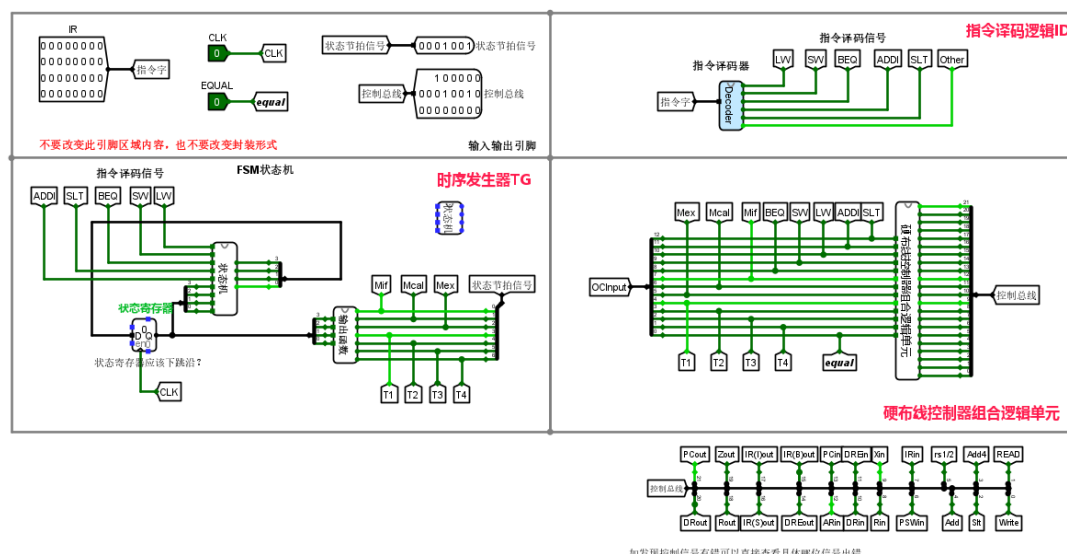
图 1-4 硬布线控制器组合逻辑单元

## ◇ 硬布线控制器设计

硬布线控制器子电路中将时序状态机与输出函数、状态寄存器正确连接，实现时序

# 华中科技大学课程实验报告

发生器逻辑，如图 1-5 所示



至此完成硬布线控制器设计，总结步骤如下：

- 1) 分析指令执行的数据通路，列出每条指令在所有寻址方式下的执行操作流程和每一步所需要的控制信号。
- 2) 对指令的操作流程进行细化，将每条指令的每个微操作分配到具体机器周期的具体时间节拍信号上，即对操作控制信号进行同步控制。
- 3) 根据控制信号同步控制方式构造合适的时序发生器。
- 4) 对每一个控制信号进行逻辑综合，得到每个控制信号的逻辑表达式。在对控制信号进行逻辑综合的过程中，要考虑每一个控制信号在不同指令、不同机器周期和不同节拍的有效情况，不能遗漏，否则对应的指令将由于缺少控制信号而不能正确执行。
- 5) 最后采用逻辑门或 PLA 或 ROM 实现逻辑表达式的功能。

## ◇ 单总线 CPU 设计

正确实现上述部件的功能后，在设计好的 CPU 进行最终联调，测试排序程序能否正常运行，仅需在 ram 中加载编写好的数据映像，然后 ctrl+k 运行即可。

# 华中科技大学课程实验报告

## 1.2.2 单总线 RISC-V CPU 设计（现代时序带中断操作）

### ✧ 指令译码器设计

由于都采用相同的指令集，译码器的设计并没有变化，同 1.2.1 中设计相同。

### ✧ 支持中断的微程序入口查找逻辑

根据指令译码信号生成 5 位的微程序入口地址，相较之前微程序入口查找逻辑，需要增加 `eret` 对应微程序入口查找逻辑，填写 excel 如图 1-6 支持中断的微程序入口查找逻辑所示，即增加 `eret` 指令的微程序入口地址即可，注意机器指令译码信号的表项应与 `logisim` 中分析组合电路自动生成的输入项保持一致，不能有多余的字母或表示。

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 1-6 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

要求根据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，



# 华中科技大学课程实验报告

并根据其生成组合逻辑电路

实验中我们需给出各个判别条件下应该选择的下址，即地址转移控制信号，如 S=0 是下一条地址，默认为当前地址+1，S=1 代表取下一条指令入口地址，S=2 是 beq 指令的 equal 条件成立的情况下的跳转地址，S=3 则是中断处理程序的地址，S=4 表示回到取指令阶段，其地址。注意 S=2 和 S=3 情况下的地址即 1.2.2 微程序入口转移程序的对应值。根据书上指导填写 excel 如图 1-7 所示。

P0	P1	P2	equal	IntR	S2	S1	S0
						0	0
1						0	1
	1		1			1	0
0	1	1	0	1		1	1
0	1	1	0	0	1	0	0
0	0	1	0	1		1	1
0	0	1	1	1		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 1-7 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

完善实验中给的微程序框架，将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，设计微程序并加载到控制存储器中。

微程序地址对应硬布线的时序，可对照硬布线控制信号输出表填写控制信号，生成相应指令字，注意除了增加的指令，之前填写好的指令字也有所变化，不能直接拷贝过去。填写 excel 表如图 1-8 所示。

微指令名称	PCOut	DRIn	Zout	Rout	WZout	DRZout	WDRZout	PCIn	ARin	DRIn	Xin	Rin	IRin	FSRin	SLZ	Add	Addr	St	READ	WRITE	EPControl	EPCIn	Labrout	STI	CLI	P0	P1	P2	微指令	微指令十六进制
取指令	0	1						1			1						1												10000000101000000000000000000000	20120000
取指令	1																1												00000000000000000000000000000000	800
取指令	2			1				1	1										1										00100000101000000000100000000000	8280200
取指令	3	1											1												1				01000000000000001000000000000100	10008004
LW	4			1							1																		00010000000010000000000000000000	4020000
LW	5				1												1												00001000000000000000100000000000	2001000
LW	6			1					1																				00100000010000000000000000000000	8100000
LW	7									1									1										00000000001000000000100000000000	80200
LW	8	1										1														1			01000000000010000000000000000000	10010001
SW	9			1							1						1												00010000000010000000000000000000	4020000
SW	10				1												1												00000100000000000010000000000000	1001000
SW	11			1					1																				00100000010000000000000000000000	8100000
SW	12			1							1						1												00010000000100001000000000000000	4042000
SW	13							1												1						1			000000010000000000001000000001	400101
BEQ	14			1							1																		00010000000010000000000000000000	4020000
BEQ	15			1											1	1										1	1		000100000000000110000000000001	4006003
BEQ	16	1									1																		10000000000100000000000000000000	20020000
BEQ	17					1											1												00000010000000000100000000000000	801000
BEQ	18			1				1																			1		00100000100000000000000000000000	8200001
SLT	19			1							1																		00010000000010000000000000000000	4020000
SLT	20			1												1		1											00010000000000001001000000000000	4002400
SLT	21			1								1</																		

P0 为判别测试位，为 1 表示要根据指令功能进行微程序分支，P1 为判别测试位，为 1 表示要根据 **equal** 标志进行微程序分支，P2 为判别测试位，为 1 表示是微程序的最后一条微指令，可能需要进行中断响应。

类型	值
顺序地址	当前指令地址+1
入口地址	地址转移逻辑部分给出
中断入口地址	26
beq 分支地址	16
取值微程序入口地址	0

在 ram 中加载好数据映像后，根据上述支持中断的微条件判别测试逻辑连接电路，如图 1-9 所示。

---



# 华中科技大学课程实验报告

其中 IE 寄存器在关中断时异步置 1，开中断时异步置 0，输出为寄存器保存内容相反部分。

mEPC 寄存器通过 mEPCin 进行输入信号的控制，通过三态门进行输出信号的控制，只有在 mEPCout 信号有效时才进行输出。

## ◇ 支持中断的现代时序硬布线控制器状态机设计

根据状态转移图填写 excel 表如图 1-11 所示。

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

图 1-11 支持中断的现代时序硬布线控制器状态机设计

# 华中科技大学课程实验报告

## ◇ 支持中断的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，完成硬布线控制器框架连接，加载数据映像如图 1-12 所示，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，因为硬布线的状态是与微程序的地址一一对应的。

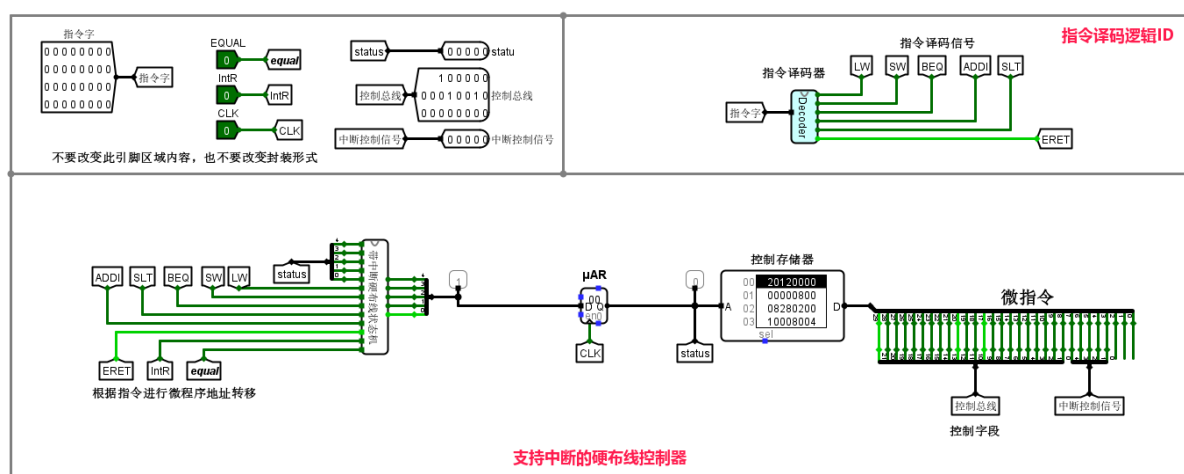


图 1-12 支持中断的现代时序硬布线控制器设计

## 1.3 实验步骤

每个实验的流程大致如下：

- (1) 分析表格/电路
- (2) 填写表格中相应表项
- (3) 自动生成逻辑表达式
- (4) 在 logisim 中利用分析组合逻辑电路功能自动生成电路，或分析后自己手动连接电路
- (5) 进行联调测试。

## 1.4 故障与调试

### 1.4.1 表达式生成问题

**故障现象：**在微程序入口查找逻辑+中断中，利用 logisim 生成组合逻辑电路时出错，无法生成，出错现象如图 1-13 logisim 故障部分截图所示。

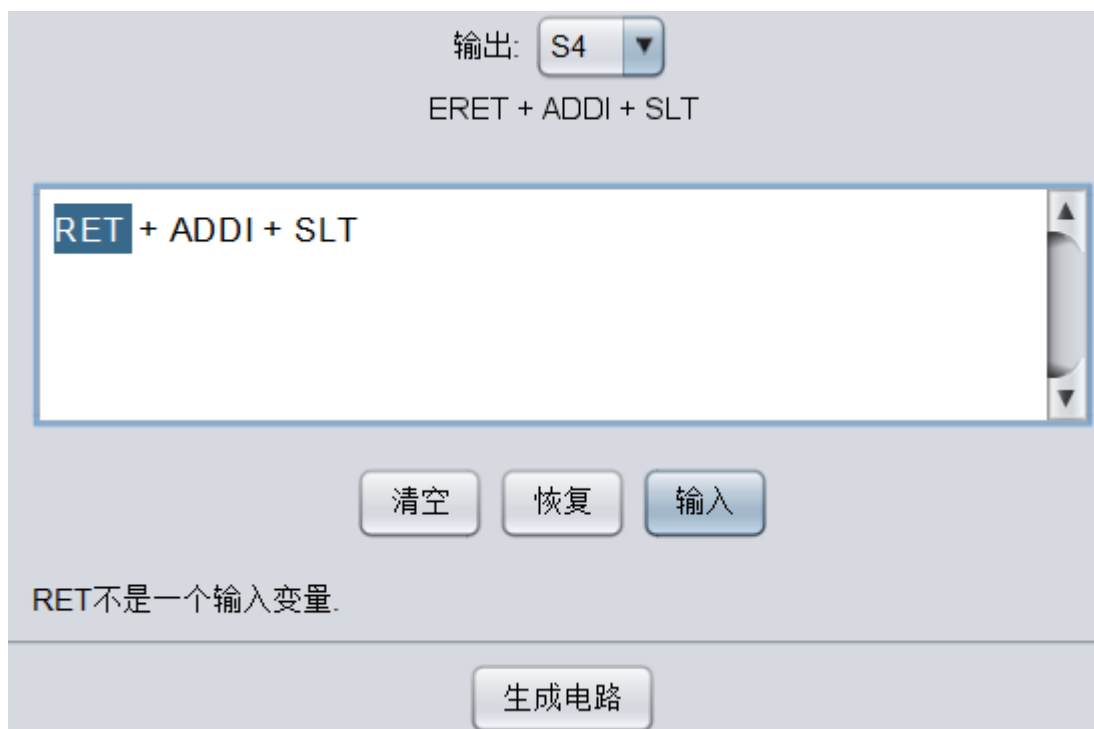


图 1-13 logisim 故障部分截图

**原因分析：**excel 表中无 eret 项，而是 ret 项，生成的表达式含有不是输入变量的 ret，logisim 无法根据其生成组合逻辑电路。

**解决方案：**将 excel 表中 ret 改成 eret，正确生成表达式即可。类似问题还有填写 3 号表时，按书上指导和实验要求更正 excel 表格即可。

### 1.4.2 微程序法填写表格时出错

**故障现象：**状态没有正常转移，educoder 上评测不通过。

**原因分析：**填写 3 号 excel 表有误。

**解决方案：**分析评测集样例，发现是 beq 的判别控制字段有误，进行更正后正确实现状态转移。

# 华中科技大学课程实验报告

## 1.4.3 现代时序中微程序条件判别逻辑出错

**故障现象：**判别逻辑出现不存在的状态，如 6 和 7。

**原因分析：**填写 4 号 excel 表有误。

**解决方案：**错误的化简了输入，更正 excel 表后正确。

## 1.4.4 带中断的现代时序联调冒泡排序时候出错

**故障现象：**正确连接电路后。点击按键相应位置没有变化。

**原因分析：**加载了错误的数据映像。

**解决方案：**更正错误的数据映像，完成逻辑。

## 1.5 测试与分析

### 1.5.1 Educoder 测试

已通过 educoder 上全部测试样例。

### 1.5.2 Logisim 测试

三级时序冒泡程序测试如图 1-14 所示，最终在执行 0x81d 个 clks，251 条指令后停下，最后一条指令为 beq 指令。

```
000 fff00413 000004932084a02300140413 004484932084a0230014041300448493 2084a02300140413004484932084a023 00140413004484932084a02300140413
010 004484932084a0230014041300448493 2084a02300140413004484932084a023 0000041301c00493200429832004aa03 0149a2b3000286632134a02321442023
020 ffc48493 00940463fe0000e3 00440413 01c0049300940463fc0008e3 00000063 00000000000000000000000000000000 00000000000000000000000000000000
030 000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
040 000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
050 000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
060 000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
070 000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
080 00000006000000050000000400000003 000000020000000100000000 ffffffff 00000000000000000000000000000000 00000000000000000000000000000000
```

图 1-14 三级时序冒泡程序测试

# 华中科技大学课程实验报告

带中断的现代时序冒泡程序测试如图 1-15 所示，最终在执行 0x1992 个 clks，252 条指令后停下，最后一条指令为 beq 指令。

```
000 fff00413 000004932084a02300140413 004484932084a0230014041300448493 2084a02300140413004484932084a023 00140413004484932084a02300140413
010 004484932084a0230014041300448493 2084a02300140413004484932084a023 0000041301c00493200429832004aa03 0149a2b3000286632134a02321442023
020 ffc48493 00940463fe0000e3 00440413 01c0049300940463fc0008e3 00000063 00000000000000000000000000000000 00000000000000000000000000000000
030 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
040 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
050 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
060 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
070 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
080 000000060000000050000000400000003 000000020000000100000000 ffffffff 00000000000000000000000000000000 00000000000000000000000000000000
```

图 1-15 带中断的现代时序冒泡程序测试

测试带中断的现代时序 CPU 终端功能，按下四次按键 1 后，测试结果如图 1-16 所示

```
070 00000000000000000000000000000000 00000000000000000000000000000000
080 000000060000000050000000400000003 000000020000000100000000 ffffffff
090 000000040000000040000000400000004 000000040000000040000000400000004
```

图 1-16 按下四下按键 1 后 RAM 中部分截图

按下两次按键 2 后，测试结果如图 1-17 所示

```
080 000000060000000050000000400000003 000000020000000100000000 ffffffff
090 00000000000000000000000000000000 00000000000000000000000000000000
0a0 fffffffe fffffffe fffffffe fffffffe fffffffe fffffffe fffffffe fffffffe
```

图 1-17 按下两下按键 2 后 RAM 中部分截图

按下三次按键 1 再按下三次按键 2，测试结果如图 1-18 按下三下按键 1 再按下三下按键 2 后 RAM 中部分截图所示。

```
080 000000060000000050000000400000003 000000020000000100000000 ffffffff
090 000000030000000030000000300000003 000000030000000030000000300000003
0a0 fffffffd fffffffd fffffffd fffffffd fffffffd fffffffd fffffffd fffffffd
```

图 1-18 按下三下按键 1 再按下三下按键 2 后 RAM 中部分截图



## 2 总结与心得

### 2.1 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：

- ✧ 在单周期 RISC-V CPU 设计（变长指令周期 3 级时序）中，根据变长三级时序设计、构造硬布线控制器，实现支持五条典型 RISC-V 指令在单总线 CPU 上运行，完成后最终 CPU 能运行内存冒泡程序。
- ✧ 在单周期 RISC-V CPU 设计（现代时序）中，根据该时序设计，构造硬布线控制器，实现支持 5 条典型 RISC-V 指令在单总线 CPU 上运行，完成后最终 CPU 能运行内存冒泡排序。
- ✧ 在 RISC-V 现代时序中断机制实现实验中，为采用现代时序的单总线结构 RISC-V CPU 设计增加中断处理机制，增加了数据通路和指令支持，正确实现多个外部按键中断事件的随机处理，完成了现代时序 CPU 中断功能的增加。

#### 2) 功能总结：

- ✧ 实现了 32 位 RISC-V 指令字定长指令的译码和指令的解析。
- ✧ 设计了时序发生器（FSM、组合逻辑）
- ✧ 设计了微程序控制器
- ✧ 设计了硬布线控制器
- ✧ 实现了微程序转移逻辑
- ✧ 完成了支持中断的现代时序单总线 CPU
- ✧ 完成了三级时序单总线 CPU

### 2.2 实验心得

心得与收获：

- ✧ 熟练使用上学期数电教学中学会的有关 logisim 的表达式，真值表等组合逻辑电路生成有关功能，并复习了和新掌握了有关元器件，更加深刻的了解了

# 华中科技大学课程实验报告

---

logisim 这个软件，为将来可能进行的分析工作打下牢固基础。

- ✧ 在硬件知识上，熟悉并掌握了单总线 CPU 不同方案下的设计架构，并自己动手设计实现了部分功能，最终以冒泡排序程序的联调来验证 CPU 的可工作性，既提升了动手能力也收获了知识。
- ✧ 在软件知识上，熟悉并掌握了 RISC-V 五条典型指令以及中断指令的编码，控制信号的产生，以及执行指令时数据通路的变化，实验不仅让我清晰的看到 CPU 是如何执行指令的，也让我学会如何去分析这些指令，从而更好地进行设计和调整。
- ✧ 光说不做假把式，光停留在课本中是远远不够的，通过实验中自己动手去设计实现 CPU，不仅倒逼自己去更认真，更深刻的理解课本，同时也在不断踩坑和调试过程中对 CPU 这一计算机关键构成部件有了更深刻的理解明白道 CPU 内部各个运算器件如何协同工作来完成 CPU 的整个功能，不得不说这次实验和教材结合的相当好，而脱离教材所学实验设计即使再好，也很难说是高明的。在此我还要感谢老师和同学们在学习中对我的帮助，特别是我的任课老师谭志虎老师在组原群里孜孜不倦的回答整个年级的问题，令人敬佩。

建议：

- ✧ 实验通过填写 excel 表格来生成表达式，通过 logisim 生成电路功能来完成复杂组合逻辑的设计而不用学生自己手连的设计是相当高明的，希望能够发扬光大。
- ✧ 实验在 educoder 平台上进行，但因学校先前是在 MIPS32 架构下进行实验，因此 educoder 平台上很多内容都没有进行更新，例如实验介绍中的测试集和实验中实际的测试集也有不一致之处，实验时可能会给学生带来不必要的时间浪费，建议尽快更正。
- ✧ educoder 平台上测试可以面向测试集调试，可以考虑适当隐藏测试集来进行学生能力的区分。

## 参考文献

- [1] DAVID A. PATTERSON(美). 计算机组成与设计硬件/软件接口(原书第 5 版). 北京:机械工业出版社.
- [2] David Money Harris(美). 数字设计和计算机体系结构(第二版). 机械工业出版社
- [3] 谭志虎, 秦磊华, 吴非, 肖亮. 计算机组成原理. 北京:人民邮电出版社, 2021 年.
- [4] 谭志虎, 秦磊华, 胡迪青. 计算机组成原理实践教程. 北京:清华大学出版社, 2018 年.

• 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： 李宇

### 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： \_\_\_\_\_