

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： CS2005

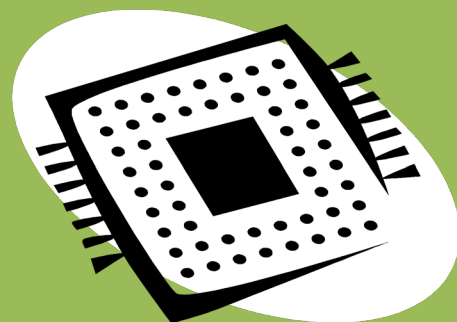
学 号： U202090063

姓 名： 董玲晶

电 话： 13067217235

邮 件： 1355532189@qq.com

完成日期： 2022-07-08



计算机科学与技术学院

1 CPU 设计实验

1.1 设计要求

本实训项目帮助学生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 MIPS 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 `eret` 的支持，需要中断服务程序配合。

1.2 方案设计

设计顺序：指令译码器设计、支持中断功能的微程序控制器设计、微程序单总线 CPU 设计、支持中断功能的现代时序硬布线控制器设计。

1.2.1 指令译码器

指令 IR 分为 OP 操作码和 FUNCT，查表可知若 OP=0X4 为 BEQ 指令；若 OP=0X23 为 LW 指令；若 OP=0X2B 为 SW 指令；若 OP=0X8 为 ADDI 指令；若 OP=0 且 FUNCT=0X2A 为 SLT 指令。

1.2.2 微程序状态与信号安排

首先要搞清楚所有状态并且安排微程序，在微程序里一个状态就相当于一条微指令，是后续电路设计的基础（包括微程序入口查询逻辑、条件判别逻辑等）。

使用实验文件包下的“6.单总线 MIPS 支持中断微程序逻辑自动生成” excel 文件，首先在第一列的微指令功能里安排程序，再在同一行选中本条微指令涉及的控制信号。

根据微程序的功能安排微指令。取指令对于所有指令来说是公操作，为取指令安排 4 个微指令，对应的状态为 0-3，0-3 号微指令的功能和控制信号如下表所示。

华中科技大学课程实验报告

表 1.2.1-1 取指令微程序各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
0	PC \rightarrow X; PC \rightarrow AR	PCout = Xin = ARin = 1	000
1	X+4 \rightarrow Z	+4 = 1	000
2	Z \rightarrow PC; M[AR] \rightarrow DR	Zout = PCin = 1; Read = DREin = 1	000
3	DR \rightarrow IR	DRout = IRin = 1	100

对于 LW 指令除去取指部分，为其指令安排五条微指令，其中前两条微指令功能为计算周期的计算指令，后三条为执行周期的执行指令，具体数据通路与控制信号如下表。在指令结束的最后一个周期给出 P3 判别字符，表示该条微指令是程序的最后一条指令，要检测是否有中断信号。

表 1.2.1-2 LW 指令各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
4	R[rs] \rightarrow X	Rout = Xin = 1	000
5	IR(I)+X \rightarrow Z	IR(I)out = ADD = 1	000
6	Z \rightarrow AR	Zout = ARin = 1	000
7	M[AR] \rightarrow DR	Read = DREin = 1	000
8	DR \rightarrow R[rt]	DRout = Rin = 1	001

同理，也为 SW 指令安排两条微指令的计算周期和三条微指令的执行周期，具体数据通路和控制信号如下表。在指令结束的最后一条微指令给出 P3 判别字符，表示该条微指令是程序的最后一条指令，要检测是否有中断信号。

表 1.2.1-3 SW 指令各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
9	R[rs] \rightarrow X	Rout = Xin = 1	000
10	IR(I)+X \rightarrow Z	IR(I)out = ADD = 1	000
11	Z \rightarrow AR	Zout = ARin = 1	000
12	R[rt] \rightarrow DR	Rout = DRin = Rs/Rt = 1	000
13	DR \rightarrow M[AR]	DREout = Write = 1	001

华中科技大学课程实验报告

BEQ 转移指令计算和执行周期微指令数同 SW 和 LW 指令，但不同点在于 BEQ 在计算周期结束时，要根据计算结果设置的 EQUAL 标志来设置程序跳转的分支地址。若 EQUAL 为 1，则根据条件判别逻辑中译码器的值设置分支跳转的地址；若 EQUAL 为 0，不进行跳转，指令直接结束。

表 1.2.1-4 BEQ 指令各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
14	$R[rs] \rightarrow X$	$R_{out} = X_{in} = 1$	000
15	$R[rt] + X \rightarrow PSW$	$R_{out} = SUB = R_s/R_t = PSW_{in} = 1$	011
16	$PC \rightarrow X$	$PC_{out} = X_{in} = 1$	000
17	$IR(A) + X \rightarrow Z$	$IR(A)_{out} = ADD = 1$	000
18	If($PSW.EQUAL == 1$): $Z \rightarrow PC$	$Z_{out} = PC_{in} = PSW.EQUAL = 1$	001

SLT、ADDI 指令和前面的三种指令不同，SLT 指令没有计算周期只有执行周期，所以不用为 SLT、ADDI 指令安排计算周期，而都只安排三条微指令的执行周期。在指令结束的最后一微指令给出 P3 判别字符，表示该条微指令是程序的最后一微指令，要检测是否有中断信号。SLT 和 ADDI 数据通路如 1.2.1-5 和 1.2.1-6 所示。

表 1.2.1-5 SLT 指令各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
19	$R[rs] \rightarrow X$	$R_{out} = X_{in} = 1$	000
20	$X - R[rt] \rightarrow Z$	$R_{out} = R_s/R_t = SLT = 1$	000
21	$Z \rightarrow R[rd]$	$Z_{out} = R_{in} = RegDst = 1$	001

表 1.2.1-6 ADDI 指令各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
22	$R[rs] \rightarrow X$	$R_{out} = X_{in} = 1$	000
23	$IR(I) + X \rightarrow Z$	$IR(I)_{out} = ADD = 1$	000
24	$Z \rightarrow R[rt]$	$Z_{out} = R_{in} = 1$	001

华中科技大学课程实验报告

支持中断功能后需要增加一个 ERET 指令，其数据通路和控制信号如下。

表 1.2.1-7 ERET 指令各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
27	EPC→PC, 1→IE	EPCout = PCin = STI = ClrInt = 1	001

除此之外还需要增加一个包含两条微指令的中断响应周期，其数据通路和控制信号如下。

表 1.2.1-8 中断响应周期各条微指令的数据通路与控制信号

	数据通路	控制信号	P1 P2 P3
25	0→IE, PC→EPC	CLI = PCout = EPCin = 1	000
26	中断程序入口→PC	IntAout = PCin = 1	001

1.2.3 微程序的入口查找逻辑

在取指令周期的最后一条微程序里，要根据译码信息计算下一条指令的分支地址，所以要设计一个入口查找逻辑，能根据机器指令的译码信号计算出微程序的入口地址，实现正确的下一条指令执行。译码信号有 LW、SW、BEQ、SLT、ADDI、ERET，根据入口地址由微程序状态机里各指令的状态地址得到，分别是十进制的 4、9、14、19、22、25。

表 1.2.2-1 支持中断的微程序地址入口表

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 S4S3S2S1S0
1						00100
	1					01001
		1				01110
			1			10011
				1		10110
					1	11001

1.2.4 微程序的条件判别测试逻辑

由于本次实验设计的微程序控制机取消了下址字段转而采用计数器法，所以就要根据判别字段设计一个条件判别逻辑的组合逻辑并分配多路选择器的各路输入来实现分支地址转移。

多路选择器的 0 号接计数器的输出端，即若判别字段全为 0 使用计数器自增得到下一条微指令的地址；1 号接微程序入口查找逻辑的输出，表示转移到指令译码信号得到的对应指令入口地址处；2 号接 BEQ 指令的执行周期地址入口，由 1.2.2 可得为 16；3 接中断响应周期的入口地址，由 1.2.2 可得为 26；4 号则接微程序的取指令入口地址，表示下一条指令为取指令。总体结构如下图所示。

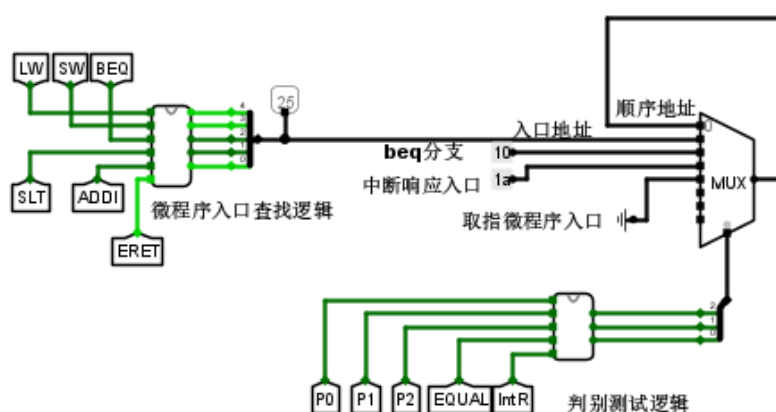


图 1.2.3-1 条件判别逻辑电路结构图

判别测试逻辑的输入为判别字段 P0 P1 P2、EQUAL 信号、intR 信号，判别测试逻辑输出接多路选择的选择端。若输入全为 0 则表示既不用回到取指令入口，也不用根据 EQUAL、intR 信号跳转，判别字段无效采用计数器顺序地址得到下一条微指令的入口地址，所以多路选择器的控制端为 1，即输出为 0。若 P0 为 1 则表示要根据指令译码信号计算下一条地址，所以输出为 1，多路选择器的控制端为 1。当 P1 为 1 时，若 EQUAL 信号的值为 1 则要跳转到 BEQ 信号执行周期的入口，多路选择器的选择端应该为 2，所以条件判别测试逻辑的输出也为 2；EQUAL 信号的值为 0 则直接回到取指

华中科技大学课程实验报告

令周期，输出则为 4。当 P2 为 1 时，若 IntR 信号的值为 1 则要进入到中断响应周期，多路选择器的选择端应该为 3，即条件判别测试逻辑的输出为 3；若 IntR 信号的值为 0 则直接回到取指令周期，输出则为 4。在 P2 和 P3 同时为 1 的情况下，P2 优先级大于 P3。

综上所述，可设计出如下组合逻辑真值表。

表 1.2.3-2 条件判别测试组合逻辑真值表

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

1.2.5 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状态机和普通的状态机大致相同，只是加上了输入信号 ERET（中断周期信号）和 IR（中断信号）、节拍数位 2 的中断响应周期、节拍数为 1 的中断指令周期。部分状态机组合逻辑真值表如图 1.2.4-2 至 1.2.4-3 所示。

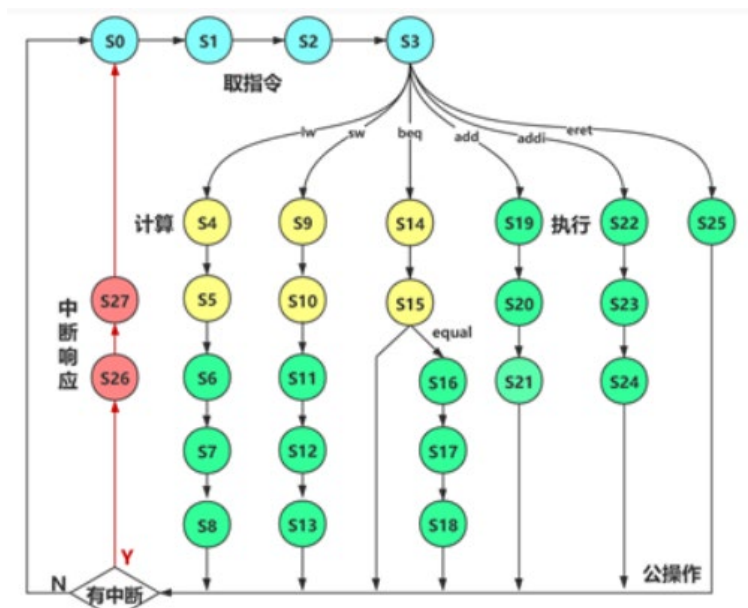


图 1.2.4-1 支持中断的控制器状态机

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	LW	SW	BEQ	SLT	ADDI	ERET	IR	EQUAL	次 态 10进	N4	N3	N2	N1	N0
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

图 1.2.4-2 状态机 BEQ 指令计算周期组合逻辑真值表

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	LW	SW	BEQ	SLT	ADDI	ERET	IR	EQUAL	次 态 10进	N4	N3	N2	N1	N0
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

图 1.2.4-3 状态机中断判别功能组合逻辑真值表

1.3 实验步骤

- (1) 指令译码器：根据 1.2.1 的思路，输入为指令码 IR，输出为各个指令的信号（LW、SW、BEQ、ADDI、SLT），用比较器、常量、与门实现判别组合逻辑，将各个指令信号与非就得到了其他指令信号 OtherInstr 信号输出，电路

华中科技大学课程实验报告

电路的控制存储器当中。

微指令	微指令十六进制
10000000100100000000000000000000	20240000
00000000000000000000100000000000	800
001000010100000000001000000000	8500200
010000000000010000000000000100	10010004
000100000001000000000000000000	4040000
000010000000000001000000000000	2001000
001000001000000000000000000000	8200000

图 1.3-3 微指令二进制和十六进制编码

(5) 完善 **logisim** 电路图中的微程序框架, 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的连接, 实现微程序的主要数据通路, 完善后的电路图如图 1.3-4 所示。

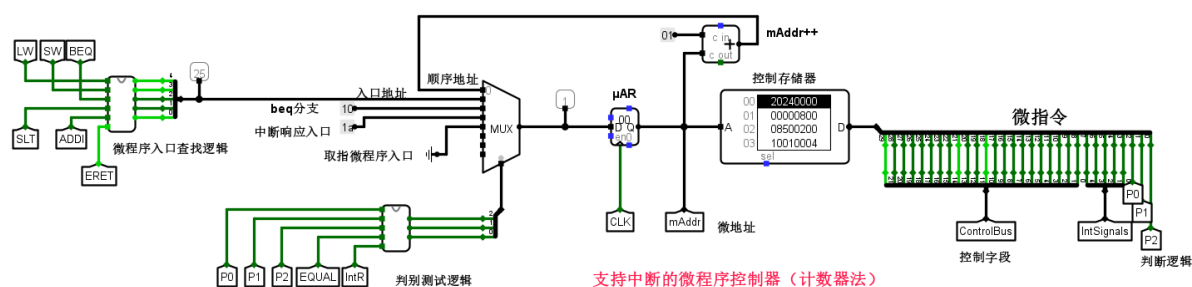


图 1.3-4 支持中断的微程序控制器电路

(6) 微程序中断逻辑：异常程序地址计数器 EPC 输入为来自内部总线的数据，因此输入接内部总线；同时要有一个控制其工作的信号，即寄存器使能端 EPCin 信号，输出端用一个三态门在 EPCout 信号的作用下将数据输出到内部总线，EPC 电路结构图如图 1.3-5 所示。

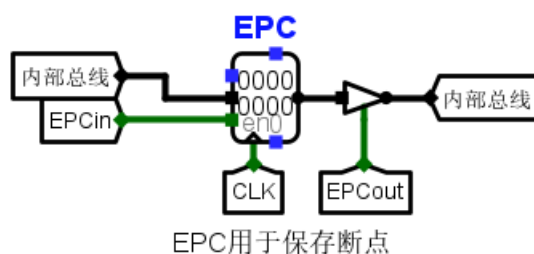


图 1.3-5 EPC 电路结构图

中断控制器的中断请求信号与中断使能寄存器输出信号相与, 使能信号为 0 则中断被屏蔽, 中断请求信号为 0; 反之则输出中断请求信号。使用一个多

路选择器，选择端即为中断号，输入为不同按键的中断服务地址（根据 MARS 汇编器汇编源程序查看），在三态门 Addrout 信号控制下输出到内部总线。中断控制信号由关中断、开中断、控制信号 Addrout、EPCin、EPCout 组成。电路图如图 1.3-6 所示。

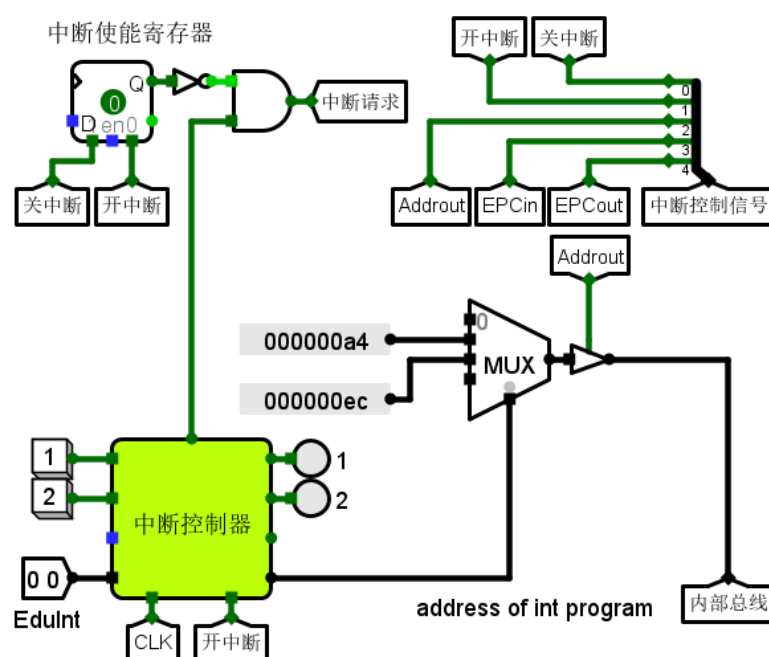


图 1.3-6 终端逻辑电路结构图

- (7) 支持中断的现代时序硬布线控制器状态机：在 5 号 excel 表格中按照 1.2.4 的思路填写状态机逻辑真值表，自动生成次态逻辑表达式后，在 logisim 相关的文件利用分析组合逻辑电路功能自动生成电路。
- (8) 完善 logisim 电路图中的现代时序硬布线控制器框架，控制存储器不需要实现而是直接采用微程序的控制存储器即可。完善后的电路图如图 1.3-7 所示。

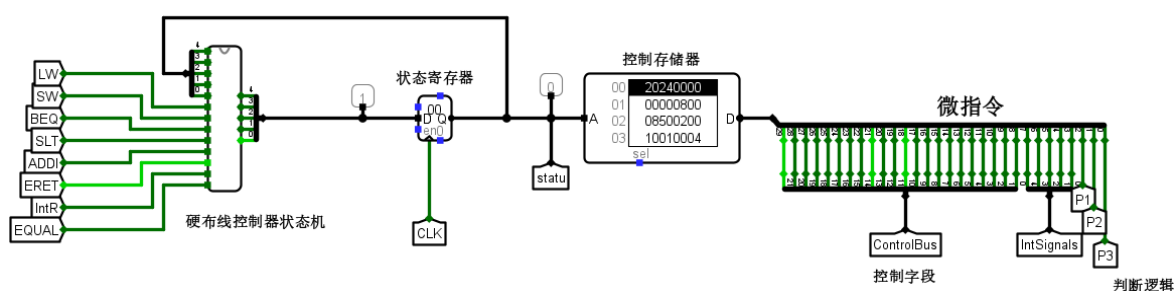


图 1.3-7 支持中断的现代时序硬布线控制器电路

华中科技大学课程实验报告

1.4 故障与调试

1.4.1 寄存器触发方式问题

故障现象：微程序控制器输入指令字最后得到的微指令地址和控制总线的值有误，如图 1.4-1 所示。

Cnt	Instr	equal	IntR	mAddr	cBus	ErrBit	Cnt	Instr	equal	IntR	mAddr	cBus	ErrBit
00	2010ffff	0	0	00	202400	xx	00	2010ffff	0	0	00	202400	xx
01	2010ffff	0	0	01	000008	xx	00	2010ffff	0	0	01	000008	15
02	2010ffff	0	0	02	085002	xx	01	2010ffff	0	0	01	000008	xx
03	2010ffff	0	0	03	100100	xx	01	2010ffff	0	0	02	085002	13
04	2010ffff	0	0	16	040400	xx	02	2010ffff	0	0	02	085002	xx
05	2010ffff	0	0	17	020010	xx	02	2010ffff	0	0	03	100100	14
06	2010ffff	0	0	18	080200	xx	03	2010ffff	0	0	03	100100	xx
07	0274402a	0	0	00	202400	xx	03	2010ffff	0	0	16	040400	14
08	0274402a	0	0	01	000008	xx	04	2010ffff	0	0	16	040400	xx
09	0274402a	0	0	02	085002	xx	04	2010ffff	0	0	17	020010	12
0a	0274402a	0	0	03	100100	xx	05	2010ffff	0	0	17	020010	xx
0b	0274402a	0	0	13	040400	xx	05	2010ffff	0	0	18	080200	13
0c	0274402a	0	0	14	040044	xx	06	2010ffff	0	0	18	080200	xx
0d	0274402a	0	0	15	080220	xx	06	2010ffff	0	0	00	202400	15
0e	42000018	0	0	00	202400	xx	07	0274402a	0	0	00	202400	xx
0f	42000018	0	0	01	000008	xx	07	0274402a	0	0	01	000008	15

图 1.4-1 测试用例出错详情

原因分析：CPU 内部结构的其他寄存器和运算器等其他部件涉及的触发器和寄存器使用的触发方式都是上升沿触发，若中断用到的状态寄存器也使用上升沿触发则会导致触发混乱，数据还没有传输完毕、一个节拍电位没有完全结束就使寄存器得到出发，数据输出到控制存储器里，造成结果错误。

解决方案：将状态寄存器的触发方式改为下降沿。

1.5 测试与分析

本次实验内容均已在 Educoder 平台测试并通过。

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：通过本次实验我在现代时序单总线结构的 MIPS CPU 的基础上，利用数字电路和计算机组成原理的知识，利用组合逻辑与时序逻辑电路增加硬件数据通路，设计了支持中断功能的微程序控制器、硬布线控制器，

以及中断需要用的中断逻辑电路如中断使能、中断控制、断点保存等。

- 2) 功能总结：实现了现代时序单总线结构的 MIPS CPU 中断功能的支持并实现多个外部按键中断事件的随机处理。
- 3) 其他总结：无。

1.7 实验心得

(1) 通过本次实验，我加深了自己对各种常见指令数据通路和所需要的控制信号的记忆和 CPU 单总线数据通路结构的理解，将在理论课堂上所学到的有关于 CPU 的基础知识相融合并应用于实践。刚上完单总线结构的时候只知道这很重要，但其实对其不太理解，对各种指令执行时 PC、运算器、内存等硬件间数据相互传输的过程印象不深，一头雾水遇到了就只能翻书找着写。还有就是在 CPU 的中断机制和实现过程方面，之前在课堂上这块我听得似懂非懂，一直不太理解中断时各种硬件如 EPC、中断控制器互相配合工作的机制，但经过这次实验的操作和设计，我彻底明白和理解了这些过程的实验原理，遇到指令也能自然地说出指令执行过程中数据的传输和各种硬件是如何进行协调、配合工作，再也不是面对书本的内容死记硬背。

(2) 在做这次实验的过程中，我体会到了学习计算机组成原理的快乐，之前理论课堂和课后学习通网课的时候真的万分痛苦，觉得组原是一本很难且无聊的学科，没有学习热情和兴趣，只能强迫自己硬着头皮去学习。但在本次实验中，自己动手设计和实现 CPU 的过程中，我却发现组原的学习特其实别有趣，激发了自己的热情。有了这一次的经历，我也没有那么畏惧和讨厌硬件的课程了，可以说是这门实验课改变了我对计算机硬件课程“除了难就是无聊”的刻板看法，也有了自己一定能够学好后续其他硬件的自信和决心。

(3) 我个人是认为组原这门课以及配套的实验课是我目前在华科上过的最好的课，这一切都离不开组原课题组在各方面的付出与努力。我也没有什么建议可以给出，就希望以后的学弟学妹们在学习这门课时都能感受到计算机的乐趣和硬件的乐趣，也希望这门课可以被更多其他的大学所看到，走出华科，成为像 CMU 15-213 那样经典的课程，帮助更多的人走进计算机、爱上计算机。

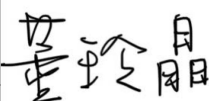
• 指导教师评定意见 •

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：董玲晶



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：_____