

基于 logisim 的多周期 MIPS CPU 硬布线控制器设计

学 号	E12214052	专 业	计算机科学与技术	姓 名	赵宸宇
实验日期	2024 年 9 月 19 日	教师签字		成 绩	

摘要

本次实验在上次“多周期微程序 mips 处理器设计”的基础上复用数据通路，主要完成将控制器的状态控制机制更换为硬布线控制器的设计。同时，作为附加项，学生可以选择是否将控制器的微命令存储器也更换为硬布线逻辑电路。本实验的硬布线控制器是纯硬布线的，不包含任何存储设备。

本次实验的实验产出有：

1. 基于硬布线设计的多周期 8 指令 mips CPU 逻辑电路图:cpu.circ
2. tex 实验报告
3. 支撑材料（用于状态机相关设计的 xlsx 表格、py 程序等）
4. 头哥网通关
5. git 日志请见<https://gitee.com/cslearnerer/AHU-CSHT>

目录

一、【实验目的】	2
二、【实验原理】	2
2.1 采用硬布线设计以提升性能	2
2.2 硬布线控制器和微程序控制器逻辑等价	2
2.3 设计方案	3
三、【实验内容】	3
3.1 数据通路的复用	3
3.2 完成控制器设计	3
3.3 完成硬布线控制器（状态翻译）设计	4
3.4 联调测试	5
四、【小结讨论】	5
4.1 工作展望	5

一、【实验目的】

实现多周期 mipsCPU 的纯（可选 | 已选）硬布线控制器设计。

完成多周期硬布线 mipsCPU 整体设计，这包含数据通路设计、控制器设计和程序联调。

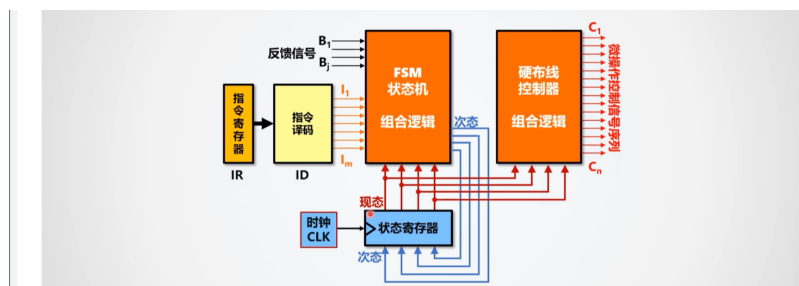
二、【实验原理】

2.1 采用硬布线设计以提升性能

在实验 1 的基础上，可以通过进一步将微程序控制器升级为硬布线控制器来提升 MIPS CPU 的运行速度。

2.2 硬布线控制器和微程序控制器逻辑等价

如图 3，硬布线控制器由一个状态机和一个组合逻辑电路组成，在功能上和第一次实验中的微程序控制器等价。



用xlsx表格，通过刻画状态转移图中的状态转移关系（有向边+条件）来完成FSM状态机这一子电路的组合逻辑设计。结果如图4所示。

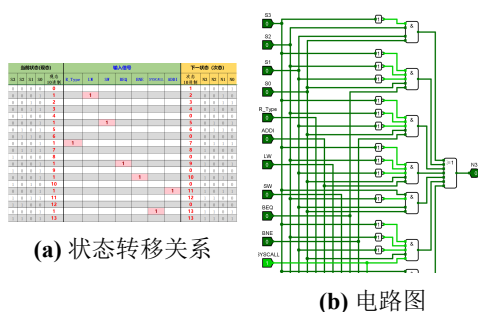


图4 FSM 状态机

3.3 完成硬布线控制器（状态翻译）设计

对于本次实验设计，可以使用硬布线控制器替代原本的只读存储器。

该硬布线控制器的本质是一个接收4bit输入，给出21bit输出（未优化地址段时）的组合逻辑电路，可以采用如图5的真值表技术配合logisim工程设计功能实现该组合电路的设计。

微指令功能	状态	微指令地址	微指令	PCsrc	AluA	AluB	MTorR	Rdst	写IR	写PC	RWrite	MWrite	MRead	BEQ	BNE	Alu-op	P	下址	微指令	十六进制
取指令	0	0000	0	0	0	01	0	0	1	1	0	0	1	0	0	00	0	0001	0000100110010000000001	13201
译码	1	0001	0	0	0	11	0	0	0	0	0	0	0	0	0	00	1	0000	0001100000000000000000	30010
LW1	2	0010	0	0	1	10	0	0	0	0	0	0	0	0	0	00	0	0011	0011000000000000000011	60003
LW2	3	0011	1	0	0	00	0	0	0	0	0	0	1	0	0	00	0	0100	10000000000000000000100	100204
LW3	4	0100	0	0	0	00	1	0	0	0	0	0	0	0	0	00	0	0000	0000010001000000000000	8800
SW1	5	0101	0	0	1	10	0	0	0	0	0	0	0	0	0	00	0	0110	00110000000000000000110	60006
SW2	6	0110	1	0	0	00	0	0	0	0	0	1	0	0	0	00	0	0000	10000000000000000000000	100400
R1	7	0111	0	0	1	00	0	0	0	0	0	0	0	0	0	11	0	1000	00100000000000000101000	40068
R2	8	1000	0	0	0	00	0	0	1	0	0	0	0	0	0	00	0	0000	00000010010000000000000	4800
BEQ	9	1001	0	1	1	00	0	0	0	0	0	0	0	0	1	0	10	0000	011000000000000001000000	C0140
BNE	10	1010	0	1	1	00	0	0	0	0	0	0	0	0	1	10	0	0000	011000000000000001000000	C00C0
ADD1	11	1011	0	0	1	10	0	0	0	0	0	0	0	0	0	00	0	1100	00110000000000000000100	6000C
ADD2	12	1100	0	0	0	00	0	0	0	1	0	0	0	0	0	00	0	0000	00000000000000000000000	800
CALL	13	1101	1	0	0	00	0	0	0	0	0	0	0	0	0	11	0	1101	10000000000000000101101	10066D

图5 微命令真值表

通过如图6所示的设计流程，即可得到完全硬布线设计的控制器。



图6 设计流程

3.4 联调测试

通过对 sort.asm 进行测试，经过验证，本次实验直到 syscall 指令共运行了 891 个 clk，和标称值 891 一致。实验测试通过。

四、【小结讨论】

在本次实验中，我通过复用相关电路，根据顶层原理独立自主完成了纯硬布线 mipsCPU 控制器的设计-调试工作。

通过本次实验，我对计算机底层的工作原理理解更深刻，对硬件调试能力得到提升。个人综合工程师素养得到提升。

4.1 工作展望

1. 优化数据通路图设计，增加可读性和可维护性
2. 简化微程序字段，这样可以简化硬布线组合逻辑电路，提升 CPU 性能（提升响应速度，降低发热量，提升 CPI）。