

课程设计

实验方式

Logisim 软件仿真：

开源免费；

有配套的慕课（MOOC）及在线实训平台；

实验内容和课程内容配合，有助于学生理解；

题目难度适中；

所有实验均可在 Educoder 在线实训平台上在线完成，在线测试，并可实现故障的快速精准定位，方便学生进行电路调试

网络资源

华中科技大学谭志虎老师《计算机硬件系统设计》

<https://www.icourse163.org/course/HUST-1205809816>

课程相关资源下载

<https://gitee.com/totalcontrol/hustzc>（码云下载）

Educoder 在线实训平台（检测及提交实验结果）

<http://www.educoder.net/paths/1426>

头歌简明教程（学生版）

<https://www.educoder.net/forums/4915>

实验环境

logisim-ita-cn-20200118, java 平台，支持 Win、Linux、MAC 平台，无需任何硬件平台。

学习内容

课前：自学慕课（MOOC）相关内容（具体见后每个实验）

课内：讲解实践

Educoder 在线实训平台：检测及提交实验结果

华中科技大学谭志虎老师《计算机硬件系统设计》

<https://www.icourse163.org/course/HUST-1205809816>

学习慕课相关章节

课程设计内容

1 单周期 MPIS CPU 设计（硬布线控制器）

学习章节：

慕课《计算机硬件系统设计》

- 原理： 6.1 单周期 MIPS CPU 数据通路
- 实验： 6.3 单周期 MIPS CPU 实验（8 条指令）

Educoder 平台：

第 1 关：单周期 MIPS CPU 设计

实验目的：

- 掌握硬布线控制器设计的基本原理
- 在 Logisim 平台中设计实现 MIPS 单周期 CPU

实验任务：

- 绘制单周期 MIPS CPU 数据通路
- 实现单周期硬布线控制器
- 测试联调

实验步骤：

(1) 构建单周期 MIPS 主机通路

- PC：32 位寄存器
- 数据存储器：地址 10 位（字地址），数据 32 位
- 寄存器文件：可从下面两种中选择其一
 - 封装好的“MIPS Regfile”电路
 - CS3410 Components 中的“Register File”（推荐这个，可以直接看到寄存器的值）
- 运算器 ALU：封装好的“MIPS ALU”电路
- 指令存储器：
 - 需自行添加一个 ROM
 - 数据位宽为 32 bits（因为 MIPS 指令字是 32 位）
 - 地址位宽合理设置，使其能够装载完整的排序程序
- 单周期硬布线控制器
 - 将其输出控制信号与对应组件相连
 - 需要设计其内部结构（见步骤 2）
- (2) 设计单周期 MIPS 控制器
 - 输入信号：指令字中的 Opcode，Func 字段
 - 输出信号：各种控制信号
 - 设计组合逻辑电路，得到以下各种信号输出：
 - 指令类型（详见 MIPS 指令集）：
 - 利用比较器，通过比较 Opcode 的值，判断指令是否为某种 I 型指令：LW、SW、BEQ、BNE 或 ADDI（指令译码逻辑）；
 - 利用比较器和简单逻辑门电路，通过比较 Func 的值，判断指令是否为某种 R 型指令：ADD、SLT 或 SysCall；
 - 利用简单逻辑门电路，判断指令是否为运算型 R 指令：R_TYPE
 - ALU_OP 的值：
 - 根据不同的指令（OP 和 FUNC 字段的值）确定执行该指令时 ALU 需要进行何种运算
 - 控制信号：
 - 利用简单逻辑门电路，得到 RegDst、RegWrite、MemToReg、MemWrite、AluSrc、Beq、Bne、Halt（详见慕课：控制信号功能说明表）
- (3) CPU 测试

实验资料中的“8 条指令单独测试用例 2019-12-7.zip”给出了测试程序用到的 8 条指令的单独测试用例；

.asm：测试用例程序的说明及对应的汇编语句；

.hex：测试用例程序的 MIPS 指令序列，16 进制表示，需要装载入指令存储器；

最后一个 sort 是完整的排序程序；

实验提示：

ALU 控制器逻辑：

- 一般方法：

利用 Excel 文件生成 ALU_OP 的表达式：

“单周期硬布线控制器表达式自动生成 2020-3-12.xlsx”

如何填写 ALU_OP，可以参照 Excel 文件中的“运算器规格”标签页或设计框架文件 cpu.circ 中 MIPS ALU 模块的电路；

- 简化方法：

由于本设计中，一共只有 8 条指令，只用到 ALU_OP 的加法（ALU_OP=5）和比较大小（ALU_OP=b）两种运算，其中用到比较大小的指令只有 SLT，所以设计可以更简单（不必对 Func 字段对应的全部运算生成组合逻辑电路）

2 多周期 MPIS CPU 设计（微程序控制器）

学习章节：

慕课《计算机硬件系统设计》

- 原理：6.2 多周期 MIPS CPU 数据通路

- 实验：6.4 多周期 MIPS 微程序控制器实验

EduCoder 平台：

- 第 2 关：微程序地址转移逻辑设计

- 第 3 关：MIPS 微程序 CPU 设计

实验目的：

- 掌握多周期 MIPS CPU 设计原理

- 掌握微程序控制器设计的基本原理

- 利用微程序控制器的设计实现多周期 MIPS 处理器

实验任务：

- 绘制多周期 MIPS CPU 数据通路

- 实现微程序控制器

- 测试联调

实验步骤：

(1) 构建多周期 MIPS 主机通路

- 存储器：地址 10 位（字地址），数据 32 位，同时存储指令和数据

- 寄存器文件：同单周期 MIPS CPU

- 运算器 ALU：同单周期 MIPS CPU

- 添加几个专用寄存器（均为 32 位）：PC，IR，DR，A，B，C

- 微程序控制器：将其输出控制信号与对应组件相连，需要设计其内部结构（见步骤 2）

(2) 设计微程序控制器

- 输入信号：指令字中的 Opcode，Func 字段，时钟信号，复位信号

- 输出信号：多路选择器的选择信号，存储器的访问控制信号，寄存器的写使能信号，微指令中的微命令

- 运算器的控制信号：设计 ALU 控制器逻辑
 - 微指令中的微命令 ALU_Control：其值决定运算器的运算选择控制信号 ALU_OP 的值
- 指令译码信号：设计指令译码逻辑（与单周期 MIPS CPU 相同）
- 微程序地址转移逻辑设计：
 - 利用 Excel 文件得到四位的微程序入口地址的逻辑表达式
“微程序地址转移逻辑自动生成 ext(2020-9-1).xlsx”
- 根据状态图构建微程序
 - 利用 Excel 文件自动得到控制存储器中存储的微指令：
“微指令自动生成(2019-4-22).xlsx”
- (3) CPU 测试：与单周期 MIPS CPU 类似

注意二者不同之处在于多周期 MIPS CPU 的指令存放在 ROM 中，系统复位后依然存在；多周期 MIPS CPU 的指令放在 RAM 中，系统复位后被清空，再次运行需要重新加载

课程设计要求

1. 以下二者中，至少完成一个 CPU 设计，在 Educoder 平台提交
 - 单周期 MIPS CPU 设计（硬布线控制器）：第 1 关
 - 多周期 MIPS CPU 设计（微程序控制器）：第 2 关、第 3 关
2. 该题可不做，有兴趣的同学也可以做
 - 多周期 MIPS CPU 设计（硬布线控制器）：第 4 关、第 5 关
3. 需要交报告（电子版和纸质版都交）
4. 现场演示验收（讲解和问答）