

北 京 邮 电 大 学

实 验 报 告

课程名称：计算机组织与结构实验

实验名称：微程序控制器、CPU 机器指令、中断原理

计算机学院 314 班 姓名叶禹赋

教师刁婷 成绩_____

2025年5月15日

注：刁老师您好，我由于对 docx 的排版、编写表格不够熟练，故采用（飞书编写-截图沾到 doc 中）的报告写法，两次打印后都会有些许模糊，请您见谅。

实验四、微程序控制器实验

一、实验任务及目的

目的：

1. 掌握微程序控制器的原理
2. 掌握 TEC-Plus/TEC-8 模型计算机中微程序控制器的实现方法，微地址转移逻辑的实现方法。

任务：

1. 正确设置操作模式开关 SWC、SWB、SWA，用单微指令方式（DP=1）跟踪控制台操作读寄存器、写寄存器、读存储器、写存储器的执行过程。
 - 记录每一步操作的微地址、判别位、下地址等信号的值。
2. 正确设置指令操作码 IR7-IR4，用单微指令方式（DP=1）跟踪 ADD、SUB、AND、JC、LD 指令的执行过程。
 - 记录每一步操作的微地址、判别位、下地址等信号的值。
 - 对于 JC 指令，跟踪 C=1 和 C=0 两种情况。

二、实验电路分析

微程序控制器的核心在于控制存储器和微地址转移逻辑，二者协同工作实现指令的微操作序列。

1. 控制存储器与微指令结构

- 控制存储器：由 5 片 58C65 E² PROM 构成，总容量为 8K×8 位。每片存储 8 位微指令字段，组合后形成完整的微指令。
- 微指令格式：包含操作控制字段（生成微操作信号）、下地址字段（N_μA）和判别位（P）。

- $N\mu A$ ：指定默认的下一条微指令地址。
- P 字段：决定是否根据条件（如指令操作码、标志位、控制台模式）修改下地址。

2. 微地址转移逻辑的实现

微地址转移逻辑的输入包括当前微地址、微指令的 $N\mu A$ 和 P 字段、外部条件信号（如 C/Z 标志、指令操作码 $IR7-IR4$ 、控制台模式 $SWC-SWA$ ），输出为下一个微地址（ UA ）。其核心逻辑如下：

(1) 无分支情况

- 顺序执行：当判别位 $P=0$ 时，直接采用微指令中的 $N\mu A$ 作为下地址，即 $UA=N\mu A$ 。

(2) 分支情况

- 条件分支：当判别位 $P=1$ 时，根据外部条件修改 $N\mu A$ ，生成新的 UA 。如：
 - 指令译码分支：根据操作码 $IR7-IR4$ 选择微程序入口。
 $UA=IR7-IR4 \parallel N\mu A$ 低位（通过逻辑门组合操作码与 $N\mu A$ ）。
 - 标志位分支：如 JC 指令中，若 $C=1$ 则跳转到目标地址，否则顺序执行。
 $UA = C ? TargetAddr : N\mu A$ 。
 - 控制台模式分支：根据 $SWC-SWA$ 直接跳转到对应控制台操作的首地址（如写寄存器模式固定进入微地址 $0x40$ ）。

(3) 硬件实现

- 多路选择器（MUX）：根据 P 字段和条件信号选择不同的地址源（ $N\mu A$ 、操作码映射地址、标志位跳转地址等）。
- 组合逻辑电路：将 $IR7-IR4$ 、 C/Z 标志、 $SWC-SWA$ 等信号转换为地址偏移量，与 $N\mu A$ 拼接生成 UA 。

三、微程序流程图分析

1. 微程序流程图的组成

微程序流程图通常包含以下关键元素：

- 微操作节点：每个节点对应一条微指令，包含操作控制信号（如 ALU 运算、寄存器读写、总线控制等）。
- 分支条件：由判别位（ P ）和 外部条件（如指令操作码、标志位 C/Z 、控制台模式 $SWC-SWA$ ）决定流程走向。
- 地址转移路径：通过 下地址（ $N\mu A$ ）和条件分支逻辑连接各个节点。

2. 微程序流程图的设计原则

- 公共取指周期：所有指令的微程序均从取指阶段开始，读取指令到 IR （指令寄存器）。
- 指令译码分支：根据 IR 的高 4 位（ $IR7-IR4$ ）跳转到对应指令的微程序入口（微地址）。

- 条件分支：根据标志位（如 C、Z）或操作模式（如控制台读写）选择不同执行路径。
- 微程序复用：公共操作（如取操作数、写回结果）可被不同指令复用，减少冗余微指令。

3. 关键流程节点分析

(1) 公共取指周期

- 微地址：0x00

所有指令的执行均从该节点开始，完成以下操作：

- 将 PC 内容送地址总线 ($MAR \leftarrow PC$)。
- 从存储器读取指令到 IR ($IR \leftarrow M[MAR]$)。
- PC 自增 ($PC \leftarrow PC+1$)。
- 分支逻辑：根据 IR7-IR4 跳转到对应指令的微程序入口（如 ADD 指令跳转到 0x10）。

(2) 指令执行阶段

以 ADD Rd, Rs 指令为例：

- ① 入口地址（译码后）：IR7-IR4=0001 → 微地址 0x10。
- ② 微指令 0x10：
 - 从寄存器堆读取 Rs 到暂存器 A ($A \leftarrow Rs$)。
 - 从寄存器堆读取 Rd 到暂存器 B ($B \leftarrow Rd$)。
 - $N \mu A = 0x11, P=0$ → 顺序执行下一条微指令。
- ③ 微指令 0x11：
 - ALU 执行加法运算 ($A+B \rightarrow C$)。
 - 结果写回 Rd ($Rd \leftarrow C$)。
 - 更新标志位 C 和 Z（根据运算结果）。
 - $N \mu A = 0x00, P=0$ → 返回取指周期。

(3) 条件转移指令（如 JC）

- 入口地址：IR7-IR4 = 0111 → 微地址 0x30。
- 微指令 0x30：
 - 检查 C 标志位。
 - 分支逻辑：
 - ◆ 若 C=1：UA=0x31（计算跳转地址， $PC \leftarrow PC+offset$ ）。
 - ◆ 若 C=0：UA=N μ A=0x00（直接返回取指周期）。

4. 微地址转移逻辑与流程图的关系

微地址转移逻辑是流程图分支的实现核心，其作用如下：

- (1) 无分支 (P=0)：直接使用微指令中的 N μ A 作为下地址。
- (2) 分支 (P=1)：
 - 指令译码分支：将 IR7-IR4 拼接到 N μ A 的高位，生成入口地址。
例如：N μ A=0x0_，IR7-IR4=0001 → UA=0x10。
 - 条件分支：根据 C/Z 标志位选择地址偏移量。
例如：JC 指令中，C=1 时 UA=0x31，C=0 时 UA=0x00。
 - 控制台模式分支：根据 SWC-SWA 直接跳转到固定地址（如写寄存器模式对应 0x40）。