

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

## 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 E<sup>2</sup> 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 μ A) 和判别位 (P)。
  - N μ 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。  
 $UA=IR7-IR4 || N\mu A$  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 $\mu$ 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 $\mu$ A 作为下地址。

(2) 分支 (P=1) :

● 指令译码分支: 将 IR7-IR4 拼接到 N $\mu$ A 的高位, 生成入口地址。

例如: N $\mu$ 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)。

#### 四、 实验过程及结果

操作模式 SWC、 SWB、 SWA	$\mu$ A	N $\mu$ A	P
100	00H	01H	P0
100	09H	08H	
100	08H	0AH	
100	0AH	0CH	
100	0CH	00H	
	00H	01H	P0
011	07H	06H	
011	06H	00H	
	00H	01H	P0
010	05H	04H	
010	04H	04H	
010	04H	04H	
010	04H	04H	
	00H	01H	P0
001	03H	04H	
001	03H	03H	
001	03H	03H	
	00H	01H	
000	01H	00H	P1
000	01H	20H	P1
000	01H	20H	P1
000	01H	20H	P1
000	01H	20H	P1

指令 IR7-IR4	$\mu A$	N $\mu A$	P
0001	21H	01H	P4
0010	22H	01H	P4
0011	23H	01H	P4
0111	27H	12H	P2
	12H	01H	P4
0111	27H	12H	P2
	13H	01H	P4
0101	25H	0EH	
	0EH	01H	P4

# 实验五、CPU 组成与机器指令的执行

## 一、实验任务及目的

目的：

1. 用微程序控制器控制数据通路，将相应的信号线连接，构成一台能够运行测试程序的 CPU。
2. 执行一个简单的程序，掌握机器指令与微指令的关系。
3. 理解计算机如何取出指令、如何执行指令、如何在一条指令执行结束之后自动取出下一条指令并执行，从而牢固建立计算机整机概念。

任务：

1. 预习任务：完成对给定程序的手工汇编。
2. 通过简单的连线构成能够运行程序的 TEC-8/TEC-Plus 模型计算机。
3. 将程序写入存储器，给寄存器 R2、R3 赋初值。
4. 跟踪执行程序，用单拍方式运行一遍，用连续方式运行一遍，详细记录实验过程及结果。
5. 用实验台操作检查程序运行结果。

## 二、程序的手工汇编结果

地址	指令	二进制机器代码	地址	指令	二进制机器代码
00H	LD R0,[R3]	01010011	0AH	INC R2	01001000
01H	INC R3	01001100	0BH	ST R2,[R2]	01101010
02H	LD R1,[R3]	01010111	0CH	AND R0,R1	00110001
03H	SUB R0,R1	00100001	0DH	OUT R2	10100010
04H	JZ 0BH	10000110	0EH	STP	11100000
05H	ST R0,[R2]	01101000	0FH	85H	10000101
06H	INC R3	01001100	10H	23H	00100011
07H	LD R0,[R3]	01010011	11H	EFH	11101111
08H	ADD R0,R1	00010001	12H	00H	00000000
09H	JC 0CH	01110010	13H	00H	00000000

### 三、实验过程及结果

AND R0, R1	100011	000001	10000	1010 0010	ODH	11H	00110001	12H	23H	02H
	000001	100000		1010 0010	ODH	11H	00110001	02H	23H	
OUT R2	101010	000001	10000	1110 0000	0EH	11H	10100010	02H	12H	12H
	000001	100000		1110 0000	0EH	11H	10100010	02H	12H	
STP	101110	000001	10000	1000 0101	OFH	11H	11100000	02H	02H	
	000001	100000	10000	1000 0101	OFH	11H	11100000	02H	02H	
85H	-	-	-	-	-	-	-	-	-	-
23H	-	-	-	-	-	-	-	-	-	-
EFH	-	-	-	-	-	-	-	-	-	-
00H	-	-	-	-	-	-	-	-	-	-
00H	-	-	-	-	-	-	-	-	-	-

● 记录连续方式下的结果

实验寄存器与内存读取结果对照表

开关设置 (SWC/SWB/SWA)	操作类型	地址/寄存器	读出值	备注
0/1/1	寄存器读取	R0	02H	寄存器低 4 位值
		R1	23H	
		R2	12H	
		R3	11H	
0/1/0	内存单元读取	[12H]	62H	指定地址数据
		[13H]	00H	未操作单元 (无关数据)

# 实验六、中断原理实验

## 一、实验任务及目的

目的：

1. 通过简单的中断系统，掌握中断控制器、中断向量、中断屏蔽等概念
2. 了解微程序控制器与中断控制器协调的基本原理
3. 掌握中断子程序和一般子程序的本质区别，掌握中断的突发性和随机性。

任务：

1. 将主程序和中断服务程序手工汇编成二进制机器代码
2. 通过简单的连线构成能够运行程序的 TEC-8 模型计算机。
3. 将主程序和中断服务程序装入存储器，给寄存器 R1 赋初值 01H，R0 赋初值 0。
4. 执行三遍主程序和中断服务程序，详细记录中断有关信号变化情况，特别记录好断点和 R0 的值。
5. 将主程序中地址为 00H 的 EI 指令改为 DI，重新运行程序，记录现象。

## 二、程序的手工汇编结果（包括主程序和中断服务程序）

地址	指令	二进制机器代码
00H	EI/DI	11010000
01H	INC R0	01000000
02H	INC R0	01000000
03H	INC R0	01000000
04H	INC R0	01000000
05H	INC R0	01000000
06H	INC R0	01000000
07H	INC R0	01000000
08H	INC R0	01000000
09H	JMP [R1]	10010100

地址	指令	二进制机器代码
45H	ADD R0,R0	00010000
46H	EI	11010000
47H	IRET	10110000

### 三、实验过程及结果

执行程序顺序	PC 断点值 EI	中断时的 R0 EI
第 1 遍	06H	77H
第 2 遍	08H	78H
第 3 遍	01H	01H
执行程序顺序	PC 断点值 DI	中断时的 R0 DI
第 1 遍	47H	EEH
第 2 遍	47H	EOH
第 3 遍	47H	02H

### 四、实验总结

根据实验四、五、六的实践，系统性地掌握了计算机核心组成原理。实验四聚焦微程序控制器，通过单微指令跟踪验证了微地址转移逻辑：控制存储器的微指令结构（N $\mu$ A、P 字段）与条件分支机制（操作码映射、标志位判断），深入理解了取指-译码-执行的微程序流程。实验五构建完整 CPU，手工汇编测试程序并验证执行过程，揭示机器指令如何分解为微操作序列（如 ADD 指令经历取数、运算、写回三阶段），强化了指令周期与数据通路的协同机制。实验六引入中断系统，对比 EI/DI 指令效果，发现中断随机触发时 PC 断点自动压栈，揭示中断向量与现场保护机制。三实验层层递进，从微指令控制到宏观指令执行，最终实现中断异步处理，完整构建了“控制器-运算器-中断系统”的计算机整机认知体系，深化了对冯·诺依曼架构运行机理的理解。