

北 京 邮 电 大 学

实 验 报 告

课程名称计算机组成与体系结构实验

实验名称微程序控制器原理与机器指令执行

计算机学院 2023211215 班 姓名米梓润

教师靳秀国 成绩

2025 年 5 月 7 日

实验四 微程序控制器组成

一、实验目标：

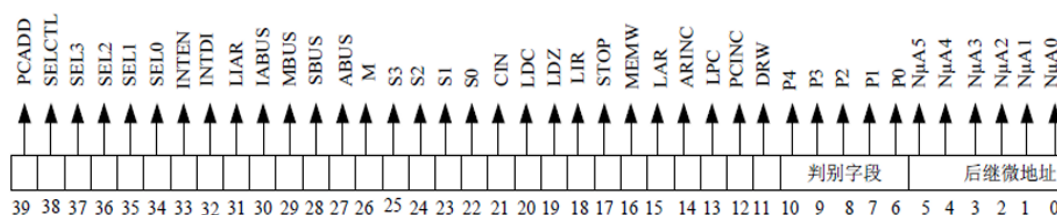
掌握微程序控制器的基本结构和功能，理解微程序控制器的设计原理和方法，熟悉微指令、微程序、微地址、微操作等基本微程序控制部件的实现方式。

二、实验内容：

用微程序控制器控制数据通路，将相应的信号线连接，构成一台能运行测试程序的 CPU；
执行一个简单的程序，掌握机器指令与微指令的关系；掌握中断原理；
理解控制器根据不同的指令类型，从微程序存储器中读取相应的微指令，生成控制信号，驱动数据通路工作的原理。
牢固建立计算机整机概念。

三、电路分析：

根据机器指令功能、格式和数据通路所需的控制信号，TEC-PLUS 采用如图所示的微指令格式。
微指令字长 40 位，顺序字段 11 位（判别字段 P4~P0，后继微地址 N μ A5~N μ A0），控制字段 29 位，微命令直接控制。



控制存储器由 5 片 58C65 组成。

58C65 是一种 8K \times 8 位的 E2PROM 器件，

TEC-PLUS 使用其中 64 个字节作为控制存储器，微地址 μ A5~ μ A0。

微地址寄存器 μ AR 由 D 触发器 74LS174 组成。

当按下复位按钮 CLR 时，产生的信号 CLR#(负脉冲)使微地址寄存器复位， μ A5~ μ A0 为 00H，在一条微指令结束时，用 T3 的下降沿将微地址转移逻辑产生的下条微指令地址 N μ A5、N μ A4~T~N μ A0~T 写入微地址寄存器。

微地址转移逻辑由若干与门和或门组成，实现“与~或”逻辑。

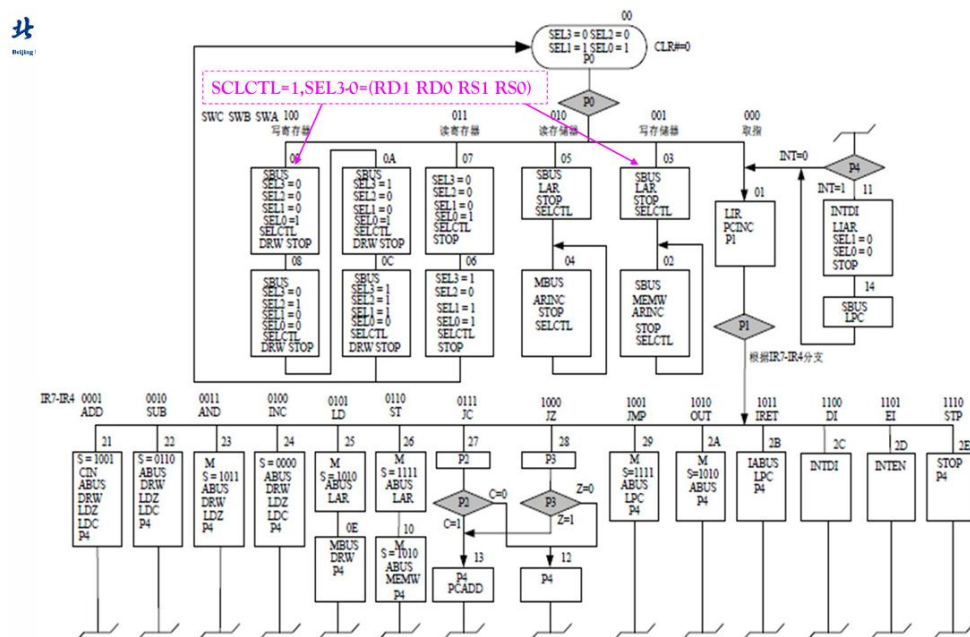
四、实验任务：

熟悉微程序流程图和微程序代码表。

正确设置模式开关 SWC、SWB、SWA，用单微指令方式（单拍开关 DP 设置为 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，记录下每一步的微地址 $\mu A5 \sim \mu A0$ 、判别位 $P4 \sim P0$ 和有关控制信号的值，写出这 4 种控制台操作的作用和使用方法。正确设置指令操作码 $IR7 \sim IR4$ ，用单微指令方式跟踪除停机指令 STP 之外的所有指令的执行过程。

记录下每一步的微地址 $\mu A5 \sim \mu A0$ 、判别位 $P4 \sim P0$ 和有关控制信号的值。

对于 JZ 指令，跟踪 $Z=1$ 、 $Z=0$ 两种情况；对于 JZ 指令，跟踪 $C=1$ 、 $C=0$ 两种情况。



五、实验步骤及数据

1. 实验台接线

| | | | | | | |
|------|-------|-------|-------|-------|-----|-----|
| 控制器 | IR4-I | IR5-I | IR6-I | IR7-I | C-I | Z-I |
| 电平开关 | K0 | K1 | K2 | K3 | K4 | K5 |

接好线后，将编程开关拨到“正常”位置，控制转换开关拨到“微程序”位置，合上电源，按 CLR#按钮，使 TEC-PLUS 实验系统处于初始状态。

2. 操作流程（实验 2、3 有具体数据）

写存储器(WRM)：按下复位按钮 CRL#，置 SWC=0，SWB=0，SWA=1。

①在 SW7~SW0 中设置存储器地址，按 QD 按钮将此地址打入 AR。

②在 SW7~SW0 置好数据，按 QD 按钮，将数据写入 AR 指定的存储器单元，这时 AR 加 1。

③返回②。依次进行下去，直到按复位按钮 CRL#为止。

实现对 RAM 的手动写入（主要是自己编写的程序和数据）。

启动程序(PR)：按下复位按钮 CRL#后，微地址寄存器清零。

置 SWC=0，SWB=0，SWA=0，用数据开关 SW7~SW0 设置 RAM 中的程序首地址，按 QD 按钮后，启动程序执行。

观察不同机器指令对应微程序的执行。



微程序控制器操作模式

| 操作模式 | 实验功能 | 备注 |
|------|----------|----|
| 000 | 启动程序运行 | |
| 001 | 写存储器 | |
| 010 | 读存储器 | |
| 011 | 读寄存器 | |
| 100 | 写寄存器 | |
| 101 | 运算器组成实验 | |
| 110 | 双端口存储器实验 | |
| 111 | 数据通路实验 | |

六、实验总结

了解了微指令、微程序、微地址、微操作等基本微程序控制部件的实现方式

实验五 CPU 组成与机器指令执行

一、实验目标：

将微程序控制器同执行部件（整个数据通路）联机，组成一台模型计算机；用微程序控制器控制模型机数据通路，将相应的信号线连接，构成一台能运行测试程序的 CPU。

通过 TEC-PLUS 执行由机器指令组成的简单程序。

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后自动取出下一条指令并执行，牢固建立的计算机整机概念。

二、实验要求：

将简单程序进行译码，按指令格式汇编成二进制机器代码。

完成控制台、时序部件、数据通路和微程序控制器之间的连线。

将程序机器代码利用控制台指令写入内存。

根据程序的需要设置通用寄存器堆中相关寄存器的数据。

单拍方式执行一遍程序，记录相关寄存器和存储器存储单元数据，与理论值比较分析。

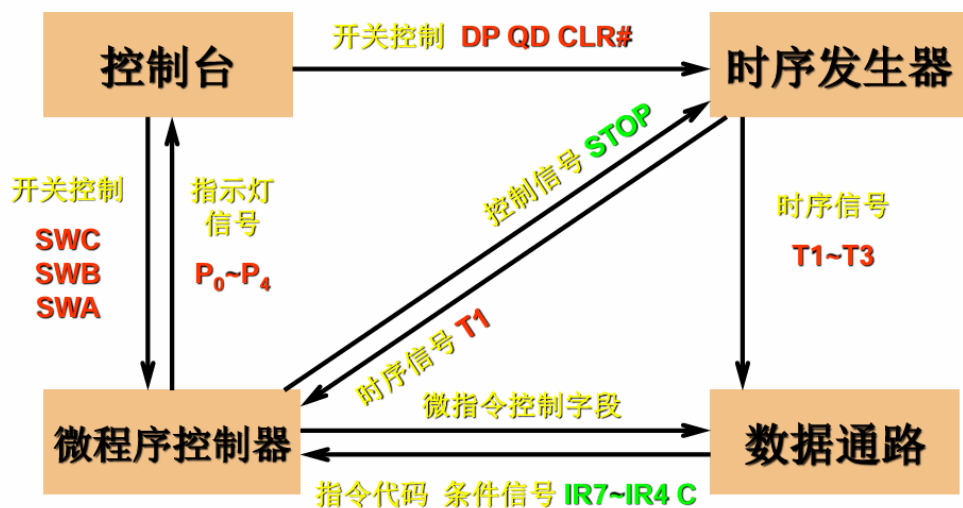
连续方式再次执行一遍程序，记录相关寄存器和存储器存储单元数据，与理论值比较分析。

三、实验任务：

在单拍方式下跟踪程序的执行

在连续方式下运行程序

四、实验内容：



五、实验步骤和数据

1. 实验台接线

| | | | | | | |
|------|-------|-------|-------|-------|-----|-----|
| 控制器 | IR4-1 | IR5-1 | IR6-1 | IR7-1 | C-1 | Z-1 |
| 数据通路 | IR4-0 | IR5-0 | IR6-0 | IE7-0 | C-0 | Z-0 |

2. 在单拍方式下跟踪程序的执行。

- (1)通过写存储器操作将程序写入存储器。
- (2)通过读操作将程序逐条读出，检查程序是否正确写入了存储器。
- (3)通过写寄存器操作设置寄存器 R2 为 12H、R3 为 0FH。
- (4)通过读寄存器操作检查设置是否正确。
- (5)将单拍开关 DP 设置为 1，使程序在单微指令下运行。
- (6)按复位按钮 CLR，复位程序计数器 PC 为 00H。将模式开关设置为 SWC=0、SWB=0、SWA=0，准备进入程序运行模式。
- (7)按一次 QD 按钮，进入程序运行。每按一次 QD 按钮，执行一条微指令，直到程序结束。在程序执行过程中，记录下列信号的值：PC7~PC0、AR7~AR0、 μ A5~ μ A0、IR7~IR0、A7~A0、B7~B0 和 D7~D0。

(8)通过读寄存器操作检查 4 个寄存器的值并记录。

(9)通过读存储器操作检查存储单元 12H、13H 的值并记录。

3. 在连续方式下运行程序

由于单拍方式下运行程序并没有改变存储器中的程序。因此 只要重新设置 R2 为 12H、R3 为 0FH。然后将单拍开关 DP 设置 为 0，按复位按钮 CLR 后，将模式开关设置为 SWC=0、 SWB=0、SWA=0，准备进入程序运行模式。按一次 QD 按钮， 程序自动运行到 STP 指令。通过读寄存器操作检查 4 个寄存器 的值并记录。通过读存储器操作检查存储单元 12H、13H 的值并记录



测试程序

#预习时要求完成的手工汇编#

| 地址 | 指令 | 机器16进制代码 |
|-----|-------------|----------------|
| 00H | LD R0, [R3] | 0101 0011 (53) |
| 01H | INC R3 | 0100 1100 (4C) |
| 02H | LD R1,[R3] | 0101 0111 (57) |
| 03H | SUB R0,R1 | 0010 0001 (21) |
| 04H | JZ 0BH | 1000 0110 (86) |
| 05H | ST R0,[R2] | 0110 1000 (68) |
| 06H | INC R3 | 0100 1100 (4C) |
| 07H | LD R0, [R3] | 0101 0011 (53) |
| 08H | ADD R0, R1 | 0001 0001 (11) |
| 09H | JC 0CH | 0111 0010 (72) |
| 0AH | INC R2 | 0100 1000 (48) |
| 0BH | ST R2,[R2] | 0110 1010 (6A) |
| 0CH | AND R0,R1 | 0011 0001 (31) |
| 0DH | OUT R2 | 1010 0010 (A2) |
| 0EH | STP | 1110 0000 (E0) |
| 0FH | 85H | 1000 0101 (85) |
| 10H | 23H | 0010 0011 (23) |
| 11H | 0EFH | 1110 1111 (EF) |
| 12H | 00H | 0000 0000 (00) |

4. 结果记录

(1)通过写存储器操作将程序写入存储器。

(2)通过读操作将程序逐条读出，检查程序是否正确写入了存储器。

(3)通过写寄存器操作设置寄存器 R2 为 12H、R3 为 0FH。

(4)通过读寄存器操作检查设置是否正确。

按复位按钮 CLR，复位程序计数器 PC 为 00H。将模式开关设置为 SWC=0、SWB=0、SWA =0

每按一次 QD 按钮，执行一条微指令，直到程序结束。记录下列信号的值：

PC7~PC0、AR7~AR0、 μ A5~ μ A0、IR7~IR0、A7~A0、B7~B0 和 D7~D0。

| QD | PC7~PC0 | AR7~AR0 | μ A5~ μ A0 | IR7~IR0 | A7~A0 | B7~B0 | D7~D0 |
|----|---------|---------|--------------------|---------|-------|-------|-------|
| ↓ | 00H | 00H | 01H | 00H | 00H | 00H | F0H |
| ↓ | 01H | 00H | 25H | 53H | 00H | 0FH | 0FH |
| ↓ | 01H | 0FH | 0EH | 53H | 00H | 0FH | 0FH |
| ↓ | 01H | 0FH | 01H | 53H | 85H | 0FH | F0H |
| ↓ | 02H | 0FH | 24H | 4CH | 0FH | 85H | 10H |
| ↓ | 02H | 0FH | 01H | 4CH | 10H | 85H | F0H |
| ↓ | 03H | 0FH | 25H | 57H | 00H | 10H | 10H |
| ↓ | 03H | 10H | 0EH | 57H | 00H | 10H | 21H |
| ↓ | 03H | 10H | 01H | 57H | 00H | 10H | F0H |
| ↓ | 04H | 10H | 22H | 21H | 85H | 23H | 62H |
| ↓ | 04H | 10H | 01H | 21H | 62H | 23H | F0H |
| ↓ | 05H | 10H | 28H | 86H | 23H | 12H | F0H |
| ↓ | 05H | 10H | 22H | 86H | 23H | 12H | F0H |
| ↓ | 05H | 10H | 01H | 86H | 23H | 12H | F0H |
| ↓ | 06H | 10H | 26H | 68H | 12H | 62H | 12H |
| ↓ | 06H | 12H | 10H | 68H | 12H | 62H | 62H |

| | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|
| ↓ | 06H | 12H | 01H | 68H | 12H | 62H | F0H |
| ↓ | 07H | 12H | 24H | 4CH | 10H | 62H | 11H |
| ↓ | 07H | 12H | 01H | 4CH | 11H | 62H | F0H |
| ↓ | 08H | 12H | 25H | 53H | 62H | 11H | 11H |
| ↓ | 08H | 11H | 0EH | 53H | 62H | 11H | EFH |
| ↓ | 08H | 11H | 01H | 53H | EFH | 11H | F0H |
| ↓ | 09H | 11H | 21H | 11H | EFH | 23H | 12H |
| ↓ | 09H | 11H | 01H | 11H | 12H | 13H | F0H |
| ↓ | 0AH | 11H | 27H | 72H | 12H | 12H | F0H |
| ↓ | 0AH | 11H | 13H | 72H | 12H | 12H | F0H |
| ↓ | 0CH | 11H | 01H | 72H | 12H | 12H | F0H |
| ↓ | 0DH | 11H | 23H | 31H | 12H | 13H | 02H |
| ↓ | 0DH | 11H | 01H | 31H | 02H | 23H | F0H |
| ↓ | 0EH | 11H | 2AH | A2H | 02H | 12H | 12H |
| ↓ | 0EH | 11H | 01H | A2H | 02H | 12H | F0H |
| ↓ | 0FH | X | X | X | X | X | X |

5、

令 SWC=1、SWB=1、SWA=0

读出 R0:02H,R1:23H,R2:12H,R3:11H

令 SWC=0、SWB=1、SWA=0

读出[12H]=62H,[13H]=00H(无关)

六、实验总结

CPU 的基本功能：

程序的顺序执行、管理控制操作、操作和执行定时、数据加工等。

控制器：程序计数器，指令寄存器，指令译码器，时序发生器，操作控制器等，取指、译码产生控制信号，控制 CPU、内存，输入/输出之间的数据传输。

运算器：算数逻辑单元 ALU，累加器 AC，数据缓冲寄存器 DR，状态条件寄存器 PSW 等，数据加工，算数逻辑运算，产生运算结果或逻辑判断。

从内存取指到执行指令结束，一条机器指令对应一个微程序。

一个机器指令周期由几个微指令周期组成。

控制部件和执行部件构成计算机的两大部分。

实验六、中断原理

一、实验目标：

从硬件、软件结合的角度，模拟单级中断和中断返回的过程；

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；

了解微程序控制器与中断控制器协调的基本原理；

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。

二、实验内容：

INT 为 1 转到微地址 11H，该微指令产生 INTDI 信号，禁止新的中断发生，产生 LIAR 信号保存当前地址(断点寄存器)，产生 STOP 信号，等待手动设置中断向量(数据开关 SD7~SD0 设置中断地址)，机器将中断向量读到 PC 后，转到中断服务程序继续执行。

执行指令 IRET，从中断地址返回，该指令产生 IABUS 信号，恢复断点地址，产生信号 LPC，将断点从数据总线装入 PC，恢复被中断的程序。

发生中断时，关中断由硬件负责。而中断现场(包括 4 个寄存器、进位标志 C 和结果为 0 标志 Z)的保存和恢复由中断服务程序完成。

中断服务程序的最后两条指令一般是开中断指令 EI 和中断返回指令 IRET。

为了保证从中断服务程序能够返回到主程序，EI 指令执行后，不允许立即被中断。

因此，EI 指令执行过程中的最后一条微指令中不包含 P4 判别位。

三、电路分析：

TEC-PLUS 的中断系统只支持单级中断、单个中断请求，有中断屏蔽功能。系统有 2 条指令用于允许和屏蔽中断（DI 关中断 指令、EI 开中断指令）。在时序发生器中，设置了一个允许中断触发器 EN_INT（DIor!EI），当它为 1 时，允许中断，当它为 0 时，禁止中断发生。复位脉冲 CLR#使 EN_INT 复位为 0。中断地址寄存器 IAR 是 74LS374。当信号 LIAR 为 1 时，在 T3 的上升沿，将 PC 保存在 IAR 中。当信号 IABUS 为 1 时，IABUS 中保存的 PC 送数据总线 DBUS。由于本实验系统只有一个断点寄存器而无堆栈，因此仅支持一级中断而不支持多级中断。

中断向量即中断服务程序的入口地址，本实验系统中由数据开关 SD7~SD0 提供。

四、实验任务：

为了保证此程序能够循环执行，应当将 R1 预先设置为 01H。R0 的初值设置为 0。将 TEC-PLUS 连接成一个完整的模型计算机。将主程序和中断服务程序装入存储器，开关 DP 设置为连续运行方式 (DP=0)，复位系统按 QD 按钮，启动程序从 00H 开始执行。PULSE 按钮，产生一个中断请求信号 PULSE，中断主程序的运行。记录断点 PC、R0 (指示灯 A7~A0 上显示) 的值。单拍开关 DP 设置为单拍方式 (DP=1)，在数据开关上设置中断服务程序的入口地址 45H。按 QD 按钮，一步步执行中断服务程序，直到返回到断点为止。将存储器 00H 的指令改为 DI，重新运行程序，记录发生的现象。列表记录中断有关信号的变化情况。特别记录好断点和 R0 的值。

五、实验步骤：

1. 试验台接线

| | | | | | | |
|------|-------|-------|-------|-------|-----|-----|
| 控制器 | IR4-1 | IR5-1 | IR6-1 | IR7-1 | C-1 | Z-1 |
| 数据通路 | IR4-0 | IR5-0 | IR6-0 | IE7-0 | C-0 | Z-0 |

接好线后, 将编程开关拨到“正常”位置, 控制转换开关拨到“微程序”位置，合上电源, 按 CLR#按钮, 使 TEC-PLUS 实验系统处于初始状态。

2. 记录数据

19:40

计算机组成与体系结构2025-未来学院 - 第二单元 .pdf

文件预览

北京邮电大学

测试程序

#预习时要求完成的手工汇编#

| 地址 | 指令 | 机器代码 |
|-----|------------|----------------|
| 00H | EI | 1101 0000 (09) |
| 01H | INC R0 | 0100 0000 (40) |
| 02H | INC R0 | 0100 0000 |
| 03H | INC R0 | 0100 0000 |
| 04H | INC R0 | 0100 0000 |
| 05H | INC R0 | 0100 0000 |
| 06H | INC R0 | 0100 0000 |
| 07H | INC R0 | 0100 0000 |
| 08H | INC R0 | 0100 0000 |
| 09H | JMP [R1] | 1001 0100 (99) |
| 45H | ADD R0, R0 | 0001 0000 (10) |
| 46H | EI | 1101 0000 |
| 47H | IRET | 1011 0000 (B9) |

71

北京邮电大学

- (1)通过写存储器操作将程序写入存储器。
- (2)通过读操作将程序逐条读出，检查程序是否正确写入了存储器。
- (3)通过写寄存器操作设置寄存器 R1 为 01H。R0 的初值设置为 0。
- (4)通过读寄存器操作检查设置是否正确。

| QD | DP | PULSE | PC7~PC0 | A7~A0 | B7~B0 | D7~D0 | SW7~SW0 |
|--------|----|-------|---------|-------|-------|-------|---------|
| ↓ | 1 | X | X | X | X | X | 00H |
| X | 1 | ↓ | 31H | 12H | 12H | F0H | X |
| X | 0 | X | 31H | 12H | 12H | F0H | 45H |
| ↓ | 0 | X | 45H | 12H | 12H | F0H | X |
| ↓ | 0 | X | 45H | 12H | 12H | F0H | X |
| ↓ | 0 | X | 45H | 24H | 24H | F0H | X |
| ↓ | 0 | X | 46H | 24H | 24H | F0H | X |
| 省略中间步骤 | X | X | X | X | X | X | X |
| | X | X | X | X | X | X | X |
| | X | X | X | X | X | X | X |
| ↓ | 0 | X | 47H | 24H | 24H | F0H | X |
| 省略中间步骤 | X | X | X | X | X | X | X |

| | | | | | | | |
|---|---|---|-----|-----|-----|-----|---|
| | X | X | X | X | X | X | X |
| | X | X | X | X | X | X | X |
| ↓ | 0 | X | 48H | 24H | 24H | FOH | X |
| ↓ | 0 | X | 31H | 24H | 24H | FOH | X |

跳转回了原中断地址

3.

令 SWC=1、SWB=1、SWA=0

读出 R0:24H

六、实验总结：

了解了 TEC-PLUS 模型计算机的中断机制和中断处理过程，掌握中断控制器、中断向量、中断屏蔽等概念，了解微程序控制器与中断控制器协调的基本原理；