

# 同济大学实验报告纸

软件工程专业 2026 届 1 班 250756 姓名 林继伟 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023 年 12 月 6 日

## [实验目的]

1. 掌握微程序控制器的组成和工作原理
2. 理解微指令编写格式
3. 理解微程序运行过程

## [实验设备]

组成原理实验箱 TD-CMA

## [实验原理]

1. 微程序控制器的组成和基本功能

### (1) 控制器基本功能

控制器是计算机的核心部件，用来实现对计算机各部件的有效控制，完成计算机程序规定的一系列操作。

根据指令操作码和时序信号，产生各种控制信号，完成取指令和执行指令的过程。

早期计算机采用的是组合逻辑控制器，运行速度快但设计负责，可维护性差，可扩展性差，目前广泛采用的是微程序控制器。

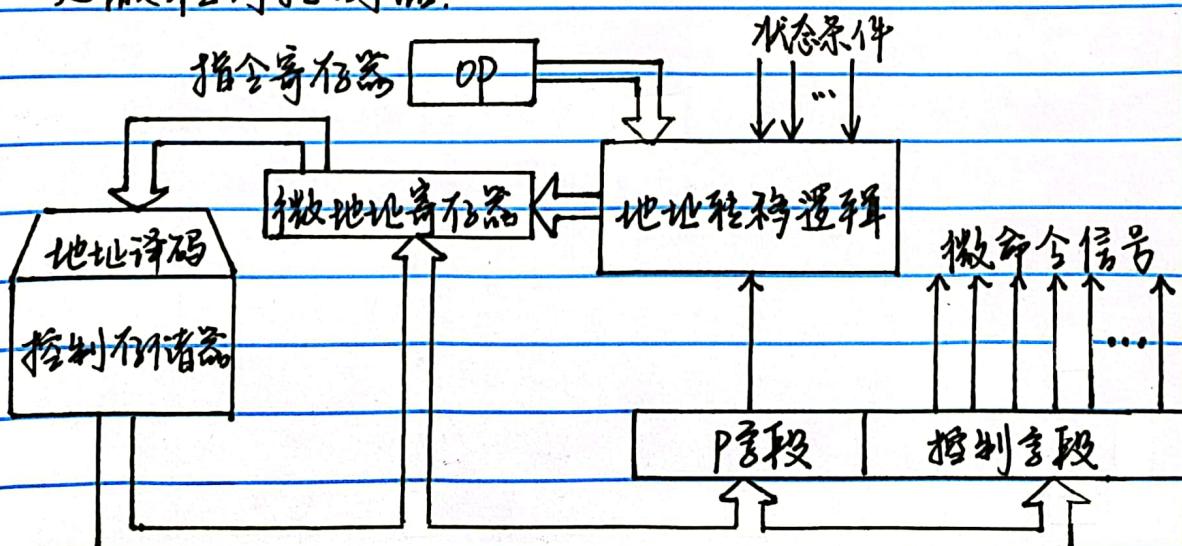


图1 微程序控制器组成原理框图。

# 同济大学实验报告纸

软件工程专业 2026届 1 班 2250758 姓名 林继申 第 组 同组人员  
课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023年 12月 6 日

## (2) 微程序控制的基本思想

把操作控制信号编成一条“微指令”存放在一个“只读存储器”中。当计算机运行的时候，按照一定的顺序逐条读出这些微指令，从而产生各种操作控制信号，使相应部件执行所规定的操作。

## (3) 微程序控制器基本任务

① 完成当前指令的翻译与执行

② 将当前指令功能转换为硬件逻辑部件工作的控

制信号

③ 完成数据的传递、处理和存储操作。

## (4) 微程序控制器的相关概念

① 微命令(微信号)：直接作用于硬件控制电路的控  
制命令(信号)

② 微操作：由微命令控制实现的一些最基本的操作。

(如取指令地址、读取指令、取操作数地址、取操作数、  
基本运算、写回结果等都是微操作)。一条微命令对应  
一个微操作。

③ 微指令：一个微周期所需的微命令的集合，用二进制编码表示。可以同时发出多个控制信号，执行一  
组微操作。一个典型的加法指令的执行可分为：取指  
令、计算地址、取操作数、执行加法运算、返回计算  
结果等步骤，每一步都由一条微指令实现。

④ 微周期：读取一条微指令并完成相应微操作所用  
的时间，通常为一个机器周期。

⑤ 微程序：若干条微指令的有序序列，每条机器指

# 同济大学实验报告纸

软件工程专业2026届 1 班250758姓名 林继申 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023年12月6日

令对应一段微程序，一条机器指令需要由多个微周期来完成。

⑥概念总结（自顶向下）：一段计算机程序由一系列机器指令组成，每一条机器指令由一段微程序实现。每一段微程序由一组微指令表示，每一条微指令由一系列微操作组成，一个微操作对应一条微命令。

## 1.5) 微程序控制器组成

### ① 控制存储器 CM (24位)

用于存放微程序，每个单元存放一条微指令代码。  
可读。

### ② 微指令寄存器 MIR

存放由控制存储器读出的一条微指令信息，包括操作控制字段和判别测试字段。

### ③ 微地址寄存器 MAR (6位)

用于存放下一条微指令的地址。

## 2. 微指令24位编写格式

微指令存放在控制存储器 CM 里，微指令字长共 24 位。

rr	rr	21	20	19	18-15	14-12	11-9	8-6	5-0
x	x	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

表1 控制位顺序表

其中 MA5-MA0 为 6 位的后续微地址，A、B、C 为三个译码字段，分别由三个控制位译码到多位。

C 字段中的 P<1> 为测试字段。

功能是根据机器指令及相应微代码进行译码，使微程序转入相应的微地址入口，从而实现完成对指令的识别，并实现微程序的分支。

# 同济大学实验报告纸

软件工程 专业 2026 届 1 班 225758 姓名 林继坤 第 1 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023 年 12 月 6 日

A 存段				B 存段				C 存段			
14	13	12	选择	11	10	9	选择	8	7	6	选择
0	0	0	MOP	0	0	0	MOP	0	0	0	MOP
0	0	1	LDA	0	0	1	AU-B	0	0	1	P<1>
0	1	0	LDR	0	1	0	RO-B	0	1	0	保留
0	1	1	LDRD	0	1	1	保留	0	1	1	保留
1	0	0	保留	1	0	0	保留	1	0	0	保留
1	0	1	保留	1	0	1	保留	1	0	1	保留
1	1	0	保留	1	1	0	保留	1	1	0	保留
1	1	1	WIR	1	1	1	保留	1	1	1	保留

表 2 微指令格式

## 3. 微程序的地址转移逻辑

在一般情况下，微指令由控制存储器读出后直接给出下一条微指令的地址，即微地址，存放于微地址寄存器中。(T2)

如果微程序不出现分支，那么下一条微指令的地址就直接由微地址寄存器给出。

如果微程序出现分支，意味着微程序出现条件转移，在这种情况下，通过判别测试字段 P 和执行部件的状态条件信息，去修改微地址寄存器的内容，并按修改后的微地址读取下一条指令，从而实现地址转移。(T4)

## 4. 微程序运行流程

本实验安排了 4 条机器指令，分别为 ADD(0000 0000)，IN(0010 0000)，OUT(0011 0000) 和 HLT(0101 0000)，指令格式如下：

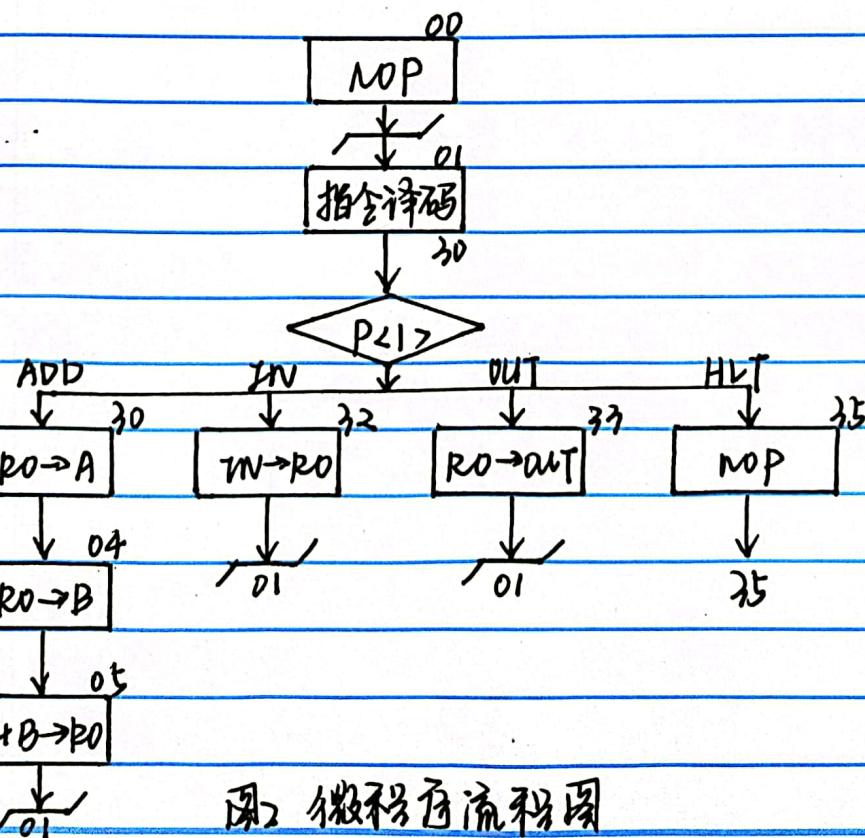
# 同济大学实验报告纸

软件工程 专业 2026 届 1 班 250758 姓名 林杨坤 中第 组 同组人员  
 课程名称 计算机组装与原理实验 实验名称 微程序控制器实验 实验日期 2023 年 12 月 6 日

助记符	机器指令码	说明	
IN	0010 0000	$IN \rightarrow R0$	
ADD	0020 0000	$R0 + R0 \rightarrow R0$	
OUT	0011 0000	$R0 \rightarrow OUT$	
HLT	0100 0000	停机	

表3 指令格式

几条机器指令对应的参考微程序流程图如图所示。图中一个矩形方框表示一条微指令，方框中的内容为该指令执行的微操作。右上角的数字是该条指令的微地址，右下角的数字是该条指令的后续微地址。所有微地址均用16进制表示。向下的箭头指出了下一条要执行的指令。P<1>为测试字，根据条件使微程序产生分支。



# 同济大学实验报告纸

软件工程 专业 2026 届 1 班 250758 姓名 林继坤 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023 年 12 月 6 日

将全部微程序按微指令格式变成二进制微代码，可得  
到二进制代码表。

地址	十六进制	高三位	S3-S0	A3段	B3段	C3段	MAS-MAO
00	00 00 01	00000	0000	000	000	000	000001
01	00 70 70	00000	0000	111	000	001	110000
04	00 24 05	00000	0000	010	010	000	000101
05	04 B2 01	00000	1001	011	001	000	000001
30	00 14 04	00000	0000	001	010	000	000100
32	18 30 01	00011	0000	011	000	000	000001
33	28 04 01	00101	0000	000	010	000	000001
35	00 00 35	00000	0000	000	000	000	110101

表4 二进制微代码表

## 【实验内容】

### 1. 实验步骤

①对照微程序流程图和二进制代码，理解微程序的编写格式

②连接实验电路

③在“编程”模式下写入微代码

④在“校验”模式下写入微代码

⑤在“运行”模式下写入微代码（单拍/单步）

### 2. 手动写入微指令

### 3. 手动校验微指令

### 4. 联机写入微程序

①[转储]-[装载]“sample\微程序控制器实验.txt”

②注意观察软件界面中的输出区和左边微地址栏的信息

# 同济大学实验报告纸

软件工程 专业 2026 届 1 班 2250758 姓名 林继冲 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023 年 12 月 6 日

## 5. 联机运行

- ① [实验]-[微控器实验]，打开实验通路图
- ② KK1 运行、KK3 运行
- ③ CON 单元 CLR 清零
- ④ 运行软件（单节拍、单周期、单机器指令）

## 6. 手动运行（观察后续微地址如何产生）

- ① KK1 运行、KK3 运行
- ② CON 单元 CLR 清零、KK2 单拍、按动 ST 开关。
- ③ T2 节拍：微控器将后续微地址打入微地址寄存器。  
当前微指令打入微指令寄存器。
- ④ T3、T4 节拍：根据 T2 节拍产生的控制信号做出相应的执行动作，如果测试信号有效，则根据机器指令及当前微地址寄存器中的内容进行译码，使微程序转入相应的微地址入口，实现微程序的分支。

## 7. 强制运行（观察机器指令何时结束）

- ① CON 单元清零、KK2 单步
- ② IN 单元置数、按动 ST 按钮
- ③ 当后级微地址显示为 100001 时，在 CON 单元的 SD27...SD20 模拟给出 IN 指令 00100000，在 T3 被打入指令寄存器 (IR)
- ④ 当后级微地址显示为 100001 时，说明当前指令已执行完毕。
- ⑤ 继续在 CON 单元的 SD27...SD20 给出其他几条指令。

## 8. 微程序控制器工作过程

- ① 机器周期开始，执行公共取址操作。

# 同济大学实验报告纸

软件工程专业 2026届 1 2250758 姓名 林继申 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023年 12月 6 日

- ② 该机器指令 送入指令寄存器 (同时修改 PC)
- ③ 进行指令译码，取出操作码产生对应的微程序入口地址，送入微地址寄存器。
- ④ 取出对应的一条微指令，送入微指令寄存器
- ⑤ 微指令操作字段经译码产生一组微命令，送往相应的执行部件
- ⑥ 在时序控制下完成微操作 (T2/T4)
- ⑦ 产生后续微地址，读取下一条微指令执行完一段微程序后，开始新的机器周期。

## 9. 微地址标号为 01、05、32 三种微指令的含义

### (1) 地址标号 01 微指令的含义

高五位	S3-S0	A 字段	B 字段	C 字段	M A5-M A0
00000	0000	111	000	001	110000

表5 地址标号 01 微指令格式

高五位均为 0，说明此时是对内存操作，不读不写。S3-S0 均为 0，此时是 F=A（直通），A 字段 111 代表 LDTR，B 字段 000 无操作，C 字段 001 代表测试字位有效，M A5-M A0 代表下一个微地址为 30（十六进制）。综上，该条微指令是将机器指令输入 R 并根据该指令及相应微代码进行判断使微程序转入相应微地址入口。

### (2) 地址标号 05 微指令的含义

高五位	S3-S0	A 字段	B 字段	C 字段	M A5-M A0
00000	1001	011	001	000	000001

表6 地址标号 05 微指令格式

高五位均为 0，说明此时是对内存操作，不读不写。S3-

# 同济大学实验报告纸

软件工程 专业 2026届 1 班 250758 姓名 林继坤 第 组 同组人员  
课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023年 12月 6 日

$S_0$  为 1001，此时是  $F = A + B$ 。A 字段 代表 LDRO，B 字段 001 代表 ALU-B。C 字段 000 无操作，MA5-MA0 代表下一个微地址为 01（十六进制）。综上，该条微指令作用是执行加法操作，加数来自于 RD，相加后结果从 ALU 流出并标明下一指令地址为 01。

## (3) 地址标号 32 指令的含义

高三位	S3-S0	A 字段	B 字段	C 字段	MA5-MA0
00011	0000	011	000	000	000001

表 7 地址标号 32 指令格式

高三位中，RD、IOM 为高电平，代表此时该 I/O 设备。S3-S0 为 0000，此时  $F = A$ （直通）。A 字段 011 代表 LDRO，B 字段 000 无操作，C 字段 000 无操作，MA5-MA0 代表下一个微地址为 01（十六进制）。综上，该条微指令作用是将 I/O 设备读取的数据加载到 RD 寄存器中。

## [实验小结]

在本次微程序控制器实验中，我掌握和理解微程序控制器的组成和工作原理。控制器是计算机的核心部件，用来实现对计算机各部件的有效控制，完成计算机程序规定的一系列操作。

我了解了微程序控制器的相关概念，其中包括微命令、微操作、微指令、微周期、微程序等。这些概念是自下向上的关系。

我学习了微指令 32 位编写格式，微指令存放在控制器 CM 里，微指令字长共 32 位。从低位到高位分别是 MA0-MA5、C 字段、B 字段、A 字段、S0-S3、IOM、RD、WR。

# 同济大学实验报告纸

软件工程 专业 2026 届 1 班 2250738 姓名 林诗中 第 组 同组人员

课程名称 计算机组成原理实验 实验名称 微程序控制器实验 实验日期 2023 年 12 月 6 日

在学习微程序的地址转移逻辑和微程序运行过程后，结合微程序流程图，我加深了对计算机组成原理知识的理解和体会。

本次实验的接线较为简单，但仍需要细心，在单拍执行时、T2 节拍结束后，我将机器指令手动打入，在 T3、T4 节拍时看到了地址转移过程，实验达到了理想的效果。分析微地址标志为 01、03、32 三条微指令的含义更加深了我对本次实验内容的掌握。本次实验加深了我对计算机组成原理知识的掌握，也提高了我的实验动手能力。