



河北师范大学软件学院
Software College of Hebei Normal University

计算机组成原理

第6章 中央处理器

6.4 微程序控制原理

微程序设计技术的实质是将程序设计技术和存储技术相结合，即用程序设计的思想方法来组织操作控制逻辑，将微操作控制信号按一定规则进行信息编码（代码化），形成控制字（微指令），再把这些微指令按时间先后排列起来，存放在一个只读存储器中。

6.4 微程序控制原理

每一条**机器指令**对应一段“**程序**”，该“程序”被存放在一个只读的**控制存储器**中，因为每段“程序”的执行结果是实现了一条机器指令的功能，所以我们将这些“程序”称为指令的**微程序**。

6.4 微程序控制原理

6.4.1 微程序控制的基本概念

1.微程序设计的提出与发展

微程序设计的概念和原理最早是由英国剑桥大学的M.V.Wilkes教授于1951年提出来的。1964年，IBM公司在IBM360系列机上成功地采用了微程序设计技术，解决了指令系统的兼容问题。70年代以来，由于VLSI技术的发展，推动了微程序设计技术的发展和应用。

6.4 微程序控制原理

2.基本术语

(1)微命令和微操作

一条机器指令可以分解成一个微操作序列，这些微操作是计算机中最基本的、不可再分解的操作。微命令是控制计算机各部件完成某个基本微操作的命令。

微命令和微操作是一一对应的。微命令是微操作的控制信号，微操作是微命令的操作过程。

6.4 微程序控制原理

微命令有兼容性和互斥性之分，**兼容性**微命令是指那些可以同时产生，共同完成某一些微操作的微命令；而**互斥性**微命令是指在机器中不允许同时出现的微命令。

6.4 微程序控制原理

(2) 微指令、微地址

微指令是指控制存储器中的一个单元的内容，即控制字，它是若干个微命令的集合。存放控制字的控制存储器的单元地址就称为微地址。

6.4 微程序控制原理

(2) 微指令、微地址

一条微指令通常至少包含两大部分信息：

① **操作控制字段**，又称微操作码字段，用以产生某一步操作所需的各微操作控制信号。

② **顺序控制字段**，又称微地址码字段，用以控制产生下一条要执行的微指令地址。

6.4 微程序控制原理

微指令有垂直型和水平型之分，垂直型微指令接近于机器指令的格式，每条微指令只能完成一个基本操作。水平型微指令则具有良好的并行性，每条微指令可以完成较多的基本操作。

6.4 微程序控制原理

(3)微周期

从控制存储器中读取一条微指令并执行相应的微命令所需的全部时间称为微周期。

(4)微程序

一系列微指令的有序集合就是微程序。一条机器指令对应于一段微程序。

6.4 微程序控制原理

程序



指令的集合 → 指令
(微程序)



微指令的集合 → 微指令



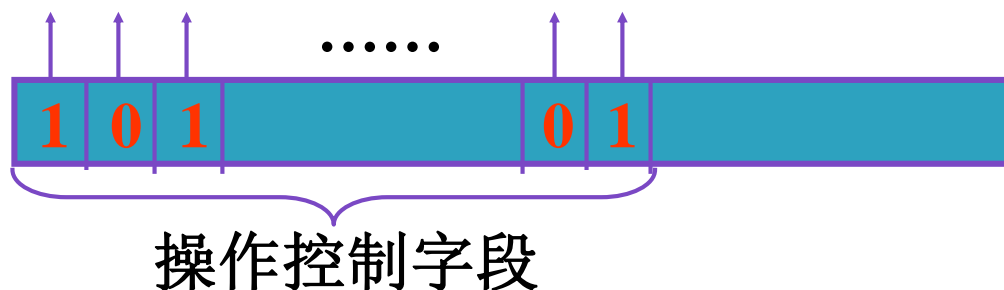
微命令的集合 → 微命令

6.4 微程序控制原理

6.4.2 微指令编码法

微指令编码法指的就是操作控制字段的编码方法。各类计算机的微指令编码法不同。

1. 直接控制法（不译码法）



6.4 微程序控制原理

2.最短编码法

最短编码使得微指令字最短。这种方法将所有的微命令统一编码，每条微指令只定义一个微命令。若微命令的总数为N，操作控制字段的长度为L，则：

$$L \geq \log_2 N$$

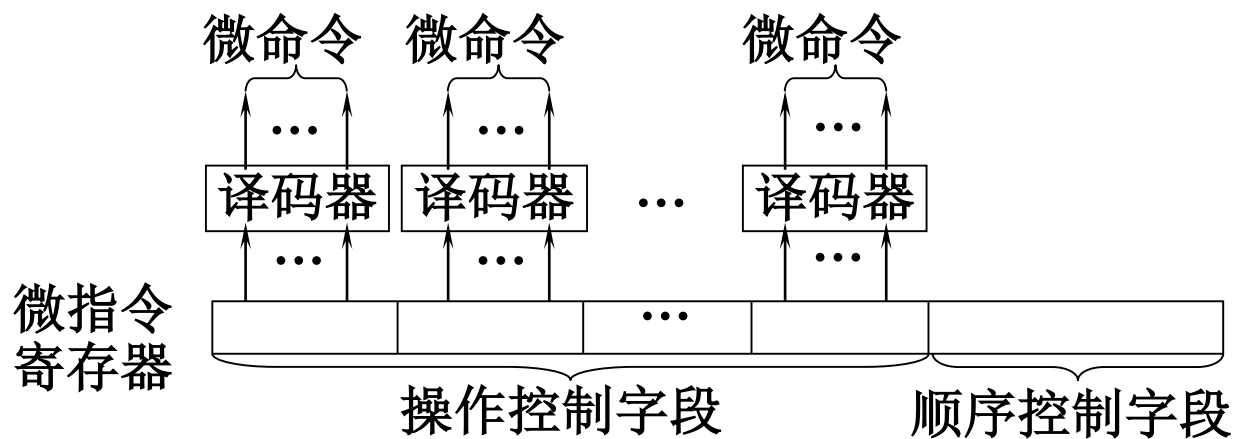
6.4 微程序控制原理

3. 字段编码法

前两种编码法的一个折衷的方法，既具有两者的优点，又克服了它们的缺点。这种方法将操作控制字段分为若干个小段，每段内采用最短编码法，段与段之间采用直接控制法。这种方法又可进一步分为字段直接编码法和字段间接编码法。

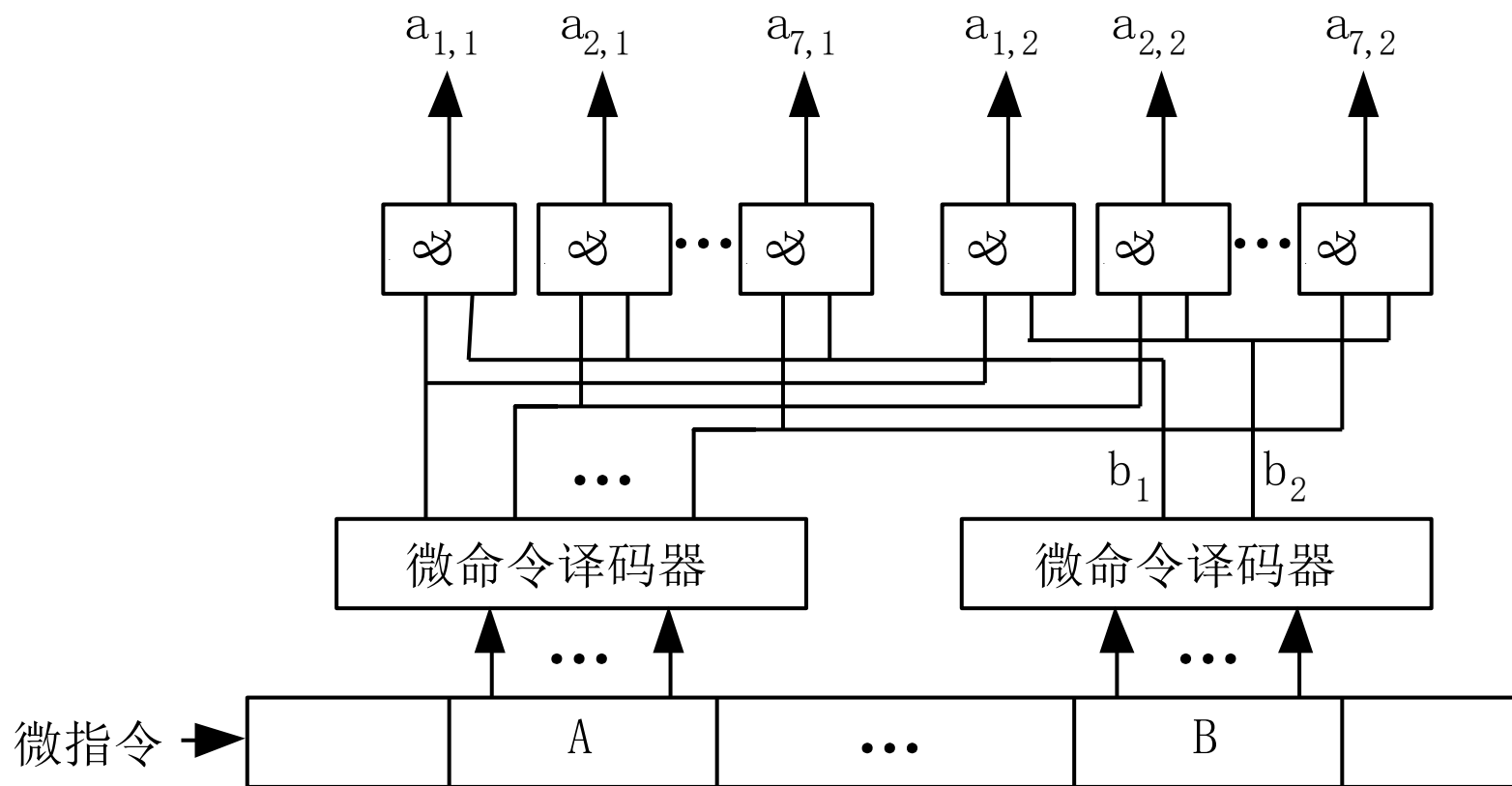
6.4 微程序控制原理

(1) 字段直接编码法



6.4 微程序控制原理

(2) 字段间接编码法



6.4 微程序控制原理

字段编码法中操作控制字段的分段原则：

① 把互斥性的微命令分在同一段内，兼容性的微命令分在不同段内。这样不仅有助于提高信息的利用率，缩短微指令字长，而且有助于充分利用硬件所具有的并行性，加快执行的速度。

6.4 微程序控制原理

② 应与数据通路结构相适应。

③ 每个小段中包含的信息位不能太多，否则将增加译码线路的复杂性和译码时间。

④ 一般每个小段还要留出一个状态，表示本字段不发出任何微命令。

6.4 微程序控制原理

例如：

- 运算器的输出控制信号有直传、左移、右移、半字交换等四个，这四个微命令是互斥的，它们可以安排在同一字段编码。
- 存储器的读和写命令也是一对互斥的微命令。
- $A \rightarrow C$ 、 $B \rightarrow C$ （ A 、 B 、 C 都是寄存器）这样的一类微命令也是互斥的微命令

6.4 微程序控制原理

假设某计算机共有256个微命令，如果采用直接控制法，微指令的操作控制字段就要有256位；而如果采用最短编码法，操作控制字段只需要8位就可以了；如果采用字段直接编码法，若4位为一个段，共需18段，操作控制字段只需72位，而且在同一时刻可以并行发出18个不同的微命令。

6.4 微程序控制原理

6.4.3 微程序控制器的组成和工作过程

1. 微程序控制器的基本组成

微程序控制器比组合逻辑控制器多出几个部件：

(1) 控制存储器 (CM)

(2) 微指令寄存器 (μIR)

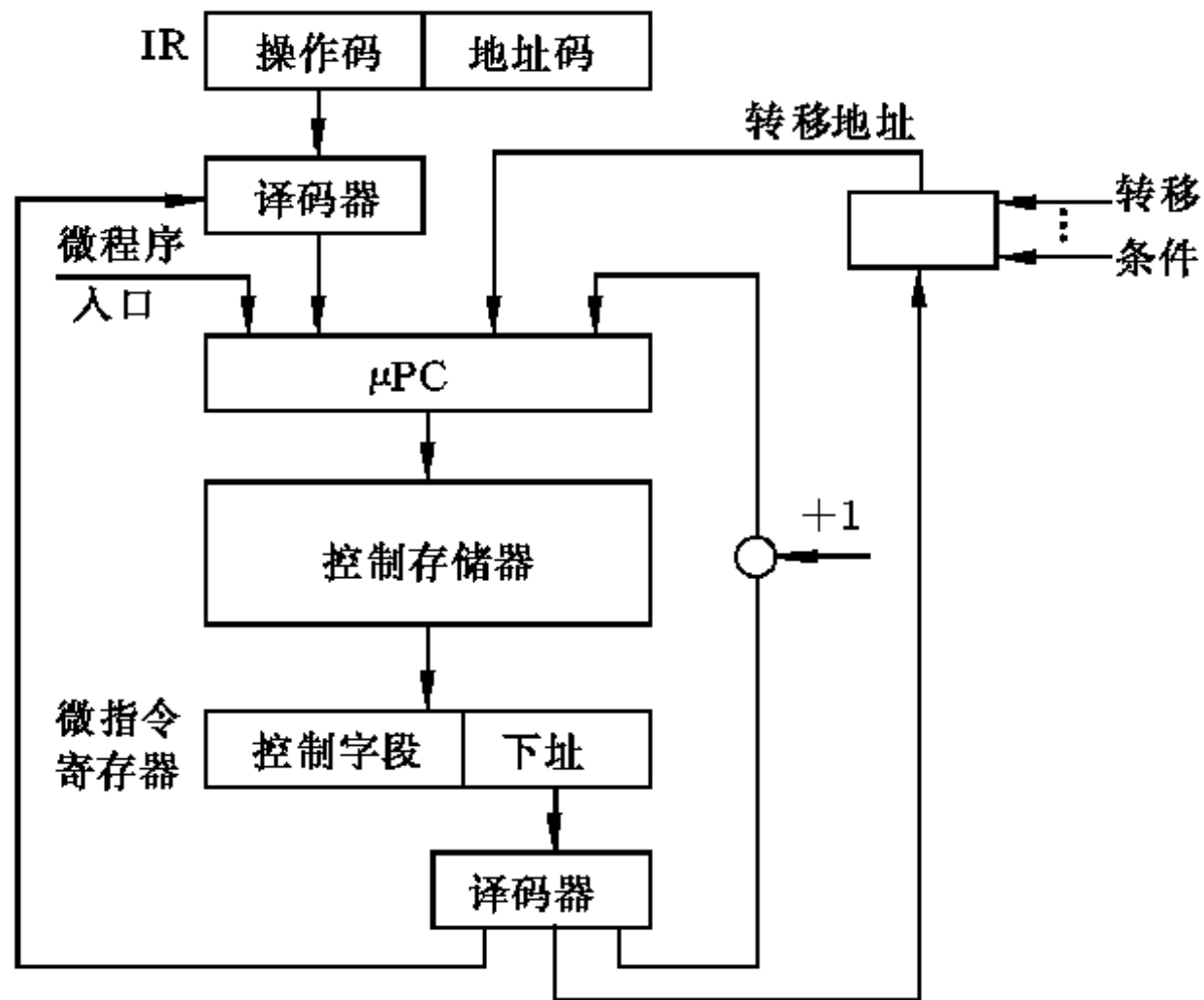
(3) 微地址形成部件

(4) 微地址寄存器 (μMAR)

6.4 微程序控制原理

2.微程序控制器的工作过程

微程序控制器的工作过程实际上就是在微程序控制器的控制下，计算机执行机器指令的过程。



6.4 微程序控制原理

(1) 执行取指令公操作。

(2) 由机器指令的操作码字段通过微地址形成部件产生出该机器指令所对应的微程序的入口地址，并送入 μMAR 。

(3) 从CM中逐条取出对应的微指令并执行。

6.4 微程序控制原理

(4) 执行完对应于一条机器指令的一段微程序后又回到取指微程序的入口地址，继续第(1)步，以完成取下条机器指令的公共操作。

以上是一条机器指令的执行过程，如此周而复始，直到整个程序执行完毕为止。

6.4 微程序控制原理

6.4.4 微程序入口地址的形成

每条机器指令对应一段微程序，当公用的取指微程序从主存中取出机器指令之后，由机器指令的操作码字段指出各段微程序的入口地址（初始微地址）。主要有三种转换方式

- 1.一级功能转换
- 2.二级功能转换
- 3.通过PLA电路实现功能转换

6.4 微程序控制原理

6.4.5 后继微地址的形成

找到初始微地址之后，可以开始执行微程序，每条微指令执行完毕都要根据要求形成后继微地址。后继微地址的形成方法对微程序编制的灵活性影响很大，它主要有两大基本类型：**增量方式和断定方式**。

6.4 微程序控制原理

6.4.6 微程序设计

1.微程序设计方法

(1)水平型微指令及水平型微程序设计

(2)垂直型微指令和垂直型微程序设计

第6章 小结

6.1 中央处理器的功能和组成

- ▶ 控制器的功能

指令流，数据流

- ▶ CPU中的主要寄存器

专用寄存器

第6章 小结

6.2 控制器的组成和实现方法

- ▶ 控制器的组成

- ▶ 控制器的硬件实现方法

组合逻辑控制器和微程序控制器的区别

6.3 时序系统与控制方式

- ▶ 时序系统

指令周期，机器周期

- ▶ 控制方式

同步，异步，联合

第6章 小结

- ▶ 指令执行的基本过程
- ▶ 指令的微操作序列

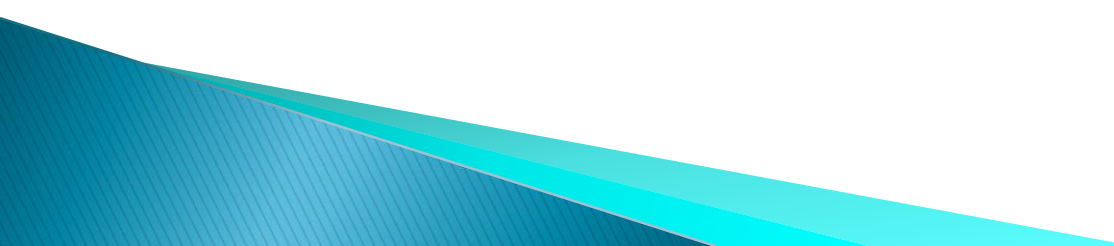
取指令操作

6.4 微程序控制原理

- ▶ 微程序控制的基本概念

微命令，微操作，微指令，微程序
两个层次

第6章 小结

- ▶ 微指令编码法
 - ▶ 微程序控制器的组成和工作过程
微程序控制器中各部件的作用
 - ▶ 微程序入口地址的形成
 - ▶ 后继微地址的形成
- 

- ▶ 某机采用微程序控制方式，微指令字长24位，水平型编码控制的微指令格式，断定方式，共有微命令30个，构成4个互斥类，各包含5个、8个、14个和3个微命令，外部条件共3个（直接编码）。
- ▶ (1) 控制存储器的容量为多少？
- ▶ (2) 设计出微指令的具体格式。

- ▶ 某计算机有8条微指令I1—I8，每条微指令所包含的微命令控制信号见下表，a—j 分别对应10种不同性质的微命令信号。假设一条微指令的控制字段仅限8位，请安排微指令的控制字段格式。

| 微指令 | a | b | c | d | e | f | g | h | i | j |
|----------------|---|---|---|---|---|---|---|---|---|---|
| I ₁ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | |
| I ₂ | ✓ | | | ✓ | | ✓ | ✓ | | | |
| I ₃ | | ✓ | | | | | | ✓ | | |
| I ₄ | | | ✓ | | | | | | | |
| I ₅ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| I ₆ | ✓ | | | | | | | ✓ | | ✓ |
| I ₇ | | | ✓ | ✓ | | | | ✓ | | |
| I ₈ | ✓ | ✓ | | | | | | ✓ | | |

- ▶ 已知某机采用微程序控制方式，控存容量为 1024×50 位，微程序可在整个控存中转移，控制转移的条件共 8 个（直接控制），微程序采用水平型格式。试说明微指令格式中各字段的位数，并画出相应的微程序控制器逻辑框图。

