



# 一、微程序控制的基本概念和工作原理

## 第一讲

### 微程序控制的基本概念和工作原理





## 7.3 微程序控制器

一

微程序控制的基本概念和工作原理

二

简单微程序控制器的设计

三

微程序设计技术

四

微程序控制方式下模型机的设计实例

五

模型机微程序设计

六

微程序控制器与硬布线控制器的比较





## 一、本节学习内容：

1. 以控制器的功能为切入点，从而引生出控制器的构架结构；
2. 微指令的形式、微程序的基本概念、原理与工作方式；
3. 微程序控制器的组成及工作原理；
4. 微程序设计方法
5. 模型机微程序控制器的设计方法





## 学习重点

1. 微程序控制器的组成及工作原理；
2. 微指令的形式、微程序的基本概念、原理与工作方式；
3. 掌握微程序设计方法





# 一、微程序控制的基本概念和工作原

控制器类型：

## 1. 硬件控制器（硬布线控制器）：

· 所有的机器指令从取指令、指令译码到产生指令所需要的控制信号均有硬件电路实现。

即：有组合逻辑电路和时序电路来实现。

· 特点：指令系统的指令越多，越复杂，电路也月复杂，

不同的指令系统，控制器也不同，

设计好后要再做修改、扩充：不方便、困难

。

## 2. 微程序控制器：

所有的机器指令由一段微程序来解释。





# 一、微程序控制的基本概念和工作原

名词定义与基本概论：

1 . **微操作**：指令执行必须完成的基本操作。

如取指令过程中的： $PC \rightarrow AR$ ,  $PC+1 \rightarrow PC$  等，

2 . **微命令**：即微操作的控制信号。

如存储器读  $MEMR\#$ ，存储器写  $MEMW\#$ ,  $PC\_B$  等控制信号等。

是形成微指令的最小单位。

3 . **微指令**：由一组完成一个或若干个微操作的微命令组成。

4 , **微程序**：解释一条机器指令一段微程序。

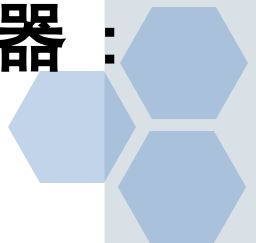
是微指令的有序集合。





## 一、微程序控制的基本概念和工作原

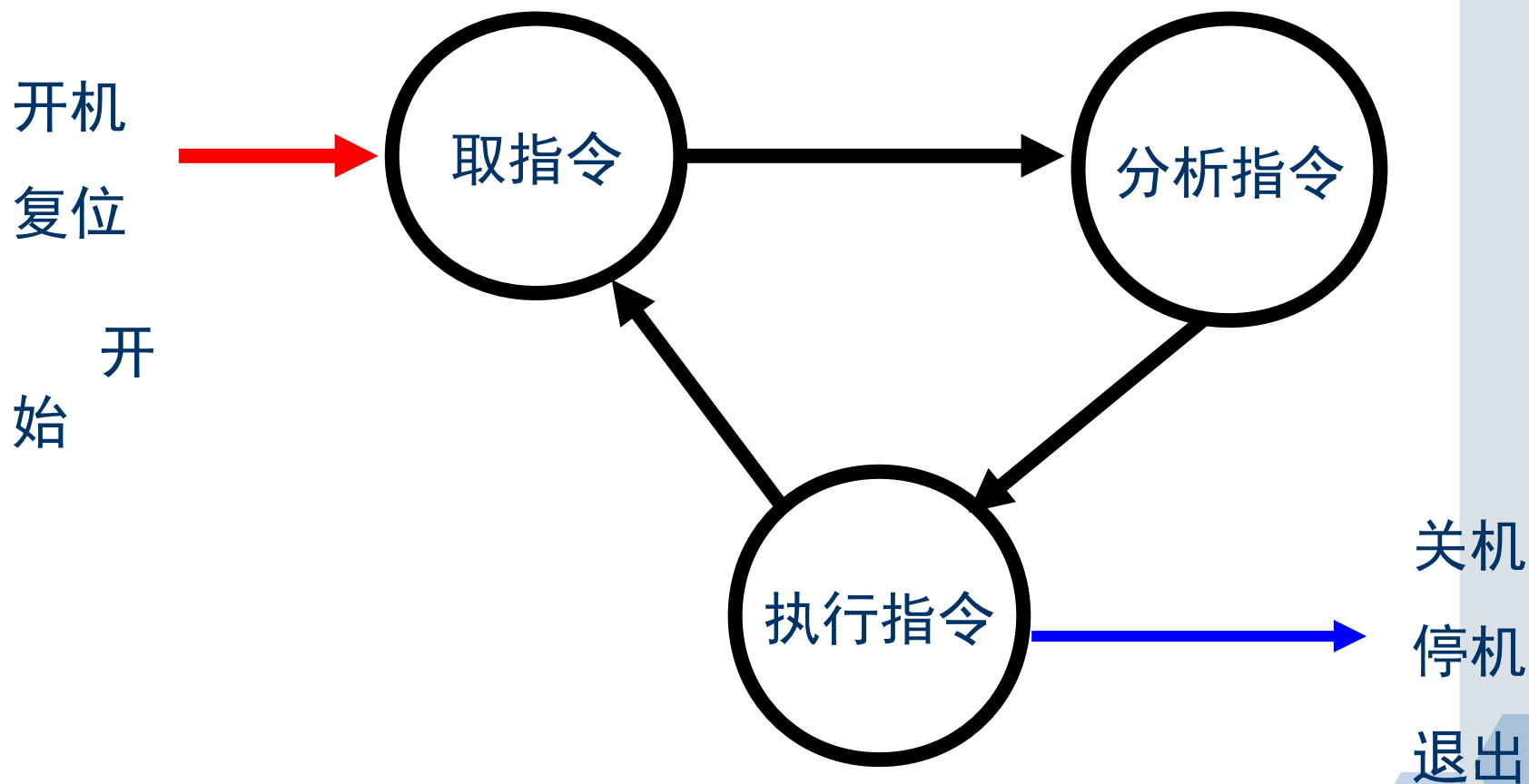
- 5 . **微指令周期（微周期）**：完成一条微指令所需的时间。
- 6 . **控制存储器**：存放微程序的存储器—微程序控制存储器，简称：控存。
- 7 . **微地址**：微指令在控存中的地址。
- 8 . **微地址寄存器**：存放微指令的寄存器： $\mu\text{AR}$ 。
- 9 . **微指令寄存器**：存放微指令地址的寄存器： $\mu\text{IR}$ 。





# 一、微程序控制的基本概念和工作原理

## 程序、指令的执行过程：

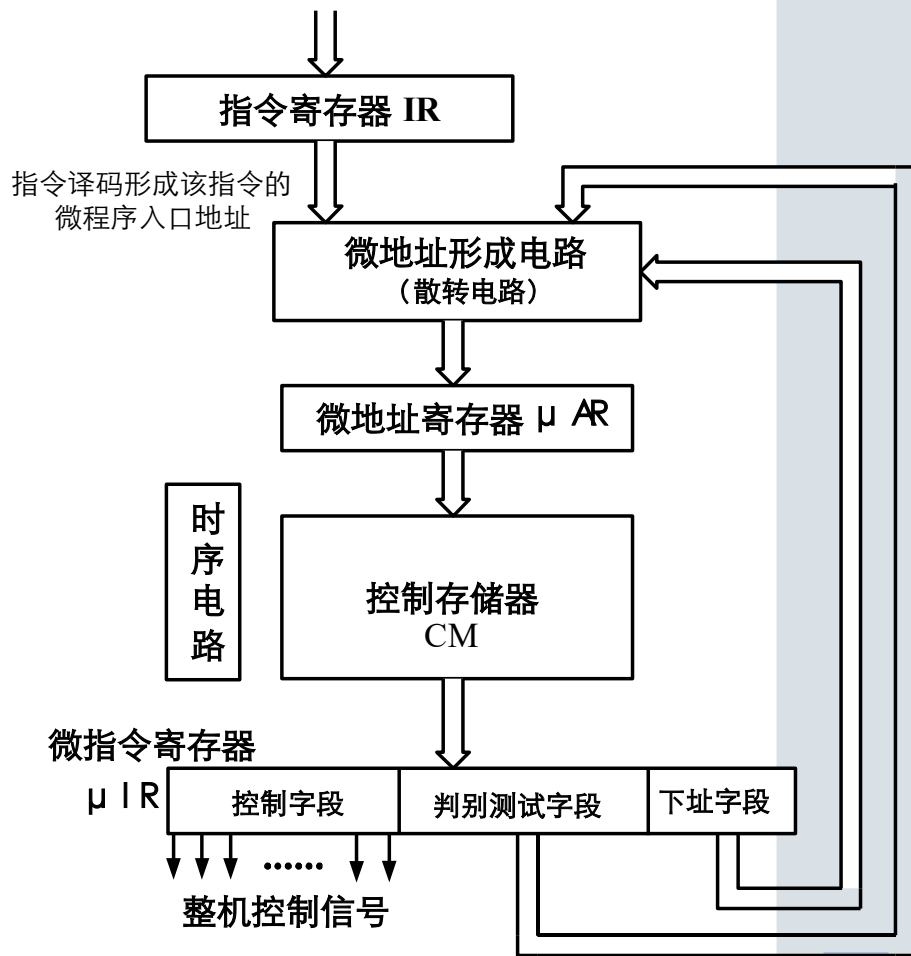






# 一、微程序控制的基本概念和工作原理

## 1. 微地址寄存器 $\mu AR$ 、控制存储器（简称控存）、微指令，微地址，微命令，微操作，微周期，微程序

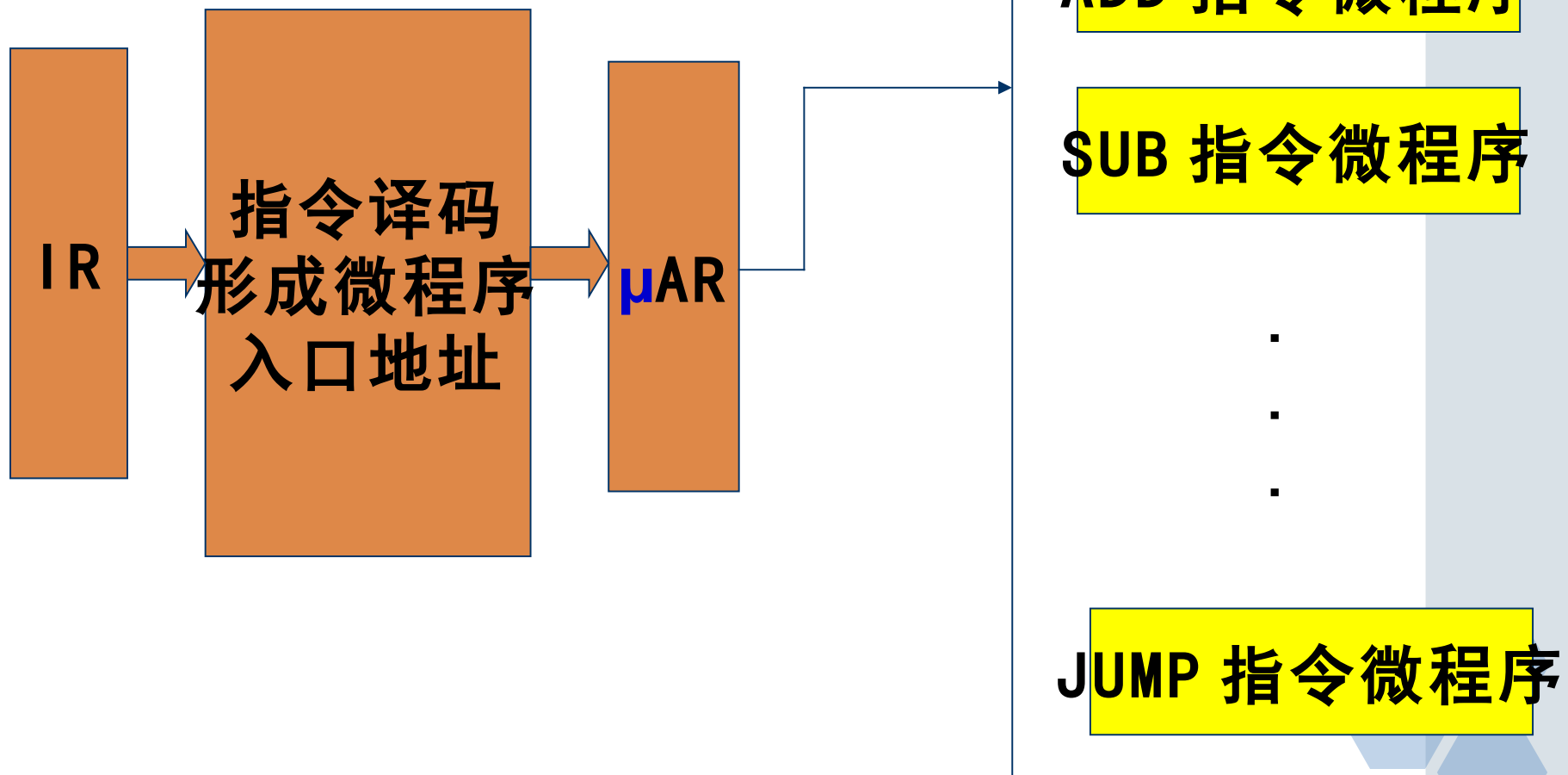




# 一、微程序控制的基本概念和工作原理

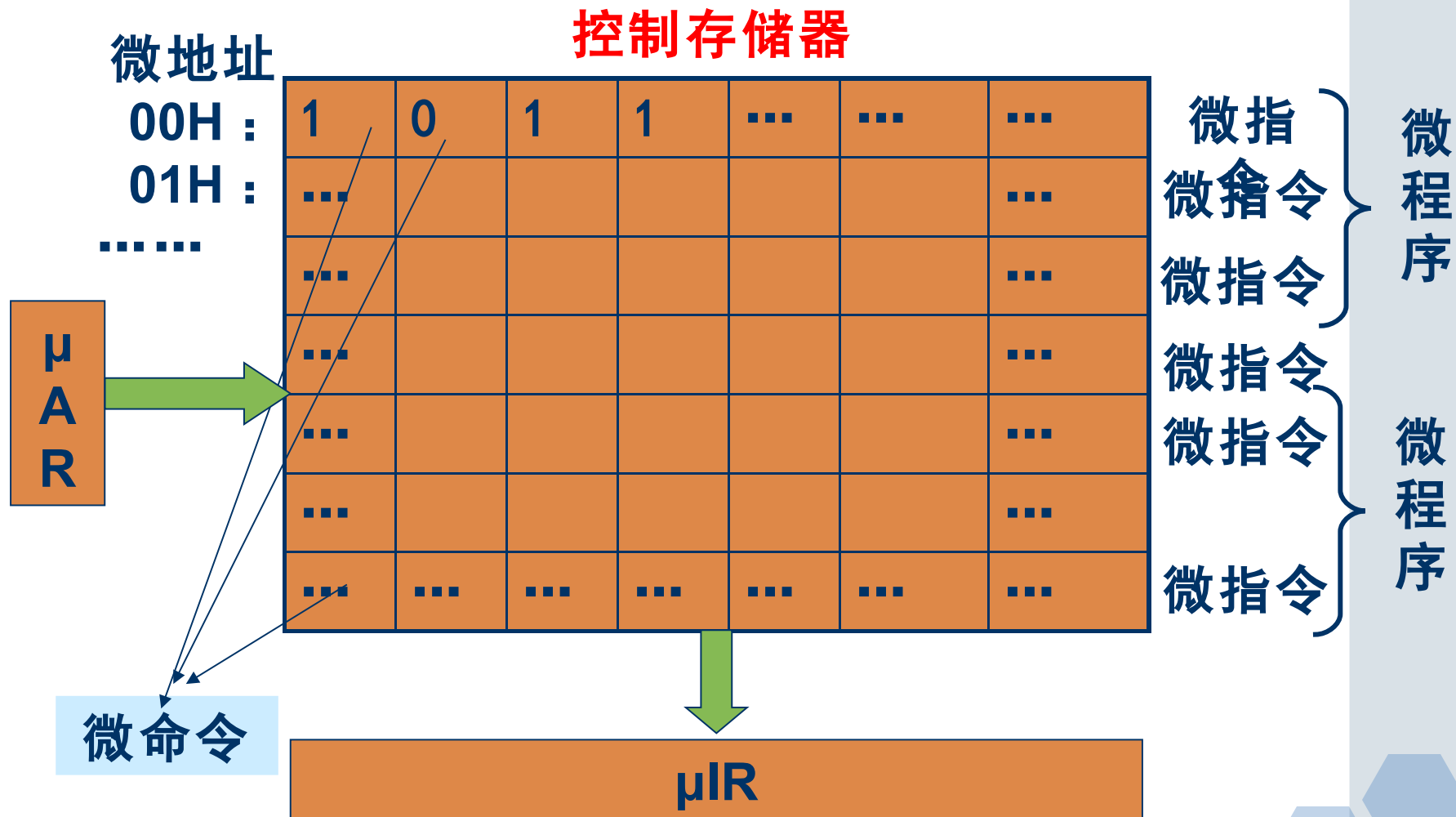
控存

IR=SUB





# 微程序基本概念关系





# 一、微程序控制的基本概念和工作原理

## ❖ 微程序控制器的工作原理：

一条机器指令由一段微程序来解释实现。

## ❖ 控制存储器中包含：

- 取指令的微程序段：公操作（所有指令共用）
- 各条指令的微程序段

## 控制存储器

取指令微程序段

MOV 指令的微程序段

ADD 指令的微程序段

SUB 指令的微程序段

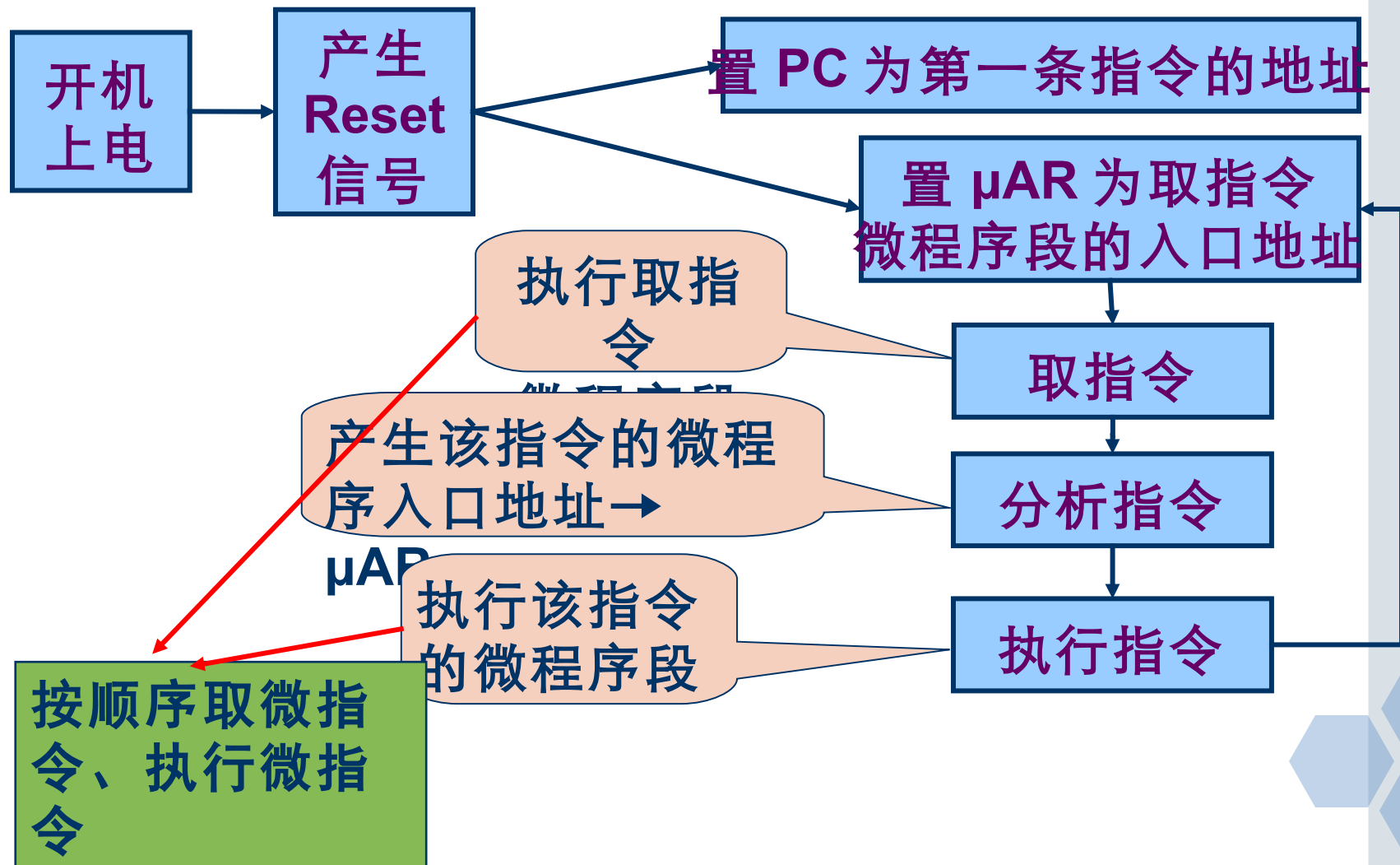
.....

HALT 指令的微程序段



# 一、微程序控制的基本概念和工作原理

## ❖ 微程序控制的计算机工作过程：



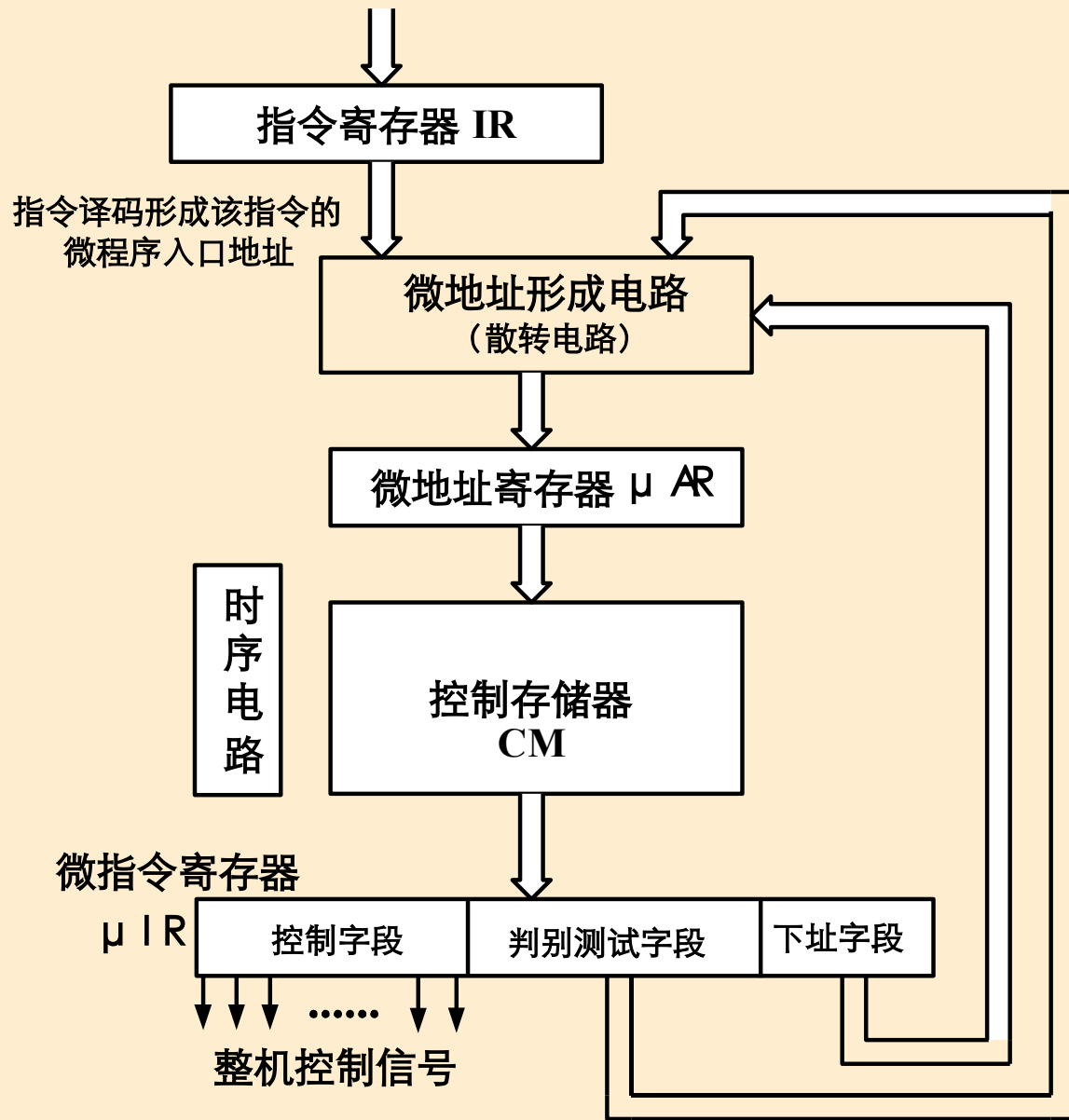


# 一、微程序控制的基本概念和工作

## 微程序控制器的组成

操作控制  
信号形成部  
件主要由以  
下 3 个部件  
构成：

- ① 控制存储器 CM
- ② 微地址寄存器  $\mu AR$
- ③ 微指令寄存器  $\mu IR$





# 一、微程序控制的基本概念和工作原

## ❖ 微指令的一般结构

- **控制字段：**包含了一组微命令信号，用于控制完成本条微指令的操作。
- **下址字段：**用于指出后继微地址（下条微指令地址）的相关信息。

控制字段	下址字段
------	------

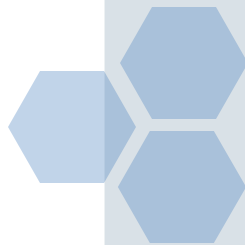




# 下址的生成方法

## ❖ 常用的下址生成方法有以下四种：

- **第一种**：下址就是控制存储器中的下一个地址，在这种情况下，控制器通常的实现方法是把当前地址加 1 来作为下址；
- **第二种**：下址是由当前微指令提供的一个绝对控存地址，这个绝对地址可能是后继微地址的全部，也可能是后继微地址的一部分，在微程序发生转移的情况下，这种方法是很有有效的；







# 下址的生成方法

## ❖ 常用的下址生成方法有以下四种：

- **第三种**：根据机器指令操作码产生该指令对应的微程序入口地址（指令译码），通常这个工作由映象逻辑（mapping logic）来完成，指令操作码输入映象逻辑后，硬件将操作码映射成该指令对应执行指令周期中的第一条微指令的地址，即微程序入口地址，将微程序入口地址装入微地址寄存器就可以转入到正确的微程序。显然，对于整条指令来说，这种映象逻辑只用一次。





## 下址的生成方法

- **第四种**：当调用微子程序时，其返回地址存储在微子程序寄存器（返回寄存器）或硬件堆栈里，微子程序寄存器或硬件堆栈都可以形成下址。

