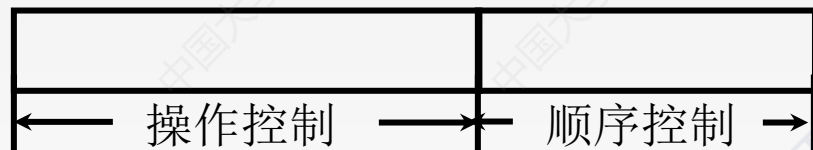


本节内容

微指令的设计

微程序控制器的工作原理

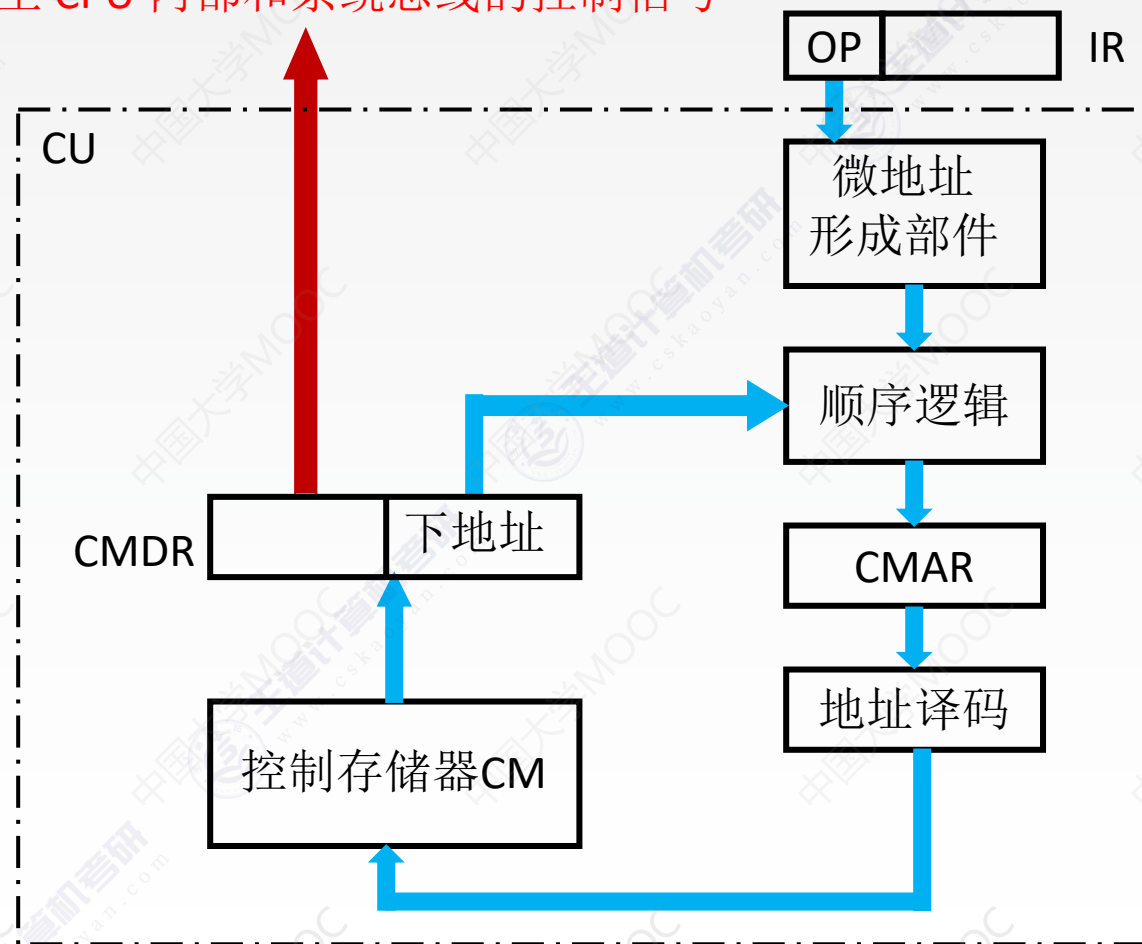
微指令基本格式



微指令的具体格式应该怎么设计？
如何根据微指令发出相应的微命令？

微命令与微操作一一对应，一个微命令对应一根输出线
有的微命令可以并行执行，因此一条微指令可以包含多个微命令

至 CPU 内部和系统总线的控制信号



微指令的格式

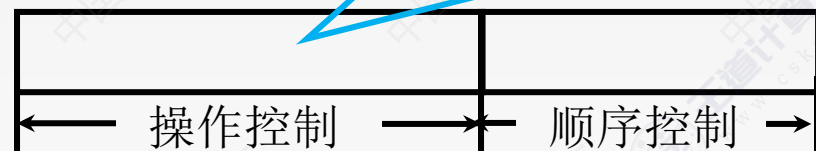
相容性微命令：可以并行完成的微命令。

互斥性微命令：不允许并行完成的微命令。

1. 水平型微指令 一条微指令能定义多个可并行的微命令。

基本格式

如何表示一系列控制信号？



优点：微程序短，执行速度快；

缺点：微指令长，编写微程序较麻烦。

微指令1
微指令2
微指令3

水平型（胖胖的）

2. 垂直型微指令 一条微指令只能定义一个微命令，由微操作码字段规定具体功能

基本格式



优点：微指令短、简单、规整，便于编写微程序；

缺点：微程序长，执行速度慢，工作效率低。

微指令0
微指令1
微指令2
微指令3
微指令4
微指令5
微指令6

垂直型
（瘦瘦的）

3. 混合型微指令 在垂直型的基础上增加一些不太复杂的并行操作。

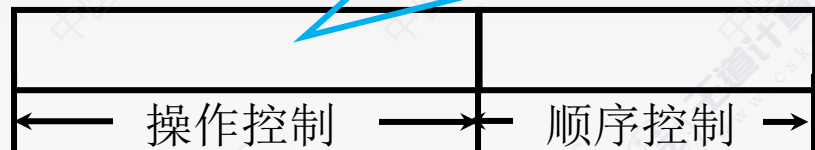
微指令较短，仍便于编写；微程序也不长，执行速度加快。

微指令的编码方式

1. 水平型微指令 一条微指令能定义多个可并行的微命令。

基本格式

如何表示一系列控制信号？



优点：微程序短，执行速度快；

缺点：微指令长，编写微程序较麻烦。

微指令的编码方式又称为微指令的控制方式，它是指如何对微指令的控制字段进行编码，以形成控制信号。编码的目标是在保证速度的情况下，尽量缩短微指令字长。

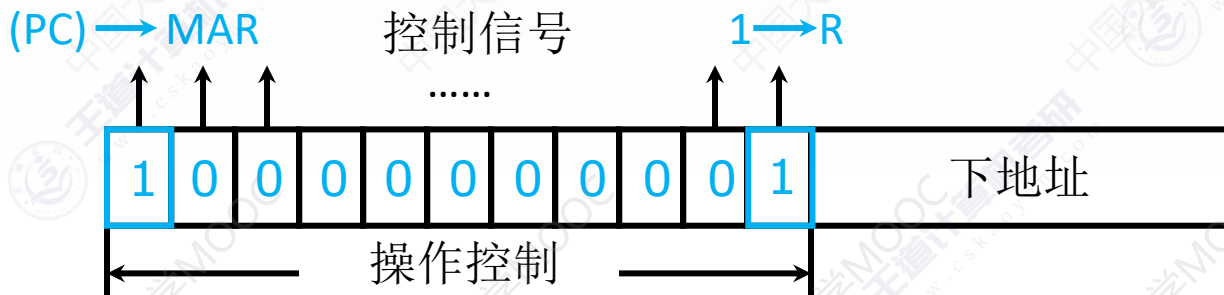
(1) 直接编码（直接控制）方式

在微指令的操作控制字段中，**每一位代表一个微操作命令**

某位为“1”表示该控制信号有效

优点：简单、直观，执行速度快，操作并行性好。

缺点：微指令字长过长， n 个微命令就要求微指令的操作字段有 n 位，造成控存容量极大。



微指令的编码方式

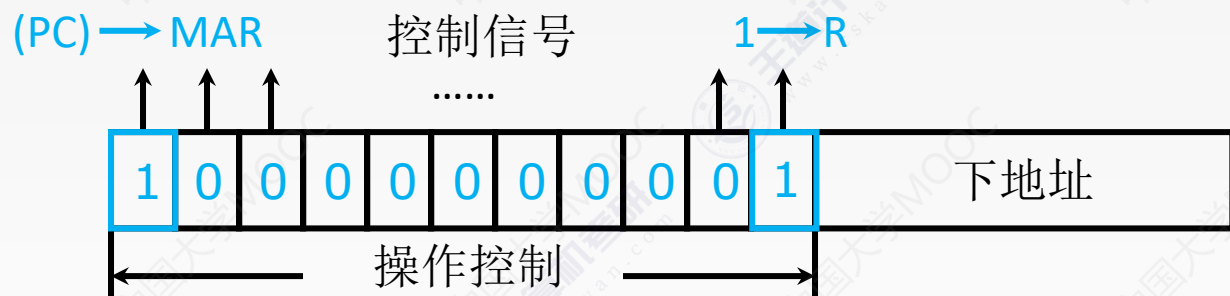
(1) 直接编码（直接控制）方式

在微指令的操作控制字段中，**每一位代表一个微操作命令**

某位为“1”表示该控制信号有效

优点：简单、直观，执行速度快，操作并行性好。

缺点：微指令字长过长， n 个微命令就要求微指令的操作字段有 n 位，造成控存容量极大。

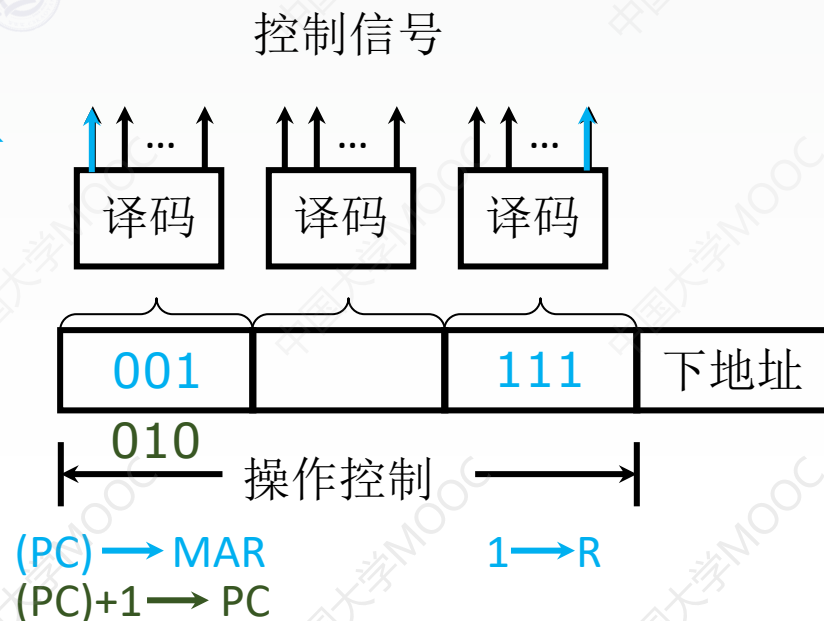


(2) 字段直接编码方式

将微指令的控制字段分成若干“段”，**每段经译码后发出控制信号**

微命令字段分段的原则：

- ① **互斥性**微命令分在**同一段内**，**相容性**微命令分在**不同段内**。
- ② **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。
- ③ 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。因此，当某字段的长度为3位时，最多只能表示7个互斥的微命令，**通常用000表示不操作**。

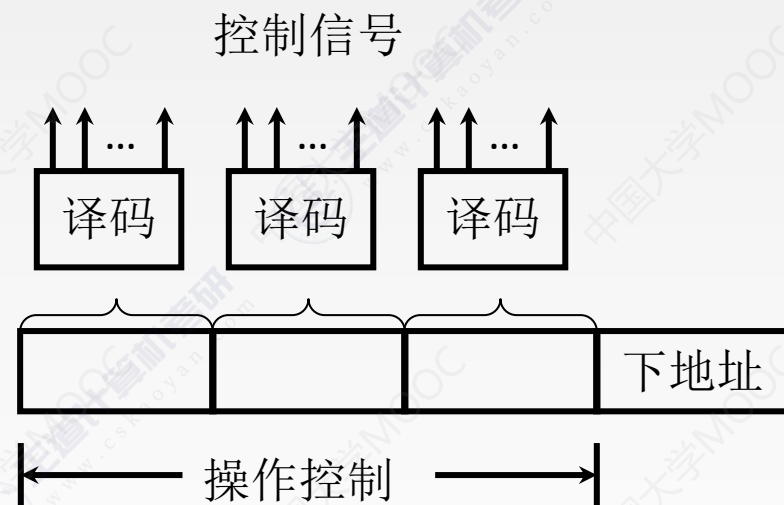


例题：字段直接编码方式

(2) 字段直接编码方式

将微指令的控制字段分成若干“段”，每段经译码后发出控制信号
微命令字段分段的原则：

- ① 互斥性微命令分在同一段内，相容性微命令分在不同段内。
- ② 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。
- ③ 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。因此，当某字段的长度为3位时，最多只能表示7个互斥的微命令，通常用000表示不操作。



某计算机的控制器采用微程序控制方式，微指令中的操作控制字段采用**字段直接编码法**，共有33个微命令，构成5个互斥类，分别包含7、3、12、5和6个微命令，则操作控制字段至少有多少位？

第1个互斥类有7个微命令，要留出1个状态表示不操作，所以需要表示8种不同的状态，故需要3个二进制位。

以此类推，后面4个互斥类各需要表示4、13、6、7种不同的状态，分别对应2、4、3、3个二进制位。

故操作控制字段的总位数为

$$3+2+4+3+3 = 15 \text{ 位}$$

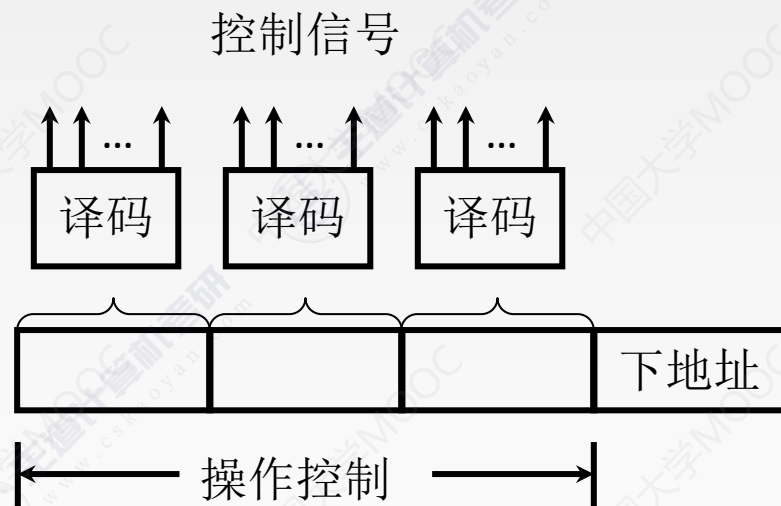
Tips: 若采用**直接编码**方式，则控制字段需要33位

微指令的编码方式

(2) 字段直接编码方式

将微指令的控制字段分成若干“段”，每段经译码后发出控制信号
微命令字段分段的原则：

- ① 互斥性微命令分在同一段内，相容性微命令分在不同段内。
- ② 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。
- ③ 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。因此，当某字段的长度为3位时，最多只能表示7个互斥的微命令，通常用000表示不操作。

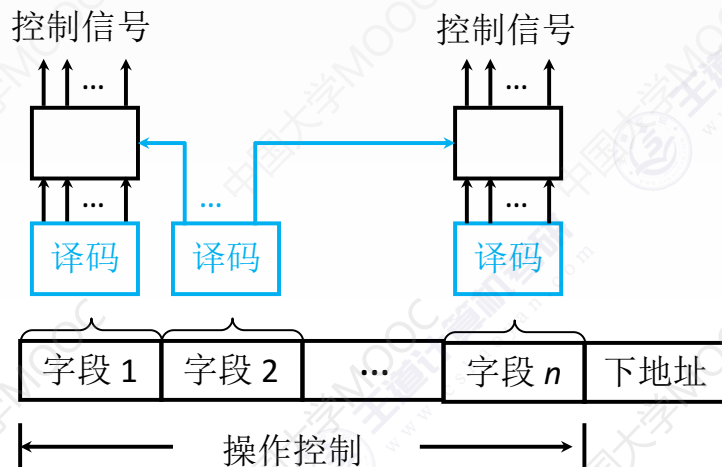


优点：可以缩短微指令字长。

缺点：要通过译码电路后再发出微命令，因此比直接编码方式慢。

(3) 字段间接编码方式

一个字段的某些微命令需由另一个字段中的某些微命令来解释，由于不是靠字段直接译码发出的微命令，故称为字段间接编码，又称隐式编码。

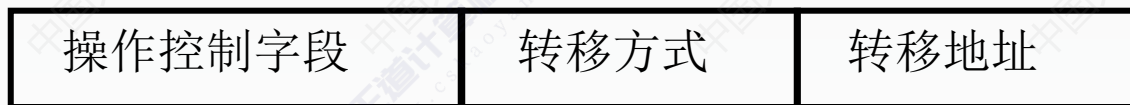


优点：可进一步缩短微指令字长。

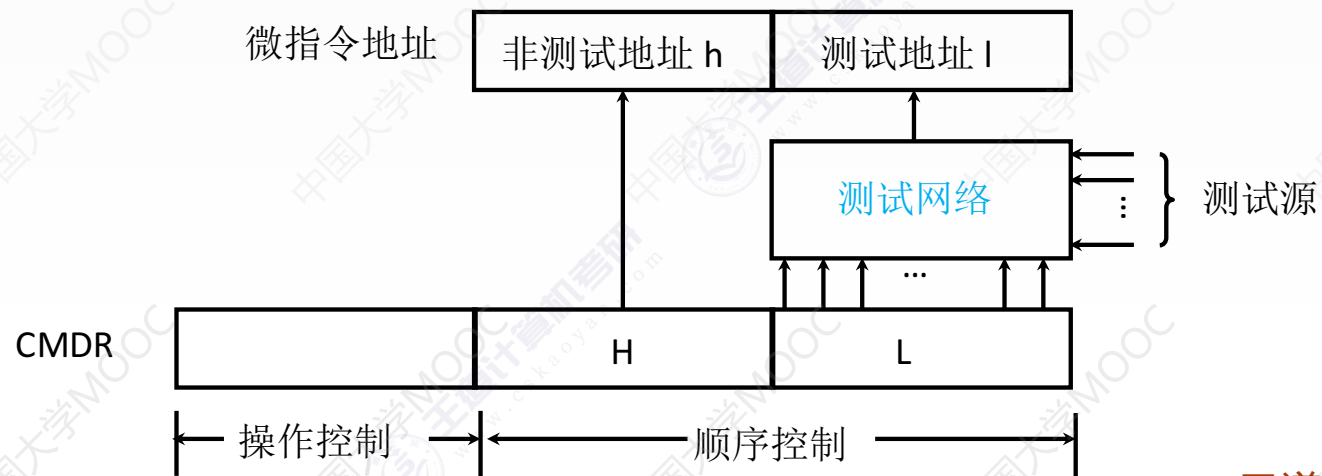
缺点：削弱了微指令的并行控制能力，故通常作为字段直接编码方式的一种辅助手段。

微指令的地址形成方式

1. 微指令的 **下地址字段** 指出 微指令格式中设置一个下地址字段，由微指令的下地址字段直接指出后继微指令的地址，这种方式又称为**断定方式**。
2. 根据机器指令的 **操作码** 形成 当机器指令取至指令寄存器后，微指令的地址由操作码经微地址形成部件形成。
3. 增量**计数器法** $(CMAR) + 1 \longrightarrow CMAR$
4. 分支转移 转移方式：指明判别条件；转移地址：指明转移成功后的去向。



5. 通过测试网络



微指令的地址形成方式

1. 微指令的 **下地址字段** 指出 微指令格式中设置一个下地址字段，由微指令的下地址字段直接指出后继微指令的地址，这种方式又称为**断定方式**。
2. 根据机器指令的 **操作码** 形成 当机器指令取至指令寄存器后，微指令的地址由操作码经微地址形成部件形成。
3. 增量**计数器法** $(CMAR) + 1 \longrightarrow CMAR$
4. 分支转移 转移方式：指明判别条件；转移地址：指明转移成功后的去向。

操作控制字段	转移方式	转移地址
--------	------	------

5. 通过测试网络
6. 由硬件产生微程序入口地址

第一条微指令地址 由专门 **硬件** 产生（用专门的硬件记录取指周期微程序首地址）

中断周期 由 **硬件** 产生 **中断周期微程序首地址**（用专门的硬件记录）

例题-断定方式

1. 微指令的 **下地址字段** 指出 微指令格式中设置一个下地址字段，由微指令的下地址字段直接指出后继微指令的地址，这种方式又称为**断定方式**。

某计算机采用微程序控制器，共有32条指令，公共的取指令微程序包含2条微指令，各指令对应的微程序平均由4条微指令组成，采用**断定法**（下地址字段法）确定下条微指令地址，则微指令中下地址字段的位数至少是多少位？

例题-断定方式

微指令
存放地
址

微指令操作码	下地址	
	1	取指周期 微程序
	2	
	3	
⋮		可以没有
转执行周期		
⋮		
转取指周期		间址周期 微程序
		中断周期 微程序
	14	对应 LDA 指令的 执行周期微程序
	15	
	0	
	17	对应 STA 指令的 执行周期微程序
	18	
	0	
⋮		

某计算机采用微程序控制器，共有32条指令，公共的取指令微程序包含2条微指令，各指令对应的微程序平均由4条微指令组成，采用**断定法**（下地址字段法）确定下条微指令地址，则微指令中下地址字段的位数至少是多少位？

总共需要存储多少条微指令？

$$32 \times 4 + 2 = 130 \text{ 条}$$

标注出130个不同的位置至少需要多少个二进制位？

$$2^7 = 128, 2^8 = 256$$

下地址字段的位数至少是8位

知识回顾

微指令的设计

微指令格式

水平型微指令



每条微指令能定义多个可并行的微命令

垂直型微指令



每条微指令只能定义一个微命令，由微操作码指明

混合型微指令



在垂直型微指令的基础上加上一些简单的并行操作

水平型微指令的编码方式

直接编码（直接控制）



控制码的每个 bit 对应一个微命令，微指令执行速度最快

字段直接编码



将互斥性的微命令分在同一个段内，相容的分在不同的段

每个段留出一个状态表示“不操作”

微指令操作码需要经过译码电路处理，因此执行速度更慢

字段间接编码



一个字段的微命令需要用另一个字段的微命令解释

可能需要多级译码电路处理，执行速度最慢

下一条微指令地址的形成方式

断定法（下地址法）：根据当前执行的微指令下地址找到下一条微指令

计数器法： $\mu PC + 1$ 顺序找到下一条微指令

根据指令操作码确定执行周期微程序首地址

由专门的硬件指明取指/中断周期的微程序首地址



公众号：王道在线



b站：王道计算机教育



抖音：王道计算机考研