

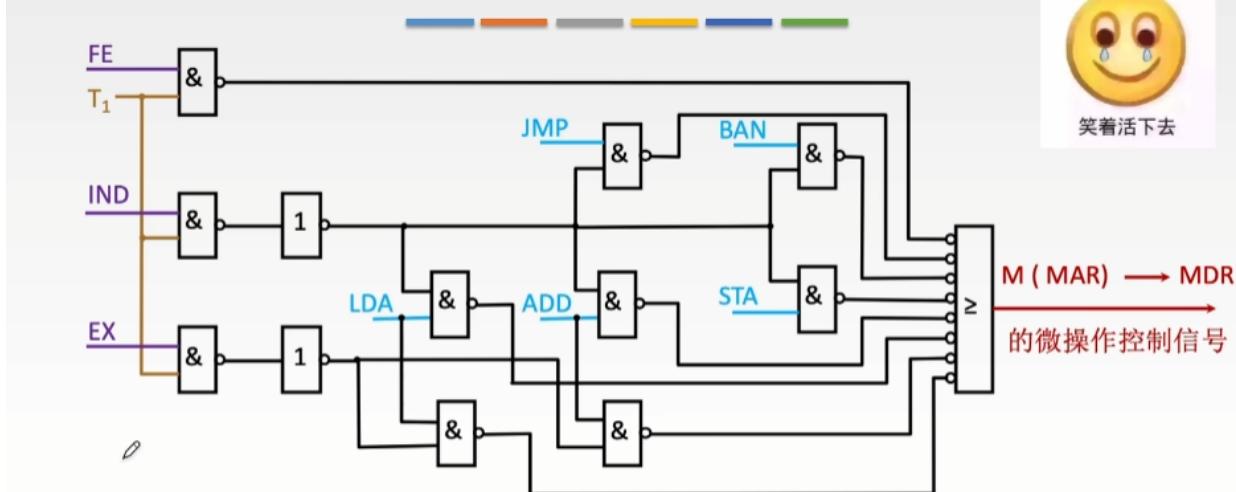
微程序控制器



硬布线控制器的设计思路

硬布线控制器：微操作控制信号由组合逻辑电路根据当前的指令码、状态和时序，即时产生

硬布线控制器的设计思路



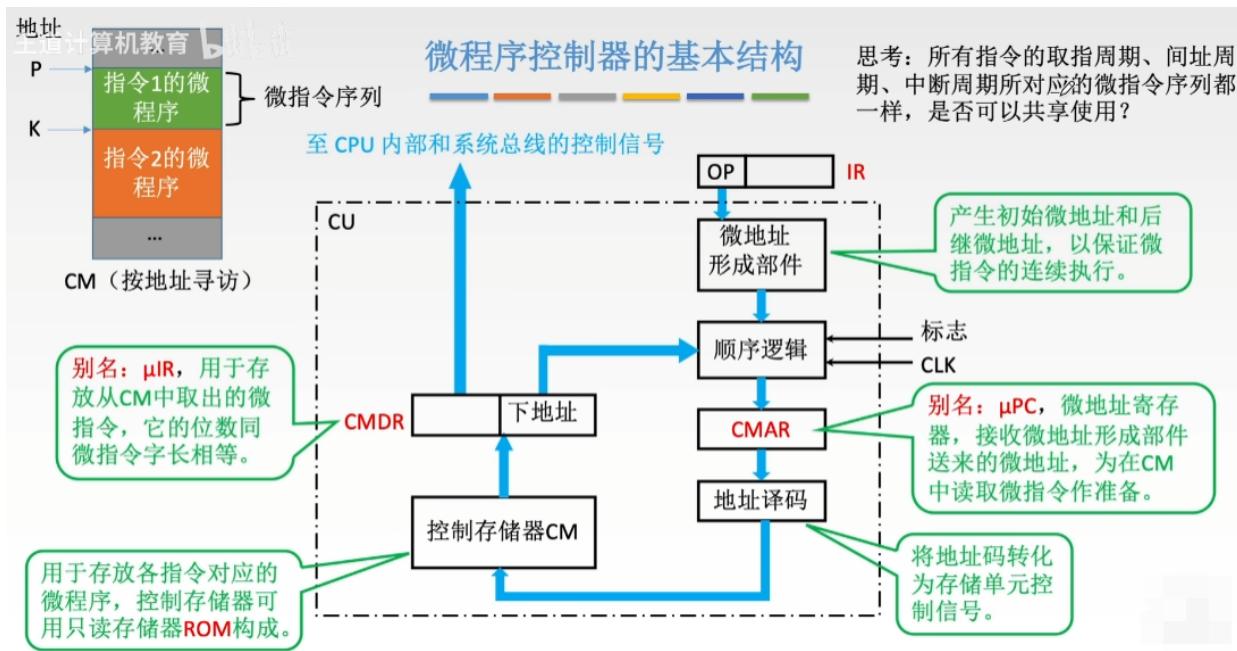
硬布线控制器：微操作控制信号由组合逻辑电路根据当前的指令码、状态和时序，即时产生

时序信息包含机器周期、节拍

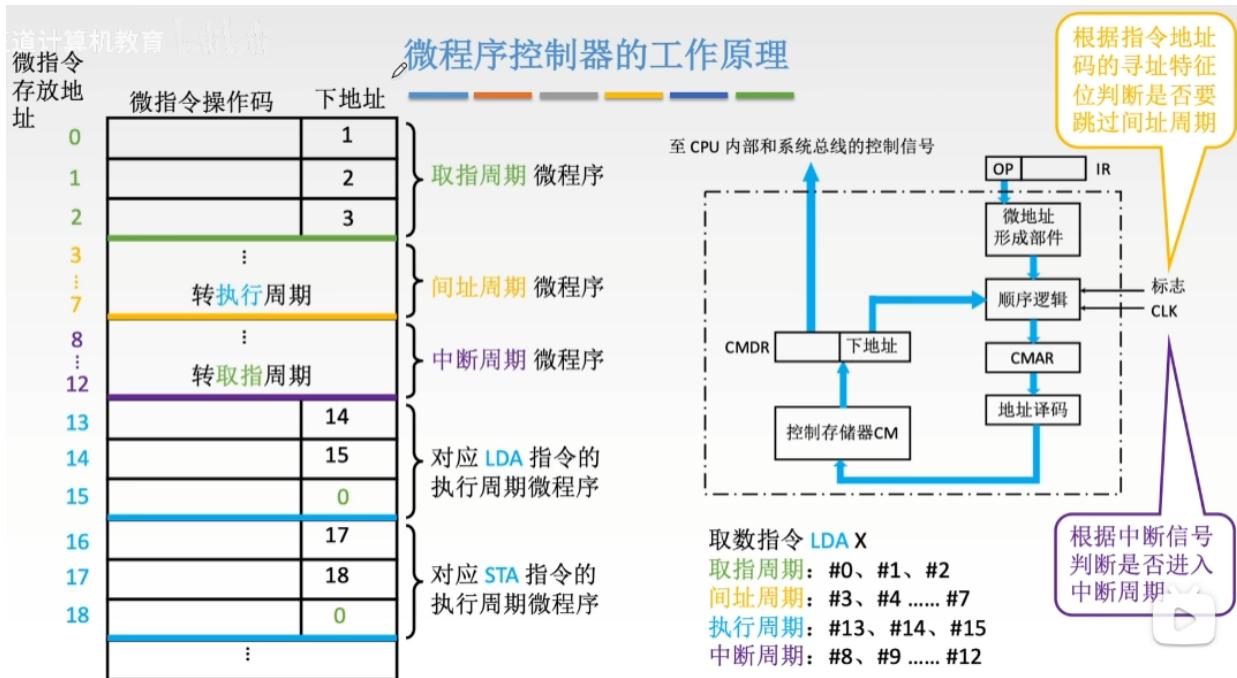
微程序控制器的设计思路

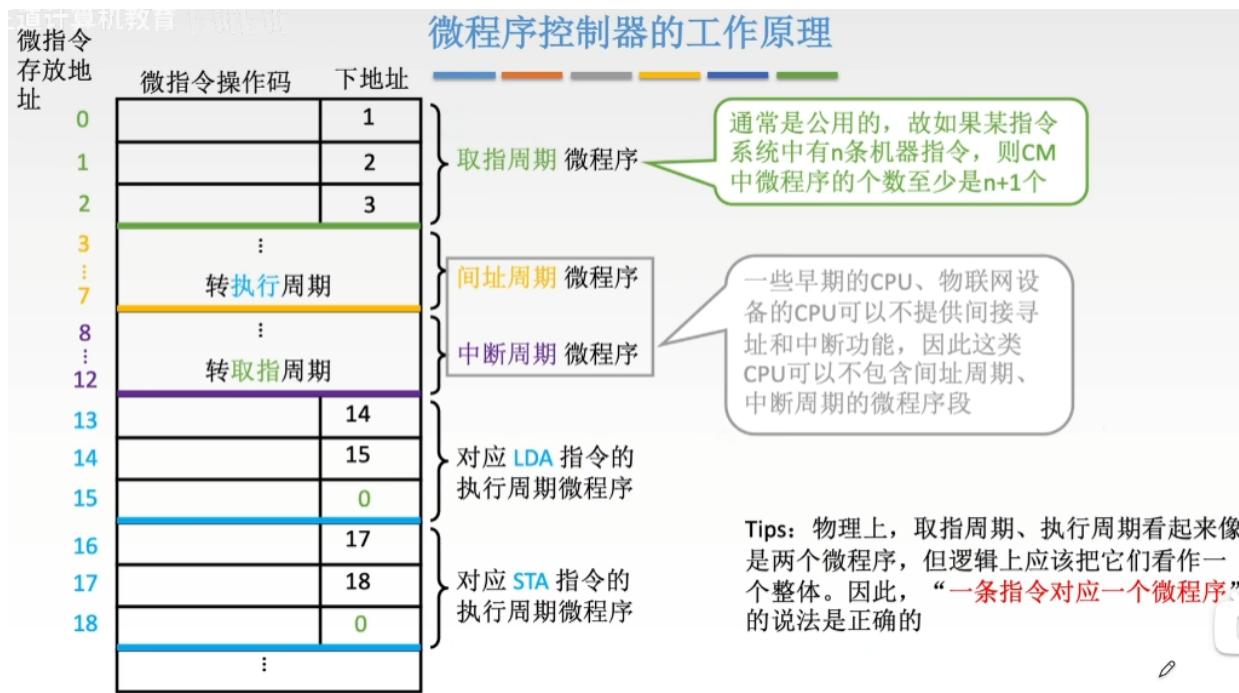


微程序控制器的基本结构

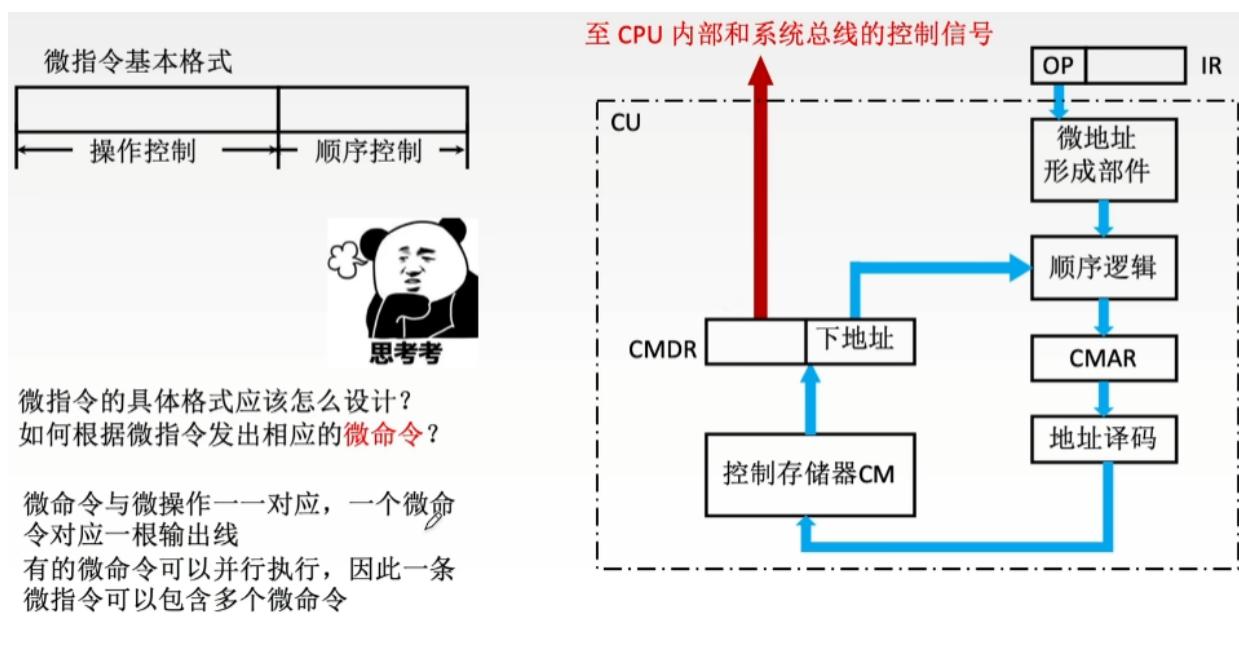


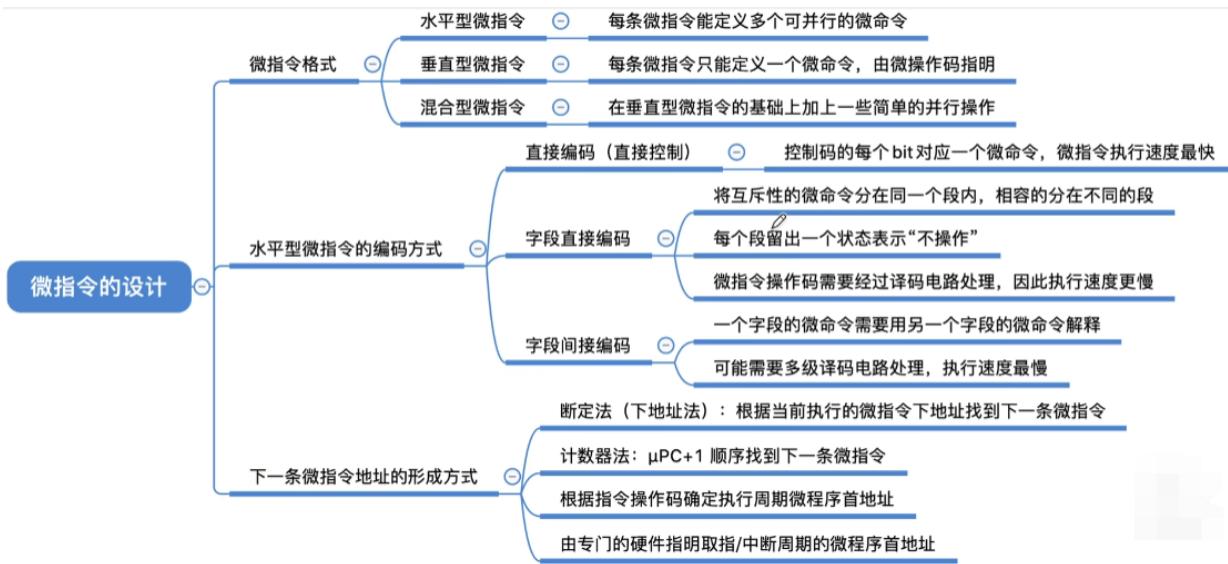
微程序控制器的工作原理





微指令的设计





微指令的格式

相容性微命令：可以并行完成的微命令。

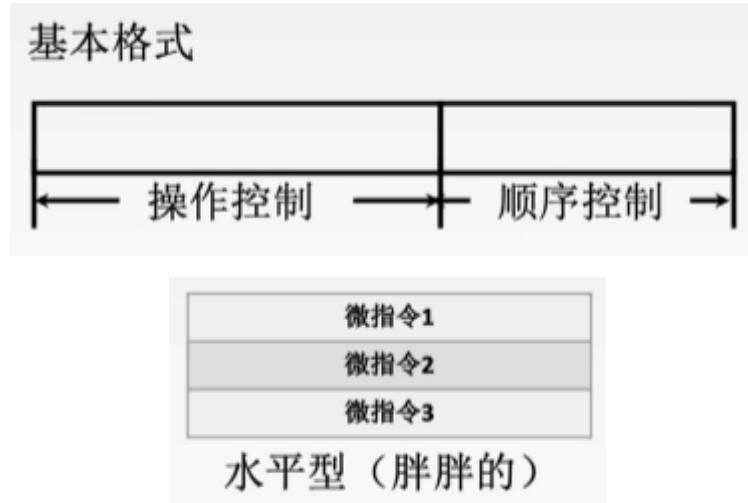
互斥性微命令：不允许并行完成的微命令。

水平型微指令

一条微指令能定义多个可并行的微命令。

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。



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

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。

基本格式



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。

计算机教育网

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？

操作控制	顺序控制
------	------

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。

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

μOP	Rd	Rs
← 微操作码 →	← 目的地址 →	← 源地址 →

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。

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。

垂直型（瘦瘦的）

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。

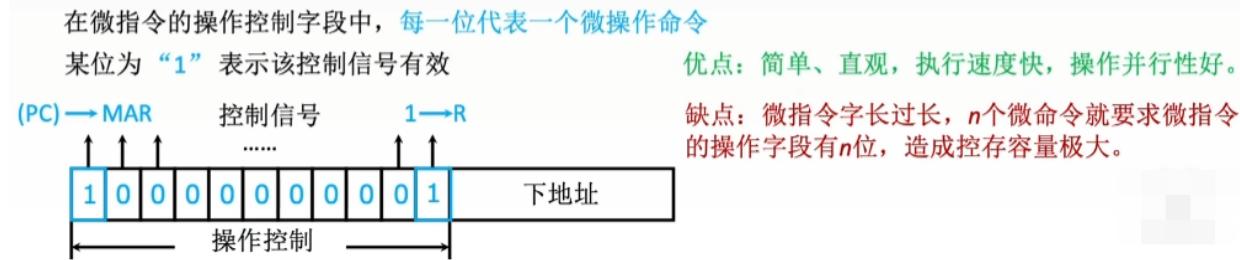
基本格式 如何表示一系列控制信号？

操作控制	顺序控制
------	------

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。

直接编码（直接控制）方式



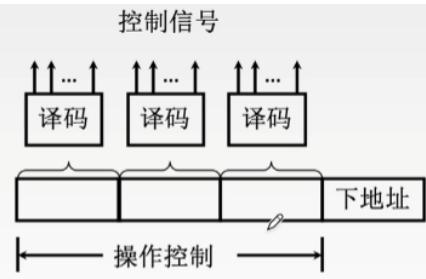
字段直接编码方式

(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。



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。

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。

故操作控制字段的总位数为

$$3+2+4+3+3 = 15 \text{ 位}$$

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。

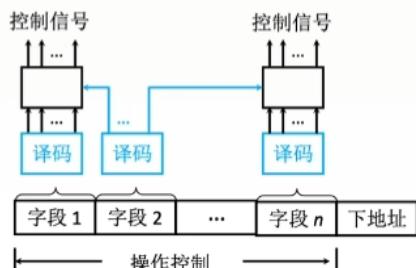
Tips：若采用直接编码方式，则控制字段需要33位



字段间接编码方式

(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。

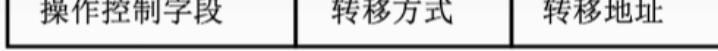


优点：可进一步缩短微指令字长。

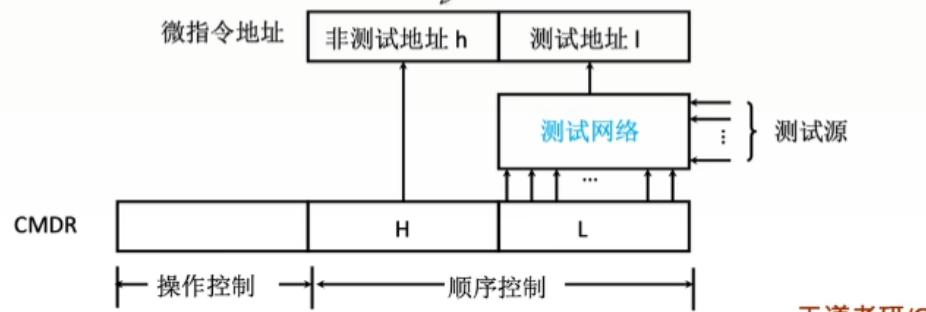
缺点：削弱了微指令的并行控制能力，故通常作为字段直接编码方式的一种辅助手段。

微指令的地址形成方式

- 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。
- 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。
- 增量计数器法 $(CMAR) + 1 \rightarrow CMAR$
- 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。



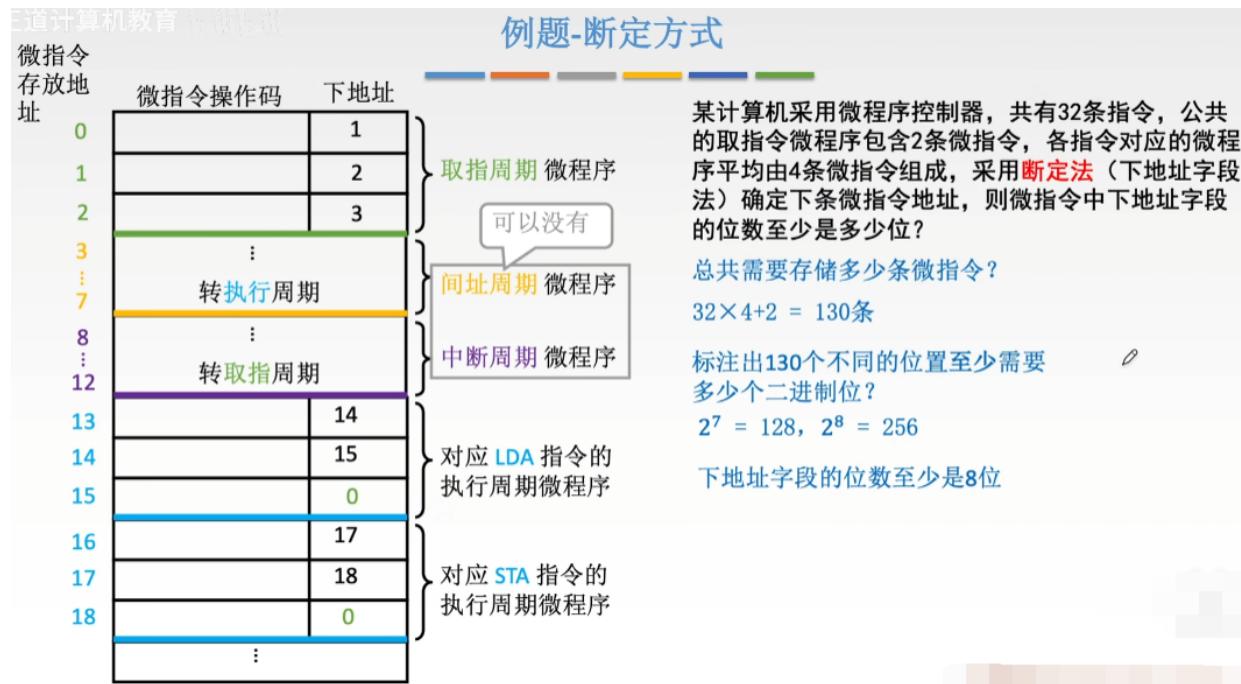
5. 通过测试网络



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）



微程序控制单元的设计

设计步骤:

1. 分析每个阶段的微操作序列
2. 写出对应机器指令的微操作命令及节拍安排
3. 确定微指令格式
4. 编写微指令码点

取指周期-硬布线控制器的节拍安排

- $T_0 \quad PC \rightarrow MAR$
- $T_0 \quad 1 \rightarrow R$
- $T_1 \quad M(MAR) \rightarrow MDR$
- $T_1 \quad (PC) + 1 \rightarrow PC$
- $T_2 \quad MDR \rightarrow IR$
- $T_2 \quad OP(IR) \rightarrow ID$

取指周期-微程序控制器的节拍安排

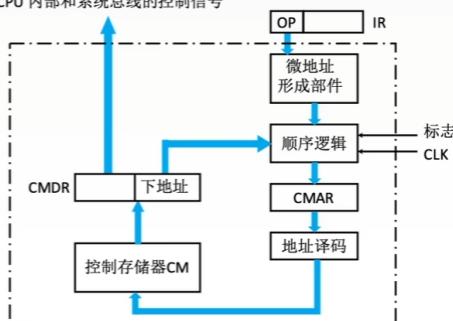
- | |
|---|
| $T_0 \quad PC \rightarrow MAR$ |
| $T_0 \quad 1 \rightarrow R$ |
| $T_1 \quad M(MAR) \rightarrow MDR$ |
| $T_1 \quad (PC) + 1 \rightarrow PC$ |
| $T_2 \quad MDR \rightarrow IR$ |
| $T_2 \quad OP(IR) \rightarrow \text{微地址形成部件}$ |

3条微指令

取指周期-硬布线控制器的节拍安排

- $T_0 \quad PC \rightarrow MAR$
- $T_0 \quad 1 \rightarrow R$
- $T_1 \quad M(MAR) \rightarrow MDR$
- $T_1 \quad (PC) + 1 \rightarrow PC$
- $T_2 \quad MDR \rightarrow IR$
- $T_2 \quad OP(IR) \rightarrow ID$

至 CPU 内部和系统总线的控制信号



取指周期-微程序控制器的节拍安排

- | | |
|---|-------------------------|
| $T_0 \quad PC \rightarrow MAR$ | 微指令a |
| $T_0 \quad 1 \rightarrow R$ | |
| $T_1 \quad Ad(CMDR) \rightarrow CMAR$ | 需要用 T_1 节拍确定下一条微指令的地址 |
| $T_2 \quad M(MAR) \rightarrow MDR$ | |
| $T_2 \quad (PC) + 1 \rightarrow PC$ | 微指令b |
| $T_3 \quad Ad(CMDR) \rightarrow CMAR$ | |
| $T_4 \quad MDR \rightarrow IR$ | 需要用 T_3 节拍确定下一条微指令的地址 |
| $T_4 \quad OP(IR) \rightarrow \text{微地址形成部件}$ | |
| $T_5 \quad \text{微地址形成部件} \rightarrow CMAR$ | 微指令c |

根据指令操作码确定其执行周期微指令序列的首地址

显然，微程序控制器的速度比硬布线控制器更慢



设计步骤：

1. 分析每个阶段的微操作序列
2. 写出对应机器指令的微操作命令及节拍安排
 - (1) 写出每个周期所需要的微操作(参照硬布线)
 - (2) 补充微程序控制器特有的微操作：
 - a. 取指周期：
Ad (CMDR) → CMAR
OP (IR) → CMAR
 - b. 执行周期：
Ad(CMDR) → CMAR

3. 确定微指令格式

根据微操作个数决定采用何种编码方式，以确定微指令的操作控制字段的位数。

根据CM中存储的微指令总数，确定微指令的顺序控制字段的位数。

最后按操作控制字段位数和顺序控制字段位数就可确定微指令字长。

4. 编写微指令码点

根据操作控制字段每一位代表的微操作命令，编写每一条微指令的码点。

微程序设计分类

静态微程序设计和动态微程序设计

静态 微程序无需改变，采用 ROM

动态 通过 改变微指令 和 微程序 改变机器指令

有利于仿真，采用 EPROM

毫微程序设计

毫微程序设计的基本概念

微程序设计 用 微程序解释机器指令

毫微程序设计 用 毫微程序解释微程序

毫微指令与微指令 的关系好比 微指令与机器指令 的关系

硬布线与微程序的比较

类 别 对比项目	微程序控制器	硬布线控制器
工作原理	微操作控制信号以微程序的形式存放在控制存储器中，执行指令时读出即可	微操作控制信号由组合逻辑电路根据当前的指令码、状态和时序，即时产生
执行速度	慢	快
规整性	较规整	烦琐、不规整
应用场合	CISC CPU	RISC CPU
易扩充性	易扩充修改	困难