第七讲

微程序设计技术（三）

微指令后续地址产生方法

实验模型机微指令字段编码表

字段间接编译法

编码 + 译码	BTO	OTB	FUNC	
			FS=1	FS=0
000	空	空	PC+1	空
001	B-DA1(t4)	ALU-B# (t2)	J 1# (t2)	M-W# (t3)
010	B-DA2(t4)	299-B# (t2)	J 2# (t2)	M_R# (t2)
011	B-IR(t3)	SR-B# (t2)	J 3# (t2)	I/O-W# (t3)
100	B-DR(t4)	DR-B# (t2)	J 4# (t2)	I/O_R# (t2)
101	B-SP(t4)	SI-B# (t2)*	J5# (t2)	INT_R# (t2)
110	B-AR(t3)	SP-B# (t2)*	CyCn# (t2)	INT_E# (t2)
111	B-PC# (t4)	PC-B# (t2)	CyNCn# (t2)	

字段直接编译法

2、微指令下址字段设计方法

① 微程序入口地址的产生

② 后继微地址的产生

① 微程序入口地址的产生

I. 指令译码产生相应微程序入口地址如果采用 PROM 来实现，这种 PROM 称为**映射存储器（MAPROM）**，它是将机器指令的操作码映射成该指令对应的微程序入口地址。

II. 指令译码产生相应微程序入口地址也可以采用**逻辑电路**来实现。

① 微程序入口地址的产生

❖举例：前述的包含两条指令的指令系统

a） 采用 MAPROM 方法：

1. ADD 的操作码 =0101，入口地址 =110，则在 MAPROM 的 5 号单元地址中写入 6 即可；

2. JMP 的操作码 =1000，入口地址 =010，则在 MAPROM 的 8 号单元地址中写入 2 即可；

可以扩展指令条数到 16 条

MAPROM	
0000	
0001	
0010	
0011	
0100	
0101	110
0110	
0111	
1000	010
……	…….
1111	

① 微程序入口地址的产

b）采用逻辑电路方法：

1. 输入：指令操作码 OP=$I_7I_6I_5I_4$；

2. 输出：该指令的微程序入口地址 =I_4

10

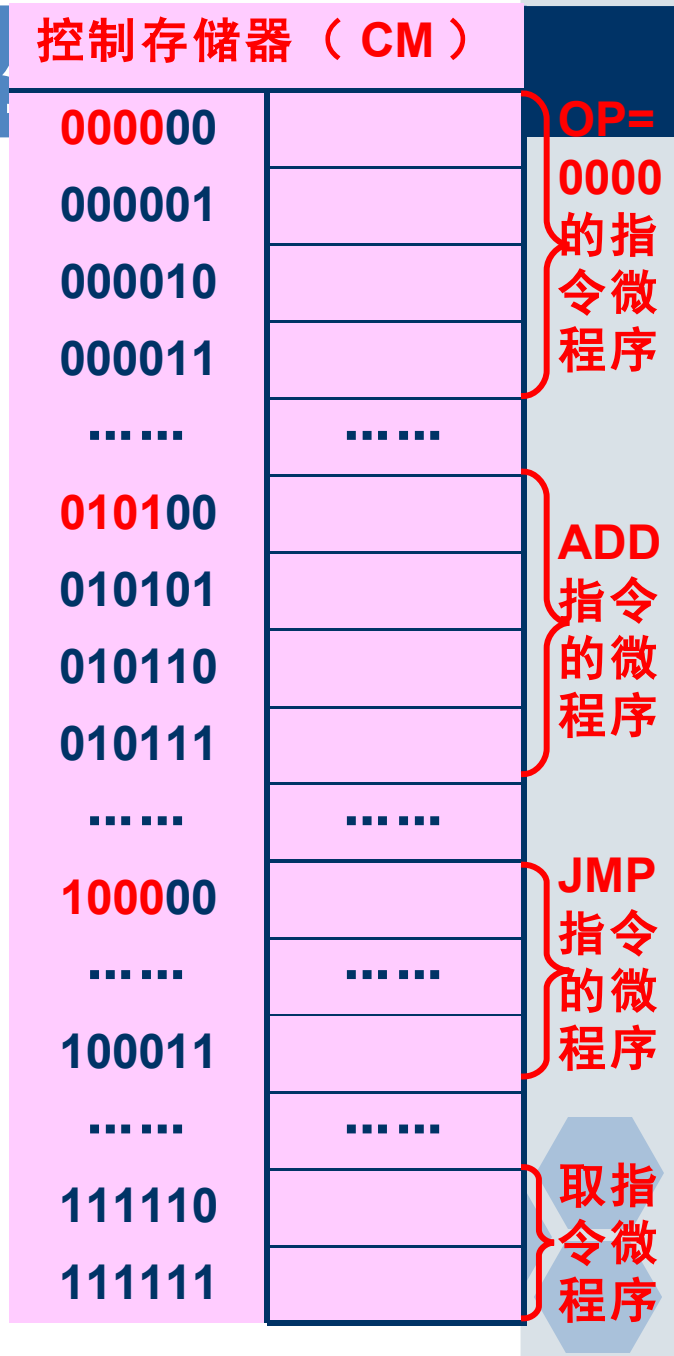
微程序入口地址产生译码电路逻辑关系：

A2=I4，

A1 ＝ 0；

A0 ＝ 0；

只能包含 2 条指令，如果需要扩充到 16 条呢？



① 微程序入口地址的产生

扩充到 16 条地址逻辑关系：　设：地址长度扩展到 6 bit

1. 输入：指令操作码 OP=$I_7I_6I_5I_4$

2. 输出：该指令的微程序入口地址 =$I_7I_6I_5I_4$ 00 ，

即：

A5 ＝ I7 ；

A4 ＝ I6 ；

A3 ＝ I5 ；

A2 ＝ I4 ；

A1 ＝ 0 ；

① 微程序入口地址的产生

- 举例：实验模型机的 7 位微程序入口地址产生方法：

a）采用逻辑电路方法：在发送 J1# 的那条微指令周期的 T4 节拍进行指令译码

设：　μAR ＝ MA6 MA5 MA4 MA3 MA2 MA1 MA0

I7 I6 ＝ 11 时，用来进行操作码扩展，译码控制信号： J1 # 。

a）输入：指令操作码 6 位　　I7 I6 I5 I4 I3 I2

b）输出：

1. 当 I7 I6≠11 时：

微程序入口地址 =MA6 MA5 MA4 I7 I6 I5 I4 ；

2. 当 I7 I6=11 时：

微程序入口地址的逻辑表达式： MA6 1 MA4 I5 I4 I3 I2 。

3. MA6~ MA0 是发送 J1# 的那条微指令的下址字段

② 后继微地址的产生

I
计数器方式

II
判定方式（下址字段法

Ⅰ. 计数器方式

a) 在微程序控制器中设置一个微程序计数器 μPC；由 μPC 来提供后继微地址

- ✓ 在顺序执行微指令时， μPC 自动 +1 。
- ✓ 遇到转移微指令时，由微指令给出转移微地址，置入 μPC 。

Ⅰ．计数器方式

- 微指令的格式有两种：

非转移类的微指令格式

0	控制字段

转移类的微指令格式

1	转移类型	下址字段（转移微地址）

- 优点：微指令字较短，便于编写微程序，后继微地址产生机构比较简单；
- 缺点：微程序较长，执行速度相对较慢

- 这种方式是将微指令的下址字段分成两部分：转移控制字段 BCF 和转移地址字段 BAF，格式如下：

控制字段	BAF	BCF

- BAF 用于给出微指令转移的部分微地址，转移控制字段 BCF 用来确定后继微地址是顺序执行还是条件转移。当微程序条件成立转移时，将 BAF 和 μPC 组合产生位地址送入 μPC （如 μPC ＋ BAF ），否则顺序执行下一条微指令（ μPC ＋ 1 ）。
- 由 BCF 所定义的微命令应能选择各种后继微地址来源。
- 转移地址字段 BAF 一般位数少，将它送到 μPC 的若干低位或高位形成后继微地址。

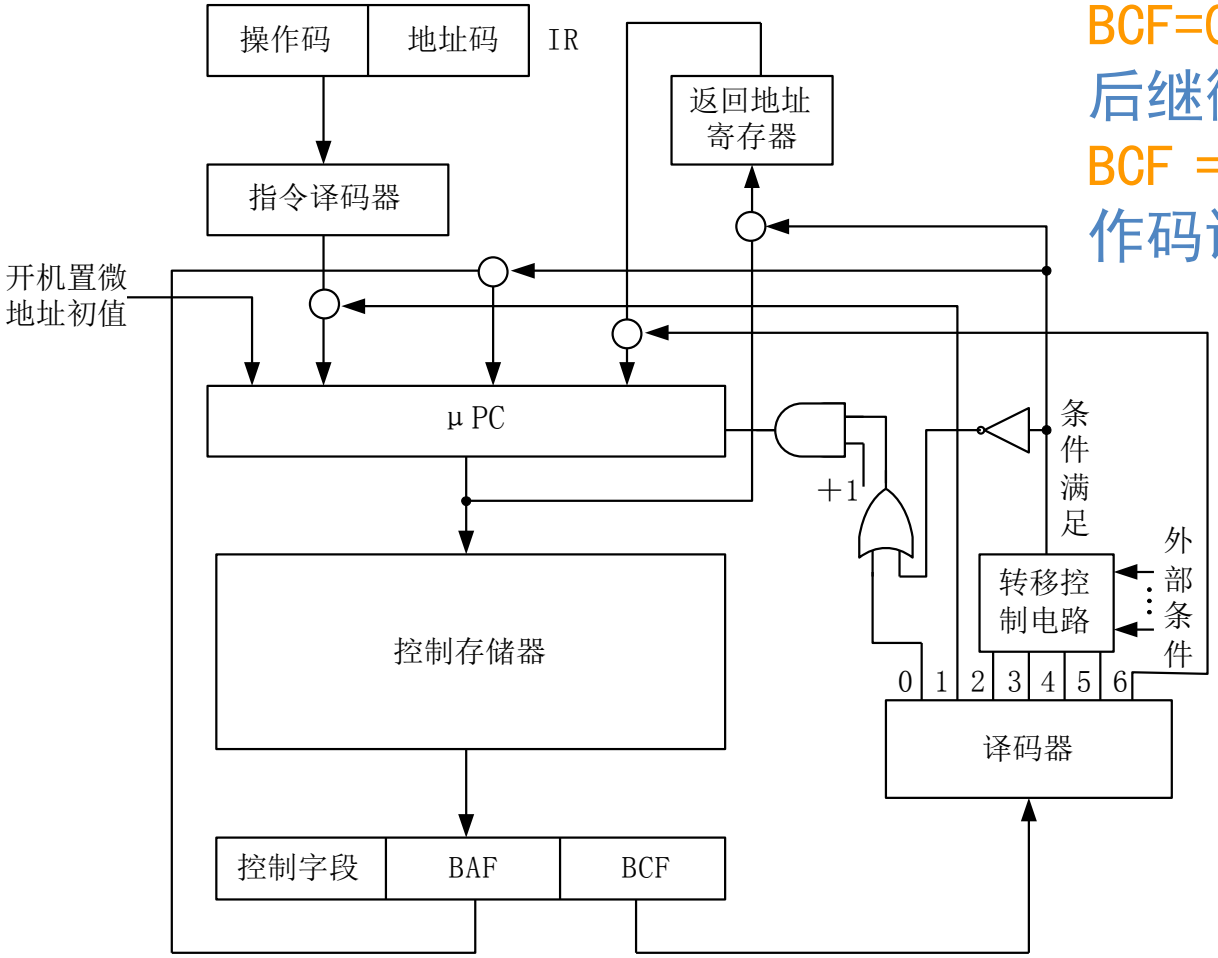


图7－19 计数器方式产生后继微地址的原理图

BCF=000，顺序执行微命令，后继微地址是 μPC＋1；
BCF ＝ 001，根据机器指令操作码译码产生后继微地址。

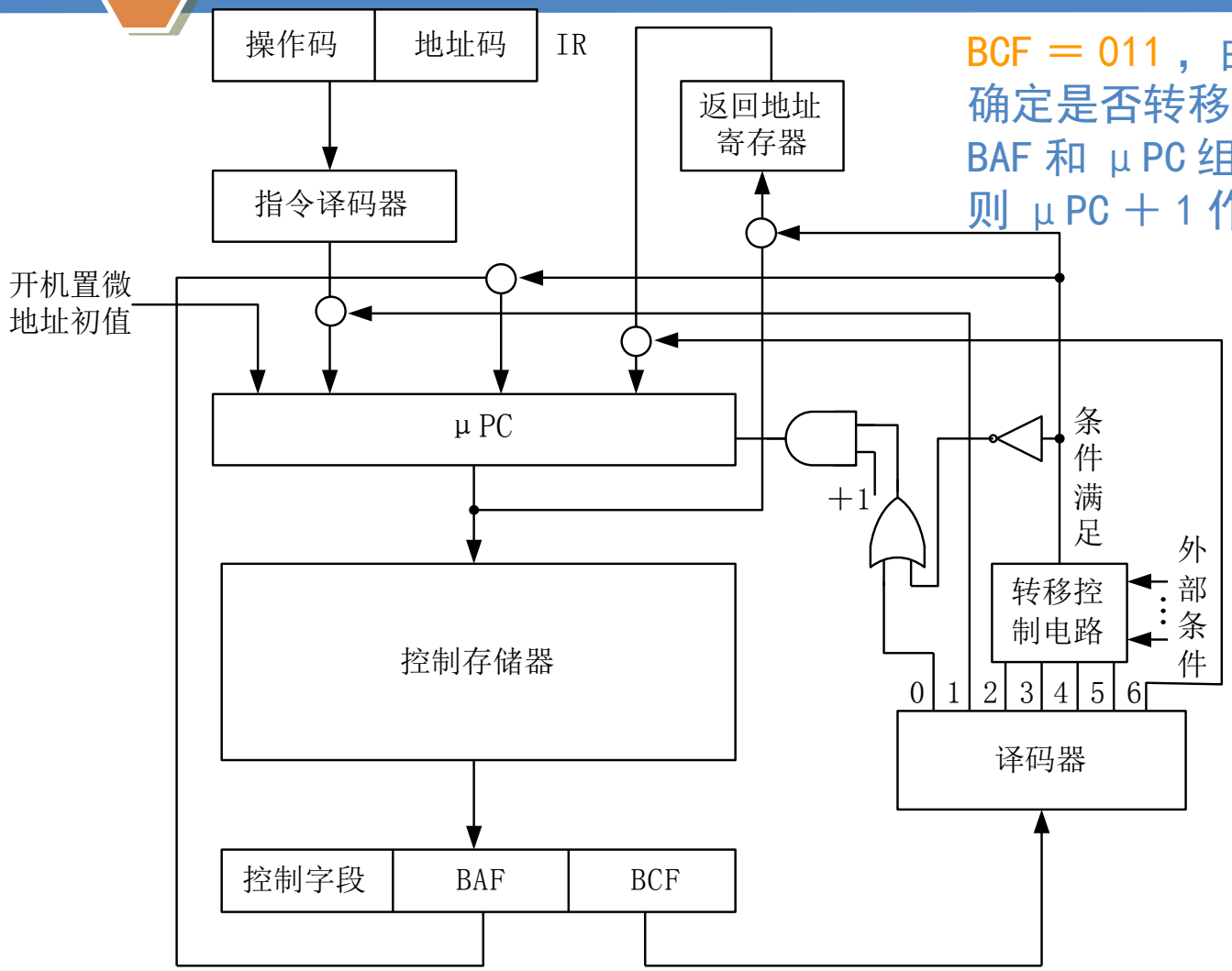


图7－19 计数器方式产生后继微地址的原理图

BCF ＝ 011，由转移条件的测试结果来确定是否转移。若测试条件满足，则 BAF 和 μPC 组合形成后继微地址；否则 μPC ＋ 1 作为后继微地址。

BCF ＝ 100，测试循环微命令，当测试条件（循环计数器内容为零）不满足时，循环继续，循环计数器减一，后继微地址由 BAF 与 μPC 组合形成；当测试条件满足时，循环结束，其后继微地址为 μPC ＋ 1，跳出循环。

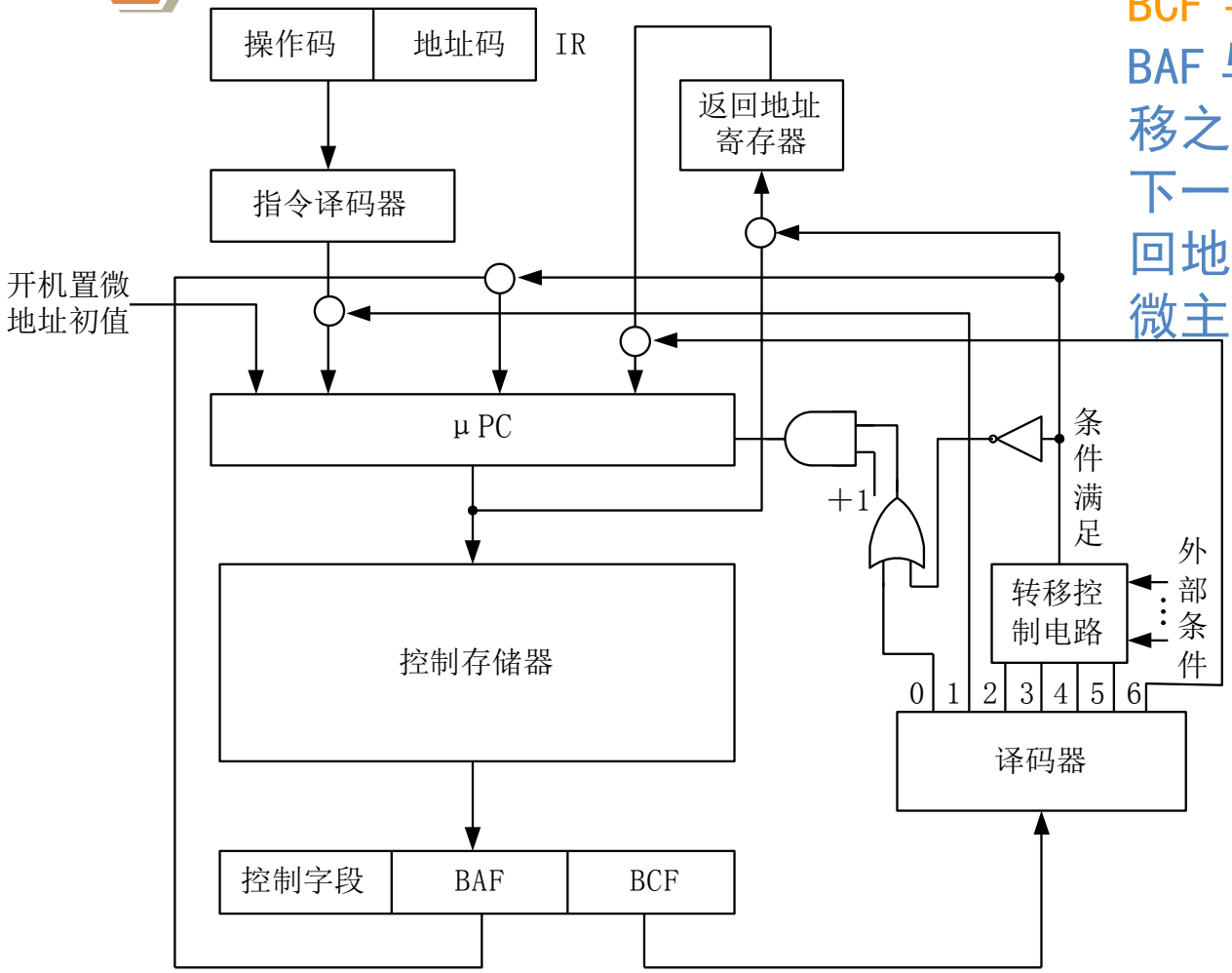


图7－19 计数器方式产生后继微地址的原理图

BCF ＝ 101，后继微地址由BAF 与 μPC 组合形成。在转移之前，要把该条微指令的下一微地址 μPC ＋ 1 送入返回地址寄存器中，以备返回微主程序。

BCF ＝ 110，后继微地址为返回地址寄存器中的内容，将其送入 μPC，从微子程序返回到原来的微主程序。

Ⅱ. 判定方式（下址字段法）

a) 后继微指令地址可由设计者指定或设计者指定的测试判别字段控制产生

- ✓ 当微程序不产生分支时，后继微指令地址直接由微指令的下址字段给出；
- ✓ 当微程序出现分支时，按测试判别字段和状态条件通过逻辑电路来形成后继微地址。

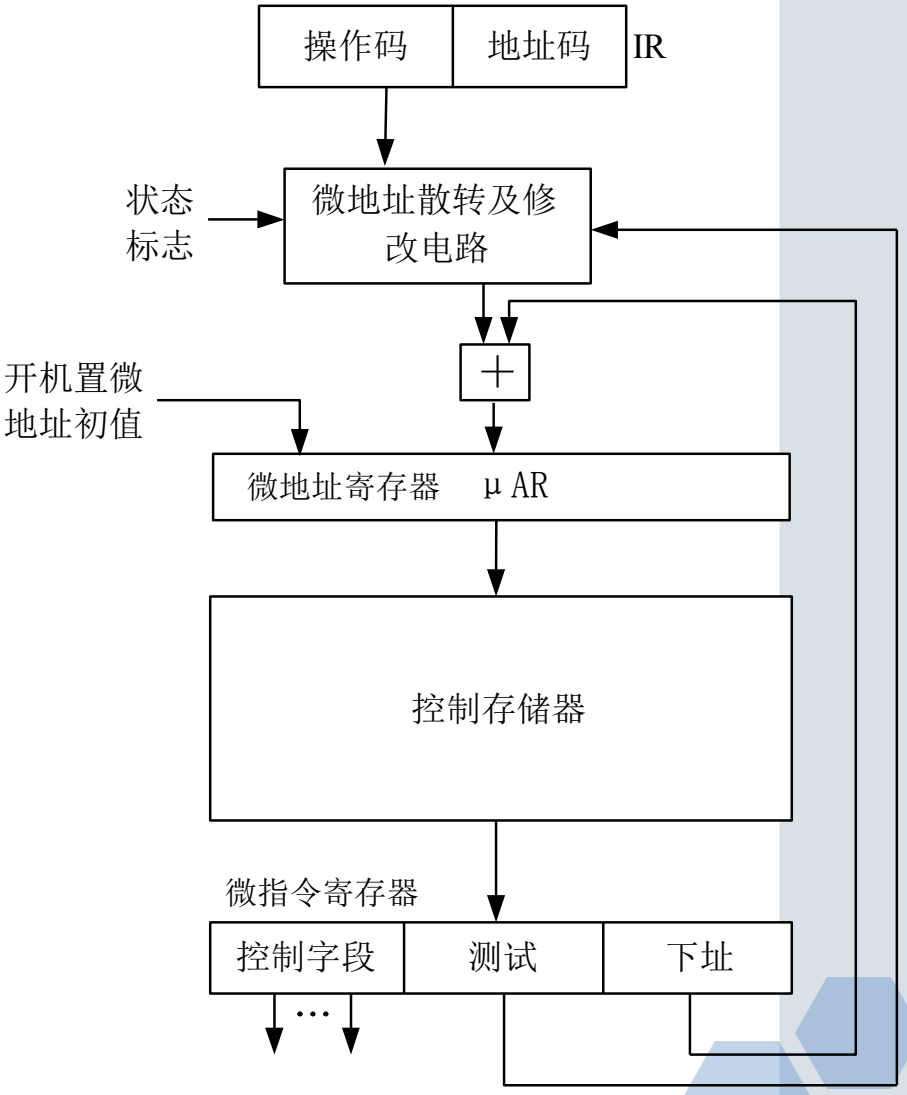
b) 在微指令格式中设置一个字段用来指明下一条要执行的微指令地址，所以也称为下址字段法

c) 每一条微指令至少都是一条无条件转移微指令，因此不必设置专门的转移微指令

判定方式产生后继微地址的原理

d）**优点：**可以实现快速多路分支，以提高微程序的执行速度，微程序在控制存储器中的物理分配方便，微程序设计灵活；

e）**缺点：**微指令字加长，形成后继微地址的结构比较复杂。



3、微指令格式的类型

- 水平型微指令

① 基本特征是：一条微指令能控制数据通路中多个功能部件并行操作。

② 优点：一条微指令可同时发许多个微命令，且微指令控制字段直接控制，微指令执行效率高，速度快，灵活，各部件执行操作的并行能力强；编制的微程序比较短。

③ 缺点：微指令字太长，明显地增加了控制存储器的横向容量。

④ 控制字段一般采用直接控制法和字段直接控

3、微指令格式的类型

- 垂直型微指令

采用完全编码的方法，将一套微命令代码化构成微指令。

① 一条微指令只能控制 1 ～ 2 种微操作，由微操作码、源地址和目标地址以及其他附带信息组成。

微操作码	源地址	目标地址	其他

1. 优点：比较直观，容易掌握和便于使用；微指令字短，减少了横向控制存储器的容量。

① 缺点：微指令要经过译码才能发出微命令，微指令的执行效率低，并行操作性比较差，增加