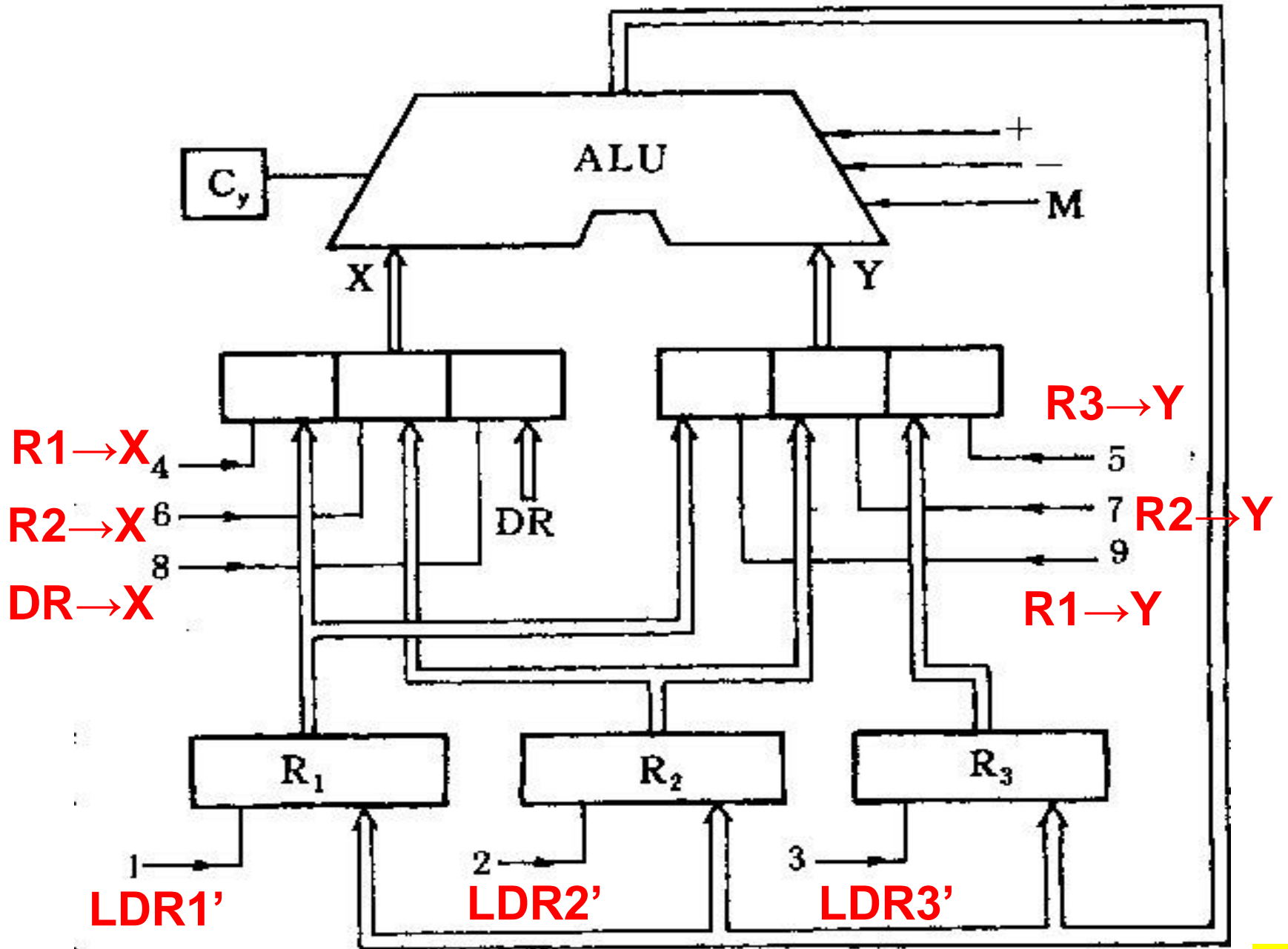


5.4 微程序控制器

5.4.1 微程序控制原理

- 将1个cpu周期内要发出的控制信号编成微指令，这样每条机器指令的所有操作可以编成一段由微指令组成的微程序；
- 将所有机器指令的微程序存到一个只读存储器——控制存储器（CM）里

CPU执行一条指令时，只需将CM中相应的一段微程序读出来，执行该段微程序，就可产生各种微操作信号，以实现该指令的功能。



□ 基本术语

- **微命令**：控制部件通过控制线向执行部件发出的**各种控制信号**。
- **微操作**：执行部件**接受微命令后所进行的操作**。
 - ◆ **相斥性**微操作：不能同时或在同一个CPU周期中出现的微操作。
 - ◆ **相容性**微操作：能同时或在同一个CPU周期中出现的操作。
- **微指令**：在机器的一个**CPU周期中**，一**组**实现一定操作功能的**微命令的组合**。
- **微程序**：实现一条机器指令功能的许多条微指令组成的序列。



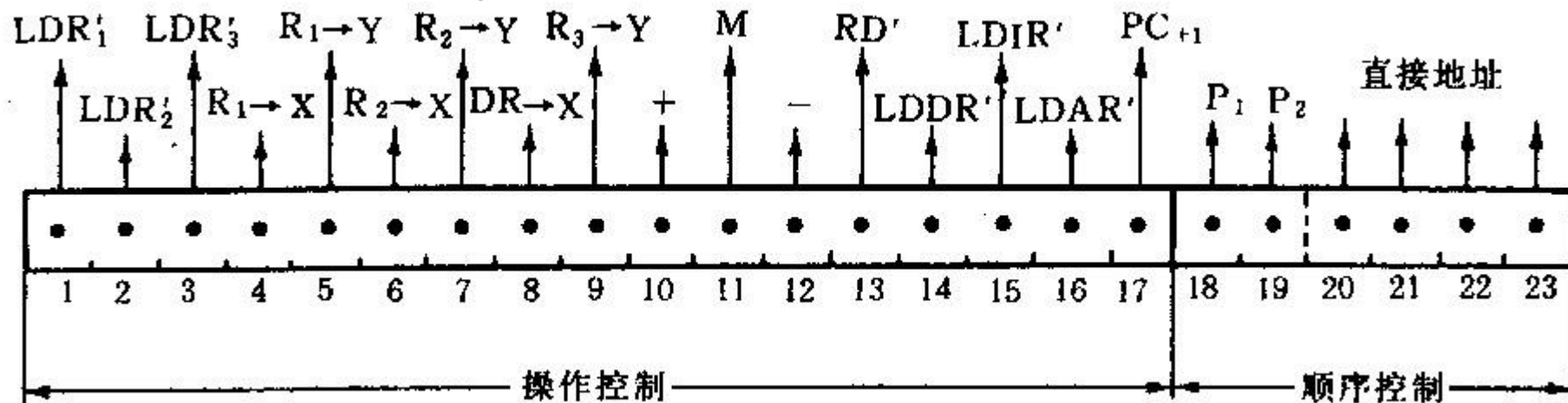
微指令格式



微指令： 在机器的一个CPU周期中，一组实现一定操作功能的微命令的组合。

- **微操作控制字段：** 产生**控制信号**。
- **顺序控制字段：** 产生**下条微指令的地址**。

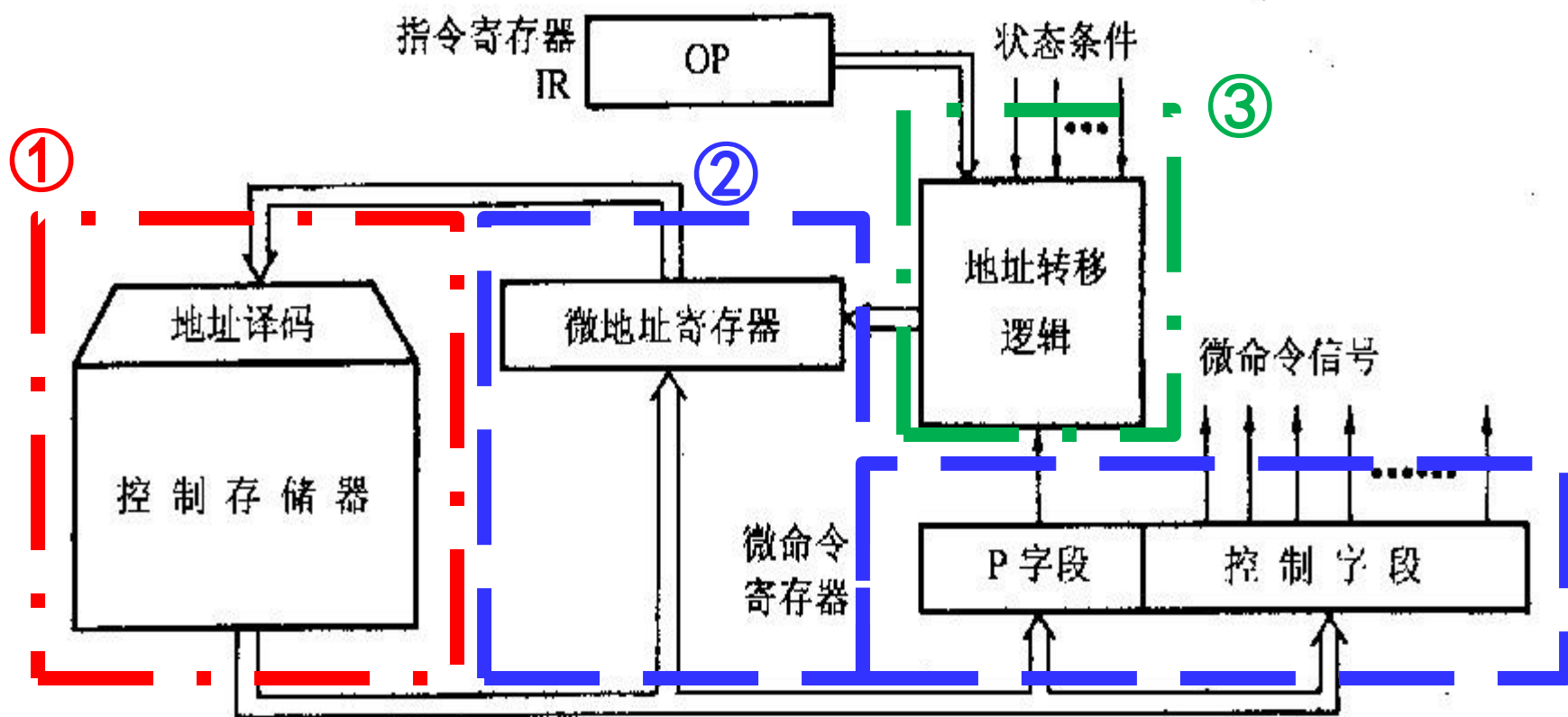
微指令给出的控制信号都是**节拍电位信号**。
部分微命令还要**和节拍脉冲信号相与**。



□ 微程序控制器组成原理



- ① 控制存储器：用于存放实现全部指令系统的微程序。
- ② 微指令寄存器：用来存放当前执行的一条微指令。
- ③ 地址转移逻辑：用于形成下条微指令的微地址。



□ 微程序举例

ADD R1, R2

A. 指令功能：
用BCD码来完成十进制的加法运算。

$R1+R2 \rightarrow R2$

B. 处理器部件

C. 指令流程

D. 微命令信号

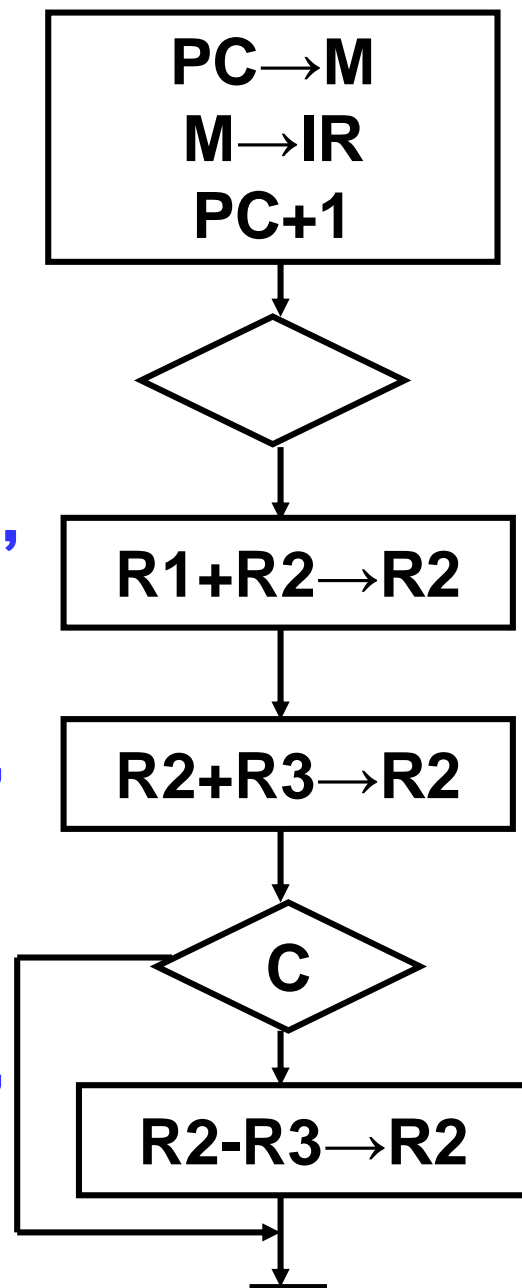


RD, LDIR,
PC+1

$R1 \rightarrow X, R2 \rightarrow Y,$
+, LDR2

$R2 \rightarrow X, R3 \rightarrow Y,$
+, LDR2

$R2 \rightarrow X, R3 \rightarrow Y,$
—, LDR2

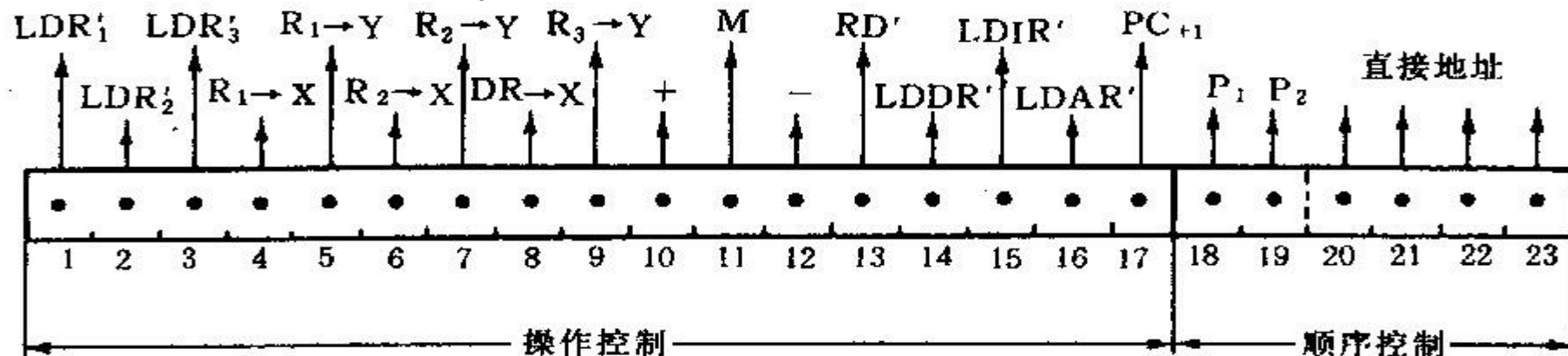


E. 微指令格式

顺序控制字段——决定下一条uI地址

- 前2位（18, 19）：测试标志；
- 后4位（20—23）：直接地址

- ① 当测试位为00，由20—23直接指示后继地址；
- ② 任意测试位为1，说明要进行相应测试，根据测试结果对20—23的某一位或几位进行修改，最终形成后继地址



F. 微指令编码



000	000	000	000	10101	10	0000
-----	-----	-----	-----	-------	----	------

010	100	100	100	00000	00	1001
-----	-----	-----	-----	-------	----	------

010	001	001	100	00000	01	0000
-----	-----	-----	-----	-------	----	------

010	001	001	001	00000	00	0000
-----	-----	-----	-----	-------	----	------

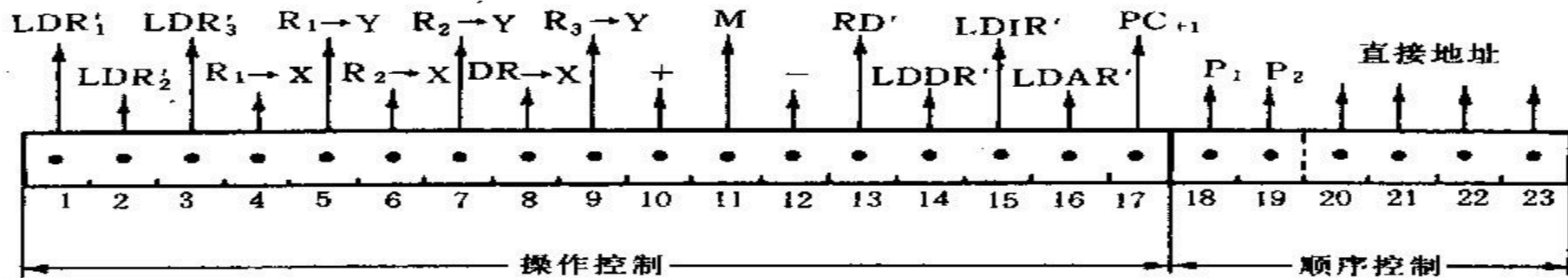
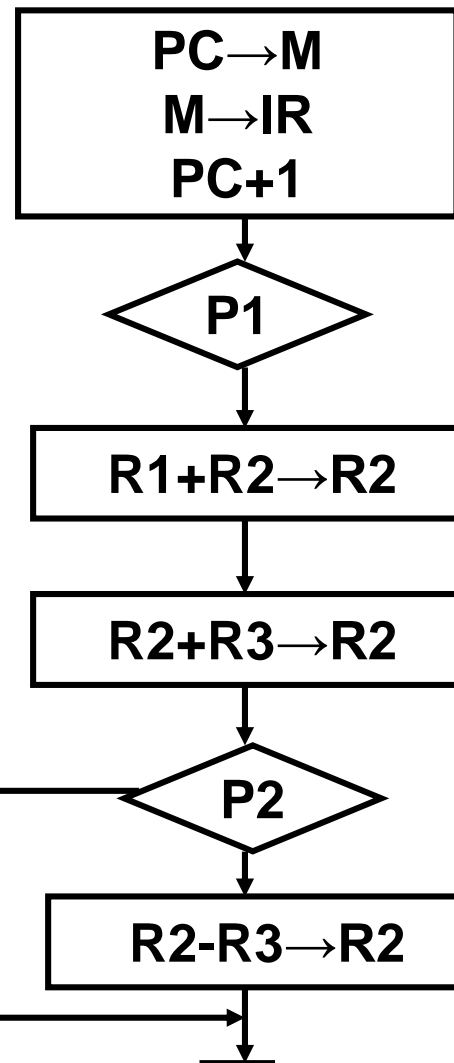
CM地址

0000

1010

1001

0001

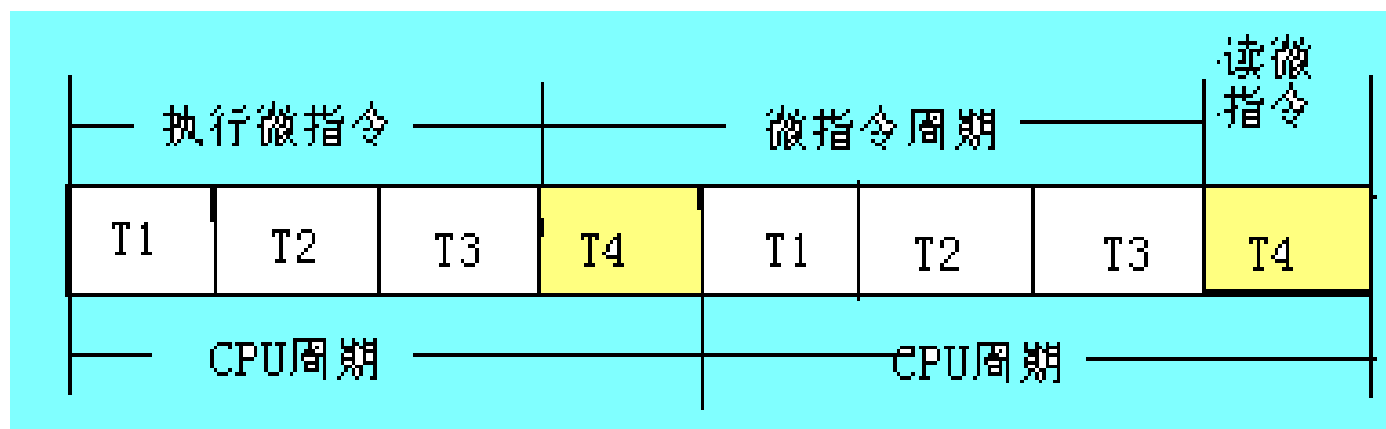


□ CPU周期与微指令周期的关系

微指令周期 = 读出微指令的时间 + 执行该条微指令的时间

机器指令与微指令的关系

- 一条机器指令的功能是由若干条微指令组成的序列来实现的。
- 机器指令存储在内存中，CPU外部；微指令存储在CM中，属于CPU内部。
- 每一个CPU周期对应一条微指令



5.4.2 微程序设计技术

□ 微命令编码方式

1. **直接表示法**：将控制字段的**每个二进制位**定义为一个**微命令**。
 - 优点：简单，执行速度快，操作并行性好。
 - 缺点：微指令字长过长，位空间利用率低。
2. **编码（分段译码）表示法**：把**一组相斥性的微命令**信号组合在一个**字段**，然后通过译码器对每个微命令信号进行译码，**译码的输出作为控制信号**。
 - 优点：可用较少的二进制位表示较多的微命令信号，缩短CM的字长。
 - 缺点：增加了译码电路，速度相对要慢一些。
3. **混合表示法**
 - 为提高灵活性，常在微指令中增设一个常数字段。



8. 某机有 8 条微指令 $I_1 \sim I_8$ ，每条微指令所包含的微命令控制信号如下表所示。

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	✓	✓						✓		

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为 8 位，请安排微指令的控制字段格式。

分析：采用混合方法——若干组互斥性的微命令，每组采用编码表示法；其余微命令采用直接表示法。

- ① 互斥的微命令（b, f, i, j）占用2位；a,c,d,e,g,h 各占1位
- ② 互斥的微命令（b, g, i）占用2位；互斥的微命令（e, h, f）占用2位；a,c,d,j 各占1位

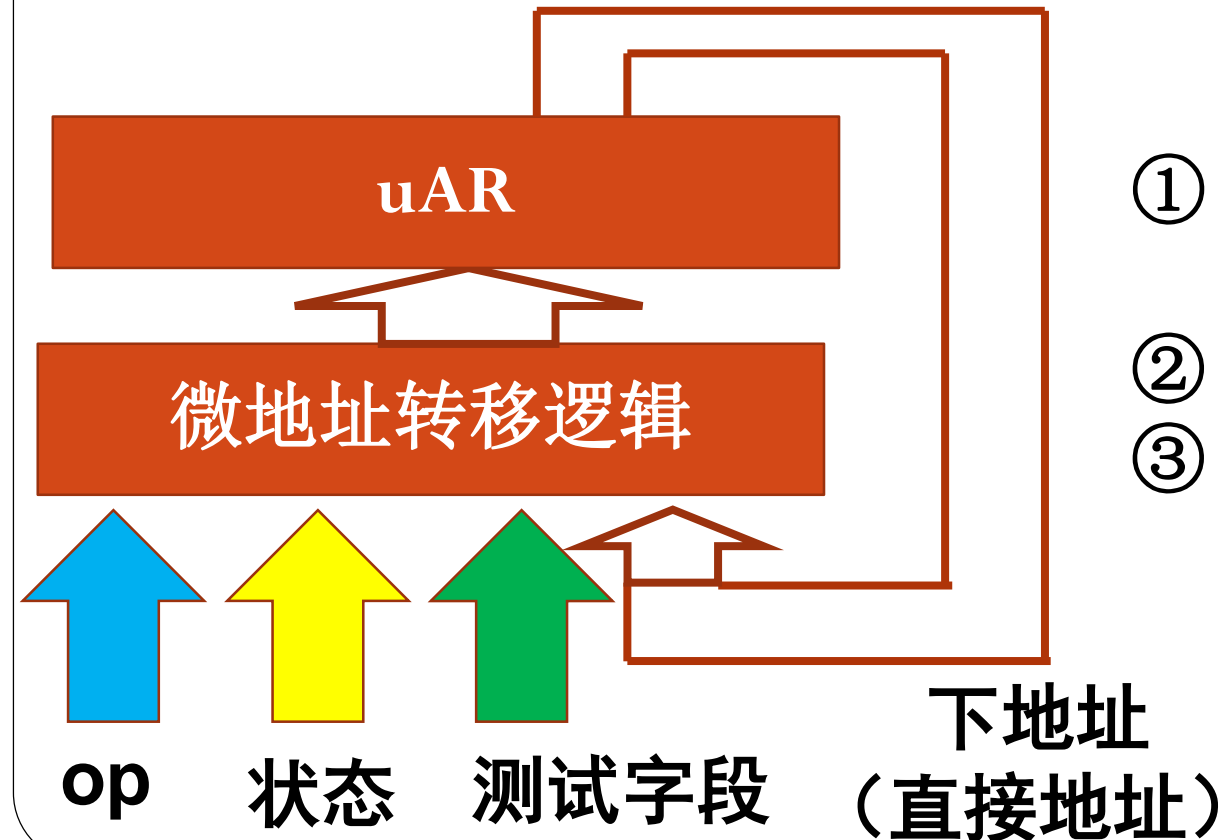
□ 微地址的形成方法

1. 计数器方式（增量方式）：顺序执行时，后继地址在现行微地址上加上一增量；非顺序执行时，需执行一条转移微指令。
 - 优点：顺序控制字段较短，设计简单。
 - 缺点：多路并行转移弱，速度较慢，灵活性差。
2. 多路转移方式（断定）：将顺序控制字段分成测试字段（P）和下地址字段二段。未出现多路分支时，后继由下地址字段直接给出；出现多路分支时，根据测试字段值和状态条件选择转移地址。
 - 优点：多路转移灵活，速度快；
 - 缺点：转移地址硬件设计复杂。



当采用多路转移方式（断定方式）应注意：

1. 直接地址字段的长度，要保证在未出现多路分支时，寻址范围能覆盖整个控制存储器（CM）
2. 出现多路分支时，转移地址由微地址转移逻辑产生。微地址转移逻辑的输入来源一般包括：



- ① 微指令中的测试字段、下地址
- ② 指令的操作码
- ③ 执行部件的状态标志

3. 当微地址转移逻辑要进行测试时

一般地，对n位的状态标志（或是操作码）进行某一种测试P，最终会相应地对uAR中的n位进行修改，则最多可产生 2^n 种分支地址。

例2 微地址寄存器有6位($\mu A5-\mu A0$)，当需要修改其内容时，可通过某一位触发器的强置端S将其置“1”。现有三种情况：

- (1)执行“取指”微指令后，微程序按IR的OP字段(IR3-IR0)进行16路分支；
 - (2)执行条件转移指令微程序时，按进位标志C的状态进行2路分支；
 - (3)执行控制台指令微程序时，按IR4，IR5的状态进行4路分支。
- 请按多路转移方法设计微地址转移逻辑。

【解】

- 按题意，微程序有三种判别测试，分别为P1，P2，P3。由于修改 $\mu A5$ - $\mu A0$ 内容具有很大灵活性，现分配如下：

(1)用P1和IR3-IR0修改 $\mu A3$ - $\mu A0$ ；

(2)用P2和C修改 $\mu A0$ ；

(3)用P3和IR5，IR4修改 $\mu A5$ ， $\mu A4$ 。

若修改地址必须在T4时钟的上升沿，则可有：

$$\mu A5 = P3 \cdot IR5 \cdot T4;$$

.....

$$\mu A3 = P1 \cdot IR3 \cdot T4$$

.....

$$\mu A0 = P1 \cdot IR0 \cdot T4 + P2 \cdot C \cdot T4$$

例 某机器微地址寄存器有**5位($\mu A4-\mu A0$)**；指令系统共包含+、-、*、/ **4种操作**，由指令中操作码**IR5、IR4**表示。

取指令后，对IR5、IR4经过P1测试可产生4路分支，分支入口微地址与操作码对应关系如下：

操作	IR5 、 IR4	μAR
+	00	01100
-	01	01101
*	10	01110
/	11	01111

Q:

- ① 请按多路转移方法设计微地址转移逻辑。
- ② 完成**取指的微指令**中，直接地址字段应设为？
- ③ 若对C进行P2测试，则可能产生几个去向？

□ 微指令的格式

1. 水平型微指令——在一个微指令周期内，同时给出多个能并行操作的微命令。
 - 全水平型
 - 字段译码型
 - 混合型
2. 垂直型微指令——在微指令中设置微操作码字段和地址码字段，采用微操作码编译法，并由微操作码规定微指令的功能。

□ 动态微程序设计

静态微程序设计：微程序设计好后，不再（能）修改；

动态微程序设计：微程序可根据需要再修改。

微程序设计流程总结：

1. 确定机器的指令系统
2. 确定数据通路结构
3. 分析每条指令，画出整个指令系统的CPU周期流程图
4. 列明每条微指令对应发出的微命令
5. 确定微指令的格式（包括微命令、微地址形成方式）
6. 给每条微指令分配好在CM中的存储单元地址
7. 根据4和微命令格式确定操作控制字段；根据6和微地址形成方式确定顺序控制字段