

3.3 微程序控制方式

3.3.1 基本概念与微程序控制的基本思想

一、基本概念：

1. **微命令**：构成控制信号序列的最小单位，又称微信号，指那些直接作用于部件或控制门电路的命令。如：打开或关闭某传送通路的电信命令，或对触发器或R进行同步打入，置位、复位等的控制脉冲。

2. **微操作**：由微命令控制实现的最基本的操作称

为微操作，如：开门、关门、选择。

3. **微周期**：从控制存储器中读取一条微指令并执行相应一步操作所需的时间，称为一个微周期或微指令周期。通常一个时钟周期为一个微周期。

4. **微指令**：每个微周期的操作所需的微命令组成一条微指令。从控制存储器的组织角度讲，每个单元存放一条微指令。

5. **微程序**：一系列微指令的有序集合称为微程序，用来解释执行一条机器指令。

6. 对应关系：一条机器指令 \longleftrightarrow 一段微程序 \longleftrightarrow 一系列微指令 \longleftrightarrow (若干) 条微指令 \longleftrightarrow (若干) 条微命令

二. 微程序控制的基本思想

1. 若干微命令编制成一条微指令，控制实现一步操作；
2. 若干微指令组成一段微程序，解释执行一条机器指令；
3. 微程序事先存放在控制存储器中，执行机器指令时再取出。

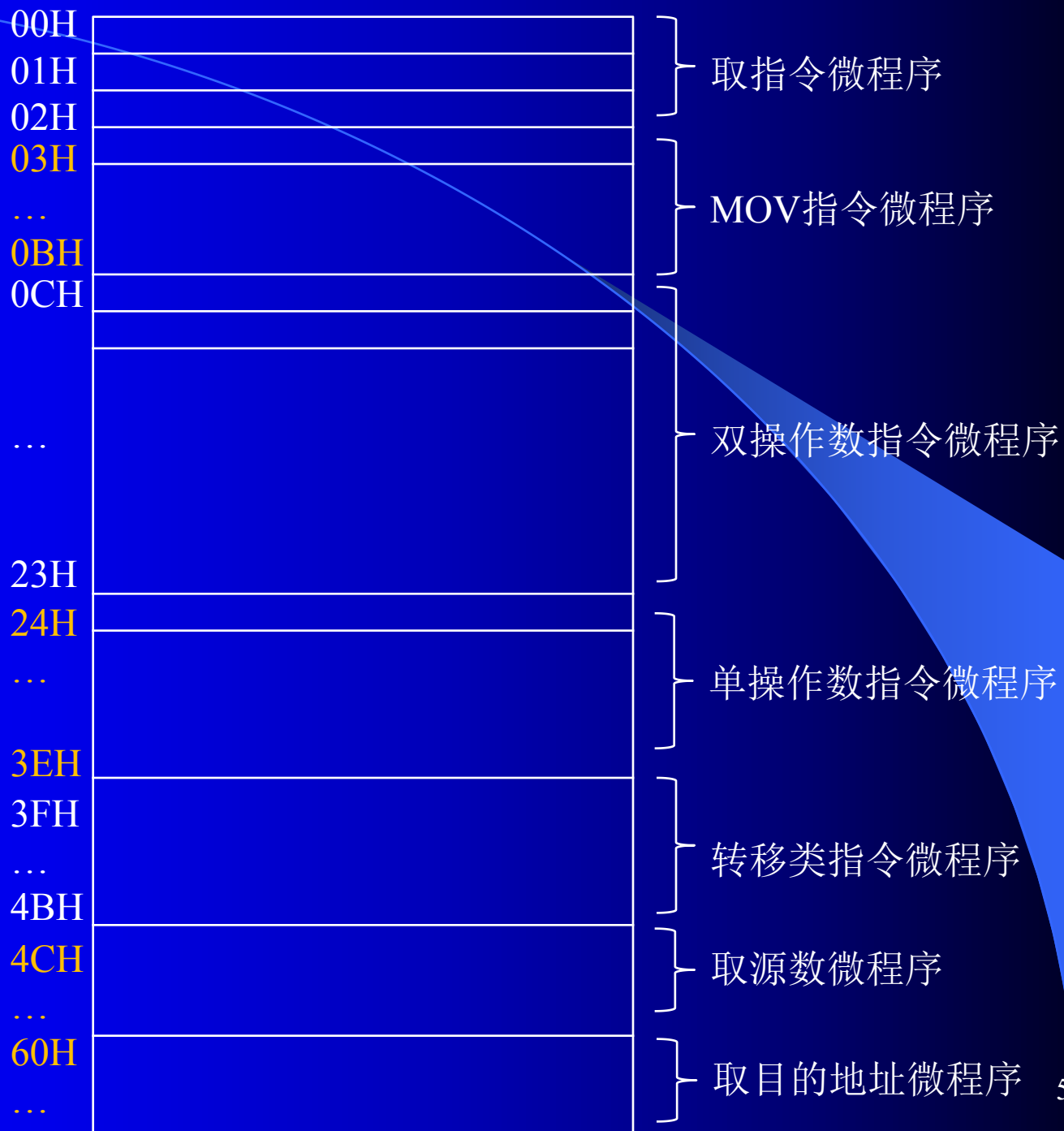
基本思想:对应关系

| | 分步 | 微命令序列 | 微指令 | |
|------------|-----------------------|--|------|-----|
| 机器指令 | $M \rightarrow IR,$ | EMAR, R, SIR... | 微指令1 | 微程序 |
| MOV R1, R0 | $PC+1 \rightarrow PC$ | $PC \rightarrow A, A+1, DM, \dots$ | 微指令2 | |
| | $R_0 \rightarrow R_1$ | $R0 \rightarrow A, \text{传}A, DM, \dots$ | 微指令3 | |
| | $PC \rightarrow MAR$ | $PC \rightarrow A, \text{传}A, DM, \dots$ | 微指令4 | |

一条**机器指令**的执行对应一段**微程序**;

一段**微程序**可包含多条**微指令**;

一条**微指令**包含一步操作所需**微命令**



例如: MOV (R₀) , (R₁)

FT: M → IR EMAR (一条微命令)

R (一条微命令)

SIR (一条微命令)

PC+1 → PC PC → A (一条微命令)

A+1 (一条微命令)

DM (一条微命令)

CPPC (一条微命令)

1 → ST (一条微命令)

CPST (一条微命令)

ST₀: R₁ → MAR R₁ → A/B (一条微命令)

A/B (一条微命令)

DM (一条微命令)

CPMAR (一条微命令)

T+1 (一条微命令)

CPT (一条微命令)

一条微指令

一条微指令

一段微程序

$ST_1: M \rightarrow MDR \rightarrow C$

EMAR (一条微命令)

R (一条微命令)

SMDR (一条微命令)

MDR \rightarrow B (一条微命令)

B (一条微命令)

DM (一条微命令)

CPC (一条微命令)

1 \rightarrow DT (一条微命令)

CPDT (一条微命令)

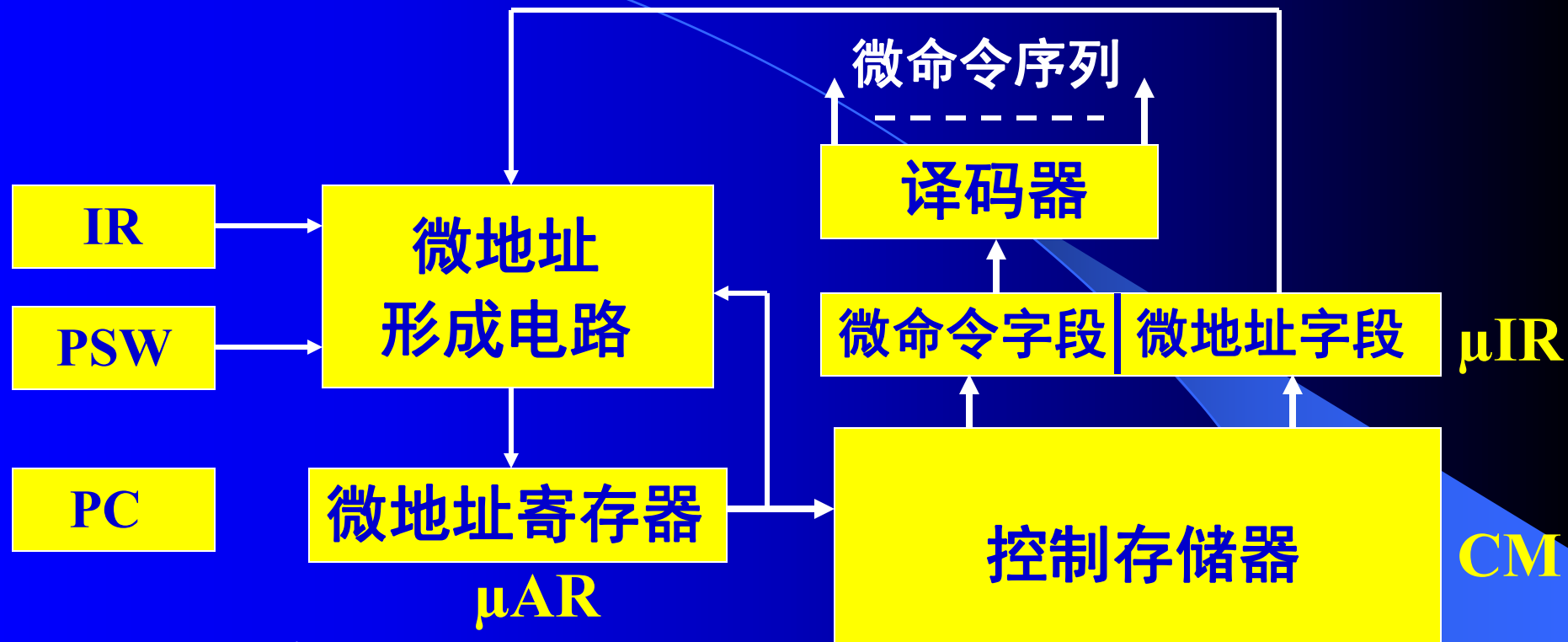
.....

$DT_0: R_0 \rightarrow MAR$

一条微指令

一段微程序

3.3.2 微程序控制器组成及工作原理

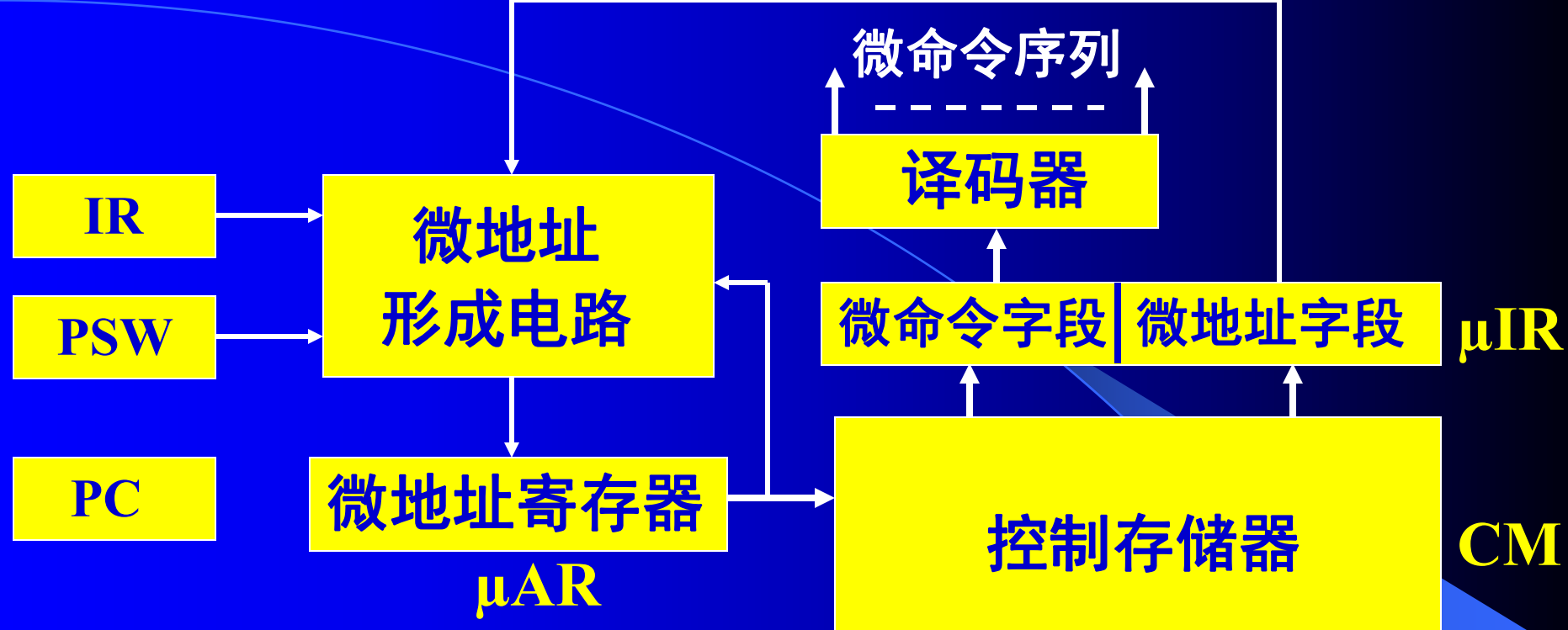


一. 组成部件

1. 控制存储器CM

功能：存放微程序。

CM属于CPU，不属于主存储器。

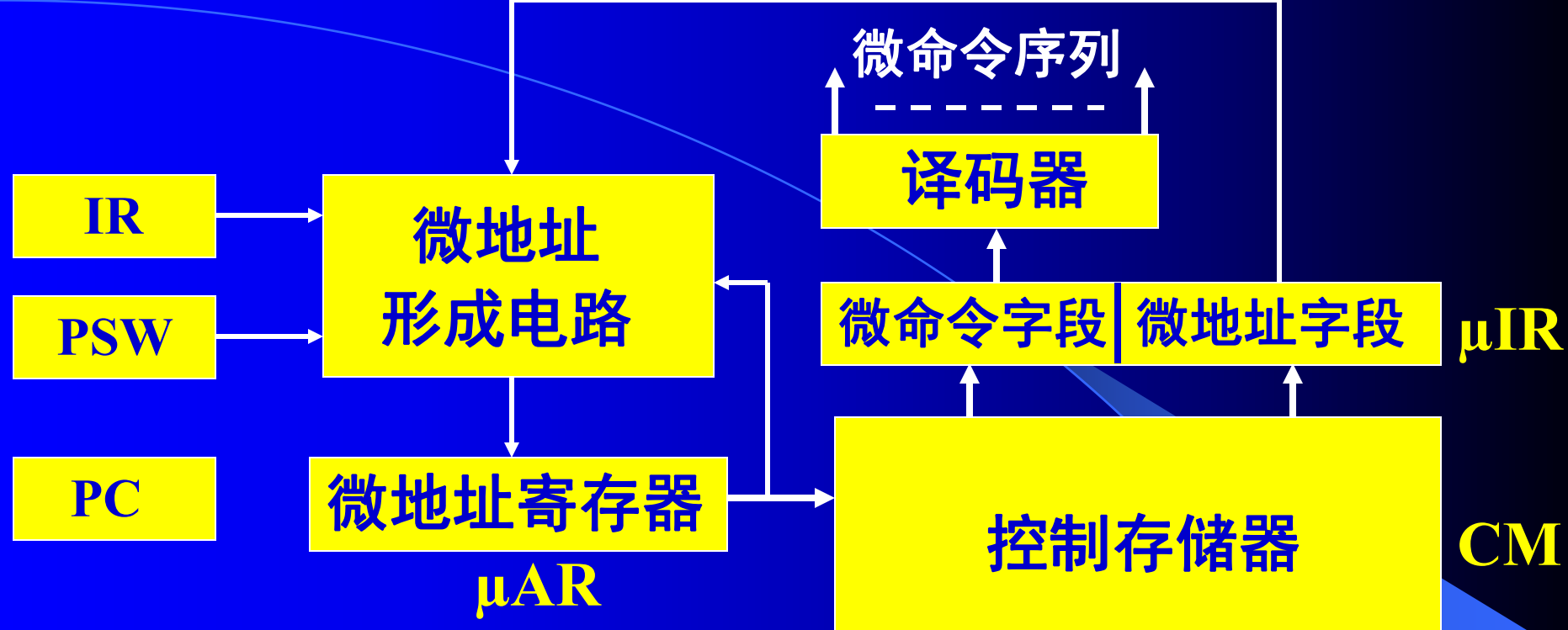


2. 微指令寄存器 μIR

功能：存放现行微指令。

微命令字段：提供一步操作所需的微命令。
(微操作控制字段)

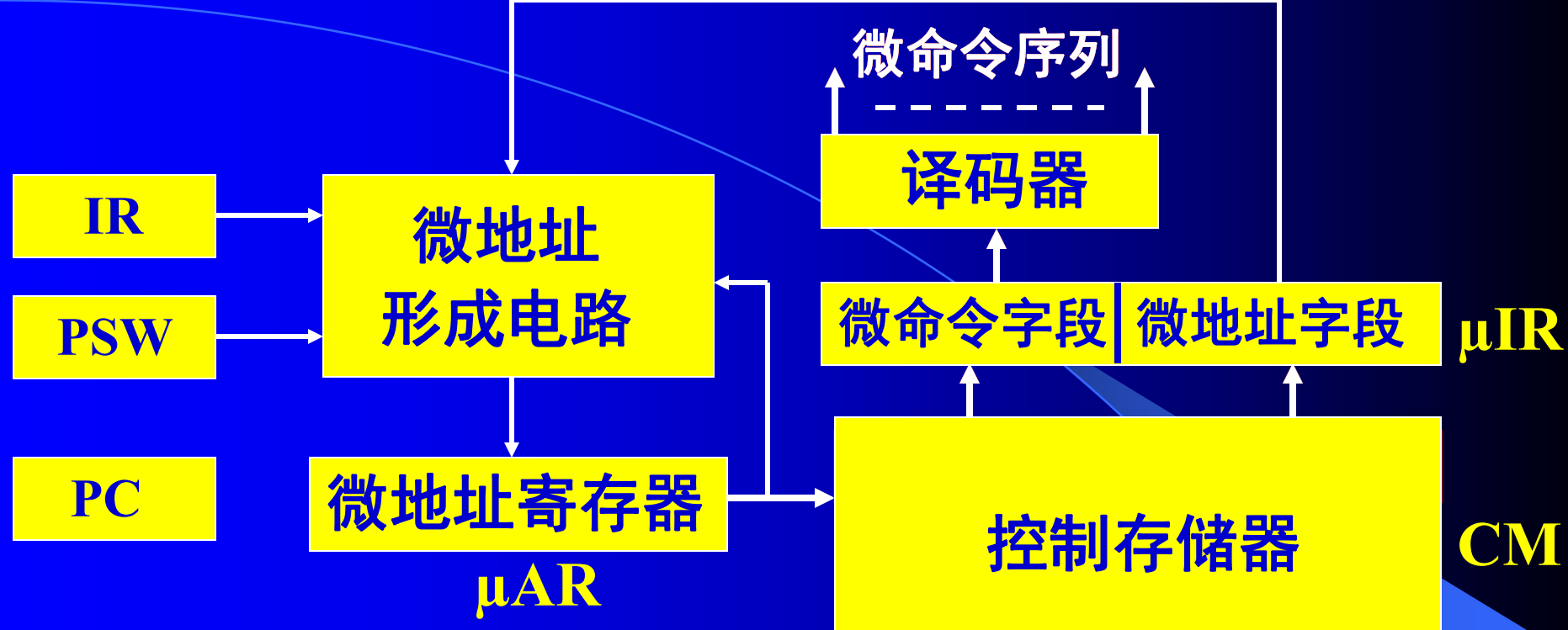
微地址字段：指明后续微地址的形成方式。
(顺序控制字段) 提供微地址的给定部分。



3. 微地址形成电路

功能：提供两类微地址。

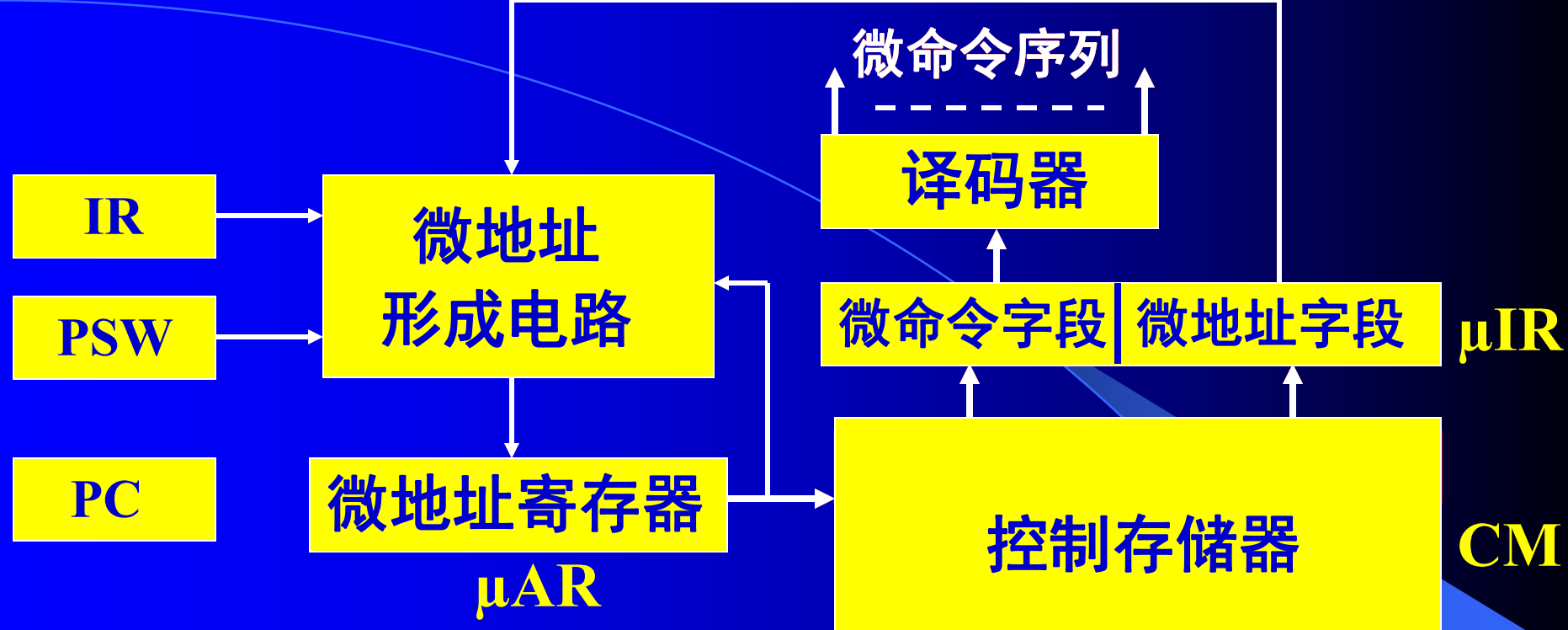
- 微程序入口地址：由机器指令操作码形成。
- 后续微地址：由微地址字段、现行微地址、运行状态等形成。



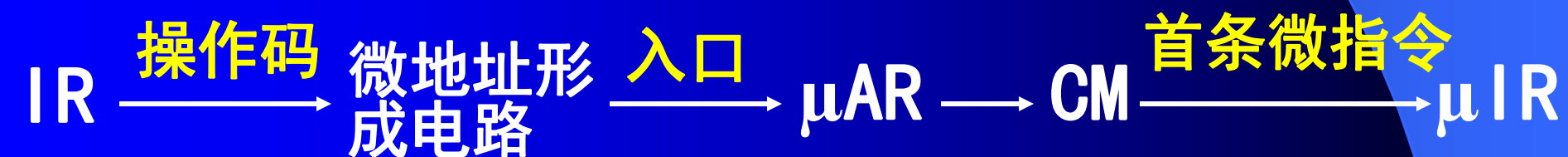
二. 工作过程

1. 取机器指令



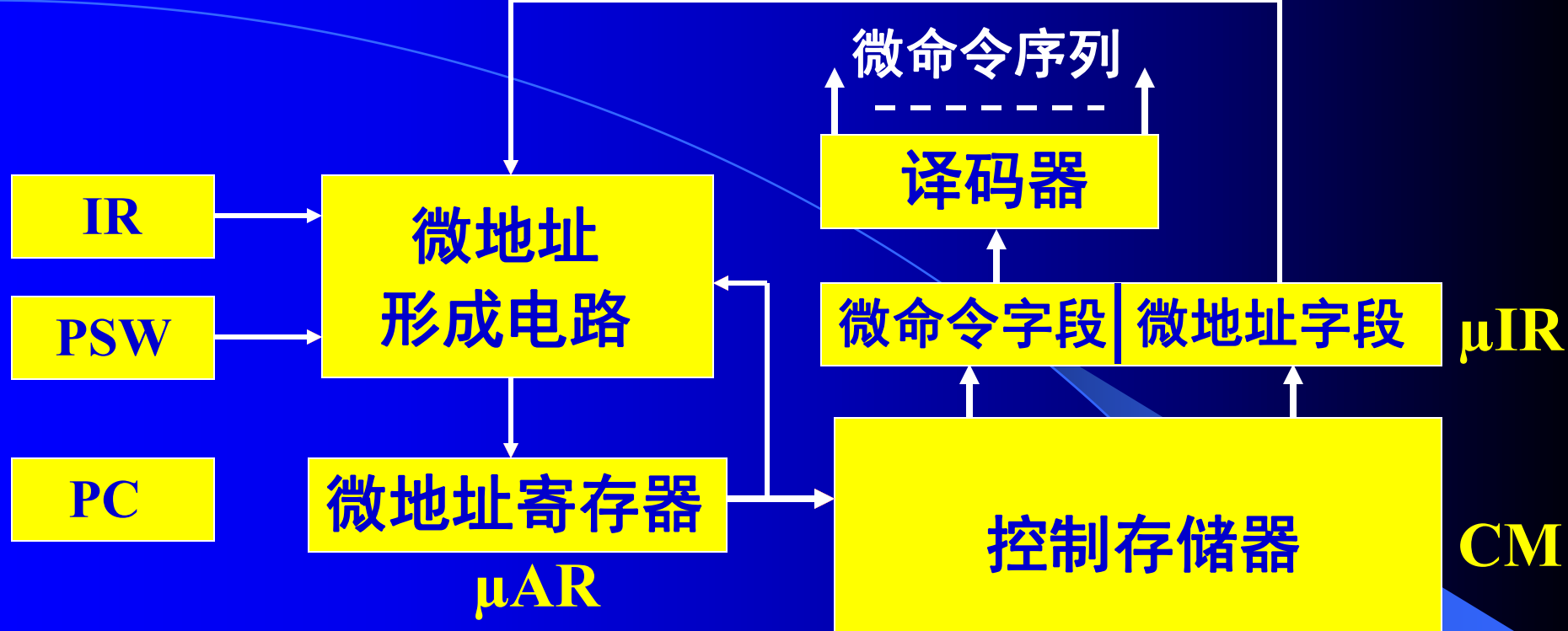


2. 转微程序入口



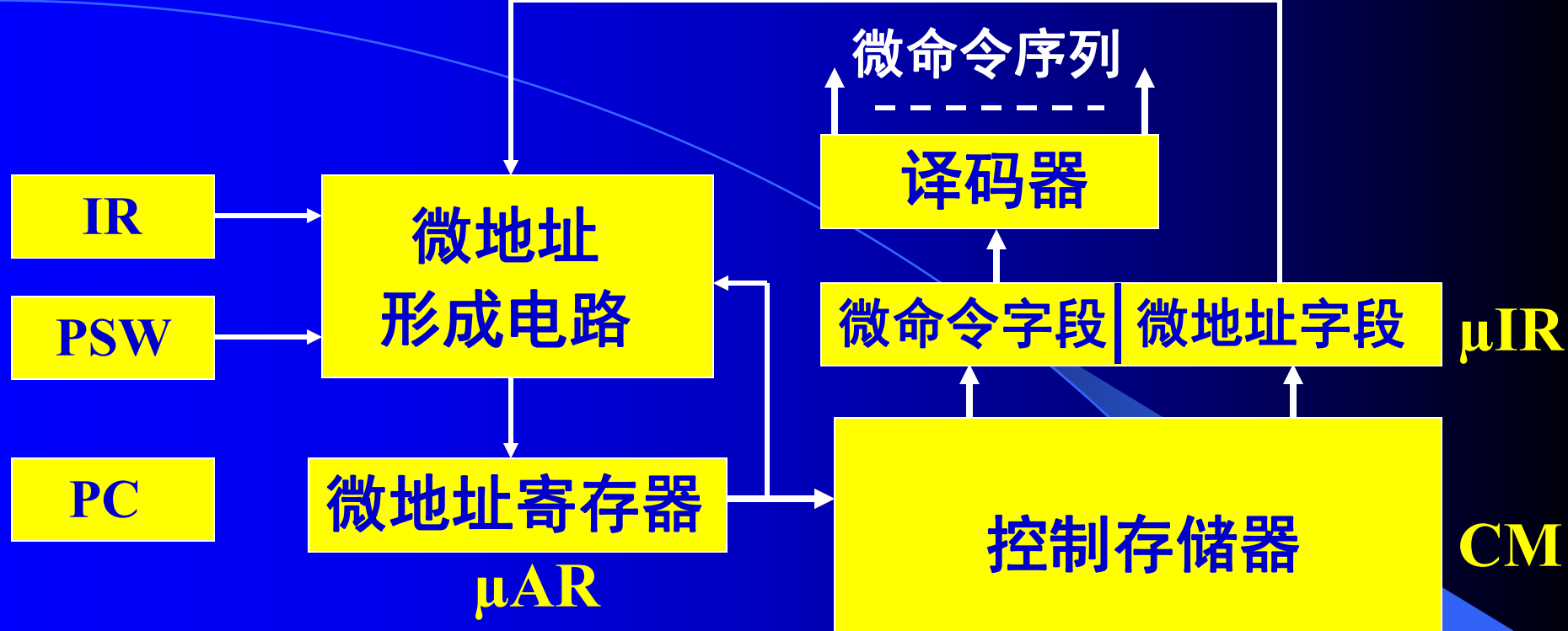
3. 执行首条微指令





4. 取后续微指令





5. 执行后续微指令

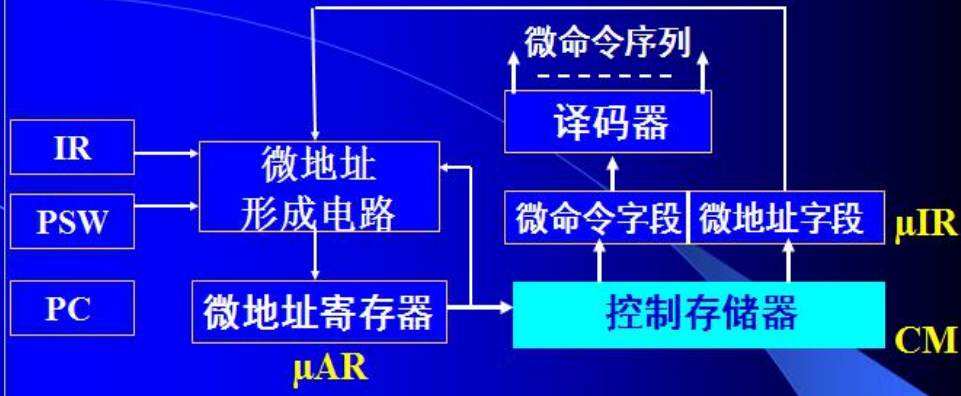
同3

6. 返回

微程序执行完，返回CM（存放取指微指令的固定单元）。

具体工作过程:

- (1) 取指微程序
- (2) 转微程序入口
即机器指令对应微程序的第一条微指令
- (3) 逐条取出微指令
译码产生微命令
- (4) 返回到取指微指令



| | | |
|-----|-----------------------------|-------------------|
| 00H | 0000000000000000011011 0000 | 取指令 微程序 |
| 01H | 1000001001001001110000 0000 | |
| 02H | 0000000000000000000000 0010 | |
| 03H | 转“取源操作数” | MOV指令 微程序 |
| ... | ! | |
| 0BH | 转“取指”入口 | |
| 0CH | 转“取源操作数” | |
| | 转“取目的地址” | 双操作数 指令微程 序 |
| ... | | |
| 23H | | |
| 24H | 转“取目的地址” | 单操作数 指令微程 序 |
| ... | | |
| 3EH | | |
| 3FH | | 转移类指 令微程序 |
| ... | | |
| 4BH | | 取源数微 程序 |
| 4CH | | |
| ... | | 取目的地 址微程序 |
| 60H | | |
| ... | | |

3.3.3 微指令格式和编码方法

一. 格式分类

1. 垂直型微指令

一条微指令定义并执行一种基本操作。

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。

2. 水平型微指令

一条微指令定义并执行几种并行的基本操作。

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。

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。

微指令不长，便于编写；微程序不长，执行速度加快。

例. 长城203微指令

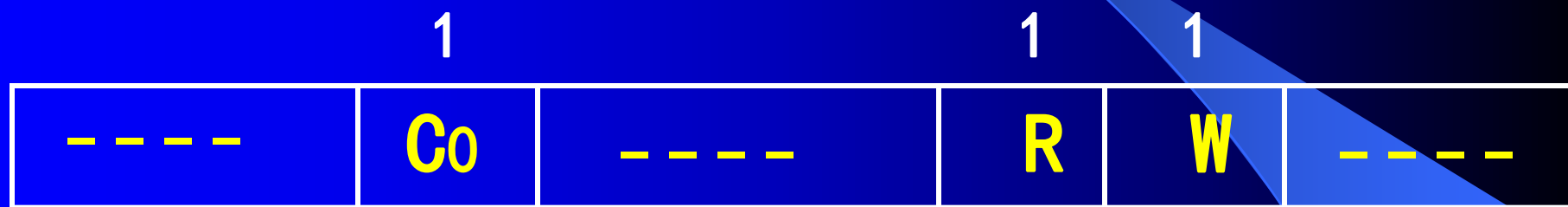


二. 编码方法

1. 直接控制法

微命令按位给出。

例. 某微指令



$C_0 = \begin{cases} 0 & \text{进位初值为0} \\ 1 & \text{进位初值为1} \end{cases}$

$R = \begin{cases} 0 & \text{不读} \\ 1 & \text{读} \end{cases}$

$W = \begin{cases} 0 & \text{不写} \\ 1 & \text{写} \end{cases}$

不需译码，产生微命令的速度快；
信息的表示效率低。

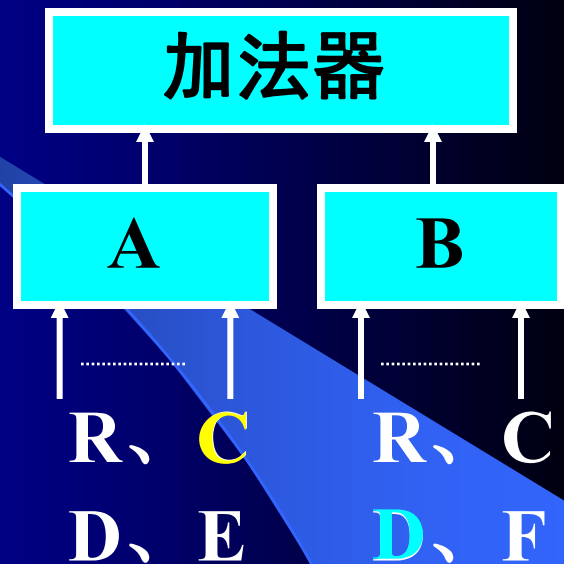
微指令中通常只有个别位采用直接控制法。

2. 分段直接编译法

微命令由字段编码直接给出。

例. 对加法器输入端进行控制。

微指令中设置AI字段，控制加法器的输入选择。

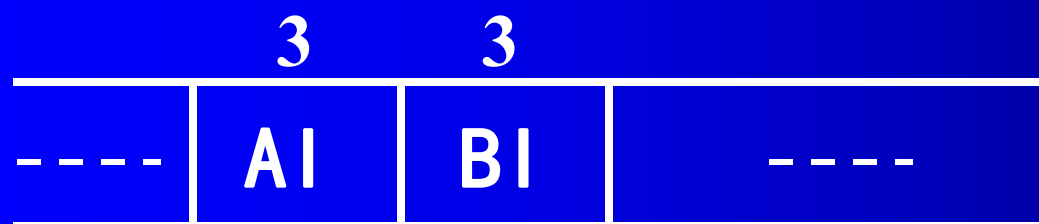


| 3 | | |
|-----|-------|-----|
| --- | AI | --- |
| 000 | 不发命令 | ? |
| 001 | R → A | |
| 010 | C → A | |
| 011 | D → B | |
| 100 | F → B | |
| ⋮ | | |

微命令分组原则：
同类操作中互斥的
微命令放同一字段。

不能同时出现

加法器A输入端的控制命令放AI字段，B输入端的控制命令放BI字段。



AI: 000 不发命令

001 R → A

010 C → A

011 D → A

100 E → A

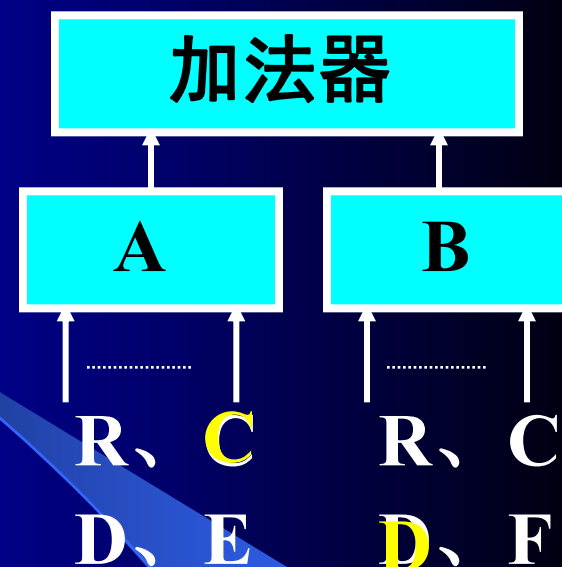
BI: 000 不发命令

001 R → B

010 C → B

011 D → B

100 F → B



操作唯一；编码较简单；一条微指令能同时提供若干微命令，便于组织各种操作。

3. 分段间接编译法

微命令由本字段编码和其他字段解释共同给出。

1) 设置解释位或解释字段

例.

| | |
|---|---|
| C | A |
|---|---|

 $C = \begin{cases} 1 & A \text{ 为某类命令} \\ 0 & A \text{ 为常数} \end{cases}$

解释位

2) 分类编译

按功能类型将微指令分类，分别安排各类微指令格式和字段编码，并设置区分标志。

4. 其他编码方法

1) 微指令译码与机器指令译码复合控制

例. 机器指令



译码器



R1

A_i门

微指令



译码器



2) 微地址参与解释

例. 微地址 微指令

| | | | |
|-----|-----|------|-----|
| 004 | --- | 取指标志 | --- |
| 011 | --- | 变址标志 | --- |

3.3.4 微地址形成方式

1. 微程序入口地址的形成

指令操作码 $\xrightarrow{\text{功能转移}}$ 微程序入口

(1) 一级功能转移

各操作码的位置、位数固定，一次转换成功。

入口地址=页号，操作码

例.

机器指令1

| | |
|---------|-------|
| 0F (8位) | ----- |
|---------|-------|

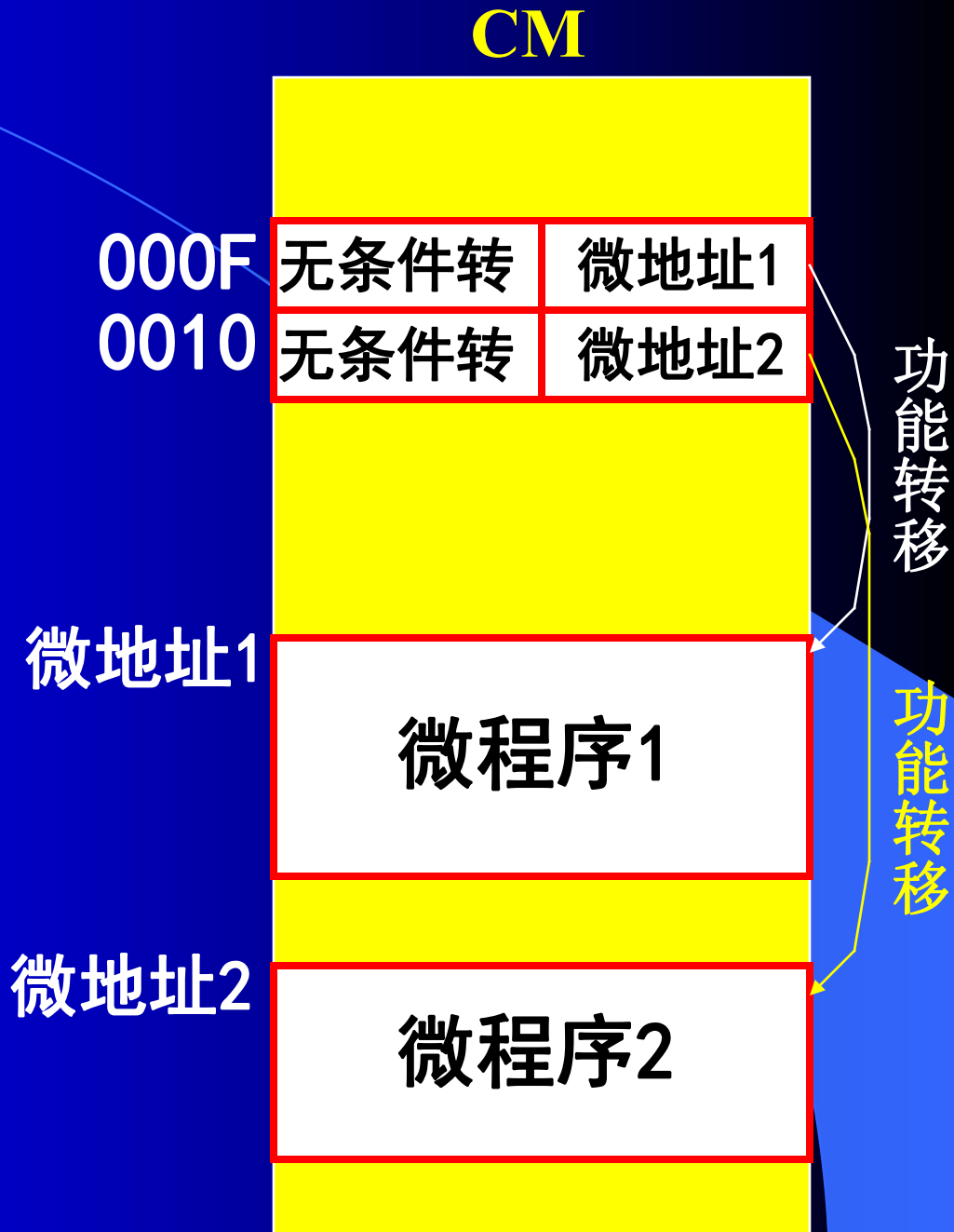
入口地址=000FH

0页

机器指令2

| | |
|---------|-------|
| 10 (8位) | ----- |
|---------|-------|

入口地址=0010H

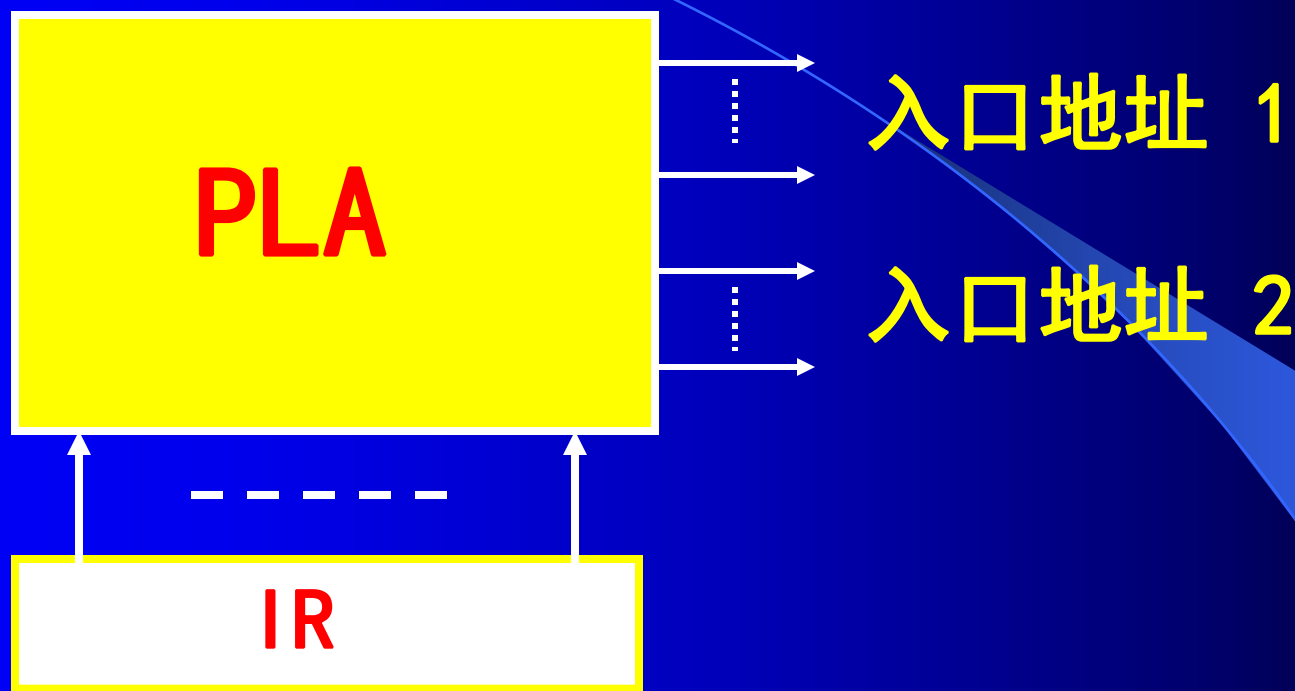


(2) 二级功能转移

各类指令操作码的位置、位数不固定，需两次转换。

{ 分类转： 指令类型标志 → 区分指令类型
功能转： 指令操作码 → 区分操作类型

(3) 用可编程逻辑阵列PLA实现功能转移



2. 后续微地址的形成

(1) 增量方式

以顺序执行为主，辅以各种常规转移方式。

顺序：现行微地址+1。

跳步：现行微地址+2。

无条件转移：现行微指令给出转移微地址。

条件转移：现行微指令给出转移微地址和转移条件。

转微子程序：现行微指令给出微子程序入口。

返回微主程序：现行微指令给出寄存器号。

R

A+1

CM

A

转移条件

D

A+1

(条件不满足)

A+2

B

C

(条件满足)

D

微子程序

R

(2) 断定方式

由直接给定和测试断定相结合形成微地址。

微指令



给定后续微地址
高位部分

指明后续微地址低
位部分的形成方式

例. 微指令



位数可变

2位

16路
分支

微地址10位, 约定:

- A = {
- 01 微地址低4位为操作码, D给定高6位;
 - 10 微地址低3位为机器指令源寻址方式编码, D给定高7位; —— 8路分支
 - 11 微地址低3位为机器指令目的寻址方式编码, D给定高7位。 —— 8路分支

3.3.5 模型机微指令格式

原则：按数据通路操作划分字段

同类操作中互斥的微命令在同一字段

26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0



基本数据通路控制字段

访M、
I/O控
制

辅助
操作

顺序
控制
字段

26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0

| | | | | | | | | | | |
|----|----|----|----------------|---|----|------|---|---|----|----|
| AI | BI | SM | C ₀ | S | Z0 | EMAR | R | W | ST | SC |
|----|----|----|----------------|---|----|------|---|---|----|----|

1. 基本数据通路控制字段

(1) AI: ALU的A输入选择 (3位)

000 无输入

001 R_i → A

010 C → A

011 D → A

100 PC → A

R0~R3、SP、PC

由机器指令指明具体的R

专用于取指/变址中选择PC

26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0

| | | | | | | | | | | |
|----|----|----|----------------|---|----|------|---|---|----|----|
| AI | BI | SM | C ₀ | S | Z0 | EMAR | R | W | ST | SC |
|----|----|----|----------------|---|----|------|---|---|----|----|

(2) BI: ALU的B输入选择 (3位)

000 无输入

001 R_i → B

R0~R3、PSW

由机器指令指明具体的R

010 C → B

011 D → B

100 MDR → B

(3) SM: ALU功能选择 S₃S₂S₁S₀M (直译法)

26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0

| | | | | | | | | | | |
|----|----|----|----------------|---|----|------|---|---|----|----|
| AI | BI | SM | C ₀ | S | ZO | EMAR | R | W | ST | SC |
|----|----|----|----------------|---|----|------|---|---|----|----|

(4) C₀: 初始进位

00: 0→C₀, 01: 1→C₀, 10: PSW₀→C₀

(5) S: 移位器选择 (2位)

00 DM 直传

01 SL 左移

10 SR 右移

11 EX 字节交换

26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0

| | | | | | | | | | | |
|----|----|----|----------------|---|----|------|---|---|----|----|
| AI | BI | SM | C ₀ | S | ZO | EMAR | R | W | ST | SC |
|----|----|----|----------------|---|----|------|---|---|----|----|

(6) ZO: 内总线输出分配 (3位)

000 无输出 → R0~R3、SP、PC、PSW

001 CPR_i

010 CPC

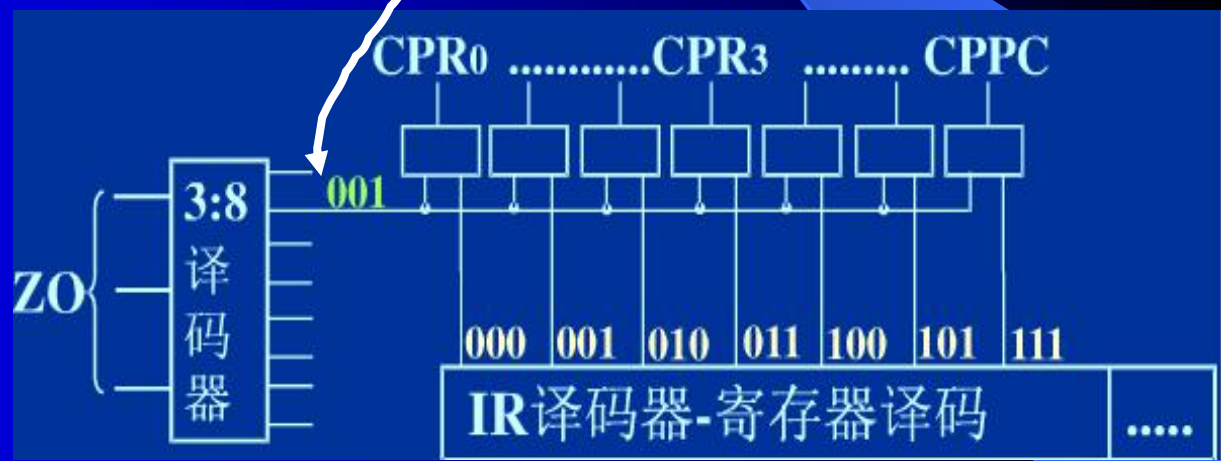
011 CPD

100 CPMAR

101 CPMDR

110 CPPC

取指令、变址寻址专用



26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0

| | | | | | | | | | | |
|----|----|----|----------------|---|----|------|---|---|----|----|
| AI | BI | SM | C ₀ | S | ZO | EMAR | R | W | ST | SC |
|----|----|----|----------------|---|----|------|---|---|----|----|

2. 访存操作 EMAR、R、W (直接控制)

(同时作为SMDR)

3. 辅助操作 (ST字段, 2位)

00 无操作 10 关中断

01 开中断 11 SIR (置入IR)

4. 顺序控制 (SC字段)

指明形成后继微地址的方式

增量方式
断定方式

26 24 23 21 20 16 15 14 13 12 11 9 8 7 6 5 4 3 0

| | | | | | | | | | | |
|----|----|----|----------------|---|----|------|---|---|----|----|
| AI | BI | SM | C ₀ | S | ZO | EMAR | R | W | ST | SC |
|----|----|----|----------------|---|----|------|---|---|----|----|

SC {

0000 顺序执行

0001 无条件转移，微指令高8位提供转移微地址

0010 按操作码OP分支

0011 按OP与DR (R型/非R型) 分支

0100 按J与PC分支

0101 按源寻址方式分支

0110 按目的寻址方式分支

0111 转微子程序，微指令高8位提供微子程序入口

1000 返回微主程序

增量方式

断定方式

增量方式

模型机微指令实例

| | | | | | | | | | | |
|----|----|----|----|---|----|------|---|---|----|----|
| 3 | 3 | 5 | 2 | 2 | 3 | 1 | 1 | 1 | 2 | 4 |
| AI | BI | SM | Co | S | ZO | EMAR | R | W | ST | SC |

取指微指令：

微地址00 $M \rightarrow IR$ ：

000 000 00000 00 00 000 1 1 0 11 0000

微地址01 $PC+1 \rightarrow PC$ ：

100 000 10010 01 00 111 0 0 0 00 0000

微地址02 按操作码分支：

000 000 00000 00 00 000 0 0 0 00 0010

3.3.6 微程序时序安排

同步控制，用统一微指令周期控制各条微指令执行。

二级时序：



3.3.7 微程序控制方式优缺点及应用

1. 优点

- (1) 设计规整，设计效率高；
- (2) 易于修改、扩展指令系统功能；
- (3) 结构规整、简洁，可靠性高；
- (4) 性价比高。

2. 缺点

- (1) 速度慢 
- (2) 执行效率不高 

3. 应用范围

特别适用于系列机
用于速度要求不高、功能较复杂的机器中。