

计算机组成原理

翁睿

哈尔滨工业大学

第 9 章 控制单元的功能

9.1 操作命令的分析

9.2 控制单元的功能

9.1 操作命令的分析

完成一条指令分 4 个工作周期

取指周期

$PC \rightarrow MAR \rightarrow \text{地址线}$

$1 \rightarrow R$

间址周期

$M(MAR) \rightarrow MDR$

执行周期

$MDR \rightarrow IR$

中断周期

$OP(IR) \rightarrow CU$

$(PC) + 1 \rightarrow PC$

9.1 操作命令的分析

完成一条指令分 4 个工作周期

取指周期

$\text{Ad}(\text{IR}) \rightarrow \text{MAR}$

间址周期

$1 \rightarrow \text{R}$

执行周期

$\text{M}(\text{MAR}) \rightarrow \text{MDR}$

中断周期

Optional

$\text{MDR} \rightarrow \text{Ad}(\text{IR})$

9.1 操作命令的分析

完成一条指令分 4 个工作周期

取指周期

间址周期

执行周期

中断周期

1. 非访存指令
2. 访存指令
3. 转移指令

9.1 操作命令的分析

完成一条指令分 4 个工作周期

取指周期

$0 \rightarrow \text{MAR}$

间址周期

$1 \rightarrow \text{W}$

执行周期

$\text{PC} \rightarrow \text{MDR}$

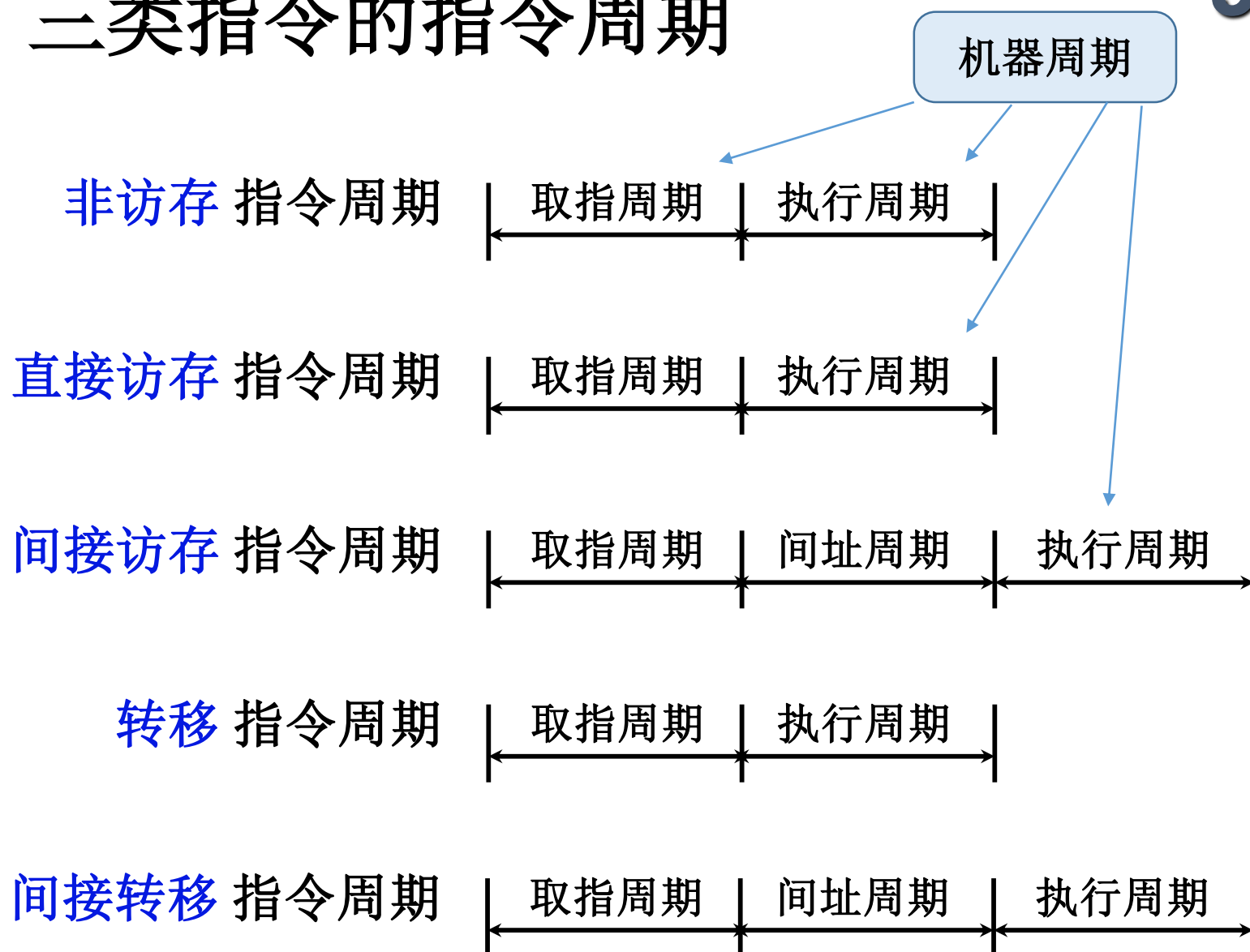
中断周期

$\text{MDR} \rightarrow \text{M}(\text{MAR})$

向量地址 $\rightarrow \text{PC}$

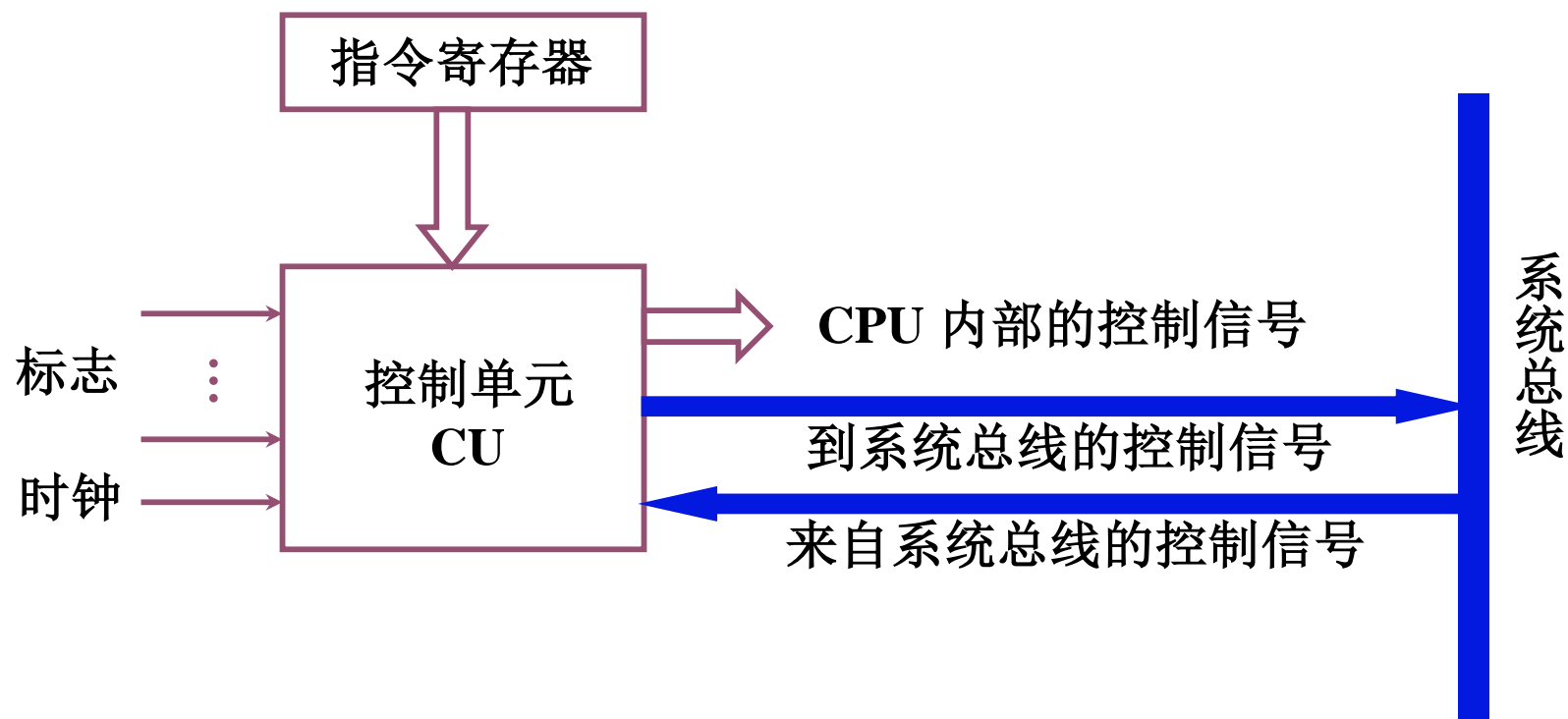
$0 \rightarrow \text{EINT}$ (置“0”)

4. 三类指令的指令周期



9.2 控制单元的功能

一、控制单元的外特性



1. 输入信号

(1) 时钟

CU是时序逻辑电路 受时钟控制

一个时钟脉冲

发一个操作命令或一组需同时执行的操作命令

(2) 指令寄存器 $OP(IR) \rightarrow CU$

控制信号 与操作码有关

(3) 标志

CU依赖CPU当前的状态 受标志控制

(4) 外来信号

如 **INTR** 中断请求
HRQ 总线请求

2. 输出信号

(1) CPU 内的各种控制信号

$R_i \rightarrow R_j$

$(PC) + 1 \rightarrow PC$

ALU +、-、与、或

(2) 送至控制总线的信号

$\overline{\text{MREQ}}$

访存控制信号

$\overline{\text{IO/M}}$

访 IO/ 存储器的控制信号

$\overline{\text{RD}}$

读命令

$\overline{\text{WR}}$

写命令

INTA

中断响应信号

HLDA

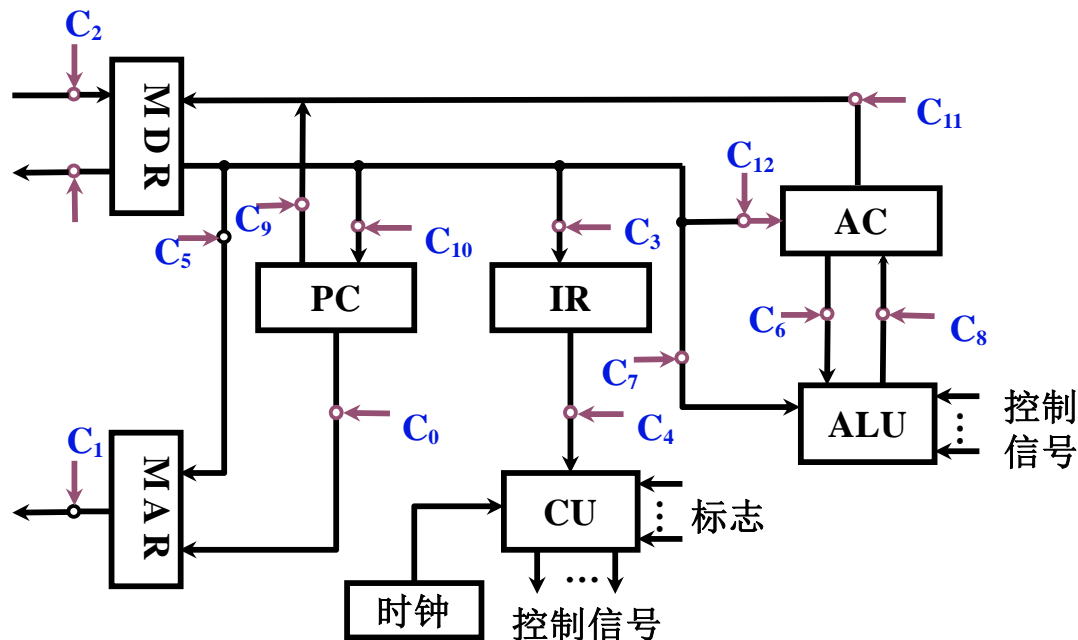
总线响应信号

二、控制信号举例

9.2

1. 不采用 CPU 内部总线的方式

功能部件间有多条独立的数据通路



2. 采用 CPU 内部总线方式

功能部件通过统一的内部总线互联

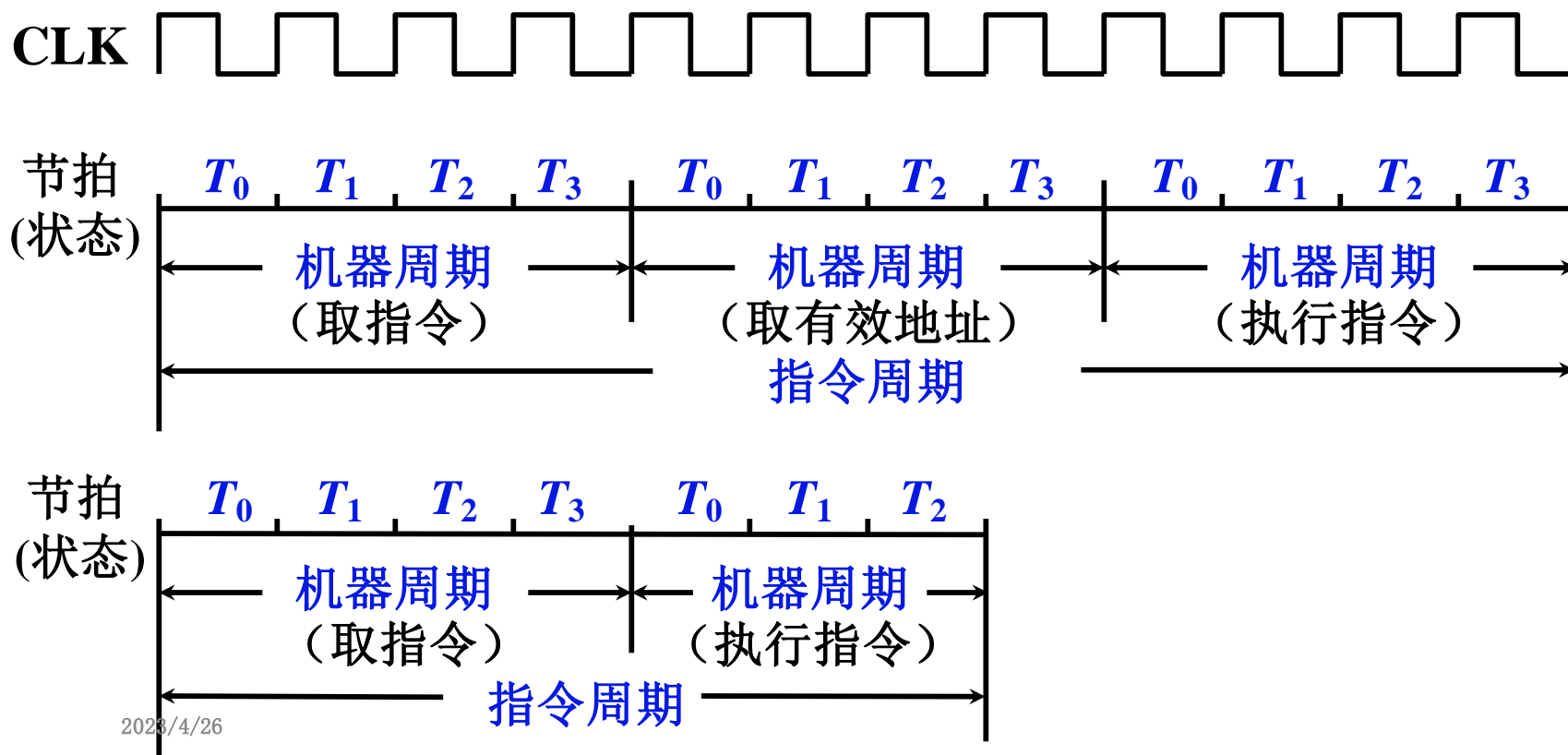
3. 多级时序系统

9.2

机器周期、节拍（状态）组成多级时序系统

一个指令周期包含若干个机器周期

一个机器周期包含若干个时钟周期



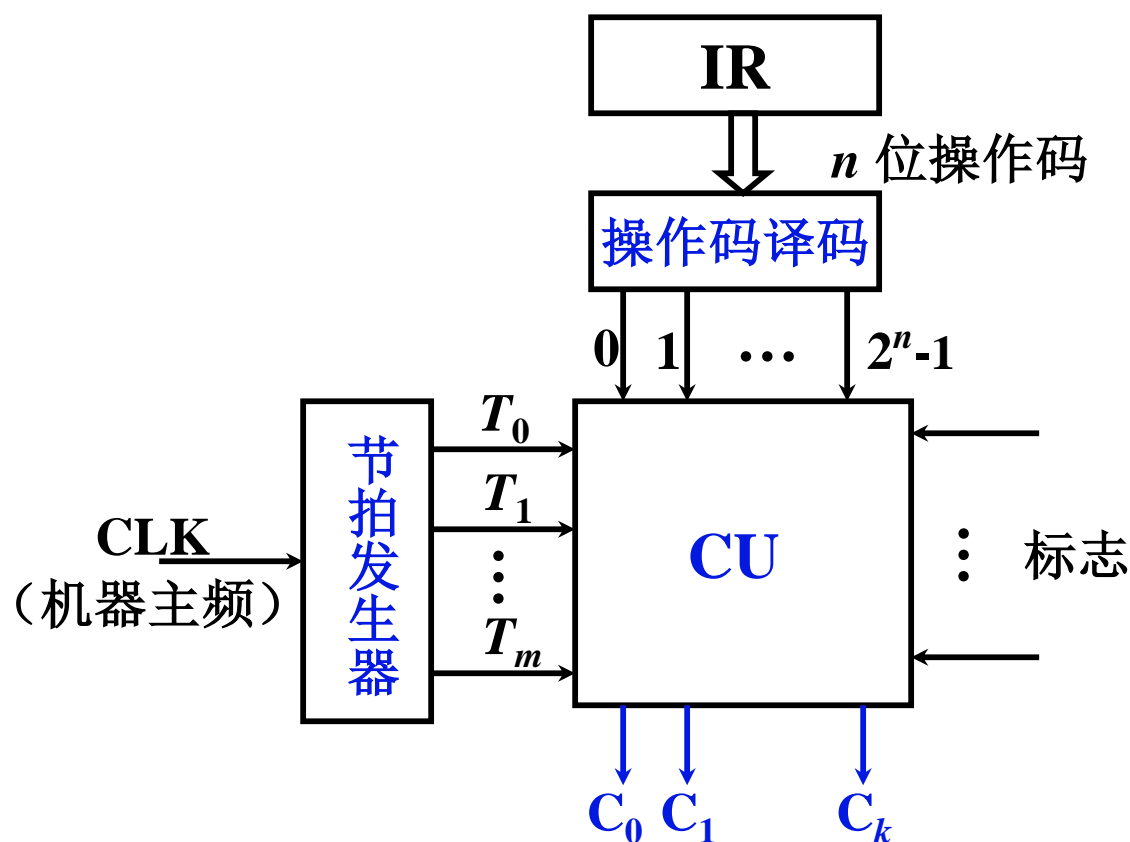
第10章 控制单元的设计

10.1 组合逻辑设计

10.2 微程序设计

10.1 组合逻辑设计

一、组合逻辑控制单元框图



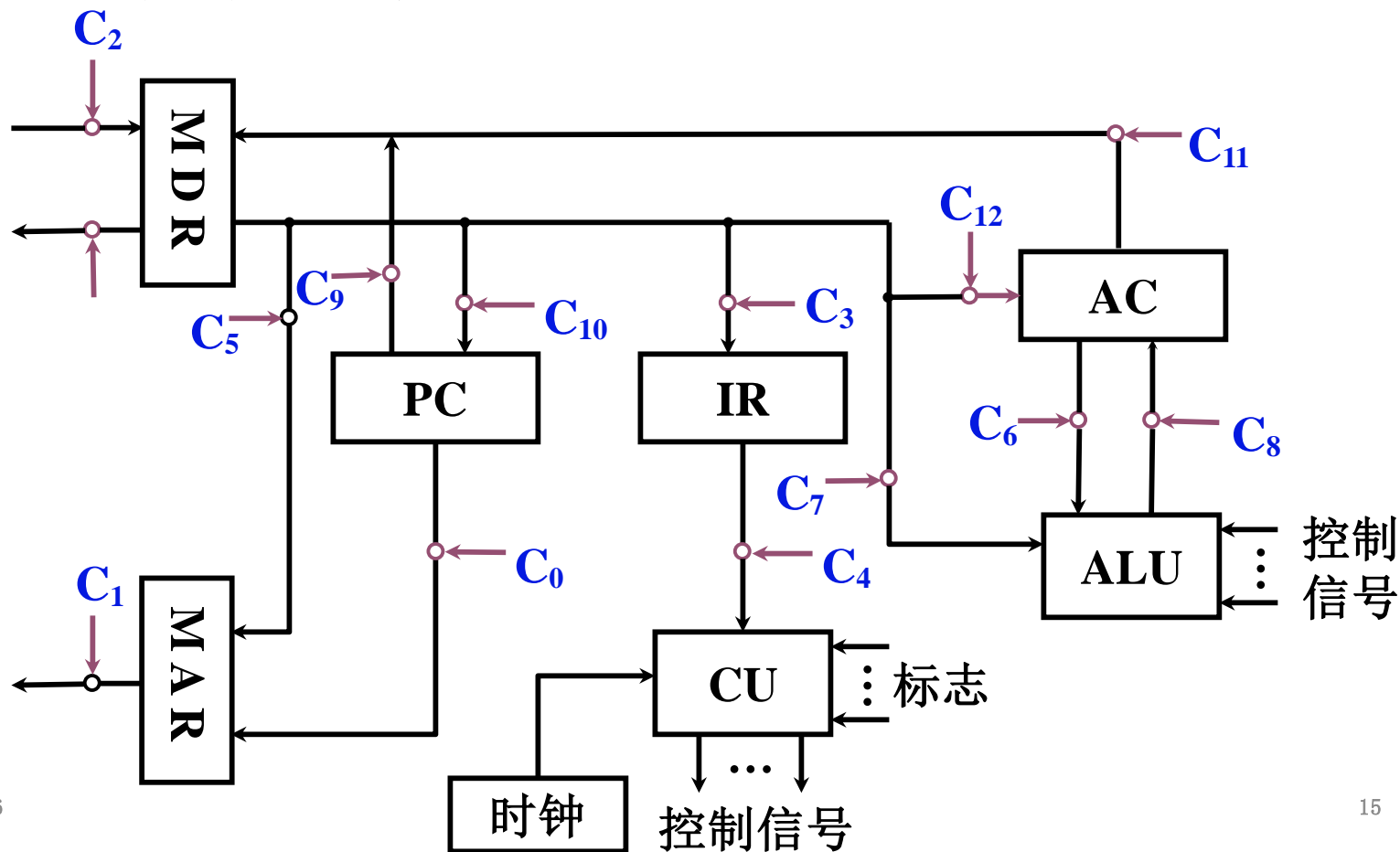
二、微操作的节拍安排

10.1

采用 同步控制方式

一个 机器周期 内有 3 个节拍（时钟周期）

CPU 内部结构采用非总线方式



1. 安排微操作时序的原则

原则一 微操作的 先后顺序不得 随意 更改

原则二 被控对象不同 的微操作

尽量安排在一个节拍 内完成

原则三 占用 时间较短 的微操作

尽量 安排在一个节拍 内完成

并允许有先后顺序

2. 取指周期 微操作的 节拍安排

T_0 $PC \longrightarrow MAR$

原则二

$1 \longrightarrow R$

T_1 $M(MAR) \longrightarrow MDR$

原则二

$(PC) + 1 \longrightarrow PC$

T_2 $MDR \longrightarrow IR$

原则三

$OP(IR) \longrightarrow ID$

3. 间址周期 微操作的 节拍安排

T_0 $Ad(IR) \longrightarrow MAR$

$1 \longrightarrow R$

T_1 $M(MAR) \longrightarrow MDR$

T_2 $MDR \longrightarrow Ad(IR)$

4. 执行周期 微操作的 节拍安排

10.1

① CLA T_0

T_1

T_2 $0 \longrightarrow AC$

② COM T_0

T_1

T_2 $\overline{AC} \longrightarrow AC$

③ SHR T_0

T_1

T_2 $L(AC) \longrightarrow R(AC)$

$AC_0 \longrightarrow AC_0$

④ CSL T_0
 T_1
 T_2 $R(AC) \longrightarrow L(AC)$ $AC_0 \longrightarrow AC_n$

⑤ STP T_0
 T_1
 T_2 $0 \longrightarrow G$

⑥ ADD X T_0 $Ad(IR) \longrightarrow MAR$ $1 \longrightarrow R$
 T_1 $M(MAR) \longrightarrow MDR$
 T_2 $(AC) + (MDR) \longrightarrow AC$

⑦ STA X T_0 $Ad(IR) \longrightarrow MAR$ $1 \longrightarrow W$
 T_1 $AC \longrightarrow MDR$
 T_2 $MDR \longrightarrow M(MAR)$

⑧ LDA X T_0 Ad (IR) \longrightarrow MAR 1 \longrightarrow R

T_1 M (MAR) \longrightarrow MDR

T_2 MDR \longrightarrow AC

⑨ JMP X T_0

T_1

T_2 Ad (IR) \longrightarrow PC

⑩ BAN X T_0

T_1

T_2 $A_0 \cdot \text{Ad (IR)} + \bar{A}_0 \cdot \text{PC} \longrightarrow \text{PC}$

5. 中断周期 微操作的 节拍安排

10.1

硬件关中断

T_0 $0 \longrightarrow \text{MAR}$ $1 \longrightarrow \text{W}$ $0 \longrightarrow \text{EINT}$

T_1 $\text{PC} \longrightarrow \text{MDR}$

T_2 $\text{MDR} \longrightarrow \text{M}(\text{MAR})$ 向量地址 $\longrightarrow \text{PC}$

中断隐指令完成

三、组合逻辑设计步骤

10.1

1. 列出操作时间表

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	ADD	STA	LDA	JMP
FE 取指	T_0		$PC \rightarrow MAR$						
			$1 \rightarrow R$						
	T_1		$M(MAR) \rightarrow MDR$						
			$(PC) + 1 \rightarrow PC$						
	T_2		$MDR \rightarrow IR$						
			$OP(IR) \rightarrow ID$						
		I	$1 \rightarrow IND$						
		\bar{I}	$1 \rightarrow EX$						

间址特征

三、组合逻辑设计步骤

10.1

1. 列出操作时间表

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	ADD	STA	LDA	JMP
IND 间址	T_0		Ad (IR) \rightarrow MAR						
			1 \rightarrow R						
	T_1		M(MAR) \rightarrow MDR						
	T_2		MDR \rightarrow Ad (IR)						
		$\overline{\text{IND}}$	1 \rightarrow EX						

间址周期标志

三、组合逻辑设计步骤

10.1

1. 列出操作时间表

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	ADD	STA	LDA	JMP
EX 执行	T_0		$Ad(IR) \rightarrow MAR$						
			$1 \rightarrow R$						
			$1 \rightarrow W$						
	T_1		$M(MAR) \rightarrow MDR$						
			$AC \rightarrow MDR$						
	T_2		$(AC) + (MDR) \rightarrow AC$						
			$MDR \rightarrow M(MAR)$						
			$MDR \rightarrow AC$						
			$0 \rightarrow AC$						

三、组合逻辑设计步骤

10.1

1. 列出操作时间表

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	ADD	STA	LDA	JMP
FE 取指	T_0		$PC \rightarrow MAR$	1	1	1	1	1	1
			$1 \rightarrow R$	1	1	1	1	1	1
	T_1		$M(MAR) \rightarrow MDR$	1	1	1	1	1	1
			$(PC) + 1 \rightarrow PC$	1	1	1	1	1	1
	T_2		$MDR \rightarrow IR$	1	1	1	1	1	1
			$OP(IR) \rightarrow ID$	1	1	1	1	1	1
		I	$1 \rightarrow IND$			1	1	1	1
		\bar{I}	$1 \rightarrow EX$	1	1	1	1	1	1

三、组合逻辑设计步骤

10.1

1. 列出操作时间表

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	ADD	STA	LDA	JMP
IND 间址	T_0		Ad (IR) \rightarrow MAR			1	1	1	1
			1 \rightarrow R			1	1	1	1
	T_1		M(MAR) \rightarrow MDR			1	1	1	1
	T_2		MDR \rightarrow Ad (IR)			1	1	1	1
		$\overline{\text{IND}}$	1 \rightarrow EX			1	1	1	1

三、组合逻辑设计步骤

10.1

1. 列出操作时间表

工作周期标记	节拍	状态条件	微操作命令信号	CLA	COM	ADD	STA	LDA	JMP
EX 执行	T_0		Ad (IR) \rightarrow MAR			1	1	1	
			1 \rightarrow R			1		1	
			1 \rightarrow W				1		
	T_1		M(MAR) \rightarrow MDR			1		1	
			AC \rightarrow MDR				1		
	T_2		(AC)+(MDR) \rightarrow AC			1			
			MDR \rightarrow M(MAR)				1		
			MDR \rightarrow AC					1	
			0 \rightarrow AC	1					

2. 写出微操作命令的最简表达式

10.1

$M(MAR) \longrightarrow MDR$

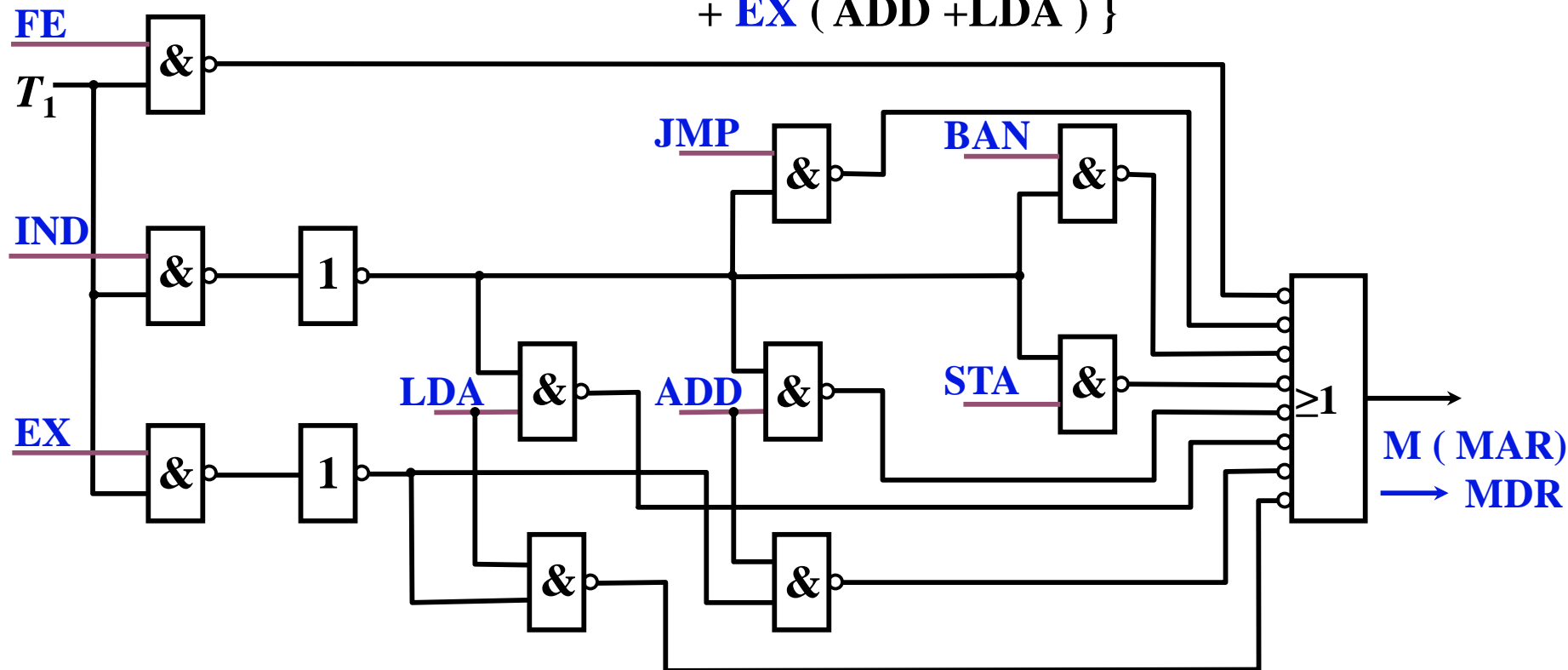
$$= FE \cdot T_1 + IND \cdot T_1 (ADD + STA + LDA + JMP + BAN) \\ + EX \cdot T_1 (ADD + LDA)$$

$$= T_1 \{ FE + IND (ADD + STA + LDA + JMP + BAN) \\ + EX (ADD + LDA) \}$$

3. 画出逻辑图

10.1

$$M(MAR) \rightarrow MDR = T_1 \{ FE + IND(ADD + STA + LDA + JMP + BAN) + EX(ADD + LDA) \}$$



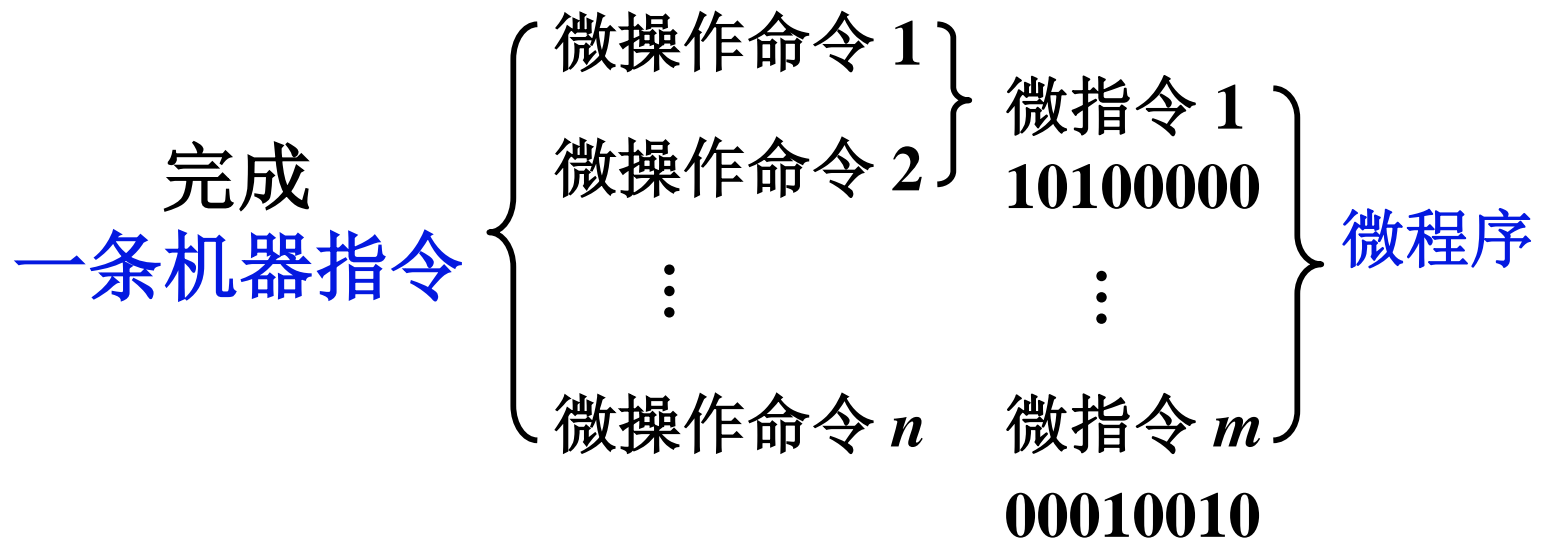
特点

- 思路清晰，简单明了
- 庞杂，调试困难，修改困难
- 速度快 (RISC)

10.2 微程序设计

一、微程序设计思想的产生

1951 英国剑桥大学教授 Wilkes



一条机器指令对应一段微程序

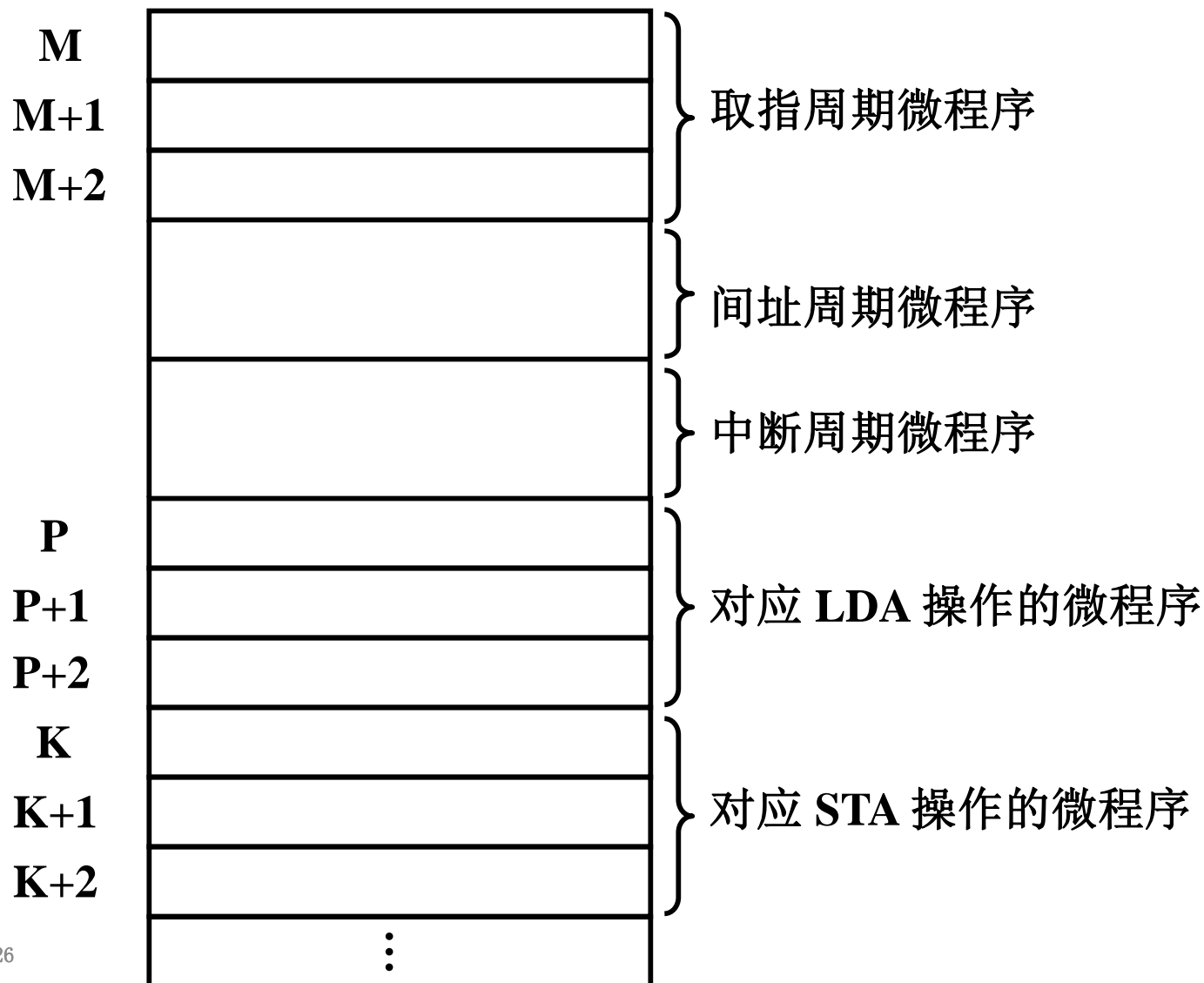
存入 ROM

存储逻辑

二、微程序控制单元框图及工作原理

10.2

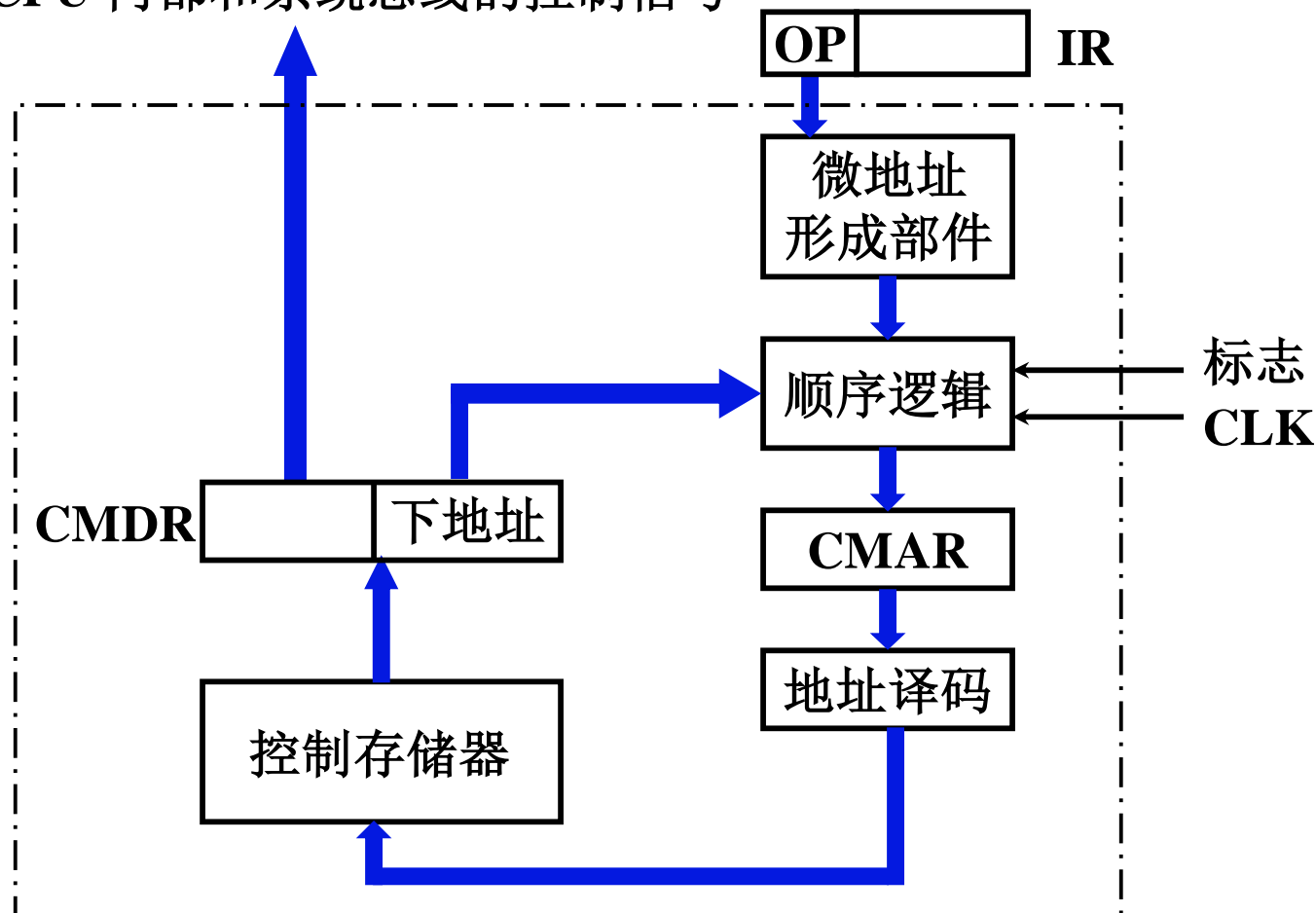
1. 机器指令对应的微程序



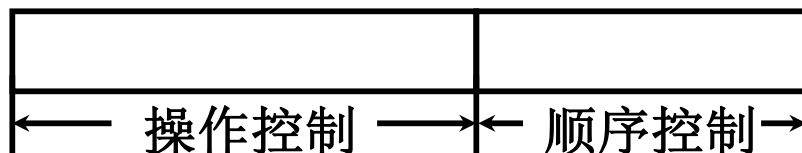
2. 微程序控制单元的基本框图

10.2

至 CPU 内部和系统总线的控制信号

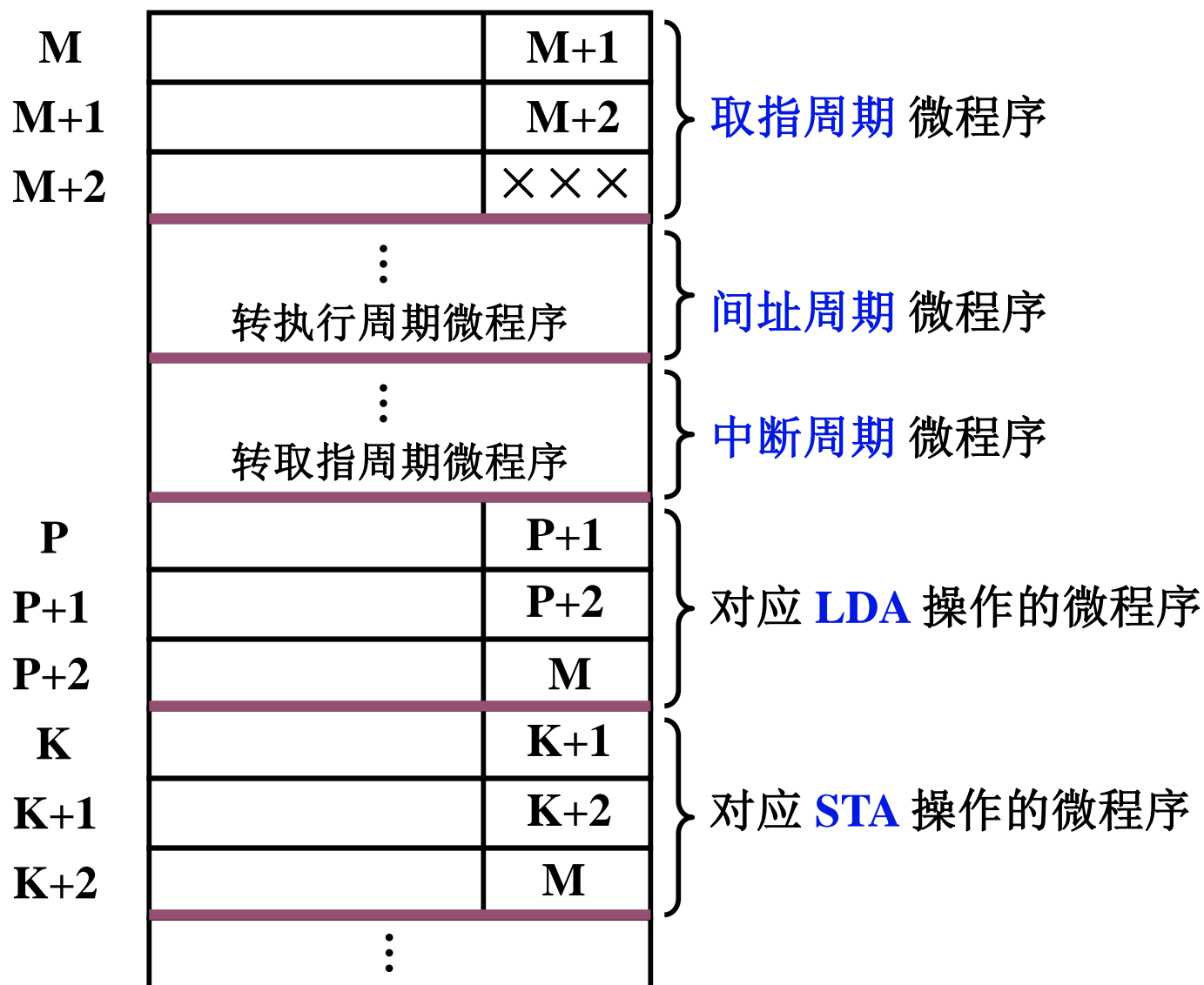


微指令基本格式



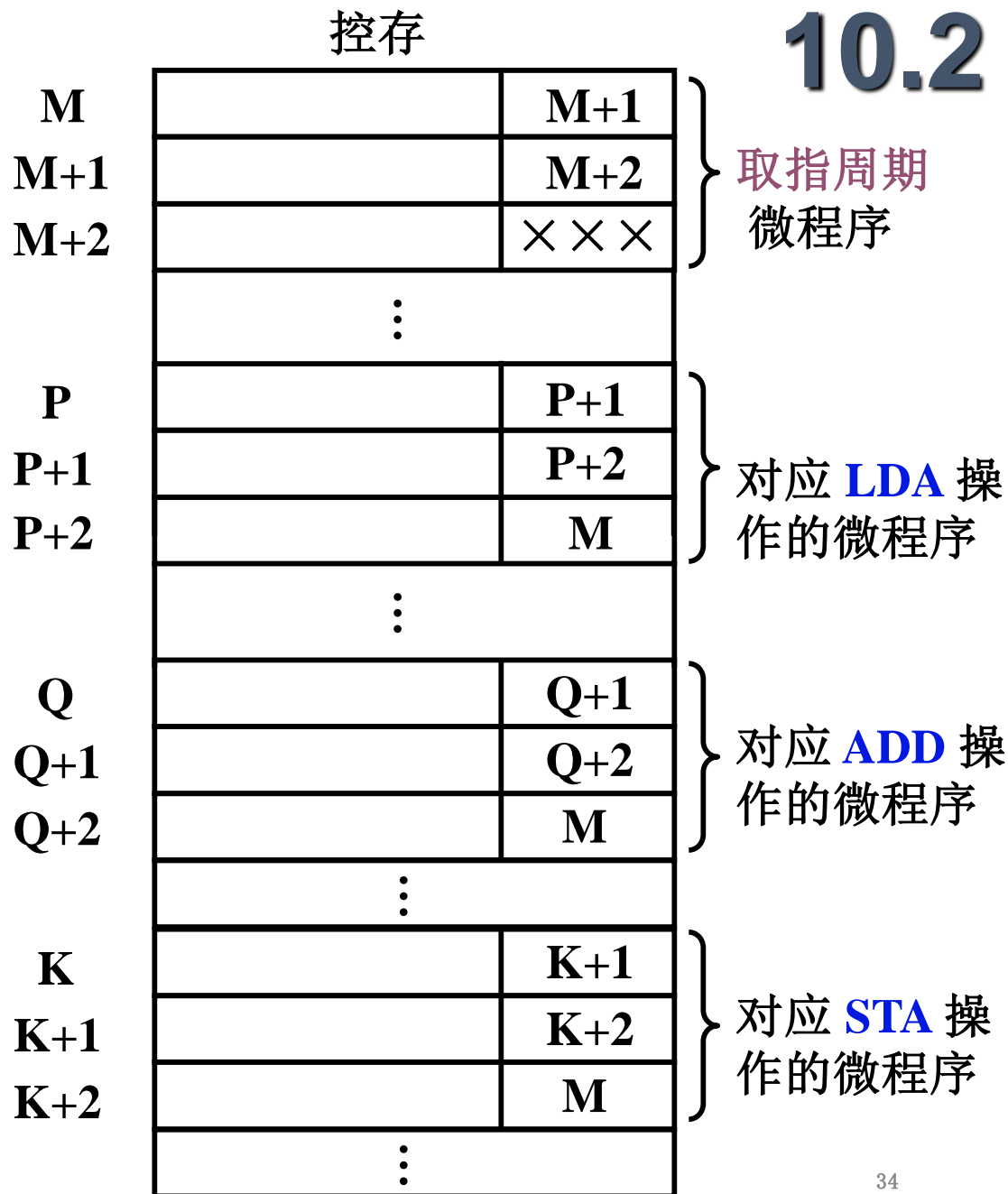
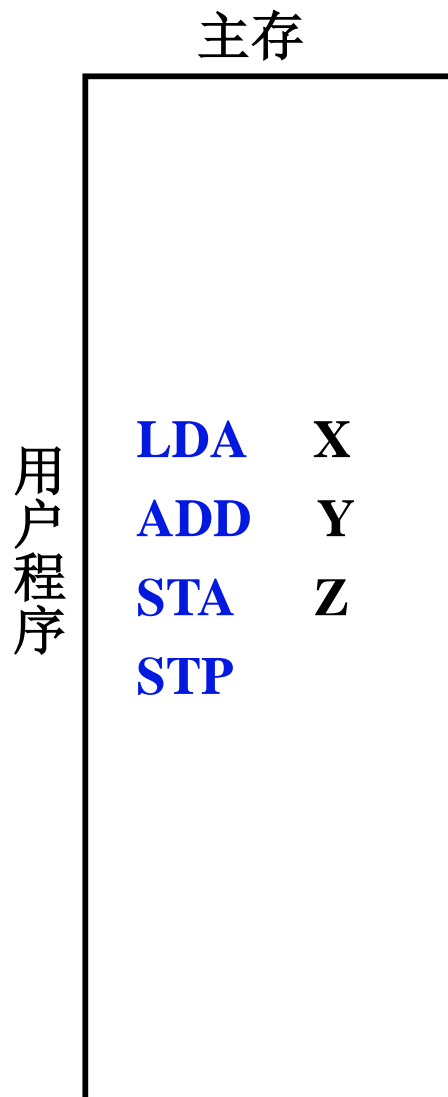
二、微程序控制单元框图及工作原理

10.2



3. 工作原理

10.2



3. 工作原理

10.2

(1) 取指阶段 执行取指微程序

$M \rightarrow CMAR$

$CM(CMAR) \rightarrow CMDR$

由 CMDR 发命令

形成下条微指令地址 $M + 1$

$Ad(CMDR) \rightarrow CMAR$

$CM(CMAR) \rightarrow CMDR$

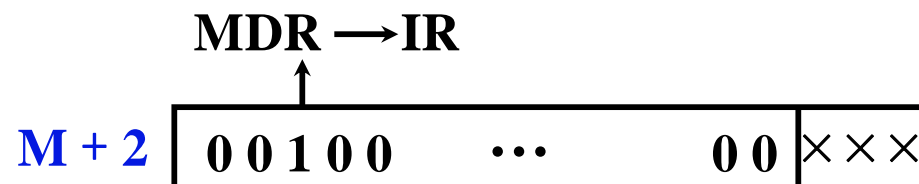
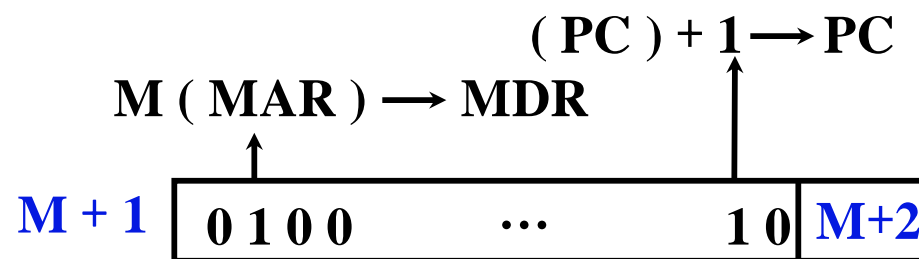
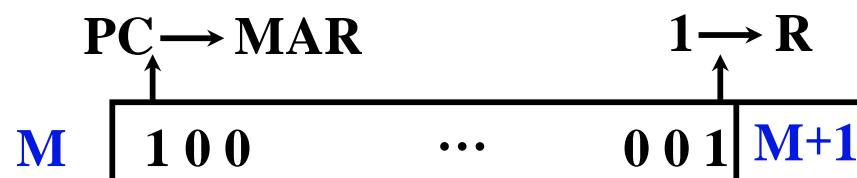
由 CMDR 发命令

形成下条微指令地址 $M + 2$

$Ad(CMDR) \rightarrow CMAR$

$CM(CMAR) \rightarrow CMDR$

由 CMDR 发命令



(2) 执行阶段 执行 LDA 微程序

10.2

$OP(IR) \longrightarrow \text{微地址形成部件} \longrightarrow \text{CMAR} \quad (P \longrightarrow \text{CMAR})$

$CM(\text{CMAR}) \longrightarrow \text{CMDR}$

由 CMDR 发命令

形成下条微指令地址

$CM(\text{CMAR}) \longrightarrow \text{CMDR}$

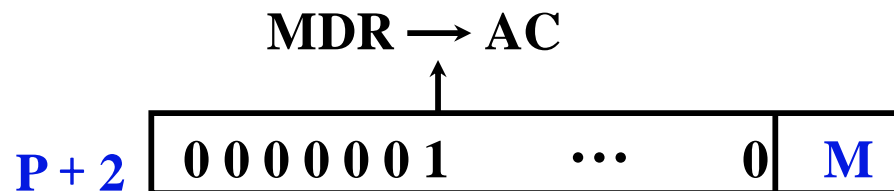
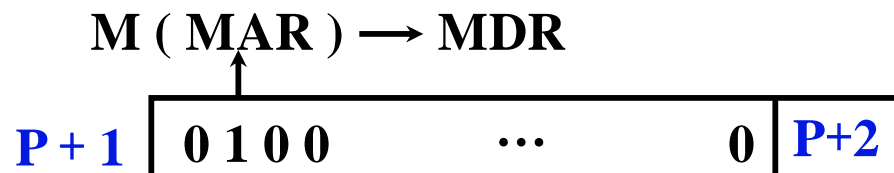
由 CMDR 发命令

形成下条微指令地址

$CM(\text{CMAR}) \longrightarrow \text{CMDR}$

由 CMDR 发命令

形成下条微指令地址



$(M \longrightarrow \text{CMAR})$

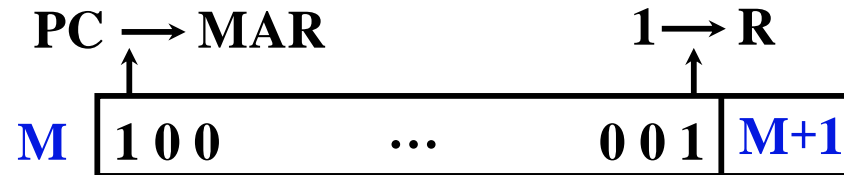
(3) 取指阶段 执行取指微程序

$M \rightarrow \text{CMAR}$

$\text{CM}(\text{CMAR}) \rightarrow \text{CMDR}$

由 CMDR 发命令

⋮



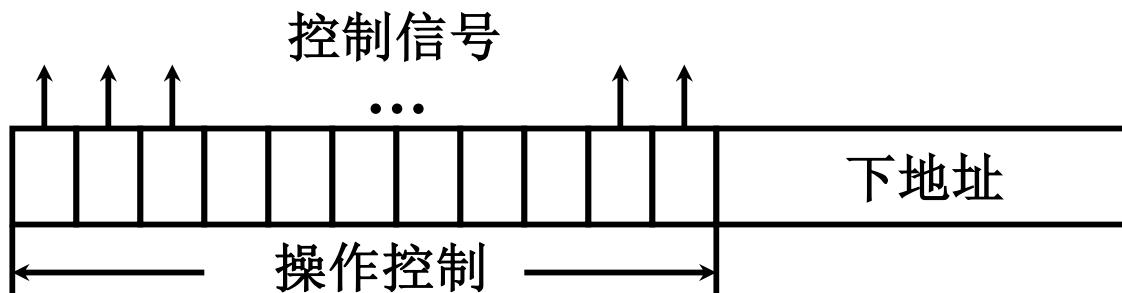
全部微指令存在 CM 中，程序执行过程中 只需读出

- 关键
- 微指令的 操作控制字段如何形成微操作命令
 - 微指令的 后续地址如何形成

三、微指令的编码方式（控制方式）

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

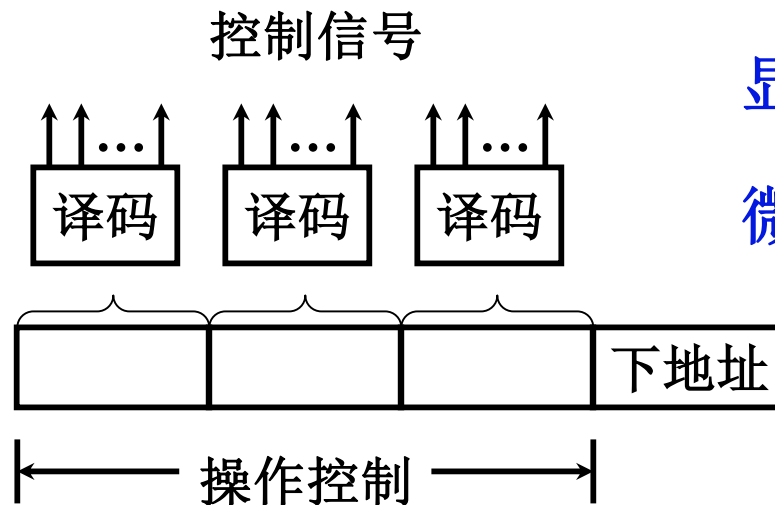


速度最快

某位为 “1” 表示该控制信号有效

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



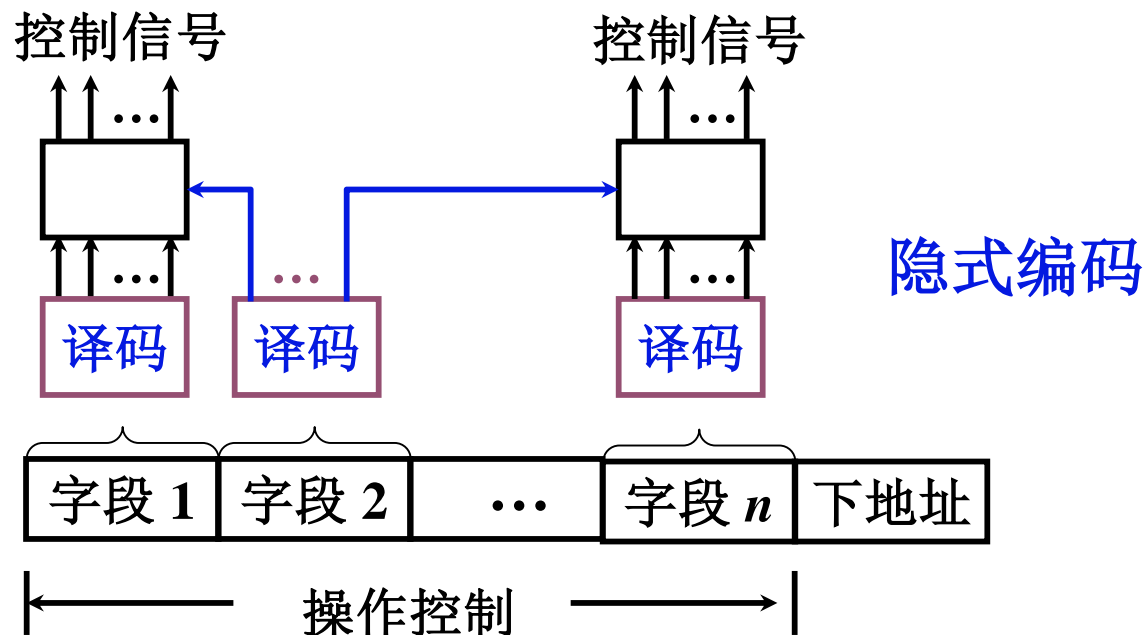
显式编码

微程序执行速度较慢

每个字段中的命令是 **互斥** 的

缩短 了微指令 **字长**，**增加** 了译码 **时间**

3. 字段间接编码方式



4. 混合编码

直接编码和字段编码（直接和间接）混合使用

5. 其他

四、微指令序列地址的形成

1. 微指令的 **下地址字段** 指出
2. 根据机器指令的 **操作码** 形成
3. **增量计数器**

$$(\text{CMAR}) + 1 \longrightarrow \text{CMAR}$$

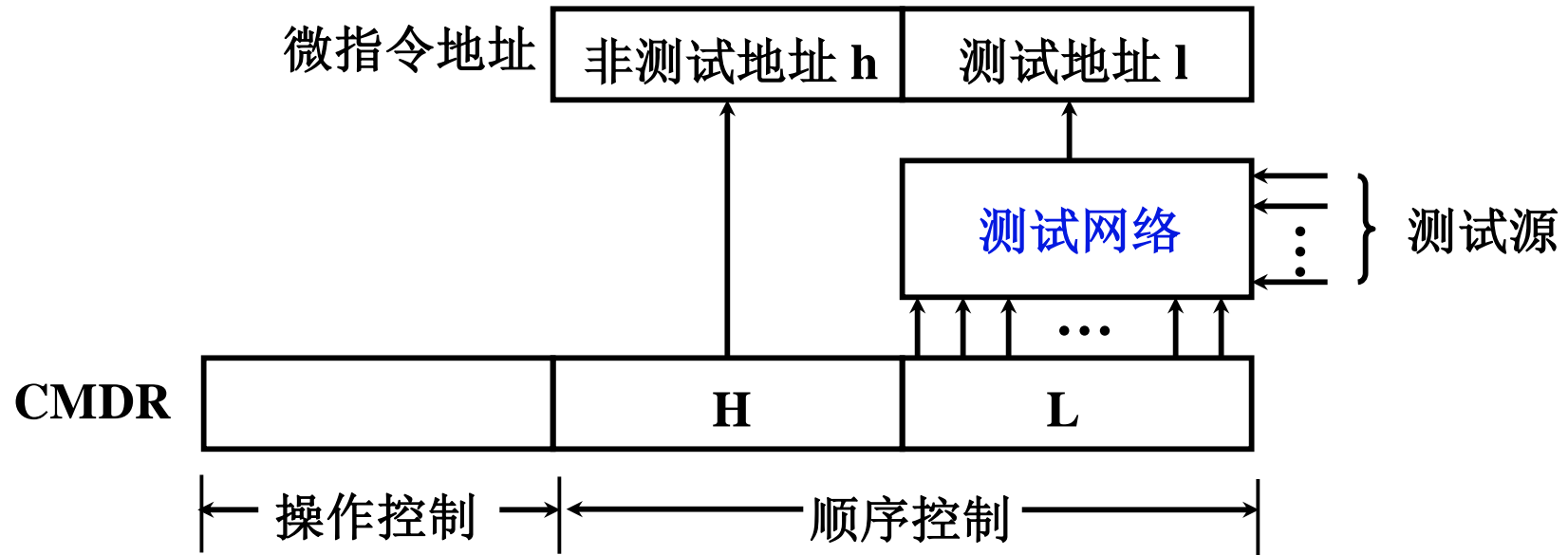
4. **分支转移**

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

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



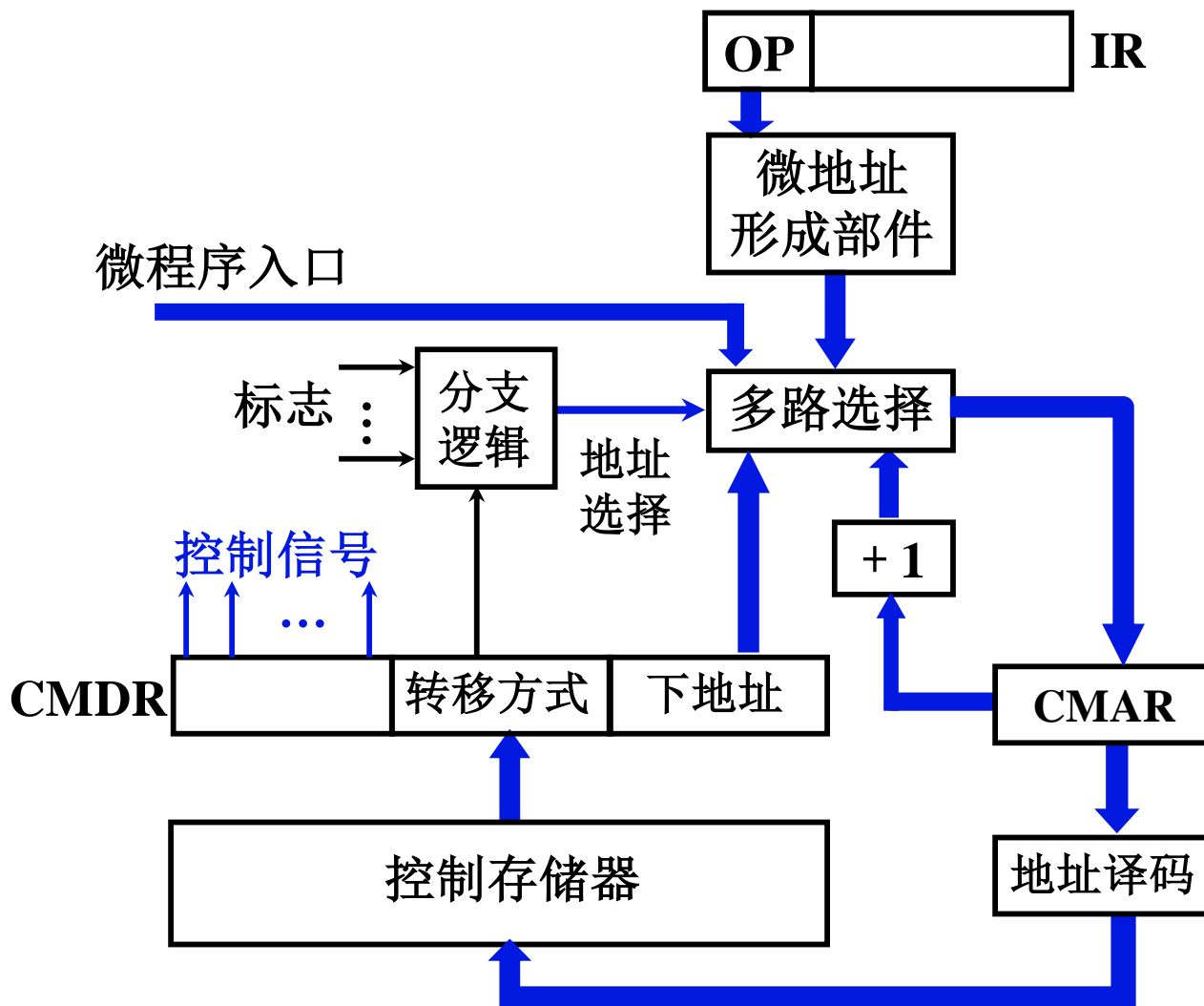
6. 由硬件产生微程序入口地址

第一条微指令地址 由专门 硬件 产生

中断周期 由 硬件 产生 中断周期微程序首地址

7. 后续微指令地址形成方式原理图

10.2



五、微指令格式

1. 水平型微指令

一次能定义并执行多个并行操作

如 直接编码、字段直接编码、字段间接编码、
直接和字段混合编码

2. 垂直型微指令

类似机器指令操作码 的方式

由微操作码字段规定微指令的功能

3. 两种微指令格式的比较

- (1) 水平型微指令比垂直型微指令 并行操作能力强，
灵活性强
- (2) 水平型微指令执行一条机器指令所要的
微指令 数目少，速度快
- (3) 水平型微指令 用较短的微程序结构换取较长的
微指令结构
- (4) 水平型微指令与机器指令 差别大

六、静态微程序设计和动态微程序设计

10.2

静态 微程序无须改变，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**

七、毫微程序设计

1. 毫微程序设计的基本概念

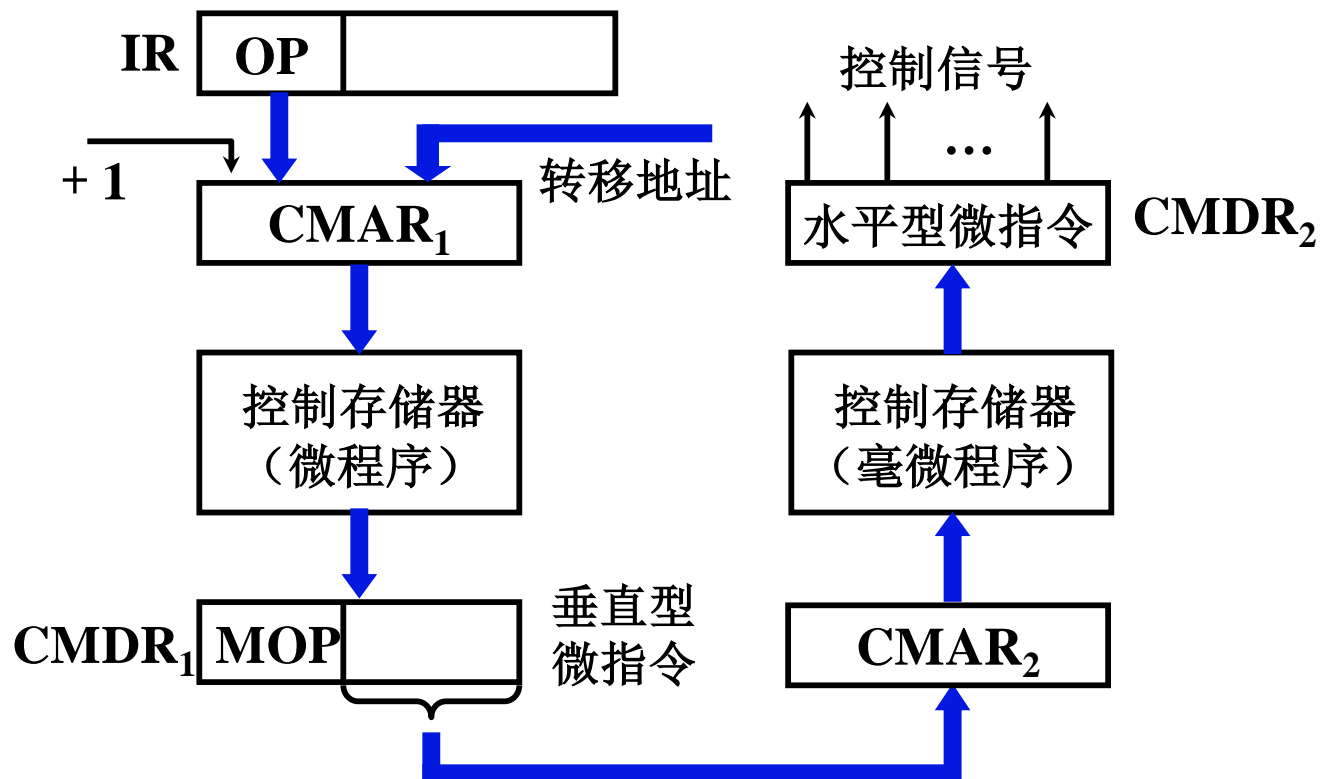
微程序设计 用 **微程序** 解释机器指令

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

2. 毫微程序控制存储器的基本组成

10.2



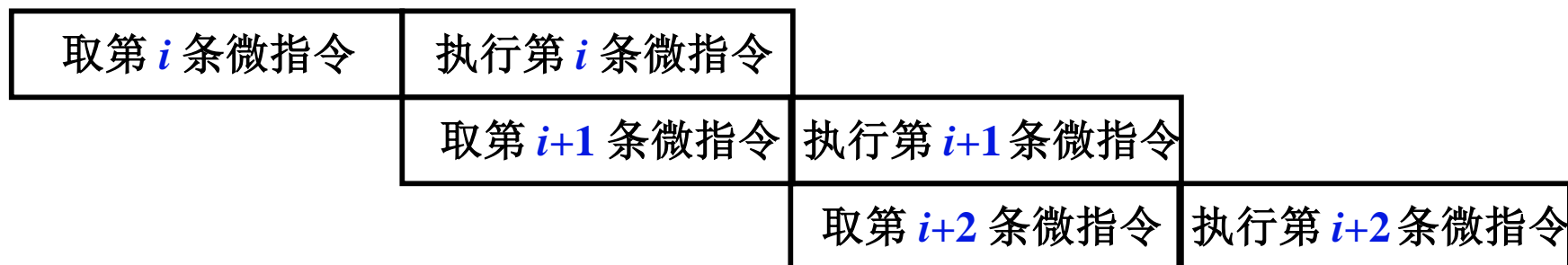
八、串行微程序控制和并行微程序控制

10.2

串行 微程序控制



并行 微程序控制



九、微程序设计举例

1. 写出对应机器指令的微操作及节拍安排

假设 CPU 结构与组合逻辑相同

(1) 取指阶段微操作分析 3 条微指令

T_0 $PC \rightarrow MAR$ $1 \rightarrow R$

T_1 $M(MAR) \rightarrow MDR$ $(PC) + 1 \rightarrow PC$

T_2 $MDR \rightarrow IR$ $OP(IR) \rightarrow$ 微地址形成部件

需考虑如何安排这条微指令？

则取指操作需 3 条微指令

$Ad(CMDR) \rightarrow CMAR$

$OP(IR) \rightarrow$ 微地址形成部件 $\rightarrow CMAR$

(2) 取指阶段的微操作及节拍安排

考虑到需要 形成后续微指令的地址

T_0 $PC \longrightarrow MAR$ $1 \longrightarrow R$

T_1 $Ad (CMDR) \longrightarrow CMAR$

T_2 $M (MAR) \longrightarrow MDR$ $(PC) + 1 \longrightarrow PC$

T_3 $Ad (CMDR) \longrightarrow CMAR$

T_4 $MDR \longrightarrow IR$ $OP (IR) \longrightarrow$ 微地址形成部件

T_5 $OP (IR) \longrightarrow$ 微地址形成部件 $\longrightarrow CMAR$

(3) 执行阶段的微操作及节拍安排

10.2

考虑到需形成后续微指令的地址

取指微程序的入口地址 M
由微指令下地址字段指出

- 非访存指令

- ① CLA 指令

$T_0 \quad 0 \longrightarrow AC$

$T_1 \quad \text{Ad (CMDR)} \longrightarrow \text{CMAR}$

- ② COM 指令

$T_0 \quad \overline{AC} \longrightarrow AC$

$T_1 \quad \text{Ad (CMDR)} \longrightarrow \text{CMAR}$

③ SHR 指令

$$T_0 \quad L(AC) \longrightarrow R(AC) \quad AC_0 \longrightarrow AC_0$$

$$T_1 \quad \text{Ad}(\text{CMDR}) \longrightarrow \text{CMAR}$$

④ CSL 指令

$$T_0 \quad R(AC) \longrightarrow L(AC) \quad AC_0 \longrightarrow AC_n$$

$$T_1 \quad \text{Ad}(\text{CMDR}) \longrightarrow \text{CMAR}$$

⑤ STP 指令

$$T_0 \quad 0 \longrightarrow G$$

$$T_1 \quad \text{Ad}(\text{CMDR}) \longrightarrow \text{CMAR}$$

- 访存指令

- ⑥ ADD 指令

T_0 $\text{Ad (IR)} \longrightarrow \text{MAR}$ $1 \longrightarrow \text{R}$

T_1 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

T_2 $\text{M (MAR)} \longrightarrow \text{MDR}$

T_3 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

T_4 $(\text{AC}) + (\text{MDR}) \longrightarrow \text{AC}$

T_5 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

- ⑦ STA 指令

T_0 $\text{Ad (IR)} \longrightarrow \text{MAR}$ $1 \longrightarrow \text{W}$

T_1 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

T_2 $\text{AC} \longrightarrow \text{MDR}$

T_3 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

T_4 $\text{MDR} \longrightarrow \text{M (MAR)}$

T_5 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

⑧ LDA 指令

10.2

T_0 $\text{Ad (IR)} \longrightarrow \text{MAR}$ $1 \longrightarrow \text{R}$

T_1 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

T_2 $\text{M (MAR)} \longrightarrow \text{MDR}$

T_3 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

T_4 $\text{MDR} \longrightarrow \text{AC}$

T_5 $\text{Ad (CMDR)} \longrightarrow \text{CMAR}$

- 转移类指令

- ⑨ JMP 指令

$$T_0 \quad \text{Ad (IR)} \longrightarrow \text{PC}$$

$$T_1 \quad \text{Ad (CMDR)} \longrightarrow \text{CMAR}$$

- ⑩ BAN 指令

$$T_0 \quad \text{A}_0 \cdot \text{Ad (IR)} + \overline{\text{A}_0} \cdot (\text{PC}) \longrightarrow \text{PC}$$

$$T_1 \quad \text{Ad (CMDR)} \longrightarrow \text{CMAR}$$

全部微操作 20个

微指令 38条

2. 确定微指令格式

(1) 微指令的编码方式

采用直接控制

(2) 后续微指令的地址形成方式

由机器指令的操作码通过微地址形成部件形成

由微指令的下地址字段直接给出

(3) 微指令字长

由 20 个微操作

确定 操作控制字段 最少 20 位

由 38 条微指令

确定微指令的 下地址字段 为 6 位

微指令字长 可取 $20 + 6 = 26$ 位

(4) 微指令字长的确定

10.2

38 条微指令中有 19 条

是关于后续微指令地址 \longrightarrow CMAR

其中 $\left\{ \begin{array}{ll} 1 \text{ 条} & \text{OP (IR) } \longrightarrow \text{微地址形成部件} \longrightarrow \text{CMAR} \\ 18 \text{ 条} & \text{Ad (CMDR) } \longrightarrow \text{CMAR} \end{array} \right.$

若用 Ad (CMDR) 直接送控存地址线

则 省去了输至 CMAR 的时间, 省去了 CMAR

同理 $\text{OP (IR) } \longrightarrow \text{微地址形成部件} \longrightarrow \text{控存地址线}$

可省去 19 条微指令, 2 个微操作

$$38 - 19 = 19$$

$$20 - 2 = 18$$

下地址字段最少取 5 位

操作控制字段最少取 18 位

3. 编写微指令码点

微程序 名称	微指令 地址 (八进制)	微指令（二进制代码）														
		操作控制字段									下地址字段					
取指		0	1	2	3	4	...	10	...	23	24	25	26	27	28	29
	00	1	1								0	0	0	0	0	1
	01			1	1						0	0	0	0	1	0
	02					1					×	×	×	×	×	×
CLA	03										0	0	0	0	0	0
COM	04										0	0	0	0	0	0
ADD	10		1					1			0	0	1	0	0	1
	11			1							0	0	1	0	1	0
	12										0	0	0	0	0	0
LDA	16		1					1			0	0	1	1	1	1
	17			1							0	1	0	0	0	0
	20										0	0	0	0	0	0

2023/4/26

59

数据通路 DataPath

- 数据通路-----执行部件间传送信息的路径 (数据流)
 - ◆通路的建立由控制信号控制，受时钟驱动 （控制流）
 - ◆不同指令、同一指令在执行的不同阶段的数据通路不同
 - ◆分类：共享通路（总线）、专用通路
 - 指令执行流程、执行效率
 - 微操作控制信号的时序安排

数据通路分类

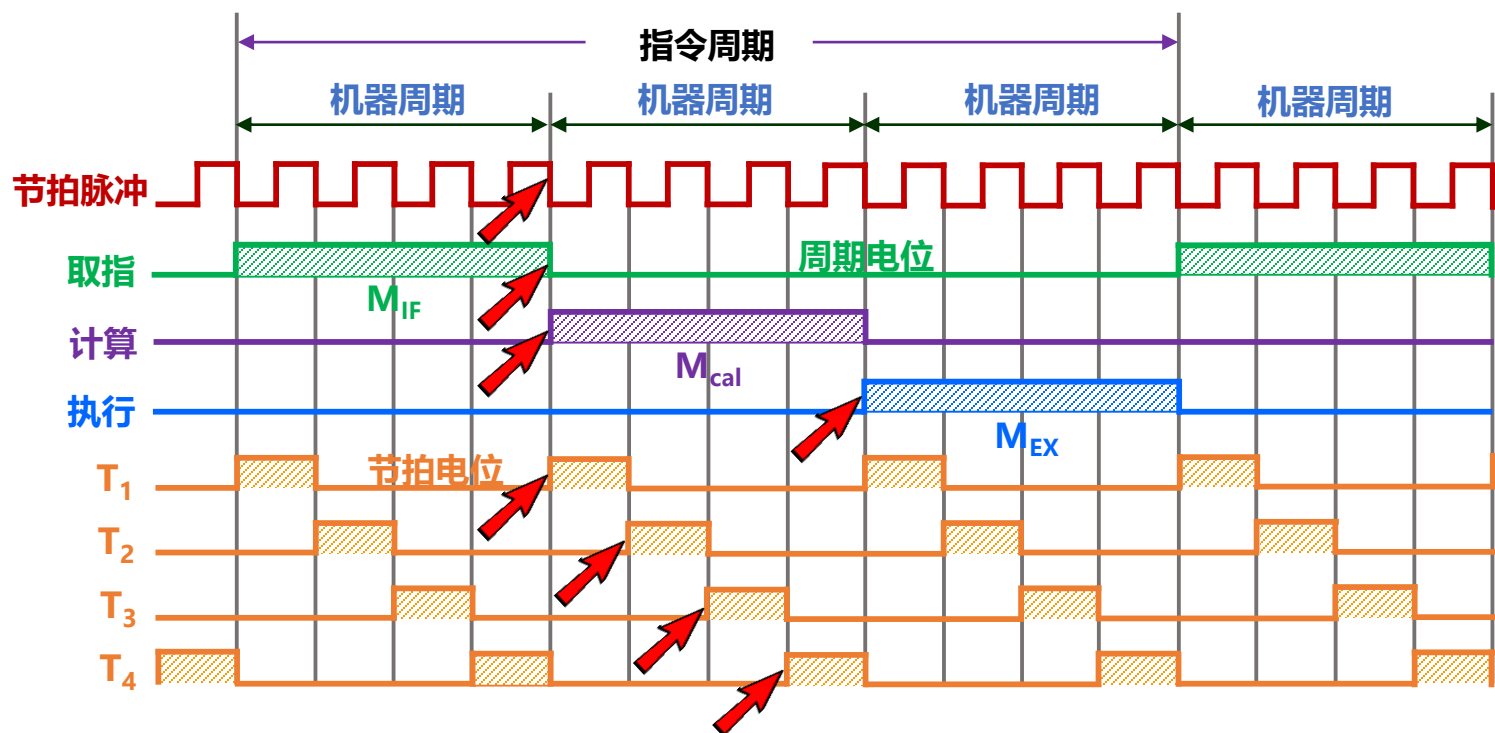
- 共享通路（总线型）

- ◆主要部件都连接在公共总线上，各部件间通过总线进行数据传输
- ◆结构简单，实现容易，但并发性较差，需分时使用总线，效率低

- 专用通路

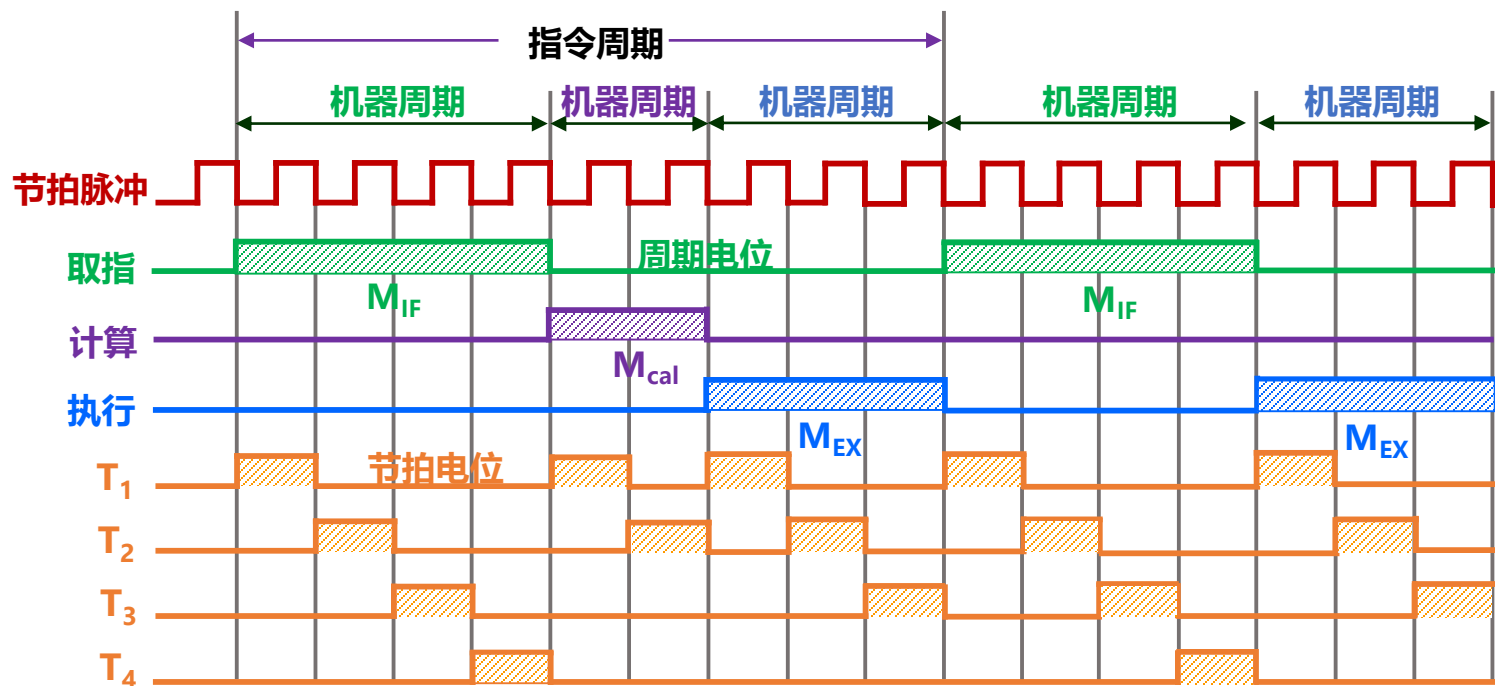
- ◆并发度高，性能佳，设计复杂，成本高
- ◆可以看作多总线结构

定长指令周期的三级时序发生器



构建时序发生器? 输入: 节拍脉冲 输出: M_{IF} , M_{cal} , M_{EX} , $T_1 \sim T_4$

变长指令周期三级时序时序发生器



机器周期数可变、节拍数可变，无周期浪费，更加灵活

计算机组成原理

全剧终

哈尔滨工业大学

第七章 课后作业

7.1 什么叫机器指令？什么叫指令系统？为什么说指令系统与机器的主要功能以及与硬件结构之间存在着密切的关系？

7.2 什么叫寻址方式？为什么要学习寻址方式？

7.6 某指令系统字长为 16 位,地址码取 4 位,试提出一种方案,使该指令系统有 8 条三地址指令、16 条二地址指令、100 条一地址指令。

7.14 设相对寻址的转移指令占两个字节,第一个字节是操作码,第二个字节是相对位移量,用补码表示。假设当前转移指令第一字节所在的地址为 2000H,且 CPU 每取出一个字节便自动完成 $(PC)+1 \rightarrow PC$ 的操作。试问当执行“JMP * +8”和“JMP * -9”指令时,转移指令第二字节的内容各为多少？

7.16 某机主存容量为 $4\text{ M} \times 16$ 位,且存储字长等于指令字长,若该机指令系统可完成 108 种操作,操作码位数固定,且具有直接、间接、变址、基址、相对、立即等六种寻址方式,试回答以下问题。

(1) 画出一地址指令格式并指出各字段的作用。

(2) 该指令直接寻址的最大范围。

(3) 一次间接寻址和多次间接寻址的寻址范围。

(4) 立即数的范围(十进制表示)。

(5) 相对寻址的位移量(十进制表示)。

(6) 上述六种寻址方式的指令中哪一种执行时间最短,哪一种最长,为什么？哪一种便于程序浮动,哪一种最适合处理数组问题？

(7) 如何修改指令格式,使指令的寻址范围可扩大到 4 M？

(8) 为使一条转移指令能转移到主存的任一位置,可采取什么措施？简要说明之。

第九章 课后作业

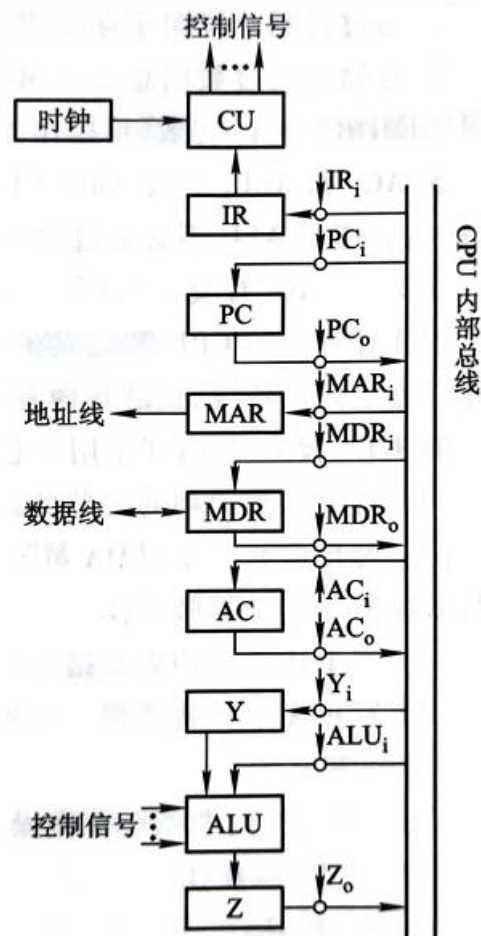
9.3 什么是指令周期、机器周期和时钟周期？三者有何关系？

9.5 设机器 A 的 CPU 主频为 8 MHz, 机器周期含 4 个时钟周期, 且该机的平均指令执行速度是 0.4 MIPS, 试求该机的平均指令周期和机器周期, 每个指令周期中含几个机器周期。如果机器 B 的 CPU 主频为 12 MHz, 且机器周期也含 4 个时钟周期, 试问 B 机的平均指令执行速度为多少 MIPS?

9.11 设 CPU 内部结构如图 9.4 所示, 此外还设有 B、C、D、E、H、L 6 个寄存器, 它们各自的输入和输出端都与内部总线相通, 并分别受控制信号控制(如 B_i 为寄存器 B 的输入控制; B_o 为寄存器 B 的输出控制)。要求从取指令开始, 写出完成下列指令所需的全部微操作和控制信号。

- (1) ADD B, C $((B) + (C) \rightarrow B)$
- (2) SUB A, H $((AC) - (H) \rightarrow AC)$

传说中的图9.4



第九章 课后作业

9.14 设单总线计算机结构如图 9.5 所示,其中 M 为主存,XR 为变址寄存器,EAR 为有效地址寄存器,LATCH 为锁存器。假设指令地址已存于 PC 中,画出“LDA * D”和“SUB X,D”指令周期信息流程图,并列出相应的控制信号序列。

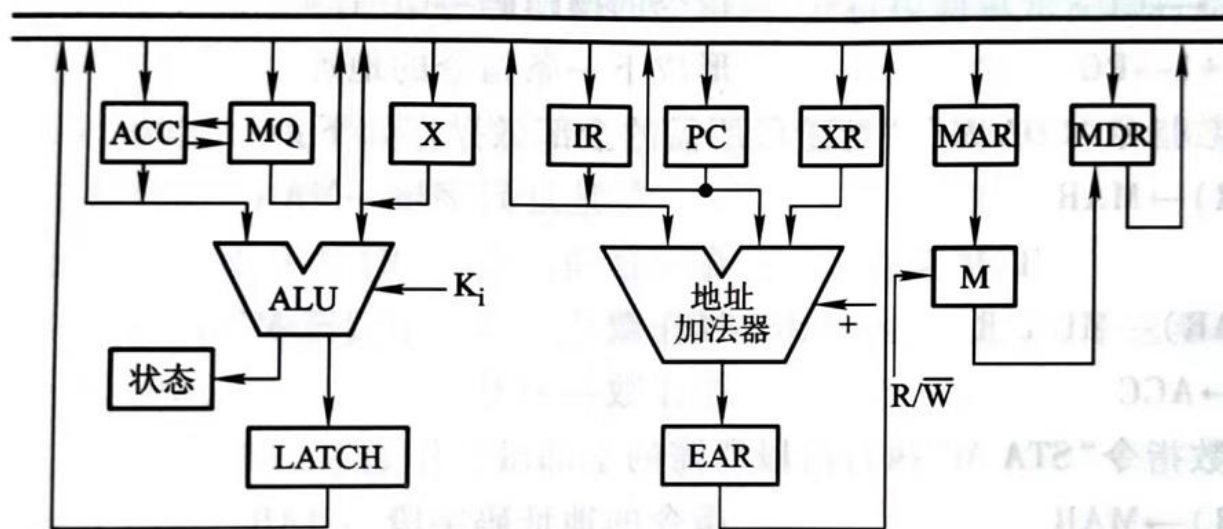
说明:

(1) “LDA * D”指令字中 * 表示相对寻址,D 为相对位移量。

(2) “SUB X,D”指令字中 X 为变址寄存器 XR,D 为形式地址。

(3) 寄存器的输入和输出均受控制信号控制,例如, PC_i 表示 PC 的输入控制信号, MDR_o 表示 MDR 的输出控制信号。

(4) 凡是需要经过总线实现寄存器之间的传送,需在流程图中注明,如 $PC \rightarrow \text{Bus} \rightarrow \text{MAR}$,相应的控制信号为 PC_i 和 MAR_i 。



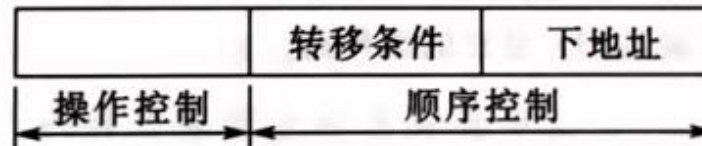
节日大礼包③:

第10章 作业

• T8 T15 T17

10.8 画出微程序控制单元的组成框图,根据指令处理过程,结合有关部件说明其工作原理。

10.15 设控制存储器的容量为 512×48 位,微程序可在整个控存空间实现转移,而控制微程序转移的条件共有 4 个(采用直接控制),微指令格式如下:



10.17 解释机器指令、微指令、微程序、毫微指令和毫微程序以及它们之间的对应关系。