



# 第 9 章 控制单元的功能

系统结构研究所·计算机组成原理



# 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

间址周期

执行周期

中断周期

# 一、取指周期

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

$1 \rightarrow R$  (启动主存读操作)

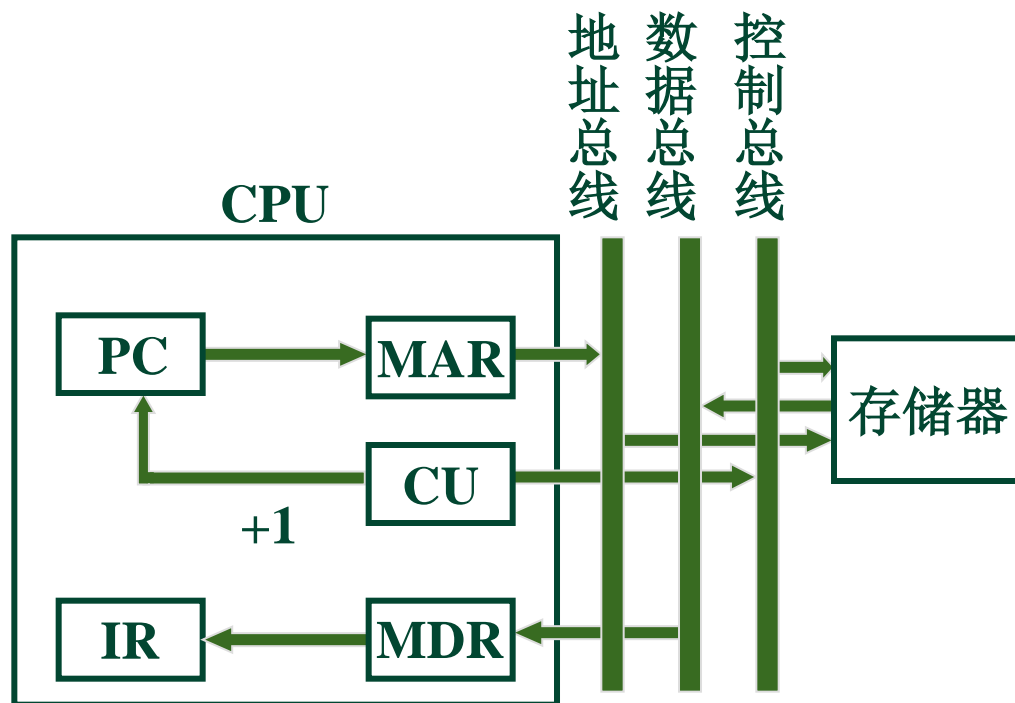
$M(MAR) \rightarrow MDR$

(将MAR所指的主存单元的内容读至MDR)

$MDR \rightarrow IR$

$OP(IR) \rightarrow CU$

$(PC) + 1 \rightarrow PC$



## 二、间址周期

9.1

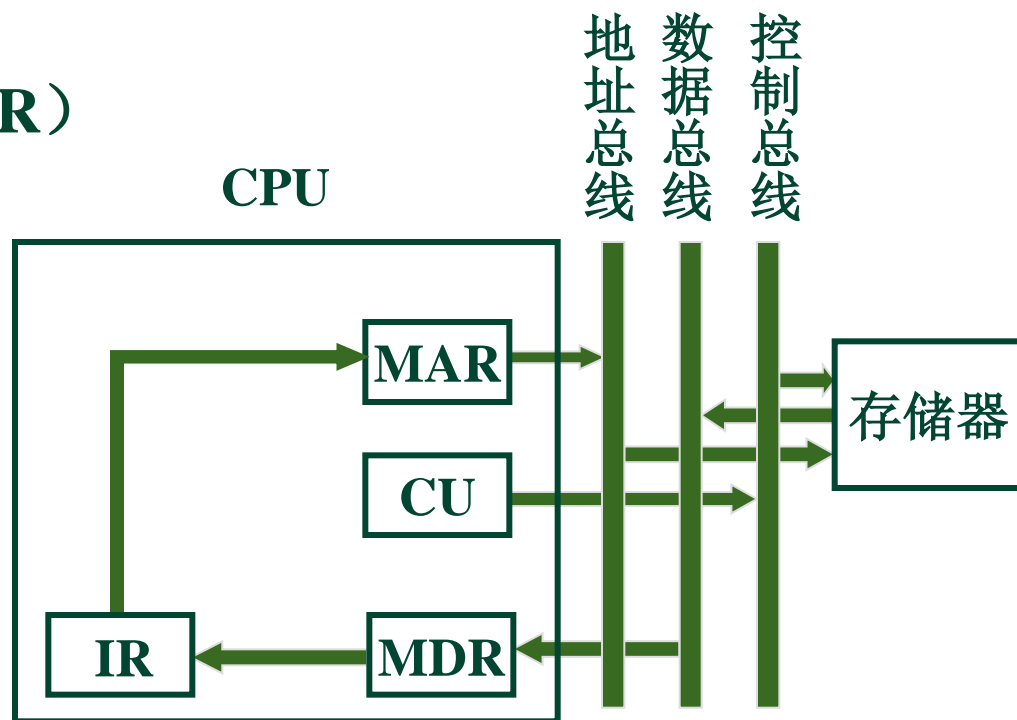
$Ad(IR) \rightarrow MAR$

(指令形式地址  $\rightarrow$  MAR)

$1 \rightarrow R$

$M(MAR) \rightarrow MDR$

$MDR \rightarrow Ad(IR)$



## 1. 非访存指令

(1) **CLA** 清A  $0 \rightarrow \text{ACC}$

(2) **COM** 取反  $\overline{\text{ACC}} \rightarrow \text{ACC}$

(3) **SHR** 算术右移  $\text{L}(\text{ACC}) \rightarrow \text{R}(\text{ACC}), (\text{ACC}_0 \rightarrow \text{ACC}_0)$

(4) **CSL** 循环左移  $\text{R}(\text{ACC}) \rightarrow \text{L}(\text{ACC}), (\text{ACC}_0 \rightarrow \text{ACC}_n)$

(5) **STP** 停机指令  $0 \rightarrow \text{G}$  (运行标志触发器)

## 2. 访存指令

### (1) 加法指令

**ADD X**

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

$1 \rightarrow \text{R}$

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

$(\text{ACC}) + (\text{MDR}) \rightarrow \text{ACC}$

### (2) 存数指令

**STA X**

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

$1 \rightarrow \text{W}$

$\text{ACC} \rightarrow \text{MDR}$

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

## (3) 取数指令

**LDA X** $\text{Ad}(\text{IR}) \rightarrow \text{MAR}$  $1 \rightarrow \text{R}$  $\text{M}(\text{MAR}) \rightarrow \text{MDR}$  $\text{MDR} \rightarrow \text{ACC}$ 

## 3. 转移指令

## (1) 无条件转

**JMP X** $\text{Ad}(\text{IR}) \rightarrow \text{PC}$ 

## (2) 条件转移

**BAN X** (负则转) $\text{A}_0 \cdot \text{Ad}(\text{IR}) + \bar{\text{A}}_0(\text{PC}) \rightarrow \text{PC}$ (结果为负即 $\text{A}_0=1$ )



## 4. 三类指令的指令周期



## 四、中断周期

9.1

程序断点存入 “0” 地址      程序断点 进栈

$0 \rightarrow \text{MAR}$

$(\text{SP}) - 1 \rightarrow \text{MAR}$

$1 \rightarrow \text{W}$

$1 \rightarrow \text{W}$

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

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

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

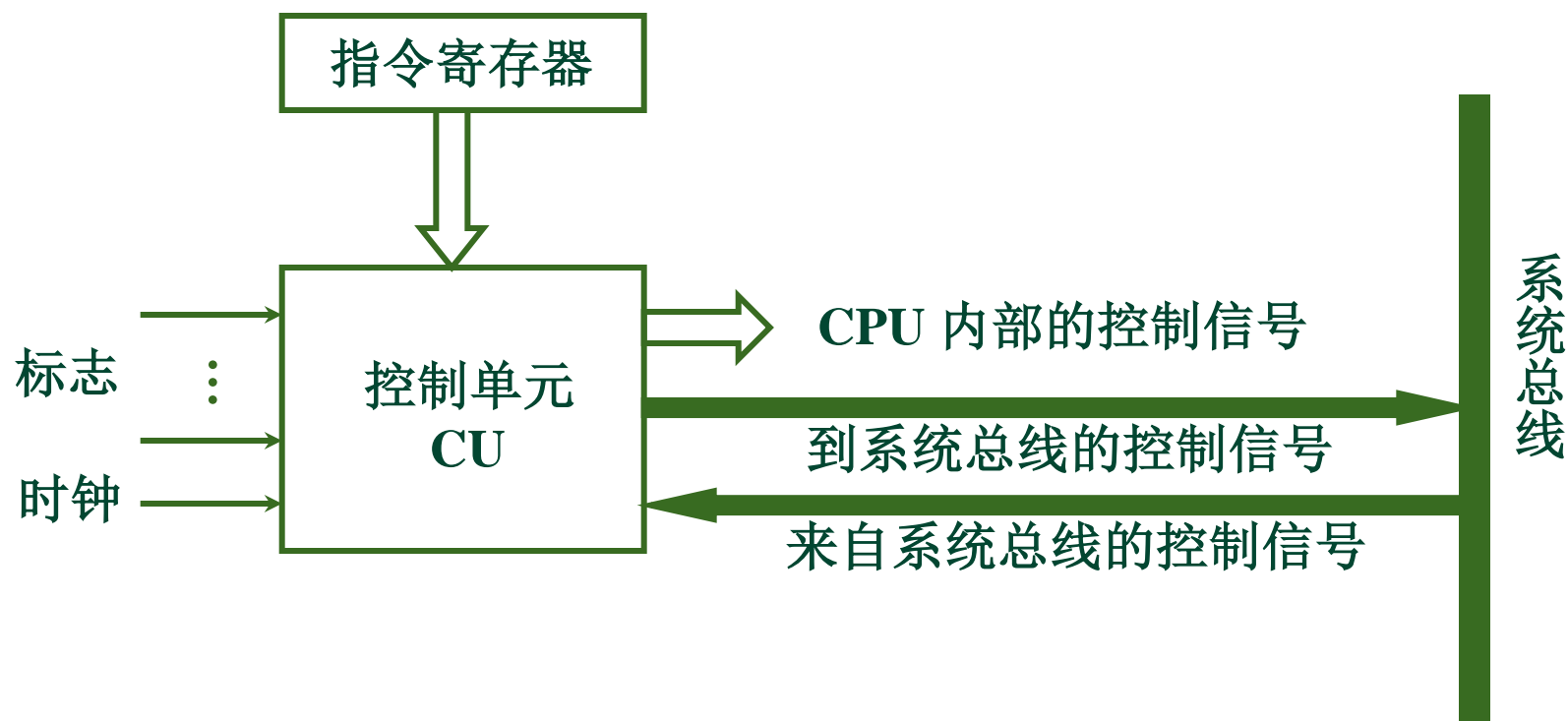
$\text{MDR} \rightarrow \text{M}(\text{MAR})$

中断识别程序入口地址  $\text{M} \rightarrow \text{PC}$

$0 \rightarrow \text{EINT} \text{ (置 “0” )}$

$0 \rightarrow \text{EINT} \text{ (置 “0” )}$

### 一、控制单元的外特性



# 1. 输入信号

9.2

## (1) 时钟

CU 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 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

$\overline{MREQ}$

访存控制信号

$\overline{IO/M}$

访 IO/ 存储器的控制信号

$\overline{RD}$

读命令

$\overline{WR}$

写命令

INTA

中断响应信号

HLDA

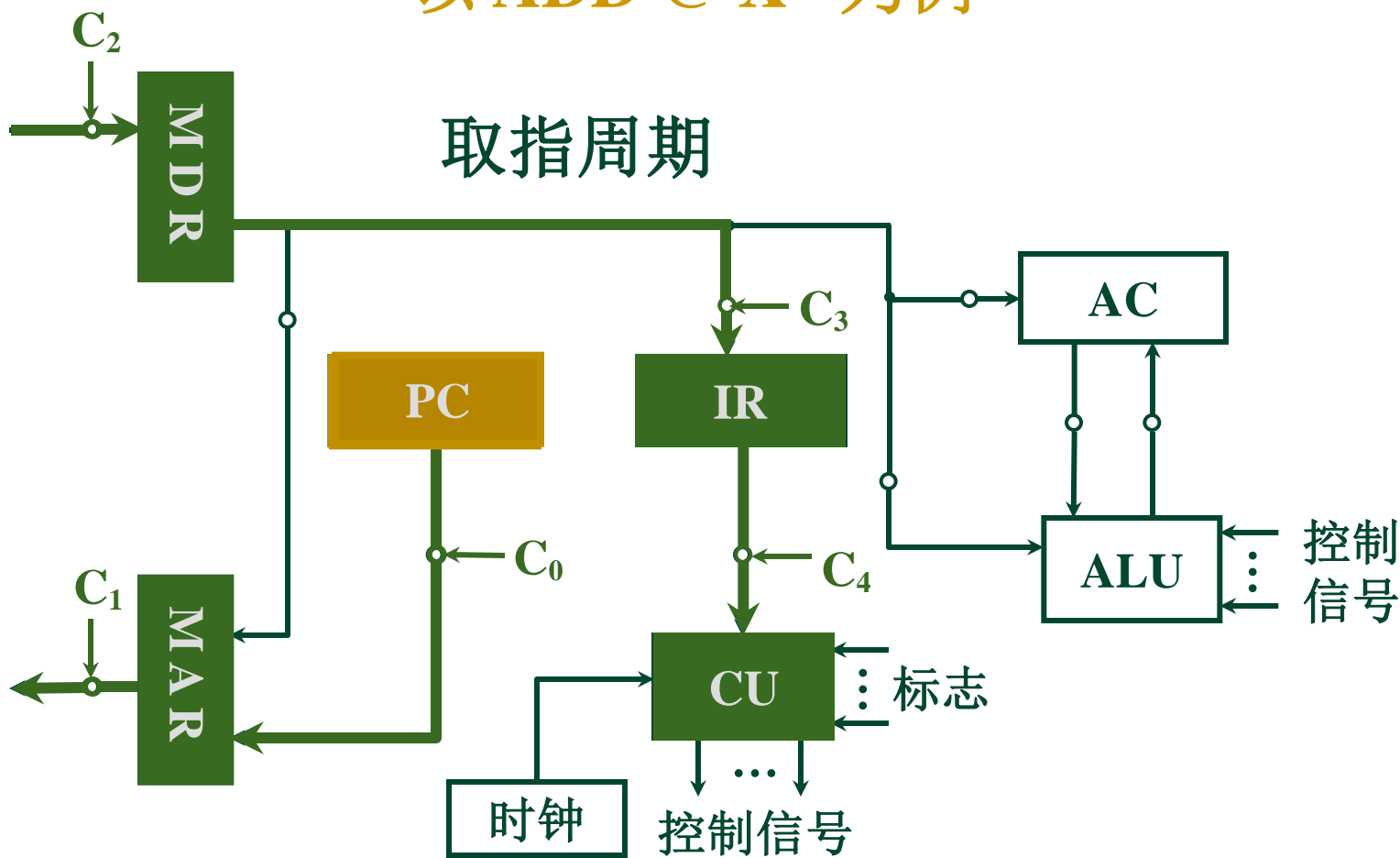
总线响应信号

## 二、控制信号举例

9.2

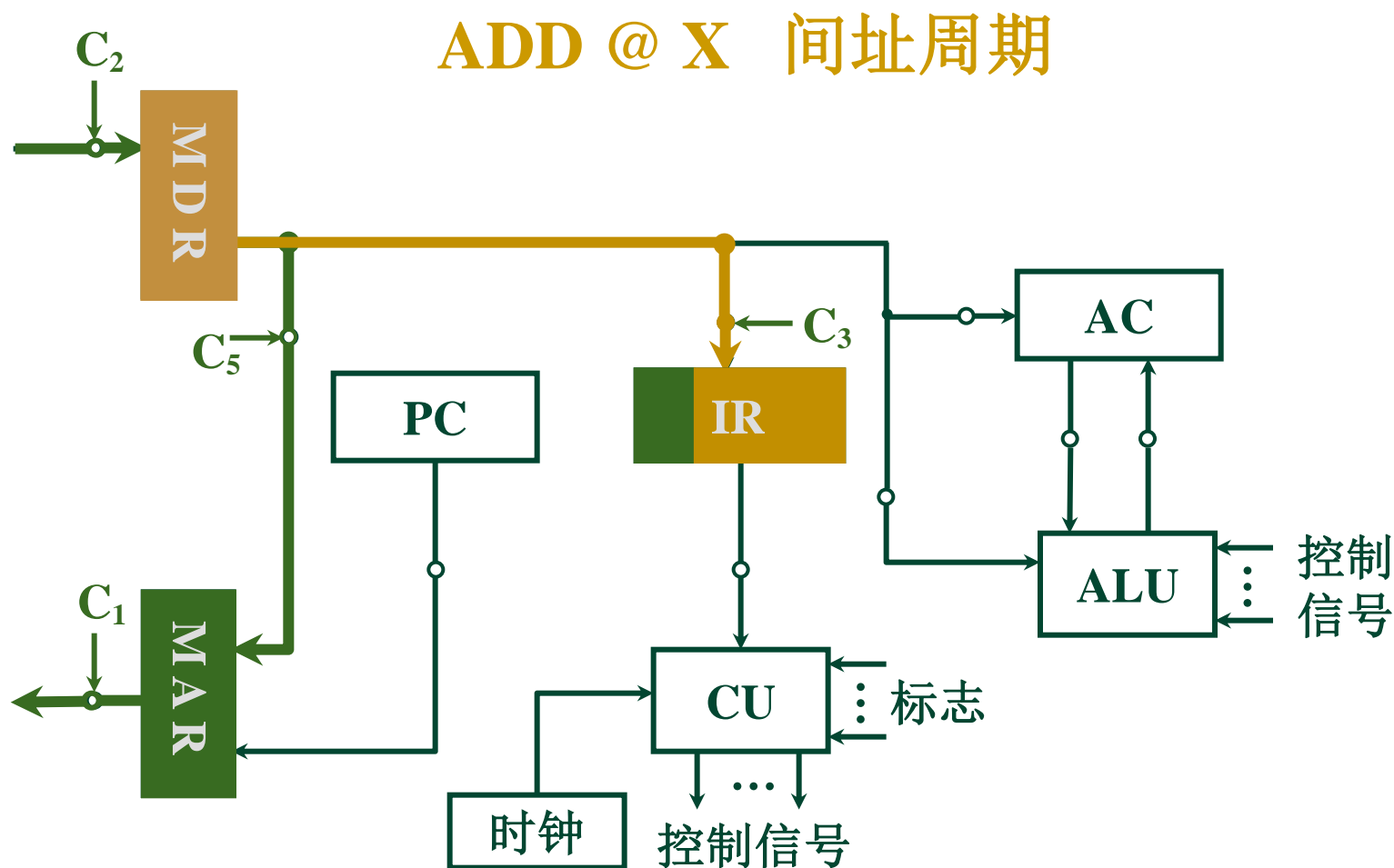
### 1. 不采用 CPU 内部总线的方式

以 **ADD @ X** 为例



## 二、控制信号举例

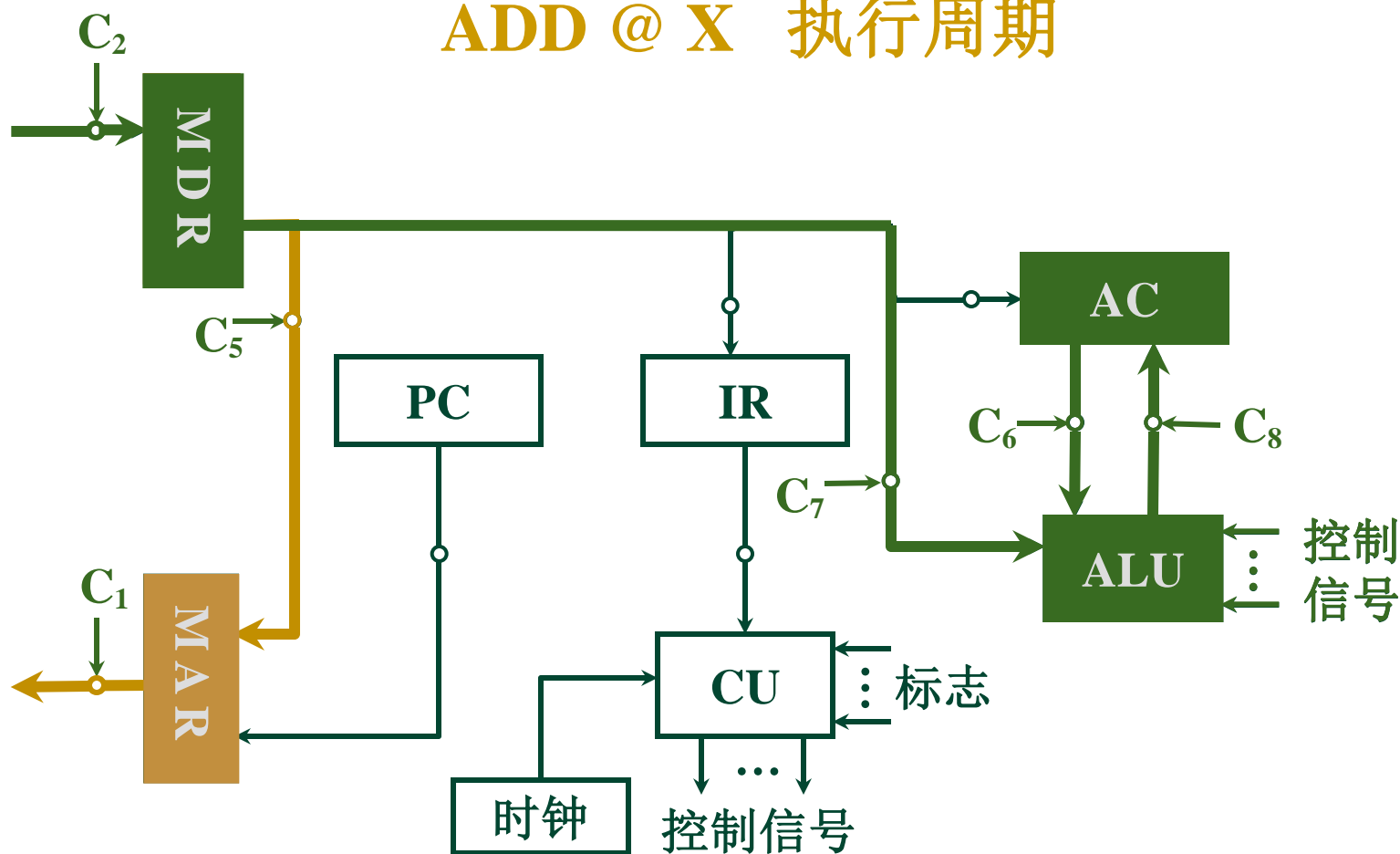
### 1. 不采用 CPU 内部总线的方式



## 二、控制信号举例

### 1. 不采用 CPU 内部总线的方式

ADD @ X 执行周期





## 2. 采用 CPU 内部总线方式

### (1) ADD @ X 取指周期

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

$PC_0 \quad MAR_i$

• CU 发读命令  $1 \rightarrow R$

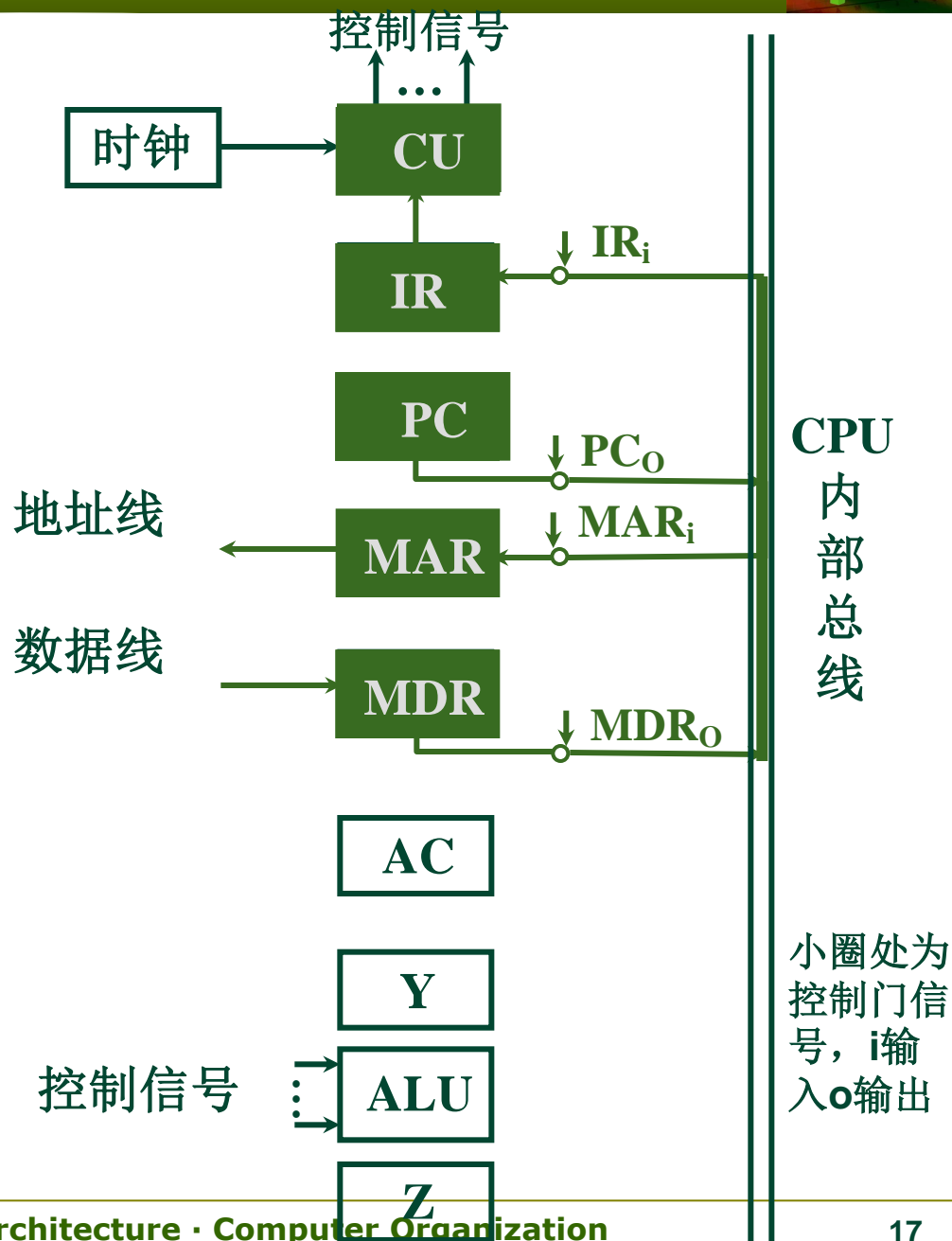
• 数据线  $\rightarrow MDR$

•  $MDR \rightarrow IR$

$MDR_0 \quad IR_i$

•  $OP(IR) \rightarrow CU$

•  $(PC) + 1 \rightarrow PC$



# (2) ADD @ X 间址周期

形式地址  $\rightarrow$  MAR

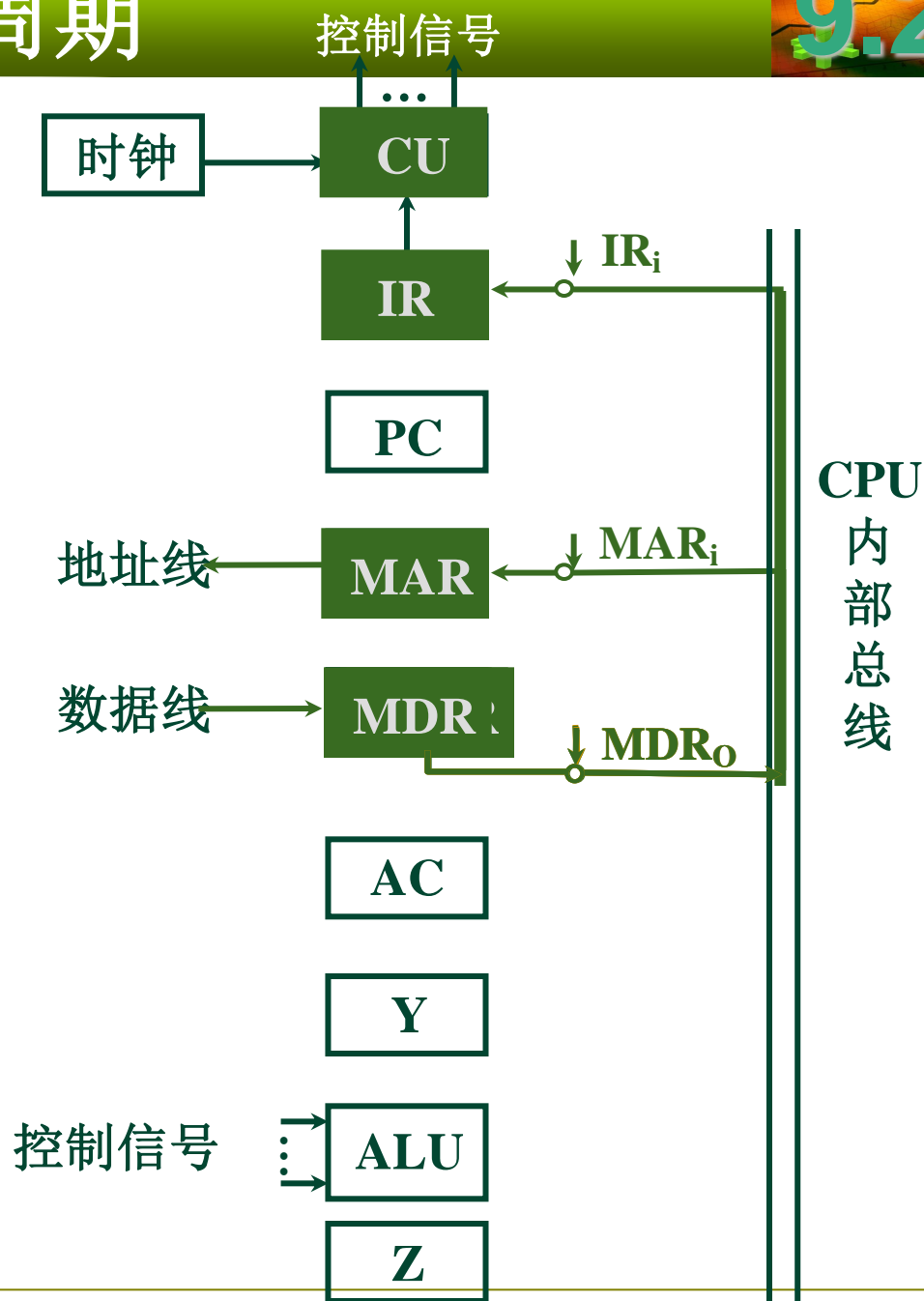
•  $\text{MDR} \rightarrow \text{MAR} \rightarrow \text{地址线}$   
 $\text{MDR}_0 \quad \text{MAR}_i$

•  $1 \rightarrow R$

• 数据线  $\rightarrow$  MDR

•  $\text{MDR} \rightarrow \text{IR}$   
 $\text{MDR}_0 \quad \text{IR}_i$

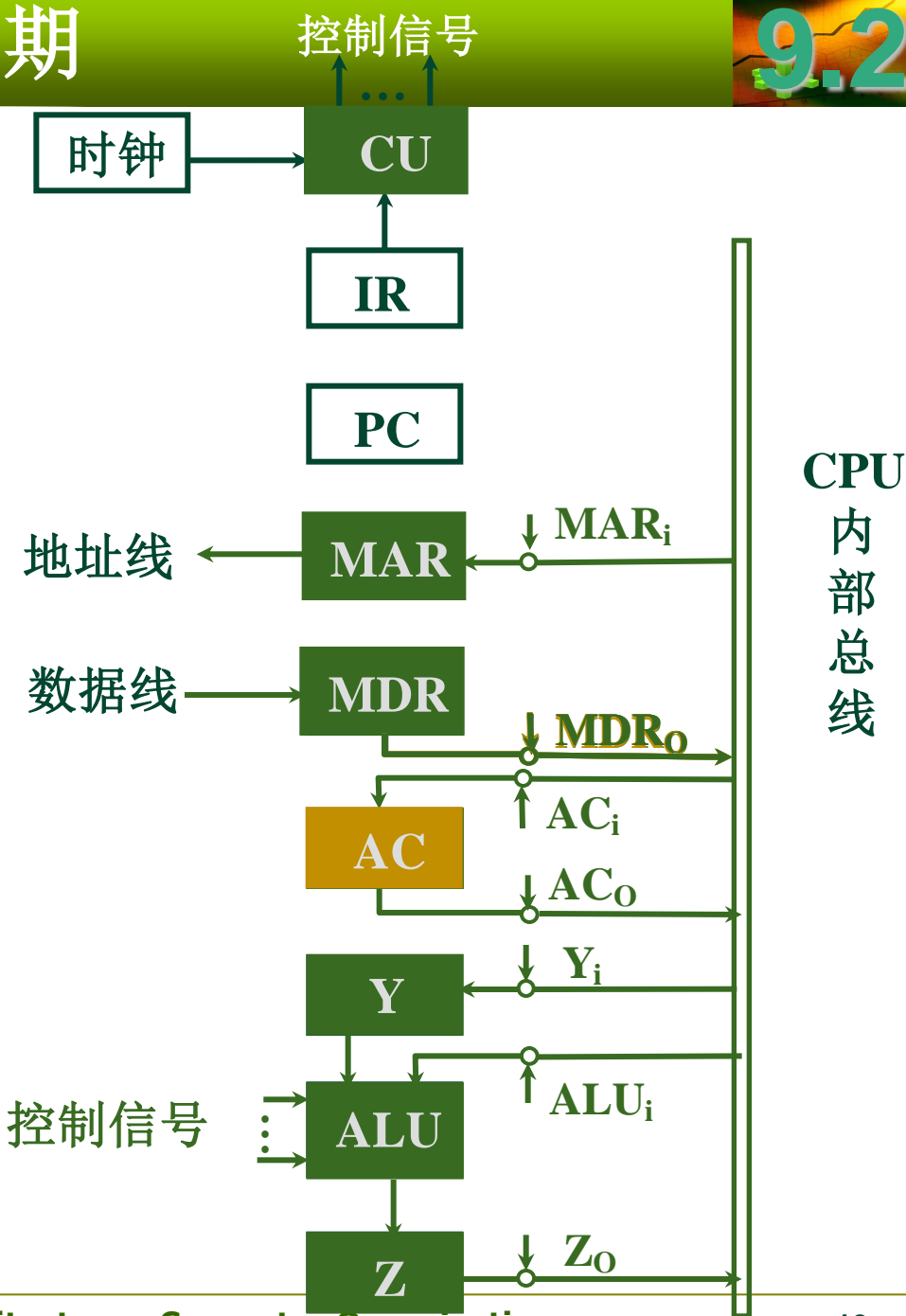
有效地址  $\rightarrow \text{Ad}(\text{IR})$



### (3) ADD @ X 执行周期

9.2

- $\text{MDR} \rightarrow \text{MAR} \rightarrow \text{地址线}$   
 $\text{MDR}_0 \quad \text{MAR}_i$
- $1 \rightarrow R$
- 数据线  $\rightarrow \text{MDR}$
- $\text{MDR} \rightarrow Y \rightarrow \text{ALU}$   
 $\text{MDR}_0 \quad Y_i$
- $\text{AC} \rightarrow \text{ALU}$   
 $\text{AC}_0 \quad \text{ALU}_i$
- $(\text{AC}) + (\text{Y}) \rightarrow \text{Z}$
- $\text{Z} \rightarrow \text{AC}$   
 $\text{Z}_0 \quad \text{AC}_i$



# 三、多级时序系统



## 1. 机器周期

### (1) 机器周期的概念

所有指令执行过程中的一个基准时间

### (2) 确定机器周期需考虑的因素

每条指令的执行 步骤

每一步骤 所需的 时间

### (3) 基准时间的确定

- 以完成 最复杂 指令功能的时间 为准
- 以 访问一次存储器 的时间 为基准

若指令字长 = 存储字长      取指周期 = 机器周期

## 2. 时钟周期（节拍、状态）

- 一个机器周期内可完成若干个微操作
- 每个微操作需一定的时间，以时钟信号来控制产生每一个微操作命令
- 时钟信号控制节拍发生器，产生节拍，每个节拍宽度对应一个时钟周期
- 将一个机器周期分成若干个时间相等的时间段（节拍、状态、时钟周期）
- 时钟周期是控制计算机操作的最小单位时间
- 用时钟周期控制产生一个或几个微操作命令

# 多级时序系统

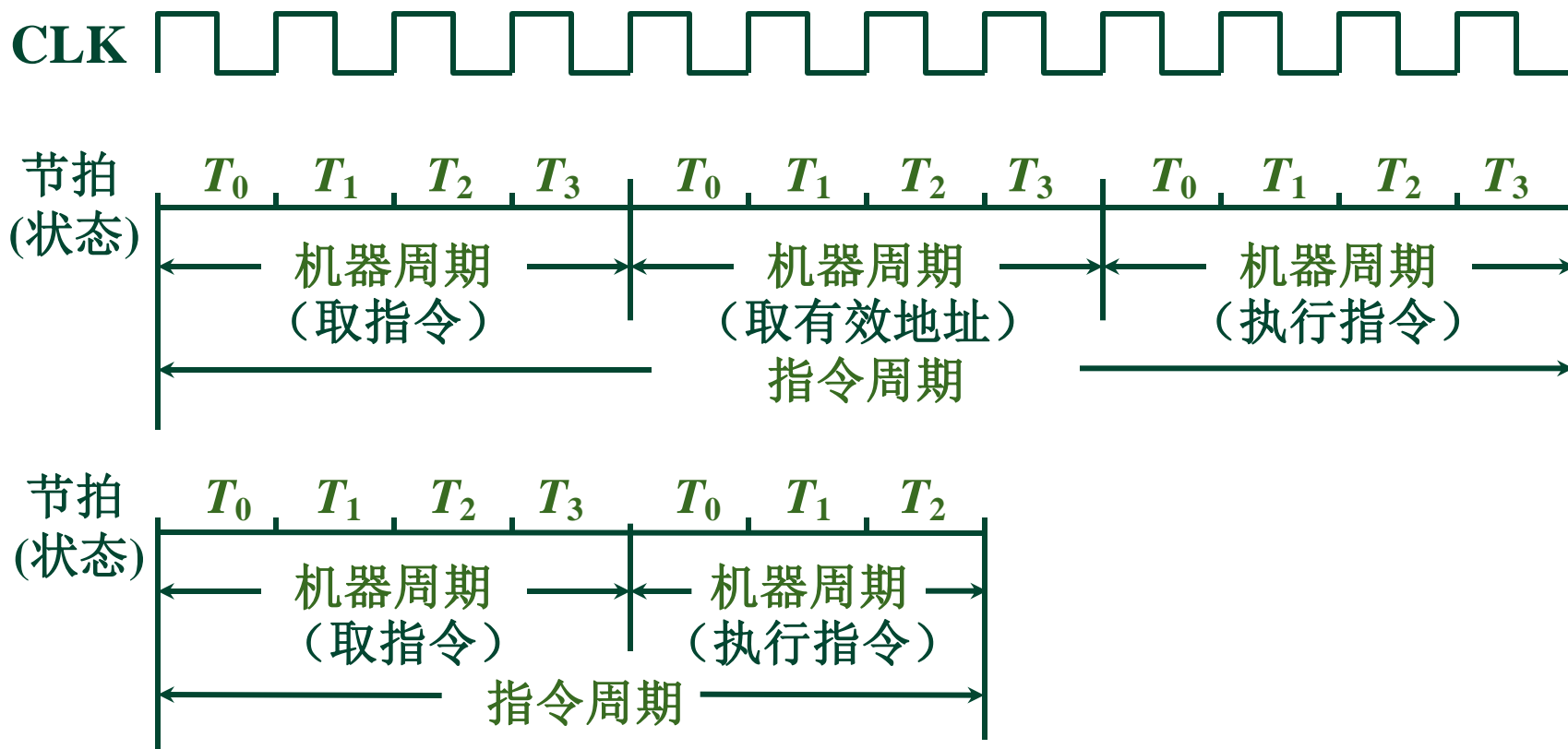
- 指令周期是从取指令、分析指令到执行完该指令所需的时间。
- 不同的指令，其指令周期长短可以不同。
- 在时序系统中通常不为指令周期设置时间标志信号，因而也不将其作为时序的一级

# 多级时序系统

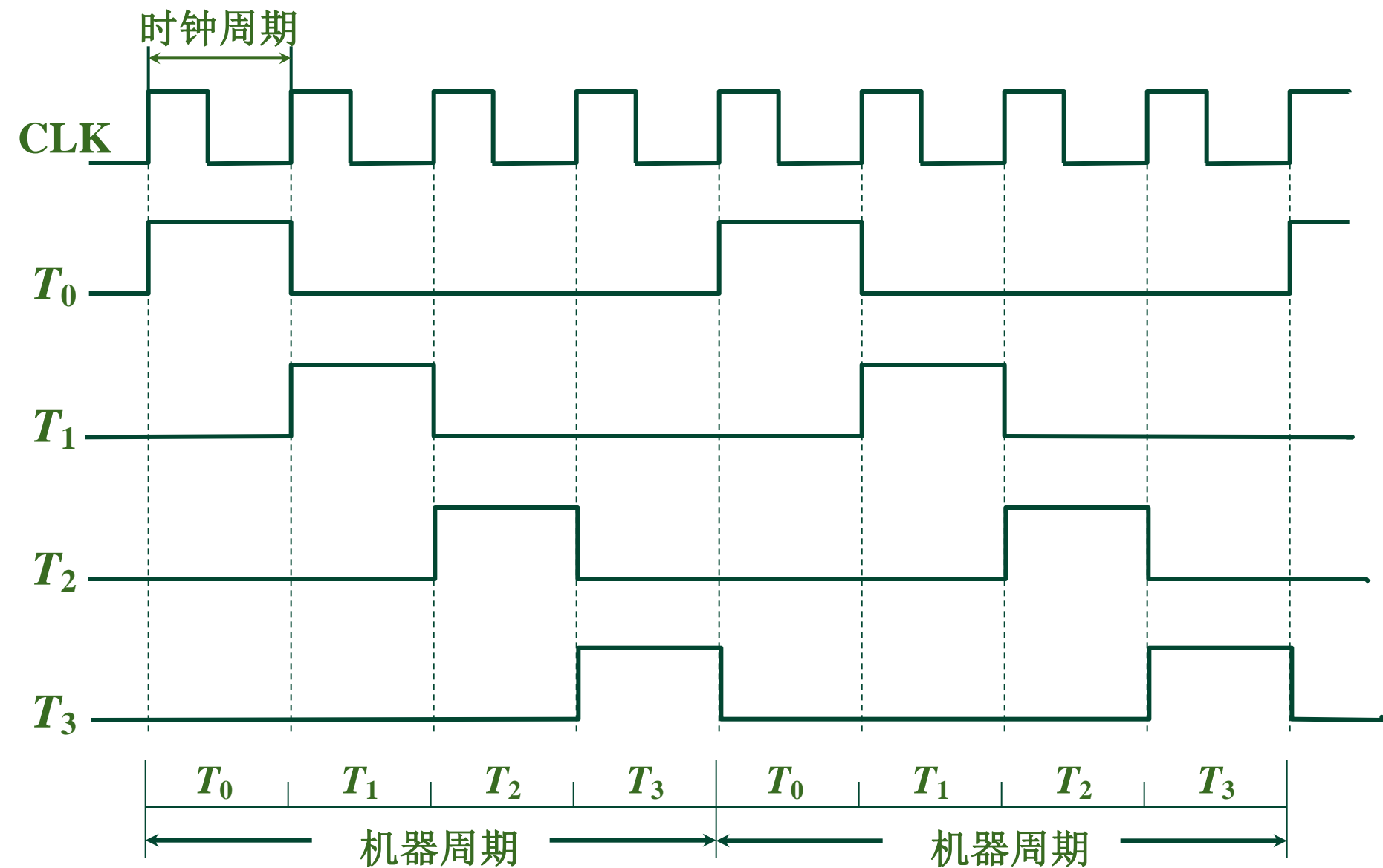
机器周期、节拍（状态）组成多级时序系统

一个指令周期包含若干个机器周期

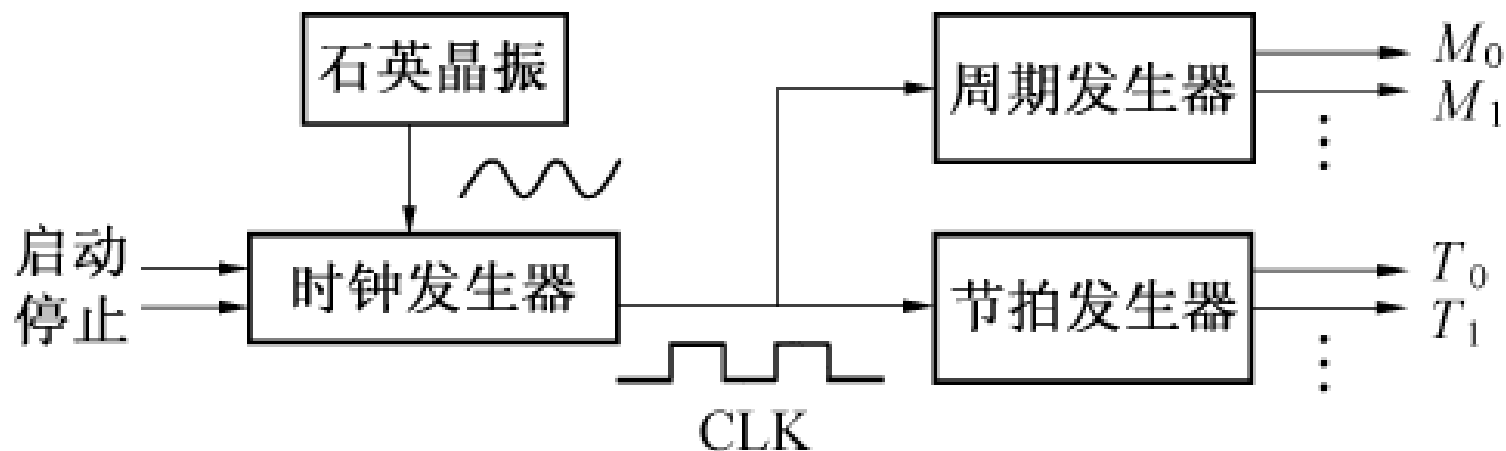
一个机器周期包含若干个时钟周期



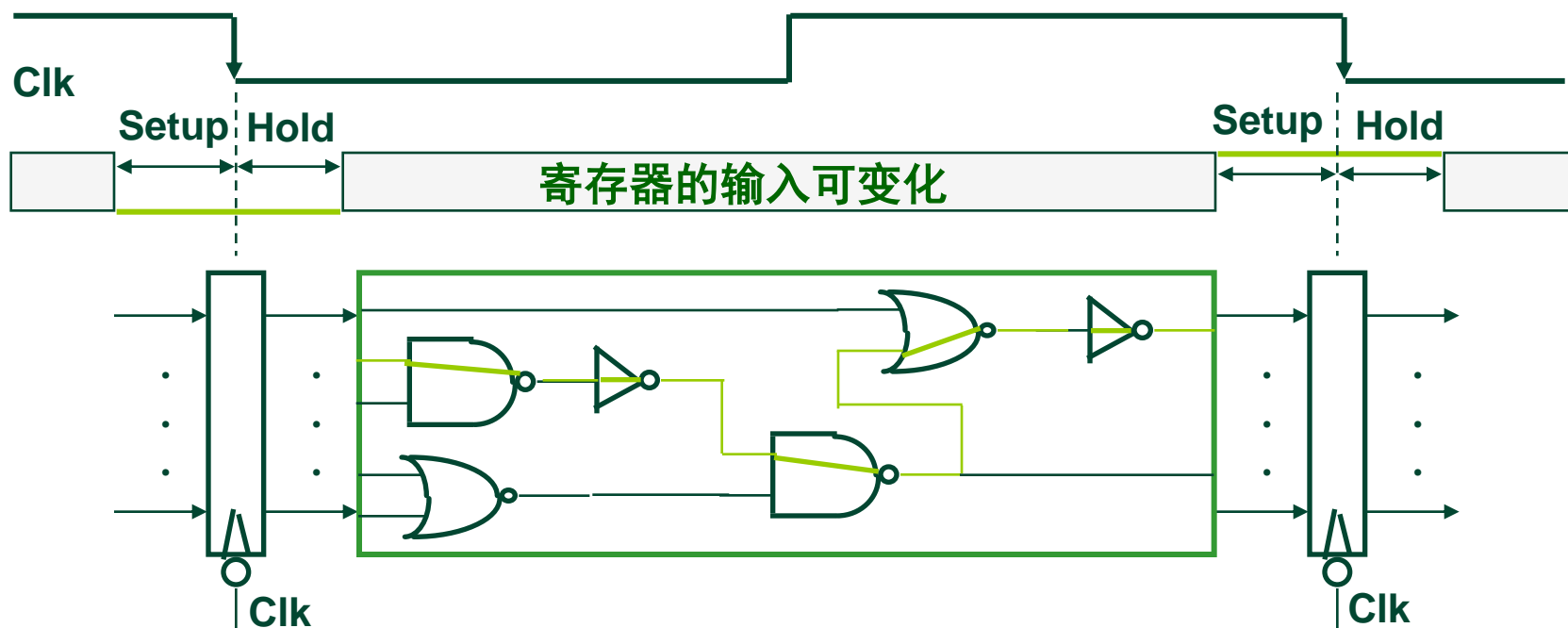
# 时钟周期（节拍、状态）







- ❖ 主振电路中的石英晶体振荡器产生一系列**正弦信号**，经时钟发生器整形分频得到**时钟脉冲信号**，成为机器的**主频脉冲**。主频脉冲再经过周期发生器和节拍发生器产生周期电位和节拍电位。



数据通路由 “... + 状态元件 + 操作元件( 组合电路) + 状态元件 + ...” 组成

只有状态元件能存储信息，所有操作元件都须从状态单元接收输入，并将输出写入状态单元中。其输入为前一时钟生成的数据，输出为当前时钟所用的数据

❖ 假定采用下降沿触发（负跳变）方式（也可以是上升沿方式）

- 所有状态单元在下降沿写入信息，经过Latch Prop (clk-to-Q) 后输出有效
- $\text{Cycle Time} = \text{Latch Prop} + \text{Longest Delay Path} + \text{Setup} + \text{Clock Skew(最大偏移)}$

# 机器速度与机器主频的关系

- 机器的主频  $f$  越快 机器的速度也越快
- 在机器周期所含时钟周期数 相同 的前提下，  
两机 平均指令执行速度之比 等于 两机主频之比

$$\frac{\text{MIPS}_1}{\text{MIPS}_2} = \frac{f_1}{f_2}$$

见例题

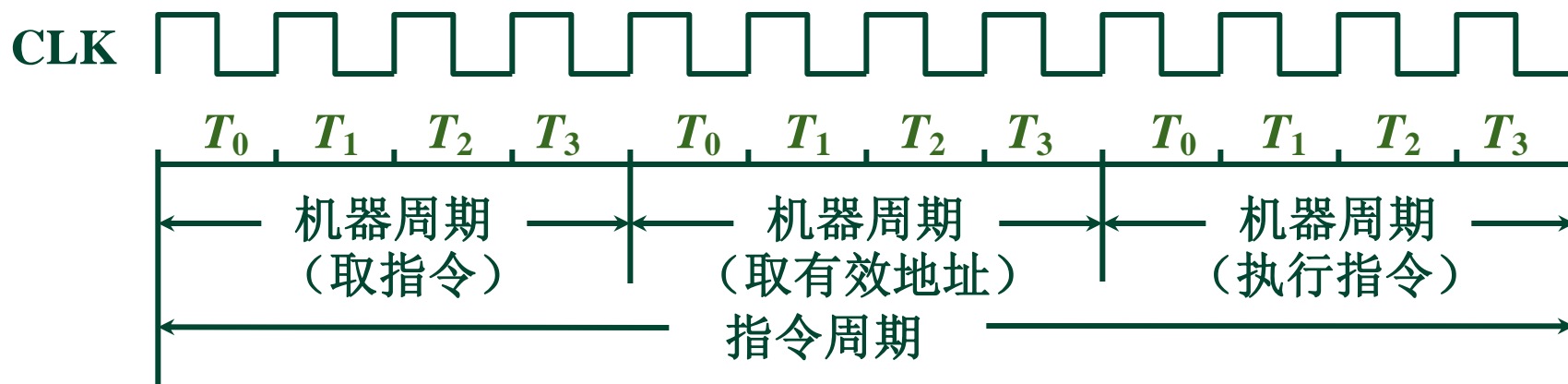
- 机器速度 不仅与 主频有关，还与机器周期中所含时钟周期（主频的倒数）数 以及指令周期中所含的 机器周期数有关
- 机器运行速度还与字长和计算机体系结构有关。  
字长越长，单位时间内完成的数据运算就越多，运算速度越快  
体系结构：存储器采用分级结构，处理器流水结构、多机结构等  
❖ 其他因素：主存速度、Cache、总线传输速度、硬盘速度

## 四、CU的控制方式

产生不同微操作命令序列所用的时序控制方式

### 1. 同步控制方式

任一微操作均由 **统一基准时标** 的时序信号控制



#### (1) 采用 **定长** 的机器周期

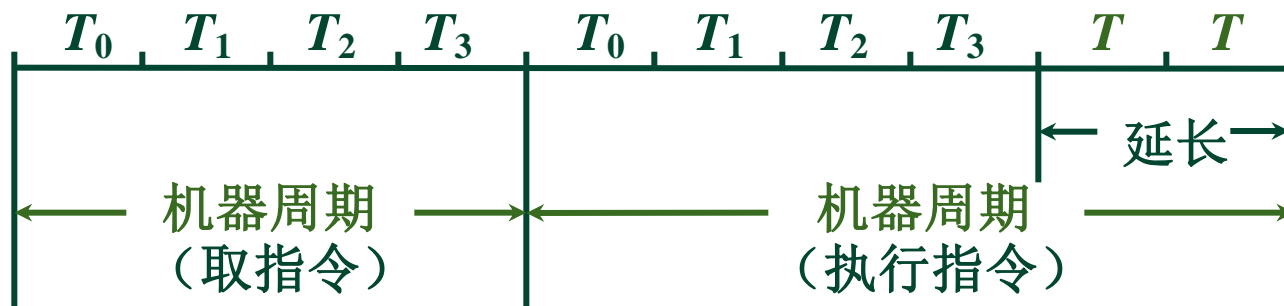
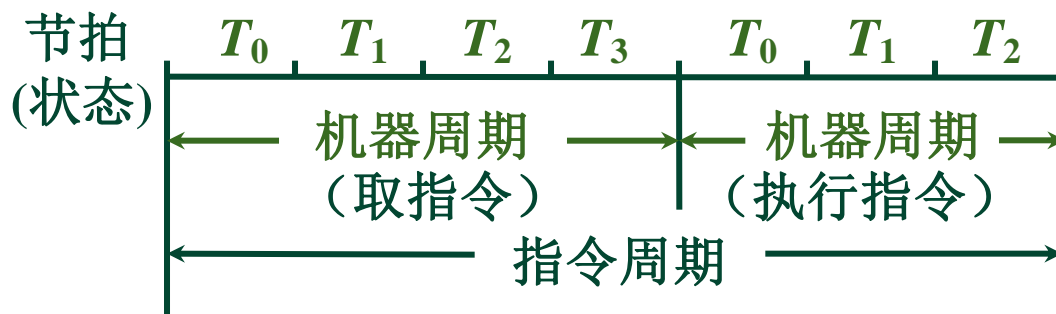
以 **最长** 的 **微操作序列** 和 **最繁** 的微操作作为 **标准**

机器周期内 **节拍数**相同

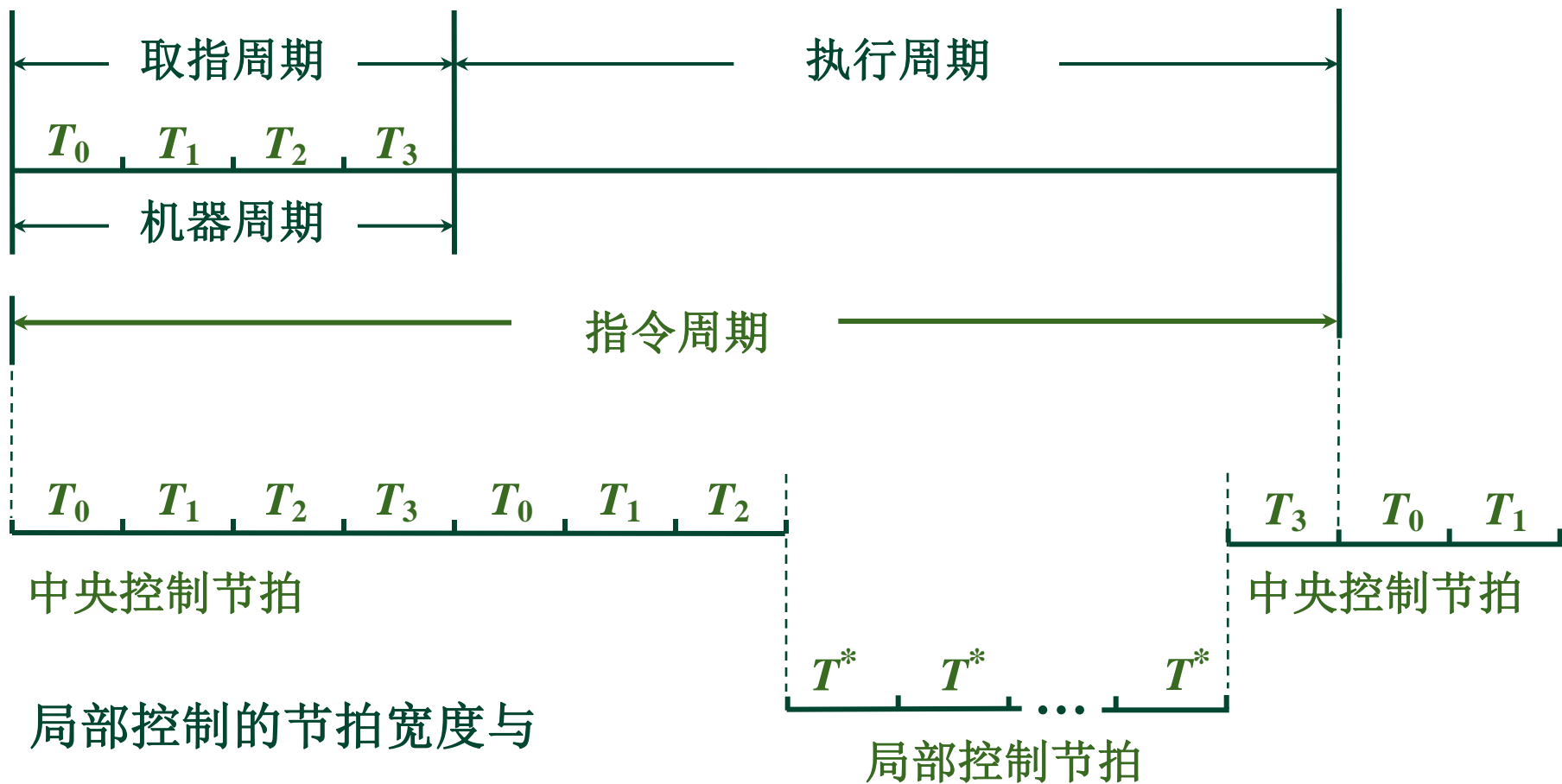
## (2) 采用不定长的机器周期

9.2

### 机器周期内 节拍数不等



### (3) 采用中央控制和局部控制相结合的方法



局部控制的节拍宽度与  
中央控制的节拍宽度一致

## 2. 异步控制方式

无基准时标信号

无固定的周期节拍和严格的时钟同步

采用 应答方式

## 3. 联合控制方式

同步与异步相结合

大部分统一、小部分区别对待

如：取指同步、I/O异步

## 4. 人工控制方式

(1) Reset

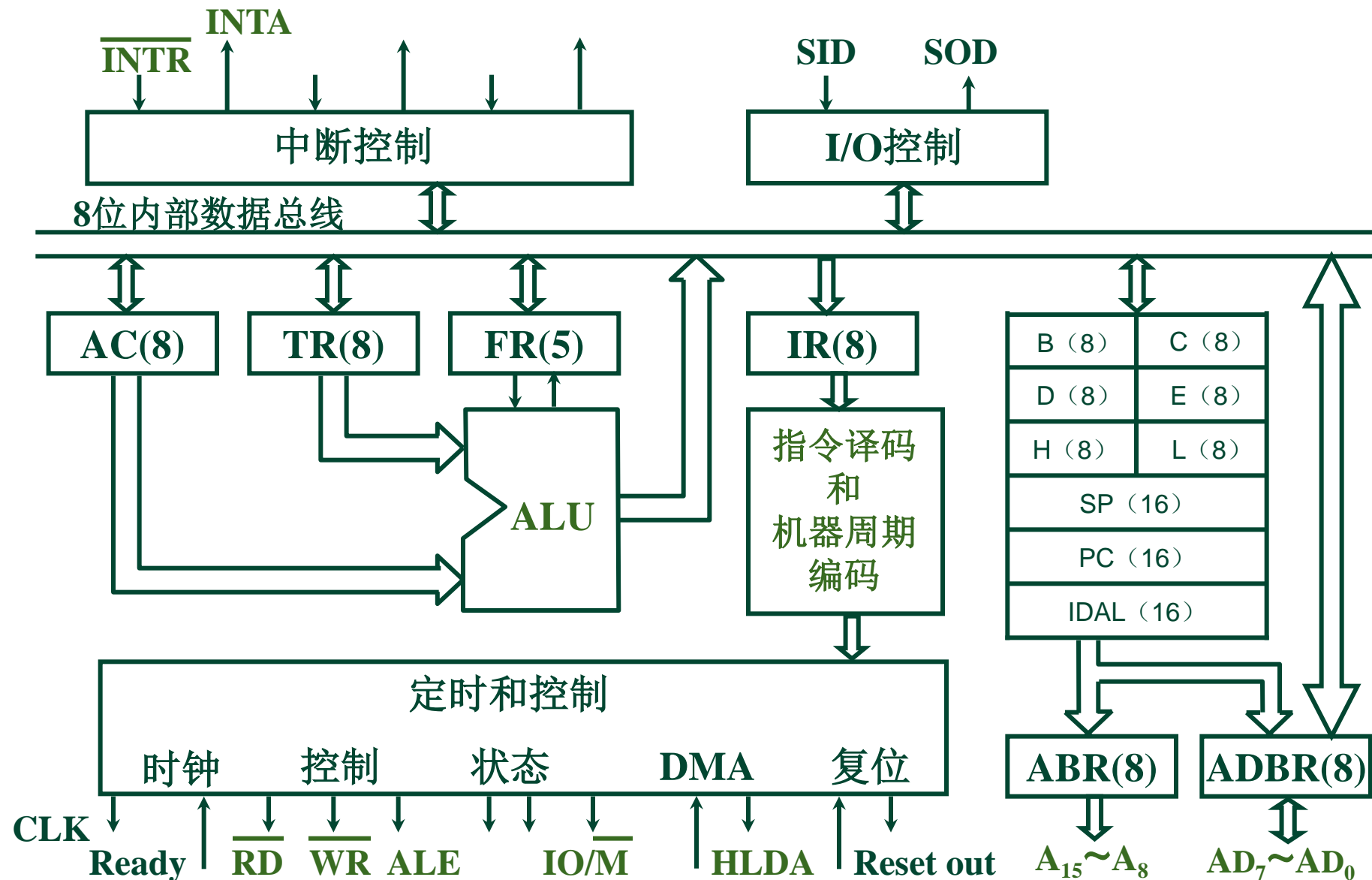
(2) 连续 和 单条 指令执行转换开关

(3) 符合停机开关

# 五、多级时序系统实例分析

9.2

## 1. 8085 的组成





## 2. 8085 的外部引脚

9.2

### (1) 地址和数据信号

$A_{15} \sim A_8$      $AD_7 \sim AD_0$

SID        SOD

### (2) 定时和控制信号

入    $X_1$     $X_2$

出   CLK   ALE    $S_0$     $S_1$   
       $IO/\overline{M}$     $\overline{RD}$     $\overline{WR}$

### (3) 存储器和 I/O 初始化

入   HOLD   Ready

出   HLDA

$X_1$	□	1	40	□	$V_{CC}$
$X_2$	□	2	39	□	HOLD
Reset out	□	3	38	□	HLDA
SOD	□	4	37	□	<u>CLK(out)</u>
SID	□	5	36	□	<u>Rstet in</u>
Trap	□	6	35	□	Ready
RST7.5	□	7	34	□	$IO/\overline{M}$
RST6.5	□	8	33	□	<u><math>S_1</math></u>
RST5.5	□	9	32	□	<u>RD</u>
$\overline{INTR}$	□	10	31	□	<u>WR</u>
INTA	□	11	30	□	ALE
$AD_0$	□	12	29	□	$S_0$
$AD_1$	□	13	28	□	$A_{15}$
$AD_2$	□	14	27	□	$A_{14}$
$AD_3$	□	15	26	□	$A_{13}$
$AD_4$	□	16	25	□	$A_{12}$
$AD_5$	□	17	24	□	$A_{11}$
$AD_6$	□	18	23	□	$A_{10}$
$AD_7$	□	19	22	□	$A_9$
$V_{SS}$	□	20	21	□	$A_8$

## (4) 与中断有关的信号

9.2

入  $\overline{\text{INTR}}$

出  $\text{INTA}$

Trap 重新启动中断

## (5) CPU 初始化

入  $\overline{\text{Reset in}}$

出  $\text{Reset out}$

## (6) 电源和地

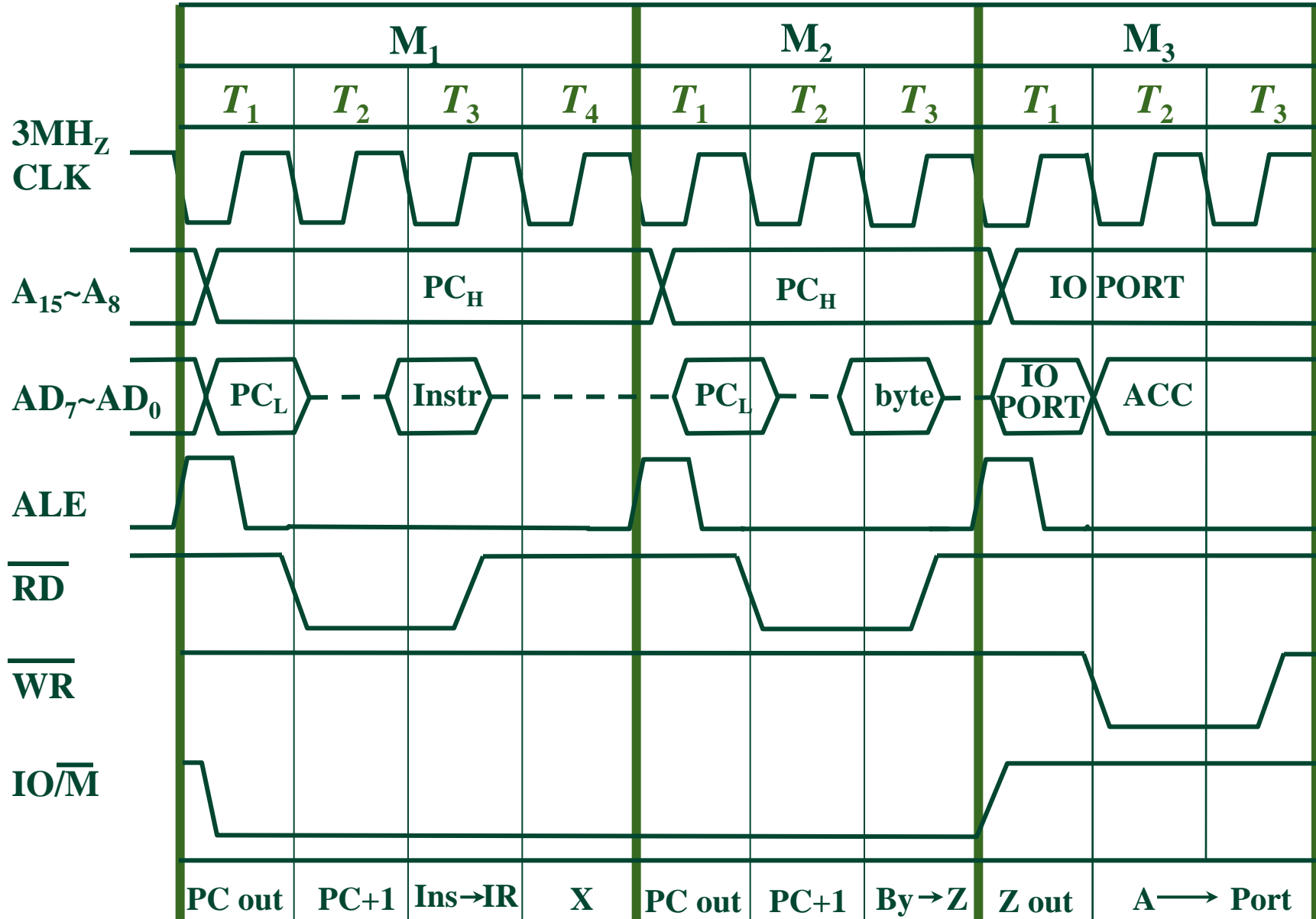
$V_{CC}$  +5 V

$V_{SS}$  地

$X_1$	1	40	$V_{CC}$
$X_2$	2	39	HOLD
Reset out	3	38	HLDA
SOD	4	37	$\overline{\text{CLK(out)}}$
SID	5	36	$\overline{\text{Rreset in}}$
Trap	6	35	Ready
RST7.5	7	34	IO/M
RST6.5	8	33	$\overline{S_1}$
RST5.5	9	32	$\overline{\text{RD}}$
$\overline{\text{INTR}}$	10	31	$\overline{\text{WR}}$
$\text{INTA}$	11	30	ALE
$\text{AD}_0$	12	29	$S_0$
$\text{AD}_1$	13	28	$A_{15}$
$\text{AD}_2$	14	27	$A_{14}$
$\text{AD}_3$	15	26	$A_{13}$
$\text{AD}_4$	16	25	$A_{12}$
$\text{AD}_5$	17	24	$A_{11}$
$\text{AD}_6$	18	23	$A_{10}$
$\text{AD}_7$	19	22	$A_9$
$V_{SS}$	20	21	$A_8$



### 3. 机器周期和节拍（状态）与控制信号的关系



以一条输出指令（I/O 写）为例：

（指令字长16位、数据线8位，分两次取）

机器周期  $M_1$  取指令操作码

机器周期  $M_2$  取设备地址

机器周期  $M_3$  执行 ACC 的内容写入设备

每个 控制 信号在 指定机器周期 的  
指定节拍  $T$  时刻 发出

# 第九章 控制单元的功能（回顾）

## ❖ 9.1 微操作命令分析：四个工作周期的微操作

## ❖ 9.2 CU的功能

- 外特性：输入、输出
- 控制信号举例 《=》 发出微操作命令
- 多级时序系统：机器周期、时钟周期
- 控制方式：
  - 同步：定长与不定长时钟周期、中央与局部控制结
  - 异步、联合、人工控制方式

例. 设某机平均执行一条指令需要两次访问内存，平均需要3个CPU周期，每个CPU周期平均包含4个节拍周期。若机器主频为240MHz，问：

(1) 若主存为“0等待”（即不需要插入等待周期），问执行一条指令的平均时间为多少？

(2) 若每次访问内存需要插入2个等待周期，问执行一条指令的平均时间又是多少？

解：因为主频为240MHz，所以节拍周期= $(1/240) \mu s$ 每个  
因为每个CPU周期平均包含4个节拍周期，所以：

$$\text{CPU周期} = \text{节拍周期} \times 4 = 4/240\text{MHz} = (1/60)\mu s$$

若访存不需要插入等待周期，则执行一条指令平均需要3个CPU周期，所以：

$$\text{指令周期} = 3 \times \text{CPU周期} = 3 \times (1/60) \mu s = (1/20)\mu s = 0.05\mu s$$

$$\text{机器平均速度} = 1/0.05\mu s = 20 \text{ MIPS}$$

(2) 平均执行一条指令需要两次访问内存，每次访问内存需要插入2个等待周期，所以：

$$\begin{aligned} \text{指令周期} &= 0.05\mu s + 2 \times (1/240)\mu s = (1/20)\mu s + (1/120)\mu s \\ &= (7/120)\mu s \end{aligned}$$

$$\text{机器平均速度} = 120/7 \approx 17 \text{ MIPS}$$

$$\frac{\text{MIPS}_1}{\text{MIPS}_2} = \frac{f_1}{f_2}$$

若某机主频为200MHz，每个指令周期平均为2.5CPU周期，每个CPU周期平均包括2个主频周期，问：

(1) 该机平均指令执行速度为多少MIPS？

(2) 若主频不变，但每条指令平均包括5个CPU周期，每个CPU周期又包含4个主频周期，平均指令执行速度又为多少MIPS？

由此可得出什么结论？

解：（1）主频为200MHz，所以主频周期 $=1/200\text{MHz}=0.005\ \mu\text{s}$   
每个指令周期平均为2.5CPU周期，每个CPU周期平均包括2个主频周期，所以一条指令的执行时间 $=2.5 \times 2 \times 0.005\ \mu\text{s}=0.025\ \mu\text{s}$

该机平均指令执行速度 $=1/0.025=40\text{MIPS}$ 。

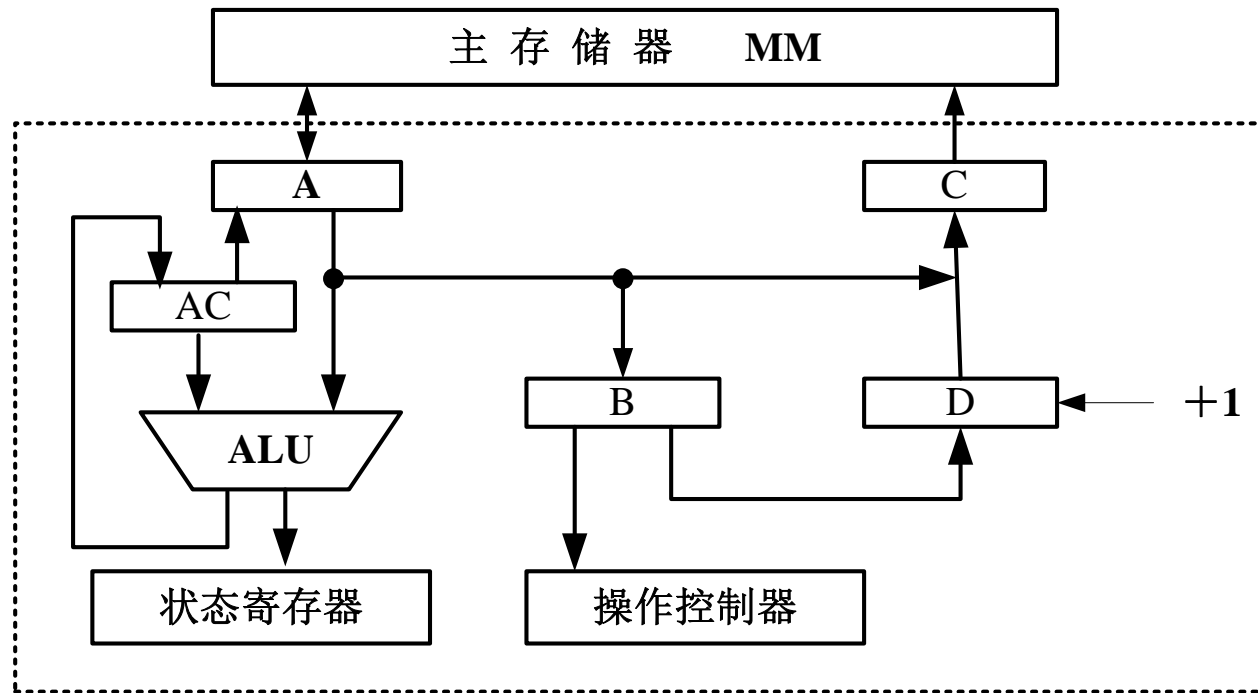
（2）每条指令平均包括5个CPU周期，每个CPU周期又包含4个主频周期，所以一条指令的执行时间 $=4 \times 5 \times 0.005\ \mu\text{s}=0.1\ \mu\text{s}$

该机平均指令执行速度 $=1/0.1=10\text{MIPS}$

(3) 说明指令的复杂程度会影响指令的平均执行速度。

例. CPU结构如图所示，其中包括一个累加寄存器AC、一个状态寄存器和其他四个寄存器，各部分之间的连线表示数据通路，箭头表示信息传送方向

(1) 标明图中四个寄存器的名称



解：A为MDR，B为IR，C为MAR，D为PC

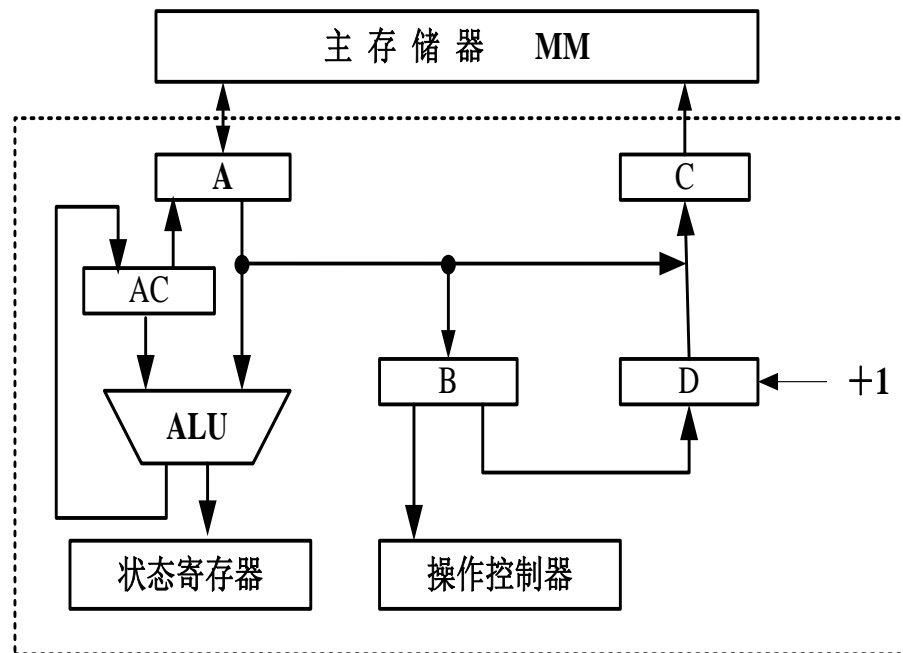


(2) 简述取指令的数据通路

(3) 简述完成指令LDA X的数据通路 (X为内存地址, LDA的功能为  $(X) \rightarrow (AC)$ )

(4) 简述完成指令ADDY的数据通路 (Y为内存地址, ADD功能为  $(AC) + (Y) \rightarrow (AC)$ )

(5) 简述完成指令STA Z的数据通路 (Z为内存地址, STA功能为  $(AC) \rightarrow (Z)$ )



解: (2)取指:  $PC \rightarrow MAR \rightarrow MM \rightarrow MDR \rightarrow IR$

(3) LDA X:  $X \rightarrow MAR \rightarrow MM \rightarrow MDR \rightarrow ALU \rightarrow AC$

(4) ADD Y:  $Y \rightarrow MAR \rightarrow MM \rightarrow MDR \rightarrow ALU \rightarrow ADD \rightarrow AC$

(5) STA Z:  $Z \rightarrow MAR, AC \rightarrow MDR \rightarrow MM$

# 微操作命令分析

设 CPU 内部采用总线连接方式,如图。

(1) 写出完成 LDA X, STA X, ADD X (X 均为主存地址) 三条指令所需的全部微操作, 并指出哪些控制信号有效。

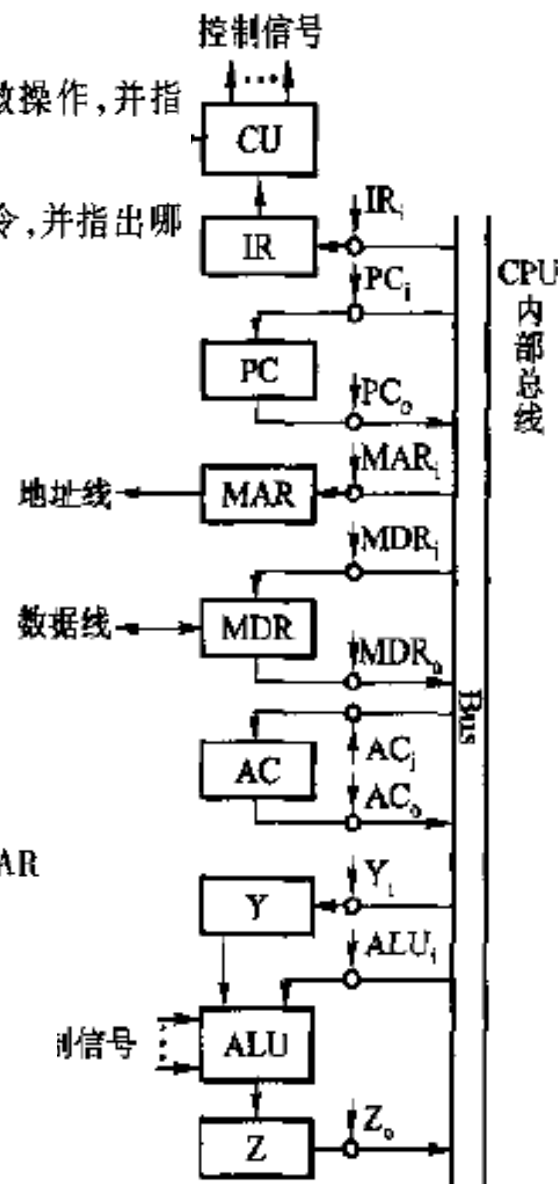
(2) 当上述三条指令均为间接寻址时, 写出完成这些指令所需的全部微操作命令, 并指出哪些控制信号有效。

(1) 上述三条指令的取指操作均相同, 即

PC → Bus → MAR	PC <sub>0</sub> 和 MAR <sub>1</sub> 有效, 现行指令地址 → MAR
1 → R	CU 发读命令
数据线 → MDR	现行指令从存储器 → 数据线 → MDR
MDR → Bus → IR	MDR <sub>0</sub> 和 IR <sub>1</sub> 有效, 现行指令 → IR
(PC) + 1 → PC	形成下条指令的地址

① LDA X 指令

Ad(MDR) → Bus → MAR	MDR <sub>0</sub> 和 MAR <sub>1</sub> 有效, 指令的地址码字段 → MAR
1 → R	CU 发读命令
数据线 → MDR	操作数从存储器 → 数据线 → MDR
MDR → Bus → AC	MDR <sub>0</sub> 和 AC <sub>1</sub> 有效, 操作数 → AC



## ② STA X 指令

$Ad(MDR) \rightarrow Bus \rightarrow MAR$

$1 \rightarrow W$

$AC \rightarrow Bus \rightarrow MDR$

$MDR \rightarrow \text{数据线}$

$MDR_i$  和  $MAR_i$  有效, 指令的地址码字段  $\rightarrow MAR$

CU 发写命令

$AC_i$  和  $MDR_i$  有效, 欲写入的数据  $\rightarrow MDR$

数据经数据线写入存储器

## ③ ADD X 指令

$Ad(MDR) \rightarrow Bus \rightarrow MAR$

$1 \rightarrow R$

数据线  $\rightarrow MDR$

$MDR \rightarrow Bus \rightarrow Y$

$(AC) + (Y) \rightarrow Z$

$Z \rightarrow AC$

$MDR_i$  和  $MAR_i$  有效, 指令的地址码字段  $\rightarrow MAR$

CU 发读命令

操作数从存储器  $\rightarrow$  数据线  $\rightarrow MDR$

$MDR_i$  和  $Y_i$  有效, 操作数  $\rightarrow Y$

$AC_i$  有效, CU 向 ALU 发加命令, 结果  $\rightarrow Z$

$Z_i$  和  $AC_i$  有效, 结果  $\rightarrow AC$

(2) 对于间接寻址的取数、存数和加法指令, 其取指周期的操作是不变的, 进入间址周期三条指令的间址操作均相同, 具体的微操作是:

$Ad(MDR) \rightarrow Bus \rightarrow MAR$

$1 \rightarrow R$

数据线  $\rightarrow MDR$

间址周期结束时有效地址在 MDR 中, 进入执行周期后, 三条指令的第一个微操作均为:

$MDR \rightarrow Bus \rightarrow MAR$

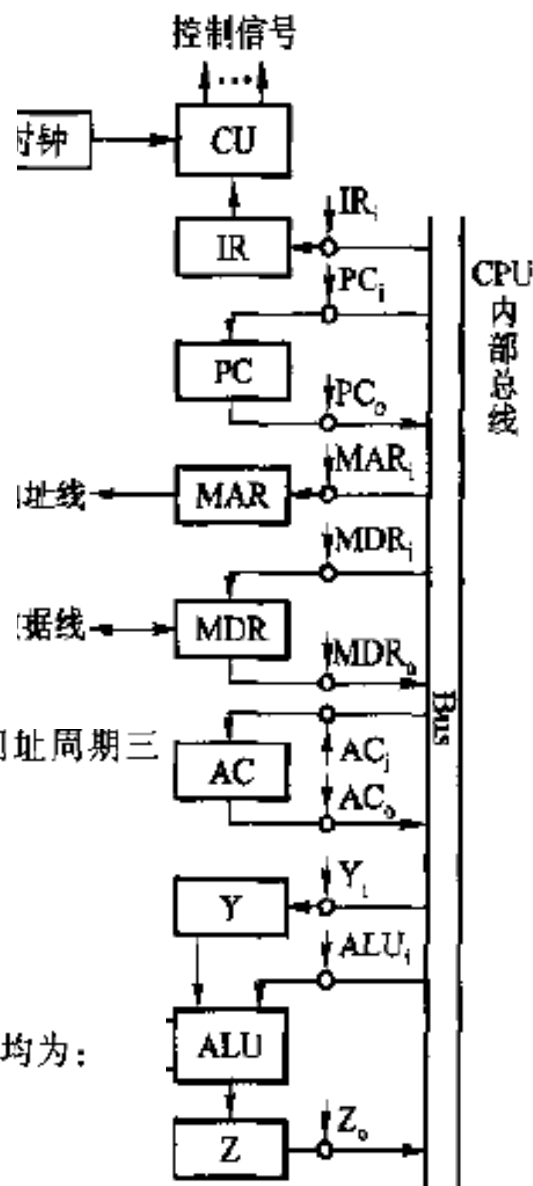
其余的微操作不变。

$MDR_i$  和  $MAR_i$  有效, 形式地址  $\rightarrow MAR$

CU 发读命令

有效地址从存储器  $\rightarrow$  数据线  $\rightarrow MDR$

$MDR_i$  和  $MAR_i$  有效, 有效地址  $\rightarrow MAR$



# 指令、机器和时钟周期

设某机主频为 8 MHz, 每个机器周期平均含 2 个时钟周期, 每条指令平均有 2.5 个机器周期, 试问该机的平均指令执行速度为多少 MIPS? 若机器主频不变, 但每个机器周期平均含 4 个时钟周期, 每条指令平均有 5 个机器周期, 则该机的平均指令执行速度又是多少 MIPS? 由此可得出什么结论?

【解】 根据主频为 8 MHz, 得时钟周期为  $1/8 = 0.125 \mu\text{s}$ , 机器周期为  $0.125 \times 2 = 0.25 \mu\text{s}$ , 指令周期为  $0.25 \times 2.5 = 0.625 \mu\text{s}$ 。

(1) 平均指令执行速度为  $1/0.625 = 1.6 \text{ MIPS}$ 。

(2) 若机器主频不变, 机器周期含 4 个时钟周期, 每条指令平均含 5 个机器周期, 则指令周期为  $0.125 \times 4 \times 5 = 2.5 \mu\text{s}$ , 故平均指令执行速度为  $1/2.5 = 0.4 \text{ MIPS}$ 。

(3) 可见机器的速度并不完全取决于主频。



# Thank You!

Computer Architecture Research Institute · Computer Organization

