



第 9 章 控制单元的功能

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

控制单元具有发出各种微操作命令序列的功能。

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)

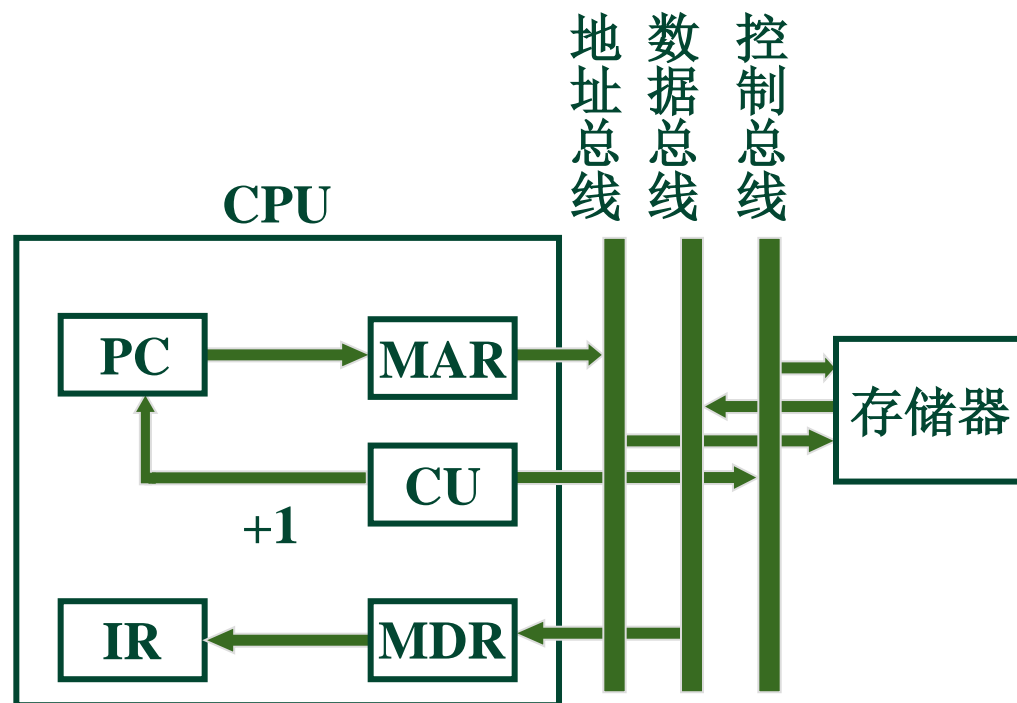
$M(MAR) \rightarrow MDR$

(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

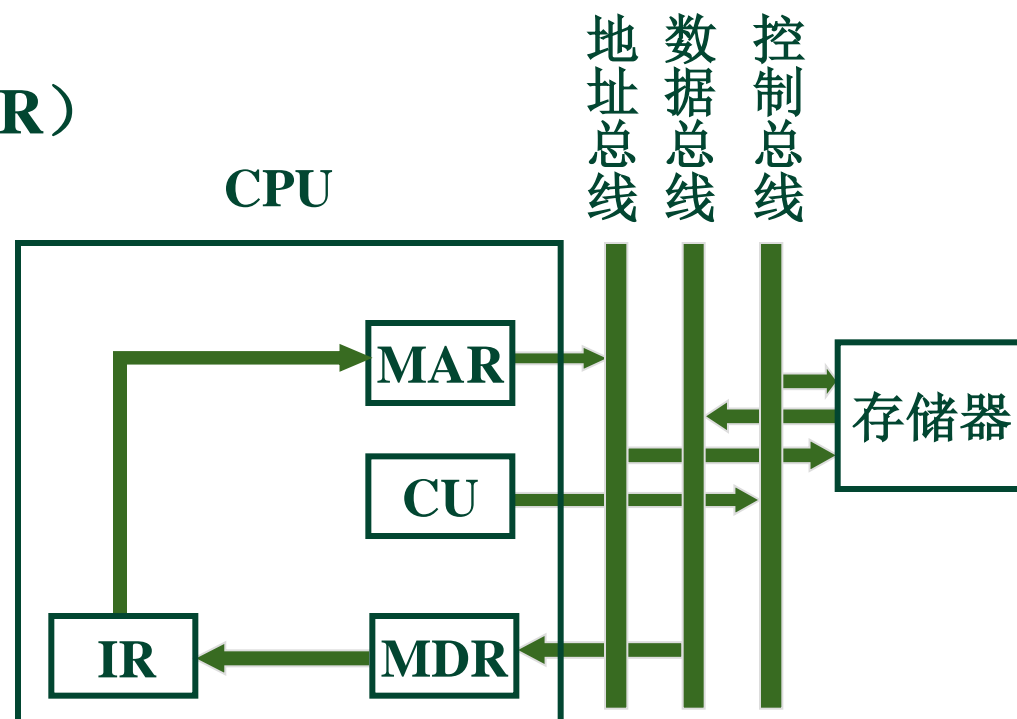
$Ad(IR) \rightarrow MAR$

(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

$0 \rightarrow \text{MAR}$

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

$1 \rightarrow \text{W}$

$1 \rightarrow \text{W}$

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

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

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

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

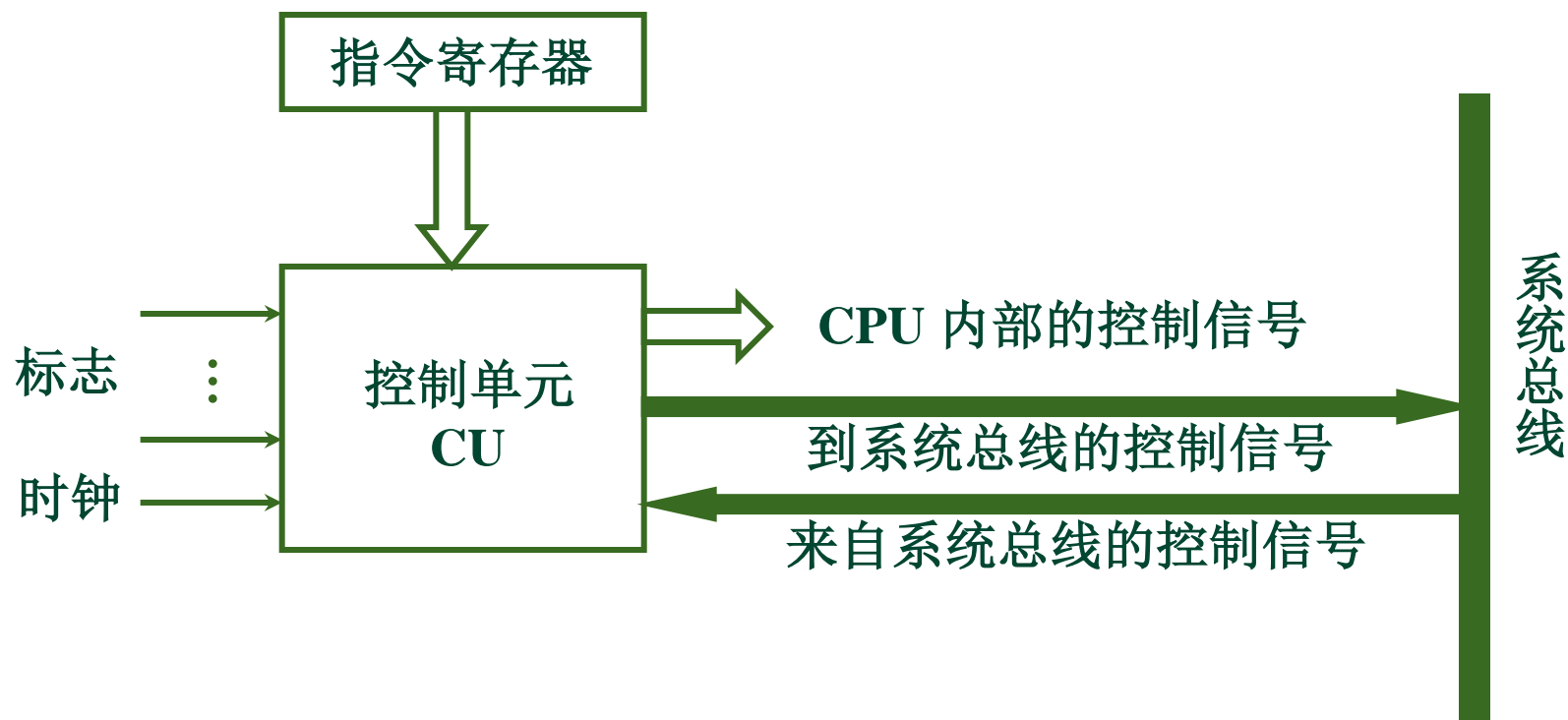
中断识别程序入口地址 $\text{M} \rightarrow \text{PC}$

$0 \rightarrow \text{EINT}$ (置 “0”)

$0 \rightarrow \text{EINT}$ (置 “0”)

9.2 控制单元的功能

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

$$\mathbf{R}_i \rightarrow \mathbf{R}_j$$
$$(\mathbf{PC}) + 1 \rightarrow \mathbf{PC}$$

ALU +、-、与、或 ……

MREQ

访存控制信号

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

访 IO/ 存储器的控制信号

RD

读命令

WR

写命令

INTA

中断响应信号

HLDA

总线响应信号

GND	1	40	VCC
AD14	2	39	AD15
AD13	3	38	A16/S3
AD12	4	37	A17/S4
AD11	5	36	A18/S5
AD10	6	35	A19/S6
AD9	7	34	BHE*/S7
AD8	8	33	MN / MX*
AD7	9	32	RD*
AD6	10	31	HOLD (RQ*/GT0*)
AD5	11	30	HLDA (RQ1*/GT1*)
AD4	12	29	WR* (LOCK*)
AD3	13	28	M / IO* (S2*)
AD2	14	27	DT / R* (S1*)
AD1	15	26	DEN (S0)
AD0	16	25	ALE
NMI	17	24	INTA
INTR	18	23	TEST*
CLK	19	22	READY
GND	20	21	RESET

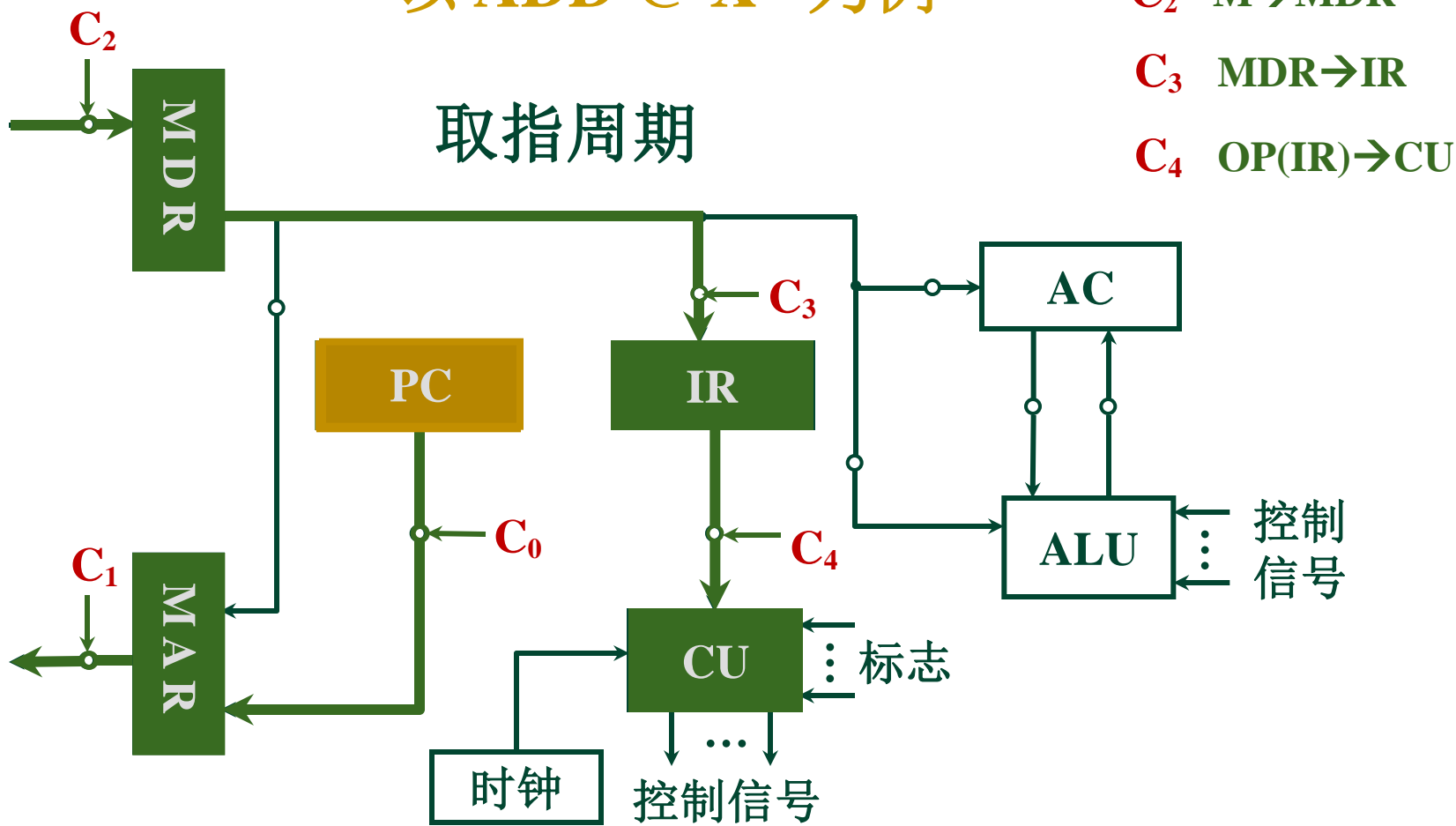
*表示低电平有效

二、控制信号举例

9.2

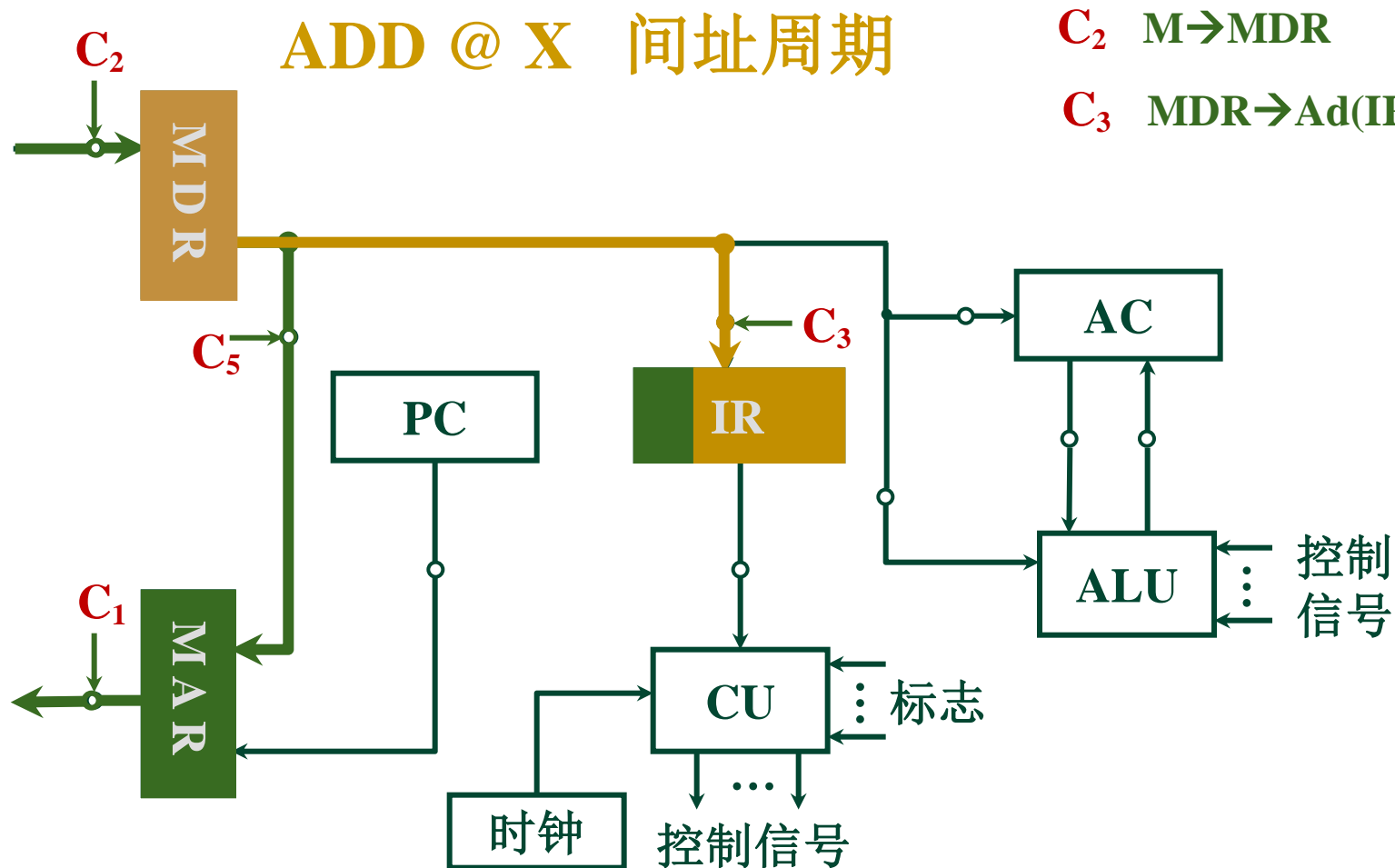
1. 不采用 CPU 内部总线的方式

以 **ADD @ X** 为例



二、控制信号举例

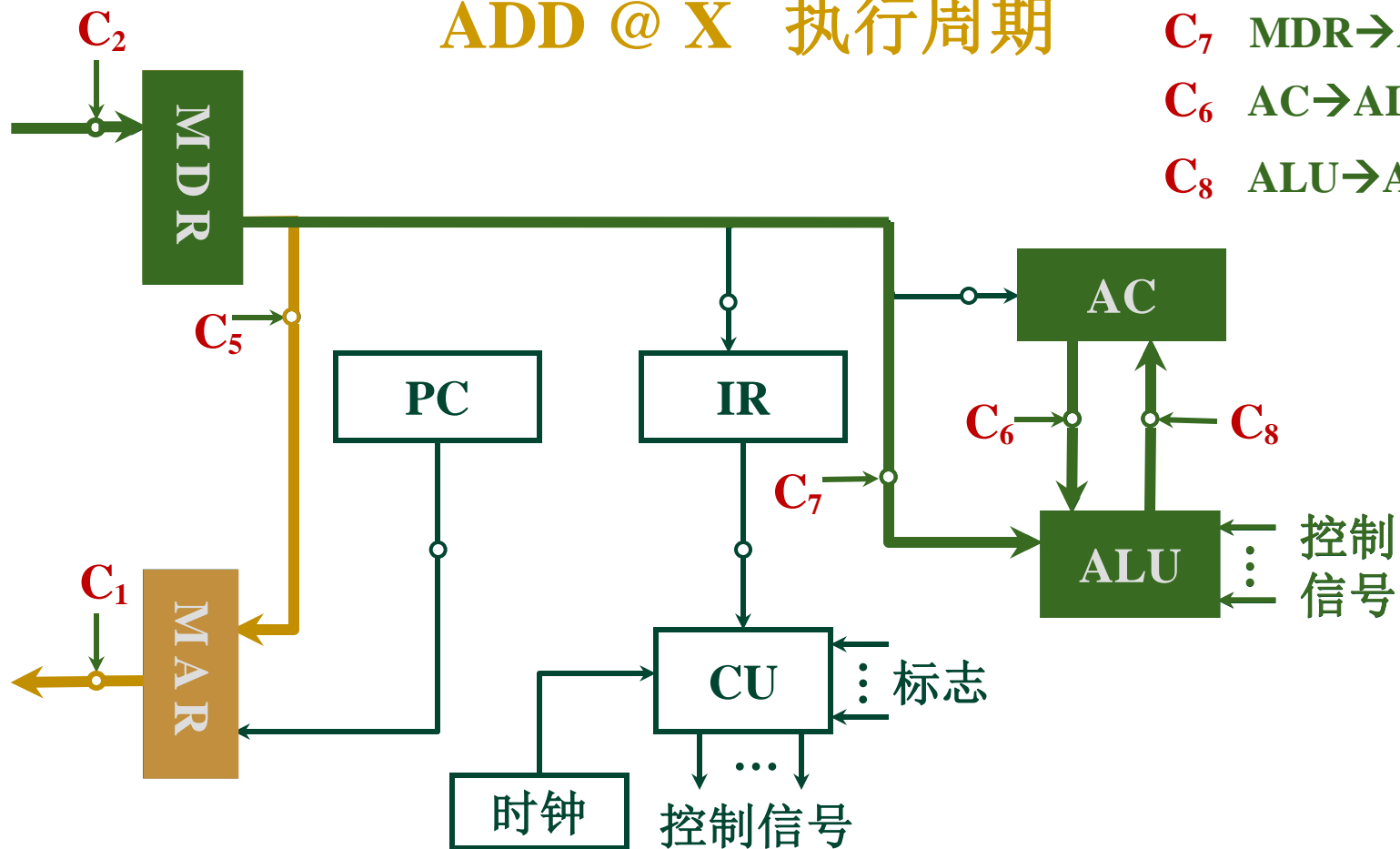
1. 不采用 CPU 内部总线的方式



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



C₅ MDR→MAR

C₁ MAR→M

C₂ M→MDR

C₇ MDR→ALU

C₆ AC→ALU

C₈ ALU→AC

2. 采用 CPU 内部总线方式

(1) ADD @ X 取指周期

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

$PC_o \quad MAR_i$

• CU 发读命令 $1 \rightarrow R$

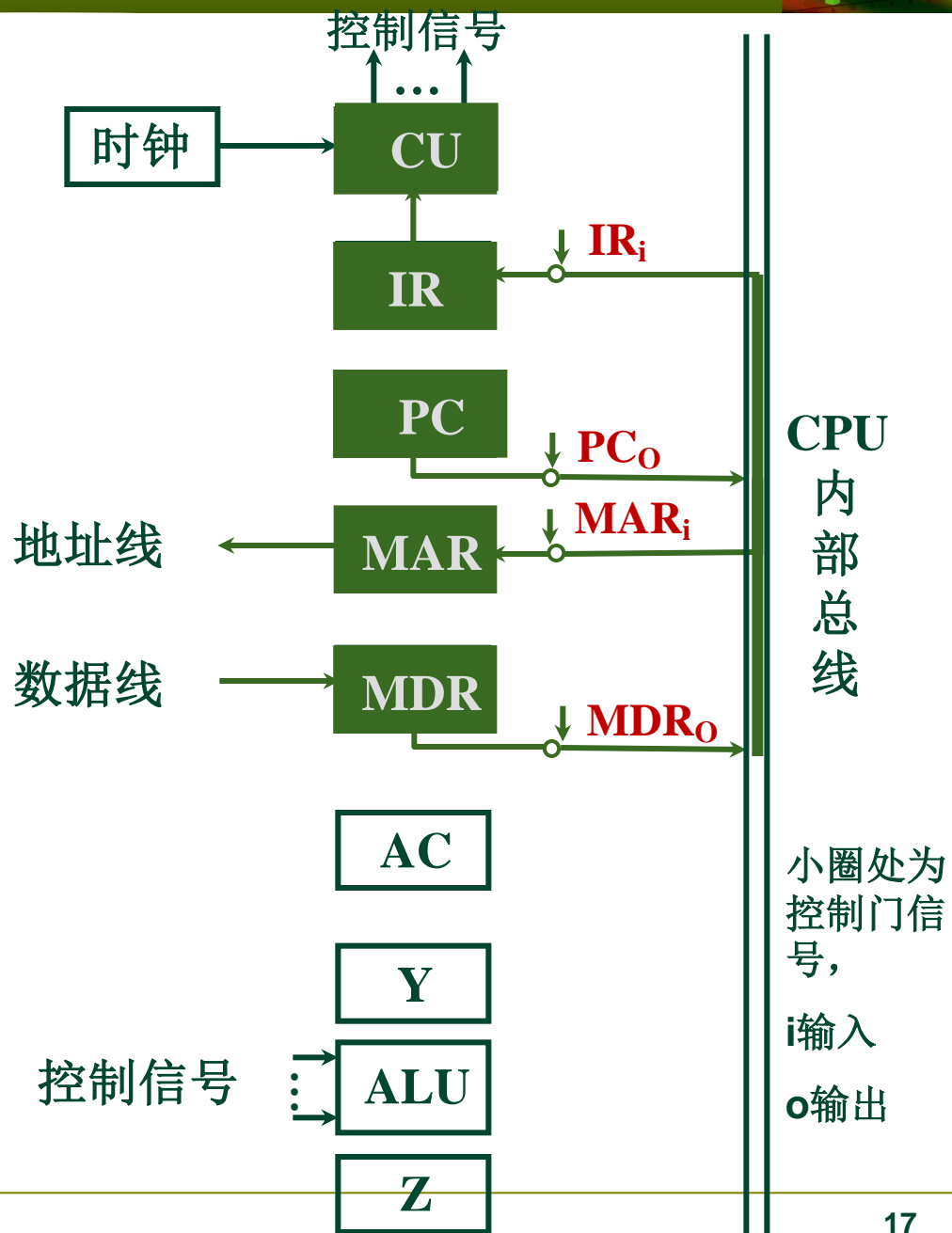
• 数据线 $\rightarrow MDR$

• $MDR \rightarrow IR$

$MDR_o \quad IR_i$

• $OP(IR) \rightarrow CU$

• $(PC) + 1 \rightarrow PC$



(2) ADD @ X 间址周期

形式地址 \rightarrow MAR

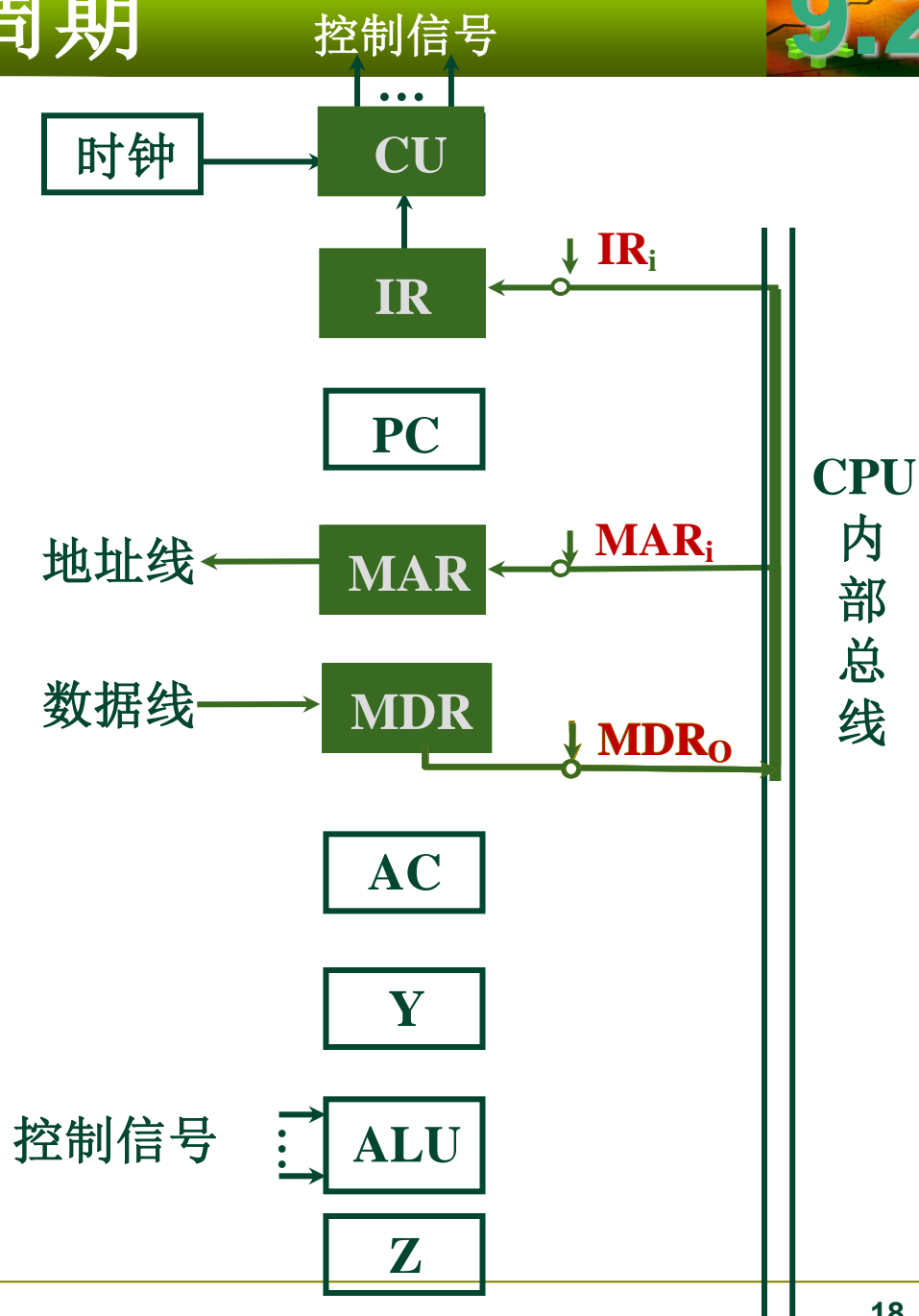
• $\text{MDR} \rightarrow \text{MAR} \rightarrow \text{地址线}$
 $\text{MDR}_0 \quad \text{MAR}_i$

• $1 \rightarrow R$

• 数据线 \rightarrow MDR

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

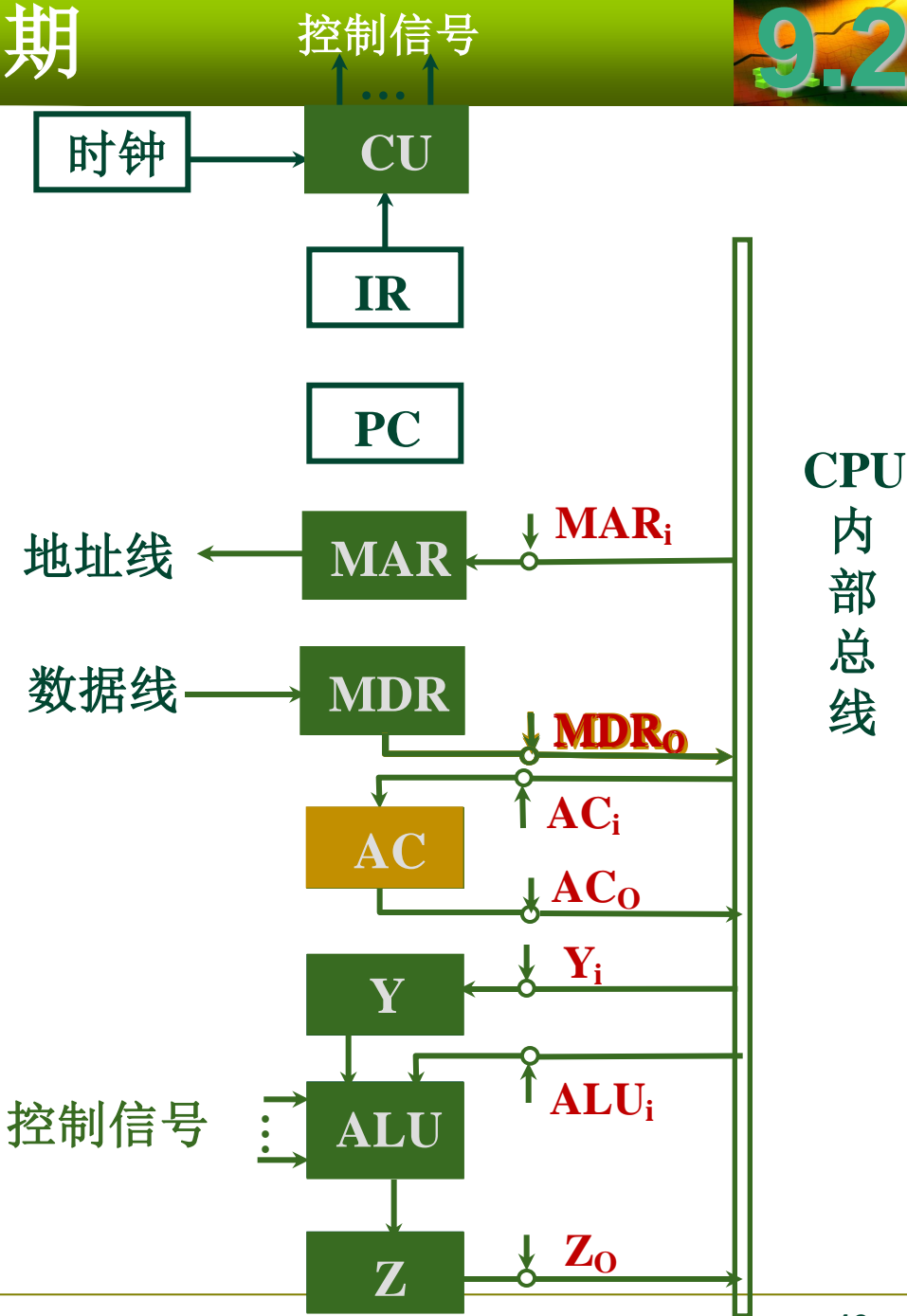
有效地址 $\rightarrow \text{Ad}(\text{IR})$



(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 $\rightarrow \text{MDR}$
- $\text{MDR} \rightarrow Y \rightarrow \text{ALU}$
 $\text{MDR}_0 \quad Y_i$
- $\text{AC} \rightarrow \text{ALU}$
 $\text{AC}_0 \quad \text{ALU}_i$
- $(\text{AC}) + (Y) \rightarrow Z$
- $Z \rightarrow \text{AC}$
 $Z_0 \quad \text{AC}_i$



三、多级时序系统



1. 机器周期(CPU周期)

教材P385

(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

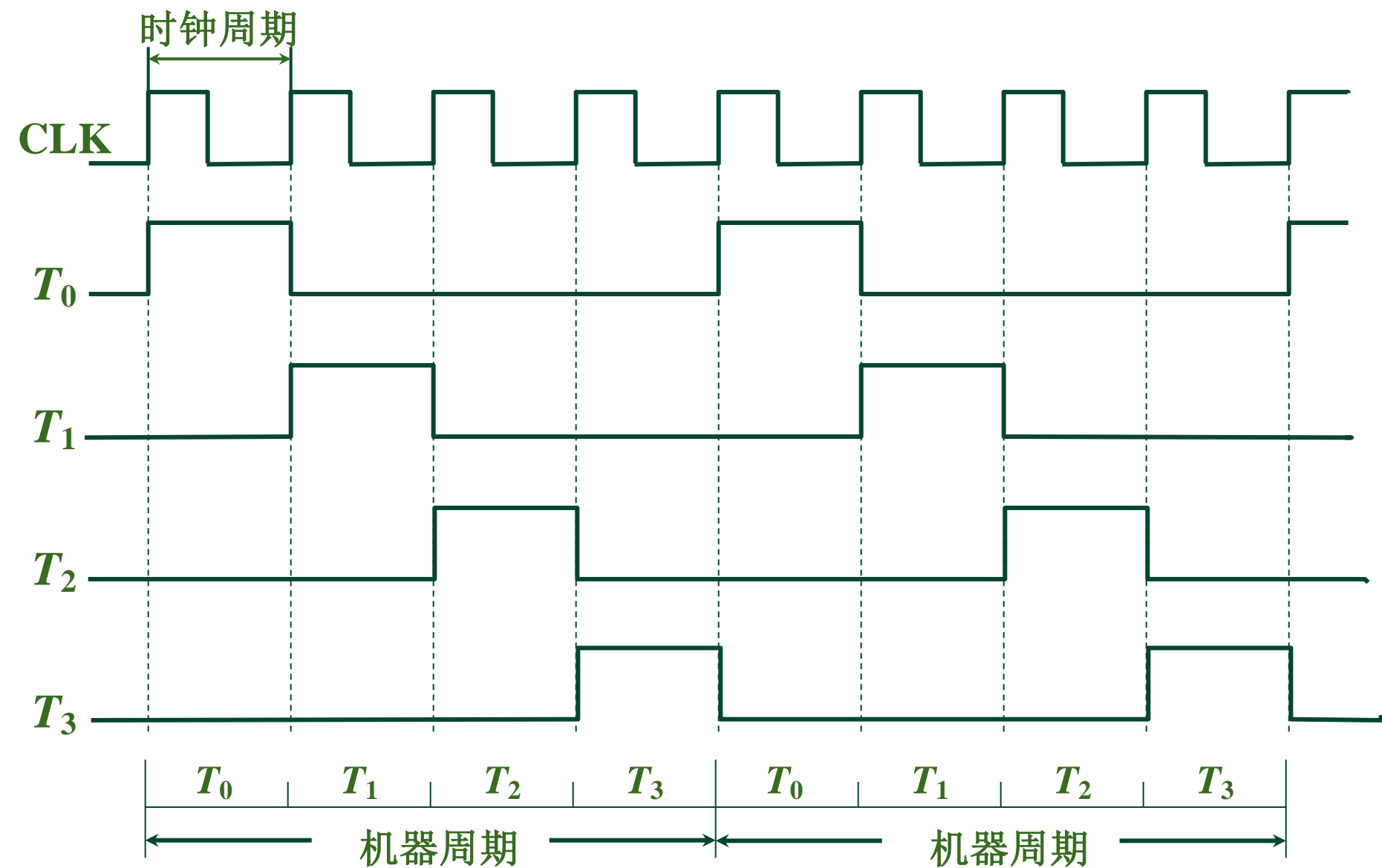
2. 时钟周期（节拍、状态）

- 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

时钟周期（节拍、状态）

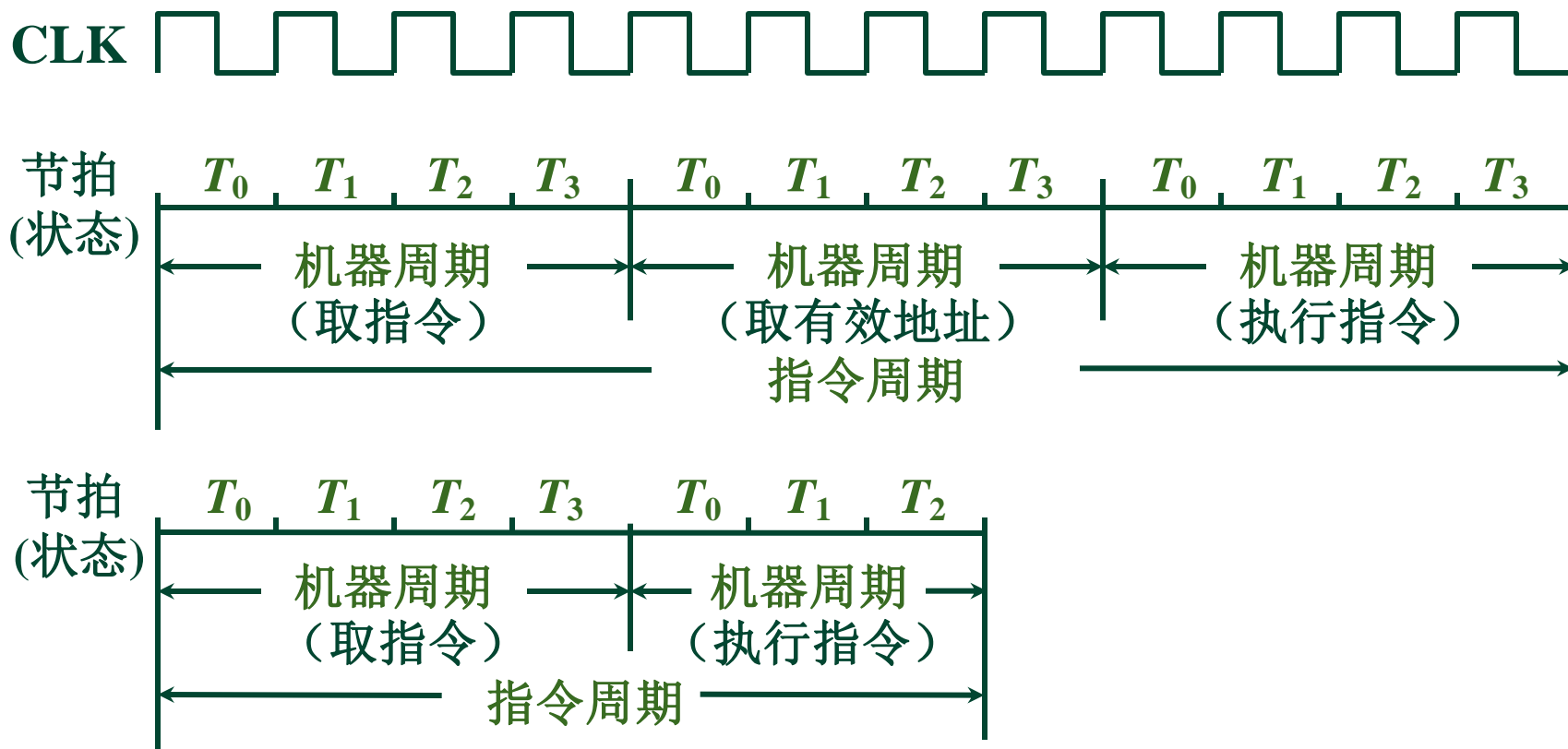


3. 多级时序系统

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



3. 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
- 三级时序系统是**小型机**常用的时序系统，在机器周期间、节拍电位间、工作脉冲间既不允许有重叠交叉，也不允许有空隙，应该是一个接一个的准确连接。
- **机器周期-节拍-脉冲** 三级时序系统。

4. 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

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

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

几个周期概念

- ❖ **时钟周期**：一个时钟脉冲所需要的时间。在计算机组成原理中又叫T周期或节拍脉冲，是**CPU**和其他单片机的基本时间单位。
- ❖ **机器周期（CPU周期）**：完成一个基本操作所需要的时间，如取指周期，执行周期。（取指令、存储器读、存储器写等，这每一项工作称为一个基本操作。）
- ❖ **指令周期**：是执行一条指令所需要的时间，一般由若干个机器周期组成。通常含一个机器周期的指令称为单周期指令，包含两个机器周期的指令称为双周期指令。
- ❖ **总线周期**：**CPU**从内存中读取指令，向内存中存取数据，对外设端口读写数据，执行总线周期。总线周期通常包含4个T状态：**T1**，**T2**，**T3**，**T4**。所谓一个T状态就是一个时钟周期。
- ❖ **存取周期**：指的是计算机进行一次访存所需要的时间，包括读写时间以及物理的恢复等待时间。
- ❖ 而总线周期是指**CPU**通过总线和存储器或**I/O**接口进行一次数据传输所需要的时间。一般情况下，存取周期都会包含有一个或多个的总线周期。
- ❖ 指令周期、总线周期和时钟周期之间的关系：一个指令周期由若干个总线周期组成，而一个总线周期时间又包含有若干个时钟周期。
- ❖ 一个总线周期包含一个（只有取址周期）或多个机器周期。

例9.2 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，等待周期=节拍周期），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= $(1/240) \mu s$

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：

$$\begin{aligned} \text{指令周期} &= 0.05\mu s + 4 \times (1/240)\mu s = (1/20)\mu s + (1/60)\mu s \\ &= (4/60)\mu s \end{aligned}$$

$$\text{机器平均速度} = 60/4 \approx 15 \text{ MIPS}$$

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

例9.3 若某机主频为**400MHZ**，每个指令周期平均为**2.5CPU**周期，每个**CPU**周期平均包括**4**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2个CPU周期，每个CPU周期又包含3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**400MHz**，所以主频周期= $1/400\text{MHz}=0.0025\mu\text{s}$

每个指令周期平均为**2.5CPU**周期，每个**CPU**周期平均包括**4**个主频周期，所以一条指令的执行时间= $2.5 \times 4 \times 0.0025\mu\text{s}=0.025\mu\text{s}$

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**2个CPU**周期，每个**CPU**周期又包含**3个主频**周期，所以一条指令的执行时间= $2 \times 3 \times 0.0025\mu\text{s}=0.015\mu\text{s}$

该机平均指令执行速度= $1/0.015=66.67\text{MIPS}$

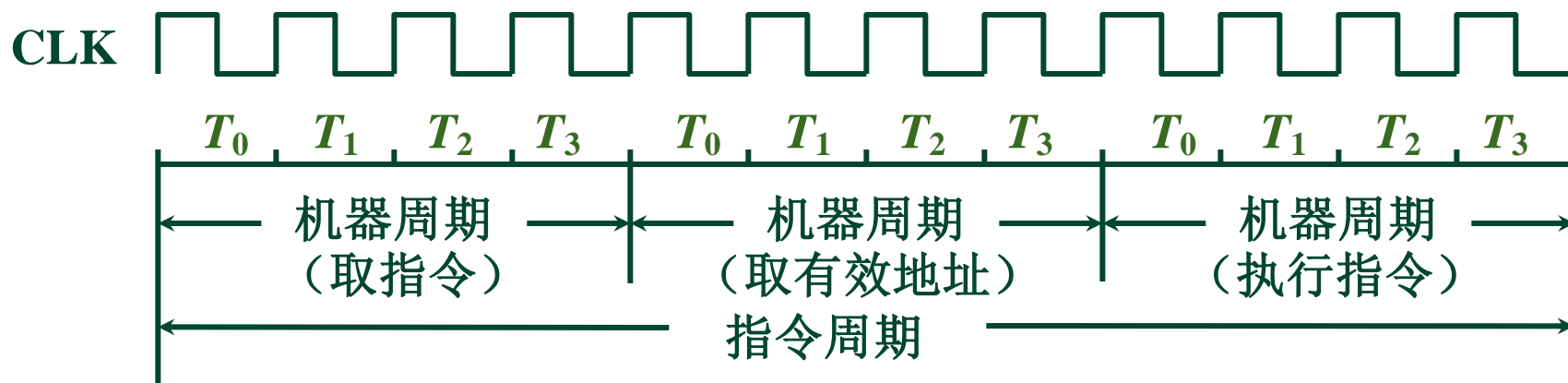
结论：指令的复杂程度会影响指令的平均执行速度。

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

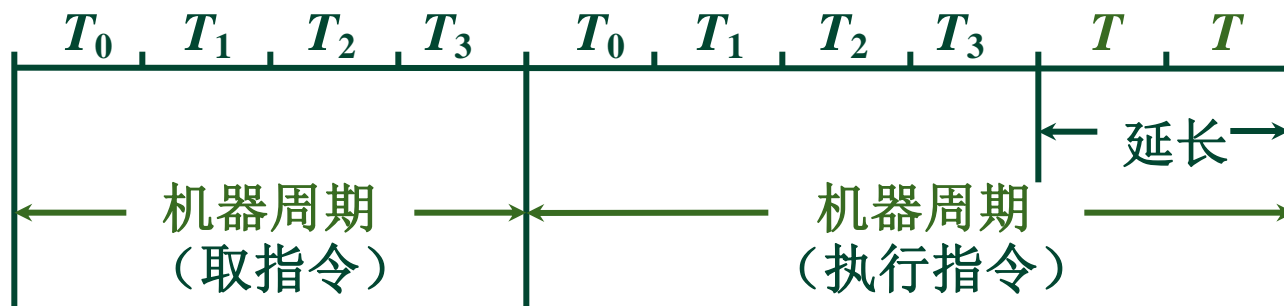
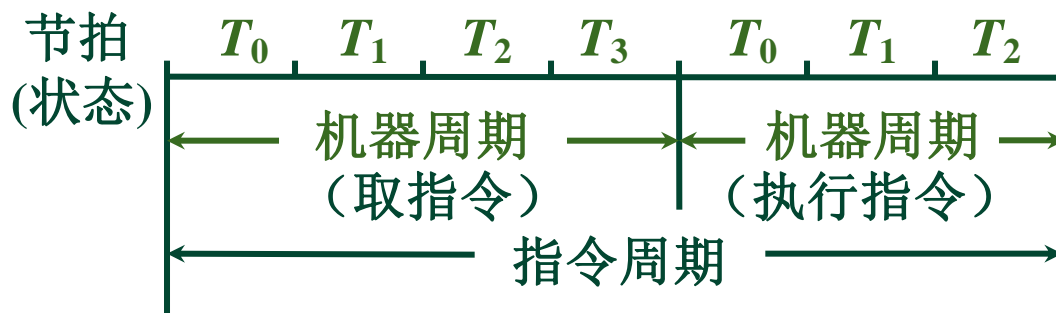
以 **最长** 的 **微操作序列** 和 **最繁** 的微操作作为 **标准**

机器周期内 **节拍数相同**

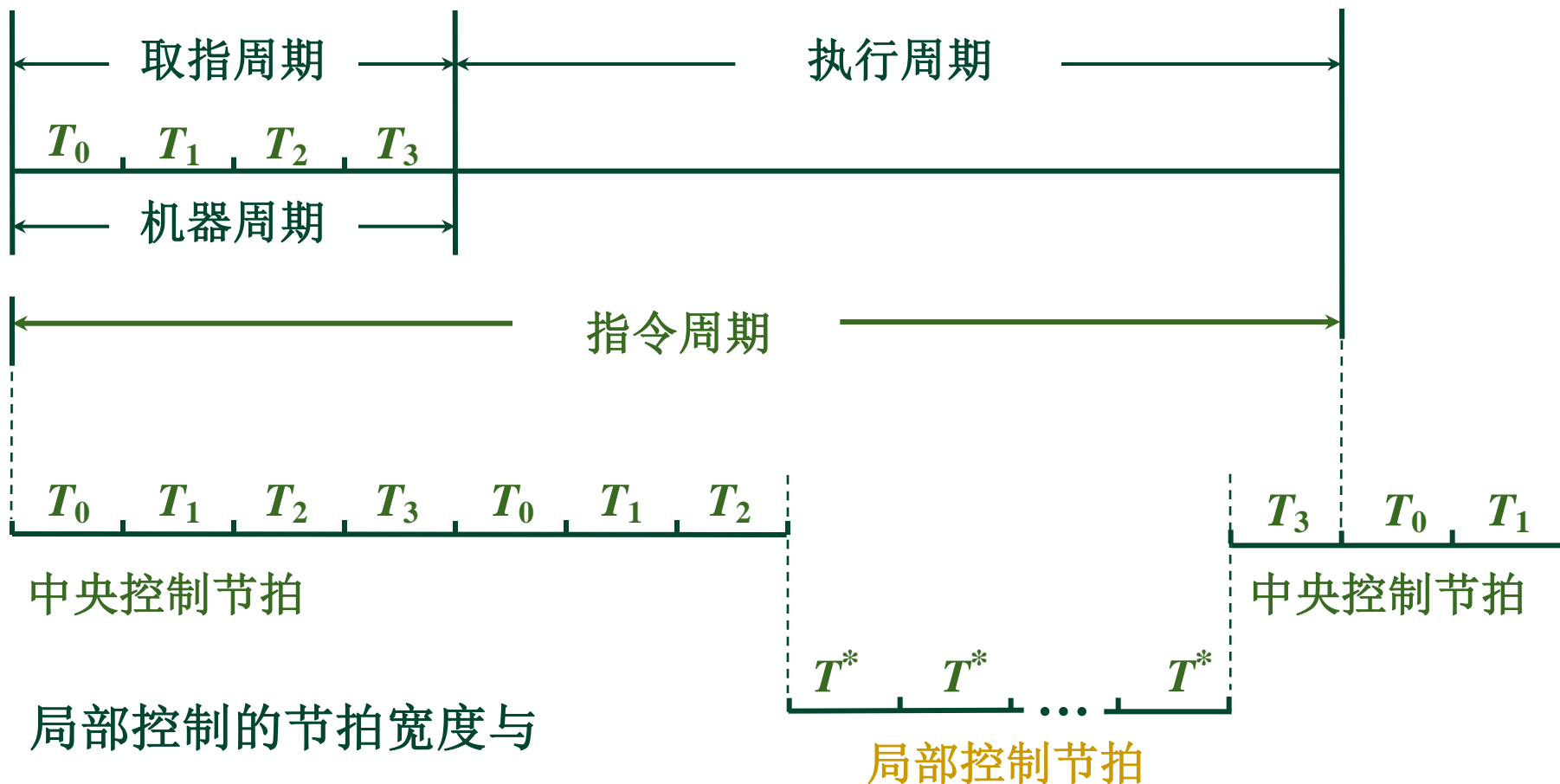
(2) 采用不定长的机器周期

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



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

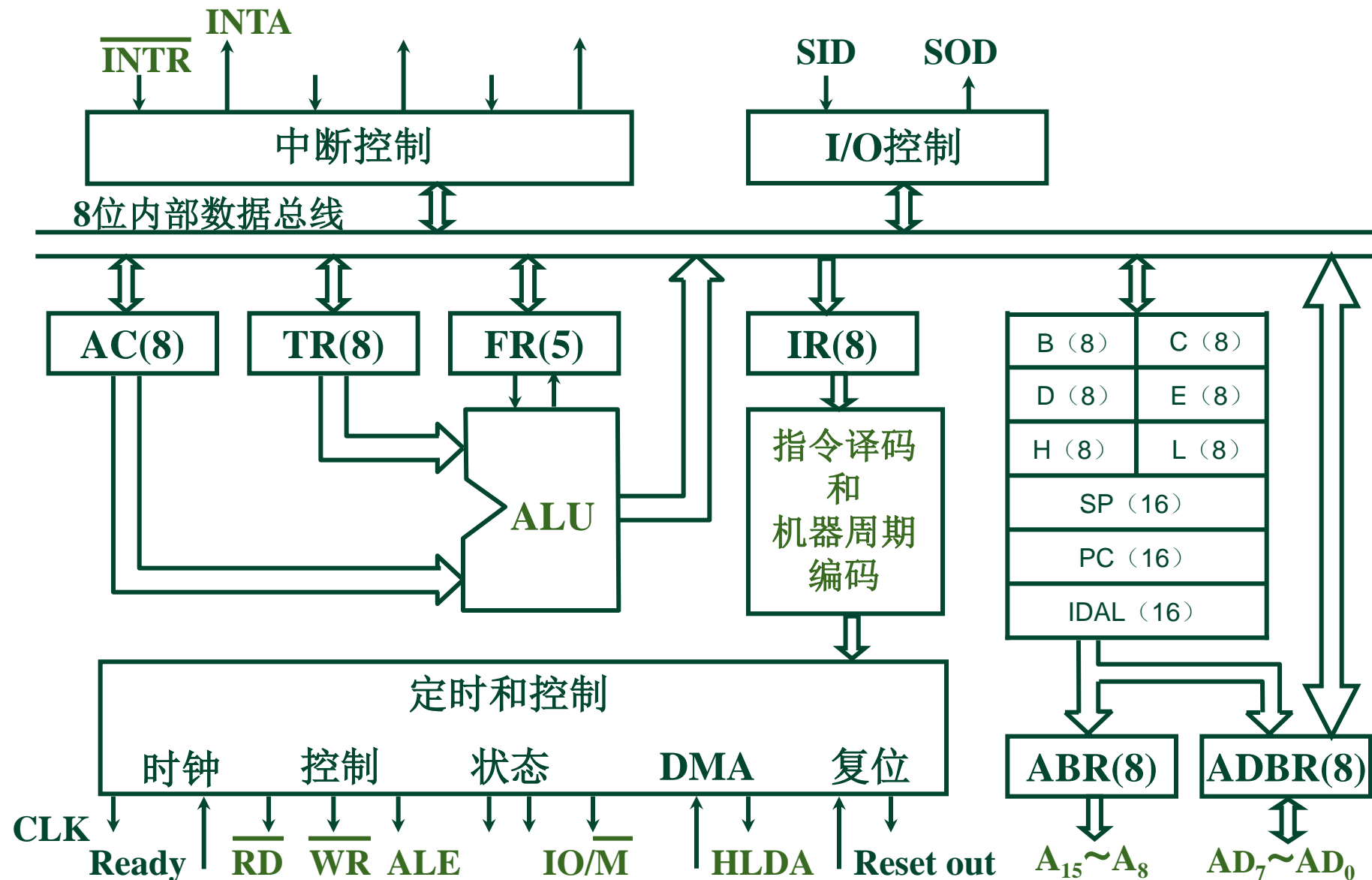
(2) 连续 和 单条 指令执行转换开关

(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	□	S_1
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 INTA

Trap 重新启动中断

(5) CPU 初始化

入 $\overline{\text{Reset in}}$

出 Reset out

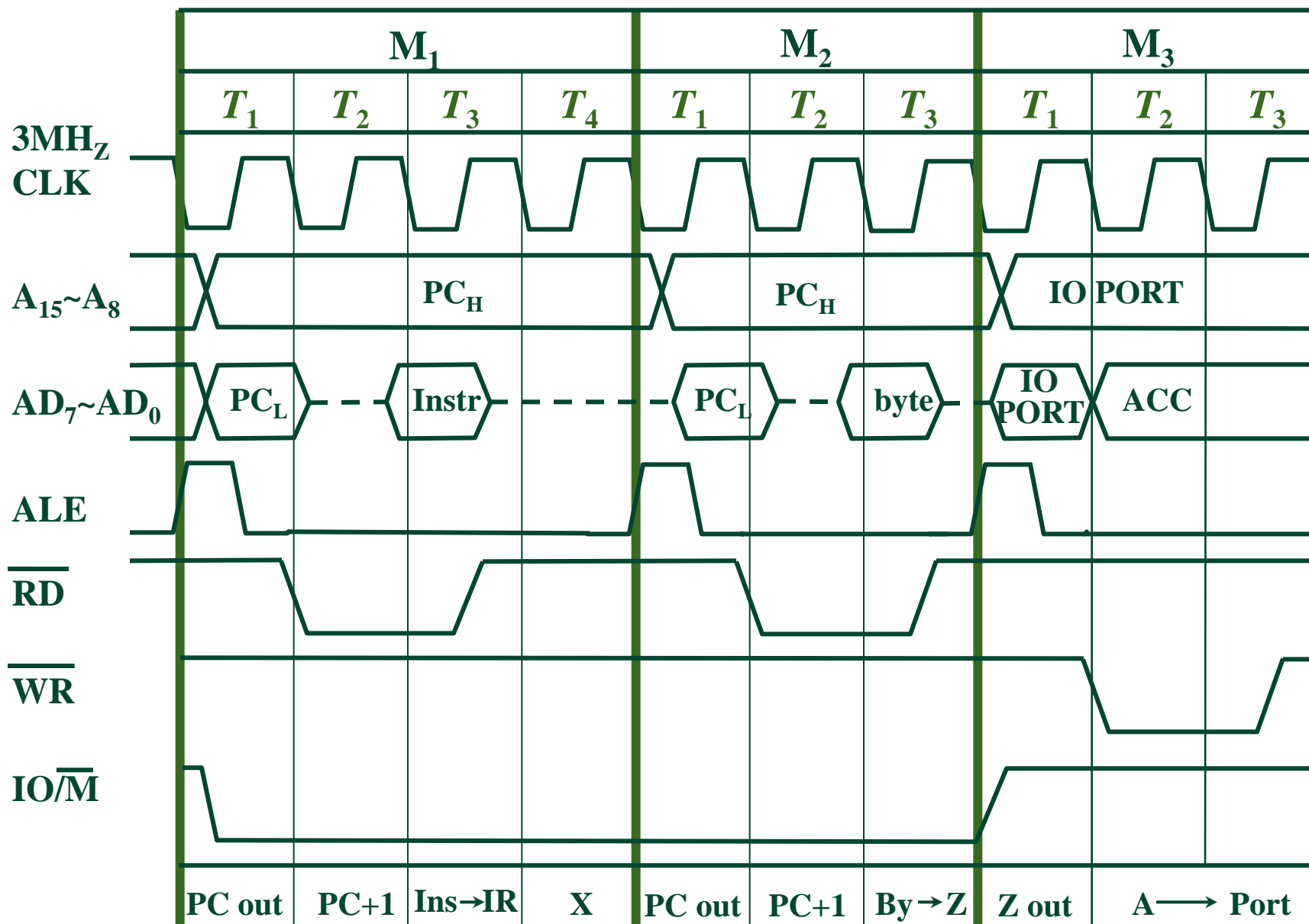
(6) 电源和地

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	$\overline{\text{WR}}$
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

3. 机器周期和节拍（状态）与控制信号的关系





Thank You!

