

第九章 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

9.1 操作命令的分析

一、取指周期

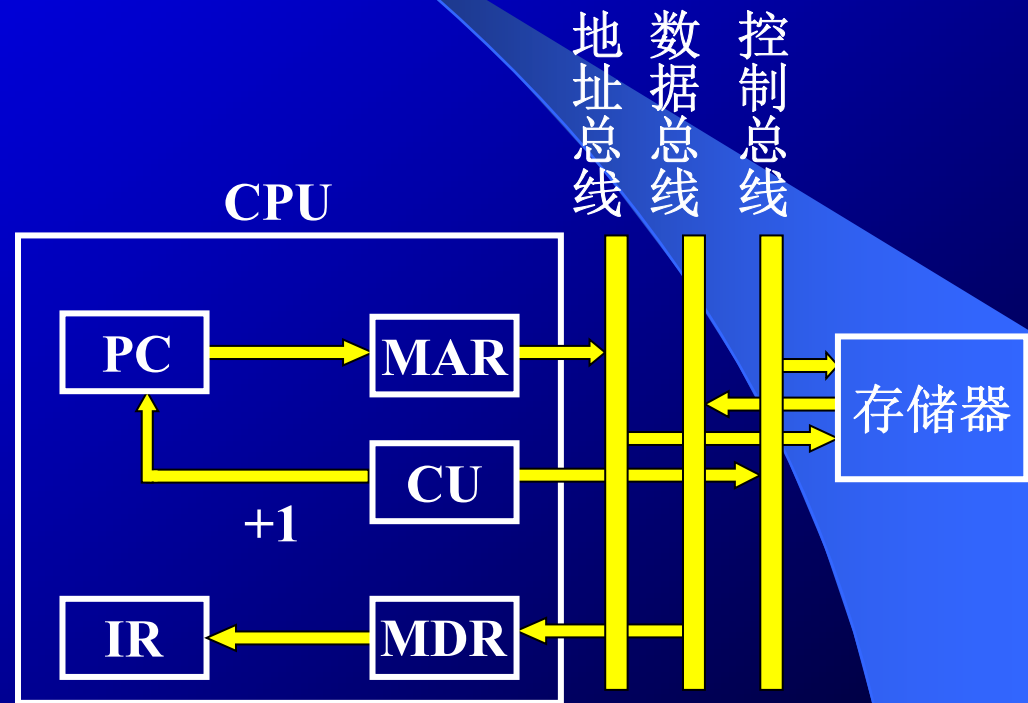
$PC \rightarrow MAR \rightarrow \text{地址线}$

$1 \rightarrow R$

$M(MAR) \rightarrow MDR$

$MDR \rightarrow IR$

$(PC) + 1 \rightarrow PC$

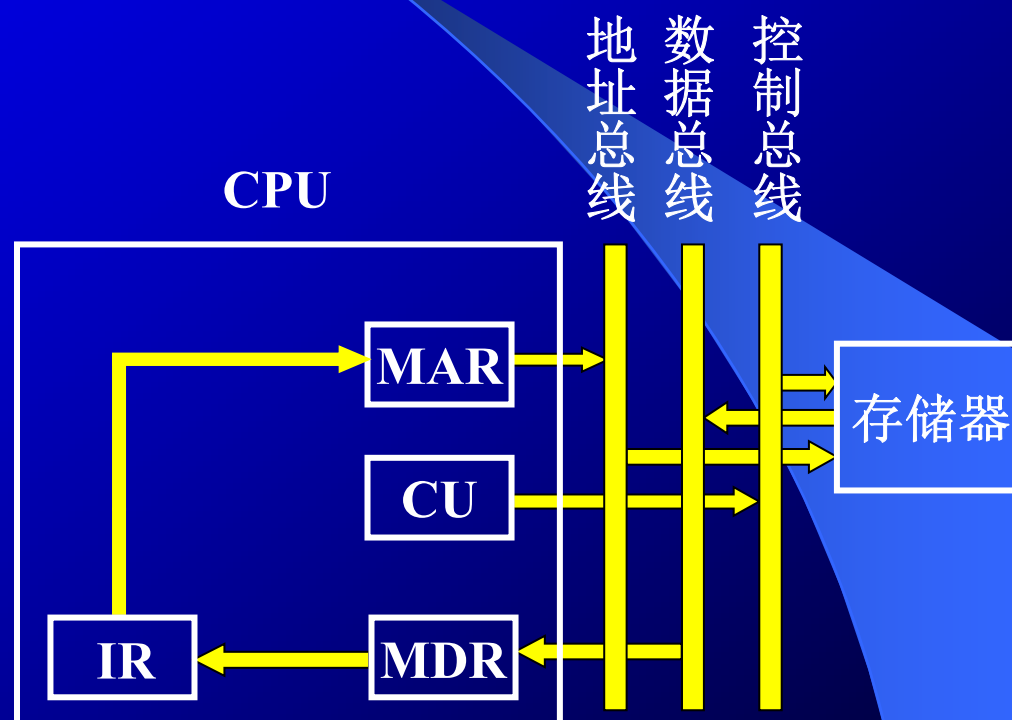


9.1

Ad (IR) \longrightarrow MAR

M (MAR) \longrightarrow MDR

MDR \longrightarrow Ad (IR)



三、执行周期

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 $0 \rightarrow \text{G}$

2. 访存指令

9.1

(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

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

$1 \rightarrow \text{W}$

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

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

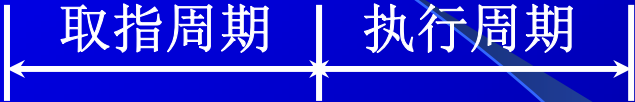
(3) 取数指令 **LDA X**
$$\text{Ad}(\text{IR}) \rightarrow \text{MAR}$$
$$1 \rightarrow \text{R}$$
$$\text{M}(\text{MAR}) \rightarrow \text{MDR}$$
$$\text{MDR} \rightarrow \text{ACC}$$

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)
$$\text{A}_0 \cdot \text{Ad}(\text{IR}) + \overline{\text{A}_0}(\text{PC}) \rightarrow \text{PC}$$

4. 三类指令的指令周期

非访存 指令周期



取指周期 | 执行周期

直接访存 指令周期



取指周期 | 执行周期

间接访存 指令周期



取指周期 | 间址周期 | 执行周期

转移 指令周期



取指周期 | 执行周期

四、中断周期

9.1

程序断点存入 “0” 地址 程序断点 进栈

$0 \rightarrow \text{MAR}$

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

$1 \rightarrow \text{W}$

$1 \rightarrow \text{W}$

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

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

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

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

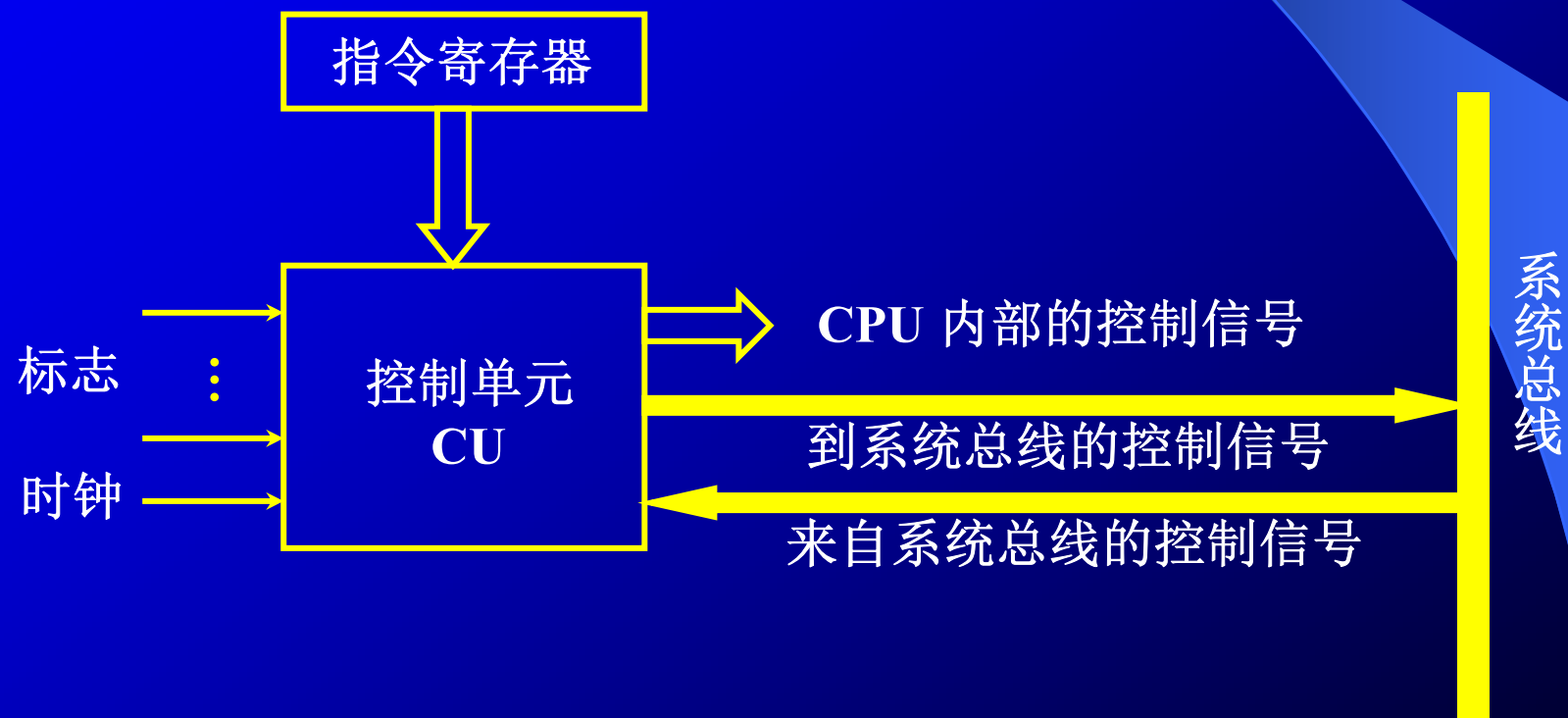
中断识别程序入口地址 $\text{M} \rightarrow \text{PC}$

$0 \rightarrow \text{EINT}$ (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

(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

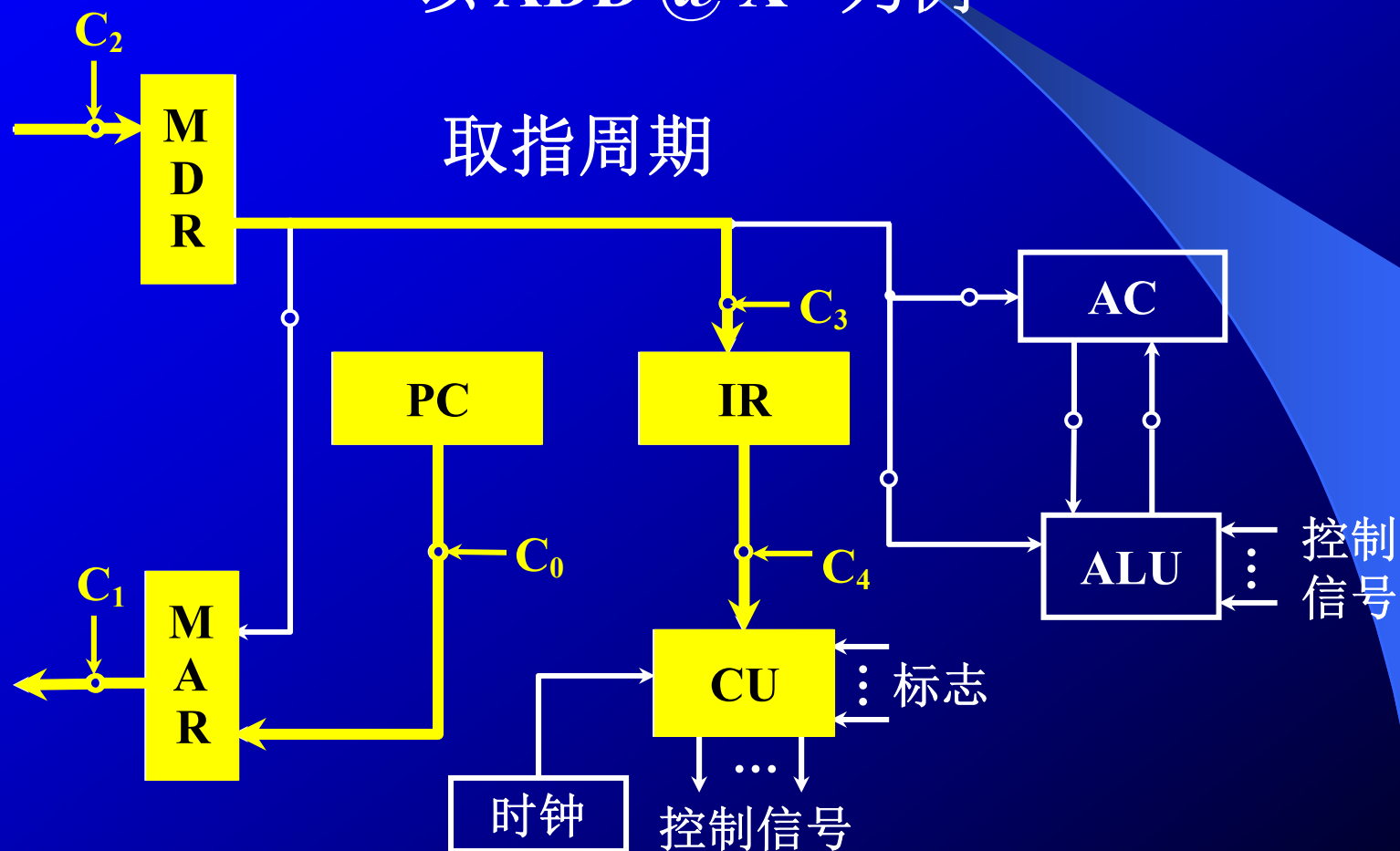
总线响应信号

二、控制信号举例

9.2

1. 不采用 CPU 内部总线的方式

以 $\text{ADD } @X$ 为例

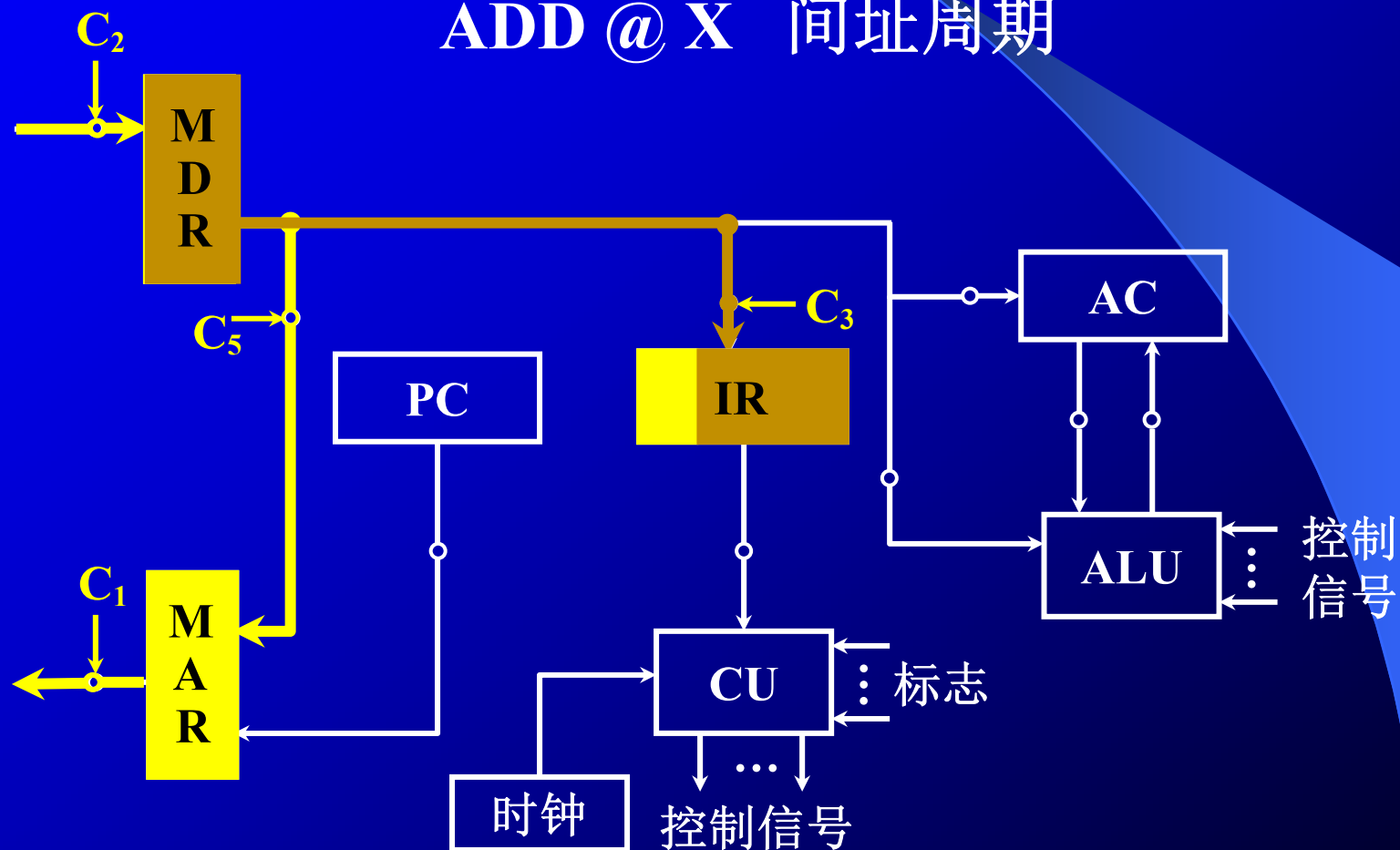


二、控制信号举例

9.2

1. 不采用 CPU 内部总线的方式

ADD @ X 间址周期

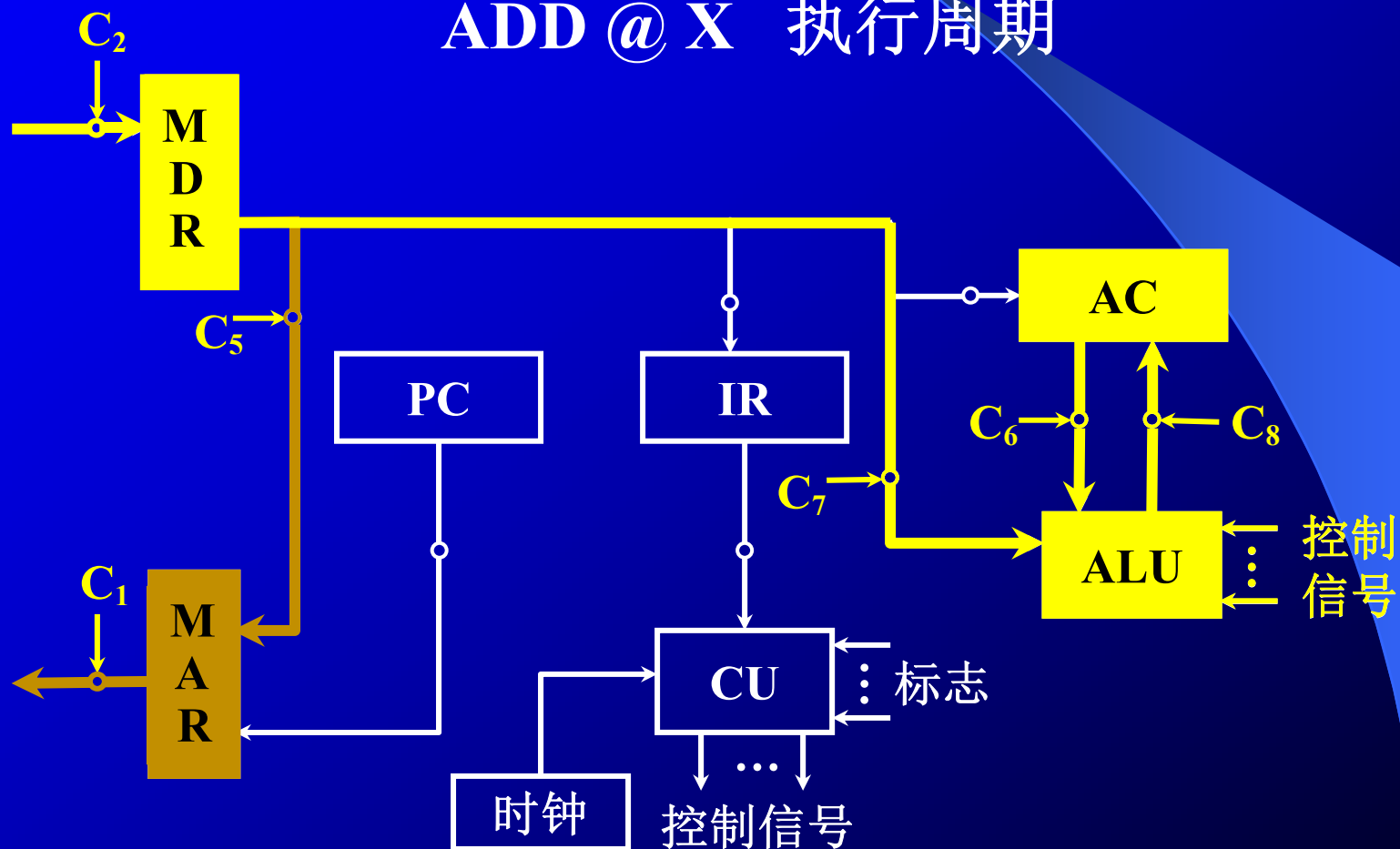


二、控制信号举例

9.2

1. 不采用 CPU 内部总线的方式

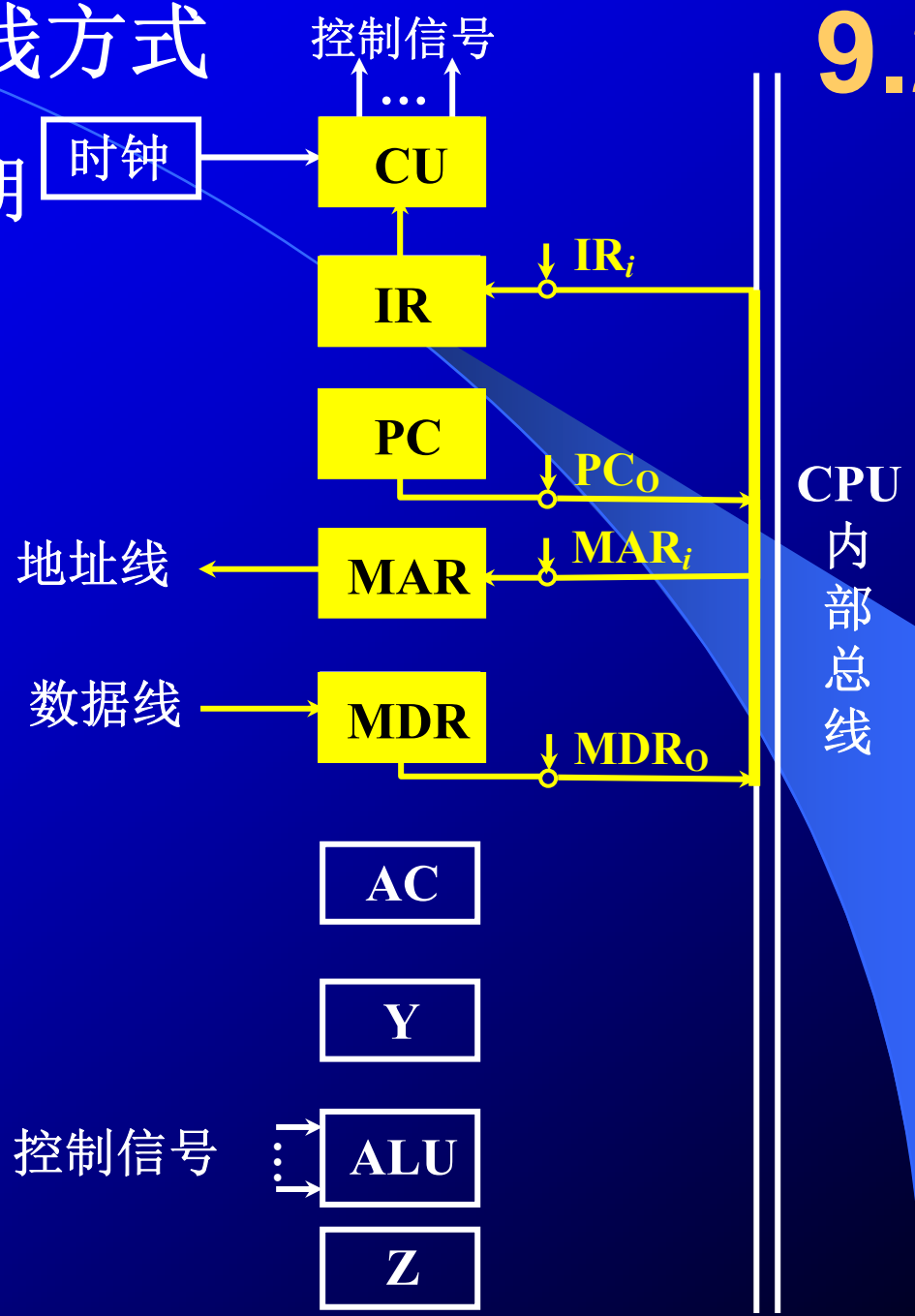
ADD @X 执行周期



2. 采用 CPU 内部总线方式

(1) ADD @ X 取指周期

- PC \rightarrow MAR \rightarrow 地址线
PC₀ MAR_i
- CU 发读命令 1 \rightarrow R
- 数据线 \rightarrow MDR
- MDR \rightarrow IR
MDR₀ IR_i
- OP (IR) \rightarrow CU
- (PC) + 1 \rightarrow PC



9.2

(2) ADD @ X 间址周期

形式地址 \rightarrow MAR

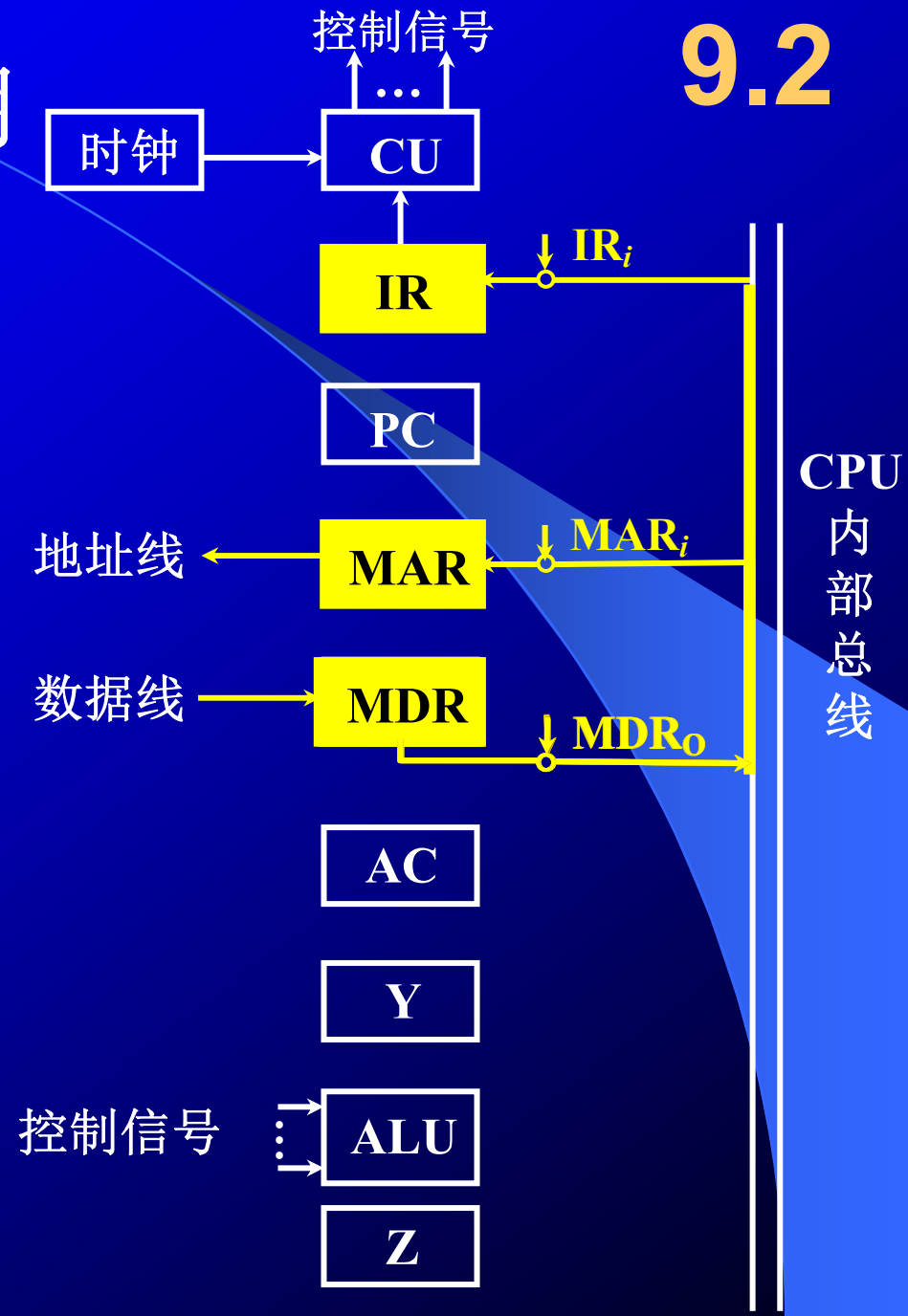
• $\text{MDR} \rightarrow \text{MAR} \rightarrow \text{地址线}$
 $\text{MDR}_0 \quad \text{MAR}_i$

• $1 \rightarrow R$

• 数据线 \rightarrow MDR

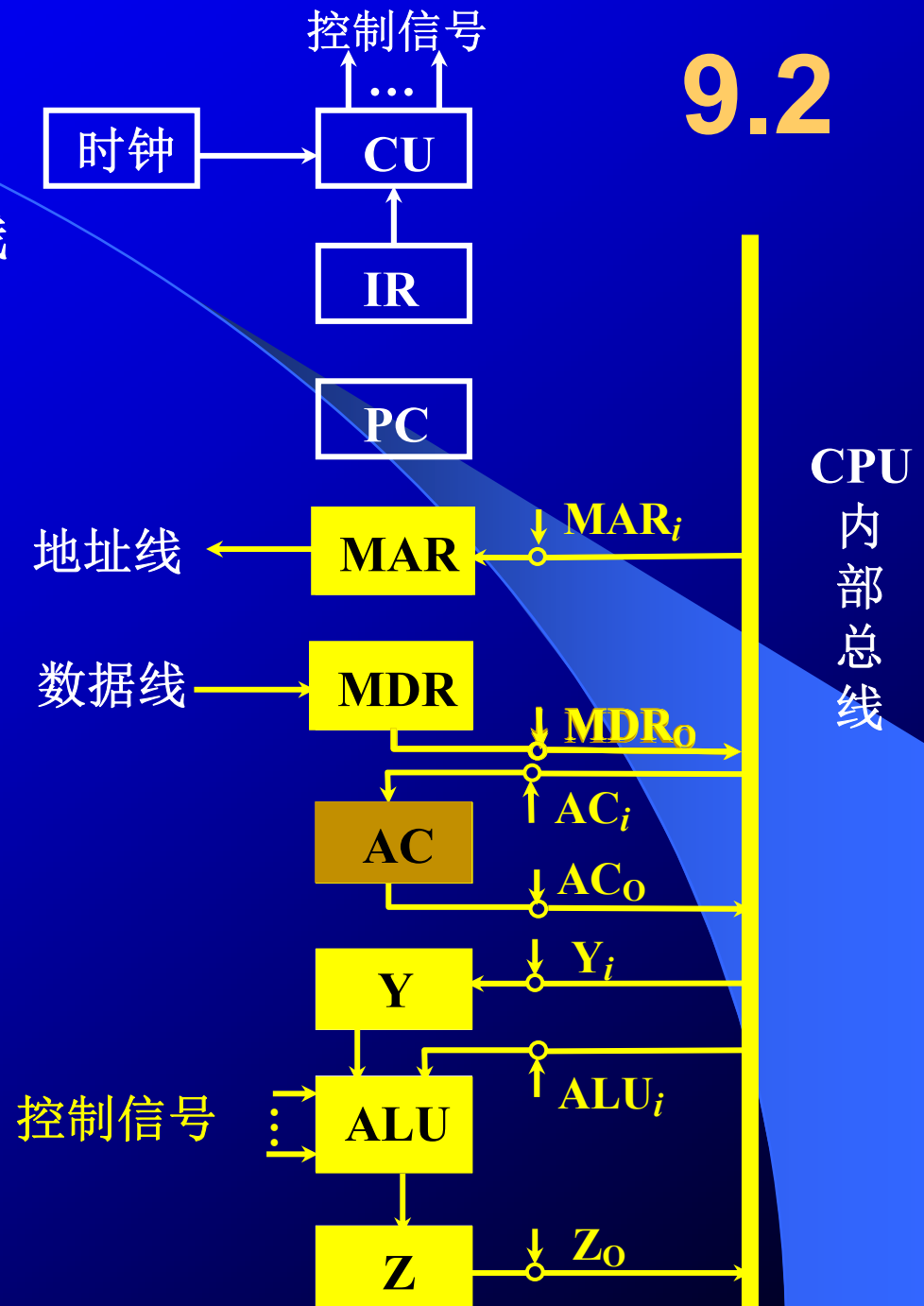
• $\text{MDR} \rightarrow \text{IR}$
 $\text{MDR}_0 \quad \text{IR}_i$

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



(3) ADD @ X 执行周期

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



三、多级时序系统

9.2

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）

9.2

一个机器周期内可完成若干个微操作

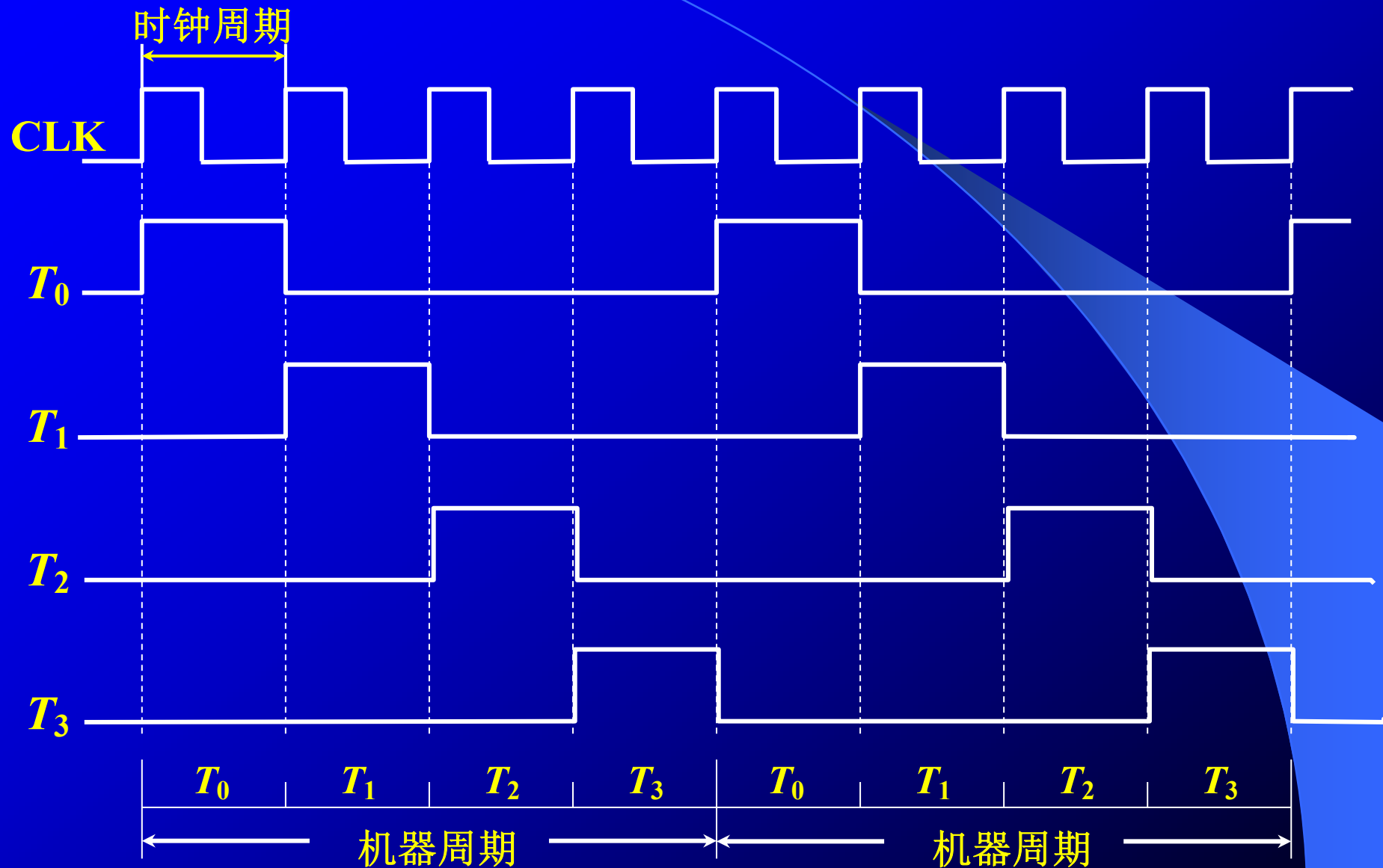
每个微操作需一定的时间

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）

用时钟周期控制产生一个或几个微操作命令

2. 时钟周期（节拍、状态）

9.2



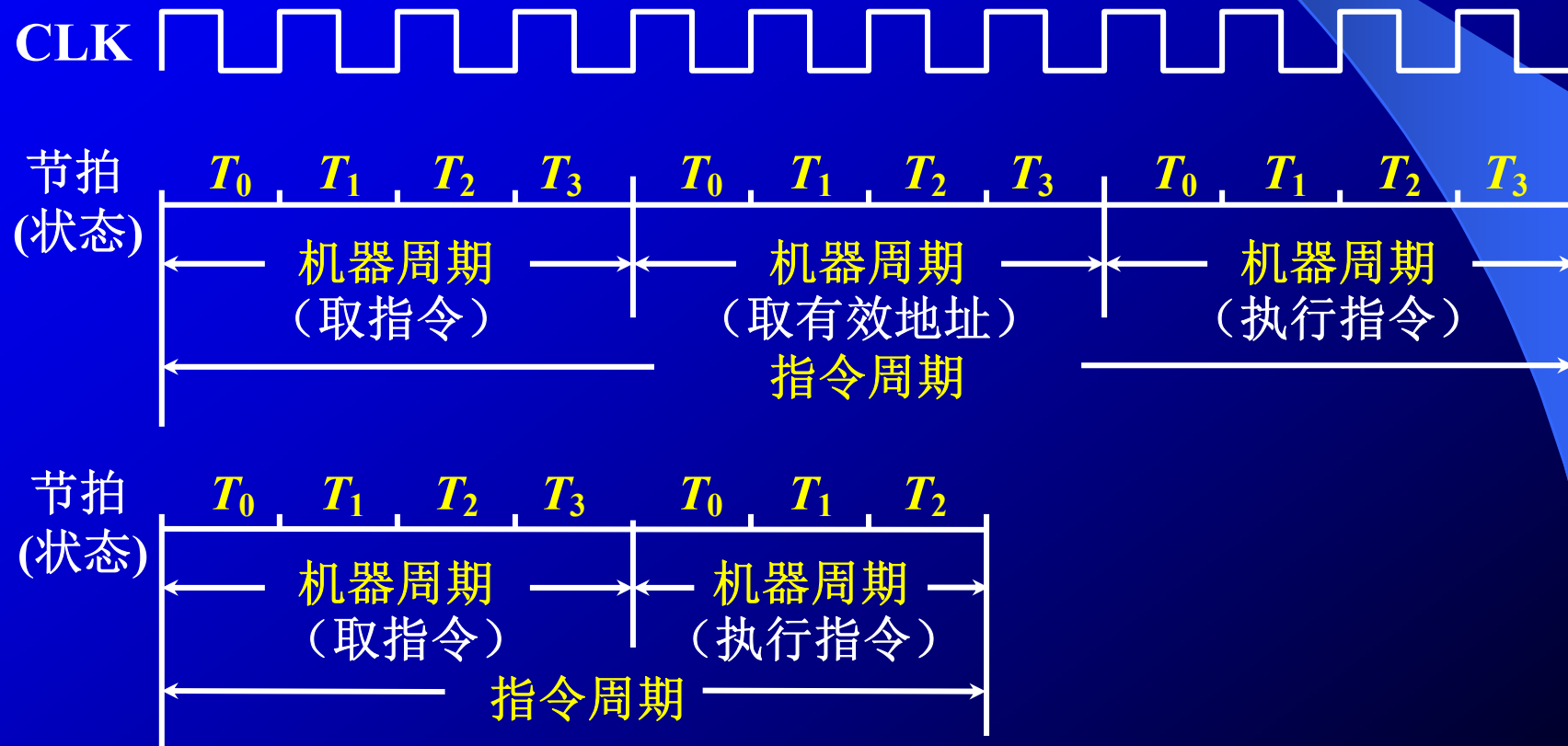
3. 多级时序系统

9.2

机器周期、节拍（状态）组成多级时序系统

一个指令周期含若干个机器周期

一个机器周期包含若干个时钟周期



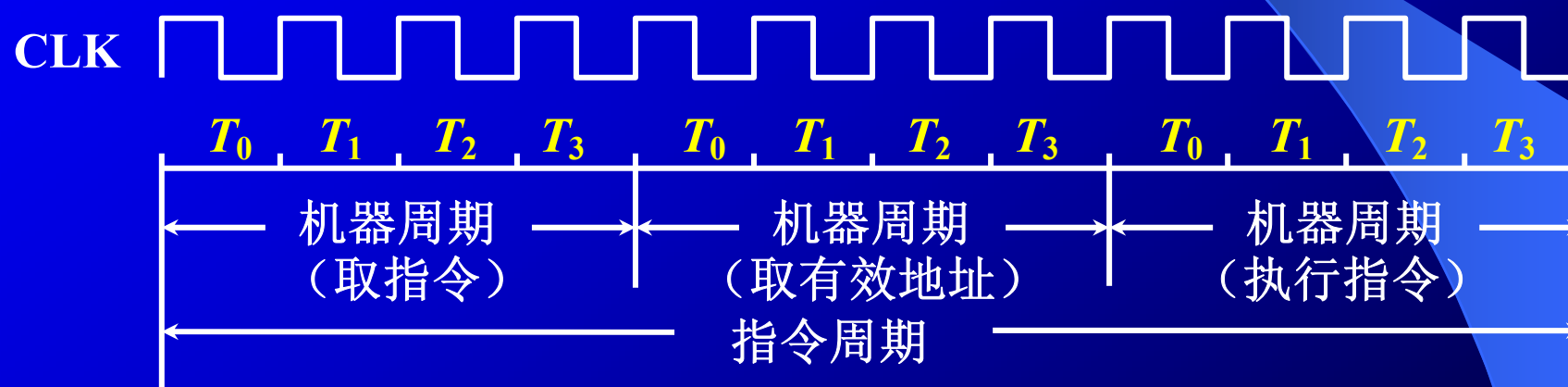
四、控制方式

9.2

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

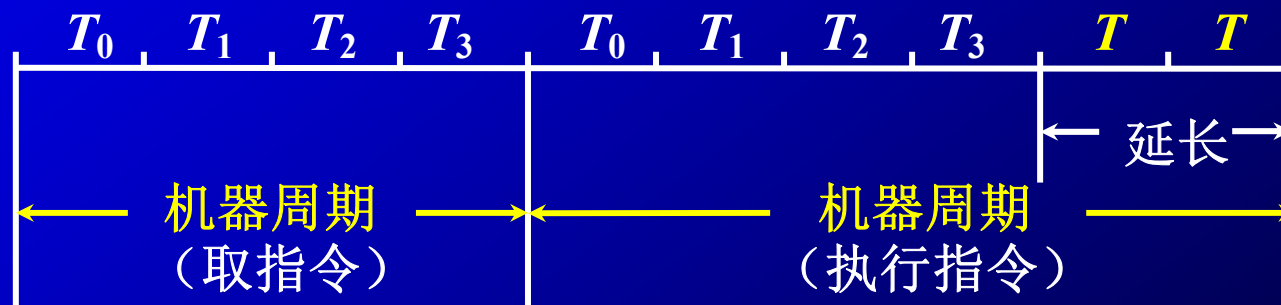
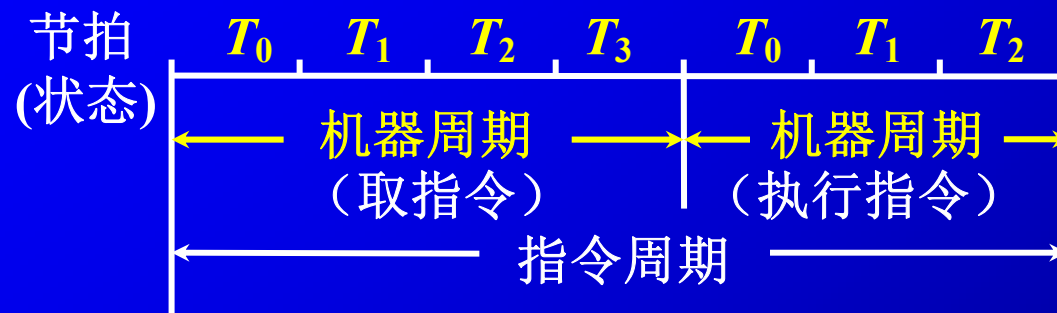
任一微操作均由 **统一基准时标** 的时序信号控制



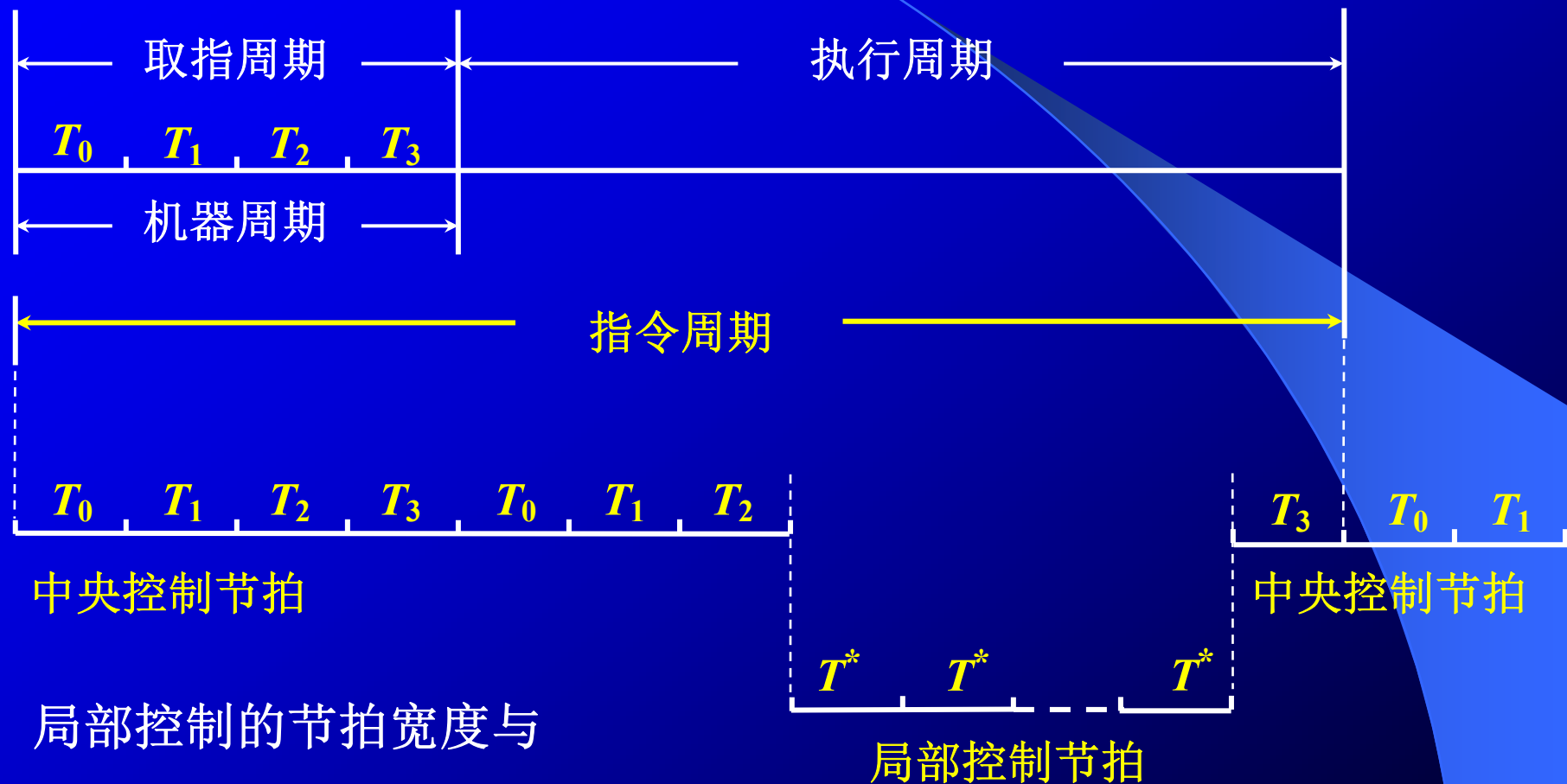
(1) 采用 **完全统一** 的机器周期和节拍

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

(2) 采用不同节拍的机器周期



(3) 采用中央控制和局部控制相结合的方法 9.2



中央控制节拍

中央控制节拍

局部控制的节拍宽度与
中央控制的节拍宽度一致

局部控制节拍

2. 异步控制方式

无基准时钟信号

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

4. 人工控制方式

(1) Reset

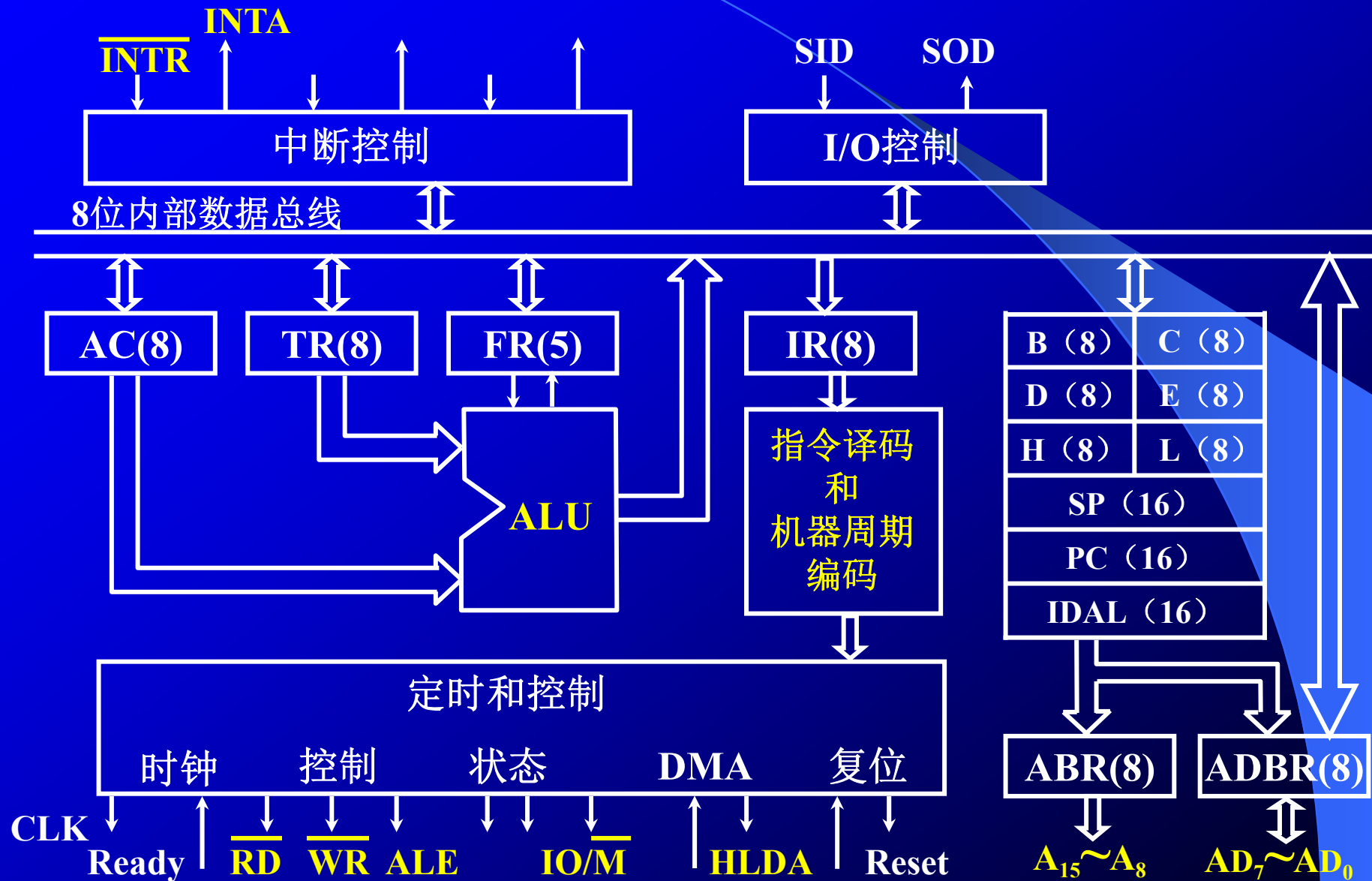
(2) 连续 和 单条 指令执行转换开关

(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	□	Vcc
X_2	□	2	39	□	HOLD
Reset out	□	3	38	□	HLDA
SOD	□	4	37	□	CLK(out)
SID	□	5	36	□	$\overline{\text{Rreset in}}$
Trap	□	6	35	□	Ready
RST7.5	□	7	34	□	IO/\overline{M}
RST6.5	□	8	33	□	S_1
RST5.5	□	9	32	□	\overline{RD}
$\overline{\text{INTR}}$	□	10	31	□	\overline{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
Vss	□	20	21	□	A_8

9.2

(4) 与中断有关的信号

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

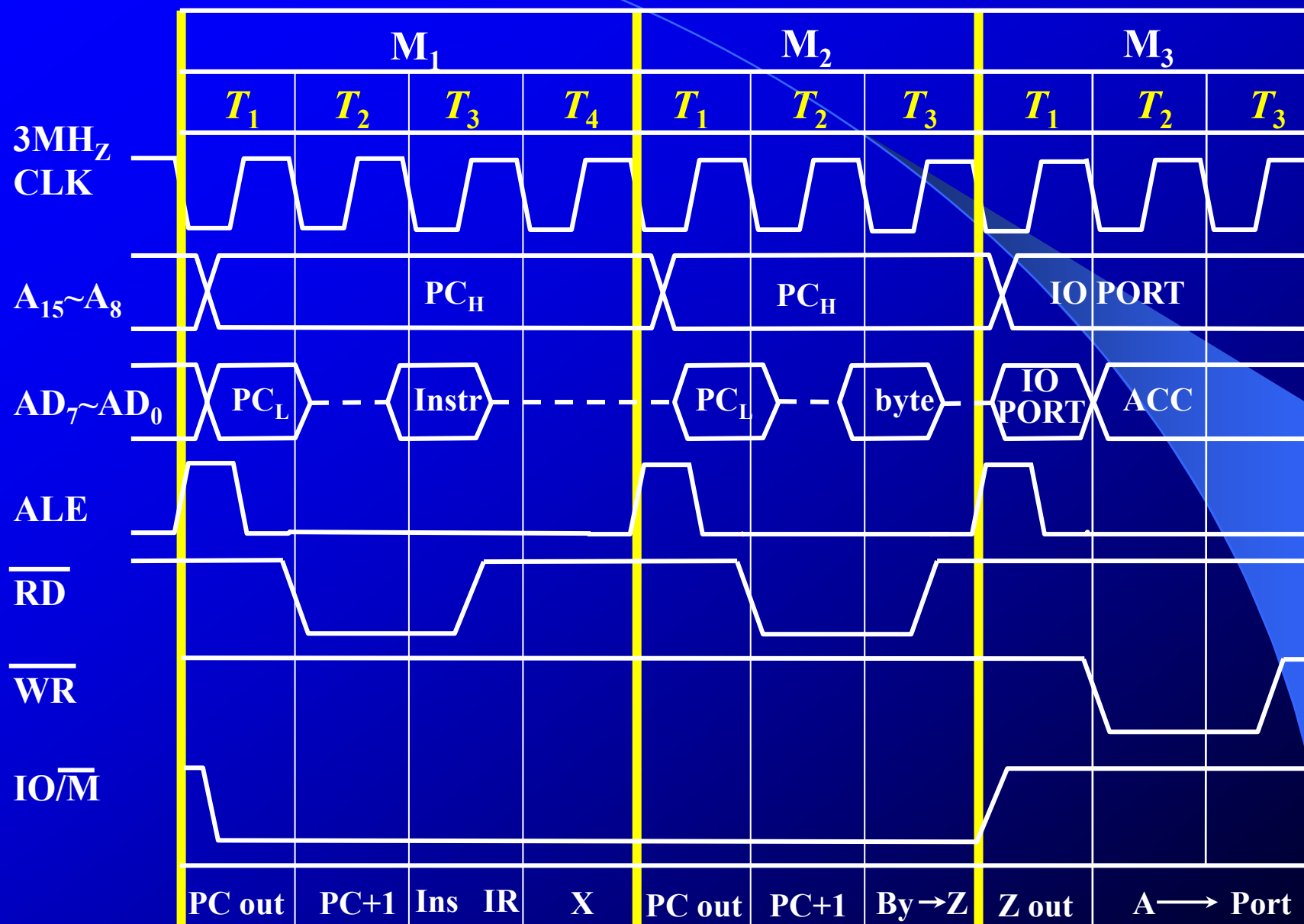
(6) 电源和地

V_{cc} +5V

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	$\overline{\text{Ready}}$
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



小结

9.2

每个 控制 信号在 指定机器周期 的
指定节拍 T 时刻 发出

以一条输出指令（**IO 写**）为例

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