

计算机组织与体系结构

第十讲

计算机科学与技术学院

舒燕君

Recap

– 定点运算

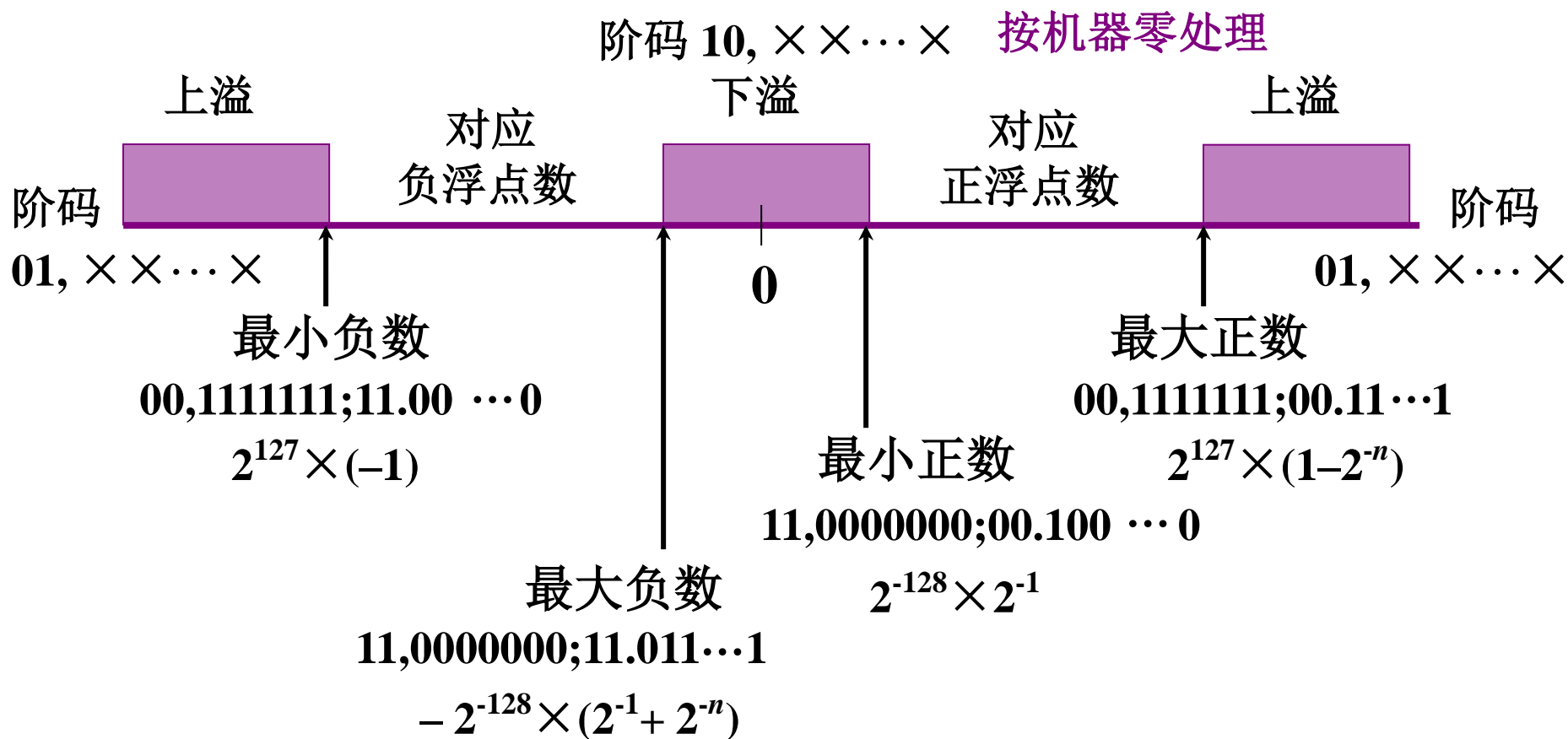
- ✓ 加减法运算（补码加减运算、溢出的判断）
- ✓ 乘法运算（由加和移位实现、硬件配置）

– 浮点四则运算

- ✓ 浮点加减法（对阶、尾数求和、规格化、舍入）

5. 溢出判断

设机器数为补码，尾数为规格化形式，并假设阶符取 2 位，阶码的数值部分取 7 位，数符取 2 位，尾数取 n 位，则该补码在数轴上的表示为



二、浮点乘除运算

$$x = S_x \cdot 2^{j_x} \quad y = S_y \cdot 2^{j_y}$$

1. 乘法

$$x \cdot y = (S_x \cdot S_y) \times 2^{j_x + j_y}$$

2. 除法

$$\frac{x}{y} = \frac{S_x}{S_y} \times 2^{j_x - j_y}$$

3. 步骤

(1) 阶码采用 补码定点加（乘法） 减（除法） 运算

(2) 尾数乘除同 定点 运算

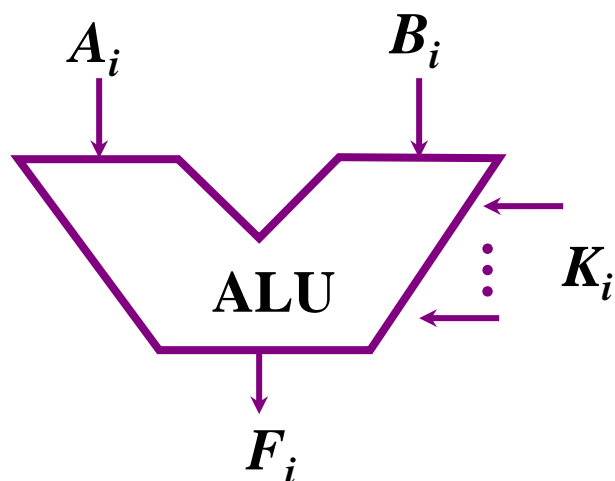
(3) 规格化

4. 浮点运算部件

阶码运算部件， 尾数运算部件

5.2.3 算术逻辑单元

一、ALU 电路



组合逻辑电路

K_i 不同取值

F_i 不同

四位 ALU 74181

$M = 0$ 算术运算

$M = 1$ 逻辑运算

$S_3 \sim S_0$ 不同取值，可做不同运算

第5章 CPU设计与实现

5.1 CPU 的结构

5.2 运算方法与ALU

5.3 多级时序系统（x86）

5.4 MIPS CPU的简单实现

5.3 多级时序系统（x86）

5.3.1 指令周期

5.3.2 微操作命令分析

5.3.3 控制单元

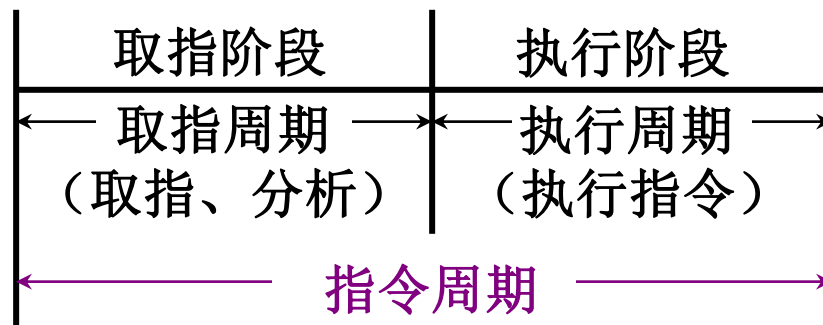
5.3.1 指令周期

一、指令周期的基本概念

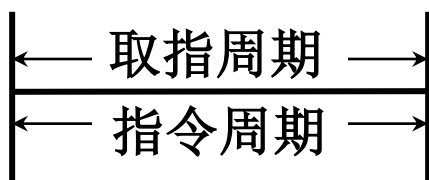
1. 指令周期

取出并执行一条指令所需的全部时间

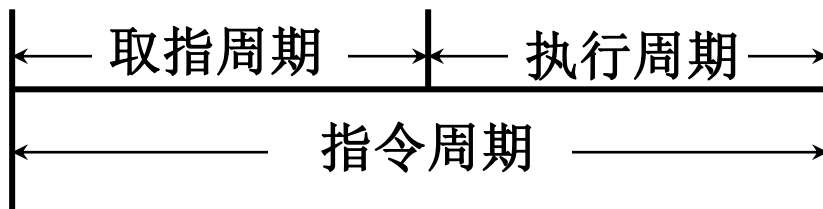
完成一条指令 { 取指、分析 取指周期
 执行 执行周期



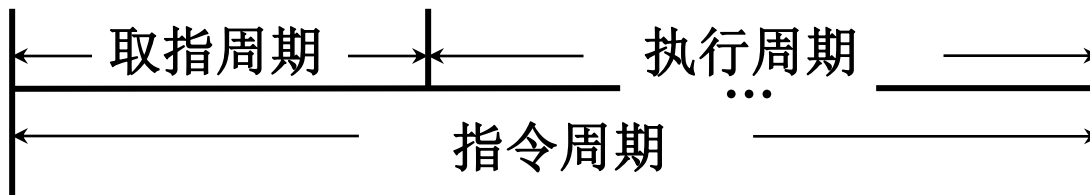
2. 每条指令的指令周期不同



NOP

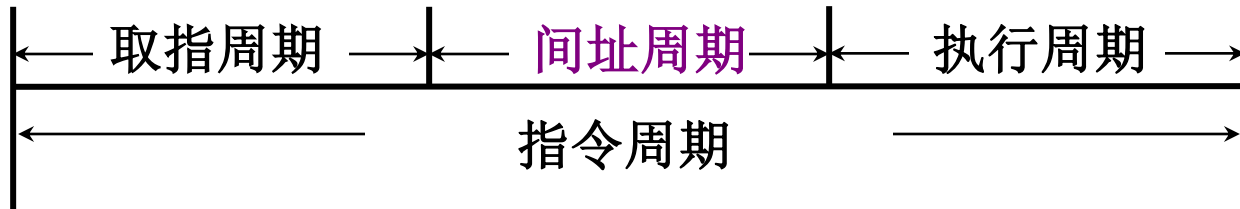


ADD mem

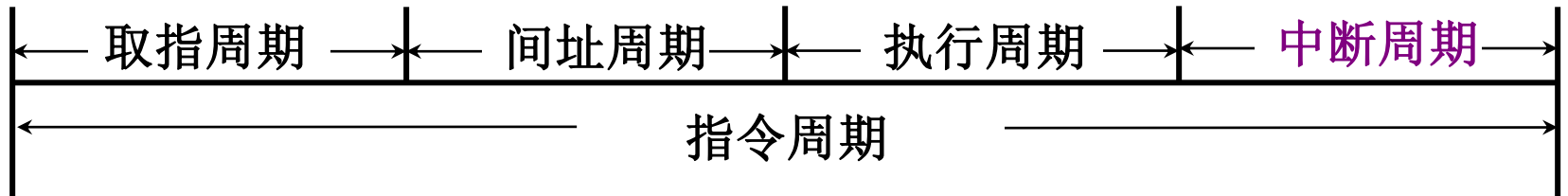


MUL mem

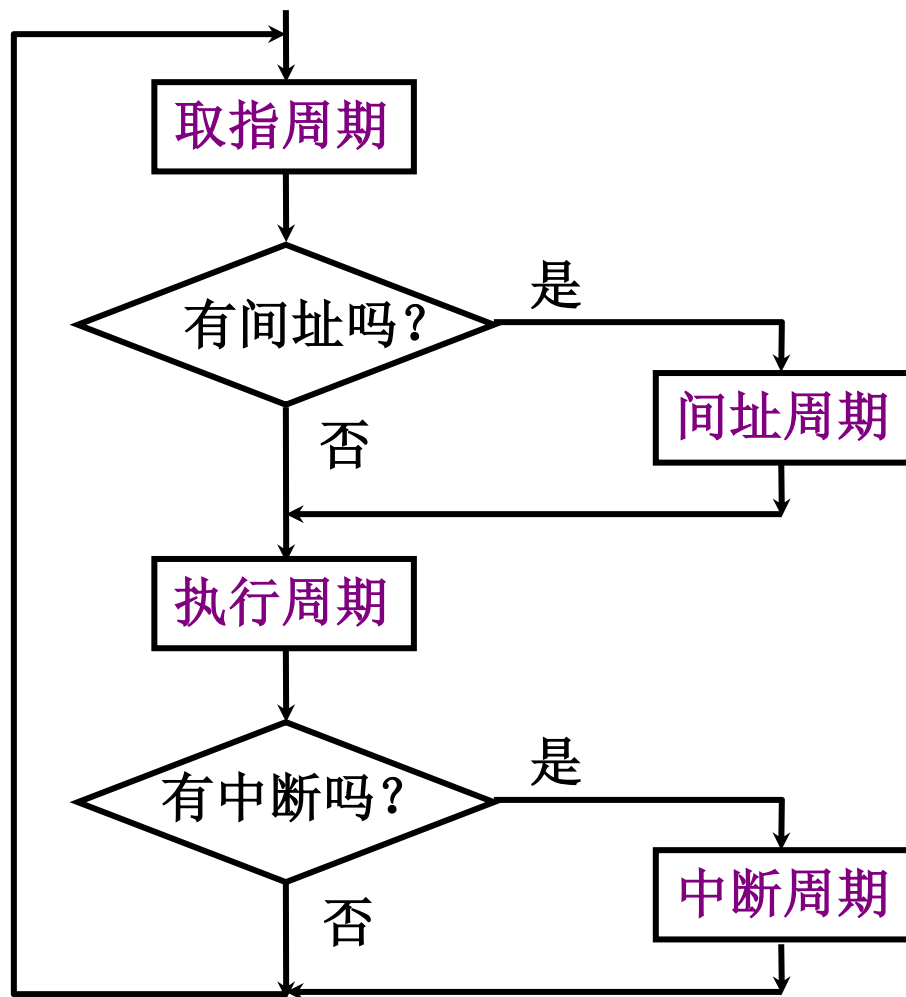
3. 具有间接寻址的指令周期



4. 带有中断周期的指令周期



5. 指令周期流程



6. CPU 工作周期的标志

CPU 访存有四种性质

取 指令

取指周期

取 地址

间址周期

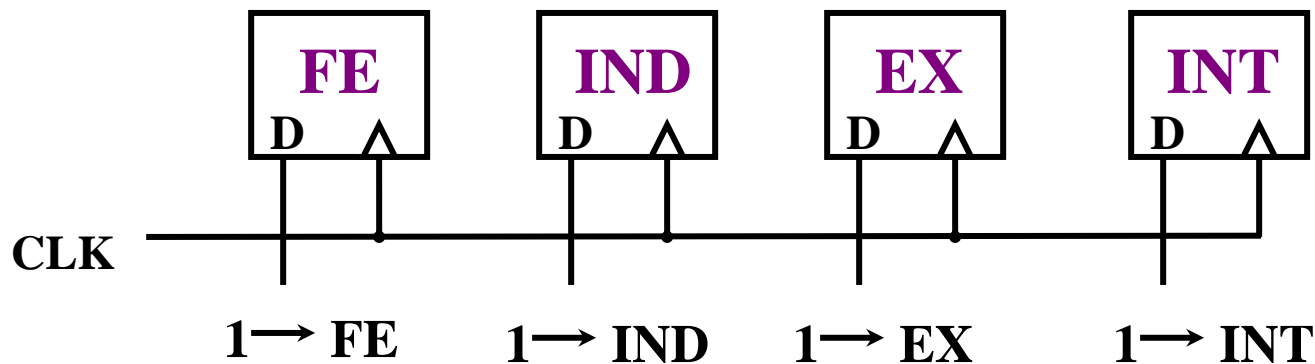
取 操作数

执行周期

存 程序断点

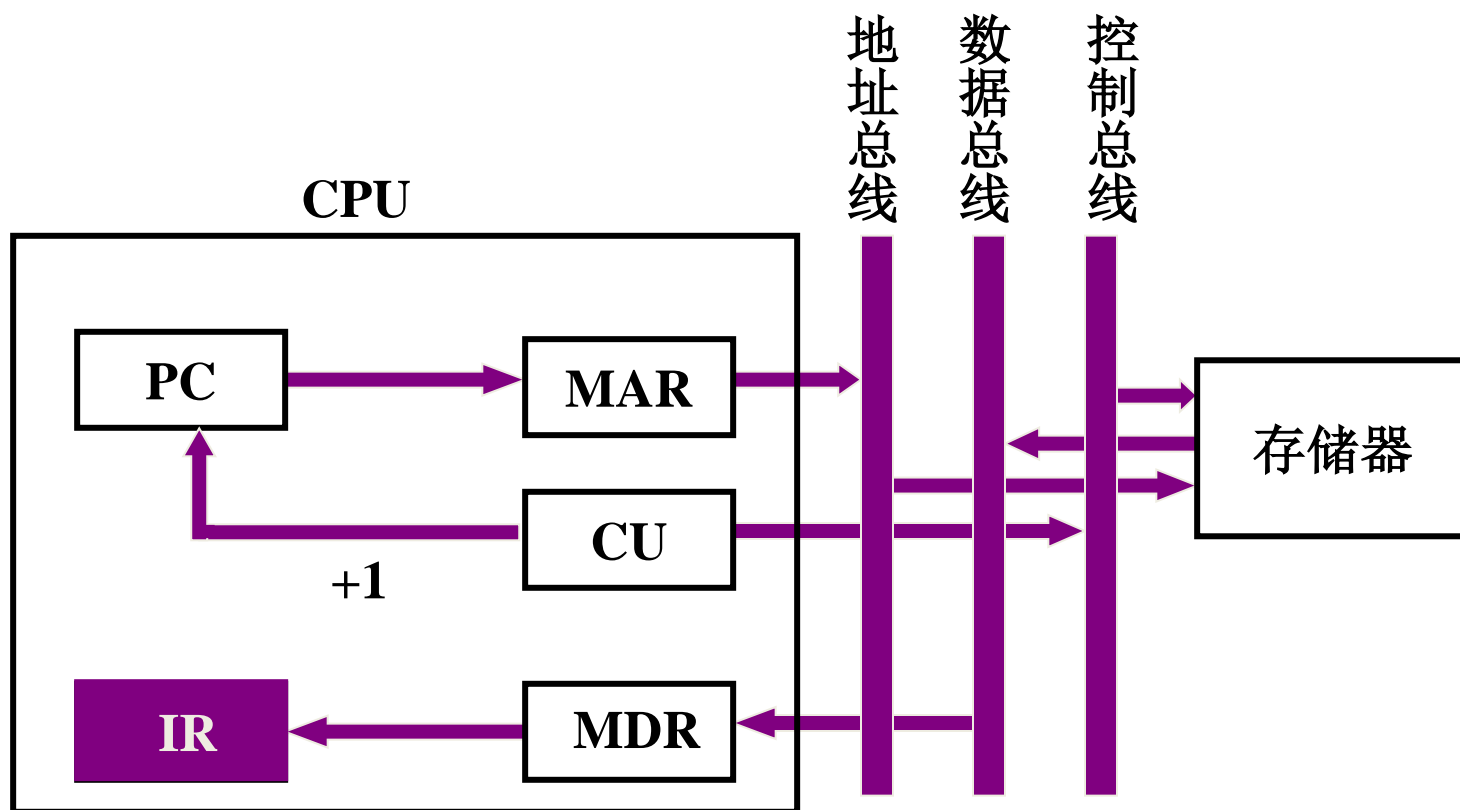
中断周期

CPU 的
4个工作周期

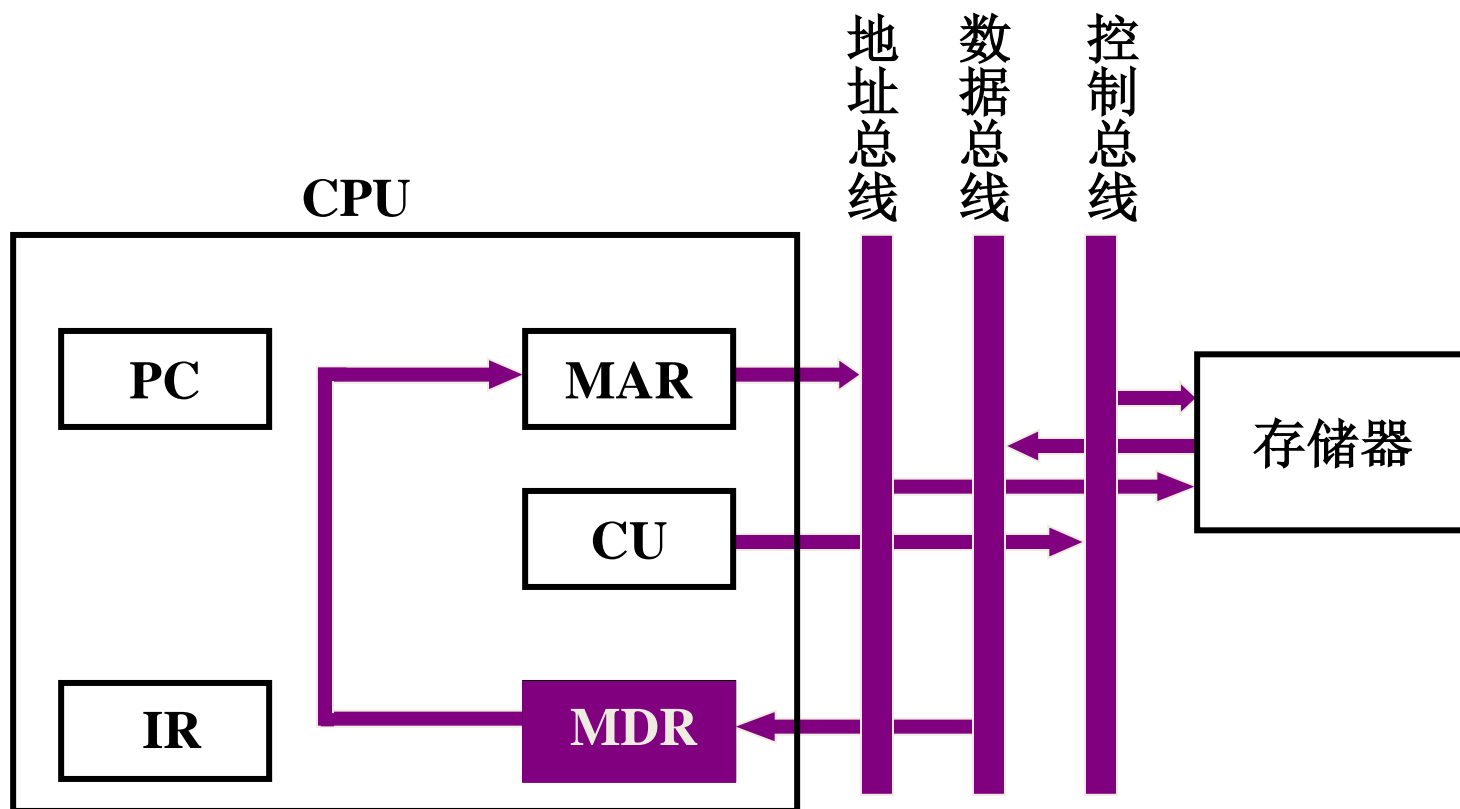


二、指令周期的数据流

1. 取指周期数据流



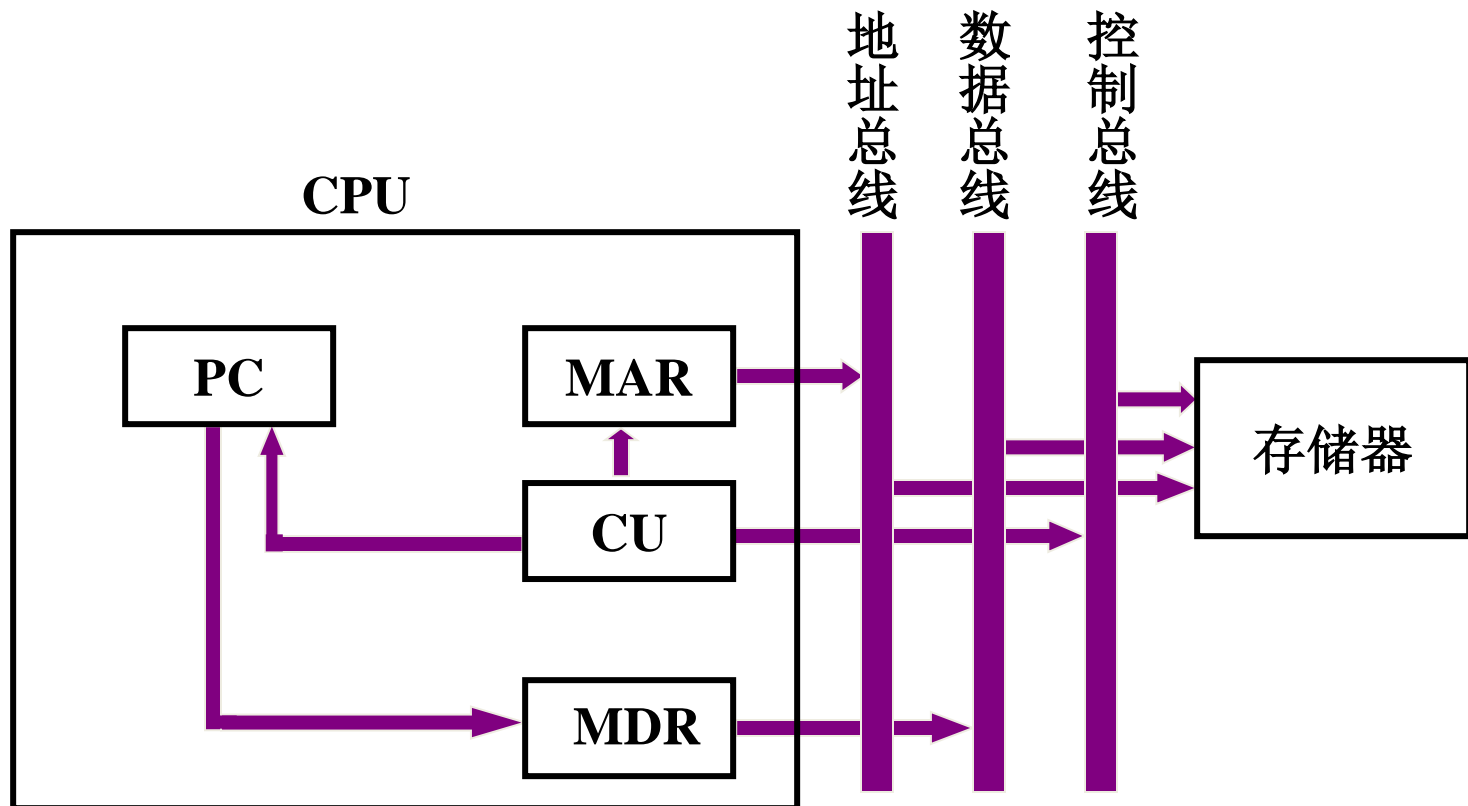
2. 间址周期数据流



3. 执行周期数据流

不同指令的执行周期数据流不同

4. 中断周期数据流



5.3.2 微操作命令分析

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

5.3.2 微操作命令分析

一、取指周期

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

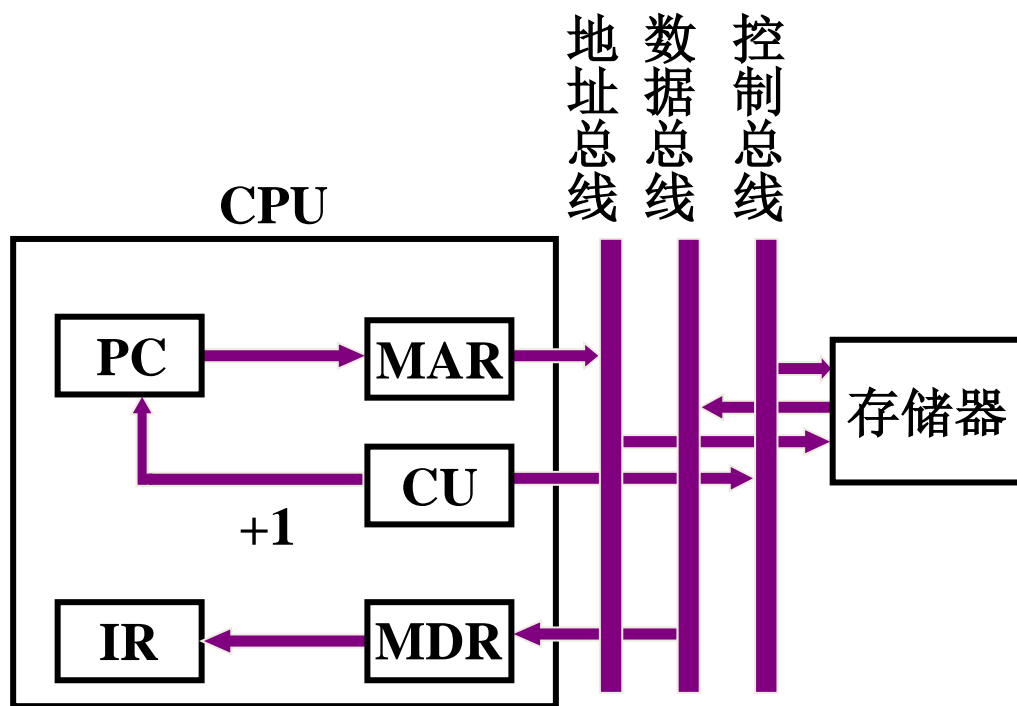
$1 \rightarrow R$

$M(MAR) \rightarrow MDR$

$MDR \rightarrow IR$

$OP(IR) \rightarrow CU$

$(PC) + 1 \rightarrow PC$



二、间址周期

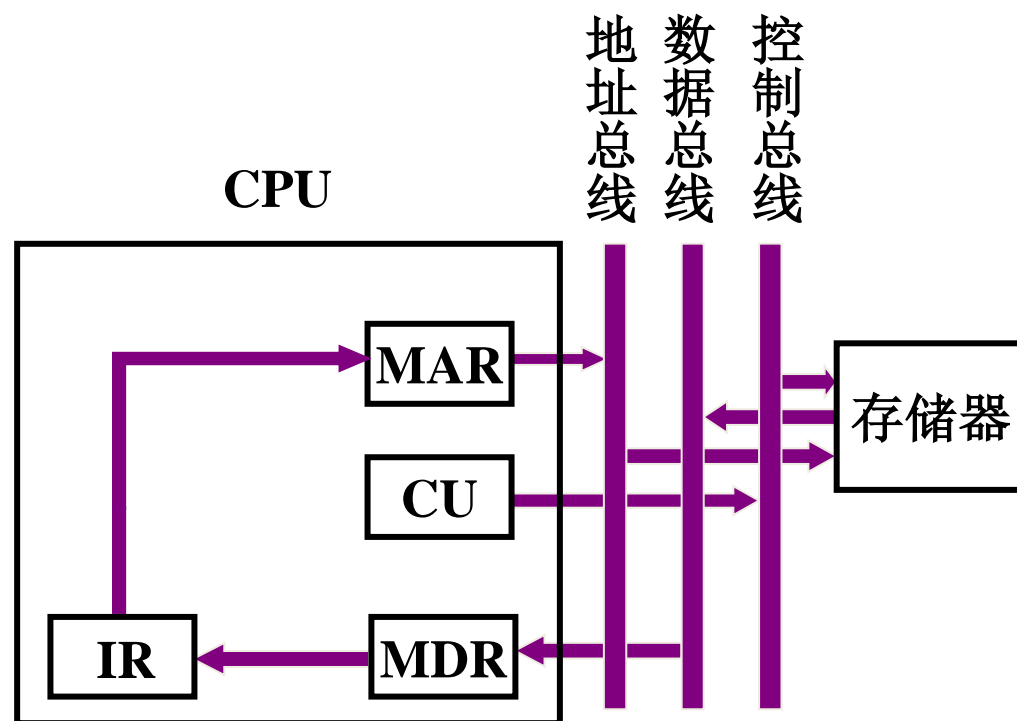
指令形式地址 \rightarrow MAR

$Ad(IR) \rightarrow MAR$

$1 \rightarrow R$

$M(MAR) \rightarrow MDR$

$MDR \rightarrow Ad(IR)$



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

(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

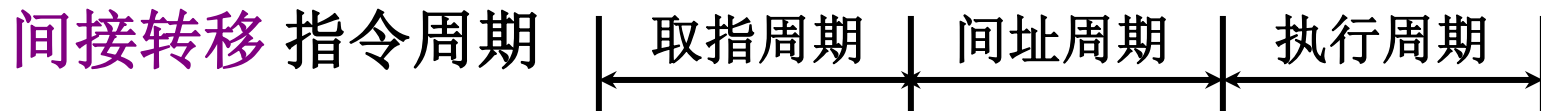
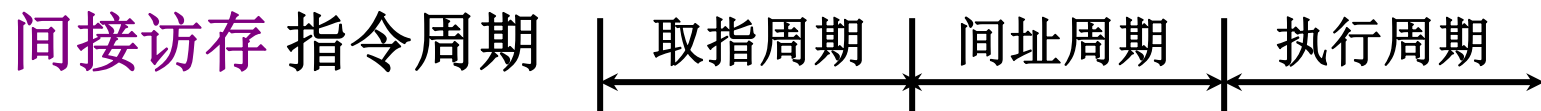
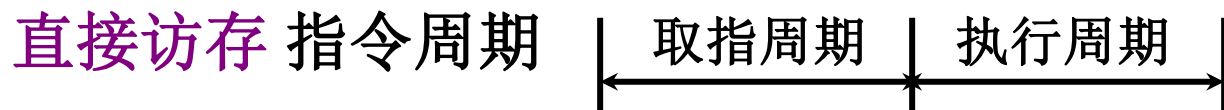
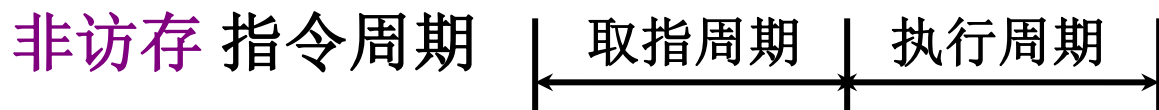
(1) 无条件转 **JMP X**

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

(2) 条件转移 **BAN X** (负则转)

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

4. 三类指令的指令周期



四、中断周期

程序断点存入 “0” 地址 程序断点 进栈

$0 \rightarrow \text{MAR}$

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

$1 \rightarrow \text{W}$

$1 \rightarrow \text{W}$

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

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

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

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

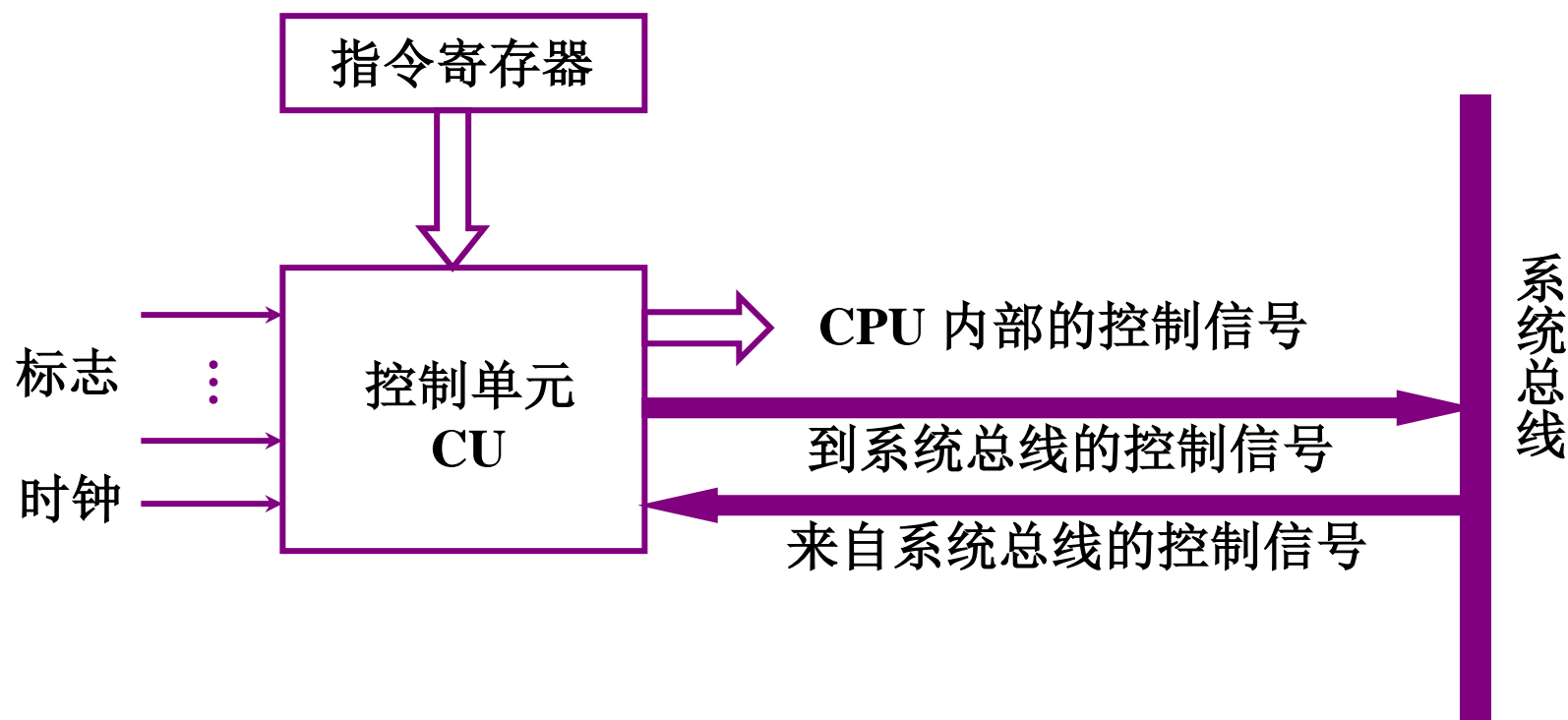
中断识别程序入口地址 $\text{M} \rightarrow \text{PC}$

$0 \rightarrow \text{EINT}$ (置 “0”)

$0 \rightarrow \text{EINT}$ (置 “0”)

5.3.3 控制单元

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

$\overline{\text{MREQ}}$

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

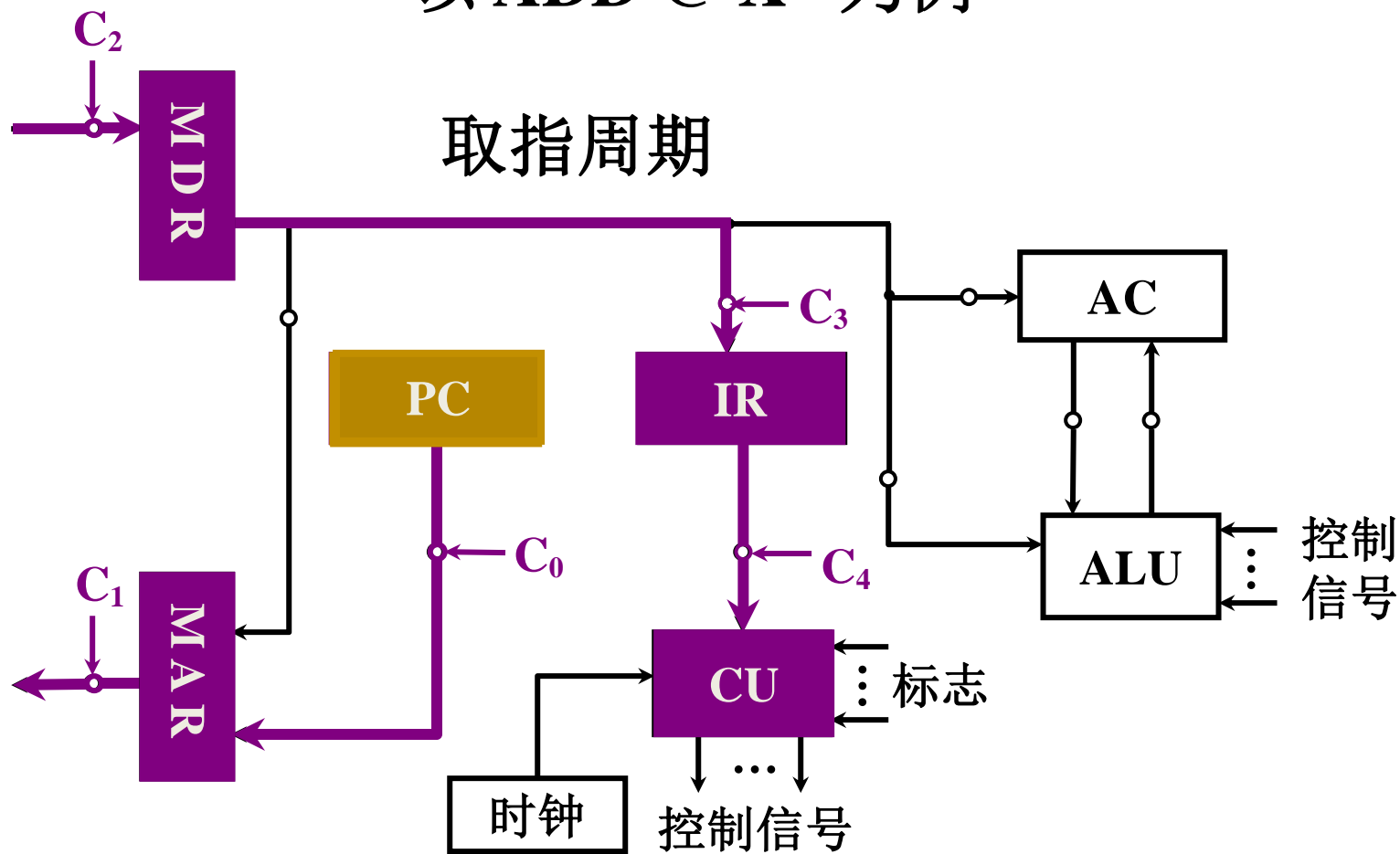
HLDA

总线响应信号

二、控制信号举例

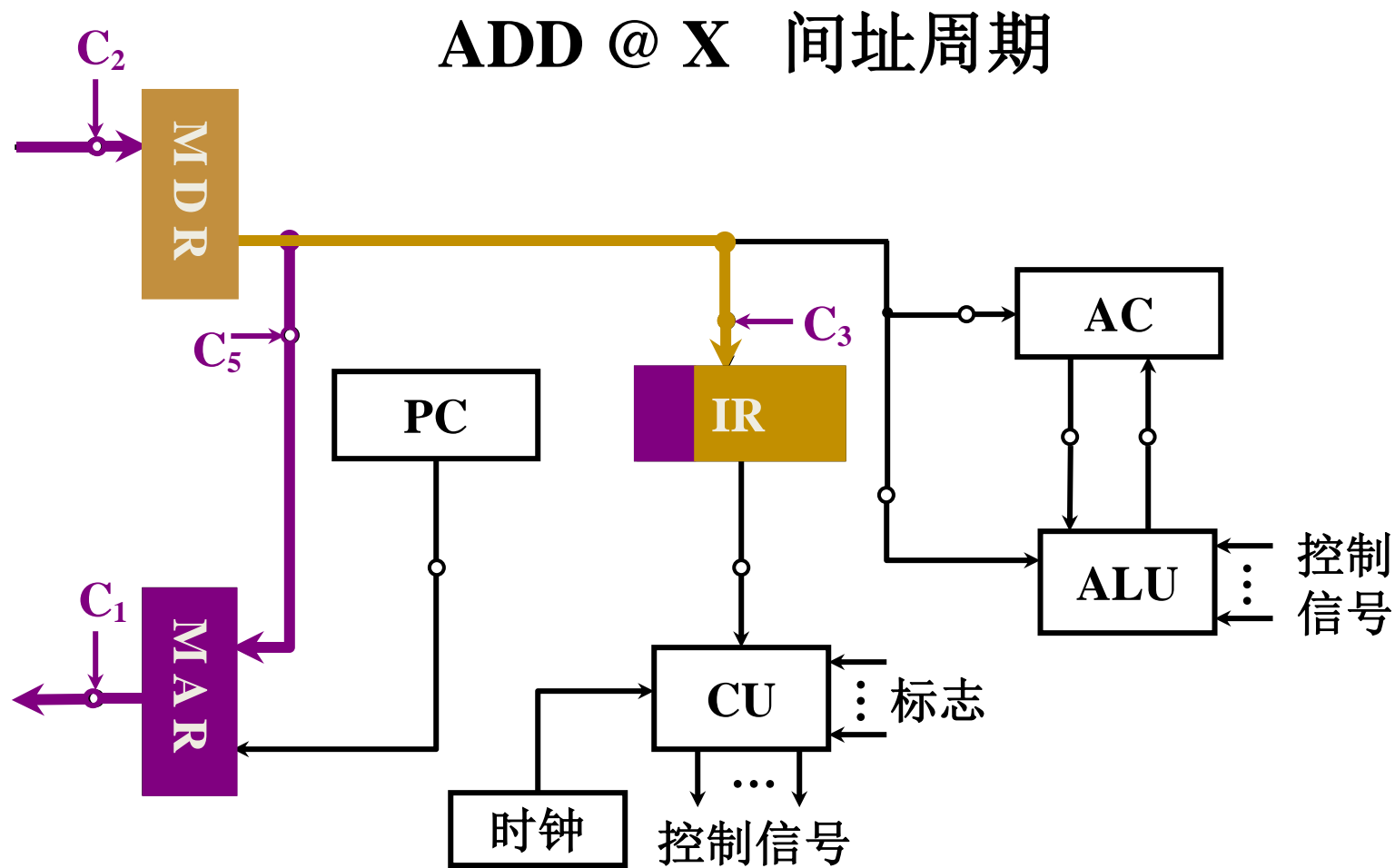
1. 不采用 CPU 内部总线的方式

以 **ADD @ X** 为例



二、控制信号举例

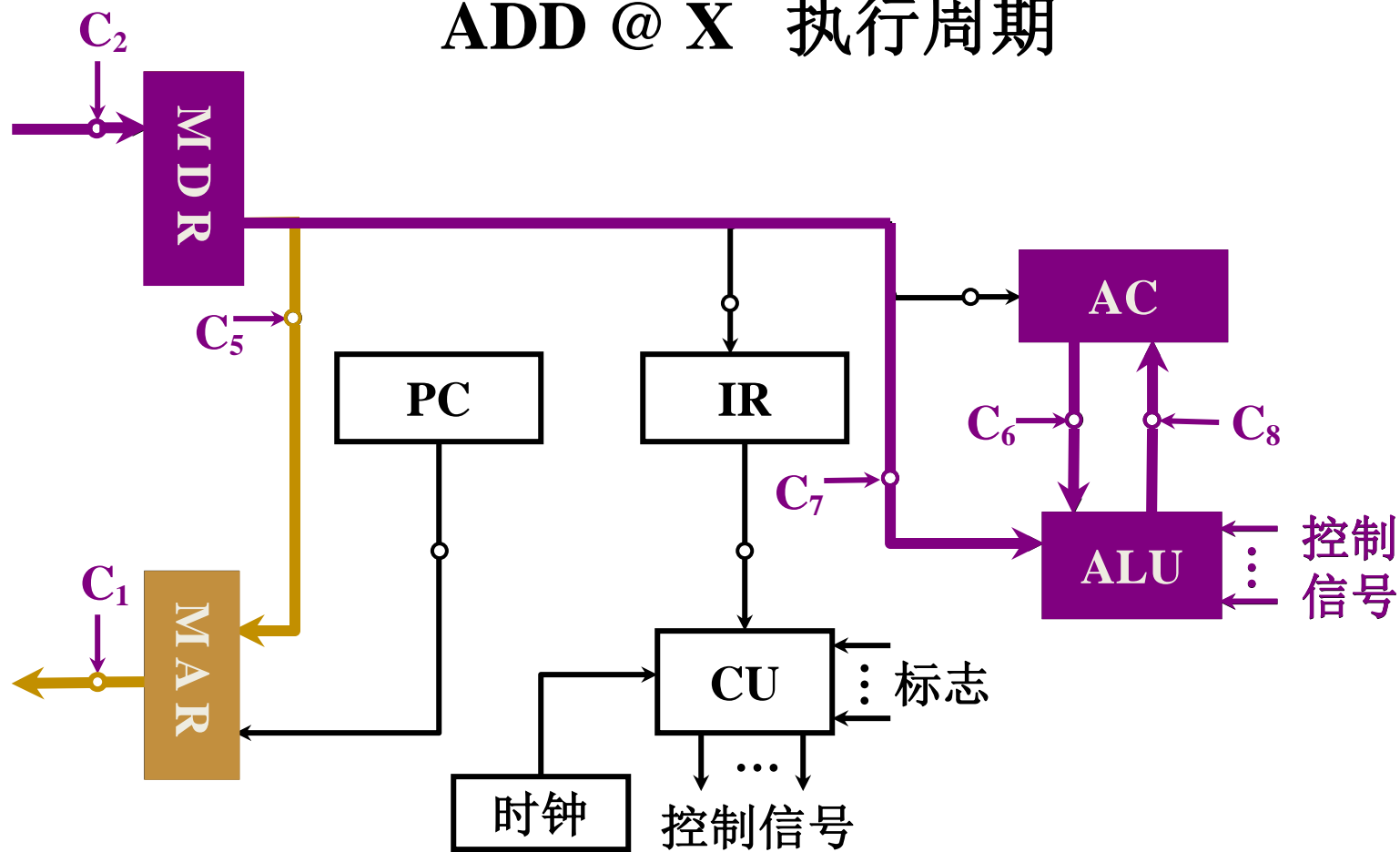
1. 不采用 CPU 内部总线的方式



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

- $PC \rightarrow MAR \rightarrow \text{地址线}$
 $PC_0 \quad MAR_i$

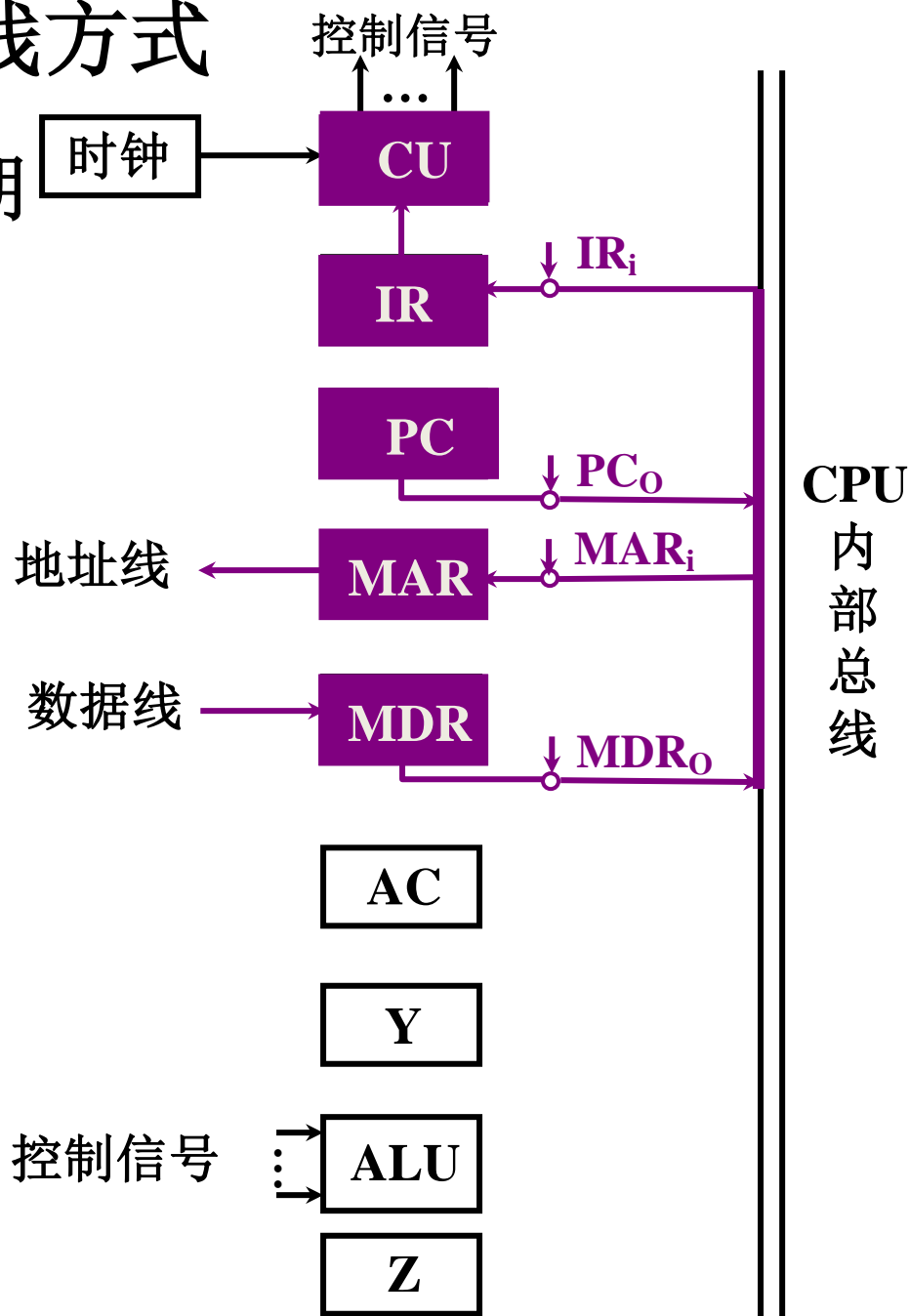
- CU 发读命令 $1 \rightarrow R$

- 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

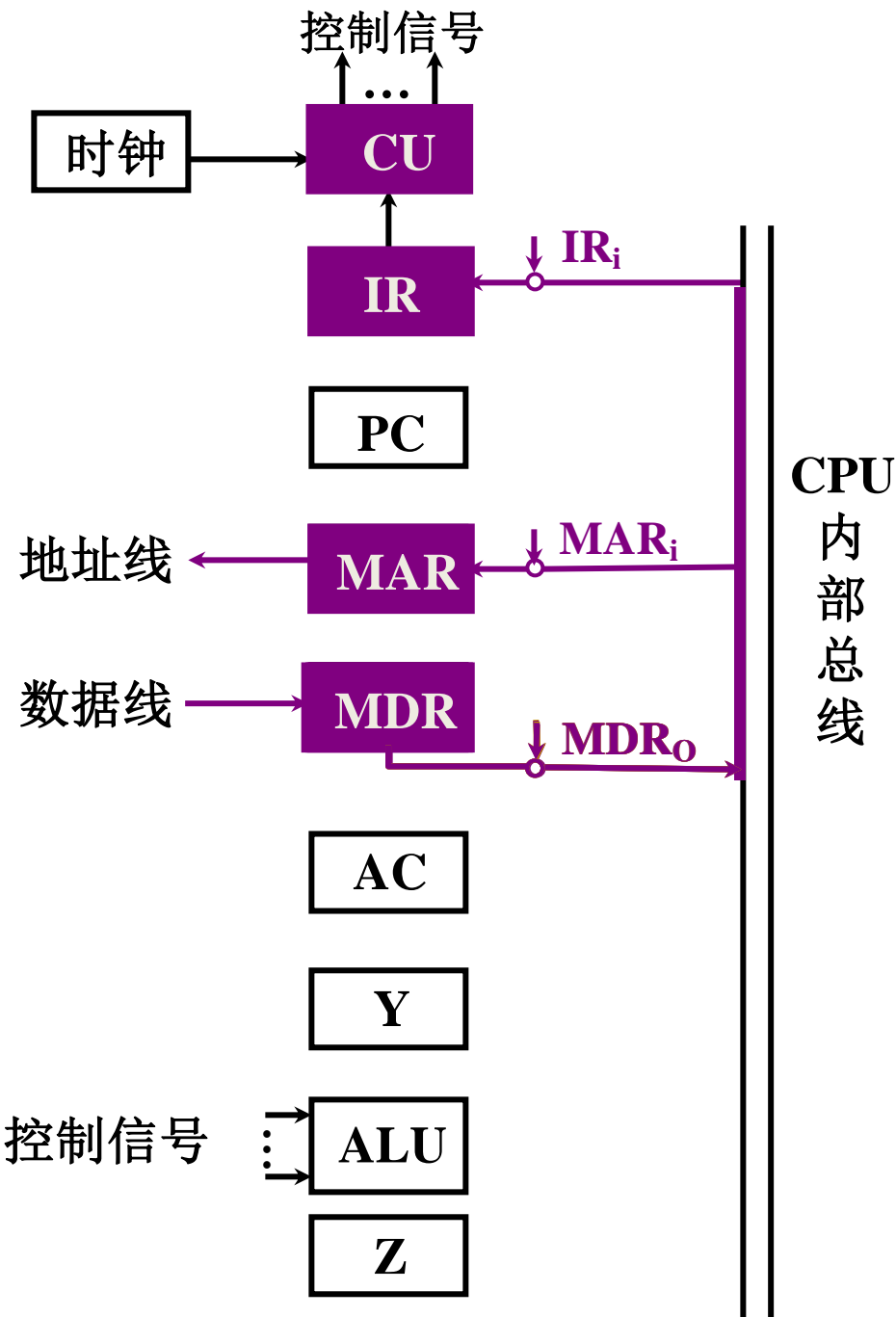
• $\text{MDR} \rightarrow \text{MAR} \rightarrow \text{地址线}$
 $\text{MDR}_0 \quad \text{MAR}_i$

• $1 \rightarrow R$

• 数据线 \rightarrow MDR

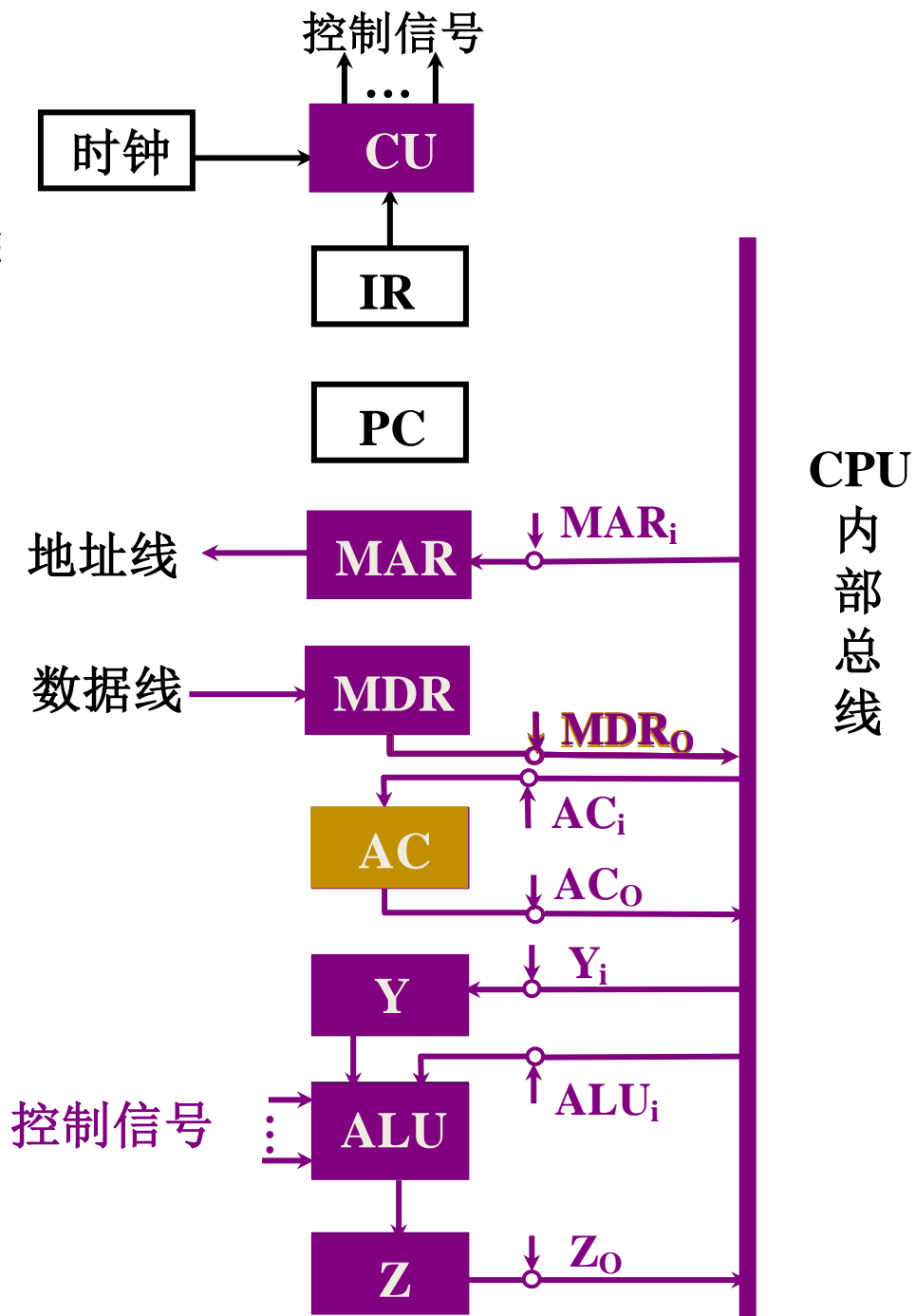
• $\text{MDR} \rightarrow \text{IR}$
 $\text{MDR}_0 \quad \text{IR}_i$

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



(3) ADD @ X 执行周期

- $\text{MDR} \longrightarrow \text{MAR} \longrightarrow \text{地址线}$
 $\text{MDR}_0 \quad \text{MAR}_i$
- $1 \longrightarrow \text{R}$
- 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

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

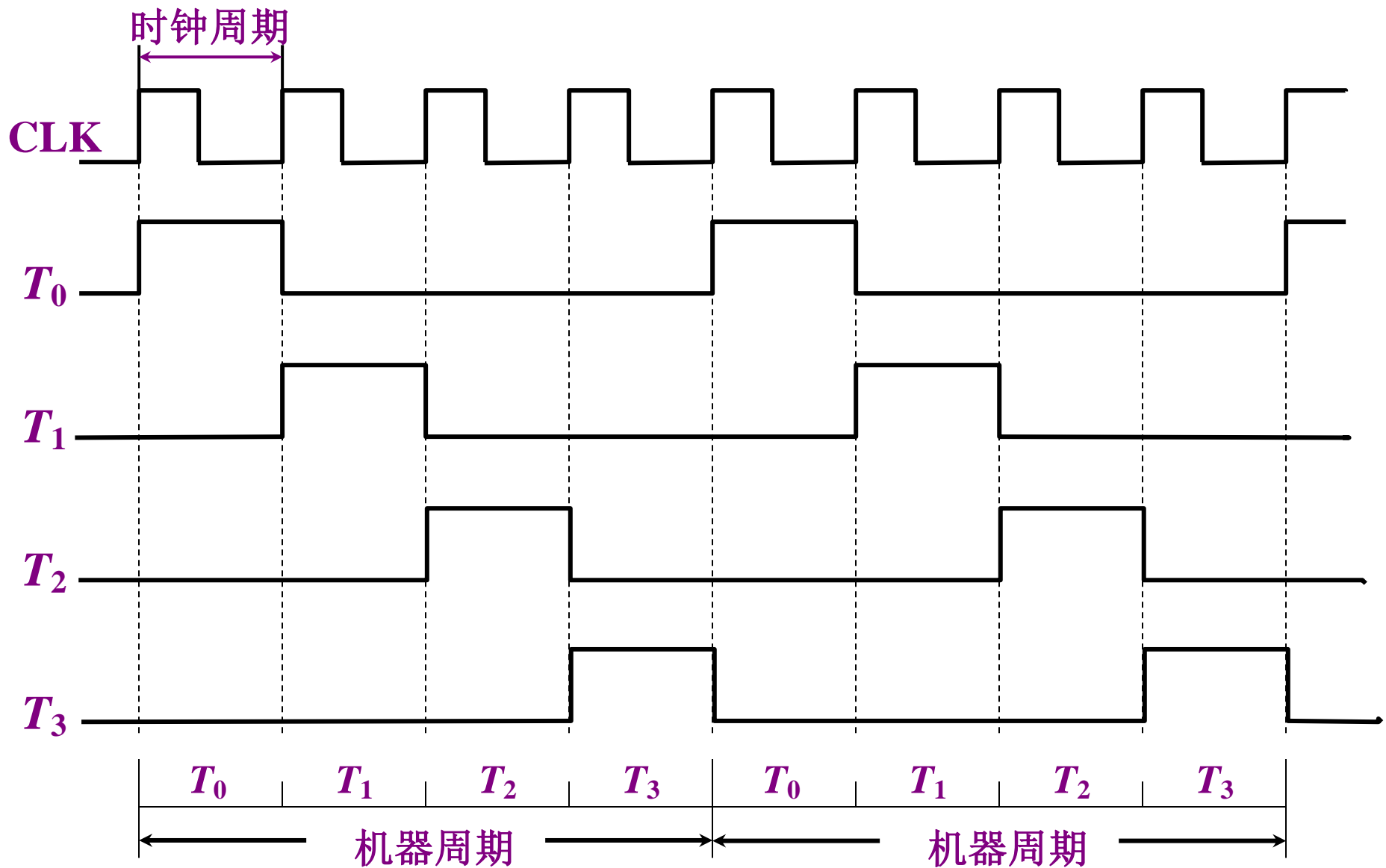
每个微操作需一定的时间

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

2. 时钟周期（节拍、状态）

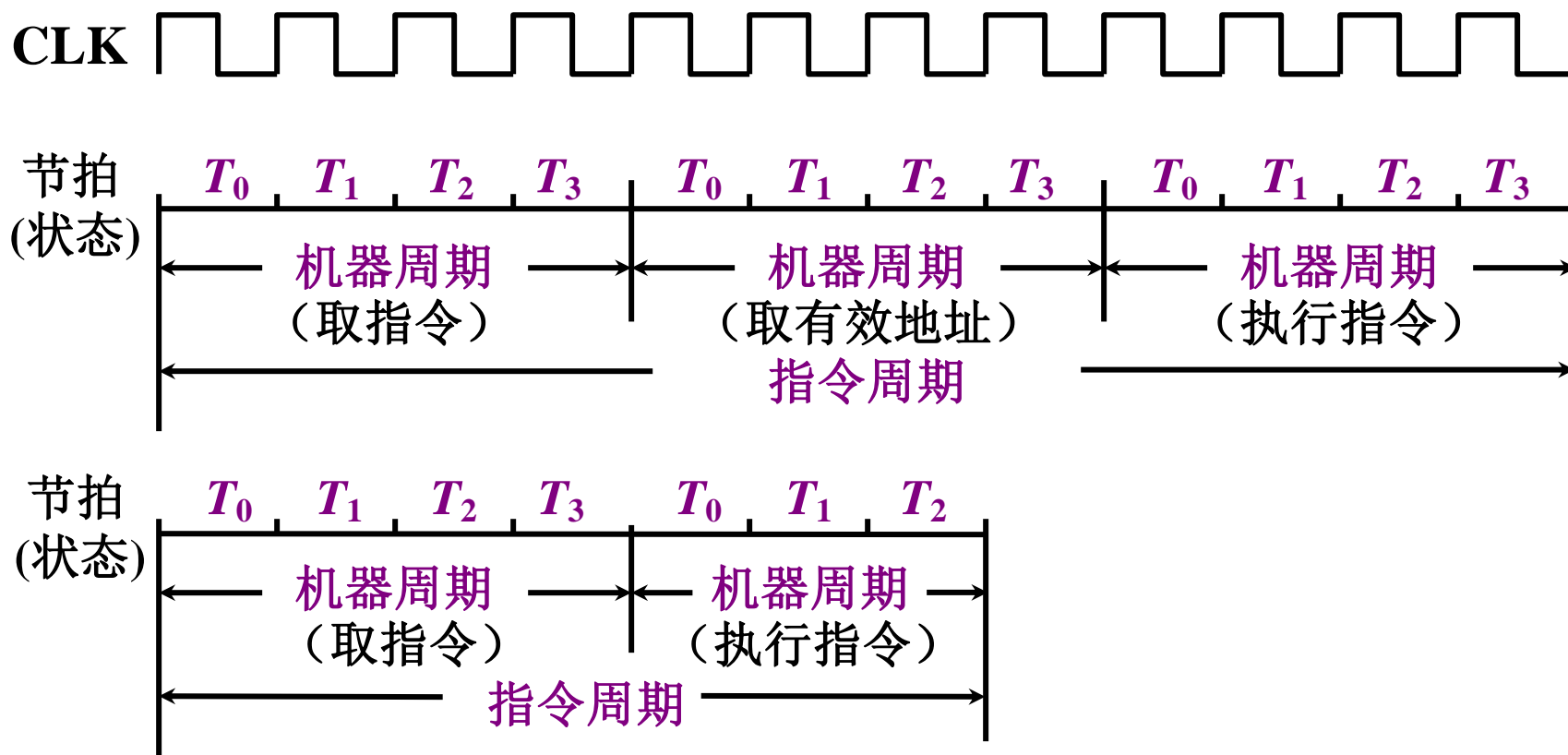


3. 多级时序系统

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



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