第9章 控制单元的功能

> 课件来源：《计算机组成原理》课件第9章（Slide 466–497，cod 版提取文本为主，全部关键图已用整页原图 slide_NNN.png 逐张核对）。本章是"指令系统"通向"CPU 设计"的枢纽：微操作序列、时序层次、控制方式是填空/简答高频考点，控制信号分析与主频-速度计算是综合题常客。

0. 本章考点清单

考点	常见题型	重要度
四个工作周期的微操作序列（取指/间址/执行/中断）	填空、简答、综合	★★★
多级时序系统（指令周期/机器周期/节拍）与时序计算	填空、计算	★★★
控制方式（同步/异步/联合/人工）及其三种同步方案	选择、简答	★★★
控制单元外特性：输入信号与输出信号	选择、填空、简答	★★★
微操作/微命令/微指令概念辨析	选择、填空	★★
两种数据通路方案的控制信号分析（ADD @X 例）	综合	★★
机器速度与主频关系（MIPS 之比 = 主频之比的前提）	计算、选择	★★
各类指令的指令周期组成	选择、填空	★★
8085 组成、外部引脚与 I/O 写时序实例	填空、选择、综合	★★

1. 基本概念与指令的四个工作周期

概念精讲

- **微操作**：指令执行过程中最基本的、不可再分解的操作（如 PC→MAR、1→R）。课件在时序部分直接使用"微操作"一词："一个机器周期内可完成若干个微操作"（课件 p.485）。
- **微命令（操作命令）**：控制部件向执行部件发出的、使执行部件完成某个微操作的控制信号。课件表述："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"（课件 p.476）。微命令与微操作一一对应：微命令是控制信号，微操作是该信号引起的操作。
- **微指令**（补充常识）：若干个同时发出的微命令的集合称为一条微指令，它是第10章控制单元设计的对象；本章只讨论微操作命令序列如何由时序系统产生，不展开微指令格式。
- **控制单元 CU 的功能**：根据指令操作码、标志、时钟和外来信号，按规定的时序发出各种微操作命令（控制信号），指挥 CPU 内部及系统总线上各部件协调工作；"每个控制信号在指定机器周期的指定节拍 T 时刻发出"（课件 p.497）。

要点：完成一条指令分 4 个工作周期（课件 p.467）

工作周期	任务	是否每条指令都有
取指周期	从存储器取出指令送 IR，PC 加 1	是
间址周期	间接寻址时取出有效地址	否（仅间接寻址指令）
执行周期	完成指令规定的操作	是
中断周期	保存断点、转中断服务程序	否（有中断请求且允许中断时）

易错警示：间址周期和中断周期不是每条指令必经的周期；"指令周期 = 取指 + 执行"只对非访存/直接访存/转移类指令成立（见第4节）。

2. 取指周期与间址周期的微操作序列

取指周期（课件 p.468）

```
PC → MAR → 地址线        ; 指令地址送地址寄存器并上地址总线
1 → R                    ; CU 发读命令
M(MAR) → MDR             ; 存储器读出指令送数据寄存器
MDR → IR                 ; 指令送指令寄存器
OP(IR) → CU              ; 操作码送控制单元译码
(PC) + 1 → PC            ; PC 加 1，指向下一条指令
```

9.1 操作命令的分析

一、取指周期

PC ⟶ MAR ⟶ 地址线

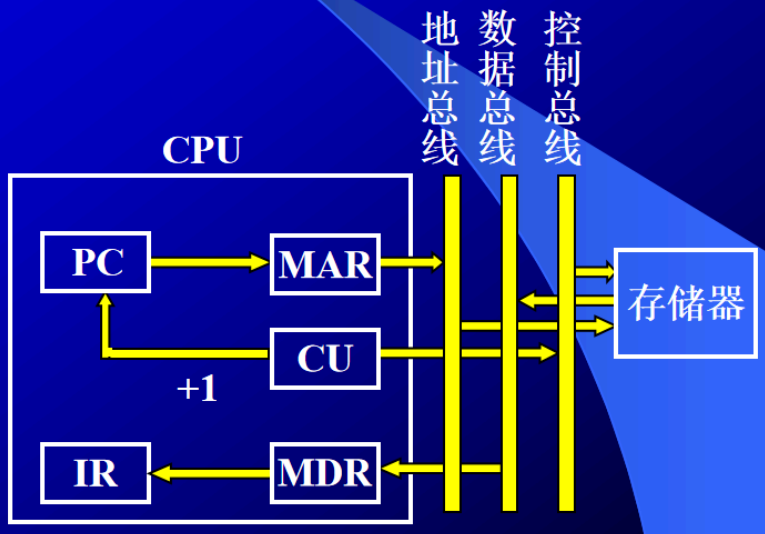
1 ⟶ R

M (MAR) ⟶ MDR

MDR ⟶ IR

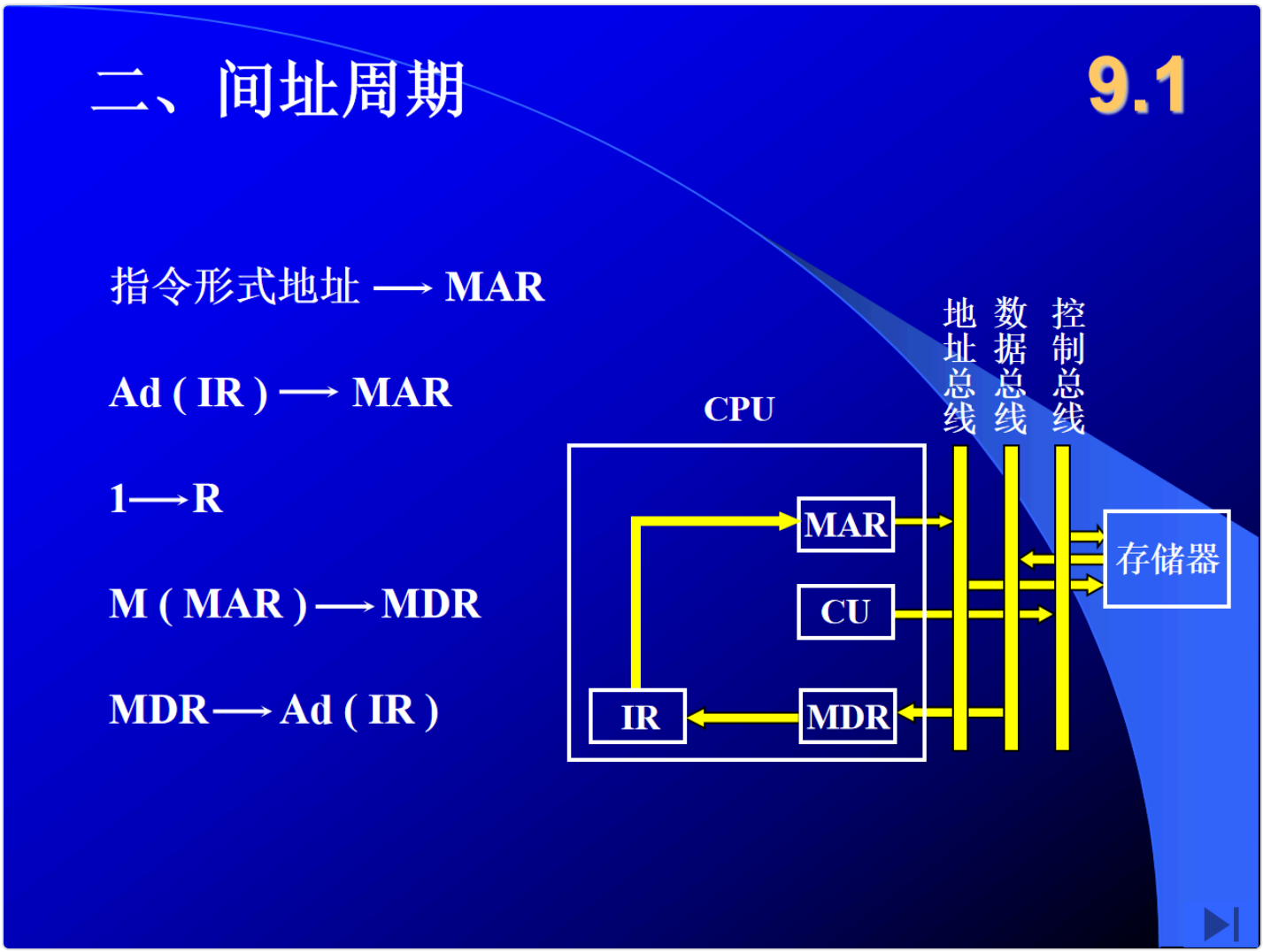
OP（IR）⟶ CU

(PC) + 1 ⟶ PC



间址周期（课件 p.469）

```
Ad(IR) → MAR          ; 指令的形式地址送 MAR
1 → R                 ; 读命令
M(MAR) → MDR          ; 读出的是有效地址（不是操作数）
MDR → Ad(IR)          ; 有效地址回送 IR 的地址字段
```



要点与对比

对比项	取指周期	间址周期
送 MAR 的地址来源	PC	Ad(IR)（形式地址）
读出的内容	指令	有效地址
读后送往	IR（整条指令）	Ad(IR)（仅地址字段）
PC 是否加 1	是	否

易错警示：间址周期最后一步是 MDR→Ad(IR)，不是 MDR→IR（整条指令会被破坏）；间址周期不修改 PC。

3. 执行周期：各类指令的微操作序列

1）非访存指令（课件 p.470）

指令	功能	微操作
CLA	清 A	0 → ACC
COM	取反	$\overline{ACC}$ → ACC
SHR	算术右移	L(ACC) → R(ACC)，ACC_0 → ACC_0（符号位不变）
CSL	循环左移	R(ACC) → L(ACC)，ACC_0 → ACC_n（最高位移入最低位）
STP	停机	0 → G（运行标志 G 清 0）

2）访存指令（课件 p.471–472）

指令	微操作序列
ADD X（加法）	Ad(IR) → MAR；1 → R；M(MAR) → MDR；(ACC) + (MDR) → ACC
STA X（存数）	Ad(IR) → MAR；1 → W；ACC → MDR；MDR → M(MAR)
LDA X（取数）	Ad(IR) → MAR；1 → R；M(MAR) → MDR；MDR → ACC

3）转移指令（课件 p.472）

指令	微操作
JMP X（无条件转）	Ad(IR) → PC
BAN X（负则转）	A_0·Ad(IR) + $\bar{A}_0$·(PC) → PC

BAN 的含义：A_0 为 ACC 的符号位。A_0=1（负）时把 Ad(IR) 送 PC 实现转移；A_0=0 时 $\bar{A}_0$=1，(PC) 送回 PC，即顺序执行。该式用一个逻辑表达式统一了"转"与"不转"两种情况（课件 p.472）。

易错警示： - SHR 是**算术**右移，符号位 ACC_0 保持不变；CSL 是**循环**左移，移出的 ACC_0 进入 ACC_n，两者不要混。 - STP 清的是运行标志 G，不是清 ACC。 - 存数指令发的是写命令 1→W，且数据流向是 ACC→MDR→M(MAR)，与取数相反。

4. 指令周期的组成与中断周期

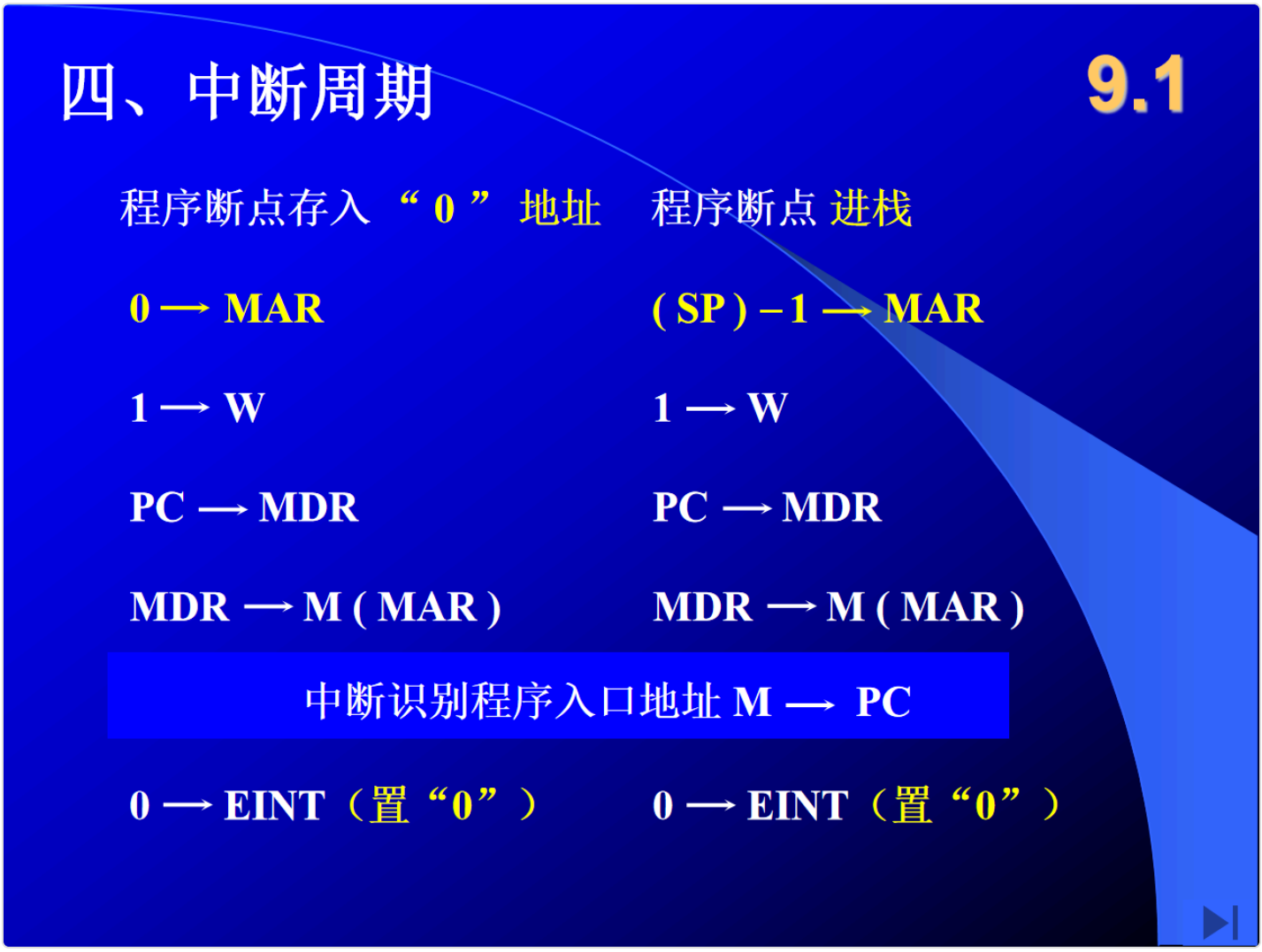
三类（五种）指令的指令周期（课件 p.473）

指令类型	指令周期组成
非访存指令	取指周期 + 执行周期
直接访存指令	取指周期 + 执行周期
间接访存指令	取指周期 + 间址周期 + 执行周期
转移指令	取指周期 + 执行周期
间接转移指令	取指周期 + 间址周期 + 执行周期



中断周期（课件 p.474）：中断周期由硬件自动完成，任务是保存断点、引出中断服务程序。课件给出两种保存断点方案：

步骤	方案一：断点存入"0"地址	方案二：断点进栈
1	0 → MAR	(SP) − 1 → MAR
2	1 → W	1 → W
3	PC → MDR	PC → MDR
4	MDR → M(MAR)	MDR → M(MAR)
5	中断识别程序入口地址 M → PC	中断识别程序入口地址 M → PC
6	0 → EINT（置"0"）	0 → EINT（置"0"）

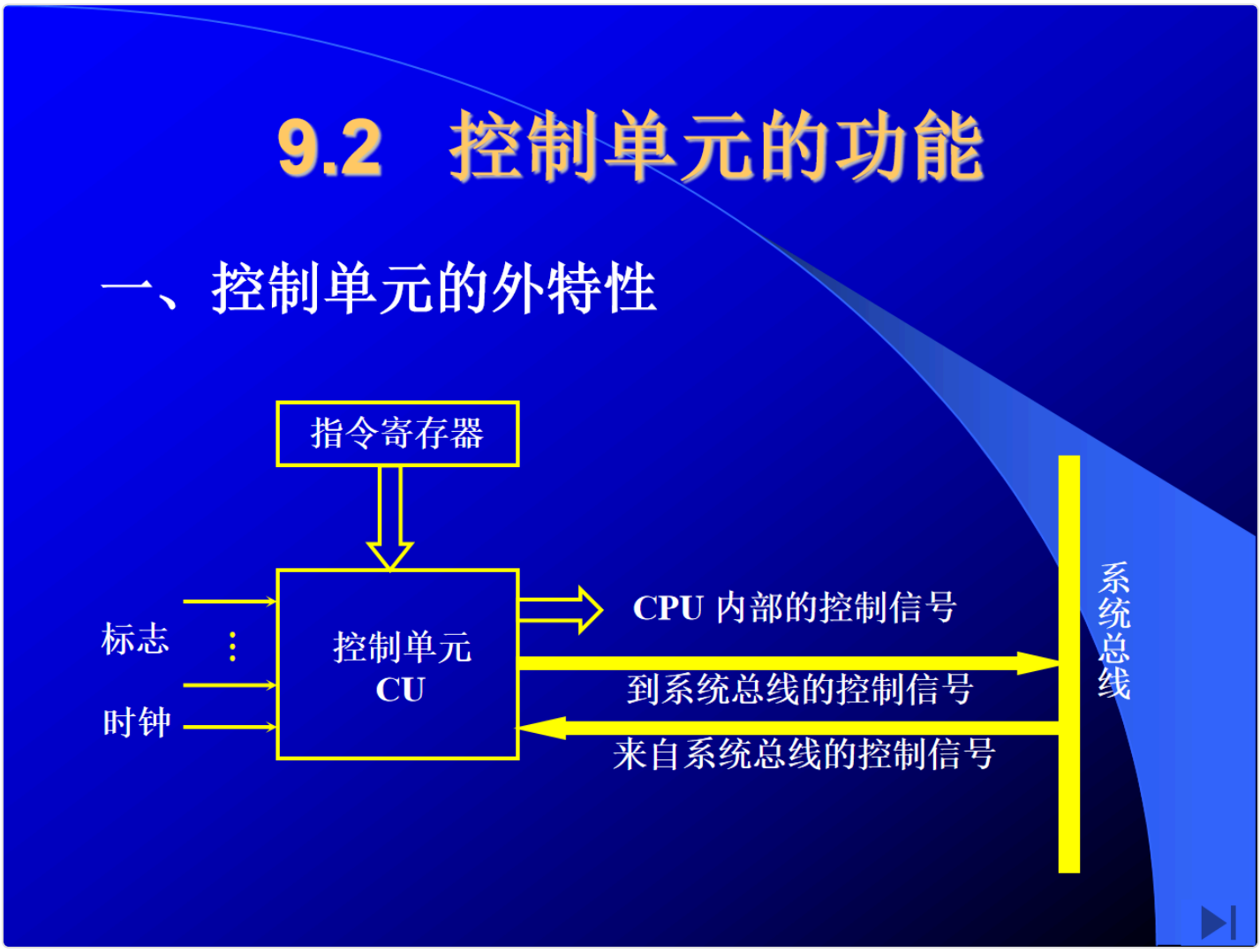


要点：- 断点就是被中断程序的下一条指令地址，即当前 PC 的内容，故保存断点 = 把 PC 写入存储器（课件 p.474）。- 0→EINT 是关中断（中断允许触发器清 0），保证保存断点和转入口的过程不被再次打断（课件 p.474）。- 进栈方案用 (SP)−1 作为栈顶地址送 MAR（课件 p.474 原图如此）。

易错警示：中断周期不是执行周期的一部分，它是独立的工作周期；"M→PC"送的是中断识别程序入口地址，不是断点。

5. 控制单元的外特性：输入信号与输出信号

外特性框图（课件 p.475）：CU 的输入有指令寄存器、标志、时钟以及来自系统总线的控制信号；输出有 CPU 内部的控制信号和到系统总线的控制信号。



1）输入信号（课件 p.476）

输入	作用
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
指令寄存器	OP(IR) → CU，控制信号与操作码有关
标志	CU 受标志控制（如条件转移依据符号位等）
外来信号	如 INTR（中断请求）、HRQ（总线请求）
来自系统总线的控制信号	外部件对 CPU 的控制/应答（课件 p.475 框图）

2）输出信号（课件 p.477）

类别	信号举例
CPU 内的各种控制信号	Ri → Rj；(PC) + 1 → PC；ALU 的 +、 - 、与、或 ……
送至控制总线的信号	$\overline{\text{MREQ}}$（访存控制信号）、IO/$\overline{\text{M}}$（访 IO/存储器控制信号）、$\overline{\text{RD}}$（读命令）、$\overline{\text{WR}}$（写命令）、INTA（中断响应信号）、HLDA（总线响应信号）

要点：带横线的信号（$\overline{\text{MREQ}}$、IO/$\overline{\text{M}}$、$\overline{\text{RD}}$、$\overline{\text{WR}}$）为低电平有效；INTA、HLDA 在课件中不带横线（课件 p.477）。

易错警示：INTR、HRQ 是 CU 的**输入**（外来请求），INTA、HLDA 才是 CU 的**输出**（响应），方向不要记反。

6. 控制信号举例：ADD @X 的两种数据通路方案

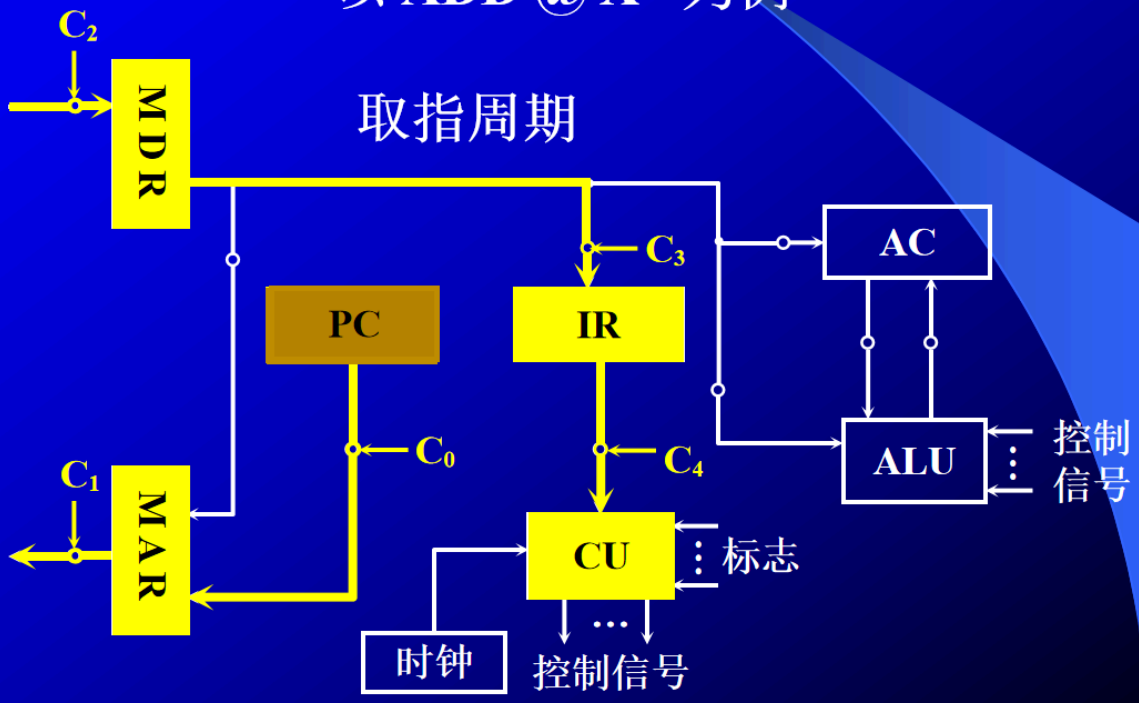
ADD @X 为间接寻址加法指令，完整经历取指、间址、执行三个周期。课件用两种数据通路结构对比控制信号的发出方式。

方案一：不采用 CPU 内部总线（课件 p.478–480）：各寄存器之间用专用连线，控制信号 C0–C8 直接开闭连线上的门。

二、控制信号举例

1. 不采用 CPU 内部总线的方式

以 ADD @ X 为例



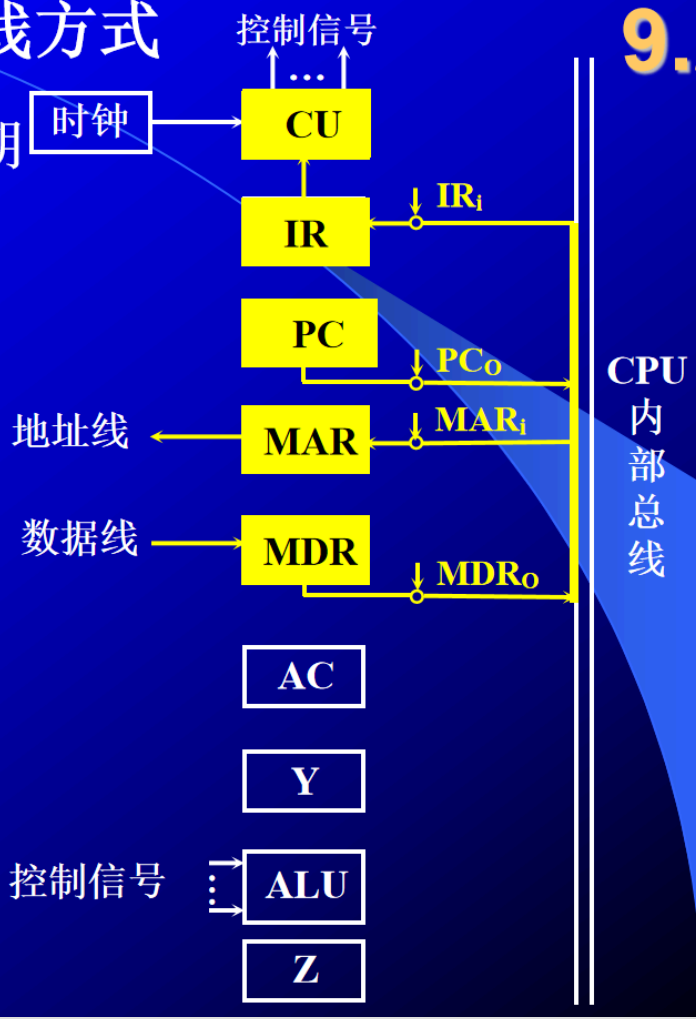
周期	微操作	控制信号
取指	PC → MAR	C0
取指	MAR → 地址线	C1
取指	数据线 → MDR（配合 1→R）	C2
取指	MDR → IR	C3
取指	OP(IR) → CU	C4
间址	Ad(IR) → MAR（顶部传送线上的地址送 MAR）	C5
间址	MAR → 地址线、1→R、数据线 → MDR	C1、C2
间址	MDR → Ad(IR)	C3
执行	MDR → MAR（有效地址）、MAR → 地址线	C5、C1
执行	1→R、数据线 → MDR	C2
执行	MDR → ALU	C7
执行	AC → ALU	C6
执行	ALU → AC（相加结果回送）	C8

方案二：采用 CPU 内部总线（课件 p.481–483）：所有寄存器挂在一组内部总线上，每个寄存器有"送总线"门（下标 o，如 PCo、MDRo、ACo、Zo）和"接总线"门（下标 i，如 MARi、IRi、Yi、ACi）。同一节拍内只能有一个寄存器向总线送数。

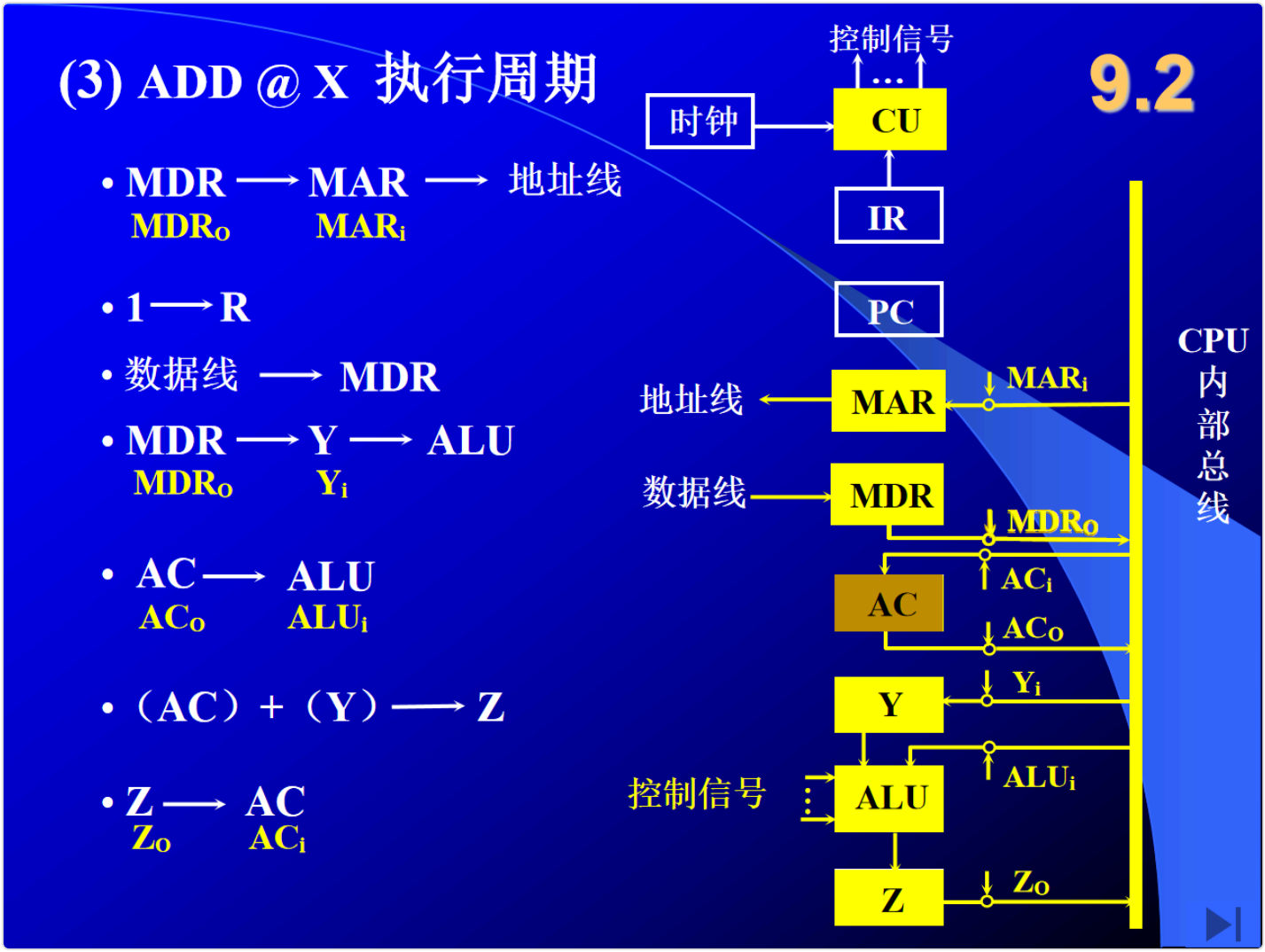
2. 采用 CPU 内部总线方式

(1) ADD @ X 取指周期

- PC⟶ MAR⟶ 地址线
  PC_o MAR_i
- CU 发读命令 1 ⟶R
- 数据线 ⟶ MDR
- MDR ⟶ IR
  MDR_o IR_i
- OP（IR）⟶ CU
- （PC）+ 1 ⟶PC



周期	微操作	门信号
取指	PC → MAR → 地址线	PCo、MARi
取指	CU 发读命令 1→R；数据线 → MDR	—
取指	MDR → IR	MDRo、IRi
取指	OP(IR) → CU；(PC)+1 → PC	—
间址	形式地址 Ad(IR) → MAR	IR 地址字段送总线门、MARi
间址	1→R；数据线 → MDR	—
间址	MDR → Ad(IR)（有效地址回送地址字段）	MDRo、IRi
执行	MDR → MAR → 地址线	MDRo、MARi
执行	1→R；数据线 → MDR	—
执行	MDR → Y → ALU	MDRo、Yi
执行	AC → ALU	ACo、ALUi
执行	(AC) + (Y) → Z	ALU 运算控制信号
执行	Z → AC	Zo、ACi



要点：- 单总线结构下"MDR→Y"与"AC→ALU"不能同一节拍完成（都要占用总线送数），必须分拍；这是总线结构省连线、费时间的代价（补充常识）。- 执行周期开始时有效地址已在 MDR 中（间址周期读出的内容），故用 MDR→MAR 而不是 Ad(IR)→MAR（课件 p.483）。- 课件 p.482 图中 IR 向总线送数的门仍标注为 IRi（与接收门同名），且该页保留了"MDR→MAR→地址线""MDR→IR"等字样。课件原文如此，存疑：送数门应为 IR 输出门（IRo），间址周期微操作应以 p.469 序列（Ad(IR)→MAR；1→R；M(MAR)→MDR；MDR→Ad(IR)）为准。

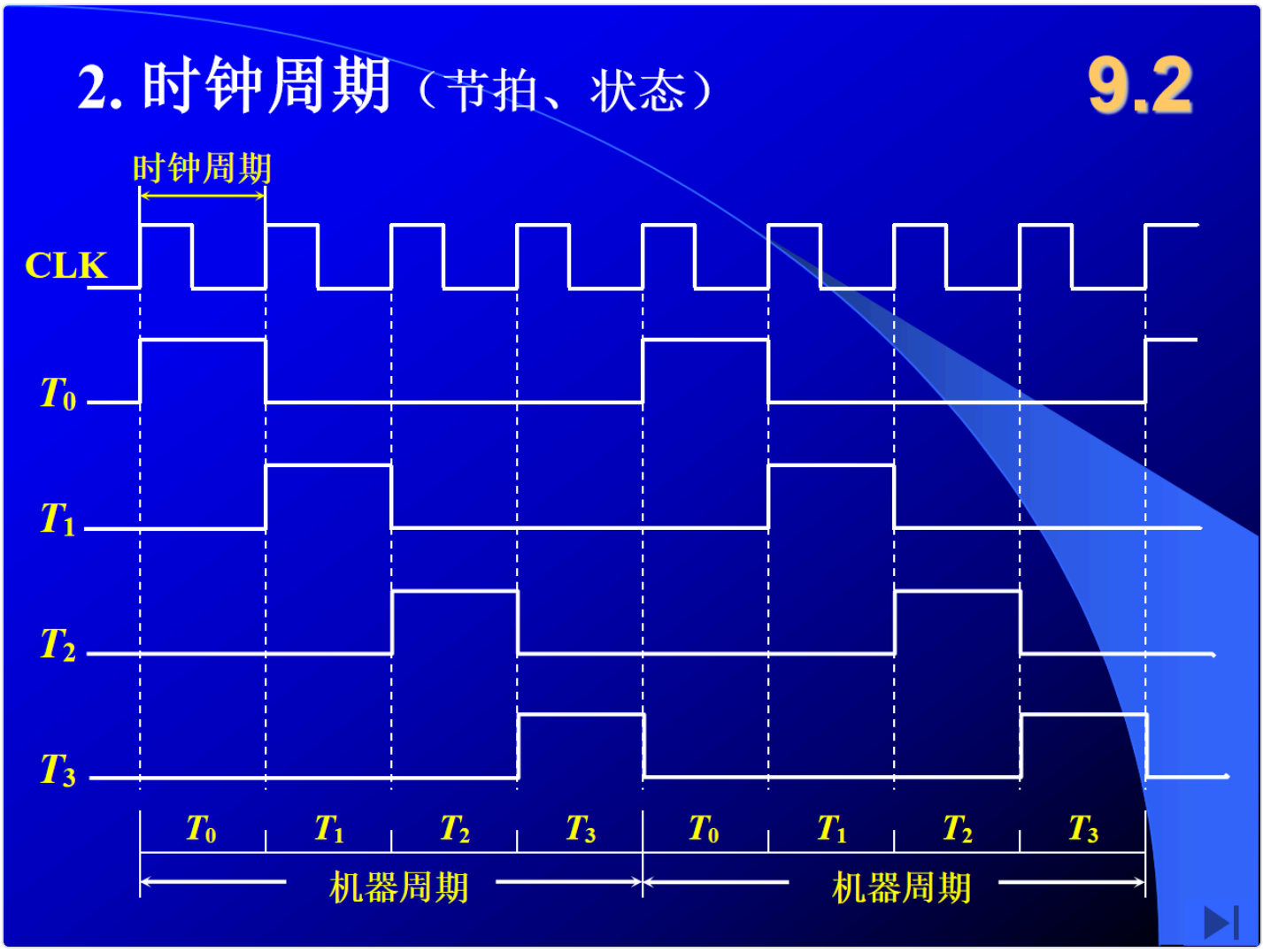
易错警示：o = out（送总线）、i = in（接总线）；写微操作序列时"谁送谁收"必须配成一对门信号，漏写接收门或送数门都扣分。

7. 多级时序系统：机器周期、节拍、时钟周期

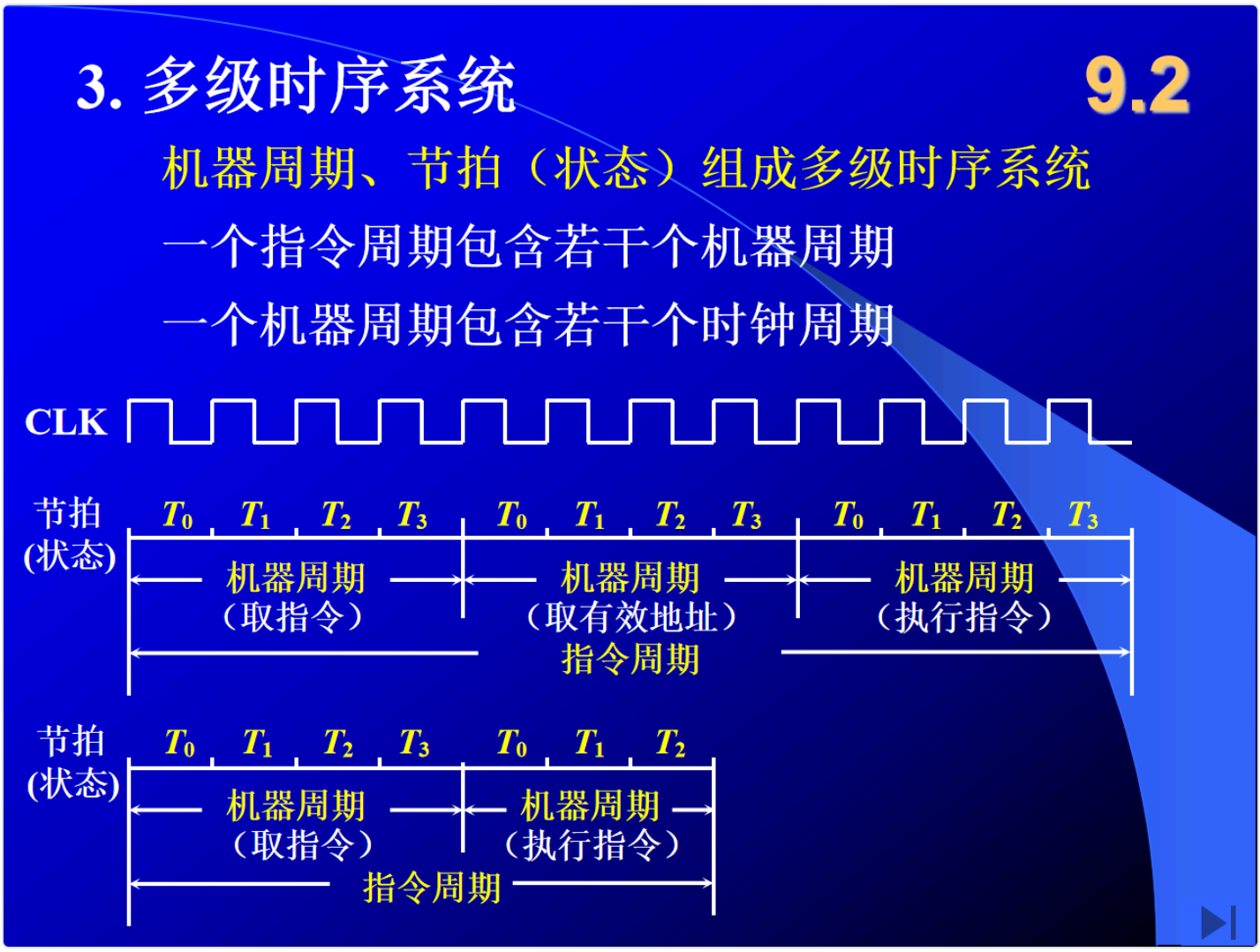
1）机器周期（课件 p.484）- 概念：所有指令执行过程中的一个**基准时间**。- 确定机器周期需考虑的因素：每条指令的执行**步骤**；每一步骤所需的**时间**。- 基准时间的确定：以完成**最复杂**指令功能的时间为准；以**访问一次存储器**的时间为基准。- 重要结论：若**指令字长 = 存储字长**，则**取指周期 = 机器周期**（课件 p.484）。

2）时钟周期（节拍、状态）（课件 p.485–486）- 一个机器周期内可完成若干个微操作，每个微操作需一定时间。- 将一个机器周期分成若干个**时间相等**的时间段，称为节拍、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（课件 p.485）。



3）多级时序系统（课件 p.487） - 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**。 - 课件示例：间接访存指令 = 3 个机器周期（取指令、取有效地址、执行指令），每周期 4 拍 T0–T3；直接访存指令 = 2 个机器周期（4 拍 + 3 拍）。



层次关系（必背）

```
指令周期 ⊃ 机器周期 ⊃ 时钟周期（节拍、状态）
指令周期：取出并执行一条指令的时间
机器周期：基准时间（常以访存一次时间为基准）
时钟周期：最小单位时间 = 主频的倒数
```

易错警示：机器周期不是"执行一条指令的时间"（那是指令周期）；节拍数在不同机器周期中可以不等（见第9节不定长方案），但每个节拍宽度相等。

8. 机器速度与机器主频的关系

课件结论（课件 p.488）： - 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：

$$MIPS_1 / MIPS_2 = f_1 / f_2$$

- 机器速度不仅与主频有关，还与**机器周期中所含时钟周期数**（时钟周期 = 主频的倒数）以及**指令周期中所含的机器周期数**有关（课件 p.488）。

计算用展开式（补充常识，由课件层次关系直接推出）：

```
时钟周期 T = 1 / f
指令执行时间 = 指令含机器周期数 × 机器周期含节拍数 × T
MIPS = f(MHz) / 每条指令所含时钟周期数
```

易错警示：MIPS 之比 = 主频之比**有前提**（机器周期所含时钟周期数相同，且指令系统/平均机器周期数可比）。主频高而每指令拍数多，速度未必快——这是计算题最爱设的陷阱。

9. 控制方式：同步、异步、联合、人工

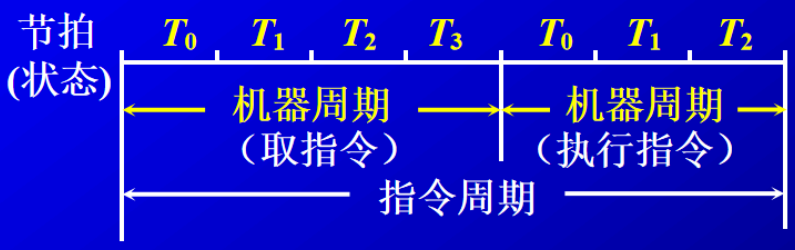
定义：控制方式是产生不同微操作命令序列所用的时序控制方式（课件 p.489）。

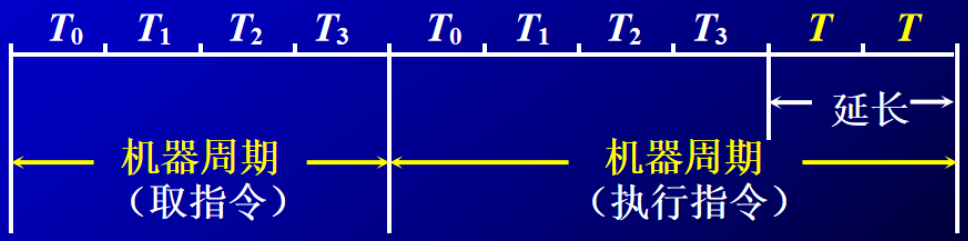
1）同步控制方式：任一微操作均由**统一基准时标**的时序信号控制（课件 p.489）。三种实现方案：

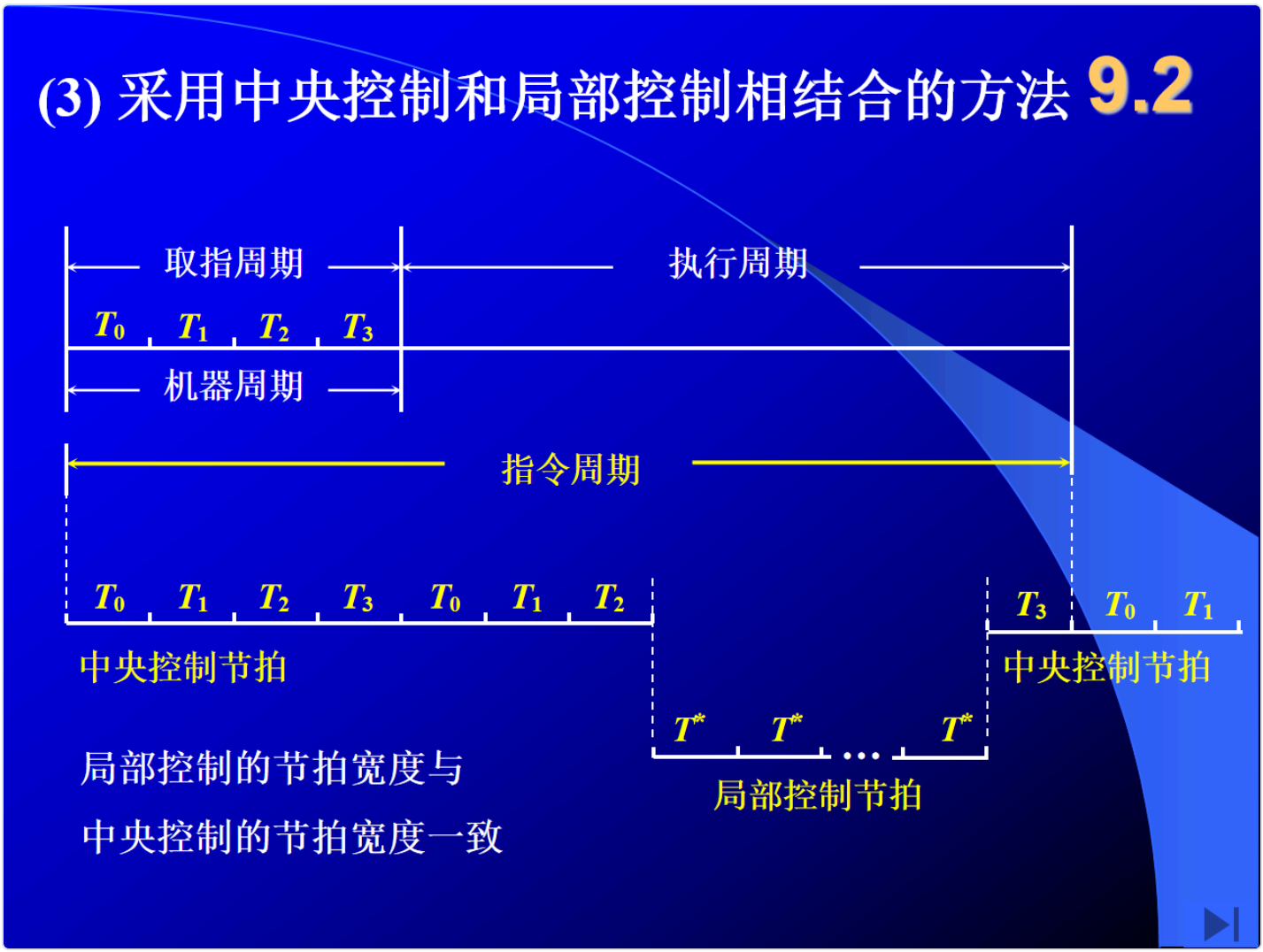
方案	特点	课件页
(1) 定长的机器周期	以**最长的微操作序列**和**最繁的微操作**作为标准；机器周期内**节拍数相同**	p.489
(2) 不定长的机器周期	机器周期内**节拍数不等**；短周期可省去多余节拍，必要时用延长节拍 T 扩展	p.490
(3) 中央控制和局部控制相结合	大部分操作用中央控制节拍；少数复杂操作用局部控制节拍 T*；**局部控制的节拍宽度与中央控制的节拍宽度一致**	p.491

(2) 采用不定长的机器周期

机器周期内 节拍数不等







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**（握手）（课件 p.492）。

3）联合控制方式：**同步与异步相结合**（课件 p.492）。

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（课件 p.492）。

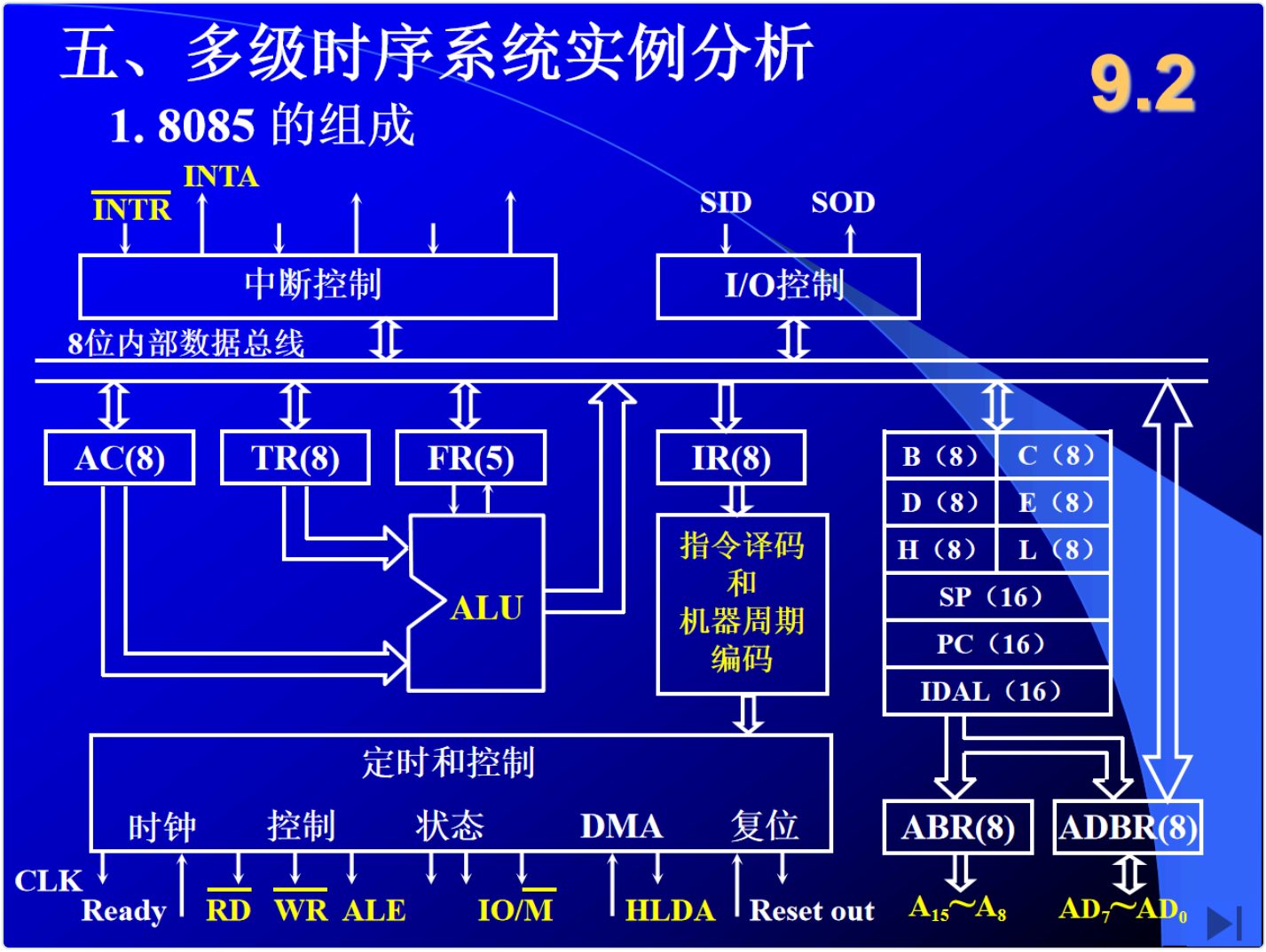
四种控制方式对比表

方式	有无统一时标	节拍/周期	典型手段
同步	有	固定或按方案变化	统一时钟、定长/不定长/中央+局部
异步	无	无固定周期节拍	应答（握手）
联合	部分有	同步为主、异步为辅	同步+异步结合
人工	由人决定	单条/连续	Reset、单条开关、符合停机开关

易错警示：不定长机器周期**仍属同步控制**（仍有统一基准时标），不要因"节拍数不等"误判为异步；异步的标志是"无基准时标 + 应答方式"。

10. 多级时序系统实例：8085

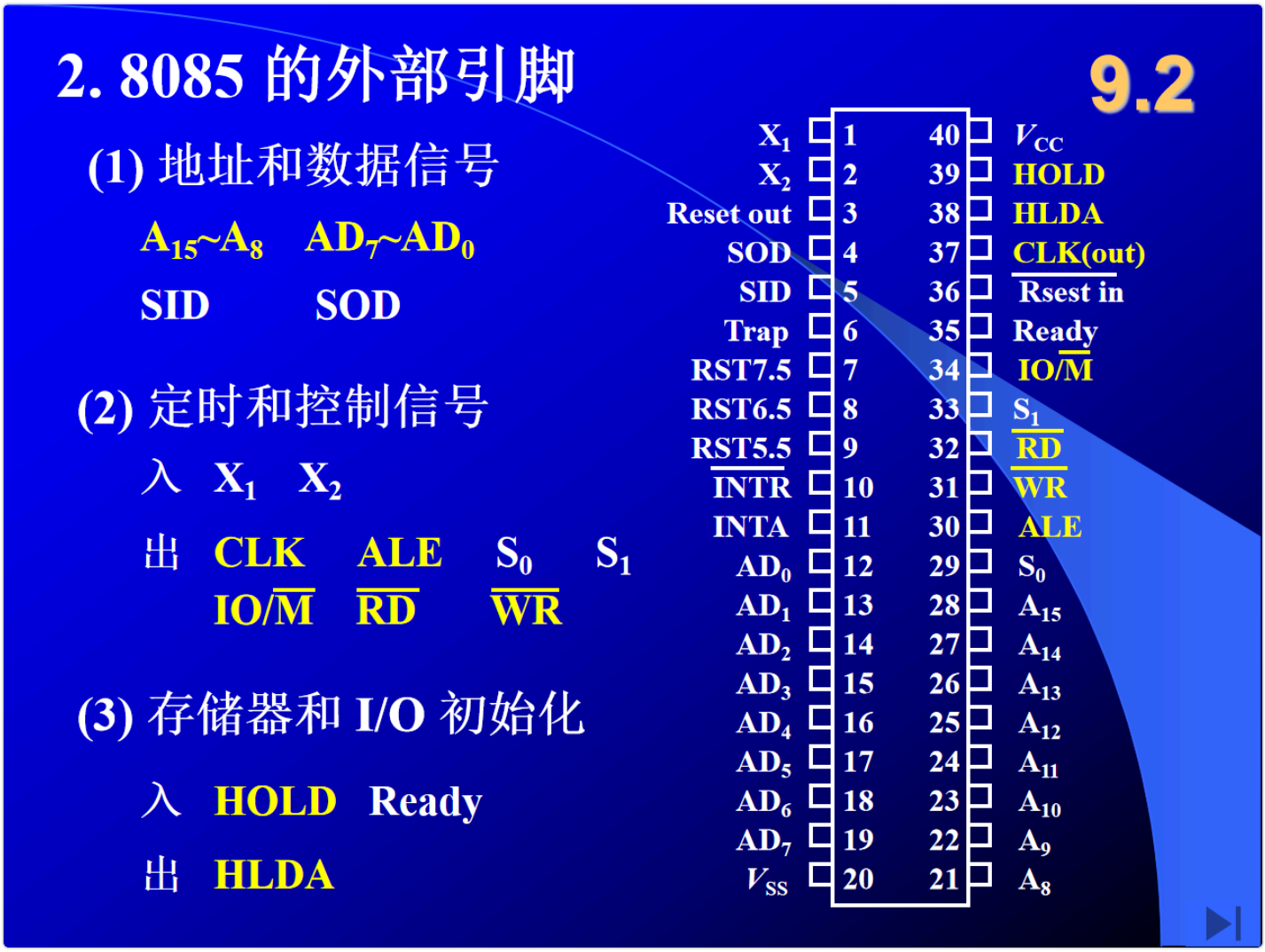
1）8085 的组成（课件 p.493）



部件	位数/说明
AC（累加器）、TR（暂存器）、IR（指令寄存器）	各 8 位
FR（标志寄存器）	5 位
B、C、D、E、H、L 通用寄存器	各 8 位
SP、PC、IDAL（增量/减量地址锁存器）	各 16 位
ABR（地址缓冲寄存器）、ADBR（地址/数据缓冲寄存器）	各 8 位
内部数据总线	8 位
指令译码和机器周期编码、定时和控制、中断控制、I/O 控制	控制部件

对外信号：CLK、Ready、$\overline{RD}$、$\overline{WR}$、ALE、IO/$\overline{M}$、HLDA、Reset out、INTR、INTA、SID、SOD、A15~A8、AD7~AD0（课件 p.493）。

2）8085 的外部引脚（40 脚）（课件 p.494–495）

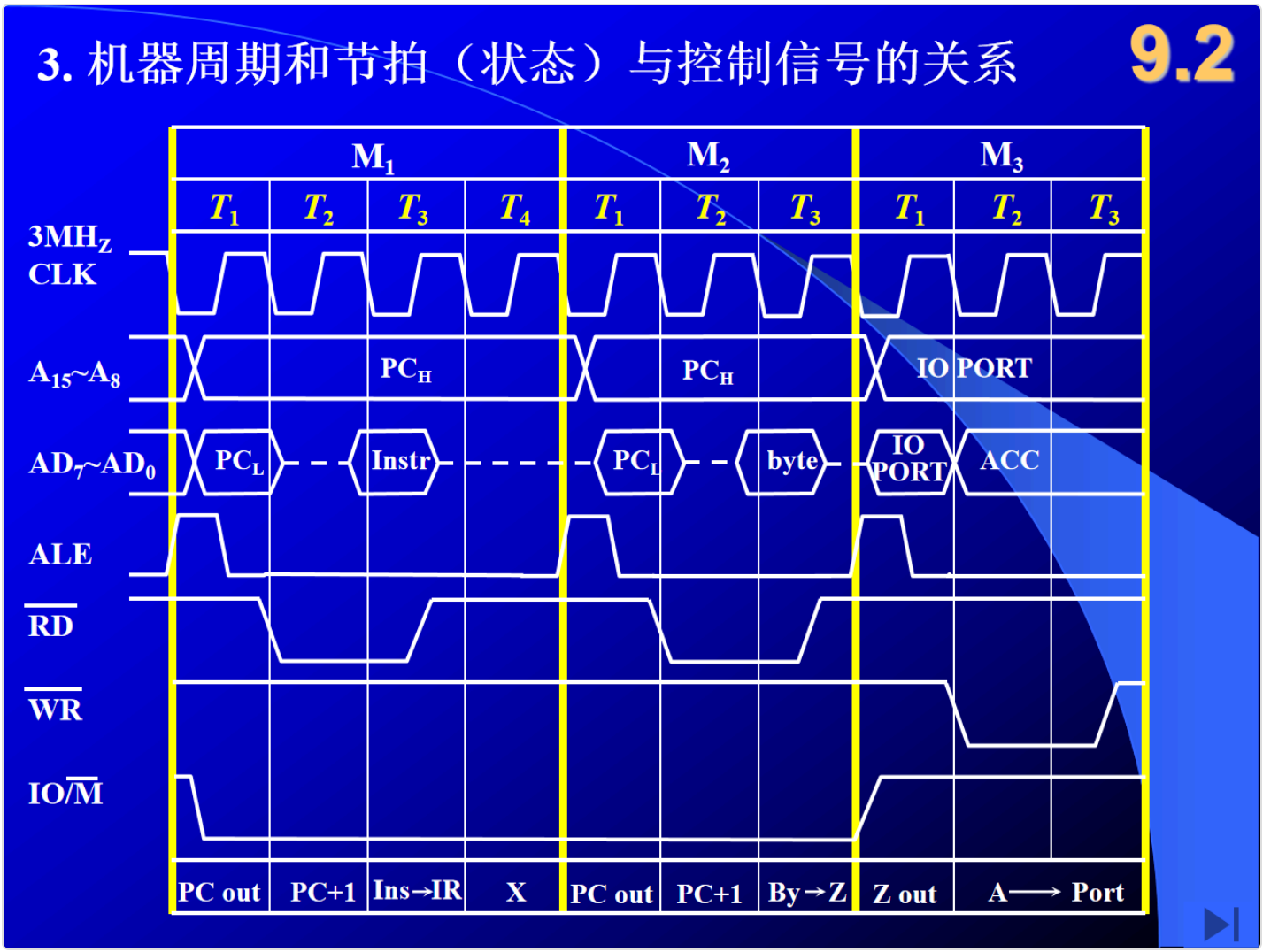


分类	信号	方向
地址和数据	A15~A8（高 8 位地址）；AD7~AD0（低 8 位地址/数据复用）	出/双向
串行数据	SID（串行输入）、SOD（串行输出）	入/出
定时和控制	X1、X2（时钟输入）；CLK(out)、ALE、S0、S1、IO/$\bar{M}$、$\overline{RD}$、$\overline{WR}$	入/出
存储器和 I/O 初始化	HOLD、Ready（入）；HLDA（出）	入/出
中断	INTR（入）、INTA（出）、Trap（重新启动中断）、RST7.5、RST6.5、RST5.5	入/出
CPU 初始化	Reset in（入）、Reset out（出）	入/出
电源和地	VCC = +5 V、VSS = 地	—

引脚号对照（课件 p.494 原图）：1 X1、2 X2、3 Reset out、4 SOD、5 SID、6 Trap、7 RST7.5、8 RST6.5、9 RST5.5、10 INTR、11 INTA、12–19 AD0–AD7、20 VSS、21–28 A8–A15、29 S0、30 ALE、31 $\overline{WR}$、32 $\overline{RD}$、33 S1、34 IO/$\bar{M}$、35 Ready、36 Reset in、37 CLK(out)、38 HLDA、39 HOLD、40 VCC。课件引脚图中 pin36 印作"Rsest in"，课件原文如此，存疑：应为 Reset in 的排版笔误。

要点：AD7~AD0 分时复用——每个机器周期的 T1 节拍由 **ALE** 脉冲把低 8 位地址锁存到外部锁存器，其余节拍传数据（课件 p.496 时序可见 ALE 只在 T1 出现）。

3）机器周期、节拍与控制信号的关系：I/O 写（输出指令）实例（课件 p.496–497）



机器周期	功能（课件 p.497）	节拍与操作	关键信号
M1（T1–T4）	取指令操作码	T1：PC out（PCH 上 A15~A8、PCL 上 AD7~AD0，ALE 锁存）；T2：PC+1，RD̄ 有效；T3：Ins→IR（操作码上数据线）；T4：X（译码）	IO/M̄=0，RD̄ 低
M2（T1–T3）	取设备地址	T1：PC out（PCH/PCL，ALE）；T2：PC+1，RD̄ 有效；T3：By→Z（读出的端口地址字节送暂存器 Z）	IO/M̄=0，RD̄ 低
M3（T1–T3）	执行：ACC 内容写入设备	T1：Z out（IO PORT 上地址总线）；T2–T3：A→Port（ACC 上 AD7~AD0，WR̄ 有效）	IO/M̄=1，WR̄ 低

小结结论（课件 p.497）：每个控制信号在**指定机器周期的指定节拍 T 时刻**发出。这正体现了 CU 的功能 = 操作码 + 时序 → 微操作命令序列。

易错警示：M1、M2 是访存（IO/M̄=0），M3 才是访 I/O（IO/M̄=1）；WR̄ 只在 M3 的 T2–T3 有效，RD̄ 只在 M1、M2 的 T2–T3 有效；ALE 每个机器周期只在 T1 出现一次。

11. 易错点与高频陷阱汇总

1. 间址周期只服务于间接寻址指令；其末步是 MDR→Ad(IR)，写成 MDR→IR 即错（课件 p.469）。
2. (PC)+1→PC 发生在**取指周期**，不是执行周期（课件 p.468）。
3. BAN X 的表达式是 $A_0 \cdot Ad(IR) + \bar{A}_0 \cdot (PC) \rightarrow PC$：$A_0=1$ 才转，$A_0=0$ 顺序执行（课件 p.472）。
4. SHR 算术右移符号位不变（$ACC_0 \rightarrow ACC_0$）；CSL 循环左移是 $ACC_0 \rightarrow ACC_n$（课件 p.470）。
5. STP 是 0→G（清运行标志），不是 0→ACC（课件 p.470）。
6. 中断周期由硬件完成；0→EINT 是关中断；M→PC 送的是中断识别程序入口地址而非断点（课件 p.474）。
7. 指令周期 ⊃ 机器周期 ⊃ 时钟周期；"取指周期=机器周期"仅在指令字长=存储字长时成立（课件 p.484、p.487）。
8. 时钟周期（节拍、状态）是最小单位时间；一个时钟脉冲可发**一组需同时执行**的操作命令，不是只能发一个（课件 p.476、p.485）。
9. $MIPS_1/MIPS_2 = f_1/f_2$ 的前提是机器周期所含时钟周期数相同；主频高≠一定快（课件 p.488）。
10. 不定长机器周期、中央+局部控制都属**同步**控制；异步的标志是无基准时标、采用应答方式（课件 p.489–492）。
11. 局部控制节拍宽度必须与中央控制节拍宽度一致（课件 p.491）。
12. INTR、HRQ、HOLD、Ready 是输入；INTA、HLDA、$\overline{RD}$、$\overline{WR}$、ALE 是输出，方向记反必丢分（课件 p.476–477、p.494）。
13. 单总线结构中同一节拍只能一个寄存器送总线：MDR→Y 与 AC→ALU 必须分拍（课件 p.483）。
14. 8085 的 AD7~AD0 复用，T1 靠 ALE 锁存低地址；IO/$\bar{M}$=1 为访 I/O、=0 为访存（课件 p.494、p.496）。
15. 微命令是控制信号、微操作是信号引起的操作、微指令是同时发出的微命令集合（补充常识），三者层次不要混。

12. 典型例题精讲

例1（微操作序列 + 总线控制信号，综合） 写出单总线结构下 ADD @X 执行周期的微操作序列，并配上门控制信号。

解析步骤： 1. 明确执行周期起点：间址周期已把有效地址读入 MDR（课件 p.482–483），故第一步用 MDR 送地址。 2. 访存取操作数：MDR→MAR→地址线（MDRo、MARi）；1→R；数据线→MDR。 3. 操作数送 ALU 暂存器 Y：MDR→Y（MDRo、Yi）。注意此拍总线被 MDR 占用。 4. 累加器送 ALU：AC→ALU（ACo、ALUi）。必须另起一拍，因为单总线一拍只能一个送数源。 5. 相加：(AC)+(Y)→Z（ALU 运算控制信号）。 6. 结果回送：Z→AC（Zo、ACi）。 结论：共 6 步微操作、占用多个节拍；评分点在"MDRo/MARi、MDRo/Yi、ACo/ALUi、Zo/ACi 成对出现"和"Y 与 AC 分拍送数"。

例2（时序计算） 某机主频 500 MHz，每个机器周期含 5 个时钟周期，平均每条指令含 2 个机器周期。求时钟周期、平均指令执行时间和 MIPS。

解析步骤： 1. $T = 1/f = 1/(500\times10^6\ Hz) = 2\times10^{-9}\ s = 2\ ns$。 2. 机器周期 = 5×T = 10 ns。 3. 指令执行时间 = 2×10 ns = 20 ns。 4. $MIPS = 1/(20\times10^{-9}\ s)/10^6 = 50$，即 50 MIPS；或用 MIPS = f(MHz)/每指令时钟周期数

= 500/10 = 50。 结论：2 ns、20 ns、50 MIPS。

例3（8085 时序读图） 读课件 p.496 时序图，回答：输出指令的 $\overline{WR}$ 在哪个机器周期哪些节拍有效？M2 周期 T3 完成什么操作？IO/$\overline{M}$ 在 M3 为何电平？

解析步骤： 1. 输出指令 = M1 取操作码 + M2 取设备地址 + M3 写端口（课件 p.497）。 2. 写操作只发生在 M3：图中 $\overline{WR}$ 在 M3 的 T2–T3 为低，故 $\overline{WR}$ 于 M3 的 T2、T3 有效。 3. M2 的 T3 标注 By→Z：把读到的端口地址字节送暂存器 Z，供 M3 的 T1 由 Z out 送上地址总线。 4. M3 访问 I/O 端口，故 IO/$\overline{M}$=1（高电平）；M1、M2 访存为 0。 结论：$\overline{WR}$ 在 M3 的 T2–T3 有效；M2-T3 为 By→Z；M3 期间 IO/$\overline{M}$=1。

附：本章速查表

微操作序列速查

周期/指令	序列
取指	PC→MAR→地址线；1→R；M(MAR)→MDR；MDR→IR；OP(IR)→CU；(PC)+1→PC
间址	Ad(IR)→MAR；1→R；M(MAR)→MDR；MDR→Ad(IR)
中断（存0地址）	0→MAR；1→W；PC→MDR；MDR→M(MAR)；M→PC；0→EINT
中断（进栈）	(SP)−1→MAR；1→W；PC→MDR；MDR→M(MAR)；M→PC；0→EINT
ADD X	Ad(IR)→MAR；1→R；M(MAR)→MDR；(ACC)+(MDR)→ACC
STA X	Ad(IR)→MAR；1→W；ACC→MDR；MDR→M(MAR)
LDA X	Ad(IR)→MAR；1→R；M(MAR)→MDR；MDR→ACC
JMP X / BAN X	Ad(IR)→PC / A_0·Ad(IR)+$\bar{A}_0$·(PC)→PC

CU 输入输出速查：输入 = 时钟、OP(IR)、标志、外来信号（INTR、HRQ）、来自系统总线的控制信号；输出 = CPU 内部控制信号（Ri→Rj、(PC)+1→PC、ALU 操作）+ 控制总线信号（$\overline{MREQ}$、IO/$\overline{M}$、$\overline{RD}$、$\overline{WR}$、INTA、HLDA）。

时序与速度公式

```
指令周期 ⊃ 机器周期 ⊃ 时钟周期（节拍、状态）
T = 1/f；指令字长=存储字长 ⇒ 取指周期=机器周期
MIPS₁/MIPS₂ = f₁/f₂（前提：机器周期所含时钟周期数相同）
指令执行时间 = 机器周期数/指令 × 节拍数/机器周期 × T
```

控制方式速查：同步（统一时标：定长/不定长/中央+局部）；异步（无时标、应答）；联合（同步+异步）；人工（Reset、连续/单条开关、符合停机开关）。

8085 速查：8 位内部总线；AC/TR/IR 8 位、FR 5 位、SP/PC/IDAL 16 位；AD7~AD0 复用、T1 用 ALE 锁存；IO/$\overline{M}$=1 访 I/O；$\overline{RD}$/$\overline{WR}$ 低有效；HOLD 入/HLDA 出；INTR 入/INTA 出；VCC=+5V；输出指令 = M1 取操作码(4拍) + M2 取设备地址(3拍) + M3 写端口(3拍)。