计算机组成原理

第三章 中央处理器

2025-8-22

信息与软件工程学院
School of Information and Software Engineering

主 要 内 容

1. 模型机的总体设计
2. 算术逻辑运算部件
3. 运算方法
4. 模型机的组合逻辑控制器
5. 模型机的微程序控制器
6. MIPS32架构CPU设计实例

引言

中央处理器（CPU）是计算机系统的核心部件：

- 运算器
- 控制器

通过本章的学习，应在CPU一级上建立起整机的概念：

1、CPU的逻辑组成：内部有哪些部件，以数据通路为核心的总体结构，与外部的连接方式。

2、CPU如何工作，即如何分时形成控制命令序列，以执行指令序列，分为：

（1）从寄存器传送级分析指令的执行流程；

（2）从微操作命令序列一级分析寄存器级传送的具体实现。

3.1 模型机的总体设计

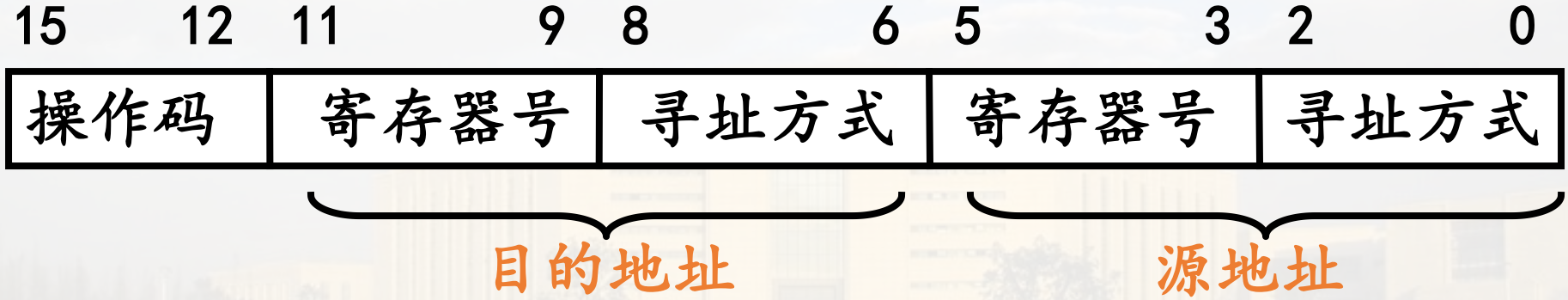
- 01. 模型机指令系统
- 02. CPU的组成
- 03. CPU的内部数据通路结构
- 04. 主机与外部的数据通路及信息传送控制方式
- 05. 时序控制方式与时序系统
- 06. 同步控制的时序系统

一、模型机指令系统

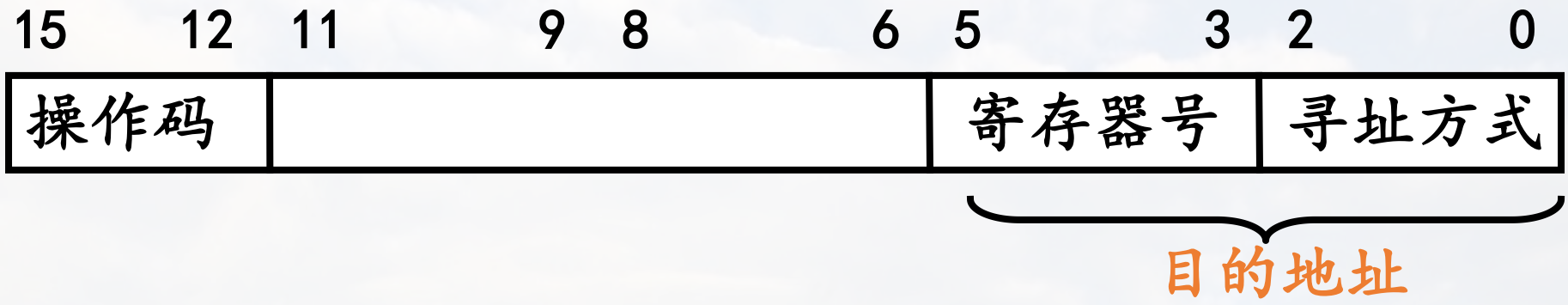
（1）指令格式

模型机指令格式分类
- 双操作数指令
- 单操作数指令
- 转移指令

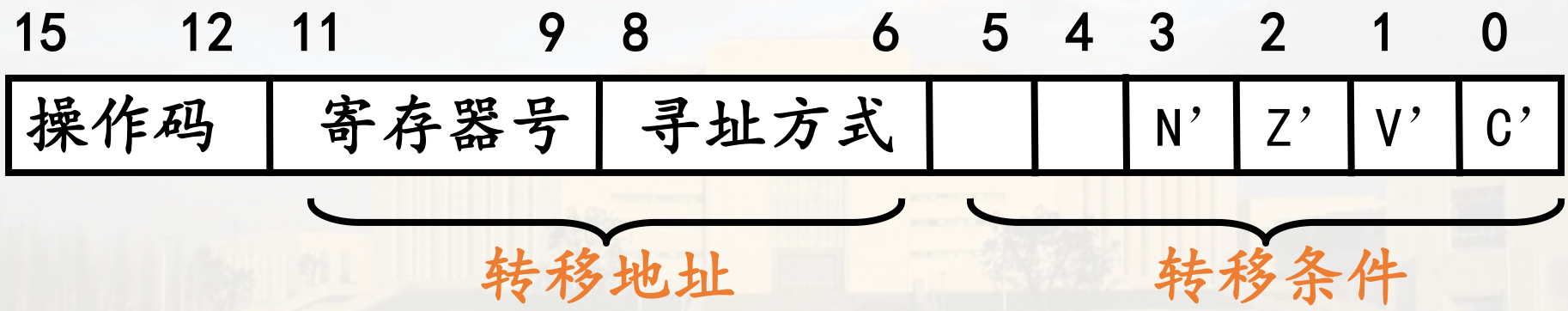
1、双操作数指令



2、单操作数指令



3、转移指令



（2）寻址方式

模型机的编址为按字编址，字长16位，采用定字长指令格式，即指令字长16位。寻址方式共以下八种：

序号	寻址方式	助记符	可指定寄存器
1	立即寻址	I	无
2	寄存器寻址	R	R_0∽R_3，SP、PC、PSW
3	寄存器间址	(R)	R_0∽R_3

一、模型机指令系统

序号	寻址方式	助记符	可指定寄存器
4	自减型寄存器间址	-(R)	R_0∽R_3、SP
5	自增型寄存器间址	(R)+	R_0∽R_3、SP、PC
6	自增型双间址	@(R)+ @(PC)+	R_0∽R_3、PC
7	变址方式	X(R) X(PC)	R_0∽R_3、PC
8	跳步方式	SKP	

一、模型机指令系统

寻址方式	操作数地址1（间址地址、形式地址的地址）	操作数地址2	操作数位置	操作数的访存次数
立即寻址：I			指令中，实际在IR中	0
寄存器直接寻址：R		指令给出R编号（即操作数地址）	R中	0
R寄存器间接寻址：(R)	指令给出R编号（间接地址）	R中给出M单元号（即操作数地址）	M中	1
自增型寄存器间接寻址：(R)+、 (SP)+、(PC)+	指令给出R编号（间接地址）	R中给出M单元号（即操作数地址）	M中	1
自减型寄存器间接寻址：-(R)、-(SP)	指令给出R编号	R中内容减1后，为M单元号（即操作数地址）	M中	1
自增型双间址：@(R)+	指令给出R编号（间址地址）	从M中取回的操作数地址（即操作数地址）	M中	2
变址寻址：X(R)	指令给出M单元号（形式地址的地址）	指令给出R编号（即变址地址N），及从M中取回的形式地址D；运算后的M单元值A=D+N（即操作数地址）	M中	2
跳步：SKP				

（3）操作类型

操作码共4位，设置15种指令（14种编码方式），

余下两种操作码组合可供扩充：

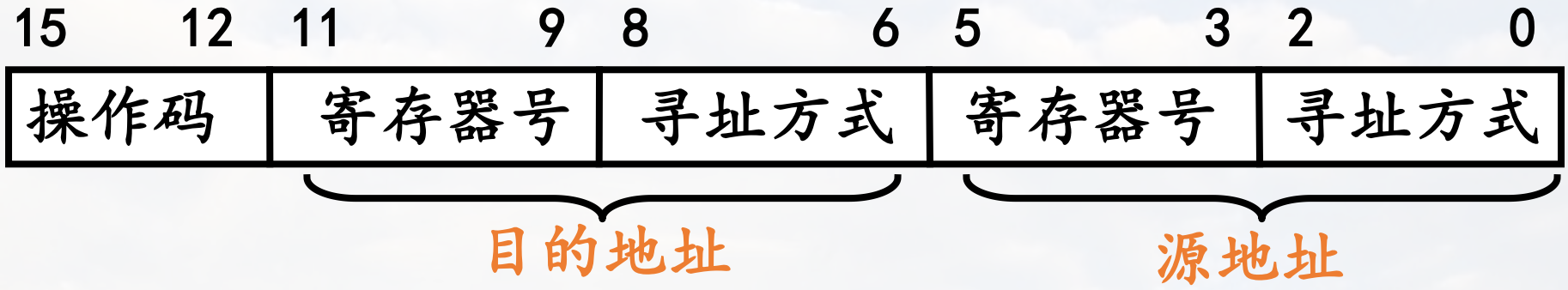
1、传送指令：一种

2、双操作数算逻指令：五种

3、单操作数算逻指令：六种

4、程序控制类指令：三种，但其中两种操作码相同

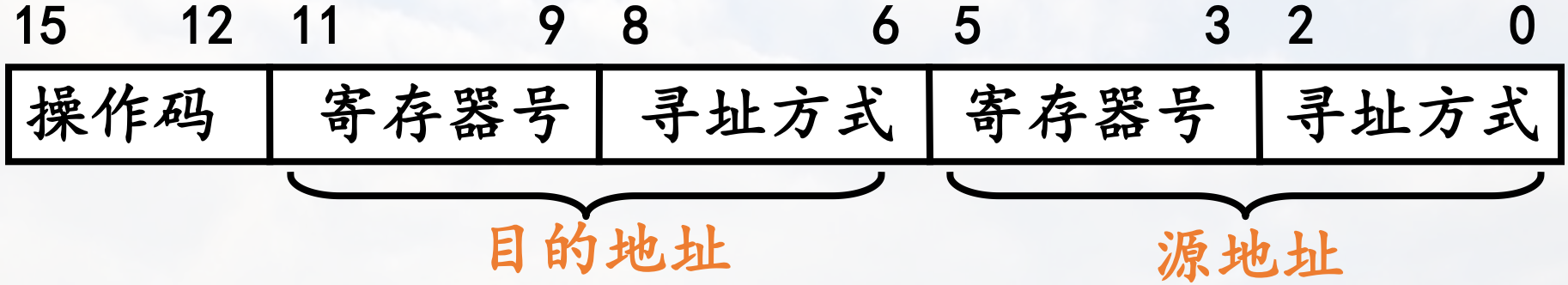
（3）操作类型



模型机中的二进制指令表示：

1）0000	000	001	001	011
2）0010	011	100	111	101
3）0101	010	000	100	010

(3) 操作类型



1) 0000 000 001 001 011

→ MOV (R_0), (R_1)+ (汇编指令表示)

2) 0010 011 100 111 101

→ SUB @(R_3)+, X(PC) (汇编指令表示)

3) 0101 010 000 100 010

→ EOR R_2, -(SP) (汇编指令表示)

二、CPU的组成

CPU组成

- **运算器**：三级，输入选择器/锁存器—>ALU—>移位器
- **寄存器组**：三组，用于处理、控制、作用于主存接口的寄存器
- **总线**：四组，CPU内总线、系统总线、部件间总线，外总线
- **控制器**：组合逻辑控制器/微程序控制器
- **时序系统**：一个脉冲源、一组计数分频逻辑

二、CPU的组成

1、运算器

运算器三级
- 输入选择器/锁存器
- ALU
- 移位器(通过斜位传送实现移位功能)

1）输入选择器/锁存器

选择数据来源，送入ALU进行运算处理，或借道ALU进行传送。

数据来源有：R_0∽R_3、C、D、PC、SP、PSW、MDR。

2）ALU部件

作各种算术，逻辑运算；由微命令M，S_0，S_1，S_2，S_3，C_0选择操作功能。

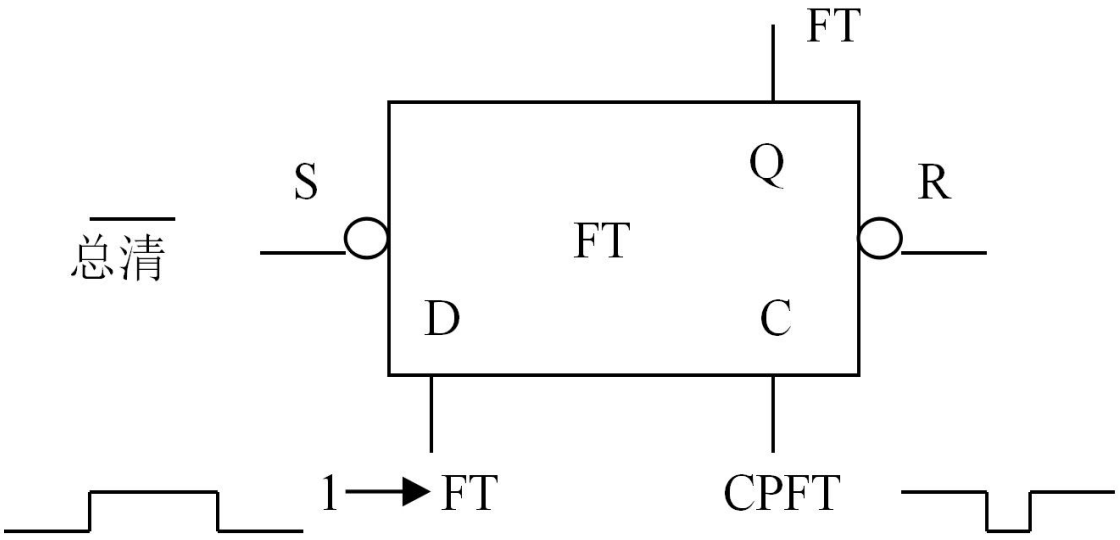
3）移位器

作直接传送、左移、右移；由微命令实现直接、左、右移。

2、寄存器组

设置的寄存器都是16位，内部结构是16个集成D触发器，代码输入至D端，CP端同步打入，还可选由R、S端异步置入（这种方式速度更快）。

集成D触发器的结构：



1）用于处理的寄存器：通用寄存器、暂存器

①通用寄存器组

一组可编程访问的寄存器。

在指令系统中为7个R分配了编号，有：R_0＝000，R_1＝001，

R_2＝010，R_3＝011，SP=100，PSW=101，PC=111

②暂存器

用户不能直接访问的R，用来暂存信息，在指令系统中没有为它们分配编号，有C、D。

暂存器C：从主存中读取源操作数或源操作数地址时使用它。

暂存器D：从主存中读取目的作数或目的操作数地址时，或中间运算结果时，使用它。

2）用于控制的寄存器：指令寄存器IR、程序计数器PC、程序状态字寄存器PSW

①指令寄存器IR(Instruction Register)

用来存放现行运行指令，它的输出是产生微操作命令序列的主要逻辑依据。

为了提高读取指令的速度，在主存与IR间建立直接传送通路，并且将IR扩充为指令队列（或指令栈），允许预取若干条指令。如：8086/8088指令队列：6/4（B）。

②程序计数器PC(Program Counter)

PC提供后继指令地址，并送往与主存相连接的地址寄存器（MAR）。

后继指令地址=PC+n，模型机中为了简化起见，令n=1

③ 程序状态字寄存器PSW（Program Status Word）

PSW的内容就是表现的现行程序的运行状态。包括：

- 特征位：进位C、溢出V、零Z、负N，允许中断I等；
- 编程设定位

3）用作主存接口的寄存器：地址寄存器MAR、

数据缓冲寄存器MBR/MDR

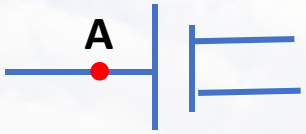
CPU访问主存时，首先送出地址码，然后送出/接收数据，即：

A. 当作用在MAR上的微命令EMAR为低电平时，MAR输出呈高阻态，与地址总线断开；

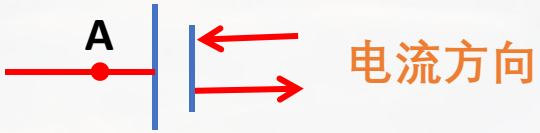
B. 当作用在MAR上的微命令EMAR为高电平时，MAR输出其内容（地址信息）送往地址总线；

二、CPU的组成

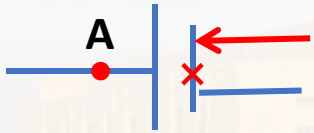
1）场效应管初始状态：



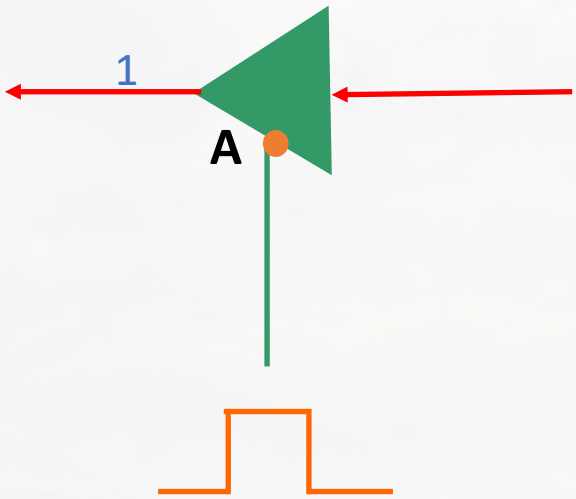
场效应管状态：当A点为高电平时，导通



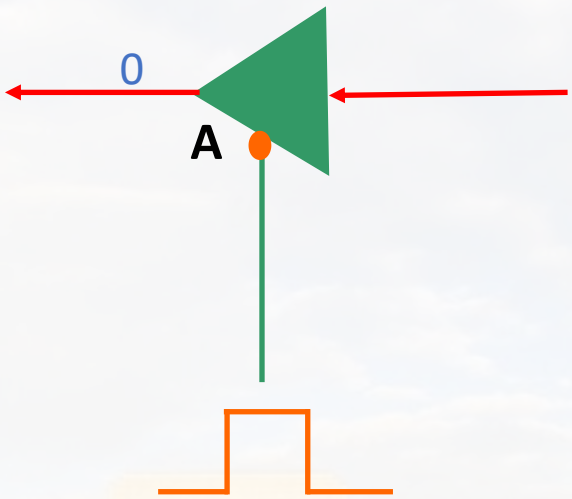
场效应管状态：当A点为低电平时，截至



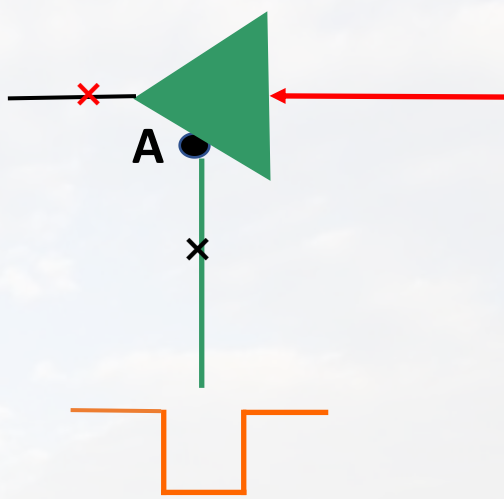
2）三态门接口：高电平（导通）、低电平（导通）、高阻态



高电平

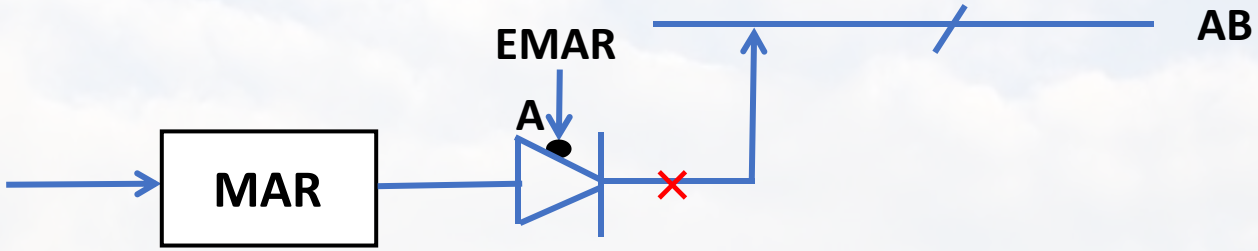


低电平

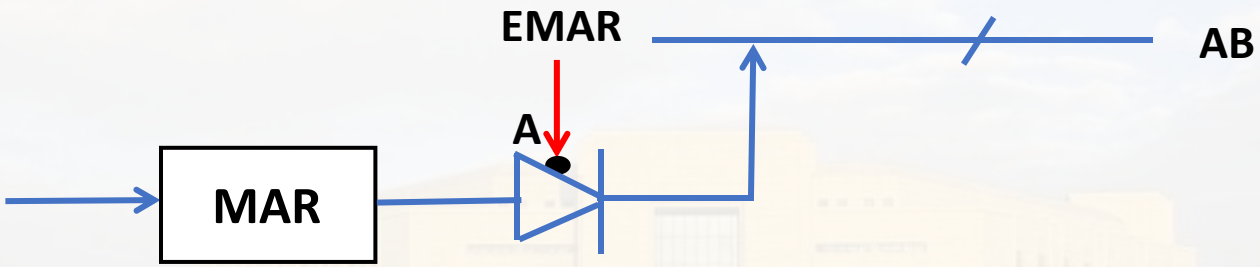


高阻态

3）当A点为低电平时，截止（输出三态门）



当A点为高电平时，导通（输出三态门）



二、CPU的组成

①地址寄存器MAR:Memory Address Register

读取指令/存取操作数/读取操作数地址时，CPU先将地址信息送入MAR，再由MAR经地址总线送往主存M，找到相应的主存单元。

②数据缓冲寄存器MBR/MDR:Memory Buffer/Data Register

由控制命令R、$\overline{W}$决定传送方向。

R：由主存单元 → 数据总线 → MDR

$\overline{W}$：由MDR → 数据总线 → 主存单元

3、总线

1）定义

数据通路结构：数据通路结构，它是CPU总体结构的核心问题。

总线：是一组能为多个部件分时共享的公共信息传送线路，及相应的控制逻辑。

2）总线类型

总线类型
- CPU内总线
- 部件间总线
- 系统总线
- 外部总线

2）总线类型

①CPU内总线：

模型机中是一组单向数据传送总线，是连接运算器、寄存器等CPU内部部件的总线。

②部件间总线：

芯片间的连接总线，包含地址线与数据线两组。

二、CPU的组成

③系统总线：

是计算机系统内各大部件进行信息交换的基础。

分类
- 地址总线AB:Address Bus
- 数据总线DB:Data Bus
- 控制总线CB:Control Bus

④外总线：

一台CS与其它设备相连接的总线。

4、时序系统

由于计算机的工作常常是分步执行的，因而就需要有一种时间信号作为分步执行的标志，如周期、节拍等。

周期、节拍、脉冲等信号称为时序信号，产生时序信号的部件称为时序发生器或时序系统，它由一个振荡器和一组计数分频器组成。

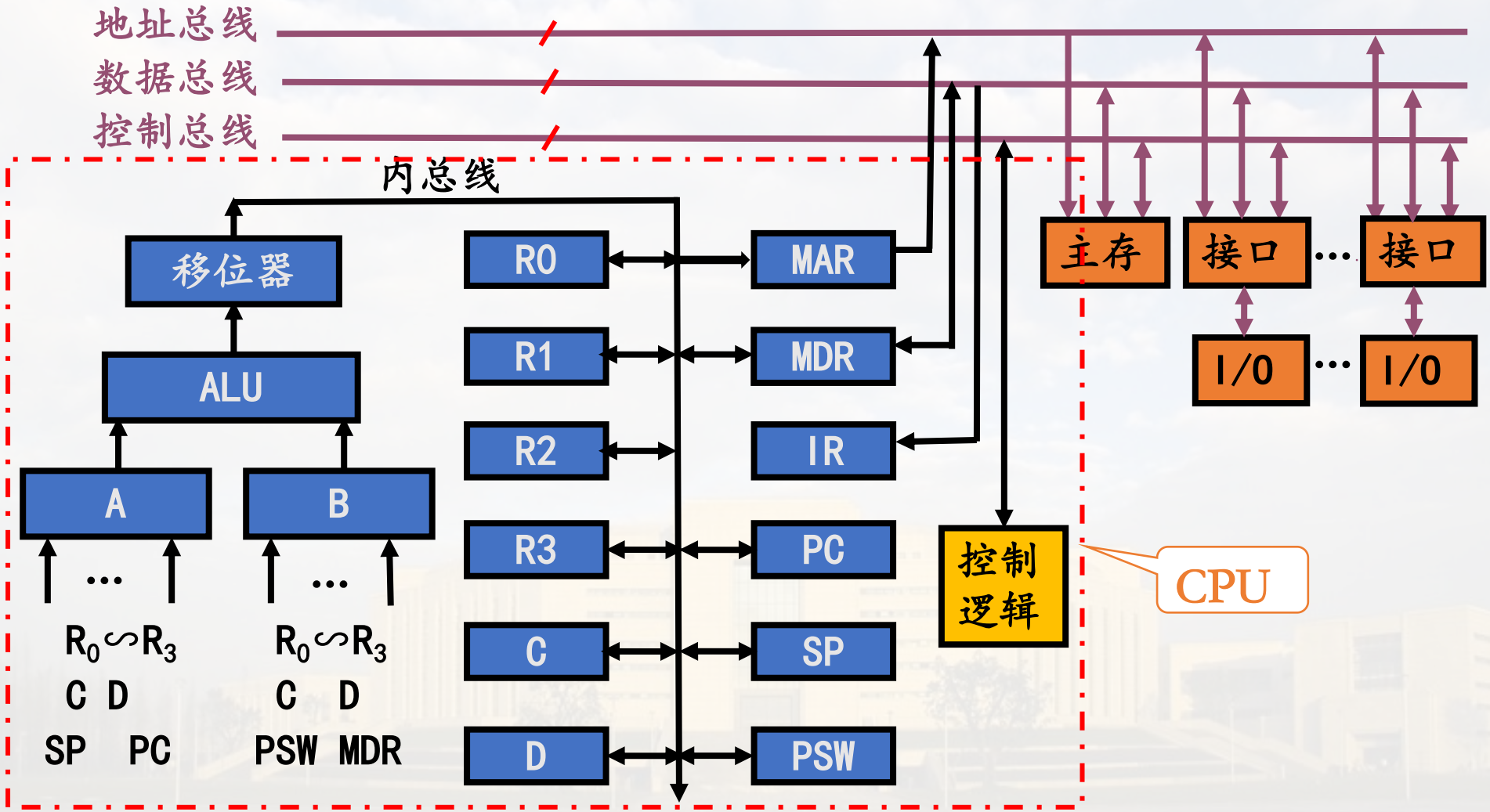
三、CPU的内部数据通路结构

模型机如何以内总线为纽带建立各部件间的数据传送通路，即CPU内部数据通路结构，这是CPU组成的核心问题。

模型机的结构为：单组、单向内总线、分立寄存器结构。

三、CPU的内部数据通路结构

1、模型机数据通路框图：



2、各类信息的传送路径

数据通路形成之后，可以在其上传送各种信息。

指令的执行基本上可以归纳为信息的传送，即控制流和数据流两大信息流。

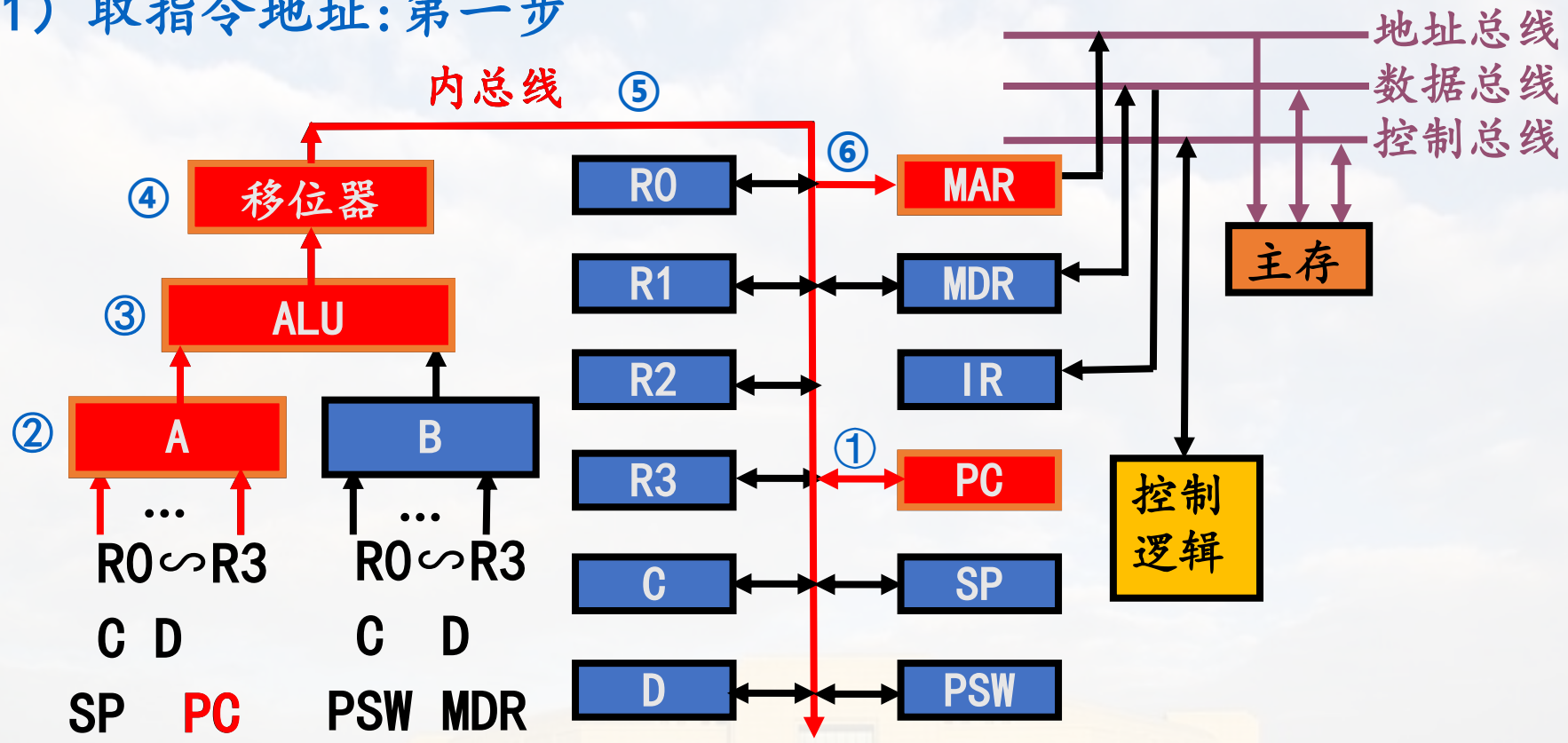
控制流：指令信息的传送，及由此产生的微命令序列。

数据流：数据信息的传送。

指令信息与数据信息的读取，依赖于地址信息。

三、CPU的内部数据通路结构

1）取指令地址：第一步

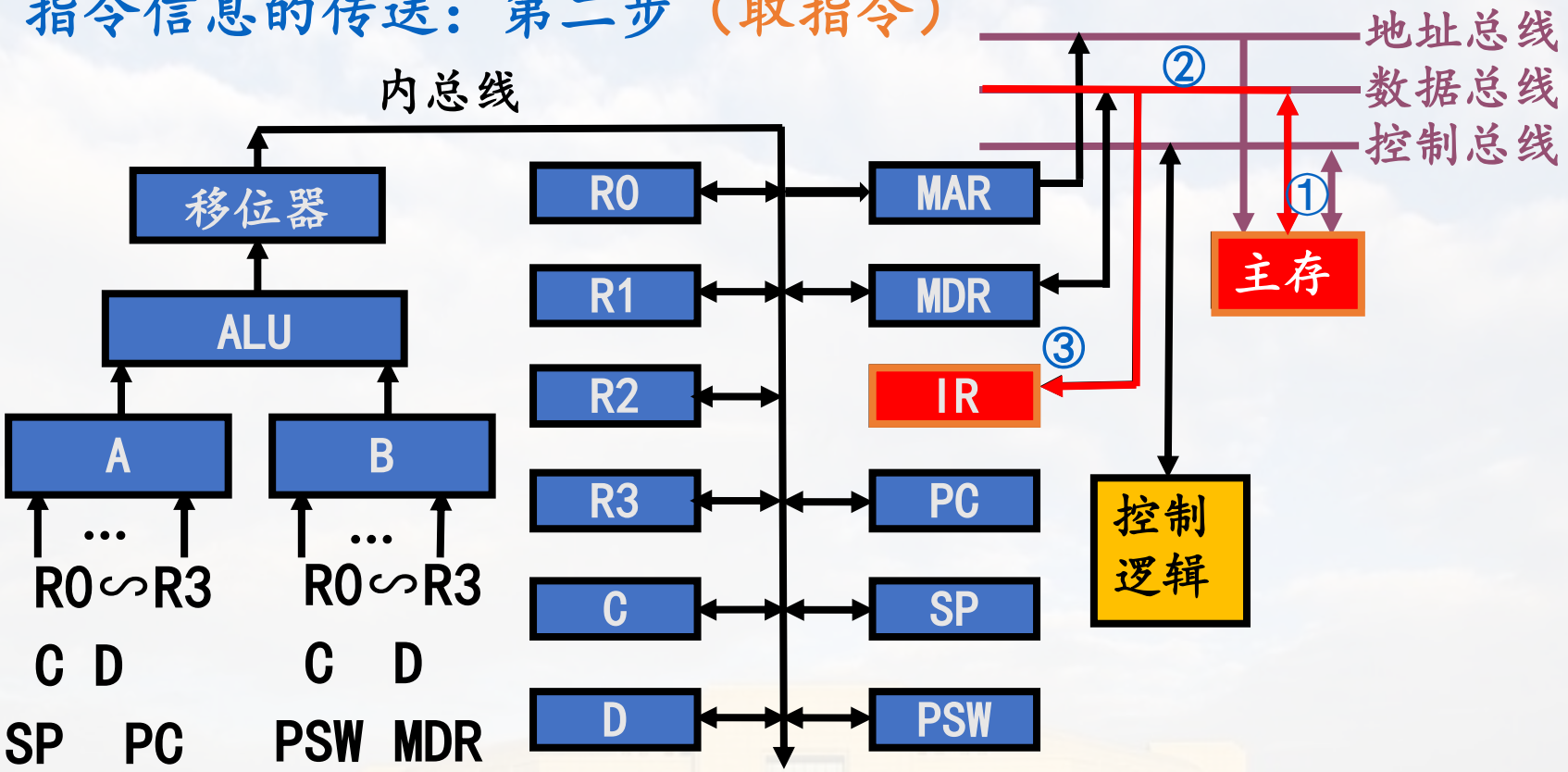


简化

PC → A → ALU → 移位器 → CPU内总线 → MAR ⇨ PC → MAR

三、CPU的内部数据通路结构

2）指令信息的传送：第二步（取指令）

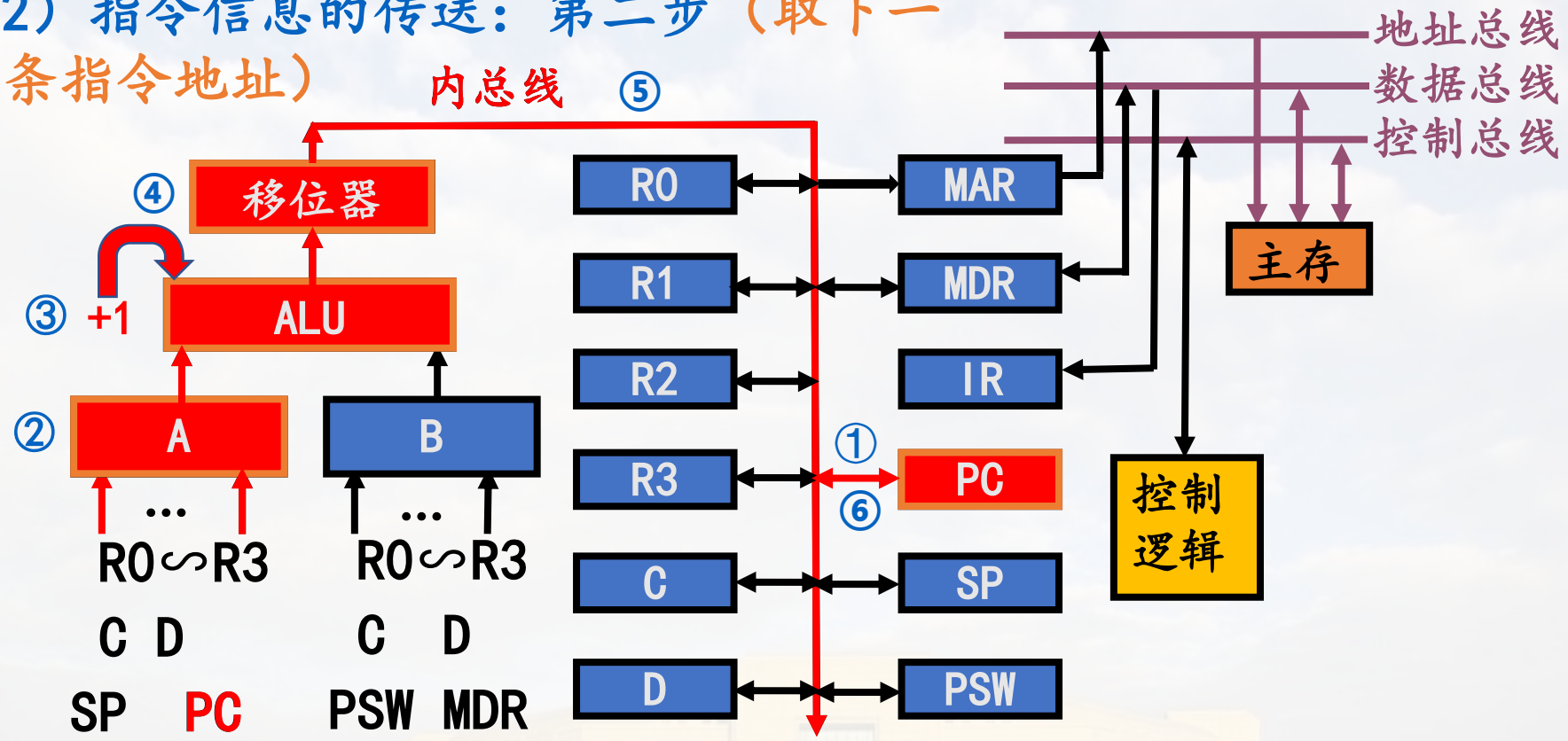


简化

M → 数总 → IR ⟹ M → IR

三、CPU的内部数据通路结构

2）指令信息的传送：第二步（取下一条指令地址）

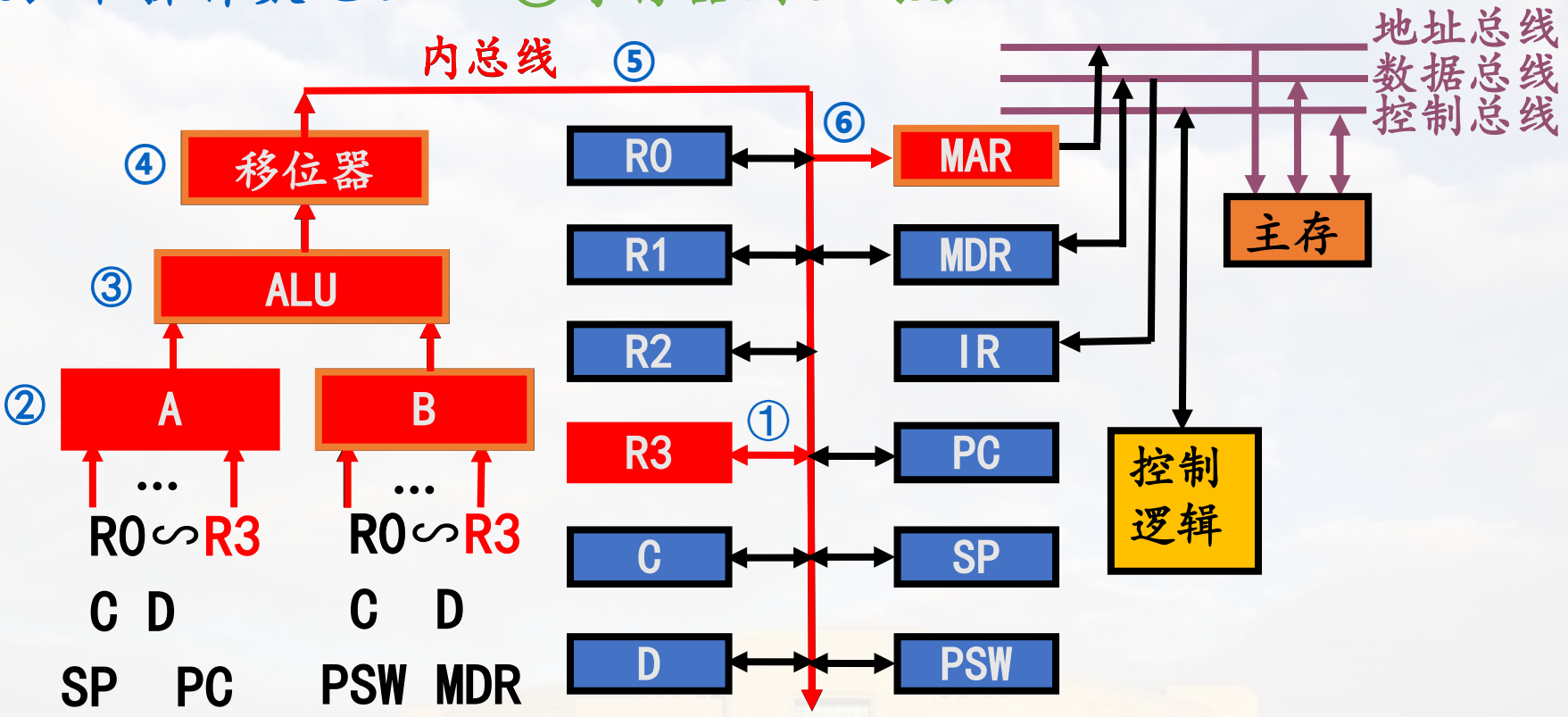


同时，后继指令地址：

PC→A→ALU(加1)→移位器→CPU内总线→PC ⇨（简化） PC+1→PC

三、CPU的内部数据通路结构

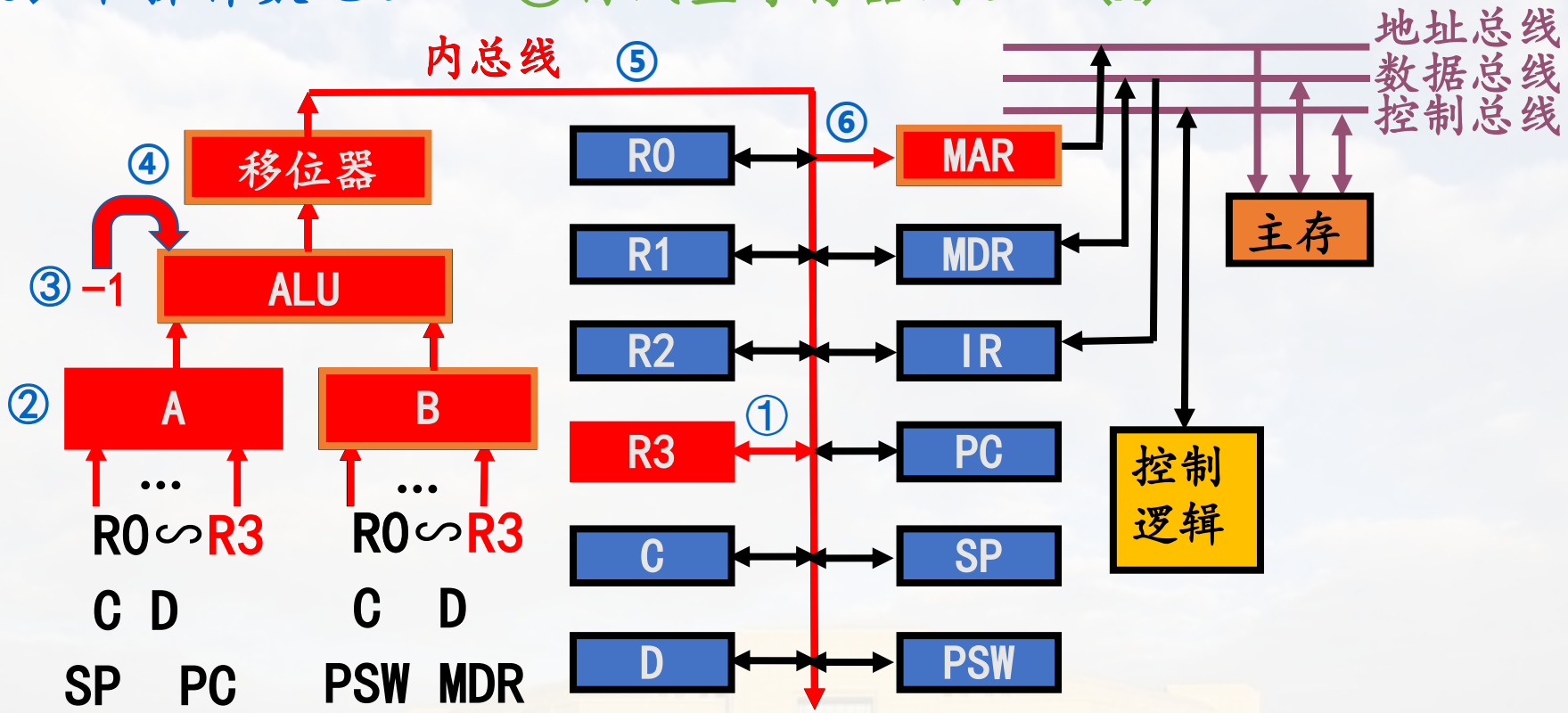
3）取操作数地址---①寄存器间址（R）



$R_i \rightarrow A/B \rightarrow ALU \rightarrow 移位器 \rightarrow CPU内总线 \rightarrow MAR \Rightarrow R_i \rightarrow MAR$

三、CPU的内部数据通路结构

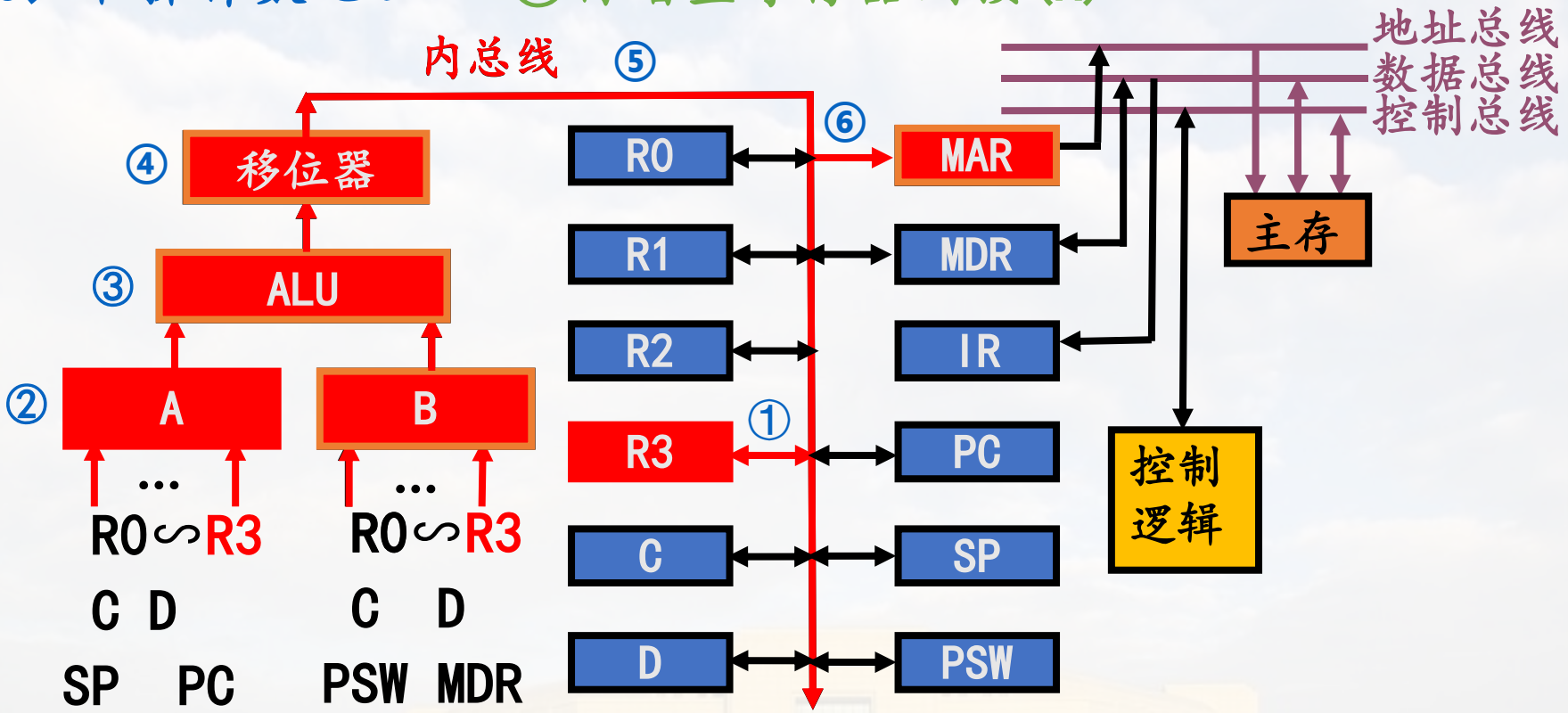
3）取操作数地址---②自减型寄存器间址 -(R)



R_i → A/B → ALU(减1) → 移位器 → CPU内总线 → MAR ⇨ R_i-1 → MAR

三、CPU的内部数据通路结构

3）取操作数地址---④自增型寄存器间接(R)+

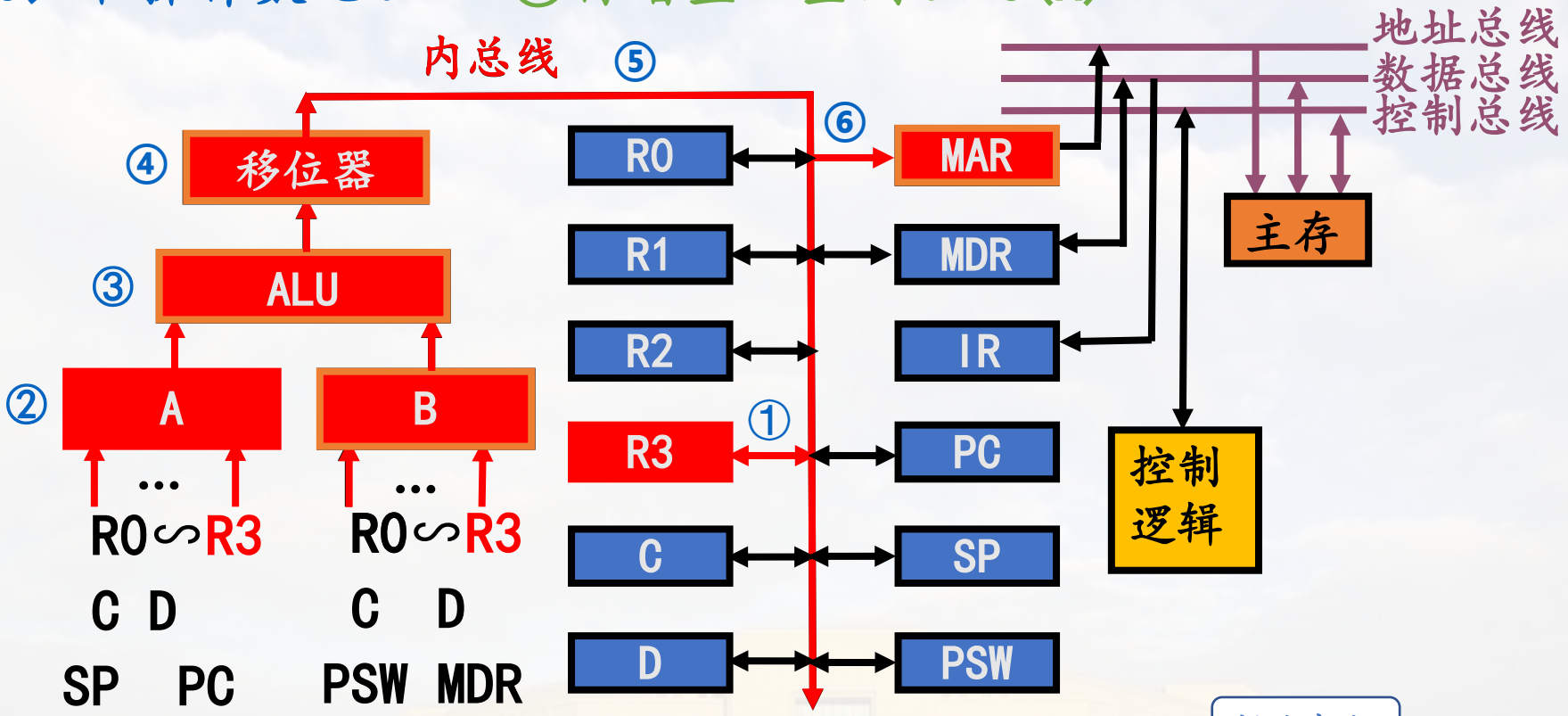


与寄存器间接寻址（R)一样

$R_i \rightarrow A/B \rightarrow ALU \rightarrow$ 移位器 $\rightarrow$ CPU内总线 $\rightarrow MAR \Rightarrow R_i \rightarrow MAR$

三、CPU的内部数据通路结构

3）取操作数地址---④自增型双重间址 @(R)+



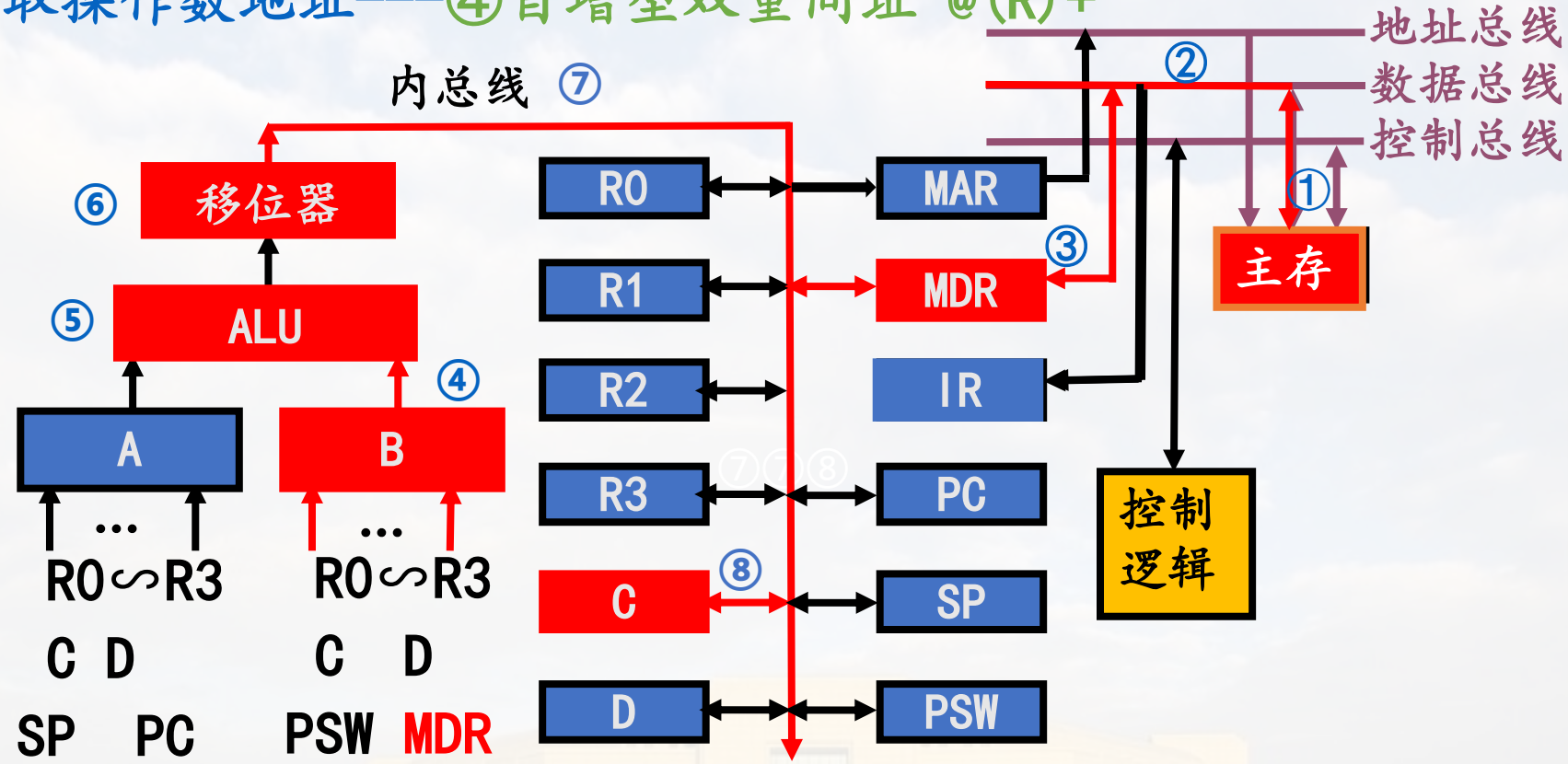
第一步（共三步）：送间址地址

间址地址

$R_i \rightarrow A/B \rightarrow ALU \rightarrow$ 移位器 $\rightarrow$ CPU内总线 $\rightarrow MAR \Rightarrow R_i \rightarrow MAR$

三、CPU的内部数据通路结构

3）取操作数地址---④自增型双重间址 @(R)+



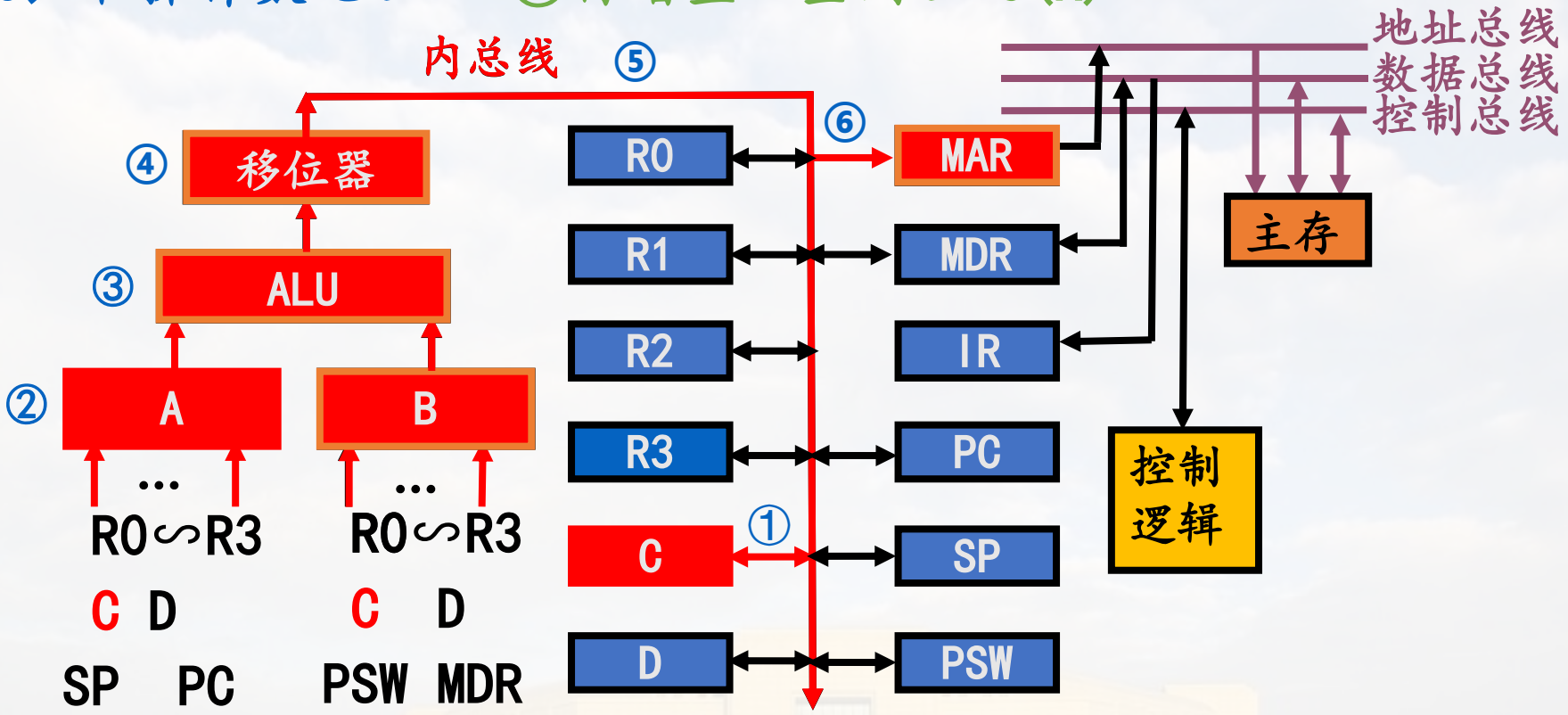
第二步（共三步）：取操作数地址

M → 数总 → MDR → B → ALU → 移位器 → 内总线 → C

⇨ M → MDR → C

操作数地址

3）取操作数地址---④自增型双重间址 @(R)+



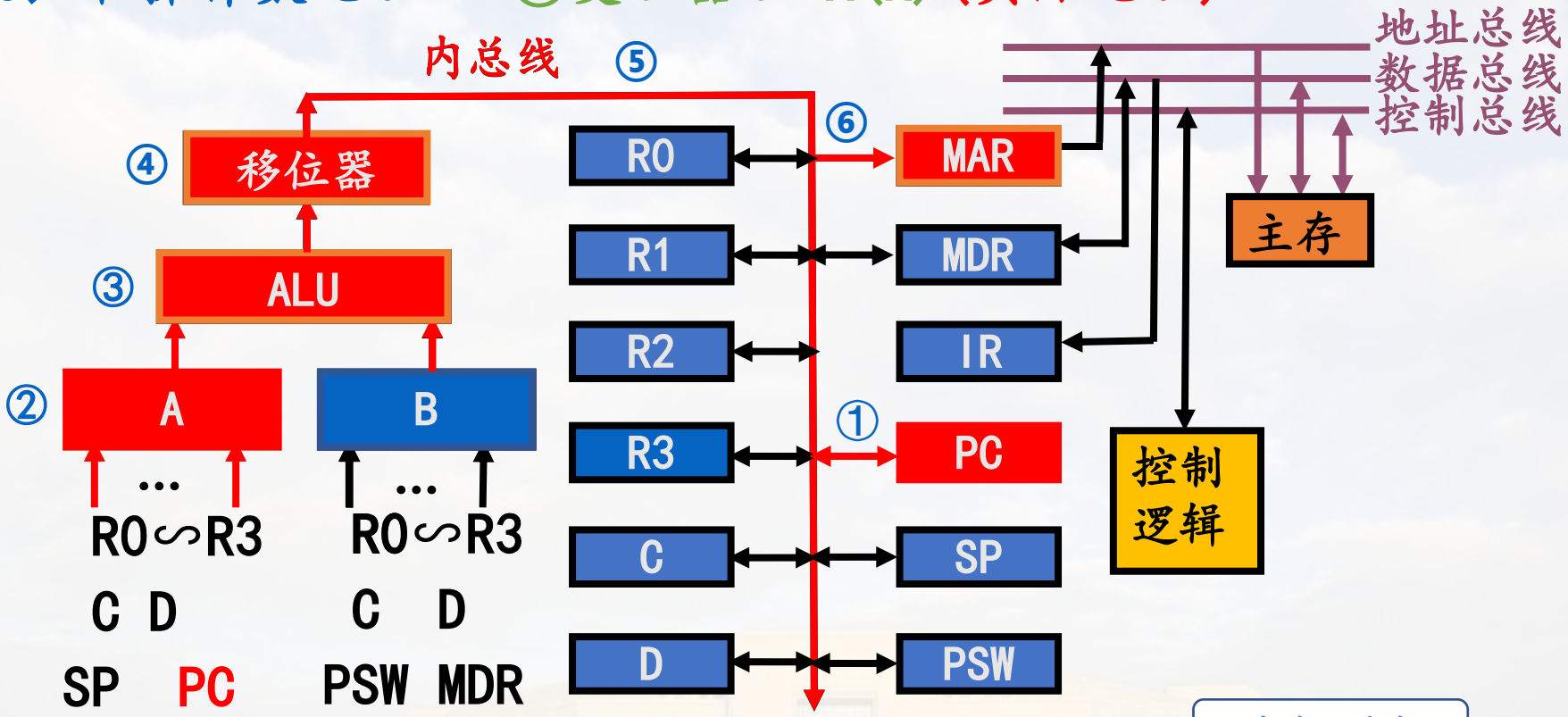
第三步（共三步）：送操作数地址

操作数地址

C → A/B → ALU → 移位器 → CPU内总线 → MAR ⇨ C → MAR

三、CPU的内部数据通路结构

3）取操作数地址---⑤变址器址 X(R)（实际地址）



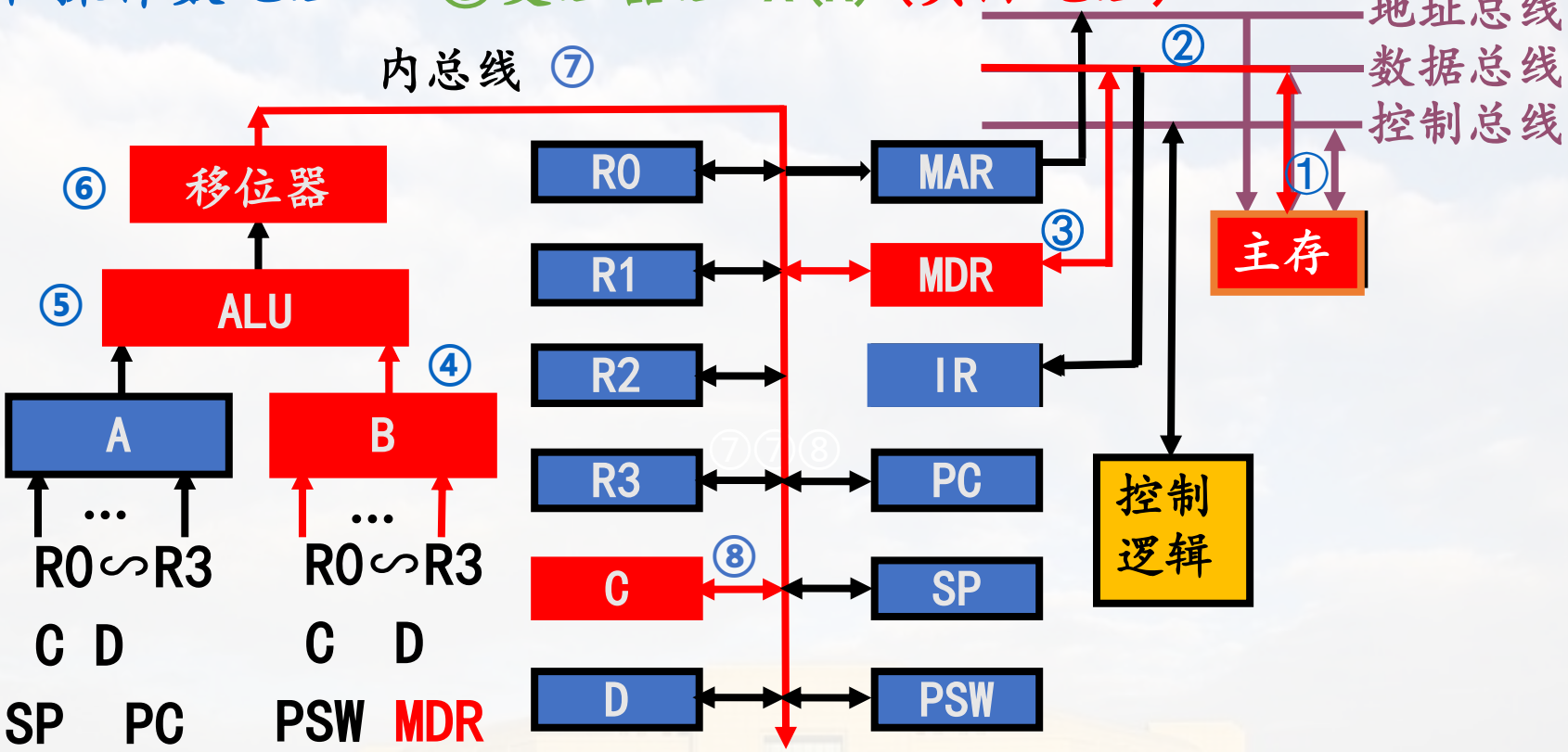
第一步（共三步）：送形式地址的地址

形式地址的地址

PC → A → ALU → 移位器 → CPU内总线 → MAR ⇨ PC → MAR

三、CPU的内部数据通路结构

3）取操作数地址---⑤变址器址 X(R)（实际地址）

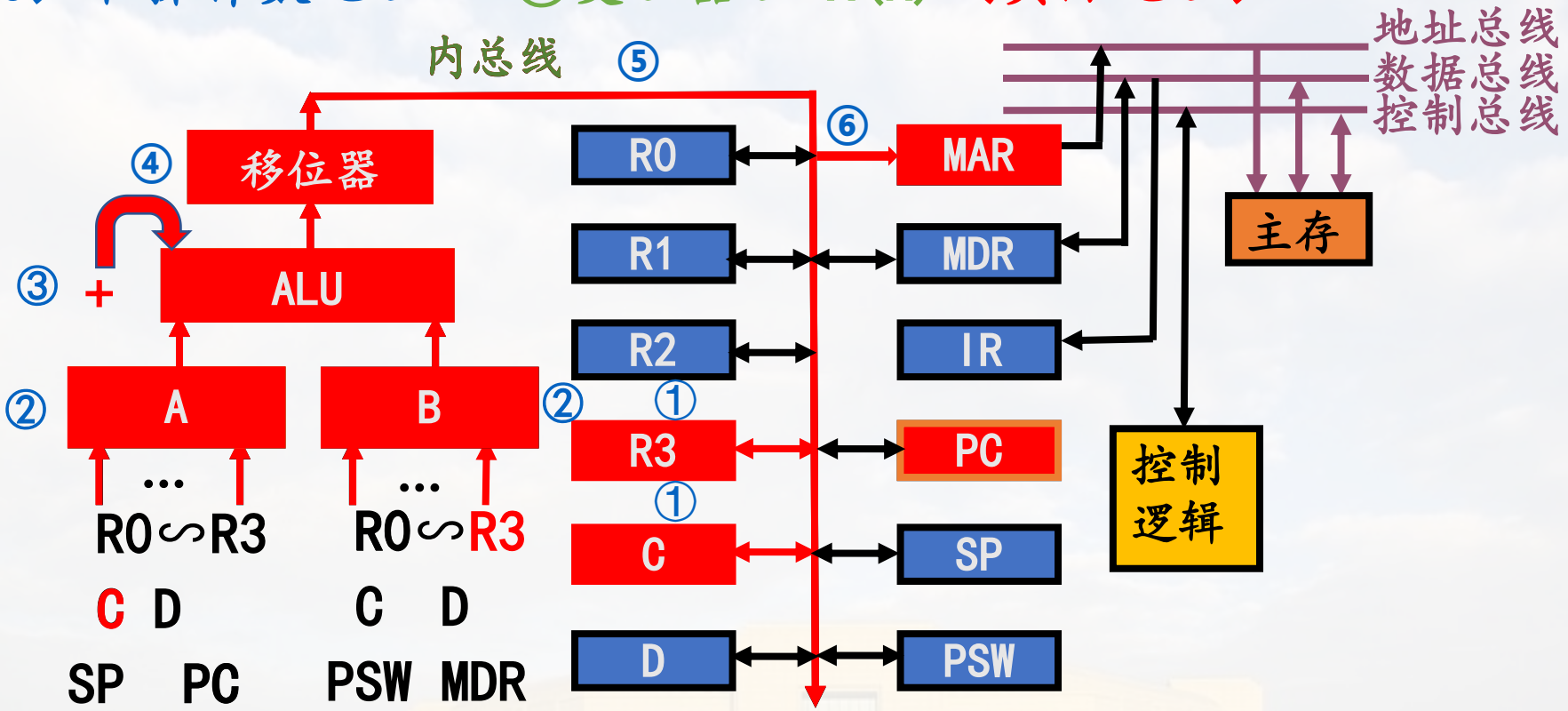


第二步（共三步）：取形式地址

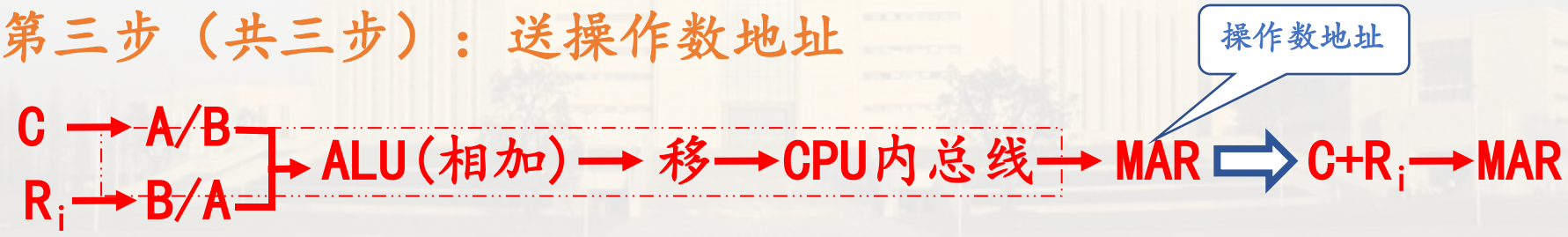
M → 数总 → MDR → B → ALU → 移位器 → 内总线 → C

⇨ M → MDR → C　形式地址

3）取操作数地址---⑤变址器址 X(R)（实际地址）

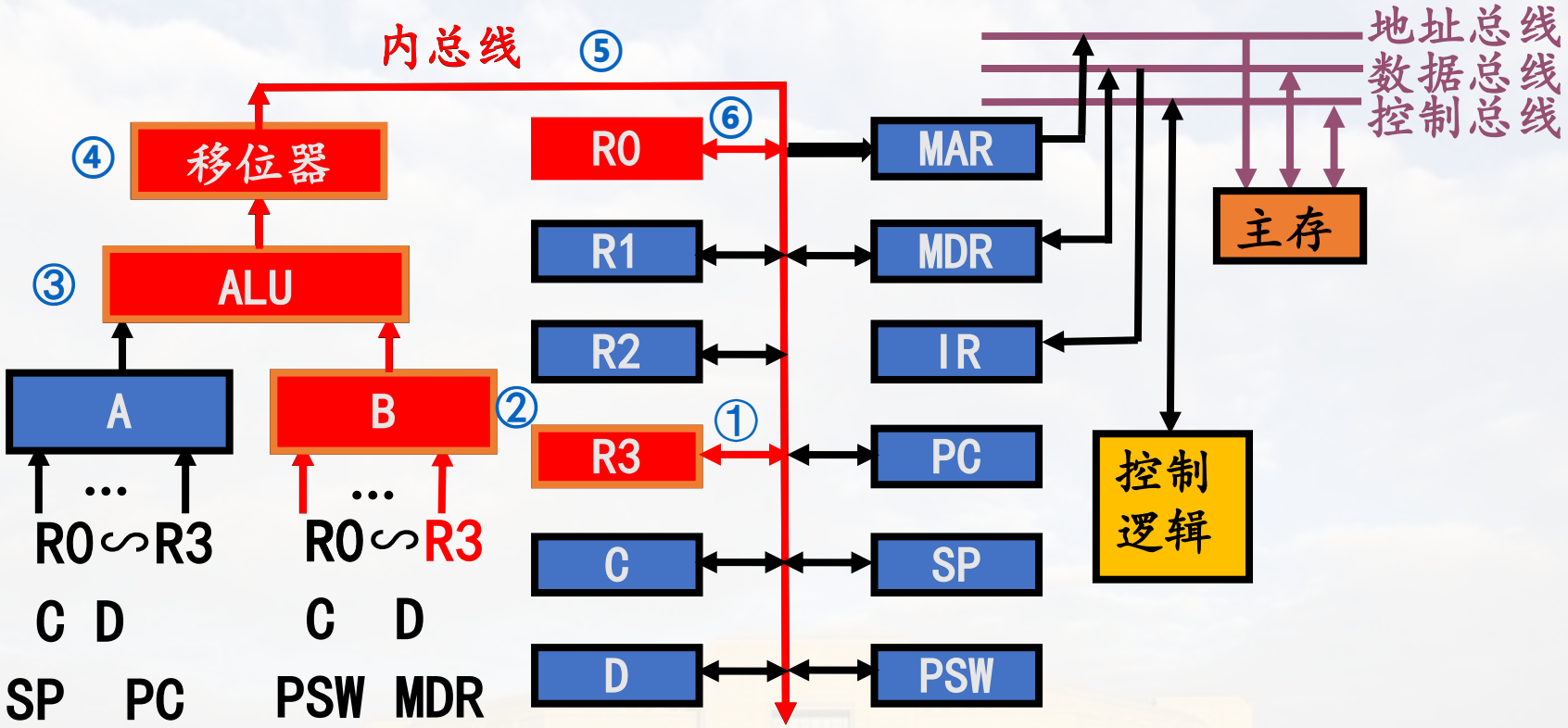


第三步（共三步）：送操作数地址



三、CPU的内部数据通路结构

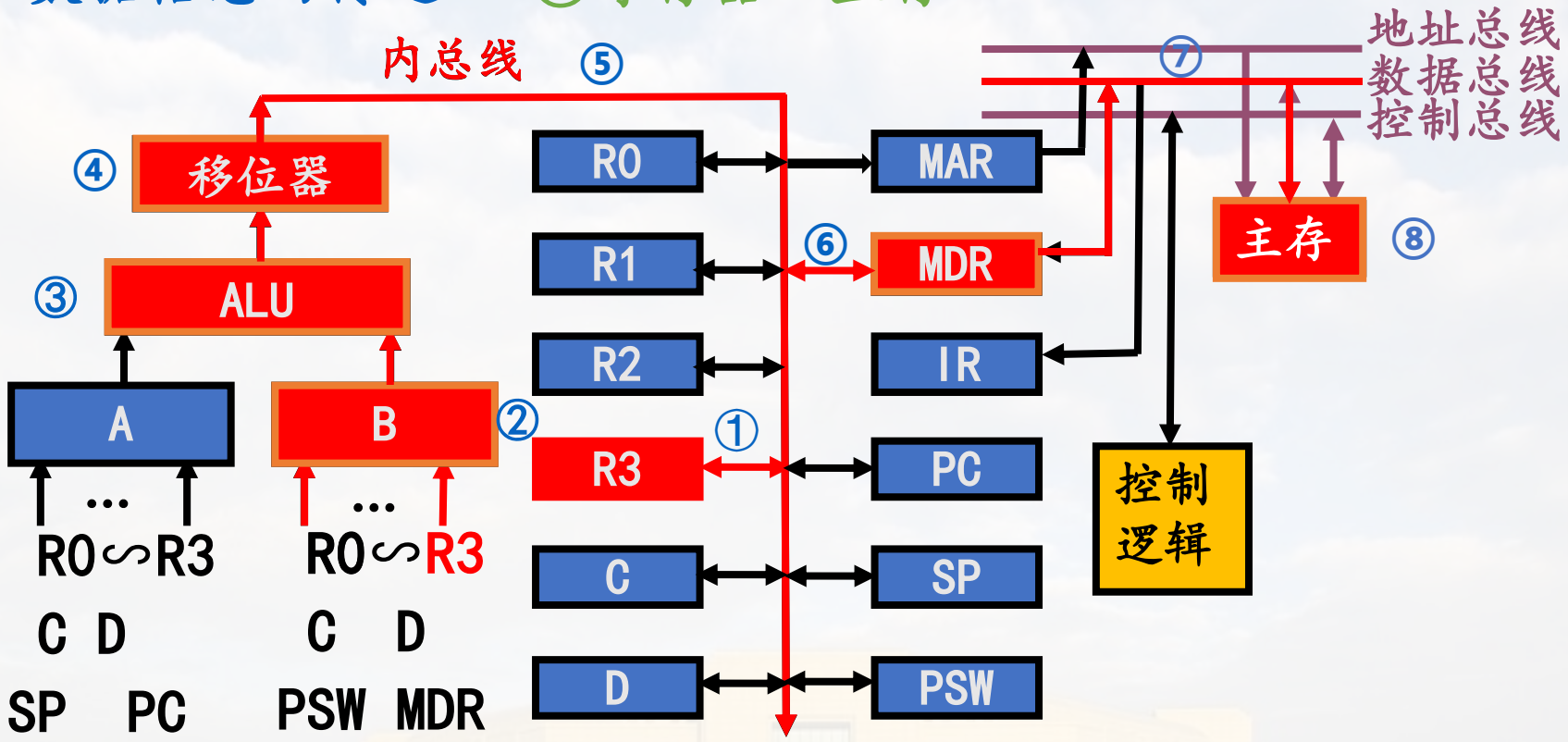
4）数据信息的传送---①寄存器→寄存器(CPU内的R)



R_i → A/B → ALU → 移位器 → CPU内总线 → R_j ⇨ $R_i \rightarrow R_j$

三、CPU的内部数据通路结构

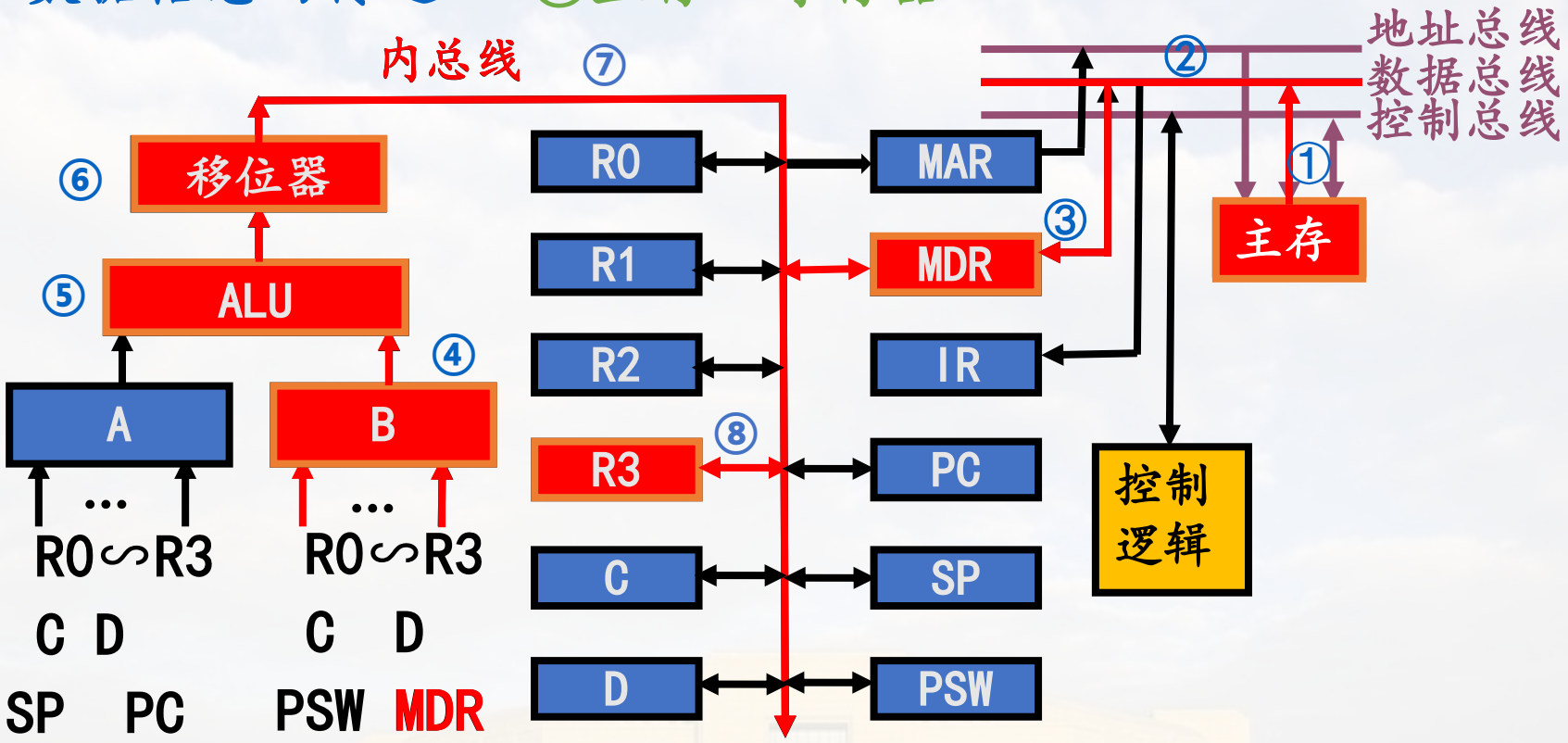
4）数据信息的传送---②寄存器→主存



R_i → A/B → ALU → 移位器 → CPU内总线 → MDR → 数总 → M

⇨ R_i → MDR，MDR → M

4）数据信息的传送----③主存→寄存器

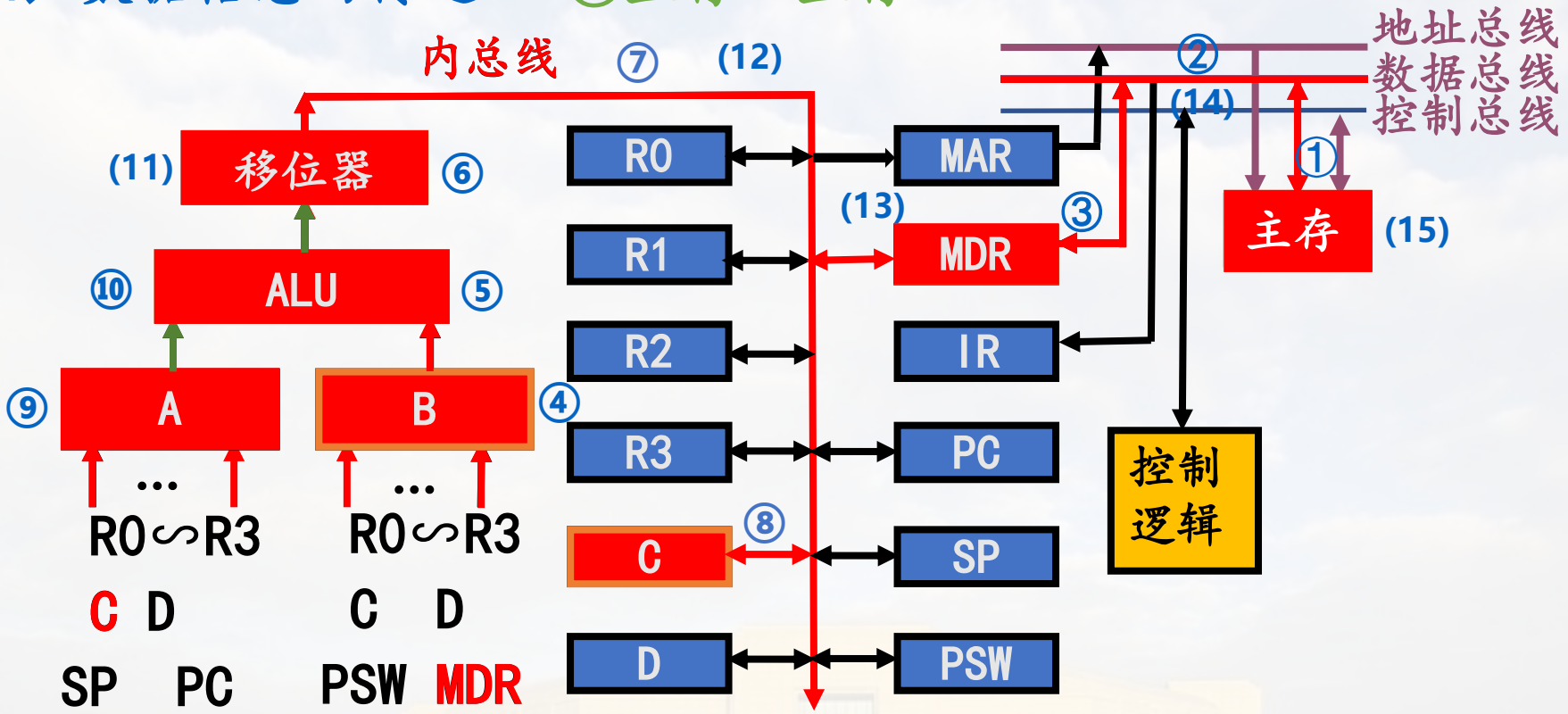


M →数总→ MDR → B → ALU → 移位器 → CPU内总线 → R_i

⇨ M → MDR → R_i

三、CPU的内部数据通路结构

4）数据信息的传送----④主存→主存

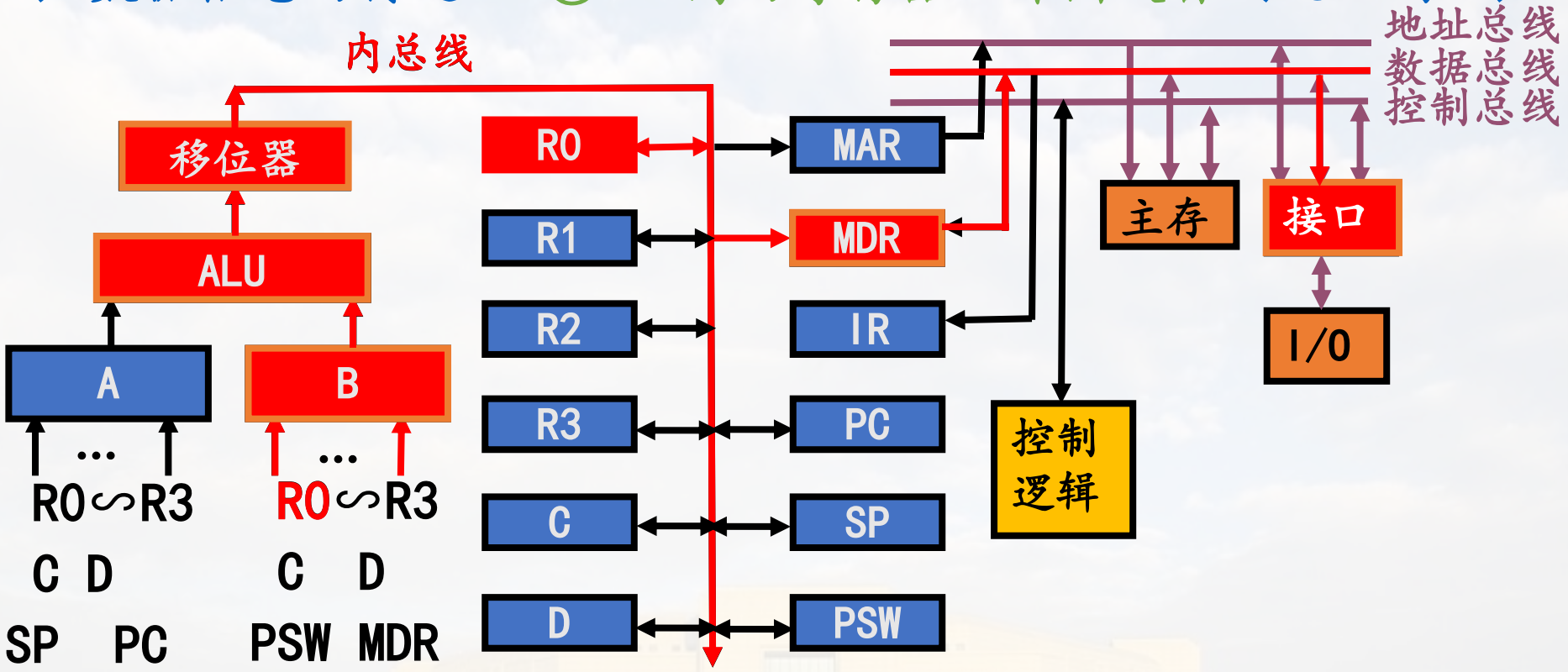


M→数总→MDR→B→ALU→移→CPU内总线→C→A/B→ALU→移

→CPU内总线→MDR→数总→M ⇨ M_1→MDR→C，C→MDR，MDR→M_2

三、CPU的内部数据通路结构

4）数据信息的传送----⑤CPU内的寄存器→外围设备（统一编址）

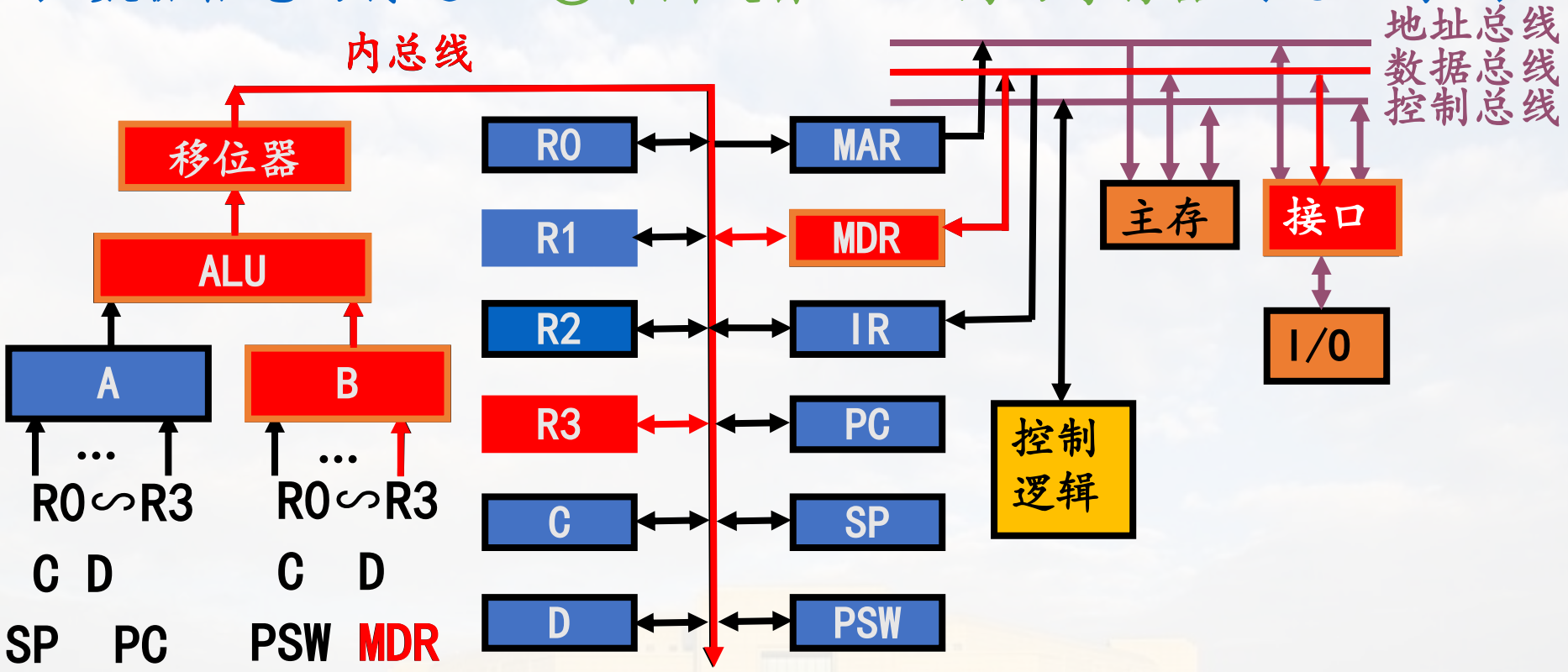


R_i → A/B → ALU → 移位器 → CPU内总线 → MDR → 数总 → R_j

（与②相同），自己完成简化路径

三、CPU的内部数据通路结构

4）数据信息的传送---⑥外围设备→CPU内的寄存器（统一编址）



R_j → 数总 → MDR → B → ALU → 移位器 → CPU内总线 → R_i

（与③相同），自己完成简化路径

（3）微命令设置

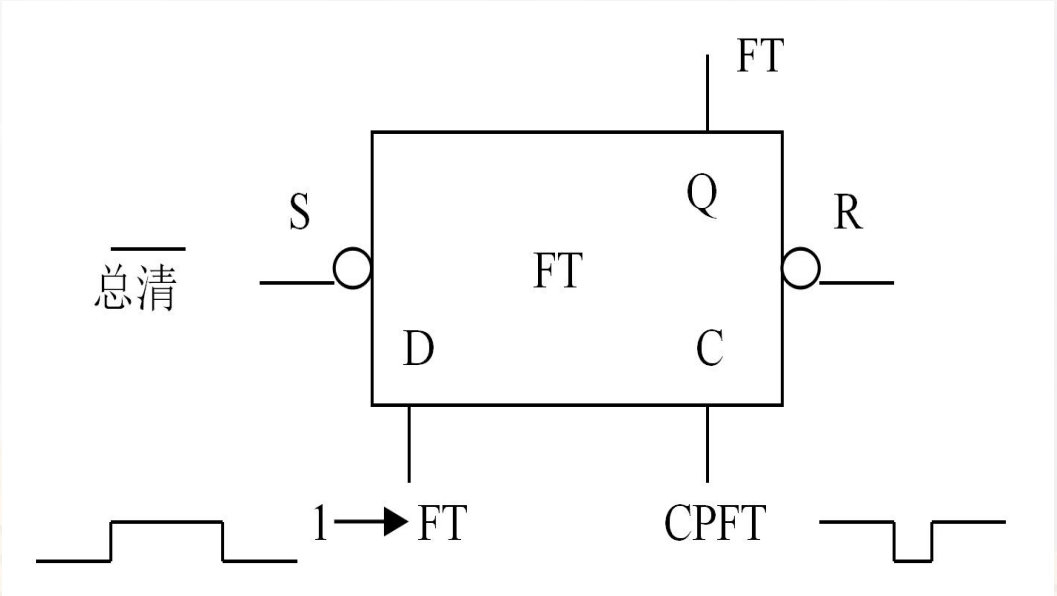
将模型机的数据传送路径分为两大类操作，并分段设置相关微命令：

首先回顾：模型机中寄存器的基本组成单元是集成D触发器。

集成D触发器工作原理：

1）同步打入(由D端打入)

2）由R/S端异步置入



3、微命令设置

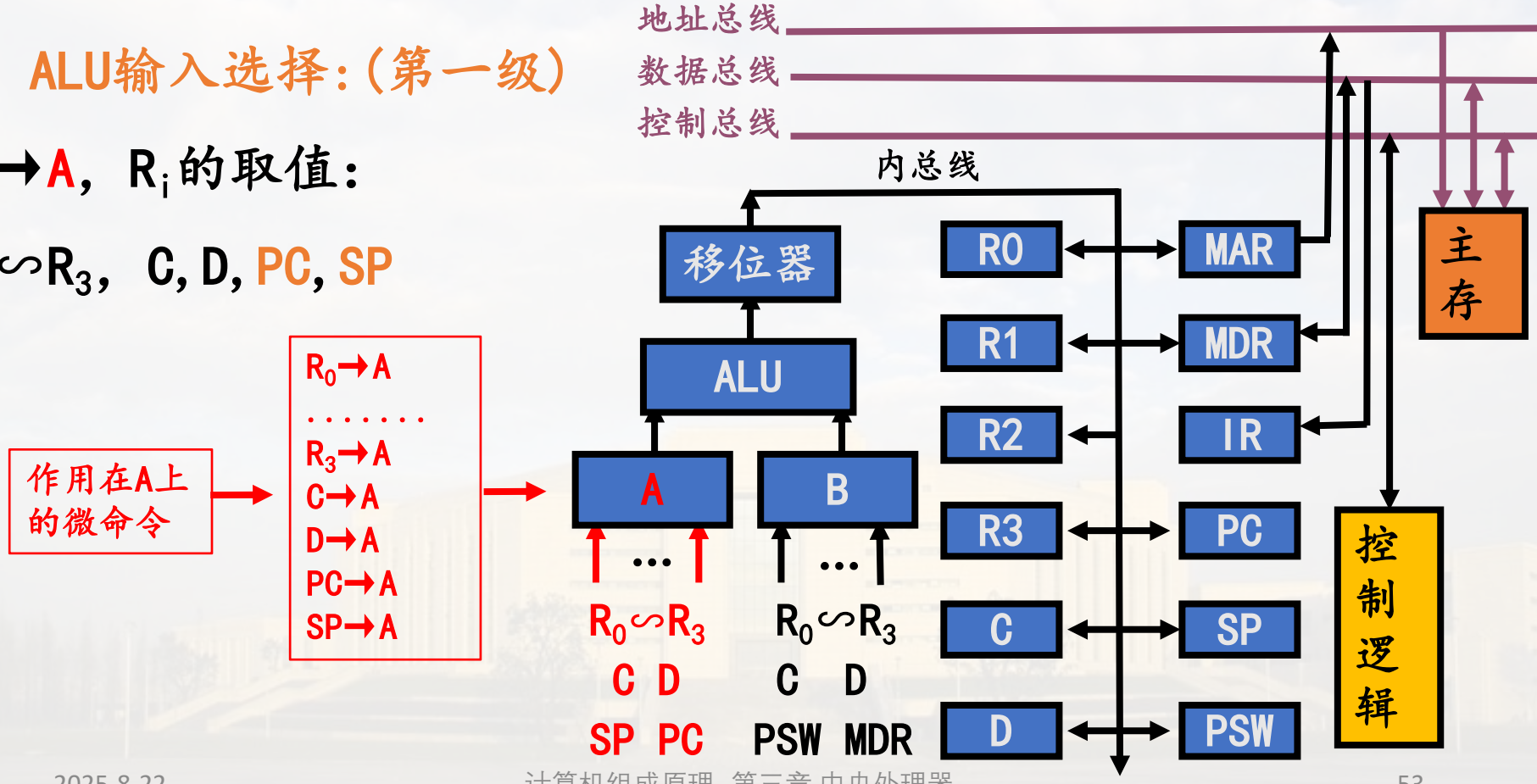
将模型机的数据传送路径分为两大类操作，并分段设置相关微命令：

1. 数据通路操作：CPU内部操作

1）ALU输入选择：（第一级）

$R_i \rightarrow A$，R_i的取值：

$R_0 \backsim R_3$，C, D, PC, SP



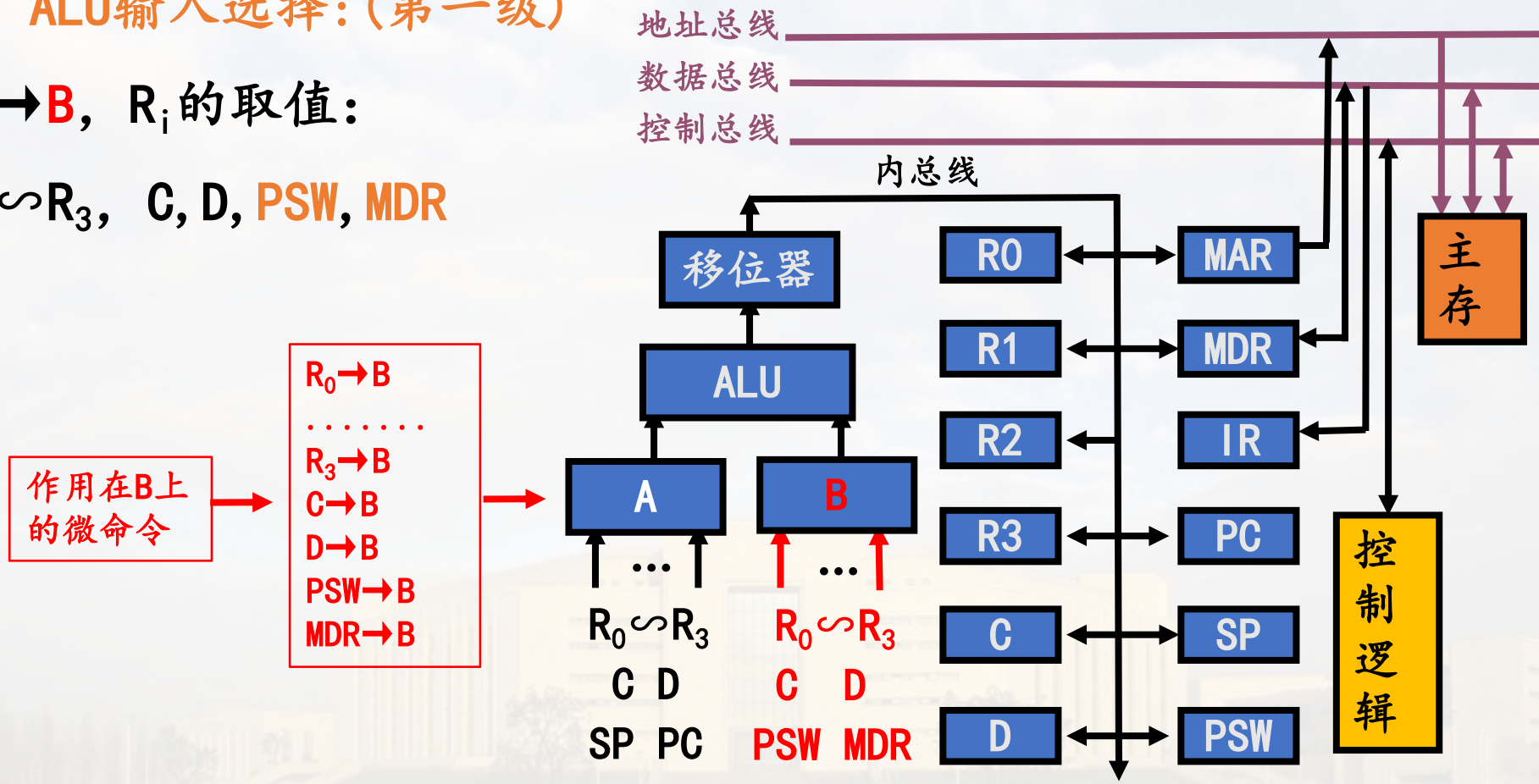
三、CPU的内部数据通路结构

（3）微命令设置

1）ALU输入选择:(第一级)

$R_i \rightarrow B$，R_i的取值：

$R_0 \backsim R_3$，C, D, PSW, MDR



（3）微命令设置

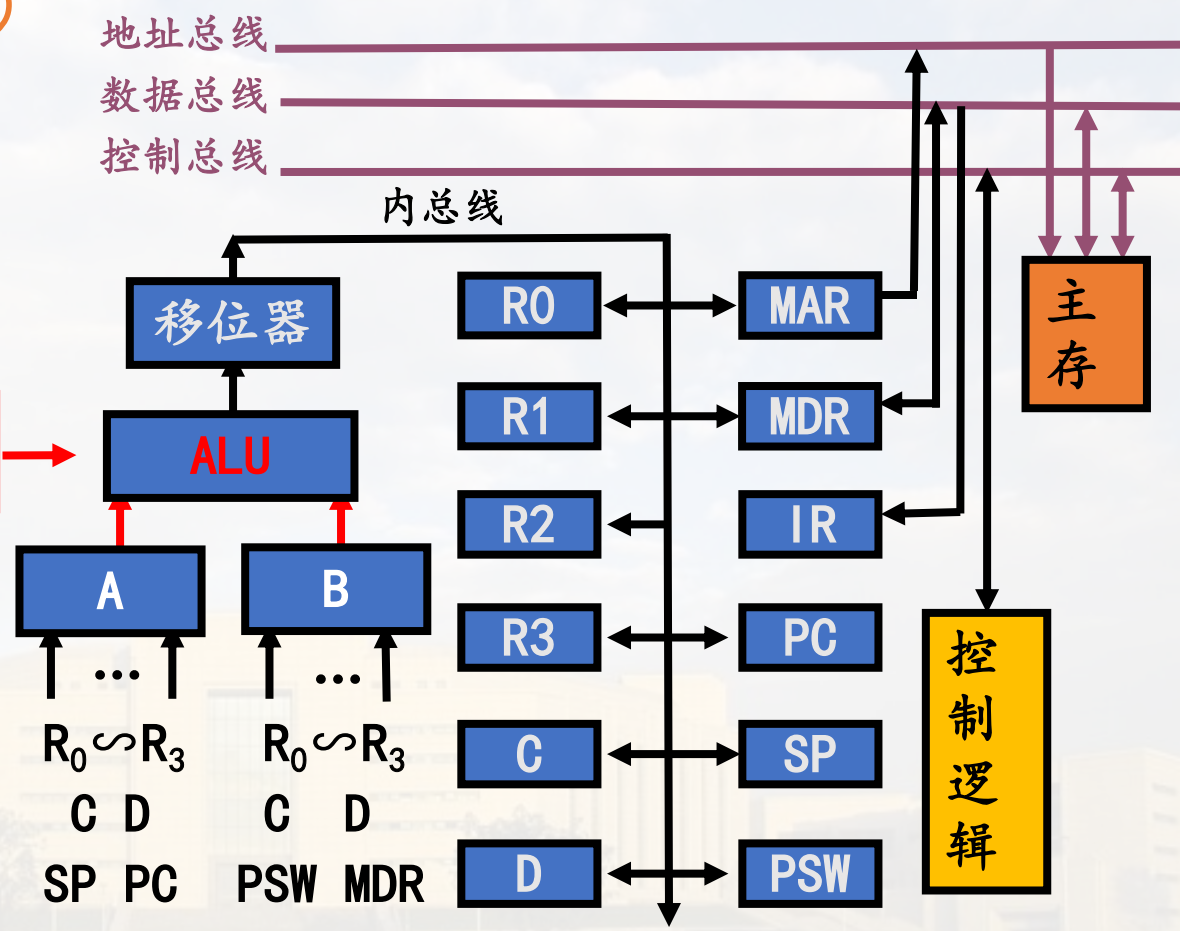
2）ALU功能选择:(第二级)

微命令$S_0 \backsim S_3$，C_0，M，

本章算术逻辑运算部件

中会讲解

作用在ALU上的微命令 → $S_0 \backsim S_3$，C_0，M的组合



三、CPU的内部数据通路结构

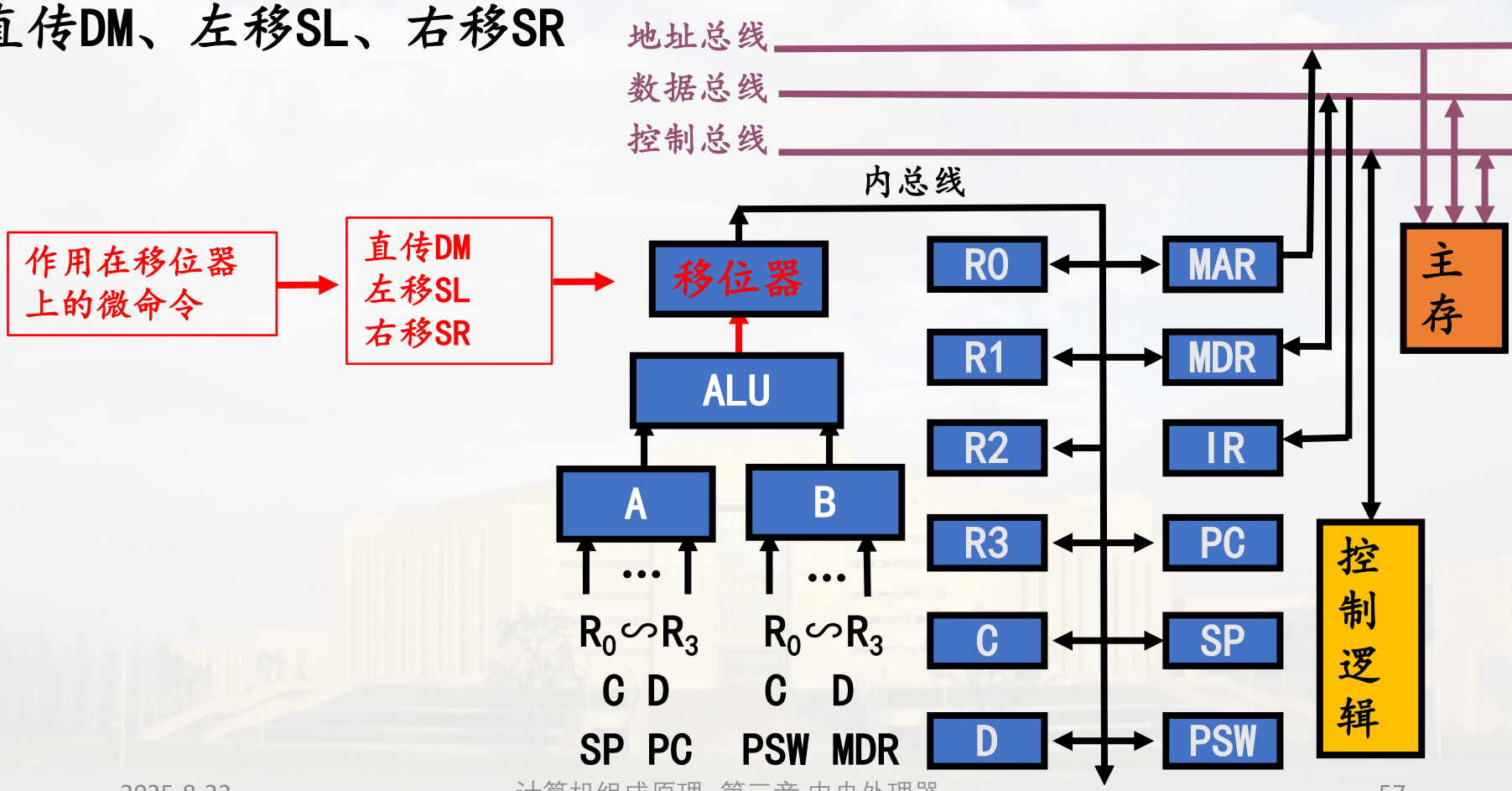
SN74181功能表（常用）

操作类型 \ 微命令	M	S_3	S_2	S_1	S_0	C_0	功能
算术运算	0	1	1	1	1	1	A+1
	0	0	0	0	0	0	A-1
	0	1	0	0	1	0	A+B
	0	0	1	1	0	1	A-B
逻辑运算	1	1	1	1	1	0	传送A
	1	0	1	1	1	0	传送B

(3) 微命令设置

3) 移位器功能选择:(第三级)

直传DM、左移SL、右移SR



（3）微命令设置

4）分配脉冲：(第四级)

$CPR_0 \backsim CPR_3$，CPC, CPD, CPMDR,

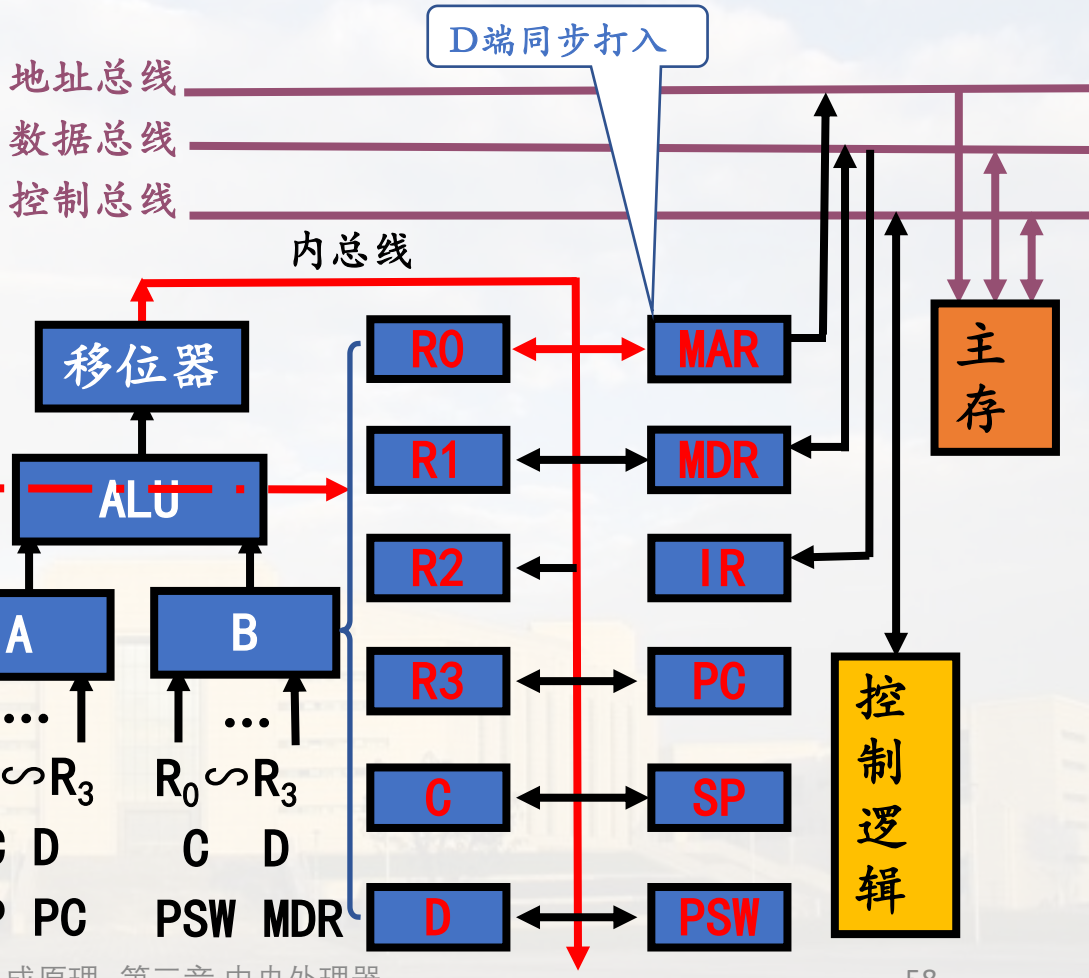
CPMAR, CPPSW, CPPC, CPSP

（IR除外）

作用在寄存器上的微命令

IR除外

CPR_0
CPR_1
CPR_2
CPR_3
CPC
CPD
CPMDR
CPMAR
CPPSW
CPPC
CPSP



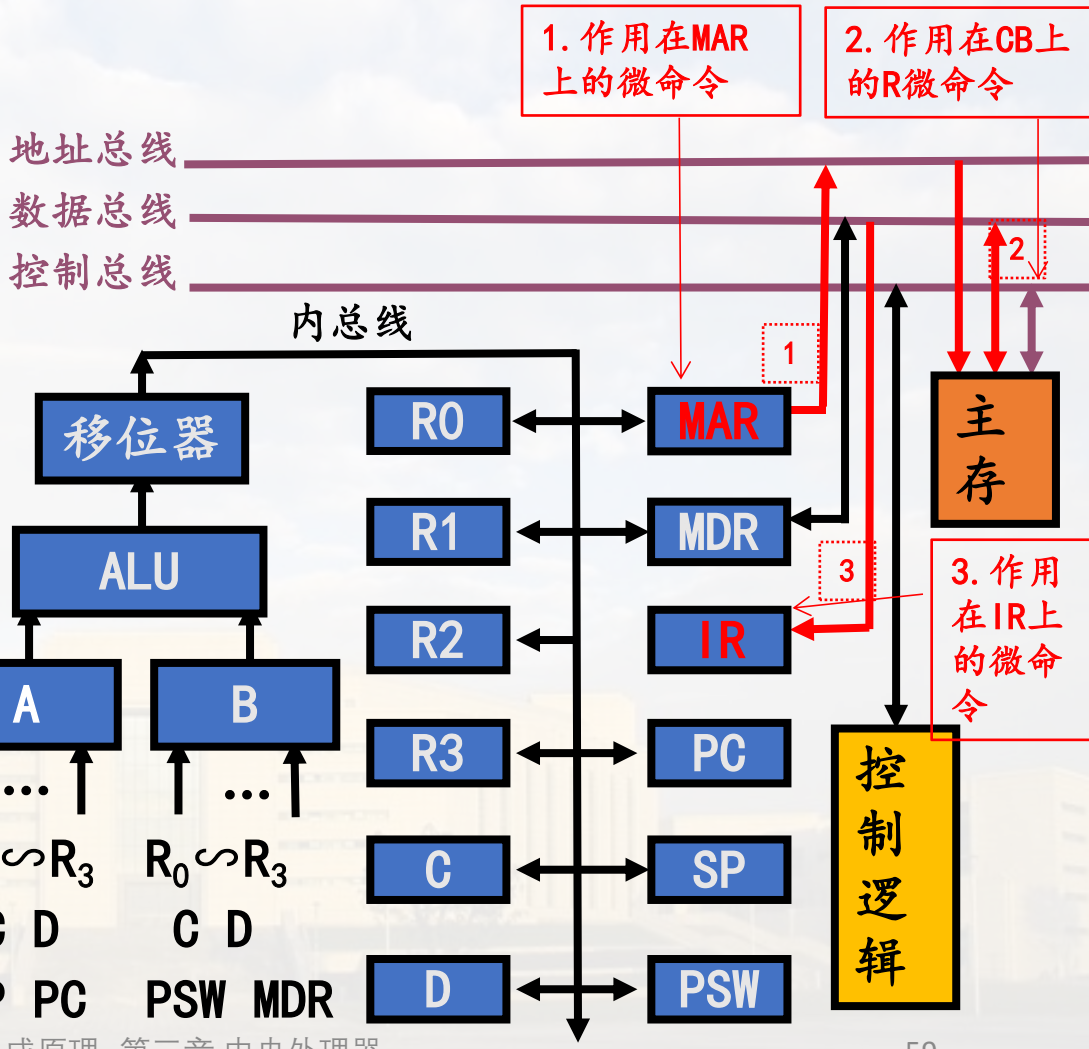
2. 与系统线及主存有关的微命令

EMAR, R, W, SIR, SMDR

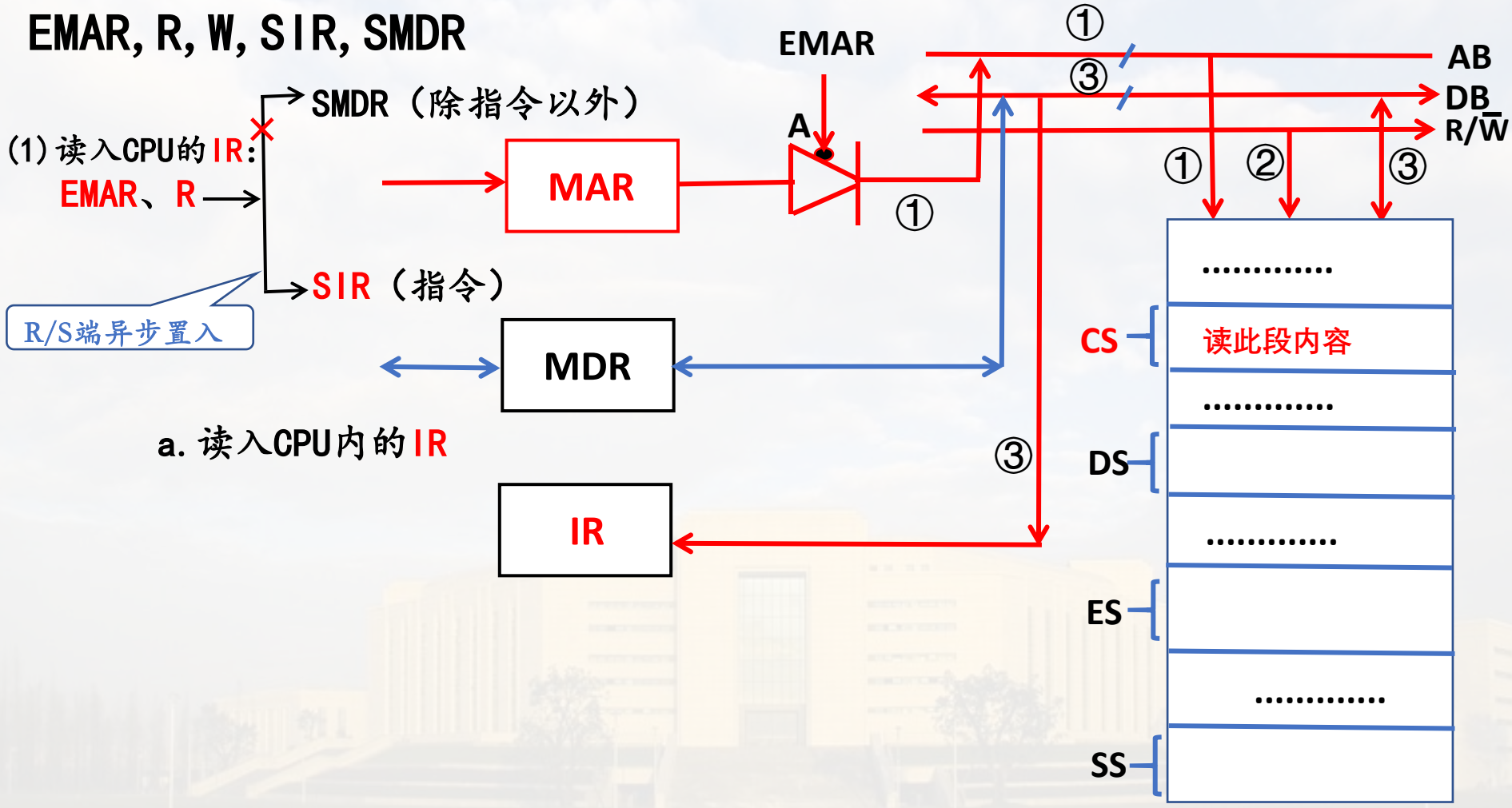
(1) 读入CPU的IR:
EMAR、R →
→ SMDR（除指令以外）
→ SIR（指令）

R/S端异步置入

a. 读入CPU内的IR



2. 与系统线及主存有关的微命令



2. 与系统线及主存有关的微命令

EMAR, R, W, SIR, SMDR

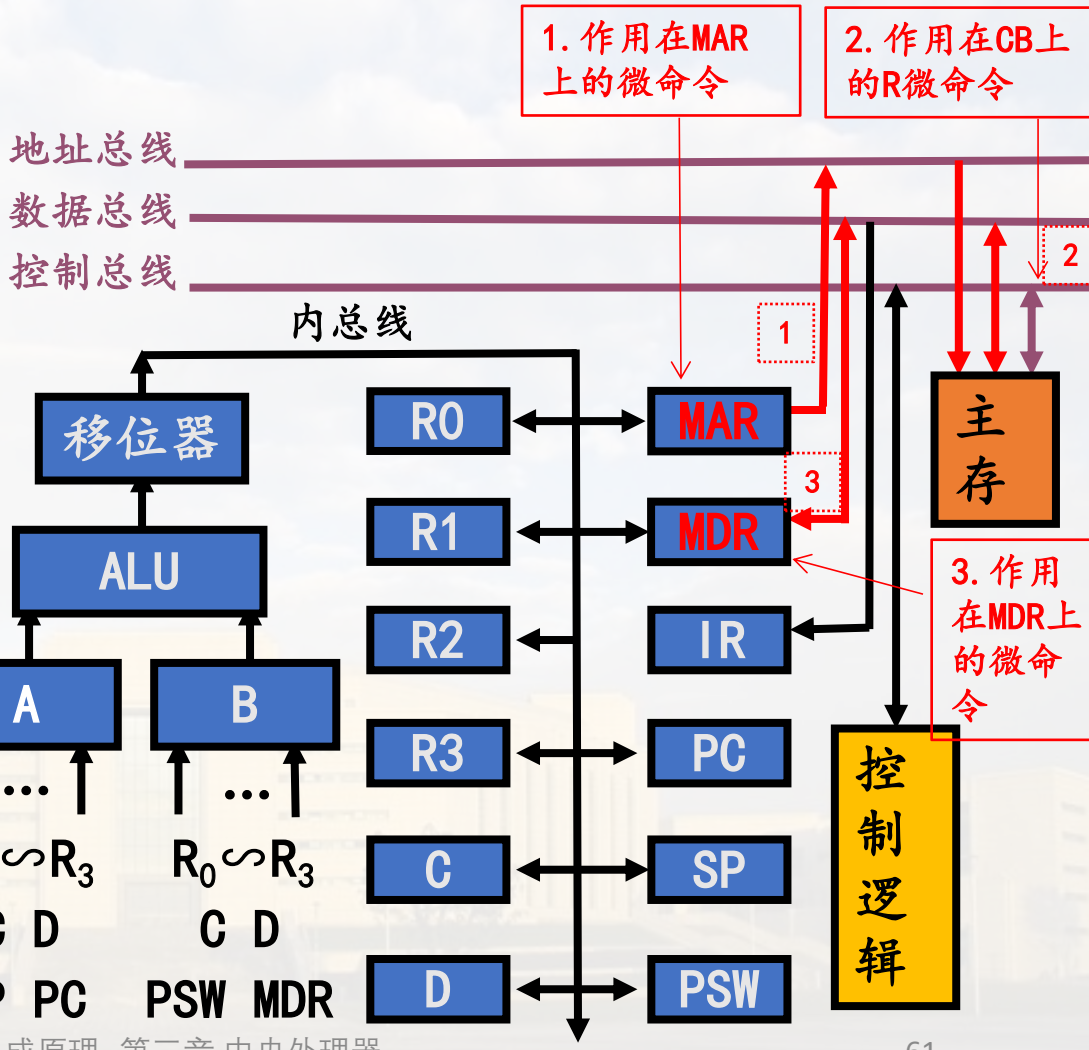
(1) 读入CPU的MDR:

EMAR、R → SMDR（除指令以外）

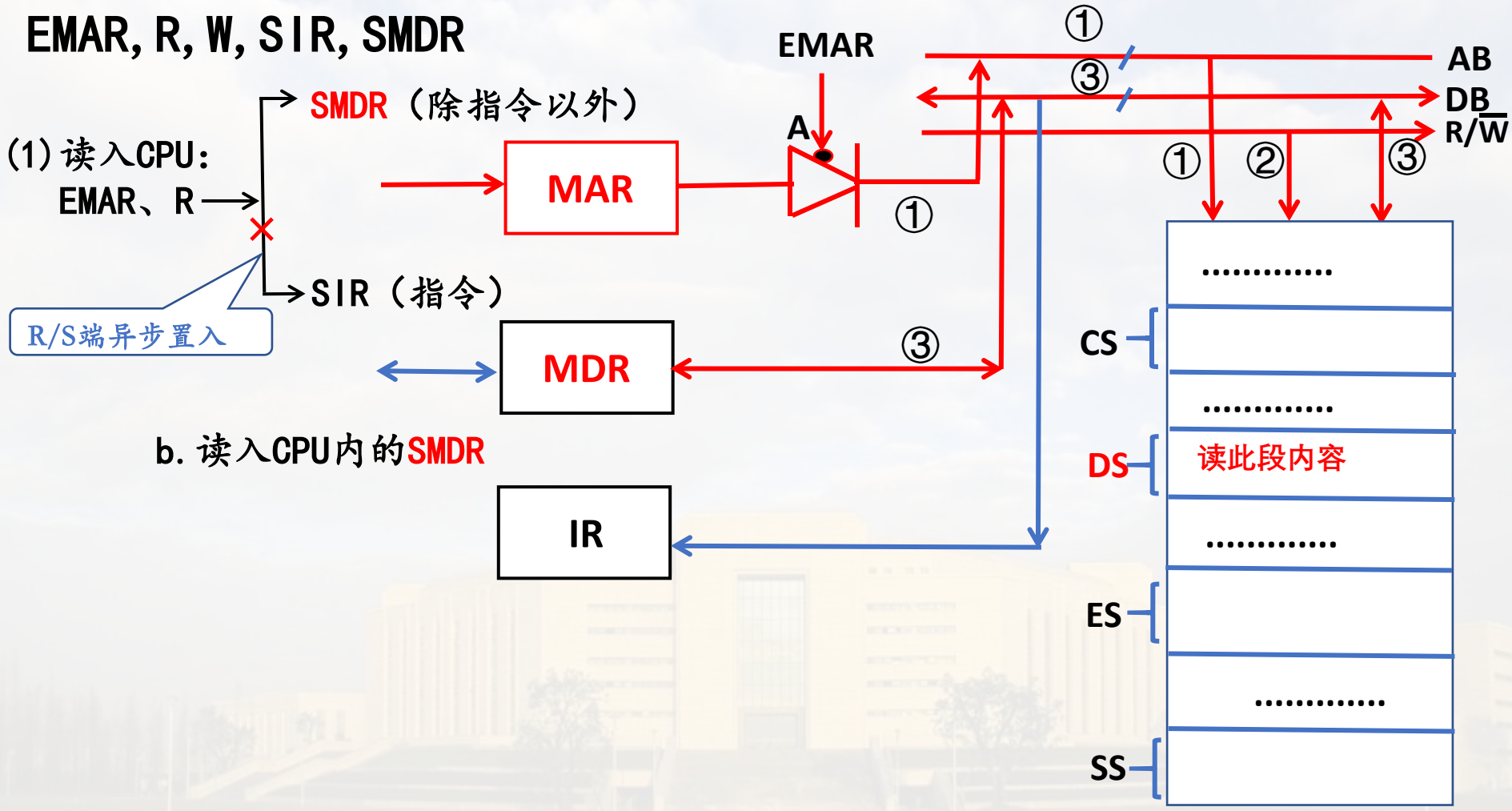
EMAR、R → SIR（指令）

R/S端异步置入

b. 读入CPU内的MDR



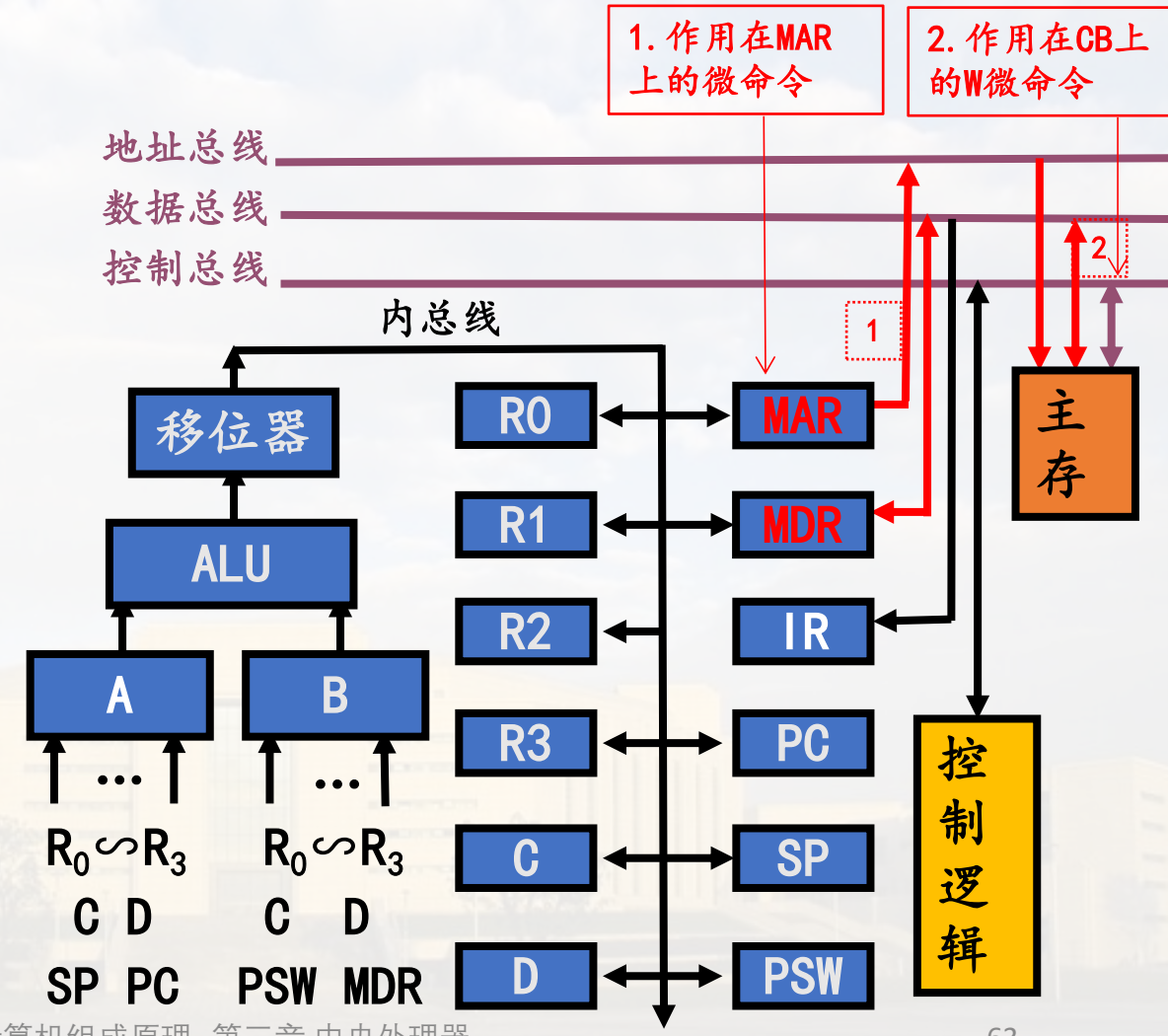
2. 与系统线及主存有关的微命令



2. 与系统线及主存有关的微命令

EMAR, R, W, SIR, SMDR

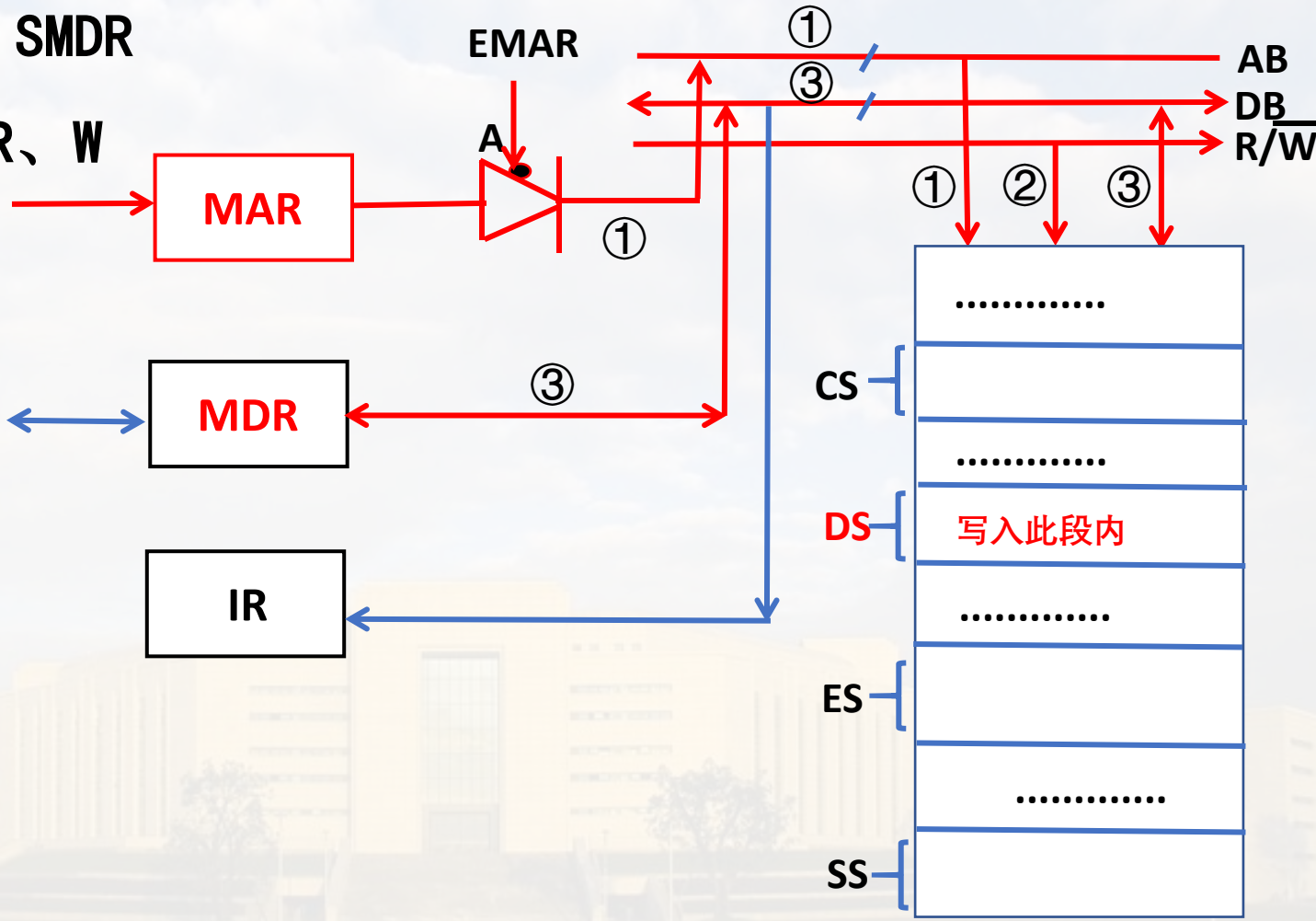
(2) 写入M：EMAR、W



2. 与系统线及主存有关的微命令

EMAR, R, W, SIR, SMDR

(2) 写入M：EMAR、W



总结：微命令设置

将模型机的数据传送路径分为两大类操作，并分段设置相关微命令：

1、数据通路操作：即CPU内部操作，按照信息传送路径分成四段

1）ALU输入选择：

$R_i \rightarrow A$，R_i的取值：$R_0 \backsim R_3$，C，D，PC，SP

$R_j \rightarrow B$，R_j的取值：$R_0 \backsim R_3$，C，D，MDR，PSW

2）ALU功能选择：

$S_0 \backsim S_3$，C_0，M

3）移位器功能选择：

直传DM、左移SL、右移SR

4）分配脉冲（打入到寄存器中的脉冲）：

$CPR_0 \backsim CPR_3$, CPC, CPD, CPMDR, CPMAR, CPPSW, CPPC, CPSP

2、与系统线及主存有关的微命令

EMAR, R, W, SIR, SMDR

(1) 读入CPU：EMAR、R → SMDR（除指令以外）
(1) 读入CPU：EMAR、R → SIR（指令）

(2) 写入M：EMAR、W

四、主机与外部的数据通路与信息传送控制方式

主机与外设的连接模式
- 辐射型/星型
- 总线型
- 通道型 ⇨ I/O处理机

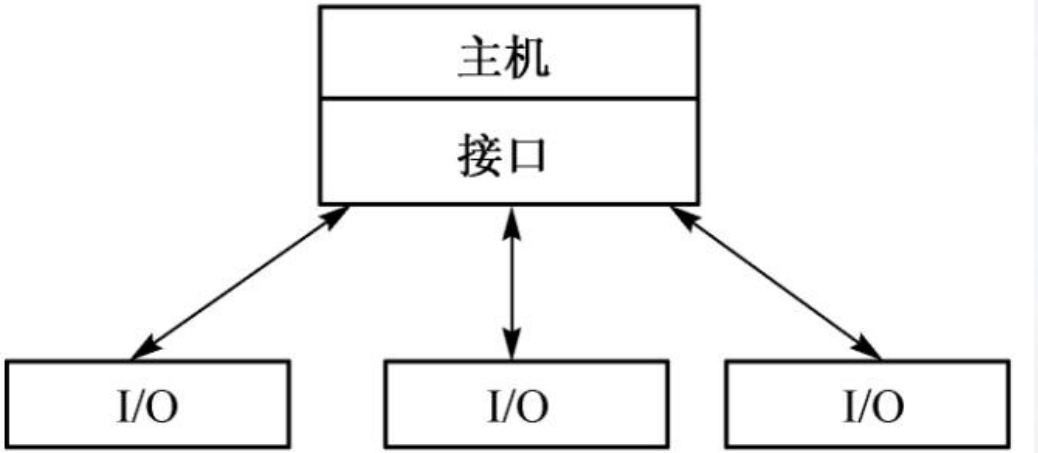
对信息传送的控制方式
- 直接程序传送方式
- 程序中断方式
- 直接有储器访问方式（DMA）

1、主机与外围设备的连接方式

① 辐射型（星型）

定义：主机与各外围设备间有单独的数据通路。

结构图：

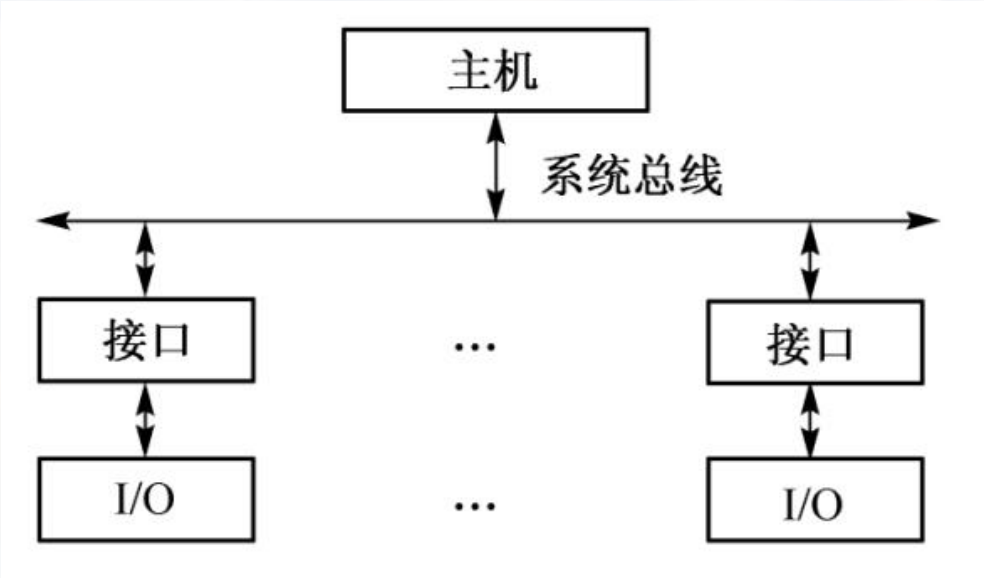


特点：速度较快，但不易扩展。

② 总线型

定义：各外设通过各自接口直接与公共的系统总线相连。

结构图：

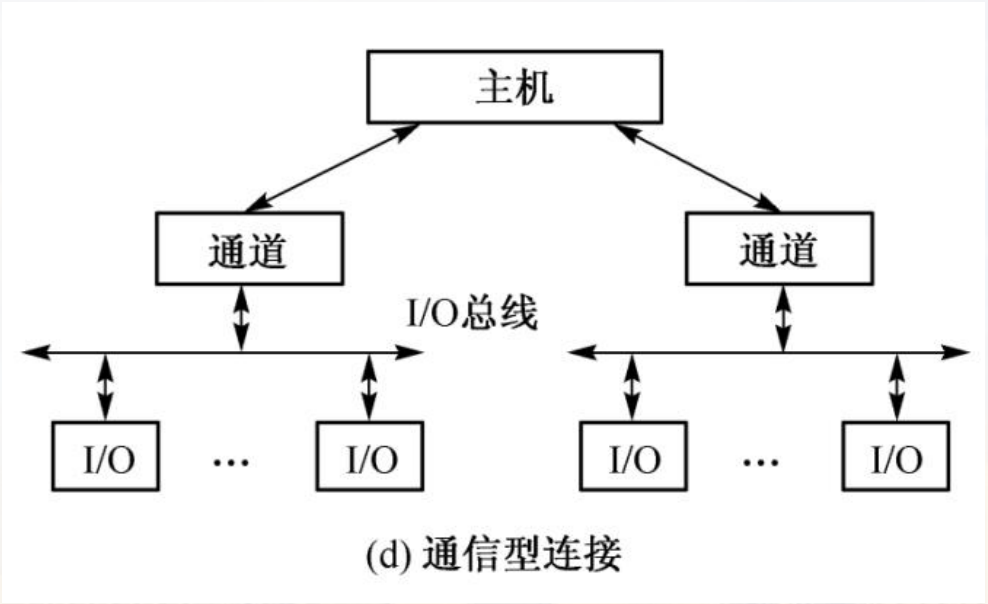


特点：结构简单，易扩展，但如果整个系统只有一组系统总线时，信息吞吐量有限，且速度较慢。

③ 通道型

定义： 采取一种称为多种连接模式的部件去连接外设，这种部件称为通道。

结构图：



(d) 通信型连接

特点： 并行能力强。

2、信息传送的控制方式

主机与外设进行信息交换时，应考虑两个问题：

(1)在CPU启动外设后,在外设准备与操作期间,是让CPU等待还是让它并行地处理执行主机的程序。

(2)如果让CPU并行地执行程序,外设将如何唤起CPU去执行I/O操作?CPU将通过子程序处理I/O操作，还是通过硬件隐指令方式处理?

① 直接程序传送方式

定义： CPU直接利用I/O指令程序实现I/O传送，在外设工作期间，CPU不执行与I/O无关的操作。

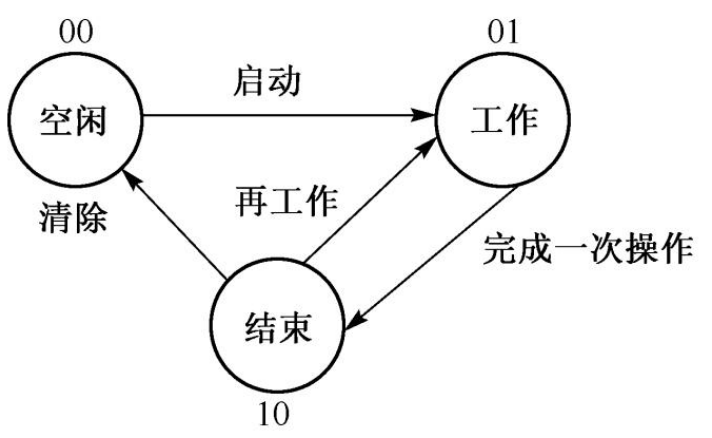
主机状态： CPU处于查询—等待—执行状态。

外设状态： 在外设接口的状态字中设置两位表示状态。

空闲——外设不工作，00

工作（忙）——外设置在执行操作，01

结束（完成）——外设完成一次操作，10



特点： CPU不能与外设并行地工作，因而CPU利用率低，并且CPU不能响应来自外部的随机请求。因此，只适用于低速外设。

② 程序中断方式

如何提高CPU的利用率？

在等待外设准备与操作这段时间内，CPU可否并行执行程序？

如行，在外设准备好或完成本次操作后，如何向CPU提出新的请求？

此外，在计算机工作过程中，还可能会遇到一些随机事态，CPU如何响应处理这些事情？

四、主机与外部的数据通路与信息传送控制方式

定义：CPU暂停执行现行程序，转去执行中断程序，以处理某些随机事态，并在处理完毕后自动恢复原程序的执行。

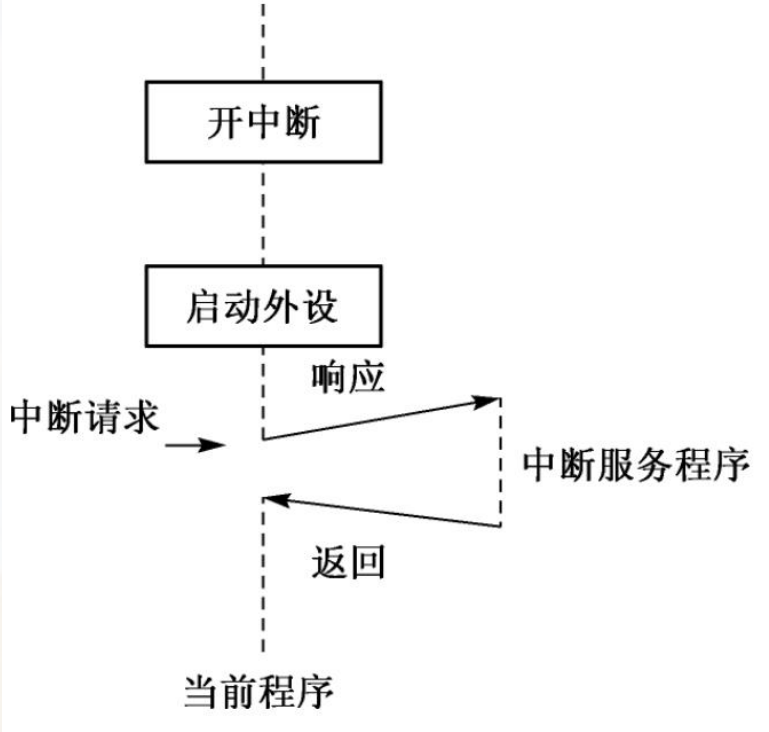
实质：程序的切换过程。包括将断点（PC内容），有关状态信息（PSW）压栈保护起来，然后根据中断号，从中断向量表中得到中断服务程序的地址，送入PC中。（这一切是硬件实现的，称为隐指令）。

特点：随机性

四、主机与外部的数据通路与信息传送控制方式

软件组织与程序安排：将系统中响应的中断请求对应的中断服务模块调入主存，并将其入口地址写入中断向量表。

工作流程：



四、主机与外部的数据通路与信息传送控制方式

硬件组织：

CPU方面：在PSW中设置一个中断位；在模型机中，外部请求只有$IREQ_0$～$IREQ_7$，因此在CPU内还应设置一个判优逻辑。

接口方面：设置了中断接口。

特点：可以处理随机的复杂事态，但程序的切换需花费一定时间，因此，其适用范围是中低速I/O操作与随机请求。

③ 直接存储器存取（DMA）方式

如果主机与高速外存或高速通信设备交换，采用什么方式？

定义： 直接依靠硬件在主存与I/O设备进行简单批量数据传送的一种工作方式，在传送期间不需CPU的程序干预。

实质： **暂停执行程序**。不存在断点，现场的保护，速度很快。

特点： **随机性**

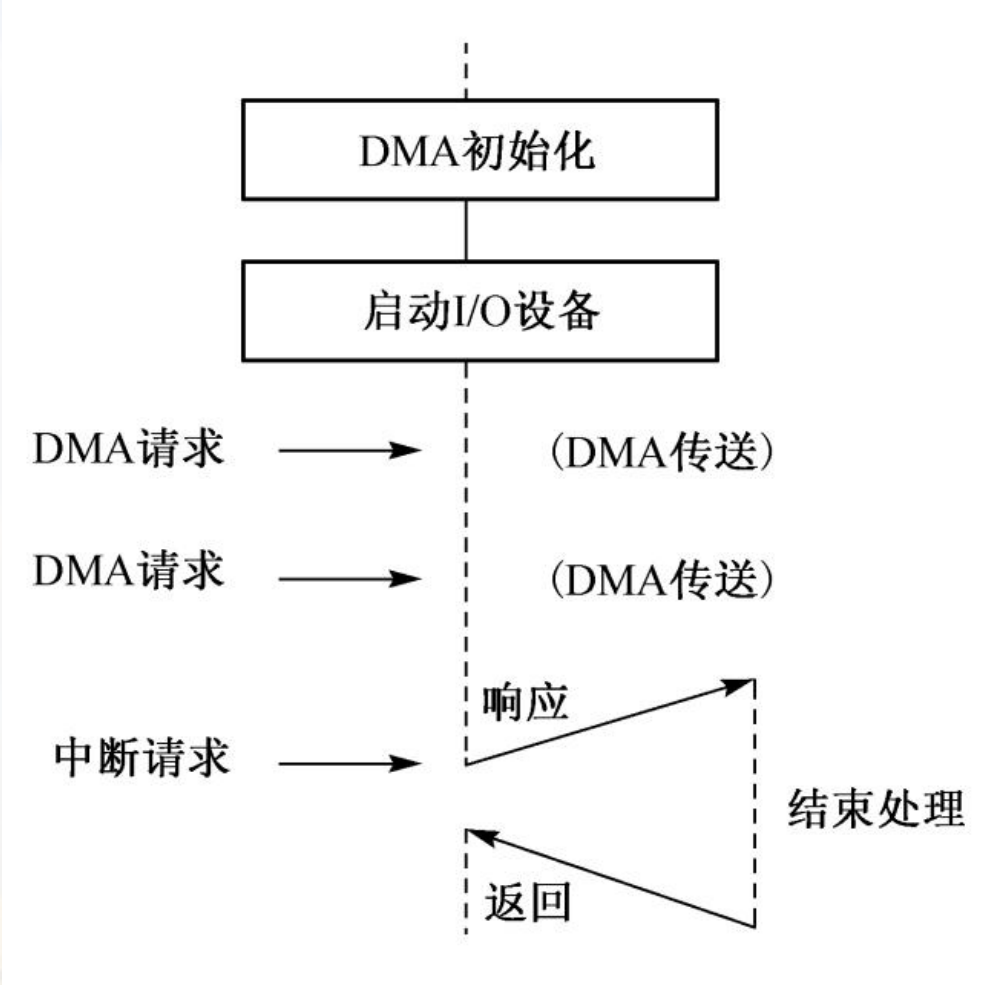
程序组织与DMA初始化：在进行DMA传送前，应对DMA控制器进行初始化，主要初始化以下三个信息：

(1) **送出控制字：**操作类型（输入或输出），外设地址。

(2) **送出主存缓冲区首址：**传送或接收数据的起始地址。

(3) **送出交换量：**DMA是批量传送的，传送的数据块数目。

DMA初始化后的工作流程：



四、主机与外部的数据通路与信息传送控制方式

硬件组织： 由DMA控制器控制管理DMA传送。

CPU方面：在CPU的时序系统中，设置专门的DMA周期。在此周期，总线由DMA控制。每当系统总线周期结束（完成一次总线传送）时，CPU对总线控制权转移作出判断，是继续由DMA控制器掌管，还是CPU收回其控制权，恢复正常程序执行。

接口方面：设置DMA控制器与接口。

特点： 适用于高速外设与主存之间的简单批量数据传送。

五、时序控制方式与时序系统

计算机的工作需要分步地执行，这就需要一种时间划分的信号标志——时序信号，以反映在什么时间段或时刻，计算机做了什么操作。

为了形成控制流，在时序方面有三个问题需要考虑：

1. 操作与时序信号之间的关系，即时序控制方式。
2. 指令之间的衔接方式；
3. 如何形成所需的时序信号，即时序系统。

1、时序控制方式

分类 { 同步控制 / 异步控制 }

① 同步控制方式

定义：如果计算机各项操作与统一的时序信号同步，称为同步控制。

时间分配（基本特征）：同步控制方式的基本特征是将操作时间划分为许多时钟周期，周期长度固定，每个时钟周期完成一步操作。

五、时序控制方式与时序系统

同步定时：在许多操作中需要严格地同步定时，如同步打入脉冲。

各部件间的协调：在CPU内，一般采用由CPU提供的统一时序信号来控制部件间信息的传送的。

特点：时序关系比较简单，在是时间安排利用上可能不经济的。

② 异步控制方式

定义：异步控制是指各项操作**按其需要**选择不同的时间，不受统一的时钟周期的约束；各操作之间的衔接与各部件之间的信息交换**采取应答方式**。

时间分配（基本特征）：没有统一的节拍划分与同步定时脉冲，但存在着申请、响应、询问、回答一类的应答关系。

主从设备的概念：申请使用总线，并获得批准后掌管总线控制权的设备，称为主设备，否则为从设备。

特点：时间紧凑，能按不同部件，设备的实际需要分配时间，实现异步应答所需的控制比较复杂。

③ 实际应用中的一些变化

在CPU或设备的内部普遍采用同步控制方式；对连接CPU、主存、外设的系统总线，有的采用同步，有的采用异步控制，但多采用异步控制。在实际应用中，同步控制甚至引入异步应答关系。

2、指令序列间的衔接方式

分类
- 串行顺序处理方式
- 单存储体重叠处理方式
- 双存储体重叠处理方式
- 多存储体重的交叉与重叠处理方式

六、同步控制的时序系统

1、时序系统

1）定义及组成

时序系统：产生节拍，脉冲等时序信号的部件，称为时序系统。

时序系统的组成：
- 一个振荡器：产生脉冲信号
- 一组计数分频逻辑

2）时序划分层次

指令周期：读取并执行一条指令所需的时间，称为指令周期。

一般不作为时序的一级。

（CPU）工作周期：在指令周期中的某一工作阶段所需的时间，称为一个工作周期。一般不同。

时钟周期（节拍）：是时序系统中最基本的时间分段。各节拍的长度相同。

定时脉冲（工作脉冲）：有的操作如打入R，还需严格的定时脉冲，以确定在哪一刻打入。

3）多级时序的划分

① 二级时序（用在微程序控制器中）

指令周期
- 时钟周期（节拍）0—工作脉冲
- ⋮
- 时钟周期（节拍）n—工作脉冲

② 三级时序（用在组合逻辑控制器中）

指令周期
- 工作周期0
  - 时钟周期（节拍）0—工作脉冲
  - ⋮
  - 时钟周期（节拍）n—工作脉冲
- ⋮
- 工作周期m
  - 时钟周期（节拍）0—工作脉冲
  - ⋮
  - 时钟周期（节拍）n—工作脉冲

谢谢观看

计算机组成原理

2025-8-22

信息与软件工程学院

School of Information and Software Engineering