CPMDR: 内总线 打入

SMDR: 外总线 置入

CPU设计的步骤

1、拟定指令系统

指令系统是CPU设计的基础。

拟定指令内容包括: 指令类型、指令功能、指令格式、寻址方式等

2、确定总体结构

- 设置哪些以及多少寄存器;
- 采用什么运算部件; 以确定提供什么控制信号
- 确定信息传送采用什么样的数据通路

3、安排CPU的工作时序

规定所有的操作完成的时间和时机。

比如:总线周期的长度、包含多少个节拍; 每条指令需要多少个总线周期、规定CPU的哪一步工作在哪个工作周期完成等。

4、规定指令流程和微命令序列

第1步: 确定指令流程

将指令每一步操作(信息在寄存器之间的传送, 即寄存器传送级)以流程图的方式描述出来;

第2步: 拟定操作时间表

结合时序, 拟定每一步操作所需要的微操作命令, 再以操作时间表的形式, 将微命令以及微命令产生的条件列出来。

微命令	规定的时间
X	产生条件
Y	产生条件

拟定指令流程和微命令序列是CPU设计过程的关键步骤。

5、形成控制逻辑

- 如果采用组合逻辑设计方法:

  根据第4步的工作, 将微命令产生的条件进行综合、简化后, 形成逻辑表达式, 用逻辑电路予以实现。

- 如果采用微程序设计方法:

  根据第4步的工作, 将对微命令进行编码, 形成微指令, 并存入到控制存储器ROM, 作为控制的核心。

步骤1(拟定指令系统)和步骤2步(确定总体结构)与控制方法(组合逻辑设计或微程序设计)基本无关。

1.模型机的指令格式

指令字长16位，采用寄存器型寻址，指令中给出寄存器号。（主存容量为64K×16位）

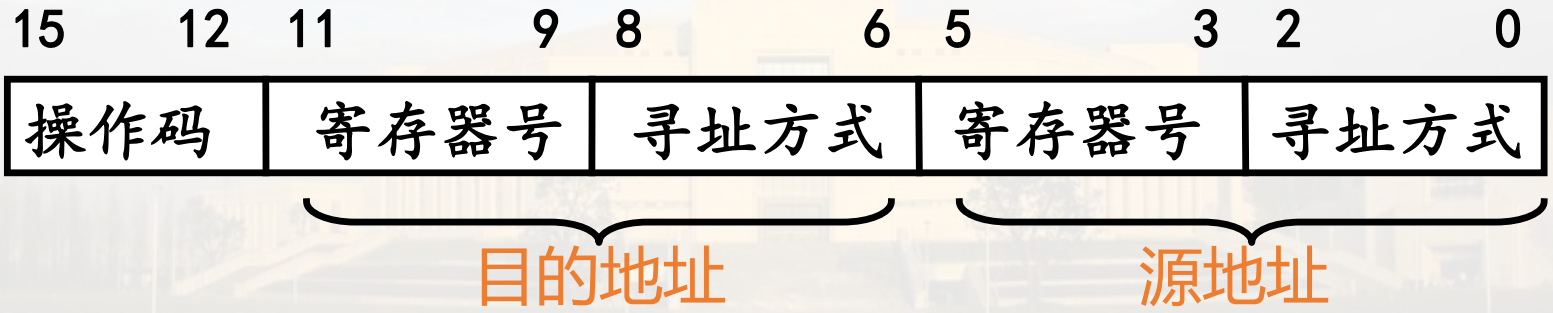
结合高级语言，考虑CPU应该有哪些基本类型的指令?

1.模型机的指令格式

(1) 指令格式

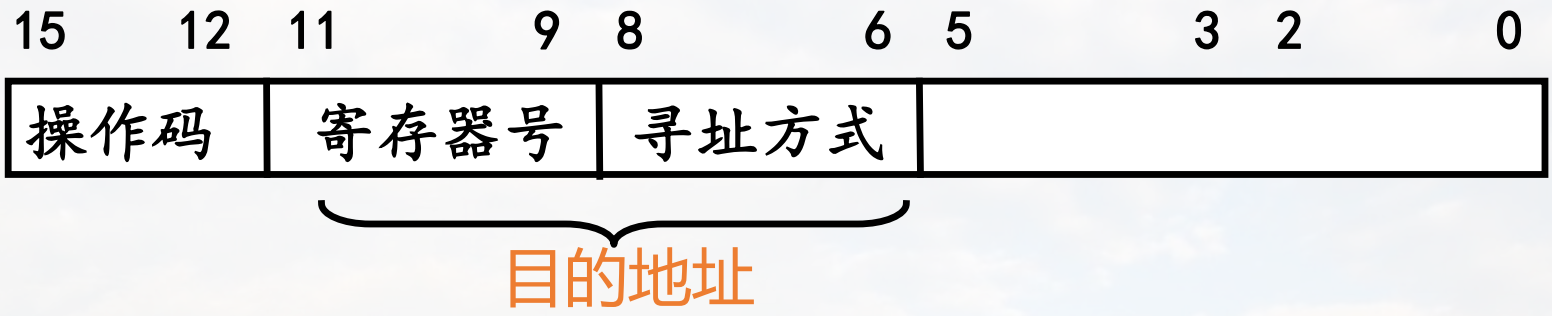
模型机指令格式分类
- 双操作数指令
- 单操作数指令
- 转移指令

1、双操作数指令

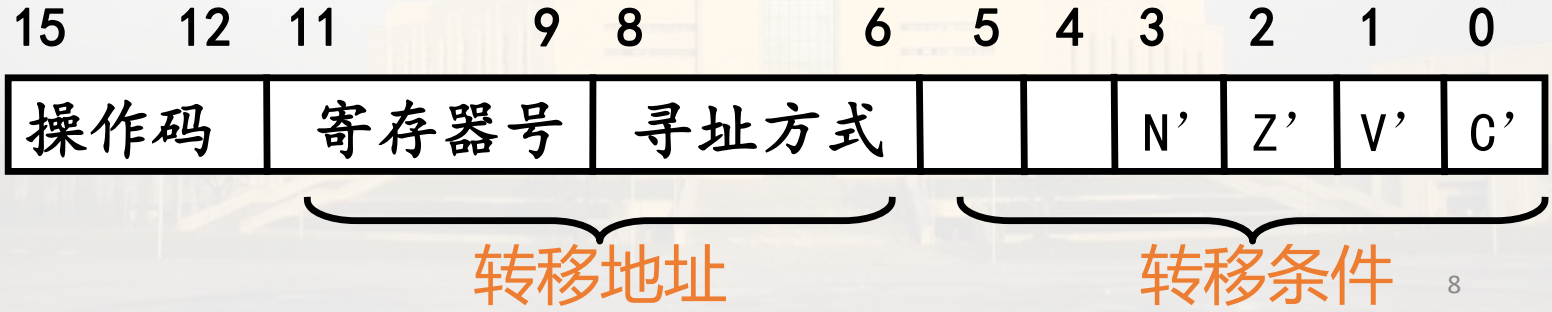


1.模型机的指令格式

2、单操作数指令



3、转移指令



2.寻址方式

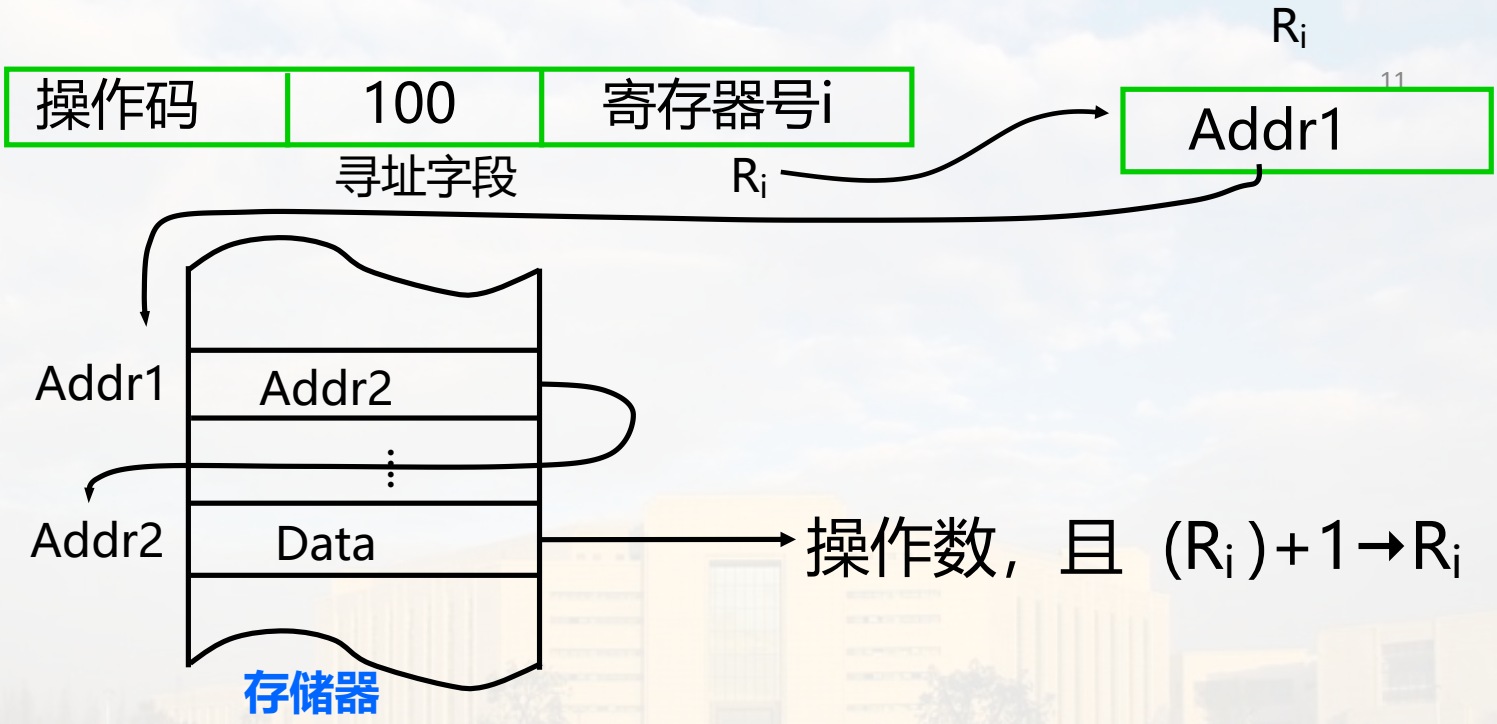
CPU可编程访问的寄存器：通用寄存器R0-R3

指令计数器PC、堆栈指针SP、程序状态字PSW

寻址方式	编码	助记符	定义 (表3-4 P130)
寄存器寻址	000	R	(R)为操作数
寄存器间址	001	(R)	(R)为操作数地址
自减型寄存器间址	010	-(R) -(SP)	(R)-1为操作数地址 (SP)-1为栈顶地址

寻址方式	编码	助记符	含义
立即/自增型寄存器间址	011	(R)+	(R)为操作数地址，访问后(R)+1
		(SP)+	(SP)为栈顶地址，出栈后(SP)+1
		(PC)+	(PC)为立即数地址，取数后(PC)+1
直接/自增型双间址	100	@(R)+	(R)为间接地址，访问后(R)+1
		@(PC)+	(PC)为间接地址，取数后(PC)+1

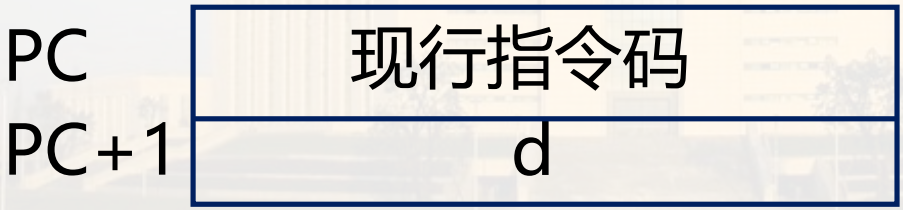
直接/自增型双间址寻址过程示意:



一、指令系统的设计

寻址方式	编码	助记符	定义
变址/相对寻址	101	X(R) X(PC)	(R)+d为有效地址 (PC)+d为有效地址
跳步	110	SKP	跳过下条指令执行

注：形式地址d存放在现行指令单元的下一个单元，因此需要根据PC值来读取该形式地址。



变址寻址流程的依据

3.目标指令集

操作码	助记符	含义 (表3-5 P131)	
0001	MOV	传送	← 用于数传、堆栈、I/O操作
0010	ADD	加	双操作数指令
⋮	⋮	⋮	
0110	EOR	异或	
1000	INC	加1	单操作数指令
⋮	⋮	⋮	
1101	NEG	变补	
1110	JMP/RST	转移/返回	
1111	JSR	转子（调用子程序）	

指令类型的说明:

① 数据传送类指令

完成数据传送、堆栈操作、I/O操作等

包括:R←→R　　R←→M　　M←→M　　R←→I/O

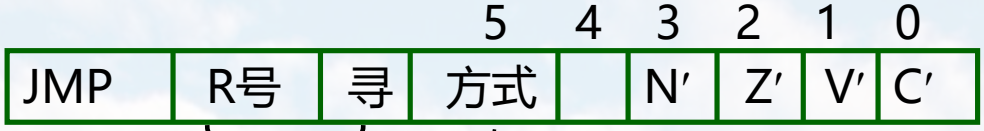
采用统一编址方式, 不需要专用I/O指令

② 转移指令

指令指明转移条件和转移地址

如下图所示:

一、指令系统的设计



转移地址

寄存器号

寻址方式

转移指令的低4位中，某一位为1，表示对它所对应的标志位进行判断

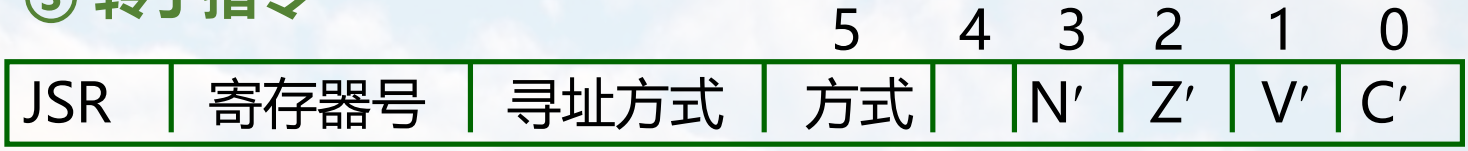
“方式”的两种状态，规定了是“1”转移还是“0”转移

0	0	0	0	0	无条件转移
0	0	0	0	1	无进位转 (C=0)
0	0	0	1	0	无溢出转 (V=0)
0	0	1	0	0	数非零转 (Z=0)
0	1	0	0	0	数非负转 (N=0)
1	0	0	0	1	有进位转 (C=1)
1	0	0	1	0	有溢出转 (V=1)
1	0	1	0	0	数为零转 (Z=1)
1	1	0	0	0	数为负转 (N=1)

为1，表示对N标志进行判断

条件满足，转转移地址；条件不满足，顺序执行。

③ 转子指令



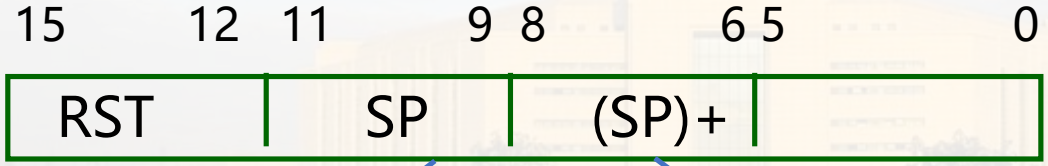
子程序入口

功能： (SP)-1
(PC) 压栈 } 保存返回地址

子程序入口地址 -> PC

④ 返回指令

采用自增型寄存器间接寻址 —(SP)+



(SP) -> PC
(SP)+1 ->SP

指明间址寄存器　　指明自增型寄存器间址

1.部件设置

(1) 主要部件设置

为了使数据传送控制简单、集中，采用以ALU为中心的总线结构。

(1) 组成

包括四个部分：ALU部件；寄存器组；存储器；控制系统；

(2) 特点

内总线采用单向数据总线(20位)；分立寄存器

- 低16位，用于连接除PSW寄存器外的其余寄存器的D输入端。
- 高4位分别对应ALU的状态标志位。并与PSW的低4位对应相连。

ALU为内部数据传送通路的中心；
与系统总线的连接通过MAR、MDR实现。

1.部件设置

(2) 寄存器

- **可编程寄存器（16位）**

通用寄存器：R0(000)、R1(001)、R2(010)、R3(011)

堆栈指针：SP(100)　　指令计数器：PC(111)

程序状态字：PSW(101)

	4	3	2	1	0
(可扩展)	I	N	Z	V	C

条件码: C=1(有进位); V=1(有溢出); Z=1(结果为0);

N=1 结果为负。

中断允许标志I: 为“1”开中断,为“0”关中断。

- **非编程寄存器（16位）**

暂存器C：约定从主存中读取源地址或源操作数时，使用C暂存。

暂存器D：约定暂存目的地址、目的操作数或运算结果。

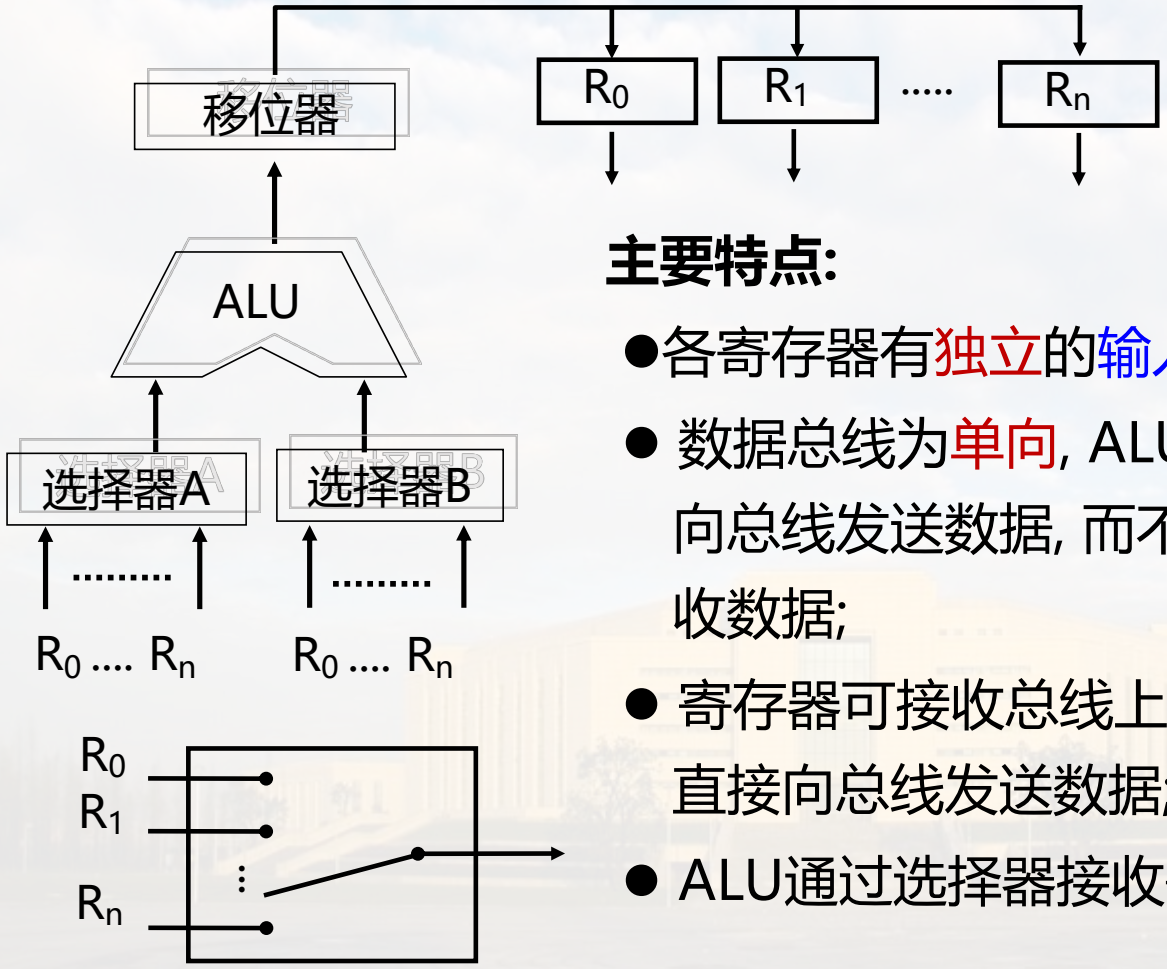
指令寄存器IR:存放现行指令。指令码经数据总线直接**置入**IR

地址寄存器MAR
数据寄存器MDR
> 实现CPU与主存的接口

用于控制的寄存器有：指令寄存器IR、程序计数器PC、程序状态字寄存器PSW

用作主存接口的寄存器是：地址寄存器MAR、数据缓冲寄存器MDR

模型机采用 - 单组内总线、分离(立)的寄存器结构

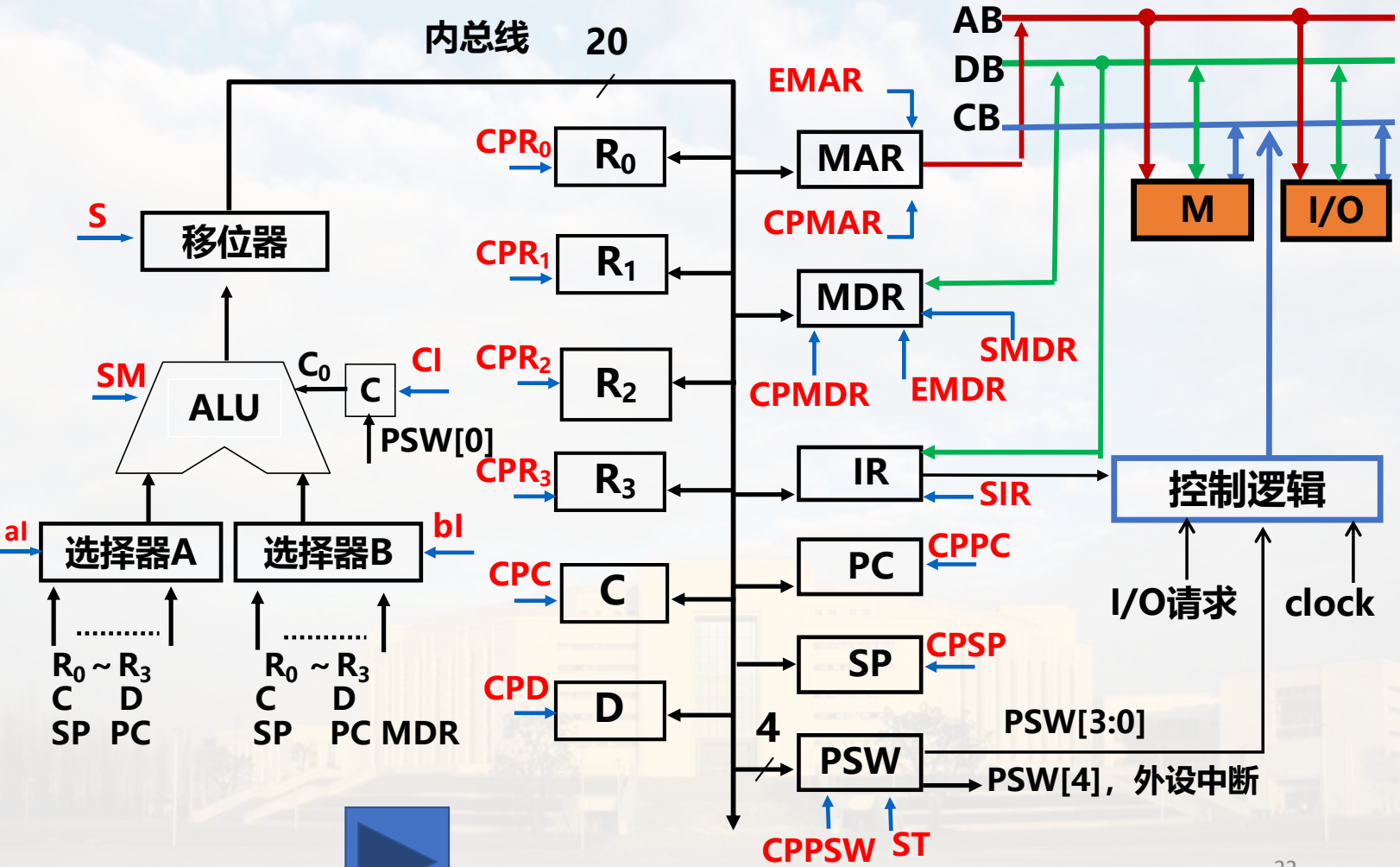


回顾

主要特点:

- 各寄存器有独立的输入口和输出口
- 数据总线为单向, ALU只能通过移位器向总线发送数据, 而不能直接从总线接收数据;
- 寄存器可接收总线上的数据，但不能直接向总线发送数据;
- ALU通过选择器接收寄存器的数据。

二、部件与与数据通路


内总线
20
AB
DB
CB
EMAR
CPR0
R0
MAR
CPMAR
S
移位器
CPR1
R1
MDR
M
I/O
SMDR
SM
ALU
C0
C
CI
CPR2
R2
CPMDR
EMDR
PSW[0]
CPR3
R3
IR
SIR
控制逻辑
al
选择器A
选择器B
bl
PC
CPPC
CPC
C
I/O请求
clock
R0 ~ R3
C D
SP PC
R0 ~ R3
C D
SP PC MDR
SP
CPSP
CPD
D
4
PSW[3:0]
PSW
PSW[4]，外设中断
CPPSW
ST

二、部件与与数据通路

- R_0、R_1、R_2、R_3：打入脉冲CPR_i
- 暂存器C、D：打入脉冲CPC、CPD
- MAR：打入脉冲CPMAR，有效时，从内总线接收地址；
  开门信号EMAR，有效时，地址输出到地址总线；
- MDR：打入脉冲CPMDR，上升沿将内部总线数据打入MDR；
  置入信号SMDR，将外部总线数据置入MDR；
  开门信号EMDR：有效时，MDR输出到DB
- PC、SP、PSW：打入脉冲CPPC、CPSP、CPPSW
- IR：置入信号SIR

●MAR

读取指令/存取操作数/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。

即：CPU访问主存时，首先送出地址码，然后送出/接收数据，需：

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；

●MDR

MDR
- 输出时
  - 输出至DB(Data Bus数据总线)（EMDR）
  - 输出至ALU的**B**选择器
- 输入时
  - 来自内总线的输入（打入）（CPMDR）
  - 来自DB的输入(置入)（SMDR）

主存的读写由控制命令R/W决定传送方向。

R：由主存单元 → 数据总线 → MDR
(需要R和SMDR都有效)

W：由MDR → 数据总线 → 数据单元
(需要W和EMDR都有效)

(3) 运算部件设置 (16位)

ALU { SN74181 4片
SN74182 1片

SM信号:M、C0、S3-S0

选择器A
选择器B
> 选择数据来源:Ri→A、Ri→B

移位器 ： 实现直送、左移、右移、字节交换

3.各类信息传送途径

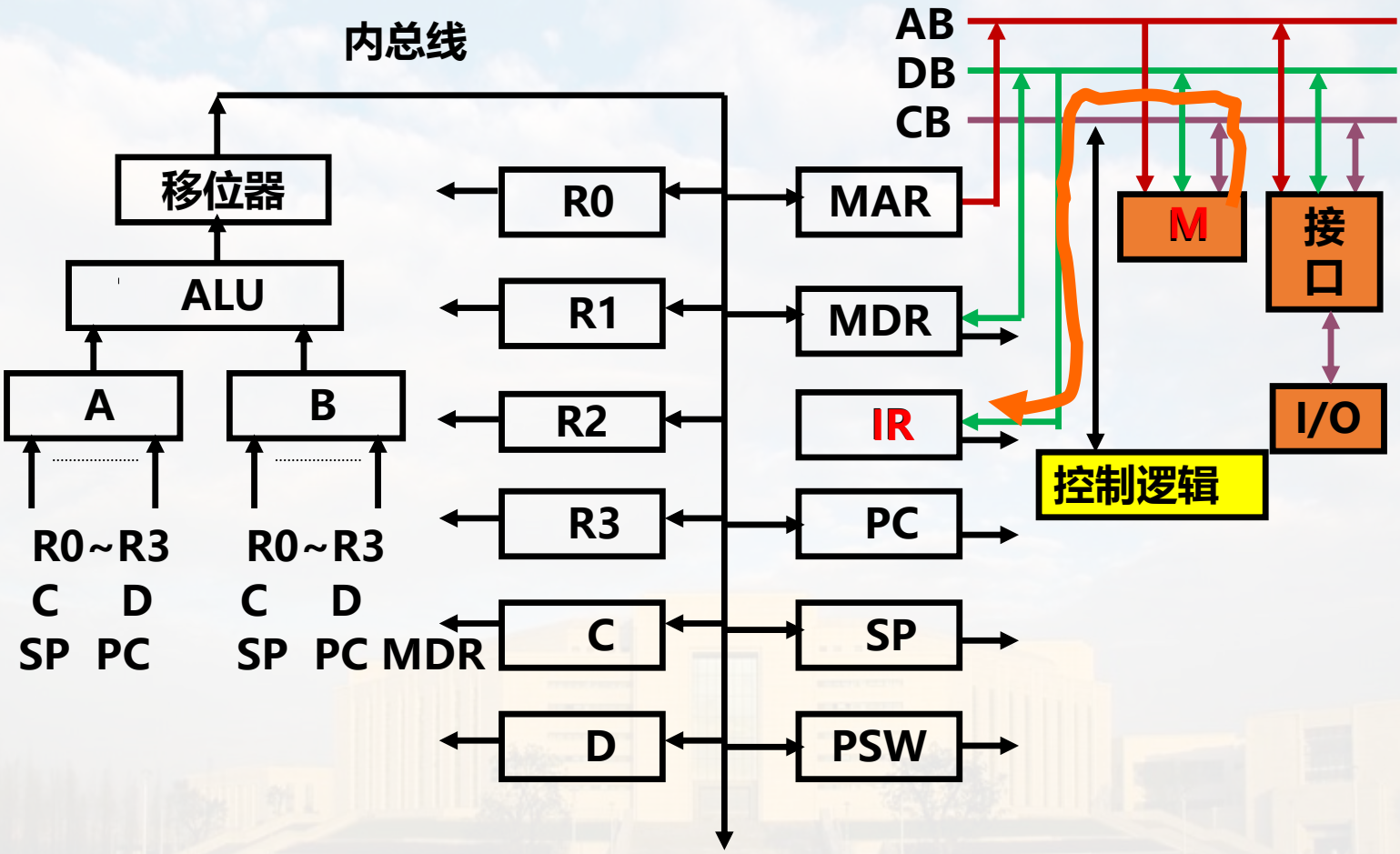
按照信息的类型分类, 包括:

(1) 指令信息

(2) 地址信息
- 1) 指令地址
- 2) 指令地址加1
- 3) 转移地址
- 4) 操作数地址

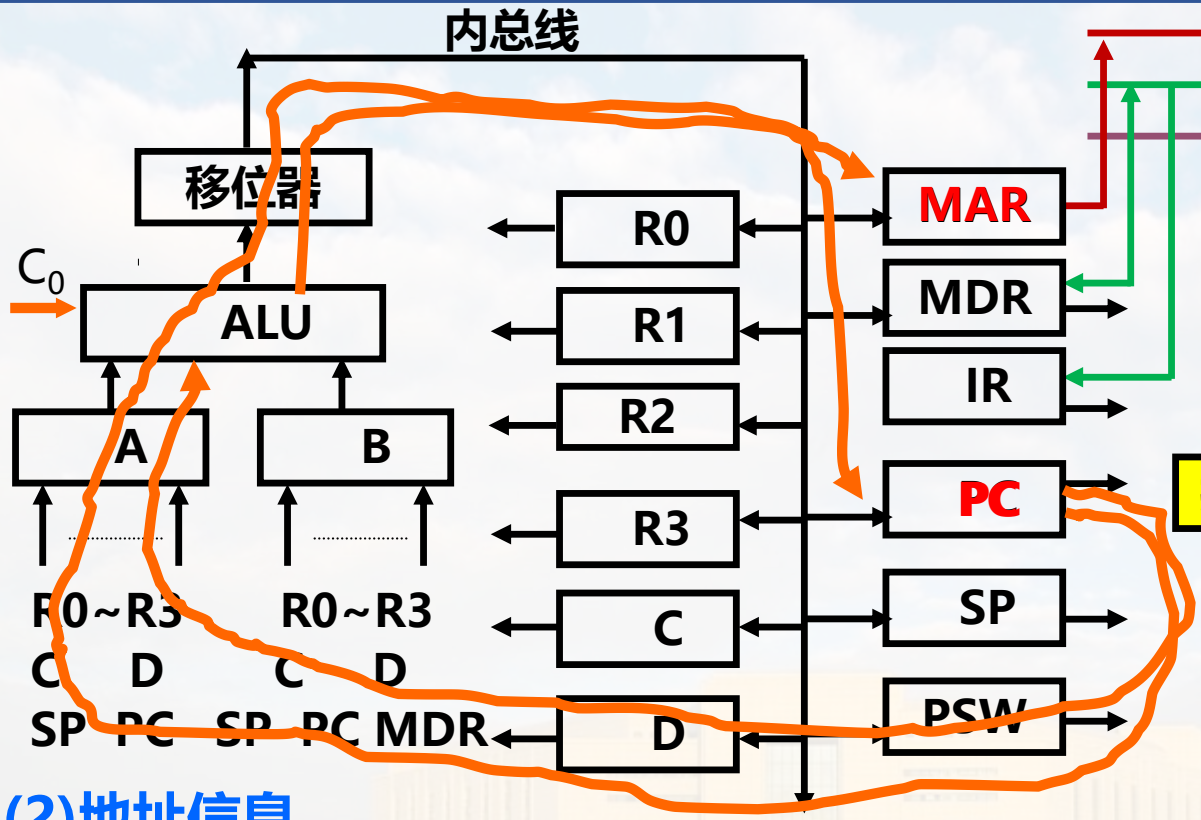
(3) 数据信息
- 1) R ⟷ R
- 2) R ⟷ M
- 3) M ⟷ M

二、部件与与数据通路



(1)指令信息 M → DB 置入 IR

二、部件与与数据通路



(2)地址信息

1）指令地址

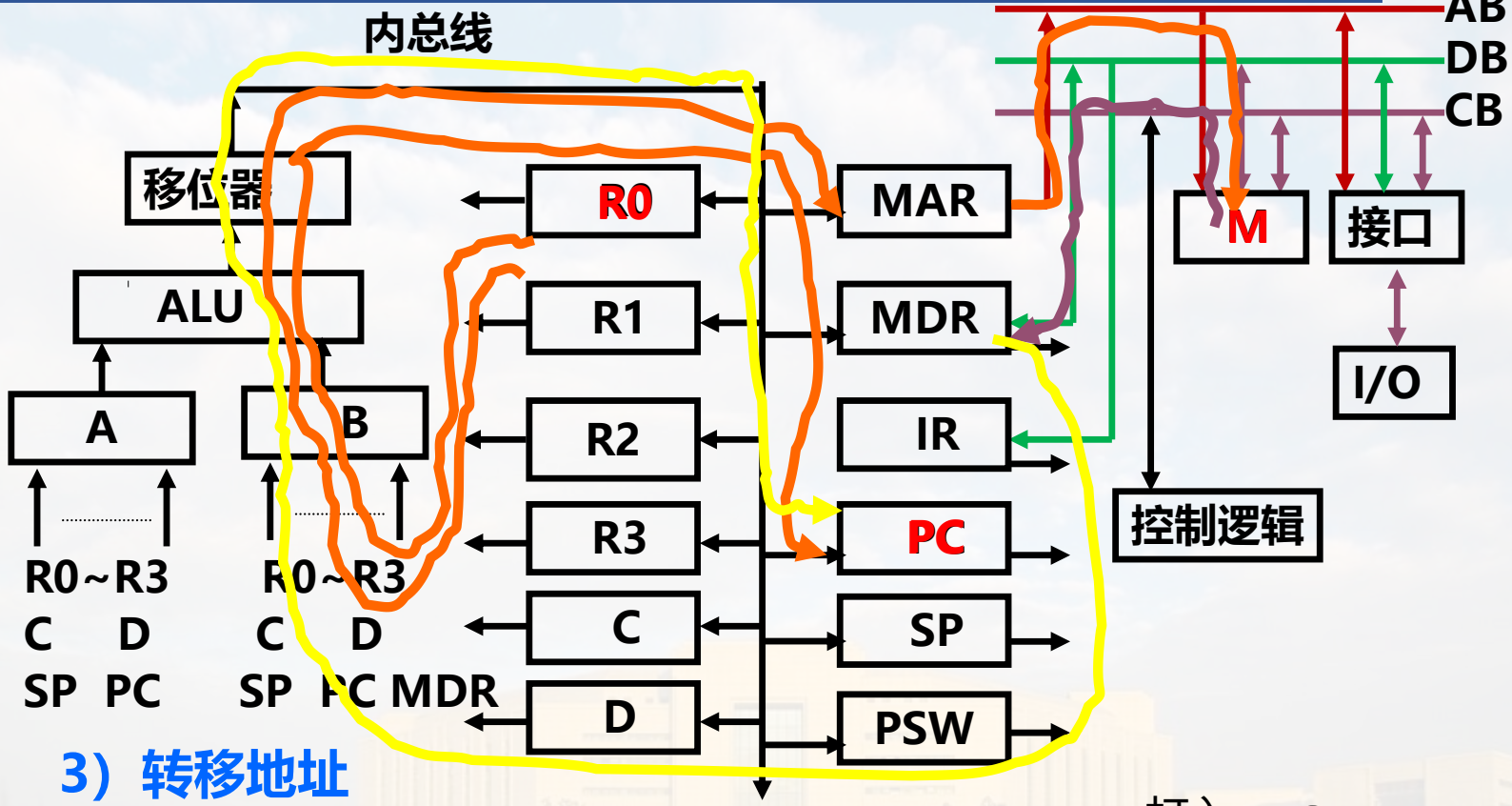
2）指令地址加1

PC → A → ALU → 移 → 内 —打入→ MAR

PC → A → ALU → 移 → 内 —打入→ PC

C0置 “1”

二、部件与与数据通路



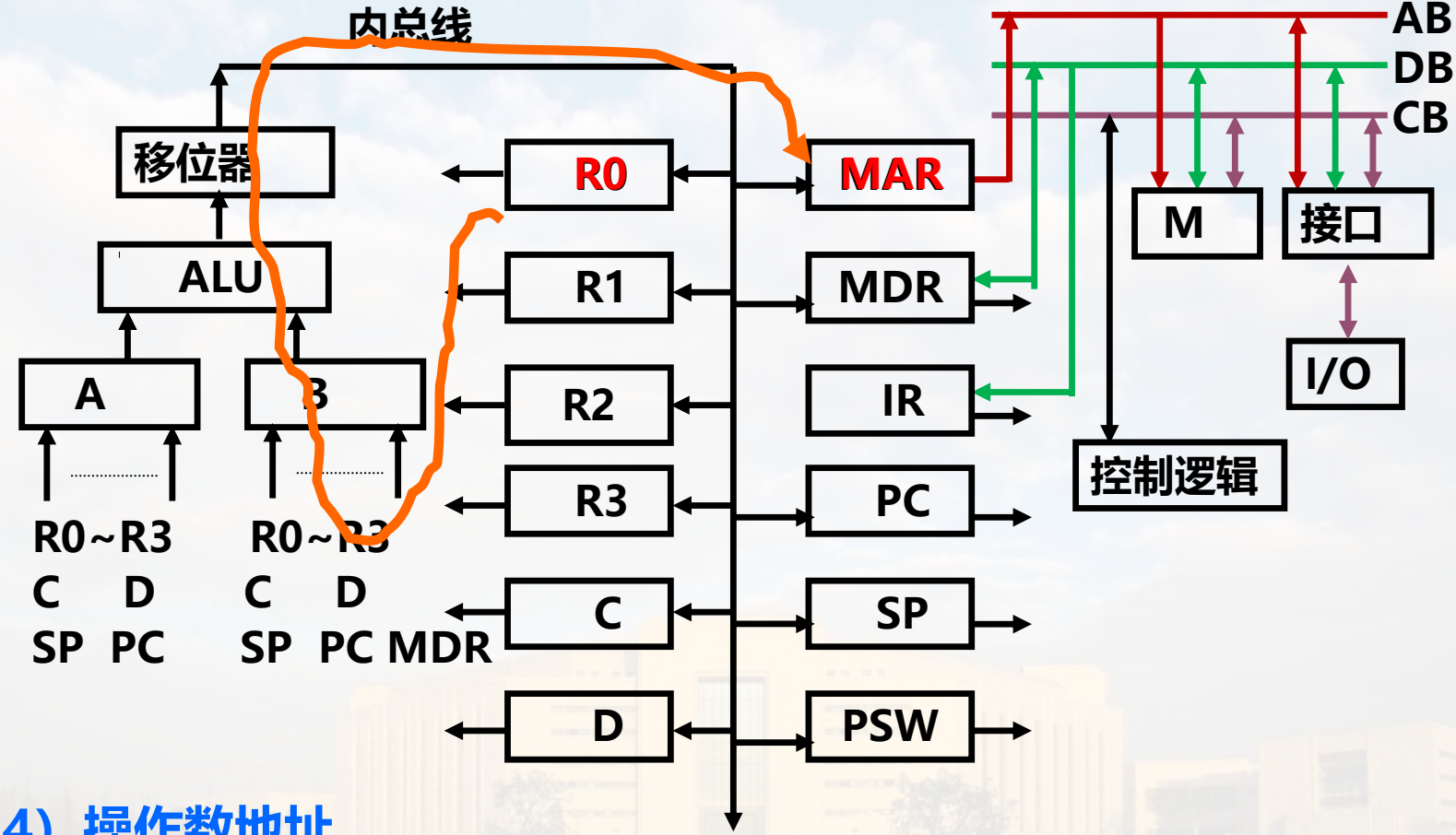
3）转移地址

寄存器寻址：R0 → B → ALU → 移 → 内 → 打入 PC

寄存器间址：R0 → B → ALU → 移 → 内 → 打入 MAR →

→ AB → M → DB → 置入 MDR → B → ALU → 移、内 → PC

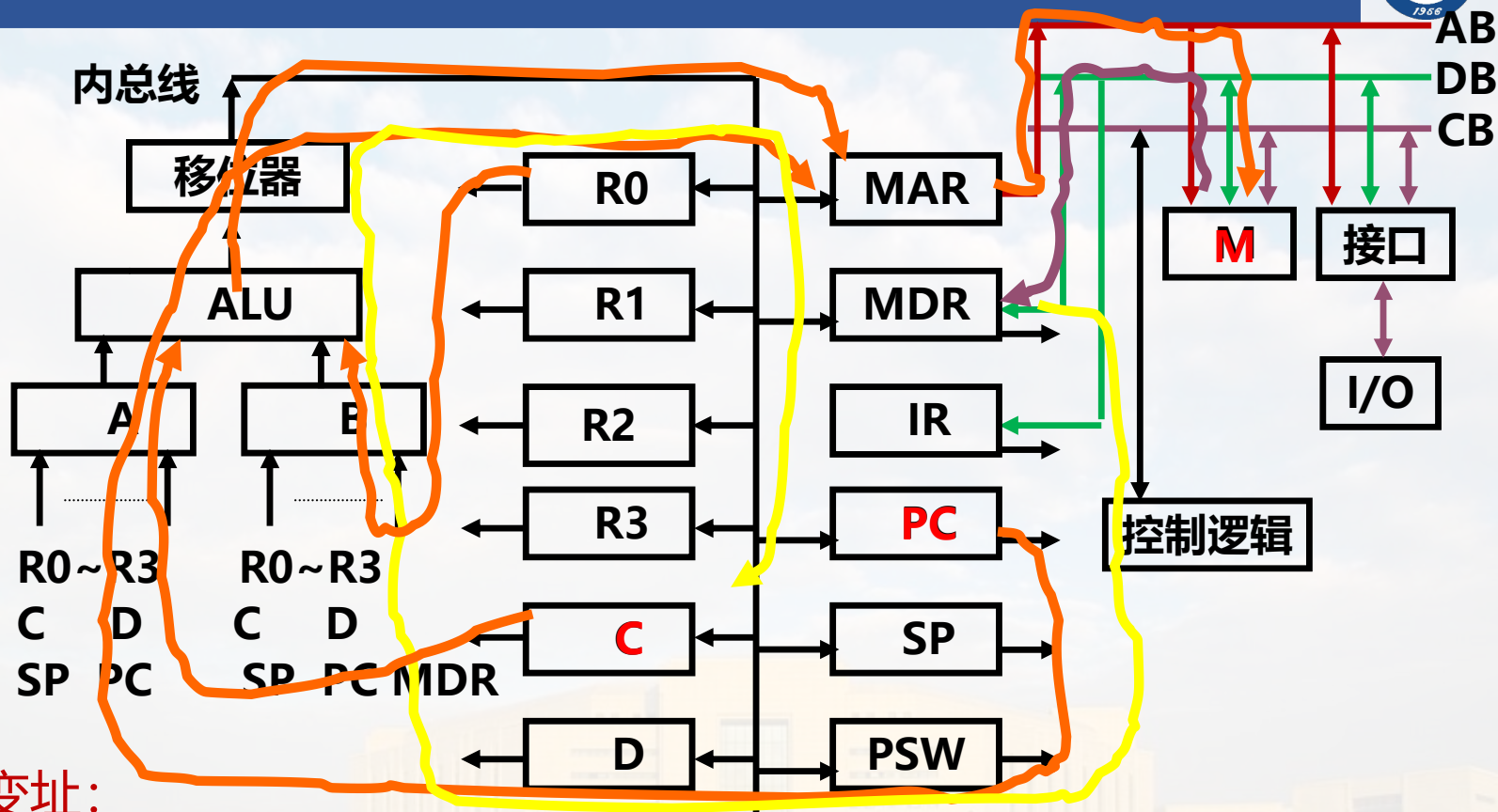
二、部件与与数据通路



4）操作数地址

寄存器间址：R0 → B → ALU → 移 → 内 —打入→ MAR

二、部件与与数据通路



变址：

PC → A → ALU → 移 → 内 → MAR → AB → M

M → DB → MDR → B → ALU → 移 → 内 → C

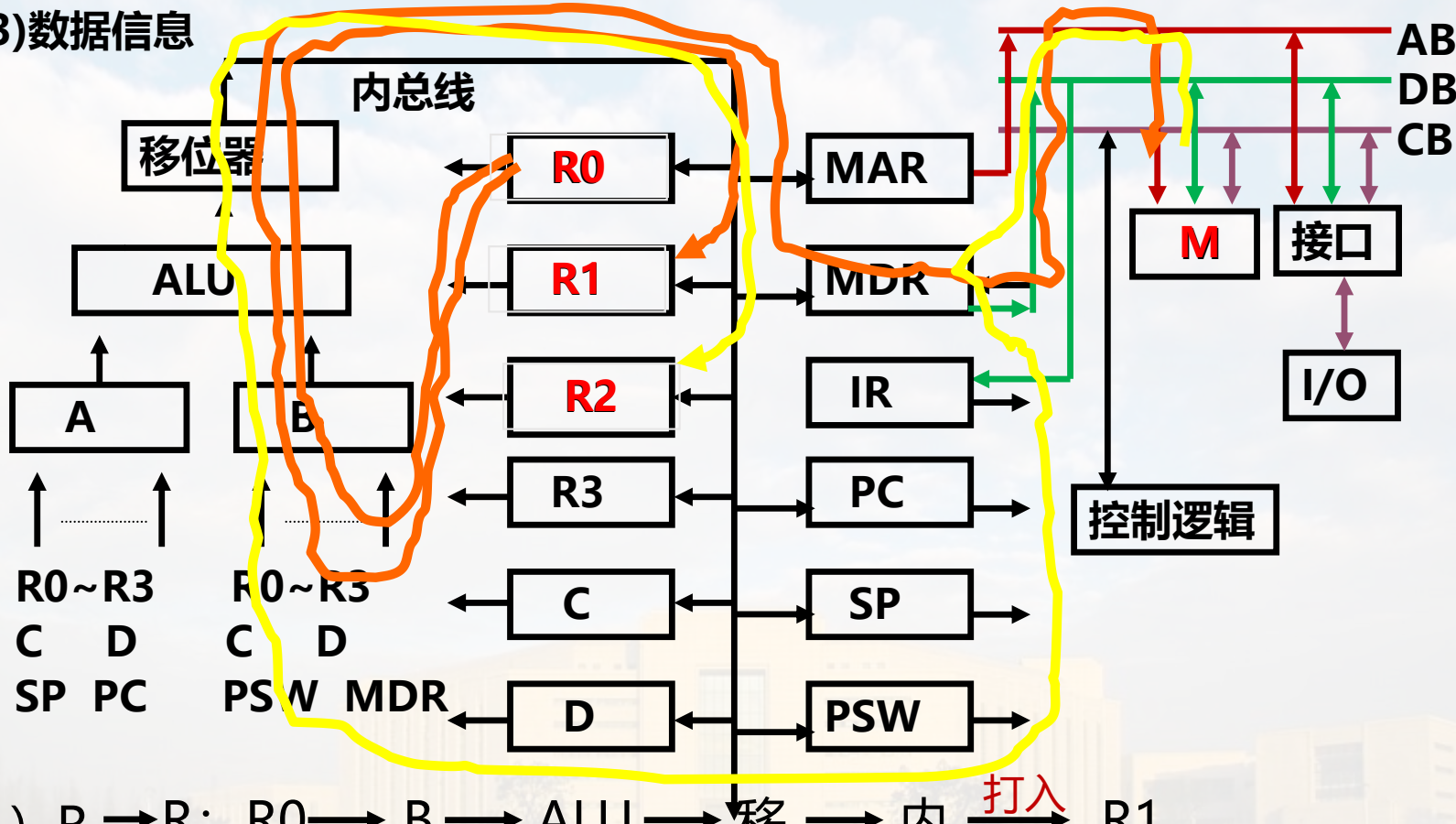
R0 → B

C → A

ALU → 移 → 内 → MAR

二、部件与与数据通路

(3)数据信息

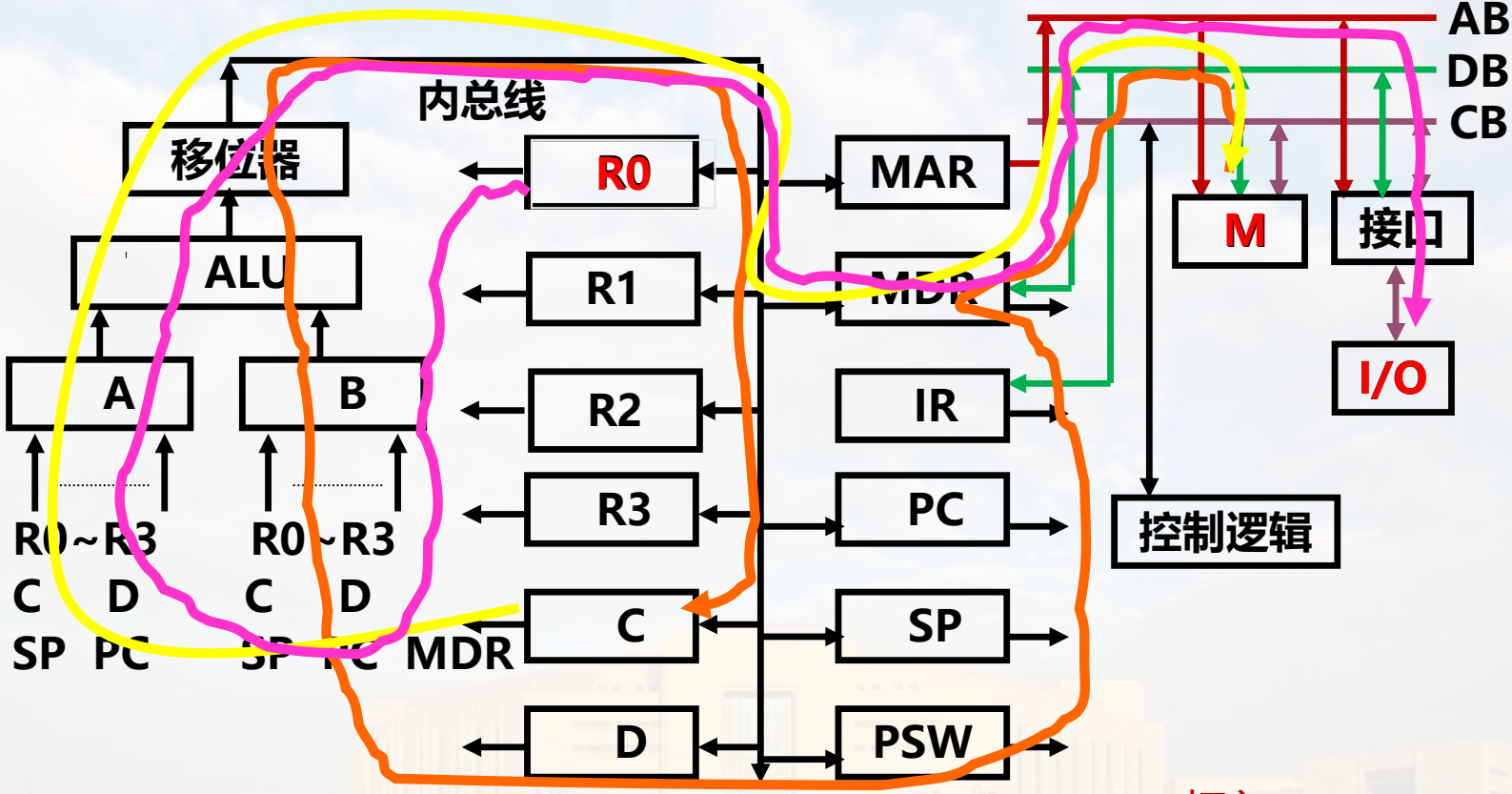


1) R→R: R0→B→ALU→移→内 —打入→ R1

2) R→M: R0→B→ALU→内 —打入→ MDR→DB→M

3) M→R: M→DB —置入→ MDR→B→ALU→移、内→R2

二、部件与与数据通路



4) M⟶M：M(源)⟶ DB ⟶MDR⟶ ALU ⟶ 内 —打入→ C
(计算目的地址)C⟶ ALU ⟶内 ⟶ MDR⟶DB⟶ M(目的)

5) R⟶ I/O： R0 ⟶ ALU⟶ 内⟶MDR⟶ DB⟶ I/O

二、部件与与数据通路

— 关于存储器之间的数据传送(M-M)的说明:

问题: 为什么要将从单元中取出的数据暂存到C寄存器，然后再存入存储单元?

M(源) ⟶ DB ⟶ MDR ⟶ ALU ⟶ 内 ⟶ C

C ⟶ ALU ⟶ 内 ⟶ MDR ⟶ DB ⟶ M (目的)

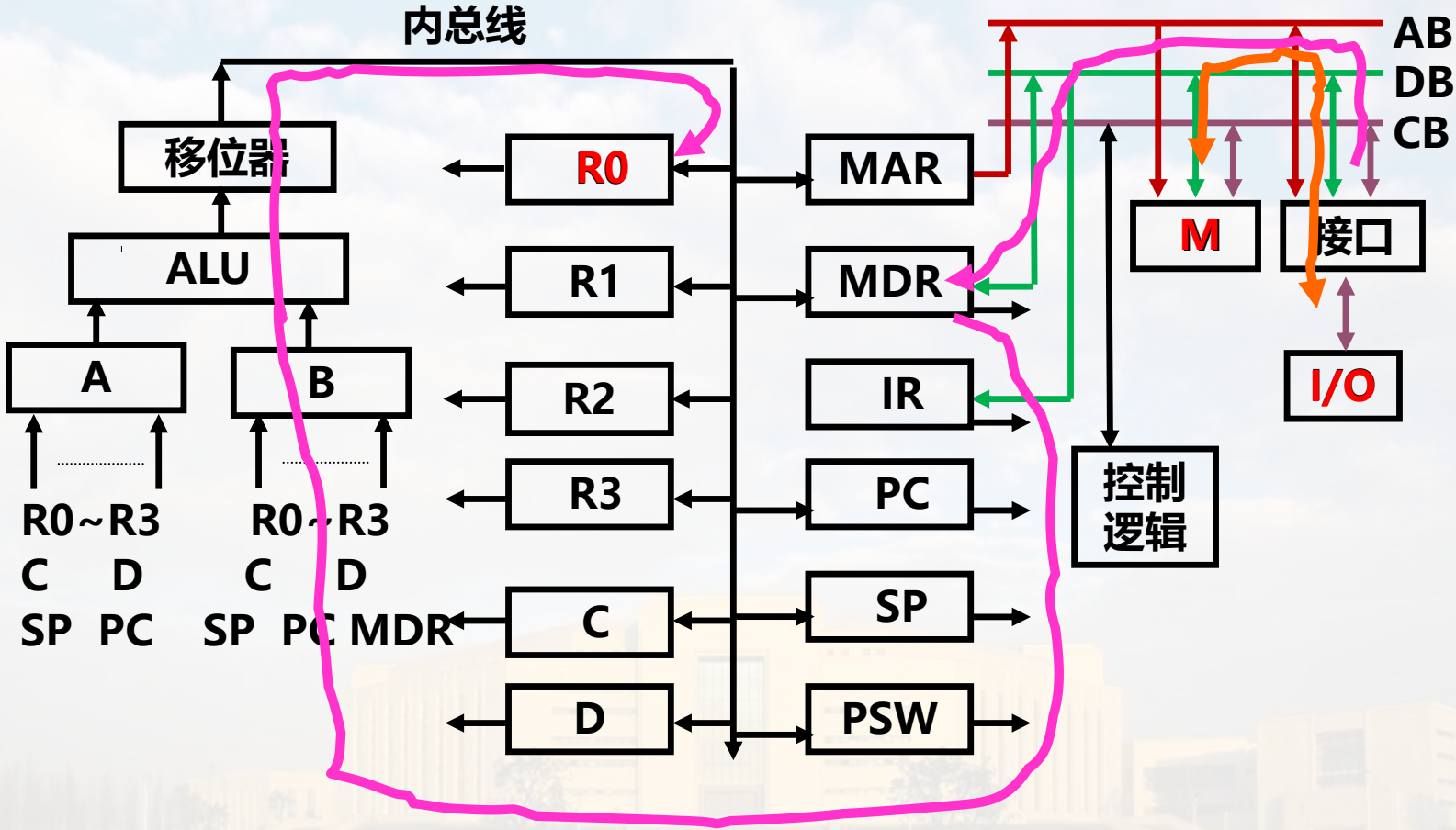
以下数据传送通路是否可行?

M(源) ⟶ DB ⟶ MDR ⟶ DB ⟶ M(目的)

由于目的单元的寻址可能是存储器间接寻址, 从存储单元中读出的内容作为目标地址, 而该内容读出后需要存入MDR, 再传送到MAR。

因此, 需要将源单元内容存入另外一个寄存器(C暂存器), 待目标地址计算完成并存入MAR后, 再将C中的内容通过MDR存入目标单元。

二、部件与与数据通路



6) I/O→R: I/O→DB→MDR→ ALU→ 内 —打入→ R0

7) I/O ⟷ M

外设与存储器之间数据传送可以有两种方式:

① 由CPU程序控制(相当于执行数据传送指令)

▶如果计算机系统没有直接 I/O ↔ M类指令, 采用两步进行(即用两条指令):

$I/O \rightarrow R_i$, $R_i \rightarrow M$; 完成I/O→M

$M \rightarrow R_i$, $R_i \rightarrow I/O$; 完成M→I/O

I/O→M的传送过程:

I/O ⟶ DB ⟶ MDR ⟶ B ⟶ ALU ⟶ 内总线 ⟶ R_i

R_i ⟶ B ⟶ ALU ⟶ 内总线 ⟶ MDR ⟶ DB ⟶ M

- 如果计算机系统提供直接I/O↔M类指令，则由指令指明存储单元地址和外设接口中的端口地址，以MDR为缓冲器：

M ⟷ DB ⟷ MDR

MDR ⟷ DB ⟷ I/O

② DMA方式

I/O ⟷ DB ⟷ M

CPU完成DMA控制器初始化后，数据传送过程无需CPU控制(数据流不经过CPU)。

二、部件与与数据通路

至此, 完成了CPU设计过程的前两个步骤的工作:

1. 拟定指令系统
   - 指令格式
   - 寻址方式
   - 指令类型

2. 总体结构和信息通路
   - 总体结构
   - 寄存器及所需命令
   - 不同信息传输通路

这两步工作与采用组合逻辑设计方法或者采用微程序设计方法无关。

后面几步的工作则与设计方法有关系。

二、部件与与数据通路

思考题：

ADD　(R2), X(PC)

该指令实现按既定寻址方式分别确定加数和被加数，并将结果保存于目的地址指定的单元，请分析其涉及到的信息传送通路。

（1）指令信息的传送路径?
（2）地址信息的传送路径?
（3）数据信息的传送路径?