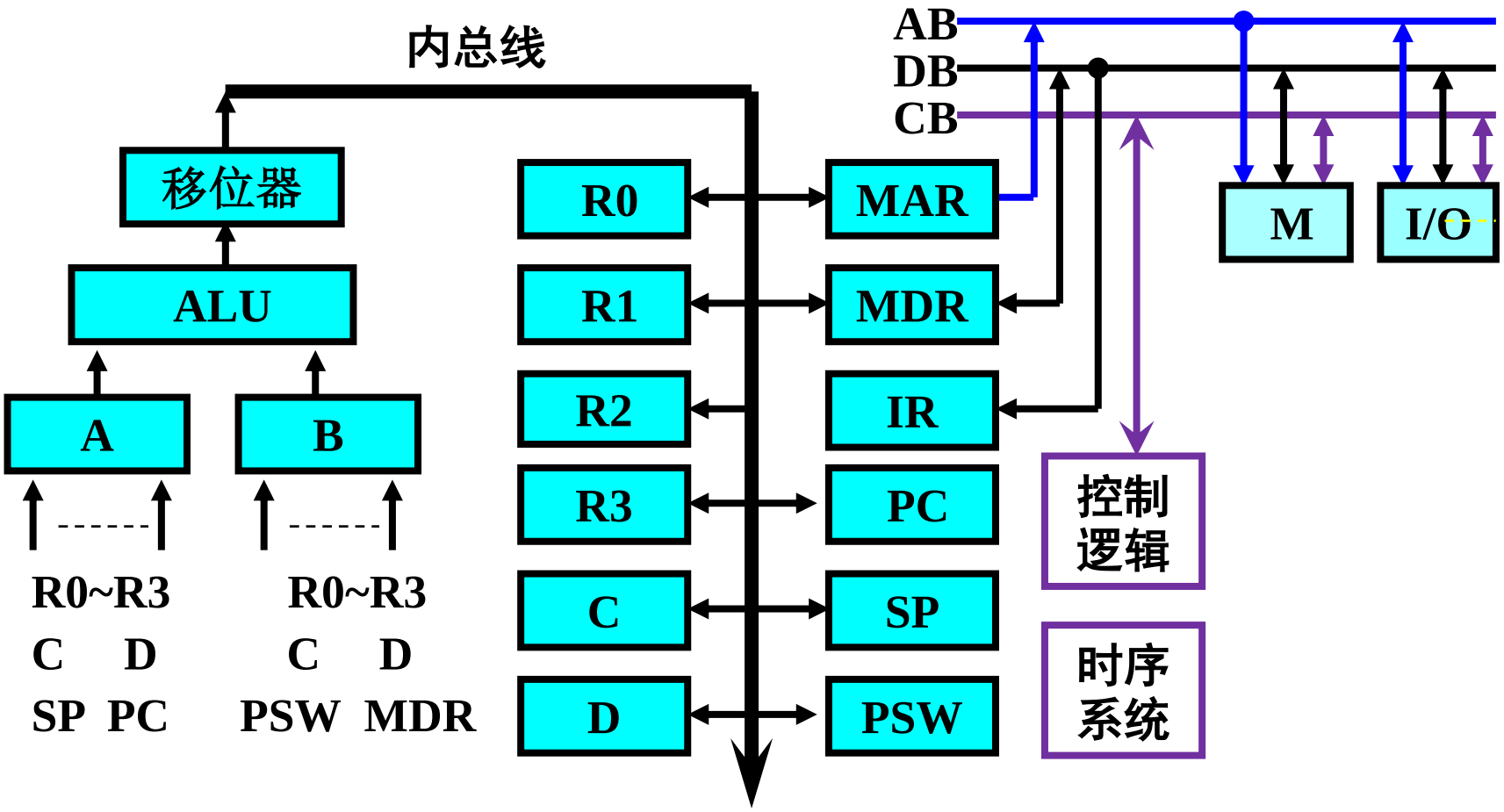
7.3 CPU模型


内总线
AB
DB
CB
移位器
ALU
A
B
R0~R3
C D
SP PC
R0~R3
C D
PSW MDR
R0
R1
R2
R3
C
D
MAR
MDR
IR
PC
SP
PSW
M
I/O
控制
逻辑
时序
系统

7.3.1 CPU设计步骤

（1）拟定指令系统

一台计算机的指令系统表明了这台机器所具有的硬件功能。

（2）确定总体结构，核心是数据通路结构

（3）确定控制信号的产生方式并安排时序

（4）拟定指令流程和微命令序列

（5）形成控制逻辑

7.3.2 模型机的指令系统

1. 指令格式

定长指令格式：16位，占据一个存储单元；

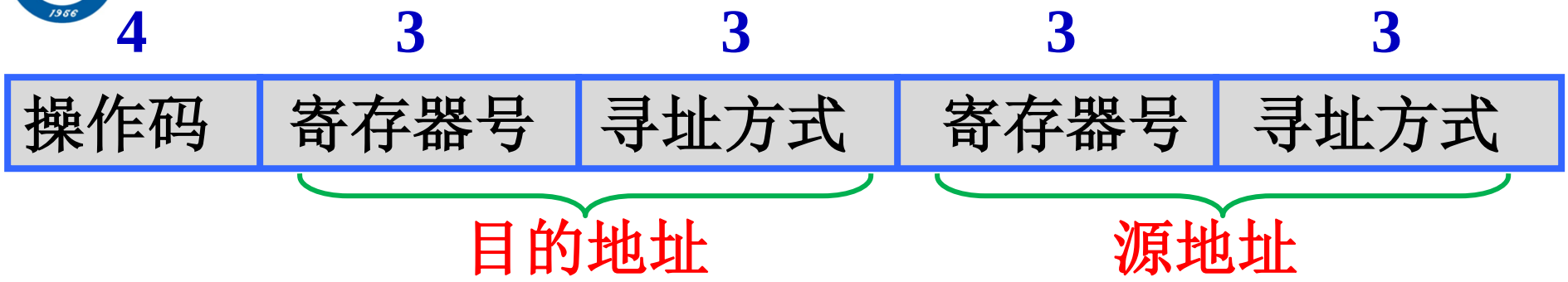
采用寄存器型寻址，即指令中给出寄存器号；

所有寄存器都是16位。

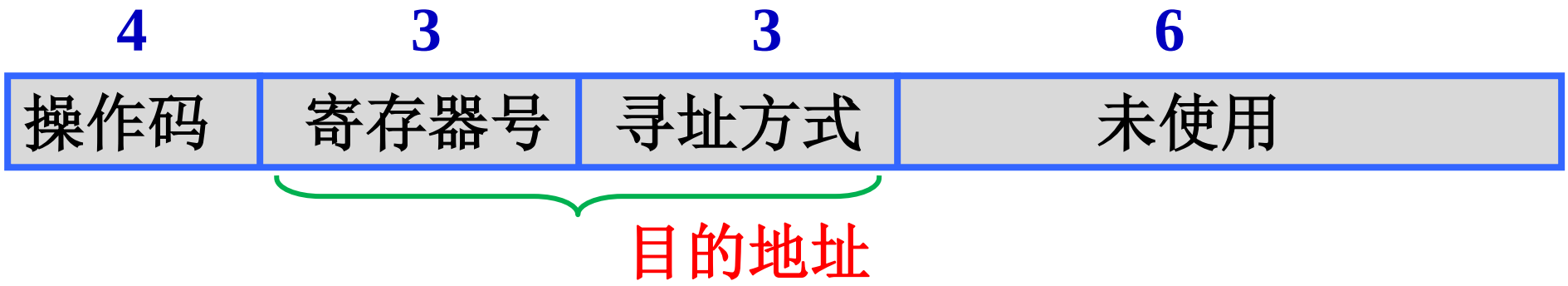
指令字长、存储字长、机器字长

指令类型
- 双地址类指令格式
- 单地址类指令格式
- 转移指令格式

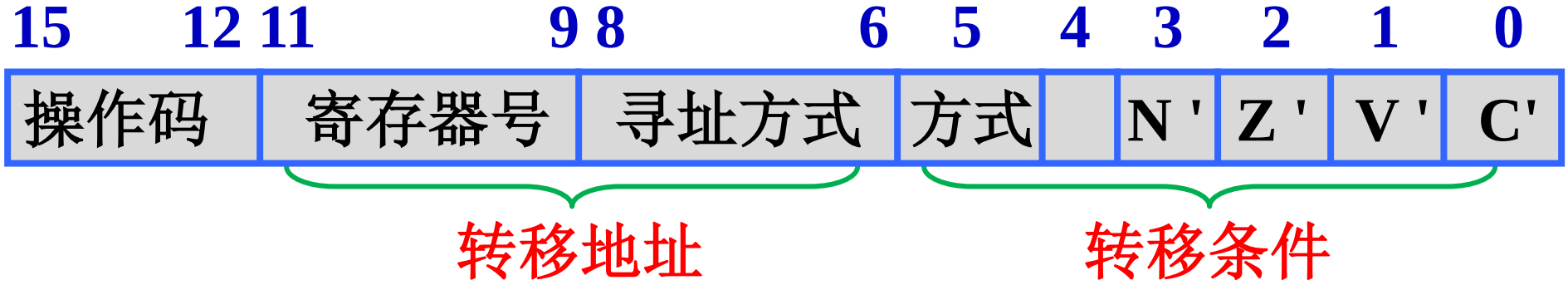
双地址类指令格式：



单地址类指令格式：

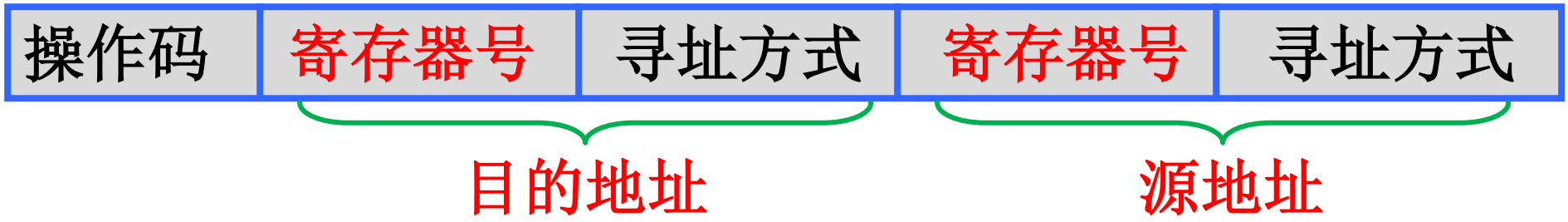


转移类指令格式：



2.寻址方式

特点：指令中直接给出寄存器编号



CPU可编程访问的寄存器（3位编号）

通用寄存器：

R_0(000)、R_1(001)、 R_2(010)、 R_3(011)

堆栈指针：SP（100）

程序状态字：PSW（101）

指令计数器：PC（111）

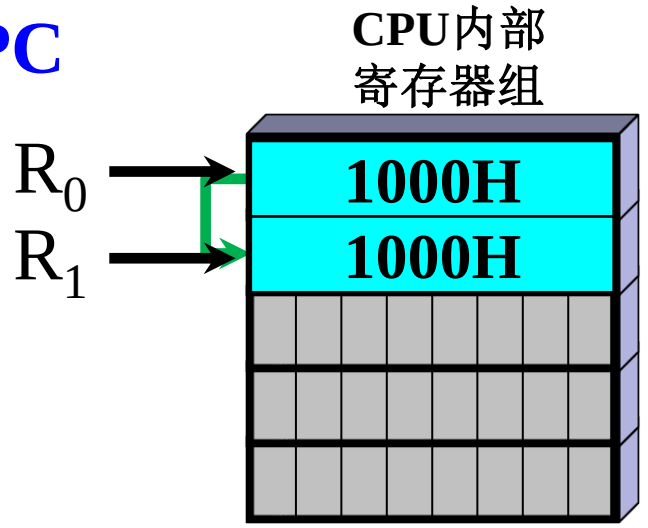
（1）0型：寄存器直接寻址

寻址方式	编码	助记符	定义
寄存器直接寻址	000	R	寄存器号为有效地址，寄存器的内容为操作数

可指定的寄存器为：

R_0、R_1、R_2、R_3 、SP、PSW、PC

例：MOV R_1，R_0



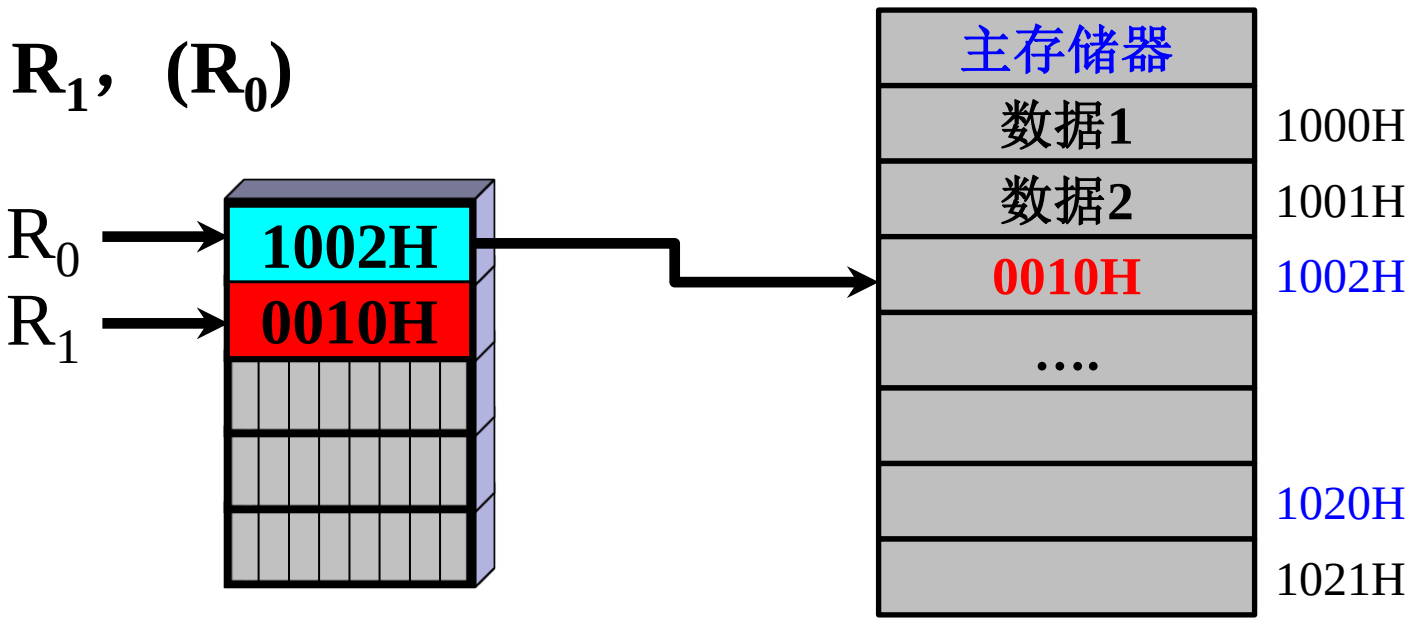
假设操作码为0000，
则指令的二进制代码？

（2）1型：寄存器间接寻址

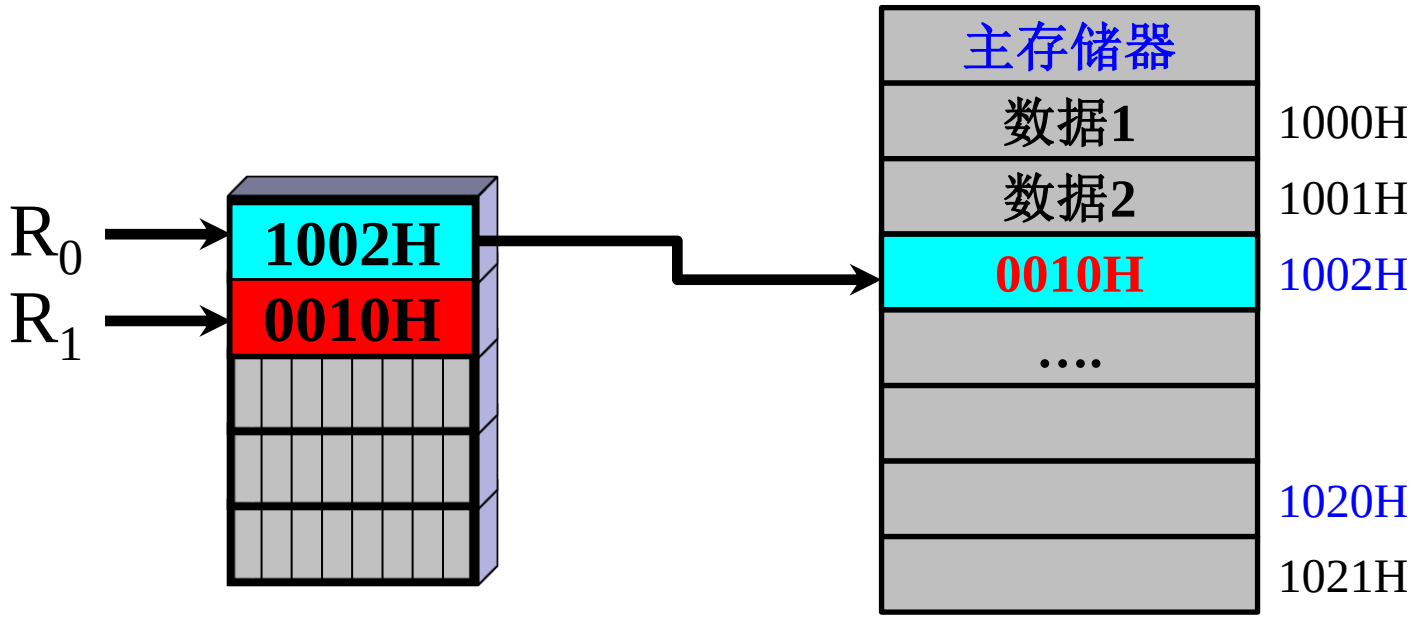
寻址方式	编码	助记符	定义
寄存器间址	001	(R)	寄存器的内容为有效地址

可指定的寄存器为：R_0、R_1、R_2、R_3

例：MOV R_1，(R_0)



例：MOV （R_0），R_1

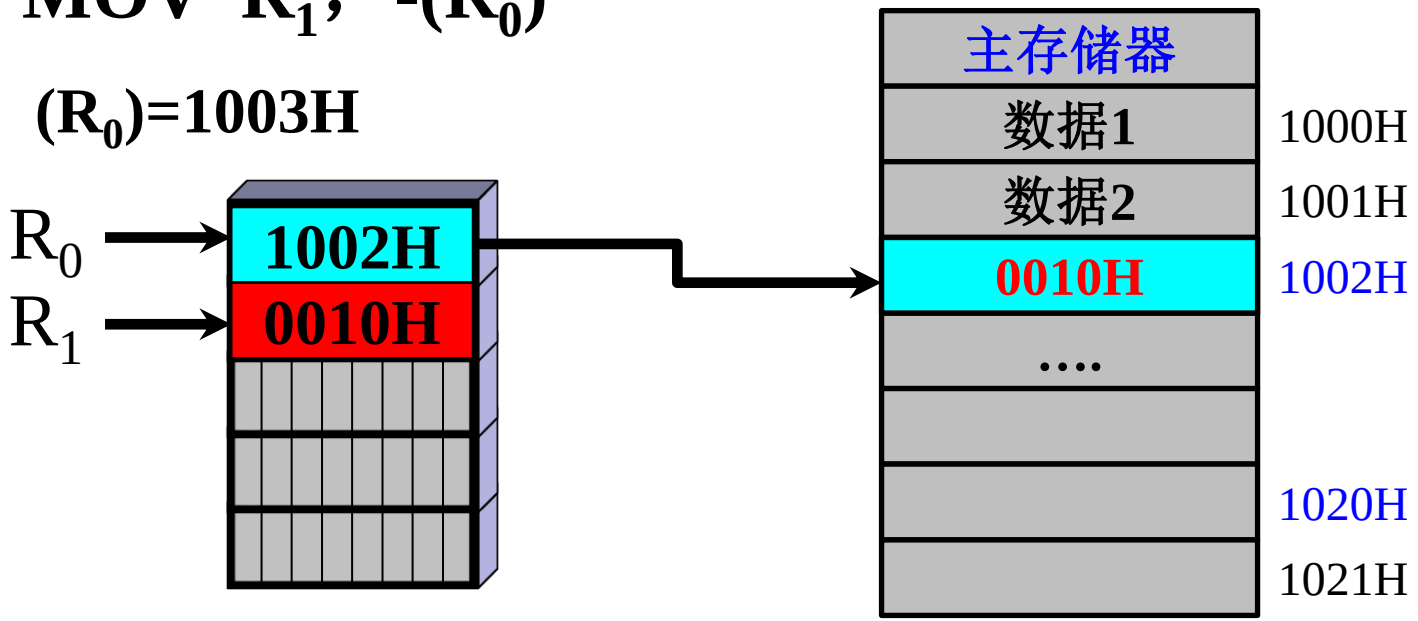

主存储器
数据1
数据2
0010H
....
1000H
1001H
1002H
1020H
1021H
R_0
R_1
1002H
0010H

（3）2型：自减型寄存器间址

寻址方式	编码	助记符	定义
自减型寄存器间址	010	-(R) -(SP)	寄存器的内容减1后作为有效地址

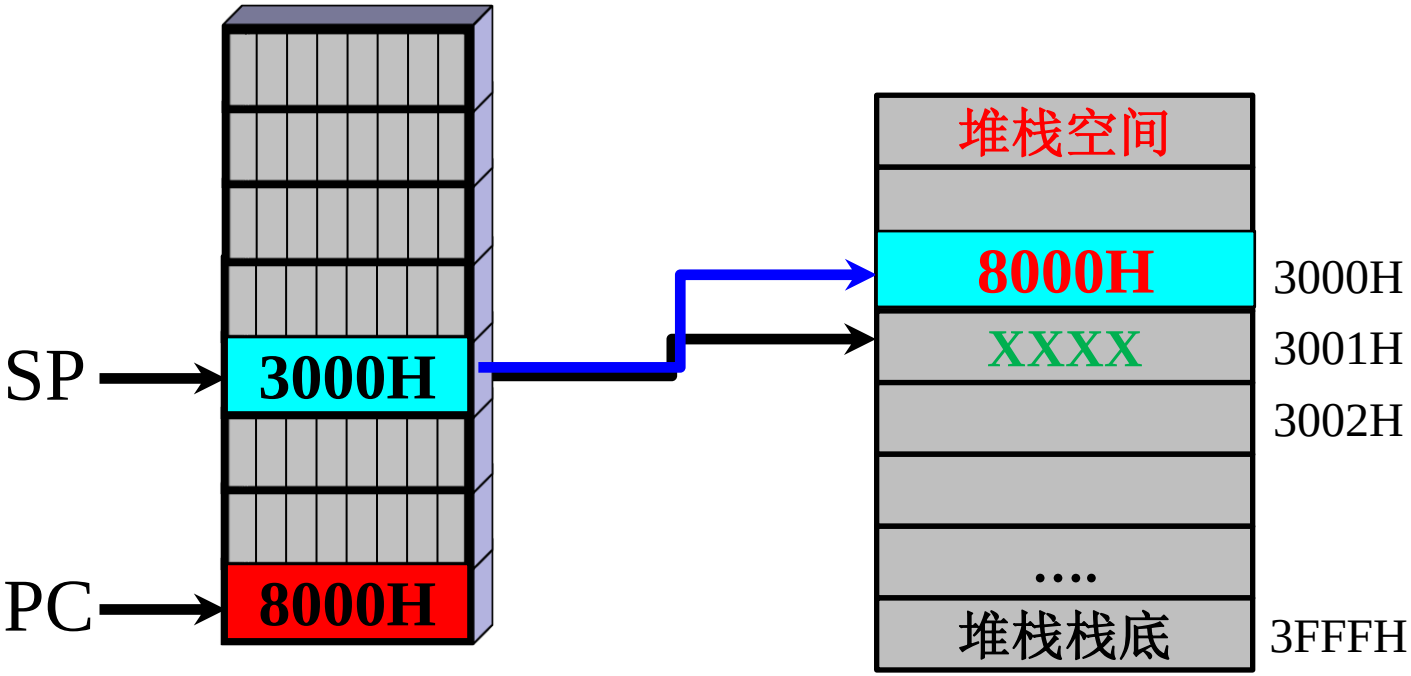
可指定的寄存器：R_0、R_1、R_2、R_3、SP

例：MOV R_1，-(R_0)



例：MOV -(SP) , PC ⟺ PUSH PC

(SP)=3001H



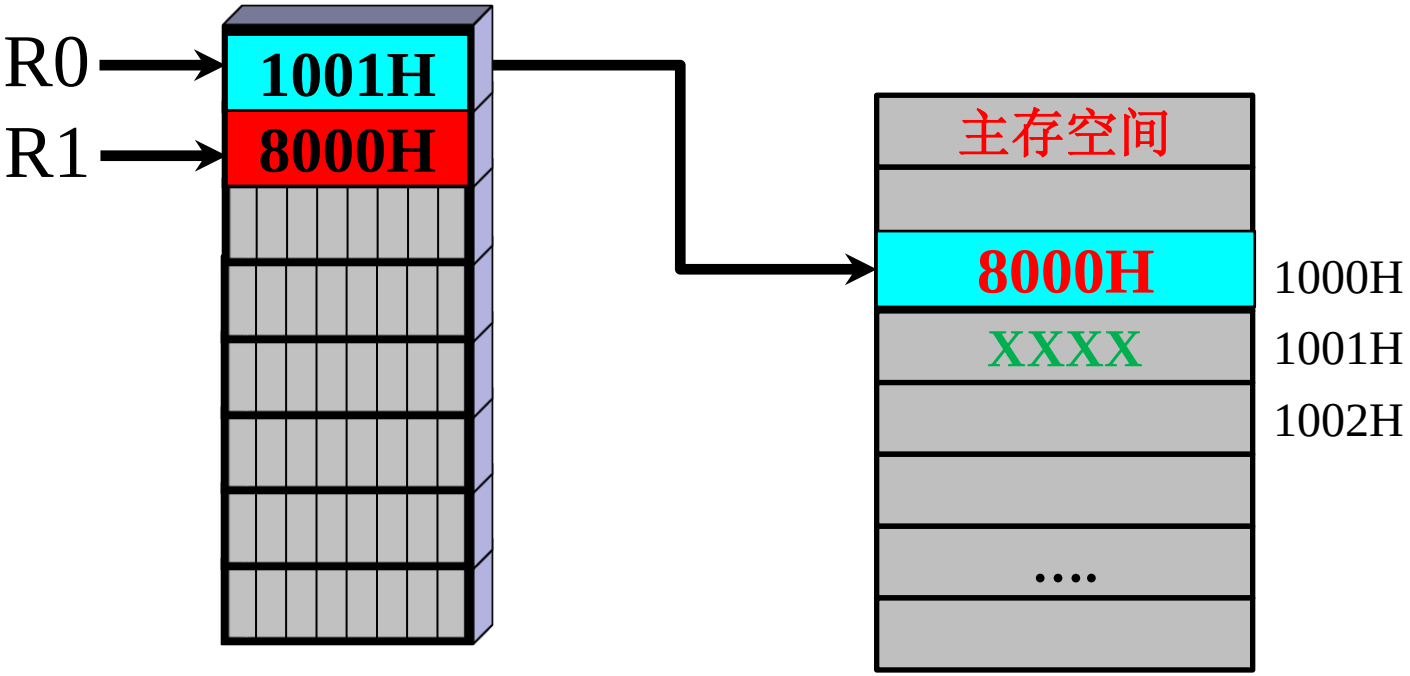
（4）3型：自增型寄存器间址

寻址方式	编码	助记符	定义
立即/自增型寄存器间址	011	(R)+ (SP)+ (PC)+	寄存器的内容为有效地址，访问该地址单元后，寄存器的内容加1。

可指定的寄存器：　R_0、R_1、R_2、R_3、SP、PC

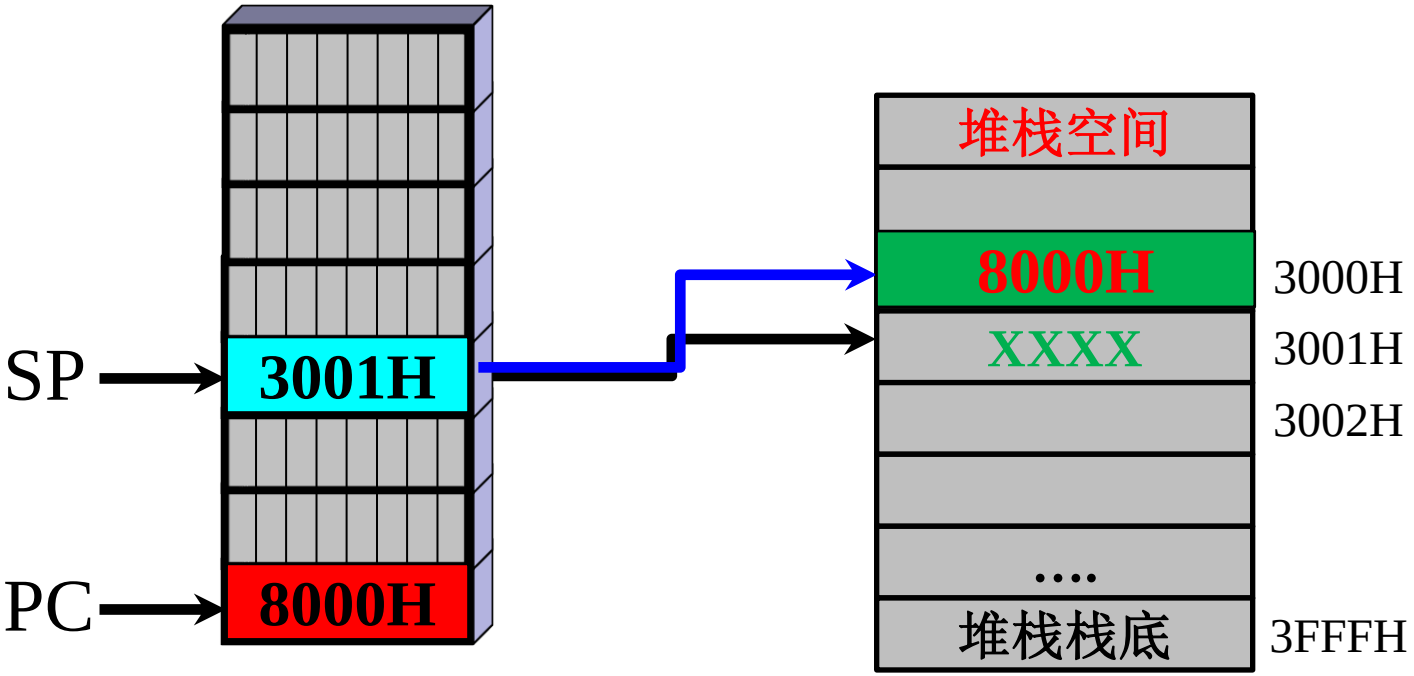
例：MOV R_1，(R_0)+

(R_0)=1000H



例：MOV PC，(SP)+ ⟺ POP PC

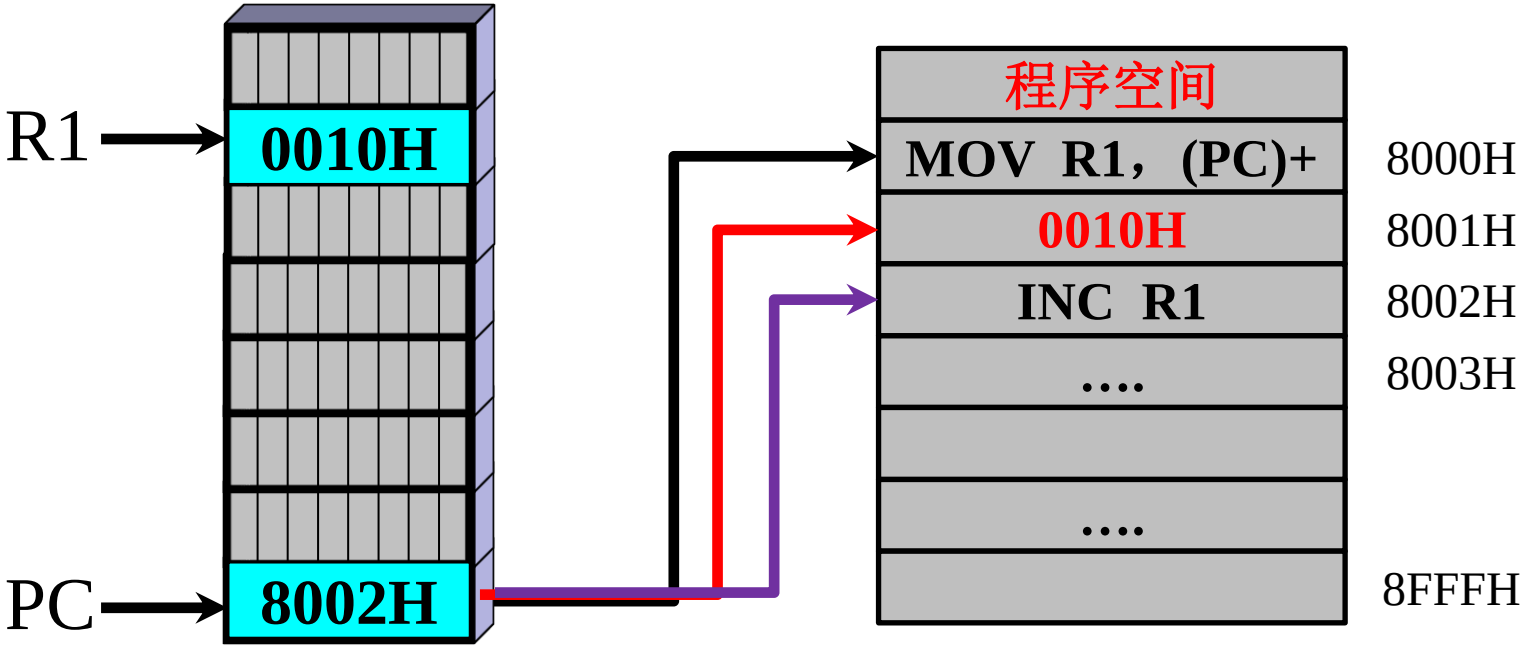
(SP)=3000H



例：MOV R1，(PC)+ ⟺ MOV R1, 10H

(PC)=8000H　　立即寻址

PC的内容随着取指、指令地运行会动态变化



（5）4型：自增型双重间址

寻址方式	编码	助记符	定义
直接/自增型双重间址	100	@(R)+	寄存器的内容为间接地址，根据该地址访存取得操作数的地址，再次访存读写操作数，然后寄存器的内容加1。
		@(PC)+	

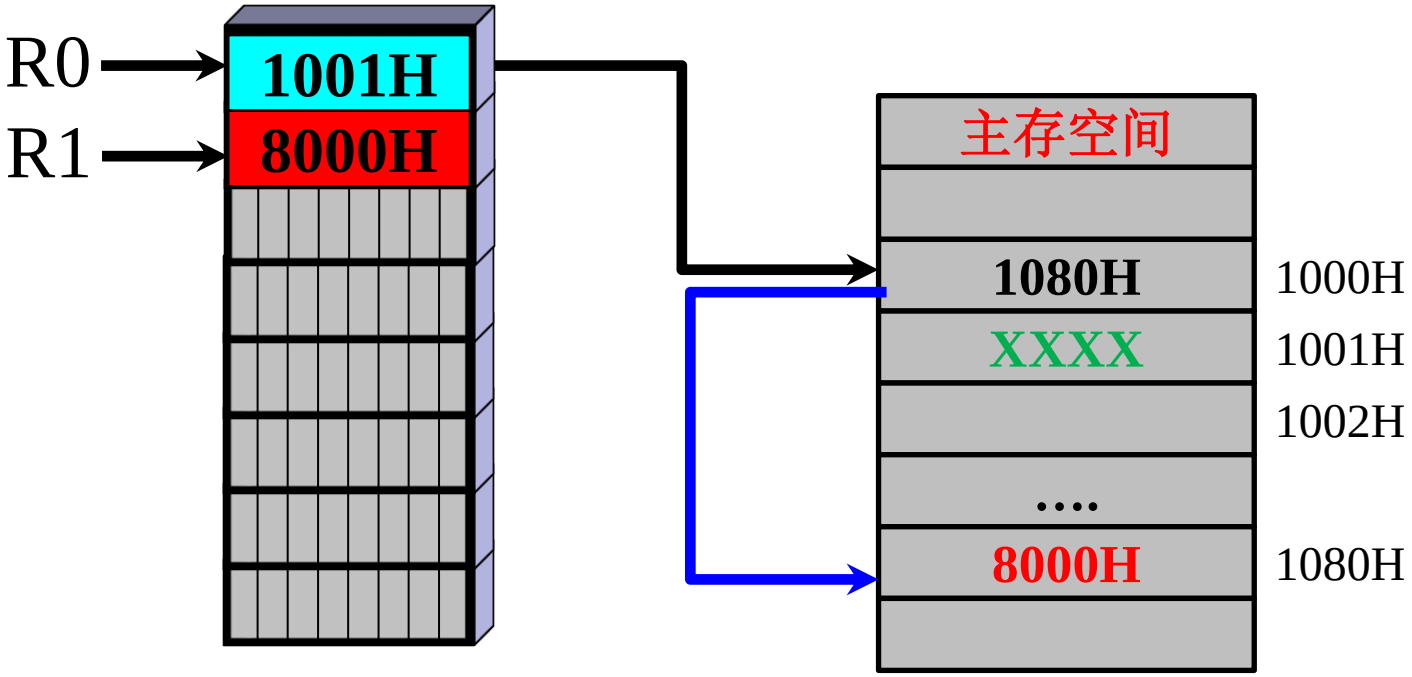
可指定的寄存器：R_0、R_1、R_2、R_3、PC

操作数地址：EA=((R))

操作数：(EA)=(((R)))

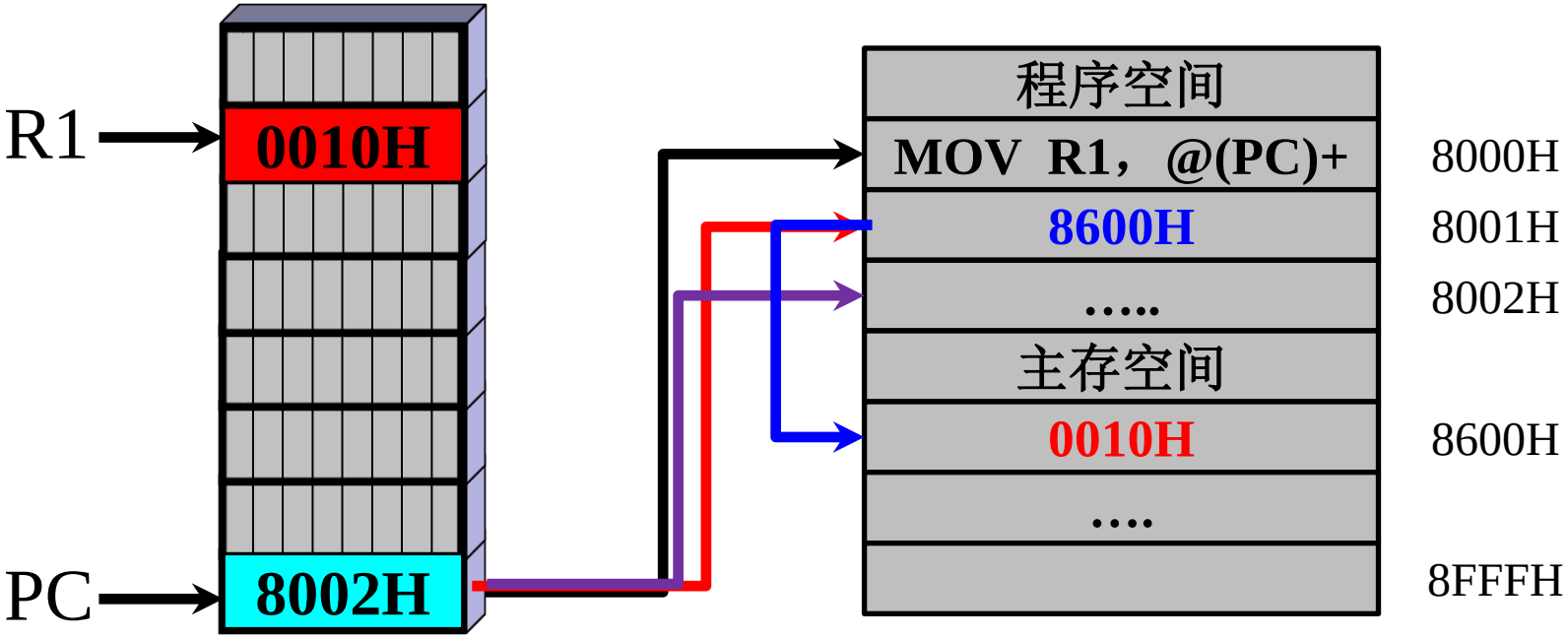
例：MOV R_1，@(R_0)+

(R_0)=1000H



例：MOV R_1，@(PC)+ ⟺ MOV R1, [8600H]

(PC)=8000H　　直接寻址



（6）5型：变址寻址/相对寻址

寻址方式	编码	助记符	定义
变址/相对寻址	101	X(R) X(PC)	寄存器的内容与形式地址之和为有效地址。

可指定的寄存器：R_0、R_1、R_2、R_3、PC

形式地址存放在紧跟指令的存储单元中。

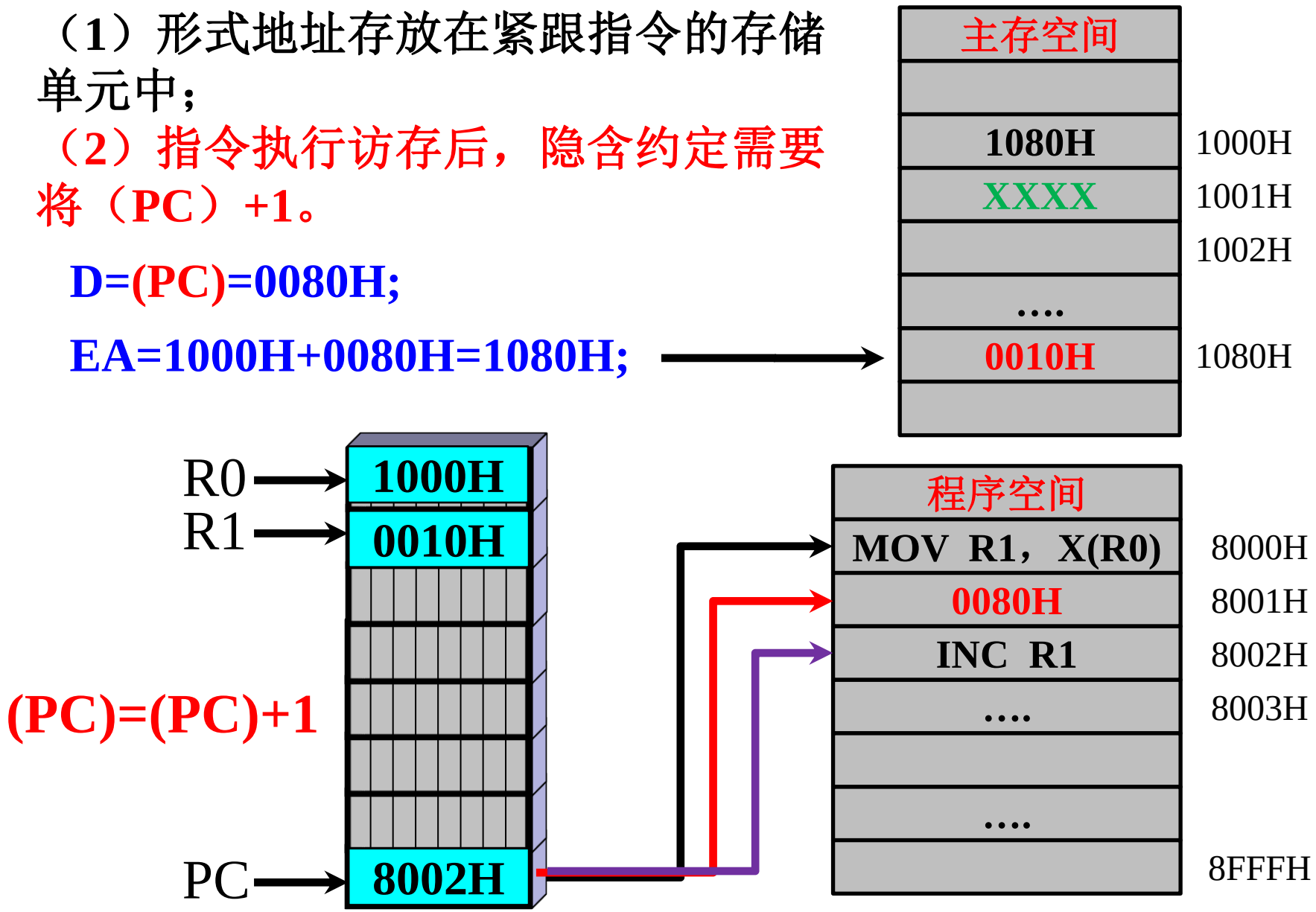
例：MOV R_1，X(R0)　　(R_0)=1000H　(PC)=8000H

（1）形式地址存放在紧跟指令的存储单元中；

（2）指令执行访存后，隐含约定需要将（PC）+1。

D=(PC)=0080H;

EA=1000H+0080H=1080H;



（7）6型：跳步寻址

寻址方式	编码	助记符	定义
跳步寻址	110	SKP	执行再下一条指令

现行指令执行后，不是顺序执行下一条指令，而是执行再下一条指令。

3.指令操作类型

操作码	助记符	含义	操作码	助记符	含义
0000	MOV	传送	1000	INC	加1
0001	ADD	加	1001	DEC	减1
0010	SUB	减	1010	SL	左移
0011	AND	与	1011	SR	右移
0100	OR	或	1100	JMP	转移
0101	EOR	异或	1100	RST	返回
0110	COM	求反	1101	JSR	转子
0111	NEG	求补			

(1)传送类指令

MOV： R<->R，R<->M，M<->M

统一编址，隐式I/O指令

(2)双地址指令

ADD，SUB，AND，OR，EOR

(3)单地址指令

COM(反)，NEG(补)，INC，DEC

SL(左移)，SR(右移)

(4)程序控制类指令　JMP, RST, JSR

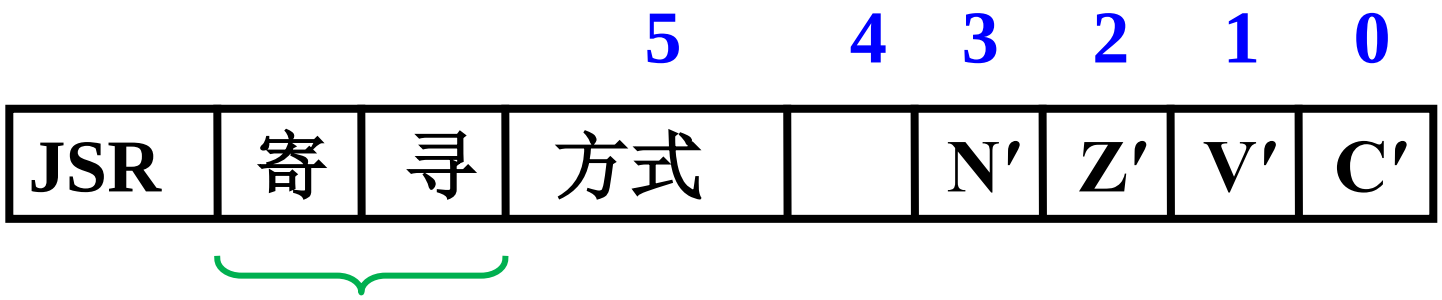
指明PSW的标志位

转移地址

根据有效地址获得的操作数作为转移地址，X(PC)特例

0	0	0	0	0	无条件转	
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	数为正转	(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)

条件满足，转向转移地址；否则顺序执行。



子程序入口地址

隐含约定：转子时返回地址压栈保存。

同时子程序的最后一条指令必须是返回指令。

7.3.3 模型机的组成和数据通路

1.部件设置

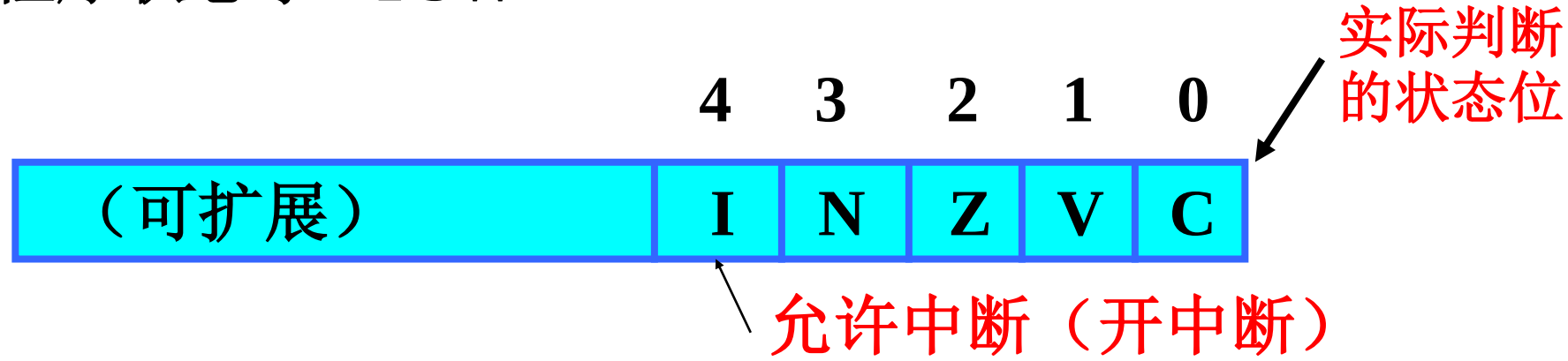
（1）寄存器

- 可编程寄存器（16位）

通用寄存器：R_0、R_1 、R_2、R_3

堆栈指针： SP　　指令计数器： PC

程序状态字：PSW

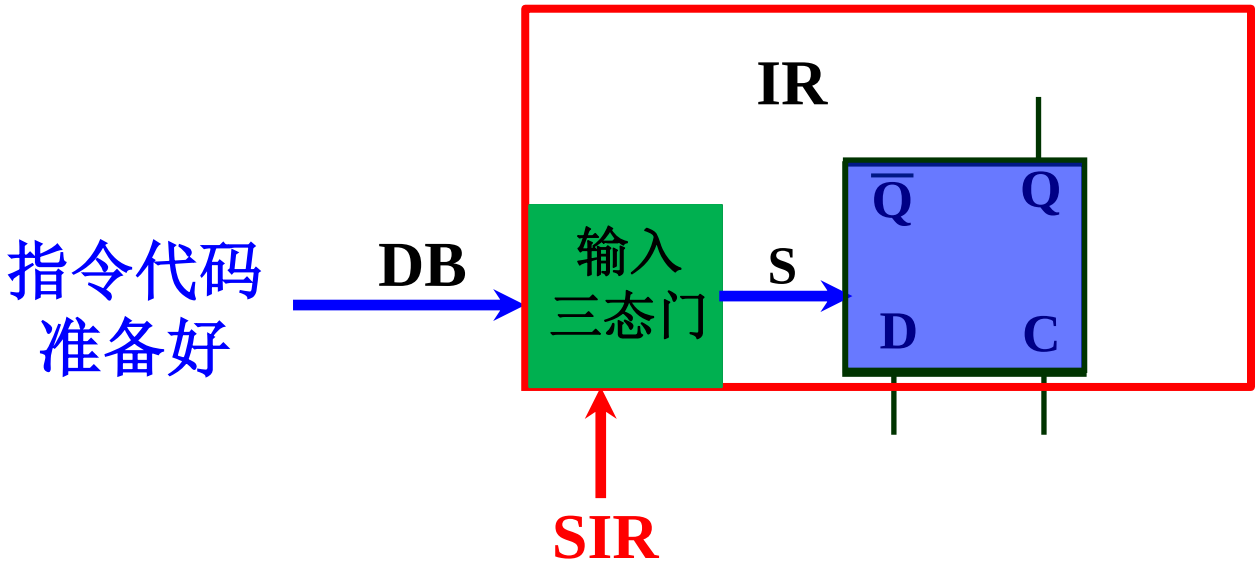


非编程寄存器（16位）

暂存器C :暂存来自主存的源地址或源数据。

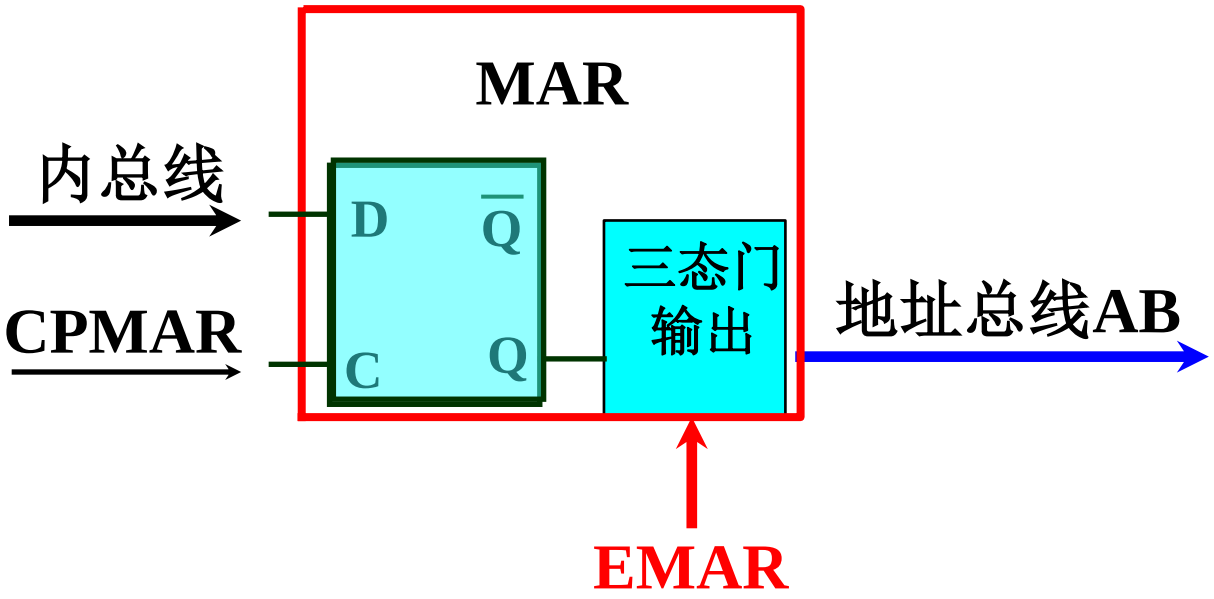
暂存器D :暂存来自主存的目的地址或目的数。

指令寄存器IR :存放现行指令。



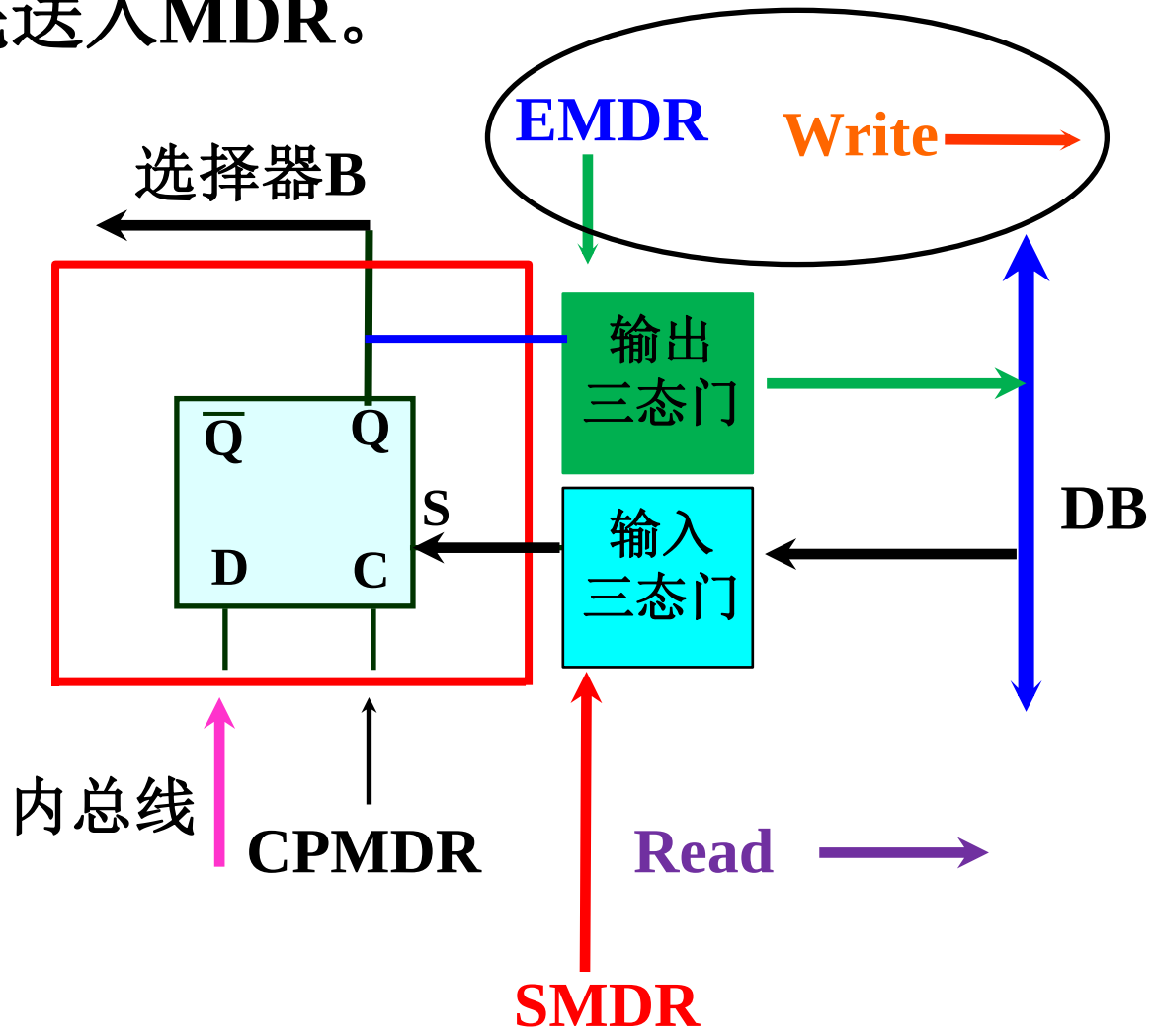
地址寄存器MAR

CPU访问主存(IO)的地址由MAR提供



数据寄存器MDR

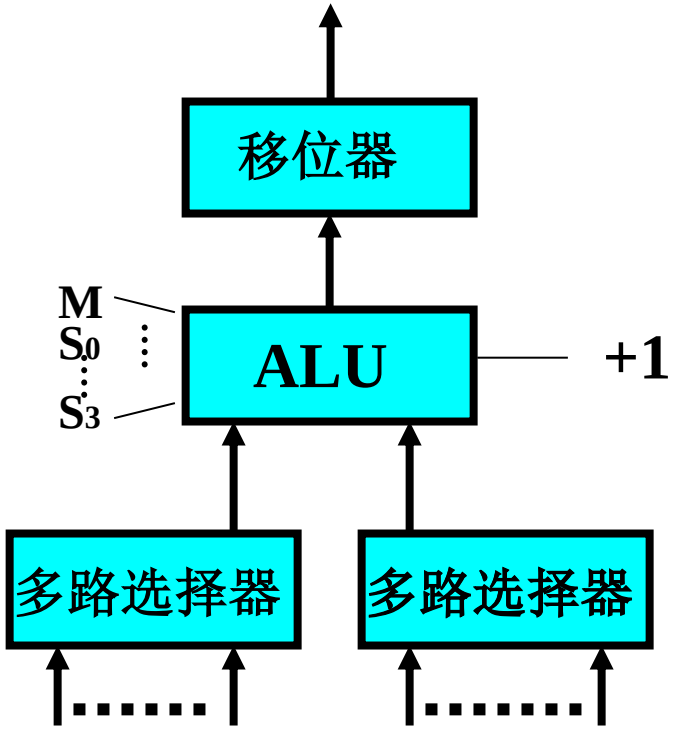
CPU存入主存(IO)的内容先放入MDR，读取主存(IO)的内容也先送入MDR。



(2) 运算部件设置 （16位）

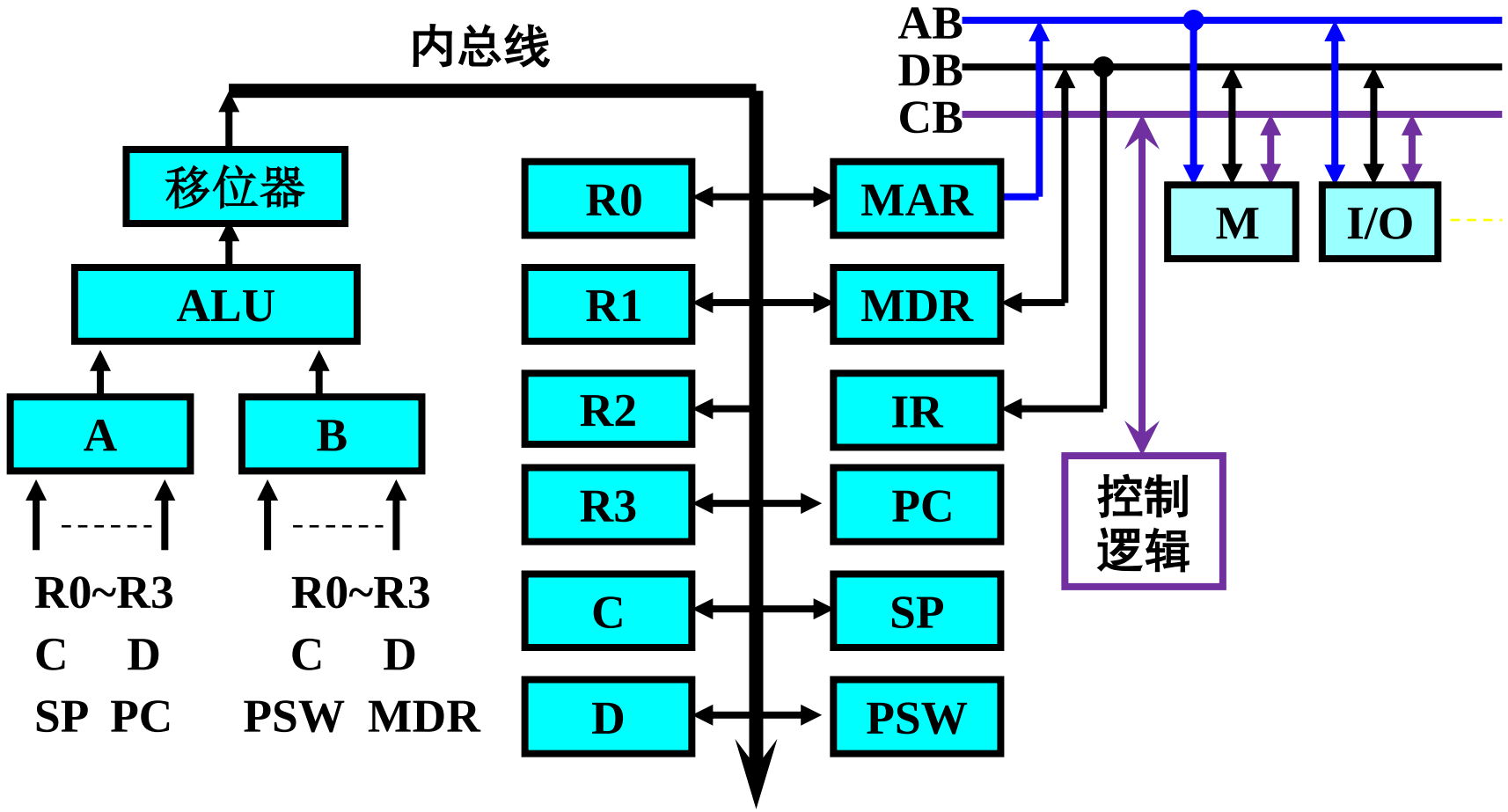
ALU { SN74181 4片
SN74182 1片

选择器A
选择器B
> 选择数据来源

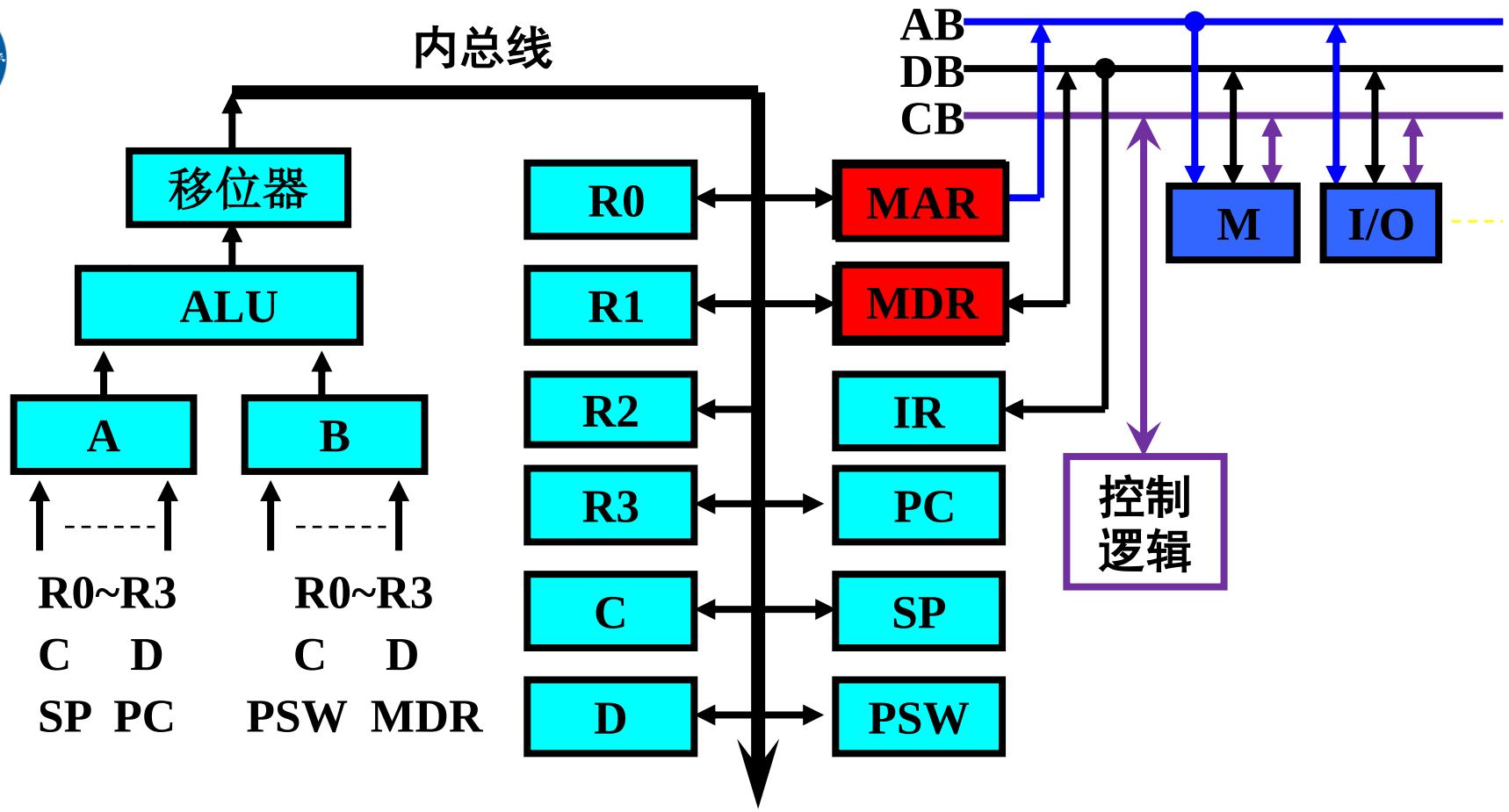


移位器 :实现直传、左移、右移、字节交换

2.总线与数据通路结构



ALU为内部数据传送通路的中心；寄存器采用分立结构；内总线采用单向数据总线；



与系统总线的连接通过MAR、MDR实现。

- MDR
  - 输入
    - 从内总线输入 (打入)
    - 从DB输入 (置入)
  - 输出
    - 输出至DB
    - 输出至ALU的B门

3.各类信息传送途径

（1）如何读取指令？

1）指令地址：PC→MAR

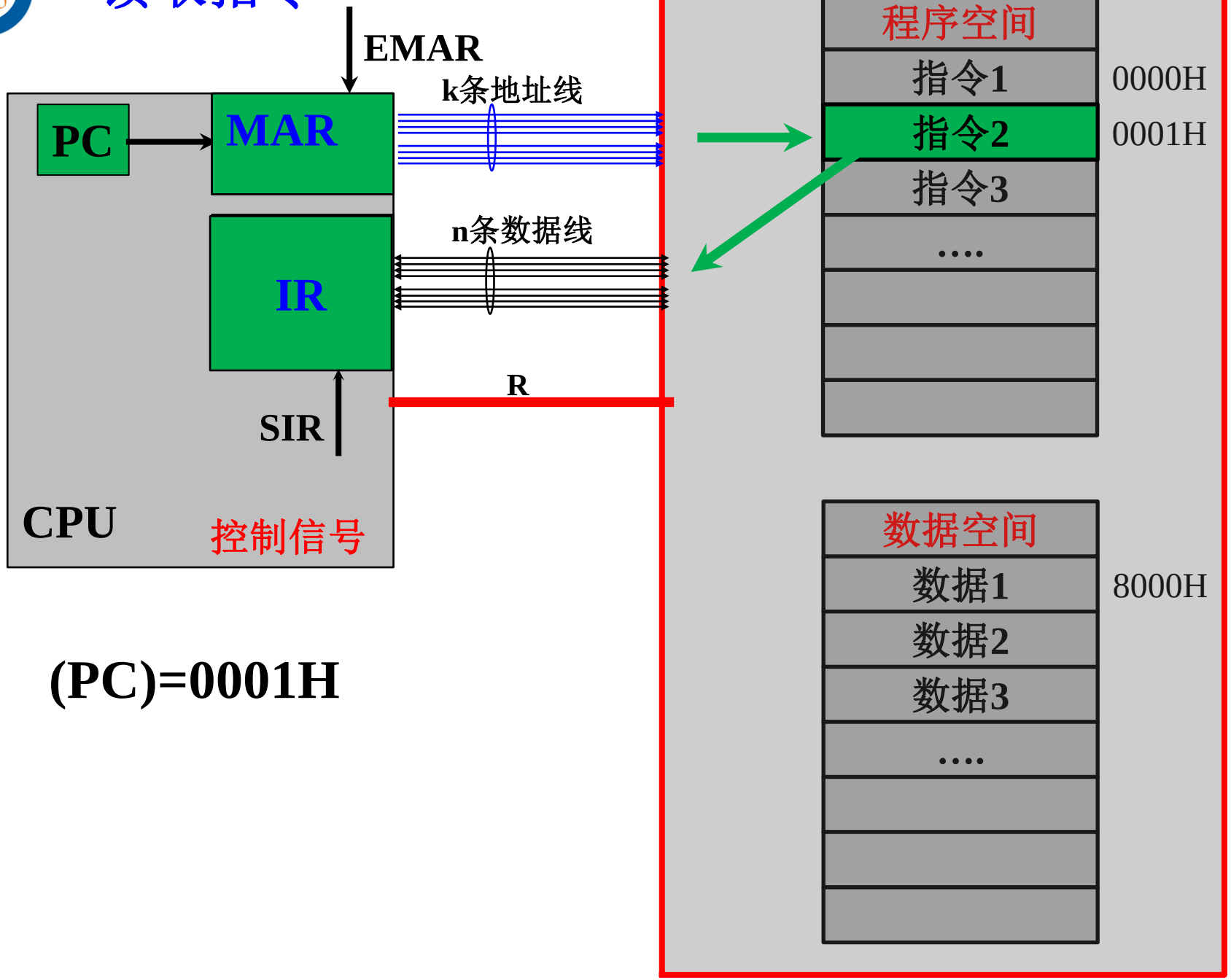
2）指令信息：M→IR

3）指令后继（顺序）地址：(PC)+1→PC

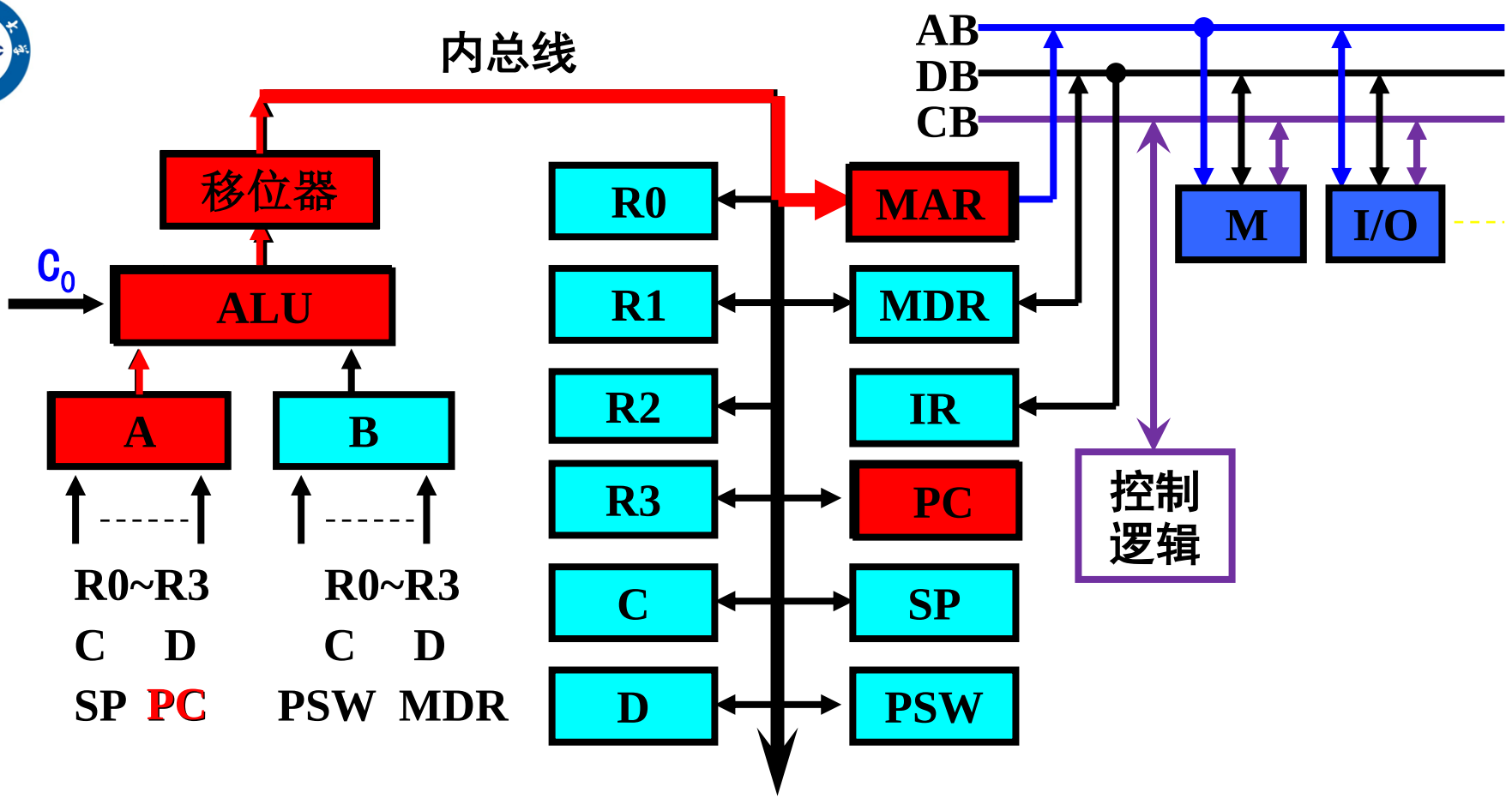
4）指令译码、取数、执行

5）若本条为转移指令，将根据寻址方式，
指令后继（转移）地址：→PC

读取指令

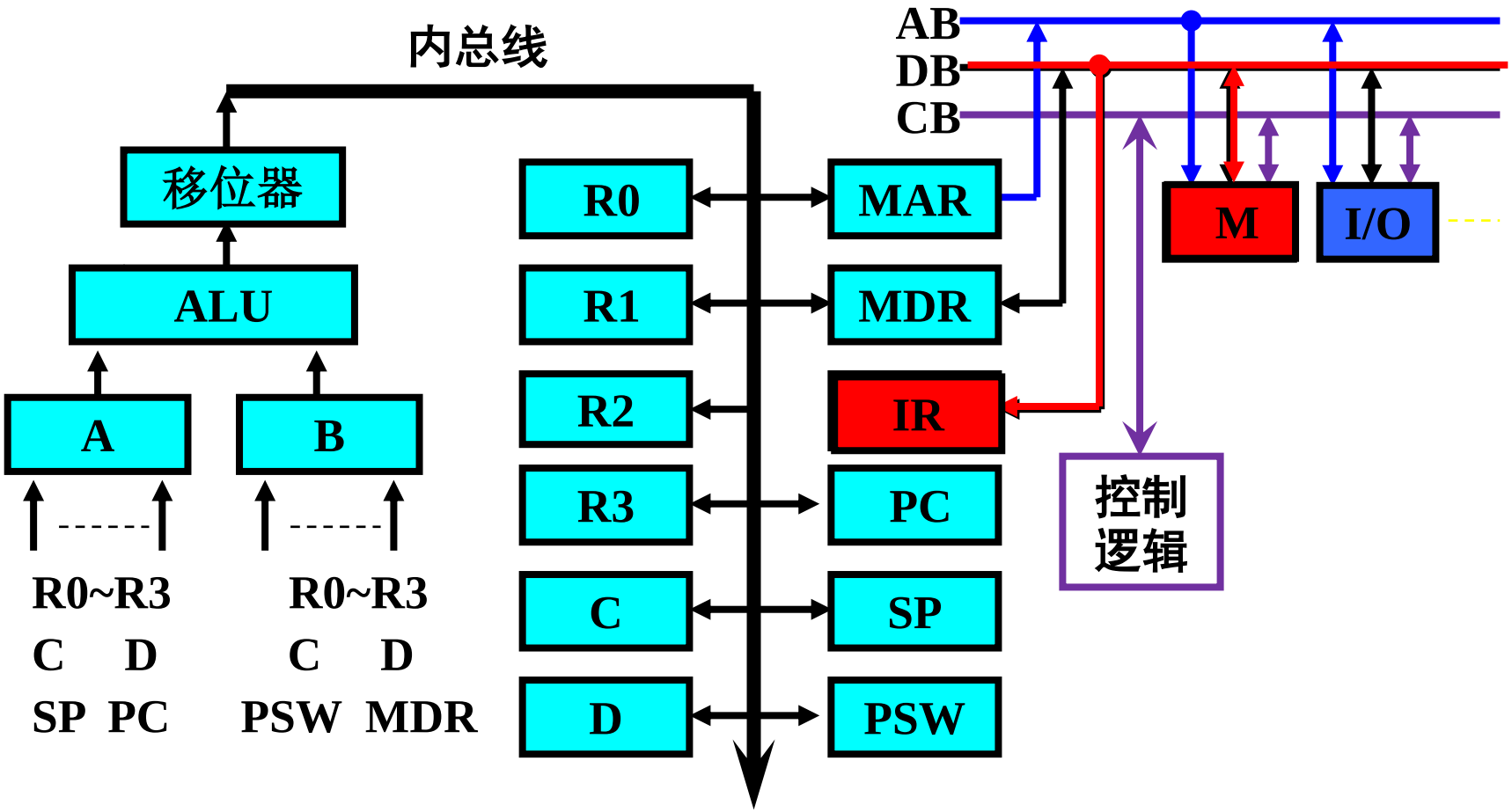


(PC)=0001H



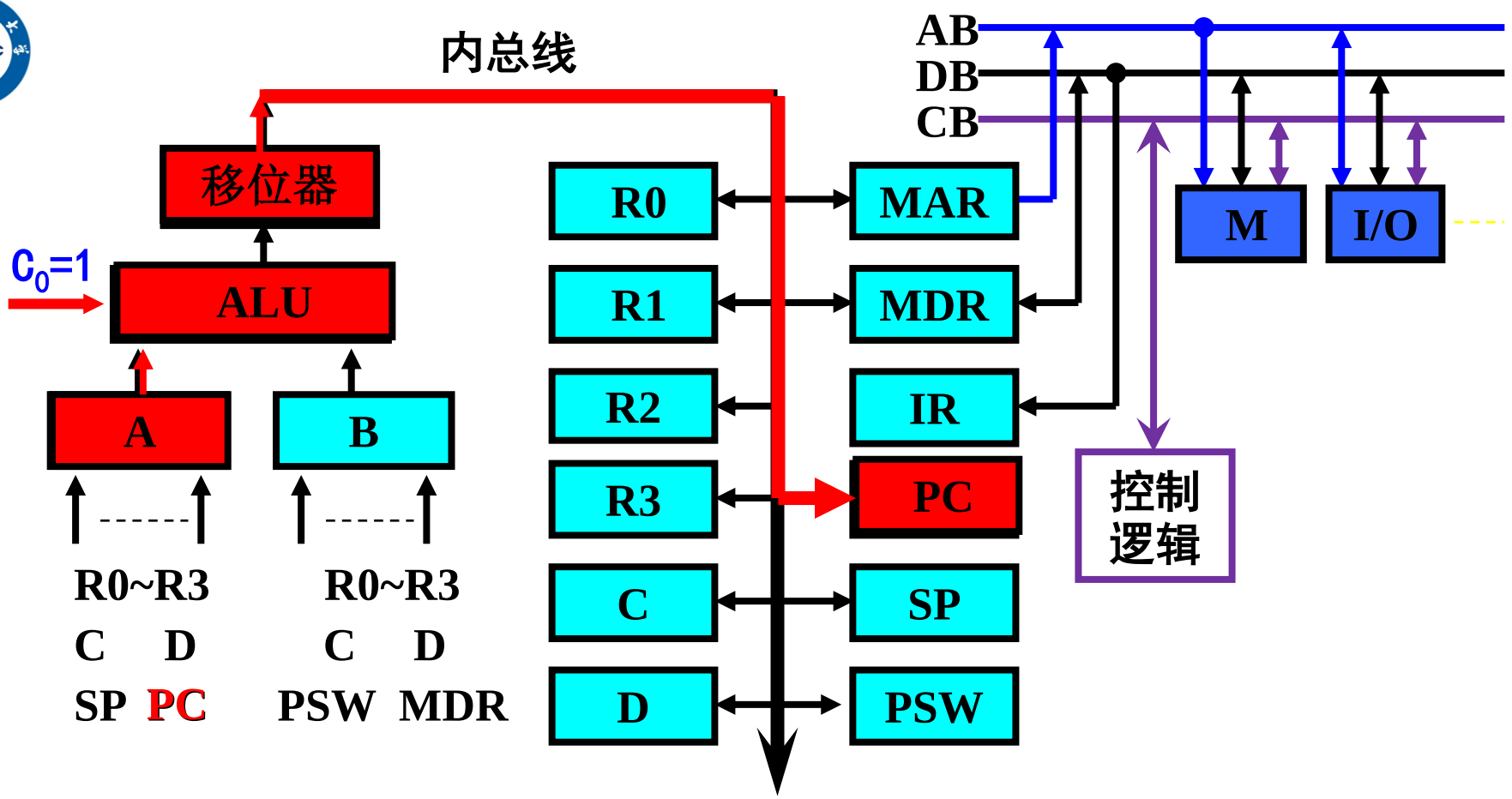
1）指令地址:PC→MAR

PC → A → ALU → 移 → 内 —打入→ MAR



2）读取指令信息

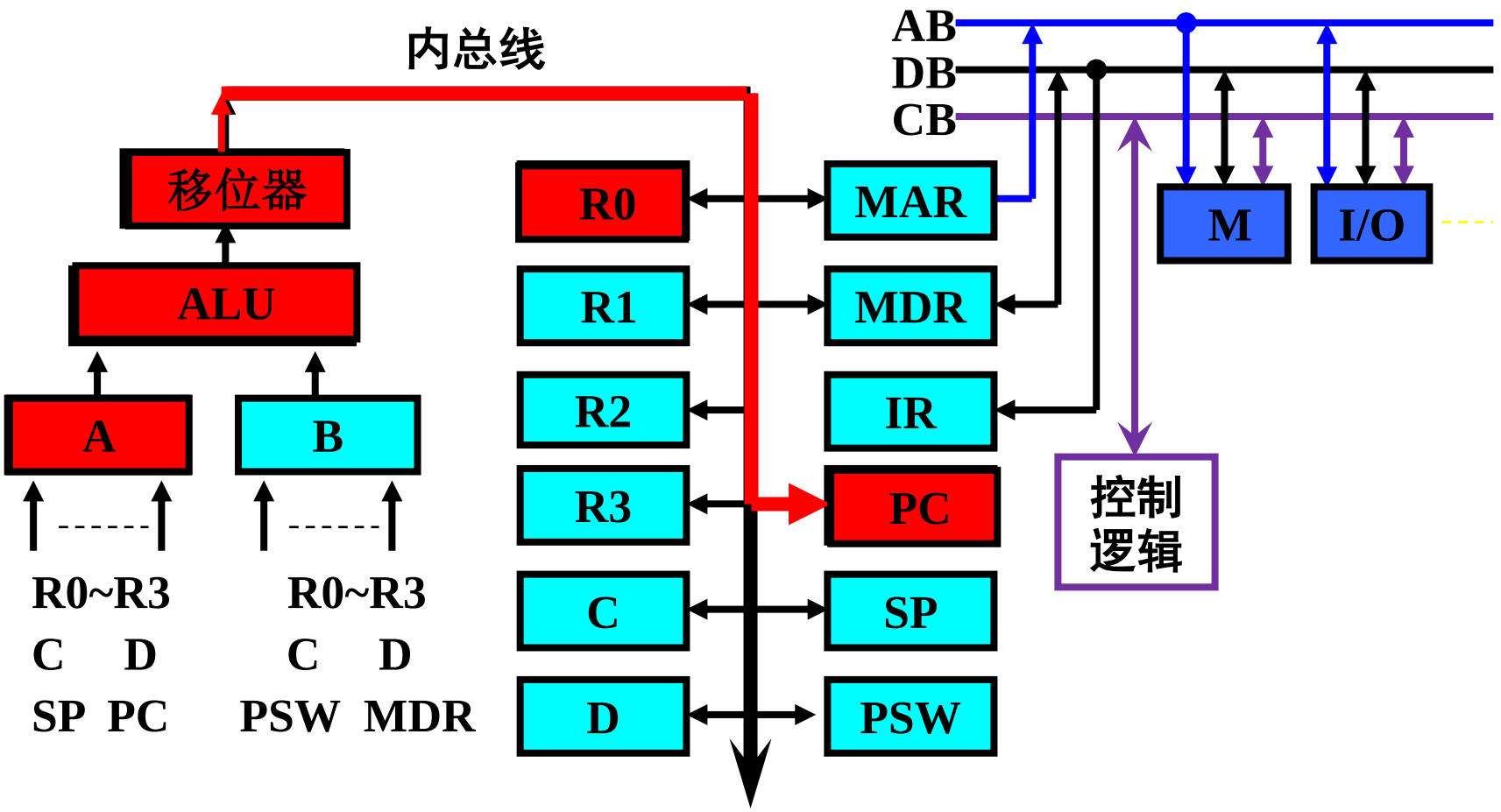
M ⟶ DB （置入） ⟶ IR



3）后继顺序地址：(PC)+1→PC

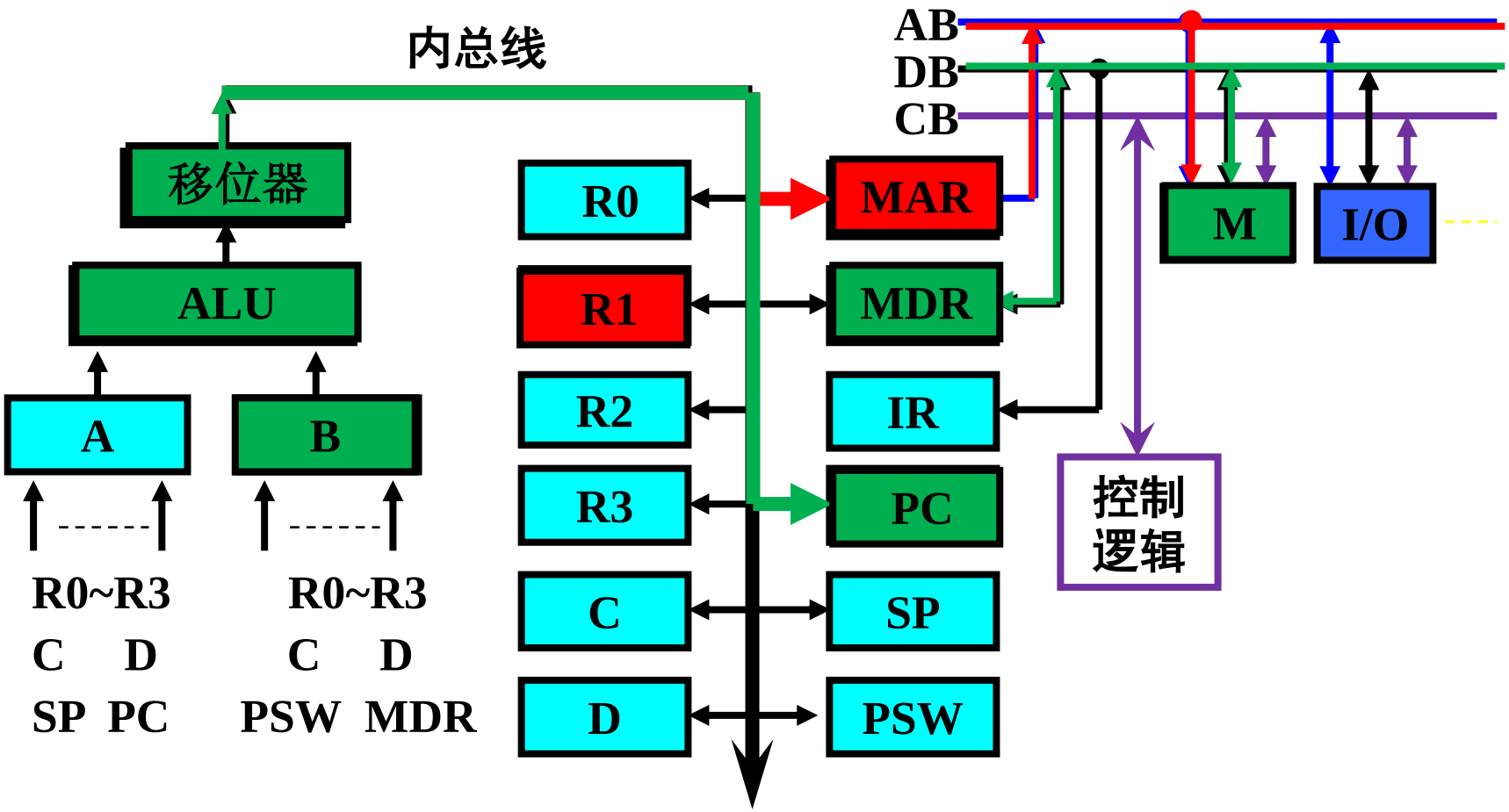
PC → A → ALU → 移 → 内 —打入→ PC

C_0 ↗



4）后继转移地址：→PC

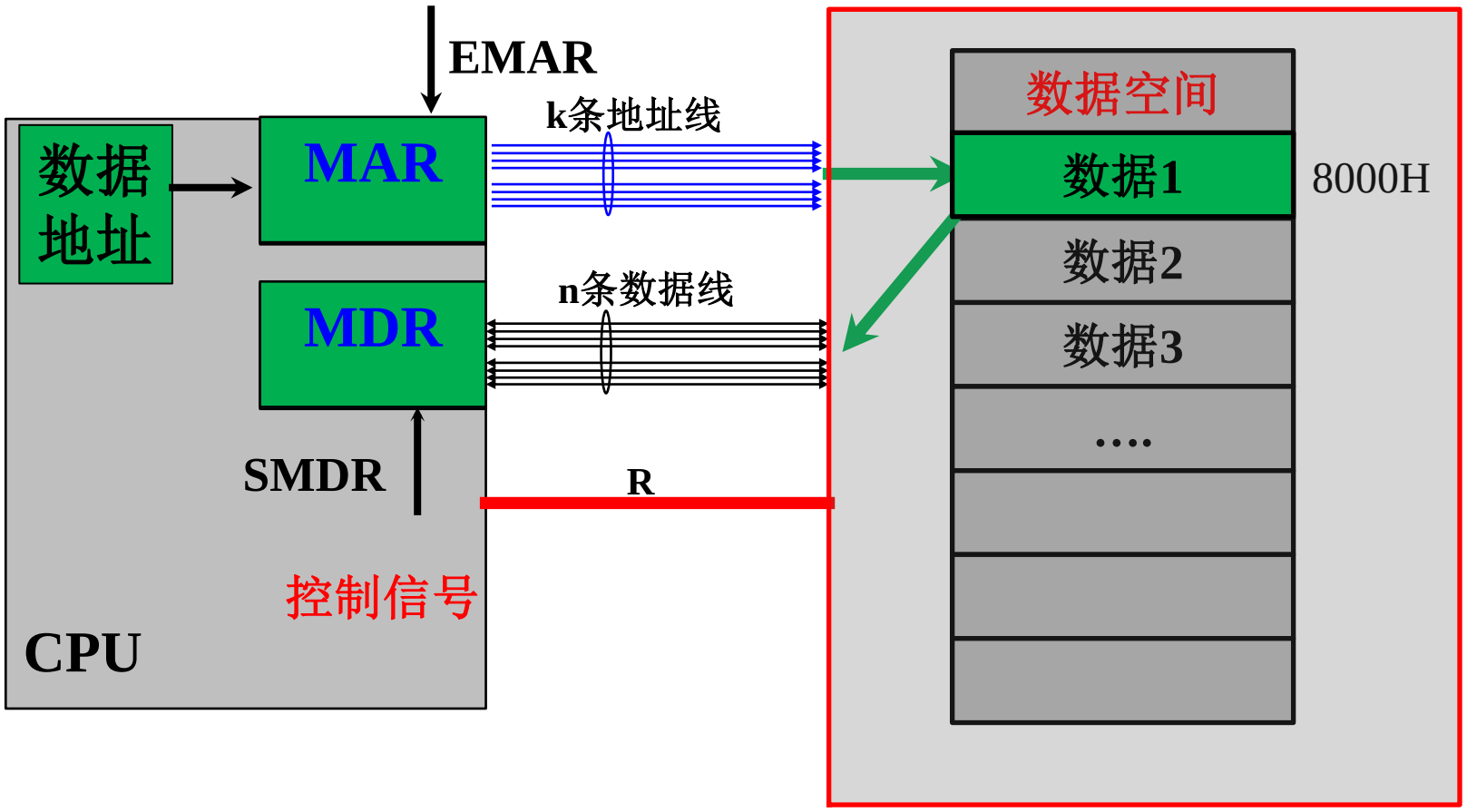
寄存器寻址：Ri → A/B → ALU → 移 → 内 → 打入 PC



寄存器间址：Ri → B → ALU → 移 → 内 —打入→ MAR →

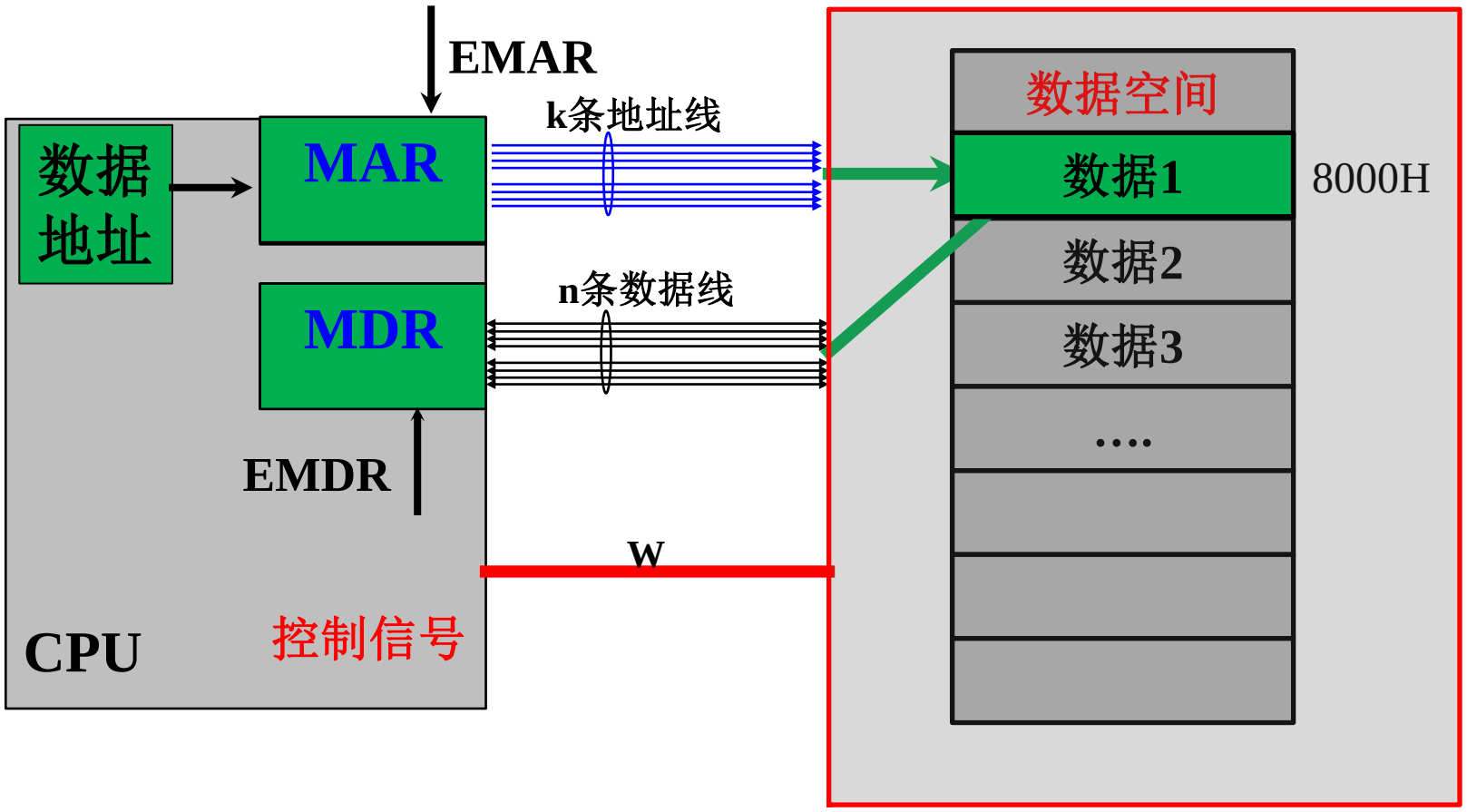
→ AB → M → DB —置入→ MDR → B → ALU → 移、内 → PC

（2）访存读取数据

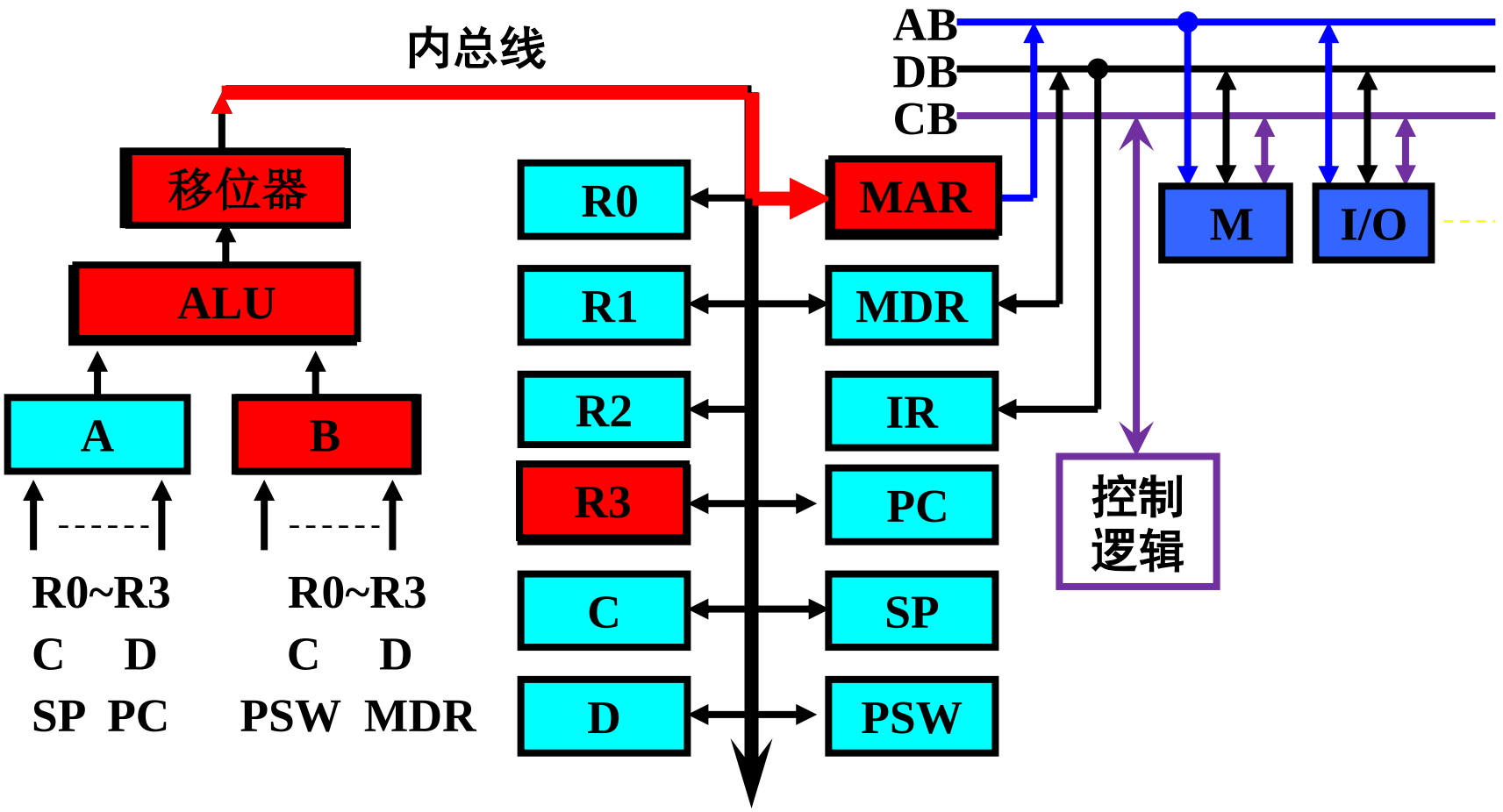


读取数据
数据地址=8000H

（3）存入数据

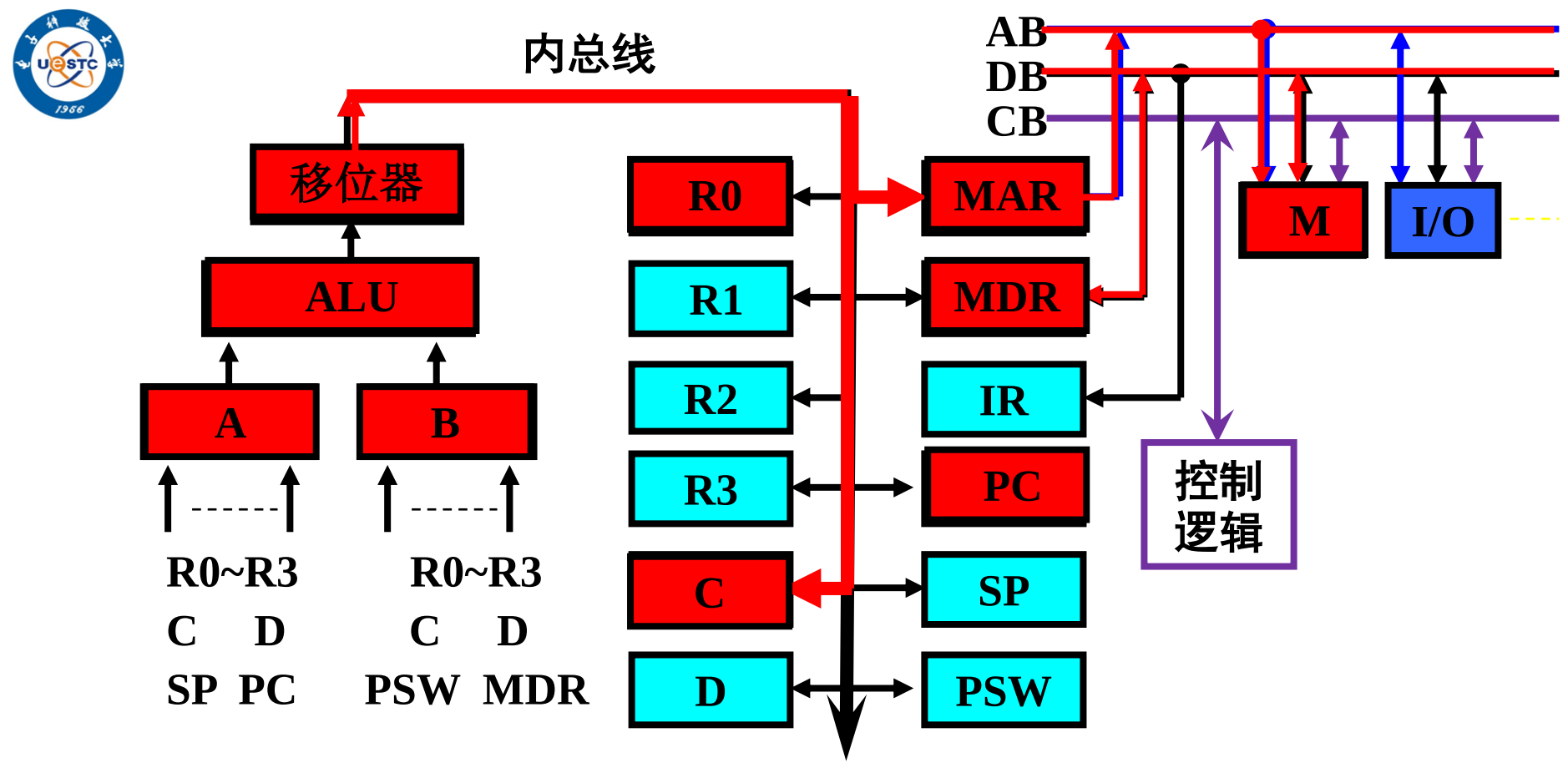


数据已送入MDR
数据地址=8000H



1）操作数地址：→MAR

寄存器间址：Ri → A/B → ALU → 移 → 内 —打入→ MAR



变址寻址：X(R)→MAR

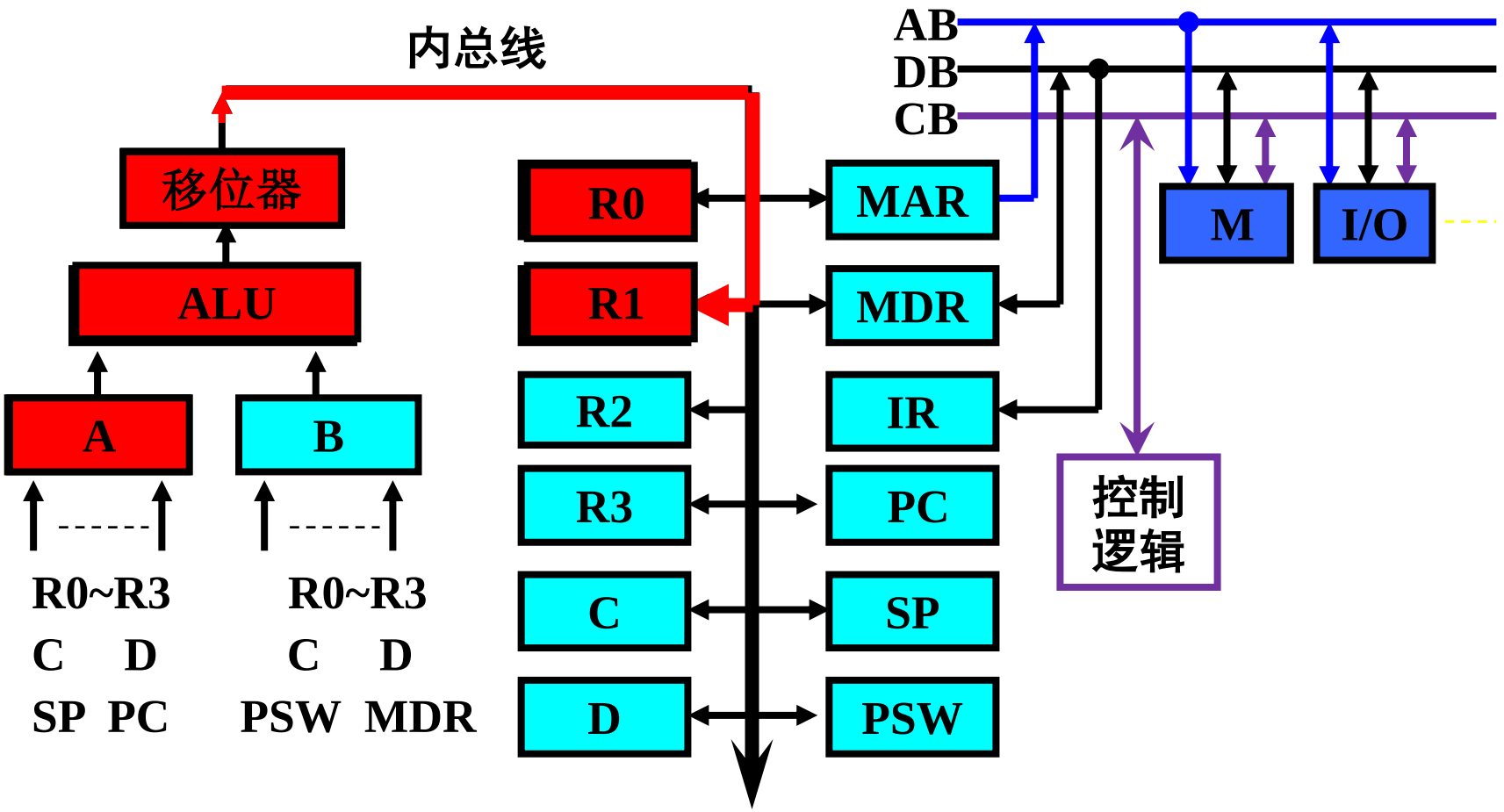
PC → A → ALU → 移 → 内 → MAR → AB → M

M → DB → MDR → B → ALU → 移 → 内 → C/D

Ri → A

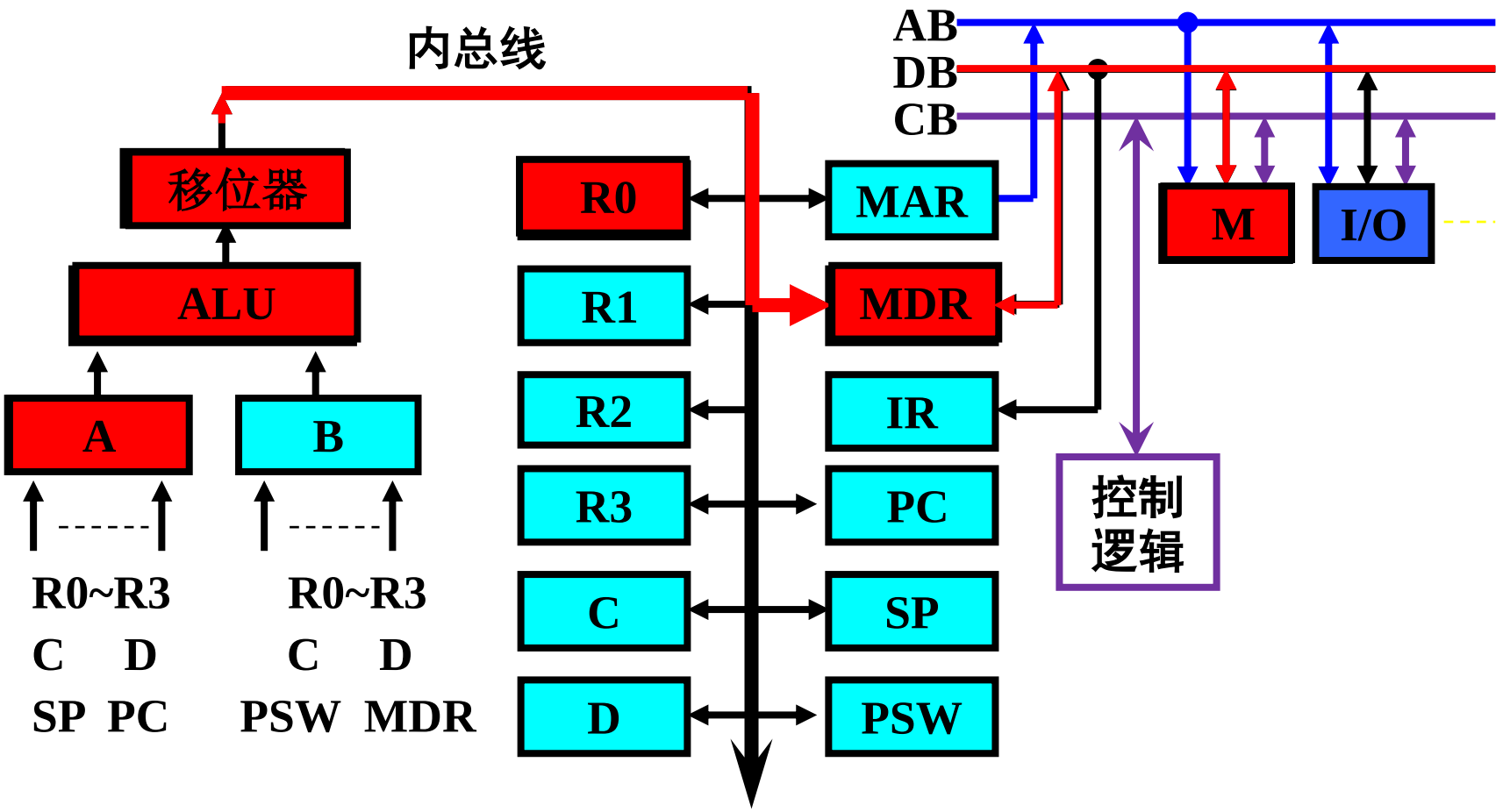
C/D → B

ALU → 移 → 内 → MAR

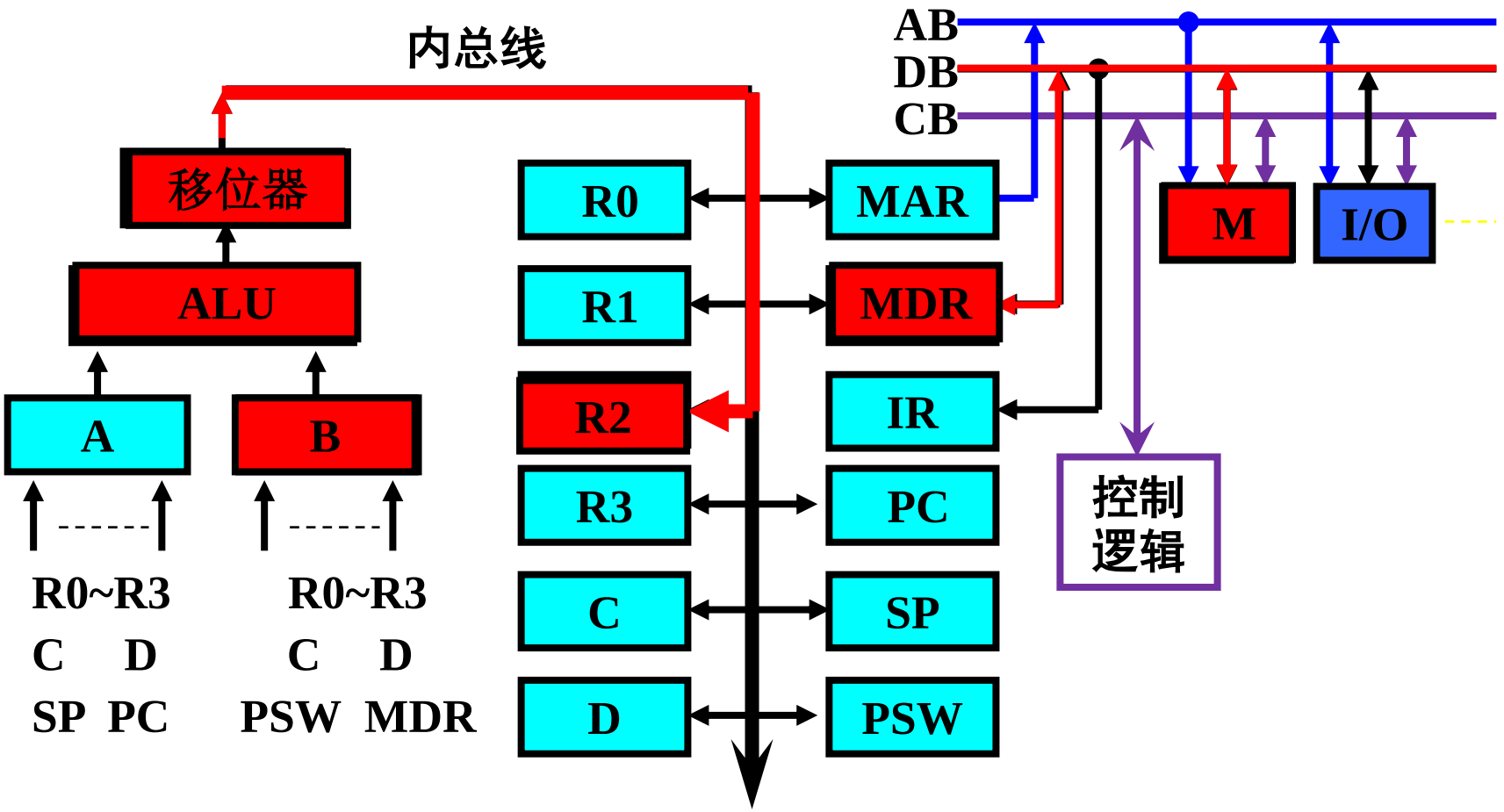


2）存取操作数

① R ⟶ R：R0 ⟶ A/B ⟶ ALU ⟶ 移 ⟶ 内 —打入→ R1

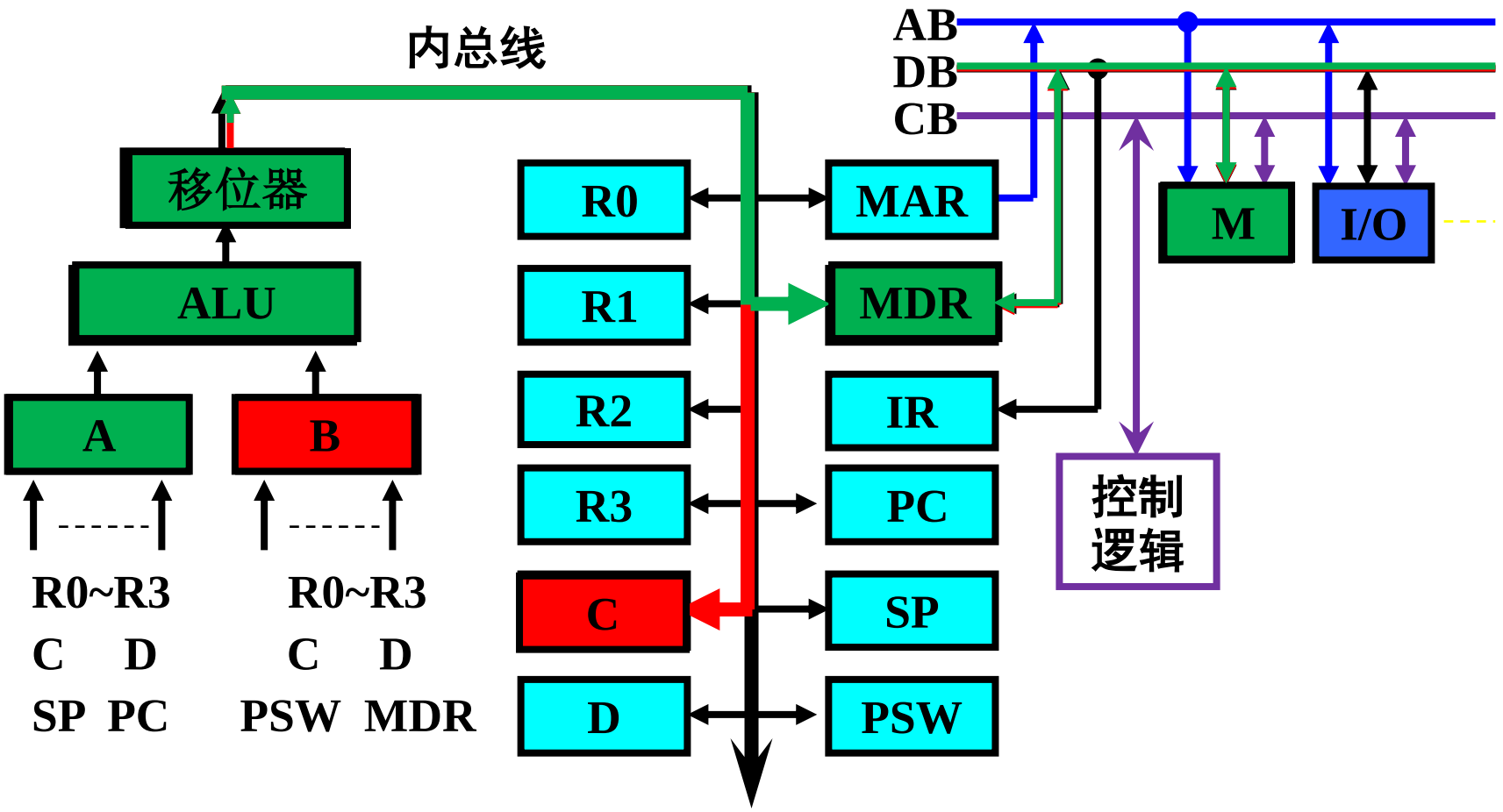


② R ⟶ M: Ri ⟶ A/B ⟶ ALU ⟶ 内 ⟶ MDR ⟶ DB ⟶ M



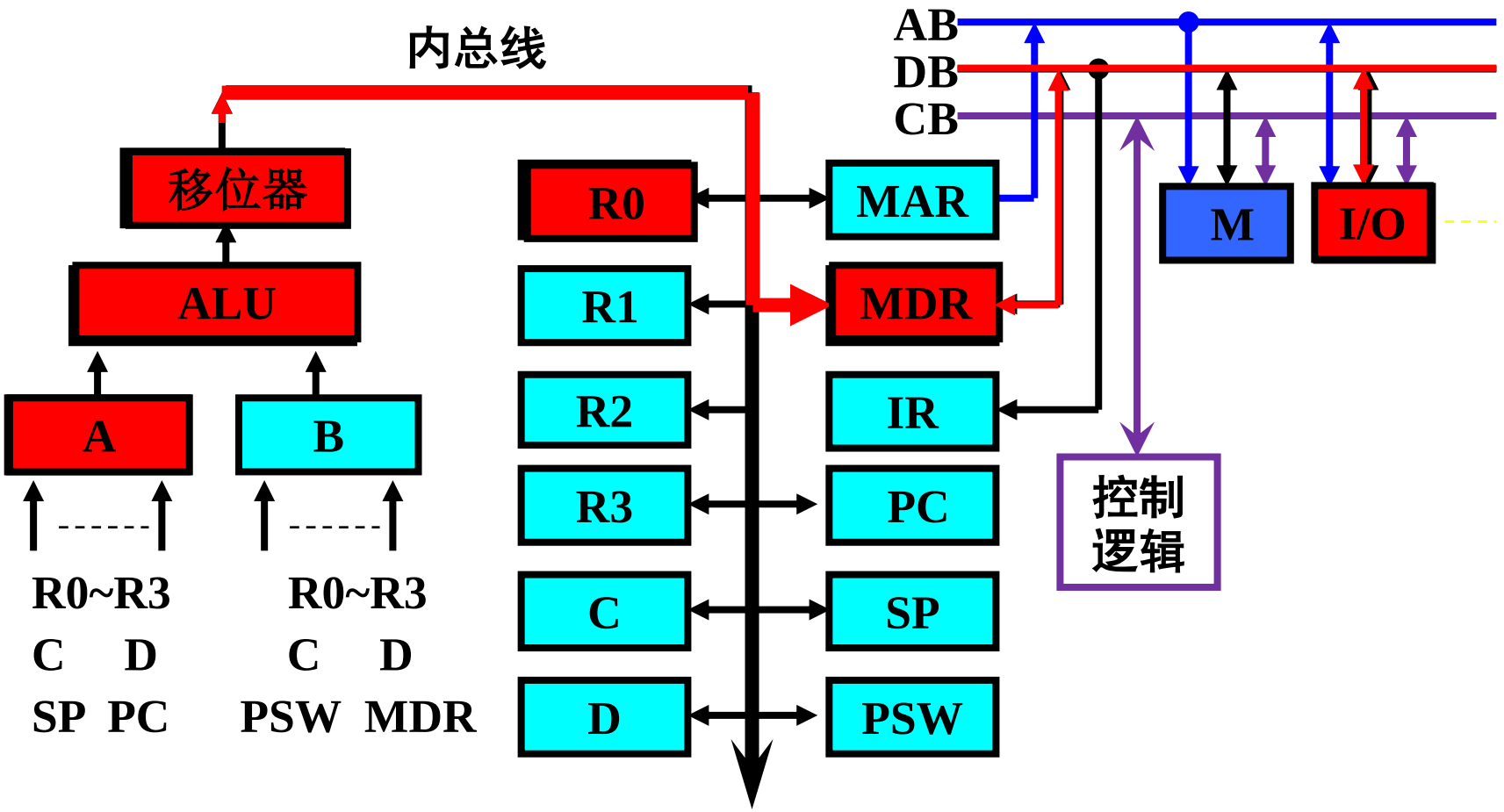
③ M ⟶ R：

M⟶DB⟶MDR⟶B⟶ALU⟶移、内⟶Ri



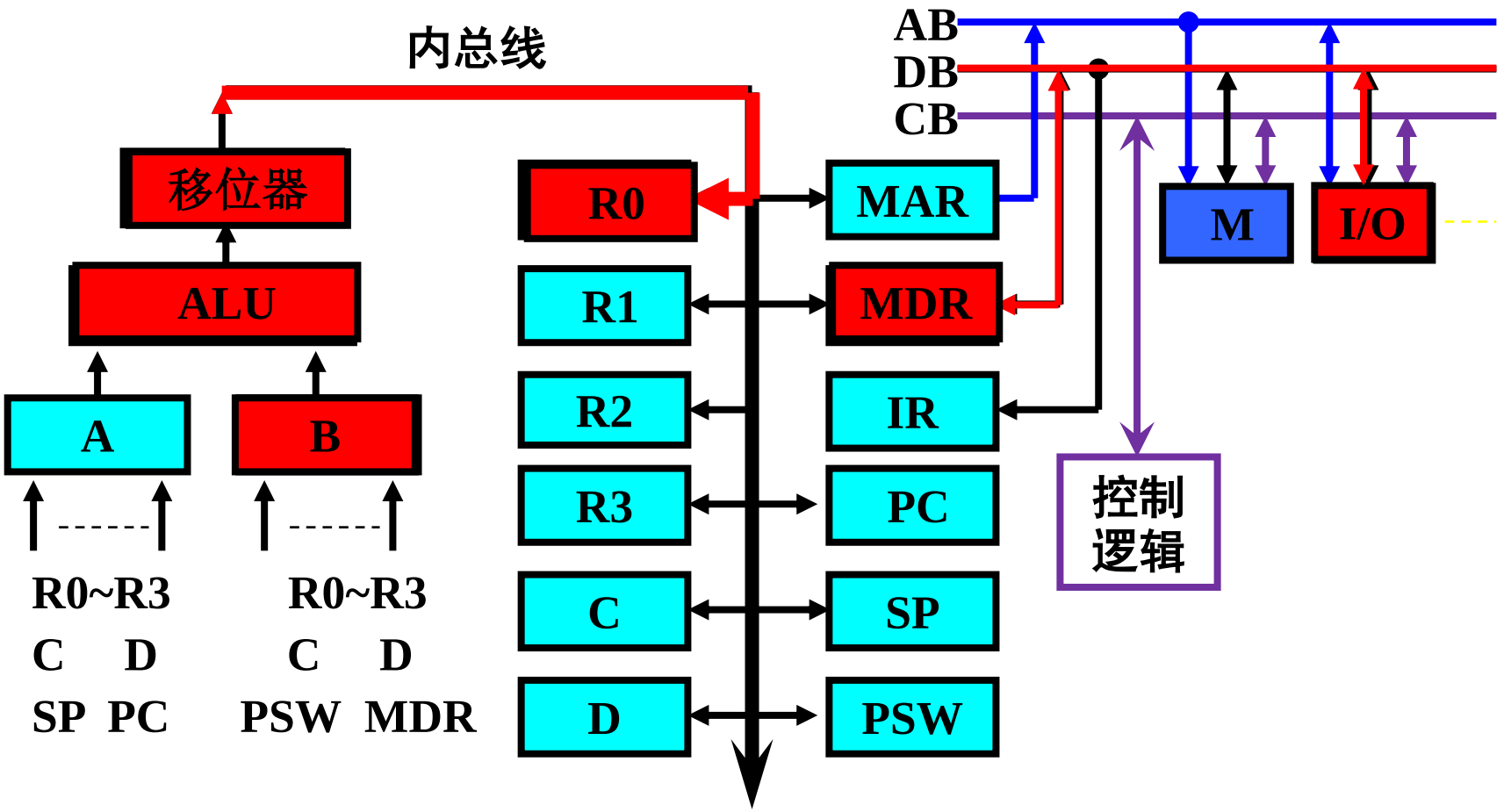
④ M ⟶ M：M(源) ⟶ DB ⟶ MDR ⟶ ALU ⟶ 内 —打入→ C

(计算目的地址) C ⟶ ALU ⟶ 内 ⟶ MDR ⟶ DB ⟶ M(目的)



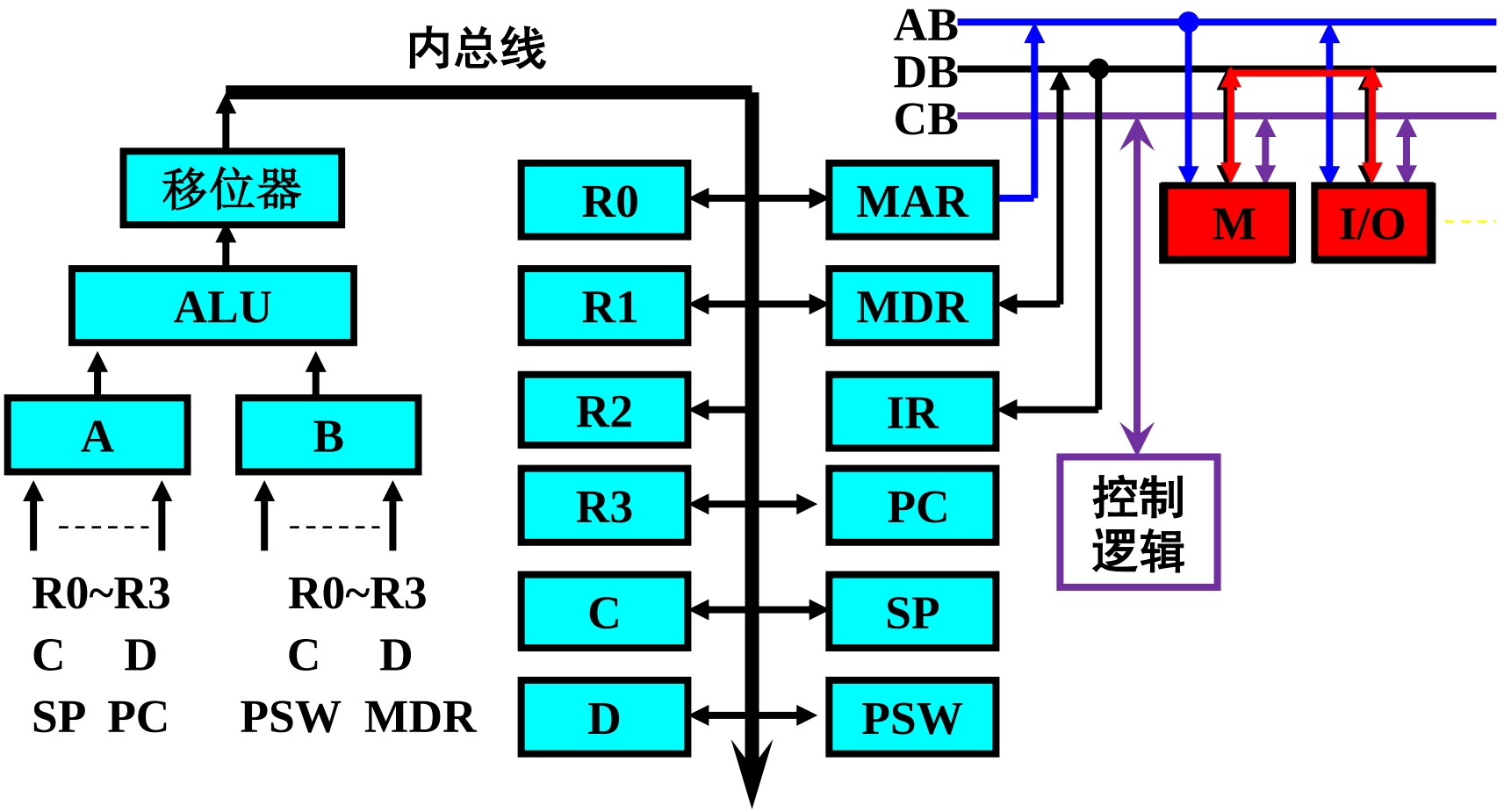
⑤ R → I/O：

Ri → ALU → 内 → MDR → DB → I/O



⑥ I/O → R

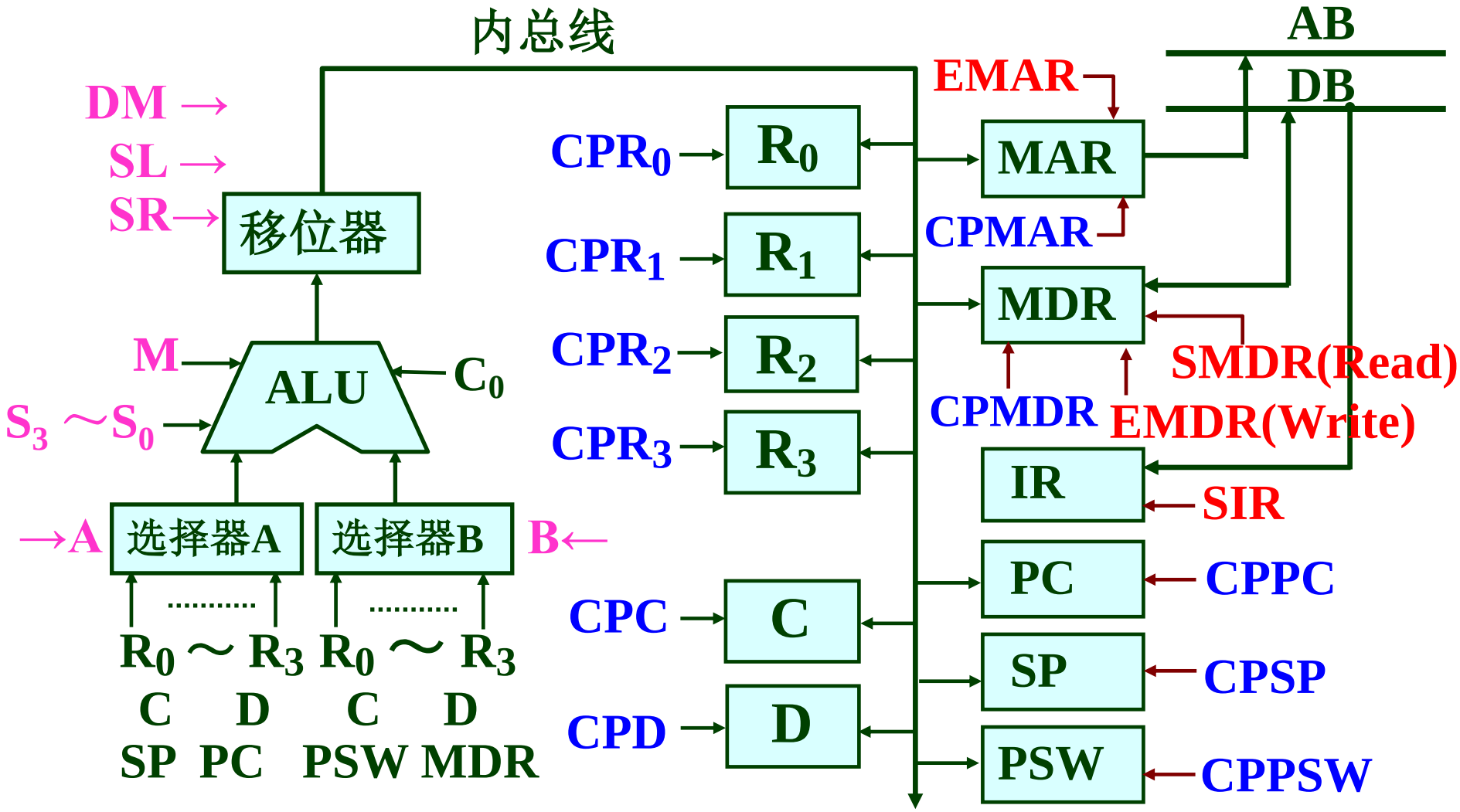
I/O → DB → MDR → ALU → 内 —打入→ Ri



⑦ I/O ⟷ M

DMA方式：I/O ⟷ DB ⟷ M

4. 模型机微命令设置


内总线
AB
DB
EMAR
DM →
SL →
SR→
移位器
CPR0
R0
MAR
CPR1
R1
CPMAR
MDR
M
ALU
C0
CPR2
R2
SMDR(Read)
S3 ～S0
CPMDR
EMDR(Write)
CPR3
R3
IR
SIR
→A
选择器A
选择器B
B←
PC
CPPC
CPC
C
SP
CPSP
CPD
D
PSW
CPPSW
R0 ～ R3
R0 ～ R3
C
D
C
D
SP
PC
PSW
MDR