

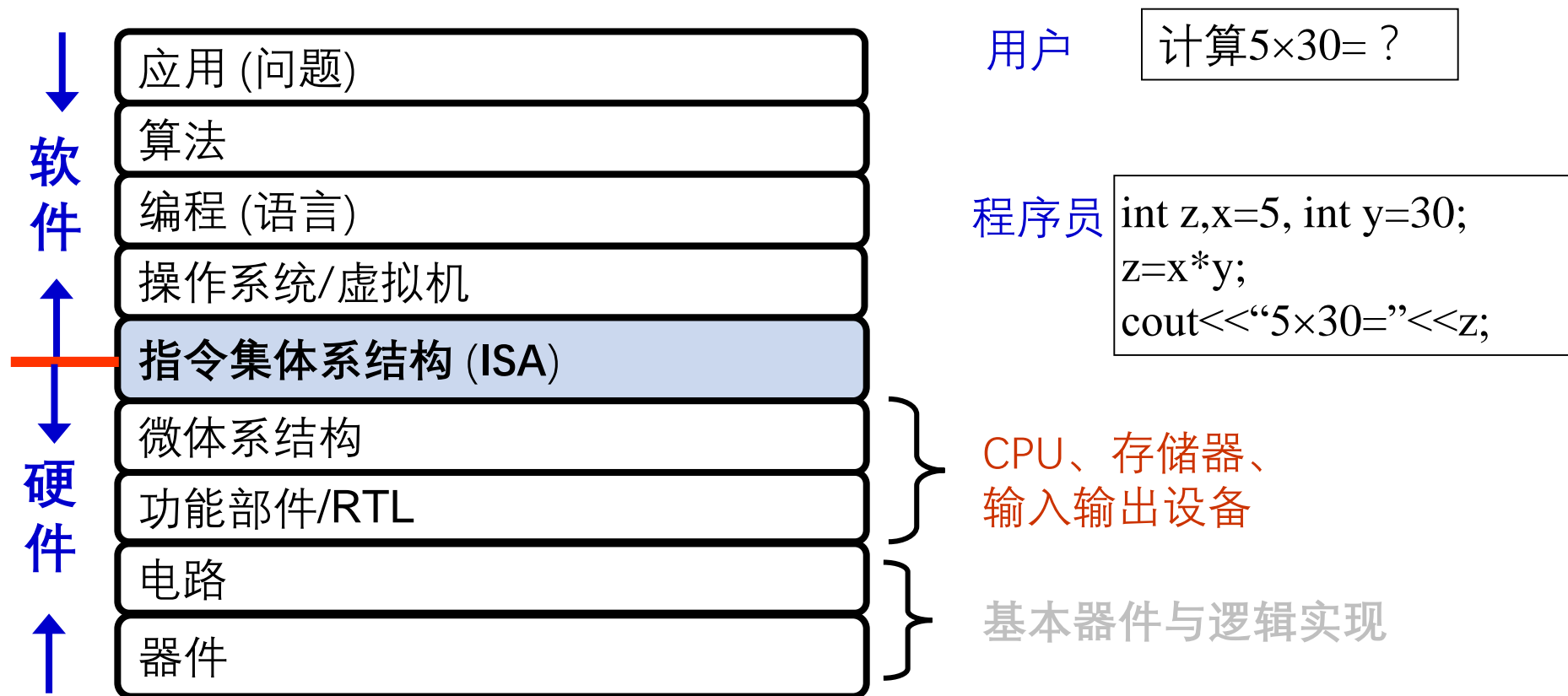
1.3 计算机的执行过程

主讲人：邓倩妮
上海交通大学





问题：计算机是怎样执行程序的呢？



计算机系统的抽象层及其转换



指令集体系结构举例

指令格式： 操作码 [地址码]

例如：

零地址指令， 如: NOP

一地址指令， 如: INC R1

二地址指令， 如: ADD R1, R2

如: ADD R1, A

如: ADD R1, #1

三地址指令， 如: ADD R1, R2, R3

指令所操作的数据可能在存储器或寄存器中， 也可以是立即数



计算机的工作过程

假设某计算机的指令系统中有如下几条指令：

(1) LOAD R1, M1

(2) ADD R1, M2

(3) STORE M3, R1

汇编语言程序	机器语言程序	功能
LOAD R1, M1	0000 0001 00000010	将M1的内容送入R1
ADD R1, M2	0001 0001 00000100	将R1的内容加上M2的内容再送回R1
STORE M3, R1	0010 0001 00000110	将R1的内容送入M3中



指令和数据都存放在内存中

汇编语言程序	机器语言程序	功能
LOAD R1, M1	0000 0001 00000010	将M1的内容送入R1
ADD R1, M2	0001 0001 00000100	将R1的内容加上M2的内容再送回R1
STORE M3, R1	0010 0001 00000110	将R1的内容送入M3中

内存地址：
按字节编码

00000010	00110110	M1的第一个字节
00000011	01000011	M1的第二个字节
.....	
00011000	0000 0001	Load 指令的第一个字节
00011001	00000010	Load 指令的第二个字节
00011010	0001 0001	ADD 指令的第一个字节
00011011	00000100	ADD 指令的第二个字节
00011100	0010 0001	STORE 指令的第一个字节
00011101	00000110	STORE 指令的第二个字节
.....		

字节：8位二进制代码

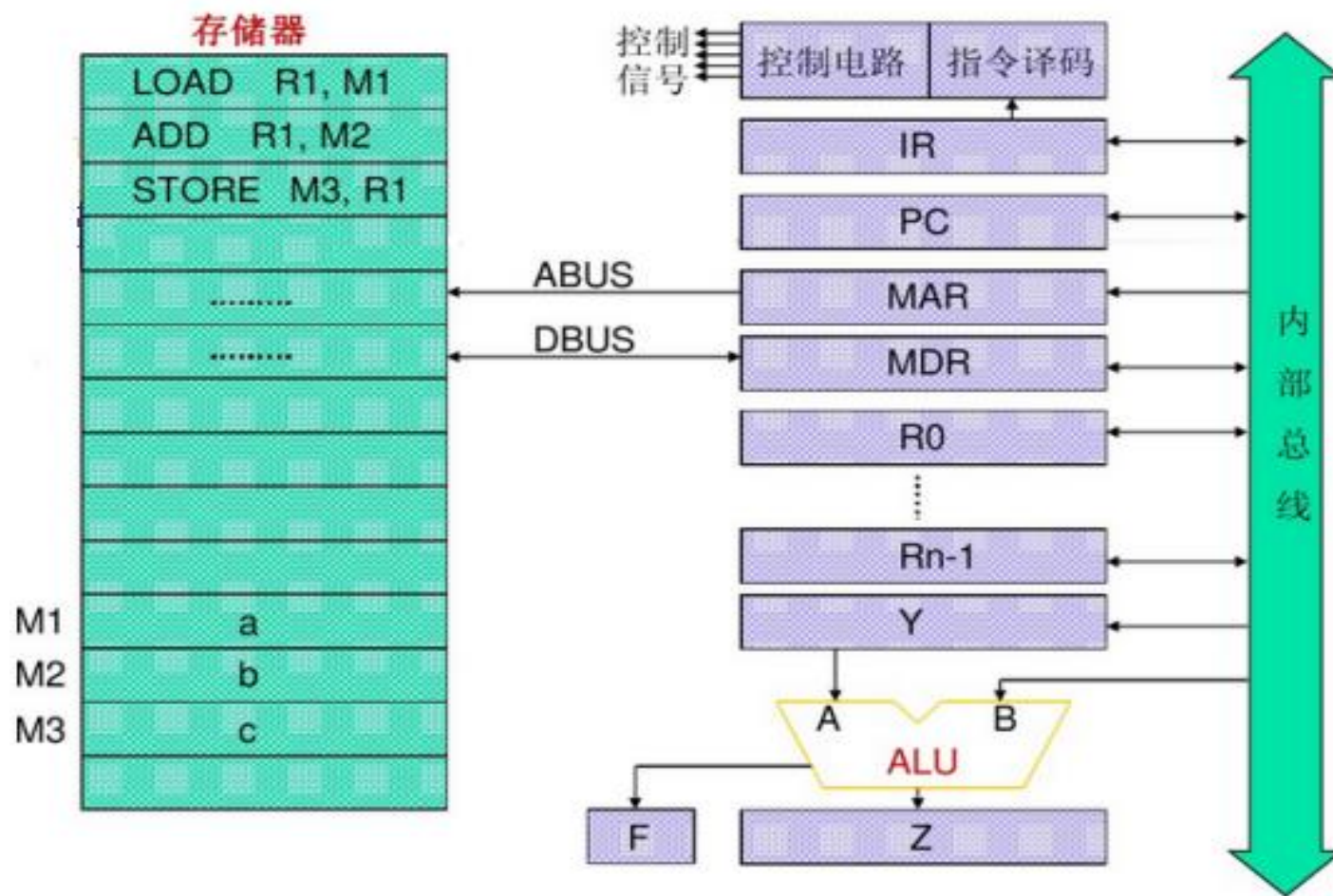
存储单元：具有特定存储地址的存储单位

存储容量：存储器中可存二进制代码的总量，单位：KB，MB，GB

指令字长：一条指令的二进制代码位数，一般是字节的倍数



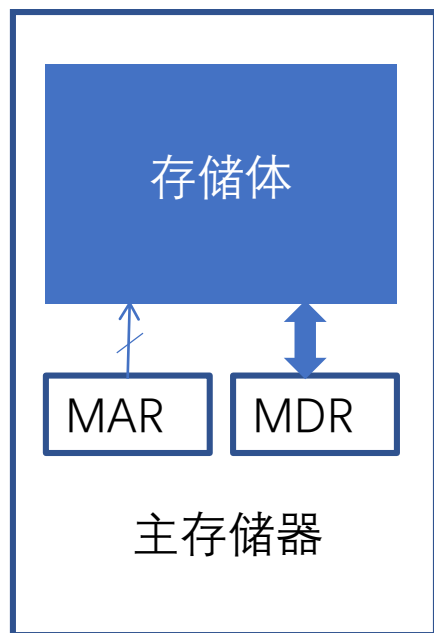
指令的执行过程



寄存器：
中央处理器内的
组成部分；有限
存储容量的高速
存储部件；可用
来暂存指令、数
据和地址



存储器的基本组成



如何进行存储器访问？（按地址访问）

MAR：存储器地址寄存器
地址长度可反映存储器的容量

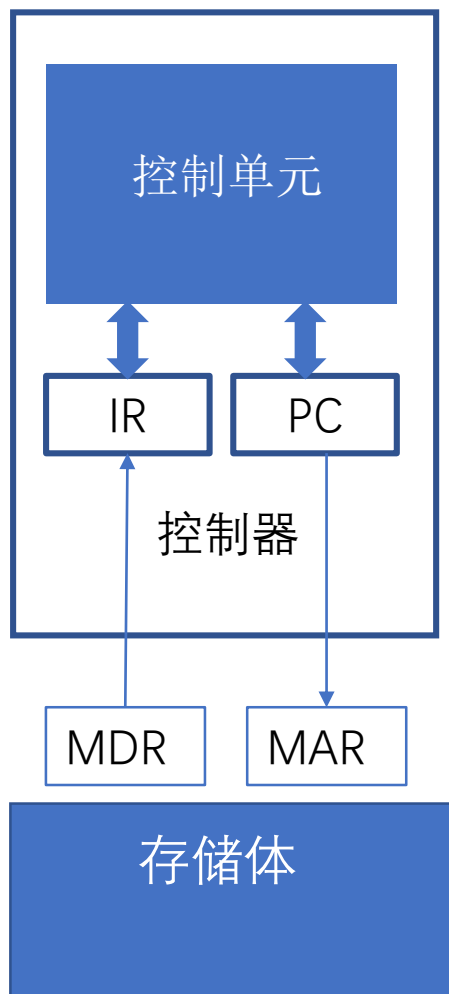
MDR：存储器数据寄存器
寄存器宽度可反映存储字长

存储字长：
每次访问存储器获得的数据的位数
（一个存储字长是字节的倍数）

vs. 指令字长



控制器的基本组成



PC：程序计数器
存放当前即将执行的指令的地址
可以自增1, $(PC) + 1$ 条指令的字节数 $\rightarrow (PC)$

IR：指令寄存器
存放当前即将执行的指令

控制器的工作：

step1. 取下一条指令

step2. 解码指令：

step3. 执行指令：

将控制信号发送给相关部件，执行相应的运算

重复 step1-step3



指令执行过程

内存地址

00000010

00110110

M1的第一个字节

00000011

01000011

M1的第二个字节

.....

00011000

0000 0001

Load 指令的第一个字节

00011001

00000010

Load 指令的第二个字节

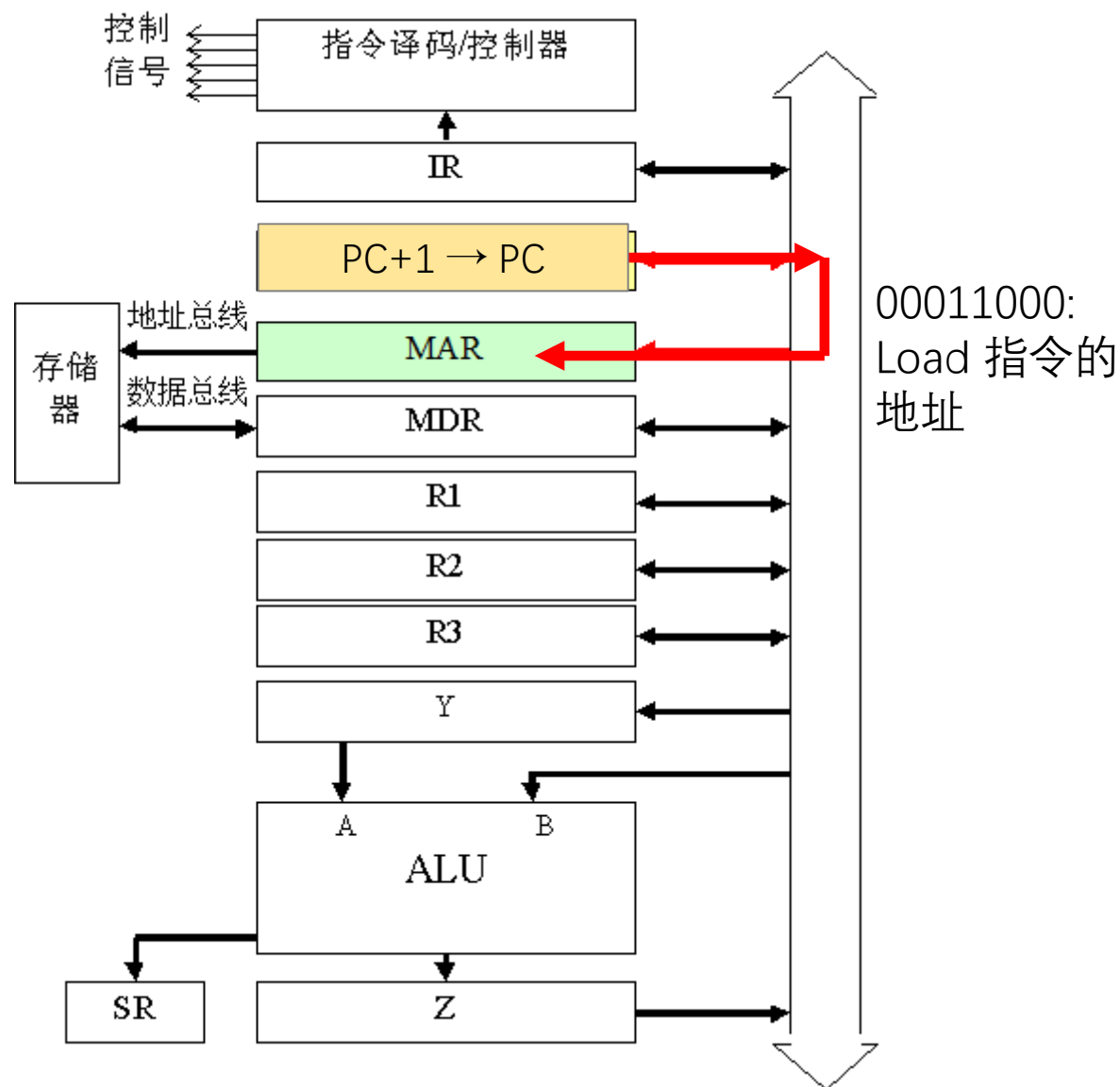
00011010

0001 0001

读操作：如LOAD R1, mem

(1) PC→MAR

(2) PC+1→PC (增加两个字节)





指令执行过程

内存地址

00000010	00110110	M1的第一个字节
00000011	01000011	M1的第二个字节
.....	
00011000	0000 0001	Load 指令的第一个字节
00011001	00000010	Load 指令的第二个字节
00011010	0001 0001	

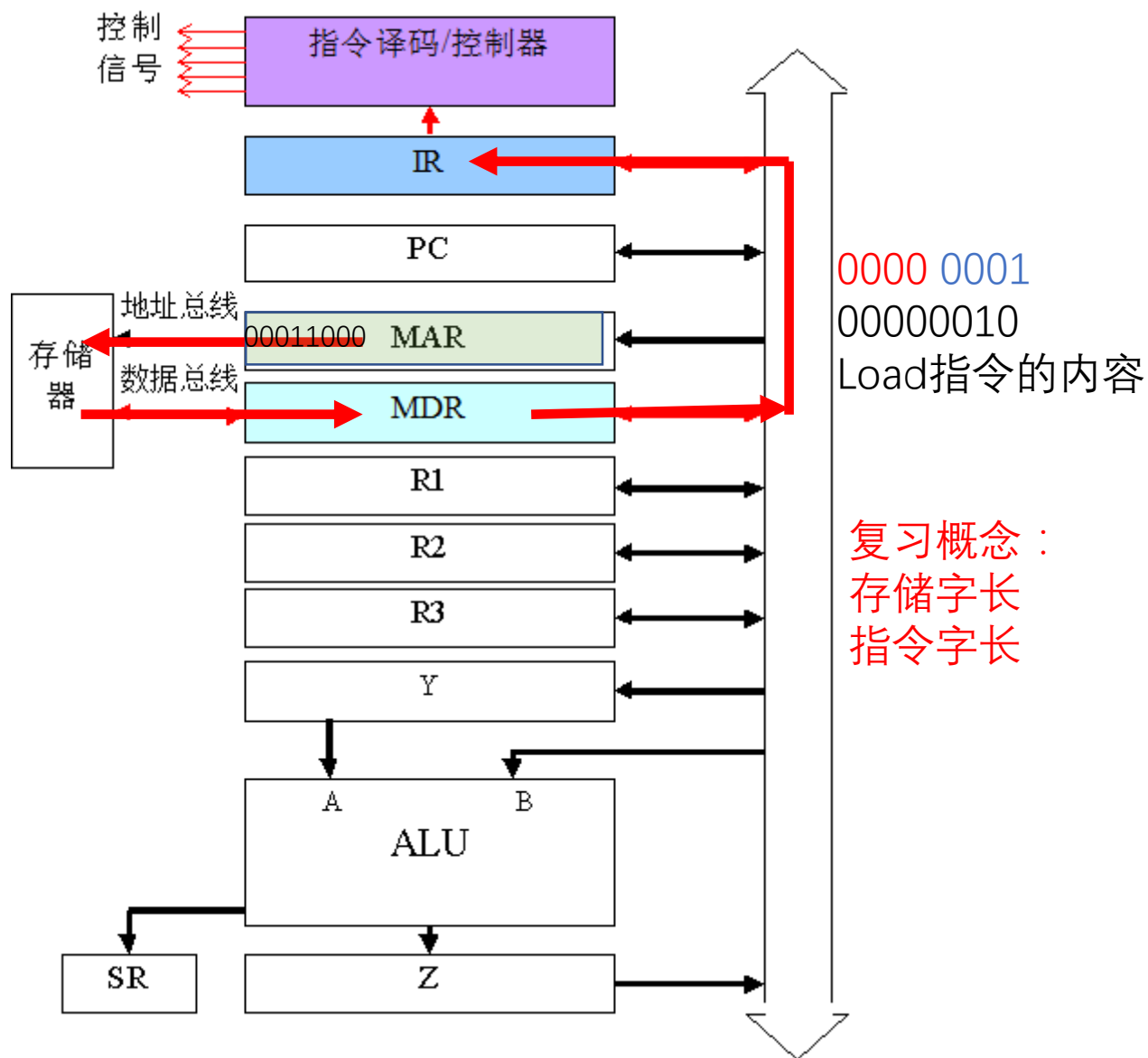
读操作：如LOAD R1, mem

(1) PC→MAR

(2) PC+1→PC

(3) DBUS→MDR

(4) MDR→IR





指令执行过程

内存地址

00000010

00110110

M1的第一个字节

00000011

01000011

M1的第二个字节

.....

.....

00011000

0000 0001

Load 指令的第一个字节

00011001

00000010

Load 指令的第二个字节

00011010

0001 0001

读操作：如LOAD R1, mem

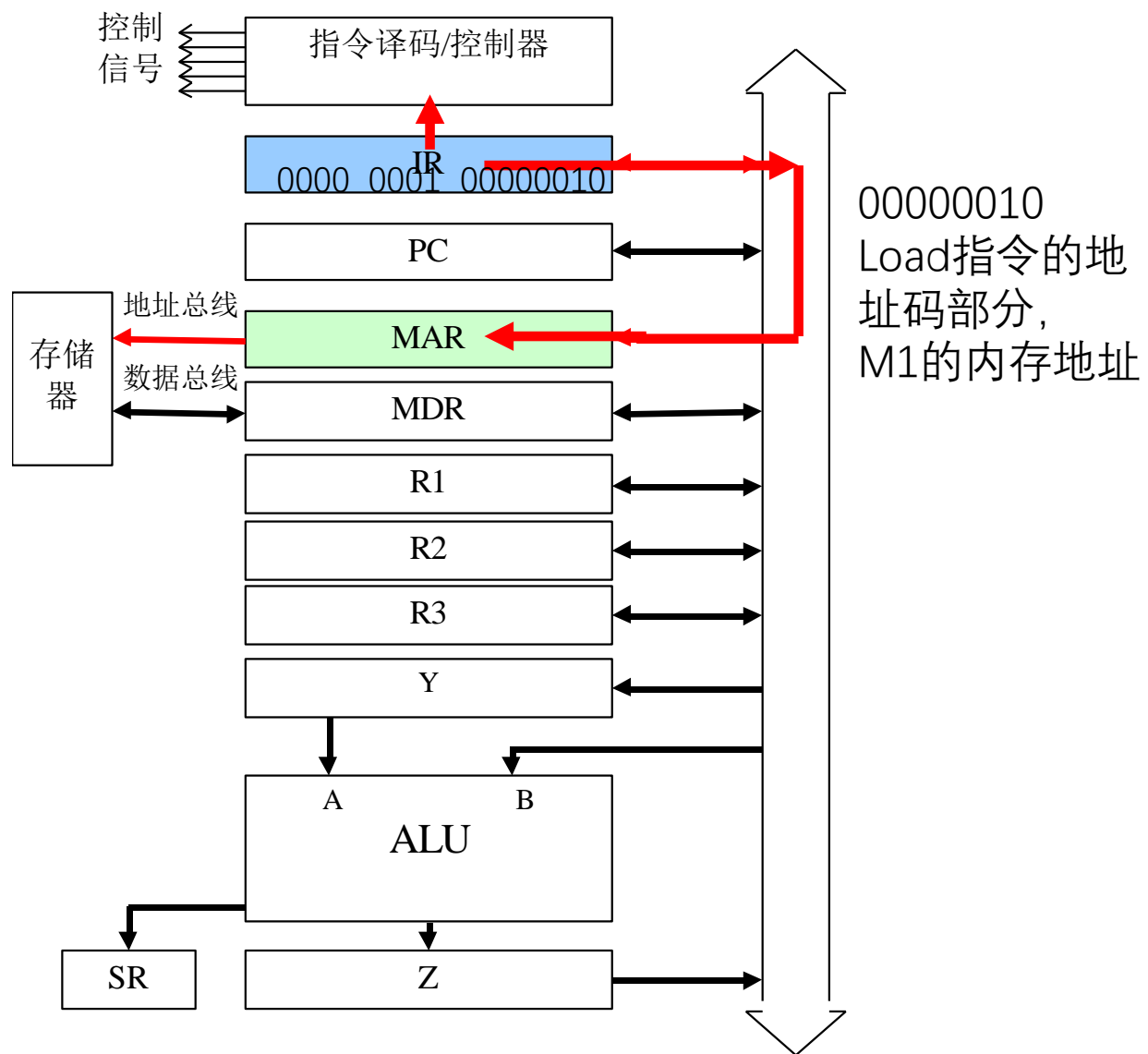
(1) PC→MAR

(2) PC+1→PC

(3) DBUS→MDR

(4) MDR→IR

(5) IR(地址段)→MAR, 读存储器



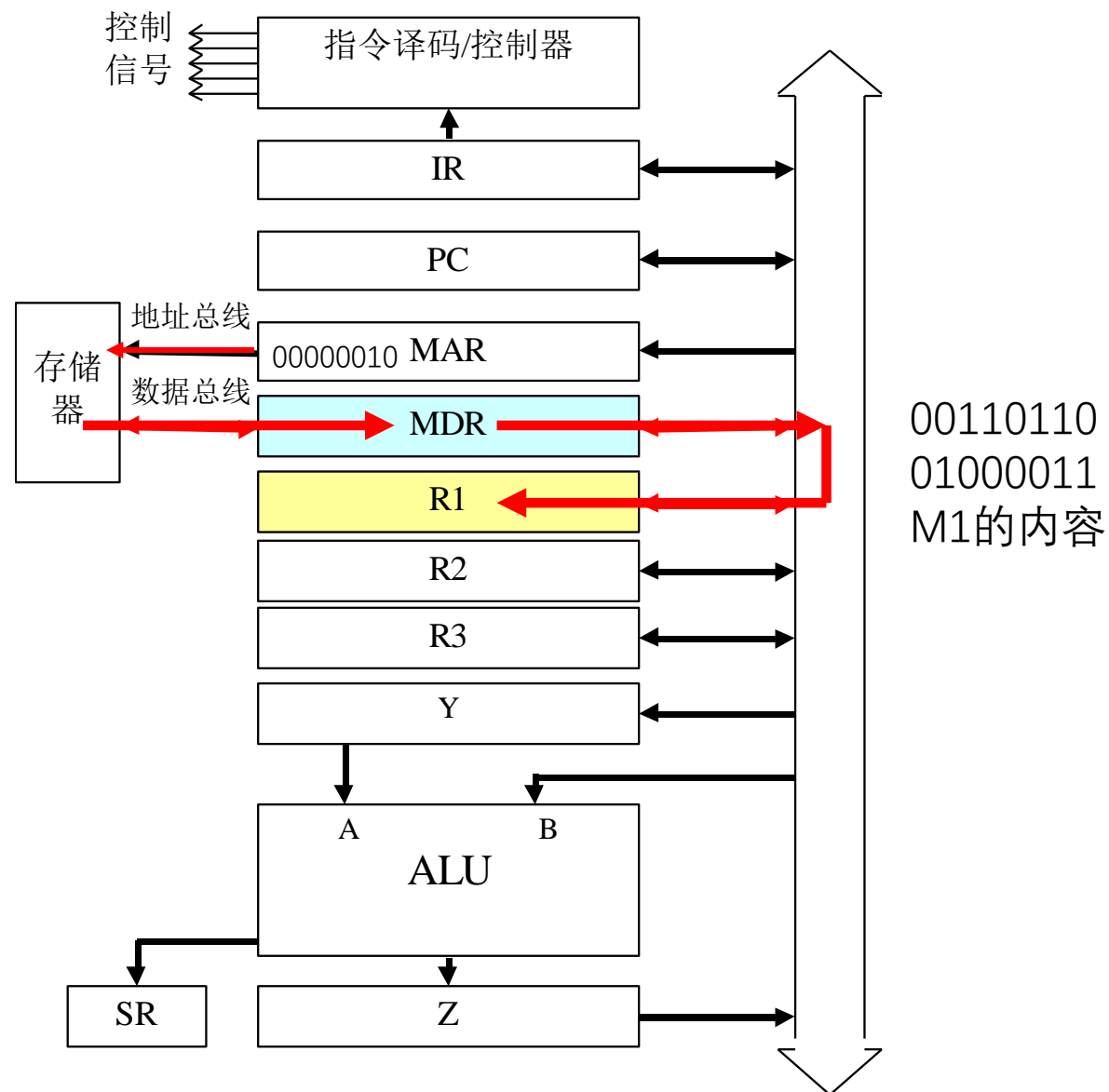


指令执行过程

内存地址		
00000010	00110110	M1的第一个字节
00000011	01000011	M1的第二个字节
.....	
00011000	0000 0001	Load 指令的第一个字节
00011001	00000010	Load 指令的第二个字节
00011010	0001 0001	

读操作：如LOAD R1, mem

- (1) PC→MAR
- (2) PC+1→PC
- (3) DBUS→MDR
- (4) MDR→IR
- (5) IR(地址段)→MAR, 读存储器
- (6) DBUS→MDR
- (7) MDR→R1





指令的执行过程 (习题)

CPU区分存储器中取出的是指令还是数据的依据是？

- A.指令译码结果不同
- B.访问指令和访问数据时寻址方式不同
- C.访问指令和访问数据时所处的指令执行阶段不同
- D.指令和数据所在的存储单元地址范围不同

答案：（C） 考察存储程序的概念； 考察一条指令执行的过程



小结

- 机器指令的执行过程
- 一些术语：
 - **寄存器**：中央处理器内的组成部分；有限存贮容量的高速存贮部件；可用来暂存指令、数据和地址
 - 字节：8位二进制代码
 - **存储单元**：具有特定存储地址的存储单位
 - 存储容量：存储器中可存二进制代码的总量，单位：KB，MB，GB
 - **指令字长**：一条指令的二进制代码位数
 - 存储字长：每次访问存储器获得的数据的位数
 - **机器字长**：CPU能同时处理的数据位数

