

# 计算机基础

51CTO学院：邹月平

# 冯·诺依曼计算机

□通过内部存储器安放存储程序，成功解决了当时计算机存储容量太小，运算速度过慢的问题。

因为有我创造101



51CTO学院

01

计算机由五大部件组成

02

指令和数据以同等地位存于存储器，可按地址寻访

03

指令和数据用二进制表示

04

指令由操作码和地址码组成

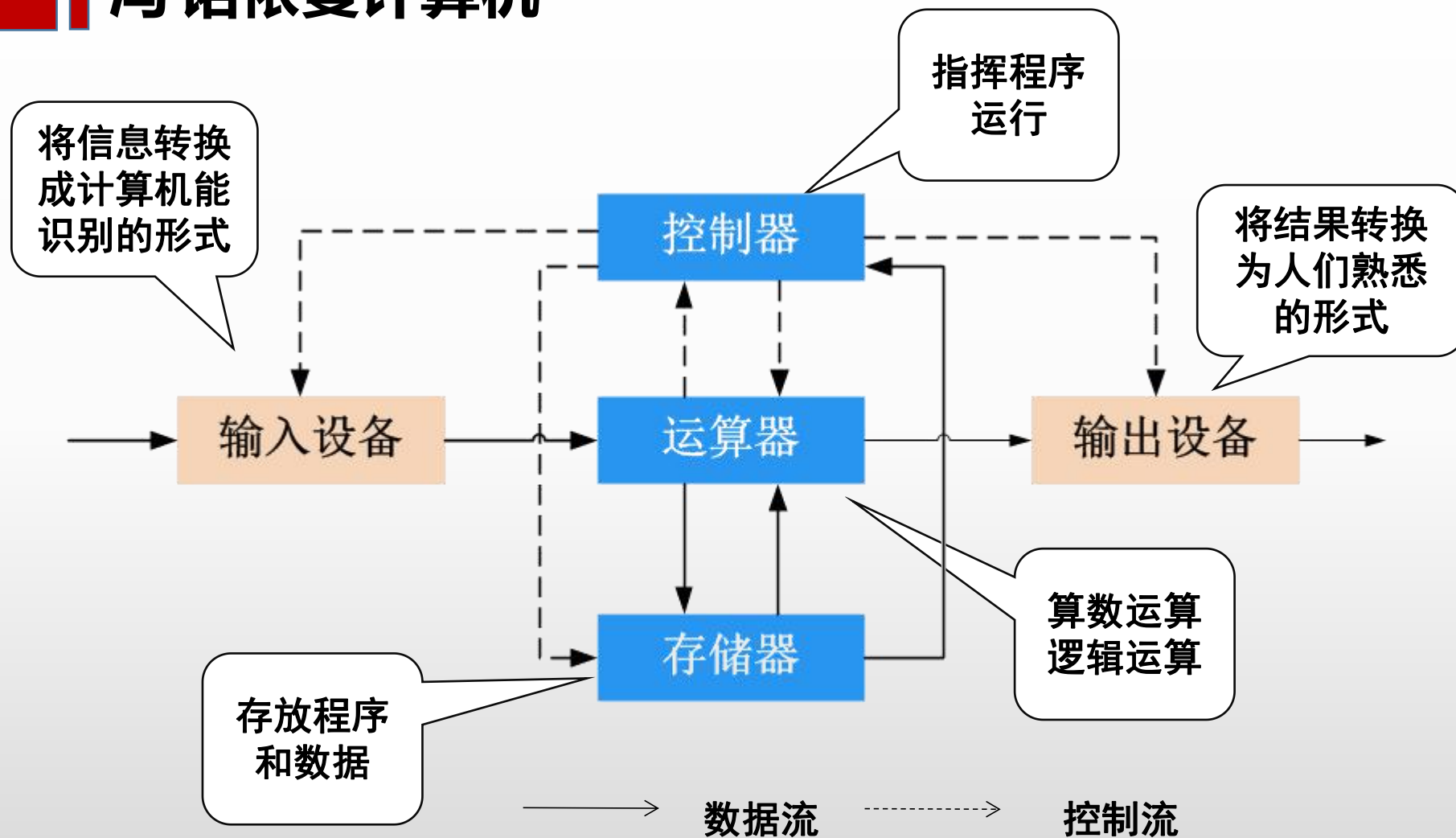
05

存储程序

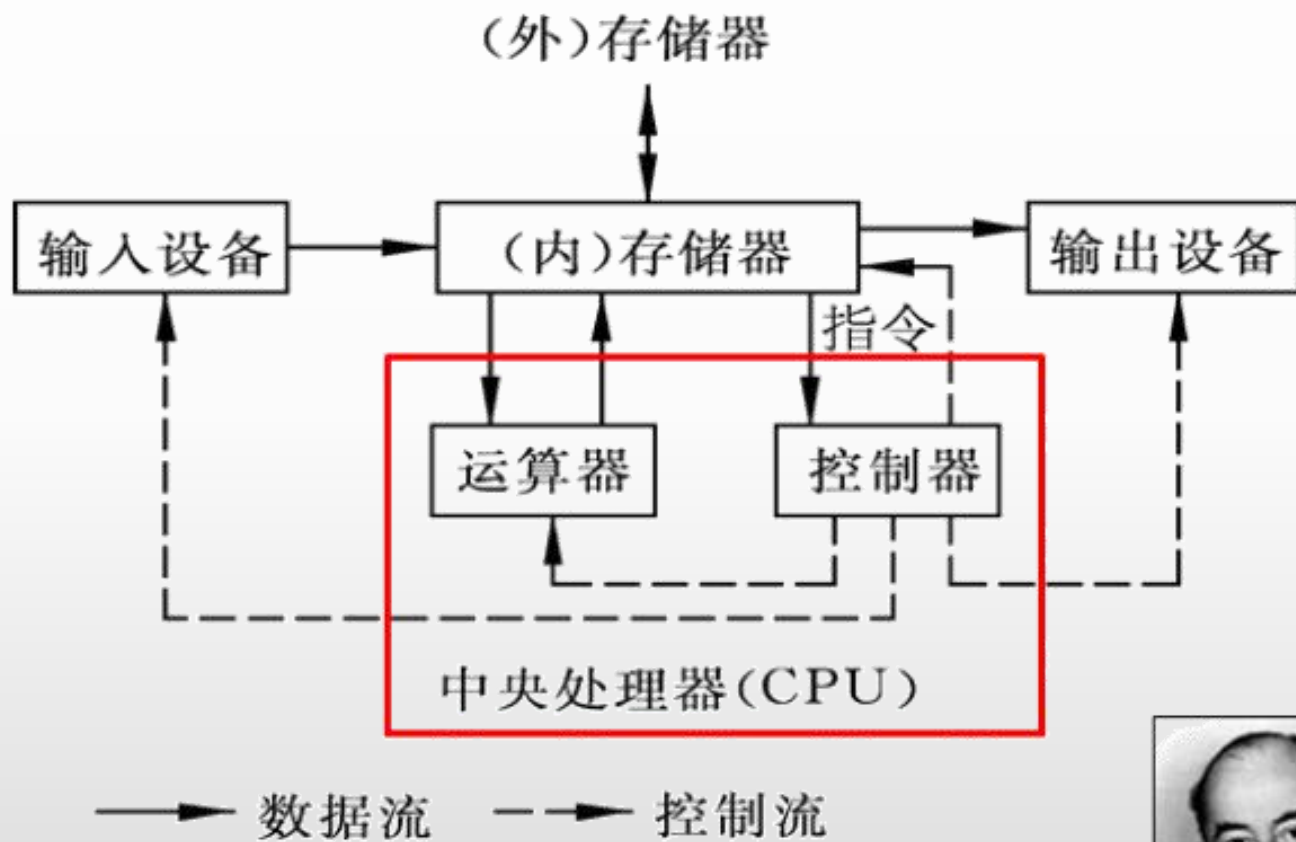
06

以运算器为中心

# 冯·诺依曼计算机



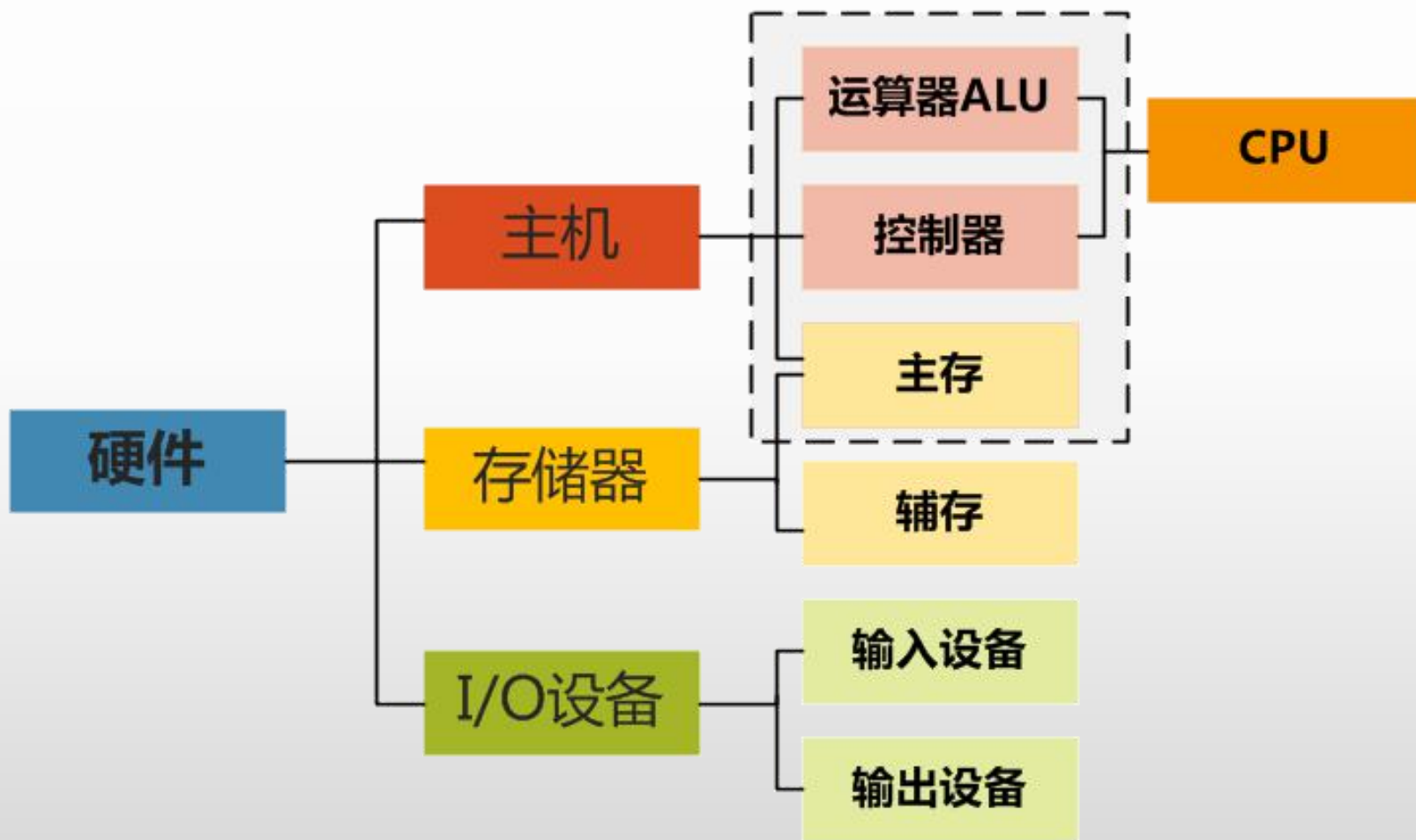
# 冯·诺依曼计算机



计算机硬件系统

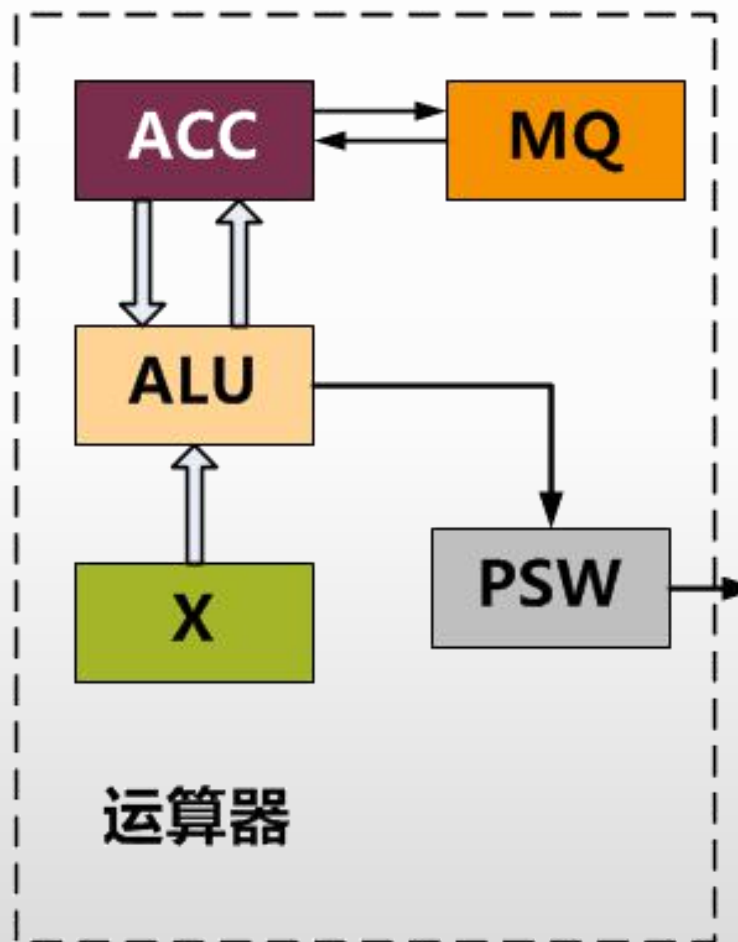


# 现代计算机硬件图



# CPU

	ACC	MQ	X
加法	被加数 和		加数
减法	被减数 差		减数
乘法	乘积高位	乘数 乘积 低位	被乘数
除法	被除数 余数	商	除数



# CPU



例如：取数A

000001 0000001000

存数B

加C

乘D

打印E

停机



$[A] \rightarrow ACC$

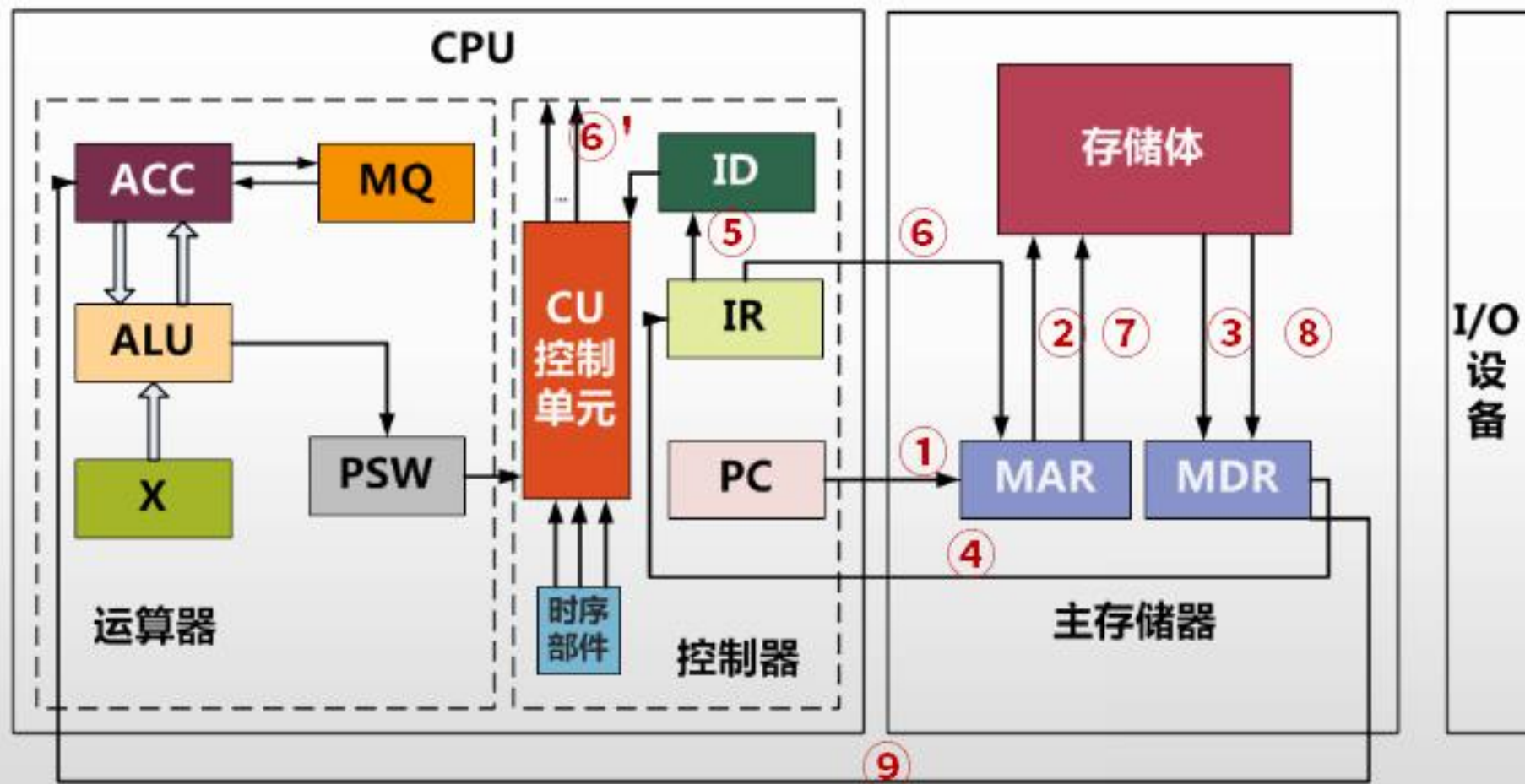
$[ACC] \rightarrow B$

$[ACC] + [C] \rightarrow ACC$

$[ACC] \times [D] \rightarrow ACC$

$[E] \rightarrow \text{打印机}$

# 现代计算机硬件图





# CPU

## 运算器

- ① 算术逻辑单元ALU
- ② 累加寄存器AC
- ③ 数据缓冲寄存器DR
- ④ 状态条件寄存器PSW

## 控制器

- ① 程序计数器PC
- ② 指令库寄存器IR
- ③ 指令译码器ID
- ④ 地址寄存器AR
- ⑤ 时序部件

一条指令的执行步骤：

取指：PC（程序计数器）→AR（地址寄存器）→M存储器→DR数据寄存器  
→IR指令寄存器

分析：OP→ID→CU控制器

执行：Ad →AR（地址寄存器）→M存储器→DR数据寄存器→AC累加器

# CPU

## □ 运算器的组成包含如下：

- ① **算术逻辑单元ALU**：数据的算术运算和逻辑运算；
- ② **累加寄存器AC**：通用寄存器，为ALU提供一个工作区，用来暂存数据；
- ③ **数据缓冲寄存器DR**：写内存时，暂存指令或数据；
- ④ **状态条件寄存器PSW**：存状态标志与控制标志。

## □ 控制器的组成包含如下：

- ① **程序计数器PC**：存储下一条要执行指令的地址；
- ② **指令寄存器IR**：存储即将执行的指令；
- ③ **指令译码器ID**：对指令中的操作码字段进行分析解释；
- ④ **地址寄存器AR**：用来保存当前CPU所访问的内存单元的地址。
- ⑤ **时序部件**：提供时序控制信号。

## 典型真题

CPU执行算术运算或者逻辑运算时，常将源操作数和结果暂存在（ ）中。

- A. 程序计数器（PC）
- B. 累加器（AC）
- C. 指令寄存器（IR）
- D. 状态寄存器（PSW）

# 典型真题

## 试题分析

选项A指令寄存器用来存放从内存中读取的指令。

选项B通用寄存器可用于传送和暂存数据，也可参与算术逻辑运算，并保存运算结果。

选项C程序计数器是用于存放下一条指令所在单元的地址。当执行一条指令时，首先需要根据PC中存放的指令地址，将指令由内存取到指令寄存器中，即将程序计数器PC中的内容送到地址总线上，此过程称为“取指令”。与此同时，PC中的地址或自动加1或由转移指针给出下一条指令的地址。此后经过分析指令，执行指令。完成第一条指令的执行，而后根据PC取出第二条指令的地址，如此循环，执行每一条指令。

选项D状态寄存器用于存放状态标志与控制标志如，中断标志、溢出标志等。

参考答案：B

## 典型真题

执行CPU指令时，在一个指令周期的过程中，首先需从内存读取要执行的指令，此时先要将指令的地址即（ ）的内容送到地址总线上。

- A. 指令寄存器（IR）
- B. 通用寄存器（GR）
- C. 程序计数器（PC）
- D. 状态寄存器（PSW）

# 典型真题

## 试题分析

PC（程序计数器）是用于存放下一条指令所在单元的地址。当执行一条指令时，处理器首先需要从PC中取出指令在内存中的地址，通过地址总线寻址获取。

选项A指令寄存器用来存放从内存中读取的指令。

选项B通用寄存器可用于传送和暂存数据，也可参与算术逻辑运算，并保存运算结果。

选项D状态寄存器用于存放状态标志与控制标志，如中断标志、溢出标志等。

参考答案：C

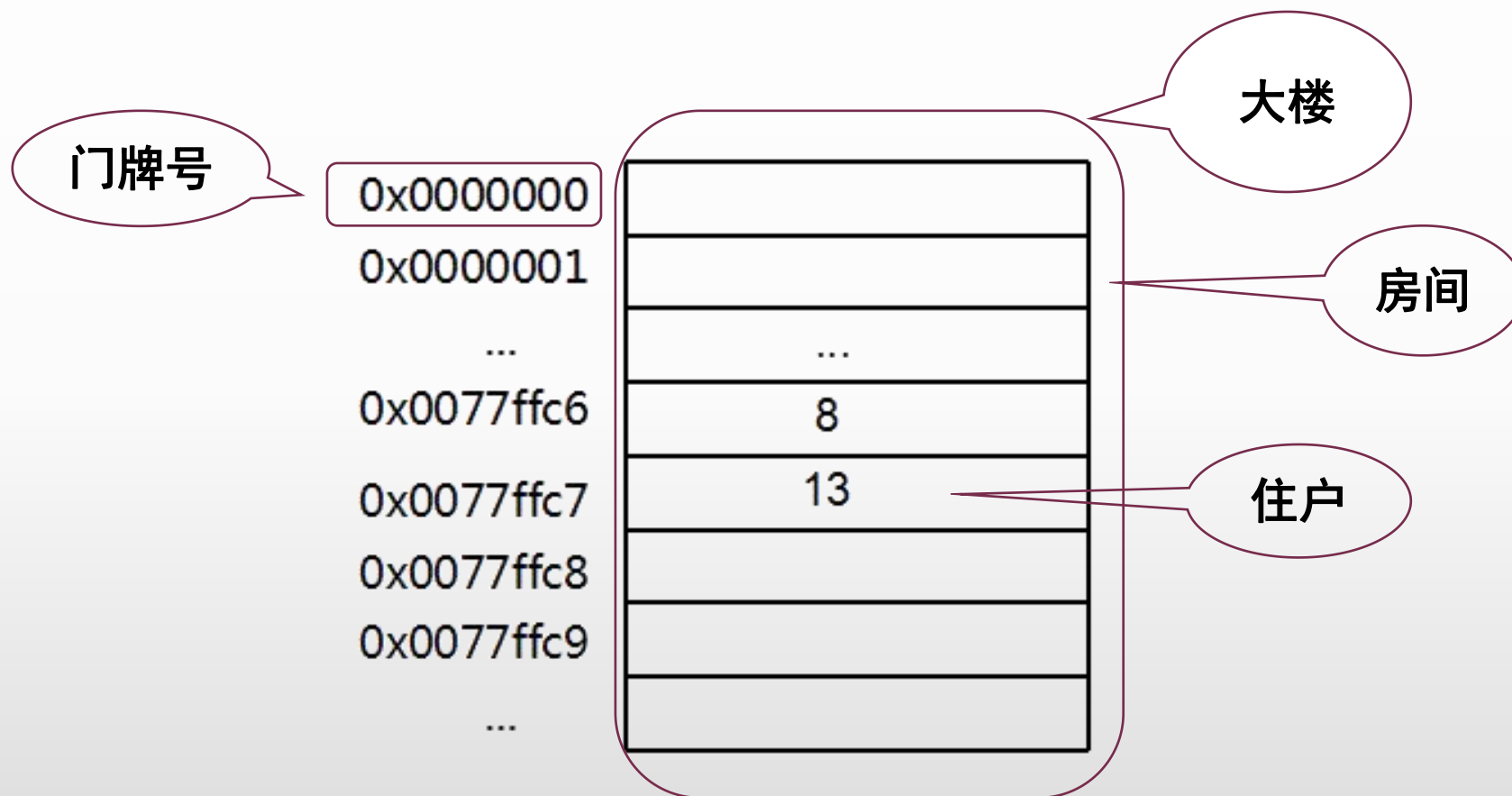
# 主存

□ 存储器的基本单位是**存储单元**，一般以**8位二进制**为一个存储单元。每个存储单元都有一个**地址**，一般用**十六进制**数表示。

0x0000000	
0x0000001	
...	...
0x0077ffc6	8
0x0077ffc7	13
0x0077ffc8	
0x0077ffc9	
...	



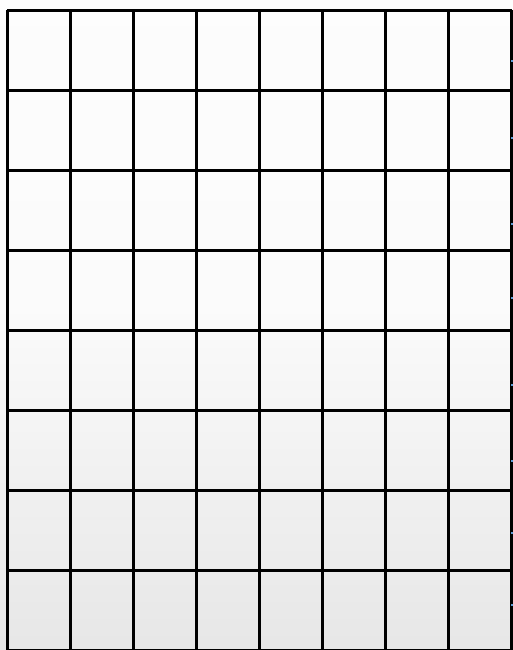
# 主存





# 主存

0x0000000  
0x0000001  
...  
0x0077ffc6  
0x0077ffc7  
0x0077ffc8  
0x0077ffc9  
...



1bit



地址总线：  
假如需要 $n$ 位二进制数来表示所有的地址，则地址总线的个数为 $n$ 。

数据总线：一次处理 $n$ 位的数据，则数据总线的长度为 $n$ 。  
 $n$ 的位数为一个字的长度。

## 典型真题

内存按字节编址，地址从A4000H到CBFFFH，共有（1）个字节。若用存储容量为 $32\text{K} \times 8\text{bit}$ 的存储芯片构成该内存，至少需要（2）片。

A、80K      B、96K      C、160K      D、192K

A、2      B、5      C、8      D、10

## 典型真题

试题分析：

1. 先求出地址总个数： $CBFFFH - A4000H + 1 = CC000 - A4000 = 28000H$ （十六进制）

$$2 * (16^4) + 8 * (16^3) = 163840 = 160 * 1024 \text{ (个)} = 160K \text{ (个)}$$

2. 求解芯片个数

$$160KB / 32KB = 5 \text{ (个)}$$

答案：C、B

## 典型真题

内存按字节编址，利用 $8K \times 4\text{bit}$ 的存储器芯片构成84000H到8FFFFH的内存，共需（ ）片。

- A. 6
- B. 8
- C. 12
- D. 24

## 典型真题

试题分析：

寻址空间范围：

$8FFFFH - 84000H + 1 = C000H = 12 \times 16^3 = 12 \times 2^{12} = 48k$ 。内存容量为  $48K \times 8bit$ ，一片芯片是容量为  $8K \times 4bit$ ，共需要12片。

参考答案：C

# 存储器

## 存储器

### 按照所处位置分类

- 内存（主存）：CPU当前使用的指令和数据
- 外存（辅存）：存放后备程序和数据

### 按照构成材料分类

- 半导体存储器
  - 静态存储器：双稳态触发器
  - 动态存储器：依靠电容上的电荷存储信息，主存
- 磁存储器：利用磁性材料两种不同的状态长期保存信息，外存
- 光存储器：利用光斑、晶像的变化表示信息，外存

### 按照工作方式分类

- 读/写存储器：RAM
- 固定只读存储器ROM：用户不能写数据
- 可编程的只读存储器PROM：用户可写入一次
- 只读存储器
  - 可擦除可编程的只读存储器EPROM：可多次编程，紫外线擦除
  - 电擦除可编程的只读存储器EEPROM：可多次编程，电擦除
  - 闪存：接近EEPROM，U盘

### 按照访问方式分类

- 按地址访问的存储器
- 按内容访问的存储器

### 按照寻址方式分类

- 随机存储器RAM：按地址访问存储器的任一单元，主存
- 顺序存储器SAM：访问时按顺序查找目标地址，磁带
- 直接存储器DAM：按照数据块所在位置访问，磁盘
- 相联存储器：按照内容进行访问，Cache

## 典型真题

计算机系统中，（1）方式是根据所访问的内容来决定要访问的存储单元，常用在（2）存储器中。

- A. 顺序存取
- B. 直接存取
- C. 随机存取
- D. 相联存取

- A. DRAM
- B. Cache
- C. EEPROM
- D. CD-ROM

# 典型真题

## 试题分析

存储器中数据常用的存取方式有顺序存取、直接存取、随机存取和相联存取等4种。

(1) 顺序存取：存储器的数据以记录的形式进行组织。对数据的访问必须按特定的线性顺序进行。磁带存储器采用顺序存取的方式。

(2) 直接存取：与顺序存取相似，直接存取也使用一个共享的读写装置对所有的数据进行访问。但是，每个数据块都拥有唯一的地址标识，读写装置可以直接移动到目的数据块的所在位置进行访问。存取时间也是可变的。磁盘存储器采用直接存取的方式。



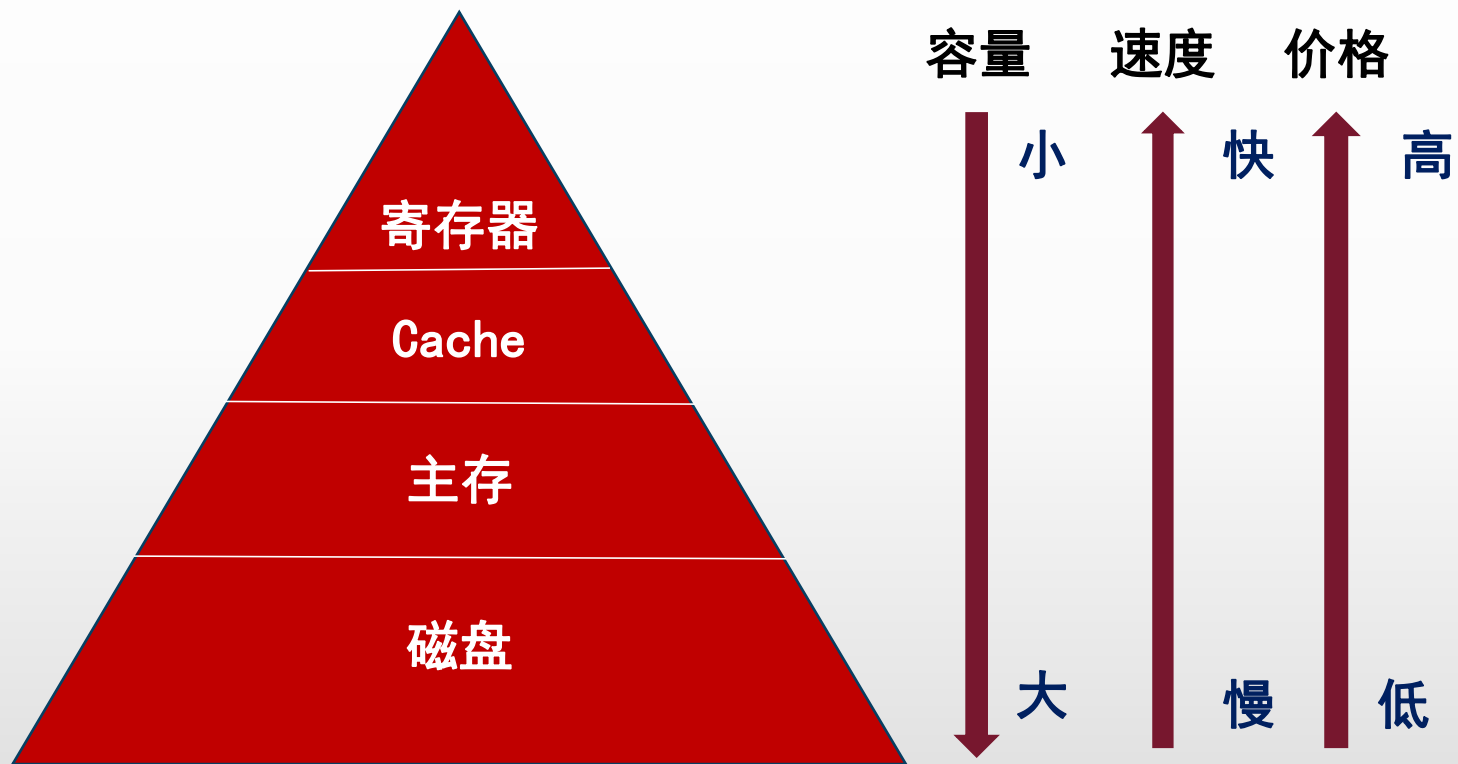
## 典型真题

(3) 随机存取：存储器的每一个可寻址单元都具有自己唯一的地址和读写装置，系统可以在相同的时间内对任意一个存储单元的数据进行访问，而与先前的访问序列无关。主存储器采用随机存取的方式。

(4) 相联存取：相联存取也是一种随机存取的形式，但是选择某一单元进行读写是取决于其内容而不是其地址。与普通的随机存取方式一样，每个单元都有自己的读写装置，读写时间也是一个常数。使用相联存取方式，可以对所有的存储单元的特定位置进行比较，选择符合条件的单元进行访问。为了提高地址映射的速度，Cache采取相联存取的方式。

参考答案： (1) D          (2) B

# 存储器



# 校验码

□信息保存在电容中，如遇电磁环境干扰，会导致电容的充放电或触发器的翻转，存在存储器的信息可能会出错。



# 校验码

## □ 码距与检错纠错：

一个编码系统的**码距**就是整个编码系统中任意(所有)两个码字的最小距离。若一个编码系统有四种编码分别为：0000，0011，1100，1111，此编码系统中0000与1111的码距为4；0000与0011的码距为2，是此编码系统的最小码距。因此该编码系统的码距为2。

➤ 在一个码组内为了检测e个误码，要求最小码距应该满足：

$$d \geq e + 1$$

➤ 在一个码组内为了纠正t个误码，要求最小码距应该满足：

$$d \geq 2t + 1$$

➤ 同时纠错检错： $d \geq e + t + 1$

## 校验码

□ 假如我们现在要对A，B两个字母进行编码。我们可以选用不同长度的编码，以产生不同码距的编码，分析它们的检错纠错能力。

1、若用1位长度的二进制编码。若A=0，B=1。这样A，B之间的最小码距为1。

合法码：{0, 1}；      非法码：{0, 1}；

2、若用2位长度的二进制编码，可选用11，00作为合法编码，也可以选用01，10作为合法编码。若以A=00，B=11为例，A、B之间的最小码距为2。

合法码：{11, 00}；      非法码：{10, 01}；

## 校验码

3、若用3位长度的二进制编码，可选用111，000作为合法编码。A，B之间的最小码距为3。

合法码：{111, 000}；      非法码：{001, 010, 011, 100, 101, 110}；

# 校验码

□ 奇偶校验：

00100011      偶校验：100100011      奇校验：000100011

以偶校验为例：假设100100011 有一位出错→100100010

假设100100011 有二位出错→100100000

只能发现奇数个位出错的情况



# 校验码

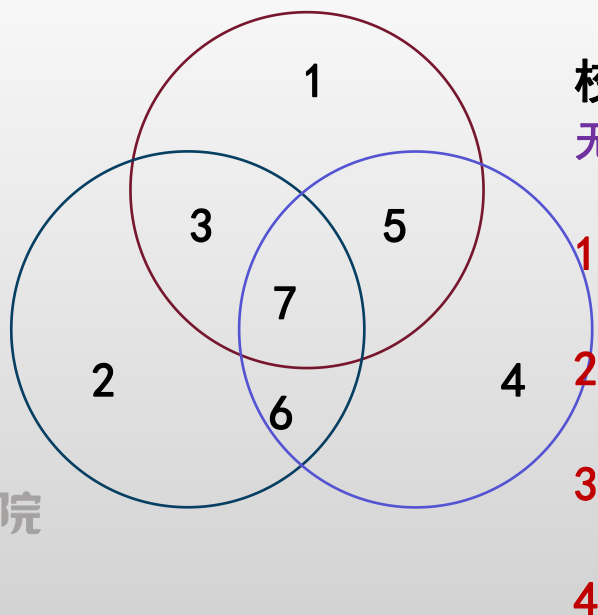
□ 海明码：奇偶校验、分组校验

00100011      偶校验：100100011      分组校验：10010 00011

例如：现有

1	2	3	4	5	6	7
---	---	---	---	---	---	---

分成3组，每组有1位校验位，共包括4位数据位。



校验结果：P<sub>3</sub> P<sub>2</sub> P<sub>1</sub>  
无错误→ 0 0 0

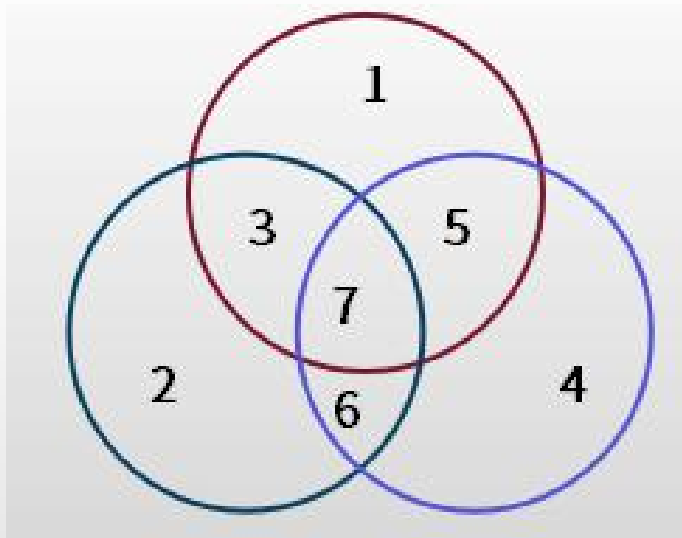
错误的位置

0  
0 0 1  
0 1 0  
0 1 1  
1 0 0  
1 0 1



# 校验码

## □ 海明码:



### 分组的方式:

第一组: XXXX**1**

1:00**1** 3:01**1** 5:10**1** 7:11**1** 9 11

第二组: XXX**1**X

2:0**1**0 3:0**1**1 6:1**1**0 7:1**1**1 10 11

第三组: XX**1**XX

4:**1**00 5:**1**01 6:**1**10 7:**1**11 12 13

第四组: X1XXX

第五组: 1XXXX

# 校验码

## □ 海明码：

海明码的校验码的位置必须是在 $2^n$ 位置（ $n$ 从0 开始，分别代表从右边数起分别是第1、2、4、8、16……），信息码也就是在非 $2^n$ 位置

...	12	11	10	9	8	7	6	5	4	3	2	1	位数
...	I8	I7	I6	I5		I4	I3	I2		I1			信息位
...					r3				r2		r1	r0	校验位

设数据位是 $n$ 位，校验位是 $K$ 位，则 $n$ 和 $k$ 必须满足以下关系：

$$2^k \geq n+k+1$$

# 校验码

例如：求0101按“偶校验” 配置的海明码

解析：n=4，根据 $2^k \geq n+k+1$ ，得K=3

7	6	5	4	3	2	1	信息位
1	0	1	r4	0	r2	r1	校验位

第一组：1357  $r1=3 \oplus 5 \oplus 7=0$

第二组：2367  $r2=3 \oplus 6 \oplus 7=1$

第三组：4567  $r4=5 \oplus 6 \oplus 7=0$       0101的海明码为0100101

## 典型真题

海明码是一种纠错码，其方法是为需要校验的数据位增加若干校验位，使得校验位的值决定于某些被校位的数据，当被校数据出错时，可根据校验位值的变化找到出错位，从而纠正错误。对于 32 位的数据，至少需要加（ 1 ）个校验位才能构成海明码。

- A. 3                      B. 4                      C. 5                      D. 6

## 典型真题

### 试题分析

海明码数据位与校验位之间的关系可以采用下述公式表示：

$2^k \geq n+k+1$ ，其中数据位是 $n$ 位，校验位是 $k$ 位。因此第5题套入公式可得需要6位校验位。

参考答案：D



## CRC循环冗余校验码

采用CRC进行差错校验,生成多项式为 $G(X)=X^4+X+1$ ,信息码字为10111,则计算出的CRC校验码是( )。

A. 0000

B. 0100

C. 0010

D. 1100

解题步骤: (1) 化解多项式为10011

(2) 信息码加0做模二除运算(不进位加法)

(3) 得到的余数即为校验码

# CRC循环冗余校验码

(1) 化解多项式为10011:

$$G(X) = X^4 + X + 1 = 2^4 + 2^1 + 2^0$$



$$2^4 + 2^1 + 2^0 \rightarrow 2^4 + 2^3 + 2^2 + 2^1 + 2^0$$

1    0    0    1    1

# CRC循环冗余校验码

(2) 信息码加0做模二除运算（不进位加法）

$$\begin{array}{r} 10100 \\ 10011 \overline{) 101110000} \\ \underline{10011} \phantom{0000} \\ 10000 \\ \underline{10011} \\ 1100 \end{array}$$

相同为0  
不同为1

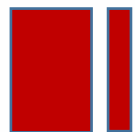
多项式最高次幂



# CRC循环冗余校验码

(3) 得到的余数即为校验码

最终的信息位+校验位为：101111100

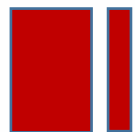


## CRC循环冗余校验码

若信息码字为111000110，生成多项式 $G(X) = X^5 + X^3 + X + 1$ ，则计算出的CRC校验码为（ ）。

- A. 01101
- B. 11001
- C. 001101
- D. 011001

51CTO学院



## CRC循环冗余校验码

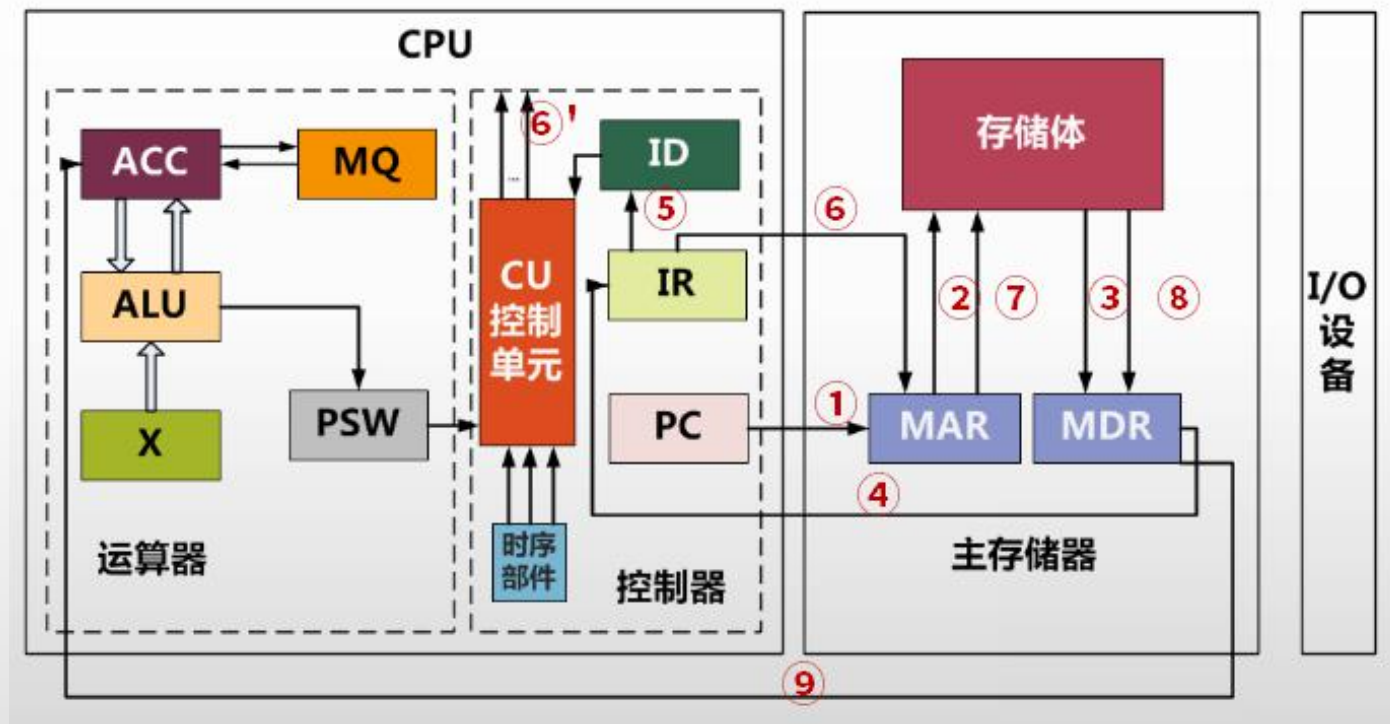
### 试题分析

多项式为101011，信息码右边补5个0后与多项式做模二除法（即  $11100011000000 \div 101011$ ），得余数即可。模2除法与算术除法类似，但每一位除的结果不影响其它位，即不向上一位借位，所以实际上就是异或（相同为0，不同为1）。

参考答案：B

# 指令

□ 指令周期：取出（解释）并执行一条指令所需的全部时间。



完成一条指令（一个指令周期）可以分为：取指周期、分析周期、执行周期

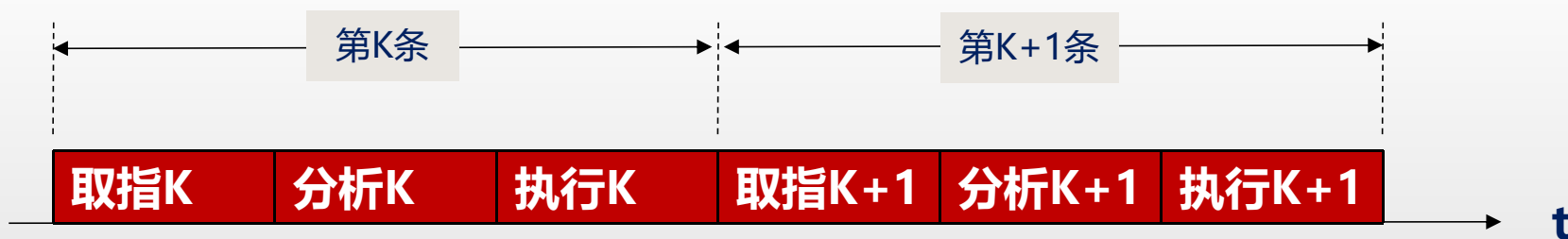
# 指令

- 指令流水技术：指令步骤的并行、提高处理器执行指令的效率。  
假设使用流水线将指令流的处理过程划分为取指、分析、执行三个并行处理的过程段。在这个流水线中，处理器有三个操作部件，同时对这三条指令进行加工，加快了程序的执行速度。几乎所有的高性能计算机都采用了指令流水线。



# 指令

**(1) 顺序方式。**各条机器指令之间顺序串行地执行，执行完一条指令后才取下一条指令。缺点是速度慢，机器各部件利用率低。



## 指令

(2) 重叠方式。在解释第K条指令的操作完成之前就可以开始解释第K+1条指令。





# 指令

## (3) 流水方式

流水线建立时间



# 指令

□ **流水线周期**：执行时间最长的一段

公式：① 理论公式：  $(t_1 + t_2 + \dots + t_k) + (n-1) * \Delta t$

② 实践公式：  $(k+n-1) * \Delta t$

例如：若流水线把一条指令分为取指、分析和执行三个部分，三部分的时间分别是取指2ns，分析2ns，执行1ns。那么流水线周期是多少？100条指令全部执行完毕需要的时间是多少？

解析：① 理论公式：  $(t_1 + t_2 + \dots + t_k) + (n-1) * \Delta t = (2+2+1) + (100-1) * 2 = 203$

② 实践公式：  $(k+n-1) * \Delta t = (3+100-1) * 2 = 204$

# 指令

□ **流水线的吞吐率和最大吞吐率：**吞吐率是指单位内流水线处理机流出的结果数。  
对指令而言，就是单位时间内执行的指令数。

$$TP = \frac{\text{指令条数}}{\text{流水线执行时间}}$$

$$TP_{\max} = \lim_{n \rightarrow \infty} \frac{n}{(k + n - 1)\Delta t} = \frac{1}{\Delta t}$$

□ **流水线加速比：**

$$S = \frac{\text{不使用流水线执行时间}}{\text{使用流水线执行时间}}$$

## 典型真题

某计算机系统采用 5 级流水线结构执行指令，设每条指令的执行由取指令( $2\Delta t$ )、分析指令( $1\Delta t$ )、取操作数( $3\Delta t$ )、运算( $1\Delta t$ )和写回结果( $2\Delta t$ ) 组成，并分别用5 个子部件完成，该流水线的最大吞吐率为 ( ) ；若连续向流水线输入10条指令，则该流水线的加速比为 ( ) 。

(1) A.  $\frac{1}{9\Delta t}$

B.  $\frac{1}{3\Delta t}$

C.  $\frac{1}{2\Delta t}$

D.  $\frac{1}{1\Delta t}$

(2) A. 1:10

B. 2:1

C. 5:2

D. 3:1

## 典型真题

### 试题分析

本题考查流水线计算。

流水线周期为： $3\Delta t$ 。

流水线的吞吐率为：指令条数/流水线执行时间。即：

$$n/(2\Delta t + 1\Delta t + 3\Delta t + 1\Delta t + 2\Delta t + (n-1)*3\Delta t) = n/(6\Delta t + 3n\Delta t)$$

流水线的最大吞吐率就是上面的式子中， $n$ 趋向于无穷大的结果。当 $n$ 趋向于无穷大时，上式的结果为： $1/3\Delta t$ 。所以应该选B。

流水线加速比=不用流水线的执行时间/使用流水线的执行时间

$$\begin{aligned} \text{10条指令不用流水线的执行时间} &= (2\Delta t + 1\Delta t + 3\Delta t + 1\Delta t + 2\Delta t) \\ &* 10 = 90\Delta t. \end{aligned}$$

## 典型真题

10条指令使用流水线的执行时间=  $(2\Delta t + 1\Delta t + 3\Delta t + 1\Delta t + 2\Delta t) + (10-1)$   
 $*3\Delta t = 36\Delta t$ 。

所以加速比为： $90\Delta t / 36\Delta t = 5:2$ 。

试题答案：B、C

# 高速缓冲存储器

◆解决CPU和主存之间的速度差异，避免CPU“空等”现象。



容量小  
速度高

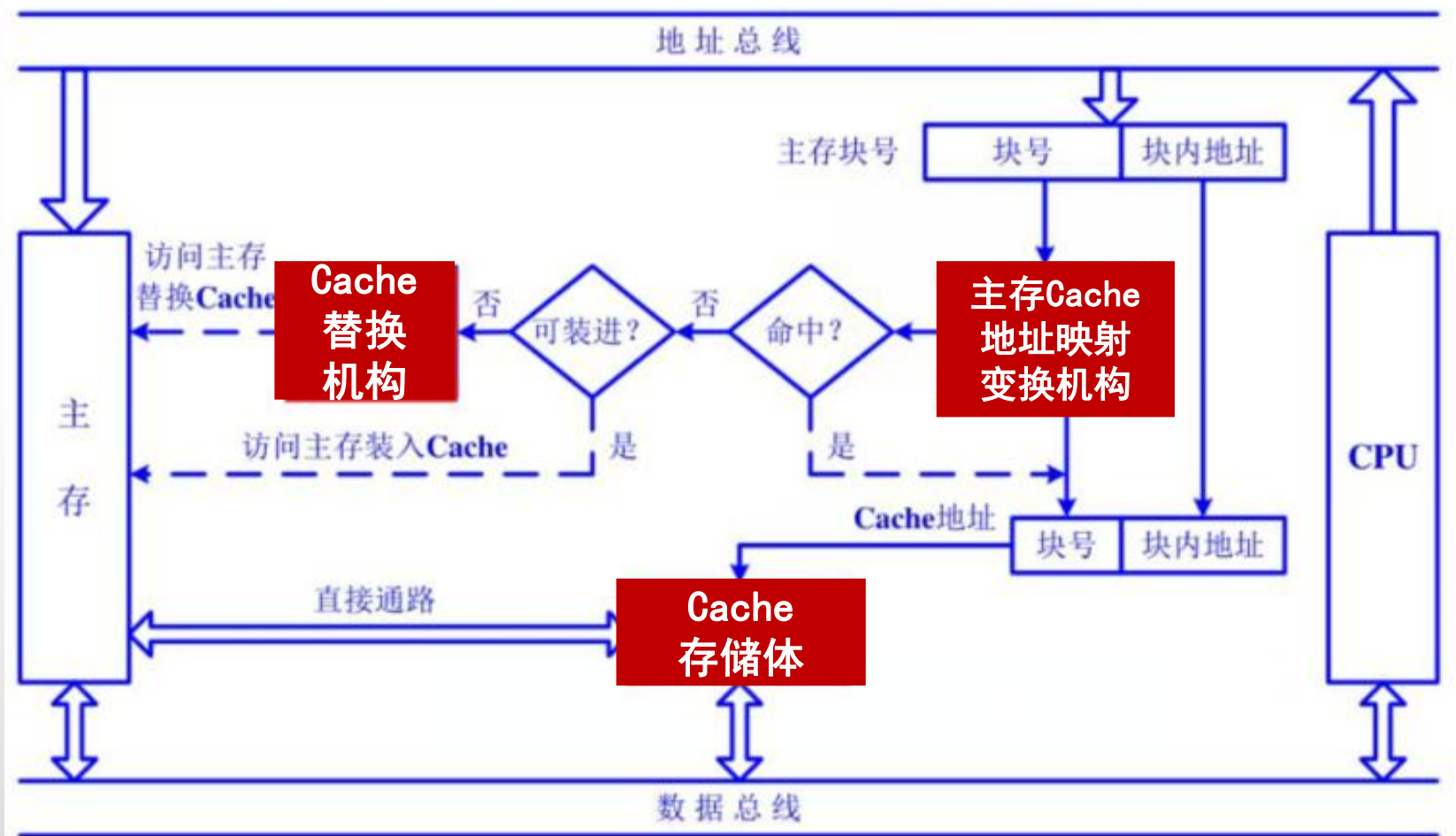
容量大  
速度低

局部性原理：

- 时间局部性
- 空间局部性



# 高速缓冲存储器



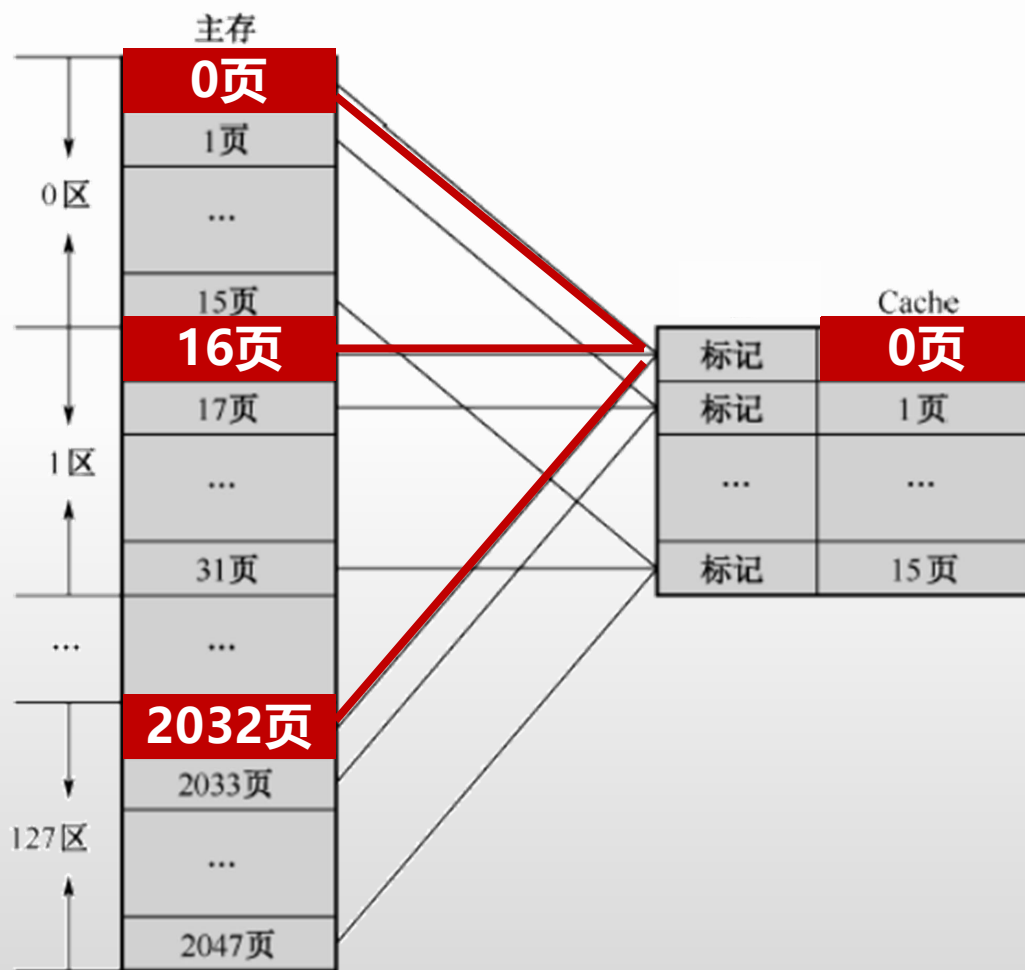


# 高速缓冲存储器

## Cache的映像方法

### ➤ 直接映像

直接映像方式的优点是地址变换很简单，缺点是不灵活，块冲突率高。

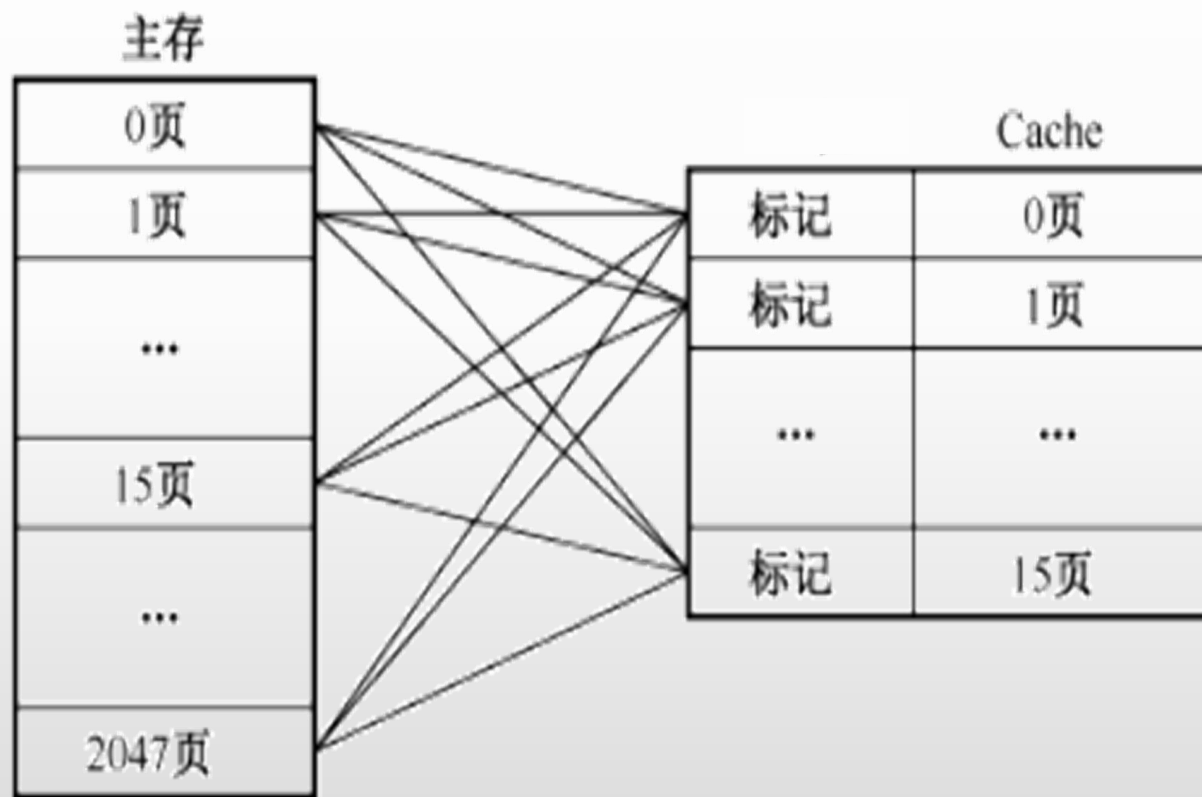


# 高速缓冲存储器

## Cache的映像方法

### ➤ 全相联映像

位置不受限制，十分灵活。其主要缺点是无法从主存块号中直接获得Cache的块号，变换比较复杂，速度比较慢。



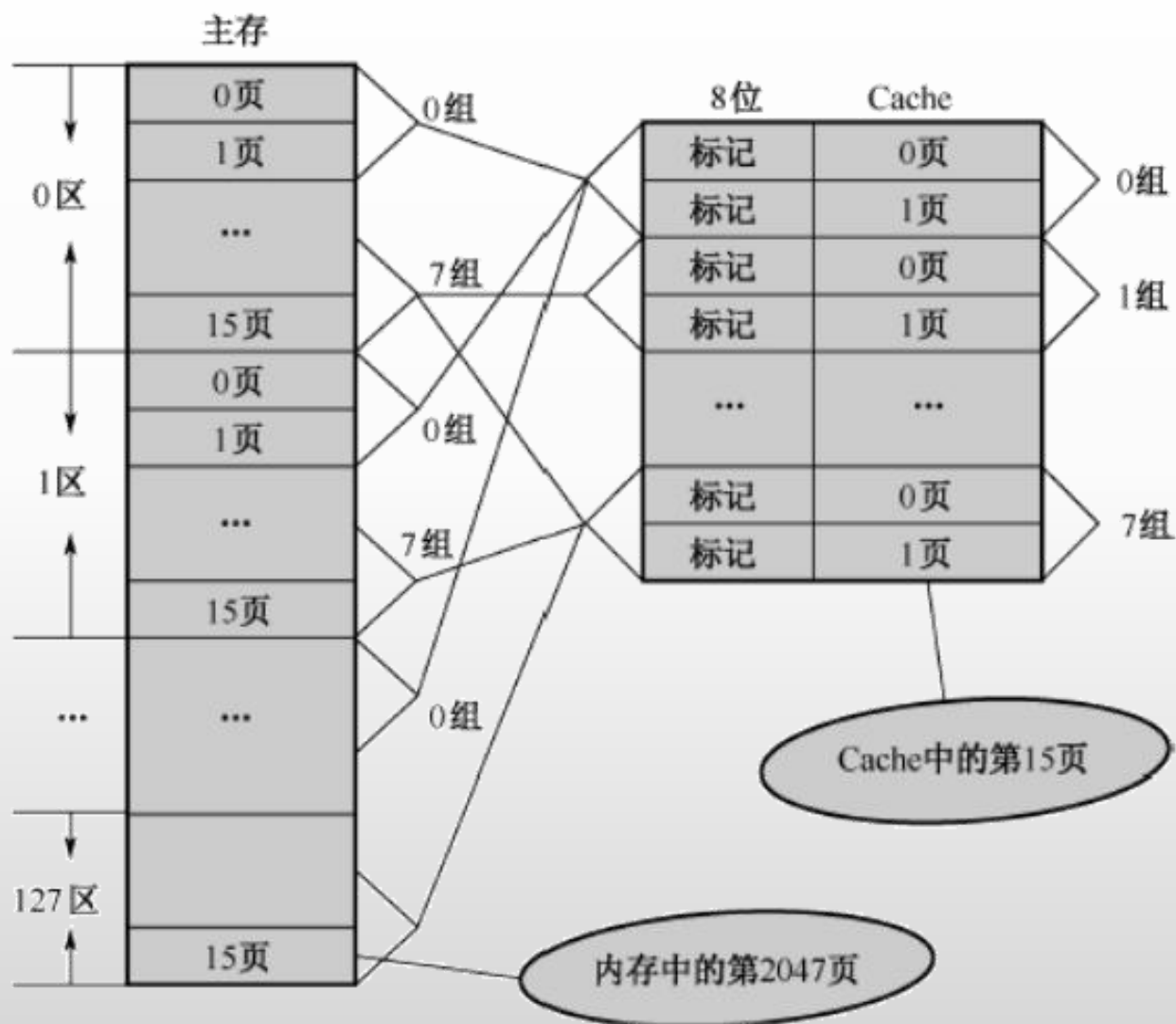
# 高速缓冲存储器

## Cache的映像方法

### ➤ 组相联映像

距离CPU较近 位置可以采用直接映像或者组相联映像

距离CPU较远的可以采用全相联映像



# 高速缓冲存储器

## cache的性能

CPU在访问内存时，首先判断所要访问的内容是否在Cache中，如果在，就称为“命中”，此时CPU直接从Cache中调用该内容；否则，就称为“不命中”，CPU只好去内存中调用所需的子程序或指令了。CPU不但可以直接从Cache中读出内容，也可以直接往其中写入内容。由于Cache的存取速率相当快，使得CPU的利用率大大提高，进而使整个系统的性能得以提升。

如果以 $H_c$ 为代表对Cache的访问命中率， $t_c$ 为Cache的存取时间， $t_m$ 为主存的访问时间，则Cache的平均访问时间 $t_a$ 为：

$$t_a = H_c t_c + (1 - H_c) t_m$$

# 高速缓冲存储器

因为cache的内容是部分主存内容的副本，应该与主存内容保持一致。而CPU对cache的写入更改了cache内容，如何与主存内容保持一致就有几种写操作工作方式可供选择，统称为写策略。

## 1. 写回法 (write--back)

当CPU对cache写命中时，只修改cache的内容不立即写入主存，只当此行被换出时才写回主存。这种策略使cache在CPU - 主存之间，不仅在读方向而且在写方向上都起到高速缓存作用。



# 高速缓冲存储器

## 2、写直达法 (write-through)

又称全写法，写透。是当cache写命中时，cache与主存同时发生写修改。

## 3、标记法

数据进入cache后，有效位置1，当cpu对该数据修改时，数据只写入主存并将该有效位置0。要从cache中读取数据时要测试其有效位，若为1则直接从cache中取数，否则从主存中取数。

# 高速缓冲存储器

**(1) 随机算法。**这是最简单的替换算法。随机法完全不管Cache块过去、现在及将来的使用情况，简单地根据一个随机数，选择一块替换掉。

**(2) 先进先出 (First In and First Out, FIFO) 算法。**按调入Cache的先后决定淘汰的顺序，即在需要更新时，将最先进入Cache的块作为被替换的块。这种方法要求为每块做一记录，记下它们进入Cache的先后次序。这种方法容易实现，而且系统开销小。其缺点是可能会把一些需要经常使用的程序块（如循环程序）替换掉。

**(3) 近期最少使用 (Least Recently Used, LRU) 算法。**LRU算法是把CPU近期最少使用的块作为被替换的块。这种替换方法需要随时记录Cache中各块的使用情况，以便确定哪个块是近期最少使用的块。LRU算法相对合理，但实现起来比较复杂，系统开销较大。通常需要对每一块设置一个称为“年龄计数器”的硬件或软件计数器，用以记录其被使用的情况。

## 高速缓冲存储器

**(4) 最不经常使用页置换(Least Frequently Used, LFU)算法。**要求在页置换时置换引用计数最小的页,因为经常使用的页应该有一个较大的引用次数。但是有些页在开始时使用次数很多,但以后就不再使用,这类页将会长时间留在内存中,因此可以将引用计数寄存器定时右移一位,形成指数衰减的平均使用次数。LFU的复杂度以及计数器规模都比LRU大,LRU只关注近期访问情况,而LFU会统计累计访问次数作为淘汰的依据。**该算法计数器位数多,实现困难。**



# 磁盘存储器

常用的外存有磁带存储器、硬盘存储器、磁盘阵列和光盘存储器。磁盘标称的容量是格式化容量，计算公式如下：

$$\text{存储容量} = n \times t \times s \times b$$

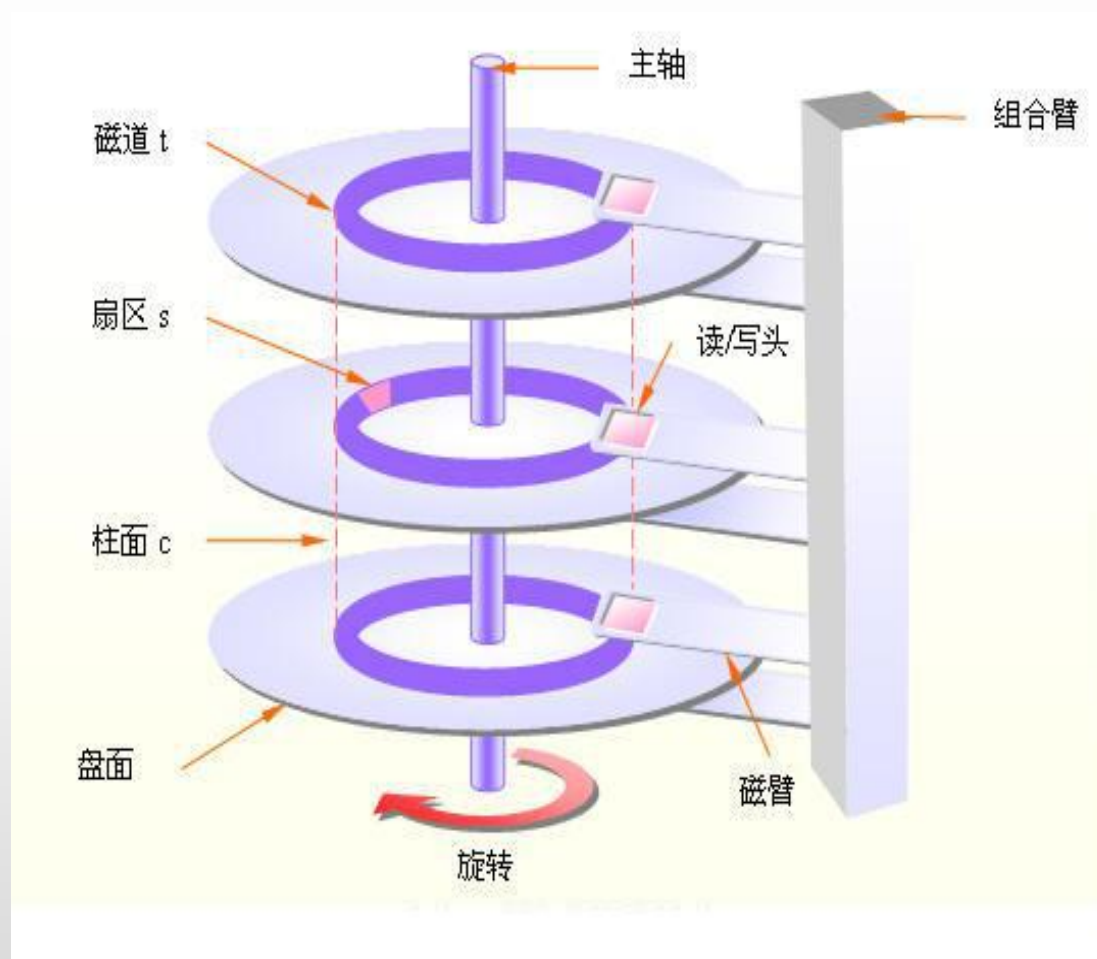
其中：n为保存数据的总记录面数，  
t为每面磁道数，s为每道的扇区数，  
b为每个扇区存储的字节数。

硬盘存取时间：

寻道时间+等待时间+读/写时间

其中读/写时间可忽略不计，

平均寻道时间+平均等待时间



## 典型真题

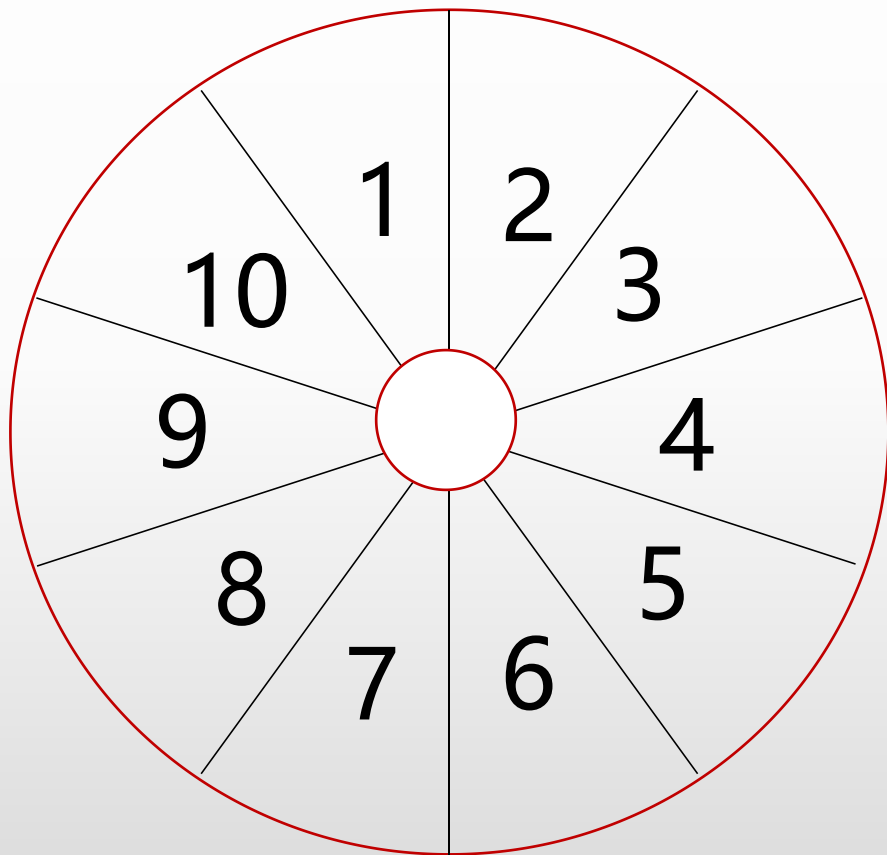
●在磁盘上存储数据的排列方式会影响 I/O 服务的总时间。假设每磁道划分成 10 个物理块，每块存放 1 个逻辑记录。逻辑记录 R1, R2, ..., R10 存放在同一个磁道上，记录的安排顺序如下表所示；

物理块	1	2	3	4	5	6	7	8	9	10
逻辑记录	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10

假定磁盘的旋转速度为 30ms/周，磁头当前处在 R1 的开始处。若系统顺序处理这些记录，使用单缓冲区，每个记录处理时间为 6ms，则处理这 10 个记录的最长时间为 (1)；若对信息存储进行优化分布后，处理 10 个记录的最少时间为(2)。

- (1) A. 189ms    B. 208ms    C. 289ms    D. 306ms  
(2) A. 60 ms    B. 90 ms    C. 109ms    D. 180ms

## 典型真题



## 典型真题

### 试题分析

系统读记录的时间为 $30/10=3\text{ms}$ 。对第一种情况：系统读出并处理记录R1之后，将转到记录R4的开始处，所以为了读出记录R2，磁盘必须再转一圈，需要 $3\text{ms}$ (读记录)加 $30\text{ms}$ (转一圈)的时间。这样，处理10个记录的总时间应为处理前9个记录(即R1, R2, ..., R9)的总时间再加上读R10和处理时间( $9\times 33\text{ms}+9\text{ms}=306\text{ms}$ )。

若对信息进行分布优化的结果对应关系所示：

物理块 1 2 3 4 5 6 7 8 9 10

逻辑记录R1 R8 R5 R2 R9 R6 R3 R10 R7 R4

记录的开始处，立即就可以读出并处理，因此处理10个记录的总时间为：

$10\times(3\text{ms}(\text{读记录})+6\text{ms}(\text{处理记录}))=10\times 9\text{ms}=90\text{ms}$

参考答案： (1) D (2) B

## 典型真题

在磁盘调度管理中，应先进行移臂调度，再进行旋转调度。假设磁盘移动臂位于21号柱面上，进程的请求序列如下表所示。如果采用最短移臂调度算法，那么系统的响应序列应为（ ）。

请求序列	柱面号	磁头号	扇区号
①	17	8	9
②	23	6	3
③	23	9	6
④	32	10	5
⑤	17	8	4
⑥	32	3	10
⑦	17	7	9
⑧	23	10	4
⑨	38	10	8

A. ②⑧③④⑤①⑦⑥⑨

B. ②③⑧④⑥⑨①⑤⑦

C. ①②③④⑤⑥⑦⑧⑨

D. ②⑧③⑤⑦①④⑥⑨

## 典型真题

### 试题分析

根据题干要求，先进行移臂调度，找到对应柱面，然后进行旋转调度，找到对应磁头和扇区。

由表可知①⑤⑦在17柱面 ( $21-17=4$ )，②③⑧在23柱面 ( $23-21=2$ )，④⑥在32柱面 ( $32-21=9$ )。因此按最短移臂算法，应该是23柱面17柱面32柱面38柱面，只有D项满足。

参考答案：D



# 计算机系统结构的分类

## Flynn 分类

1966年, Michael. J. Flynn 提出根据指令流、数据流的多倍性特征对计算机系统进行分类 (通常称为 Flynn 分类法), 有关定义如下。

(1) 指令流: 指机器执行的指令序列;

(2) 数据流: 指由指令流调用的数据序列, 包括输入数据和中间结果, 但不包括输出数据。



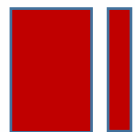
# 计算机系统结构的分类

Flynn根据不同的指令流-数据流组织方式，把计算机系统分成以下四类。

(1) 单指令流单数据流 (Single Instruction stream and Single Data stream, SISD) : SISD 其实就是传统的顺序执行的单处理器计算机，其指令部件每次只对一条指令进行译码，并只对一个操作部件分配数据。

(2) 单指令流多数据流 (Single Instruction stream and Multiple Data stream, SIMD) : SIMD 以并行处理机（矩阵处理机）为代表，并行处理机包括多个重复的处理单元，由单一指令部件控制，按照同一指令流的要求为它们分配各自所需的不同数据。





# 计算机系统结构的分类

**(3) 多指令流单数据流 (Multiple Instruction stream and Single Data stream, MISD) :** MISD 具有 $n$ 个处理单元, 按  $n$  条不同指令的要求对同一数据流及其中间结果进行不同的处理。一个处理单元的输出又作为另一个处理单元的输入。这类系统实际上很少见到。

**(4) 多指令流多数据流 (Multiple Instruction stream and Multiple Data stream, MIMD) :** MIMD 是指能实现作业、任务、指令等各级全面并行的多机系统。如多核处理器、多处理机属于 MIMD。

# 指令系统

## 复杂指令系统CISC的特点

- (1) 指令数量众多。指令系统拥有大量的指令，通常有 100 ~ 250 条。
- (2) 指令使用频率相差悬殊。最常使用的是一些比较简单的指令，仅占指令总数的20%，但在程序中出现的频率却占80%。而大部分复杂指令却很少使用。
- (3) 支持很多种寻址方式。支持的寻址方式通常为 5 ~ 20 种。
- (4) 变长的指令。指令长度不是固定的，变长的指令增加指令译码电路的复杂性。

# 指令系统

**(5) 指令可以对主存单元中的数据直接进行处理。典型的 CISC 通常都有指令能够直接对主存单元中的数据进行处理，其执行速度较慢。**

**(6) 以微程序控制为主。CISC 的指令系统很复杂，难以用硬布线逻辑（组合逻辑）电路实现控制器，通常采用微程序控制。**

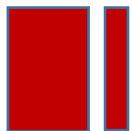
# 指令系统

## 精简指令系统RISC的特点

(1) 指令数量少。优先选取使用频率最高的一些简单指令和一些常用指令，避免使用复杂指令。只提供了 LOAD（从存储器中读数）和 STORE（把数据写入存储器）两条指令对存储器操作，其余所有的操作都在 CPU 的寄存器之间进行。

(2) 指令的寻址方式少。通常只支持寄存器寻址方式、立即数寻址方式和相对寻址方式。

(3) 指令长度固定，指令格式种类少。因为 RISC 指令数量少、格式少、相对简单，其指令长度固定，指令之间各字段的划分比较一致，译码相对容易。



## 指令系统

(4) 以硬布线逻辑控制为主。为了提高操作的执行速度，通常采用硬布线逻辑（组合逻辑）来构建控制器。

(5) 单周期指令执行，采用流水线技术。因为简化了指令系统，很容易利用流水线技术，使得大部分指令都能在一个机器周期内完成。少数指令可能会需要多周期，例如，LOAD/STORE 指令因为需要访问存储器，其执行时间就会长一些。

(6) 优化的编译器：RISC 的精简指令集使编译工作简单化。因为指令长度固定、格式少、寻址方式少，编译时不必在具有相似功能的许多指令中进行选择，也不必为寻址方式的选择而费心，同时易于实现优化，从而可以生成高效率执行的机器代码。

(7) CPU 中的通用寄存器数量多，一般在 32 个以上，有的可达上千个。

## 典型真题

**RISC(精简指令系统计算机) 的特点不包括：（ ）。**

- A. 指令长度固定，指令种类尽量少**
- B. 寻址方式尽量丰富，指令功能尽可能强**
- C. 增加寄存器数目，以减少访存次数**
- D. 用硬布线电路实现指令解码，以尽快完成指令译码**

# 典型真题

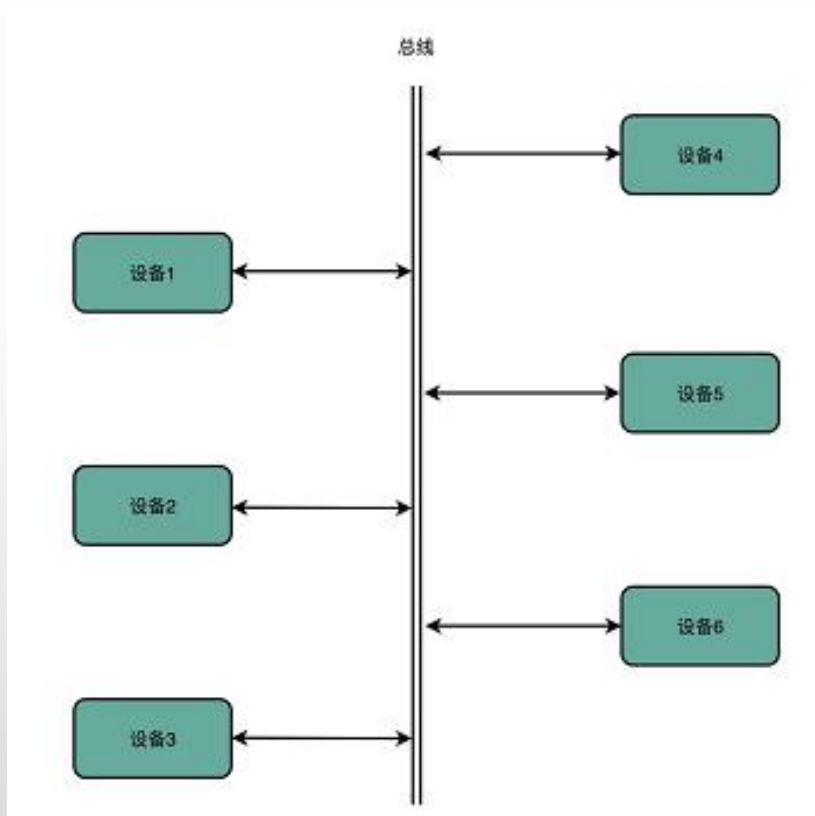
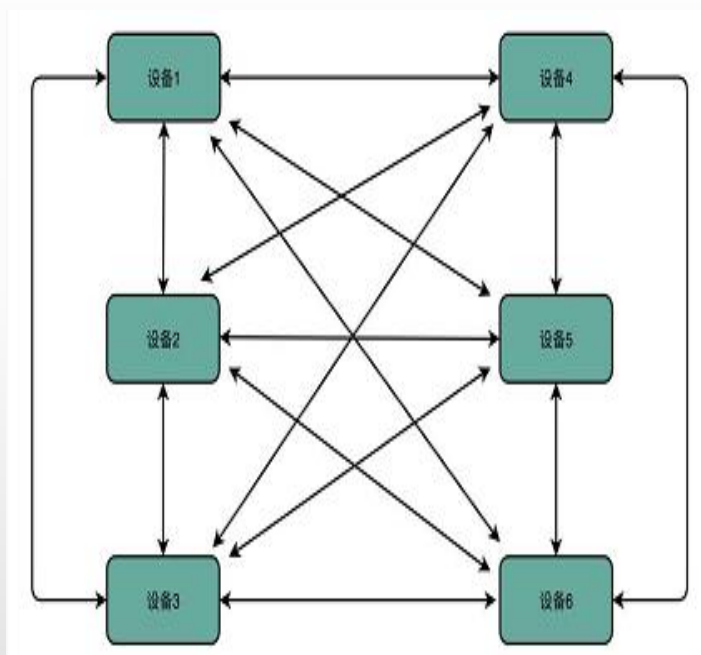
## 试题分析

指令系统类型	指令	寻址方式	实现方式	其它
CISC (复杂)	数量多, 使用频率差别大, 可变长格式	支持多种	微程序控制技术 (微码)	研制周期长
RISC (精简)	数量少, 使用频率接近, 定长格式, 大部分为单周期指令, 操作寄存器, 只有 Load/Store操作内存	支持方式少	增加了通用寄存器; 硬布线逻辑控制为; 适合采用流水线	优化编译, 有效支持高级语言

试题答案: B

# 总线

总线是一组能为多个部件分时共享的公共信息传送线路。





# 总线

按总线相对于 CPU 或其他芯片的位置可分为内部总线和外部总线两种。

按总线功能来划分，又可分为地址总线、数据总线、控制总线三类，地址总线用来传送地址信息，数据总线用来传送数据信息，控制总线用来传送各种控制信号。

按照总线中数据线的多少，可分为并行总线和串行总线。

名称	数据线	特点	应用
并行总线	多条双向数据线	有传输延迟，适合近距离连接。	系统总线 (计算机各部件)
串行总线	一条双向数据线 或两条单向数据线	速率不高，但适合长距离连接。	通信总线 (计算机之间或 计算机与其他系统间)

## 典型真题

下列说法中正确的是（12）。

- A. 半双工总线只在一个方向上传输信息，全双工总线可在两个方向上轮流传输信息
- B. 半双工总线只在一个方向上传输信息，全双工总线可在两个方向上同时传输信息
- C. 半双工总线可在两个方向上轮流传输信息，全双工总线可在两个方向上同时传输信息
- D. 半双工总线可在两个方向上同时传输信息，全双工总线可在两个方向上轮流传输信息

## 典型真题

### 试题分析

对端到端通信总线的信号传输方向与方式的分类定义如下：

单工是指A只能发信号，而B只能接收信号，通信是单向的。

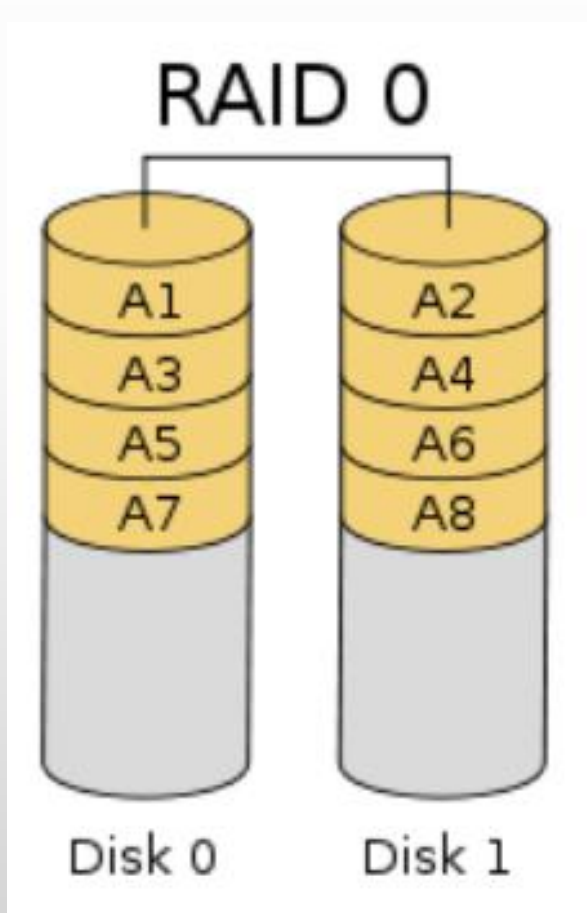
半双工是指A能发信号给B，B也能发信号给A，但这两个过程不能同时进行。

全双工比半双工又进了一步，在A给B发信号的同时，B也可以给A发信号，这两个过程可以同时进行互不影响。

参考答案：C

## 磁盘阵列

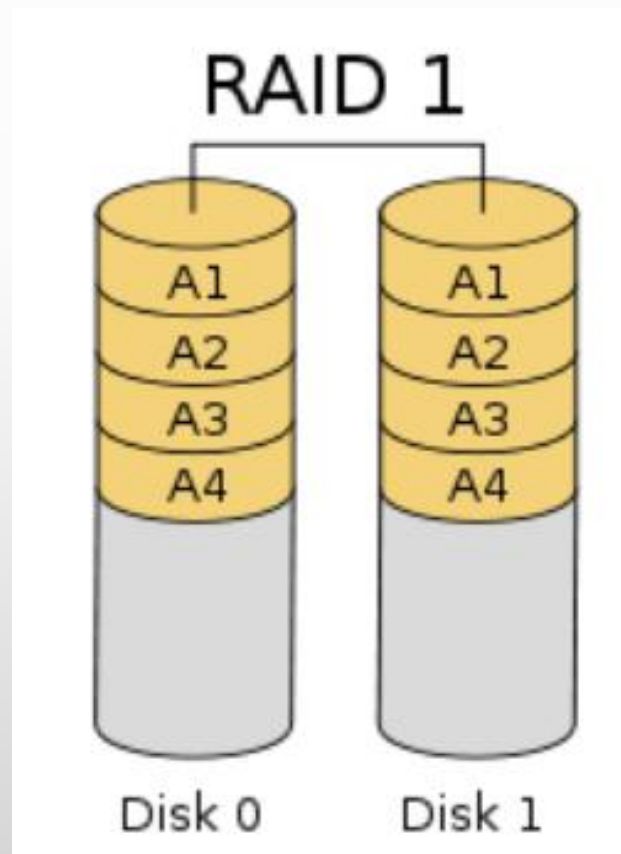
① RAID 0（无冗余和无校验的数据分块）代表了所有RAID级别中最高的存储性能。RAID 0提高存储性能的原理是把连续的数据分散到多个磁盘上存取，这样，系统有数据请求就可以被多个磁盘并行的执行，每个磁盘执行属于它自己的那部分数据请求。这种数据上的并行操作可以充分利用总线的带宽，显著提高磁盘整体存取性能。



## 磁盘阵列

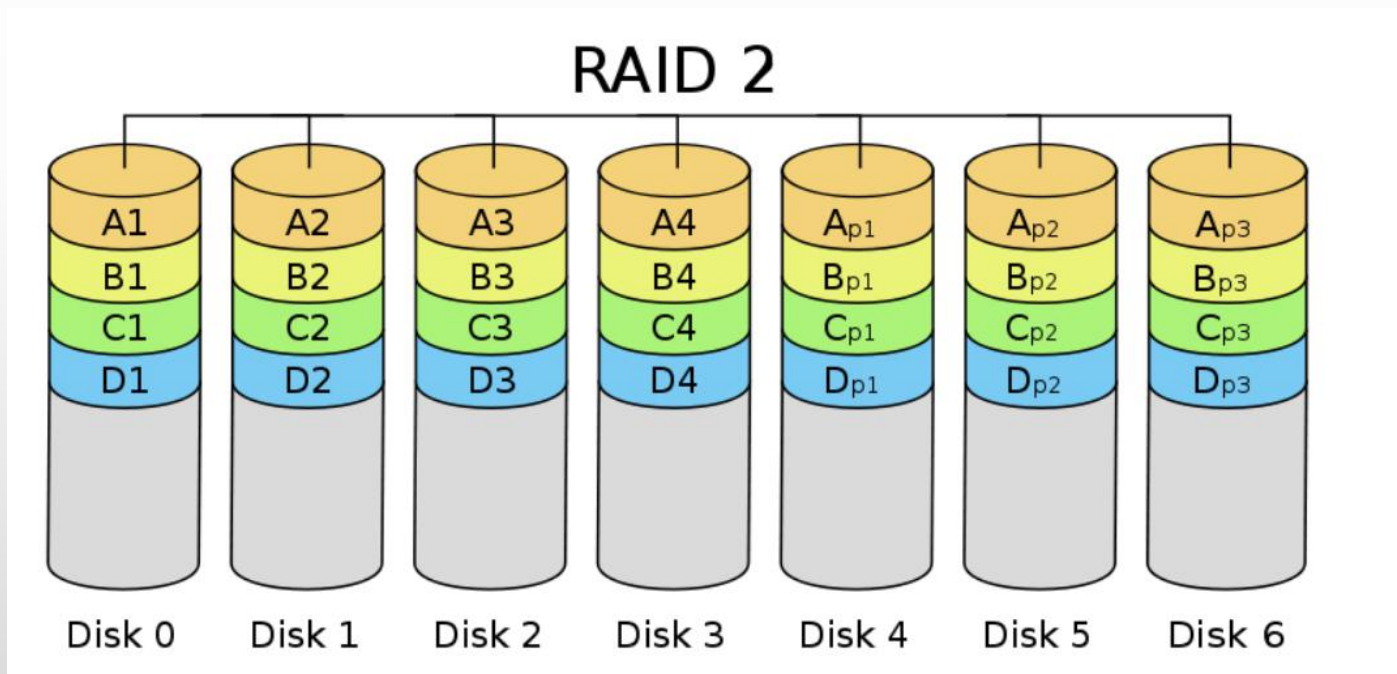
### ② RAID 1（磁盘镜像阵列）

RAID 1 称为镜像，它将数据完全一致地分别写到工作磁盘和镜像磁盘，  
磁盘空间利用率为50%。



## 磁盘阵列

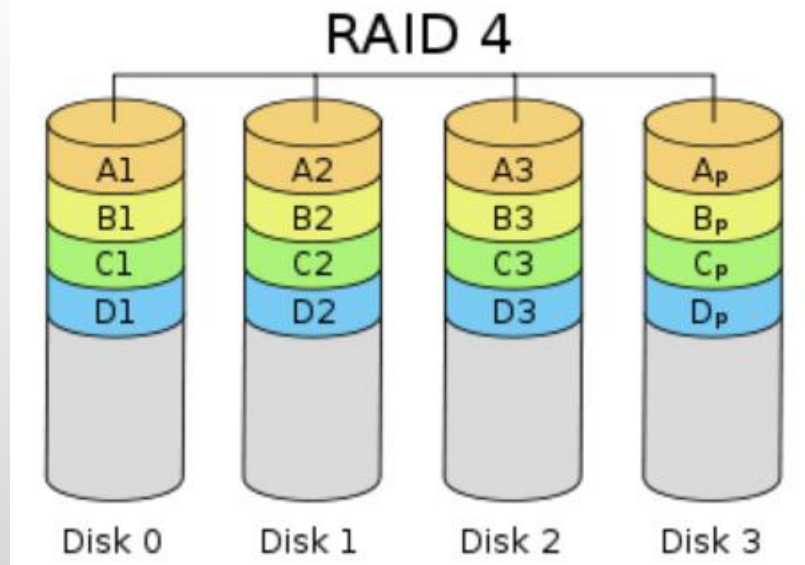
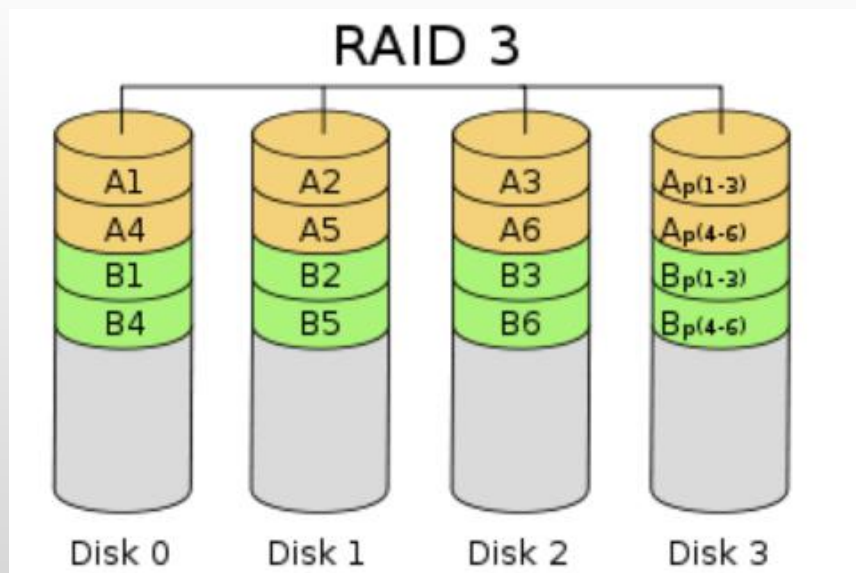
③ RAID 2（采用纠错海明码的磁盘阵列）：采用了海明码纠错技术，用户需增加校验盘来提供单纠错和双验错功能。对数据的访问涉及阵列中的每一个盘。大量数据传输时I/O性能较高，但不利于小批量数据传输。实际应用中很少使用。



## 磁盘阵列

④ RAID 3和RAID 4（采用奇偶校验码的磁盘阵列）：把奇偶校验码存放在一个独立的校验盘上。如果有一个盘失效，其上的数据可以通过对其他盘上的数据进行异或运算得到。读数据很快，但因为写入数据时要计算校验位，速度较慢。

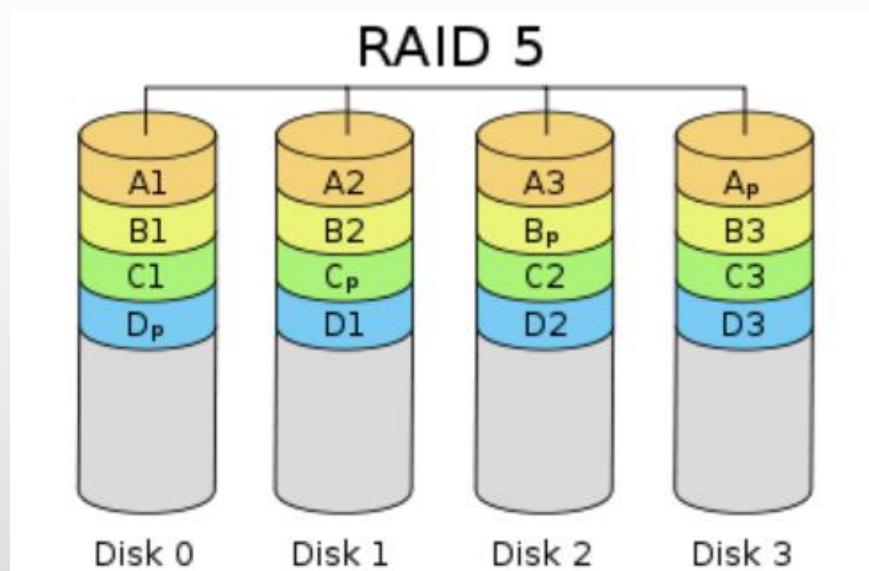
RAID 3采用位交叉奇偶校验，RAID 4采用块交叉奇偶校验码。RAID 3适用于大型文件且I/O需求不频繁的应用，RAID 4适用于大型文件的读取。



## 磁盘阵列

⑤ RAID5（无独立校验盘的奇偶校验码的磁盘阵列）：无独立校验盘，校验信息分布在组内所有盘上，对于大批量和小批量数据的读写性能都很好，适用于I/O需求频繁的应用。

当有N块阵列盘时，用户空间为N-1块盘容量。





## 典型真题

- 假如有3块容量是80G的硬盘做RAID 5阵列，则这个RAID 5的容量是（ ）；而如果有2块80G的盘和1块40G的盘，此时RAID 5的容量是（ ）。
- A. 240G   B. 160G   C. 80G   D. 40G
- A. 40G   B. 80G   C. 160G   D. 200G

## 典型真题

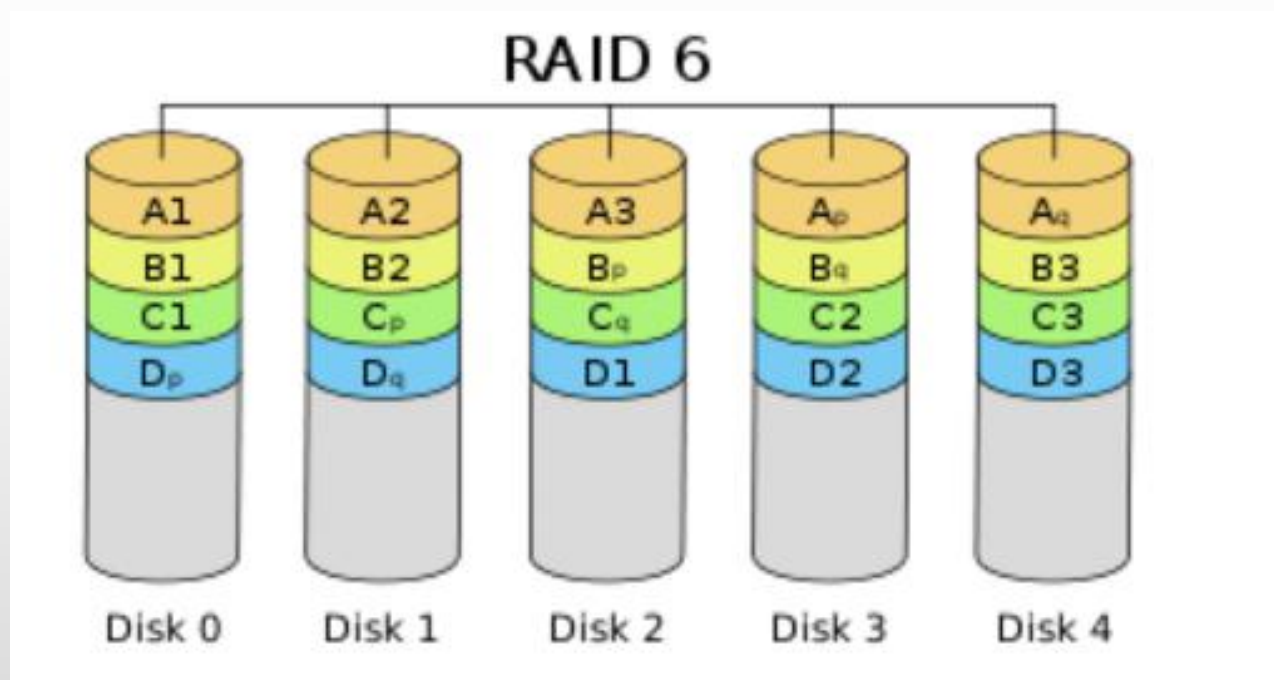
### 试题分析

共有3块80G的硬盘做RAID 5，则总容量为 $(3-1) \times 80 = 160\text{G}$ ；如果有2块80G的盘和1块40G的盘，则以较小的盘的容量为计算方式，总容量为 $(3-1) \times 40 = 80\text{G}$ 。

参考答案： B、 B

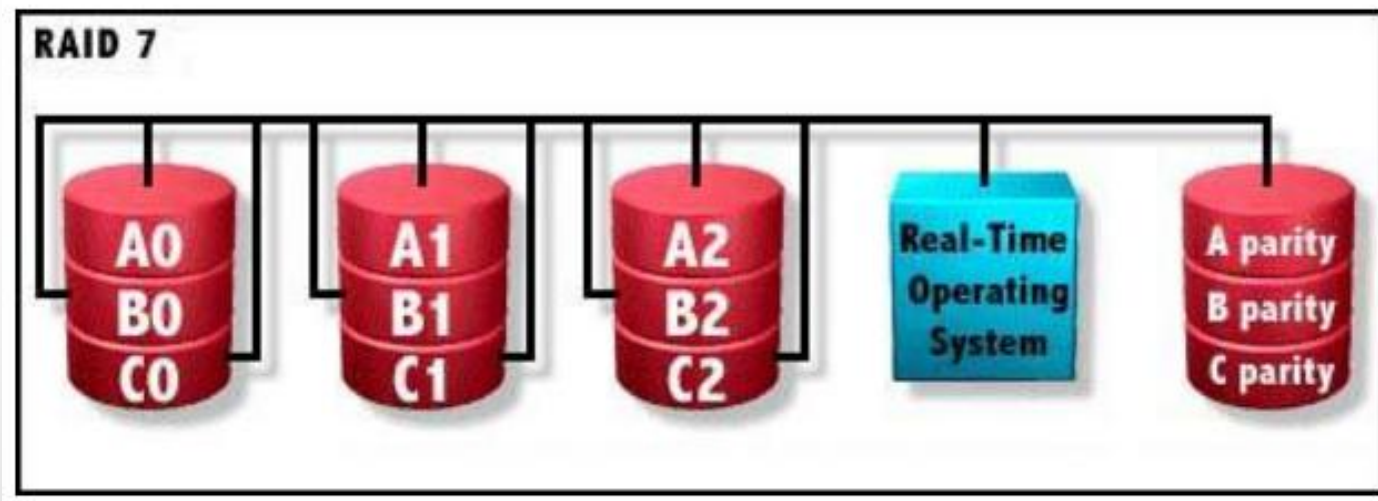
## 磁盘阵列

⑥ RAID6（独立的数据硬盘与两个独立的分布式校验方案）：RAID 6技术是在RAID 5基础上，为了进一步加强数据保护而设计的一种RAID方式，实际上是一种扩展RAID 5等级。当有N块阵列盘时，用户空间为N-2块盘容量。



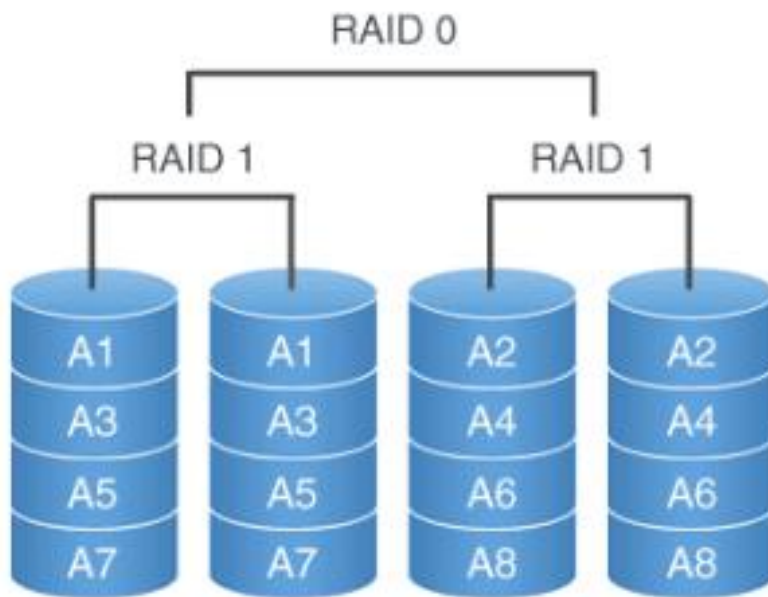
## 磁盘阵列

⑦ RAID7（最优化的异步高I/O速率和高数据传输率）：RAID 7完全可以理解为一个独立存储计算机，它自身带有操作系统和管理工具，完全可以独立运行。



## 磁盘阵列

⑧ RAID10（最可靠与高性能）：RAID 1+0 也被称为RAID 10标准，实际是将RAID 1和RAID 0标准结合的产物。RAID 1是一个冗余的备份阵列，而RAID 0负责数据读写的阵列。由于利用了RAID 0极高的读写效率和RAID 1较高的数据保护和恢复能力，使RAID 10成为了一种性价比较高的等级。



# 技术成就梦想