计算机组织结构

书面习题选讲及相关知识点回顾

刘博涵

2023年12月12日

南京大学
NANJING UNIVERSITY

1、一个测试程序在一个40MHz的处理器上运行，其目标代码有100000条指令，由如下各类指令及其时钟周期计数混合组成：

指令类型	指令计数	时钟周期计数
整数算术	45000	1
数据传送	32000	2
浮点数	15000	2
控制传送	8000	2

请确定这个程序的有效CPI（结果：小数点后2位）、MIPS速率（结果：小数点后1位）和执行时间（单位：毫秒；结果：小数点后2位）。

回顾: CPU性能

- 指令执行
  - 处理器由时钟驱动，时钟具有固定的频率f，或等价为固定的时钟周期t
  - 如果用 CPI_i 来表示指定类型i所需要的周期数，用 I_i 表示在某一给定程序中所执行的i类指令的条数
  - 则我们可以计算整个CPI如下：

$$CPI = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}, \quad I_c = \sum_{i=1}^{n} I_i$$

CPU执行一个给定程序时，平均执行每条指令的周期数

执行一个给定程序的处理时间表示为：

$$T = I_c \times CPI \times t$$

$$T = I_c \times [p + (m \times k)] \times t$$

存储器周期时间和处理器周期时间之比

译码和执行指令　存储器访问次数

在处理器和存储器之间传输数据

- 每秒百万条指令（MIPS）：

$$MIPS = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

1、一个测试程序在一个40MHz的处理器上运行，其目标代码有100000条指令，由如下各类指令及其时钟周期计数混合组成：

指令类型	指令计数	时钟周期计数
整数算术	45000	1
数据传送	32000	2
浮点数	15000	2
控制传送	8000	2

请确定这个程序的有效CPI（结果：小数点后2位）、MIPS速率（结果：小数点后1位）和执行时间（单位：毫秒；结果：小数点后2位）。

$$CPI = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}, I_c = \sum_{i=1}^{n} I_i$$

$$CPI = \frac{1 \times 45000 + 2 \times 32000 + 2 \times 15000 + 2 \times 8000}{45000 + 32000 + 15000 + 8000} = 1.55 \text{ 个周期/条指令}$$

$$MIPS = \frac{f}{CPI \times 10^6} = \frac{40 \times 10^6}{1.55 \times 10^6} \approx 25.8 \text{ 百万条指令/秒}$$

$$T = I_c \times CPI \times t = (45000 + 32000 + 15000 + 8000) \times 1.55 \times \frac{1}{40 \times 10^6} \approx 3.88 \text{ 毫秒}$$

2、假设在三台计算机上执行了4个测试程序，结果如下：

	计算机A	计算机B	计算机C
程序1	1	10	20
程序2	1000	100	20
程序3	500	1000	50
程序4	100	800	100

表中表示的每个程序执行10^8条指令所用的执行时间（单位：秒）。请计算MIPS算术平均值和调和平均值（结果：小数点后3位），并对计算机的性能排序。

回顾: CPU性能

算术均值:

$$R_A = \frac{1}{n}\sum_{i=1}^{n} R_i$$

- **适用场景**：适合度量所有**测量值的总和是一个有意义的数值**，如：系统执行时间
- **原因**：计算方便，适用性较强。但容易受极端值影响

调和均值:

$$R_H = \frac{n}{\sum_{i=1}^{n}\frac{1}{R_i}}$$

- **适用场景**：适用于以**系统的执行速率**来衡量系统价值
- **原因**：速率的总和没有意义，而速率的调和均值与总时间成反比，调合均值的比较更有实际意义

几何均值:

$$R_G = (\prod_{i=1}^{n} R_i)^{\frac{1}{n}}$$

- **适用场景**：适用于衡量计算机的**相对性能**，如：基准测试
- **原因**：几何均值对所有数据赋予了相同的权重，在度量机器的相对性能时提供了一致的结果

每个程序的执行速率 → $R_i = \frac{Z}{t_i}$ ← 操作数，假设每个程序的相同

每个程序的执行时间 ←

2、假设在三台计算机上执行了4个测试程序，结果如下：

	计算机A	计算机B	计算机C
程序1	1	10	20
程序2	1000	100	20
程序3	500	1000	50
程序4	100	800	100

表中表示的每个程序执行10^8条指令所用的执行时间（单位：秒）。请计算MIPS算术平均值和调和平均值（结果：小数点后3位），并对计算机的性能排序。

$$MIPS = \frac{I_c}{T \times 10^6}$$

	计算机A	计算机B	计算机C
程序1	100	10	5
程序2	0.1	1	5
程序3	0.2	0.1	2
程序4	1	0.125	1

$$MIPS_A = \frac{1}{n}\sum_{i=1}^{n} MIPS_i$$

$$MIPS_H == \frac{n}{\sum_{i=1}^{n} \frac{1}{MIPS_i}}$$

	计算机A	计算机B	计算机C
算数均值	25.325	2.806	3.25
调和均值	0.250	0.209	2.105

算数均值：计算机性能排序为：A > C > B

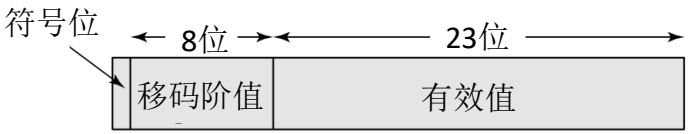
调和均值：计算机性能排序为：C > A > B

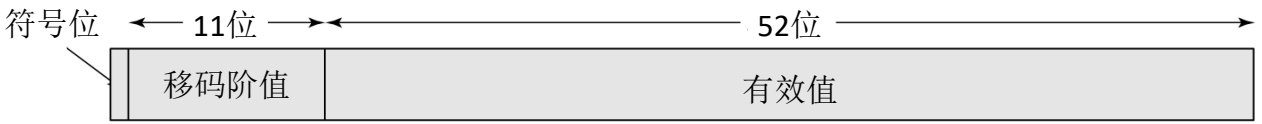
3、假定变量int i = 1234567890、float f = 1.23456789e9，sizeof(int)=4，判断以下表达式的结果（True / False）

- i == (int)(float)f
- i == (int)(double)f
- i == (float)(int)f
- i == (float)(double)f

回顾: IEEE 754 标准

- **定义32位的单精度和64位的双精度两种格式**





- **定义两种拓展格式**
  - 扩展单精度浮点格式（≥43 位，不常用）。
  - 扩展双精度浮点格式（≥79 位，一般情况下，Intel x86 结构的计算机采用的是 80 位，而 SPARC 结构的计算机采用的是 128 位）。

3、假定变量int i = 1234567890、float f = 1.23456789e9，sizeof(int)=4，判断以下表达式的结果（True / False）

- **i == (int)(float)f**

  假：float即IEEE754单精度，尾数为(23+1)位，$10^7 < 2^{24} < 10^8$，所以理论上对应十进制有效位数为7位，而 i 有9位十进制有效位。

- **i == (int)(double)f**

  假：i ==(int)(double) i 为真，但f已经损失了精度，double(f)并不能还原精度

- **i == (float)(int)f**

  假：理由同上。还需要注意，将一个大于$(2^{31}-1)$的float数转为int也将为假

- **i == (float)(double)f**

  假：理由同上。

精度考虑：数据类型

- X为int型，F为float型，D为double型，判断下列关系是否恒为真：
  - 否 • X == (int)(float) X
  - 是 • X == (int)(double) X
  - 是 • F == (float)(double) F
  - 否 • D == (float) D
  - 否 • X == (float) X
  - 否 • X * X >= 0
  - 是 • F * F >= 0
  - 否 • (D+F) – D == F
  - 否 • (D+F) – F == D
  - 否 • 1/3 == 1/3.0

4、某计算机在信息传输中采用基于偶校验的海明码，对每个字节生成校验位。假设所传输信息的十六进制表示为8F3CAB96H，且将信息与校验码按照故障字的顺序排列后一起传输。如果传输中没有发生任何错误，写出所接收到信息（含校验码）的十六进制表示。

示例：FFFFFFFFFFFFH

回顾: 海明码

- **数据位划分**
  - 假定数据位为8位 $D = D_8 \ldots D_2 D_1$, 校验码为4位$C = C_4 C_3 C_2 C_1$
  - 数据位/校验码与故障字的关系

故障字	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
数据位	D_8	D_7	D_6	D_5		D_4	D_3	D_2		D_1		
校验位					C_4				C_3		C_2	C_1

  - 数据位划分

$$C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

$$C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$$

4、某计算机在信息传输中采用基于偶校验的海明码，对每个字节生成校验位。假设所传输信息的十六进制表示为8F3CAB96H，且将信息与校验码按照故障字的顺序排列后一起传输。如果传输中没有发生任何错误，写出所接收到信息（含校验码）的十六进制表示。

示例：FFFFFFFFFFFFH

根据海明码的计算规则：

$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$

$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$

$C3 = D2 \oplus D3 \oplus D4 \oplus D8$

$C4 = D5 \oplus D6 \oplus D7 \oplus D8$

对8F3CAB96H各个字节计算出校验码：

8FH = 1000 1111B，校验码（C4C3C2C1）为1011

3CH = 0011 1100B，校验码（C4C3C2C1）为0010

ABH = 1010 1011B，校验码（C4C3C2C1）为0111

96H = 1001 0110B，校验码（C4C3C2C1）为0110

所以，将信息和校验码按照故障字的顺序排列后的二进制表示为：

1000 1111 0111 0011 0110 0010 1010 0101 1111 1001 0011 1010

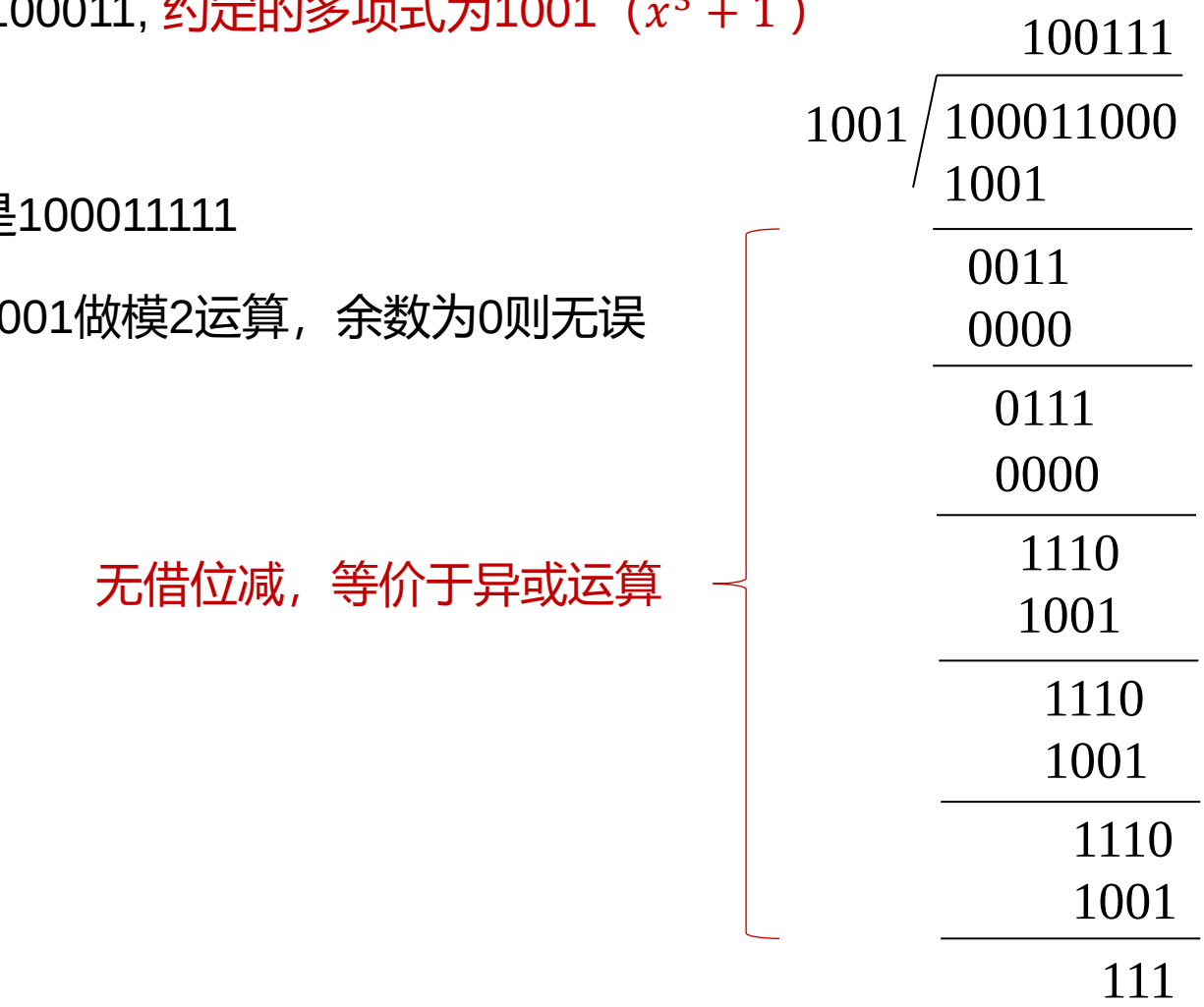
十六进制表示为：**8F7362A5F93AH**

5、假设要传送的数据信息为 100011，若约定的生成多项式位$G(x) = x^3 + 1$。如果传输中没有出现错误，接收到的信息是什么?

示例：000000000

回顾：循环冗余校验

- 假设数据是 100011, 约定的多项式为1001（x^3+1）
- 校验码是111
- 发出的数据是100011111
- 接收方也对1001做模2运算，余数为0则无误



5、假设要传送的数据信息为 100011，若约定的生成多项式位$G(x)=x^3+1$。如果传输中没有出现错误，接收到的信息是什么?

示例：000000000

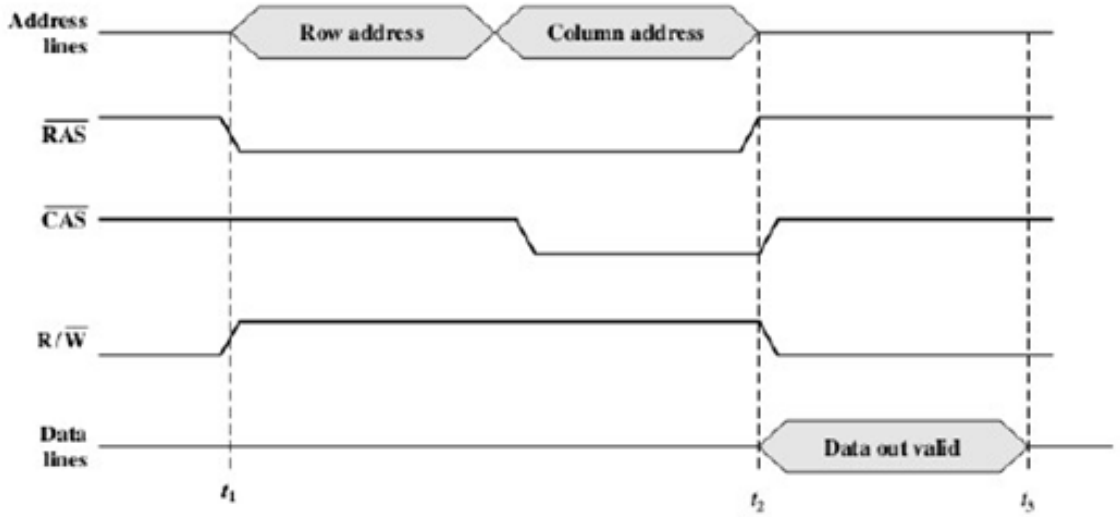
生成多项式$G(x)$为1001，所以将数据左移3位后，进行模2除法：

```
          100111
    ------------
1001/ 100011000
      1001
      ----
       0011
       0000
       ----
        0111
        0000
        ----
         1110
         1001
         ----
          1110
          1001
          ----
           1110
           1001
           ----
            111
```

校验码为**111**。

如果传输中没有出现错误，接收到的信息是：**100011111**。

6、假设采用分散式刷新，下图表示一个DRAM经由总线的读操作的简化时序。存取时间认为是由$t1$到$t2$。由$t2$到$t3$是刷新时间，此期间 DRAM 芯片必须再充电，然后处理器才能再次存取它们。



a) 假定存取时间是60ns，刷新时间是40ns。问：存储周期是多少（单位：ns，精度：整数）？假定1位输出，这个DRAM所支持的最大数据传输率是多少（单位Mbps，精度：整数）？

b）使用这些芯片构成一个32位宽的存储器系统，其产生的数据传输率是多少（单位Mbps，精度：整数）？

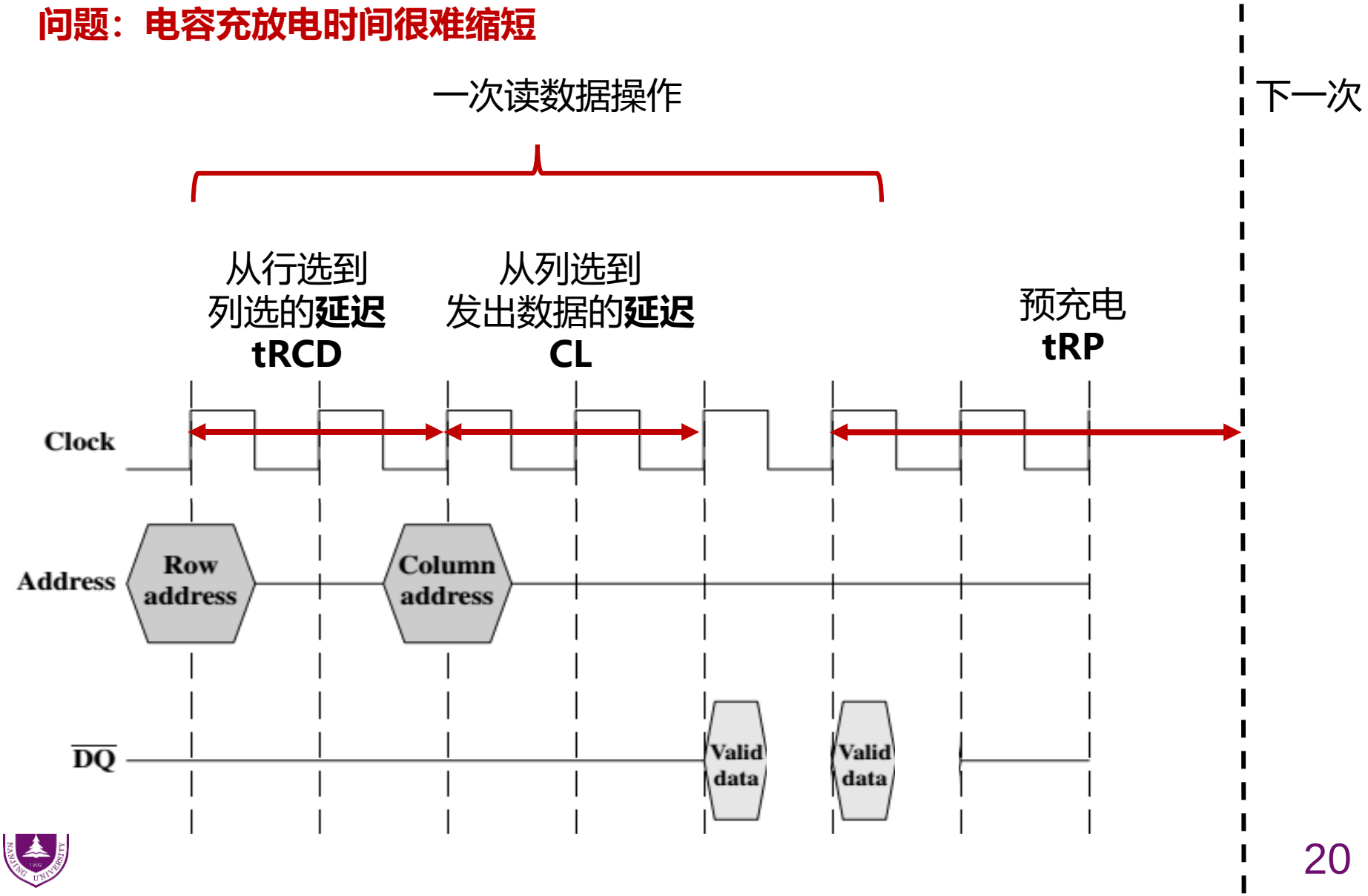
回顾：如何刷新

约束：刷新会占用片选线、地址线、地址译码器

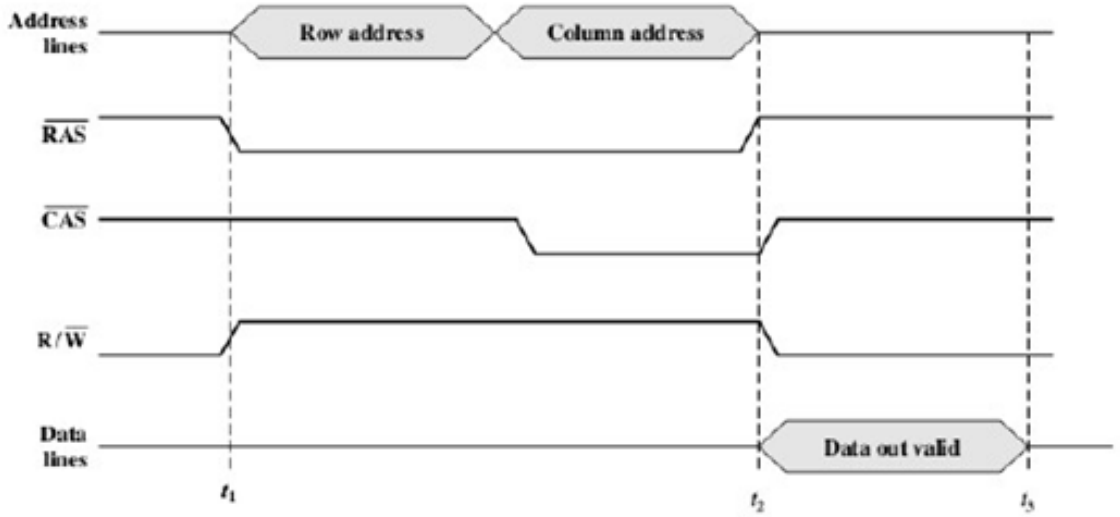
- **集中式刷新（Centralized refresh）**
  - 停止读写操作，并逐行刷新
  - 刷新时无法操作内存（死区）
- **分散式刷新（Decentralized refresh）**
  - 在每个存储周期中，当读写操作完成时进行刷新
  - 一次读写刷新一行，逐行刷新
  - 会增加每个存储周期的时间
- **异步刷新（Asynchronous refresh）**
  - 每一行各自以固定间隔（小于最大刷新周期，毫秒级）刷新
  - 将DRAM的刷新安排在CPU对指令的译码阶段，可有效避免死区
  - 效率高：常用

回顾：SDRAM

问题：电容充放电时间很难缩短



6、假设采用分散式刷新，下图表示一个DRAM经由总线的读操作的简化时序。存取时间认为是由$t1$到$t2$。由$t2$到$t3$是刷新时间，此期间 DRAM 芯片必须再充电，然后处理器才能再次存取它们。



a) 假定存取时间是60ns，刷新时间是40ns。问：存储周期是多少（单位：ns，精度：整数）？假定1位输出，这个DRAM所支持的最大数据传输率是多少（单位Mbps，精度：整数）？

Tc=60ns+40ns=100ns

数据传输率V=1bit/100ns=**10Mbps**

b) 使用这些芯片构成一个32位宽的存储器系统，其产生的数据传输率是多少（单位Mbps，精度：整数）？

数据传输率V′ =32×V=32×10Mbps=**320Mbps**

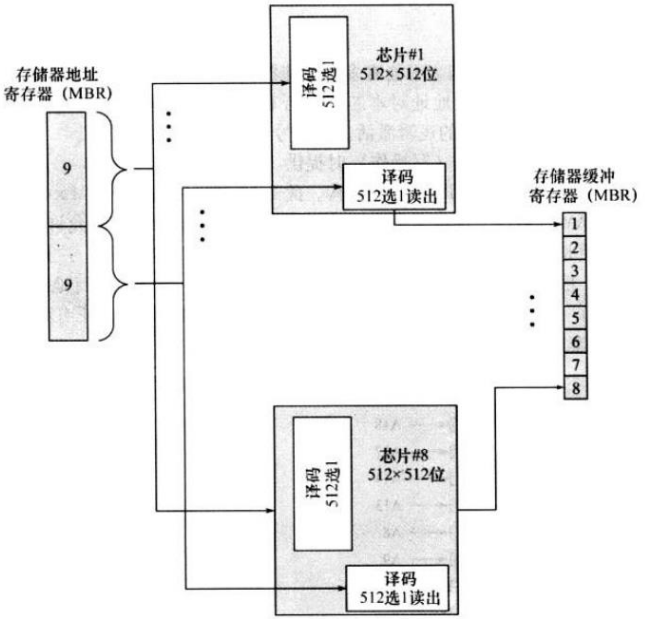
7、已知某机主存容量为 64KB，按字节编址。假定用 1K×4 位的 DRAM 芯片构成该存储器，请问：

a) 需要多少个这样的DRAM芯片?

b) 主存地址共多少位？哪几位用于选片？哪几位用于片内选址?

回顾：从位元到主存

- **模块组织**
  - **位扩展**：地址线不变，数据线增加
    - 使用 8 块 4K*1 bit 的芯片组成 4K*8 bit 的存储器
  - **字扩展**：地址线增加，数据线不变
    - 使用 4 个 16K*8 bit 的芯片组成 64K*8 bit 的存储器
  - **字、位同时扩展**：地址线增加，数据线增加
    - 使用 8 个 16K*4 bit 的芯片组成 64K*8 bit 的存储器



7、已知某机主存容量为 64KB，按字节编址。假定用 1K×4 位的 DRAM 芯片构成该存储器，请问：

a) 需要多少个这样的DRAM芯片?

N=64KB/(1K×4bit)=**128（1B=8bit)**

b) 主存地址共多少位？哪几位用于选片？哪几位用于片内选址?

主存容量为64KB，按字节寻址，所以寻址空间为64K=2^16，**主存地址为16位**。
由于片内为1K个地址，所以**低10位为片内地址**，**高16-10=6位用于选片**。

注意：此处不能算为128=2^7，所以高7位选片，低16-7=9位用于片内选址。因为片内的选址单元是4位，需要位扩展后才能按字节编址（整体上是字位扩展），即选片时都是同时选中2个芯片。

8、计算机系统包含容量为32K×16位的主存，按字编址，每字16位。Cache采用4路组关联的映射方式，数据区大小为4K字，主存块大小为64字。假设Cache初始时是空的，处理器顺序地从存储单元（每个存储单元中包含1个字）0,1,...,4351中取数，然后再重复这一顺序9次，并且Cache的速度是主存的10倍，同时假设块替换用LRU算法。请说明使用Cache后的速度为原来的多少倍（精度：小数点后1位）。

回顾：组关联映射

- 标记
  - 地址中最高n位，$n = log_2 M - log_2 S$

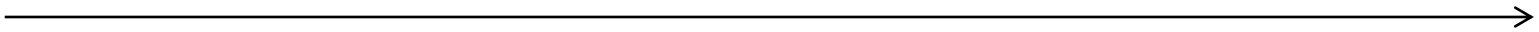
主存地址	标记	Cache组号	块内地址

- 示例
  - 假设cache有4行，每行包含8个字，分成2个组；主存中包含128个字。访问主存的地址长度为7位，则：
    - 最低的3位：块内地址
    - 中间的1位：映射时所对应的Cache中的组
    - 最高的3位：区分映射到同一组的不同块，记录为Cache标记

回顾：最近最少使用算法（LRU）

例：假设cache有**2组**，**每组4行**，每行**8个字**，主存地址长度为**8位**

主存地址 00000000 10000010 01000001 11000100 00000000 11100011



Cache

LRU	标记
0	0000

未命中
载入

LRU	标记
1	0000
0	1000

未命中
载入

LRU	标记
2	0000
1	1000
0	0100

未命中
载入

LRU	标记
3	0000
2	1000
1	0100
0	1100

未命中
载入

LRU	标记
0	0000
3	1000
2	0100
1	1100

命中

LRU	标记
1	0000
0	1110
3	0100
2	1100

未命中
替换

- **LRU位需要额外的硬件实现**
  - **K路需要$log(\mathrm{K!}) \approx K \cdot log(K)$位，例如4路时每行需要2位，共需8位**
- **LRU会增加cache访问时间**

8、计算机系统包含容量为32K×16位的主存，按字编址，每字16位。Cache采用4路组关联的映射方式，数据区大小为4K字，主存块大小为64字。假设Cache初始时是空的，处理器顺序地从存储单元（每个存储单元中包含1个字）0,1,...,4351中取数，然后再重复这一顺序9次，并且Cache的速度是主存的10倍，同时假设块替换用LRU算法。请说明使用Cache后的速度为原来的多少倍（精度：小数点后1位）。

主存地址为：标记5，组号4，块内地址6

4352/64=68，即在前68块中操作10次。但Cache只有16组，每组4行。

	第0行	第1行	第2行	第3行
第0组	0 / 64 / 48	16 / 0 / 64	32 / 16	48 / 32
第1组	1 / 65 / 49	17 / 1 / 65	33 / 17	49 / 33
第2组	2 / 66 / 50	18 / 2 / 66	34 / 18	50 / 34
第3组	3 / 67 / 51	19 / 3 / 97	35 / 19	51 / 35
第4组	4	20	36	52
⋮	⋮	⋮	⋮	⋮
第15组	15	31	47	63

第1轮未命中68次
第2~10轮未命中号为：
0,1,2,3,16,17,18,19,32,33,34,
35,48,49,50,51,64,65,66,67
轮番替换，上面这20个块永远各未命中1次

命中率 P=(4352*10-68-20*9)/43520=99.43%

设cache的读取时间为T，主存的读取时间为10T，则使用cache后，系统效率提高到原来的N倍。

N= 10T/(T+0.0057*10T)=**9.5**

9、考虑一个每行16个字节的4行Cache，主存按每块16个字节划分，即块0有地址0到15的16个字节，等等。现考虑一程序，它以如下地址顺序访问主存：

一次：63～70

循环10次：15～32，80～95

a) 假设Cache组织成直接映射式。块0、4、...指派到行0，块1、5、...指派到行1，如此类推。请计算命中率（形式：小数，非百分数；精度：小数点后3位）。

b) 假设Cache组织成两路组关联映射式，共有两组，每组两行。偶序号块指派到组0，奇序号块指派到组1。使用LRU替换策略，请计算命中率（形式：小数，非百分数；精度：小数点后3位）。

回顾：直接映射

- **（块）标记**
  - 地址中最高n位, $n = log_2 M - log_2 C$ ← $i = j \ mod \ C$

同行号则低$log_2 C$位相同

块的数量　　行的数量

主存地址	标记	Cache行号	块内地址

- **例**
  - 假设cache有4行，每行包含8个字；主存中包含128个字。访问主存的地址长度为7位，则：
    - 最低的3位：块内地址
    - 中间的2位：映射时所对应的Cache行号
    - 最高的2位：区分映射到同一行的不同块，记录为Cache标记

9、考虑一个每行16个字节的4行Cache，主存按每块16个字节划分，即块0有地址0到15的16个字节，等等。现考虑一程序，它以如下地址顺序访问主存：

一次：63～70

循环10次：15～32，80～95

a) 假设Cache组织成直接映射式。块0、4、…指派到行0，块1、5、…指派到行1，如此类推。请计算命中率（形式：小数，非百分数；精度：小数点后3位）。

一次有63，64未命中，循环第一次有15，16，32，80未命中，以后9次有16,80未命中。所以命中率：

P=(8+18*10+16*10-2-4-2*9)/348=**0.931**

b) 假设Cache组织成两路组关联映射式，共有两组，每组两行。偶序号块指派到组0，奇序号块指派到组1。使用LRU替换策略，请计算命中率（形式：小数，非百分数；精度：小数点后3位）。

前面一样，后9次循环都命中。所以

P=(348-6)/348=**0.983**

10、考虑一个存取时间为1ns和命中率H=0.95的L1 Cache。假设我们修改了此Cache的设计（Cache的容量、组织），从而使得命中率提升到0.97，但也使存取时间增大到1.5ns。如果要使得新设计能导致性能改善，cache的速度必须是主存的多少倍以上（精度：整数）？

回顾：平均访问时间

- 假设p是**命中率**， T_C 是cache的访问时间， T_M 是主存的访问时间，使用cache时的**平均访问时间**T_A为

$$T_A = p \times T_C + (1-p) \times (T_C + T_M)$$
$$= T_C + (1-p) \times T_M$$

Cache返回数据的时间忽略不计

降低Cache访问时间 **矛盾** 提高命中率

降低主存访问时间（即降低未命中惩罚） **很难**

- 命中率p越大， T_C越小， 效果越好
- 如果想要 $T_A < T_M$, 必须要求

$$p > T_C / T_M$$

- 难点：cache的容量远远小于主存的容量（随机访问时p = Cache容量 /主存容量）

10、考虑一个存取时间为1ns和命中率H=0.95的L1 Cache。假设我们修改了此Cache的设计（Cache的容量、组织），从而使得命中率提升到0.97，但也使存取时间增大到1.5ns。如果要使得新设计能导致性能改善，cache的速度必须是主存的多少倍以上（精度：整数）？

设主存的读写所耗为Tns，必须满足：

T*(1-0.95)+1>T*(1-0.97)+1.5

算得T>25

所以cache的速度需要是主存的

25/1.5=**17倍以上**。

11、假设某处理器的时钟频率为1.2GHz，当L1 cache无缺失时的CPI为1（即CPU可以快速地从L1 cache中读取指令，并在1个时钟周期内完成）。访问一次主存的时间为100ns（包括所有缺失处理），L1 cache的局部缺失率为2%。若增加一个L2 cache，并假定L2 cache的访问时间为5ns，而且其容量足够大到使全局缺失率仅为0.5%。分析增加L2 cache后处理器执行程序的效率为原来的多少倍（精度：小数点后3位）？

CPU时钟周期为1/1.2GHZ=0.833ns

未增加L2时读一条指令平均耗时：

T1=0.833ns+100ns*2%=2.833ns

增加L2后：

T2=0.833ns+2%*5ns+100*0.5%ns=1.433ns

则效率提高到了原来的T1/T2=**1.977倍**。

12、考虑一个单片磁盘，它有如下参数：旋转速率是7200rpm，一面上的磁道数是30000，每道扇区数是600，寻道时间是每越过一百个磁道用时1ms。假定开始时磁头位于磁道0，收到一个存取随机磁道上随机扇区的请求。

a) 平均寻道时间是多少（精度：小数点后2位，单位：s）？

b) 平均旋转延迟是多少（精度：小数点后2位，单位：ms）？

c) 一个扇区的传送时间是多少（精度：小数点后4位，单位：ms）？

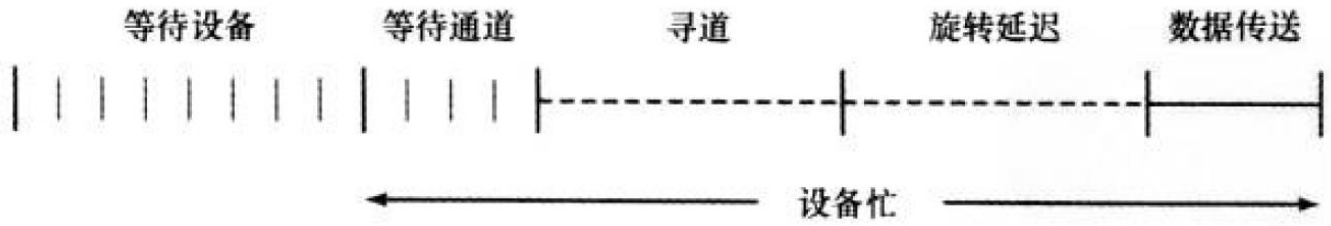
d) 完成访问请求的总的平均时间是多少（精度：小数点后2位，单位：ms）？

回顾：硬磁盘I/O访问时间

- **寻道时间（seek time）**：磁头定位到所需移动到的磁道所花费的时间
  - 初始启动时间，跨越若干磁道所用的时间
- **旋转延迟（rotational delay）**：等待响应扇区的起始处到达磁头所需的时间
  - 通常是磁道旋转半周所需的时间
- **传送时间（transfer time）**：数据传输所需的时间

$$T = \frac{b}{rN}$$

T = 传送时间
b = 传送的字节数
N = 每磁道的字节数
r = 旋转速率，单位是转/秒



回顾：硬磁盘I/O访问时间

- **平均访问时间**

取平均时间　取实际时间

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

T_s 是平均寻道时间

- 当连续访问多个相邻的磁道时，**跨越磁道**：
  - 对于每个磁道都需要考虑旋转延迟
  - **通常**只需要考虑第一个磁道的寻道时间，但在明确知道跨越每个磁道需要的时间时需要考虑

磁道非常多，顺序移到下一个磁道的耗时是非常少的

12、考虑一个单片磁盘，它有如下参数：旋转速率是7200rpm，一面上的磁道数是30000，每道扇区数是600，寻道时间是每越过一百个磁道用时1ms。假定开始时磁头位于磁道0，收到一个存取随机磁道上随机扇区的请求。

a) 平均寻道时间是多少（精度：小数点后2位，单位：s）？

$$T_S = \frac{1}{100} ms * \frac{29999}{2} \approx 150ms = \mathbf{0.15s}$$

b) 平均旋转延迟是多少（精度：小数点后2位，单位：ms）？

$$\frac{1}{2 * 7200\, r/min} * 60\, s/min = \frac{1}{240} s \approx \mathbf{4.17ms}$$

c) 一个扇区的传送时间是多少（精度：小数点后4位，单位：ms）？

$$T = \frac{b}{rN} = \frac{60\, s/min}{7200\, r/min} * \frac{1}{600} = \frac{1}{72000} \text{s} \approx \mathbf{0.0139ms}$$

d) 完成访问请求的总的平均时间是多少（精度：小数点后2位，单位：ms）？

$$T_A = T_S + \frac{1}{2\text{r}} + \frac{\text{b}}{\text{rN}} = 0.15 + \frac{1}{240} + \frac{1}{72000} \approx \mathbf{154.18ms}$$

13、假定一个程序重复完成将磁盘上一个4KB的数据块读出，进行相应处理后，写回到磁盘的另外一个数据区。各数据块内信息在磁盘上连续存放，并随机地位于磁盘的一个磁道上。磁盘转速为7200rpm，平均寻道时间为10ms，磁盘最大数据传输率为320Mbps，没有其他程序使用磁盘和处理器，并且磁盘读写操作和磁盘数据的处理时间不重叠。若程序对磁盘数据的处理需要20000个时钟周期，处理器时钟频率为500MHz，则:

a）该程序完成一次数据块“读出-处理-写回”操作所需要的时间为多少（精度：小数点后2位，单位：毫秒）？

平均旋转延迟:

1/(2*7200)*60=1/240 s≈4.17ms

因为块内信息连续存放,所以数据传输时间:

4KB/320Mbps=(4*1024*8)/(320*10^6) s≈0.1ms

则存取时间，即平均存取时间:

T= 10ms+4.17ms+0.1ms=14.27ms

数据块的处理时间:

20000/500MHz=0.04ms

“读出-处理-写回”需要2次存取，1次处理，故完成一次操作时间:

14.27*2+0.04=**28.58ms**

b）每秒钟可以完成多少次这样的数据块操作（精度：整数）？

$$\left\lfloor \frac{1\text{s}}{28.58\text{ms}} \right\rfloor = \textbf{34}\text{次}$$

14、某个磁盘的磁道编号为0～999。磁头寻道时，每跨越1个磁道所需的平均时间为0.01ms（例如磁头从磁道2移动到磁道3需要0.01ms）。磁盘的平均旋转速度为6000转/分钟。每个磁道上的扇区数量为1000个。

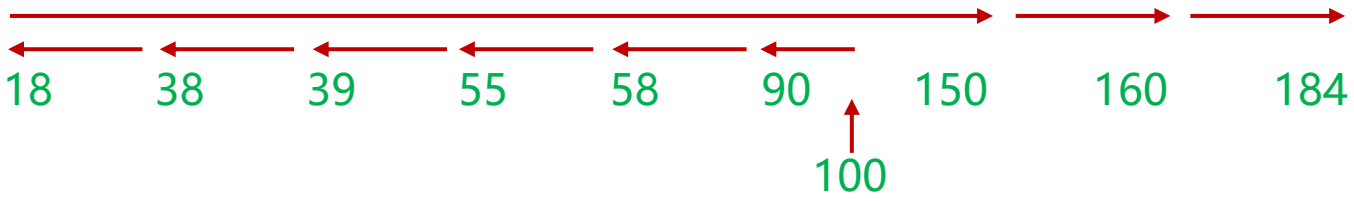
已知当前磁盘为空，有5个写入数据的任务同时到达

任务	1	2	3	4	5
开始写入的磁道	300	170	220	90	470
写入数据大小	3MB	40KB	1MB	500KB	600KB

假设磁头的初始位置为磁道200，采用最短寻道时间优先算法（即优先处理开始写入位置与当前磁头位置最接近的任务），且每个磁道上都从0号扇区写入，多于1个磁道时向磁盘中心移动。请问完成这5个写入任务所需要的总时间为多少?

回顾：磁头寻道/磁盘调度

- **最短寻道时间优先（SSTF）**
  - **优先处理起始位置与当前磁头位置最接近的读写任务**
  - **优点**：每次的寻道时间最短（局部最优），平均寻道时间缩短
  - **缺点**：可能产生饥饿现象，尤其是位于两端的磁道请求
  - **示例**：假设磁头的初始位置是100号磁道，有多个任务先后陆续的请求访问55，58，39，18，90，160，150，38，184号磁道



  - 磁头总共移动了（100-18）+（184-18）=248个磁道
  - 平均寻道长度为248/9=**27.5个磁道**

14、某个磁盘的磁道编号为0～999。磁头寻道时，每跨越1个磁道所需的平均时间为0.01ms（例如磁头从磁道2移动到磁道3需要0.01ms）。磁盘的平均旋转速度为6000转/分钟。每个磁道上的扇区数量为1000个。

已知当前磁盘为空，有5个写入数据的任务同时到达

任务	1	2	3	4	5
开始写入的磁道	300	170	220	90	470
写入数据大小	3MB	40KB	1MB	500KB	600KB
	6.144磁道	0.08磁道	2.048磁道	1磁道	1.2磁道

假设磁头的初始位置为磁道200，采用最短寻道时间优先算法（即优先处理开始写入位置与当前磁头位置最接近的任务），且每个磁道上都从0号扇区写入，多于1个磁道时向磁盘中心移动。请问完成这5个写入任务所需要的总时间为多少?

磁盘旋转一周的时间：$\frac{60s}{6000RPM} = 10ms$

平均旋转延迟：$10ms \times \frac{1}{2} = 5\ ms$

每个磁道可存储数据：$512B \times 1000 = 500KB$

$200 \longrightarrow 220\sim222 \longrightarrow 170 \longrightarrow 90 \longrightarrow 300\sim306 \longrightarrow 470\sim471$

寻道长度：$(222-200+222-170+170-90+306-90+471-306) = 535$磁道

总时间：$535 \times 0.01ms + 5ms \times (7+1+3+1+2) + 10ms \times (6.144+0.08+2.048+1+1.2) = 5.35+75+104.72 = 180.07ms$

15、假定有两个用来存储10TB数据的RAID系统，每个磁盘的大小均为2TB。系统A使用RAID 1技术，系统B使用RAID 5技术。请问：

a) 系统A需要比系统B多用多少存储容量（单位：TB）？

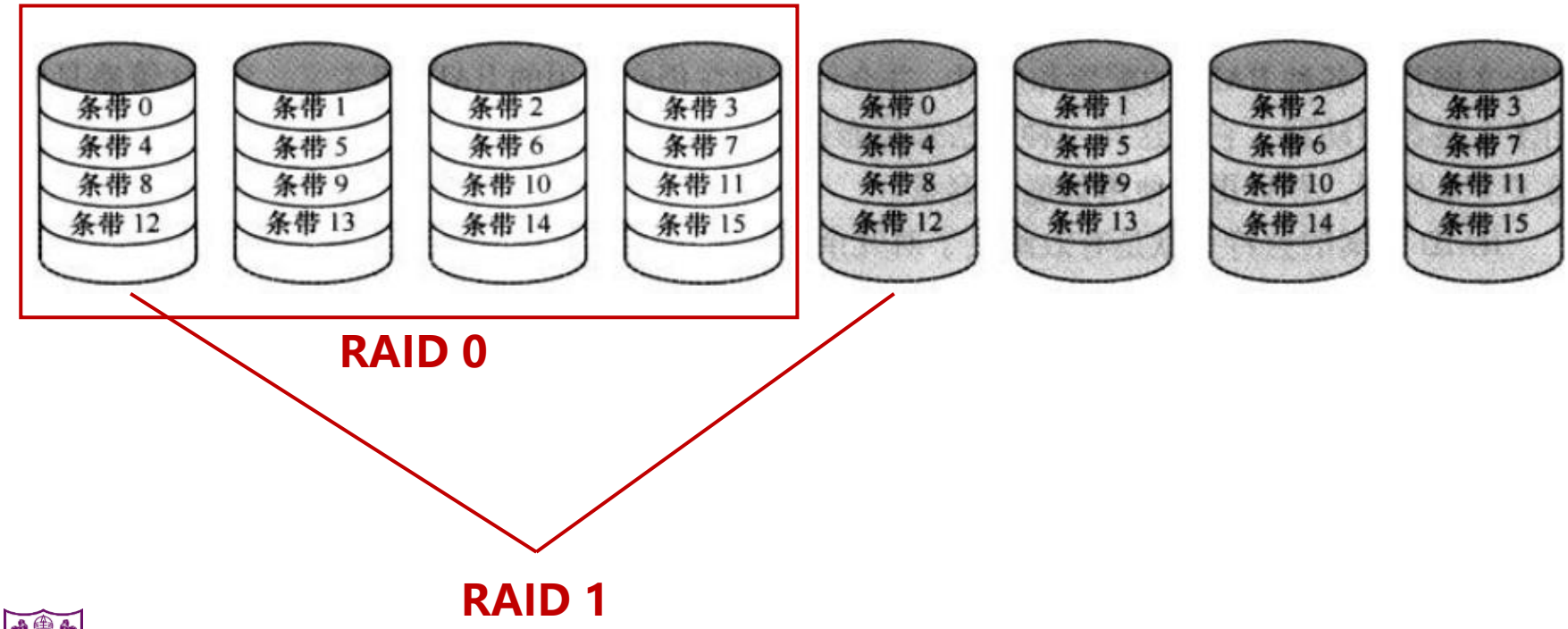
b) 假定一个应用需要向磁盘写入一块数据，若磁盘读或写一块数据的时间为30ms，则最坏情况下，在系统A上写入一块数据需要多少时间（单位：毫秒）？

c) 如果问题b）是在系统B上写入一块数据，需要多少时间（单位：毫秒）？

d) 哪个系统更加可靠?

回顾：RAID 1

- 采用了数据条带
- 采用简单地备份所有数据的方法来实现冗余

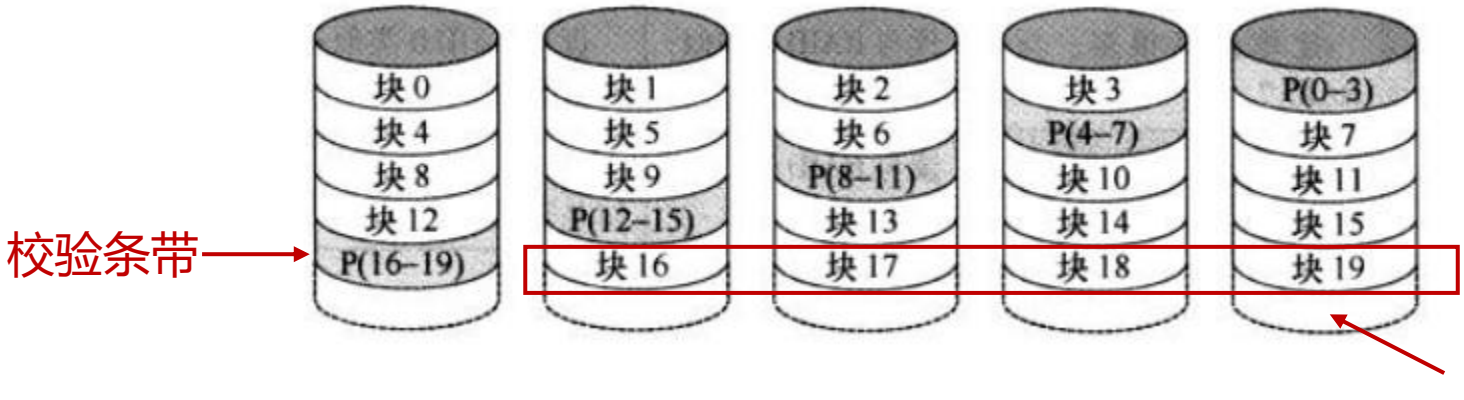


回顾：RAID 5

- 与RAID 4 组织方式相似（常用）
- 在所有磁盘上都分布了奇偶校验条带
  - 避免潜在的I/O瓶颈问题
- 访问时的“两读两写”：读在写前，读/写不需要并行

$$P'(B) = P(B) \oplus B_0 \oplus B_0'$$

读　写



5	块交错分布式奇偶校验	$N+1$	比单盘高很多；与RAID 2、3、4 差不多	读与RAID 0 类似；写低于单盘	读与RAID 0 类似；写显著低于单盘

15、假定有两个用来存储10TB数据的RAID系统，每个磁盘的大小均为2TB。系统A使用RAID 1技术，系统B使用RAID 5技术。请问：

a) 系统A需要比系统B多用多少存储容量（单位：TB）？

A系统需20TB存储容量；B系统采用6个磁盘，需12TB存储容量，**多用8TB**

b) 假定一个应用需要向磁盘写入一块数据，若磁盘读或写一块数据的时间为30ms，则最坏情况下，在系统A上写入一块数据需要多少时间（单位：毫秒）？

30ms

c) 如果问题b）是在系统B上写入一块数据，需要多少时间（单位：毫秒）？

120ms（最坏情况：两读两写）

d) 哪个系统更加可靠?

假设每一个盘损坏的概率均为p

RAID1的可靠性为：$R1 = 1-p^2$

RAID5的可靠性为：$R5 = (1-p)^n C_n^0 + p(1-p)^n C_n^1 = [(n-1)p + 1](1-p)^{n-1}$

RAID1的可靠性比RAID5更高，且RAID5的磁盘数越多，可靠性越低。

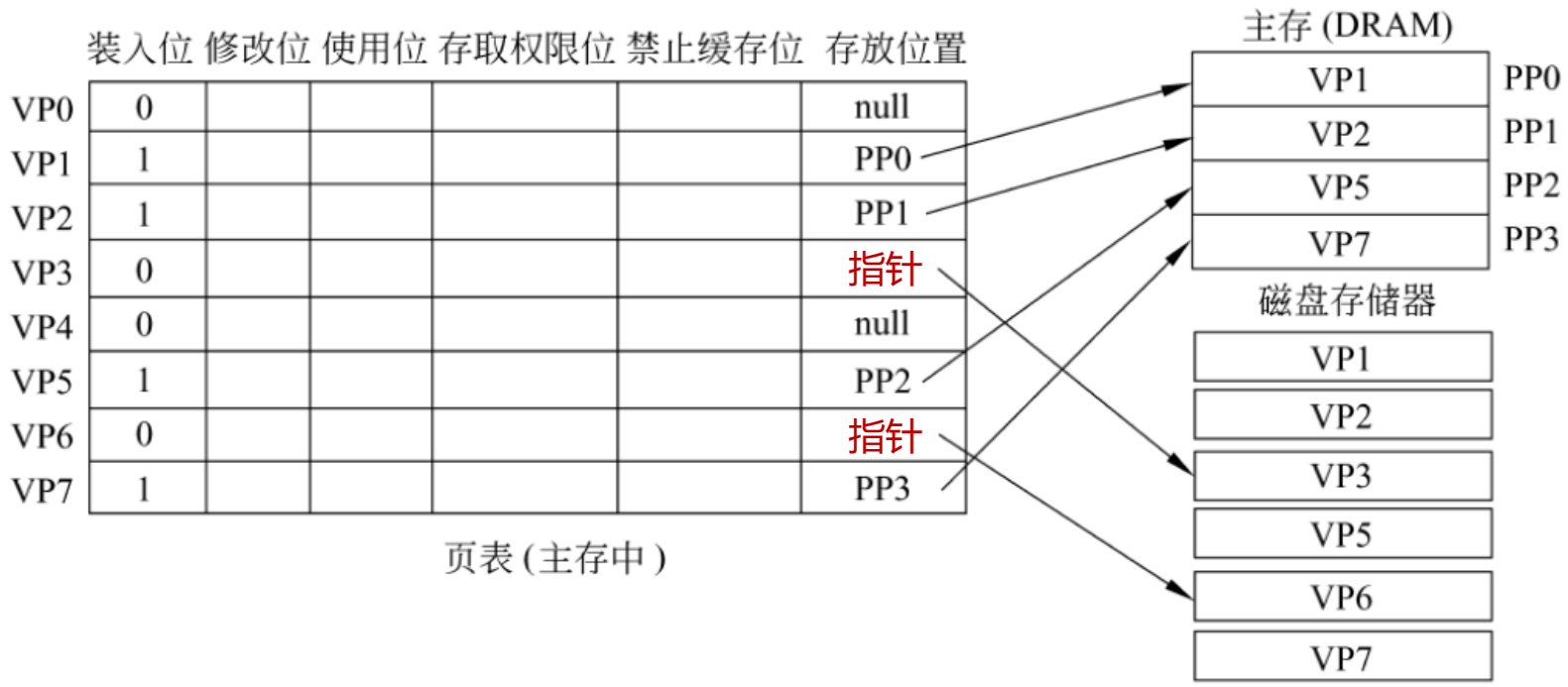
16、假设一个分页虚拟存储系统的虚拟地址为40位，物理地址为36位，页大小为16KB，按字节编址。若页表中的有效位、存储保护位、修改位、使用位共占4位，磁盘地址不在页表中。则该存储系统中每个程序的页表大小为多少（单位：MB）？（说明：1.假设每个程序都能使用全部的虚拟内存；2.页表项的长度必须为字节的整数倍）

回顾：分页式虚拟存储器的页表

根据页表中记录的物理页存放位置，可以将虚拟地址转化为物理地址

虚拟页号 + 页内偏移量 ➔ 物理页号 + 页内偏移量

虚拟地址　　物理地址



16、假设一个分页虚拟存储系统的虚拟地址为40位，物理地址为36位，页大小为16KB，按字节编址。若页表中的有效位、存储保护位、修改位、使用位共占4位，磁盘地址不在页表中。则该存储系统中每个程序的页表大小为多少（单位：MB）？（说明：1.假设每个程序都能使用全部的虚拟内存；2.页表项的长度必须为字节的整数倍）

按字节编址，故：
虚拟主存页面个数：
2^(40-14)=2^26
物理主存页面数：
2^(36-14)=2^22
页表项的最小长度：
22+4=26
根据说明2，取32位（4B）
页表大小：
2^26*4B=**256MB**

谢谢

bohanliu@nju.edu.cn

南京大學
NANJING UNIVERSITY