

计算机组成与系统结构

第三章 多层次的存储器

吕昕晨

lvxinchen@bupt.edu.cn

网络空间安全学院



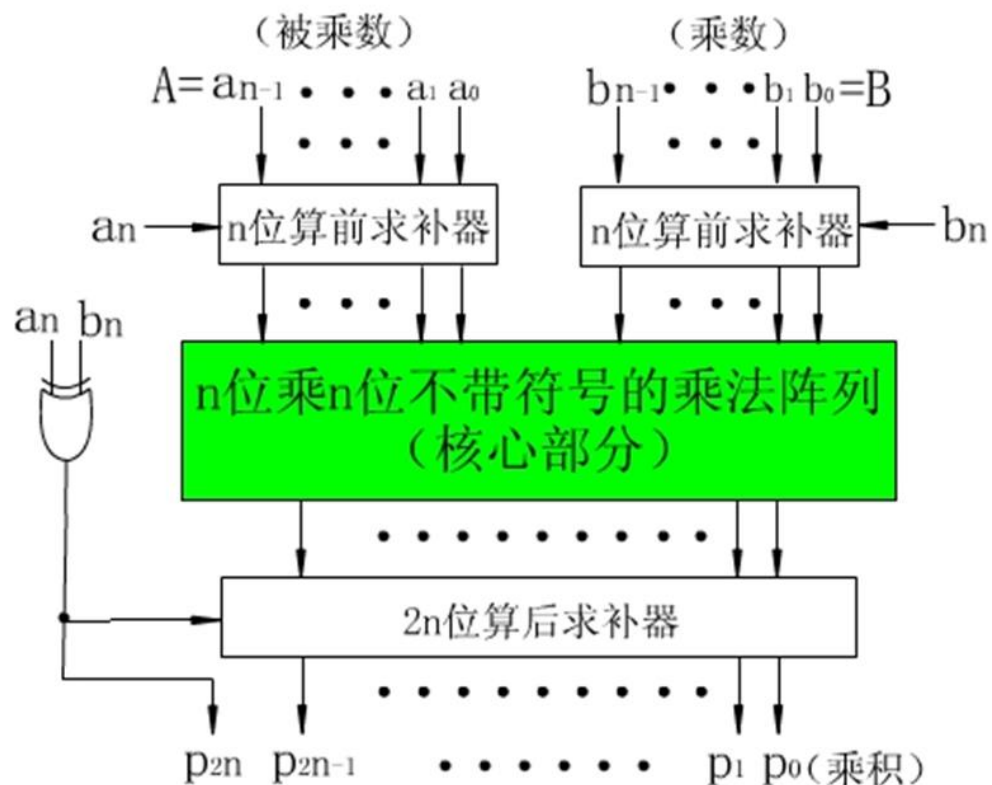
复习

- 定点乘法（带符号、原码补码乘法）
- 定点除法
- 浮点加减法
- 浮点乘法

带符号乘法器



- 补码性质
 - $[[A]_{\text{补}}]_{\text{补}} = [A]_{\text{原}}$
- 带符号乘法器构思路
 - 算前求补 (输入补码)
 - 乘法器
 - 算后求补 (输出补码)

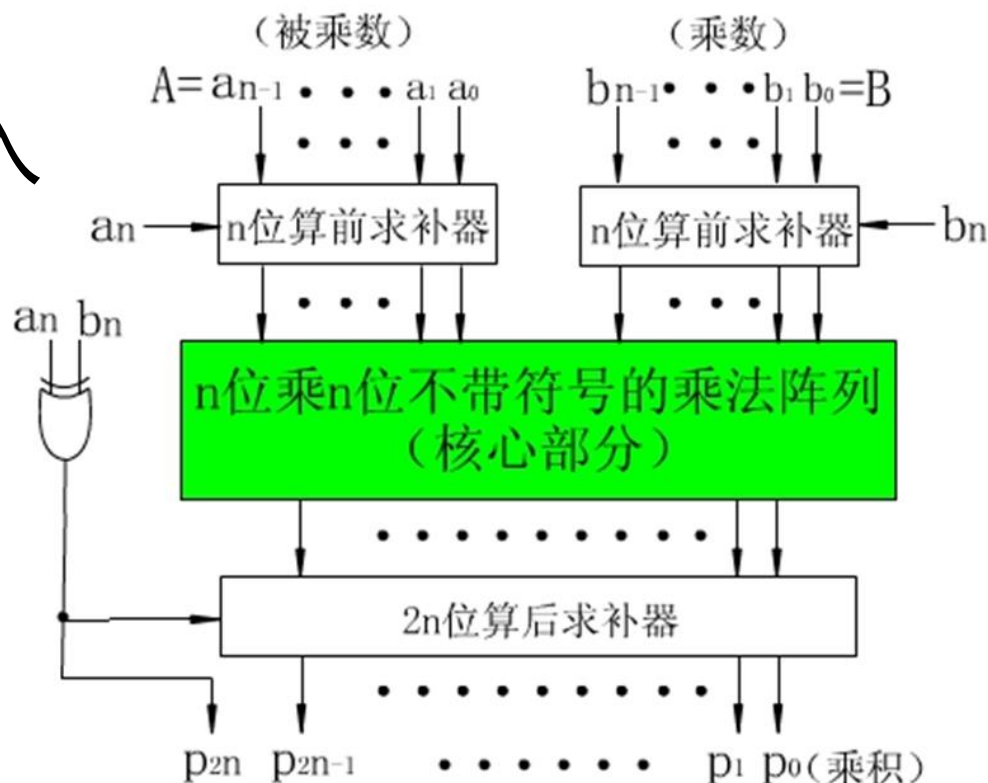


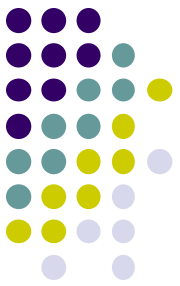
- 补码转换性质
 - 最右端往左边扫描，直到第一个1的时候，该位和右边各位保持不变，左边各数值位按位取反 (扫描)

带符号乘法器



- 求解步骤
 - 判断原码/补码输入
 - 符号位计算
 - (算前求补)
 - 乘法 (不带符号)
 - (算后求补)





习题[2-7-1] 已知 $x=11011$, $y=-11111$
用原码阵列乘法器、补码阵列乘法器, 计算 $x \times y$?

- 原码乘法 (输入原码、输出原码)

- 符号位: $0 \oplus 1 = 1$

- 无需算前算后求补

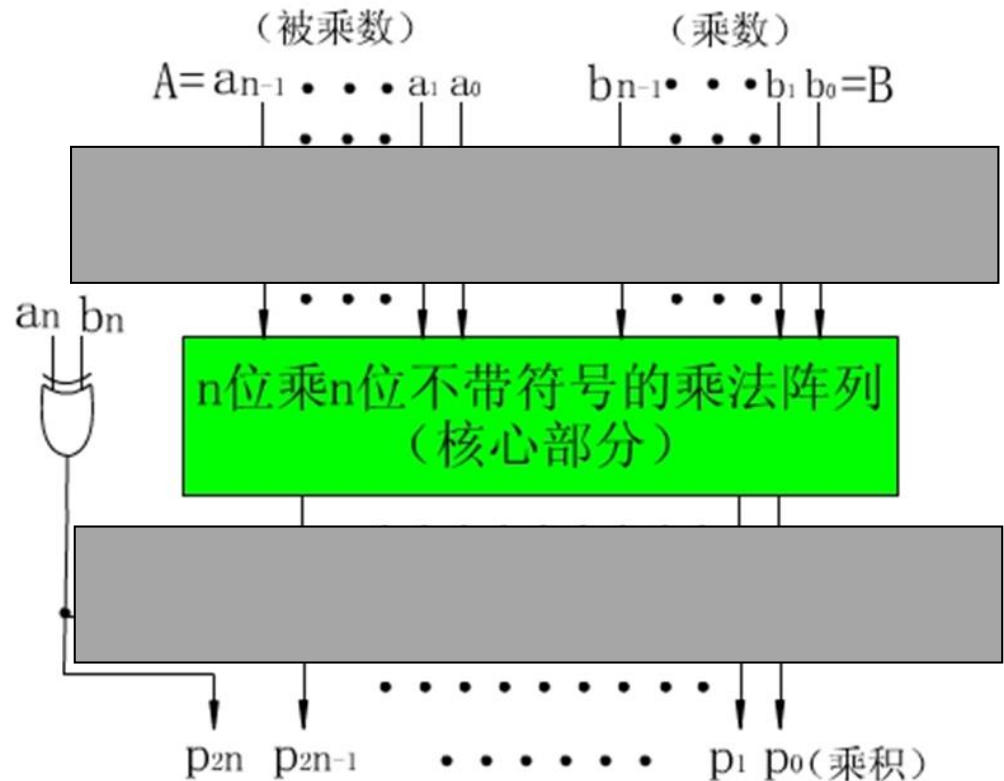
- $|x|=11011$, $|y|=11111$

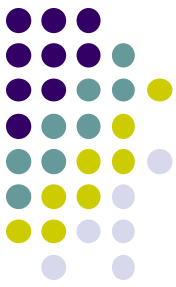
- 乘法:

$$\begin{array}{r}
 11011 \\
 \times 11111 \\
 \hline
 11011 \\
 11011 \\
 11011 \\
 11011 \\
 11011 \\
 + 11011 \\
 \hline
 1101000101
 \end{array}$$

1 1 0 1 0 0 0 1 0 1

- $[x \times y]_{\text{原}} = 1 \text{ } 1101000101$





习题[2-7-1)] 已知 $x=11011$, $y=-11111$
用原码阵列乘法器、补码阵列乘法器, 计算 $x \times y$?

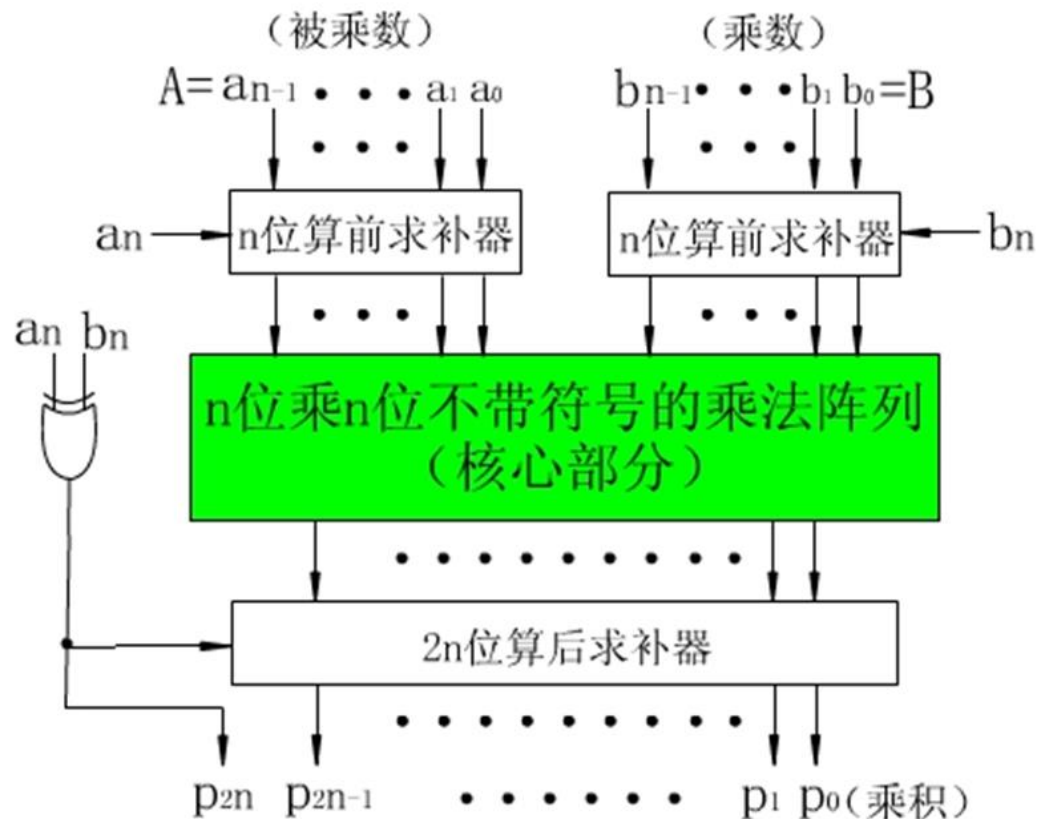
- 补码乘法 (输入补码、输出补码)
- $[x]_{\text{补}}=0\ 11011$, $[y]_{\text{补}}=100001$

- 符号位: $0 \oplus 1 = 1$
- 算前求补
- $|x|=11011$, $|y|=11111$

乘法:

$$\begin{array}{r}
 11011 \\
 \times 11111 \\
 \hline
 11011 \\
 11011 \\
 11011 \\
 11011 \\
 11011 \\
 + 11011 \\
 \hline
 1101000101
 \end{array}$$

- $[x \times y]_{\text{原}} = 1\ 1101000101$
- 算后求补 $[x \times y]_{\text{补}} = 1\ 0010111011$





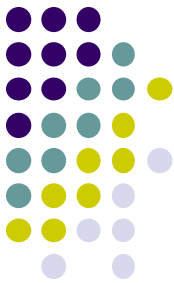
复习

- 定点乘法
- 定点除法（加减交替、位数终止、余数）
- 浮点加法
- 浮点乘法



计算机除法流程

- 人工除法时，人可以比较被除数（余数）和除数的大小来确定商1（够减）或商0（不够减）
- 机器除法时，余数为正表示够减，余数为负表示不够减。不够减时必须恢复原来余数，才能继续向下运算。这种方法叫**恢复余数法**，控制比较复杂。
- 不恢复余数法（**加减交替法**）
 - 余数为正，商1，下次除数右移做减法
 - 余数为负，商0，下次除数右移做加法



补码除法流程——加减交替法

[例23] $x = 0.101001$, $y = 0.111$, 求 $x \div y$ 。

[解:] $[x]_{\text{补}} = 0.101001$, $[y]_{\text{补}} = 0.111$, $[-y]_{\text{补}} = 1.001$

起始位置 0.101001 ; 被除数
 $+ [-y]_{\text{补}} \rightarrow 1.001$; 第一步减除数 $y = 0.111$
小数点后3位

$1.110001 < 0$ $q_0 = 0$; 余数为负, 商0
 $+ [y]_{\text{补}} \rightarrow 0.0111$; 除数右移1位加

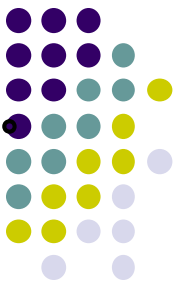
$0.001101 > 0$ $q_1 = 1$; 余数为正, 商1
 $+ [-y]_{\text{补}} \rightarrow 1.11001$; 除数右移2位减

$1.111111 < 0$ $q_2 = 0$; 余数为负, 商0
 $+ [y]_{\text{补}} \rightarrow 0.000111$; 除数右移3位加

补码 $0.000110 > 0$ $q_3 = 1$; 余数为正, 商1

商 $q = q_0.q_1q_2q_3 = 0.101$, 余数 $r = 0.000110$

真值



习题[2-8-1) $x = 11000$, $y = -11111$, 用原码除法, 求 $x \div y$

- 符号位: $0 \oplus 1 = 1$
- $|x| = 11000$, $|y| = 11111$
- 纯小数表示, 小数点左移5位, $|x| = 0.11000$, $|y| = 0.11111$ 小数点后5位
- $[|x|] = 0.11000$, $[|y|]_{\text{补}} = 0.11111$, $[-|y|]_{\text{补}} = 1.00001$

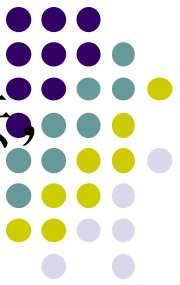
| | | |
|---|---|-----------|
| | 0.1 1 0 0 0 | ; 被除数 |
| + | [-y] _补 1.0 0 0 0 1 | ; 第一步减除数y |

| | | |
|---|---|-------------------------|
| | 1.1 1 0 0 1 | <0 $q_0 = 0$; 余数为负, 商0 |
| + | [y] _补 → 0.0 1 1 1 1 1 | ; 除数右移1位加 |

| | | |
|---|---|-------------------------|
| | 0.0 1 0 0 0 1 | >0 $q_1 = 1$; 余数为正, 商1 |
| + | [-y] _补 → 1.1 1 0 0 0 0 1 | ; 除数右移2位减 |

| | | |
|---|---|-------------------------|
| | 0.0 0 0 0 0 1 1 | >0 $q_2 = 1$; 余数为正, 商1 |
| + | [y] _补 → 1.1 1 1 0 0 0 0 1 | ; 除数右移3位减 |

| | | |
|--|-------------------|-------------------------|
| | 1.1 1 1 0 0 1 1 1 | <0 $q_3 = 0$; 余数为负, 商0 |
|--|-------------------|-------------------------|



习题[2-8-1] $x = 11000$, $y = -11111$, 用原码除法,
求 $x \div y$ 。

$1.1\ 1\ 1\ 0\ 0\ 1\ 1\ 1 < 0\ q_3 = 0$; 余数为负, 商0
+ $[y]_{\text{补}} \rightarrow 0.0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1$; 除数右移4位加

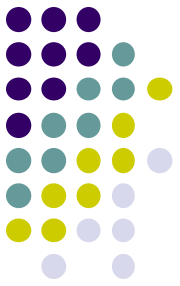
$1.1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1 < 0\ q_4 = 0$; 余数为负, 商0
+ $[y]_{\text{补}} \rightarrow 0.0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1$; 除数右移5位加

$1.1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1 < 0\ q_5 = 0$; 余数为负, 商0

- 商真值 $|x \div y| = 0.11000$, 原码除法 $[x \div y]_{\text{原}} = 1.11000$
- 余数: 0.0000011
- 小数点右移5位 (补偿): 0.11

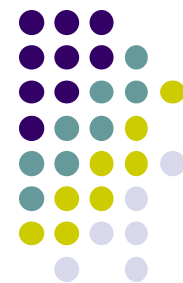
$0.0\ 0\ 0\ 0\ 0\ 1\ 1 > 0\ q_2 = 1$; 余数为正, 商1

$q_0 = 0,$
 $q_1 = 1,$
 $q_2 = 1$



复习

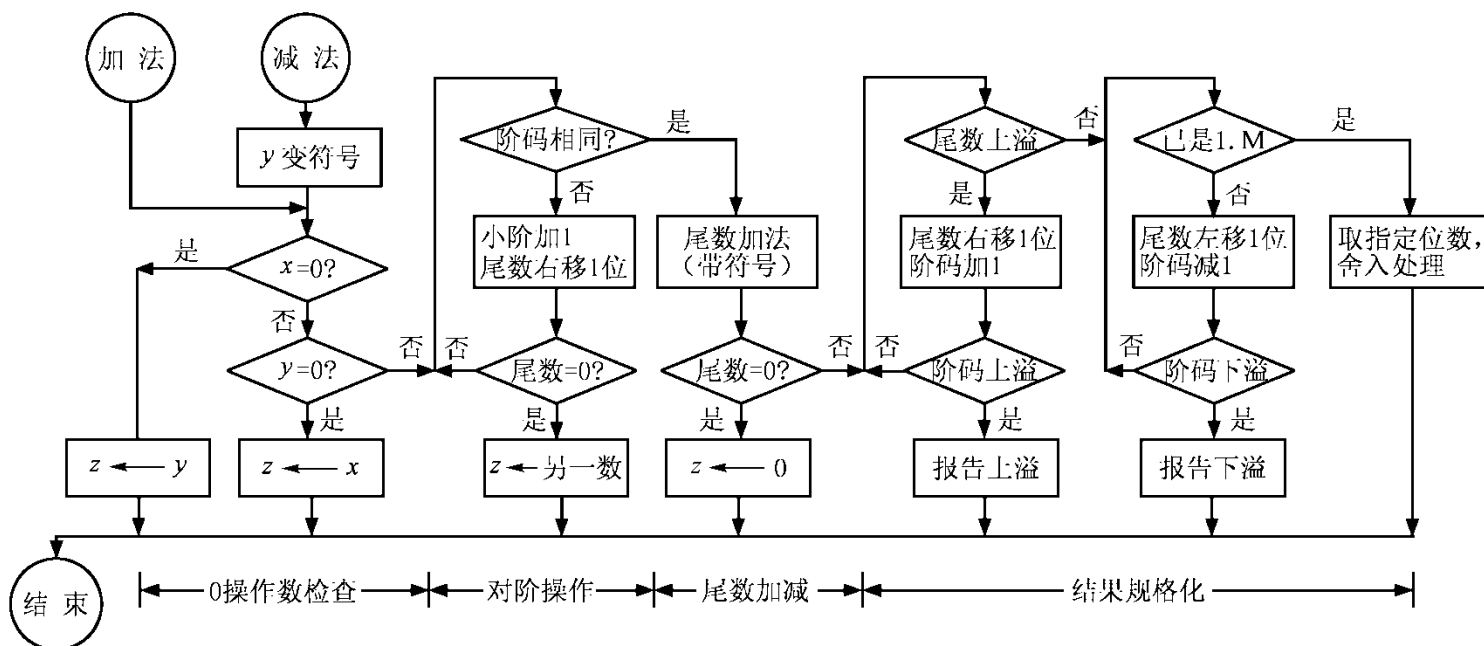
- 定点乘法
- 定点除法
- 浮点加减法（规格化、舍入）
- 浮点乘法

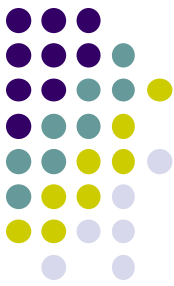


浮点加减法运算运算步骤

- 操作数检查;
- 比较阶码并完成对阶 (小阶向大阶对齐) ;
- 尾数求和运算;
- 结果规格化;
- 舍入处理

$$x \pm y = (M_x 2^{E_x - E_y} \pm M_y) 2^{E_y}$$





浮点加减法示例一 (1)

[例28] 设 $x = 2^{010} \times 0.11011011$, $y = -2^{100} \times 0.10101100$, 求 $x+y$

- 1、0操作数检查 (非0)
- 2、对阶：阶码对齐后才能加减。规则是阶码小的向阶码大的数对齐；
 - 若 $\Delta E = 0$, 表示两数阶码相等, 即 $E_x = E_y$;
 - 若 $\Delta E > 0$, 表示 $E_x > E_y$;
 - 若 $\Delta E < 0$, 表示 $E_x < E_y$ 。
 - 当 $E_x \neq E_y$ 时, 要通过尾数的移动以改变 E_x 或 E_y , 使之相等。
- $[x]_{\text{浮}} \rightarrow E_x = 00010, M_x = 0.11011011$
- $[y]_{\text{浮}} \rightarrow E_y = 00100, M_y = 1.01010100$ (补码);
- 阶差 $= [E_x]_{\text{补}} - [E_y]_{\text{补}} = 00010 - 00100 = 00010 + 11100$ (取反加一) $= 11110$
- 即阶差为-2, M_x 右移两位, E_x 加2。
- $[x]_{\text{浮, 对阶}} \rightarrow E_x = 00100, M_x = 0.00110110(11)$, 阶码精度



浮点加减法示例一 (2)

3、尾数相加 (带符号位)

$$\begin{array}{r} 0.00110110(11) \\ + 1.01010100 \\ \hline \end{array}$$

$$1.10001010(11)$$

4、1.M规格化

● 纯小数格式应为 (+/-) 0.1XXXXXX

● 规则

● 尾数右移1位, 阶码加1

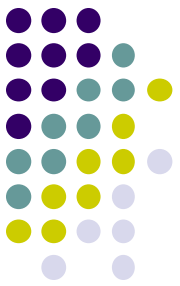
● 尾数左移1位, 阶码减1

$$0.01 * 10^5 = 0.1 * 10^4$$

正数1前; 负数1后

$$E_x = 00100$$

2020/3/7 左规处理, 结果为1.00010101(10), 阶码为00011



浮点加减法示例一 (3)

- 舍入处理 (对阶和向右规格化时)
 - 就近舍入(0舍1入):类似“四舍五入”,丢弃的最高位为1,进1
 - 朝0舍入:截尾
 - 朝 $+\infty$ 舍入:正数多余位不全为“0”,进1;负数,截尾
 - 朝 $-\infty$ 舍入:负数多余位不全为“0”,进1;正数,截尾

采用0舍1入法处理, 得到1.00010110。

1.00010101(10)

- 溢出判断和处理
 - 阶码上溢, 一般将其认为是 $+\infty$ 和 $-\infty$ 。
 - 阶码下溢, 则数值为0。

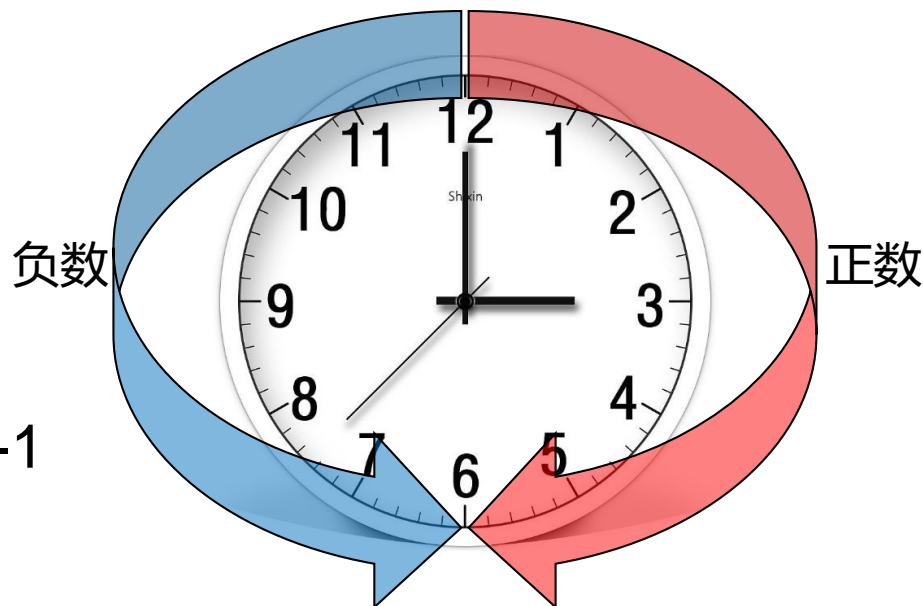
阶码符号位为00, 不溢出。得最终结果为

$$x+y = 2^{011} \times (-0.11101010)$$



补码舍入处理

- 就近舍入（0舍1入）
 - 正数→四舍五入法
 - 负数
 - $[x]_{\text{补}} = 1\ 110. (1)$
 - $[x]_{\text{原}} = 1\ 001. (1) = -1.5 \rightarrow -1$
 - $[x\text{-舍入}]_{\text{原}} = 1\ 001$
 - $[x\text{-舍入}]_{\text{补}} = 1\ 111$





$$x=0.1101 \times 2^{01} ; y=-0.1010 \times 2^{11}$$

尾数和阶符都采用补码表示，都采用双符号位表示法，就近舍入规则。求 $x+y=?$

- ☐ A 0.1011×2^{01}
- ☐ B 0.1011×2^{10}
- ☐ C 0.1101×2^{01}
- ☒ D 0.1101×2^{10}



习题求解过程

$[x]_{\text{浮}} = 0001, 00.1101$

$[y]_{\text{浮}} = 0011, 11.0110$

阶差 = 1110, 即为 -2

M_x 应当右移 2 位,

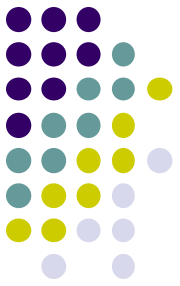
$[x]_{\text{浮}} = 0011, 00.0011 (01)$

尾数和为 11.1001 (01)

左规 11.0010 (10), 阶码减 1 为 0010

舍入 (就近舍入) 11.0011 丢弃 10

$x + y = -0.1101 * 2^{10}$



复习

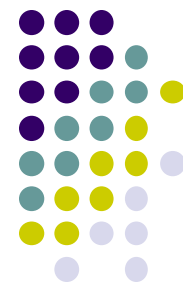
- 定点乘法
- 定点除法
- 浮点加减法
- 浮点乘除法（规格化、舍入）



浮点乘法和除法运算示例一

[例30] 设有浮点数 $x = 2^{-5} \times 0.0110011$, $y = 2^3 \times (-0.1110010)$

- 求 $[x \times y]_{\text{浮}}$ 。
- 尾数(含符号位)用8位补码表示。
- 要求用补码完成尾数乘法运算,运算结果尾数保留高8位(含符号位),并用尾数低位字长值处理舍入操作。



[解:]

阶码采用双符号位,尾数原码采用单符号位,则有

$$[M_x]_{\text{原}} = 0.0110011, [M_y]_{\text{原}} = 1.1110010$$

$$[E_x]_{\text{补}} = 11011, [E_y]_{\text{补}} = 00011$$

$$[x]_{\text{浮}} = 11011, 0.0110011, [y]_{\text{浮}} = 00011, 1.1110010$$

(1) 求阶码和: $[E_x]_{\text{补}} + [E_y]_{\text{补}} = 11011 + 00011 = 11110$ (补码形式-2)

(2) 尾数乘法运算可采用原码阵列乘法器实现, 即有

$$\begin{aligned} [M_x]_{\text{原}} \times [M_y]_{\text{原}} &= [0.0110011]_{\text{原}} \times [1.1110010]_{\text{原}} \\ &= [1.0101101, 0110110]_{\text{原}} \end{aligned}$$

规格化: 1前

(3) 规格化处理: 乘积不是规格化的数, 需要左规。尾数左移1位变为1.1011010, 1101100, 阶码变为11101 (-3)。

(4) 舍入处理: 尾数为负数, 取高位字长, 按舍入规则舍去低位字长, 故尾数为1.1011011。

$$\text{最终相乘结果为 } [x \times y]_{\text{浮}} = 11101, 1.1011011$$

$$\text{其真值为 } x \times y = 2^{-3} \times (-0.1011011)$$



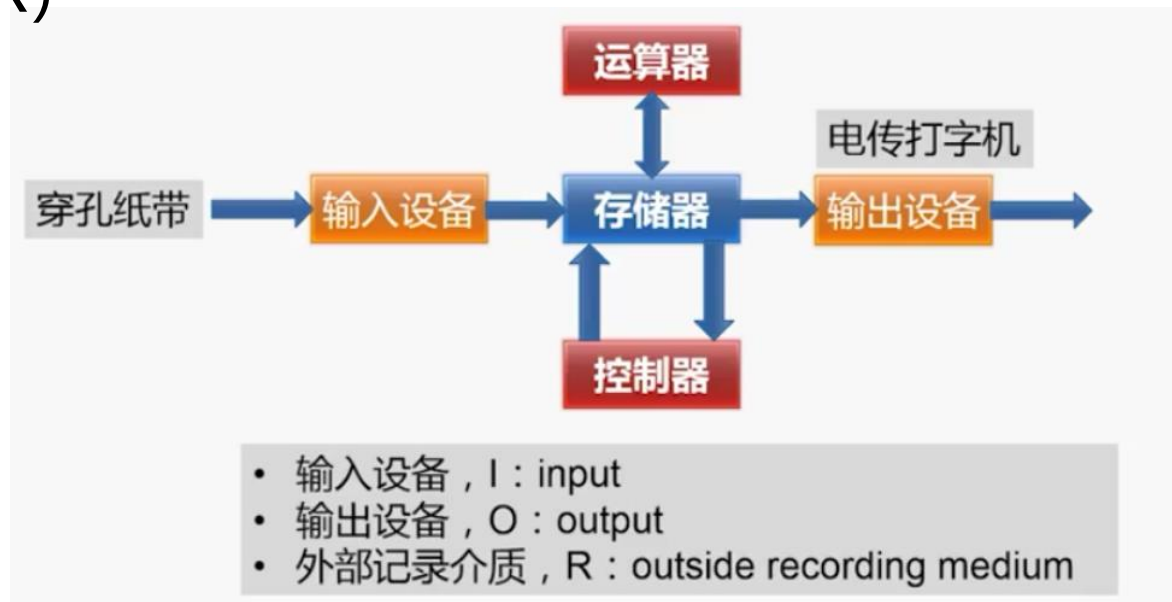
第三章 多层次的存储器

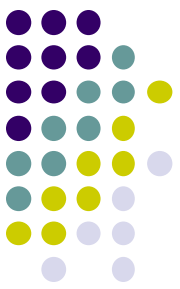
- 存储器分类
- 存储器技术指标
- 层次缓存结构
- 其他分层存储结构



冯·诺依曼结构

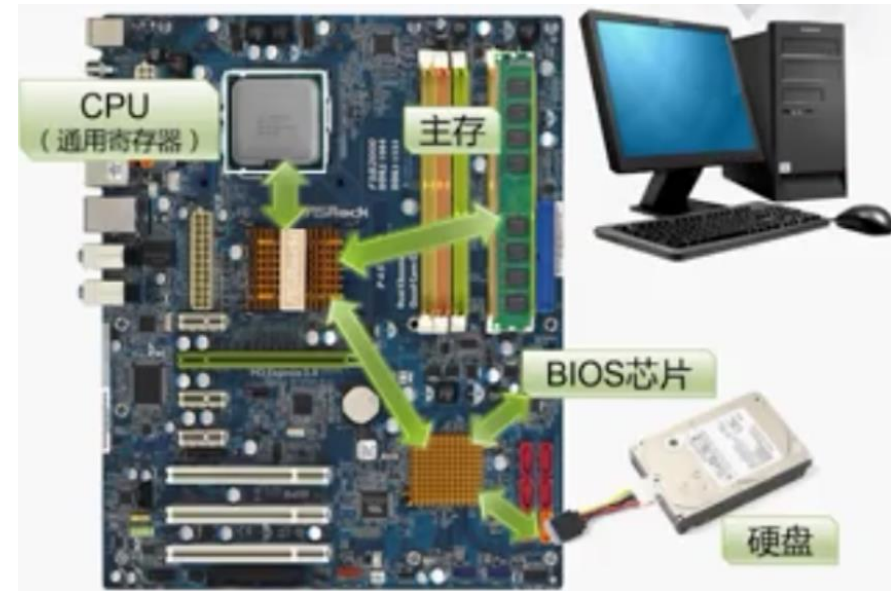
- **运算器**, CA (Central Arithmetical)
- 控制器, CC (Central Control)
- **存储器**: M (Memory)
- 输入设备, I (Input)
- 输出设备, O (Output)
- **外部记录介质** (R)





存储器分类与特性（1）

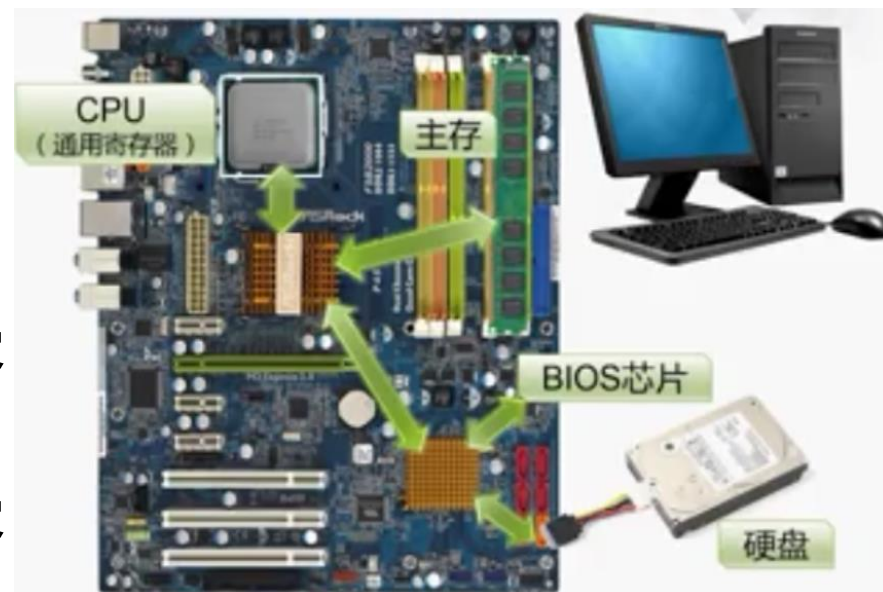
- 内容是否易失
 - 非易失
 - 硬盘、BIOS
 - 易失性
 - 内存、Cache
- 可读可写
 - BIOS（只读）
 - 内存、硬盘等（可读可写）
- 随机访问/顺序访问
 - 磁带、硬盘
 - 内存、Cache、BIOS





存储器分类与特性（2）

- 主要性能
 - 容量
 - 访问速度
 - 价格
- 现有存储器特性
 - 速度快的存储器价格贵，容量小
 - 价格低的存储器速度慢，容量大
- 在存储器系统设计时，应当在存储器容量、速度和价格方面的因素作折中考虑

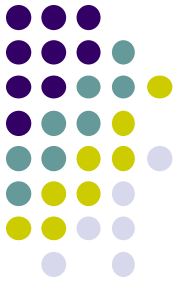


此题未设置答案，请点击右侧设置按钮



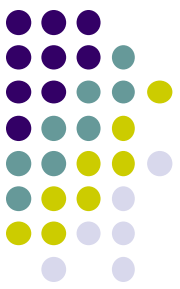
BIOS芯片属于？

- ☐ A 易失性、可读可写存储器
- ☐ B 非易失性、可读可写存储器
- ☒ C 易失性、只读存储器
- ☐ D 非易失性、只读存储器



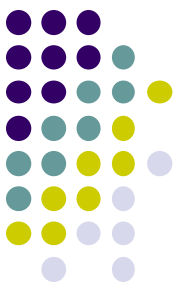
第三章 多层次的存储器

- 存储器分类
- 存储器技术指标
- 层次缓存结构
- 其他分层存储结构



存储器技术指标（2）

- 存取时间：
 - 存储器访问时间，
 - 是从存储器接收到读/写命令开始到信息被读出或写入完成所需的时间
- 存储周期（存取周期）
 - 存储器连续读写过程中一次完整的存取操作所需的时间
 - CPU连续两次访问存储器的最小间隔时间
 - 通常，存储周期略大于存取时间
- 存储器带宽（数据传送速率，频宽）
 - 单位时间里存储器所存取的信息量，通常以位/秒或字节/秒做度量单位
 - 若系统的总线宽度为W位，则带宽= $W/\text{存取周期}$ （b/s）



存储器技术指标（1）

- 存储容量

- 存储容量指一个存储器中可存储的信息比特数，常用比特数（b）或字节数(B)来表示，也可使用KB、MB、GB、TB等单位
- $1KB=2^{10}B$, $1MB=2^{20}B$, $1GB=2^{30}B$, $1TB=2^{40}B$
- 存储容量也可表示为
 - 存储字数（存储单元数）×存储字长（每单元的比特数）
 - 1Mb容量的存储器可以组织成 $1M \times 1 \text{ bit}$ ，也可组织成 $128K \times 8 \text{ bit}$ 或者 $512K \times 4 \text{ bit}$ 。

此题未设置答案，请点击右侧设置按钮



小明用大小为4GB的U盘去存储大小为3.8GB的文件，是否能正确存储？

A 能

B 不能

4,001,333,248 字节
 $\approx 3.72 \times 1024^3 \text{ B}$
 $= 3.72 \text{ GB}$



此题未设置答案，请点击右侧设置按钮



在32位CPU与操作系统的主机中，装入了4GB大小的内存，此时访存最大字节数为？

- A 2^{32}
- B 4×10^9

内部存储器容量

- 高速缓存：32KB = 32×1024 Byte
- 内存：2GB = 2×1024^3 Byte

外部存储器容量

- 优盘：4GB = 4×1000^3 Byte
- 硬盘：1TB = 1×1000^3 Byte

此题未设置答案，请点击右侧设置按钮



CPU主频为3GHz，对应的具体频率为(Hz)?

A

$$3 \times 2^{30}$$

B

$$3 \times 10^9$$

数据传输率、时钟频率

- 以太网：100Mbps = 100×1000^2 bits per second
- SATA-2：3Gbit/s = 3×1000^2 bits per second
- CPU主频：3GHz = 3×1000^3 Hz



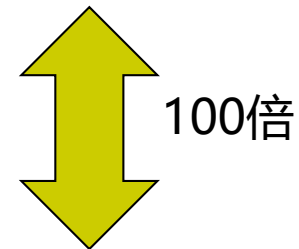
第三章 多层次的存储器

- 存储器分类
- 存储器技术指标
- 层次缓存结构
- 其他分层存储结构



CPU与DRAM发展

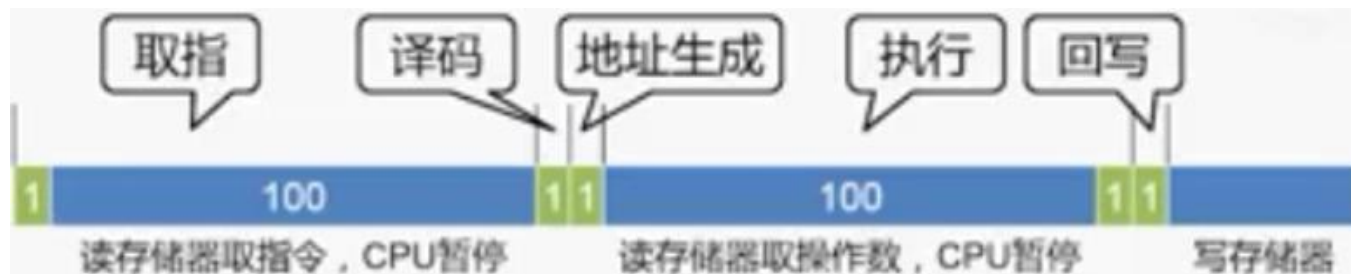
| CPU | | 1980 | 1990 | 2000 | 2010 | 2010:1980 |
|-----|--------------------------|-------|------|------------|---------|-----------|
| | Name | 8080 | 386 | Pentium II | Core i7 | / |
| | Clock rate(MHz) | 1 | 20 | 600 | 2,500 | 2,500 |
| | Cycle time(ns) | 1,000 | 50 | 1.6 | 0.4 | 2,500 |
| | Cores | 1 | 1 | 1 | 4 | 4 |
| | Effective Cycle time(ns) | 1,000 | 50 | 1.6 | 0.1 | 10,000 |



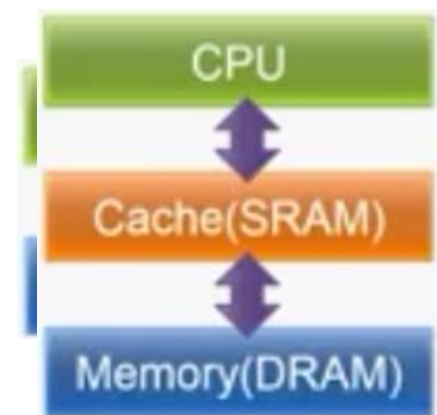
| DRAM | \$/MB | 8,000 | 100 | 1 | 0.06 | 130,000 |
|------|------------------|-------|-----|----|-------|---------|
| | access time(ns) | 375 | 100 | 60 | 40 | 9 |
| | typical size(MB) | 0.064 | 4 | 64 | 8,000 | 125,000 |



存储器性能影响



- 考虑指令执行过程 `ADD R0, [6]`
 - 取指：读指令，100时钟周期
 - 译码：1时钟周期
 - 执行：访存，100时钟周期
 - 回写：1时钟周期
- 性能下降100倍以上





CPU、SRAM与DRAM

| CPU | | 1980 | 1990 | 2000 | 2010 | 2010:1980 |
|-----|--------------------------|-------|------|------------|---------|-----------|
| | Name | 8080 | 386 | Pentium II | Core i7 | / |
| | Clock rate(MHz) | 1 | 20 | 600 | 2,500 | 2,500 |
| | Cycle time(ns) | 1,000 | 50 | 1.6 | 0.4 | 2,500 |
| | Cores | 1 | 1 | 1 | 4 | 4 |
| | Effective Cycle time(ns) | 1,000 | 50 | 1.6 | 0.1 | 10,000 |

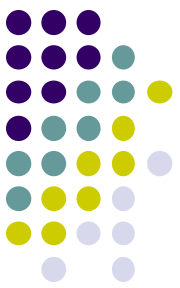


3~4倍

| SRAM | \$/MB | 19,200 | 320 | 100 | 60 | 320 |
|------|-----------------|--------|-----|-----|-----|-----|
| | access time(ns) | 300 | 35 | 3 | 1.5 | 200 |



| DRAM | \$/MB | 8,000 | 100 | 1 | 0.06 | 130,000 |
|------|------------------|-------|-----|----|-------|---------|
| | access time(ns) | 375 | 100 | 60 | 40 | 9 |
| | typical size(MB) | 0.064 | 4 | 64 | 8,000 | 125,000 |



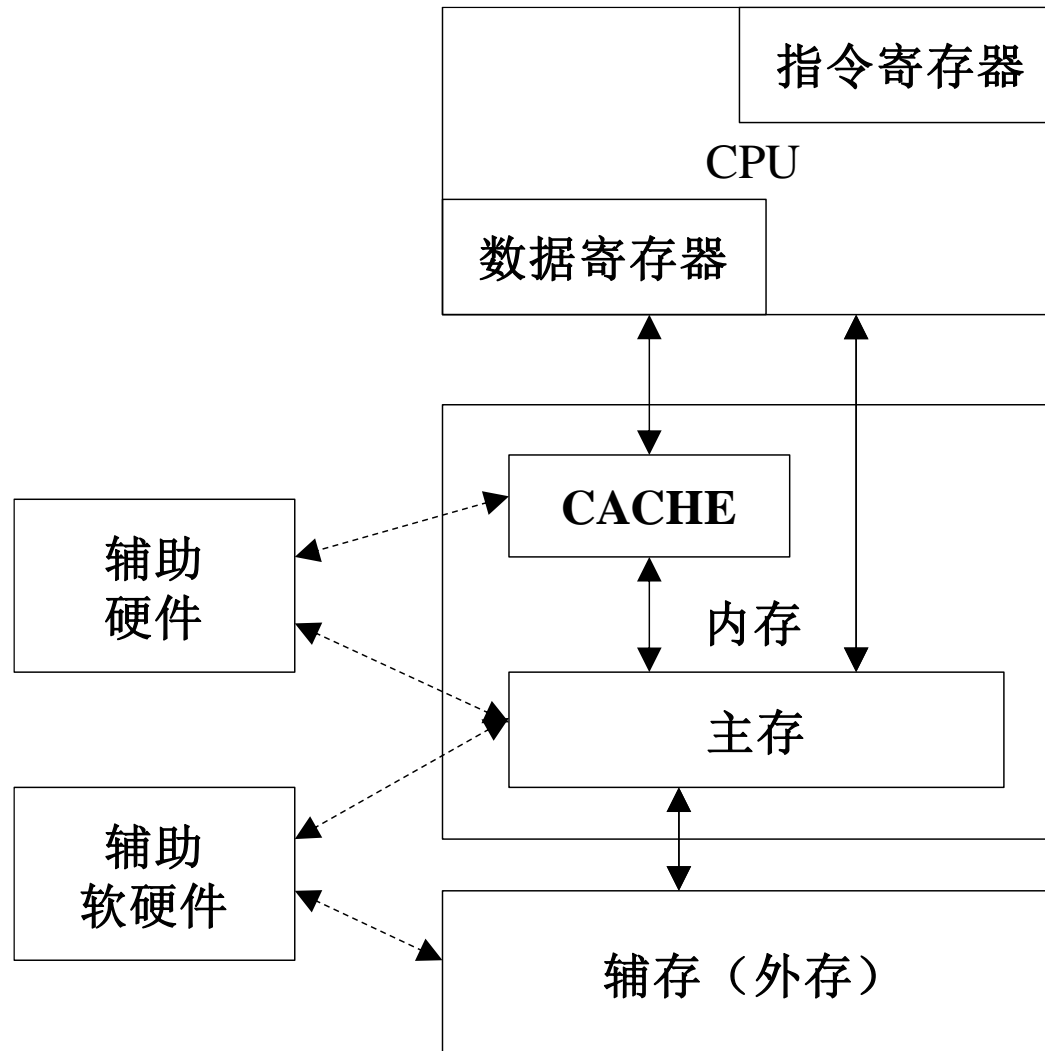
Cache发展趋势

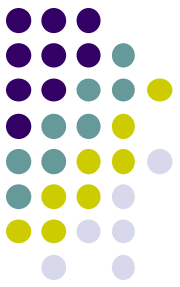
- 高速缓冲存储器简称Cache，它是计算机系统中的一个高速小容量半导体存储器。

| CPU | 典型主频 | 访存周期 | DRAM延迟 | Cache设计 |
|-------|--------------------|--------------|--------|----------------|
| 8088 | 4.77MHz (210ns) | 4 (840ns) | 250ns | 无需Cache |
| 80286 | 10MHz (100ns) | 2 (200ns) | 220ns | 无需Cache |
| 80386 | 25MHz (40ns) | 2 (80ns) | 190ns | 片外Cache |
| 80486 | 33MHz (30ns) | 2 (60ns) | 165ns | 8KB片内 Cache |



多级存储系统（片外Cache）

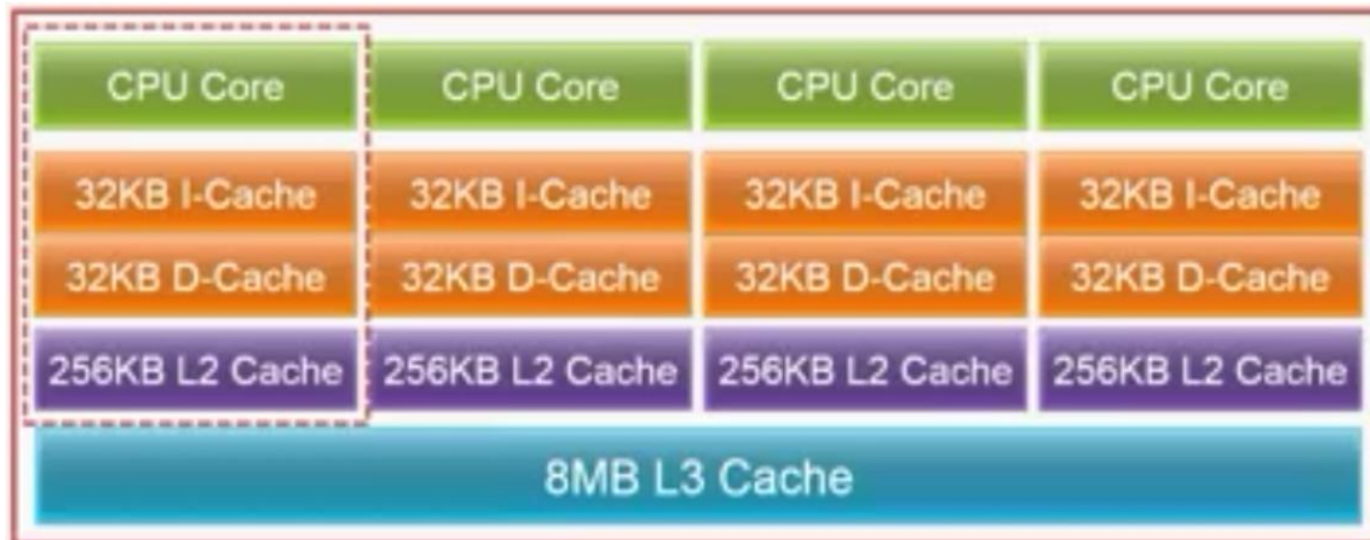




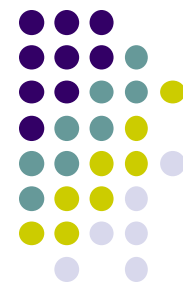
片内Cache—Intel Core i7



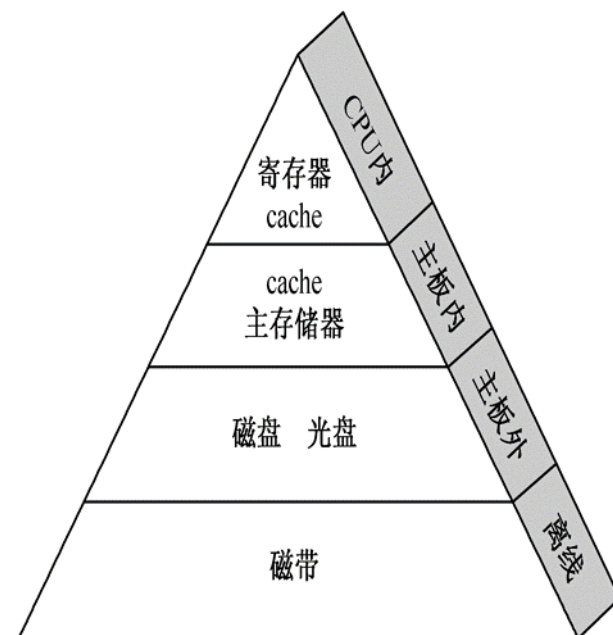
- 一级Cache：数据与指令分离
 - 4周期，32KB
- 二级Cache：统一数据与指令
 - 11周期，256KB
- 三级Cache：多核共享
 - 30~40周期，8MB



存储器分级结构



- 金字塔形
 - 容量大、速度慢（底层）
 - 容量小、速度快（高层）
- 核心关键
 - 调用缓存方法
 - 高效利用多层缓存结构

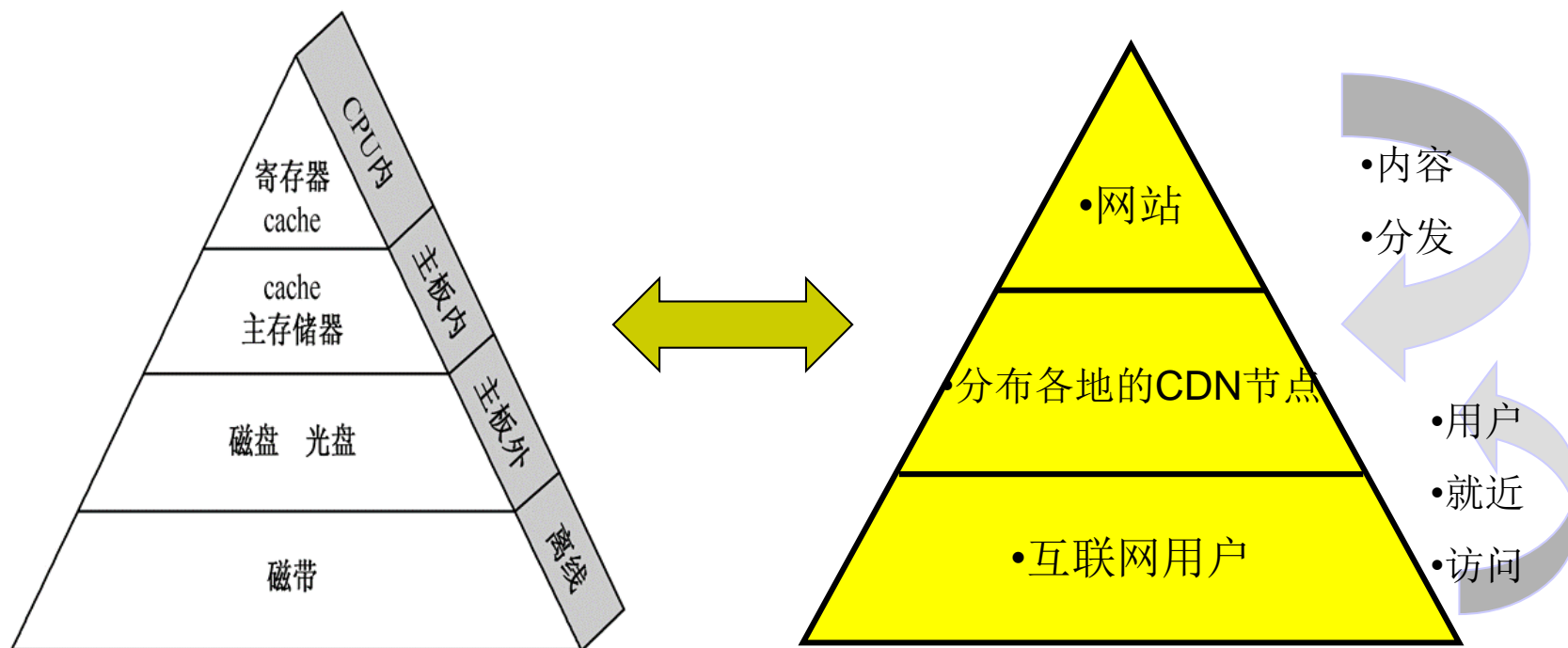
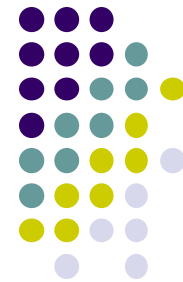


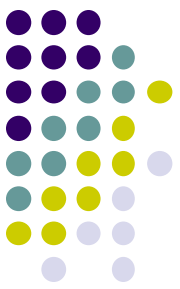


第三章 多层次的存储器

- 存储器分类
- 存储器技术指标
- 层次缓存结构
- 其他分层存储结构

基本业务模式类比（CDN）





基于CDN技术的流媒体方案

- **CDN** （ Content Delivery Networks ）
 - 内容分发网络
 - 通过在现有的**Internet**中增加一层新的网络架构，将网站或服务商的内容发布到最接近用户的网络“边缘”，使用户可以就近取得所需的内容，提高用户访问网站的响应速度。
- **CDN**是构建在现有网络基础上的一张网中网，为内容的分发提供高效、高质量的“高速公路”
- 该技术在**1998**年由美国最著名的高校**MIT**的教授提出，其成立的公司**Akamai**目前已成为一家市值超过**40**亿美金的世界**500**强企业

为什么要有CDN?



- CDN的好处
 - 降低客户建设网站基础设施的投资，并减小运维成本
 - 由于内容被分发到离用户更近的地方，因此可以绕过网络中的拥塞
 - 提高内容分发的质量、速度以及可靠性
 - 减轻源服务器负载
 - 通过统计复用，降低用户的服务成本

CDN的发展历程

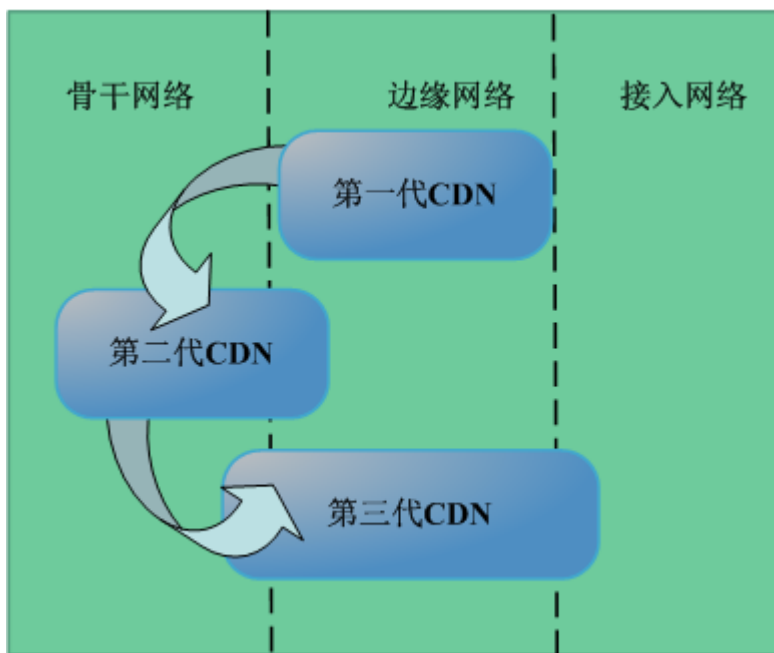


- CDN发展的推动力

- 新应用的拉动
- 技术的发展

- CDN发展的阶段划分

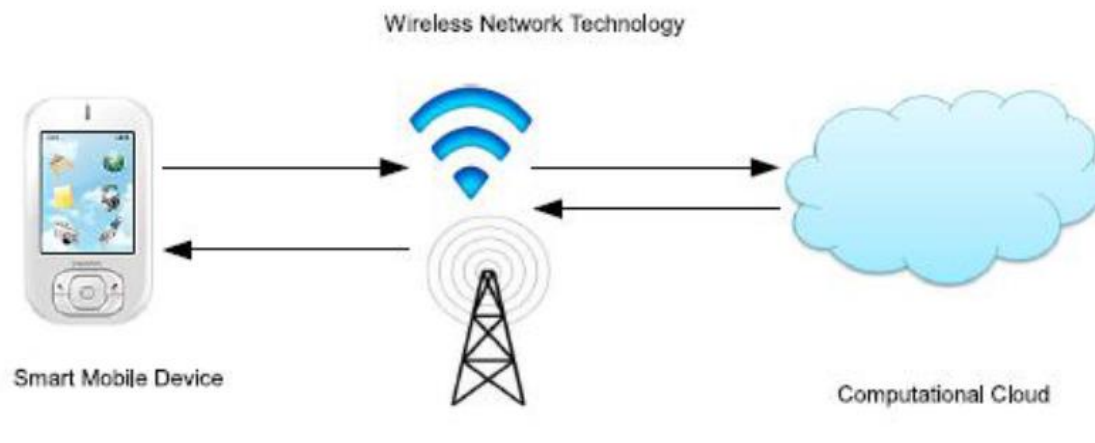
- 第一代CDN: Cache-assisted CDN
 - 发展时间: 1998年~2001年
 - 代表公司: Akamai
- 第二代CDN: Fiber-assisted CDN
 - 发展时间: 2001年~2006年
 - 代表公司: Limelight
- 第三代CDN: Peer-assisted CDN
 - 发展时间: 2006年~
 - 代表公司: ChinaCache



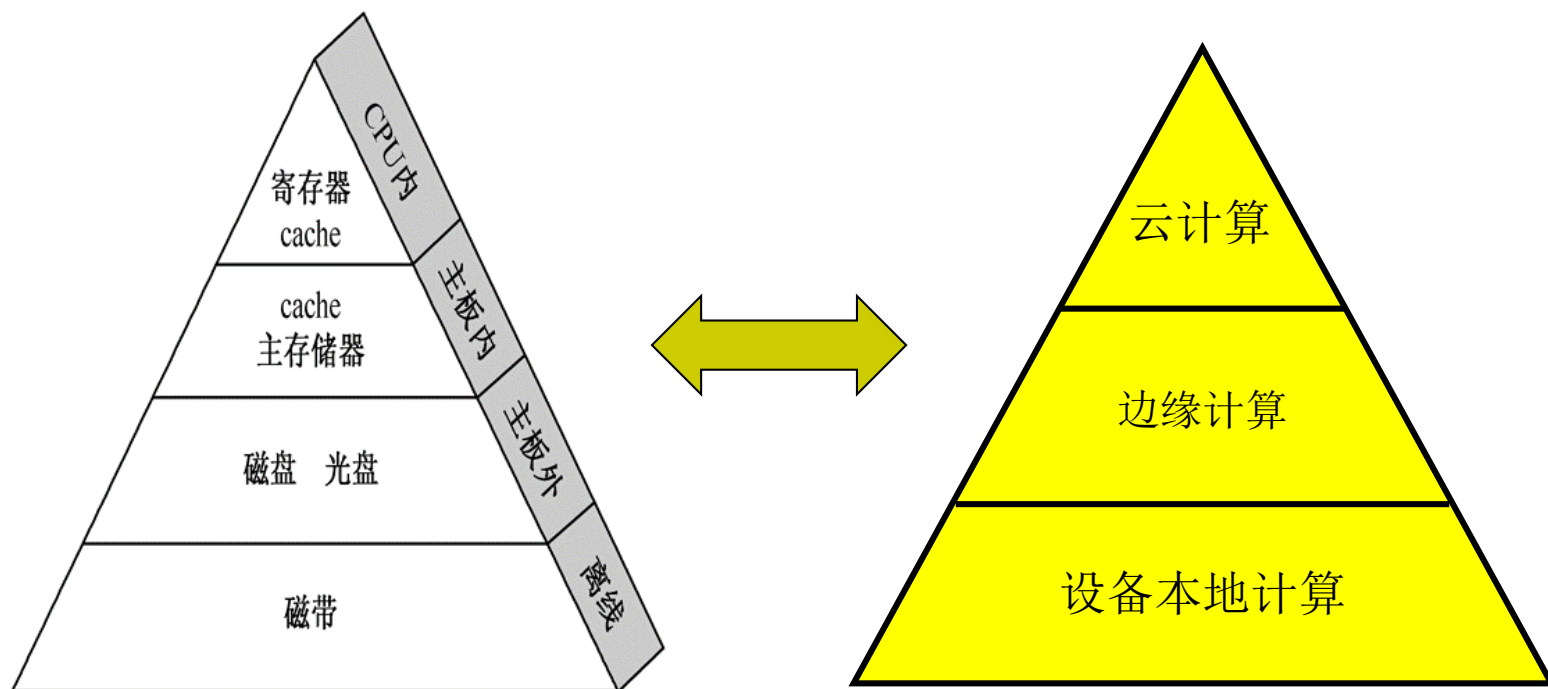
存储服务 v.s. 计算服务



- 由于移动设备计算/存储能力受限，需要将复杂移动应用迁移到云服务提供商处理
 - 提升用户体验
 - 减少能耗，缩短执行时间
- 支持复杂移动应用
 - 自动驾驶、语音识别、增强现实等



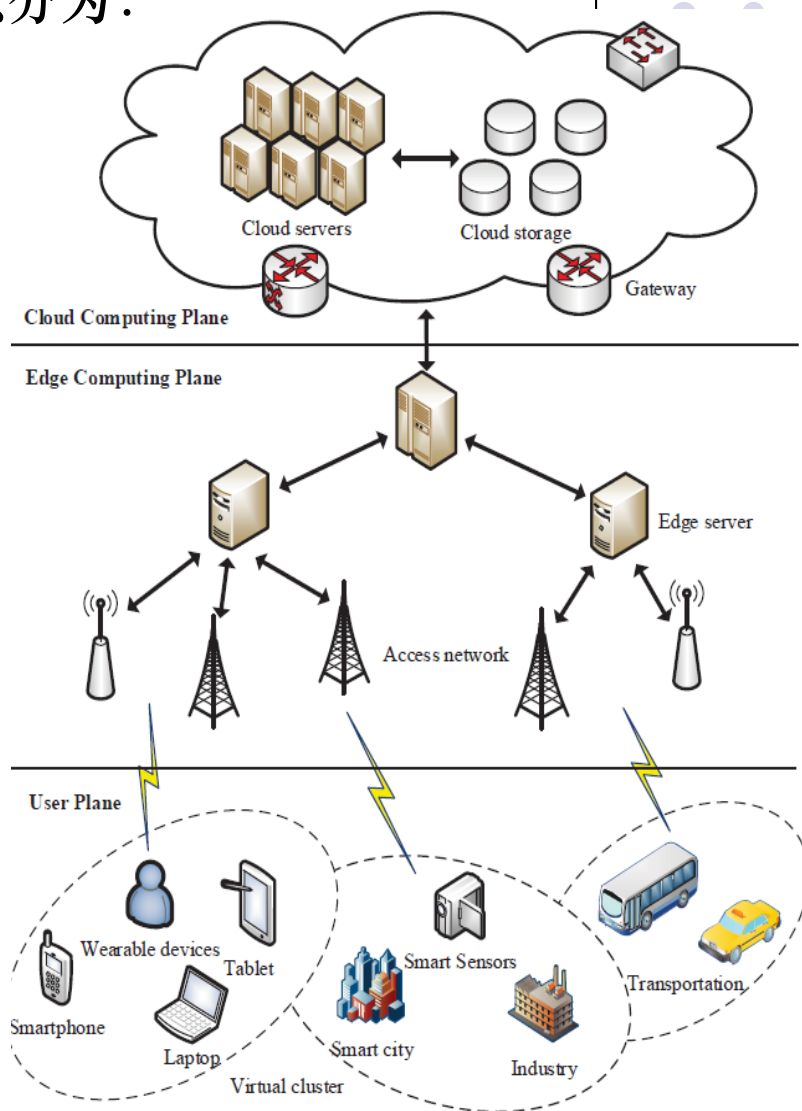
基本业务模式类比（MEC）



移动云计算/边缘计算

根据计算资源与用户的距离，计算服务可以分为：

- 移动云计算（2009-2014）
 - 服务由传统云服务商提供，云服务器位于核心网
 - 数据传输/结果返回需经过回传链路，无法保障服务时延
- 移动边缘计算（2014-至今）
 - 服务由运营商提供，计算资源部署在网络边缘设备(如宏基站/小小区/交换机/路由器/网关等)
 - 直接在网络边缘提供计算服务
- 雾计算/多接入边缘计算

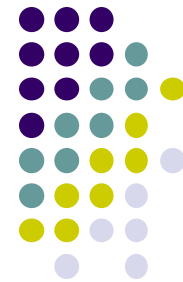


边缘计算特点



- 低时延、高可靠的计算服务
 - 移动边缘计算缩短用户与云服务资源距离，避免移动云计算中核心网（WAN）传输时延
- 高带宽、低回传开销
 - 将数据处理从云服务器分布至各网络设备，在提升网络数据处理能力同时，降低数据回传消耗
- 鲁棒性、安全性
 - 分布式节点进行任务处理，避免单点故障，系统鲁棒性强
 - 分散受攻击节点

总结



- 存储器分类
 - 易失、随机、可读可写
- 存储器性能指标
- 存储器分层结构
- 类比分层结构
 - 内容分发网络
 - 移动边缘计算

