

系统分析师

《基础知识部分》



(希) 希赛

计算机组成与体系结构



课程内容提要

(CS) 希赛

- 编码及浮点数运算 (★)
- 计算机体系结构 (★★★★)
- 流水线技术 (★★★★)
- 嵌入式系统 (★)
- 存储系统 (★★★★★★)
- 总线 (★★)
- 校验码 (★)



浮点数运算

(希赛)

浮点数表示：

$$N = \text{尾数} * \text{基数}^{\text{指数}}$$

运算过程：

对阶 > 尾数计算 > 结果格式化

特点：

- 1、一般尾数用补码，阶码用移码
- 2、阶码的位数决定数的表示范围，位数越多范围越大
- 3、尾数的位数决定数的有效精度，位数越多精度越高
- 4、对阶时，小数向大数看齐
- 5、对阶是通过较小数的尾数右移实现的



浮点运算

(一) 希赛

浮点数在机器中的表示形式如下所示，若阶码的长度为 e ，尾数的长度为 m ，则以下关于浮点表示的叙述中，正确的是（ ）。

阶符	阶码	数符	尾数
----	----	----	----

- ① e 的值影响浮点数的范围， e 越大，所表示的浮点数值范围越大
- ② e 的值影响浮点数的精度， e 越大，所表示的浮点数精度越高
- ③ m 的值影响浮点数的范围， m 越大，所表示的浮点数范围越大
- ④ m 的值影响浮点数的精度， m 越大，所表示的浮点数精度越高

A ①③

B ②③

C ①④

D ②④



计算机体系结构 – Flynn分类法

(一) 希赛

体系结构类型	结构	关键特性	代表
单指令流单数据流 SISD	控制部分：一个 处 理 器：一个 主存模块：一个		单处理器系统
单指令流多数据流 SIMD	控制部分：一个 处 理 器：多个 主存模块：多个	各处理器以异步的形式执行同一条指令	并行处理机 阵列处理机 超级向量处理机 (GPU)
多指令流单数据流 MISD	控制部分：多个 处 理 器：一个 主存模块：多个	被证明不可能，至少是不实际	目前没有，有文献称流水线计算机为此类
多指令流多数据流 MIMD	控制部分：多个 处 理 器：多个 主存模块：多个	能够实现作业、任务、指令等各级全面并行	多核处理器 (SMP、BMP、MP) 多处理机系统 (MPP) 多计算机



计算机体系结构 – Flynn分类法

(希赛)

Flynn分类法根据计算机在执行程序的过程中（ ）的不同组合，将计算机分为4类。当前主流的多核计算机属于（ ）计算机。

- A 指令流和数据流
- B 数据流和控制流
- C 指令流和控制流
- D 数据流和总线带宽

- A SISD
- B SIMD
- C MISD
- D MIMD



计算机体系结构 – Flynn分类法

(希赛)

以下关于CPU和GPU的叙述中，错误的是（ ）。

- A CPU适合于需要处理各种不同的数据类型、大量的分支跳转及中断等场合
- B CPU利用较高的主频、高速缓存（Cache）和分支预测等技术来执行指令
- C GPU采用MISD（Multiple Instruction Single Data）并行计算架构
- D GPU的特点是比CPU包含更多的计算单元和更简单的控制单元



计算机体系结构 – Flynn分类法

(一) 希赛

关于大规模并行处理器MPP，以下说法不正确的是（ ）。

- A 大多数MPP系统使用标准的CPU作为它们的处理器
- B 其互连网络通常采用商用的以太网实现
- C 是一种异步的分布式存储器结构的MIMD系统
- D 使用特殊的硬件和软件来监控系统、检测错误并从错误中平滑地恢复



计算机体系结构 – CISC与RISC

(希赛)

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；硬布线逻辑控制为主；适合采用流水线	优化编译，有效支持高级语言

复杂指令集计算机 (CISC, Complex Instruction Set Computers)

精简指令集计算机 (RISC, Reduced Instruction Set Computers)



计算机体系结构 – CISC与RISC

(希赛)

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

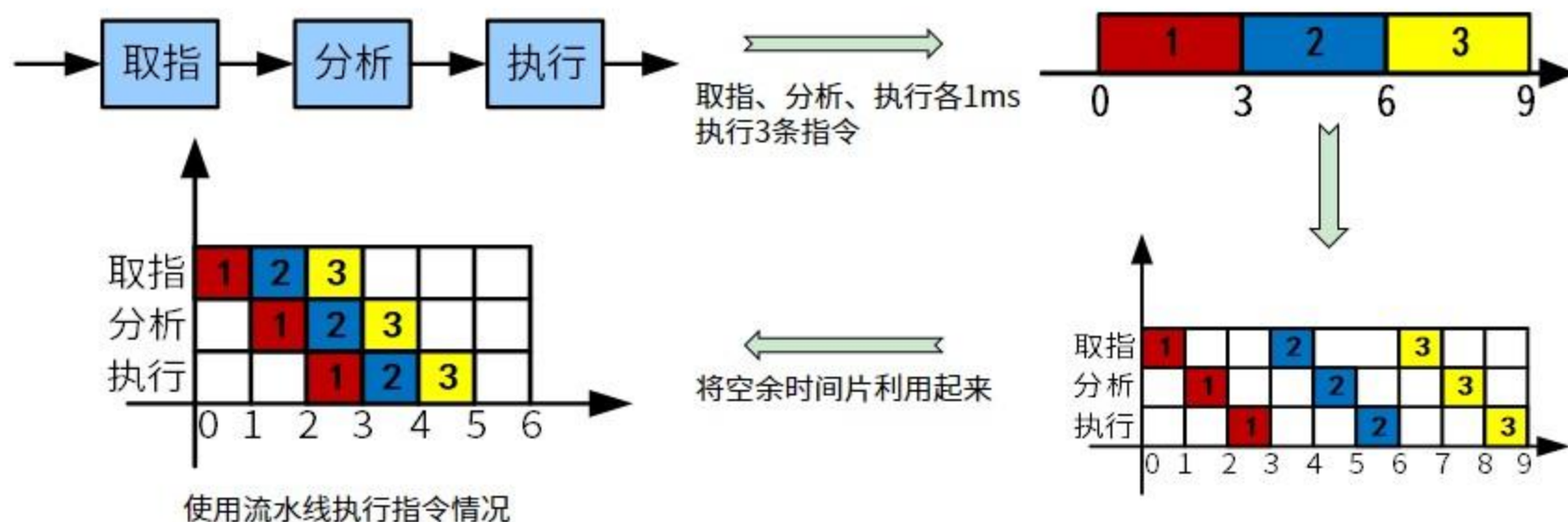


流水线

(一) 希赛

☀ 相关参数计算：流水线执行时间计算、流水线吞吐率、流水线加速比等。

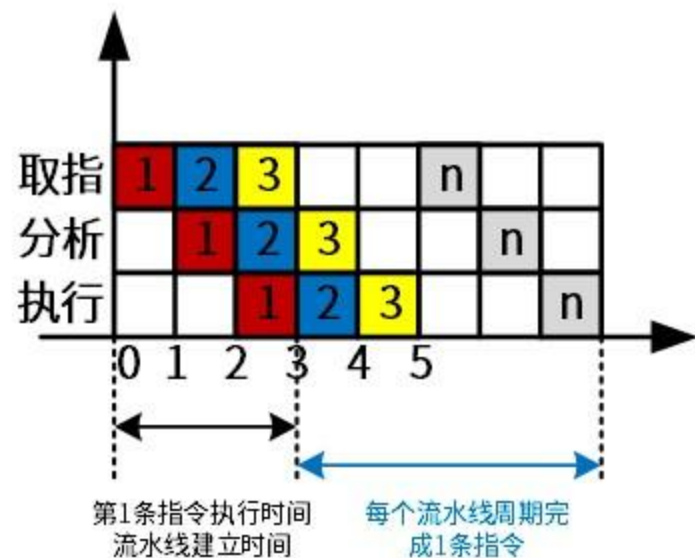
☀ 流水线是指在程序执行时多条指令重叠进行操作的一种准并行处理实现技术。各种部件同时处理是针对不同指令而言的，它们可同时为多条指令的不同部分进行工作，以提高各部件的利用率和指令的平均执行速度。





流水线

(希赛)



✓ 流水线周期为执行时间最长的一段

✓ 流水线计算公式为：

1条指令执行时间 + (指令条数-1) * 流水线周期

①理论公式: $(t_1+t_2+\dots+t_k)+(n-1)*t$

②实践公式: $k*t+(n-1)*t$

一条指令的执行过程可以分解为取指、分析和执行三步，在取指时间 $t_{\text{取指}}=3\Delta t$ 、分析时间 $t_{\text{分析}}=2\Delta t$ 、执行时间 $t_{\text{执行}}=4\Delta t$ 的情况下，若按串行方式执行，则10条指令全部执行完需要 () Δt ；若按流水线的方式执行，流水线周期为 () Δt ，则10条指令全部执行完需要 () Δt 。

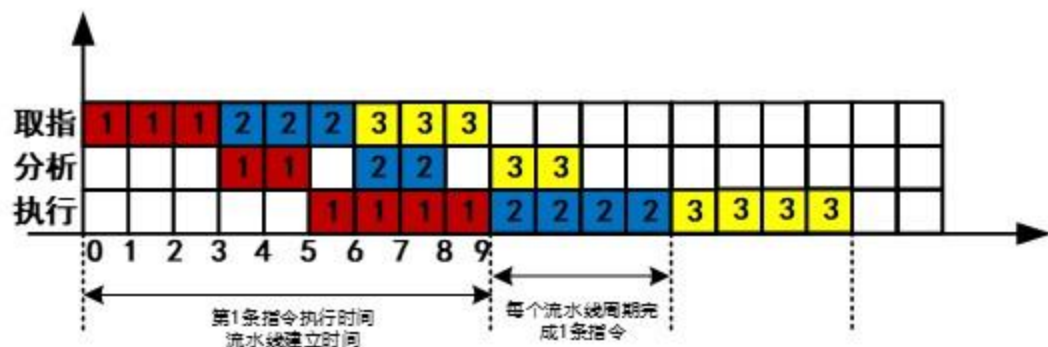


流水线

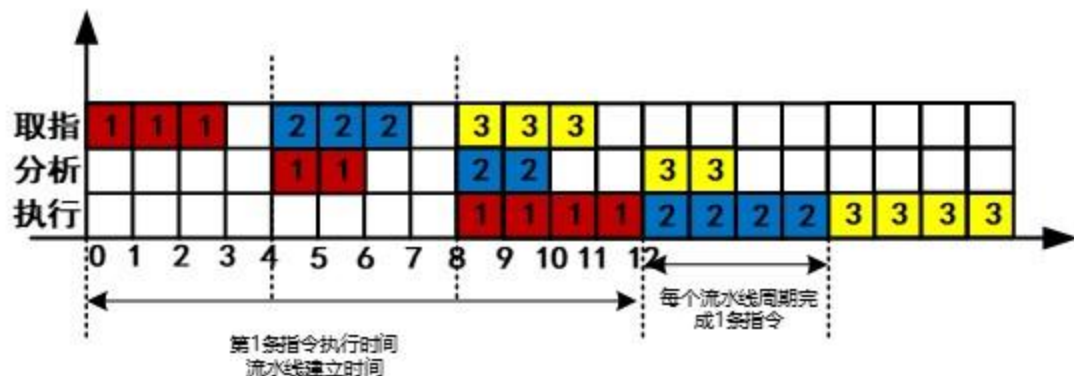
(一) 希赛



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



实践公式: $k * t + (n-1) * t$





流水线

(一) 希赛

☀ 流水线的吞吐率 (Through Put rate, TP) 是指在单位时间内流水线所完成的任务数量或输出的结果数量。计算流水线吞吐率的最基本的公式如下:

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

☀ 流水线最大吞吐率:

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

一条指令的执行过程可以分解为取指、分析和执行三步, 取指时间 $t_{\text{取指}}=3\Delta t$ 、分析时间 $t_{\text{分析}}=2\Delta t$ 、执行时间 $t_{\text{执行}}=4\Delta t$ 。10条指令的吞吐率? 最大吞吐率?



流水线 – 流水线加速比计算

(希赛)

完成同样一批任务，不使用流水线所用的时间与使用流水线所用的时间之比称为流水线的加速比。计算流水线加速比的基本公式如下：

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



流水线

(希赛)

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 ()。

A $1/9 \Delta t$

B $1/3 \Delta t$

C $1/2 \Delta t$

D $1/1 \Delta t$

A 1:10

B 2:1

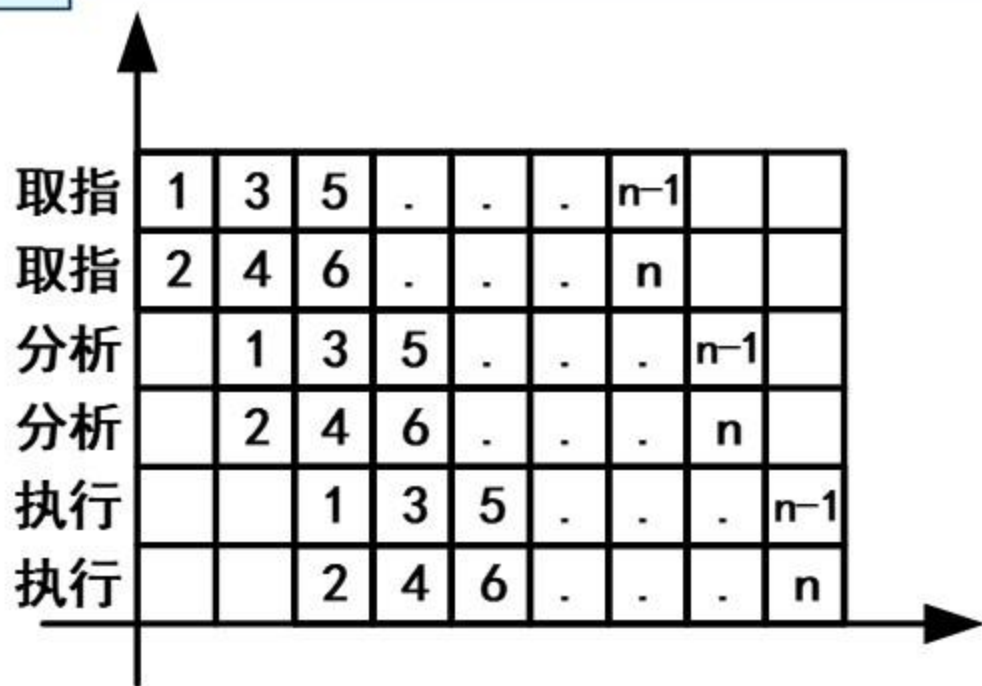
C 5:2

D 3:1



流水线 – 超标量流水线

(希赛)



一条指令的执行过程可以分解为取指、分析和执行三步，在取指时间 $t_{\text{取指}}=3\Delta t$ 、分析时间 $t_{\text{分析}}=2\Delta t$ 、执行时间 $t_{\text{执行}}=4\Delta t$ 的情况下，若按流水线的方式执行，流水线周期为（ ） Δt ，则10条指令全部执行完需要（ ） Δt 。

度为2的超标量流水线，执行时间为（ ） Δt 。

度为3的超标量流水线，执行时间为（ ） Δt 。



流水线

(一) 希赛

设每条指令由取指、分析、执行3个子部件完成，并且每个子部件的执行时间均为 Δt 。若采用常规标量单流水线处理机（即该处理机的度为1），连续执行16条指令，则共耗时（ ） Δt 。若采用度为4的超标量流水线处理机，连续执行上述16条指令，则共耗时（ ） Δt 。

- | | | | |
|------|------|------|------|
| A 16 | B 18 | C 32 | D 48 |
| A 4 | B 5 | C 6 | D 8 |



流水线

(一) 希赛





嵌入式-体系结构

(希赛)

冯·诺依曼结构

冯·诺依曼结构也称普林斯顿结构，是一种将程序指令存储器和数据存储器合并在一起的存储器结构。

特点：

- (1) 一般用于PC处理器，如I3，I5，I7处理器
- (2) 指令与数据存储器合并在一起
- (3) 指令与数据都通过相同的数据总线传输

哈佛结构

哈佛结构是一种将程序指令存储和数据存储分开的存储器结构。哈佛结构是一种并行体系结构，它的主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问。

特点：

- (1) 一般用于嵌入式系统处理器 (DSP) 数字信号处理 (DSP, Digital Signal Processing)
- (2) 指令与数据分开存储，可以并行读取，有较高数据的吞吐率
- (3) 有4条总线：指令和数据的数据总线与地址总线



嵌入式系统开发设计 - 芯片

(希赛)

1、DSP

DSP芯片，也称数字信号处理器，是一种特别适合于进行数字信号处理运算的微处理器，其主要应用是实时快速地实现各种数字信号处理算法。

2、SoC

System on Chip，简称SoC，也叫片上系统。从狭义角度讲，它是信息系统核心的芯片集成，是将系统关键部件集成在一块芯片上；从广义角度讲，SoC是一个微小型系统，如果说中央处理器（CPU）是大脑，那么SoC就是包括大脑、心脏、眼睛和手的系统。

3、MPU

微机中的中央处理器（CPU）称为微处理器（MPU），是构成微机的核心部件，也可以说是微机的“心脏”。它起到控制整个微型计算机工作的作用，产生控制信号对相应的部件进行控制，并执行相应的操作。

4、MCU

微控制单元（Microcontroller Unit；MCU），又称单片微型计算机（Single Chip Microcomputer）或者单片机，是把中央处理器（Central Process Unit；CPU）的频率与规格做适当缩减，并将内存（memory）、计数器（Timer）、USB、A/D转换、UART、PLC、DMA等周边接口，甚至LCD驱动电路都整合在单一芯片上，形成芯片级的计算机，为不同的应用场合做不同组合控制。



嵌入式

(希赛)

目前处理器市场中存在CPU和DSP两种类型处理器，分别用于不同场景，这两种处理器具有不同的体系结构，DSP采用（ ）。

- A 冯·诺依曼结构
- B 哈佛结构
- C FPGA结构
- D 与GPU相同的结构



嵌入式

(希赛)

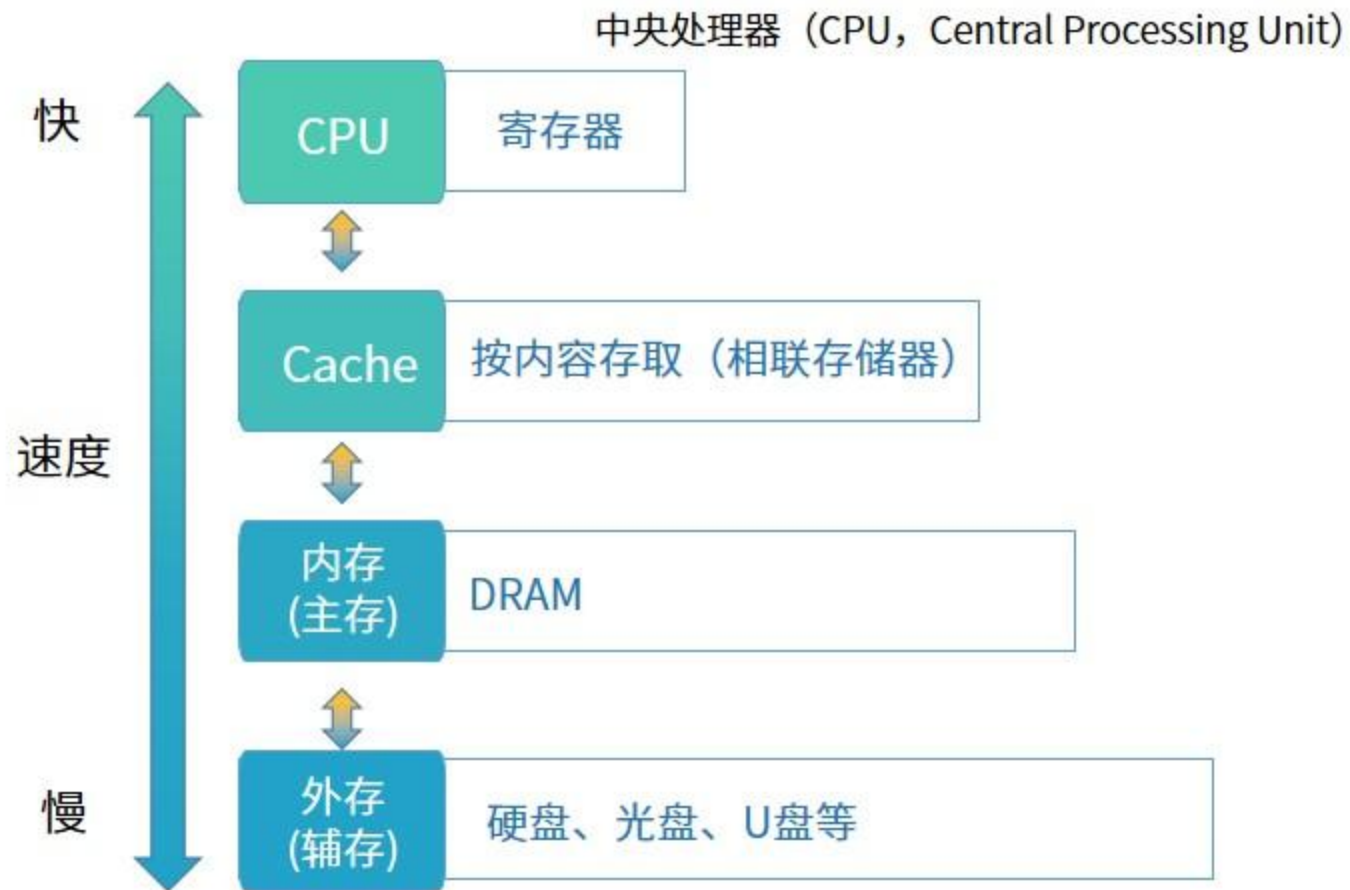
嵌入式处理器是嵌入式系统的核心部件，一般可分为嵌入式微处理器（MPU）、微控制器（MCU）、数字信号处理器（DSP）和片上系统（SOC）。以下叙述中，错误的是（ ）。

- A MPU在安全性和可靠性等方面进行增强，适用于运算量较大的智能系统
- B MCU典型代表是单片机，体积小从而使功耗和成本下降
- C DSP处理器对系统结构和指令进行了特殊设计，适合数字信号处理
- D SOC是一个有专用目标的集成电路，其中包括完整系统并有嵌入式软件的全部内容



存储系统-层次化存储结构

(希赛)





存储系统-层次化存储结构

(希赛)

计算机采用分级存储体系的主要目的是为了（ ）。

- A 解决主存容量不足的问题
- B 提高存储器读写可靠性
- C 提高外设访问效率
- D 解决存储的容量、价格和速度之间的矛盾



存储系统-Cache

(希赛)

- ✓ Cache的功能：提高CPU数据输入输出的速率，突破冯·诺依曼瓶颈，即CPU与存储系统间数据传送带宽限制。
- ✓ 在计算机的存储系统体系中，Cache是访问速度最快的层次。
- ✓ Cache对程序员来说是透明的。
- ✓ 使用Cache改善系统性能的依据是程序的局部性原理。

时间局部性

空间局部性



存储系统-Cache

(希赛)

- ✓ **时间局部性**：指程序中的某条指令一旦执行，不久以后该指令可能再次执行，典型原因是由于程序中存在大量的循环操作。
- ✓ **空间局部性**：指一旦程序访问了某个存储单元，不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址可能集中在一定的范围内，其典型情况是程序顺序执行。
- ✓ **工作集理论**：工作集是进程运行时被频繁访问的页面集合。

例：

```
int i,j,s=0,n=10000;  
for(i=1;i<=n;i++)  
    for(j=1;j<=n;j++)  
        s+=j;  
printf("结果为:%d",s)
```




存储系统-Cache

(希赛)

☀ 如果以 h 代表对Cache的访问命中率， t_1 表示Cache的周期时间， t_2 表示主存储器周期时间，以读操作为例，使用“Cache+主存储器”的系统的平均周期为 t_3 ，则：

$$t_3 = h \times t_1 + (1-h) \times t_2$$

其中， $(1-h)$ 又称为失效率（未命中率）。



存储系统

(希赛)

在嵌入式系统设计时，下面几种存储结构中对程序员是透明的是（ ）。

- A 高速缓存
- B 磁盘存储器
- C 内存
- D flash存储器



存储系统-Cache

(希赛)

以下关于Cache的叙述中，正确的是（ ）。

- A 在容量确定的情况下，替换算法的时间复杂度是影响Cache命中率的关键因素
- B Cache的设计思想是在合理的成本下提高命中率
- C Cache的设计目标是容量尽可能与主存容量相等
- D CPU中的Cache容量应大于CPU之外的Cache容量



Cache页面淘汰

(希赛)

映像方式：

- ✓ 直接相联映像：硬件电路较简单，但冲突率很高。
- ✓ 全相联映像：电路难于设计和实现，只适用于小容量的Cache，冲突率较低。
- ✓ 组相联映像：直接相联与全相联的折中。

- ✓ Cache页面淘汰算法
- ✓ Cache的读写过程

- ✓ 先进先出算法 (FIFO)
- ✓ 近期最少使用算法 (LRU)

- ✓ 写直达：同时写Cache与内存
- ✓ 写回：只写Cache，淘汰页面时，写回内存
- ✓ 标记法：只写入内存，并将标志位清0，若用到此数据，需要再次调取



存储系统-Cache

(希赛)

Cache的替换算法中，（ ）算法计数器位数多，实现困难。

- A FIFO
- B LFU
- C LRU
- D RAND



存储系统-主存编址

(希赛)

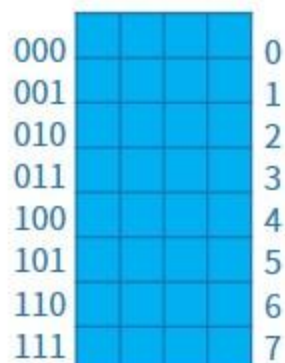
bit (比特位)



4bit (字长为4bit)



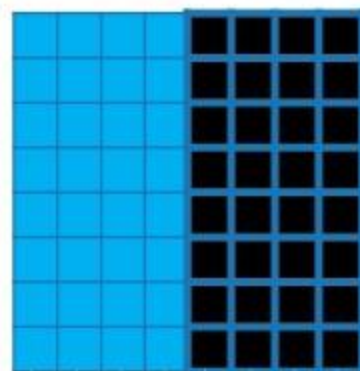
8bit (字节1B=8bit)



8*4位的存储器



16*4位的存储器



8*8位的存储器

每行有多少位?

一共多少行?

总容量多少位?



存储系统-主存编址

(一) 希赛



存储单元

存储单元个数=最大地址-最小地址+1



编址内容

按字编址：存储体的存储单元是字存储单元，即最小寻址单位是一个字。

按字节编址：存储体的存储单元是字节存储单元，即最小寻址单位是一个字节。



总容量=存储单元个数*编址内容



根据存储器所要求的容量和选定的存储芯片的容量，就可以计算出所需芯片的总数，即：

总片数=总容量/每片的容量



存储系统-主存编址

(希赛)

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

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

存储单元个数=最大地址-最小地址+1=最大地址+1-最小地址

$8FFFF\text{H} + 1 = 90000\text{H}$

$90000\text{H} - 84000\text{H}$

90000H
 $- 84000\text{H}$

$C000\text{H}$

总容量=存储单元个数*编址内容

$12 * 16^3 * 8\text{bit}$

单位芯片容量： $8K * 4\text{bit} = 8 * 2^{10} * 4\text{bit}$

总片数=总容量/单位芯片容量

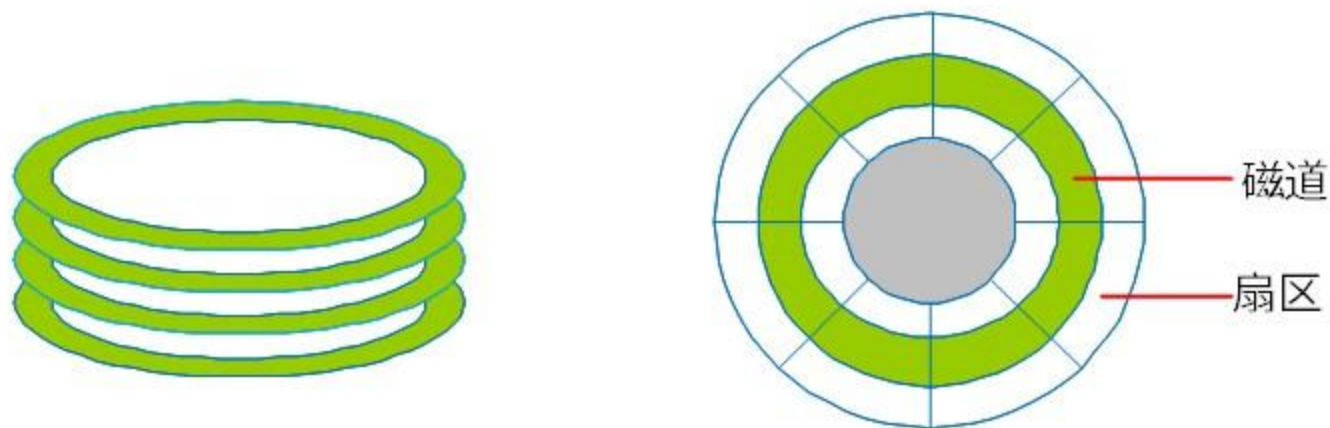
$12 * 16^3 * 8\text{bit}$

$8 * 2^{10} * 4\text{bit}$



存储管理 - 磁盘管理

(希赛)

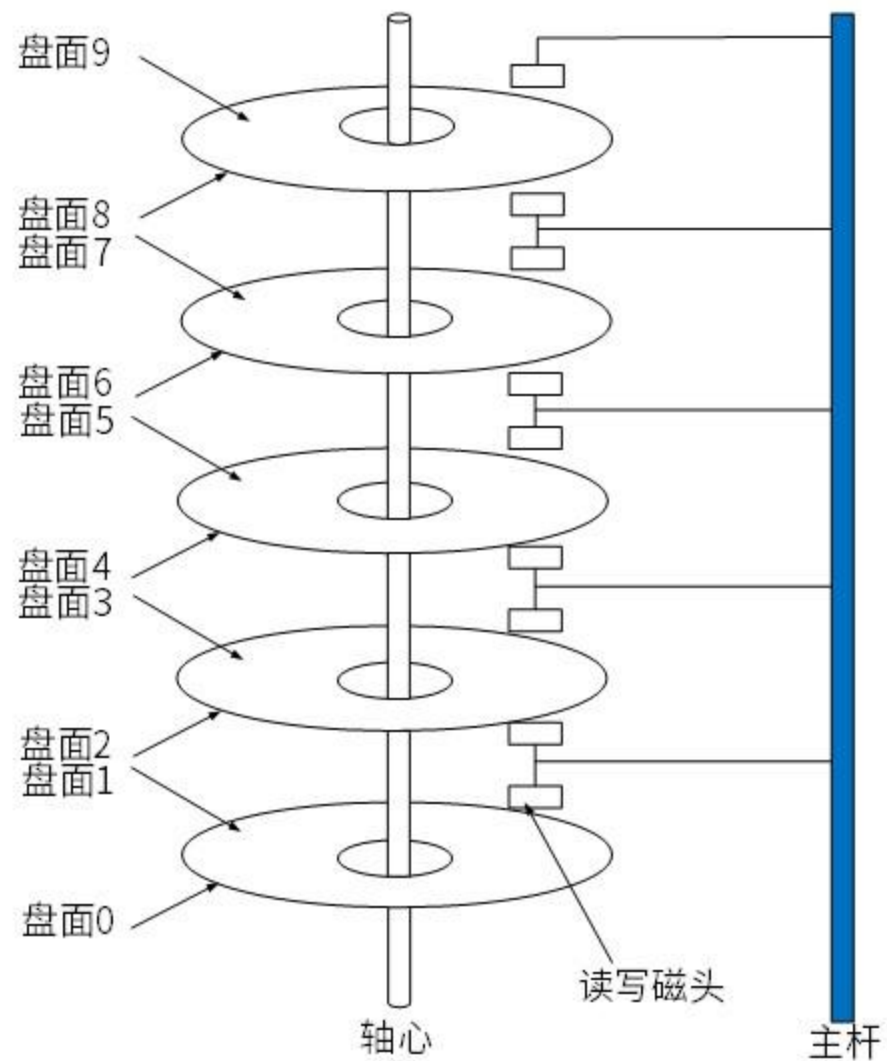


存取时间=寻道时间+等待时间，寻道时间是指磁头移动到磁道所需的时间；等待时间为等待读写的扇区转到磁头下方所用的时间。



存储管理 - 磁盘管理

(希赛)





存储管理 – 磁盘管理

(希赛)

读取磁盘数据的时间应包括以下三个部分：

- (1) 找磁道的时间。
- (2) 找块（扇区）的时间，即旋转延迟时间。
- (3) 传输时间。



存储管理 – 磁盘管理

(希赛)

◆ 学员疑问：

存取时间=寻道时间+等待时间（平均定位时间+转动延迟）这个公式我认为有问题。

我认为应是 存取时间=寻道时间+转动延迟+存（或取）磁盘扇区内容的时间；

麻烦到时说一下讲义里如果是对的该怎么理解，我的如果是错的错在哪里？

◆ 平均存取时间（Average Access Time），是指磁头找到指定数据的平均时间。

平均存取时间是指磁头找到指定数据的平均时间，通常它是硬盘平均寻道时间和平均潜伏时间（等待时间）之和。平均存取时间最能代表硬盘找到某一数据所用的时间，数值越小越好。

◆ 平均访问时间=平均寻道时间+平均等待时间



存储管理 – 磁盘管理

(希赛)

某磁盘磁头从一个磁道移至另一个磁道需要10ms。文件在磁盘上非连续存放，逻辑上相邻数据块的平均移动距离为10个磁道，每块的旋转延迟时间及传输时间分别为100ms和2ms，则读取一个100块的文件需要（ ）ms。

A 10200 B 11000 C 11200 D 20200

$((10 \times 10) + 100 + 2) \times 100$



存储管理 – 磁盘管理

(希赛)

假设某磁盘的每个磁道划分成11个物理块，每块存放1个逻辑记录。逻辑记录R₀, R₁, ..., R₉, R₁₀存放在同一个磁道上，记录的存放顺序如下表所示：

物理块	1	2	3	4	5	6	7	8	9	10	11
逻辑记录	R ₀	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉	R ₁₀

如果磁盘的旋转周期为33ms，磁头当前处在R₀的开始处。若系统使用单缓冲区顺序处理这些记录，每个记录处理时间为3ms，则处理这11个记录的最长时间为（ ）；若对信息存储进行优化分布后，处理11个记录的最少时间为（ ）。

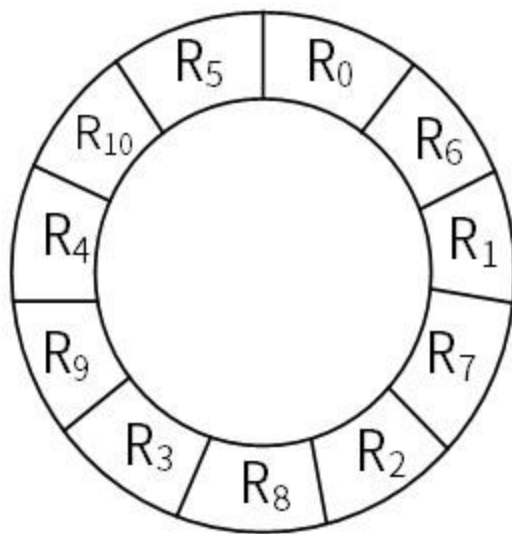
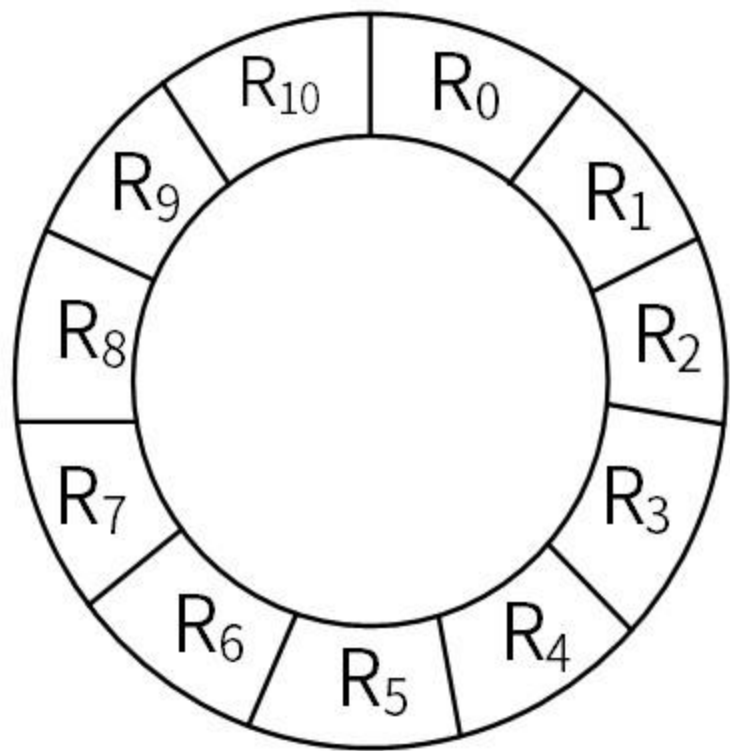
A 33ms B 336ms C 366ms D 376ms

A 33ms B 66ms C 86ms D 93ms



存储管理 - 磁盘管理

(希赛)





存储管理 – 磁盘管理

(希赛)

- ✓ 先来先服务 (FCFS)
- ✓ 最短寻道时间优先 (SSTF)
- ✓ 扫描算法 (SCAN)
- ✓ 循环扫描 (CSCAN) 算法



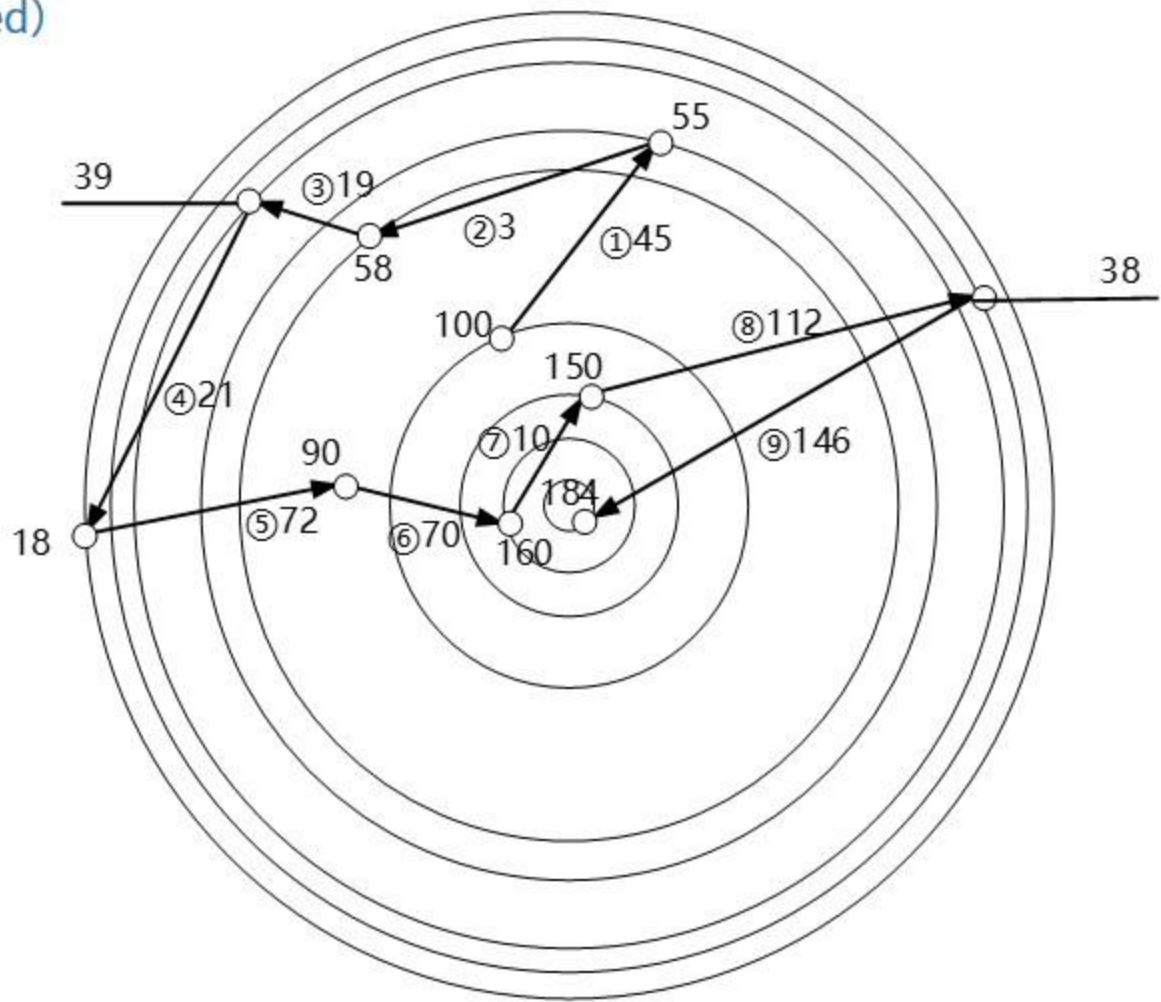
磁盘调度 - FCFS

(希赛)

先来先服务FCFS (First-Come, First Served)

(从100号磁道开始)

序号	被访问的下一个磁道号	移动距离 (磁道数)
①	55	45
②	58	3
③	39	19
④	18	21
⑤	90	72
⑥	160	70
⑦	150	10
⑧	38	112
⑨	184	146
平均寻道长度: 55.3		





存储管理 – 磁盘管理

(希赛)

假设磁盘臂位于15号柱面上，进程的请求序列如下表表示，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	12	8	9
②	19	6	5
③	23	9	6
④	19	10	5
⑤	12	8	4
⑥	28	3	10

A ①②③④⑤⑥

B ⑤①②④③⑥

C ②③④⑤①⑥

D ④②③⑤①⑥

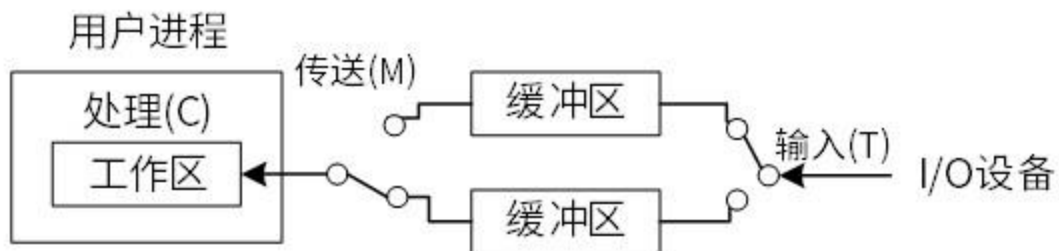


综合应用

(●) 希赛

假设磁盘块与缓冲区大小相同，每个盘块读入缓冲区的时间为 $15\mu\text{s}$ ，由缓冲区送至用户区的时间是 $5\mu\text{s}$ ，在用户区内系统对每块数据的处理时间为 $1\mu\text{s}$ ，若用户需要将大小为10个磁盘块的Doc1文件逐块从磁盘读入缓冲区，并送至用户区进行处理，那么采用单缓冲区需要花费的时间为（ ） μs ；采用双缓冲区需要花费的时间为（ ） μs 。

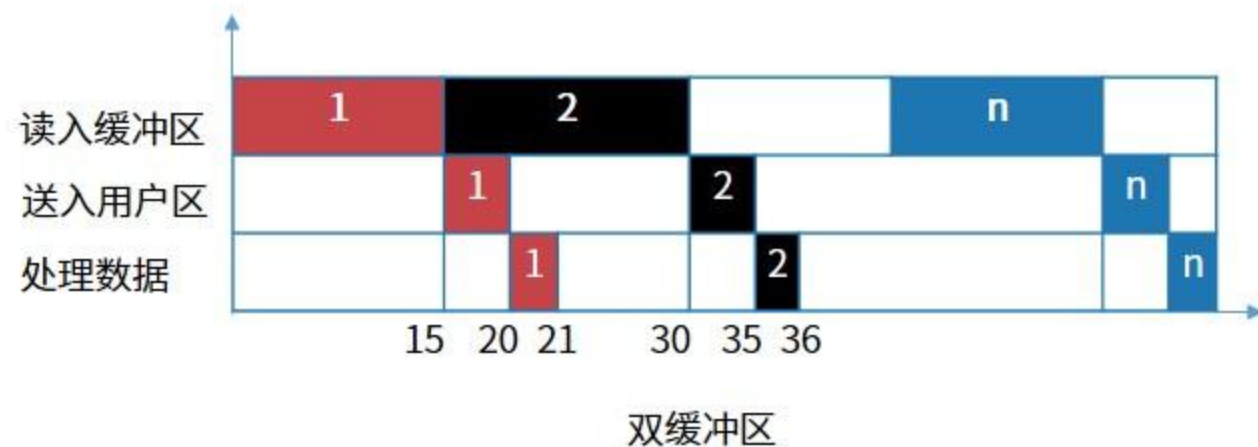
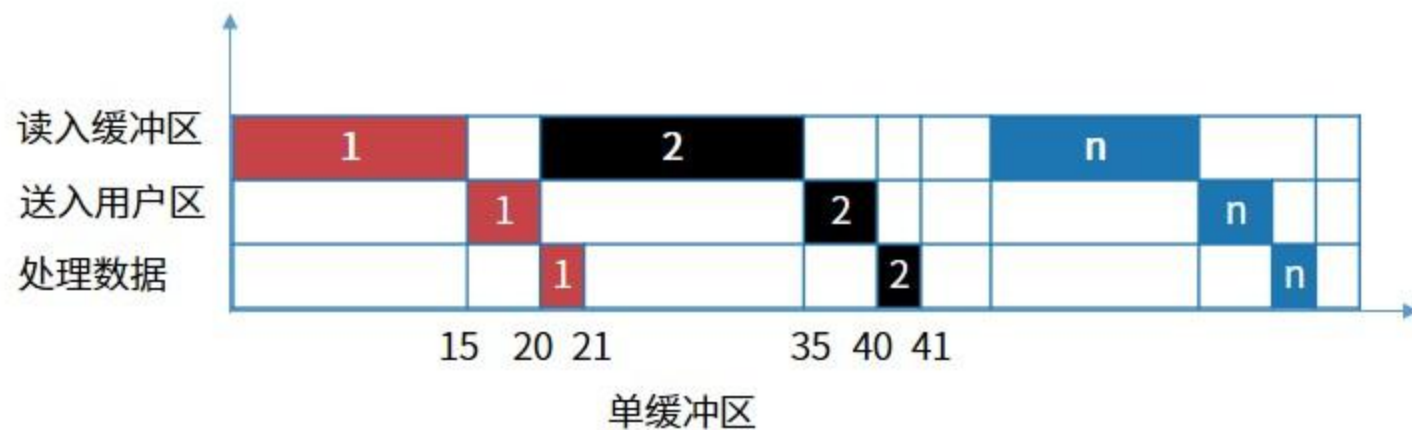
- A 150 B 151 C 156 D 201
A 150 用户进程 B 151 C 156 D 201





综合应用

(希赛)





综合应用

(希赛)

这道题中处理数据有三个处理步骤：

- 1、从磁盘读入到缓冲区 (15us) ；
- 2、从缓冲区读入到 (内存) 用户区 (5us) ；
- 3、处理 (内存) 用户区数据 (1us) 。

步骤1和2都需要访问临界资源——缓冲区，所以需要合并成一个操作阶段。
使用缓冲区时不能并行，必须分开执行，时间为 $15+5=20\text{us}$ ；其次处理数据 1us 。
构造成流水线后，整个过程划分为2个阶段，分别是 20us ， 1us ，根据流水线执行公式，流水线执行时间为 $20\text{us}+1\text{us}+(10-1)*20\text{us}=201\text{us}$ 。

双缓冲区：

可以实现读入到缓冲区2和从缓冲区1读入到用户区的并发。

读入缓冲区，和从缓冲区读入用户区，可以对不同的缓冲区进行，也就是说，可以并行处理。

对于这里构造成流水线后，整个过程划分为3个阶段，1、从磁盘读入到缓冲区 (15us) ； 2、从缓冲区读入到用户区 (5us) ； 3、处理 (内存) 用户区数据 (1us) 。根据流水线执行公式，流水线执行时间为 $15\text{us}+5\text{us}+1\text{us}+(10-1)*15\text{us}=156\text{us}$ 。






总线

(希赛)

- ✓ 总线是一组能为多个部件**分时共享**的公共信息传送线路。
 - **共享**
 - **分时**：是指同一时刻仅允许一个部件向总线发送信息，但允许多个部件同时从总线上接收相同的信息。
- ✓ 串行总线（适合长距离传输）与并行总线（适合近距离连接，不宜过长）
- ✓ 单工、半双工与全双工
- ✓ 总线宽度与总线带宽

总线的分类：

-  **数据总线 (Data Bus, DB)**：在CPU与RAM之间来回传送需要处理或是需要储存的数据。
-  **地址总线 (Address Bus, AB)**：用来指定在RAM (Random Access Memory) 之中储存的数据的地址。
-  **控制总线 (Control Bus, CB)**：将微处理器控制单元 (Control Unit) 的信号，传送到周边设备。



总线

(希赛)

挂接在总线上的多个部件，（ ）。

- A 只能分时向总线发送数据，并只能分时从总线接收数据
- B 只能分时向总线发送数据，但可同时从总线接收数据
- C 可同时向总线发送数据，并同时从总线接收数据
- D 可同时向总线发送数据，但只能分时从总线接收数据



总线

(希赛)

下列说法中正确的是（ ）。

- 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



总线

(希赛)


以下关于串行总线的说法中，正确的是（ ）。


- A 串行总线一般都是全双工总线，适宜于长距离传输数据
- B 串行总线传输的波特率是总线初始化时预先定义好的，使用中不可改变
- C 串行总线是按位（bit）传输数据的，其数据的正确性依赖于校验码纠正
- D 串行总线的数据发送和接收是以软件查询方式工作



校验码-奇偶校验

(CS) 希赛

 奇偶校验码的编码方法是：由若干位有效信息（如一个字节），再加上一个二进制位（校验位）组成校验码。

 奇校验：整个校验码（有效信息位和校验位）中“1”的个数为奇数。

偶校验：整个校验码（有效信息位和校验位）中“1”的个数为偶数。

奇偶校验，可检查1位的错误，不可纠错。





校验码-循环校验码CRC

(希赛)


CRC校验，可检错，不可纠错。

循环冗余校验 (CRC, Cyclic Redundancy Check)

 CRC的编码方法是：在k位信息码之后拼接r位校验码。应用CRC码的关键是如何从k位信息位简便地得到r位校验位（编码），以及如何从k+r位信息码判断是否出错。

 循环冗余校验码编码规律如下：

- ① 把待编码的N位有效信息表示为多项式 $M(X)$ ；
- ② 把 $M(X)$ 左移K位，得到 $M(X) \times X^K$ ，这样空出了K位，以便拼装K位余数（即校验位）；
- ③ 选取一个K+1位的产生多项式 $G(X)$ ，对 $M(X) \times X^K$ 做模2除；
- ④ 把左移K位以后的有效信息与余数 $R(X)$ 做模2加减，拼接为CRC码，此时的CRC码共有N+K位。

 把接收到的CRC码用约定的生成多项式 $G(X)$ 去除，如果正确，则余数为0；如果某一位出错，则余数不为0。不同的位数出错其余数不同，余数和出错位序号之间有唯一的对应关系。



校验码-循环校验码CRC

(希赛)

什么是模2除法，它和普通的除法有何区别？

模2除法是指在做除法运算的过程中不计其进位的除法。

例如，10111对110进行模2除法为：

$$\begin{array}{r} 110 \\ 110 \overline{) 10111} \\ \underline{110} \\ 0111 \\ \underline{110} \\ 0011 \end{array}$$

普通除法运算过程：

$$\begin{array}{r} 11 \\ 110 \overline{) 10111} \\ \underline{0110} \\ 1011 \\ \underline{0110} \\ 0101 \end{array}$$



校验码-循环校验码CRC

(希赛)

例：原始报文为“10111”，其生成多项式为：“ $G(x)=x^4+x+1$ ”。对其进行CRC编码后的结果为？

$$\begin{array}{c} x^4+x+1 \\ \downarrow \\ x^4+0 \times x^3+0 \times x^2+x+1 \\ 1 \quad 0 \quad 0 \quad 1 \quad 1 \end{array}$$

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



校验码

(CS) 希赛

	校验码位数	校验码位置	检错	纠错	校验方式
奇偶校验	1	一般拼接在头部	可检奇数位错	不可纠错	奇校验：最终1的个数是奇数个； 偶校验：最终1的个数是偶数个；
CRC循环冗余校验	生成多项式最高次幂决定	拼接在信息位尾部	可检错	不可纠错	模2除法求余数，拼接作为校验位
海明校验	$2^r \geq m+r+1$	插入在信息位中间(2^k 位置)	可检错	可纠错	分组奇偶校验



校验码-循环校验码CRC

希赛

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

$$\begin{array}{c} x^5+x^3+x+1 \\ \downarrow \\ x^5+0 \times x^4+x^3+0 \times x^2+x+1 \\ 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \end{array}$$

$$\begin{array}{r} 101011 \overline{) 11100011000000} \\ \underline{101011} \\ 100111 \\ \underline{101011} \\ 110010 \\ \underline{101011} \\ 110010 \\ \underline{101011} \\ 110010 \\ \underline{101011} \\ 110010 \\ \underline{101011} \\ 110010 \\ \underline{101011} \\ 11001 \end{array}$$