

## lecture12 计算机的指令系统

指令功能：数据运算、数据传输、输入输出指令、控制指令、其他

指令格式：指令字（操作码+操作数地址）

THCO MIPS 指令系统：

16 字长，按字编址

指令的寻址方式：顺序寻址

操作数寻址方式：立即寻址，寄存器寻址，变址寻址（8 位偏移）

I/O：与主存共享地址空间（统一地址空间），双串口

总线：双 16 位地址总线，双 16 位数据总线，独立的访问 FLASH 总线

共设计有 44 条指令：R 型（21 条）I 型（14 条）B 型（5 条）J 型（4 条）

思考：

指令系统的作用和地位？指令系统是硬、软件之间的接口部分，其优劣是一个计算机系统是否成功的关键因素。

为实现 THCO 指令系统,ALU 应该具备哪些功能?+->><<&||

## lecture13 数据表示及检错纠错

变长字符编码，提高存储空间利用率

原码、反码和补码（正数一致，零的原码和反码均有 2 个编码，补码只 1 个码）

负数表示形式：

原码（Sign Magnitude）：符号位&&数的绝对值

反码（One's Complement）：符号位&&数值按位求反

补码（Two's Complement）：反码的最低位+1（不同位数补码相加减时，要符号扩展）

补码加法： $[x+y]_{\text{补}}=[x]_{\text{补}}+[y]_{\text{补}}$

补码减法： $[x-y]_{\text{补}}=[x+(-y)]_{\text{补}}=[x]_{\text{补}}+[-y]_{\text{补}}$

$[-x]_{\text{补}}$ ： $[x]_{\text{补}}$ 连同符号位在内，逐位求反，再在最低位加 1。

$[x/2]_{\text{补}}$ ： $[x]_{\text{补}}$ 算术右移 1 位

大端：数据的低位保存在内存的高地址字节中。

小端：数据的低位保存在内存的低地址字节中。

奇偶校验码：用于并行数据传送中，码距为 2

汉明校验码：用于并行数据传送中  $2^r \geq m+r+1$ ，码距为 4

循环冗余校验码：用于串行数据传送中

•汉明校验码每个校验位强迫包括自己在内的一些位的奇偶值为偶数（或奇数）。

p1（整个码字的第 1 位）的校验规则：从当前位数 1 起，校验 1 位，跳过 1 位，再校验 1 位，再跳过 1 位，……。校验的所有位的 1 的数目应该为偶数（偶校验）。

p2（整个码字的第 2 位）的校验规则：从当前位数 2 起，校验 2 位，跳过 2 位，再校验 2 位，再跳过 2 位，……。校验的所有位的 1 的数目应该为偶数（偶校验）。

p3（整个码字的第 4 位）的校验规则：从当前位数 4 起，校验 4 位，跳过 4 位，再校验 4 位，再跳过 4 位，……。校验的所有位的 1 的数目应该为偶数（偶校验）。

## lecture14 算术运算及电路实现

运算器：

由控制器产生的控制信号驱动，取得操作数，完成**算术、逻辑运算**，得到运算结果的状态，输出、存放运算结果，暂存运算的中间结果。

1 位 ALU 设计

布斯算法

恢复余数法、加减交替除法

## lecture15 运算器部件组成

定点运算器和浮点运算器

位片结构运算器 Am2901：

4 位运算器

8 种运算功能

8 种数据来源组合

8 种数据输出方式

VHDL 语言简介

思考：

A、B 口锁存器的作用？如果不设置该锁存器，需要如何调整时序？

锁存器，在时钟脉冲的高电平期间，其输入信号直接“穿”过线路送到其输出端，用脉冲下降沿把输入信号存储起来用于后续的输出。

## lecture16 浮点数表示和运算

一个浮点数的组成分为三个部分：符号位+阶码域+尾数域

记符号位值为  $s$ ，阶码域的整数值为  $e$ ，尾数域的小数值为  $f$ 。

对于规格化数（阶码  $e$  不为 0），阶码值  $E=e-Bias$ ，尾数值  $M=1+f$ ；（隐藏位技术）

对于非规格化数（阶码  $e$  全 0），阶码值  $E=1-Bias$ ，尾数值  $M=f$ ；

$V=(-1)^s * M * 2^E$ （其中  $Bias$  为偏置，单精度为 127，双精度为 1023）

s00000000 跟全 0 尾数是浮点数 0，

s11111111 跟全 0 尾数表示无穷大，

s11111111 跟非全 0 尾数表示非法数值 NaN。

单精度： $2.0 \times 10^{-38}$  至  $2.0 \times 10^{38}$

双精度： $2.0 \times 10^{-308}$  至  $2.0 \times 10^{308}$

思考：

IEEE754 浮点数尾数用原码，阶码用移码的原因？

运算器主要功能是什么？都是怎样在电路上实现的？

运算器的功能如此简单，为什么程序能完成十分复杂的功能？

## lecture31 层次存储器系统动态存储器

**存储器作用：**存放程序和数据，存储程序使计算机走向通用

几种常用的存储方式：磁颗粒、半导体(电平/电容)、光

主存储器：寄存器 Register，高速缓存 Cache，主存储器 Main Memory

辅助存储器：磁盘 Disk，磁带 Tape，光盘 Compact Disk

**不同类型存储器比较（从左到右，速度、价格/位、可靠性、CPU 访问频度递减，容量递增）：**

**Register、Cache、Main Memory、Magnetic Disk、Tape、Optical Disk**

按访问方式分类：随机访问存储器 RAM（半导体），顺序访问存储器 SAM（磁带），直接访问存储器 DAM（磁盘），关联访问存储器 CAM（Cache+TLB）

**存储器系统设计目标：**大容量、高速度、低成本、高可靠性

存储器对性能的影响：

假定某台计算机的处理器主频= 1GHz (机器周期为 1ns), CPI = 1.1, 50%算逻指令, 30% 存取指令, 20% 转移指令。其中 10%的存取指令会发生数据缺失, 需要 50 个周期的延迟。  
 $CPI = \text{理想 CPI} + \text{每条指令的平均延迟} = 1.1 + (0.30 \times 0.10 \times 50) = 1.1 + 1.5 = 2.6 \text{ cycle}$

程序运行的局部性原理

时间局部性：最近被访问过的程序和数据很可能再次被访问

空间局部性：CPU 很可能访问最近被访问过的地址单元附近的地址单元。

层次之间应满足的原则：

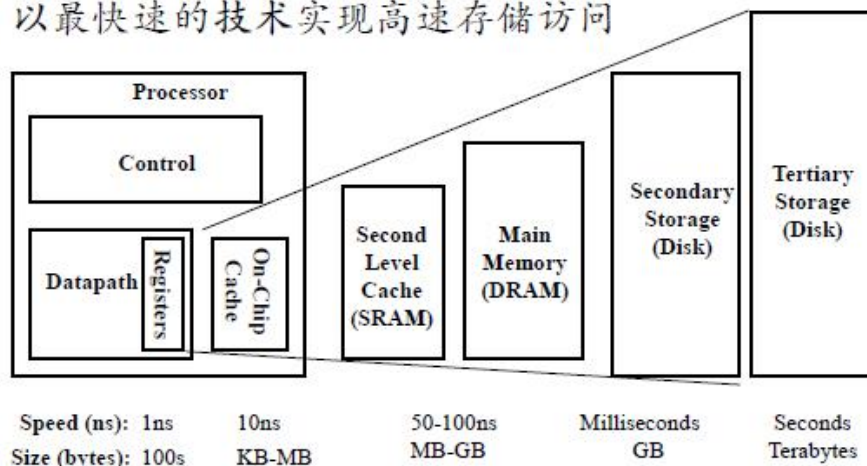
一致性原则：处在不同层次存储器中的同一个信息应保持相同的值。

包含性原则：处在内层的信息一定被包含在其外层的存储器中，反之则不成立。

**层次存储器系统**

❖ 以最低廉的价格提供尽可能大的存储空间

❖ 以最快速的技术实现高速存储访问



并行技术：主存的一体多字（一个读写体，每次多字）。单字多体（多个读写体，交叉编址）。

主存储器的作用和连接：

**地址总线**用于选择主存储器的一个存储单元（字或字节），其位数决定了能够访问的存储单元的最大数目，称为最大可寻址空间。

**数据总线**用于在计算机各功能部件之间传送数据，其位数（总线的宽度）与总线时钟频率的乘积，与该总线所支持的最高数据吞吐（输入/输出）能力成正比。

**控制总线**用于指明总线的工作周期类型和本次入/出完成的时刻。

**动态存储器 DRAM：**用金属氧化物半导体（MOS）的单个 MOS 管存储一个二进制位（bit）信息。信息被存储在 MOS 管 T 的源极的寄生电容  $C_s$  中，有电荷表示 1，无电荷表示 0。集成度高、容量大、能耗低、速度慢。工作特点：破坏性读出，需定期刷新，快速分页组织。

DRAM 写访问开始于 RAS\_L 信号有效。两种写方式：WE\_L 信号早和晚于 CAS\_L 信号有效。

DRAM 读访问开始于 RAS\_L 信号有效。两种读方式：OE-L 信号早于或晚于 CAS\_L 有效。

思考

程序的局部性原理指什么?为什么层次存储器系统能同时达到高性能/低成本/大容量的指标?

### lecture32 静态存储器及高速缓冲存储器

**静态存储器**：速度快，存储密度低，数据入/出共用管脚，能耗高，价格高。

	存储信息	读破坏性	刷新	送行列地址	访问速度	集成度	发热量	存储成本
SRAM	触发器	否	否	同时	快	低	大	高
DRAM	电容	是	需要	分两次	慢	高	小	低

层次存储器系统：根据程序的局部性原理，将最近被访问的信息项装入到 Cache 中，将最近被访问的信息项临近的信息一起装入到 Cache 中。

**高速缓冲存储器 Cache**：主存和 CPU 之间的存储器，用高速的静态存储器实现，缓存了 CPU 频繁访问的信息。高速，透明。

要解决的问题

1. 地址之间的映射关系：如何根据主存地址得到 Cache 中的数据?
2. 数据之间一致性：Cache 中的内容是否已经是主存对应地址的内容?
3. 数据交换的粒度：Cache 中的内容与主存内容以多大的粒度交换?
4. Cache 内容装入和替换策略：如何提高 Cache 的命中率?

#### Cache 参数

块 (Line)：数据交换的最小单位

命中 (Hit)：在较高层次中发现要访问的内容

命中率 (Hit Rate)：命中次数/访问次数

命中时间：访问在较高层次中数据的时间

失效 (Miss)：需要在较低层次中访问块

失效率 (Miss Rate)：1-命中率

失效损失 (Miss Penalty)：替换较高层次数据块的时间+将该块交付给处理器的时间

平均访问时间=HR\*命中时间+(1-HR)\*失效损失

全相联方式的地址映射关系 (cache：有效位-标记-数据。主存：块号-块内地址)

1. 主存的字块可以和 Cache 的任何字块对应，利用率高，方式灵活。
2. 标志位较长，比较电路的成本太高。如：n 位的主存地址，块内地址为 b 位，Cache 有 m 块，则需要 n-b 位标志位，m 个比较电路。

**直接映射方式** (cache：有效位-标记-数据。主存：标记-索引-块内地址)

1. 主存的字块只可以和固定 Cache 字块对应，方式直接，利用率低，命中率低，效率较低。

2. 标志位较短, 比较电路的成本低。如果主存空间有  $2^n$  块, Cache 有  $2^m$  块, 则只要  $n-m$  位标志位, 仅需要 1 次比较。

### lecture33 高速缓冲存储器 Cache

**多路组相联方式** (cache: 有效位-标记-数据。主存: 标记-索引-块内地址)

1. 前两种方式的综合方案。先通过直接映射方法确定组, 再在组内通过全相联方式确定行。
2. 集中了两个方式的优点。成本也不太高。

**一致性保证:**

写直达 (Write through) 在 Cache 中命中, 同时修改 Cache 和对应的主存内容; 没有在 Cache 中命中, 写分配 (Write allocate), 非写分配 (not Write allocate)。强一致性保证, 效率低。

拖后写 (Write back) 替换时再写主存 (主动替换, 被动替换), 通过监听总线上的访问操作来实现。弱一致性保证, 实现复杂, 效率较高。

**提高存储访问的性能:** 平均访问时间 = 命中时间  $\times$  命中率 + 失效损失  $\times$  缺失率

提高命中率、缩短缺失时的访问时间、提高 Cache 本身的速度

**Cache 缺失的四类原因:** 必然缺失 Compulsory Miss (开机或者是进程切换、首次访问数据块), 容量缺失 Capacity Miss (活动数据集超出了 Cache 的大小), 冲突缺失 Conflict Miss (多个内存块映射到同一 Cache 块、某一 Cache 组块已满但空闲的 Cache 块在其他组), 无效缺失 (其他进程修改了主存数据)。

经验总结: 容量为  $N$ 、采用直接映射方式 Cache 的缺失率和容量为  $N/2$ 、采用 2 路组相联映射方式 Cache 的缺失率相当

**影响命中率的因素**

Cache 容量: 适当提高

Cache 块大小: 适当块数, 最佳块数

地址映射方式: 多路组相联

直接映射——主存中的一块只能映射到 Cache 中唯一的一个位置

全相联映射——主存中的一块可以映射到 Cache 中任何一个位置

$N$  路组相联映射——主存中的一块可以选择映射到 Cache 中  $N$  个位置

替换算法: 最近最少使用的 (LRU)、先进先出 (FIFO)、随机替换 (RAND)

**多级 Cache:**

采用两级或更多级 cache 来提高命中率 (增加 Cache 层次, 增加了用户的选择);

将 Cache 分解为指令 Cache 和数据 Cache (指令流水的现实要求, 根据具体情况选用不同的组织方式和容量)。

**Cache 接入系统的体系结构**

1. 侧接法: 像入出设备似的连接到总线上, 优点是结构简单, 成本低, 缺点是不利于降低总线占用率。

2. 隔断法: 把原来的总线打断为两段, 使 CACHE 处在两段之间, 优点是有利于提高总线利用率, 支持总线并发操作, 缺点是结构复杂, 成本较高。

**一致性保证策略(MESI)**

思考

Cache 命中率和哪些因素有关? 如何提高 Cache 的命中率?

Cache 写有许多策略, 试进行比较。



## lecture34 虚拟存储器

虚拟地址：程序员编程使用的地址

虚拟地址空间：虚拟地址的集合

逻辑地址：程序员使用的地址

存储器地址：物理存储器的地址，也称物理地址（Physical Address）或实地址（Real Address）

每个进程有独立的逻辑地址空间，实现内存共享，实现内存保护，通过页表将虚地址转换为实地址。

### 虚拟存储器目的

提供大容量：获得运行比物理存储器更大空间程序的能力

简化存储管理：内存的分配以及虚实地址转换

保护：操作系统可以对虚拟存储空间进行特定的保护

灵活：程序的某部分可以装入主存的任意位置

提高存储效率：只在主存储器中保留最重要的部分

提高并行度：在进行段页替换的同时可以执行其它进程

可扩展：为对象提供了扩展空间的能力.页表中存放有访问权限。

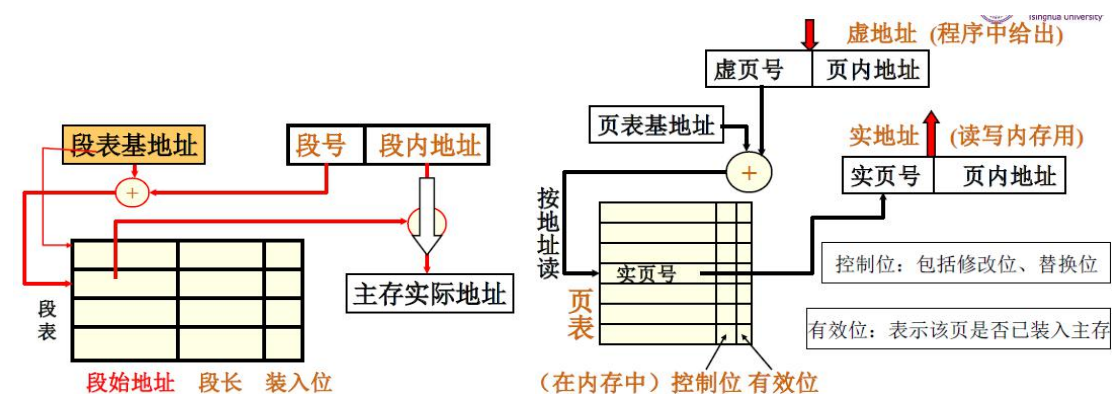
虚存	Cache
主要目的是实现存储管理，帮助解决存储容量的问题。	主要目的是解决存储速度问题，使存储器访问速度不太影响 CPU 的运行速度。
单位时间内数据交换次数少，每次交换数据量大	单位时间内数据交换的次数多，每次交换数据量小
由操作系统管理	由硬件实现

虚拟存储器管理：

段式存储管理：以段作为存储管理的对象，段长动态可变，段起点、终点不定。易于编译、管理、修改、保护和共享，空间分配困难，容易产生碎片。

段（Segment）程序模块化设计的结果（过程、函数、数组、…），逻辑上相对独立，是共享和保护的最小单元。

实现：设置段表进行管理，段表基地址，段起始地址，段长，装入位，保护、共享等标志。



页式存储管理

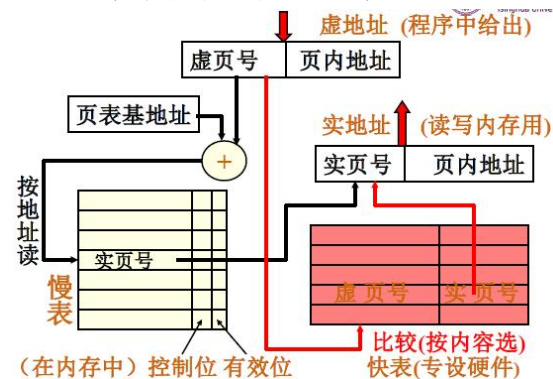
将主存和虚存划分为固定大小的页，以页为单位进行管理和数据交换。

虚地址=虚页号+页内地址；实地址=实页号+页内地址。

实现：通过页表进行管理，页表基地址寄存器，实页号，控制位。

页式虚拟存储器的访问过程

1. 得到程序给出的虚地址；
2. 由虚地址得到虚页号；
3. 访问页表，得到对应的实页号；
4. 若该页已在内存中，则根据实页号得到实地址，访问内存；
5. 否则，启动输入输出系统，读出对应页装入主存，再进行访问。



### 转换旁路缓冲 (TLB)

为页表设置的专用 Cache，高效实现虚页号到实页号的转换，容量较小，128~256 个表项。

TLB 缺失：流水线停，通知操作系统，读页表，将表项写入 TLB，返回用户程序，重新访问。

页面大小：缩小页面大小可减少内部碎片，但需更大的页表。

页面替换算法：最近最少使用 (LRU)

### 段页式虚拟存储管理

先把程序按逻辑单位分为段，再把每段分成固定大小的页。

操作系统对主存的调入调出按页面进行，但又可按段实现共享和保护，兼取页式和段式系统的优点。地址映射通过一个段表和一组页表来进行，需要多次查表。

### Pentium 的虚存管理

不分段也不分页模式，虚拟存储的地址空间和物理存储空间大小相同。

页式管理模式，将主存分成固定长度的页，通过页进行存储保护和管理。

段式管理模式，按程序本身的逻辑段来划分主存空间，段的长度可变，段表存放在内部寄存器，存储访问时间是可以预知的。

段页式管理模式，按程序的逻辑结构将主存分段，对段进行访问控制，在段内分页，按页进行数据交换。

段号:

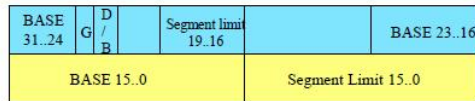


0: GDT 1: LDT      线性地址格式:

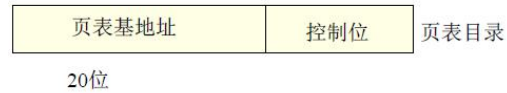
段表基地址



段表:



线性地址=BASE + OFFSET



思考

是否还有继续使用虚拟存储器的必要?

## lecture35 磁表面存储设备

磁芯存储器: 存储原理简单, 工艺复杂, 可靠性低, 大容量, 成本低廉, 断电后保存数据。

**磁表面存储设备:** 容量大, 成本低, 断电保存, 脱机保存, 串行访问, 数据交换频率低, 数据交换量大。一般为磁盘、光盘等。如何保存? 如何表示? 如何组织? 如何管理?

用磁颗粒的不同磁化偏转方向来区分不同的状态, 主存中存放 CPU 要立即访问的程序和数据, 辅助存储器中存放 CPU 不立即使用的信息, 在需要时再调入主存中。

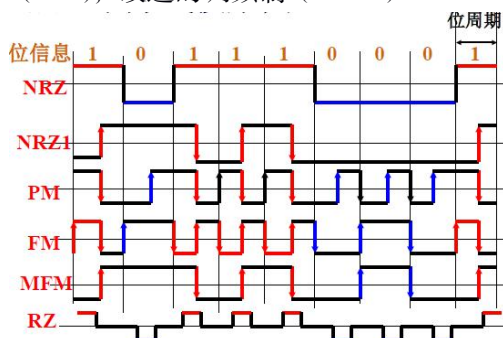
随机访问: 随机访问任何单元, 访问时间与存放位置无关, 每一位都有各自的读写设备。

串行访问: 顺序按位进行, 访问时间与存储位的物理位置有关, 共用一个读写设备。

主要指标: 存储密度, 存储容量, 寻址时间, 数据传输率, 误码率, 价格。

**磁记录方式:** 一种编码方法, 即如何将一串二进制信息, 通过读写电路变换成磁层介质中的磁化翻转序列。(编码效率, 自同步能力, 读写可靠性)

归零制 (RZ), 不归零制 (NRZ), 见 1 翻转的不归零制 (NRZ1), 调相制 (PM), 调频制 (FM), 改进的调频制 (MFM)



## 磁盘

长期存储, 断电后存储, 容量大, 价格低廉, 但速度慢, 可用在层次存储器的最底层。

使用旋转托盘上的表面磁颗粒来存储数据, 可移动的读/写头来访问磁盘。

硬盘比软盘: 硬质托盘, 面积大; 密度高; 转速快, 传输率高; 可以多个盘片组合。

扇区是磁盘访问的最小单位, 每面多少磁道, 每个磁道多少扇区。

访问过程: 寻道 (将磁头移动到正确磁道), 寻找扇区 (等待磁盘上扇区旋转到磁头下), 传



输数据（读写数据）。平均旋转延迟应为磁盘旋转半周的时间。

**磁盘访问时间 = 寻道时间 + 平均旋转延迟 + 传输时间 + 磁盘控制器延迟**

例：平均寻道时间 = 12ms，旋转速度 = 5400rpm，磁盘控制器延迟 = 2ms，传输速度 = 5MB，扇区大小 = 512 bytes，页大小 8KB。

旋转 1 周：1/5400 minutes = 11.1ms => 1/2 周：5.6 ms

读 1 个扇区：12ms + 5.6ms + .5K/5MB s + 2ms = 12 + 5.6 + .1ms + 2ms = 19.7 ms

读 1 页的时间：12 ms + 5.6ms + 8K/5MB s + 2ms = 12ms + 5.6ms + 1.6ms + 2ms = 21.2 ms

思考

页容量大扇区却如此小：可靠性（损坏时丢弃扇区，检错快）；灵活性（支持不同操作系统）。

采用流水方式和大容量传输方式（每次读取多个扇区）克服磁盘控制器延迟。

额外开销在总开销中比例较大 => 一次传输大量数据比较有效。

将页面存放在相邻扇区中可以避免额外的寻道开销。

## RAID 技术

廉价磁盘的冗余阵列（Redundant Arrays of Inexpensive Disks）

目标：N 个磁盘的容量，1/N 的访问时间，更高的性价比，用冗余技术提高存储信息可用性。

提高磁盘的可用性和性能

RAID0~ RAID7

思考

磁表面存储设备的原理和特点，它在层次存储器系统中的地位和作用。

如何将文件分布在扇区使访问速度加快？

## lecture36 光盘的存储原理

CD-ROM

CD-R

CD-RW

DVD

存储技术的发展

优点：容量适宜，价格低，可靠性，便于携带，ROM，用于计算机间交换数据和分发程序。

### 光盘的存储原理

光电转换：入射光和反射光叠加

物理变化：在涂有玻璃表层的主盘上，用高能红外激光束烧出 0.8 毫米直径的小孔制成。

往模子上注入熔化的多种碳酸盐脂。在碳酸盐脂上沉淀上一薄层的反射铝，再覆盖上一层起保护作用的表层。

碳酸盐脂底基的凹陷部分叫作凹区，凹区两边未经过烧制的部分叫作凸区。

读出原理：用低能量激光二极管照射，从凹区反射的激光的强度为从凸区反射光的强度的一半，感知反射光的强弱，可区分凸区和凹区，编码方式为见 1 就翻转的不归 0 制。

### 只读光盘（CD-ROM）

数字化存储，只读，便于携带。

将每个字节编码成 14 位的符号，可以对字节进行汉明编码。连续的 42 个符号一组，构成了 588 位的帧，包含 192 位数据位（24 个字节），其它的 396 位用于纠错和控制位。将 98 帧作为一个 CD-ROM 扇区，每个扇区由 16 个字节的引导区开始，其中前 12 个字节为引导区起始标志，然后 3 个字节是扇区号，最后的一个字节是光盘的数据存放格式。

### **CD-R 存储原理、CD-RW 存储原理、DVD**

#### **存储技术的新发展**

不同层次的存储器作用：存储器组成（数字逻辑层，存放一位二进制数 0 或 1）；存储器（指令系统层，存放可单独访问的最小单元（字节、字、扇区））；存储器（操作系统层，段、页管理）；存储器（用户层，文件）。

存储器设计目标：高速度，大容量，高可靠性，低价格。

思考

计算机存储器系统是如何为数据库系统服务的？

## **lecture41 输入/输出系统概述和输入/输出方式**

### **输入输出系统的作用、功能**

与计算机外部交换信息的通道

### **输入/输出系统组成**

设备：键盘、鼠标；显示器、网卡、打印机、喇叭  
多（种类繁多）；繁（功能繁杂）；异（速度不一）。

### **要解决的问题**

控制方式：CPU 如何控制输入/输出？（输入/输出方式）

传输方式：传输通道、方式、速率等（总线、接口）

数据识别和转换：数/模转换、语音识别等，转为字符、数据等计算机能识别的格式（设备）。

### **输入/输出方式**

**程序直接控制：**CPU 直接使用输入/输出指令来控制外部设备。

CPU 方查询接口状态（循环等待），直到外设已经接收到该字符，读字符；外设方往接口数据缓冲中送字符，处理完后，置状态寄存器，等待下一个字符。

**程序中断：**外部设备请求，CPU 暂停正在执行的程序并响应，CPU 与外设并行工作，完成后继续执行原来的程序。有中断优先级。

中断源：外中断（I/O 设备等），异常（内中断：处理器硬件故障、程序“出错”，Trap），中断触发器，中断状态寄存器。

完整过程：中断请求（中断源设备设置中断触发器），中断响应（响应条件：允许中断、当前指令结束、优先级，响应实现：硬件实现的中断隐指令，保存断点），中断处理（保存现场信息，运行中断服务程序，中断返回）。

关中断；保存断点；判中断源，转中断服务；开中断；执行中断服务程序；关中断；恢复断点；开中断；返回断点。

**直接存储访问（DMA）**I/O 设备和主存储器之间的直接数据通路，为专设的硬件，用于高速 I/O 设备和主存储器之间成组传送数据。

数据传送开始前和结束后通过程序或中断方式对 DMA 进行预处理和后处理。

DMA 工作方式：独占总线方式，周期窃取方式。

Cache 一致性：主存中的数据可能不是最新的，采用直接写会带来性能的降低，DMA 查询 Cache 会降低性能，直接设计硬件控制。

**通道** I/O 通道是系统中代替 CPU 管理控制外设的独立部件，能执行有限 I/O 指令集合。

根据 CPU 要求选择某一指定外设与系统相连，向该外设发出操作命令，进行初始化。

指出外设读/写信息的位置以及与外设交换信息的主存缓冲区地址。

控制外设与主存之间的数据交换。

指定数据传送结束时的操作内容，检查外设的状态

通道类型：字节多路通道（简单的共享通道，分时处理，面向低、中速字符设备）；选择通道（选择一台外设独占整个通道，以成组传送方式传送数据块，效率高，适合快速设备）；

数组多路通道（上两种方式的结合，效率高，控制复杂）。

### 外围处理机

通道型处理机：共享内存

程序直接控制	成本低，效率低，严重占用 CPU 资源，适用早期计算机中高速设备。
程序中断方式	提高 CPU 的效率，对 CPU 干扰较大，可管理多个外部设备，适用传输速度不高或传输量不大。
直接存储访问（DMA）	数据传输过程由 DMA 自行控制，主存储器需支持成组传送，与设备一对一服务，对 CPU 干扰适中（初始化、周期挪用），无法适用大量高速设备管理。
通道控制方式	一对多的连接关系，适应不同速度、不同种类的外部设备，可并行工作
外围处理机	通用计算机，独立完成输入/输出功能，通过通道方式与主机进行交互。

思考

输入/输出方式解决了什么问题？

它们各自有哪些特点？

## lecture42 总线

### 计算机总线

共享的信息通道，用于连接计算机多个子系统（部件），线也是连接复杂巨系统的一种基本工具。处理器和其他组成部件的接口。

优点：解决外部设备“繁”的问题（容易增添新的设备，使用相同总线标准的外部设备可在不同计算机间兼容），降低成本（总线可供多个设备共享），简化设计。

不足：容易成为信息通道的瓶颈（总线故障，总线带宽限制了整条总线的吞吐量）。

总线的最高速度主要由下列因素决定：总线长度，总线负载的设备数，负载设备的特性。

1、单总线计算机（主板总线）：使用一条总线完成处理器和主存储器之间通信，主存储器和输入/输出设备之间通信。

优点：简单、成本低

缺点：速度慢，总线将成为系统瓶颈

应用：PDP-1、IBM PC - ISA EISA

2、双总线系统：输入/输出总线通过适配器和处理器-主存总线相连，处理器-主存总线用于处理器和主存储器之间的通信，输入/输出总线为输入/输出设备提供信息。

应用举例：Apple Macintosh II、NuBus、SCSI 总线。

3、三总线系统：主板总线连接到处理器-主存总线，处理器-主存总线用于处理器和主存之间数据交换，I/O 总线连接到主板总线。

优点：大大减少处理器-主存总线负载

例：现代 PC 采用的结构

### 总线分类

处理器-主存总线（专用）	传输距离短、速度高，主存储器专用，保证主存储器-处理器之间的高带宽，直接和处理器连接，优化处理使之适应 Cache 块传送。
输入/输出总线（行业标准）	通常距离较长，速度较慢，需要适应多种输入/输出设备，和处理器-主存总线通过桥连接（或通过主板总线）
主板总线（行业标准或专门设计）	主板：连接各部件器件的底盘，应允许处理器、主存储器和输入/输出设备互连，应有价格优势：所有组件连接在一条总线上。

### 总线组成

**控制线**：标记总线事务的开始和结束，指明数据线上传输信息的类型。

**数据线**：在源设备和目标设备间传送信息。

相关概念：

总线主设备：有能力控制总线，发起总线事务

总线从设备：响应主设备请求

总线通信协议：定义总线传输中的事件顺序和时序要求

异步总线传输：控制信号（请求，应答）作为总控信号

同步总线传输：使用共同的时钟信号

总线事务：发起命令（和地址），传输数据。

### 总线仲裁

问题：如何为需要使用总线的设备安排总线？

可通过主-从设备的安排来避免冲突：只允许主设备发起总线事务，控制所有总线请求，从设备响应主设备的读写请求。

最简单的设计：处理器作为唯一的总线主设备，所有总线请求均由处理器控制，主要缺点是处理器被卷入到每一个总线事务中。

总线仲裁的基本要求：某总线主设备使用总线前应发出总线请求，只有得到授权后，主设备才能使用总线，使用完毕后，主设备应通知仲裁器。

**优先权**：优先级高的设备应得到优先服务。**公平性**：低优先级的设备应有得到服务的机会。

总线仲裁方式：

#### 1、集中仲裁

1) 菊链仲裁（所有设备共用一个总线请求信号）

简单。但无法保证公平性，低优先级设备可能得不到总线使用权；总线授权信号的逐级传递限制了申请总线的速度。

2) 集中平行仲裁（通过集中的仲裁器进行）

用于几乎所有处理器-主存总线和一些高速输入/输出总线。

**2、分布仲裁** 1) 通过自我选择进行分布式仲裁（每个要使用总线的设备将自己的标识放在总线上）。2) 碰撞检测（以太网）。

### 通信方式

同步总线：控制线中包含有一根时钟信号线，传输协议根据时钟信号制定。

例如：主设备提出总线请求后 5 个时钟周期，可以获得能否使用总线的信号。

优点：逻辑简单、高速。缺点：总线上设备须按时钟频率工作，为防止时钟信号扭曲，高速工作时总线距离须够短。

异步总线：不使用统一的时钟，可适应设备的不同速度，不用担心时钟信号扭曲，距离可较长，使用握手协议。

增加总线带宽——

增加总线的宽度：可增加每个周期传送数据的量，提高了成本。

分别设置数据总线和地址总线：可同时传送数据和地址，提高了成本。

采用成组传送方式：一个总线事务传送多个数据，每次只需要在开始的时候传送一个地址，直到数据传送完毕才释放总线，复杂度提高，延长后续总线请求的等待时间。

多主设备总线提高事务数量——

仲裁重叠：在当前事务时，为下一总线事务进行仲裁。

总线占用：没有其他主设备请求总线的情况下，某主设备一直占用总线，完成多个总线事务。地址、数据传送重叠。

**总线举例 PCI 总线**

参数选择	提高性能	降低成本
总线宽度	将地址和数据线分开	互用地址和数据线
数据宽度	越宽越快（32 位）	越窄越廉价（8 位）传送
大小	多字可减少总线开销	单字传送简单
主设备	多主设备（仲裁）	单主设备
时钟	同步	异步
协议	并行	串行

思考

计算机总线作用？总线仲裁应考虑哪些方面？

总线数据传输模式有哪些？各有什么特点？

### **lecture43 接口电路和外部设备**

#### **接口的基本功能**

提供主机识别（指定、找到）使用的 I/O 设备的支持：为每个设备规定几个地址码或编号。

建立主机和设备之间的控制与通信机制：接收处理器（主设备）的命令，并提交给外部设备，同时，为主设备提供外部设备的状态。

提供主机和设备之间信息交换过程中的数据缓冲机构。

提供主机和设备之间信息交换过程中的其他特别需求支持：屏蔽外部设备的差异。

#### **接口内部组成**

设备识别电路

数据缓冲寄存器（输入/输出）

控制寄存器

状态寄存器

中断电路

其他电路

#### **串行通信**

同步传送：采用同步信号（内同步：同步字符。外同步：硬件同步信号）

异步传送（起始位、停止位；波特率）

全双工（通信双方有各自的接收和发送部件，两条数据线）

#### **串行传送中的有关概念**

串行、异步、同步、单工、半双工、全双工。

停止位、数据位、起始位。

奇偶校验、传送的波特率、波特率因子、数据采样。

#### **USB 接口**



用户不必再设置卡上、设备上的开关或跳线  
不必打开机箱来安装新的输入输出设备  
应该只需要一根电缆线就可以将所有设备连接起来  
输入/输出设备应可以从电缆上得到电源  
单台计算机最多可以连接 127 个设备  
系统应能支持实时设备（声卡、电话）  
可在计算机运行时安装设备  
不必重新启动计算机  
成本低

### **接口**

连接外部设备：设备识别，数据缓冲，协议实现，屏蔽差异。  
通过总线与主机进行通信。

### **外部设备功能**

完成数据的输入和/或输出：信号转换，数据采样  
与接口进行连接：接口信号，电平标准等  
与主机进行通信：通过总线进行，速度，控制方式

指令流水执行性能比较

k 个阶段，每段的时间都为 T，则 n 条指令所用的时间为

顺序执行： $t = nkT$

一次重叠执行： $t = nkT - (n-1)T$

两次重叠执行： $t = nkT - (n-2)T$

m 次重叠执行： $t = nkT - (n-m)T$