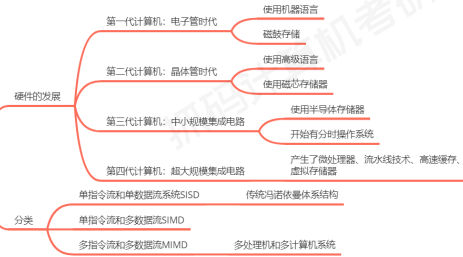
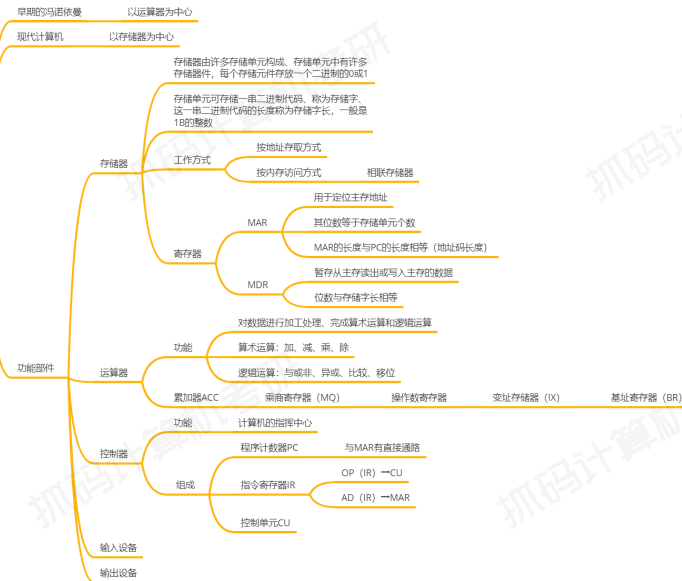


计组概述

计算机的发展



计算机的组成



计算机系统层次结构



计算机的性能指标

机器字长

计算机进行一次整数运算所能处理的二进制数据的位数

机器字长

计算机进行一次整数运算所能处理的二进制数据的位数，一般等于内部寄存器的大小，它决定了计算机的运算精度

辨析

指令字长

一个指令所包含的二进制位数

存储字长

一个存储单元的二进制代码长度

数据通路带宽

数据总线一次能并行传送的信息的位数

主存容量

主存储器所能存储信息的最大容量

MAR位数 n ，MDR长度 m （即存储字长），主存容量 = $2^n \times m$

MAR位数反应存储单元的个数，即可寻址的范围

运算速度

吞吐量和响应时间

吞吐量

系统在单位时间内处理的请求数

主要取决于主存的存取周期

响应时间

包括CPU时间（执行时间）+ 等待时间（I/O时间等）

主频和CPU时钟周期

主频

主频越高、完成指令的一个执行步骤所用的时间就越短，指令执行速度越快

CPU时间周期

CPU中最小的时间单位、每个动作至少需要一个时钟周期

互为倒数关系，主频通常以MHz为单位，1Hz表示1次/秒

CPI

执行一条指令所需的时钟周期数

CPU执行时间

CPU执行时间 = 时钟周期数 / 主频 = (指令条数 * CPI) / 主频

MIPS、MFLOPS、GFLOPS、TFLOPS

MIPS

每秒执行多少百万条指令

$MIPS = \text{主频} / CPI = \text{指令条数} / \text{执行时间} \times 10^6$

MFLOPS

百万次浮点运算

GFLOPS

十亿次浮点运算

TFLOPS

万亿次浮点运算



数据的表示和运算

编码

二进制、八进制、十进制、十六进制

注意：不是每个十进制都能转换成二进制

采用四位二进制表示一位十进制数

BCD码

六种冗余状态，超过9，需要加6（二进制0110）修正

ASCII码

7位二进制编码，存储时按一个字节存储、最高位为0

字符编码

汉字编码

国际码 = (16进制区位码) + 2020H

汉字内码 = (16进制区位码) + 8080H

★ 校验码

奇偶校验

奇校验

整个校验码（有效位+信息位）“1”为奇数个

偶校验

整个校验码（有效位+信息位）“1”为偶数个

只能发现奇数个错误，不能纠正错误

n位信息位，K位校验位，满足 $n+k \leq 2^k - 1$

纠错能力恒小于校验能力，即校验错的位数大于纠正的错位数

海明码

校验过程

①算出k②写出n+k为H编号③分校验组④校验位 = 校验组异或⑤写出完整的海明码H

校验原理

$S_i = P_i$ 异或校验组，全为0不出错，不全为0，转换为十进制i，找H_i取反即可

CRC循环冗余校验码

写出多项式，最高次数 = 信息位补0个数

补0后的信息位与多项式进行模2除法

将余数替换掉信息位后面的0

校验原理

接收方收到数据与多项式进行模2除法、余数为0，没有出错，余数不为0，将所得余数对应的信息位取反



数

数制与编码

数值：BCD码，用四位二进制来表示十进制，有冗余状态

字符

ASCII码

采用7位二进制数表示大小写字母、数字、和专用符号

汉字编码

包括输入编码、汉字内码、汉字字形码

校验

奇偶校验码

在原编码上加一个校验位，检验整个校验码中的个数是奇数还是偶数

海明校验码

按照某种规律分成若干组，每组安排一个检验位进行奇偶性测试，产生多位检验信息

循环冗余校验码

在K位信息码后面加上R位检验码，利用模2除法检验编码正确性

★ 定点数

表示

原码：用机器数的最高位表示数的符号，其余各位表示数的绝对值

补码：正数与原码一样，负数除符号位外，各位是原码的按位取反

计算

反码：整数与原码一样，负数除符号位外，各位是原码的按位取反

移位、原码以及补码下的加减乘除

★ 浮点数

表示

组成：阶符、阶码、数符、尾数

规格化：规定尾数的最高位必须是一个有效值

IEEE754标准：尾数采取隐藏位策略的原码表示，且阶码用移码表示的浮点数

加减步骤

对阶、小阶向大阶看齐

尾数求和

规格化

算数逻辑单元

一位全加器

最基本的加法单元，输入两位加数及低位的进位、输出和与最高进位

串行加法器

一位全加器的简单相连、串行进位

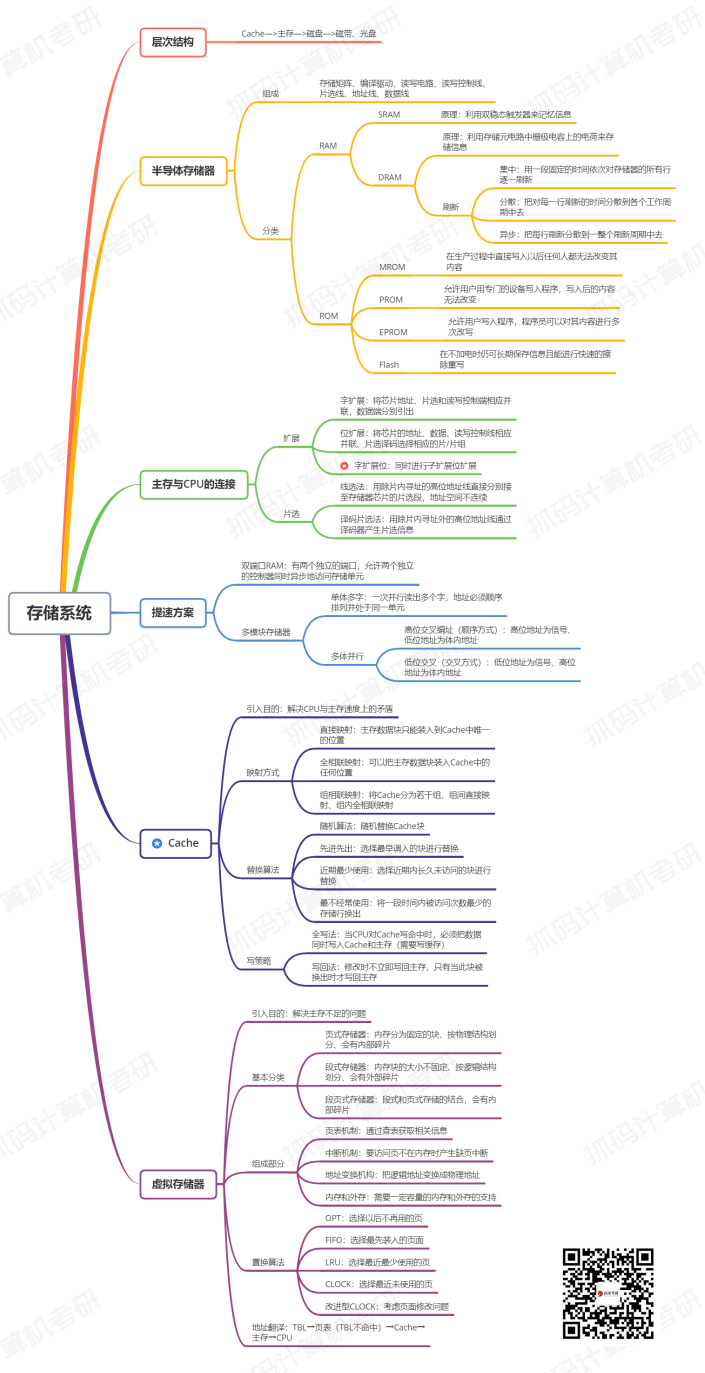
并行加法器

使用先行进位提高加法器的运算速度

ALU

功能较强的组合逻辑电路，它能进行多种算术运算和逻辑运算





虚拟存储器

基本概念

- 虚地址
 - 也称为逻辑地址，虚地址对应的空间为虚拟空间或程序空间
 - 虚地址 = 虚页存储 + 页内字地址
- 实地址
 - 也称物理地址，实地址对应的空间是主存的地址空间
 - 实地址 = 主存页号 + 页内字地址
- 辅存地址
 - 辅存地址 = 磁盘号 + 盘面号 + 磁道号 + 扇区号

页式存储器

- 概念：以页面长度固定、页表简单、调入方便
- 空间划分
 - 虚拟空间
 - 虚地址 = 虚页号 + 页内地址
 - 主存地址
 - 实地址 = 实页号 + 页内地址
- 虚实地址的变换
 - 页表
 - 虚实地址的变换由页表来实现，页表在主存中
 - 记录：虚页号与实页号的对照信息、装入信息
 - 变换后的实地址：实页号 + 页内地址
- 优缺点
 - 优点
 - 页面长度固定、页表简单、调入方便
 - 缺点
 - 页不是逻辑上独立的实体，处理保护共享不方便

段式存储器

- 概念
 - 段式虚拟存储器中的段是按程序的逻辑结构划分的，长度各异
- 虚拟地址划分
 - 虚地址 = 段号 + 段内地址
- 虚实地址的变换
 - 段表
 - 段表是程序的逻辑段和在主存存放位置的对照表
 - 记录：某个段的段号，装入位、段起点、段长
 - 变换后的实地址：段起始地址 + 段内地址
- 优缺点
 - 优点
 - 段分界与程序分界对应，具有逻辑独立性
 - 易于编译、管理、修改、保护、共享
 - 缺点
 - 容易在段时间留下碎片、造成浪费

段页式存储器

- 概念
 - 把程序按照逻辑结构划分段，每段再划分固定大小的页
- 虚拟地址划分
 - 虚地址 = 段号 + 段内页号 + 页内地址
- 虚实地址变换
 - 段表：根据段号一段表地址，从段表中找到页表起始地址 + 段内页号一页表地址
 - 页表：在页表中取出实页号 + 页内地址—实地址

快表TLB

- 快表存在的原因：因为CPU找通过虚地址都要访问一次，因为表就在主存中，找到表再通过地址变换得到实地址，再根据实地址访问才能得到想要的结果
- 快表TLB：将经常访问的页放在高速缓存存储器中，找到表中再通过地址变换得到实地址，再根据实地址访问才能得到想要的结果
- 慢表Page：在主存中的表
- 注：查找时，快表和慢表是一起查的，若快表找到了、慢表查找停止、快表没找到就继续在慢表查，没查到则执行页面替换策略

虚拟存储器与Cache的比较

- 相同点
 - 都有映射地址、替换算法、更新策略
 - 都根据程序访问的局部特性原理
 - 都为了提高系统性能
- 不同点
 - Cache解决系统速度、虚拟存储器解决主存容量问题
 - Cache由硬件实现，对所有程序员透明，虚拟存储器是OS和硬件共同实现，系统程序员不透明、应用程序员透明
 - 虚拟存储器不命中对系统性能影响更大
 - Cache能与CPU直接通信、辅存不能



指令系统

指令格式

- 基本格式
 - 操作码 指出指令中该指令应该执行什么性质的操作和具体有何种功能
 - 地址码 给出被操作的信息的地址或操作数本身，可以有多个地址码
- 定长操作码 在指令的最高位部分分配固定的若干位表示操作码
- ★ 扩展操作码 全部指令的操作码字段的位数不固定

寻址方式

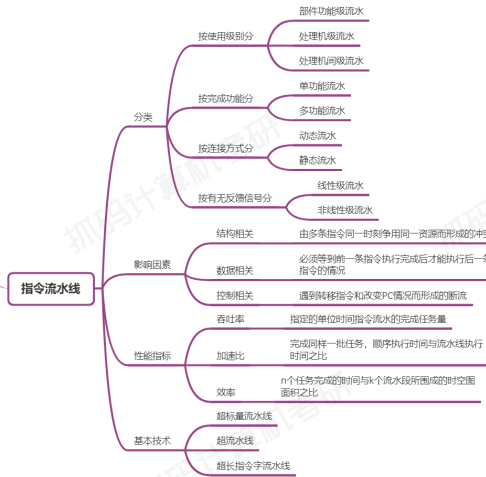
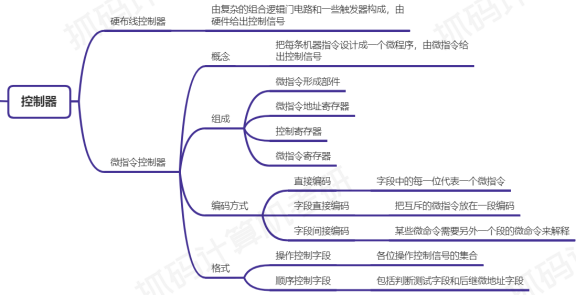
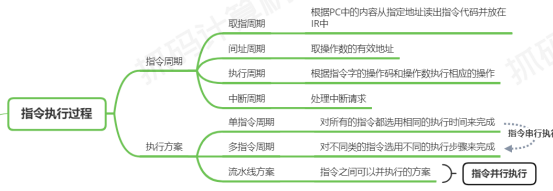
- 指令寻址方式
 - 顺序寻址 通过程序计数器PC加1，自动形成下一条指令的地址
 - 跳跃寻址 通过转移指令直接或间接给出下一条指令的地址
- 数据寻址方式
 - 隐含寻址 操作数不明显给出，而是隐含在指令中
 - 立即寻址 给出的不是操作数地址，而是操作数本身
 - 直接寻址 直接给出操作数的有效地址
 - 间接寻址 直接给出操作数有效地址的地址
 - 寄存器寻址 直接给出操作数所在的寄存器编号
 - 寄存器间接寻址 给出的是寄存器编号，但该寄存器编号中的地址才是有效地址
 - ★ 偏移寻址
 - 基址寻址 将基址寄存器中的内容加上指令格式中的形式地址才是有效地址
 - 变址寻址 将变址寄存器中的内容加上指令格式中的形式地址才是有效地址
 - 相对寻址 把程序计数器PC的内容加上指令格式中的形式地址才是有效地址
 - 堆栈寻址 从规定的栈中取出操作数

设计模式

- CISC 指令数目多、字长不固定、寻址方式多、寄存器数量少、一般为微程序控制
- RISC 指令数目少，字长固定，寻址方式少，寄存器数量多，一般为组合逻辑控制



中央处理器



总线

概述

- 定义：一组能为多个部件分时共享的公共信息传送线路
- 分类
 - 片内总线：用来连接芯片内部的各个部件
 - 系统总线：用于连接计算机系统内部各功能部件
 - 通信总线：用于连接计算机系统之间或计算机系统与其他系统
- 性能指标：传输周期、时间周期、总线宽度、总线带宽

仲裁

- 集中式
 - 链式查询
 - 计数器定时查询
 - 独立请求方式
- 分布式：不要中央仲裁器，每个潜在的主模块都有自己的仲裁器

总线周期

- 申请分配阶段：主设备向总线提出使用请求
- 寻址阶段：通过总线发出本次要访问的从模块的地址及有关的命令
- 传输阶段：主模块和从模块进行数据交换，可单向或双向进行数据传送
- 结束阶段：主模块的有关信息从系统总线上擦除，让出总线使用权

定时

- 同步：系统采用一个统一的时钟信号来协调发送和接收双方的传递的定时关系
- 异步
 - 概念：没有统一的时钟，依靠相互制约的“握手”来实现定时通信
 - 分类：不互锁，半互锁，全互锁



输入/输出系统

外部设备

- 输入设备 鼠标键盘等
- 输出设备 打印机, 显示器等
- 外部存储器 硬盘存储器、磁盘阵列、光盘存储器等

I/O接口

- 主要功能
 - 实现主机和外设的通信控制
 - 进行地址译码和设备选择
 - 实现数据缓冲
 - 信号格式的转换
 - 传送控制命令和状态信息
- 组成 数据缓冲寄存器, 设备选择电路, 命令寄存器, 命令译码器, 内部接口, 外部接口, 设备状态标志, 控制逻辑电路
- 编址
 - 统一编址 把I/O端口当作存储器的单元进行地址分配
 - 独立编址 I/O端口独立编址, 设置专门的输入输出指令访问接口

I/O方式

- 程序查询 由程序不断的查询外设的状态, 直到外设设备就绪
 - 中断请求: 中断源向CPU发出中断信号
 - 中断判优: 判断多个中断源的优先级
 - 响应条件
 - 有中断源提出中断请求
 - CPU允许中断并且开中断
 - 一条指令执行完毕且没有更紧迫的任务
 - 中断隐指令: 完成关中断, 保存断点 (PC值) 引出中断服务程序
 - 中断向量 查询中断服务程序的入口地址
 - 中断处理 执行中断服务程序、最好恢复现场、中断返回
- 程序中断
 - 概念 当CPU处理中断时, 又有更高优先级的中断请求
 - 条件
 - 中断服务程序中提前设置开中断指令
 - 优先级别高的中断源有权中断优先级别低的中断源
- DMA方式
 - 组成
 - 主存地址计数器 存放要交换的主存地址
 - 传送长度计数器 用来记录传送数据的主存地址
 - 数据缓冲寄存器 用于暂存每次传送的数据
 - DMA请求触发器 I/O设备准备好数据后使DMA请求触发器置位
 - 控制/状态逻辑 由控制和时序电路及状态标志完成
 - 中断机构 数据块传送完毕后触发中断机构, 提出CPU对主存的访问
 - 传送方式
 - 停止CPU访存 当需要传送数据时, 停止CPU对主存的访问
 - 交替访存 将CPU周期分为DMA访存和CPU访存两个部分
 - 周期挪用 I/O设备需要访存时, 挪用一個或几个存取周期
 - 传送过程
 - 预处理 完成寄存器置初值之类的准备工作
 - 数据传送 占有总线传输数据, 数据传输完全由DMA控制
 - 后处理 CPU执行中断服务程序做结束DMA处理

