

相关链接

[数据结构总结与知识网图](#) [计算机网络知识总结及知识网图](#) [操作系统总结及知识网图](#) [计算机组成原理总结及知识网图](#)

第一章 计算机系统概述

知识网图



冯诺依曼机的特点：

1) 计算机硬件系统由运算器、存储器、控制器、输入设备、输出设备5大部件组成。2) 指令和数据以同等地位存储在存储器中，并可按地址寻址。3) 指令和数据均用二进制代码表示。4) 指令由操作码和地址码组成。操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置。5) 指令在存储器内按顺序存放。通常，指令是顺序执行的，在特定条件下可根据运算结果或根据设定的条件改变执行顺序。6) 早期的冯诺依曼机以运算器为中心，输入/输出设备通过运算器和存储器传送数据。

不同级别的语言：

机器语言：(二进制语言) 计算机唯一可以直接识别和执行的语言。**汇编语言：**用英文单词或其缩写代替二进制的指令代码，易于人们记忆和理解。**高级语言：**为方便程序设计人员写出解决问题的处理方案和解题过程的程序。

计算机的工作过程：

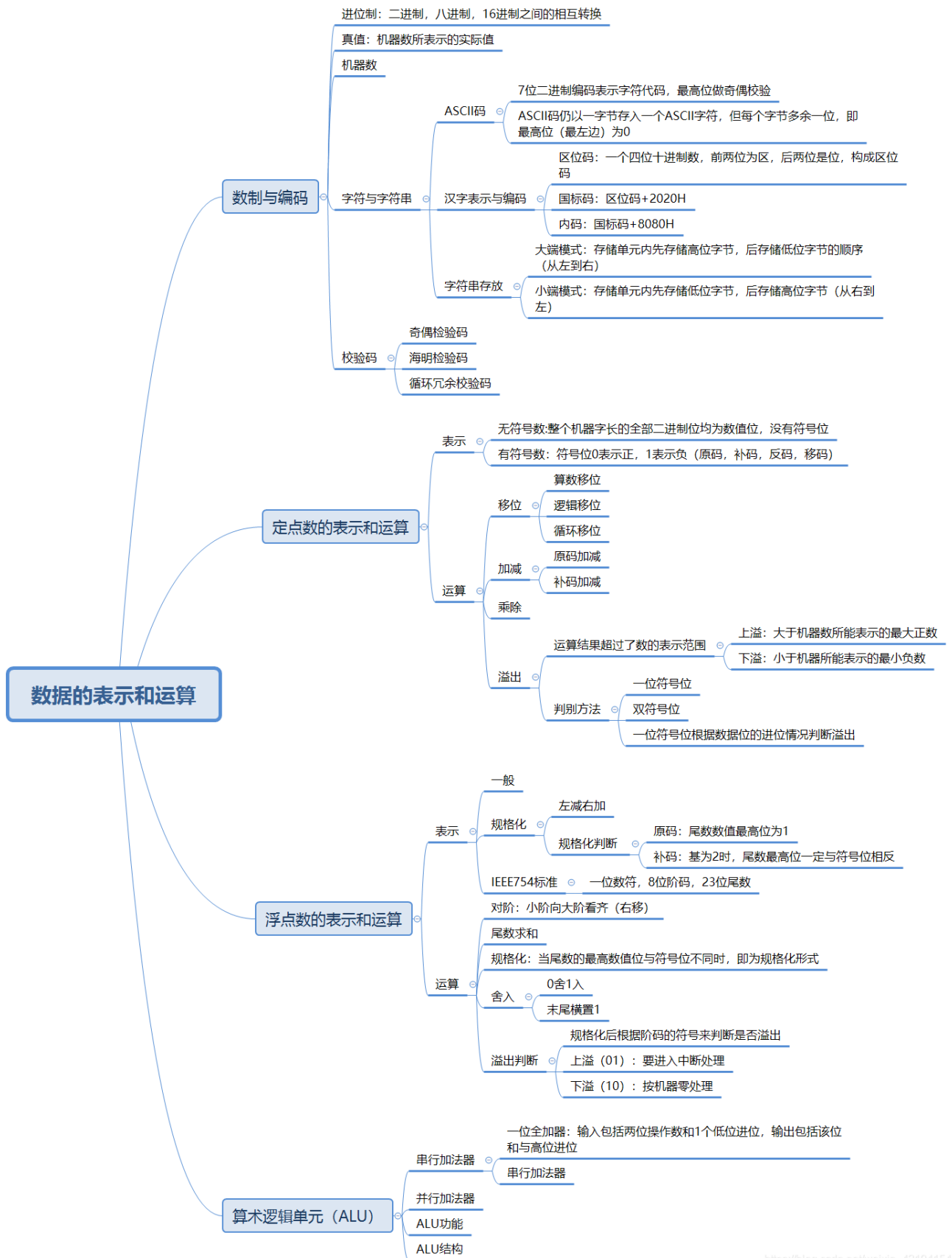
1)把程序和数据装入主存储器。2)将源程序转换成可执行文件。3)从可执行文件的首地址开始逐条执行指令。

计算机的性能指标：

机器字长：指计算机进行一次整数运算所能处理的二进制数据的位数。(通常与CPU的寄存器、加法器有关。**数据通路宽带：**指数据总线一次所能并行传送信息的位数(指外部数据总线的宽度，与CPU内部的数据总线宽度有可能不同)**主存容量：**指主存储器所能存储信息的最大容量，通常以字节来衡量，也可用字数*字长来表述存储容量。**运算速度 1) 吞吐量：**指系统在单位时间内处理请求的数量**2) 响应时间：**指从用户向计算机发送一个请求到系统对该请求做出响应并获得所需结果的等待时间。通常包括CPU时间(运行一个程序所花费的时间)与等待时间(用于磁盘访问、存储器访问、I/O操作、操作系统开销等时间)**主频和CPU时钟周期：****1) CPU时钟周期：****2) 主频：**CPI：执行一条程序所用的时钟周期数**CPU执行时间：**指运行一个程序所花费的时间。 $\text{CPU执行时间} = \text{CPU时钟周期数} / \text{主频} = (\text{指令条数} * \text{CPI}) / \text{主频}$ **MIPS、MFLOPS、GFLOPS、TFLOPS**

第二章 数据的表示和运算

知识网图



https://blog.csdn.net/weixin_42104154

在计算机系统内部，所有信息都用二进制进行编码的原因有以下几点：

- 1) 二进制只有0和1两种状态，使用有两个稳定状态的物理器件就可以表示二进制的每一位。制造成本低。
- 2) 二进制位1和0正好与逻辑值“真”和“假”对应，为计算机实现逻辑运算和程序中的逻辑判断提供了便利条件。
- 3) 二进制的编码和运算规则简单，通过逻辑门电路便能方便实现算术运算。

有符号的机器表示

(n+1)位	整数表示范围	小数表示范围
原码和反码	$-(2^n - 1) \leq x \leq 2^n - 1$	$-(1 - 2^{-n}) \leq x \leq 1 - 2^{-n}$
补码	$-2^n \leq x \leq 2^n - 1$	$-1 \leq x \leq 1 - 2^{-n}$
移码	$-2^n \leq x \leq 2^n - 1$	移码全为0真值最小，移码全为1真值最大

第三章 存储系统

知识网图

存储系统

存储器分类

- 层次划分
 - 主存储器（内存）
 - 辅助存储器（硬盘等）
 - 高速缓存（Cache）
- 存储介质
 - 磁表面存储器（磁盘，磁带）
 - 磁心存储器半导体（MOS型存储器，双极存储器）
 - 光存储器（光盘）
- 存取方式
 - 随机存储器（RAM）：存储器的任何一个存储单元的内容都可以随机存取
 - 只读存储器（ROM）：存储器的内容只能随即读出而不能写入
 - 串行访问存储器：按某种顺序存取，存取时间的长短与信息在存储体上的物理位置有关
- 信息的可保存性
 - 断电易失：如RAM
 - 断电不丢失：如ROM，磁盘，光盘
 - 破坏性读：如DRAM
 - 非破坏性读：如SRAM，磁盘等

性能指标

- 实际容量=存储单元个数*单元长度
- 理论容量=存储字数*存储字长
- 存取周期=存取时间+存储器状态恢复时间=连续读或写的最小时间间隔（存取时间>=存取时间）
- 主存带宽=传输速率=每秒能传输多少bit

层次化结构

CPU--寄存器--Cache--主存--辅存

随机存储器

- SRAM
 - 用双稳态触发器存储信息，多用于高速缓存
 - 非破坏读出，断电易失
 - 引脚中除了地址线 and 数据线外，还有CS(片选),WE(可写),OE(可读)
- DRAM
 - 用电容电荷存储信息，用于主存
 - 刷新方式
 - 集中刷新：在一个刷新周期内，利用一端固定的时间，依次对存储器的所有行进行逐一再生
 - 分散刷新：将工作周期分为读写周期和刷新周期，增加了存取周期，无死区
 - 异步刷新：刷新周期（2ms）除以行数，得到时间间隔t,每时间t产生一次刷新，缩短死区时间
 - 刷新对CPU是透明的，SRAM刷新单位是行，刷新类似读操作，无需片选
 - 引脚中除了地址线 and 数据线外，还有RAS(行选，起CS的作用)，CAS(列选)，WE,OE
- 只读存储器(ROM)
 - 结构简单，具有非易失性，可靠性高
 - 类型
 - 掩模式只读存储器（MROM）
 - 一次可编程只读存储器（PROM）
 - 闪速存储器（Flash Memory）
 - 固态硬盘（SSD）

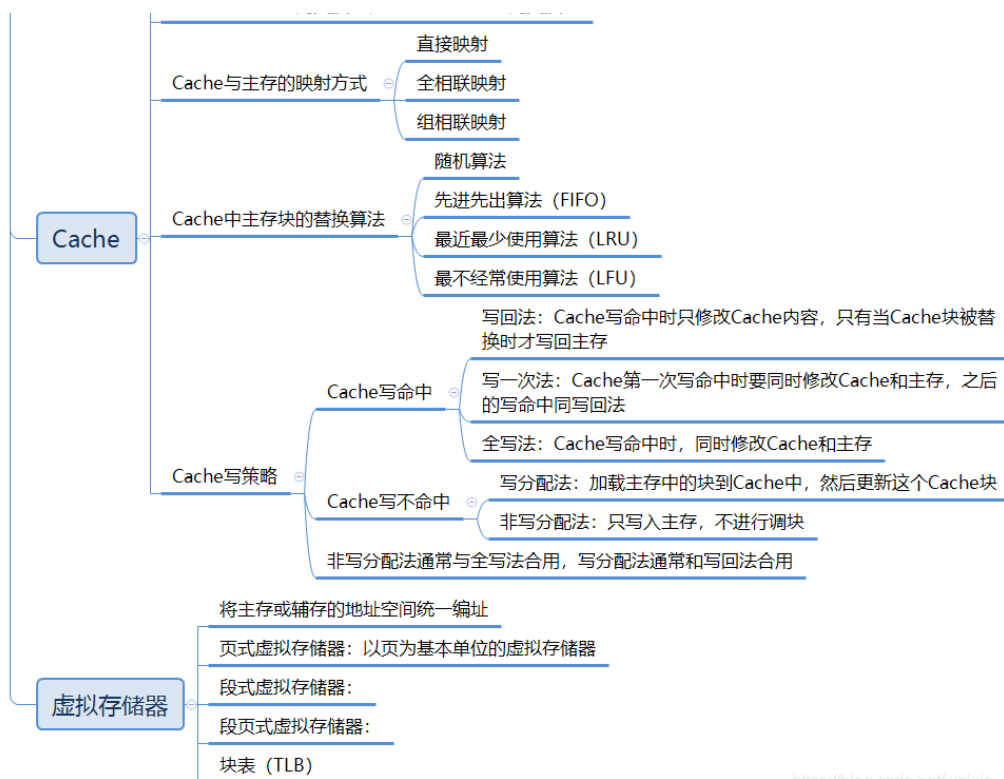
主存与CPU的连接

- 主存与CPU通过数据总线、地址总线、控制总线相连接
- 主存容量扩展
 - 存储容量=存储单元个数*存储单元字长
 - 字扩展
 - 位扩展
 - 字位同时扩展

双端口RAM与多模块存储器

- 双端口RAM：两个输入输出端，可并行读写数据，是空间并行技术
- 多模块存储器
 - 顺序方式：高位交叉编址，一个存储周期只能读出一个字的数据
 - 交叉方式：低位交叉编址，一个存取周期能读出字数*字长的数据

工作原理：程序访问的局部性原理
Cache与主存之间交换块，Cache与CPU之间交换字



https://blog.csdn.net/weixin_42104154

DRAM刷新方式

1) 集中刷新 在一个刷新周期内, 利用一段固定的时间, 依次对存储器的所有行进行逐一再生, 期间停止对存储器的读写操作, 即死时间, 访存“死区”。优点: 读写操作时不受刷新工作的影响, 系统的存取速度较高。缺点: 在集中刷新期间 (死区) 不能访问存储器。 **2) 分散刷新** 把对每行的刷新分散到各个工作周期中, 一个存储器的系统工作周期分为两部分: 前半部分用于正常读写或保持; 后半部分用于刷新某一行。优点: 没有死区。缺点: 加长了系统的存取周期, 降低了整机的速度。 **3) 异步刷新** 将刷新周期除以行数, 得到两次刷新操作之间的时间间隔 t , 每间隔时间 t 产生一次刷新请求。前两种方法的结合, 即可缩短“死时间”, 又能充分利用最大刷新间隔 $2ms$ 的特点。

程序访问的局部性原理

程序访问的局部性原理包括时间局部性和空间局部性。 **时间局部性** 指在最近的未来要用的信息很可能是现在正在使用的信息, (因为程序中存在循环) **空间局部性** 是指在最近的未来要用到的信息, 很可能与现在正在使用的信息在存储空间上是邻近的, 因为指令通常是顺序存放, 顺序执行的。 **高速缓冲技术** 利用程序访问的局部性原理, 把程序中正在使用的部分存放在一个高速的容量较小的Cache中, 使CPU的访存操作大多数针对Cache进行, 从而大大提高程序的执行速度。

虚拟存储器程序的执行过程

CPU使用续存时, 由辅存硬件找出虚地址与实地址之间的对应关系, 并判断这个虚地址对应的存储单元是否已装入主存。若已在主存中, 则通过地址转换, CPU可直接访问主存指示的实际单元; 若不在主存中, 则把包含这个字的一页或一段调入主存后再由CPU访问。若主存已满, 则采用替换算法置换主存的一页或一段。

快表

通常采用全相联或组相联方式, 每个TLB项由页表表项内容加上一个TLB标记字段组成, TLB标记用来表示该表项曲子页表中的哪个虚页号对应的页表项, 所以TLB标记的内容在**全相联方式**下表示该页表项对应的虚页号; **组相联方式**下表示对应虚页号的高位部分, 虚页号的低位部分用于选择TLB组的组索引。

虚拟存储器与Cache的比较

共同点： 1) 均是为了提高系统性能，二者都有容量、速度、价格的梯度； 2) 均把数据划分为小信息块，并作为基本得传递单元，虚存系统的信息块更大； 3) 都有地址的映射、替换算法、更新策略等问题； 4) 依据程序的局部性原理，将活跃的数据放在相对高速的部件中。 **不同点：** 1) Cache主要解决系统速度，虚存为了解决主存容量； 2) Cache由硬件实现，是硬件存储器，对所有程序员透明；而虚拟存储器由OS和硬件共同实现，是逻辑上的存储器，对系统程序员不透明，对应用程序员透明； 3) 虚拟存储器系统不命中时对系统性能影响更大； 4) CPU与Cache和主存都建立了直接访问的通路，辅存与CPU无直接通路。

第四章 指令系统

知识网图



https://blog.csdn.net/weixin_42104154

常见指令寻址方式的特点及适用情况：

1) **立即寻址**: 操作数获取便捷, 通常用于给寄存器赋初值。2) **直接寻址**: 相对于立即寻址, 缩短了指令长度。3) **间接寻址**: 扩大了寻址范围, 便于编制程序, 易于完成子程序返回。4) **寄存器寻址**: 指令字较短, 指令执行速度较快。5) **寄存器间接寻址**: 扩大了寻址范围。6) **基址寻址**: 扩大了操作数寻址范围, 适用于多道程序设计, 常用于为程序或数据分配存储空间。7) **变址寻址**: 主要用于处理数组问题, 适合编制循环程序。8) **相对寻址**: 用于控制程序的执行顺序、转移等。

CISC与RISC特点

CISC的主要特点: 1) 指令系统复杂庞大, 指令数量一般大于200+; 2) 指令的长度不固定, 指令格式多, 寻址方式多; 3) 可以放村的指令不受限制; 4) 各种指令的使用频度相差很大; 5) 各种指令执行时间相差很大, 大多数指令需要多个时钟周期才能完成; 6) 控制器大多数采用微程序控制, 有些指令非常复杂, 难以采用硬连线控制; 7) 难以用优化编译生成高效的目标代码程序。 **RISC的主要特点**: 1) 选取使用频率最高的一些简单指令, 复杂指令的功能由简单指令的组合来实现; 2) 指令长度固定, 指令格式种类少, 寻址方式种类少; 3) 只有Load/Store指令访存, 其余指令的操作都在寄存器之间进行; 4) CPU中通用寄存器的数量相当多; 5) RISC采用指令流水线技术, 大部分指令在一个时钟周期内完成; 6) 以硬布线控制为主, 不用或少用微程序控制; 7) 重视编译优化工作, 减少程序执行时间。

与CISC相比, RISC的优点:

1) 更能充分利用VLSI芯片的面积; 2) 更能提高运算速度; 3) 便于设计, 可降低成本, 提高可靠性; 4) 有利于编译程序代码优化。

第五章 中央处理器

知识网图

中央处理器

CPU的功能

- 指令控制
- 操作控制
- 时间控制
- 数据加工
- 中断处理

CPU的结构

运算器

- 算术逻辑单元：主要功能是进行算术/逻辑运算
- 暂存寄存器：暂存从主存读来的数据（对应用程序员是透明的）
- 累加寄存器：是一个通用寄存器，用于暂存ALU运算的结果信息
- 通用寄存器：存放操作数及各种地址信息
- 程序状态字寄存器：保留由算术逻辑运算指令或测试指令的结果而建立的各种状态信息
- 移位器：对操作数或运算结果进行移位运算
- 计数器：控制乘除运算的操作步数

控制器

- 程序计数器：指出下一条指令在主存中的存放地址
- 指令寄存器：保存当前正在执行的指令（无需用户的任何干预，对用户是透明的。）
- 指令译码器：仅对操作码字段进行译码，向控制器提供特定的操作信号
- 存储器地址寄存器（MAR）：存放要访问的主存单元地址
- 存储器数据寄存器（MDR）：存放向主存写入的信息或从主存读出的信息
- 微操作信号发生器：根据IR、PSW等内容及时序信号，产生控制整个计算机系统所需的各种控制信号，其结构有组合逻辑型和存储逻辑型两种。

指令执行过程

指令周期--机器周期--时钟周期--工作脉冲

指令周期 ⊖ 取指--间址--执行--中断

- 单指令周期：所有指令都选用相同的执行时间来完成
- 多指令周期：对不同类型的指令选用不同的执行步骤来完成
- 流水线方案：指令之间可以并行执行的方案

数据通路

数据通路：数据在功能部件之间传送的路径

功能

- 寄存器之间的数据传送
- 主存与CPU之间的数据传送
- 执行算术或逻辑运算

结构

- CPU内部单总线方式：所有寄存器的输入端和输出端都连接到一条公共通路上
- CPU内部三总线方式：将所有寄存器的输入端和输出端都连接到多条公共通路上
- 专用数据通路方式：根据指令执行过程中的数据和地址的流动方向安排连接线路，避免使用共享的总线

控制器

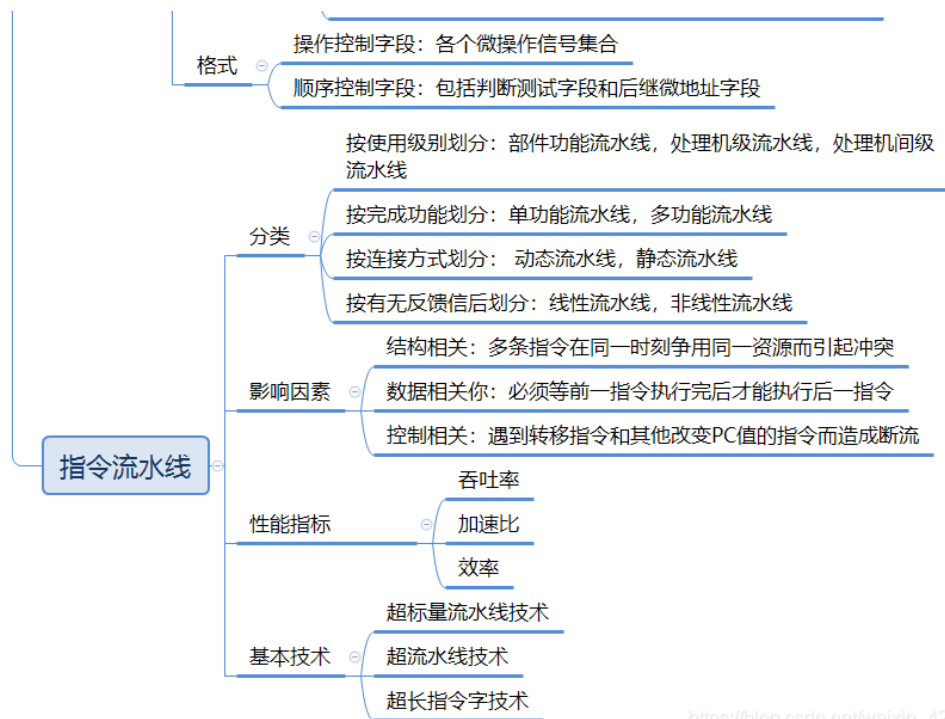
硬布线控制器：由复杂的组合逻辑门电路和一些触发器构成，由硬件给出控制信号

微指令控制器

- 概念：将每条机器指令设计成一个微程序，由微指令给出控制信号
- 组成：控制存储器--微指令控制器--微指令形成部件--微地址寄存器

编码方式

- 直接编码：字段中每一位表示一个微命令
- 字段直接编码：把互斥微命令放在一段编码
- 字段间间接编码：某些微命令需要另一个段的微命令来解释



https://blog.csdn.net/weixin_42104154

CPU的具体功能

1) **指令控制**: 完成取指令, 分析指令和执行指令的操作, 即程序的顺序执行。2) **操作控制**: CPU管理并产生由内存取出的每条指令的操作信号, 把各种操作信号送往相应的部件, 从而控制这些部件按指令的要求进行动作。3) **时间控制**: 对各种操作加以时间上的控制。4) **数据加工**: 对数据进行算数和逻辑运算。5) **中断处理**: 对计算机运行过程中出现的异常情况 & 特殊请求进行处理。

控制器的工作原理

根据指令操作码、指令的执行步骤 (微命令序列) 和条件信号来形成当前计算机各部件要用到的控制信号。计算机整机各硬件系统在这些控制信号的控制下协同运行, 产生预期的执行结果。1) 运算器部件通过数据总线与内存、输入输出设备传送数据。2) 输入输出设备通过接口电路与总线相连接。3) 内存、输入输出设备从地址总线接收地址信息, 从控制总线得到控制信号, 通过数据总线与其他部件传送数据。4) 控制器部件从数据总线接收指令信息, 从运算器部件接收指令转移地址, 送出指令地址到地址总线, 还要向系统中的部件提供它们运行所需要的控制信号。

控制器的主要功能

1) 从主存中取出一条指令并指出下一条指令在主存的位置。2) 对指令进行译码或测试, 产生相应的操作控制信号, 以便启动规定的动作。3) 指挥并控制CPU、主存、输入输出之间的数据流动方向。

CPU的控制方式

1) **同步控制方式** 系统有一个统一的时钟, 所有的信号均来自这一统一的时钟信号。控制电路简单, 但运行速度慢。2) **异步控制方式** 不存在基准时钟信号, 各部件按自身固有的速度工作, 通过应答方式进行联络。运行速度快, 但控制电路比较复杂。3) **联合控制方式** 介于同步、异步之间的一种折中。对各种不同的指令的微操作实行大部分采用同步控制, 小部分采用异步控制的办法。

硬布线控制单元设计步骤:

1) **列出微操作命令的操作时间表**。根据微操作节拍安排，列出微操作命令的操作时间表（包含各个机器周期，节拍下的每条指令完成的为操作控制信号）。2) **进行微操作信号综合**。对微操作信号进行综合分析、归类，根据微操作时间表可写出各微操作控制信号的逻辑表达式并进行适当的简化。3) **画出微操作命令的逻辑图**。根据逻辑表达式可画出对应每个微操作信号的逻辑电路图，并用逻辑门电路实现。

硬布线 and 微程序控制器的特点

硬布线控制器的特点：速度快（控制器的速度取决于电路延迟），一旦设计完成后，就不能通过其他额外修改添加新功能（将控制部件视为专门产生固定时序控制信号的逻辑电路，用最少数元件和取得最高速度作为设计目标）。**微程序控制器**：具有规整性，灵活性，可维护性等优点，但由于微程序控制器采用了存储程序原理，所以每条指令都要从控制存储器中取一次，影响速度。

第六章 总线

知识网图

总线

总线概述

概念

总线：是一组能为多个部件分时共享的公共信息传送线路

分时：指同一时刻只允许有一个部件向总线发送信息，若系统中有多个部件，则需要分时地向总线发送信息

共享：总线上可以挂接多个部件，各个部件之间互相交换的信息都可以通过这组线路分时共享

分类 (按功能划分)

片内总线：芯片内部的总线，是CPU芯片内部寄存器与寄存器之间、寄存器与ALU之间的公共连接线

系统总线

- 数据总线：用来传输各功能部件之间相互连接的总线
- 地址总线：用来指出数据总线上的源数据或目的数据所在的主存单元或I/O端口的地址
- 控制总线：传输的是控制信息，包括CPU送出的控制命令和主存返回CPU的反馈信号

通信总线：在计算机系统之间或计算机系统与其他系统之间传送信息的总线

总线结构

单总线结构

- 将CPU、主存、I/O设备都挂在一组总线上
- 结构简单，成本低，易于接入新设备，但带宽低负载重，不支持并发

双总线结构

- 两条总线，一条是主存总线，一条是I/O总线
- 将低速设备从单总线上分离出来，实现存储器总线和I/O总线分离，但需要增加通道等硬件设备

三总线结构

- 主存总线，IO总线，DMA总线
- 提高I/O设备的性能及系统的吞吐量，系统工作效率较低

总线性能指标

总线的传输周期（总线周期）：一次总线操作所需的时间（申请，寻址，传输，结束）

总线时钟周期：即机器时钟周期

总线工作频率：总线周期的倒数

总线时钟频率：即机器时钟频率

总线宽度：总线上能同时传输数据的位数，取决于数据总线根数

总线带宽：总线工作频率*总线宽度/8 (B/s)

总线复用：分时复用

信号线数：数据线，控制线，地址线三种总线总和

总线仲裁

采用总线仲裁部件，以某种方式选择一个主设备优先获得总线控制权

集中式仲裁方式

- 链式查询方式
- 计数器定时查询方式
- 独立请求方式

分布仲裁方式

总线操作和定时

同步定时方式：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送定时系统。

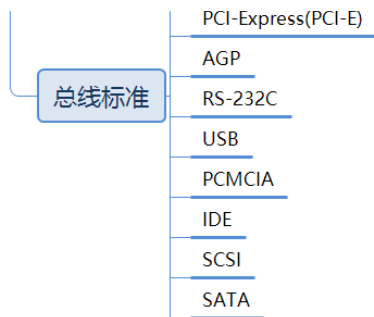
异步方式

- 通过应答信号实现总线控制
- 类型
 - 不互锁方式
 - 半互锁方式
 - 全互锁方式

总线传输阶段

- 申请分配阶段
- 寻址阶段
- 传输阶段
- 结束阶段

- ISA
- EISA
- VESA
- PCI



https://blog.csdn.net/weixin_42104154

总线传输的4个阶段

1) 申请分配阶段。由需要使用总线的主模块提出申请，经总线仲裁机构决定将下一传输周期的总线使用权授予某一申请者（分为传输请求，总线仲裁两阶段）。**2) 寻址阶段。**取得使用权的主模块通过总线发出本次要访问的从模块（或从设备）的地址及相关命令，启动参与本次传输的从模块。**3) 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。**4) 结束阶段。**主模块的有关信息均从系统总线上撤除，让出总线的使用权。

第七章 输入/输出系统

知识网图

输入/输出系统

I/O概念

外设

输入设备：键盘鼠标
输出设备：显示器打印机
存储设备：磁盘光盘RAID

I/O接口

- I/O接口的功能
 - 选址
 - 传送命令
 - 传送数据
 - 反映I/O设备的工作状态
- I/O接口的基本结构
 - 缓冲+译码+状态+命令+逻辑电路
 - 内部接口：与系统总线相连，并行传输
 - 外部接口：与外设相连，传输方式不定
 - 端口是可读写的寄存器，若干端口+逻辑电路构成接口
- 端口与编址
 - I/O端口：指接口电路中可被CPU直接访问的寄存器，主要有数据端口，状态端口和控制端口
 - 编址方式
 - 统一编址（存储器映射方式）：把I/O端口当作存储器的单元进行地址分配。CPU无需专门的I/O指令，用统一的访问指令便可访问I/O端口
 - 独立编址（I/O映射方式）：设置专门的I/O指令来访问I/O端口

IO接口设置数据缓冲寄存器和设备状态寄存器

程序查询方式

- CPU不断查询状态寄存器中的状态，直到外设准备就绪则开始传输，CPU与IO串行工作
- CPU花费很多时间在查询和等待，效率低

中断源

- 内中断：处理器和内存内部的中断，如运算错，页面错，，地址错
- 外中断：处理器和内存外部的中断，如IO，ESC

中断请求

- 硬中断
- 软中断
- 非屏蔽中断
- 屏蔽中断

中断判优

中断响应条件

- 有中断请求
- CPU允许中断及开中断
- 指令执行完毕且无更紧迫的任务

中断隐指令

- 不是指令系统中的真正指令，转去执行中断服务程序由硬件直接实现的
- 操作
 - 保存断点
 - 关中断
 - 中断服务程序寻址

中断向量：即中断服务程序入口地址

I/O方式

程序中断方式

中断处理

- 关中断
- 保存断点
- 中断服务程序寻址
- 保存现场及屏蔽字
- 开中断
- 执行中断服务程序
- 关中断
- 恢复现场和屏蔽字
- 开中断
- 中断返回

多重中断

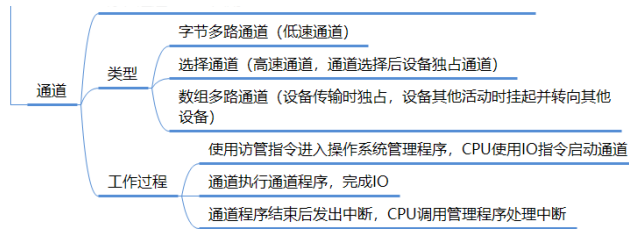
- 允许中断嵌套
- 中断服务程序中提前设置开中断，优先权高的有权中断优先级低的

中断屏蔽

DMA方式

- DMA控制器
 - 外设与内存之间有直接传输通道，数据传输过程不需要CPU参与，中断仅在故障及传输结束
 - 特点
 - IO与主机并行工作，程序和数据传输并行工作
 - 主存既可以被CPU访问，也可被设备访问
- DMA传输过程
 - 预处理：CPU启动IO设备，IO设备准备好后向DMA发送DMA请求，DMA向CPU发送总线请求
 - 传输：传输阶段完全由DMA硬件控制
 - 后处理：DMA发送中断请求，CPU检查传输数据的正确性并决定是否继续传输

是一个处理器，有自己的指令系统，通道执行的程序称为通道程序，通道与主存共用内存



https://blog.csdn.net/weixin_42104154

程序查询方式的工作流程

1) CPU执行初始化程序, 并预置传送参数; 2) 向I/O接口发送命令字, 启动I/O设备; 3) 从外设接口读取其状态信息; 4) CPU不断查询I/O设备状态, 直到外设准备就绪; 5) 传送一次数据; 6) 修改地址和计数器参数; 7) 判断传送是否结束, 若未结束转3), 直到计数器为0

程序中中断的作用

1) 实现CPU与I/O设备的并行工作; 2) 处理硬件故障和软件错误; 3) 实现人机交互, 用户干预机器需要用到中断系统; 4) 实现多道程序, 分时操作, 多道程序的切换; 5) 实时处理需要借助中断系统来实现快速响应; 6) 实现应用程序和操作系统的切换; 7) 多处理器系统中各处理器之间的信息交流和任务切换。

中断处理流程

关中断 保存断点 引出中断服务程序 保存断点和屏蔽字 开中断 执行中断服务程序 关中断 恢复现场和屏蔽字 开中断

DMA方式与中断方式的区别

1) 中断方式时程序的切换, 需要保护和恢复现场; 而DMA处理预处理和后处理, 其他操作均不占用CPU的任何资源 2) 对中断请求的响应只发生在每条指令执行完毕时, 而对DMA请求的响应可以发生在每个机器周期结束时, 只要CPU不占用总线即可 3) 终端传送需要CPU的干预, 而DMA传送过程无需CPU的干预 4) DMA请求的优先级高于中断请求 中断方式具有对异常事件的处理能力, 而DMA仅限于传送数据块的I/O操作 6) 中断方式靠程序传送数据, DMA方式靠硬件传送数据

知识网图下载

https://download.csdn.net/download/weixin_42104154/14922915