计算机组成原理

1 计算机系统概述

1. 计算机的发展 1946年第一台计算机ENIAC问世，电子管时代->晶体管时代->中小规模集成电路->大规模集成电路
2. 摩尔定律 每隔18个月计算机性能可以提高一倍
3. 计算机按指令和数据流可以分为 单指令流和单数据流（SISD，诺依曼体系）；单指令流和多数据流（SIMD）；多指令流和单数据流（MIMD，实际上不存在）；多指令流和多数据流（MIMD，多处理器和多计算机）
4. 微型计算机的发展是以微型处理器为标准的
5. 逻辑电路的高低电平和二进制的0，1对应，所以采用二进制简单可靠
6. 对于某一功能，其在硬件上实现和在软件上实现是等价的，则称其在软硬件上是 **等效** 的（而非等价！！）
7. 冯诺依曼机的特点
    - 运算器 控制器 存储器 输入设备 输出设备组成（运算器和控制器合起来是CPU）
    - 指令和数据以同等地位存储于存储器（无法在存储器上分辨是指令还是数据）
    - 指令由操作码和地址码组成，
    - 以运算器为核心，外设通过运算器（CPU）和存储器传送数据（程序直接控制方式）
    - 基本工作方式是控制流驱动方式
    - 工作的特点是按地址访问并顺序执行指令
8. 现代的计算机以存储器为核心
9. 主存储器是通过按存储单元的地址进行存取的，这种方式被称为按地址存取方式（相联存储器是按内容访问的）（一个存储单元能存储的0，1位数称为存储字长）
10. MAR 用于寻址，位数对应着存储单元的个数；MDR 用于存储一个存储单元的数据；现代计算机中将MAR和MDR放在CPU中
11. 运算器 和运算相关，核心是算术逻辑单元ALU，累加器ACC，乘商寄存器MQ，操作数寄存器X，变址寄存器IX，基址寄存器BR，程序状态字寄存器（和计算结构进行比较），移位器，计数器，前三个必须有
12. 控制器 和指令相关，核心是控制单元CU，程序计数器PC（存放下一条指令的地址），指令寄存器IR
13. 系统软件有 操作系统OS，数据库软件DBMS，语言处理程序，分布式软件系统，网络软件系统，标准库程序，服务性程序（DBMS和数据库系统DBS不同，一般由数据库、数据库管理软件，数据库管理员和应用系统组成，是DBMS+OS+DBA）
14. (PC)->MAR 将PC中的数据传到MAR
15. 地址译码器是主存的构成部分
16. 计算机的性能指标

- 机器字长 一次整数运算所能处理的二进制位数，与CPU寄存器的位数、加法器的位数有关
- 指令字长 一个指令字包含的二进制位数，通常为存储字长的整数倍
- 存储字长 一个存储单元能存储的位数
- 吞吐量 系统在单位时间内处理请求的数量，取决于主存的存取周期
- 响应时间 发送一个请求需要多久才会响应
- CPU时间周期 最小的时间单位，每个动作至少要一个时钟周期，主频是周期的倒数
- CPI 每条指令使用的时钟周期
- MIPS 每秒执行百万条指令；MFLOPS 每秒执行百万条浮点运算；GFLOPS 每秒执行十亿次运算（标记性能最有用的量）

17. 系列机 具有基本的体系结构
18. 固件 将程序固定在ROM中组成的部件称为固件
19. 兼容 计算机软件或者硬件可以在同一系列不同型号的计算机之间通用

2 数据的表示和运算

1. 十进制转为二进制 整数部分：除2取余法（由低到高） 小数部分：乘基取整法（由高到低）
2. BCD码 二进制编码的十进制数 8421码（大于9需要加6修正） 余3码（多3） 2421码（权值排列）
3. ASCII码 127是DEL码，32是SPACE码，编码只32-126的字符是可印刷字符
4. 小端模式 先存储低位字节，后存储高位字节 (high)11223344(low)
5. 校验码 码距：任意两个合法码字之间最少变化的二进制位数称为码距；码距越大，检错能力越强；码距>=2才有校验能力
6. 易错点 一位主存错误码，使用一位奇偶校验码一定可以校验
7. 原码 补码 反码 移码

码	整数正数范围	整数负数范围	小数正数范围	小数负数范围
原码	0-2^n-1	$1-2^n$-0	0-$1-2^{-n}$	$2^{-n}-1$-0
补码	0-2^n-1	-2^n-0	0-$1-2^{-n}$	-1 -0
反码	0-2^n-1	$1-2^n$-0	0-$1-2^{-n}$	$2^{-n}-1$-0
移码	0-2^n-1	-2^n-0	-----	-----

8. 补码的算术移位 将符号位和数值位一起右移一位，并保持补码的符号位不变，可实现除法功能
9. 移码和补码都有唯一的真值0表示
10. 算术移位 正数：原码、补码、反码移位都补0；负数：原码补0，补码左移补0右移补1，反码补1
11. 逻辑移位 将操作数当作无符号数移动，统统补0
12. 循环移位 不带标志位CF的左移右移也需要改变CF的值

13. 符号扩展 正数：原码、补码、反码扩展都补0；负数：原码补0，补码整数补1小数补0，反码补1
14. 溢出判断 三种方法都有异或门实现
    - 单符号位 参与操作的两数符号位相同，结果也没变，则没有溢出
    - 双符号位（模4补码） 00，11 没有变，01 正溢出，10 负溢出
    - 单符号位和数据位的进位 若相同则无溢出，不相同则溢出
15. 定点数乘除法运算总结

乘法类型	符号位参与运算	累加/加减次数	移位方向	移位次数	说明
原码一位乘法	否	n	右	n	部分积两位，乘数1位
补码Booth乘法	是	n+1	右	n	部分积两位，乘数1位
原码加减交替法	否	n+1或者n+2	左	n	若余数最终为负，需要恢复余数
补码加减交替法	是	n+1	左	n	商末位恒置1

16. 凡是原码运算，不论加减乘除，符号位都单独处理，其中乘除运算的结果符号由参与运算的两个操作数符号相 **异或** 得到
17. 强制类型转换 char类型只占1个字节，不是一位
18. 模4补码 只有一个符号位，因为每一个正确的模4补码的两个位都是相同的
19. 浮点数
    - 左规 当结果需要规格化处理时，将尾数算术左移一位，阶码减一，可能需要多次左规
    - 右规 浮点数结果溢出时（01或10），将尾数算术右移一位，阶码加一，只需要右规一次
    - 规格化结果 尾数基数为2时，原码规格化后尾数首位一定是1；补码尾数首位一定和符号位相反；当尾数基数为3时，原码规格化后尾数最高两位不全为0
    - 上溢会报错，下溢当作0处理；下溢的运算结果的绝对值小于机器所能表示的最小绝对值
    - IEEE 754标准 数符+阶码E（移码）+尾数M（原码） 阶码的范围是（1-254 全0表示非规格化数）

类型	阶码	尾数	总位数	偏置值	最小值	最大值	值表示

类型	阶码	尾数	总位数	偏置值	最小值	最大值	值表示
短浮点数	8	23	32	127	$1*2^{1-127}$	$1.111...*2^{254-127}$	$(-1)^s1.M*2^{E-127}$
长浮点数	11	52	64	1023	----	----	

- 浮点数的加减运算
  - 小阶向大阶对齐，阶码小的尾数右移直到阶码相等 舍弃有效位可能会产生误差，影响精度
  - 尾数求和
  - 规格化 左规（补码）：尾数首位和符号位相同；右规（补码）：溢出时
  - 舍入 0舍1入法：右移时被移去的尾数最高数值位为1加1，为0加0；恒置1法：使末位恒置1
  - 溢出判断 01上溢；10下溢
  - 只有对阶和右规需要尾数舍入；右规和尾码都可能引起阶码上溢

20. 类型转换 int->float->double
    - 32bit int 转到 float 只有1+23位的精度，如果int为32为，那么转换也会有损失
    - 如果将float 转到 int，float的小数部分也会有损失
    - 将float 转为double不会丢失精度
21. 使用浮点数是为了增加数据的表示精度
22. 多位加法器设计采用快速进位，对加法器的每一位都会生成两个信号，进位信号g=XY，进位传递信号p=X异或Y

3 存储系统

1. 磁盘分类
   - 存储介质 磁表面存储器（磁盘，磁带）；磁芯存储器（MOS型，双极型）；光存储器（光盘）
   - 存取方式 随机存储器（RAM） 只读存储器ROM 串行访问存储器（顺序存取（磁带），直接存取（磁盘））
2. 直接存取方式和随机存取不同，是介于顺序存取和直接存取之间的方式
3. 存取周期 两次独立的存取操作之间的时间间隔（包括恢复时间）；存取时间 存取操作需要的时间
4. Cache（主存）中的内容是主存（辅存）中的一部分

5. Cache与主存之间的数据调动是硬件自动完成的，对用户透明；而主存和辅存之间的数据调动则是由硬件和系统共同完成的，对应用程序员透明，对系统程序员不透明
6. 破坏性读出和易失性
7. 静态RAM SRAM 双稳态触发器记忆信息 非破坏性读出 易失性存储器（断电没有内容） 存取速度快，集成度低，功耗大，一般用于高速缓冲器
8. 动态RAM DRAM 电路中的栅极电容记忆信息 破坏性读出 易失性存储器 相对慢，集成度高，功耗小，容量大，价格低
    - DRAM采用地址复用技术，地址线为原来的一般，且地址信号分行、列两次传送
    - 刷新方式 集中刷新（在“死区”（此期间停止读写）集中刷新）；分散刷新（把刷新安排到每个存取周期）；异步刷新（每个t时间刷新一次）；透明刷新（刷新的时间在不需要访问存储器的译码阶段，不需要浪费时间）
    - 刷新单位是行，所以只需要行地址；刷新不需要选片，即整个存储器刷新
9. 只读存储器ROM 非易失性（U盘） 掩膜式只读存储器MROM（出厂一次性）、可编程PROM（一次性）、可擦除可变成EPROM（多次编程）、闪存（MOS）、固态硬盘SSD
10. 技巧 SRAM的引脚芯片包括：地址线+数据线+片选线（1条）+读写控制线（可1可2）；DRAM的引脚芯片包括：地址线/2（分用）+数据线+行、列片选（2）+读写控制（可1可2）
11. 主存容量扩展 位扩展法（8b到16b），连接地址线的方式相同，但是连接数据线的方式不同，某一时刻要选到所有芯片；字扩展法（8K到16K），连接数据线的方式和地址线的方式都相同，片选
12. 片选有效信号和访存控制信号有关，低电平有效
13. 双端口RAM 两组相互独立的地址线、数据线、控制线，允许同时异步访问存储单元，同时读、边写边读会发生错误
14. 多模块存储器 核心思想：CPU的执行速度比访存速度快，因此一次从存储器取出多条数据让CPU执行
    - 单体多字存储器 一次读出多条指令
    - 多体并行存储器 高位交叉编址（指令放到一个连续空间，顺序存储器，不能满足程序的局部性原理）；低位交叉编址（指令放到不同存储体上，模块数>=T/r，T存取周期，r总线传送周期）
15. 交叉存储器判断访存冲突：给定的访存地址在相邻的4（4个存储体）次访存中是否出现了同一个访存体
16. Cache与CPU之间的数据交换以字为单位，Cache与主存之间的数据交换是以Cache块为单位的
17. 地址映射 把主存空间映射到Cache；地址变换 访存时把主存地址变换成Cache地址的过程
18. Cache与主存的映射方式
    - 直接映射 主存字块标记+Cache字块地址+字块内地址
    - 全相联映射 主存字块标记+字块内地址 （按内容寻址）
    - 组相联映射 组间直接映射，组内全相联映射 主存字块标记+组地址+字块内地址
19. Cache的总容量包括：存储容量+标记阵列容量（有效位，标记位，一致性维护位，替换算法控制位；前两个一定有，后两个看题眼）
20. Cache的写策略

- 全写法（命中时，必须把数据同时写入Cache和主存），非写分配法（没命中时，只把数据写到内存，不进行调块）
- 写回法（命中时，只把数据写到Cache，只有在换出时才写到主存），写分配法（没命中时，先加载数据到Cache，然后在Cache中写入）

21. 指令Cache和数据Cache分离的主要目的是减少指令流水线资源冲突
22. 对于应用程序员，虚拟存储器是透明的；对于系统程序员，虚拟存储器是不透明的
23. 快表TLB的内容是页表的一部分（虚拟地址）；Cache的内容是主存的一部分（物理地址）

4 指令系统

1. 指令 指令是计算机运行的最小功能单位，一台计算机上所有的指令的集合构成该机的指令系统，位于软件和硬件的交界面上
2. 定长指令字结构（所有指令字长相等） 变长指令字结构 单字长指令 半字长指令 双字长指令
3. 根据指令操作数地址码的数目的不同：
   - 零地址指令 1.不需要操作数 2.两个操作树隐含的从栈顶和次栈顶弹出，运算结果隐含的压入栈中
   - 单地址指令 1.只需要一个地址 2.另一个操作数由累加器ACC提供
   - 二地址指令 目的操作数和源操作数
   - 三地址指令 两个源操作数和一个存放结果
   - 四地址指令 包含下址
4. 程序控制类指令主要包括 无条件转移 有条件转移 子程序调用和返回 循环指令等
5. 指令寻址方式 EA：有效地址
   - 指令寻址 顺序寻址（PC） 跳跃寻址（将下址保存到PC，所有下一条指令仍然是由PC给出）
   - 隐含寻址 隐含的使用ACC作为第二操作数的地址
   - 立即数寻址 指令中的地址字段给出的不是操作数的地址而是操作数
   - 直接寻址 EA=A
   - 间接寻址 EA=(A) 一次间址需要两次访存
   - 寄存器寻址
   - 寄存器间接寻址 EA=(R)
   - 相对寻址 EA=(PC)+A（最重要）
   - 基址寻址 EA=(BR)+A
   - 变址寻址 EA=(IX)+A（便于处理数组问题）
   - 堆栈寻址 硬堆栈（寄存器），软堆栈（内存）
6. 进位/借位标志：CF；零标志ZF；符号标志：SF；溢出标志：OF
7. CISC & RISC
   - 复杂指令系统计算机CISC 长度不固定，指令格式多，寻址方式多，大多兼容，绝大多数为微程序控制

- 精简指令系统计算机RISC 长度固定，指令格式少，寻址方式少，可访存指令只有load/store，通用寄存器数量多，绝大多数为组合逻辑控制（硬布线）

5 中央处理器

1. 控制器的组成 程序计数器PC、指令寄存器IR、指令译码器、地址寄存器MAR、数据寄存器MDR、时序系统、微操作信号发生器
2. 用户可见（透明）：通用寄存器，程序状态字寄存器；用户不可见（不透明）：MAR、MDR、IR
3. 转移指令使用时，需要用到PSW寄存器判断是否满足去转移，如果可以转移就将PC修改为转移指令的目标地址，否则PC的值为下一条地址的地址
4. 指令译码只是对指令的操作码字段译码
5. CPU中不包括地址译码器
6. 间址周期的作用是取操作数的有效地址EA
7. 程序计数器的地址位数是根据能有多少个 **指令** 来确定的；而不是通过计算机主存地址大小（MAR）
8. 指令周期 CPU从主存中每取出并执行一条指令所需要的全部时间；由若干个机器周期组成；每个机器周期又可分为若干个时钟周期
9. 无条件转移指令 JMP X，只包含取址和执行两个机器周期
10. 对于间址寻址的指令，为了取操作数，需要有间址周期访问主存取得有效地址EA 取址-间址-执行
11. 如果CPU采用中断方式与IO通信，还会有中断周期 取址-间址-中断
12. 中断周期中的栈操作是 SP-1，因为计算机中的堆栈的实现都是向低地址增加
13. 指令周期中的数据流（四个机器周期）
    - 取指 PC->MAR->ABUS->MEM; CU发出控制信号->CBUS->MEM; MEM->DBUS->MDR->IR; CU发出读命令->PC=(PC)+1
    - 间址 Ad(IR)(或MDR)->MAR->ABUS->MEM; CU发出读命令->CBUS->MEM; MES->DBUS->MDR(EA);
    - 执行 计算
    - 中断 CU->(SP-1); SP->MAR->ABUS->MEM; CU发出写命令->CBUS->MEM; PC->MDR->DBUS->MEM(SP) 中断程序入口函数
14. 取指操作是自动进行的，控制器不需要得到相应的命令
15. 机器周期的长度通常由 **存取周期** 来确定，在存储字长等于指令字长的条件下，取指周期也可看作机器周期
16. 数据通路的基本结构 CPU内部单总线、CPU内部三总线、专用数据通路
17. 硬布线控制器 由复杂的组合门电路和一些触发器构成，又称组合逻辑控制器
    - CU的信号来源
      - 指令译码器译IR的操作码与时钟配合产生控制信号
      - 时序系统产生的机器周期信号和时钟周期信号
      - 来自执行单元的反馈信号（PSW状态的信号）

    - 系统总线（中断，DMA）
  - CPU控制方式 同步控制（所有信号来自统一的时钟信号），异步控制信号（各部件按照自己的速度工作），联合控制方式（大部分采用同步，小部分采用异步）

18. 微程序控制器 将微操作信号代码化，将微程序存入一个专门的存储器（控制存储器）中，微程序控制信号由微指令产生
    - 微操作有互斥性和相容性之分，互斥性的不允许出现在同一个微周期
    - 微周期 一条微指令执行需要的时间
    - 微指令 比如一个取指周期对应的命令就是微指令，分为微操作码和微地址码 CMAR CMDR（放微指令）
    - 控制存储器 CPU内部，使用ROM实现（非易失）
    - 微指令编码方式 1.直接编码 2.字段直接编码（互斥性指令放在一起通过译码器编码，多空出一个状态） 3.字段间接编码（隐式编码）
    - 水平型微指令（对应每一个控制信号的输出，可以执行几种并行操作，微程序短，执行速度快；缺点是微指令字长，编写程序麻烦）
    - 垂直型微指令（操作码+目的地址+源地址，一条指令定义并执行一个操作，微指令字短，编写程序容易，缺点是微程序字长，执行慢）
19. 机器指令 <==> 微程序；机器周期的指令 <==> 微指令；每个小操作 <==> 微操作/命令
20. 微程序控制器的工作过程 机器开始运行时，自动将取指微程序的入口地址放到CMAR，根据CMAR从CM中取出相应的微指令送到CMDR，之后的每条微操作/命令的CM地址就从CMDR的下地址字段给出；将CMDR的操作码交给CPU处理取到微指令（IR），将Ad(IR)通过微地址生成器生成微地址放到CMAR中，通过CMAR在CM中找到下一条微指令的开始地址；如此反复
21. 若指令系统中有n种机器指令，则控存中的微程序个数至少是n+1，多1个是公共的取指微程序
22. 流水线 部件功能级（将复杂的算术逻辑运算组成） 处理机级（一条指令解释成多个子过程） 处理机间级（每个处理机专门处理一个任务）
    - 影响因素 1.结构相关（资源冲突，CPU） 2.数据相关（数据冲突，解决：阻塞、使用上一条命令的ACC结果而不等待上一条命令回写） 3.控制相关（控制冲突，转移指令等造成断流 解决：预测）
    - 性能指标
      - 吞吐率（单位时间内完成的任务数量） $TP = \frac{n}{n-1+k}$
      - 流水线的加速比 $S = \frac{kn}{(k+n-1)}$
      - 流水线的效率 面积之比
23. 超标量流水线
    - 超标量流水线技术 通过编译优化技术，把可并行执行的指令搭配起来（一个周期完成多条指令）
    - 超流水线技术 通过编译优化，在一个时钟周期内再分段
    - 超长指令字 通过编译优化，将多条能并行执行的指令组合成一条指令

6 总线

1. 总线的特点 分时和共享
2. 猝发传送方式 一个总线周期内传输存储地址连续的多个数据字的总线传输方式
3. 片内总线 CPU内部寄存器之间、寄存器与ALU之间的公共连接线
4. 系统总线 CPU、内存、IO接口之间相互连接的总线（数据、地址、控制）
    - 单总线 CPU、内存、IO设备都连到一个系统总线上
    - 双总线 CPU、内存和通道连一个总线，所有IO设备连另一个总线，由通道管理
    - 三总线 CPU与内存使用主存总线，CPU和IO设备使用IO总线，主存和IO设备使用DMA总线
5. 总线周期 总线的传输周期，一次总线操作的时间（申请、寻址、传输、结束）
6. 总线宽度 通常指数据总线的根数 32根数据总线叫32为总线
7. 总线带宽=总线宽度*总线频率
8. 计算机使用总线结构便于增减外设，并使减少了信息传输线的条数，但相对于专线来说，降低了并行性
9. $2^{16} = 65536$
10. 总线仲裁 解决总线竞争问题

仲裁方式	控制线	解释
链式查询方式	BG:1;BR:1;BS:1	离仲裁器近的优先级高
计数器定时查询方式	BR:1;BS:1;选择线: $log_2 n$	当计数器从上一次的终点开始时，设备使用总线的优先级相等
独立请求方式	BG:n;BR:n;BS:1	每个设备都有独立的请求

11. 总线忙信号的建立者是获得总线控制权的设备
12. 总线的定时方式
    - 同步定时 一个时钟
    - 异步定时 缺点：比同步控制方式复杂，速度比同步定时方式慢；有不互锁、半互锁、全互锁；传送操作按需分配时间
13. 总线标准
    - 系统总线 ISA(Industry Standard Architecture，并行), EISA,
    - 局部总线 VESA(Video), PCI(显卡，并行), PCI-E, AGP(Graphics),
    - 设备总线 USB（串行）

7 输入输出系统

1. IO指令是机器指令的一类，但与其他通用指令有不同的地方
2. 通道程序放在主存中而不是通道中
3. 按照打印机的工作原理，将打印机分为击打式和非击打式两类；按工作方式可分为点阵、针式、喷墨式、激光

4. 统一编址的情况下，没有专门的IO指令。使用访存指令实现IO操作，区分存储单元和IO设备是靠它们各自不同的地址码（不是不同的地址线）
5. 独立编址的情况下，有专门的IO指令，依靠不同的指令来区分存储单元和IO设备，且有专门的访问端口
6. IO指令实现的数据传送通常发生在通用寄存器和IO设备之间
7. 非屏蔽中断和屏蔽中断都是外中断；非屏蔽中断不受中断标志位IF的影响，在IF=0（关中断）的情况下也可以被响应
8. 中断判优 故障中断优先级最高，然后是IO中断；可以由硬件也可以由软件实现
    - 硬件故障中断 > 软件故障中断
    - DMA > IO设备传输的中断
    - 高速设备 > 低速
    - 输入 > 输出
    - 实时设备 > 普通设备
9. 中断隐指令 CPU响应中断后，转去执行中断服务程序，这个操作是由硬件直接完成的，称为中断隐指令；无操作码，不是一条真正的指令，不能被用户执行
    - 关中断（IF=0） 保存断点 引出中断服务程序（将中断服务程序的入口地址传到PC）
10. 中断程序需要做的事：保护现场 开中断（让更高优先级执行） 中断时间处理 关中断 恢复现场 开中断 中断返回
11. 中断向量 不同的设备有不同的中断服务程序，不同的中断服务程序有不同的入口地址，这个入口地址称为中断向量
12. 系统把全部的中断向量集中保存在存储器的某一区域，这个存储区叫中断向量表，即中断服务程序入口地址表
13. 中断向量地址是中断服务程序的入口地址的地址
14. 寻址中断服务程序入口地址（中断向量）的方法 1.硬件向量（产生中断类型号，通过中断类型号找到中断向量存放的地址）
15. 中断服务程序的最后一条指令通常是中断返回指令，使其返回到原服务程序的完成
16. 多重中断和屏蔽 1表示屏蔽该中断源 0表示可以正常申请
17. 中断屏蔽标志可以改变中断处理的次序，而不是中断响应的次序，后者是由硬件电路决定的
18. Cache属于存储设备，不能提出中断
19. 浮点数下溢表示0，不能提出中断
20. DMA是在主存和外设IO直接建立直接的通道，在DMA传输期间，DMA控制器将接管CPU的三大总线，CPU的主存控制信号将被禁止使用，当DMA完成后，才恢复CPU对主存的访问权利，因此DMA控制器必须具有控制系统总线的能力
21. DMA的传输方式（防止DMA和CPU同时访问主存） 1.禁止CPU访问 2.交替访问（CPU的工作周期大于主存存取周期） 3.周期挪用（CPU和DMA控制器同时请求时CPU暂时放弃对主线的控制）
22. 中断和DMA的区别
    - 中断需要保存和恢复现场，而DMA除了预处理和后处理，不需要CPU干预

- 对中断请求的响应只能发生在每条指令执行完（执行周期之后），而对DMA的响应可以发生在任何机器周期之后（取指、间址、执行），只要CPU不占用总线
- 中断传输过程需要CPU干预，而DMA不需要

23. 主存故障引起的中断属于机器校验中断，属于内中断
24. 当用户需要输入输出时，会引起访管中断
25. 程序查询方式中，CPU与外设、传送和主程序都是串行的；中断方式中CPU和外设是并行的，传送和主程序是串行的；而在DMA中二者都是并行的
26. DMA的数据传输是由硬件（DMA控制器）完成的；而中断是通过软件完成的

Date: 2018-12-07

Author: hiro

Created: 2018-12-11 二 22:39