**1、说明计算机系统的层次结构。**

虚拟机器M4（高级语言机器）用编译程序翻译成汇编语言程序

虚拟机器M3（汇编语言机器）用汇编程序翻译成机器语言程序

虚拟机器M2（操作系统机器）用机器语言解释操作系统

传统机器M1（传统语言机器）用微程序解释机器指令

微程序机器M0（微指令系统）由硬件直接执行微指令

**2、说明冯诺依曼体系结构的特点。**

①计算机由运算器、存储器、控制器、输入设备和输出设备五大部件组成

②指令和数据均以同等地位存放于存储器内，并可按地址寻访

③指令和数据均用二进制数表示

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，地址码用来标识操作数在存 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机器以运算器为中心，输入输出设备与存储器间的数据传送通过运算器完成

**3、什么是机器字长、指令字长、存储字长?**

①机器字长：指CPU一次能处理数据的位数，与CPU中寄存器的位数有关（P17）

②指令字长：一个指令字中包含二进制代码的总位数。指令字长取决于操作码的长度、操作码地址的长度和操作码地址的个数。

③存储字长：一个存储单元存储二进制代码的位数（P13）

**4、系统总线包括哪三类？起到什么作用？**

①数据总线：用来传输各功能部件之间的数据信息。

②地址总线：用来指出数据总线上的源数据或目的数据在主存单元的地址或I/O设备的地址

③控制总线：用来发出各种控制信号传输线；还起到监视各部件状态的作用。

**5、总线判优控制是解决什么问题的？总的来说可以分为哪两种方式？**

总线判优控制是为了解决当总线上多个主设备同时申请总线时的使用权分配的问题。按一定的优先等级顺序确定哪个主设备能使用总线，只有获得总线使用权主设备才能开始传送数据。

总线判优控制可分集中式和分布式两种，前者将控制逻辑集中在一处(如在CPU中)，后者将控制逻辑分散在与总线连接的各个部件或设备上。

**6、集中式总线控制优先权仲裁方式有哪三种？各有什么特点？**

①链式查询：只需要很少几根线就能按一定优先次序实现总线控制，很容易扩充设备，但对电路故障很敏感，且优先级别低的设备可能很难获得请求。离总线控制部件最近的设备具有最高的优先级。（仅用2根线确定总线使用权属于哪个设备）【BS:总线忙；BR:总线请求；BG:总线同意】

②计数器定时查询：计数可以从“0”开始，此时一旦设备的优先次序被固定，设备的优先级就按0，1，……，n的顺序降序排列，而且固定不变；计数也可以从上一次计数的终止点开始，即是一种循环方法，此时设备使用总线的优先级相等；计数器的初始值还可由程序设置，故优先次序可以改变。这种方式对电路的故障不如链式查询方式敏感，但增加了控制线数，控制也较复杂。（n个设备需要tmp7827条控制线）【书上写的没有+2】

③独立请求方式：响应速度快，优先次序控制灵活（通过程序改变），但控制线数量多，总线控制更复杂。（n个设备需要2n条控制线）

**7、简述总线周期的4个阶段。**

通常将完成一次总线操作的时间称为总线周期。

①申请分配阶段：由需要使用总线的主模块（或主设备）提出申请，经总线仲裁机构决定下一传输周期的总线使用权授于某一申请者。

②寻址阶段：取得了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）的地址及有关命令，启动参与本次传输的从模块。

③传数阶段：主模块和从模块进行数据交换，数据由源模块发出，经数据总线流入目的模块。

④结束阶段：主模块的有关信息均从系统总线上撤除，让出总线使用权。

**8、总线通信控制是解决什么问题的？总的来说有哪几种控制方式？各自的特点是什么？**

总线通信控制主要解决通信双方如何获知传输开始和传输结束，以及通信双方如何协调如何配合。通常用四种方式：同步通信、异步通信、半同步通信和分离式通信。

1. 同步通信：

优点是规定明确、统一， 模块间的配合简单一致。其缺点是主、从模块时间配合属于强制性“同步”，必须在限定时间内完成规定的要求。并且对所有从模块都用同一限时，这就势必造成，对各不相同速度的部件而言，必须按最慢速度的部件来设计公共时钟，严重影响总线的工作效率，也给设计带来了局限性，缺乏灵活性。同步通信一般用于总线长度较短、各部件存取时间比较一致的场合。在同步通信的总线系统中，总线传输周期越短，数据线的位数越多，直接影响总线的数据传输率。

1. 异步通信

克服了同步通信的缺点，允许各模板速度的不-致性，给设计者充分的灵活性和选择余地。它没有公共的时钟标准，不要求所有部件严格的统一操作时间， 而是采用应答方式(又称握手方式)。即当主模块发出请求(Request)信号时，一直等待从模块反馈回来“响应”(Acknowledge) 信号后，才开始通信。要求主、从模块之间增加两条应答线（握手交互信号线）。异步通信的应答方式又可分为不互锁、半互锁和全互锁三种类型。异步通信可用于并行传送或串行传送。

1. 半同步通信：

既保留了同步通信的基本特点，如所有的地址、命令、数据信号的发出时间，都严格参照系统时钟的某个前沿开始，而接收方都采用系统时钟厚沿时刻来进行判断识别；同时又像异步通信那样，允许不同速度的模块和谐地工作.为此增设了一条“等待”（WAIT非）响应信号线，采用插入时钟（等待）周期的措施来协调通信双方的配合问题。

半同步通信适用于系统工作速度不高但又包含了由许多工作速度差异较大的各类设备组成的简单系统。半同步通信控制方式比异步通信简单，在全系统内各模块又在统一的系统时钟控制下同步工作，可靠性较高，同步结构较方便。其缺点是对系统时钟频率不能要求太高，故从整体上来看，系统工作的速度还不是很高。

（4）分离式通信：

①各模块欲占用总线使用权都必须提出申请.

②在得到总线使用权后，主模块在限定的时间内向对方传送信息，采用同步方式传送，不再等待对方的回答信号

③各模块在准备数据的过程中都不占用总线，使总线可接受其他模块的请求

④总线被占用时都在做有效工作，或者通过它发送命令，或者通过它传送数据，不存在空闲等待时间，充分地利用了总线的有效占用，从而实现了总线在多个主、从模块间进行信息交叉重叠方式传送，这对大型计算机系统是极为重要的。

但这种方式控制比较复杂，一般在普通微型计算机系统很少采用，主要用于大型计算机系统。

**9、串行传输和并行传输有何区别？各适用于什么场合？**

①串行传输是指数据在一条线路上按位依次进行传输，线路成本低，但速度慢，适合于远距离的数据传输。

②并行传输是每个数据位都有一条独立的传输线，所有的数据位同时传输，其传输速度快、成本高，适应于近距离、高速传输的场合。

**10、试比较RAM和ROM。**

（1）RAM

①RAM随机存取存储器，在程序的执行过程中既可以读出又可以写入。

②RAM是易失性的，当断电后数据会丢失。

1. ROM

①ROM是只读存储器，在程序的执行过程中只能读出信息，不能写入信息。

②ROM是非易失性的，断电后数据不会丢失，可以长时间断电保存。

**11、从实现技术的角度，试比较SRAM和DRAM。**

SRAM和DRAM都属随机储存器，即在程序的执行过程中既可读入信息又可写入信息。 SRAM是用触发器原理存储信息的，因此即使信息读出后，它仍保持其原状态，不需要再生，但电源掉电时，原存信息丢失。DRAM是靠电容存储电荷的原理来存储信息，即使电源不掉电，由于电容要放电，信息也会丢失，故需不断地刷新。

（1）SRAM：

①SRAM是一种基于触发器的存储器，使用稳定的存储电路来存储和保持数据。

②每个存储单元由一个存储器单元和控制电路组成，其中存储器单元由多个触发器构成，能够存储比特数据。

③由于采用了触发器结构，SRAM在不断刷新的过程中保持数据的稳定性。

④SRAM的存储密度相对较低，因为每个存储单元需要更多的晶体管来实现，通常需要6个晶体管。

⑤SRAM不需要进行定期刷新操作，数据可以一直保持稳定，无需周期性刷新。

⑥SRAM的访问速度非常快，因为数据存储在触发器中，可以立即读取和写入。

（2）DRAM：

①DRAM是一种基于电容的存储器，使用电容来存储和表示数据。

②每个存储单元由一个电容和一个访问晶体管组成。

③电容在存储器中充电或放电来表示数据的0和1。

④由于电容会逐渐漏电，DRAM需要定期刷新以保持数据的正确性。

⑤DRAM的存储密度较高，因为每个存储单元只需要一个电容和一个访问晶体管，通常只需要1个晶体管和1个电容。

⑥DRAM的访问速度相对较慢，因为数据存储在电容中，需要经过访问晶体管的操作。

在同样大小的芯片中，DRAM（动态RAM）的集成度远高于SRAM（静态RAM）

DRAM行、列地址按先后顺序输送，减少了芯片引脚，封装尺寸也减少

DRAM的功耗比SRAM小

DRAM的价格比SRAM便宜，容量大，存取周期长

DRAM的速度比SRAM低

DRAM需要再生，故需配置再生电路，也需要消耗一部分功率

**12、存储芯片内的地址译码方式有几种？各自特点及应用场合？**

半导体存储芯片的译码驱动方式有两种：线选法和重合法。

（1）线选法：用一根字选择线（字线），直接选中一个存储单元的各位（如一个字节）。这种方式结构较简单，但只适于容量不大的存储芯片。适用于地址线较少的芯片。其特点是地址信号只须经过一个方向的译码就可以选中某一存储单元的所有位。

（2）重合法：适用于地址线较多的芯片。其特点是地址线分成两组，分别经行、列两个方向译码，只有行、列两个方向均选中的存储元才能进行读/写。

**13、什么是动态RAM的刷新？刷新有几种方式？简要说明之。**

刷新的过程实质上是先将原存信息读出，再由刷新放大器形成原信息并重新写入的再生过程。由于存储单元被访问是随机的，有可能某些存储单元长期得不到访问，不进行存储器的读/写操作，其存储单元内的原信息将会慢慢消失。为此，必须采用定时刷新的方法，它规定在一定的时间内，对动态 RAM 的全部基本单元电路必作一次刷新，一般取2 ms ，这个时间称为刷新周期，又称再生周期。刷新是一行行进行的，必须在刷新周期内，由专用的刷新电路来完成对基本单元电路的逐行刷新，才能保证动态 RAM 内的信息不丢失。通常有三种方式刷新：集中刷新、分散刷新和异步刷新。

（1）集中刷新：在规定的一个刷新周期内，对全部存储单元集中一段时间逐行进行刷新，此刻必须停止读/写操作。这段时间称为“死时间”。

（2）分散刷新：对每行存储单元的刷新分散到每个存取周期内完成。不存在停止读/写操作的死时间，但存取周期长了，整个系统速度降低了。

（3）异步刷新：是前两种方式的结合，既可缩短“死时间”，又充分利用最大刷新间隔为2ms的特点。如果将动态 RAM 的刷新安排在 CPU 对指令的译码阶段，由于这个阶段 CPU 不访问存储器，所以这种方案既克服了分散刷新需独占 0.5 µs 用于刷新，使存取周期加长且降低系统速度的缺点，又不会出现集中刷新的访存“死区”问题，从根本上提高了整机的工作效率。

**14、存储容量扩展有哪几种方法？**

（1）字扩展。位扩展是指增加存储字长，

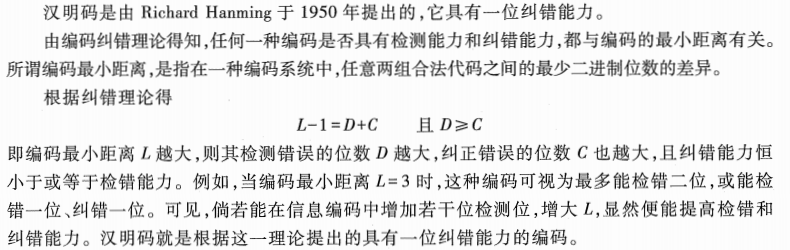
（2）位扩展。字扩展是指增加存储器字的数量。

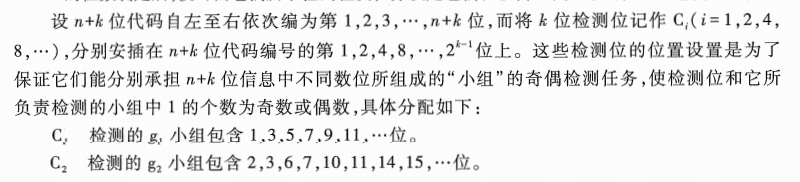
（3）字、位扩展。字、位扩展是指既增加存储字的数量，又增加存储字长。

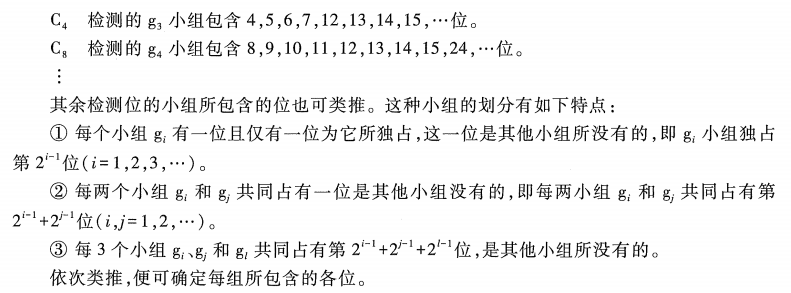
**15、使用汉明码进行校验时，需要增加k位检测位，且k位的检测位和n位的待检测二进制代码位之间要满足什么关系？为什么？**

设欲检测的二进制代码为n位，为使其具有纠错能力，需增添k位检测位，组成n+k位的代码。为了能准确对错误定位以及指出代码没错，新增添的检测位数k应满足：2^k ≥ n+k+1。

汉明码至少要能检验出n+k个1位出错情况，和不出错的情况，共n+k+1种，而k位能检测出2k种错误，所以校验位个数k要满足2^k≥n+k+1。







**16、已知收到的海明码为0100111（按配偶原则配置），试问欲传送的信息是什么？**

P1 = 1⊕3⊕5⊕7 = 0，P2 = 2⊕3⊕6⊕7 = 1，P4 = 4⊕5⊕6⊕7 = 1

所以P4P2P1 = 110，故第六位出错，可纠正为0100101，故要求传送的信息为0101

**17、什么是多体并行存储器？有哪两种存储体编址方式，各自有什么特点？**

多体并行系统就是采用多体模块组成的存储器。每个模块有相同的容量和存取速度，各模块各自都有独立的地址寄存器（MAR）、数据寄存器（MDR）、地址译码、驱动电路和读/写电路，他们能并行工作，又能交叉工作。

多体并行存储器分为高位交叉编址(顺序方式)和低位交叉编址(交叉方式)两种。

①高位交叉：高位地址可表示体号，低位地址为体内地址。按这种编址方式，只要合理调动，使不同的请求源同时访问不同的体，便可实现并行工作。这种编址方式由于一个体内的地

址是连续的，有利于存储器的扩充。

②低位交叉：地位地址用来表示体号，高位地址为体内地址。这种编址方式又称为模M编址（M等于模块数），一般模块数取2的方幂，使硬件电路比较简单。可以在不改变每个模块存取周期的前提下，提高存储器的带宽。

**18、Cache、主存和辅存构成的三级存储系统分别的目的是什么？设计依据是？**

①目的：提高访存速度；扩大存储容量。缓存－主存层次主要解决 CPU 和主存速度不匹配的间题。主存－辅存层次主要解决存储系统的容量问题。

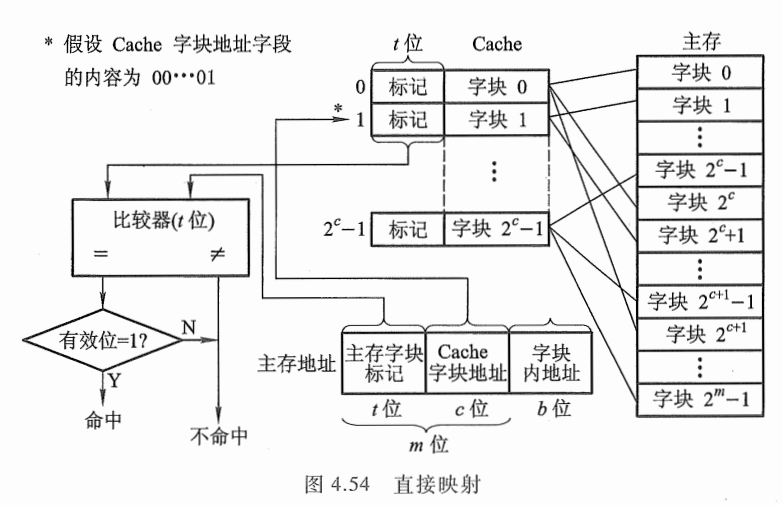
②设计依据：程序访问的局部性原理。CPU 从主存取指令或取数据，在一定时间内，只是对主存局部地址区域的访问。这是由于指令和数据在主存内都是连续存放的，并且有些指令和数据往往会被多次调用（如子程序、循环程序和一些常数），即指令和数据在主存的地址分布不是随机的，而是相对的簇聚，使得 CPU 在执行程序时，访存具有相对的局部性。

**19、请简述Cache的基本工作原理。**

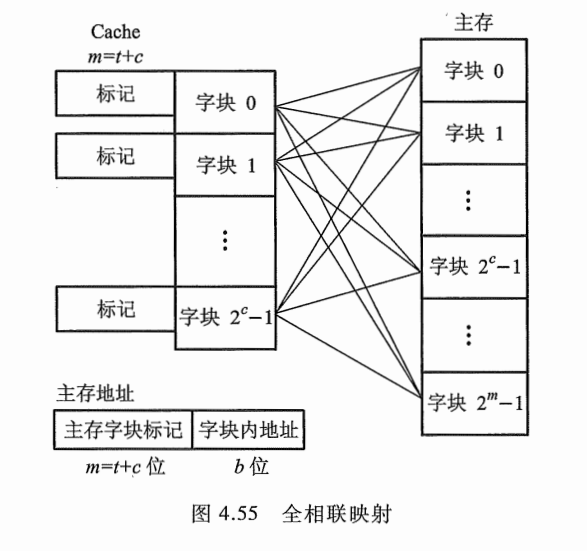
CPU欲读取主存某字时，有两种可能：一种是所需要的字已经在缓存中，即可直接访问Cache；另一种是所需的字不在Cache内，此时需将该字所在的主存整个字块一次调入Cache中（Cache与主存之间是字块传送)。当Cache未满时，主存块可被调入缓存块中，称该主存块与缓存块建立了对应关系。当Cache已满时，无法接收来自主存块的信息，就由Cache内的替换机构按一定的算法从Cache内移除哪块返回主存，并把新的主存块调入Cache中。

**20、试比较主存-Cache管理中三种地址映射的方法，并分别给出在三种地址映射下，主存地址划分。**

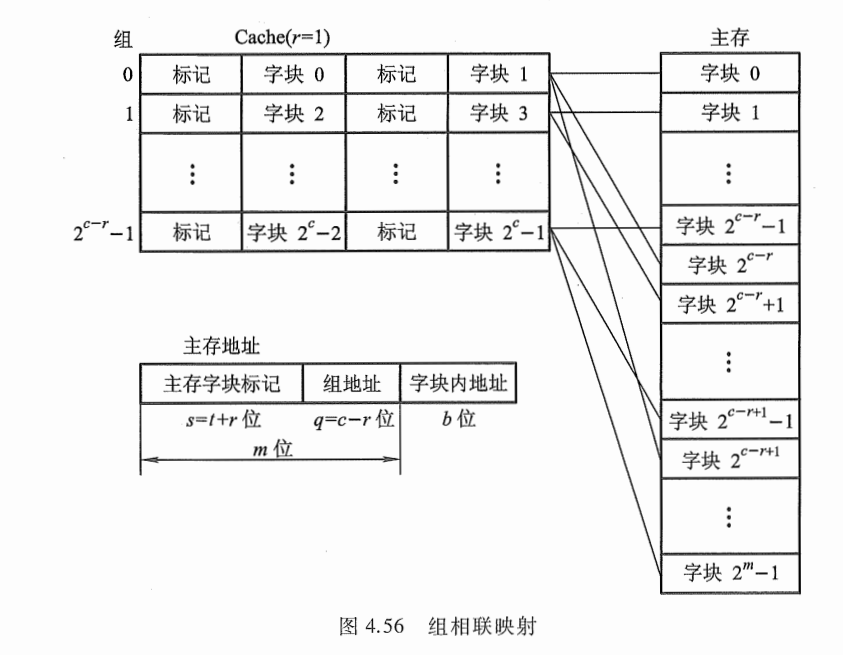
①直接映射：假设缓存共有C块，每个主存块j只与一个缓存块i对应，而每个缓存块可以和多个主存块对应，它们的映射关系是i=j mod C。实现简单，但主存块只能固定地对应某个缓存块，不够灵活，命中率低。



②全相联映射：主存中每一字块映射到Cache中的任何一块位置上，这种方式灵活，命中率也更高，缩小了块冲突率。但所需的逻辑电路甚多，成本高，实际的 Cache 还要采用各种措施来减少地址的比较次数。



③组相联映射：是直接映射和组相联映射的一种折中。假设缓存共有Q组，每组内有R块，并用i表示缓存的组号，j表示主存的块号，它们的映射关系是i=j mod Q，即某个主存块j按模Q映射到缓存第i组内的任一块。这种方法比直接映射灵活，命中率高，比全相联映射成本低，是它们的折中，被广泛应用。



**21、Cache-主存间的替换策略有哪些，请简述。**

（1）先进先出( First-In-First-Out，FIFO)算法

FIFO算法选择最早调入Cache的字块进行替换，它不需要记录各字块的使用情况，比较容易实现，开销小,但没有根据访存的局部性原理，故不能提高Cache的命中率。因为最早调人的信息可能以后还要用到,或者经常要用到，如循环程序。

（2）近期最少使用( Least Recently Used，LRU)算法

LRU算法比较好地利用访存局部性原理，替换出近期用得最少的字块。它需要随时记录Cache中各字块的使用情况，以便确定哪个字块是近期最少使用的字块。它实际是种推测的方法，比较复杂，-般采用简化的方法，只记录每个块最近一次使用的时间。LRU算法的平均命中率比FIFO的高。

（3）随机法

随机法是随机地确定被替换的块.比较简单,可采用一个随机数产生器产生一个随机的被替换的块,但它也没有根据访存的局部性原理，故不能提高Cache的命中率。

**22、发生主存写请求时，如何保证Cache和主存的一致性？**

①写直达法(Write-through)，又称为存直达法(Store-hrough)，即写操作时数据既写人Cache又写人主存。它能随时保证主存和Cache的数据始终一致 ,但增加了访存次数。

②写回法( Write-back),又称为拷回法( Copy-back),即写操作时只把数据写人Cache而不写人主存，但当Cache数据被替换出去时才写回主存。可见写回法Cache中的数据会与主存中的不一致。为了识别Cache中的数据是否与主存一致Cache中的每一块要增设个 标志位。该位有两个状态:“清”(表示未修改过，与主存一致)和“浊”(表示修改过:与主存不一致)。在Cache替换时，“清”的Cache块不必写回主存,因为此时主存中相应块的内容与Cache块是一致的。在写Cache时，要将该标志位设置为“浊”替换时此Cache块要写回主存，同时要使标志位为“清”。

写回法和写直达法各有特色。在写直达法中，由于 Cache 中的数据始终和主存保持一致，在读操作 Cache 失效时，只需选择一个替换的块（主存块）调入 Cache ，被替换的块 (Cache 块）不必写回主存。可见读操作不涉及对主存的写操作。因此这种方法更新策略比较容易实现。但是在写操作时，既要写入 Cache 又要写入主存，因此写直达法的”写“操作时间就是访问主存的时间。

在写回法中，写操作时只写入 Cache ，故”写“操作时间就是访问 Cache 的时间，因此速度快。这种方法对主存的写操作只发生在块替换时，而且对 Cache 中一个数据块的多次写操作只次写入主存，因此可减少主存的写操作次数。但在读操作 Cache 失效时要发生数据替换，引起被替换的块写回主存的操作，增加了 Cache 的复杂性。

**23、I/O地址码的编址方式有哪两种？**

通常将 I/0 设备码看作地址码，对 I/0 地址码的编址可采用两种方式：统一编址或不统一编址。

①统一编址就是将 I/0 地址看作存储器地址的一部分。对I/O设备的访问，所用的指令与访存指令相似。显然统一编址占用了存储空间，减少了主存容量，但无须专用I/0 指令。

②不统一编址就是指 I/0 地址和存储器地址是分开的，所有对 I/0备的访问必须有专用的 I/0 指令。不统一编址由千不占用主存空间，故不影响主存容量，但需设 I/0 专用指令。

**24、I/O与主机交换信息有哪几种控制方式？各有何特点？**

（1）程序查询方式

程序查询方式是由 CPU 通过程序不断查询 I/0 设备是否己做好准备，从而控制 I/0 设备与主机交换信息。其特点是主机与I/O串行工作。CPU启动I/O后，，若设备准备就绪，CPU便转入处理I/O与主机间传送信息的程序；若设备未做好准备，则CPU反复查询，“踏步”等待直到I/O准备就绪为止。可见这种方式CPU效率很低。

（2）程序中断方式

程序中断方式是指CPU 在启动 I/0 设备后，不查询设备是否已准备就绪，继续执行自身程序，只是当 I/0 设备准备就绪并CPU 发出中断请求后才予以响应。其特点是主机与I/O并行工作。CPU启动I/O后，不必时刻查询I/O是否准备好，而是继续执行程序。当I/O准备就绪时，向CPU发中断请求信号，CPU在适当的时候响应I/O的中断请求，暂停现行程序为I/O服务。这种方式消除了“踏步”现象，提高了CPU的效率

1. DMA方式

主存与 I/0 设备之间有一条数据通路，主存与 I/0 设备交换信息时，无须调用中断服务程序。其特点是主机与I/O并行工作，主存与I/O之间有一条直接数据通路。CPU启动I/O后，不必查询I/O是否准备好，当I/O准备就绪后，发出DMA请求，此时CPU不直接参与I/O和主存间的信息交换，只是把外部总线（地址线、数据线和有关的控制线）的使用权暂时交赋予DMA，仍然可以完成自身内部的操作（如加法、移位等），故不必中断现行程序，只需暂停一个存取周期访存（即周期挪用），CPU的效率更高。

1. 通道方式

通道是一个具有特殊功能的处理器，CPU把部分权力下放给通道，由它实现对外围设备的统一管理和外围设备与主存之间的数据交换，大大提高了CPU的效率，但它是以花费更多的硬件为代价的

1. I/O处理机方式

它是通道方式的进一步发展，CPU将I/O操作及外围设备的管理权全部交给I/O处理机，其实质是多机系统，因而效率有更大提高

**25、什么是I/O接口，它与端口有何区别？为什么要设置I/O接口，I/O接口如何分类。**

（1）I/O接口通常是指主机与I/O设备之间设置的一个硬件电路及其相应的软件控制。

（2）接口可以看作两个系统或两个部件之间的交接部分，它既可以是两种硬设备之间的连接电路，也可以是两个软件之间的共同逻辑边界。

端口：指接口电路中的一些寄存器，这些寄存器分别用来存放数据信息、控制信息和状态信息。，相应的端口分别称为数据端口、控制端口和状态端口。若干个端口加上相应的控制逻辑才能组成接口。 CPU 通过输入指令，从端口读入信息，通过输出指令，可将信息写入端口中。

（3）设置I/O接口的理由如下：

①一台机器通常配有多台I/O设备，他们各自有其设备号（地址），通过接口可实现I/O设备的选择。

②I/O设备种类繁多，速度不一，与CPU速度相差可能很大，通过接口可实现数据缓冲，达到速度匹配。

③有些I/O设备可能串行传送数据，而CPU一般为并行传送，通过接口可实现数据串 - 并格式的转换。

④I/O设备的输入输出电平可能与CPU的输入输出电平不同，通过接口可实现电平转换

CPU启动I/O设备工作，要向I/O设备发各种控制信号，通过接口可传送控制命令。

⑤I/O设备需将其工作状态（如“忙”、“就绪”、“错误”等）及时向CPU报告，通过接口可监视设备的工作状态，并可保存状态信息，供CPU查询。

（4）I/O接口的分类：

①按数据传送方式分类：并行接口和串行接口

并行接口是将一个字节（或一个字）的所有位同时传送；；串行接口是在设备与接口间一位一位传送。

②按功能选择的灵活性分类：可编程接口和不可编程接口

可编程接口的功能及操作方式可用程序来改变或选择；；不可编程接口不能由程序来改变其功能，但可通过硬连线逻辑来实现不同的功能。

③按通用性分类：通用接口和专用接口

通用接口可供多种 I/0 设备使用；专用接口是为某类外设或某种用途专门设计的。

④按数据传送的控制方式分类：程序型接口和DMA型接口

程序型接口用于连接速度较慢的 I/0 设备；DMA 型接口用千连接高速I/0 设备。

**26、程序查询方式和程序中断方式都由程序实现外围设备的输入输出，它们有何不同？**

（1）程序查询方式是用户在程序中安排一段输入输出程序，它由I/O指令、测试指令和转移指令等组成。CPU一旦启动I/O后，就进入这段程序，时刻查询I/O准备的情况，若未准备就绪就踏步等待；若准备就绪就实现传送。在输入输出的全部过程中，CPU停止自身的操作。

（2）程序中断方式虽也要用程序实现外部设备的输入、输出，但它只是以中断服务程序的形式插入到用户现行程序中。即CPU启动I/O后，继续自身的工作，不必查询I/O的状态。而I/O被启动后，便进入自身的准备阶段，当其准备就绪时，向CPU提出中断请求，此时若满足条件，CPU暂停现行程序，转入该设备的中断服务程序，在服务程序中实现数据的传送。

从宏观上分析，程序中断方式克服了程序查询方式中的 CPU" 踏步“现象，实现CPU I/0 的并行工作，提高了 CPU 的资源利用率。

**27、以I/O设备的中断处理过程为例，说明一次程序中断的全过程。**

①由CPU发启动I/O设备指令，将接口中的B置“1”，D置“0”

②接口启动输入设备开始工作

③输入设备将数据送入数据缓冲寄存器

④输入设备向接口发出“设备工作结束”信号，将D置为“1”，B置为“0”，标志设备准备就绪

⑤当设备准备就绪（D=1），且本设备未被屏蔽（MASK=0）时，在指令执行阶段的结束时刻，由CPU发出中断查询信号

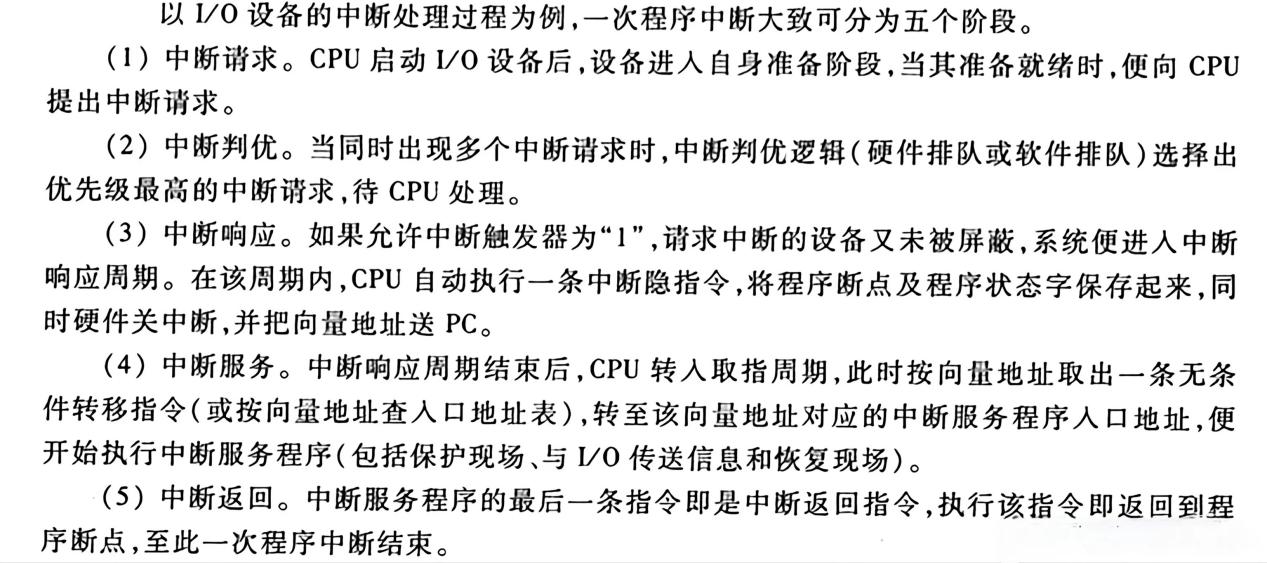
⑥设备中断请求触发器INTR被置为“1”，标志设备向CPU提出中断请求。与此同时，INTR送至排队器，进行中断判优

⑦若CPU允许中断（EINT=1），设备又被排队选中，即进入中断相应阶段，由中断响应信号INTA将排队器输出送至编码器形成向量地址

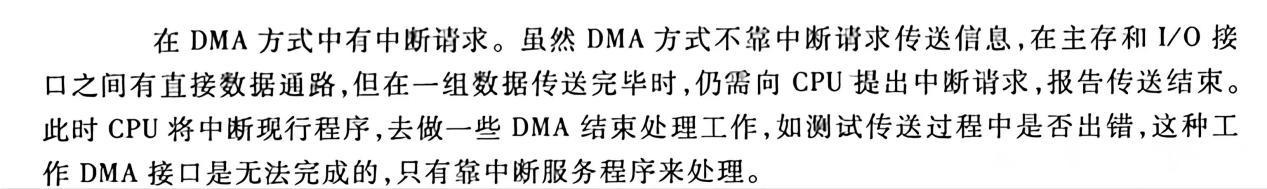
⑧向量地址送至PC，作为下一条指令的地址

⑨由于向量地址中存放的是一条无条件转移指令，故这条指令执行结束后即无条件转至该设备的服务程序入口地址，开始执行中断服务程序，进入中断服务阶段，通过输入指令将数据缓冲寄存器的输入数据送至CPU的通用寄存器，再存入主存相关单元

⑩中断服务程序的最后一条指令是中断返回指令，当其执行结束后，即中断返回至原程序的断点处。至此，一个完整的程序中断处理过程即告结束

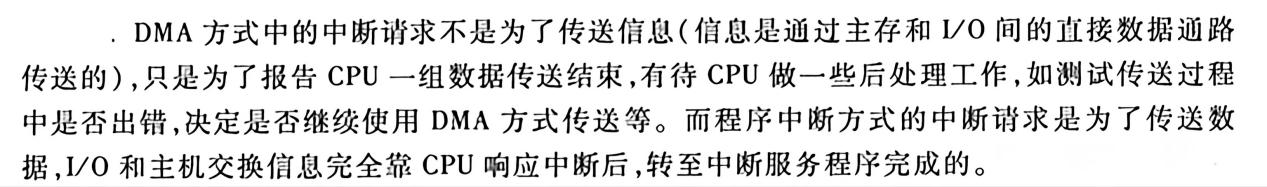


**28、在DMA方式中有没有中断请求，为什么？DMA接口电路中应设置哪些硬件。**



硬件：主存地址寄存器（AR）、字计数器（WC）、数据缓冲寄存器（BR）、DMA控制逻辑、中断机构、设备地址寄存器（DAR）

**29、DMA方式中的中断请求和程序中断中的中断请求有何区别？**

****

**29-1、与程序中断方式相比，DMA方式有哪些特点？**

①从数据传送看，程序中断方式靠程序传送，DMA方式靠硬件传送

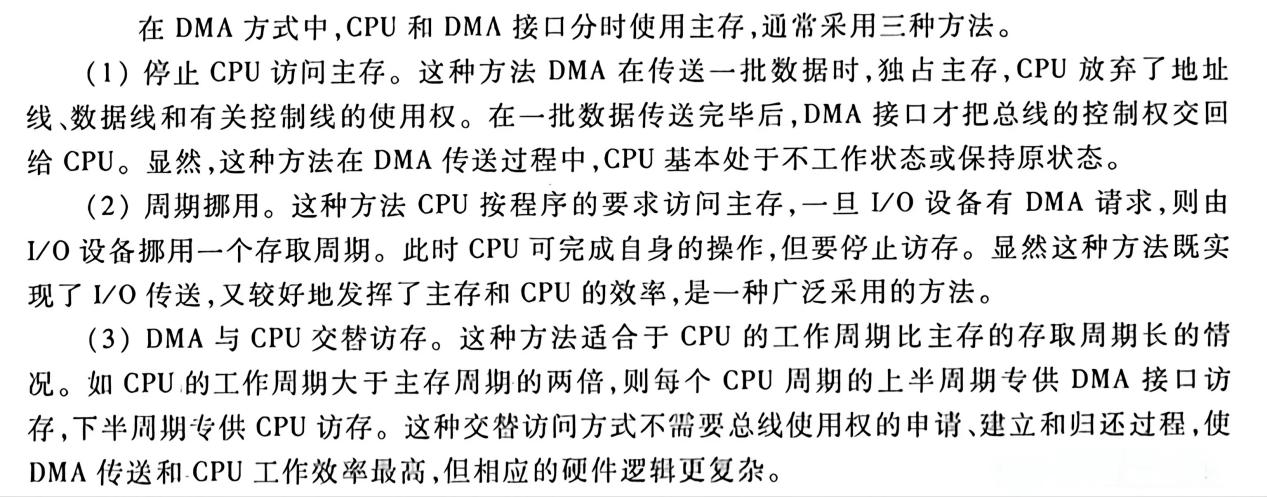
②从CPU响应时间看，程序中断方式是在一条指令执行结束时响应，而DMA方式可在指令周期内的任一存取周期结束时响应

③程序中断方式有处理异常事件的能力，而DMA方式没有这种能力，主要用于大批数据的传送，如硬盘存取、图像处理、高速数据采集系统等，可提高数据吞吐量

④程序中断方式需要中断现行程序，故需保护现场；DMA方式不中断现行程序，无须保护现场

⑤DMA的优先级比程序中断的优先级高

**30、DMA和主存交换数据时有哪三种方法？分别简述。**



**31、简述DMA的工作过程。**

DMA 的数据传送过程分为预处理、数据传送和后处理3个阶段。

1. 预处理：

主存起始地址→DMA

设备地址→DMA

传送数据个数→DMA

启动设备

【在DMA 接口开始工作之前， CPU 必须给它预置如下信息。

①给DMA 控制逻辑指明数据传送方向是输入（写主存）还是输出（读主存）。

②向DMA 设备地址寄存器送入设备号，并启动设备。

③向 DMA 主存地址寄存器送入交换数据的主存起始地址。

④对字计数器赋予交换数据的个数。】

（2）数据传送：继续执行主程序；同时完成一批数据的传送

【主存地址送总线

数据送I/O设备

修改主存地址

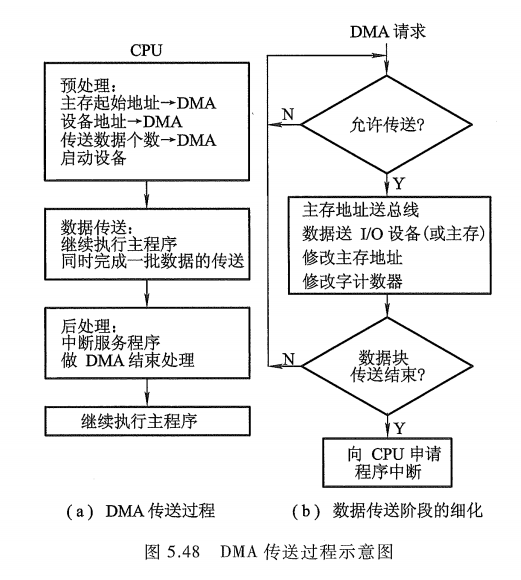
修改字计数器

直到数据块传送结束为止】

（3）后处理：中断服务程序；做DMA结束处理

检验送入主存的数是否正确，决定是否继续用DMA，测试传送过程中是否发生错误，错则转诊断程序。

继续执行主程序



**32、DMA接口的类型有哪几种？**

（1）选择型DMA接口

在物理上可连接多个设备，在逻辑上只允许连接一个设备，即在某一段时间内， DMA 接口只能为一个设备服务，关键是在预处理时将所选设备的设备号送入设备地址寄存器。适用于数据传输率很高的设备。

（2）多路型DMA接口

①链式多路型DMA接口

②独立请求多路型DMA接口

在物理上可以连接多个设备，在逻辑上也允许多个设备同时工作，各个设备采用字节交叉的方式通过DMA接口进行数据传送。适合于同时为多个数据传输率不十分高的设备服务。

**33、说明浮点加/减法运算步骤。**

①对阶：使两数的小数点位置对齐。（小阶对大阶）

②尾数求和：将对阶后的两尾数按定点加减运算规则求和（差）。

③规格化：为增加有效数字的位数，提高运算精度，必须将求和（差）后的尾数规格化。

④舍入：为提高精度，要考虑尾数右移时丢失的数值位。（0舍1入/恒置1法）

⑤溢出判断：即判断结果是否溢出。

**34、说明浮点乘/除法运算的基本步骤。**

①阶码运算。乘法加法、除法减法，同时还要判断是否溢出。

②尾数运算。同定点运算

③尾数规格化

④舍入处理

**35、试比较基址寻址和变址寻址，并举例说明其适用的场合。**

①基址寻址：EA=A+(BR)；变址寻址：EA=A+(IX)。

②都可有效地扩大指令寻址范围

③基址寻址时，基准地址由基址寄存器给出，地址的改变反映在位移量A的取值上；变址寻址时，基准地址由A给出，地址的改变反映在变址值的自动修改上，变址值由变址寄存器给出

④基址寄存器内容通常由系统程序设定，在程序的执行过程中不可变，形式地址A可变。变址寄存器内容通常由用户设定，在程序的执行过程中可变，形式地址A不变。

⑤基址寻址适用于程序的动态重定位，变址寻址适用于数组或字符串处理

**36、任意写出五种寻址方式，令EA为有效地址，A为形式地址，写出这五种寻址方式计算**有效地址的表达式。

①立即寻址: 形式地址A就是操作数

②直接寻址: EA = A

③隐含寻址:操作数地址隐含在操作码中

④间接寻址:EA=(A)

⑤寄存器寻址:EA=Ri

⑥寄存器间接寻址:EA=(Ri)

⑦基址寻址:EA = A + (BR) BR:基址寄存器

⑧变址寻址: EA = A + (IX) IX:变址寄存器

⑨相对寻址:EA = (PC) + A

⑩堆栈寻址:

**37、指令中有哪些字段？各有何作用？如何确定各字段的位数？**

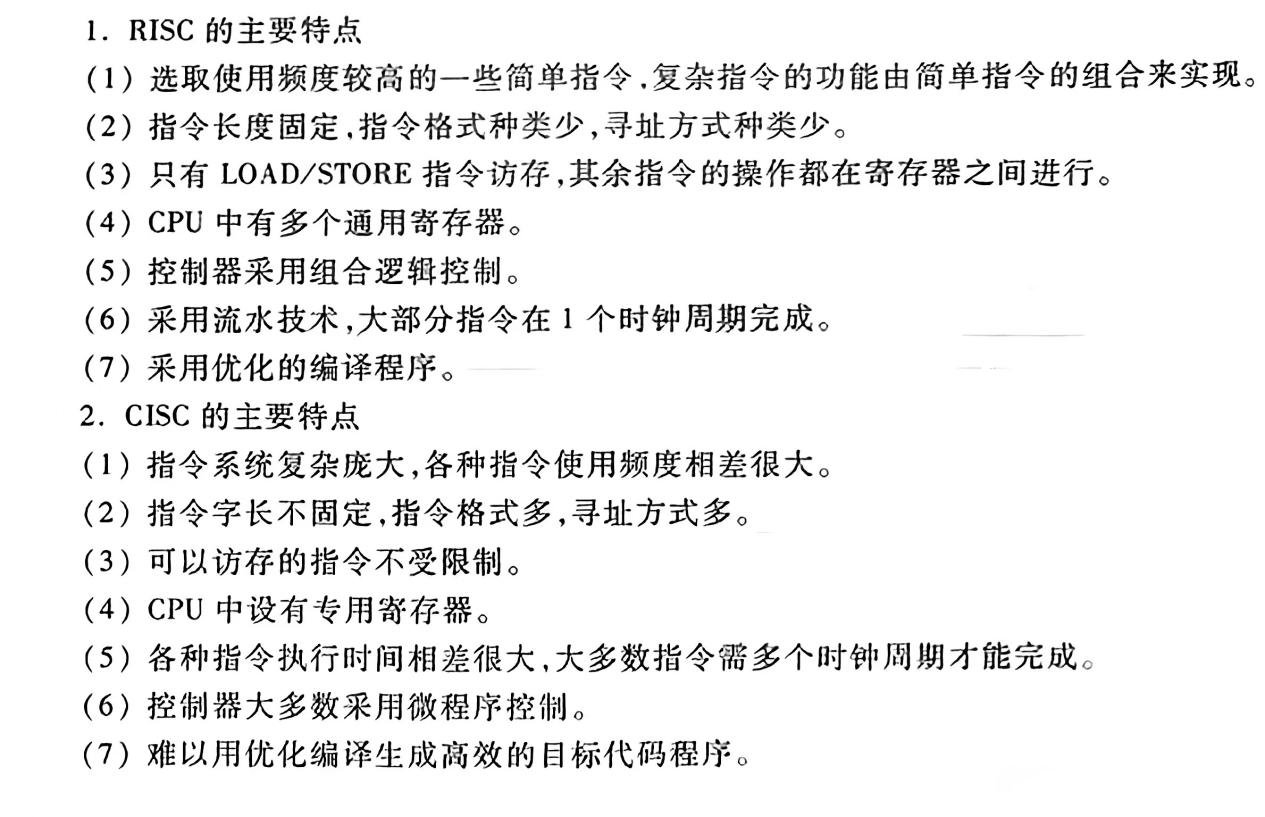
指令中有三种字段：操作码字段、寻址特征字段和地址码字段。

①操作码字段指出机器完成某种操作，其位数取决于指令系统的操作种类。

②寻址特征字段指出该指令以任何方式寻找操作数的有效地址，其位数取决于寻址方式的种类。

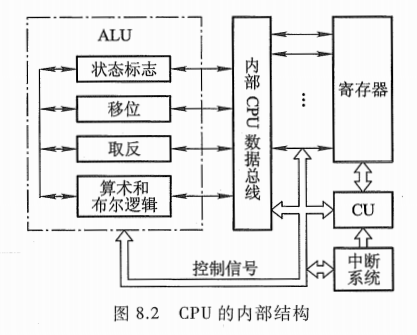
③地址码字段和寻址特征字段共同指出操作数或指令的有效地址，其位数与寻址范围有关。

**38、RISC和CISC指令系统各自的特点。**



**39、CPU有哪些功能？画出其内部的组成框图，并说明每个部件的作用。**

CPU的主要功能：指令控制、操作控制、时间控制、数据加工、处理中断等



寄存器：存放当前指令的地址，存放操作数

CU：发出各种操作命令序列

ALU：实现算术逻辑运算

中断系统：处理异常情况和特殊请求

**40、影响流水线性能的因素有哪些？分别由什么原因导致的？分别采用什么方法解决？**

（1）结构相关：当指令在重叠执行过程中，不同指令争用同一功能部件产生资源冲突时产生的。 解决办法之一是在完成前一条指令对数据的存储器访问时，暂停（一个时钟周期）取后一条指令的操作。

解决办法之二是设置两个独立的存储器分别存放操作数和指令，以免取指令和取操作数同时进行时相互冲突。

（2）数据相关：是流水线中的各条指令因重叠操作，可能改变对操作数的读写访问顺序，从而导致了数据相关冲突。

解决办法之一是后推法，即遇到数据相关时，就停顿后继指令的运行，直至前面指令的结果已经生成。

解决办法之二是采用定向技术，，又称为旁路技术或相关专用通路技术。其主要思想是不必待某条指令的执行结果送回到寄存器后，再从寄存器中取出该结果，作为下一条指令的源操作数，而是直接将执行结果送到其他指令所需要的地方。

（3）控制相关：主要是由转移指令引起的。

解决办法是可以采用尽早判别转移是否发生，尽早生成转移目标地址；预取转移成功或不成功两个控制流方向上的目标指令；加快和提前形成条件码；提高转移方向的猜准率等方法。

**41、流水线性能评价指标有吞吐率、加速比、效率。各自的定义是什么？**

①吞吐率：指单位时间内流水线所完成的指令或输出结果的数量。吞吐率又有最大吞吐率和实际吞吐率之分。

②加速比：指m段流水线的速度与等功能的非流水线的速度之比

③效率：指流水线中各功能段的利用率

**42、某机有五个中断源L0、L1、L2、L3、L4，按中断响应的次序由高向低排序为L0→L1→L2→L3→L4，现要求中断处理次序为L1→L3→L4→L0→L2，写出各中断源的屏蔽字。**

L0：10100

L1：11111

L2：00100

L3：10111

L4：10101

**43、什么是中断？中断系统需要解决哪些问题？**

中断：中断是指计算机运行过程中，出现某些意外情况需主机干预时，机器能自动停止正在运行的程序并转入处理新情况的程序，处理完毕后又返回原被暂停的程序继续运行。简单来说，就是CPU停下当前的工作任务，去处理其他事情，处理完后回来继续执行刚才的任务。

设计中断系统需解决的问题：

①各中断源如何向CPU提出中断请求

②当对个中断源同时提出中断请求时，中断系统如何确定优先响应哪个中断源的请求

③CPU在什么条件、什么时候、以什么方式来响应中断

④CPU响应中断后如何保护现场

⑤CPU响应中断后，如何停止原程序的执行而转入中断服务程序的入口地址

⑥中断处理结束后，CPU如何恢复现场，如何返回到原程序的间断处

⑦在中断处理过程中又出现了新的中断请求，CPU该如何处理

**44、CPU响应中断的条件和时间是什么？**

（1）条件：

①允许中断触发器必须为“1”

②中断源提出请求且未被屏蔽，并且排上队

（2）时间：每条指令执行周期结束时刻

**45、CPU进入中断周期后，完成哪些操作？**

（1）保存断点：保存当前正在执行的程序的地址，以便在中断处理完毕后能返回被中断的原程序继续执行。

（2）形成中断服务程序的入口地址：确定中断服务程序的入口地址，并将其装入程序计数器。（硬件向量法/软件查询法）

（3）关闭中断：在响应中断后，CPU会自动关闭外部硬件中断，以防止在处理中断的过程中再次被中断。【前三个为中断隐指令，CPU自动完成（由硬件）】

（4）保护现场：保存当前程序的运行状态，包括程序计数器、程序状态字、通用寄存器等的内容，以便在中断处理完毕后能恢复到中断前的状态。

（5）中断服务：执行中断服务程序，处理中断请求。

（6）恢复现场：在中断处理完毕后，恢复被保存的程序运行状态。

（7）中断返回：在中断服务子程序的最后要安排一条中断返回指令，执行该指令，系统自动将堆栈内保存的程序计数器和程序状态字值弹出，从而恢复主程序断点处的地址值，使CPU转到被中断的程序中继续执行。

**46、什么是中断隐指令？它有哪些功能。**

中断隐指令：，即在机器指令系统中没有的指令，它是CPU 在中断周期内由硬件自动完成的一条指令。CPU响应中断之后，经过某些操作，转去执行中断服务程序。这些操作是由硬件直接实现的。

功能：保护程序断点，寻找中断程序的入口地址，关中断

**47、什么是中断屏蔽字，它的作用是什么？**

对应每个中断请求触发器就有一个屏蔽触发器，将所有屏敝触发器组合在一起，便构成一个屏蔽寄存器，屏蔽寄存器的内容称为屏蔽字。屏蔽字与中断源的优先级别是一一对应的。

作用：

①在中断服务程序中设置适当的屏蔽字，能起到对优先级别不同的中断源的屏蔽作用。

②采用了屏蔽技术后，可以改变 CPU 处理各中断源的优先等级，从而改变 CPU 执行程序的

轨迹。

**48、什么是多级时序？**

机器周期和节拍（状态）组成了多级时序系统。

机器周期：机器周期可看做是所有指令执行过程中的一个基准时间，机器周期取决于指令的功能及器件的速度

节拍：用时钟信号控制节拍发生器所产生，每个节拍的宽度对应一个时钟周期，是控制计算机操作的最小时间单位。

一个指令周期包含若干个机器周期，一个机器周期又包含若干个时钟周期（节拍），每个指令周期内的机器周期数可以不等，每个机器周期内的节拍数也可以不等。

**49、简述微程序设计的思想。**

将一条机器指令编写成一个微程序，每一个微程序包含若干条微指令，每一条微指令对应一个或几个微操作命令。然后把这些微程序存到一个控制存储器中,用寻找用户程序机器指令的方法来寻找每个微程序中的微指令。由于这些微指令是以二进制代码形式表示的，每位代表一个控制信号(若该位为1,表示该控制信号有效;若该位为0，表示此控制信号无效),因此，逐条执行每一条微指令，也就相应地完成了一条机器指令的全部操作。可见，微程序控制单元的核心部件是一个控制存储器。由于执行一条机器指令必须多次访问控制存储器，以取出多条微指令来控制执行各个微操作，因此要求控制存储器的速度较高。

**50、什么是水平型微指令，什么是垂直型微指令，说明其各自的特点。**

微指令格式与微指令的编码方式有关，通常分为水平型微指令和垂直型微指令两种。

（1）水平型微指令

水平型微指令的特点是一次能定义并执行多个并行操作的微命令。图 10.7 就是典型的水平型微指令。从编码方式看，直接编码、字段直接编码、字段间接编码以及直接和字段混合编码都属千水平型微指令。其中，直接编码速度最快，字段编码要经过译码，故速度受影响。

（2）垂直型微指令

垂直型微指令的特点是采用类似机器指令操作码的方式，在微指令字中，设置微操作码字段，由微操作码规定微指令的功能。通常一条微指令有 1~2 个微命令，控制 1~2 种操作。这种微指令不强调其并行控制功能。

（3）两种微指令格式的比较

①水平型微指令比垂直型微指令并行操作能力强、效率高、灵活性强。

②水平型微指令执行一条机器指令所需的微指令数目少，因此速度比垂直型微指令的速度快。

③水平型微指令用较短的微程序结构换取较长的微指令结构，垂直型微指令正相反，它以较长的微程序结构换取较短的微指令结构。

④水平型微指令与机器指令差别较大，垂直型微指令与机器指令相似。