

组成原理

本题库可用来参考复试出题范围，答案可能存在一些问题，感谢群友们随时指出~

1、DRAM三种刷新方式

集中刷新、分散刷新、异步刷新

集中刷新是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。

分散刷新是对每行存储单元的刷新分散到每个存取周期内完成。其中，把机器的存储周期分为两半，前半段用来读写操作，后半段用来刷新。

异步刷新是前两种方法的结合，他即可缩短死时间，又可以充分利用最大刷新间隔为2ms的特点。即若存取周期0.5us，排列从128*128的存储芯片，可采取在2ms内对128行各刷新一遍，即每隔 $2000\text{us}/128=15.6\text{us}$ 刷新一行，每行刷新时间为0.5us

2、精简指令集(RISC)和复杂指令集(CISC)的区别

从硬件角度来看CISC处理的是不等长指令集，它必须对不等长指令进行分割，因此在执行单一指令的时候需要进行较多的处理工作。而RISC执行的是等长精简指令集，CPU在执行指令的时候速度较快且性能稳定。因此在并行处理方面RISC明显优于CISC，RISC可同时执行多条指令，它可将一条指令分割成若干个进程或线程，交由多个处理器同时执行。由于RISC执行的是精简指令集，所以它的制造工艺简单且成本低廉。从软件角度来看，CISC运行的则是我们所熟悉的DOS、Windows操作系统。而且它拥有大量的应用程序。因为全世界有65%以上的软件厂商都理为基于CISC体系结构的PC及其兼容机服务的，象赫赫有名的Microsoft就是其中的一家。而RISC在此方面却显得有些势单力薄。虽然在RISC上也可运行DOS、Windows，但是需要一个翻译过程，所以运行速度要慢许多。

3、2路映射，内存单位16字节，一个缓存块16行，要访问的块地址是1000 0001 1101 0011，怎么判断命中？

二路映射将16行分为8组，每组两块，地址为1000 0001 1101 0011的块，首先将地址mod 8，得3，则查看cache中，第4，5行是否储存地址为1000 0001 1101 0011的块的内容。

4、浮点数标准化的目的？说一下IEEE标准的方法？

浮点数的出现是实际需要，但是用01无法直接表示浮点数，早期的计算机厂商有各自的浮点数表示方案，这会导致不同计算机运算相同数据得到不同的结果，浮点数标准化的方法具有合理性与先进性，现在被广泛采用

IEEE 754为标准浮点数方法，将浮点数分为数符，阶码，和尾数，32位中，符号位1位，阶码8位，偏移量为 2^7-1 ，尾数23位

符号位：1表示负数，0表示正数

阶码：采用移码，实际阶码为数字阶码减去偏移量

尾数：除非阶码为全1或全0，尾数默认为1.xxxx中小数点后面的部分

5、高级语言和低级语言的概念。高级语言相比低级语言的优势

低级语言：泛指机器语言与汇编语言，，机器语言由01组成，直接对应CPU指令与内存位置，可以被CPU直接识别，汇编语言用字母组合来表示01组合，方便人类记忆。

高级语言：高级语言对于低级语言进行封装和抽象，让他更接近我们平常的思维，需要由翻译程序翻译成低级语言才能被计算机识别。

高级语言的优势：它有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好的描述各种算法，而且容易学习掌握

6、什么是单中断，什么是多重中断。

单重中断和多重中断的区别在于“开中断”的设置时间不同。对于单重中断，开中断指令设置在最后“中断返回”之前，意味着在整个中断服务处理过程中，不能再响应其他

中断源的请求。而对于多重中断，开中断指令提前至“保护现场”之后，意味着在保护现场

之后，若有更高级别的中断源提出请求，CPU也可以响应，从而实现中断嵌套，这是二者

的主要区别。

总之多重中断就是当CPU正在执行某个中断服务程序时，另一个中断源又提出了新的中断请求，而cpu又响应了这个新的请求，暂时停止正在运行的服务程序，转去执行新的中断服务程序。。

7、数据线，控制信号线，地址线的数目对计算机有什么影响。总线带宽的决定因素是什么。

数据线的数目决定计算机的存储字长，地址线的数据决定计算机的存储单元个数，控制线的数目决定计算机能够发出信号的种类数目。总线带宽由总线工作频率和总线宽度决定，如频率为33mhz，宽度为32bit（4B）则总线带宽为 $33 \times 4 = 132\text{Mbps}$ 。

8、中断处理的时候，恢复现场为啥要关中断？

中断是指当出现需要时，CPU暂时停止当前程序的执行转而执行处理新情况的程序和执行过程。即在程序运行过程中，系统出现了一个必须由CPU立即处理的情况，此时，CPU暂时中止程序的执行转而处理这个新的情况的过程就叫做中断。

而关中断是指在此中断处理完成前，屏蔽掉其他中断信号，不处理其它中断。保存和恢复现场的时候如果允许其他中断发生，干扰正在处理中的操作系统的信息，会导致很难恢复到原有的状态甚至发生错误

9、指令流水线有哪些冲突？如何去克服它们？

有结构相关、数据相关、控制相关三种冲突。

结构相关：当多条指令进入流水线后，硬件资源满足不了指令重叠执行的要求的产生

的，即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，又称为资源相关。解决方法：可以让流水线在完成前一条指令对数据的存储器访问时，暂停取后一条指令的操作。

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，从而导致数据相关冲突，即后继指令需要用到前面指令的执行结果时产生的。解决方法：后推法：即遇到数据相关时，就停顿后继指令的执行，直至前面指令的结果已经生成。

控制相关：主要是由转移指令引起的，即当流水线遇到分支指令和其他改变PC值的指令时引起的。解决方法：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。

10、中断控制器的功能

中断源的识别、中断源的屏蔽、中断优先级处理等。

11、外存储系统的类别。

硬磁盘、软磁盘、磁带、光盘

12、什么是指令集ISA？它在计算机层次中有什么作用？

1) ISA

指令集体系结构(Instruction Set Architecture, ISA)，简称体系结构或系统结构,它是软件和硬件之间接口的一个完整定义。

ISA定义了一台计算机可以执行的所有指令的集合，每条指令规定了计算机执行什么操作，所处理的操作数存放的地址空间以及操作数类型。ISA规定的内容包括数据类型及格式，指令格式，寻址方式和可访问地址空间的大小，程序可访问的寄存器个数、位数和编号，控制寄存器的定义，I/O空间的编制方式，中断结构，机器工作状态的定义和切换，输入输出结构和数据传送方式，存储保护方式等。因此，可以看出，指令集体系结构是指软件能够感知到的部分，也称软件可见部分。

2) 作用

ISA在编译器编写者（CPU软件）和处理器设计人员（CPU硬件）之间提供了一个抽象层，定义处理器上的软件如何构建，这是ISA的最重要内涵，现代处理器都是支持高级语言编程、操作系统等等特性，ISA要定义出指令集内的指令是如何支撑起C语言里堆栈、过程调用，操作系统里异常、中断，多媒体平台里数字图像处理、3D加速等等。

13、alu的13种指令

对操作数加1，对操作数减1，两个操作数加和减，两个数的与、或、和异或等按位逻辑运算，操作数按位取反，左移、右移。

14、汇编里面怎么判断两个数溢出，各位标志位 NF DF ZF 的变化和表示情况

AF:辅助进位标志位。运算过程中看最后四位，不论长度为多少。最后四位向前有进位或者借位，AF=1,否则AF=0;

DF:方向标志位。在串处理指令中，每次操作后，如果DF=0，si、di递增，如果DF=1，si、di递减；注意此处DF的值是由程序员进行设定的 cld命令是将DF设置为0，std命令是将DF设置为1；

ZF:零标志位。相关指令执行后结果为0那么ZF=1,结果不为0则ZF=0；

15、程序从读取到执行的流程

1) 取指令

根据指令地址（由PC提供），从存储器中取出所要执行的指令

2) 分析指令

① 对取出的指令进行译码分析。确定指令应完成的操作，产生相应操作的控制电位，参与形成该指令功能所需要的全部控制命令（微操作控制信号）。

② 根据寻址方式的分析和指令功能要求，形成操作数的有效地址，并按此地址取出操作数据（运算型指令）或形成转移地址(转移类指令)，以实现程序转移。

3) 执行指令

根据指令分析所产生的操作控制信号和形成的有效地址，按一定的算法形成指令操作控制序列，控制有关部件完成指令规定的功能。一条指令执行结束，若没有异常情况和特殊请求，则按程序顺序，再去取出并执行下一条指令。

16、问题：RISC指令集能否直接操作存储器中的数据？为什么？

不能

RISC与CISC存储器操作的不同，通常RISC拥有很多的通用寄存器，每个通用寄存器可以存放数据或地址，这些寄存器可以为所有的数据操作提供快速的局部数据存储空间，同时处理器也只处理通用寄存器中的数据，使用Load和Store来完成寄存器和外部存储器的数据传送，因为存储器的访问很耗时，把存储器访问和数据处理分开，这样可以反复使用存储在寄存器中的数据，提高处理器效率。而CISC 机器的存储器操作指令多，可直接操作存储器。

17、cache的物理位置

cache是高速缓冲寄存器，他的物理位置介于CPU和主存储器之间，来解决CPU和主存之间速度不匹配的问题。

18、IEEE754对浮点数规格化，什么时候左规什么时候右规

以基数为2为例，其规格化形式为 $1/2 \leq |S| < 1$ ，如果采用双符号位的补码，正数规格化形式为00.1xx...x，负数的规格化形式为11.0xx...x，即尾数的最高位与符号位不同

时，即为规格化形式。

左规：档位数出现00.0xx...x或11.1xx...x时，需左规。左规是尾数左移一位，阶码减一，知道为规格化数为止。

右规：当尾数出现01.xx...x或10.xx...x时，表示尾数溢出，需要进行右规，右规时尾数右移一位，阶码加一。