

抓码计算机考研·组成原理考前必背合集

编前语：

计算机组成原理考前必背资料由抓码专业团队历时两年迭代完善，梳理组成原理必背重要知识点，以简答题的形式呈现。

计组以抽象难啃闻名，如果你是自命题考生，知识点务必多背诵、理解记忆，这是重要考题之一。如果你是 408 考生，熟悉内容、消理解即可，能够帮助你梳理掌握重要知识点。

此外，抓码运营组基于计算机组成原理必背文本制作了带背音频、视频，方便大家在冲刺抢分阶段利用碎片化时间反复回顾、温习知识点。

音频带背：抓码计算机考研微信公众号搜索“音频带背”；喜马拉雅 FM、网易云音乐扫码收听



计算机考研计算机组成原理 必背知识点

计算机考研冲刺阶段必听！利用碎片时间助你...



长按识别

在网易云音乐收听节目



扫描二维码在线收听



每一天的精神食粮



视频带背：微博、B 站、小红书平台搜索“抓码计算机考研”

该博主还有 2166 条精彩微博

快来微博一起看看吧



你感兴趣的视频都在B站

保存图片
打开哔哩哔哩APP
扫码查看UP主



分享于2022-11-16 11:21

小红书

扫描二维码
在小红书找到我



一、计算机系统概述

1. 试说明冯诺依曼计算机的基本特征，请画出其框图并简要说明每个部分的主要功能。

答：1、采用二进制代码形式表示信息。2、采用**存储程序**工作方式。3、计算机硬件系统由五大部件（存储器、运算器、控制器、输入设备和输出设备）组成。4、以**运算器**为中心。

运算器：完成算术和逻辑运算；存储器：存储指令和数据；控制器：负责全机操作；输入输出部件：信息的输入和输出。

CPU：运算器+控制器

主机：CPU+主存储器（主存）

2. 存储程序控制方式：即事先编写程序，再由计算机把这些信息存储起来，然后连续地、快速地执行程序，从而完成各种运算过程。

3. 计算机内部有哪两种信息流，它们之间有什么关系？

答：计算机内部有控制信息流和数据信息流。控制信息流包括指令信息、状态信息、时序信息，这些信息的组合产生各类控制信号，对数据信息进行加工处理，并控制数据信息的流向，实现计算机的各项功能。

4. 试举例说明计算机硬件和软件功能在逻辑上的等价性

答：在计算机中，实际上有许多功能既可以直接由硬件实现，也可以在硬件支持下依靠软件实现，对用户而言，在功能上是等价的。这种情况称为硬、软件在功能上的逻辑等价。例如：硬件可以直接做乘法运算，也可以通过软件用相加和移位的方式实现乘法运算。

5. 说明计算机系统的层次结构。

计算机系统可分为：微程序机器级，一般机器级（或称机器语言级），操作系统级，汇编语言级，高级语言级。

6. 简述 CPU 的主要功能。

CPU 主要有以下四方面的功能：

- (1) 指令控制：程序的顺序控制，称为指令控制。
- (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：对各种操作实施时间上的控制，称为时间控制。
- (4) 数据加工：对数据进行算术运算和逻辑运算处理，完成数据的加工处理。

7. 衡量计算机运算速度的指标。

(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：指系统对请求做出响应的时间，响应时间为 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、主存储器访问、I/O 操作、操作系统开销等时间）的总和。

(3) CPU 时钟周期：通常为节拍脉冲或 T 周期，即主频的倒数，它是 CPU 中最小的时间单位，简称时钟周期（s）

(4) 主频（CPU 时钟频率）：机器内部主时钟的频率，是衡量机器速度的重要参数。

(5) CPI：指令执行所需要的时钟周期数。

(6) CPU 执行时间：运行一个程序所花费的时间

$$CPU \text{ 执行时间} = \frac{\text{时钟周期数}}{\text{时钟频率}} = \frac{IC \times CPI}{\text{时钟频率}} \quad (IC: \text{程序所含指令数})$$

(7) MIPS：表示每秒执行多少百万条指令。

$$MIPS = \frac{\text{指令条数}}{\text{执行时间} \times 10^6} = \frac{\text{主频}}{CPI} \quad (\text{公式注意单位})$$

(8) MFLOPS：表示每秒执行多少百万次浮点运算。

$$MFLOPS = \frac{\text{浮点操作次数}}{\text{执行时间} \times 10^6}$$

(9) 加速比

$$\text{加速比} = \frac{\text{改进前的总执行时间}}{\text{改进后的总执行时间}}$$

8. 单位换算。

| 字符 | 通常意义 | 实际表示 |
|----|-----------|-----------------------------|
| K | 10^3 | $2^{10} = 1024$ |
| M | 10^6 | $2^{20} = 1048576$ |
| G | 10^9 | $2^{30} = 1073741824$ |
| T | 10^{12} | $2^{40} = 1099511627776$ |
| P | 10^{15} | $2^{50} = 1125899906842624$ |

二、数据的表示与运算

1. 试述浮点数规格化的目的和方法

答：浮点的规格化是为了使浮点数尾数的最高数值位为有效数位。当尾数用补码表示时，若符号位与小数点后的第一位不相等，则被定义为已规格化数，否则便是非规格化数。通过规格化，可以保证运算数据的精度。

通常，采用向左规格化，即尾数每左移一位，阶码减 1，直至规格化完成。

2. 请简要说明什么是计算机系统硬件与软件之间的界面，其主要功能是什么？

答：从程序的编制与执行角度看，指令规定了计算机的操作类型及操作数地址，它们是产生各种控制信号的基础。另外，从硬件设计角度看，在设计计算机的时候先要确定硬件能够直接执行哪些操作，表现为一组指令集合，称之为计算机的指令系统。因此，指令系统体现了一台计算机的软、硬件界面。

3. 如果堆栈采用自底向上生成方式，对于下述两种情况，分别讨论压入和弹出时，应先后做哪些操作？

(1) 栈顶单元是已存数据的实单元

(2) 栈顶单元是待存元素的空单元

答：如果是实单元：压栈时先 $SP-1 \rightarrow SP$ ，后存入数据；弹出时先取出数据，后 $SP+1 \rightarrow SP$ 。如果是空单元：压栈时先存入数据，后 $SP-1 \rightarrow SP$ ；弹出时先 $SP+1 \rightarrow SP$ ，后取出数据。

4. 简要叙述三种不同的判断溢出的方法，并分别写出其判溢的表达式。

溢出判断表达式：

$$\text{“溢出”} = \overline{S_A} \overline{S_B} S_f + S_A S_B \overline{S_C}$$

$$\text{“溢出”} = C_f \oplus C$$

$$\text{“溢出”} = S_{f1} \oplus S_{f2}$$

5. 试论证在浮点数加减为什么要对阶，对阶原则反过来为什么不行？

答：对阶目的：使被加数和加数的小数点对齐，即使其阶码相等

对阶原则：小阶向大阶看齐

原因：如果大阶向小阶看齐，随阶码的值减少，为保持数的值不变，则尾数必须左移相应位数，有可能发生符号位及尾数低位的丢失，这会影响精度不会产生错误。

6. 由于浮点数的表示范围宽广，在实际应用中很少出现溢出，仅在理论上的两种极端情况下，才可能出现上溢和下溢，试描述这两种情况。

答：上溢：同号数相加，其中一数的绝对值很大，阶码已达正最大，而和的绝对值又大于 1，则尾数需要右规，阶码加 1，超出了阶码的最大表示范围，此时称为上溢，运算结果错误，CPU 报错停机，并置 PSW 中溢出标志位为 1。

下溢：异号数相加前，两数的绝对值很小，使阶码达到负绝对值最大，且两数相差很小，相加后需左规，则尾数左规时，阶码减小，超出了阶码表示的最小的负数，此时称为下溢，一般当做机器零处理，计算机不报错。

7. 试论证在浮点数除法运算中，为什么经过“尾数调整”的操作后，其结果就必定是规格化结果。

答：尾数调整是指检测被除数尾数的绝对值是否小于除数尾数的绝对值，以确保商的尾数为小数，如果不是，则将被除数尾数右移一位，并相应调整其阶码。由于两操作数均已是规格化数，即 $|M| \geq 1/2$ ，相除后其商的绝对值必然大于等于 $1/2$ ，不需要左规。在进行“尾数调整”后，商的绝对值必然小于 1，不需要右规。所以按上述操作产生的商不需要进行规格化处理。

8. 原码、补码、反码、移码（有符号数）

原码：数值部分为数绝对值，符号位正数就是 0，负数就是 1



反码：正数和原码一样；负数时符号位不变，数值位取反

补码：正数和原码一样，负数时，符号位不变，数值位取反+1

移码：补码的符号位取反。

(1) 对于真值 0，原码和反码各有两种不同的表示形式，而补码只有唯一的表现形式。

(2) 原码、反码表示的正、负数范围相对零来说是对称的；但补码负数表示范围较正数表示范围大，能多表示一个负数，其值等于 -2^n （纯整数）或 -1 （纯小数）。

9、溢出

两个数相加（相减可以转换成相加）

若 X 和 Y 异号，则一定不会产生溢出

若 X 和 Y 同号，则可能会产生溢出。两个正数相加产生的溢出称为正溢出；两个负数相加产生的溢出称为负溢出。

补码的溢出的判断可以用双符号位：

$S_{s1}S_{s2} = 00$ ，结果为正数，无溢出

$S_{s1}S_{s2} = 01$ ，结果正溢

$S_{s1}S_{s2} = 10$ ，结果负溢

$S_{s1}S_{s2} = 11$ ，结果为负数，无溢出

10、IEEE 754 浮点数

| 类型 | 数 符 | 阶码 | 尾数数值 | 总位数 | 偏置值 | |
|-------|--------|----|------|-----|-------|-------|
| | | | | | 十六进制 | 十进制 |
| 短浮点数 | 1 | 8 | 23 | 32 | 7FH | 127 |
| 长浮点数 | 1 | 11 | 52 | 52 | 3FFH | 1023 |
| 临时浮点数 | 1 | 15 | 64 | 64 | 3FFFH | 16383 |

浮点数只有阶码发生溢出才是真的溢出。若阶码用双符号位补码表示，则

01,XXX...X, 表示上溢。此时, 浮点数真正溢出, 计算机需停止运算, 进行溢出中断处理。

10,XXX...X, 表示下溢。浮点数数值趋于零, 计算机不进行溢出处理, 而是当做机器零处理。

三、存储器层次结构、指令系统

1、请说明三级存储体系分别由哪些部分组成, 并比较“CACHE—主存”和“主存—辅存”这两个存储层次的相同点和不同点。

答: 三级存储体由 “CACHE—主存”和“主存—辅存”构成。在 CACHE 和主存之间, 主存和辅存之间分别有辅助硬件和辅助软件负责信息调度, 以便各级存储器能够组成有机的三级存储体。CACHE 和主存构成的系统的内存, 而主存和辅存依靠辅助硬件和辅助软件构成虚拟存储器。

相同点: 出发点相同, 原理相同。

不同点: 目的不同, 数据通路不同, 透明度不同, 未命中时损失不同。

2、CACHE—主存的地址映像方式主要有哪几种?

答: 主存与高速缓存之间的地址映像方式有直接映像、全相联映像、组相联映像。

直接映像方式下主存块只能映像到高速缓存的指定块。这种方式地址转换速度快, 但冲突率较高。

全相联映像方式下主存中的块可以映像到高速缓存中的任意块。

而组相联映像是二者的结合, 它将主存和高速缓存分组, 组间采用直接映像, 组内采用全相联映像。组相联映像和全相联映像速度比较低, 通常适合小容量 CACHE

3、什么是存储器的刷新? 刷新有哪些典型的方式? 每种刷新方式的特点是什么?

答：动态随机存取存储器，利用存储元中的栅极电容存储电荷，电容上有电荷表明存放数据 1，无电荷表示存放数据 0。由于存储元电路存在漏电流，使栅极电容上的电荷流失，因此必须每隔一定时间对存储体中的所有记忆单元的栅极电容补充电荷，这个过程称为刷新。

有三种典型的刷新控制方式：

(1) 集中刷新方式。在最大刷新时间间隔内，前一段时间进行读写保持，后一段时间集中进行刷新。这种方式的主要缺点是在集中刷新这段时间内不能进行存取访问，称之为死区。

(2) 分散刷新方式：这种方式把 CPU 对内存的存取周期扩展为两倍，前半段用来进行读写保持，后半段作为刷新时间。因刷新过于频繁，故影响了系统的速度，但它不存在死区。这种方式不适合高速内存。

(3) 异步刷新方式：将以上两种方式结合起来。首先用最大刷新时间间隔除以要刷新的行数，然后将每段时间分割为两部分，前段时间用于读写保持，后一小段时间用于刷新。这样既保证了内存的刷新，又没有明显死区。

4、微程序编码有哪三种方式？微指令格式有哪几种微程序？控制器的基本结构是怎样的，哪些特点？

答：微程序编码方式有直接表示法，编码表示法，混合表示法。

微指令的格式大体分为两种：水平型微指令和垂直型微指令。

微程序的控制器具有规整性、可维护性、灵活性的优点，可实现复杂指令的操作控制，使得在计算机中可以较方便的增加和修改指令，甚至可以实现其他计算机的指令。

它主要由控制存储器、微指令寄存器和地址转移逻辑三大部分组成。

5、请说明微指令地址的形成方式主要有哪两种？分别是从哪里获得的下一条微指令的微地址的。

答：微地址形成方式：

初始微地址的形成：取机器指令；功能转移

后续微地址的形成：



增量方式：这种方式与工作程序的顺序控制方式类似，即以顺序执行为主，配合各种常规转移方式，比如：顺序执行，无条件转移，条件转移，转子与返回等。

断定方式：这是一种直接给定微地址与测试判定微地址相结合的方式。为实现多路分支，将微地址的若干低位作为可断定的部分，相应地在微指令的顺序控制字段中设置或注明断定条件，即微地址低位的形成条件。

6、与组合逻辑控制方式相比，微程序控制器有何优点？

组合逻辑（三级时序）控制器速度快，但控制较复杂，且功能扩展较难。

微程序控制器有规整性、可维护性的优点。他是一种利用软件设计硬件的技术，可实现复杂指令的操作控制。另外，微程序设计便于计算机功能的扩充，可较方便地增加和修改指令，只需增加或修改一些微程序。

7、请说明 SRAM 的组成结构，与 SRAM 相比，DRAM 在电路组成上有什么不同之处？

SRAM 存储器由存储体、读写电路、地址译码电路、控制电路组成，DRAM 还需要有动态刷新电路。

8、什么是 RISC？RISC 指令系统的特点是什么？

RISC 是精简指令系统计算机，它有以下特点：

- (1) 选取使用频率最高的一些简单指令，以及很有用但不复杂的指令。
- (2) 指令长度固定，指令格式种类少，寻址方式种类少。
- (3) 只有取数/存数指令访问存储器，其余指令的操作都在寄存器之间进行。
- (4) 大部分指令在一个机器周期内完成。
- (5) CPU 中通用寄存器数量相当多。
- (6) 以硬布线控制为主，不用或少用微指令码控制。

一般用高级语言编程，特别重视编译优化工作，以减少程序执行时间。



9、什么是 CISC? CISC 指令系统的特点是什么?

CISC 是复杂指令系统计算机的英文缩写。其特点是:

- 1: 指令系统复杂庞大, 指令数目一般多达 2、3 百条。
- 2: 址方式多
- 3: 指令格式多
- 4: 指令字长不固定
- 5: 可访存指令不加限制
- 6: 各种指令使用频率相差很大
- 7: 各种指令执行时间相差很大
- 8: 大多数采用微程序控制器

10、指令和数据均存放在内存中, CPU 如何从时间和空间上区分它们是指令还是数据?

从时间上讲,取指令事件发生在“取指周期”, 取数据事件发生在“执行周期”。从空间上讲, 从内存读出指令流流向控制器 (指令寄存器)。从内存读出数据流流向运算器 (通用寄存器)。

11、边界对齐的数据存放方式

边界对齐简单地说就是使存储多字节的起始单元刚好是某个多字节读取模块的开始单元, 所以边界对齐的数据存放方式对数据的存放位置有下列要求:

- (1) 8 位数据占一个存储单元, 其地址为 XXXXX (任意)
- (2) 16 位数据占用两个存储单元, 存放数据的起始地址为 X...XXX0 (2 的整数倍)
- (3) 32 位数据占用 4 个存储单元, 存放数据的起始地址为 X...XX00 (4 的整数倍)
- (4) 64 位数据占用 8 个存储单元, 存放数据的起始地址为 X...X000 (8 的整数倍)

12、映射方式关联度分析

关联度是指一个主存块映射到 Cache 中可能存放的位置个数。直接映射的关联度最低，为 1；全相联映射的关联度最高，为 Cache 总行数；N 路组相联映射的关联度居中，为 N。

当 Cache 大小和主存块大小一定时，关联度和命中率、命中时间、标记所占额外开销有如下关系：

(1) 关联度越低，命中率越低。因此，直接映射的命中率最低，全相联映射的命中率最高。

(2) 关联度越低，判断命中的开销越小，命中时间越短。因此，直接映射的命中时间最短，全相联映射的命中时间最长。

(3) 关联度越低，标记所占额外空间开销越小。因此，直接映射额外空间开销最少，全相联映射额外空间开销最大。

13、Cache 的替换算法

直接映射时主存中的每一块只能被放置到 Cache 中唯一的指定位置，若这个位置已有内容，则产生块冲突，原来的块将无条件地被换。而采用全相联映射和组相联映射方式则不同，当从主存向 Cache 传送一个块，而 Cache 中的空间已被占满时，就需要把原来存储的一块替换掉。

3 种常见的替换算法：

1) 随机算法：随机算法完全不管 Cache 过去、现在及未来的使用情况，简单地根据一个随机数，选择一块替换掉。

2) 先进先出算法：按调入 Cache 的先后决定淘汰顺序，即在需要更新时，将最先进入 Cache 的块作为被替换的块。

3) 近期最少使用算法：是把 CPU 近期最少使用的块作为被替换的块。

14、Cache 更新策略

Cache 更新策略又称为写策略，可分为直写法和写回法两种。

直写法是指 CPU 在执行写操作时，必须把数据同时写入 Cache 和主存。当某一块需要替换时，也不必把这一块再写入主存，新调入的块可以立即把这一块覆盖掉。这种实现简单，而且能随时保持主存数据的正确性，但可能增加多次不

必要的主存写入，会降低存取速度。

写回法是指 CPU 在执行写操作时，被写数据只写入 Cache，不写入主存。仅当需要替换时，才把已经修改过的 Cache 块写入主存。在采用这种更新策略的 Cache 块表中，一般有一个标志位（称为脏位），当一块中的任何一个单元被修改时，该标志位被置 1。在需要替换这一块时，如果标志位为 1（表示该块数据已经“脏”了），则必须先把这一块写入主存之后，才能再调入新的块；如果标志位为 0（表示该块数据还是“干净”的），则这一块不必写入主存，只要用新调入的块覆盖这一块即可。这种方法操作速度快，但因主存中的字块未及时修改而可能出错。

15、Cache 和虚拟存储器的异同点

为了解决存储容量、存取速度和价格之间的矛盾，通常把各种不同存储容量、不同存取速度的存储器按一定的体系结构组织起来，形成存储系统。Cache 存储系统和虚拟存储系统都利用了程序的局部性原理，在地址映射和替换算法上也有许多相似之处。它们的主要不同有以下 4 点：

(1) Cache 主要解决主存和 CPU 之间的速度差距，而虚拟存储器主要用来弥补主存和辅存之间的容量差距。

(2) Cache 每次传送的信息块是定长的，只有几十字节；而虚拟存储器信息块划分方案很多，有页式、段式、段页式等，长度为几百字节至几百千字节(KB)。

(3) 在 Cache 存储系统中，CPU 与 Cache 和主存都建立了直接访问的通道，一旦不命中，CPU 就直接访问主存并同时向 Cache 调度信息块，从而减少了 CPU 的等待时间。而辅助存储器与 CPU 之间没有直接通路，一旦主存不命中，只能从辅存调块到主存。因为辅存的速度相对于 CPU 的差距太大，调度需要毫秒级的时间。因此，CPU 一般会去执行另一个程序，等到调度完成后才返回原来的程序继续工作。

(4) Cache 存取信息的过程、地址转换和替换策略全部用硬件实现，对所有程序员均是透明的。而主存-辅存层次的虚拟存储器基本是由操作系统的存储管理软件管理的，仅辅以一些硬件，所以对设计存储管理软件的系统程序员来说是不透明的，而对应用程序员来说是透明的。



16、TLB、页表、Cache 缺失分析

在一个具有 Cache 和虚拟存储器的系统中，CPU 的一次访存操作可能会涉及 TLB、页表、Cache、主存和磁盘的访问。

CPU 访存过程中存在以下 3 种缺失（未命中）情况：

- (1) TLB 缺失：要访问的页对应的页表项不在 TLB 中。
- (2) 缺页：要访问的页不在主存中
- (3) Cache 缺失：要访问的主存块不在 Cache 中。

| 序号 | TLB | 页表 | Cache | 说明 |
|----|-----|----|-------|---------------------------------------|
| 1 | 命中 | 命中 | 命中 | 可能,TLB 命中则页一定命中,信息在主存,就可能在 Cache 中 |
| 2 | 命中 | 命中 | 缺失 | 可能,TLB 命中则页一定命中,信息在主存,但不可能在 Cache 中 |
| 3 | 命中 | 缺失 | 命中 | 不可能,页缺失,说明信息在不在主存中,TLB 中一定没有该页表项 |
| 4 | 命中 | 缺失 | 缺失 | 不可能,页缺失,说明信息在不在主存中,TLB 中一定没有该页表项 |
| 5 | 缺失 | 命中 | 命中 | 可能,TLB 缺失但页可能命中,信息在主存,就可能在 Cache 中 |
| 6 | 缺失 | 命中 | 缺失 | 可能,TLB 缺失但页可能命中,信息在主存,但可能不在 Cache 中 |
| 7 | 缺失 | 缺失 | 命中 | 不可能,页缺失,说明信息不在主存,Cache 中一定也没有该信息 |
| 8 | 缺失 | 缺失 | 缺失 | 可能,TLB 缺失,页也可能缺失,信息不在主存,一定也不在 Cache 中 |

17、相联存储器

常规存储器是按地址访问的，即送一个地址码，选中相应的一个存储单元，然后进行读写操作。而在信息检索一类工作中，需要的却是按信息内容选中相应的存储单元，进行读写。相联存储器不是根据地址访问存储器，而是根据所存数据字的全部内容或部分内容进行存取的，是一种按内容寻址的存储器。相联存储器的特点是每个存储单元都必须有一个处理单元。当对其写入数据时，能够自动



选择一个未用的空单元进行存储；当要读出数据时，不是给出其存储单元的地址，而是直接给出该数据或该数据的一部分内容，相联存储器同时、并行进行的，所以这种基于数据进行读写的机制，其速度比基于地址进行读写的方式要快许多。

18、存储器分类

(1)、按与 CPU 的连接和功能分类

1) 主存储器

CPU 能够直接访问的存储器为主存储器，用以存放当前运行的程序和数据。由于它设在主机内部，又称为内存，因此简称内存或主存

2) 辅助存储器

辅助存储器是为解决主存容量不足而设置的存储器，用以存放当前不参加运行的程序和数据，当需要运行时，成批调入内存供 CPU 使用，CPU 不能直接访问它。由于它是外部设备的一种，所以又称为外存储器，简称外存。

3) 高速缓冲存储器

高速缓冲存储器是一种介于主存与 CPU 之间用于解决 CPU 与主存间速度匹配问题的高速容量小的存储器。它被用于存放 CPU 立即要运行或刚使用过的程序和数据

(2)、按存取方式分类

1) 随机存取存储器 (Random Access Memory —— RAM)

存储器任何单元的内容均可按其地址随机地读取或写入，而且存取时间与单元的物理位置无关。一般主存储器主要由 RAM 组成。

2) 只读存储器 (Read Only Memory —— ROM)

存储器单元任何的内容只能随机地读出，而不能写入新信息，称为只读存储器。只读 可以作为主存储的一部分，用以存放不变的程序和数据。只读存储器可以用作其他固定存储器。例如存放微程序的控制存储器存放字符点阵图案的字符发生器等。

3) 顺序存取存储器 (Sequential Access Memory —— SAM)

存储器所有信息的排列、寻址和读写操作均是按顺序进行的，并且存取时间

与信息在存储器中的物理位置有关，这种存储器称为顺序存取存储器

4) 直接存取存储器 (Direct Access Memory —— DAM)

这种存储器既不像 RAM 那样能随机地访问任何存储单元，也不像 SAM 那样完全被顺序存取，而是介于 RAM 与 SAM 之间的存储器。目前使用的磁盘就属于直接存取存储器。当要存放所需信息时，它要进行两个逻辑动作，第一步寻道，使磁头指向被选磁道，第二步在被选磁道上顺序存取。

19、半导体只读存储器:

1) 掩膜 ROM

掩膜 ROM 中的内容是由半导体存储芯片制造厂家在制造该芯片时，直接写入 ROM 中的，即掩膜 ROM 不是用户可编程 ROM。掩膜 ROM 的主要优点是比其他类型的 ROM 便宜，但是一旦掩膜 ROM 中的某个代码或数据有错误，整批的掩膜 ROM 都得扔掉。

2) 可编程 ROM

可编程 ROM (PROM) 是一种提供给用户，将要写入的信息烧入 ROM。PROM 为一次可编程 ROM。

对 PROM 写入信息需要用一个叫 ROM 编程器的特殊设备来实现这个过程

3) 紫外线擦除 PROM

紫外线实现擦除的 PROM (EPROM) 的目的是要使已写入 PROM 中的信息能被修改，且可被编程、擦除几千次。EPROM 的问题是：需要紫外线设备，擦除芯片的内容耗时为分钟级

4) 电擦除 PROM

与 EPROM 比，电擦除的 PROM (EEPROM) 。一是用电来擦除原有的信息，实现瞬间擦除；二是可以选择擦除具体字节单元的内容；三是可以直接在电路板上进行擦除和编程；要求在电路板上设置擦除和编程电路。

5) 闪存存储器

闪存是电擦除，又被称为闪烁电擦除可编程 ROM。闪存可擦写的次数是有限的。

20、闪存存储器

闪存以单晶体管作为二进制信号的存储单元，它的结构与普通的半导体晶体管非常类似，区别在于闪存的晶体管加入了浮动栅和控制栅——浮动栅用于存储电子。

通道热电子编程（CHE），该方法通过对控制栅施加高电压，使传导电子在电场的作用下突破绝缘体的屏障进行浮动栅内部。

隧道效应法（FN），直接在绝缘层两侧施加高电压形成高强度电场，帮助电子穿越氧化层通道进出浮动栅

闪存存储器可以分为：

1、NOR 型闪存

NOR 型闪存工作时同时使用 CHE 和 FN 两种方法，CHE 用于数据写入，支持单字节或单字编程；FN 法则用于擦除。但 NOR 不能单字节擦除，必须以块为单位或对整片区域执行擦除操作。无法胜任纯数据存储和文件存储之类的应用。支持代码本地直接运行，即应用程序可以直接在闪存内运行，不必再把代码读到系统 RAM 中。手机就是使用 NOR 型闪存的大户

2、NAND 型闪存

NAND 型闪存工作时采用 FN 法写入和擦除，单晶体管的结构相对简单，存储密度较高，擦除动作很快，但缺陷在于读出性能平平且不支持代码本地运行。

NAND 型闪存以块为单位进行擦除操作。闪存的写入操作必须在空白区域进行，如果目标区域已经有数据，必须先擦除后写入。

NAND 型闪存主要用来存储资料，它常常被应用于诸如数码照相机、数码相机及闪存卡等数码产品。

21、总线事务

从请求总线到完成总线使用的操作序列称为总线事务，它是在一个总线周期中发生的一系列活动。典型的总线事务包括请求操作、裁决操作、地址传输、数据传输和总线释放。

（1）请求操作。主设备（CPU 或 DMA）发出总线传输请求，并获得总线使用权授予某一申请者。



(2) 仲裁阶段：总线总裁机构决定下一个传输周期的总线使用权授予某一申请者。

(3) 寻址阶段。主设备通过总线给出要访问的从设备地址及有关命令，启动从模块。

(4) 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。

(5) 释放阶段。主模块的有关信息均从系统总线上撤除，让出总线使用权。

四、CPU、总线和 I/O 系统

1. 简述中断的步骤及中断响应的条件

请求中断的五个步骤：中断请求、中断判优、中断响应、中断处理、中断返回。

进入中断响应的条件：有中断请求、开中断、一般应等待一条指令执行完。

2. CPU 进入中断响应周期要完成什么操作，这些操作由谁完成？

答：（1）关中断（2）保存断点（3）获得服务程序的入口。（4）转向程序运行状态，以开始执行中断服务程序。

以上操作是在中断周期中直接依靠硬件实现的。

3. CPU 响应中断必须满足的三个条件是什么，中断优先权控制电路的功能是什么？

答：三个条件：（1）必须有中断请求（2）必须开中断（3）一般应待一条指令执行完。

中断优先权控制电路的功能：当同时有若干个中断请求时，选优先权最高的中断请求送 CPU。

4. 简述程序中断处理的三个主要步骤，说明为什么在第一步和第三步



处理的前后都要有关中断和开中断。

答：三个步骤：（1）保存被中断程序的断点和现场，判断中断条件，转入相应的中断服务程序入口（2）执行中断服务程序（3）恢复以前保存的现场和断点。

|

“保存断点和现场”以及“恢复现场”都应整体执行，不能被中断，因此执行前应关中断，执行后应开中断，即允许响应其他的中断请求。

5. 简述多重中断系统中 CPU 响应处理一次中断的步骤。

- (1) 关中断
- (2) 保存现场
- (3) 判别中断条件
- (4) 开中断
- (5) 执行中断服务程序
- (6) 关中断
- (7) 恢复现场信息
- (8) 开中断

7、一个计算机系统中的总线，大致分为哪几类？

(1) 同一部件如 CPU 内部连接各寄存器及运算部件之间的总线，称为内部总线。

(2) 同一台计算机系统的各部件，如 CPU、内存、通道和各类 I/O 接口间互相连接的总线，称为系统总线。

8、比较 DMA 方式和程序中断方式，并指出它们各自应用在什么性质的场合。在输入输出系统中，DMA 方式是否可以替代中断方式。



答：DMA 仅限于数据块的输入和输出操作；而程序中断除用于输入和输出之外，还用于故障诊断等任务，意义要广泛的多。中断一般只发生在一条指令周期结束而另一条指令尚未开始的时刻；而 DMA 则可在两个存储周期之间的任意时刻响应。DMA 过程，若遇到出错信号或接受到新的启动指令，则可中断现行的程序，进入中断服务程序，而程序中断除对 DMA 提供后处理外，还可以测试 DMA 的状态或中断条件，以实施对 DMA 及有关设备控制器的监控。

DMA 适用于：高速、批量数据的简单传送。中断方式适用于：处理复杂随机事件、控制中低速 I/O 设备。

不可以。DMA 方式中也需要用中断机制进行同步，而且 DMA 方式适用于大批量数据的高速传输，不适用低速的单字节的数据输入、输出。

9、请简要说明常见的主机与外围设备之间信息传送的控制方式，并指出采用哪种方式 CPU 效率最低，哪种方式效率最高。

四种：程序查询方式、中断方式、DMA 方式和通道方式。程序查询方式 CPU 效率最低。

10、何谓“总线仲裁”？一般采用何种策略进行仲裁，简要说明它们的应用环境。

答：连接到总线上的功能模块有主动和被动两种形态。主方可以启动一个总线周期，而从方只能响应主方的请求。每次总线操作，只能有一个主方占用总线控制权，但同一时间里可以有一个或多个从方。

除 CPU 模块外，I/O 功能模块也可以提出总线请求。为了解决多个主设备同时竞争总线控制权，必须具有总线仲裁部件，以某种方式选择其中一个主设备作为总线的下一次主方。

一般来说，采用优先级或公平策略进行仲裁。在多处理器系统中对 CPU 模块的总线请求采用公平原则处理，而对 I/O 模块的总线请求采用优先级策略。

11、DMA 控制器的组成

(1) 主存地址计数器：用来存放主存中要交换的数据的地址。该计数器的

初始值为主存缓冲区的首地址。当以 DMA 方式传送数据时，每传送一个数据，将主存地址计数器加 1，从而以增量方式给出主存中要交换的一批数据的地址，直至这批数据传送完毕为止。

(2) 传送长度计数器：用来记录传送数据块的长度，其初始值为传送数据块的总字数或总字节数，每传送一个字或一个字节，传送长度计数器自动减 1，当其内容为全 0 时，表示数据块已全部传送完毕

(3) 数据缓冲寄存器。用来暂存每次传送的数据。输入时，数据由外设（如磁盘）先送往数据缓冲寄存器，再通过数据总线送到主存。输出时，数据由主存通过数据总线送到数据缓冲寄存器，再送到外设。

(4) DMA 请求触发器。外设准备好数据后给出一个控制信号，使 DMA 请求触发器置位。

(5) 控制/状态逻辑。该电路由控制和时序电路以及状态标志灯组成，用于指定传送方向，修改传送参数，并对 DMA 请求信号和 CPU 响应信号进行协调和同步。

(6) 中断机构：当一个数据块传送完毕后触发中断机构，向 CPU 提出中断请求，CPU 将进行 DMA 传送的结尾处理。

12、DMA 传送过程

DMA 传送过程可分为 3 个阶段：预处理阶段、数据传送阶段和后处理阶段

(1) 预处理

在 DMA 传送之前必须做准备工作，这些都是由 CPU 完成。CPU 首先执行几条 I/O 指令，用于测试外设的状态、向 DMA 控制器的有关寄存器置初值、设置传送方向、启动该外部设备等。

在这些工作完成之后，CPU 继续执行原来的程序，在外设准备好发送的数据（输入）或接收的数据已处理完毕（输出）时，外设向 DMA 控制器发送 DMA 请求，再由 DMA 控制器向 CPU 发送总线请求。

(2) 数据传送

DMA 的数据传送可以以字（字节）为基本单位，也可以以数据块为基本单位。在数据传送阶段，由 DMA 控制器自动地完成外设与主存之间的数据传送。



此时 CPU 可以继续执行原来的程序，不需要插手任何与数据传送相关的工作。

(3) 后处理

当传送长度计数器计到全 0 时，DMA 传送结束，DMA 控制器向 CPU 发中断请求，CPU 停止原来程序的执行，转去执行中断服务程序，做 DMA 结束处理工作。

13、DMA 控制器的操作过程如下：

以从外设传送一个数据块至主存为例，

- (1) 由外设向 DMA 控制器发出请求信号
 - (2) DMA 控制器向 CPU 发出总线请求信号
 - (3) CPU 向 DMA 控制器发出总线响应信号，此时 DMA 控制器获取了总线的控制权。
 - (4) DMA 控制器向外设发出 DMA 响应信号，表示 DMA 控制器已控制了总线，允许外设与主存交换数据
 - (5) DMA 控制器按主存地址计数器的内容发出地址信号作为主存地址的选择，同时主存地址计数器的内容加 1（或减 1）
 - (6) DMA 控制器发出 IOR 信号到外设，将外设数据读入总线，同时发出 MEMW 信号，将数据总线的数据写入地址总线选中的主存单元
 - (7) 传送长度计数器减 1
- 重复 (5) ~ (7) 步骤，直到传送长度计数器减到全 0 为止，数据块的 DMA 方式传送工作完成。这时,DMA 控制器的 HRQ 降为低电平，将总线控制权交还 CPU。

14、DMA 方式的特点

- (1) 使主存与 CPU 的固定联系脱钩，主存既可被 CPU 访问，又可被外设访问。
- (2) 在数据块传送时，主存地址的确定、传送数据的计数等都由硬件电路直接实现。
- (3) 主存中要开辟专用缓冲区，及时向外设提供数据和接收外设的数据

- (4) 数据传送速度快, CPU 和外设并行工作, 提高了系统的效率
- (5) 在传送开始前要通过程序进行预处理, 在传送结束后要通过中断方式进行后处理。

15、程序查询、程序中断和 DMA 方式的对比

程序查询方式下, 需要将 I/O 设备的工作时间串行插入 CPU 执行程序的时间 (工作时间) 中。由于 I/O 设备速度相对很慢, CPU 将花费大量的时间来等待 I/O 设备 (比如等待打印机完成一行字符的打印, 在等待期间, CPU 无法响应其他的工作)

程序中断方式下, 只有在需要 I/O 操作时发出 I/O 设备的启动命令, 然后 CPU 就可以继续执行程序的其他部分。当 I/O 设备就绪后发出中断请求信号通知 CPU 对 I/O 设备进行一次响应 (比如在打印机打印一行字符期间, CPU 依然能够响应其他的工作)。中断处理时, 插入 CPU 工作时间中的是执行中断服务程序的时间, 这一时间远远小于 I/O 设备完成工作所需的时间

DMA 方式下, 插入 CPU 工作时间中的仅仅是一个存取周期 (对于周期挪用而言) 所占用的时间。对于大量数据的传送来说, 虽然需要插入多个存取周期, 但显然对 CPU 的干扰很小。同时, 由于每次传送能够由硬件在一个存取周期内完成, 从而实现了 I/O 数据的高速传送。

16、I/O 接口的功能

I/O 接口也称 I/O 控制器, 是主机和外设之间的交接界面, 通过接口可以实现主机和外设之间的信息交换。

I/O 接口的主要功能如下:

- (1) 实现主机和外设的通信联络控制
- (2) 进行地址译码和设备选择
- (3) 实现数据缓冲
- (4) 实现数据格式的转换
- (5) 传递控制命令和状态信息

17、总线的主要性能指标

(1) 总线宽度

总线宽度指的是总线的线数,对总线宽度最直接的影响是地址线和数据线的数量,地址线的宽度指明了总线能直接访问存储器的地址空间范围,数据线的宽度指明了访问一次存储器或外设时能够交换的数据位数。

(2) 总线带宽

总线带宽定义为总线的最大数据传输率,即每秒传输的字节数。在同步通信中,总线带宽与总线时钟密不可分,总线时钟频率的高低决定了总线带宽的大小。

总线带宽 = 总线宽度 × 总线时钟频率

(3) 总线负载

总线负载指连接在总线上的最大设备数量,大多数总线的负载能力是有限的。

(4) 总线复用

总线复用是指在不同时段利用总线上同一个信号线传送不同的信号,例如地址总线 and 数据总线共用一组信号线。采用这种方式的目的是减少总线数量,提高总线的利用率。

(5) 总线突发传输

突发数据传输是一种总线传输方式,即在一个总线周期中可以传输存储地址连续的多个数据。

18、总线的主要特性

(1) 物理特性: 又称机械特性,是指总线在物理连接上的特性,它规定了总线的线数以及总线的插头、插座的形状、尺寸和信号线的排列方式等要素

(2) 功能特性: 描述总线中每一根线的功能。例如, CPU 发出的各种控制命令 (如存储器读写、I/O 读写)、外设与主机的同步匹配信号、中断信号、DMA 控制信号等。

(3) 电气特性: 定义了每根线上信号的传递方向及有效电平范围。

(4) 时间特性: 规定了每根线在什么时间有效以及不同信号之间相互配合的时间关系。只有规定了总线上各信号有效的时间关系, CPU 才能正确无误地

使用。

19、组合逻辑控制和微程序控制的比较

组合逻辑（硬布线）控制和微程序控制的主要区别在于控制单元的实现方式不同。具体说明如下：

(1) 组合逻辑控制的控制功能是由组合逻辑电路实现的，由于各个微操作控制信号的逻辑表达式的繁简程度不同，由此组成的控制电路零乱、复杂。而微程序控制的控制功能是由存放微程序的控制存储器和存放当前正在执行的微指令的寄存器实现的，控制电路比较规整。

(2) 对组合逻辑控制来说，因为所有控制信号的逻辑表达式用硬连线固定下来，当需要修改和增加指令时就很麻烦，有时甚至可能需要重新进行设计。而在微程序控制中，各条指令的微操作控制信号的差别仅反映在控制存储器的内容上。如果想扩展和改变计算机的功能，只需在控制存储器中增加新的微指令或修改某些原来的微指令即可。

(3) 在同样的半导体工艺条件下，组合逻辑控制比微程序控制的速度快。这是因为组合逻辑控制的速度主要取决于逻辑电路的延迟，而微程序控制执行的每条微指令都要从控制存储器中读取，影响了速度。

20、控制器的组成

(1) 指令部件。包括程序计数器、指令寄存器、指令译码器、地址形成部件等

(2) 时序部件。包括脉冲源、启停控制逻辑、节拍信号发生器等

(3) 微操作信号发生器（控制单元）

(4) 中断控制逻辑

更多计算机考研资源
扫码关注微信公众号



抓 机 遇 ， 码 未 来