

简答题:

第一章

1)计算机由哪几部分组成?以哪部分为中心?

计算机由运算器、控制器、存储器、输入设备及输出设备五大部分构成,现代计算机通常把运算器和控制器集成在一个芯片上,合称为中央处理器。

而在微处理器面世之前,运算器和控制器分离,而且存储器的容量很小,故而设计成以运算器为中心的结构,其他部件都通过运算器完成信息的传递。

随着微电子技术的进步,同时计算机需要处理、加工的信息量也与日俱增,大量 I/O 设备的速度和 CPU 的速度差距悬殊,故以运算器为中心的结构不能满足计算机发展的要求。现代计算机已经发展为以存储器为中心,使 IO 操作尽可能地绕过 CPU,直接在 IO 设备和存储器之间完成,以提高系统的整体运行效率。

2)主频高的 CPU 一定比主频低的 CPU 快吗?为什么?

衡量 CPU 运算速度的指标有很多,不能以单独的某个指标来判断 CPU 的好坏。CPU 的主频,即 CPU 内核工作的时钟频率。CPU 的主频表示 CPU 内数字脉冲信号振荡的速度,主频和实际的运算速度存在一定的关系,但目前还没有一个确定的公式能够定量两者的数值关系,因为 CPU 的运算速度还要看 CPU 的流水线的各方面的性能指标(架构、缓存、指令集、CPU 的位数、Cache 大小等)。由于主频并不直接代表运算速度,因此在一定情况下很可能会出现主频较高的 CPU 实际运算速度较低的现象。

3) 翻译程序、汇编程序、编译程序、解释程序有什么差别?各自的特性是什么?

翻译程序是指把高级语言源程序翻译成机器语言程序(目标代码)的软件。

翻译程序有两种:一种是编译程序,它将高级语言源程序一次全部翻译成目标程序,每次执行程序时,只需执行目标程序,因此只要源程序不变,就无须重新翻译,请注意同种高级语言在不同体系结构下,编译成目标程序是不一样的,目标程序与体系结构相关,但仍不是计算机硬件能够直接执行的程序。另一种是解释程序,它将源程序的一条语句翻译成对应的机器目标代码,开立即执行,然后翻译下一条源程序语句并执行,直至所有源程序语句全部被翻译并执行完。所以解释程序的执行过程是翻译一句执行一句,并且不会生成目标程序。汇编程序也是一种语言翻译程序,它把汇编语言源程序翻译为机器语言程序。汇编语言是一种面向机器的低级语言,是机器语言的符号表示,与机器语言一一对应。

编译程序与汇编程序的区别:若源语言是诸如 C、C++、Java 等“高级语言”,而目标语言是如汇编语言或机器语言之类的“低级语言”,则这样的一个翻译程序称为编译程序。若源语言是汇编语言,而目标语言是机器语言,则这样的一个翻译程序称为汇编程序。

4)不同级别的语言编写的程序有什么区别?哪种语言编写的程序能被硬件直接执行?

机器语言和汇编语言与机器指令对应,而高级语言不与指令直接对应,具有较好的可移植性。其中机器语言可以被硬件直接执行。

5) .同一个功能既可以由软件实现又可以由硬件实现吗?

软件和硬件是两种完全不同的形态,硬件是实体,是物质基础;软件是一种信息,看不见、摸不到。但在逻辑功能上,软件和硬件是等效的。因此,在计算机系统中,许多功能既可以由硬件直接实现,又可以在硬件的配合下由软件实现。

例如,乘法运算既可用专门的乘法器(主要由加法器和移位器组成)实现,也可用乘法子程序(主要由加法指令和移位指令等组成)来实现。

6) 什么是透明性?透明是指什么都能看见吗?

在计算机领域中,站在某类用户的角度,若感觉不到某个事物或属性的存在,即“看“不到某个事物或属性,则称为“对该用户而言,某个事物或属性是透明的”。例如,对于高级语言程序员来说,浮点数格式、乘法指令等这些指令的格式、数据如何在运算器中运算等都是

透明的:而对于机器语音或汇编语言程序员来说,指令的格式、机器结构、数据格式等则不是透明的。

在 CPU 中, IR、MAR 和 MDR 对各类程序员都是透明的。

7) 机器字长、指令字长、存储字长的区别和联系?

机器字长:计算机能直接处理的二进制数据的位数,机器字长一般等于内部寄存器的大小,它决定了计算机的运算精度。

指令字长:一个指令字中包含的二进制代码的位数。

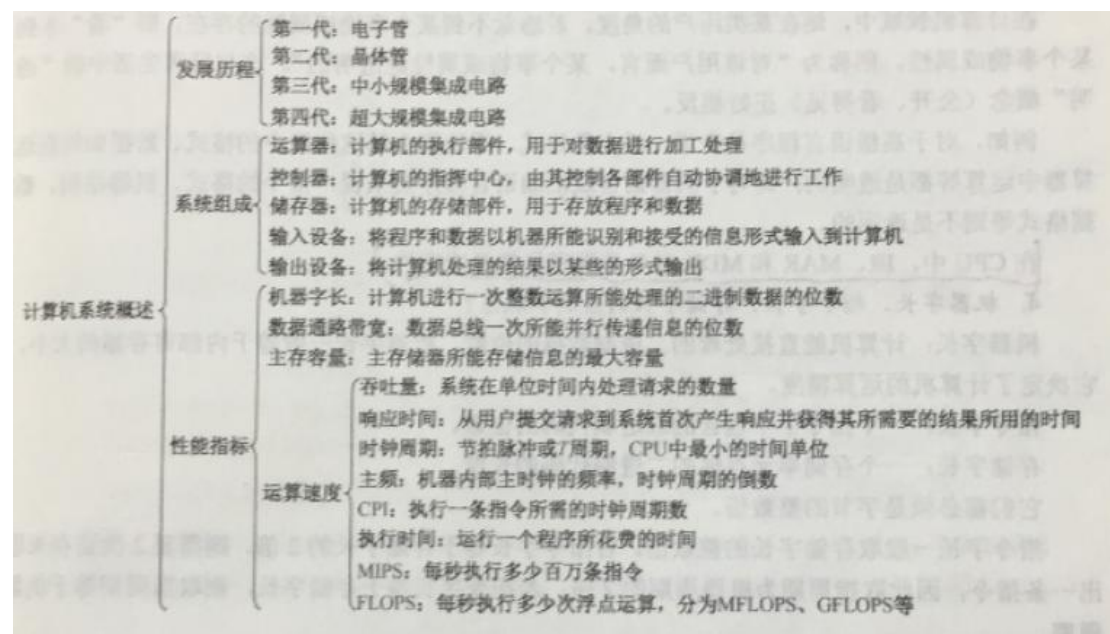
存储字长:一个存储单元存储的二进制代码的长度。

它们都必须是字节的整数倍。

指令字长一般取存储字长的整数倍,若指令字长等于存储字长的 2 倍,则需要 2 次访存来取出一条指令,因此取指周期为机器周期的 2 倍;若指令字长等于存储字长,则取指周期等于机器周期。

早期的计算机存储字长一般和机器的指令字长 与数据字长相等,故访问一次主存便可取出一条指令或一个数据。随着计算机的发展,指令字长可变,数据字长也可变,但它们必须都是字节的整数倍。

请注意 64 位操作系统是指特别为 64 位架构的计算机而设计的操作系统,它能够利用 64 位处理器的优势。但 64 位机器既可以使用 64 位操作系统,又可以使用 32 位操作系统。而 32 位处理器是无法使用 64 位操作系统的。



第二、三章

1) 在计算机中, 为什么要采用二进制来表示数据?

从可行性来说, 采用二进制, 只有 0 和 1 两个状态, 能够表示 0、1 两种状态的电子器件很多, 如开关的接通和断开、晶体管的导通和截止、磁元件的正负剩磁、电位电平的高与低等, 都可表示 0、1 两个数码。使用二进制, 电子器件具有实现的可行性。

从运算的简易性来说, 二进制数的运算法则少, 运算简单, 使计算机运算器的硬件结构大大简化(十进制的乘法九九口诀表有 55 条公式, 而二进制乘法只有 4 条规则)。

从逻辑上来说, 由于二进制 0 和 1 正好和逻辑代数的假(false) 和真(true) 相对应, 有逻辑代数的理论基础, 用二进制表示二值逻辑很自然。

2)计算机在字长足够的情况下能够精确地表示每个数吗?若不能,请举例。

计算机采用二进制来表示数据,在字长足够时,可以表示任何一个整数。而二进制表示小数时只能够用 1(2)的任意组合表示,即使字长很长,也不可能精确表示出所有小数,只能无限逼近。例如 0.1 就无法用二进制精确地表示。

3)字长相同的情况下,浮点数和定点数的表示范围与精度有什么区别?

字长相同时,浮点数取字长的一部分作为阶码,所以表示范围比定点数要大,而取一部分作为阶码也就代表着尾数部位的有效位数减少,而定点数字长的全部位都用来表示数值本身,精度要比同字长的浮点数更大。

4)用移码表示浮点数的阶码有什么好处?

移码的两个好处:

①浮点数进行加减运算时,时常要比较阶码的大小,相对于原码和补码,移码比较大小更方便。

②检验移码的特殊值(0 和 max)时比较容易。阶码以移码编码时的特殊值如下。0:表示指数为负无穷大,相当于分数分母无穷大,整个数无穷接近 0,在尾数也为 0 时可用来表示 0;尾数不为零表示未正规化的数。max:表示指数正无穷大,若尾数为 0,则表示浮点数超出表示范围(正负无穷大);尾数不为 0,则表示浮点数运算错误。

第四、七章

1)存储器的层次结构主要体现在何处?为何要分这些层次?计算机如何管理这些层次?

存储器的层次结构主要体现在 Cache-主存和主存辅存这两个存储层次上,Cathe 主存层次在存储系统中主要对 CPU 访存取起如速作用,即从整体运行的效果分析,CPU 访存速度加快,接近于 Cache 的速度,而寻址空间和位价却接近于主存,主存-辅存层次在存储系统中主要起扩容作用,即从程序员的角度看,他所使用的存储器的容量和位价接近于辅存,而速度接近于主存。

综合上述两个存储层次的作用,从整个存储系统来看,就达到了速度快、容量大、位价低的优化效果。

主存与 Cache 之间的信息调度功能全部由硬件自动完成。而主存与辅存层次的调度目前广泛采用虚拟存储技术实现,即将主存与辅存的一部分通过软/硬结合的技术组成虚拟存储器,程序员可用这个比主存实际空间(物理地址空间)大得多的虚拟地址空间(逻辑地址空间)编程,当程序运行时,再由软硬件自动配合完成虚拟地址空间与主存实际物理空间的转换。因此,这两个层次上的调度或转换操作对于程序员来说都是透明的。

2)存取周期和存取时间有何区别?

存取周期和存取时间的主要区别是:存取时间仅为完成一次操作的时间:而存取周期不仅包含操作时间,而且包含操作后线路的恢复时间,即存取周期=存取时间+恢复时间。

3)在虚拟存储器中,页面是设置得大一些好还是设置得小一些好?

页面不能设置得过大,也不能设置得过小。因为页面太小时,平均页内剩余空间较少,可节省存储空间,但会使得页表增大,而且页面太小时不能充分利用访存的空间局部性来提高命中率:页面太大时,可减少页表空间,但平均页内剩余空间较大,会浪费较多存储空间,页面太大还会使页面调入/调出时间较长。

4) Cache 行的大小和命中率之间有什么关系?

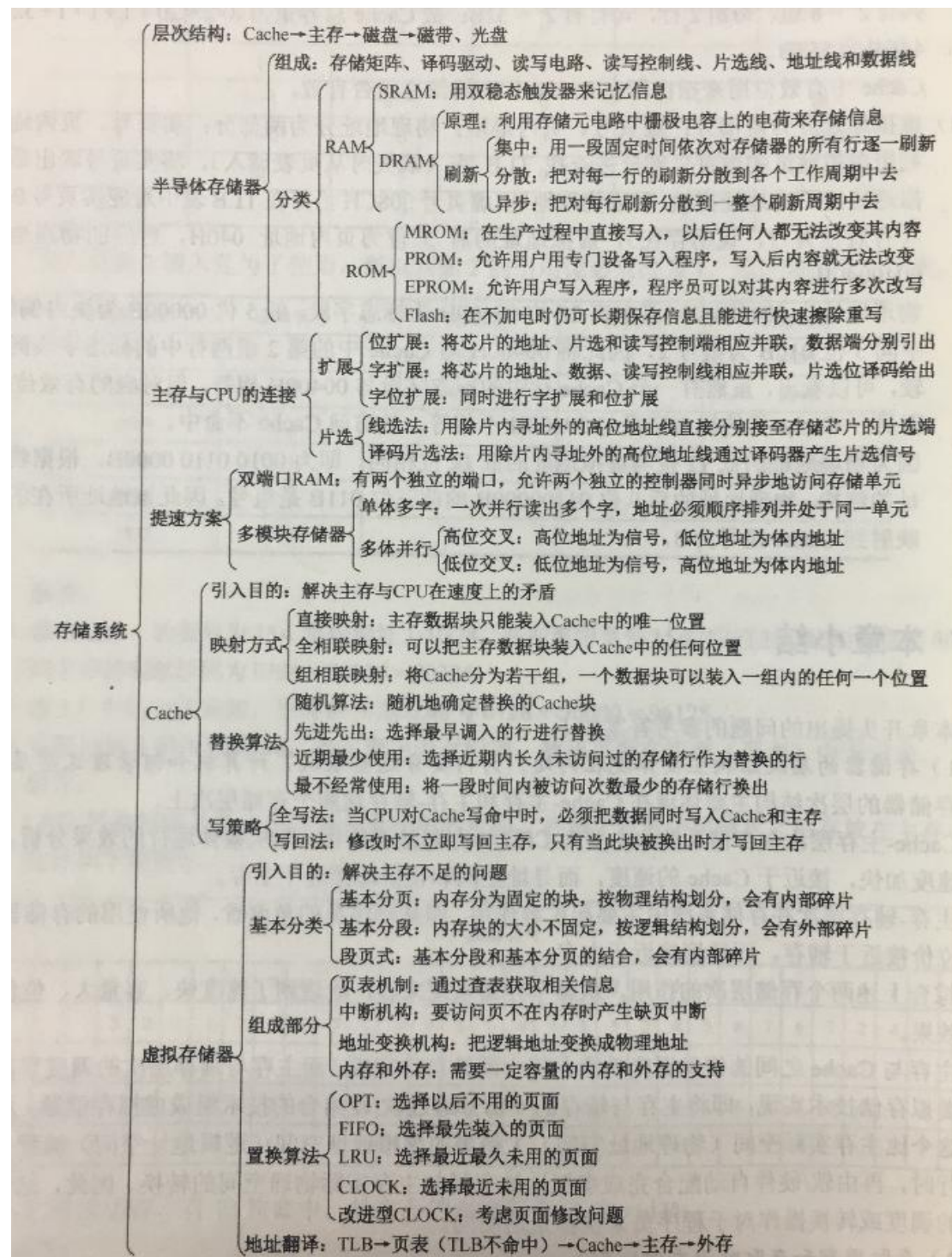
行的长度较大。可以充分利用程序访问的空间局部性,使一个较大的局部空间被一起调到 h 中,因而可以增加命中机会。但是,行长也不能太大,主要原因有两个:

a 行长大使失效损失变大。也就是说,若未命中,则需花更多时间从主存读块。

b 行长太大,Cache 项数变少,因而命中的可能性变小。

5) 发生取指令 Cache 缺失的处理过程是什么?

- 程序计数器恢复当前指令的值。
- 对主存进行读的操作。
- 将读入的指令写入 Cache 中，更改有效位和标记位。
- 重新执行当前指令。



第五章

1) 什么是指令?什么是指令系统?为什么要引入指令系统?

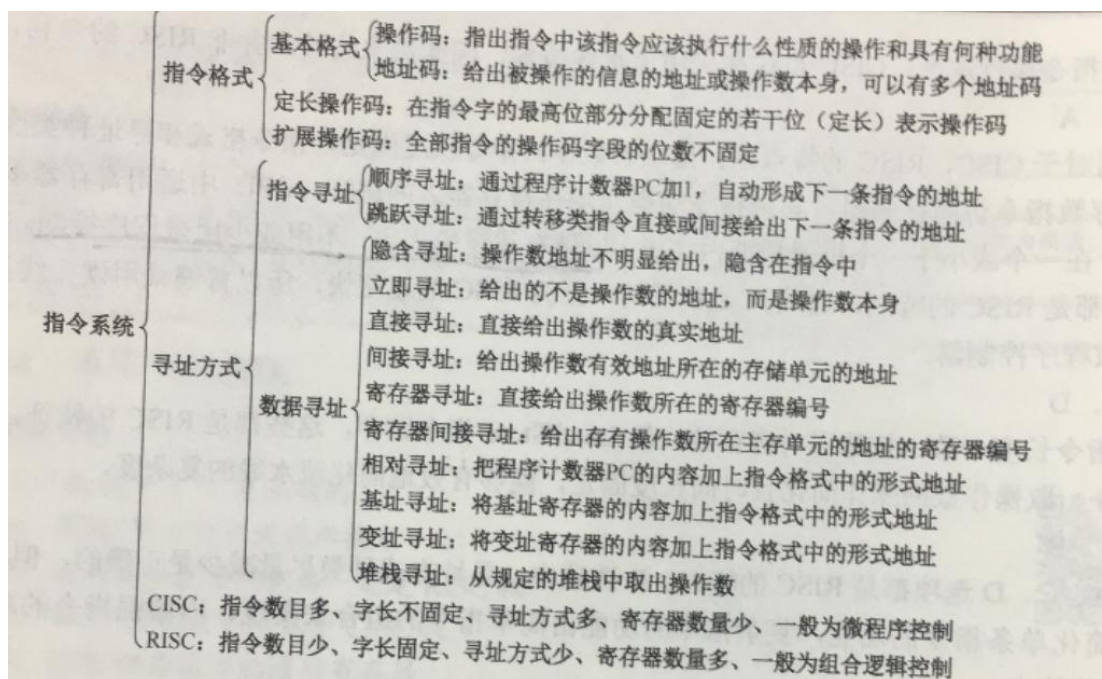
指令就是要计算机执行某种操作的命令,一台计算机中所有机器指令的集合,称为这台计算机的指令系统。引入指令系统后,避免了用户与二进制代码直接接触,使得用户编写程序更为方便。另外,指令系统是表征一台计算机性能的重要因素,它的格式与功能不仅直接影响到机器的硬件结构,而且也直接影响到系统软件,影响到机器的适用范围。

2) 一般来说,指令分为哪些部分?每部分有什么用处?

一条指令通常包括操作码字段和地址码字段两部分。其中,操作码指出指令中该指令应该执行什么性质的操作和具有何种功能,它是识别指令、了解指令功能与区分操作数地址内容的组成和使用方法等的关键信息。地址码用于给出被操作的信息(指令或数据)的地址,包括参加运算的一个或多个操作数所在的地址、运算结果的保存地址、程序的转移地址、被调用子程序的入口地址等。

3) 对于一个指令系统来说,寻址方式多和少有什么影响?

寻址方式的多样化能让用户编程更为方便,但多重寻址方式会造成 CPU 结构的复杂化,也不利于指令流水线的运行。而寻址方式太少虽然能够提高 CPU 的效率,但对于用户而言,少数几种寻址方式会使编程变得复杂,很难满足用户的需求。



第六章

1) CPU 分为哪几部分?分别实现什么功能?

CPU 分为运算器和控制器。其中运算器主要负责数据的加工,即对数据进行算术和逻辑运算,控制器是整个系统的指挥中枢,对整个计算机系统进行有效的控制,包括指令控制、操作控制、时间控制和中断处理。

2) 指令和数据均存放在内存中,计算机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,从内存读出的指令流流向控制器(指令寄存器),从内存读出的数据流流向运算器(通用寄存器)。

3) 什么是指令周期、机器周期和时钟周期?它们之间有何关系?

CPU 每取出并执行一条指令所需的全部时间称为指令周期；机器周期是在同步控制的机器中，执行指令周期中一步相对完整的操作(指令步)所需的时间，通常安排机器周期长度=主存周期；时钟周期是指计算机主时钟的周期时间，它是计算机运行时最基本的时序单位，对应完成一个微操作所需的时间，通常时钟周期=计算机主频的倒数。

4)指令周期是否有一个固定值?为什么?

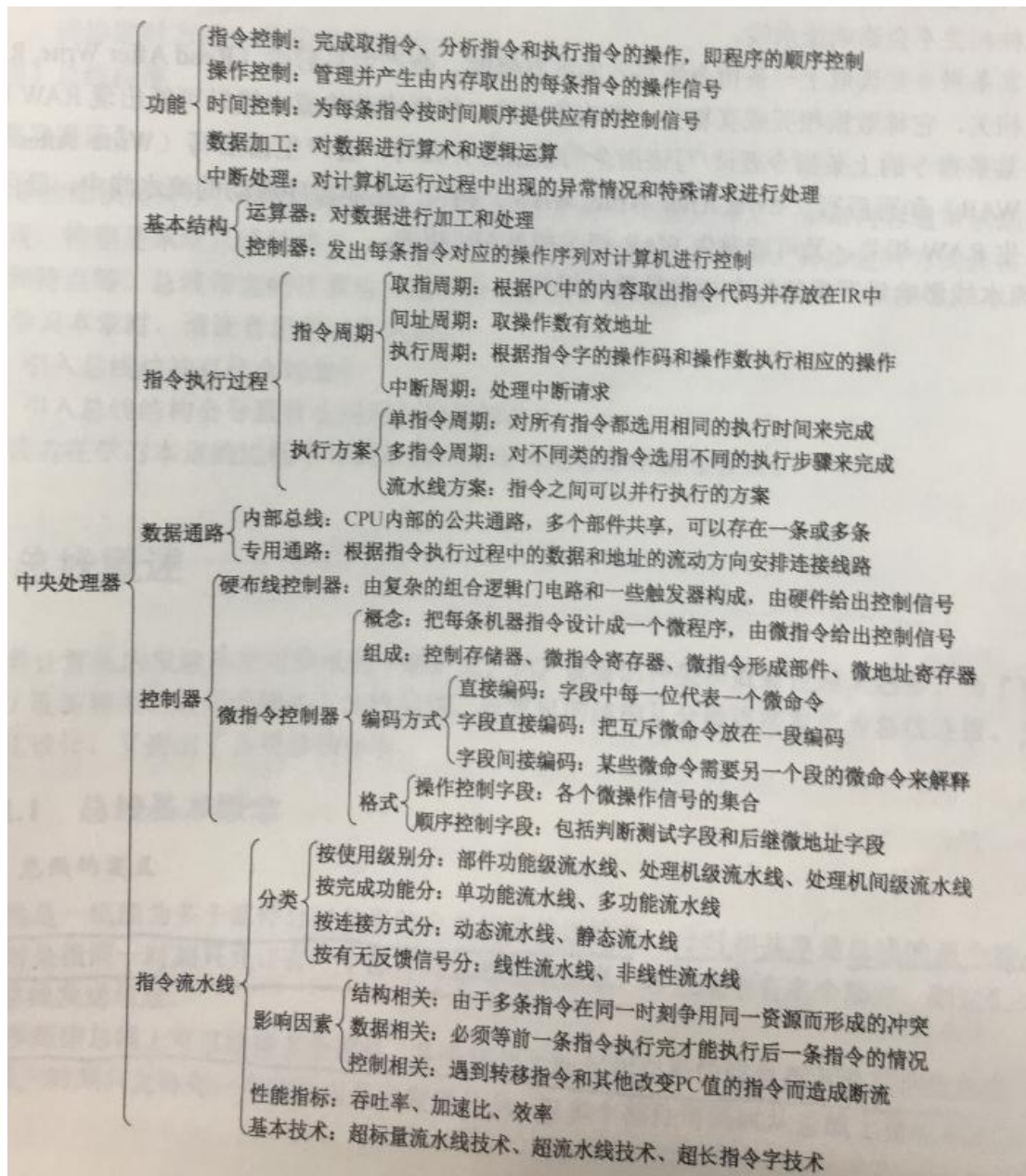
由于计算机中各种指令执行所需的时间差异很大，因此为了提高 **CPU** 的运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，即指令周期对不同的指令来说不是一个固定值。

5)什么是微指令?它和机器指令有什么关系?

控制部件通过控制线向执行部件发出各种控制命令，通常把这种控制命令称为微命令，而一组实现一定操作功能的微命令的组合，构成 1 条微指令，许多条微指令组成的序列构成微程序。微程序完成对指令的解释执行。指令，即指机器指令。每条指令可以完成一个独立的算术运算或逻辑运算操作。在采用微程序控制器的 **CPU** 中，一条指令对应一个微程序，一个微程序由许多微指令构成。一条微指令会发出很多不同的微命令。

6)什么是指令流水线? 指令流水线相对于传统计算机体系结构的优势是什么?

指令流水线是把指令分解为若干子过程，通过将每个子过程与其他子过程并行执行，来提高计算机的吞吐率的技术。采用流水线技术只需增加少量硬件就能把计算机的运算速度提高几倍，因此成为计算机中普遍使用的一种并行处理技术，通过在同一个时间段使用各功能部件，使得利用率明显提高。



综合题:

第三章

乘法运算

第四章

存储器的扩展计算及连接画图

第五章

指令扩展计算

不同寻址方式下, 地址的计算

第六章

数据通路, 控制信号的分析

第七章

地址映射关系, 通过虚拟地址计算物理地址