

考点脉络



计算机硬件系统是软件运行的基础，掌握一些基本硬件的工作原理是软件技术提高的前提，所以硬件基础知识是软件设计师考试中的一个必考模块。在此科目中，涉及到的知识点非常多，但真正正常考的却不多，本章将对考点脉络进行梳理，对重要知识点进行精讲，并辅以习题，以便考生加深印象。

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

（1）数据的表示：数制及其转换、原码、反码、补码、移码、浮点数、溢出、算术运算、逻辑运算、校验码。

（2）计算机系统的组成、体系结构分类及特性：CPU、存储器的组成、性能和基本工作原理、常用I/O设备、通信设备的性能及基本工作原理、I/O接口的功能、类型和特性、CISC/RISC、流水线操作、多处理机、并行处理。

（3）存储系统：虚拟存储器基本工作原理、多级存储体系、RAID类型和特性。

（4）可靠性与系统性能评测基础知识：诊断与容错、系统可靠性分析评价、校验方法、计算机系统性能评测方法。

从历年的考试情况来看，本章的考点主要集中在以下方面。

在数据的表示中，主要考浮点数运算、溢出、算术、逻辑运算。

在计算机系统的组成与体系结构中，主要考查CPU的构成，常见寄存器的作用、计算机体系结构分类、指令系统基础、CISC与RISC、流水线操作的相关内容。

在存储系统中，主要考查Cache存储器。

在可靠性与系统性能评测基础知识中，主要考查系统可靠性分析和校验方法。

版权方授权希赛网发布，侵权必究

本书简介

下一节

数据的表示

在数据的表示这个考点中，主要涉及到数制转换、数据编码、浮点数计算三个方面的内容，其中难度最高的是浮点数计算。

版权方授权希赛网发布，侵权必究

上一节

本书简介

下一节

1. 数制转换

(1) R进制数转换成十进制数

R进制数转换成十进制数通常使用按权展开法。具体操作方式为：将R进制数的每一位数值用 R^k 形式表示，即幂的底数是R，指数为k，k与该位和小数点之间的距离有关。当该位位于小数点左边，k值是该位和小数点之间数码的个数，而当该位位于小数点右边，k值是负值，其绝对值是该位和小数点之间数码的个数加1。

例如二进制数10101.01的值可计算如下：

$$10101.01 = 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-2}$$

按照上面的表示法，即可计算出R进制数十进制的值。

(2) 十进制数转换为R进制数

最常用的是“除以R取余法”。例如将十进制数85转换为二进制数：

$$\begin{array}{r} 2 \overline{)85} \text{余} 1 \\ 2 \overline{)42} 0 \\ 2 \overline{)21} \quad 1 \\ 2 \overline{)10} \\ 2 \overline{)5} \quad 1 \\ 2 \overline{)2} \quad 0 \\ 1 \quad 1 \end{array}$$

将所得的余数从低位到高位排列（1010101）₂就是85的二进制数。

(3) 二进制数与八进制数、十六进制数之间的转换

二进制转八进制：将每3个二进制数转换为八进制数；

二进制转十六进制数：将每4个二进制数转换为八进制数；

八进制转二进制：将每个八进制数转换为3位二进制数；

十六进制转二进制：将每个十六进制数转换为4位二进制数。

上面的转换都是以小数点作为计算数码个数的起点。八进制数和十六进制数转换可先转换为二进制数，然后再转换为目标进制。

2. 原码、反码、补码、移码

在计算机中，数据编码方式可以有多种，最为常见的有原码、反码、补码、移码。一个正数的原码、补码、反码是相同的，负数则不同。

(1) 原码

将最高位用做符号位（0表示正数，1表示负数），其余各位代表数值本身的绝对值的表示形式。这种方式是最容易理解的。

例如，+1的原码是0000 0001，-1的原码是1000 0001。

但是直接使用原码在计算时却会有麻烦，比如 $(1)_{10} + (-1)_{10} = 0$ ，如果直接使用原码则：

$$(0000\ 0001)_2 + (1000\ 0001)_2 = (1000\ 0010)_2$$

这样计算的结果是-2，也就是说，使用原码直接参与计算可能会出现错误的结果。所以，原码的符号位不能直接参与计算，必须和其它位分开，这样会增加硬件的开销和复杂性。

(2) 反码

正数的反码与原码相同。负数的反码符号位为1，其余各位为该数绝对值的原码按位取反。这个取反的过程使得这种编码称为“反码”。

例如，-1的反码：1111 1110。

同样对上面的加法，使用反码的结果是：

$$(0000\ 0001)_2 + (1111\ 1110)_2 = (1111\ 1111)_2$$

这样的结果是负0，而在人们普遍的观念中，0是不分正负的。反码的符号位可以直接参与计算，而且减法也可以转换为加法计算。

(3) 补码

正数的补码与原码相同。负数的补码是该数的反码加1，这个加1就是“补”。

例如，-1的补码：1111 1110+1 = 1111 1111。

再次做加法是这样的：

$$(0000\ 0001)_2 + (1111\ 1111)_2 = (0000\ 0000)_2$$

直接使用补码进行计算的结果是正确的。

对于一个补码表示的数，要计算其原码，只要对它再次求补，可得该数的原码。

由于补码能使符号位与有效值部分一起参加运算，从而简化运算规则，同时它也使减法运算转换为加法运算，进一步简化计算机中运算器的电路，这使得在大部分计算机系统中，数据都使用补码表示。

(3) 移码

移码是对补码的符号位取反得到的一种编码。移码只用于表示浮点数的阶码，所以只用于整数。

例如，-1的移码为：0111 1111。

3. 浮点数计算

在数学中，要表示一个很大的数时，我们常常使用一种称为科学计数法的方式：

$$N = M \cdot R^e$$

其中M称为尾数，e是指数，R为基数。

浮点数就是使用这种方法来表示大范围的数，其中指数一般是2，8，16。而且对于特定机器而言，指数是固定不变的，所以在浮点数中指数并不出现。从这个表达式可以看出：浮点数表示的精度取决于尾数的宽度，范围取决于基数的大小和指数的宽度。

浮点数的运算主要有三个步骤：对阶、尾数计数、结果格式化。

(1) 对阶

首先计算两个数的指数差，把指数小的向指数大的对齐，并将尾数右移指数差的位数，这样两个浮点数就完成了对阶的操作。可以看出，对阶的过程可能使得指数小的浮点数失去一些有效位。如果两个浮点数阶数相差很大，大于指数小的浮点数的尾数宽度，那么对阶后那个浮点数的尾数就变成了0，即当做机器零处理了。

(2) 尾数计算

对阶完成后，两个浮点数尾数就如同定点数，计算过程同定点数计算。

(3) 结果格式化

尾数计算后，可能会产生溢出，此时将尾数右移，同时指数加1，如果指数加1后发生了溢出，

则表示两个浮点数的运算发生了溢出。

如果尾数计算没有溢出，则尾数不断左移，同时指数减1，直到尾数为格式化数。如果这个过程中，指数小于机器能表达的最小值，则将结果置“机器零”，这种情况称为下溢。

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第1章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

一点一练

试题1

若某计算机采用8位整数补码表示数据，则运算__(1)__将产生溢出。

- (1) A. $-127+1$ B. $-127-1$ C. $127+1$ D. $127-1$

试题2

原码表示法和补码表示法是计算机中用于表示数据的两种编码方法，在计算机系统中常采用补码来表示和运算数据，原因是采用补码可以__(2)__。

- (2) A. 保证运算过程与手工运算方法保持一致
B. 简化计算机运算部件的设计
C. 提高数据的运算速度
D. 提高数据的运算精度

试题3

计算机中的浮点数由三部分组成：符号位S，指数部分E（称为阶码）和尾数部分M。在总长度固定的情况下，增加E的位数、减少M的位数可以__(3)__。

- (3) A. 扩大可表示的数的范围同时降低精度
B. 扩大可表示的数的范围同时提高精度
C. 减小可表示的数的范围同时降低精度
D. 减小可表示的数的范围同时提高精度

试题4

计算机中常采用原码、反码、补码和移码表示数据，其中， ± 0 编码相同的是__(4)__。

- (4) A. 原码和补码 B. 反码和补码
C. 补码和移码 D. 原码和移码

试题5

计算机在进行浮点数的相加（减）运算之前先进行对阶操作，若x的阶码大于y的阶码，则应将__(5)__。

- (5) A. x的阶码缩小至与y的阶码相同，且使x的尾数部分进行算术左移
B. x的阶码缩小至与y的阶码相同，且使x的尾数部分进行算术右移
C. y的阶码扩大至与x的阶码相同，且使y的尾数部分进行算术左移
D. y的阶码扩大至与x的阶码相同，且使y的尾数部分进行算术右移

解析与答案

试题1分析

采用8位整数补码表示数据，数据的表示范围是-128至127，因此在各选择中运算会发生溢出的是C。

试题1答案

(1) C

试题2分析

在计算机中，各类运算都等可以采用补码进行，特别是对于有符号数的运算。在计算机中设计补码的目的一是为了使符号位能与有效值部分一起参加运算，从而简化运算规则，使运算部件的设计更简单；二是为了使减法运算转换为加法运算，进一步简化计算机中运算器的线路设计。因此在计算机系统中常采用补码来表示和运算数据，原因是采用补码可以简化计算机运算部件的设计。

试题2答案

(2) B

试题3分析

浮点数的表示形式如下：

$$N=M\times r^E$$

其中r是浮点数阶码的底，与尾数的基数相同，通常r=2。E和M都是带符号的定点数，E叫做阶码，M叫做尾数。浮点数的一般格式如图所示，浮点数的底是隐含的，在整个机器数中不出现。阶码的符号位为Es，阶码的大小反映了在数N中小数点的实际位置；尾数的符号位为Ms，它也是整个浮点数的符号位，表示了该浮点数的正、负。

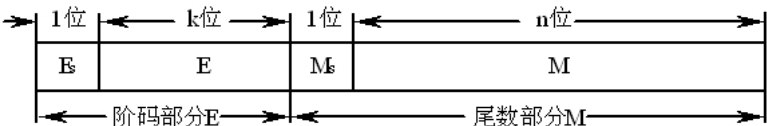


图1-1浮点数的一般格式

浮点数的大小由阶码部分决定，而其精度由尾数部分决定，因此增加E的位数、减少M的位数可以扩大可表示的数的范围同时降低精度。

试题3答案

A

试题4分析

本题考查计算机系统数据编码基础知识。

设机器字长为n（即采用n个二进制位表示数据），最高位是符号位，0表示正号，1表示负号。

原码表示方式下，除符号位外，n-1位表示数值的绝对值。因此，n为8时， $[+0]_{原}=0$

0000000， $[-0]_{原}=1\ 0000000$ 。

正数的反码与原码相同，负数的反码则是其绝对值按位求反。n为8时，数值0的反码表示有两种形式： $[+0]_{\text{反}}=0\ 0000000$ ， $[-0]_{\text{反}}=11111111$ 。

正数的补码与其原码和反码相同，负数的补码则等于其反码的末尾加1。在补码表示中，0有唯一的编码： $[+0]_{\text{补}}=0\ 0000000$ ， $[-0]_{\text{补}}=00000000$ 。

移码表示法是在数X上增加一个偏移量来定义的，常用于表示浮点数中的阶码。机器字长为n时，在偏移量为 2^{n-1} 的情况下，只要将补码的符号位取反便可获得相应的移码表示。

试题4答案

(4) C

试题5分析

本题考查浮点数运算的基本过程。该过程分为如下3步：

第一步：对阶，规则是向大阶看齐。具体方法是：阶码小的那个尾数右移，移位的次数等于参加运算的两个数的阶码之差。

第二步：尾数相加。实际上进行的是定点数加法。

第三步：结果的后处理。包括如下3个方面：

(1) 如果运算结果的尾数部分溢出，则需要向右规格化一位。具体方法是：尾数部分右移一位，阶码加1。

(2) 如果运算结果的最高位为0，则需要向左规格化，并且要重复进行，直到尾数的最高位不为0。尾数部分每左移一位，阶码必须减1。

(3) 进行舍入处理。

所以本题的正确答案为D。

试题5答案

(5) D

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

计算机系统的组成与体系结构

在计算机系统的组成与体系结构中，计算机体系结构分类、计算机的硬件组成、指令系统基础、CISC与RISC、流水线操作等内容是最为重要的，下面将详细介绍这几个方面的知识。

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

考点精讲

1. 计算机体系结构分类

计算机体系结构分类有多种方式，其中最为常见的是：Flynn分类法与冯氏分类法。而考试中主要考查的是Flynn分类法。

Flynn分类法是根据指令流、数据流和多倍性三方面来进行分类的，如表1-1所示。

表1-1 Flynn分类法

体系结构类型	结构	关键特性	代表
单指令流单数据流 SISD	控制部分：一个 处理器：一个 主存模块：一个		单处理器系统
单指令流多数据流 SIMD	控制部分：一个 处理器：多个 主存模块：多个	各处理器以异步的形式 执行同一条指令	并行处理机 阵列处理机 超级向量处理机
多指令流单数据流 MISD	控制部分：多个 处理器：多个 主存模块：多个	被证明不可能，至少是 不实际	理论模型，未见成形产品
多指令流多数据流 MIMD	控制部分：多个 处理器：多个 主存模块：多个	能够实现作业、任务、 指令等各级全面并行	多处理机系统 多计算机

2. 计算机的硬件组成

计算机硬件系统是依照冯·诺依曼所设计体系结构，即包括运算器、控制器、存储器、输入设备和输出设备五大部件组成，如图1-2所示。

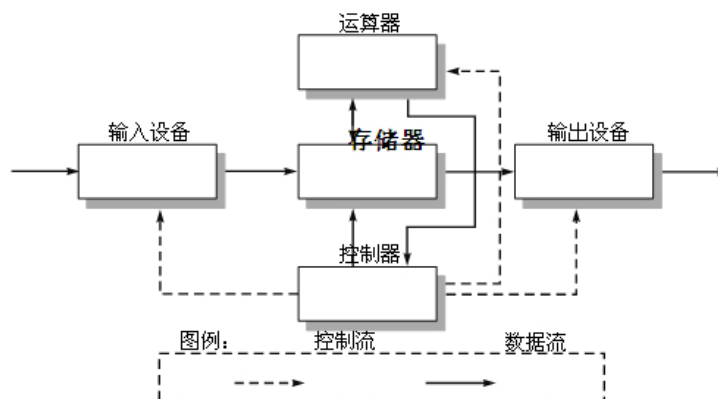


图1-2 计算机硬件组成结构示意图

其中运算器和控制器组成中央处理器（CPU）。运算器负责完成算术、逻辑运算功能，通常由ALU（算术/逻辑单元）、寄存器、多路转换器、数据总线组成；控制器则负责依次访问程序指令，进行指令译码，并协调其他设备，通常由程序计数器（PC）、指令寄存器、指令译码器、状态/条件寄存器、时序发生器、微操作信号发生器组成。

下面介绍几个主要部件的功能：

程序计数器：用于存放下一条指令所在单元的地址的地方。由于大多数指令都是按顺序来执行的，所以修改的过程通常只是简单的对PC加1。当遇到转移指令时，后继指令的地址（即PC的内容）必须从指令寄存器中的地址字段取得。在这种情况下，下一条从内存取出的指令将由转移指令来规定，而不像通常一样按顺序来取得。因此程序计数器的结构是具有寄存信息和计数两种功能的结构。

指令寄存器：用来保存当前正在执行的一条指令。当执行一条指令时，先把它从内存取到数据寄存器中，然后再传送至指令寄存器。指令划分为操作码和地址码字段，由二进制数字组成。为了执行任何给定的指令，必须对操作码进行测试，以便识别所要求的操作。指令译码器就是做这项工

作的（下一词条将详细介绍其工作过程）。指令寄存器中操作码字段的输出就是指令译码器的输入。操作码一经译码后，即可向操作控制器发出具体操作的特定信号。

指令译码器：译码是编码的逆过程，在编码时，每一种二进制代码，都赋予了特定的含义，即都表示了一个确定的信号或者对象。把代码状态的特定含义“翻译”出来的过程叫做译码，实现译码操作的电路称为译码器。或者说，译码器是可以将输入二进制代码的状态翻译成输出信号，以表示其原来含义的电路。

3. 指令系统基础

在计算机中，CPU都会定义出自己特定的指令系统，不过都遵循着统一的标准格式。指令的基本格式是由操作码和地址码两个部分组成的。操作码指出该指令要完成什么操作，地址码则是提供原始的数据。指令系统中定义操作码的方式可以分为规整型（定长编码）和非规整型（变长编码）两种，如表1-2所示。

表1-2 指令系统操作码定义分类方法比较表

编码方式	编码方式	平均码长
定长编码	采用相等码长，每个操作码的长度相等	码长 $\geq \log_2(\text{操作码数})$ 如 14 个操作码，就应该是 4 位： $2^3=8$ ，不够， $2^4=16$ ，多 2 个
变长编码	根据使用频度选择不同长度的编码	将码分为几类，然后再对每类进行编码 平均码长为：将每个码长乘以频度，再累加其和

而在指令系统中用来确定如何提供操作数或提供操作数地址的方式称为寻址方式和编址方式。操作数可以存放在CPU中的寄存器（用寄存器名操作）、主存储器（指出存储单元地址）、堆栈（先进后出的存储机制，用栈顶指针SP来标出其当前位置）、外存储器或外围设备中。不过在运算时，数据均在主存储器中，操作数可以采用以下几种寻址方式：

（1）立即寻址方式

指令的地址字段指出的不是操作数的地址，而是操作数本身。这种方式的特点是指令执行时间很短，不需要访问内存取数。题目中所说的“操作数包含在指令中的寻址方式”就是立即寻址。

例如，单地址的移位指令格式为

OP（移位）	F	D
--------	---	---

这里D不是地址，而是一个操作数。F为标志位，当F=1时，操作数进行右移；当F=0时，操作数进行左移。

（2）直接寻址方式

直接寻址特点是：在指令格式的地址字段中直接指出操作数在内存的地址D。

采用直接寻址方式时，指令字中的形式地址D就是操作数的有效地址E，即E=D。因此通常把形式地址D又称为直接地址。此时，由寻址模式给予指示。如果用S表示操作数，那么直接寻址的逻辑表达式为 $S=(E)=(D)$

（3）间接寻址方式

间接寻址的情况下，指令地址字段中的形式地址D不是操作数的真正地址，而是操作数地址的指示器，D单元的内容才是操作数的有效地址。

如果把直接寻址和间接寻址结合起来，指令有如下形式：

操作码	I	D
-----	---	---

寻址特征位I=0，表示直接寻址，这时有效地址E=D；I=1，表示间接寻址，这时有效地址E=

(D)。

间接寻址方式是早期计算机中经常采用的方式，但由于两次访问内存，影响指令执行速度，现在已不大使用。

(4) 寄存器寻址方式和寄存器间接寻址方式

当操作数不放在内存中，而是放在CPU的通用寄存器中时，可采用寄存器寻址方式。此时指令中给出的操作数地址不是内存的地址单元号，而是通用寄存器的编号。这也就是题目中所说的“操作数在寄存器中的寻址方式”。

寄存器间接寻址方式与寄存器寻址方式的区别在于：指令格式中的寄存器内容不是操作数，而是操作数的地址，该地址指明的操作数在内存中。这也就是题目中所说的“操作数的地址在寄存器中的寻址方式”。

(5) 相对寻址方式

相对寻址是把程序计数器PC的内容加上指令格式中的形式地址D而形成操作数的有效地址。程序计数器的内容就是当前指令的地址。“相对”寻址，就是相对于当前的指令地址而言的。

采用相对寻址方式的好处是程序员无须用指令的绝对地址编程，所编程序可以放在内存任何地方。

此时形式地址D通常称为偏移量，其值可正可负，相对于当前指令地址进行浮动。

4. CISC与RISC

为了提高操作系统的效率，人们最初选择了向指令系统中添加更多、更复杂的指令，而随着不断地升级和向后兼容的需要，指令集也越来越大。这种类型的计算机，我们称之为复杂指令计算机CISC。而后来研究发现，计算机指令系统如果使用少量结构简单的指令会提高计算机的性能，这就是精简指令集计算机RISC。计算机执行程序所需的时间P由三方面因素决定：编译后产生的机器指令数I、执行每条指令所需的平均周期数CPI，以及每个机器周期的时间T。它们的关系是 $P=I \times CPI \times T$ 。RISC正是通过简化指令的途径使计算机结构更合理，减少指令执行周期数，提高运算速度。虽然RISC编译后产生的机器指令数(I)增多了，但指令所需的周期数(CPI)和每个周期的时间(T)都可以减少。它与CISC可谓各有特色，如表1-3所示。

表1-3 指令系统操作码定义分类方法比较表

指令系统类型	指令	寻址方式	实现方式	其它
CISC (复杂)	数量多，使用频率差别大，可 变长格式	支持多种	微程序控制技术	
RISC (精简)	数量少，使用频率接近，定长 格式，大部分为单周期指令， 操作寄存器，只有 Load/Store 操作内存	支持方式少	增加了通用寄存器；硬 布线逻辑控制为主；采 用流水线	优化编译，有效 支持高级语言

典型的RISC处理器包括：DEC的Alpha 21164、IBM的Power PC620、HP的PA-8000、SGI MIPS分部的TS、Sun的Ultra SPARC。目前RISC处理器技术的发展方向是采用并行处理技术（包括超级流水线、超级标量、超长指令字）大幅度提高运算速度。

5. 流水线

流水线是指在程序执行时多条指令重叠进行操作的一种准并行处理实现技术。各种部件同时处理是针对不同指令而言的，它们可同时为多条指令的不同部分进行工作，以提高各部件的利用率和指令的平均执行速度。

指令流水线是将指令执行分成几个子过程，每一个子过程对应一个工位，我们称为流水级或流水节拍，这个工位在计算机里就是可以重叠工作的功能部件，称为流水部件。

如图1-3所示，IF，ID，EX，WD分别是流水线的流水部件。

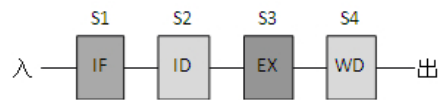


图1-3几个部件组成的流水线

流水线要求所有的流水级部件必须在相同的时间内完成各自的子过程。在流水线中，指令流动一步便是一个机器周期，机器周期的长度必须由最慢的流水级部件处理子过程所需的时间来决定。

那么我们为什么要提出流水线这个概念，以及流水线是如何提高系统吞吐量的呢？下面我们来看几个图，概念自然就清楚了。

图1-4是一个非流水线结构系统执行指令时空图。

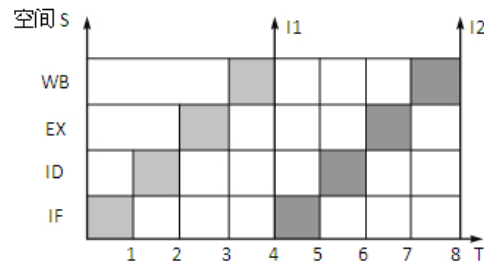


图1-4非流水线结构系统执行指令时空图

我们从图1-4中可以看到，任意一个系统时间都有大量的设备处于空闲状态，如第一个时间段有ID，EX，WB空闲，则第二个时间段有IF，EX，WB空闲。

我们再来看采用了流水线结构的时空图1-5。

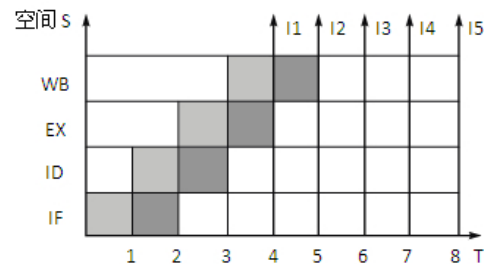


图1-5流水线结构指令时空图

显然，采用流水线可以大大提升系统资源的利用率，以及整个系统的吞吐量。

在考试中，流水线方面的考题主要考查两个知识点：计算流水线执行时间和分析影响流水线的因素。

(1) 计算流水线执行时间

假定有某种类型的任务，共可分成N个子任务，执行每个子任务需要时间t，则完成该任务所需的时间即为Nt。若以传统的方式，则完成k个任务所需的时间是kNt；而使用流水线技术执行，花费的时间是Nt+(k-1)t。也就是说，除了第一个任务需要完整的时间外，其他都通过并行，节省了大量的时间，只需一个子任务的单位时间就够了。

另外要注意的是，如果每个子任务所需的时间不同，则其速度取决于其执行顺序中最慢的那个（也就是流水线周期值等于最慢的那个指令周期），要根据实际情况进行调整。

例如：若指令流水线把一条指令分为取指、分析和执行三部分，且三部分的时间分别是取指2ns，分析2ns，执行1ns。那么，最长的是2ns，因此100条指令全部执行完毕需要的时间就是：
 $(2ns+2ns+1ns) + (100-1) \times 2ns = 203ns$ 。

另外，还应该掌握几个关键的术语：流水线的吞吐量（等于任务数/完成时间），加速比（不采用流水线的执行时间/采用流水线的执行时间）。

（2）影响流水性的主要因素

如图1-5所示，流水线的关键在于“重叠执行”，因此如果这个条件不能够满足，流水线就会被破坏。这种破坏主要来自两种情况：

转移指令：因为前面的转移指令还没有完成，流水线无法确定下一条指令的地址，因此也就无法向流水线中添加这条指令。从这里的分析可以看出，无条件跳转指令是不会影响流水线的。

共享资源访问的冲突：也就是后一条指令需要使用的数据，与前一条指令发生的冲突，或者相邻的指令使用了相同的寄存器，这也会使得流水线失败。

响应中断：当有中断请求时，流水线也会停止。对于这种情况有两种响应方式，一种是立即停止——精确断点法，能够立即响应中断；另一种是流水线中的指令继续执行，不再新增指令到流水线——不精确断点法。

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第1章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

一点一练

试题1

若某条无条件转移汇编指令采用直接寻址，则该指令的功能是将指令中的地址码送入__(1)__。

- (1) A．PC（程序计数器） B．AR（地址寄存器）
C．AC（累加器） D．ALU（算逻运算单元）

试题2

CPU中译码器的主要作用是进行__(2)__。

- (2) A．地址译码 B．指令译码 C．数据译码 D．选择多路数据至ALU

试题3

编写汇编语言程序时，下列寄存器中程序员可访问的是__(3)__。

- (3) A．程序计数器（PC） B．指令寄存器（IR）
C．存储器数据寄存器（MDR） D．存储器地址寄存器（MAR）

试题4

在CPU中用于跟踪指令地址的寄存器是__(4)__。

- (4) A．地址寄存器（MAR） B．数据寄存器（MDR）
C．程序计数器（PC） D．指令寄存器（IR）

试题5

若CPU要执行的指令为：MOV R1，#45（即将数值45传送到寄存器R1中），则该指令中采用的寻址方式为__(4)__。

- (4) A．直接寻址和立即寻址 B．寄存器寻址和立即寻址
C．相对寻址和直接寻址 D．寄存器间接寻址和直接寻址

试题6

在CPU的寄存器中，__(6)__对用户是完全透明的。

(6) A. 程序计数器 B. 指令寄存器 C. 状态寄存器 D. 通用寄存器

试题7

指令系统中采用不同寻址方式的目的是__(7)___。

(7) A. 提高从内存获取数据的速度 B. 提高从外存获取数据的速度
C. 降低操作码的译码难度 D. 扩大寻址空间并提高编程灵活性

试题8

总线复用方式可以__(8)___。

(8) A. 提高总线的传输带宽 B. 增加总线的功能
C. 减少总线中信号线的数量 D. 提高CPU利用率

试题9

一条指令的执行过程可以分解为取指、分析和执行三步，在取指时间 $t_{取指}=3\Delta t$ 、分析时间 $t_{分析}=2\Delta t$ 、执行时间 $t_{执行}=4\Delta t$ 的情况下，若按串行方式执行，则10条指令全部执行完需要__(9)___ Δt ；若按流水线的方式执行，则10条指令全部执行完需要__(10)___ Δt 。

(9) A. 40 B. 70 C. 90 D. 100
(10) A. 20 B. 30 C. 40 D. 45

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

第 1 章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

解析与答案

试题1分析

本题主要考查寄存器的相关内容。

程序计数器是用于存放下一条指令所在单元的地址的地方。在程序执行前，必须将程序的起始地址，即程序的一条指令所在的内存单元地址送入程序计数器，当执行指令时，CPU将自动修改程序计数器的内容，即每执行一条指令程序计数器增加一个量，使其指向下一个待指向的指令。程序的转移等操作也是通过该寄存器来实现的。

地址寄存器一般用来保存当前CPU所访问的内存单元的地址，以方便对内存的读写操作。

累加器是专门存放算术或逻辑运算的一个操作数和运算结果的寄存器。

ALU是CPU的执行单元，主要负责运算工作。

试题1答案

(1) A

试题2分析

在计算机中，通常都是二进制代码，如果我们要将一个信息放到计算机中去表述，就都需要将其编码为二进制代码，在编码时，每一种二进制代码，都赋予了特定的含义，即都表示了一个确定的信号或者对象。而译码就是编码的逆过程。

CPU中的译码器的主要作用是对指令进行译码。

试题2答案

(2) B

试题3分析

程序计数器是用于存放下一条指令所在单元的地址的地方。在程序执行前，必须将程序的起始地址，即程序的一条指令所在的内存单元地址送入程序计数器，当执行指令时，CPU将自动修改程序计数器的内容，即每执行一条指令程序计数器增加一个量，使其指向下一个待指向的指令。程序的转移等操作也是通过该寄存器来实现的。

指令寄存器一般用来保存当前正在执行的一条指令。

存储器数据寄存器主要是用来保存操作数和运算结果等信息的，其目的是为了节省读取操作数所需占用总线和访问存储器的时间。

存储器地址寄存器一般用来保存当前CPU所访问的内存单元的地址，以方便对内存的读写操作。

作为程序员，应该要能控制其所编写程序的执行过程，这就需要利用程序计数器来实现，因此程序员能访问的是程序计数器。

试题3答案

(3) A

试题4分析

请参看试题3分析。

试题4答案

(4) C

试题5分析

本题主要考查各种寻址方式。

立即寻址的特点是：指令的地址字段指出的不是操作数的地址，而是操作数本身；

直接寻址特点是：在指令格式的地址字段中直接指出操作数在内存的地址；

寄存器寻址的特点是：指令中给出的操作数地址不是内存的地址单元号，而是通用寄存器的编号（当操作数不放在内存中，而是放在CPU的通用寄存器中时，可采用寄存器寻址方式）；

寄存器间接寻址方式与寄存器寻址方式的区别在于：指令格式中操作数地址所指向的寄存器中存放的内容不是操作数，而是操作数的地址，通过该地址可在内存中找到操作数；

相对寻址的特点是：把程序计数器PC的内容加上指令格式中的形式地址来形成操作数的有效地址。

在本题中，指令中的两个操作数，分别使用的是寄存器寻址和立即寻址，因为在这个指令中，其第一个操作数字段是一个寄存器编号，而第二个操作数字段就是操作数本身。

试题5答案

(5) B

试题6分析

指令寄存器用来存放当前正在执行的指令，对用户是完全透明的。

状态寄存器用来存放计算结果的标志信息，如进位标志、溢出标志等。

通用寄存器可用于传送和暂存数据，也可参与算术逻辑运算，并保存运算结果。

试题6答案

(6) B

试题7分析

寻址是指寻找操作数的地址或下一条将要执行的指令地址。数据和指令一般都需要存放在一些存储器的存储单元中，存储器对这些存储单元进行编号，这些编号就是数据和指令的地址，如果在应用中需要用到这些数据或指令时，就通过它们的地址到存储器中去寻找，这就是寻址。

假如某主机的主存容量可达1MB，而指令中的地址码字段最长仅16位，只能直接访问主存的一小部分，而无法访问到整个主存空间，而采用不同的寻址方式可以实现对整个主存空间的访问。就是在字长很长的大型机中（地址码字段足够长），即使指令中能够拿出足够的位数来作为访问整个主存空间的地址，但为了灵活、方便地编制程序，也需要对地址进行必要的变换。

综上所述，可知本题答案选D。

试题7答案

(7) D

试题8分析

总线复用，顾名思义就是一条总线实现多种功能。常见的总线复用方式有总线分时复用，它是指在不同时段利用总线上同一个信号线传送不同信号，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。

试题8答案

(8) C

试题9分析

串行执行时，每条指令都需三步才能执行完，没有重叠。总的执行时间为： $(3+2+4) \times 10 = 90$ 。

按流水线方式执行，系统在同一时刻可以进行第k条指令的取指，第k+1条指令的分析，第k+2条指令的执行，所以效率大大提高了。我们平时用的流水线计算公式是：第一条指令顺序执行时间+（指令条数-1）×周期，而周期与三个步骤时间最长的保持一致，因此本题的计算结果为： $9 + (10-1) \times 4 = 45$ 。

试题9答案

(9) C (10) D

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

考点精讲

在整个计算机系统中，存储系统的地位非常重要，在本书中，有多个章节涉及存储相关内容。本章主要描述存储系统中与硬件相关的部分，而在操作系统章节将描述存储调度相关内容。

1. Cache

由于在CPU与存储系统间存在着数据传送带宽的限制，因此在其中设置了Cache（高速缓冲存储器，通常速度比内存快），以提高整体效率。但由于其成本更高，因此Cache的容量要比内存小得多。Cache是一种相联存储器（即按内容进行存储的存储器）。

(1) Cache原理、命中率、失效率

使用Cache改善系统性能的主要依据是程序的局部性原理。通俗地说，就是一段时间内，执行的语句常集中于某个局部。而Cache正式将访问集中的内容放在速度更快的Cache上，以提高性能。引入Cache后，CPU在需要数据时，先找Cache，如果没有再找内存。

如果Cache的访问命中率为 h （通常 $1-h$ 就是Cache的失效率），而Cache的访问周期时间是 t_1 ，主存储器的访问周期时间是 t_2 ，则整个系统的平均访存时间就应该是：

$$t_3 = h \times t_1 + (1-h) \times t_2$$

从公式可以看出，系统的平均访存时间与命中率有着很密切的关系。灵活地应用这个公式，可以计算出所有情况下的平均访存时间。

例如：设某流水线计算机主存的读/写时间为100ns，有一个指令和数据合一的Cache，已知该Cache的读/写时间为10ns，取指令的命中率为98%，取数的命中率为95%。在执行某类程序时，约有1/5指令需要存/取一个操作数。假设指令流水线在任何时候都不阻塞，则设置Cache后，每条指令的平均访存时间约为多少？其实这是应用该公式的简单数学题：

$$(2\% \times 100\text{ns} + 98\% \times 10\text{ns}) + 1/5 \times (5\% \times 100\text{ns} + 95\% \times 10\text{ns}) = 14.7\text{ns}$$

(2) Cache存储器的映射机制

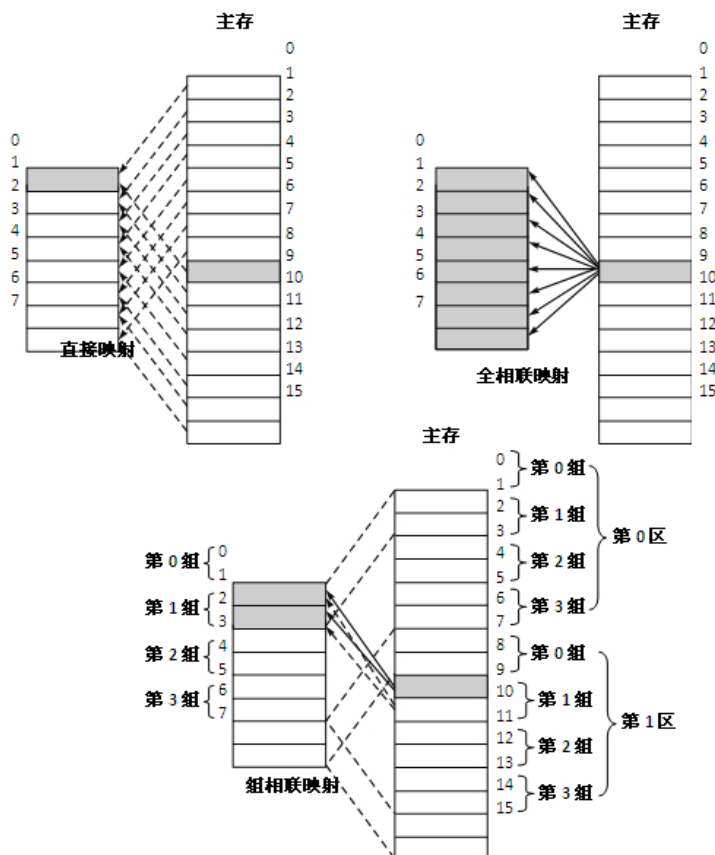


图1-6 Cache映射规则图解

CPU发生访存请求时，会先让Cache判断是否包括，如果命中（即包括请求的内容）就直接使用。这个判断的过程就是Cache地址映射，这个速度应该尽可能快，常见的映射方法有直接映射、全相联映射和组相联映射三种，其原理如图1-6所示。

直接映射：是一种多对一的映射关系，但一个主存块只能够复制到Cache的一个特定位置上去。Cache的行号 i 和主存的块号 j 有函数关系： $i = j \% m$ （其中 m 为Cache总行数）。例如：某Cache容量为16KB（可用14位表示），每行的大小为16B（可用4位表示），则说明其可分为1024行（可用10位表示）。则主存地址的最低四位为Cache的行内地址，中间10位为Cache行号。如果内存地

址为1234E8F8H的话，那么最后四位就是1000（对应16进制数的最后一位），而中间10位，则应从E8F（111010001111）中获取，得到1010001111。

相联映射：将主存中一个块的地址与块的内容一起存于Cache的行中。速度更快，但控制复杂。

组相联映射：是前两种方式的折中方案。它将Cache中的块再分成组。然后通过直接映射方式决定组号，再通过相联映射的方式决定Cache中的块号。

要注意的是，在Cache映射中，主存和Cache存储器将均分成容量相同的块。

例如：容量为64块的Cache采用组相联方式映像，字块大小为128个字，每4块为一组。若主存容量为4096块，且以字编址，那么主存地址应该为多少位？主存区号为多少位？这样的题目，首先根据主存与Cache块的容量需一致，因此内存也是128个字，因此共有 128×4096 个字，即 $2^{19}(2^7 + 2^{12})$ 个字，因此主存地址需要19位；而内存所需要分为 $4096/64$ 块，即 2^6 ，因此主存区号需要6位。

（3）Cache淘汰算法

当Cache数据已满，并且出现未命中情况时，就是淘汰一些老的数据，更新一些新的数据。而选择淘汰什么数据的方法就是淘汰算法，常见的方法有三种：随机淘汰、先进先出（FIFO）淘汰（淘汰最早调入Cache的数据）、最近最少使用（LRU）淘汰法。其中平均命中率最高的是LRU算法。

（4）Cache存储器的写操作

在使用Cache时，需要保证其数据与主存一致，因此在写Cache时就需要考虑与主存间的同步问题，通常使用以下三种方法：写直达（写Cache时，同时写主存）、写回（写Cache时不马上写主存，而是等其淘汰时回写）、标记法。

2. 主存（内存）

（1）主存储器的种类

RAM：随机存储器，可读写，断电后数据无法保存，只能暂存数据。

SRAM：静态随机存储器，在不断电时信息能够一直保持。

DRAM：动态随机存储器，需要定时刷新以维持信息不丢失。

ROM：只读存储器，出厂前用掩膜技术写入，常用于存放BIOS和微程序控制。

PROM：可编程ROM，只能一次写入，需用特殊电子设备进行写入。

EPROM：可擦除的PROM，用紫外线照射15~20分钟可擦去所有信息，可写入多次。

E²PROM：电可擦除ERPOM，可以写入，但速度慢。

闪速存储器：现在U盘使用的种类，可以快速写入。

记忆时，抓住几个关键英文字母。A，即Access，说明读写都行；O，即Only，说明只读；P，即Programmable，说明可通过特殊电子设备写入；E，即Erasable，说明可擦写；E平方说明是两个E，第二个E是电子。

（2）主存储器的组成

实际的存储器总是由一片或多片存储器配以控制电路构成的（如图1-7所示）。其容量为 $W \times B$ ，W是存储单元（word，即字）的数量，B表示每个word由多少bit（位）组成。如果某一芯片规格为 $w \times b$ ，则组成 $W \times B$ 的存储器需要用 $(W/w) \times (B/b)$ 个芯片。

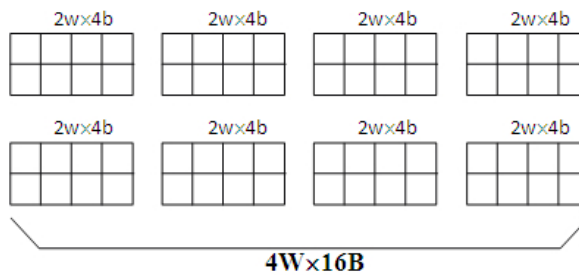


图1-7主存储器组成示意图

(3) 主存储器的地址编码

主存储器（内存）采用的是随机存取方式，需对每个数据块进行编码，而在主存储器中数据块是以word来标识的，即每个字一个地址，通常采用的是16进制表示。例如，按字节编址，地址从A4000H到CBFFFH，则表示有(CBFFF-A4000)+1个字节，28000H个，也就是163840个字节，等于160KB。

要注意的是，编址的基础可以是字节，也可以是字（字是由1个或多个字节组成的），要算地址位数，首先应计算要编址的字或字节数，然后求2的对数即可得到。

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

第1章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

一点一练

试题1

位于CPU与主存之间的高速缓冲存储器Cache用于存放部分主存数据的拷贝，主存地址与Cache地址之间的转换工作由__(1)__完成。

- (1) A. 硬件 B. 软件 C. 用户 D. 程序员

试题2

在程序的执行过程中，Cache与主存的地址映像由__(2)__。

- (2) A. 专门的硬件自动完成 B. 程序员进行调度
C. 操作系统进行管理 D. 程序员和操作系统共同协调完成

试题3

相联存储器按__(3)__访问。

- (3) A. 地址 B. 先入后出的方式
C. 内容 D. 先入先出的方式

试题4

内存单元按字节编址，地址0000A000H~0000BFFFH共有__(4)__个存储单元。

- (4) A. 8192K B. 1024K C. 13K D. 8K

试题5

若内存容量为4GB，字长为32，则__(5)__。

- (5) A. 地址总线和数据总线的宽度都为32

B．地址总线的宽度为30，数据总线的宽度为32

C．地址总线的宽度为30，数据总线的宽度为8

D．地址总线的宽度为32，数据总线的宽度为8

试题6

若某计算机系统的I/O接口与主存采用统一编址，则输入输出操作是通过__(6)__指令来完成的。

(6) A．控制 B．中断 C．输入输出 D．访存

试题7

设用2K×4位的存储器芯片组成16K×8位的存储器（地址单元为0000H～3FFFH，每个芯片的地址空间连续），则地址单元0B1FH所在芯片的最小地址编号为__(7)__。

(7) A．0000H B．0800 H C．2000 H D．2800 H

试题8

正常情况下，操作系统对保存有大量有用数据的硬盘进行__(8)__操作时，不会清除有用数据。

(8) A．磁盘分区和格式化 B．磁盘格式化和碎片整理

C．磁盘清理和碎片整理 D．磁盘分区和磁盘清理

试题9

__(9)__不属于按寻址方式划分的一类存储器。

(9) A．随机存储器 B．顺序存储器

C．相联存储器 D．直接存储器

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

解析与答案

试题1分析

从Cache-主存层次实现的目标看，一方面既要使CPU的访存速度接近于访Cache的速度，另一方面为用户程序提供的运行空间应保持为主存容量大小的存储空间。在采用Cache-主存层次的系统中，Cache对用户程序而言是透明的，也就是说，用户程序可以不需要知道Cache的存在。因此，CPU每次访存时，依然和未使用Cache的情况一样，给出的是一个主存地址。但在Cache-主存结构中，CPU首先访问的是Cache，并不是主存。为此，需要一种机制将主存地址转换成对应的Cache地址，这个处理过程对速度要求非常高，因此它是完全由硬件来完成的。

试题1答案

(1) A

试题2分析

Cache与主存的地址映像需要专门的硬件自动完成，使用硬件来处理具有更高的转换速率。

试题2答案

(2) A

试题3分析

相联存储器一种按内容进行存储和访问的存储器。

试题3答案

(3) C

试题4分析

本题考查计算机中的存储部件组成。

内存按字节编址，地址从0000A000H到0000BFFFH时，存储单元数为0000BFFFH

-0000A000H + 1H = 00002000H，转换为二进制后为0010 0000 0000 0000即 2^{13} ，即8K个存储单元。

试题4答案

(4) D

试题5分析

字长是指在同一时间中CPU处理二进制数的位数叫字长。

数据总线是用于在计算机中传送数据的总线，它可以把CPU的数据传送到存储器或输入输出接口等其它部件，也可以将其它部件的数据传送到CPU。数据总线的位数是微型计算机的一个重要指标，通常与字长一致，所以字长32也就意味着数据总线的宽度为32。

地址总线是传送地址信息的总线，根据地址总线的多少可以确定内存容量的大小，如32位的地址总线可以允许 2^{32} 次方的内存容量。

试题5答案

(5) A

试题6分析

I/O接口与主存采用统一编址，即将I/O设备的接口与主存单元一样看待，每个端口占用一个存储单元的地址，其实就是将主存的一部分划出来作为I/O地址空间。

访存指令是指访问内存的指令，显然，这里需要访问内存，才能找到相应的输入输出设备，一次需要使用访存指令。

而控制类指令通常是指程序控制类指令，用于控制程序流程改变的指令，包括条件转移指令、无条件转移指令、循环控制指令、程序调用和返回指令、中断指令等。

试题6答案

(6) D

试题7分析

芯片的大小为 $2k \times 4$ 位，而存储器的大小为 $16k \times 8$ 位，不难得出要获得这样一个大小的存储器，需要16片 $2k \times 4$ 位的芯片。

如果按字节编址，对应一个大小为 $16k \times 8$ 位的存储器，需要14位地址，其中高4位为片选地址，低10位为片内地址，而题目给出的地址0B1FH转换为二进制为00 1011 0001 1111，其高4位为0010，即片选地址为2。因此，地址0B1FH对应第2片芯片，该芯片的起始地址（最小地址）为00 1000 0000 0000，即0800H。

试题7答案

(7) B

试题8分析

这题主要考查我们对磁盘分区、格式化、碎片整理、磁盘清理等概念的理解。

磁盘分区是指将一块容量相对较大的磁盘划分为多块容量相对较小的磁盘，磁盘分区并不删除磁盘上的数据，但一般情况下，磁盘分区后要经过格式化后才能正式使用。

磁盘格式化在往磁盘的所有数据区上写零的操作过程，它是一种纯物理操作，同时对硬盘介质做一致性检测，并且标记出不可读和坏的扇区。那么格式化后，磁盘原有的数据被清除。

磁盘清理也可以删除计算机上的文件，但它主要用于删除计算机上那些不需要的文件。

碎片整理是指通过系统软件或一些专业的磁盘碎片整理软件对计算机磁盘在长期使用过程中产生的碎片和凌乱文件重新整理，以释放出更多可用的存储空间。

综上所述，不会清除计算机中有用数据的是C。

试题8答案

(8) C

试题9分析

本题考查存储器类型的划分。

按寻址方式划分，存储器可分为：顺序存储（如：磁带机），直接存储器（如：磁盘、硬盘、光盘），随机存储器（如：内存、U盘）。

试题9答案

(9) C

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

第1章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

考点精讲

可靠性计算在软件设计师考试中出现频度较高，出题方式也较为单一，主要是计算，是一个很好得分的知识点，另外一个知识点是校验码，出现频度较低，需要了解概念及计算过程，本节将详细介绍这部分内容。

1. 可靠性计算

可靠性计算主要涉及三种系统，即串联系统、并联系统和冗余系统，其中串联系统和并联系统的可靠性计算都非常简单，只要了解其概念，公式很容易记住。冗余系统要复杂一些，有些学员常常搞不清楚。

(1) 串联系统

假设一个系统由n个子系统组成，当且仅当所有的子系统都能正常工作时，系统才能正常工作，这种系统称为串联系统，如图1-8所示。



图1-8串联系统

设系统各个子系统的可靠性和失效率分别用 R_1, R_2, \dots, R_n 表示，则系统的可靠性 $R = R_1 \times R_2 \times \dots \times R_n$ 。

如果系统的各个子系统的失效率分别用 $\lambda_1, \lambda_2, \dots, \lambda_n$ 来表示，则系统的失效率

$$\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_n。$$

(2) 并联系统

假如一个系统由n个子系统组成，只要有一个子系统能够正常工作，系统就能正常工作，如图1-9所示。

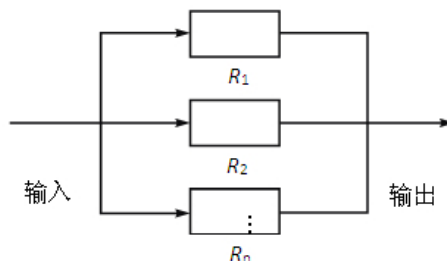


图1-9 并联系统

设系统各个子系统的可靠性分别用 R_1, R_2, \dots, R_n 表示，则系统的可靠性 $R = 1 - (1 - R_1) \times (1 - R_2) \times \cdots \times (1 - R_n)$ 。

假如所有子系统的失效率均为 λ ，则系统的失效率为 μ ：

$$\mu = \frac{1}{\frac{1}{\lambda} \sum_{j=1}^n \frac{1}{j}}$$

在并联系统中只有一个子系统是真正需要的，其余n-1个子系统都被称为冗余子系统。该系统随着冗余子系统数量的增加，其平均无故障时间也会增加。

(3) 模冗余系统

m模冗余系统由m个（ $m=2n+1$ 为奇数）相同的子系统和一个表决器组成，经过表决器表决后，m个子系统中占多数相同结果的输出可作为系统的输出，如图1-10所示。

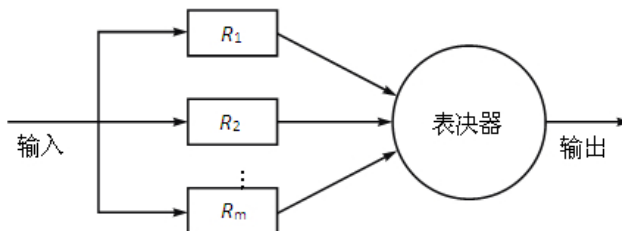


图1-10 模冗余系统

在m个子系统中，只有n+1个或n+1个以上的子系统能正常工作，系统就能正常工作并输出正确结果。假设表决器是完全可靠的，每个子系统的可靠性为 R_0 ，则m模冗余系统的可靠性为：

$$\sum_{i=n+1}^m C_m^i R_0^i (1 - R_0)^{m-i}$$

2. 校验码

为了实现数据的自动检错与纠错，引入了校验码。而最简单的就是奇偶校验码，它分为奇校验和偶校验两种，均是添加1位校验位，根据信息码中1的个数来决定校验位的取值，使得填入校验位后，使得1的个数为奇数（奇校验）或偶数（偶校验）。这方面知识更深入的考察点主要包括以下几个方面：

(1) 海明码距

海明的冗余数据位检测和纠正代码差错的理论和方法指出：可以在数据代码上添加若干冗余位组成码字。而将一个码字变成另一个码字时必须改变的最小位数就是码字之间海明距离，简称码距。从这里将得出：没有加冗余校验码的任何编码，它们的码距就是1，即只要改一位，就可以变成

另一个码字了；而奇偶校验码则添加了1位校验码，使得要变成另一个码字最少要修改两位，这就使其码距变成2了；根据定义得知，码距是不同码字的海明距离的最小值。判断码距时，可以列出一些码进行判断，找出最小的位数即可。

另外，还需要记住以下几个关键的关系：

可查出多少位错误：根据海明的研究发现，可以发现“ \leq 码距-1”位的错误。

可以纠正多少位错误：根据海明的研究发现，可以纠正“ $<$ 码距/2”位的错误，因此如果要能够纠正n位错误，则所需最小的码距应该是“ $2n+1$ ”。

(2) 海明校验码

要计算海明校验码，首先要知道海明校验码是放置在2的幂次位上的，即“1、2、4、8、16、32……”，而对于信息位为m的原始数据，需加入k位的校验码，它满足 $m+k+1 < 2^k$ 。计算时总令人感到头痛。而有一种简单的方法，则是从第1位开始写，遇到校验位留下空格。例如：原始信息为101101100，并采用偶校验则：

		1		0	1	1		0	1	1	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13

图1-11填入原始信息

然后根据以下公式填充校验位“1、2、4、8”：

$$\text{Bit } 1 = B_3 \oplus B_5 \oplus B_7 \oplus B_9 \oplus B_{11} \oplus B_{13} = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$\text{Bit } 2 = B_3 \oplus B_6 \oplus B_7 \oplus B_{10} \oplus B_{11} = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$\text{Bit } 4 = B_5 \oplus B_6 \oplus B_7 \oplus B_{12} \oplus B_{13} = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$\text{Bit } 8 = B_9 \oplus B_{10} \oplus B_{11} \oplus B_{12} \oplus B_{13} = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

(注： \oplus 指的是异或运算； B_n 代表位数)

然后将结果填入，得到：

1	1	1	0	0	1	1	0	0	1	1	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13

图1-12完成计算

而如果给出一个加入了校验码的信息，并说明有一位错误，要找出，则可以采用基本相同的方法，假如给出的是：

1	1	1	0	0	1	1	0	0	1	0	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13

图1-13有一位错误

	B8	B4	B2	B1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1

可根据以下公式计算：

$$\text{Bit 1} = B1 \oplus B3 \oplus B5 \oplus B7 \oplus B9 \oplus B11 \oplus B13 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$\text{Bit 2} = B2 \oplus B3 \oplus B6 \oplus B7 \oplus B10 \oplus B11 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0 = 1$$

$$\text{Bit 4} = B4 \oplus B5 \oplus B6 \oplus B7 \oplus B12 \oplus B13 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0 = 0$$

$$\text{Bit 8} = B8 \oplus B9 \oplus B10 \oplus B11 \oplus B12 \oplus B13 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1$$

然后从高位往下写，得到1101，即十进制的11，因此出错的位数为第11位。而剩下的问题就是这个公式如何来的？首先计算校验码时，1、2、4、8位都是空的，因此在公式的左边；当进行校验时，1、2、4、8位都已经有了值，因此要参与计算。而这些值是根据右表得到的，也就是生成B1、B2、B4、B8四个公式，而公式中要参与计算的位，是在表格中出现“1”的那个位。要说明的是，右边的表格，就是对数据位的二进制描述。

由于海明码距在计算和纠错过程中，计算都过于复杂，无法很容易地使用硬件实现，因此在实际的应用中并不是应用得很广泛。

(3) CRC校验码

由于CRC的实现原理十分易于用硬件实现，因此被广泛地应用于计算机网络上的差错控制。而CRC的考察点主要有两个：计算CRC校验码；验算一个加了CRC校验的码是否有错误。

计算CRC校验码

要计算CRC校验码，需根据CRC生成多项式进行。例如：原始报文为“11001010101”，其生成多项式为：“ $x^4 + x^3 + x + 1$ ”。在计算时，是在原始报文的后面若干个0（等于校验码的位数，而生成多项式的最高幂次就是校验位的位数，即使用该生成多项式产生的校验码为4位）作为被除数，除以生成多项式所对应的二进制数（根据其幂次的值决定，得到11011，因为生成多项式中除了没有 x^2 之外，其他位都有）。然后使用模2除，得到的商就是校验码。

$$\begin{array}{r}
 11011 \overline{) 110010101010000} \\
 \underline{11011} \\
 10010 \\
 \underline{11011} \\
 10011 \\
 \underline{11011} \\
 10000 \\
 \underline{11011} \\
 10111 \\
 \underline{11011} \\
 11000 \\
 \underline{11011} \\
 11000 \\
 \underline{11011} \\
 0011
 \end{array}$$

图1-14计算CRC校验码

然后将0011添加到原始报文的后面就是结果：110010101010011。

检查信息码是否有CRC错误

要想检查信息码是否出现了CRC错误的计算很简单，只需待检查的信息码做被除数，除以生成多项式，如果能够整除就说明没有错误，否则就是出错了。另外要注意的是，当CRC检查出现错误时，它是不会进行纠错的，通常是让信息的发送方重发一遍。

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

一点一练

试题1

某计算机系统的部件构成如图1-15所示，假定每个部件的千小时可靠度都为R，则该系统的千小时可靠度为__(1)__。

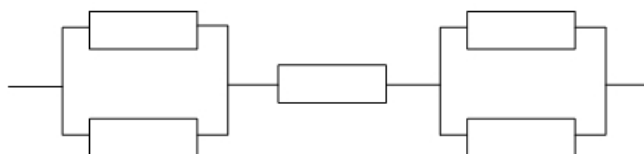


图1-15可靠性计算示意图

- (1) A . $R+2R/4$ B . $R+R^2/4$ C . $R(1-(1-R)^2)$ D . $R(1-(1-R)^2)^2$

试题3

下面关于校验方法的叙述，__(3)__是正确的。

- (3) A . 采用奇偶校验可检测数据传输过程中出现一位数据错误的位置并加以纠正
 B . 采用海明校验可检测数据传输过程中出现一位数据错误的位置并加以纠正
 C . 采用海明校验，校验码的长度和位置可随机设定

D. 采用CRC校验，需要将校验码分散开并插入数据的指定位置中

试题4

某数据处理软件包括两个完全相同的数据处理部件和一个数据存储部件，且采用图1-16给出的容错方案。当数据处理部件的可靠性为0.6时，为使整个软件系统的可靠性不小于0.66，则数据存储部件的可靠性至少应为_(4)_____。

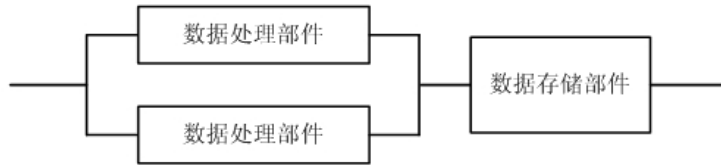


图1-16 先并联后串联系统

(4) A. 0.6 B. 0.66 C. 0.79 D. 1.0

试题5

若计算机采用CRC进行差错校验，生成多项式为 $G(X) = X^4 + X + 1$ ，信息字为10110，则CRC校验码是_(5)_____。

(5) A. 0000 B. 0100 C. 0010 D. 1111

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

解析与答案

试题1分析

本题考查系统可靠性。

计算机系统是一个复杂的系统，而且影响其可靠性的因素也非常繁复，很难直接对其进行可靠性分析。若采用串联方式，则系统可靠性为每个部件的乘积 $R = R_1 \times R_2 \times R_3 \times \dots \times R_n$ ；若采用并联方式，则系统的可靠性为 $R = 1 - (1 - R_1) \times (1 - R_2) \times (1 - R_3) \times \dots \times (1 - R_n)$ 。

在本题中，既有并联又有串联，计算时首先我们要分别计算图中两个并联后的可靠度，它们分别为 $1 - (1 - R)^2$ ，然后是三者串联，根据串联的计算公式，可得系统的可靠度为 $R \times 1 - (1 - R)^2 \times 1 - (1 - R)^2 = R(1 - (1 - R)^2)^2$ 。因此本题答案选D。

试题1答案

(1) D

试题2

冗余技术通常分为4类，其中_(2)_____按照工作方法可以分为静态、动态和混合冗余。

(2) A. 时间冗余 B. 信息冗余 C. 结构冗余 D. 冗余附加技术

试题2分析

冗余技术一般包括时间冗余、信息冗余、结构冗余和冗余附加技术，其中结构冗余按照工作方法可以分为静态、动态和混合冗余。

试题2答案

(2) C

试题3分析

本题考查基本校验码，题目中提到的3种校验方式都是需要考生掌握的。

其先进度排名为：奇偶校验<CRC校验<海明校验。

奇偶校验码是在源信息码的基础上添加一个校验位，使整个信息位呈奇性或偶性。这种校验码只能根据收到的信息奇偶性判断信息在传输中是否产生了1位错误，同时不能判断具体是哪一位出了错。

CRC校验即循环冗余码校验（Cyclical Redundancy Check），它是利用除法及余数的原理来做错误侦测（Error Detecting）的。实际应用时，发送装置计算出CRC值并随数据一同发送给接收装置，接收装置对收到的数据重新计算CRC并与收到的CRC相比较，若两个CRC值不同，则说明数据通信出现错误。

海明校验是由Richard Hamming于1950年提出、目前还被广泛采用的一种很有效的校验方法，它只要增加少数几个校验位，就可以发现“≤码距-1”位的错误，可以纠正“<码距/2”位的错误。因此如果要能够纠正n位错误，则所需最小的码距应该是“2n+1”；如果要纠正1位错误，则最小码距为3。

有了以上的知识基础，问题就非常容易解决了，接下来逐个分析备选答案中存在的概念错误。

A. “采用奇偶校验可检测数据传输过程中出现一位数据错误的位置并加以纠正”，奇偶校验码只能检测一位的错误，不能确定错误的位置，更无法对错误进行纠正，描述错误。

B. “采用海明校验可检测数据传输过程中出现一位数据错误的位置并加以纠正”，海明码能发现错误位置并予以改正，描述正确。

C. “采用海明校验，校验码的长度和位置可随机设定”，海明码的编码过程有严格要求，对于信息位与校验位的放置也是有约定的，不能随机设定，描述错误。

D. “采用CRC校验，需要将校验码分散开并插入数据的指定位置中”，CRC码的校验位都是置于编码的最后部分（最右端），描述错误。

故正确答案为：B。

试题3答案

(3) B

试题4分析

本题考查系统可靠性，是常考的知识点。

计算机系统是一个复杂的系统，而且影响其可靠性的因素也非常繁复，很难直接对其进行可靠性分析。若采用串联方式，则系统可靠性为每个部件的乘积 $R=R_1 \times R_2 \times R_3 \times \dots \times R_n$ ；若采用并联方式，则系统的可靠性为 $R=1-(1-R_1) \times (1-R_2) \times (1-R_3) \times \dots \times (1-R_n)$ 。本道题当中的数据处理部件是并联的，每个处理部件的可靠性为0.6，把这两个数据处理部件作为整体M看，则它的可靠性为： $1-(1-0.6) \times (1-0.6) = 0.84$ 。由于M和数据存储部件是串联的，而M的可靠性为0.84，要求整个软件系统的可靠性不小于0.66，来求数据存储部件的可靠性H。利用串联公式， $0.84 \times H \geq 0.66$ ，解这个方程得到H为：0.7857，约等于0.79，所以选择C答案。

试题4答案

(4) C

试题5分析

本题考查数据校验知识。

CRC即循环冗余校验码，是数据通信领域中最常用的一种差错校验码，其特征是信息字段和校验字段的长度可以任意选定。

在CRC校验方法中，进行多项式除法（模2除法）运算后的余数为校验字段。

信息字为10110，对应的多项式 $M(x)=X_4+X_2+X$ ，生成多项式为 $G(X)=X_4+X+1$ ，对应的代码为10011。

校验码计算过程为：先将信息码左移4位（生成码长-1），得到101100000，然后反复进行异或运算（即除数和被除数最高位对齐，按位异或），如下所示：

$10110 \oplus 10011 = 00101$ ，00101左移两位得到10100再与10011异或；

$10100 \oplus 10011 = 00111$ ，00111左移两位得到11100再与10011异或；

$11100 \oplus 10011 = 01111$ ，其结果为CRC校验码，即余数1111。

第 1 章：计算机硬件基础知识

作者：希赛教育软考

习题解析



版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

第 1 章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

考前冲刺

试题1

利用高速通信网络将多台高性能工作站或微型机互连构成机群系统，其通信形式属于__ (1) __计算机。

- (1) A . 单指令流单数据流 (SISD) B . 多指令流单数据流 (MISD)
C . 单指令流多数据流 (SIMD) D . 多指令流多数据流 (MIMD)

试题2

阵列处理机属于__ (2) __计算机。

- (2) A . SISD B . SIMD C . MISD D . MIMD

试题3

每一条指令都可以分解为取指、分析和执行三步。已知取指时间 $t_{\text{取指}}=5 \Delta t$ ，分析时间 $t_{\text{分析}}=2 \Delta t$ ，执行时间 $t_{\text{执行}}=5 \Delta t$ 。如果按顺序方式从头到尾执行完500条指令需__ (3) __ Δt 。如果按照[执行] k 、[分析] $k+1$ 、[取指] $k+2$ 重叠的流水线方式执行指令，从头到尾执行完500条指令需__ (4) __ Δt 。

- (3) A . 5590 B . 5595 C . 6000 D . 6007
(4) A . 2492 B . 2500 C . 2510 D . 2515

试题4

若内存按字节编址，用存储容量为 $32K \times 8$ 比特的存储器芯片构成地址编号为A0000H ~ DFFFFH的内存空间，则至少需要__ (5) __片。

(5) A . 4 B . 6 C . 8 D . 10

试题5

__ (6) __ 不属于计算机控制器中的部件。

- (6) A . 指令寄存器 IR B . 程序计数器 PC
C . 算术逻辑单元 ALU D . 程序状态字寄存器 PSW

试题6

下面的描述中, __ (7) __ 不是 RISC 设计应遵循的设计原则。

- (7) A . 指令条数应少一些
B . 寻址方式尽可能少
C . 采用变长指令, 功能复杂的指令长度长而简单指令长度短
D . 设计尽可能多的通用寄存器

试题7

系统响应时间和作业吞吐量是衡量计算机系统性能的重要指标。对于一个持续处理业务的系统而言, 其__ (8) __。

- (8) A . 响应时间越短, 作业吞吐量越小 B . 响应时间越短, 作业吞吐量越大
C . 响应时间越长, 作业吞吐量越大 D . 响应时间不会影响作业吞吐量

试题8

在指令系统的各种寻址方式中, 获取操作数最快的方式是__ (9) __。若操作数的地址包含在指令中, 则属于__ (10) __方式。

- (9) A . 直接寻址 B . 立即寻址 C . 寄存器寻址 D . 间接寻址
(10) A . 直接寻址 B . 立即寻址 C . 寄存器寻址 D . 间接寻址

试题9

在计算机系统中采用总线结构, 便于实现系统的积木化构造, 同时可以__ (11) __。

- (11) A . 提高数据传输速度 B . 提高数据传输量
C . 减少信息传输线的数量 D . 减少指令系统的复杂性

试题10

利用海明码 (Hamming Code) 纠正单位错, 如果有 9 位信息位, 则需要加入__ (12) __位冗余位。

- (12) A . 4 B . 5 C . 7 D . 8

试题11

以下关于CISC (Complex Instruction Set Computer, 复杂指令集计算机) 和 RISC (Reduced Instruction Set Computer, 精简指令集计算机) 的叙述中, 错误的是__ (13) __。

- (13) A . 在CISC中, 其复杂指令都采用硬布线逻辑来执行
B . 采用CISC技术的CPU, 其芯片设计复杂度更高
C . 在RISC中, 更适合采用硬布线逻辑执行指令
D . 采用RISC技术, 指令系统中的指令种类和寻址方式更少

试题12

循环冗余校验码 (CRC) 利用生成多项式进行编码。设数据位为k位, 校验位为r位, 则CRC码的格式为__ (14) __。

- (14) A. k个数据位之后跟r个校验位
B. r个校验位之后跟k个数据位
C. r个校验位随机加入k个数据位中
D. r个校验位等间隔地加入k个数据位中

版权方授权希赛网发布，侵权必究

[上一节](#) [本书简介](#) [下一节](#)

第1章：计算机硬件基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

习题解析

试题1分析

本题考查计算机组成与体系结构的Flynn分类法，是常考的知识点。关于Flynn分类法请参看表1-1。

题中是利用高速通信网络将多台高性能工作站或微型机互连构成机群系统，事实上是采用了多处理机。

试题1答案

(1) D

试题2分析

本题考查计算机组成与体系结构的Flynn分类法，是常考的知识点。关于Flynn分类法请参看表1-1。

根据阵列机的定义，它将大量重复设置的处理单元互连构成阵列，在单一控制部件的控制下，向各处理单元分配各自的数据，以达到并行执行同一条指令的目的。因此，阵列处理机是单指令流多数据流（SIMD）计算机。

试题2答案

(2) B

试题3分析

按顺序方式执行指令，每条指令从取指到执行共耗时 $12\Delta t$ ，所以500条指令共耗时： $12 \times 500 = 6000\Delta t$ 。

采用流水线方式时，系统在同一时刻可以进行第k条指令的取指，第k+1条指令的分析，第k+2条指令的执行，所以效率大大提高了。采用流水线的执行示意图如图1-17所示。

平时大家看到的都是这样的示意图，但是平时我们看到的图都是笼统的。这里把所有周期都定为统一长度，这样流水线的总时间为： $(n+2) \times \text{周期}$ 。如此题中为： $(500+2) \times 5 = 2510$ 。

但我们平时用的流水线计算公式是：第一条指令顺序执行时间+(指令条数-1)×周期这个公式是怎么来的呢？

请大家看图1-18。

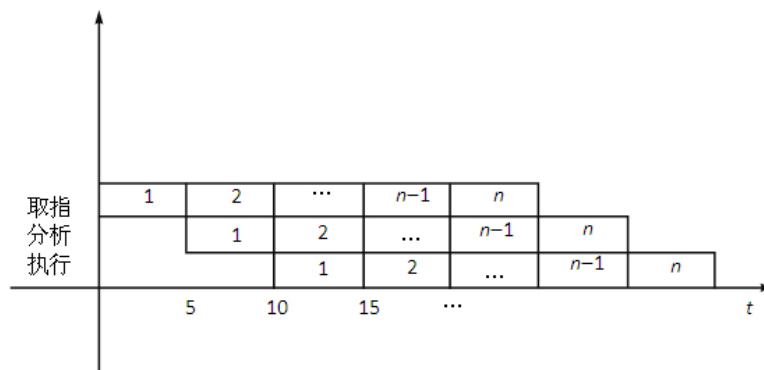


图1-17流水线执行示意图一

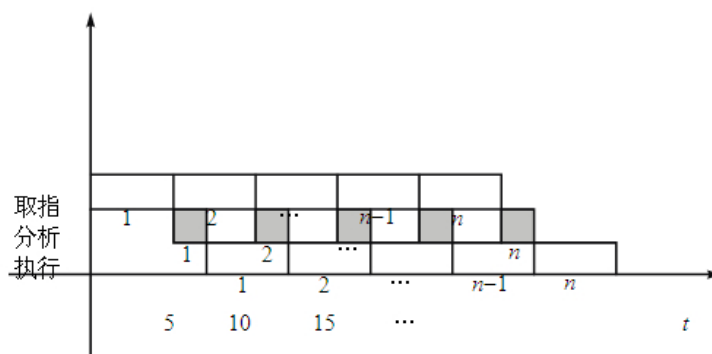


图1-18流水线执行示意图二

对于此题而言，关键在于指令的分析时间，周期是 5Δ ，而实际完成分析只需要 2Δ 时间，所以正常运行时空图应如图1-18所示，其中黑色块是分析的真实发生时间。所以采用流水线的耗时为： $5+2+5\times(500-1)+5=2507$ 。

题目是按图1-17来计算的，计算结果为2510。

试题3答案

(3) C (4) C

试题4分析

此题的解题思路是先计算出地址编号为A0000H~DFFFFH的内存空间大小，然后用空间大小除以芯片容量，得到芯片数量。在这个操作过程中，运算单位及数制的一致性特别需要注意，在进行运算之前，一定得把单位转化成相同的。下面是具体运算过程。

$DFFFFH - A0000H + 1 = 40000H$ ，转化为十进制为 2^{18} 。由于内存是按字节编址，所以空间大小应为 2^8KB ，即256KB， $32K \times 8$ 比特的芯片即 $32K \times 1$ 字节的芯片，所以 $256KB / 32KB = 8$ 。所以正确答案为C。

试题4答案

(5) C

试题5分析

控制器是分析和执行指令的部件，也是统一指挥和控制计算机各个部件按时序协调操作的部件。控制器的组成包含如下部分：①程序计数器PC；②指令寄存器IR；③指令译码器；④时序部件；⑤程序状态字寄存器PSW；⑥中断机构。故C答案的算术逻辑单元ALU不属于控制器，是运算器。

试题5答案

(6) C

试题6分析

RISC（精简指令系统计算机）的设计原则有：

- （1）只使用频度高的及最有用的指令，一般为几十条指令；
- （2）指令格式简单化、规格化；
- （3）每条指令在一个机器周期内完成；
- （4）只有存数和取数指令访问存储器；
- （5）以最简单、有效的方式支持高级语言。

很显然，C答案错误。

试题6答案

（7）C

试题7分析

系统响应时间是指用户发出完整请求到系统完成任务给出响应的时间间隔。作业吞吐量是指单位时间内系统完成的任务量。若一个给定系统持续地收到用户提交的任务请求，则系统的响应时间将对作业吞吐量造成一定影响。若每个任务的响应时间越短，则系统的空闲资源越多，整个系统在单位时间内完成的任务量将越大；反之，若响应时间越长，则系统的空闲资源越少，整个系统在单位时间内完成的任务量将越小。

试题7答案

（8）B

试题8分析

此题考查的是考生对操作数几种基本寻址方式的理解。关于基本寻址方式的描述，请参看“1.3 计算机系统的组成与体系结构”的“指令系统基础”内容。

试题8答案

（9）B（10）A

试题9分析

总线是在计算机中连接两个或多个功能部件的一组共享的信息传输线，它的主要特征就是多个部件共享传输介质。它是构成计算机系统的骨架，是各个功能部件之间进行信息传输的公共通道，借助总线的连接，计算机各个部件之间可以传送地址、数据和各种控制信息。在计算机系统中采用总线结构，便于实现系统的积木化构造，同时可以有效减少信息传输线的数量。

试题9答案

（11）C

试题10分析

按照海明的理论，纠错码的编码就是把所有合法的码字尽量安排在 n 维超立方体的顶点上，使得任一对码字之间的距离尽可能大。如果任意两个码字之间的海明距离是 d ，则所有少于等于 $d-1$ 位的错误都可以检查出来，所有少于 $d/2$ 位的错误都可以纠正。一个自然的推论是，对某种长度的错误串，要纠正错误就要用比仅仅检测它多一倍的冗余位。如果对于 m 位的数据，增加 k 位冗余位， $n=m+k$ 位的纠错码，则有： $m+k+1 < 2^k$ 对于给定的数据位 m ，上式给出了 k 的下界，即要纠正单个错误， k 必须取最小值。在本题中， $m=9$ ， $9+k+1 < 2^k$ ，可取 $k=4$ ，得到 $9+4+1=14 < 2^4=16$ 。

试题10答案

（12）A

试题11分析

本题考查指令系统和计算机体系结构基础知识。

CISC (Complex Instruction Set Computer,复杂指令集计算机) 的基本思想是：进一步增强原有指令的功能，用更为复杂的新指令取代原先由软件子程序完成的功能，实现软件功能的硬件化，导致机器的指令系统越来越庞大而复杂。CISC计算机一般所含的指令数目至少300条以上，有的甚至超过500条。

RISC (Reduced Instruction Set Computer，精简指令集计算机) 的基本思想是：通过减少指令总数和简化指令功能，降低硬件设计的复杂度，使指令能单周期执行，并通过优化编译提高指令的执行速度，采用硬布线控制逻辑优化编译程序。在20世纪70年代末开始兴起，导致机器的指令系统进一步精炼而简单。

试题11答案

(13) A

试题12分析

本题考查循环冗余校验码。

循环冗余校验码的编码组成结构为：数据位+校验位。所以本题应选A。此外，可以做一些知识扩展，即海明码的数据与校验位的组成方式为： 2^n 位置放校验位，其它位置放数据位。

试题12答案

(14) A

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)

第 2 章：操作系统基础知识

作者：希赛教育软考学院 来源：希赛网 2014年05月04日

考点脉络

软件设计师需要有扎实的理论知识，而操作系统作为计算机科学最为基本的理论基础和分支领域之一，是软件设计师必须重点掌握的知识。本章将介绍操作系统相关的考点，并辅以练习题，以便考生切实掌握相关内容。

根据考试大纲，本章要求考生掌握以下几个方面的知识。

- (1) 操作系统的内核。
- (2) 操作系统的五大管理功能：进程管理、存储管理、设备管理、文件管理、作业管理。
- (3) 网络操作系统和嵌入式操作系统基础知识。
- (4) 操作系统的配置。

从历年的考试情况来看，本章主要考查进程状态转换图、信号量与PV操作、死锁问题、银行家算法、段页式存储、页面置换算法、磁盘调度、树形文件系统。

[版权方授权希赛网发布，侵权必究](#)

[上一节](#) [本书简介](#) [下一节](#)