上海市电子计算机应用软件人员水平考试参考资料

计算机硬件与软件基础知识

《微电脑世界》编辑部 编

上海科学技术出版社

上海市电子计算机应用软件人员水平考试参考资料

计算机硬件与软件基础知识

《微电脑世界》编辑部　编

上海科学技术出版社

出 版 说 明

电子计算机为人类的智力解放建立了一个里程碑。科学家们认为一场新的技术革命正以计算机科学技术为中心在世界范围内逐步展开。计算机的研究水平、生产能力和应用程度已成为一个国家现代化的重要标志。我国四个现代化的宏伟大业急需大批计算机科技人材，特别是从事计算机软件工作的专业人员。上海市人民政府决定从1985年起，每年举行一次计算机软件人员的水平考试。为了给应试的同志提供一本较为合适的复习资料，《微电脑世界》编辑部根据水平考试委员会公布的大纲编写了这本《计算机硬件与软件基础知识》。不论在题材范围、实例选取和内容深度上，我们都力求做到简明扼要，实用有效。本书不仅是报考者的一本合适的参考书，也可作为知识青年的入门书和计算机知识的科普读物。

一九八四年元月

计算机硬件与软件基础知识
《微电脑世界》编辑部　编
上海科学技术出版社出版
（上海瑞金二路450号）
新华书店上海发行所发行　上海市崇明县永南印刷厂
开本850×1156　1/16　印张8　字数187,000
1985年3月第1版　1985年3月第1次印刷
印数1—10,000

书号：13119·1304　　定价：1.52元

目　录

第一部分　计算机系统硬件基础知识

第二部分　软件基础知识

第一部分　计算机系统硬件基础知识

我们知道，一个现代的计算机系统是由“硬件”与“软件”两大部分组成的。(当然,还有电源设备。)硬件指的是系统的物理实体即计算机的机体。包括组成计算机的那些机械的、磁性的、电子的装置和部件，如运算器、存贮器、控制器、输入/输出设备、通道和外存等等。软件指的是为了方便用户，扩大计算机功能、提高计算机效率而建立的各种程序和有关资料。这种程序可称系统程序，包括操作系统、汇编程序、编译程序、程序库、检查与诊断程序、专用程序包和各种程序设计语言等。机器系统(硬件)和程序系统(软件)便构成了一个完整的计算机系统。

- 电子计算机系统
  - 硬件
    - 主机
      - 中央处理机
        - 运算器
        - 控制器
      - 内存贮器(主存)
    - 外部设备
      - 输入设备
      - 输出设备
      - 外存贮器(辅助存贮器)
    - 外围设备
      - 模—数转换器
      - 数—模转换器
      - 其他
  - 软件
    - 系统软件
      - 操作系统
      - 编译系统
      - 诊断系统
      - 其它
    - 程序设计语言
      - 机器语言
      - 汇编语言
      - 高级语言
    - 应用软件

下面要介绍的是关于数字电子计算机系统硬件的基础知识，分为九个题目。

一、数制及进位制数的换算

二、算术运算与逻辑运算

三、计算机内数的表示方式

四、非数值数据在计算机内的表示

五、几种常用的校验码
六、电子计算机的基本结构
七、指令与指令系统的概念
八、存贮器的种类、功能和特征
九、输入/输出设备的种类与特点

第一章　数制及进位制数的换算

1.1　计　数　制

计算机系统中采用的数制有十进制、二进制、八进制和十六进制。这些进位制共同具有如下三个特点。

(1) 每种进位制都有一个基数。十进制的基数是10，二进制、八进制与十六进制的基数分别为2，8和16。

(2) 各种进位制有自己的基本数字。十进制的基本数字是0，1，2，3，4，5，6，7，8，9，二进制与八进制的基本数字分别为0，1与0，1，2，3，4，5，6，7，而十六进制的基本数字则为0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F。

(3) 在一种数制中，用一串基本数字来表示一个数，而每个基本数字(除 0 外)在数中的大小与它的位置有关。例如十进制的2，在个位上(小数点左边第一位)表示 $2\times10^0=2$，在十位上(小数点左边第二位)表示$2\times10^1=20$，在百位上表示$2\times10^2=200$，而在十分位上(小数点右边第一位)却表示 $2\times10^{-1}=0.2$，如此等等。这里的 10^0，10^1，10^2，10^{-1}，……为十进制赋与各个数位的“权”。

二进制的权：(逢二进一)

$$\cdots\cdots,2^3,\ 2^2,\ 2^1,\ 2^0.2^{-1},\ 2^{-2},\ 2^{-3},\ \cdots\cdots$$

$$\uparrow$$

小数点

八进制的权：(逢八进一)

$$\cdots\cdots,8^3,\ 8^2,\ 8^1,\ 8^0.8^{-1},\ 8^{-2},\ 8^{-3},\ \cdots\cdots$$

十六进制的权：(逢十六进一)

$$\cdots\cdots,16^3,\ 16^2,\ 16^1,\ 16^0.16^{-1},\ 16^{-2},\ 16^{-3},\ \cdots\cdots$$

例如，十进制数4286.57可以写成如下形式

$$(4286.57)_{10}=4\times10^3+2\times10^2+8\times10^1+6\times10^0+5\times10^{-1}+7\times10^{-2}$$

1.2　进位制数的换算

(1) 十进制数与二进制数的换算

电子计算机内部是使用二进制进行工作的，但人们却习惯于十进制数，送入计算机的原始数据大多是十进制的。要想用计算机处理十进制数，必须把十进制的原始数据先转换成二进制数，机器才能进行运算。而计算机运算出的二进制结果，也需要转化成十进制数显示或打印出来，才容易为人们所接受。

那么，怎样把十进制数换算成二进制数?又怎样将二进制数换算为十进制数呢?

把一个十进制整数化成二进制数的规则是：“除二取余”。即将该十进制整数除以2，得

到商和余数后再将商用 2 除，得到新的商和余数。如此继续下去，直到商等于零为止。然后将各次所得余数，以最后得到的余数为最高位数字，依次排列起来，就是所求的与原十进制整数等值的二进制数。

例 1　要把十进制数 725 转换为二进制数，设

$$(725)_{10}=(k_nk_{n-1}\cdots\cdots k_1k_0)_2$$

除 2 取余的计算过程如下：

2	725	余数 $=1\cdots\cdots k_0$
2	362	余数 $=0\cdots\cdots k_1$
2	181	余数 $=1\cdots\cdots k_2$
2	90	余数 $=0\cdots\cdots k_3$
2	45	余数 $=1\cdots\cdots k_4$
2	22	余数 $=0\cdots\cdots k_5$
2	11	余数 $=1\cdots\cdots k_6$
2	5	余数 $=1\cdots\cdots k_7$
2	2	余数 $=0\cdots\cdots k_8$
2	1	余数 $=1\cdots\cdots k_9$
	0	

于是

$$(725)_{10}=(1011010101)_2$$

把一个十进制纯小数化成二进制数的规则是："乘二取整"。即用 2 乘十进制纯小数，取出整数部分后再用 2 去乘剩下的纯小数部分，又得到新的整数部分。如此继续下去，直到剩下的纯小数部分为零或达到所要求的精确度为止。然后把每次取出的整数部分按出现的先后顺序依次排列起来，就是与原十进制纯小数等值的二进制数。

例 2　要把十进制数 0.5625 表示成二进制形式，设

$$(0.5625)_{10}=(0,\ k_{-1}k_{-2}k_{-m})_2$$

乘 2 取整的计算过程如下：

$$\begin{array}{r}0.5625\\ \times\quad 2\\ \hline 1.1250\end{array}\qquad \text{整数部分}=1\cdots\cdots k_{-1}$$

$$\begin{array}{r}0.1250\\ \times\quad 2\\ \hline 0.2500\end{array}\qquad \text{整数部分}=0\cdots\cdots k_{-2}$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
0.5000
\end{array}
\qquad 整数部分 = 0 \cdots\cdots k_{-3}
$$

$$
\begin{array}{r}
\times \quad 2 \\
\hline
1.0000
\end{array}
\qquad 整数部分 = 1 \cdots\cdots k_{-4}
$$

于是

$$(0.5625)_{10} = (0.1001)_2$$

对具有整数和小数两个部分的十进制数，分别按上述规则将整数部分和小数部分分别换算成二进制形式，然后加起来即可。例如，根据上面的换算结果，知

$$(725)_{10} = (1011010101)_2$$

$$(0.5625)_{10} = (0.1001)_2$$

加起来便得到

$$(725.5625)_{10} = (1011010101.1001)_2$$

反过来 要想把一个二进制数转化为十进制数，只需将二进制数的各位数字乘以相应的"权"，然后相加。

例 3

$$
\begin{aligned}
&(101.101)_2 \\
&= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
&= 4 + 1 + 0.5 + 0.125 \\
&= (5.625)_{10}
\end{aligned}
$$

又例如，

$$
\begin{aligned}
&(11001.1001)_2 \\
&= 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-4} \\
&= 16 + 8 + 1 + 0.5 + 0.0625 \\
&= (25.5625)_{10}
\end{aligned}
$$

(2) 十进制数与八进制和十六进制数的换算

原理和方法跟十进制数与二进制数的换算相同。

例 4 将十进制数 725.6875 转换为八进制数。

先用"除 8 取余"的方法将 725 化成八进制形式，得到

$$(725)_{10} = (1325)_8$$

再用"乘 8 取整"的方法化 0.6875，可得

$$(0.6875)_{10} = (0.54)_8$$

因此，最后结果为

$$(725.6875)_{10} = (1325.54)_8$$

例 5 $(2047)_{10} = (7FF)_{16}$

例 6 $(0.634)_{10} = (0.A24E)_{16}$(取四位有效数字。)

(3) 二进制数与八进制和十六进制数的换算

一位八进制数正好与一个三位二进制数相当，一位十六进制数正好与一个四位二进制数相当。它们之间的这种对应关系给换算工作带来了极大的方便。

例 7 将八进制数 67.721 化为二进制形式。

由于 6 7 . 7 2 1

$$\begin{array}{cccccc} 6 & 7 & . & 7 & 2 & 1 \\ \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 110 & 111 & & 111 & 010 & 001 \end{array}$$

故有

$$(67.721)_8 = (110111.111010001)_2$$

例 8 将二进制数 11111101.01001111 化为八进制形式。

先将二进制数从小数点开始，每三位划为一组，即

011 111 101 . 010 011 110

然后写出每组相应的八进制数：

3 7 5 . 2 3 6

于是

$$(11111101.010011111)_2 = (375.236)_8$$

例 9
$$\begin{aligned}&(3E82.C7)_{16}\\ &= (\underline{0011}\ \underline{1110}\ \underline{1000}\ \underline{0010}.\underline{1100}\ \underline{0111})_2\end{aligned}$$

例10
$$\begin{aligned}&(1011010110000110.1111001)_2\\ &= (\underline{1011}\ \underline{0101}\ \underline{1000}\ \underline{0110}.\underline{1111}\ \underline{0010})\\ &= (B586.F2)_{16}\end{aligned}$$

1.3 二-十进制（二进制编码的十进制）

用若干二进制数码表示十进制的 10 个基本数字，表示的方法很多 常用的有 8—4—2—1 码，2—4—2—1 码和余 3 码。这几种编码的对应关系列于下表。

表 1

十进制数字	8421 码	2421 码	余 3 码
0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0

8421 码、2421 码和余 3 码都是用四位二进制数码表示出十进制的 10 个基本数字。8421 码赋予这四个数位的权，从左至右分别为 8，4，2，1，与通常的二进制数一样，因而这

种编码最自然、最简单也最容易识别。例如(十进制)数 427，写成 8421 码形式就是 0100，0010，0111。2421 码给四个数位赋予的“权”，从左至右分别为 2，4，2，1，而余 3 码是把 8421 码加上 0011(=3)而得到的。余 3 码是无“权”码，另两种为有权码。

要使计算机进行十进制与二进制的相互转换，常借助 8421 码作为中间过渡。比如，要输入的数为 427，先将它的 8421 码形式即 0100，0010，0111 输入机器，再由机器自动地将其转换为真正的二进制数。在某些计算机中，二-十进制形式还被用来直接进行十进制的各种运算。

习　题

1．将二进制数 100，1101，0.101，11.01，11011，-0.1001，101011.0011 转换成十进制数。

2．将下列各数按“权”展开

$$(1011.011)_2 \qquad (4211.375)_8 \qquad (3AF.4D9C)_{16}$$

3．设 $A=a_1a_2a_3a_4a_5a_6a_7$，是一个七位的二进制正整数，如何判断 A 是 4 的倍数？

4．说明十进制数转换成二进制数的原理与方法。

5．说明二进制数转换成八进制数的原理和方法。

6．用二进制表示一个四位长的十进制数最少需要几位？

7．将十进制数 5，9，11，-2，-5，$-\frac{1}{4}$，6.125，199，0.375，0.2，69.3，-523.11 转换成二进制数(二进制小数取六位)。

8．十二位长的二进制整数的表示范围有多大？

9．将下列二进制数写成八进制和十六进制数。

110111011110.110111

1111011000111.11011001

10．写出十进制数 659，7028，41378 的 8421 码形式。

第二章　算术运算与逻辑运算

绝大多数电子计算机都采用二进制来表示数和进行运算，这是因为二进制比起其他进位制来有许多优点。

(1) 二进制只有两个基本数字 0 和 1，因此任何具有两个不同稳定状态的元件，例如开关元件，都可用来表示二进制数的每一位，而制造双稳态元件要比制造多稳态元件容易得多。

(2) 二进制数的运算简单，就运算操作的简便性而言，二进制是最方便的一种计数制。

(3) 用二进制表示数还可以节省设备。

(4) 采用二进制就可以借助逻辑代数(开关代数)这一数学工具来分析和综合计算机中有关的逻辑线路，使设计数字计算机的工作得到一个有力的助手。

现代计算机所进行的基本操作大致可以分为四类：算术操作、逻辑操作、数据传送和控制操作。算术操作也就是进行算术运算，包括加、减、乘、除四则运算，求负数，求绝对值等。逻辑操作也就是进行逻辑运算。逻辑运算有逻辑加法(或运算)，逻辑乘法(与运算)，逻辑否定(非运算)、比较、选择、移位等等。

2.1 算 术 运 算

这里仅考察二进制数的加、减、乘、除四种基本的算术运算。

(1) 加法

二进制的加法规则：

$$0+0=0 \qquad 0+1=1$$
$$1+0=1 \qquad 1+1=10$$

例如，

$$\begin{array}{r} 1\,0\,1\,1 \\ +\ 1\,1\,1\,0 \\ \hline 1\,1\,0\,0\,1 \end{array}$$

(2) 减法

二进制的减法规则：

$$0-0=0 \qquad 1-0=1$$
$$1-1=0 \qquad 10-1=1$$

例如，

$$\begin{array}{r} 1\,1\,0\,1 \\ -\ \ 1\,1\,0 \\ \hline 1\,1\,1 \end{array}$$

(3) 乘法

二进制的乘法规则为

$$0\times0=0 \qquad 0\times1=0$$
$$1\times0=0 \qquad 1\times1=1$$

例如，

$$\begin{array}{r} 1101 \\ \times\quad 1010 \\ \hline 0000 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10000010 \end{array}$$

从这个实例可以看出，二进制数的乘法是由加被乘数和移位两种操作实现的。

(4) 除法

二进制的除法运算与十进制的除法相仿，例如

$$\begin{array}{r} 11100\cdots\cdots\text{商} \\ 1101\sqrt{101110101} \\ 1101 \\ \hline 10100 \\ 1101 \\ \hline 1111 \\ 1101 \\ \hline 1001\cdots\cdots\text{余数} \end{array}$$

可见，二进制的除法可以由减除数和移位两种操作实现。

2.2 逻辑运算

这里仅仅考察逻辑加法、逻辑乘法和逻辑否定三种基本的逻辑运算。在逻辑代数中变量的取值只有两种可能，或为 1 或为 0，在逻辑线路中常用高电位来表示 1，用低电位表示 0。

(1) 逻辑加法

逻辑加法的运算表如下：

$$0+0=0 \quad \text{或} \quad 0\vee0=0$$
$$0+1=1 \quad \text{或} \quad 0\vee1=1$$
$$1+0=1 \quad \text{或} \quad 1\vee0=1$$
$$1+1=1 \quad \text{或} \quad 1\vee1=1$$

逻辑加法也叫“或运算”。实现或运算的逻辑电路就是“或门”。或门的表示符号如图 1 所示。

A，B 为或门的输入，P **为输出。**以逻辑代数的眼光看来，A 和 B 为逻辑变量，取值 0 或 1，P 为逻辑加法的运算结果，其值由 A，B 决定，也就是说 $P = A + B$。

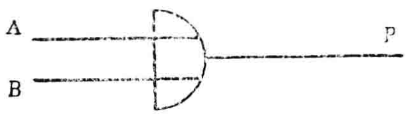

图 1

(2) 逻辑乘法

逻辑乘法的运算表如下：

$0 \cdot 0 = 0$ 或 $0 \wedge 0 = 0$

$0 \cdot 1 = 0$ 或 $0 \wedge 1 = 0$

$1 \cdot 0 = 0$ 或 $1 \wedge 0 = 0$

$1 \cdot 1 = 1$ 或 $1 \wedge 1 = 1$

逻辑乘法也叫“与运算”。实现与运算的逻辑电路就是“与门”。与门的表示符号如图2所示。A，B 为与门的输入，P 为输出。用逻辑代数的眼光看来，A 和 B 为逻辑变量，取值 0 或 1，P 为逻辑乘法的运算结果，其值由 A，B 决定，也就是说 $P = A \cdot B$。

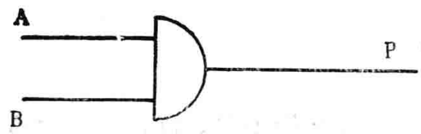

图 2

(3) 逻辑否定

这种逻辑运算就是把 1 变为 0，把 0 变为 1，在普通代数中是没有的，但在逻辑代数中却经常遇到。逻辑否定的运算表为

$\overline{0} = 1$　　　$\overline{1} = 0$

逻辑否定也叫“非运算”。实现非运算的逻辑电路就是“非门”，也就是常说的反向器。非**门的表示符**号为

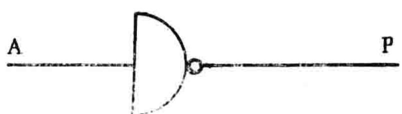

图 3

A 为非门的输入，P 为输出，在逻辑代数中它们之间的关系可表为 $P = \overline{A}$。

2.3 逻辑代数的基本公式

(1) $A + 0 = A$	$A \cdot 0 = 0$
(2) $A + 1 = 1$	$A \cdot 1 = A$
(3) $A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
(4) $A + B = B + A$	$A \cdot B = B \cdot A$
(5) $(A + B) + C = A + (B + C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
(6) $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
(7) $A + A = A$	$A \cdot A = A$
(8) $\overline{A + B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$
(9) $A + (A \cdot B) = A$	$A \cdot (A + B) = A$
(10) $\overline{\overline{A}} = A$	

说明：上面这些公式有的可利用真值表证明，也就是列举出变量可能出现的所有情况并

代入等式两端逐一验证。以第(8)组公式为例，变量只有 A 和 B 两个，而 A，B 又只能取值 0 和 1，故只有四种可能的组合，利用真值表方法对(8)组中的公式可证明如下：

表 2

A	B	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

表 3

A	B	$\overline{A\cdot B}$	$\overline{A}+\overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

有的公式我们也能采用数学推导的办法证明。例如，对公式(9)可推证如下：

$$
\begin{aligned}
& A+(A\cdot B) & \\
&=(A\cdot 1)+(A\cdot B) & \text{(根据公式(2))}\\
&=A\cdot(1+B) & \text{(根据公式(6))}\\
&=A\cdot(B+1) & \text{(根据公式(4))}\\
&=A\cdot 1 & \text{(根据公式(2))}\\
&=A & \text{(根据公式(2))}
\end{aligned}
$$

2.4 半加器与全加器

这是一种实现二进制一位加法的电路，半加器的输入端只有两个，即加数与被加数，而全加器却还要考虑到从下一个低位到来的进位。两个半加器可以组成一个全加器。

(1) 半加器

令半加器的两个输入为 A 与 B。半加器的输出也有两个，一个是 A 与 B 的和，记为 S，一个是 A 与 B 相加的进位，记为 C。A，B，S，C 之间的关系如下表所示。

表 4

输入		输出	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

一个半加器可以用“或门”，“与门”，“非门”来构成，下面就是一种实现方案。