

●电子工业工人技术等级培训教材

# 计算机英语

## 计算机英语

### 计算机英语

●黄国兴 李德宗 姚雪珍 编



English

电子工业出版社

19.6  
542

电子工业工人技术等级培训教材

# 计算机英语

黄国兴 李德宗 姚雪珍 编

4528/1



# (京)新登字 055 号

## 内 容 提 要

本书是用于电子技工培训的计算机专业英语教材,由课文、计算机屏幕上出现的英文提示信息 and 计算机常用词汇三部分构成。课文的内容都是关于计算机方面的基本知识,对于每篇课文都有相应的词汇、注释和译文,便于读者自学。

## 计算机英语

黄国兴 李德宗 姚雪珍 编

责任编辑 赵 平

\*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京市顺义县天竺颖华印刷厂印刷

\*

开本:787×1092毫米 1/32 印张:6.125 字数:120千字

1994年4月第1版 1994年10月第2次印刷

印数:10100-20200册 定价:5.80元

ISBN 7-5053-2296-6/TP·640

# 出 版 说 明

为了适应电子科学技术飞速发展,提高电子工业技术工人素质,劳动部与电子工业部颁发了《电子工业工人技术等级标准》。根据新标准,电子工业部组织有关省市电子工业主管部门和企事业单位有关人员成立了“电子整机专业”,“家用电器产品维修专业”,“真空电子器件、接插件、继电器、绝缘介质专业”,“半导体器件及集成电路专业”,“计算机专业”,“磁性材料、电池专业”,“电子元件专业”共七个工人技术培训教材编审委员会。制定了 19 个专业、311 个工种的教学计划、教学大纲。并根据计划大纲的要求,制定了 1993~1995 年培训教材编审出版规划。列入规划的教材 78 种和相应的教学录像带若干种。

这套教材的编写是按“技术工人要按岗位要求开展技术等级培训”的要求,以文化课为专业课服务,专业课为提高工人实际操作和分析解决生产实际问题的能力服务为原则。教材既注重了电子工业技术工人要有一定专业理论知识的要求,又克服了以往工人培训教材片面强调理论的倾向;保证了必要的知识传授,又强调了技能培训和解决生产实际问题能力的培养。

这套教材在认真研究了 311 个工种的共性基础知识要求的基础上,编写了八种统编教材,供 311 个工种工人进行基础知识培训时选用;并以 19 个专业为基础,根据每个专业共性的专业知识、专业技能编写了 70 种教材供 311 个工种工人进行专业知识、专业技能培训时使用。

每种教材在反映初、中、高三级技术工人培训的不同要求

的基础上,注意了基础知识、专业知识、专业技能培训的系统性。因此,多数教材是初、中、高三级合在一起的,更好地体现由浅入深、由低及高的教学规律。

在教材编写上,针对工人培训的特点,突出教材的实用性、针对性,力求文字简练、通俗易懂、内容上紧密结合教学大纲要求,在讲授理论知识的同时还注意了对生产工艺和操作技能的要求,使教师易于施教,工人便于理解和操作。知识性强的教材,每章后配有练习题和思考题,以便巩固应掌握的知识。技能性强的教材,配有适当的技能训练课目,以便提高工人操作技能。在有关工艺和设备的教材中,主要介绍了通用性较强的内容和典型产品、设备,对于使用过类似教材的工厂企业,由于各自的产品、设备不同可自编相应的补充讲义与教材结合起来进行培训。另外,为适应技术发展、工艺改革、设备更新的需要,这套教材在编写中还注意了新技术、新工艺、新设备及其发展趋势,以拓宽工人的知识面。

参加这套教材编审工作的有北京、天津、上海、江苏、陕西五省市电子工业主管部门和河北、河南、山东、山西、辽宁、江西、四川、广东、湖南、湖北等十个省市的有关单位的专家、技术人员、教师等。在此谨向为此付出艰辛劳动的全体编审人员和各地、各单位支持这项工作的领导表示衷心感谢。

由于电子工业的迅速发展,这套教材的涉及面广、实用性强,加之编写时间仓促,教材中肯定有不妥之处,恳请使用单位提出宝贵意见。以便进一步修订,使之更加完善。

电子工业部

1993年7月

# CONTENTS 目 录

## Part One Text 课文

Lesson One	Digital Logic ..... 1
	数字逻辑
Lesson Two	Basic Logic Gates (part one) ... 7
	基本门电路(上)
Lesson Three	Basic Logic Gates (part two) ... 17
	基本门电路(下)
Lesson Four	The computer ..... 23
	计算机
Lesson Five	Number Systems (part one) ... 33
	数制(上)
Lesson Six	Number Systems (part two) ... 39
	数制(下)
Lesson Seven	Transmission Codes ..... 46
	传输码
Lesson Eight	File Store ..... 58
	文件储存
Lesson Nine	Peripherals ..... 65
	外部设备
Lesson Ten	Input / Output Bus and Input / Output Interface ..... 71
	输入 / 输出总线及输入 / 输出接口

Lesson Eleven	What the Operating System Does .....	78
	操作系统能做什么	
Lesson Twelve	The 'High-Level Language' ...	85
	高级语言	
Lesson Thirteen	What Is A Database Management System .....	92
	什么是数据库管理系统	
Lesson Fourteen	Computer and Information Management .....	98
	计算机和信息管理	
Lesson Fifteen	Software Development .....	105
	软件开发	
Lesson Sixteen	LANS and WANS and now MANS .....	116
	局域网、广域网及现在的市域网	
Lesson Seventeen	Factories of the 90s .....	123
	九十年代的工厂	
Lesson Eighteen	Computer Virus .....	129
	计算机病毒	
<b>Part Two</b>	<b>English Signals Pointed out on the Screech of the Computer</b>	
	计算机屏幕上出现的英文提示信息 .....	136
<b>Part Three</b>	<b>Vocabularies for Computer</b>	
	计算机常用词汇800条 .....	160

## Lesson one      Digital Logic

Digital logic systems are used to process information, and to control plant and machinery in the 'real world'. The use of digital systems and, in particular, computers, has become increasingly widespread over the past few decades, and the rate of expansion shows no signs of slowing down. This has occurred for two simple reasons. Firstly, information can be easily represented and manipulated in binary form, and is easily stored and manipulated using very simple electronic circuits, usually known as gates. Secondly, the ability to fabricate integrated circuits onto silicon chips, has meant that the cost of a digital logic system has continued to fall in real terms. The obvious example of such a logic systems is the small computer, or microcomputer as it is frequently known.

At the highest, or most abstract level a digital logic system may be regarded as a black box. This box has a number of inputs and a number of outputs. Each input and each output may take only one of two values. These values are referred to as TRUE and FALSE. When a logic system is implemented with an electronic circuit, voltages are used to represent these values. Usually a low voltage, close to 0 volts, is used to represent FALSE, and a high voltage, usually in the region 5–15 volts, is used to represent TRUE.

As with all sciences, the specification and design of digital logic systems requires a set of basic propositions and



rules, i.e. a mathematics. The mathematics we use for digital design is called Boolean algebra, or switching theory, and is based on Boole's work on the algebra of propositions published in 1854. Boolean algebra is also known as propositional calculus, and the form in which it is used was developed by Shannon in 1938.

### Words and Phrases

1. digital / 'didʒɪtl / a. 数字的
2. logic / 'lɒdʒɪk / n. 逻辑、逻辑性
3. process / 'prəʊses, 'prɒses / n. 过程 vt. 加工、处理
4. machinery / mə'ʃi:nəri / n. [总称]机器、机械
5. computer / kəm'pjʊ:tə / n. 计算机、电子计算机
6. increase / in'kri:s / vi. vt. n. 增加、增多、扩大
7. decade / 'dekeɪd / n. 十、十个组成的一组、十年
8. expansion / ɪks'pænjən / n. 张开、伸开、扩张
9. occur / ə'kɔ: / (occurred; occurring) vi. 发生、出现、存在
10. represent / ,reprɪ'zent / vt. 描绘、描述、代表
11. manipulate / mə'nɪpjuleɪt / vt. 熟练地使用、操作、处理
12. binary / 'baɪnəri / a. 二、双、复、二进制的
13. fabricate / 'fæbrɪkeɪt / vt. 制作、装配、组合
14. integrate / 'ɪntɪgreɪt / vt. 使结合 integrated circuit 集成电路
15. silicon / 'sɪlɪkən / n. 硅 silicon chip 硅片

16. microcomputer / 'maikroukəm'pju: tə / n. 微型计算机
17. frequent / 'fri: kwənt / a. 时常发生的、频繁的 -ly ad.
18. abstract / 'æbstrækt / a. 抽象的
19. voltage / 'vouldʒ / n. 电压、伏特数
20. region / 'ri: dʒən / n. 地区、地带; (科学的)领域、范围
21. specification / ,spesifi'keiʃən / n. 详细说明 [复数] 规格、说明书
22. require / ri'kwaɪə / vt. 需要、要求
23. proposition / ,prɒpə'zɪʃən / n. [逻辑]命题
24. Boolean / 'bu: lin / n. (人名)布尔
25. algebra / 'ældʒibrə / n. 代数学
26. switching theory 开关理论
27. calculus / 'kælkjuləs / n. 计算、演算
28. Shannon / 'ʃenən / n. (人名)香农
29. be used to (+v): 被用于(做……)
30. in particular: 特别地,尤其
31. widespread over: 广泛传播,广泛展开,遍及
32. be regarded as: 视作,认为
33. a number of: 许多,一些(后面可跟可数名词或不可数名词)
34. be referred to as: 被称作为
35. be implemented with: 由……执行,由……完成
36. be based on: 以……为基础,以……为依据

## Notes

1. 在此句中, be used to (+v.) 表示“被用于做……”。

注意区别另两种类似形式的不同意义和用法:

used to do sth: 过去常常做……

be used to (+n): 习惯于……

This instrument is used to measure the depth of the sound.

这台仪器被用来测量声音的高低。

Formerly people used to burn kerosene to get light.

过去人们经常点煤油来照明。

The students are used to making experiments at night now.

现在学生们已习惯于晚上做实验了。

to process…和to control…是两个并列的动词不定式短语,作主语补足语,补充说明 digital logic systems。

2. 此句中, in particular, 是插入语, 表示强调。

“the use of …”一句与“the rate of …”一句是相并列的。

widespread是一个合成词,由wide与spread合成。

此句中widespread是过去分词,作become的表语。

few(几乎没有几个),后面加可数名词,表示否定的意思,而 a few(几个),同样跟可数名词,表示肯定的意思。

试比较: Few people know it. 几乎没有人知道这一点。

A few people know it. 有几个人知道这一点。  
在本句中, few decades, 可解释为“短短的几十年”;  
slowing down: 是动名词短语, 作介词of的宾语。

3. 此句中, using very simple electronic circuits是现在分词短语, 作状语, 表示条件。

usually known as gates 中, usually是副词, 修饰 known as gates, 而 known as gates 是过去分词短语, 作定语, 修饰 electronic circuits, 意思是“被称作为”, 与 called as 有相同的意思。

4. 此句中, in which it is used是由“介词+关系代词”所引导的定语从句, 此类句子介词后只能用 which, 在口语中一般都把介词放到从句后面去, which it is used in. 这时关系代词可用 which 也可用 that, 但介词后不能用 that。

## 第一课 数字逻辑

在现实世界中, 数字逻辑系统用于处理信息, 控制工厂及机器的运转。在过去短短的几十年中, 数字系统的使用, 尤其是计算机的使用, 已日益广泛地向各个领域渗透, 其扩展的速度至今毫无减慢的迹象。这种现象的发生有两个简单的原因: 第一, 信息不但很容易以二进制表示和处理, 而且能用简单的线路(通常称为“门”电路)进行储存和处理。第二, 将集成电路封装在硅片上的能力表明: 数字逻辑系统的实际成本一直在持续不断地下降。这类逻辑系统明显的例子是人们所

熟悉的小型计算机及微型计算机。

从最抽象的角度来看,可以把数字逻辑系统视为黑盒子。该黑盒子有许多输入和输出,每一输入或输出只能传送一个两值信号,而这两个值即为 TRUE(真)和 FALSE(假)。当逻辑系统由电子线路实现时,可用电压来表示这两个值。通常用一个接近 0 伏的低电压表示 FALSE,用一个 5~15 伏的高电压表示 TRUE。

象所有的科学一样,数字逻辑系统的说明及设计需要一套基本定理和规则,这也意味着需要数字理论。用于该类系统设计的数字称为布尔代数或开关理论。它是以 1854 年出版的布尔著作中的命题代数为基础的。布尔代数也以命题演算所著称,它目前所使用的演算方式,是 1938 年由香农所发展的。

## Lesson Two Basic Logic Gates (part one)

Digital logic circuits are divided into two distinct types, namely combinatorial logic circuits and sequential logic circuits. However, most real logic systems, such as computers, are a mixture of both combinatorial and sequential logic circuits. A combinatorial circuit is one in which the values of the outputs depend only on the present values of the input. Combinatorial circuits have no memory, so the outputs can not depend on the order in which the values of the inputs are changed, or on any previous values of those inputs. A sequential circuit, on the other hand, is one in which the outputs depend not only on the present values of the inputs, but also on the past values of those inputs, i.e. on the history of the circuit. Clearly a sequential circuit requires a memory to store the necessary history of the circuit. But here, only the combinatorial circuits are discussed.

Figure 2.1a shows the simplest possible combinatorial logic circuit: a single input labelled A and a single output labelled Z. The circuit of Fig. 2.1a is known as a buffer. The output Z always has the same logic level as the input, so the Boolean equation <sup>1</sup> defining its behaviour is  $Z = A$  and is read as Z equals A.

Figure 2.1b is an inverter or NOT gate. The value of the output is always the inverse, <sup>2</sup> or complement, of the input. The logic equation is  $Z = \bar{A}$  and is read as Z equals NOT A. The line, or bar, over the A implies NOT, in this case NOT

A. The logic symbol of the NOT gate is really composed of two parts, a triangle and a circle, or bubble as it is usually called. <sup>3</sup> The triangle means that the element is a buffer and the bubble means that this buffer is actually an inverting buffer so that the element performs the NOT function. Note for the present that the bubble may be drawn either on the input to, or the output from, the buffer. The simplicity of the NOT gate hides its importance: <sup>5</sup> without the NOT gate, the design of many useful digital logic systems would not be possible.

<sup>6</sup> Slightly more complicated are the combinatorial circuits shown in Fig.2.2, each of which has two inputs, labelled A and B, and one output, labelled Z. <sup>7</sup> Since each of the inputs A and B can take one of only two possible values, 0 or 1, there are 4 possible input combinations. There are five fundamental elements, namely AND, OR, NAND, NOR and EOR. Figure 2.2 shows the accepted logic symbols of these five basic elements, or gates <sup>8</sup> as they are frequently known. Each circuit is defined by a Boolean logic equation and by a truth table; both definitions are shown in Fig.2.2. These gates will be considered in more detail in next part.



$$Z = A$$

Buffer

A	Z
0	0
1	1

Fig 2.1a



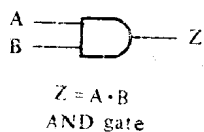
$$Z = \bar{A}$$

NOTgate

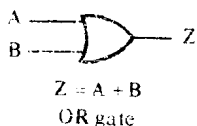
A	Z
0	1
1	0

Fig 2.1b

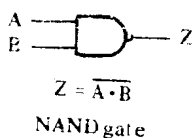




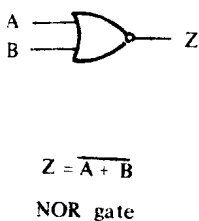
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



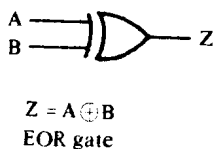
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1



A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0



A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Fig 2.2