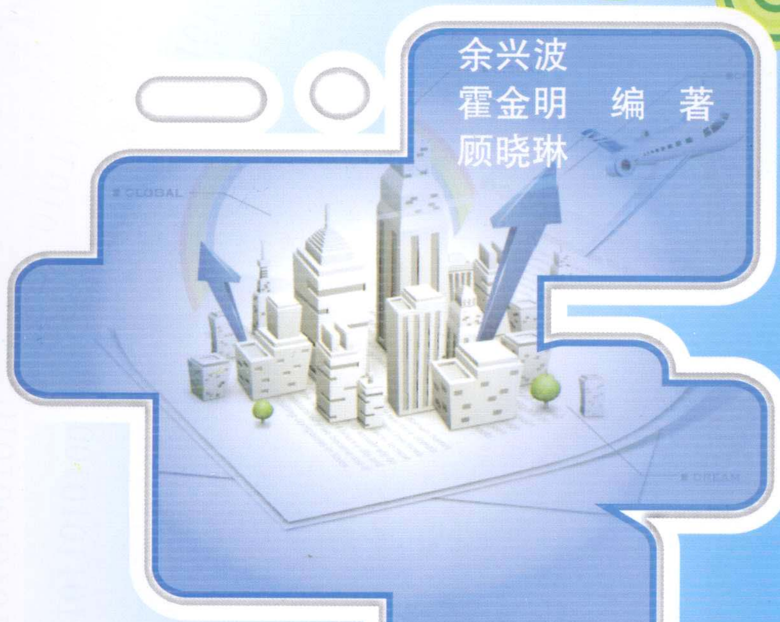
21世纪全国本科院校电气信息类创新型应用人才培养规划教材

电气信息工程专业英语

余兴波
霍金明 编著
顾晓琳



精心挑选英文原文和文章
细心讲解专业词汇及短语
耐心翻译长复句子与难点

提供 电子课件 www.pup6.com
提供 习题参考答案 www.pup6.com
赠送 参考译文 www.pup6.com

北京大学出版社
PEKING UNIVERSITY PRESS

21世纪全国本科院校电气信息类创新型应用人才培养规划教材

电气信息工程专业英语

余兴波　霍金明　顾晓琳　编　著

北京大学出版社
PEKING UNIVERSITY PRESS
1902

内 容 简 介

本书以电气信息工程专业知识为背景，通过英文课文的阅读，学习用科技英文术语和观念理解电气信息工程的专业知识和概念，对专业英语词汇、短语等及词法、语法、句型等进行详细的讲解，对专业文献的翻译进行详细的分析。本书按照知识由浅入深，共分为 12 个单元，主要内容涉及计算机、数字电子技术、模拟电子技术、通信工程、电力电子、电子元器件等方面。

在参考了大量同类著作和国外期刊的基础上，本书精心挑选了具有代表性的英文原文文章，书中每个单元均包括正文和阅读两部分，其中正文部分附有课文中出现的专业词汇、短语、长难句的详细讲解和练习题，且两部分均有参考译文，能够帮助读者更好地掌握所学内容，还可以使学生适应毕业后到合资、外企等单位的工作环境。

图书在版编目(CIP)数据

电气信息工程专业英语/余兴波，霍金明，顾晓琳编著. —北京：北京大学出版社，2013.8

(21 世纪全国本科院校电气信息类创新型应用人才培养规划教材)

ISBN 978-7-301-22920-0

Ⅰ.①电…　Ⅱ.①余…②霍…③顾…　Ⅲ.①电子技术—英语—高等学校—教材②信息工程—英语—高等学校—教材　Ⅳ.①H31

中国版本图书馆 CIP 数据核字(2013)第 178232 号

书　　　名：电气信息工程专业英语

著作责任者：余兴波　霍金明　顾晓琳　编著

策 划 编 辑：程志强

责 任 编 辑：程志强

标 准 书 号：ISBN 978-7-301-22920-0/TP・1298

出 版 发 行：北京大学出版社

地　　　址：北京市海淀区成府路 205 号　100871

网　　　址：http://www.pup.cn　新浪官方微博：@北京大学出版社

电 子 信 箱：pup_6@163.com

电　　　话：邮购部 62752015　发行部 62750672　编辑部 62750667　出版部 62754962

印　刷　者：**北京飞达印刷有限责任公司**

经　销　者：新华书店

787 毫米×1092 毫米　16 开本　10 印张　228 千字

2013 年 8 月第 1 版　2013 年 8 月第 1 次印刷

定　　　价：26.00 元

前　　言

在市场竞争日趋激烈的今天，民办高校、独立学院必须要走出一条属于自己的教育之路。引进职业化教育并创新实践教育体系，是保证民办高校、独立学院走出自己应用型本科特色道路的有效措施。

为此，作为“大学应用本科实践教学与职业化教育系列教材”，编者 2010 年组织编写了一套《机械信息工程专业英语教程》，主编余兴波，分 I、II 两册，2011 年 3 月由吉林出版集团有限责任公司出版，缓解了独立学院机械工程、信息工程专业学生专业英语教材短缺的局面。该教材自出版使用以来，收到了较好的教学效果，因为作为教材的课文均是经过精心挑选且具有代表性的英文原文，在教材内容和难度上，适合独立学院学生的英语水平，循序渐进，实践练习，容易入门，且激发学生学习兴趣，教材体现了一切为学生发展的宗旨，以学生就业为中心，本校机械、信息类学生的就业实例证明，专业外语的学习提高了学生在独资、合资、外企等企业的就业能力。

由于教学需求，并且按照学校领导关于教材建设的要求和安排，在北京大学出版社的支持和鼓励下，编者认真研究并采纳了一些学校的师生在使用教材过程中提出的宝贵意见，同时将编写教材中存在的错误进行了逐一修订。修订内容分为“正文”、“阅读课文”的专业译文、注释、Word-Study、Sentence Patterns 和 Exercises 等。全书由 12 篇课文组成。修订再版的教材，其书稿的质量由修订者“文责自负”。

本书的修订工作由长春光华机械工程学院副院长余兴波教授、霍金明（常务）、顾晓琳（常务）、任婷共同完成。《电气信息工程专业英语》教材修订体系分工如下：

霍金明负责编写第 1、2 单元和第 5、6 单元；

余兴波、于大海负责编写第 3、4 单元和第 7、10 单元；

余兴波、王金莉负责编写第 8、9 单元和第 11、12 单元；

顾晓琳负责编写第 1～6 单元 Word-Study, Sentence Patterns, Exercises, Translation；

任婷负责编写第 7～12 单元 Word-Study, Sentence Patterns, Exercises, Translation。

全书的统稿、校稿工作由余兴波、霍金明、顾晓琳、任婷共同完成。

本书在编写过程中得到了学院董事长、院长等各位领导的大力支持和帮助；学院公共外语教研部主任滕玉梅教授审阅了书稿；北京大学出版社的编辑在教材的策划、修订、编写、版式设计等方面做了大量工作，在此一并表示感谢！

由于编写水平有限以及时间紧迫，有些是属专业英语课程实践教学体系改革方面所做的点滴尝试，其中难免存在不当和疏漏之处，敬请读者批评指正。

编　者

2013 年 5 月于长春

目　录

Unit 1 1

Text 1: Octal and Hexadecimal Numbers 1

Reading 1: Positional Number Systems 8

Unit 2 10

Text 2: Floating-Point Numbers 10

Reading2: ANSI/IEEE-754 Format 15

Unit 3 17

Text 3: Diodes 17

Reading 3: Power Converters 22

Unit 4 25

Text 4: Digital Waveforms 25

Reading 4: A Digital Waveform Carries Binary Information 31

Unit 5 34

Text 5: Basic Logic Operations 34

Reading 5: The Nyquist Criterion: Relating Bandwidth to Symbol Rate 40

Unit 6 42

Text 6: Review of Four Basic Analog Filter Approximations 42

Reading 6: Summary of Memory Byte Pools 47

Unit 7 49

Text 7: The Growth of Bandwidth and the Digital Revolution 49

Reading 7: The Transmission of Images 54

Unit 8 56

Text 8: Superposition 56

Reading 8: Linear Systems 64

Unit 9 66

Text 9: Low Pass Filters 66

Reading 9: Active Filters 73

Unit 10 75

Text 10: RFID Systems 75

Reading 10: Auto-ID Technologies 81

Unit 11 82

Text 11: Fixed-Function Integrated Circuits 82

Reading 11: Summary of Memory Block Pools 89

Unit 12 91

Text 12: Little Endian and Big Endian Configuration 91

Reading 12: Processes 97

附录 1　索引 99

附录 2　参考答案 109

附录 3　参考译文 117

参考文献 150

Unit 1

Text 1: Octal and Hexadecimal Numbers

Radix 10 is important, because we use it in everyday business, and radix 2 is important, because binary number can be processed directly by digital circuits. Numbers in other radices are not often processed directly, but may be important for documentation or other purposes. In particular, the octal and hexadecimal numbers can provide convenient shorthand representations for multibit numbers in a digital system.

The octal number system uses radix 8, while the hexadecimal number system uses radix 16. Table 1-1 shows the binary integers from 0 to 1111 and their octal, decimal, and hexadecimal equivalents. The octal system needs 8 digits, so it uses digits 0~7 of the decimal system. The hexadecimal system needs 16 digits, so it supplements decimal digits 0~9 with the letters A~F.

The octal and hexadecimal number systems are useful for representing multibit numbers, because their radices are powers of 2. Since a string of three bits can take on eight different combinations, it follows that each 3-bit string can be uniquely represented by one octal digit, according to the third and fourth columns of Table 1-1. Likewise, a 4-bit string can be represented by one hexadecimal digit according to the fifth and sixth columns of the Table 1-1.

Table 1-1 Binary, decimal, octal and hexadecimal numbers

Binary	Decimal	Octal	3-Bit String	Hexadecimal	4-Bit String
0	0	0	000	0	0000
1	1	1	001	1	0001
10	2	2	010	2	0010
11	3	3	011	3	0011
100	4	4	100	4	0100
101	5	5	101	5	0101
110	6	6	110	6	0110
111	7	7	111	7	0111
1000	8	10	—	8	1000
1001	9	11	—	9	1001
1010	10	12	—	A	1010
1011	11	13	—	B	1011
1100	12	14	—	C	1100

续表

Binary	Decimal	Octal	3-Bit String	Hexadecimal	4-Bit String
1101	13	15	—	D	1101
1110	14	16	—	E	1110
1111	15	17	—	F	1111

Thus, it is very easy to convert a binary number to octal number. Starting at the binary point and working left, we simply separate the bits into groups of three and replace each group with the corresponding octal digit.

$$100011001110_2=100\ 011\ 001\ 110_2=4316_8$$

$$11101101110101001_2=011\ 101\ 101\ 110\ 101\ 001_2=355651_8$$

The procedure for binary to hexadecimal conversion is similar, except we use groups of four bits.

$$100011001110_2=1000\ 1100\ 1110_2=8CE_{16}$$

$$11101101110101001_2=01\ 1101\ 1011\ 1010\ 1001_2=1DBA9_{16}$$

In these examples we have freely added zeroes on the left to make the total number of bits a multiple of 3 or 4 as required.

If a binary number contains digits to the right of the binary point, we can convert them to octal or hexadecimal number by starting at the binary point and working right. Both the left-hand and right-hand sides can be padded with zeroes to get multiples of three or four bits, as is shown in the example below:

$$10.1011001011_2=010\ .\ 101\ 100\ 101\ 100_2=2.5454_8$$

$$=0010\ .\ 1011\ 0010\ 1100_2=2.B2C_{16}$$

Converting in the reverse direction, from octal or hexadecimal to binary, is also very easy. We simply replace each octal or hexadecimal digit with the corresponding 3-bit or 4-bit string, as shown below:

$$1357_8=001\ 011\ 101\ 111_2$$

$$2046.17_8=010\ 000\ 100\ 110.001\ 111_2$$

$$BEAD_{16}=1011\ 1110\ 1010\ 1101_2$$

$$9F.46C_{16}=1001\ 111\ .0100\ 0110\ 1100_2$$

The octal number system was quite popular 25 years ago because of certain minicomputers that had their front-panel lights and switches arranged in groups of three. However, the octal number system is not used much today because of the preponderance of machines that process 8-bit bytes. It is difficult to extract individual byte values in multi-byte quantities in the octal representation. For example, what are the octal values of the four 8-bit bytes in the 32-bit number with octal representation 12345670123_8?

In the hexadecimal system, two digits represent an 8-bit byte, and 2n digits represent an n-byte word; each pair of digits constitutes exactly one byte. For example, the 32-bit hexadecimal number $5678ABCD_{16}$ consists of four bytes with values 56_{16}, 78_{16}, AB_{16} and CD_{16}. In this context, a 4-bit hexadecimal digit is sometimes called a nibble; a 32-bit (4-byte) number has eight nibbles. Hexadecimal numbers are often used to describe a computer's memory address space. For example, a computer with 16-bit addresses might be described as having read/write memory installed at addresses $0\sim EFFF_{16}$ and read-only memory at addresses $F000\sim FFFF_{16}$. Many computer programming languages use the prefix "0x" to denote a hexadecimal number. For example, 0xBFC0000[1].

Words and Expressions

1.	radix	['rediks]	*n.*	①[数]根 ②[统]基数 ③[植]根
2.	binary	['bainəri]	*a.*	①二进制的 ②仅基于两个数字的，二元的
			n.	二进制数
3.	process	['prəuses]	*vt.*	①加工；处理 ②数据处理
4.	digital	['didʒitəl]	*a.*	数码的，数字信息系统的，数字的
			n.	数字电视，数字仪表
5.	circuit	['sə:kit]	*n.*	①电路，线路 ②环行，环行道
6.	purpose	['pə:pəs]	*n.*	①目的；意图 ②作用；用途
7.	provide	[prə'vaid]	*vt.*	提供，供应
8.	convenient	[kən'vi:njənt]	*a.*	①方便的，便利的 ②实用的
9.	representation	[ˌreprizen'teiʃən]	*n.*	①表示 ②代表
10.	multibit	[mʌltibit]	*a.*	多位的
11.	hexadecimal	['heksədesiməl]	*a.*	十六进制的
12.	integer	['intidʒə]	*n.*	整数
13.	octal	['ɔktl]	*a.*	八进制的
14.	equivalent	[i'kwivələnt]	*n.*	①相等的东西；等量 ②[数学]等价；等值
			a.	①相等的，相当的 ②等价的；等值的
15.	decimal	['desiməl]	*a.*	十进位的，小数的
			n.	小数
16.	supplement	['sʌplimənt]	*vt.*	增补，补充
			n.	①增补(物) ②(报纸的)增刊

17.	represent	[repri'zent]	*vt.*	①代表，表示 ②相当于
18.	uniquely	[ju:'ni:kli]	*ad.*	独特地，唯一地
19.	combination	[kɔmbi'neiʃən]	*n.*	①结合，组合 ②联合体，组合物
20.	column	['kɔləm]	*n.*	①[计算机]纵列，纵向排列 ②纵行
21.	convert	[kən'və:t]	*vt.& vi.*	(使)转变，(使)转化
22.	replace	[ri'pleis]	*vt.*	①代替，替换 ②(用……)替换
23.	separate	['sepəreit]	*vt.& vi.*	分开，隔离
			a.	个别的，各自的
24.	corresponding	[kɔri'spɔndiŋ]	*a.*	①对应的，相应的 ②符合的，一致的
25.	procedure	[prə'si:dʒə]	*n.*	[计算机]过程，程序，步骤；算法
26.	conversion	[kən'və:ʃən]	*n.*	变换，转化
27.	require	[ri'kwaiə]	*vt.& vi.*	①需要 ②要求；规定
28.	pad	[pæd]	*vt.*	填塞；封填
			n.	一捆；一束
29.	direction	[di'rekʃən]	*n.*	①方向，趋向 ②指示，说明
30.	panel	['pænəl]	*n.*	(汽车或其他机械的)控制板，仪表盘
31.	switch	[switʃ]	*n.*	开关
			vt.& vi.	转变，改变
32.	arrange	[ə'reindʒ]	*vt.& vi.*	安排；准备
33.	preponderance	[pri'pɔndərəns]	*n.*	优势，主体
34.	extract	[iks'trækt]	*vt.*	①提取，提炼 ②摘录；选录
35.	individual	[indi'vidjuəl]	*a.*	个别的，独特的
			n.	个人
36.	constitute	['kɔnstitju:t]	*vt.*	构成，组成
37.	nibble	['nibl]	*n.*	[计算机]半字节
38.	describe	[di'skraib]	*vt.*	①描述 ②描绘，叙述，形容
39.	install	[in'stɔ:l]	*vt.*	①安装 ②安顿，安置
40.	denote	[di'nəut]	*v.*	表示；意指
41.	prefix	['pri:fiks]	*n.*	前缀
			vt.	在……前加前缀
42.	a string of			一串，一行，一列
43.	take on			承担，接受，从事

44. according to 按照，根据
45. because of 因为，由于
46. consist of 由……组成

Notes

(1) Radix 10 is important because we use it in everyday business, and radix 2 is important because binary numbers can be processed directly by digital circuits.

句中的 because 引导原因状语从句。because 引导的原因状语从句一般放于主句之后，because 表示直接原因，语气最强，最适合回答以 why 引导的疑问句。

(2) The octal number system uses radix 8, while the hexadecimal number system uses radix 16.

句中 while 作并列连词，译为“而，却”，表对照关系。

(3) Since a string of three bits can take on eight different combinations, it follows that each 3-bit string can be uniquely represented by one octal digit, according to the third and fourth columns of Table 1-1.

句中的 since 引导原因状语从句。since 引导的原因状语从句一般放于主句之前表示已知的、显然的理由(通常被翻译成“既然”)，较为正式，语气比 because 弱。

(4) The octal number system was quite popular 25 years ago because of certain minicomputers that had their front-panel lights and switches arranged in groups of three.

句中，关系代词 that 引导一个定语从句修饰 certain minicomputers，关系代词 that 在从句中作主语。

Word-Study

I. Binary

(1) (computing, mathematics) using only 0 and 1 as a system of numbers.

- the binary system
- binary arithmetic

(2) (technical) based on only two numbers; consisting of two parts.

- binary codes / numbers 二进制代码/数
- One million binary bits. 两个，双(两)个的东西，尤指双星。

II. Process, Processing

(1) **Process** means to treat raw material, food ,etc. in order to change it, preserve it etc.

(2) (computing) to perform a series of operations on data in a computer.

(3) **Processing** is a course on color photograph and processing.

Most of the food we buy is processed in some way.

processed cheese

a sewage processing plant

the food processing industry

III. Convenient, Convenience, Conveniently

(1) **Convenient** usually means easy or quick to do; not causing problems.

(2) **Convenience** is the quantity of being useful, easy or suitable for sb.

(3) **Conveniently** usually means easily or quickly to do; not causing problems.

It is very convenient to pay by card.

I'll call back at a more convenient time.

We have provided seats for the convenience of our customers.

In this resort you can enjoy all the comfort and convenience of modern tourism.

The report can be conveniently divided into three main sections.

The hotel is conveniently situated close to the beach and the shops.

IV. Representation, Represent

(1) **Representation** is the act of presenting sb./sth. in a particular way; something that shows or describes sth.

(2) **Represent** means[often passive] to be a member of a group of people and act or speak on their behalf at an event, a meeting, etc.

the negative representation of single mothers in the media

The snake swallowing its tail is a representation of infinity.

The competition attracted over 500 contestants representing 8 different countries.

The president was represented at the ceremony by the vice-president.

Sentence Patterns

Formal Subject "it"

It is { wrong to tell a lie. / no use arguing about it. / uncertain who will come.

It + be + adj.+ to do sth. / doing/ that …	(1) It is very important to learn a foreign language. (2) It is useless crying over the spilt milk. (3) It was really surprising that she married a man like that.
It + be +n. (phr.) + doing / that …	(1) It is no good telling lies. (2) It is a pity that you didn't go to see the film yesterday. (3) It is a truth that there would be no new China without the Communist Party.

续表

It + be + past participle + that ... (usually say, hope, think, suppose, expect, report, know, believe, decide, etc.)	(1) It is said that they have invented a new type of computer. (2) It is believed that China will become one of the strongest countries in the world. (3) It was reported that more than 170 thousand people died in the 2004 tsunami.
It + takes + (sb.) + some time + to do sth.	(1) It took me some time to read the reading materials. (2) How long does it take you to go to Beijing from Qingdao by train?
It + seems / appears / happens, etc. (intransitive verb) + that ...	(1) It seems that he enjoys pop songs very much. (2) It appears that Tom might change his mind.
Interrogative Sentences	(1) Does it matter much that they won't come tomorrow? (2) Is it true that he will go abroad next week?

Exercises

Ⅰ. Answer the following questions with the information from the passage.

1. Why are radix 10 and radix 2 important?
2. What the octal number system and the hexadecimal system use?
3. How many digits do the octal system and the hexadecimal system need?
4. What can each 3-bit string and a 4-bit string be represented by?
5. How can we convert a binary number to octal?
6. How can we convert a binary number to octal or hexadecimal, if a binary number contains to the right of the binary point?
7. What is converting in the reverse direction, from octal or hexadecimal?
8. Why was the octal number system quite popular 25 years ago?
9. What do two digits represent in the hexadecimal system?
10. What do many computer programming languages use to denote a hexadecimal number?

Ⅱ. Complete the following sentences with phrases and expressions given below.

A. be useful for **B.** replace ... with... **C.** convert ... to... **D.** consist of **E.** according to

1. We've __________ the old adding machine __________ a computer.
2. All electronic computers __________ five units although they are of different kinds.
3. Courses taken that would __________ computer programming are Computer science, systems design and analysis, FORTRAN programming, PASCAL programming, operating systems, systems management.
4. __________ expert opinions, they gave up the experiment immediately.
5. __________ code __________ ordinary language.

Ⅲ. Fill in the blanks with the words given below. Change the form when necessary.

process	convenient	uniquely	require	arrange	extract

1. The computer enables people to communicate with each other more __________.
2. Your __________ that she wait till next week is reasonable.
3. How fast does the computer __________ the data?
4. A computer word that specifies to __________ some parts of another computer word.
5. I have __________ that one of my staff will meet you at the airport.
6. He's caught the __________ opportunity.

Ⅳ. Choose an appropriate translation from Column B for each of the words in Column A.

		Column A		Column B
(　)	1.	binary number	A.	基数
(　)	2.	octal number	B.	十六进制
(　)	3.	hexadecimal number	C.	半字节
(　)	4.	decimal number	D.	八进制
(　)	5.	Radix	E.	十进制
(　)	6.	Nibble	F.	电路
(　)	7.	multi-byte	G.	二进制
(　)	8.	Circuit	H.	多字节

Reading 1: Positional Number Systems

The traditional number system that we learned in school and use every day in business is called a positional number system. In such a system, a number is represented by a string of digits where each digit position has an associated weight. The value of a number is a weighted sum of the digits, for example

$$1734=1\times1000+7\times100+3\times10+4\times1$$

Each weight is a power of 10 corresponding to the digit's position. A decimal point allows negative as well as positive powers of 10 to be used, for example

$$5185.68=5\times1000+1\times100+8\times10+5\times1+6\times0.1+8\times0.01$$

In general, a number D of the form $d_1d_0.d_{-1}d_{-2}$ has the value

$$D=d_1\times10^1+d_0\times10^0+d_{-1}\times10^{-1}+d_{-2}\times10^{-2}$$

Here 10 is called the base or radix of the number system. In a general positional number system, the radix may be any integer $r\geqslant2$ and a digit in position i has weight r^i. The general form of a number in such a system is

$$d_{p-1}\,d_{p-2}\ldots d_1d_0.d_{-1}d_{-2}\ldots d_{-n}$$

where there are p digits to the left of the point and n digits to the right of the point, "." is called the radix point. If the radix point is missing, it is assumed to be to the right of the rightmost digit. The value of the number is the sum of each digit multiplied by the corresponding power of the radix

$$D=\sum_{i=-n}^{p-1} d_i \cdot r^i$$

Except for possible leading and trailing zeroes, the representation of a number in a positional number system is unique. (Obviously, 0185.6300 equals 185.63, and so on.) The leftmost digit in such a number is called the high-order or most significant digit; the rightmost is called the low-order or least significant digit.

Digital circuits have signals that are normally in one of only two conditions—low or high, charged or discharged, off or on. The signals in these circuits are interpreted to represent binary digits (or bits) that have one of two values, 0 and 1. Thus, the binary radix is normally used to represent numbers in a digital system. The general form of a binary number is

$$b_{p-1}\, b_{p-2} \ldots b_1 b_0 . b_{-1} b_{-2} \ldots b_{-n}$$

and its value is

$$B=\sum_{i=-n}^{p=-1} b_i \cdot 2^i$$

In a binary number, the radix point is called the binary point. When dealing with binary and other nondecimal numbers, we often use a subscript to indicate the radix of each number, unless the radix is clear from the context. Examples of binary numbers and their decimal equivalents are given below.

$$10011_2=1\times16+0\times8+0\times4+1\times2+1\times1=19_{10}$$
$$100010_2=1\times32+0\times16+0\times8+0\times4\times1+2+0\times1=34_{10}$$
$$101.001_2=1\times4+0\times2+1\times1+0\times0.5+0\times0.25+1\times0.125=5.125_{10}$$

The leftmost bit of a binary number is called the high-order or most significant bit (MSB); the rightmost is the low-order or least significant bit (LSB)[1].

Unit 2

Text 2: Floating-Point Numbers

Floating-point notation can be used conveniently to represent both large as well as small fractional or mixed numbers. This makes the process of arithmetic operations on these numbers relatively much easier. Floating-point representation greatly increases the range of numbers, from the smallest to the largest, that can be represented using a given number of digits. Floating-point numbers are in general expressed in the following form

$$N = m \times b^e \tag{2.1}$$

Where m is the fractional part, called the significant or mantissa, e is the integer part, called the exponent and b is the base of the number system or numeration. Fractional part m is a p-digit number of the form (±d.dddd …dd), with each digit d being an integer between 0 and $b-1$ inclusive. If the leading digit of m is nonzero, then the number is said to be normalized.

According to equation (2.1), in the case of decimal, hexadecimal and binary number systems can be written as follows:

Decimal system

$$N = m \times 10^e \tag{2.2}$$

Hexadecimal system

$$N = m \times 16^e \tag{2.3}$$

Binary system

$$N = m \times 2^e \tag{2.4}$$

For example, decimal numbers 0.0003754 and 3754 will be represented in floating-point notation as 3.754×10^{-4} and 3.754×10^{3} respectively. A hexadecimal number 257. ABF will be represented as $2.57\text{ABF}\times16^{2}$. In the case of normalized binary numbers, the leading digit, which is the most significant bit, is always '1' and thus it does not need to be stored explicitly.

Similarly, while expressing a given mixed binary number as a floating-point number, the radix point is so shifted as to have the most significant bit immediately to the right of the radix point as a '1'. Both the mantissa and the exponent can have a positive or a negative value.

The mixed binary number $(110.1011)_2$ will be represented in floating-point notation as $.1101011\times2^{3}$ = .1101011e+0011. Here, .1101011 is the mantissa and e+0011 implies that the exponent is +3. As another example, $(0.000111)_2$ will be written as .111e−0011, with .111 being the mantissa and e−0011 implying an exponent of −3. Also, $(-0.00000101)_2$ may be written as $-.101\times2^{-5}$=−.101e−0101, where −.101 is the mantissa and e−0101 indicates an exponent of −5. If we wanted to represent the mantissas by using eight bits, then .1101011 and .111 would be represented as .11010110 and .11100000[2].

Words and Expressions

1.	notation	[nəu'teiʃən]	*n.*	标记法；(数学、科学和音乐中的)符号
2.	fractional	['frækʃənəl]	*a.*	①微不足道的，极小的，极少的 ②分数的；小数的
3.	arithmetic	[ə'riθmətik]	*n.*	①算术　②计算　③算术运算；四则运算
4.	relatively	['relətivli]	*ad.*	相当程度上；相当地；相对地
5.	range	[reindʒ]	*n.*	(变动或浮动的)范围，界限，区域
6.	significant	[sig'nifikənt]	*n.*	①有意义的事物　②象征，标志
			a.	①有重大意义的　②显著的
7.	numeration	[nju:mə'reiʃən]	*n.*	计算，编号，读数法
8.	inclusive	[in'klu:siv]	*a.*	①包括……的　②范围广泛的
9.	equation	[i'kweiʃən]	*n.*	①方程式，等式　②相等，平衡
10.	respectively	[ri'spektivli]	*ad.*	各自地，分别地
11.	normalized	['nɔ:məlaizd]	*a.*	规范化的，规格化的
12.	explicitly	[ik'splisitli]	*ad.*	明白地，明确地
13.	imply	[im'plai]	*vt.& vi.*	①暗示，暗指　②说明，表明
14.	indicate	['indikeit]	*vt.*	表明，表示
15.	as well as			(除……之外)也，又
16.	in the case of			就……来说；在……的情况下

Notes

(1) Floating-point notation can be used conveniently to represent both large as well as small fractional or mixed numbers.

句中短语 be used to do 意思是“被用来做……”，as well as 意思是“也”。