时代教育·国外高校优秀教材精选

数字设计基础

（双语教学版）

The Essence of Digital Design

（英）巴里·威尔金森（Barry Wilkinson） 著
（英）史蒂文·奎克利（Steven Quigley） 增补
江 捷 注释

机械工业出版社
CHINA MACHINE PRESS

时代教育·国外高校优秀教材精选

数字设计基础

（双语教学版）

THE ESSENCE OF DIGITAL DESIGN

（英）巴里·威尔金森（Barry Wilkinson） 著
（英）史蒂文·奎克利（Steven Quigley） 增补
江 捷 注释

机 械 工 业 出 版 社

The original work by Prof. Barry Wilkinson
Additional material for this edition by Dr. Steven Quigley
Chinese note by Prof. Jiang Jie

本书是专为我国高校双语教学设计的一本教材,是在 Barry Wilkinson 所著的《数字设计基础》一书的基础上,结合目前数字设计技术的发展和双语教学的需要,增加了 Steven Quigley 提供的 VHDL 硬件描述语言及其典型设计实例,补充了江捷对重点、难点内容的汉语注释。

本书共分 12 章,主要讨论了逻辑代数基础、逻辑门、组合逻辑电路、触发器与计数器、时序逻辑电路、可编程逻辑器件、逻辑电路测试、VHDL 硬件描述语言基础和 VHDL 程序设计等内容。本书简明扼要,示例丰富,各章后附有适量习题、推荐阅读文献和习题解答,配合汉语注释,易于阅读。

本书可作为高等学校电子信息类、电气信息类各专业本科生"数字电子技术"双语教学的教科书,也可作为相关专业"专业英语"教学的参考书。

图书在版编目(CIP)数据

数字设计基础:双语教学版/(英)威尔金森(Wilkinson, B.)著;江捷注释.—北京:机械工业出版社,2007.10
时代教育.国外高校优秀教材精选
ISBN 978-7-111-22838-7

Ⅰ.数… Ⅱ.①威…②江… Ⅲ.数字电路—电路设计—高等学校—双语教学—教材 Ⅳ. TN79

中国版本图书馆 CIP 数据核字(2007)第 178040 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
责任编辑:王保家 版式设计:霍永明
封面设计:饶 薇 责任印制:洪汉军
北京振兴源印务有限公司印刷厂印刷
2008 年 1 月第 1 版第 1 次印刷
210mm×239mm · 13.666 印张 · 543 千字
标准书号:ISBN 978-7-111-22838-7
定价:32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话:(010)68326294
购书热线电话:(010)88379639 88379641 88379643
编辑热线电话:(010)88379711
封面无防伪标均为盗版

出版说明

随着我国加入 WTO，国际间的竞争越来越激烈，而国际间的竞争实际上也就是人才的竞争、教育的竞争。为了加快培养具有国际竞争力的高水平技术人才，加快我国教育改革的步伐，国家教育部近来出台了一系列倡导高校开展双语教学、引进原版教材的政策。以此为契机，机械工业出版社陆续推出一系列国外影印版教材，其内容涉及高等学校公共基础课，以及机、电、信息领域的专业基础课和专业课。

引进国外优秀原版教材，在有条件的学校推动开展英语授课或双语教学，自然也引进了先进的教学思想和教学方法，这对提高我国自编教材的水平，加强学生的英语实际应用能力，使我国的高等教育尽快与国际接轨，必将起到积极的推动作用。

为了做好教材的引进工作，机械工业出版社特别成立了由著名专家组成的国外高校优秀教材审定委员会。这些专家对实施双语教学做了深入细致的调查研究，对引进原版教材提出了许多建设性意见，并慎重地对每一本将要引进的原版教材一审再审，精选再精选，确认教材本身的质量水平以及权威性和先进性，以期所引进的原版教材能适应我国学生的外语水平和学习特点。在引进工作中，审定委员会还结合我国高校教学课程体系的设置和要求，对原版教材的教学思想和方法的先进性、科学性严格把关。同时尽量考虑原版教材的系统性和经济性。

这套教材出版后，我们将根据各高校的双语教学计划，举办原版教材的教师培训，及时地将其推荐给各高校选用。希望高校师生在使用教材后及时反馈意见和建议，使我们更好地为教学改革服务。

机械工业出版社

前　　言

我国加入 WTO 后，如何与国际接轨、适应全球经济一体化的需要、满足社会对复合型高级专业人才的需求，成为教育特别是高等教育的当务之急。双语教学是推进我国高等教育国际化、培养具有国际竞争力人才的一项重要举措，教育部已将“稳步推进双语教学”列为 2001 年以来“确保高校教学质量的 20 项工作”中的第 8 项，提出加强大学本科的双语教学，将双语课程的开设率作为高校教学质量评估的一项重要指标，双语教学已成为当前高校教学改革的热点。

教材是决定双语教学成效的关键。选用原版教材，可以使教师感悟国外先进的教育理念，体验培养学生创新能力的教学方法，使学生感受并逐步适应国外的自主学习模式。

近年来，国内出版社先后引进和影印了许多电子技术类原版教材，为我们提供了丰富的选材资源，但仔细分析后，不难发现这些教材普遍存在以下问题。第一，篇幅过长。原版教材的突出特点是实用性强、内容新颖、叙述详尽，而且习题数量多、质量高，便于英语水平较高的读者阅读。但一般篇幅较长，因此，受课程学时限制，一般只能讲授大约 1/2 的内容。第二，缺少中文注释。通篇的英文、繁多的专业词汇，不利于学生，特别是低年级学生的课外阅读和自学；而且存在较大的语言障碍，容易使学生对双语教学产生畏难心理。第三，编排体系与国内教材差异较大。多数数字电子技术类教材缺少硬件描述语言（如 VHDL 或 Verilog）内容，因此，讲授该部分内容需另购教材。第四，价格较高，加重了学生的经济负担。

2005 年，笔者在英国伯明翰大学电子工程学院研修期间，在认真调研该院电子信息类专业本科生“数字逻辑设计”课程教材及其使用效果的基础上，建议机械工业出版社引进由 Pearson Education（培生教育出版集团）出版的、Barry Wilkinson 编著的《数字设计基础》一书，并在此基础上，结合目前数字设计技术的发展和双语教学的需要，补充伯明翰大学电子工程学院 Steven Quigley 博士提供的该院“计算机硬件与数字设计”课程教学使用的 VHDL 硬件描述语言及其程序设计部分，增加北京工业大学电子信息与控制工程学院江捷副教授对重点、难点内容的中文注释。

改编后全书共分 12 章，第 1 ~ 7 章主要介绍逻辑代数基础、逻辑门、组合逻辑电路、触发器与计数器、时序逻辑电路、可编程逻辑器件、逻辑电路测试等内容；增补的第 8 ~ 12 章介绍 VHDL 硬件描述语言的设计动机、程序结构、行为描述与结构描述、仿真测试和时序电路的 VHDL 描述方法等内容。本书简明扼要，示例丰富，各章后附有适量习题、推荐阅读文献和习题解答。

VHDL是一种复杂的硬件描述语言，本书并没有对其做全面介绍，而是从应用实例出发，采用循序渐进的方法加以引导，介绍常见基本模块的设计方法，适合学生快速入门，即学即用。

重点、难点内容和专业词汇的中文注释，克服了课外阅读的语言障碍，便于学生自学。这种"双语教学版"教材，既可以让学生不完全依赖译文，又能很好地理解教材内容；既突出了国外教材实用性强的特点，又兼顾了我国学生的实际英语水平。本书作为参考教材试用以来，受到学生们的普遍欢迎。

在本书即将正式出版之际，感谢曾经给笔者提供过很多帮助的伯明翰大学电子工程学院的Steven Quigley、Phil Atkins和Frederick Huang。

由于时间仓促，作者水平有限，书中难免出现错误和不妥之处，恳请广大读者批评指正。

江　捷

Preface

The purpose of this book is to present all the key aspects of logic design as a first course for undergraduate Electrical Engineering and Computer Science students. The material has been developed specifically for one semester or course module.

Chapter 1 introduces the concept of a digital system and its application areas. The concept of a logic signal, which has one of two values, is introduced. All digital systems use logic signals that take one of two values, 0 and 1, both for control operations and for numbers (binary numbers). The binary number representation and binary arithmetic are outlined, in so far as it applies in the design of a digital system. Being an "essence" book, excessive details on binary numbers such as how to convert numbers to and from different number representations is given only a passing mention. References are given throughout to texts that extend the presentation in the book.

The underlying mathematical basis for a digital system is Boolean algebra, an algebra which, in this context, defines operations on two-valued variables. Boolean operations are implemented with logic gates, making the logic gate the fundamental component of a digital system. Logic gates are described in Chapter 2 with the relevant Boolean algebra underpinning. At the end of the chapter, quite simple electronic circuit details of some logic gates are outlined. This section will be of most interest to electrical engineers. Computer science students could omit the section without loss of continuity.

Logic gates are called combinational logic circuits because their outputs depend upon a combination of input values. Chapter 3 explores how to design more complex "combinational logic" digital systems using logic gates. One of the key aspects in the design of digital systems is the number and types of gates used. Often a minimum number of gates is desired. Techniques that lead to a reduced number of gates is described, notably the Karnaugh map minimization technique. Minimization of very complex logic circuits is often done by a computer program. Techniques more applicable to computer programs can be found in the list of further reading.

Most digital systems contain memory that can store information about past events. The response of such digital systems can then depend upon the past events. The basic logic components having memory are flip-flops which are described in Chapter 4. Flip-flops form the basis of more complex "sequential logic" circuits which can create sequences of output values. One very common sequential logic circuit is the counter whose outputs follow numeric sequences such as binary increasing sequences. The design of counters is also described in Chapter 4. Knowledge of the internal design of flip-flops can be found in the list of further reading, though this detail is often unnecessary since most designs use prefabricated flip-flops or standard designs.

Chapter 5 continues the design of sequential circuits. The method of producing a design starting with a state diagram is described through the use of examples. All the designs considered are so-called synchronous sequential logic circuits, those which are controlled by an external clock signal. Synchronous sequential logic circuits form the vast majority of designs because of the simplicity of timing operations with a clock signal. The design of asynchronous sequential logic circuits, which do not use a synchronizing clock signal, is described in the list of further reading.

Chapter 6 is devoted to one logic device family, the programmable logic device (PLD). PLDs are logic components that have alterable internal connections which enable various logic designs to be implemented totally within the device. PLDs are widely used to reduce the number of components and the cost. PLDs have to be "programmed" to obtain the desired internal connections. This programming is specified using a PLD programming language as described in Chapter 6. Sufficient details of a PLD language are given to enable designs to programmed. Complete details of PLD languages can be found in the further reading.

The final chapter, Chapter 7, introduces the area of logic circuit testing to detect faults that might have occurred during manufacture or subsequently. Testing is extremely important but has been omitted from many previous texts on logic design. Proper consideration should be given to manufacturing a working system as well as obtaining a logically correct design, and testing should be part of the design process. Chapter 7 is intended as a overview of the important techniques. More details of testing can be found in specialized texts listed in the further reading section.

As can be seen from the description of the chapters, there are several pointers to where the material can be developed. A solutions manual is also available to instructors.

I wish to record my continued appreciation to Christopher Glennie of Prentice Hall for his guidance throughout the preparation of this book, and his work in obtaining constructive reviews of the manuscript. I also wish to thank Jacqueline Harbor for handling the manuscript in its final stages and all of the Prentice Hall production staff for their professionalism. Finally, I thank the anonymous reviewers for their efforts and helpful comments. I would greatly appreciate any further suggestions or corrections to be send to me at abw@uncc.edu so that they can be incorporated into subsequent printings of this text.

Barry Wilkinson

Contents

目 录

CHAPTER 1

Digital systems and the representation of information[1]

1 数字系统与信息表示

Aims and objectives[2]

The purpose of this chapter is to establish the elements of a digital system and its application areas. The main part of this chapter concentrates upon how information is represented inside a digital system. The concept of a logic signal having two values is introduced and it is shown how such two-valued signals can be used to represent both control actions and numbers[3].

2 目标与任务
3 本章介绍数字系统及其应用领域的基础知识，重点讲述数字系统中信息的表示方法，提出二值逻辑信号的概念，阐述用逻辑信号表示控制作用及数值的方法。
4 数字系统领域
5 密切相关的任务

1.1 The realm of digital systems[4]

A digital system is often designed to satisfy two closely interrelated tasks[5]:

1. To control apparatus[6]
2. To perform calculations[7].

A digital system might perform some calculation and, on the basis of the result, take certain control actions[8]. Examples of such digital "control" systems would be a system to control (a) industrial equipment and (b) an automobile[9].

6 控制设备
7 执行运算
8 数字系统能够进行运算，并根据运算结果，采取相应控制。
9 工业设备和汽车

1.1.1 Control applications

Let us start with a simple industrial control application of controlling the amount of material held in a hopper, as shown in Figure 1.1. The task here is to maintain material in the hopper between level L1 and level L2. The material is taken from the hopper at intervals. The digital control system controls a flow valve supplying material to the hopper. Two sensors are provided, one to detect when the level L1 is reached by the material in the hopper, and one to detect when level L2 is reached by the material. Each sensor generates a signal when the material has passed the level set by the sensor. This signal would typically be a voltage, say 5 volts, to indicate that the material has passed the sensor,[10] and perhaps 0 volts to indicate that the material has not passed the sensor. Hence the input signals have two values, one to indicate

10 首先让我们分析一个简单的控制漏斗中物料存量的工业控制实例。物料不断地从漏斗中流出，数字系统通过调节流量阀门的位置，控制流入漏斗的物料，使物料存量保持在 L1 与 L2 之间。系统中装有两个传感器，分别检测漏斗中的物料是否达到 L1 和 L2。若超过指定位置，则传感器输出一个 5V 的电压信号。

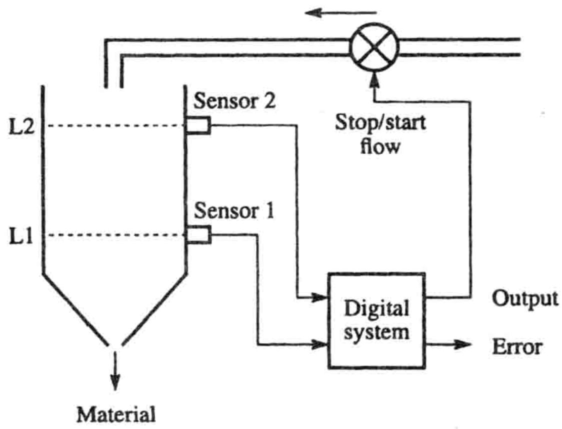

Figure 1.1 Controlling flow of material into a hopper[1]

1 控制流入漏斗的物料

the presence[2] of material and one to indicate the absence[3] of material. The flow control valve will also be controlled by a two-state signal[4] ON and OFF in this case. For example, a 50 volts signal might turn the valve on and 0 volts might turn it off. The flow valve is to be turned on when the material is below L1 and kept on until L2 is reached when the valve is turned off. Again the valve is kept off until the material drops to L1[5] The digital system must generate a signal which implements this "algorithm[6]". The signal has two values, one for ON and one for OFF.

Two levels are sensed in the system so that material can be allowed into the hopper for a reasonable period. Maintaining a single level by turning the valve on when below the level and turning the valve off when above the level would create a situation where the valve would be continually turned on and off.[7]

2 存在
3 不存在
4 流量阀门也是由一个二值信号控制
5 物料低于 L1,打开流量阀门,升至 L2,关闭阀门;物料降至 L1,则再次打开阀门。
6 算法
7 系统中设置两个位置检测点,可使物料在一段时间内持续流入漏斗。若只设置一个检测点,低于该点开阀门,高于该点关阀门,则势必造成阀门的连续开关。

Logic signals and functions[8]

8 逻辑信号与函数

In subsequent chapters, we will study how to design a digital system such as that shown in Figure 1.1. The important point to observe here is that the input and output signals have two values, and two voltages will be used, one for each value. The digital system itself might use 5 volts and 0 volts to represent the two values. Rather than refer to specific voltages, we will refer to two-valued *logic* signals[9] The digital system will be designed with components that can accept two-valued logic signals and generate two-valued logic signals[10] The two values in our example correspond to[11] ON and OFF. Sometimes the terms TRUE and FALSE are used, e.g. the signal from sensor 1 is TRUE when the material level is above level 1, and FALSE that the material level is below level 1. TRUE corresponds to ON and FALSE corresponds to OFF. The numerical values 1 and 0 are usually used in place on TRUE and FALSE in logic design; 1 corresponds to TRUE and 0 corresponds to FALSE. Table 1.1 relates the situation here and the corresponding "logic" signals.

9 我们所指的是二值逻辑信号,而不是特定的电压值。
10 数字系统中的器件,输入、输出的都是二值逻辑信号。
11 相应于

Table 1.1 *Logic signals*

Control situation	*Logic voltage*	*Switch*	*Logic value*	
Below level	0 V	OFF	FALSE	0
Above level	5 V	ON	TRUE	1

Positive and negative logic representation[1]

The actual voltages used in a system are typically 0 volts and 5 volts, but can be lower notably for the internal gates of complex components such as microprocessors.[2] Given two voltages, say 0 volts and 5 volts, we must decide which voltage will be used to represent which logic value. For example, we could use 0 volts to represent a logic 1 and 5 volts to represent a logic 0. Whatever decision is made concerning the voltages, the same voltages are used throughout the system. Usually the higher voltage is chosen to represent a logic 1 and the lower voltage used to represent a logic 0 (e.g. 5 volts = 1 and 0 volts = 0). This is known as *positive logic representation*. An alternative is to use the higher voltage to represent a logic 0 and the lower voltage to represent a logic 1 which is known as *negative logic representation*. It is possible to use both positive and negative logic representations in a single logic system[3](while still using only two voltages). Using both representations in the same system is know as *mixed logic representation*.[4] Then a special notation is needed for naming logic signals (see Fletcher, 1980 or Tinder, 1991).

1 正负逻辑表示
2 系统使用的电压通常为 0V 和 5V，但在某些复杂组件如微处理器的内部，电压的实际值却要低得多。
3 同一系统，允许正、负逻辑并用。
4 同一系统使用两种逻辑，称为混合逻辑。

Logic functions

To be able to develop the "algorithm" or logic function for a digital system, logic signals must be represented by names just as in algebra.[5] For example, the two inputs in Figure 1.1 might be called S_1 and S_2 (for sensor 1 and sensor 2 respectively). The output might be called Z. In Figure 1.1, a second output signal is shown, a signal indicating that an error has occurred. An example of an error condition would be sensor 1 showing the material below level L1 and sensor 2 showing the material above level L2, which is of course impossible if all the components are working. Let the error output be E. The "algorithm" for the error signal is:

$$E = 1 \text{ if } S_1 = 0 \text{ and } S_2 = 1$$

This logic function can be written in a two-valued algebra called *Boolean algebra*[6]:

$$E = \overline{S}_1 \cdot S_2$$

In this notation, $\overline{S}_1$ will be a 0 when S_1 is a 1 and $\overline{S}_1$ will be a 1 when S_1 is a 0. The dot (·) corresponds to "and".

There could be more than one error condition, say E_1, E_2 and E_3, each of which indicates an error. If E_1 is a 1 or E_2 is a 1 or E_3 is a 1, we might want a general alarm[7] to be sounded. The algorithm for this alarm is:

5 逻辑函数
为生成数字系统的控制算法或逻辑函数，逻辑信号需用变量名表示，这与普通代数中相同。
6 布尔代数
7 报警

Alarm = 1 if $E_1 = 1$ or $E_2 = 1$ or $E_3 = 1$.

The logic function for the alarm is written in Boolean Algebra as:

$$\text{Alarm} = E_1 + E_2 + E_3$$

Here + corresponds to "or". In fact, we have now introduced the three fundamental operations of Boolean algebra, AND (·), OR (+) and NOT (¯).[1] From these operations, we can create any logical function. The full range of this algebra will be developed in Chapter 2. For the controller application, we need a logic circuit having two inputs, S_1 and S_2 and a single output, E, which will be a 1 if $S_1 = 0$ and $S_2 = 1$, otherwise $E = 0$. This requires a so-called *combinational logic circuit* because the output value will depend upon certain combinations of input values.[2] The alarm function is also implemented with a combinational logic circuit.

The function for the output, Z, is slightly more complex as it requires the circuit to memorize logic values.[3] Z will change to a 1 when $S_1 = 0$ but does not change back to a 0 when $S_1 = 1$ but only when $S_2 = 1$. This is a so-called *sequential logic circuit*, because generally the outputs of such circuits will follow sequences.[4] Designing such logic circuits is addressed first in Chapter 4.

1 布尔代数的三种基本运算,逻辑与、或、非。

2 组合逻辑电路因电路的输出值取决于输入变量的某些组合而得名。

3 因为要记忆电路以前的逻辑值,所以输出函数 Z 的表达式会变得复杂些。

4 时序逻辑电路因其输出在通常情况下,按一定时序规律变化而得名。

Calculations

More elaborate[5] control applications might use sensors which take measurements. There might be a sensor which measures temperature. The material in our hopper might need to be maintained at a specific temperature by the use of heating elements. For example, the heater[6] might be turned on when the temperature is below 40°C and kept on until a temperature of 45°C is reached. The heater is then turned off until the temperature has fallen to below 40°C. This is a very similar algorithm to that used to control the hopper, except now we need to handle numbers rather than simply ON/OFF values.[7] In this and other applications, there might be many measurements and complex calculations necessary based upon the measurements. For example, a modern automobile engine[8] is usually controlled internally by a digital system. Here the control of actions within the engine for optimum performance are dependent upon several variables including the current speed, the load, temperature, etc.[9] Again, we now must operate upon numbers, not just two-valued logic variables.

5 复杂的

6 加热器

7 这和前面控制漏斗的算法非常类似,只是现在我们需要处理的是数值大小,而不再是简单的开关量。

8 汽车发动机

9 使发动机性能最优的控制策略取决于包括当前速度、负载、温度等在内的多个变量。

Digital computer

Performing calculations in a semi-automatic way[10] has long been a desire to relieve the burden of manual calculations.[11] A mechanical[12] calculator was devised by Babbage in the 1850s, and electronic versions began to be developed during the second world war, for example to calculate paths of projectiles.[13] Babbage conceived the idea of a universal programmable calculator,[14] the electronic version we now call a "digital computer". The digital computer represents one of the most complex digital systems, a system which manipulates numbers and calculates values from these numbers using a program of steps stored in the memory of the computer.[15] Most complex

10 半自动式

11 减轻手工计算的负担

12 机械的

13 计算导弹轨迹

14 构思设计通用的可编程计算器

15 存放在计算机存储器中的程序