Prentice Hall

时代教育·国外高校优秀教材精选

应用PLD的数字电子技术

(英文版)

Digital Electronics with PLD Integration

(美)奈杰尔 P.库克(Nigel P.Cook)著

机械工业出版社
CHINA MACHINE PRESS

时代教育·国外高校优秀教材精选

应用PLD的数字电子技术

（英文版）

Digital Electronics with PLD Integration

（美） 奈杰尔 P. 库克（Nigel P. Cook） 著

江苏工业学院图书馆
藏书章

机械工业出版社

English reprint copyright © 2002 by Pearson Education North Asia Limited and China Machine Press.

Original English language title: Digital Electronics with PLD Integration by Nigel P. Cook
Copyright © 2001 by Prentice-Hall, Inc.
All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice-Hall, Inc.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong and Kong and Macau).

本书英文影印版由美国 Pearson Education（培生教育出版集团）授权机械工业出版社在中国大陆境内独家出版发行，未经出版者许可，不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封面贴有 Pearson Education 培生教育出版集团激光防伪标签，无标签者不得销售。
北京市版权局著作权合同登记号：图字：01-2002-5029

图书在版编目（CIP）数据

应用 PLD 的数字电子技术/（美）库克（Cook, N. P.）著. —北京：机械工业出版社，2003. 1
（时代教育：国外高校优秀教材精选）
ISBN 7-111-11297-0

Ⅰ. 应… Ⅱ. 库… Ⅲ. 可编程序逻辑器件—高等学校—教材—英文 Ⅳ. TP211

中国版本图书馆 CIP 数据核字（2002）第 097549 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：周 娟
封面设计：鞠 杨 责任印制：付方敏
北京铭成印刷有限公司印刷·新华书店北京发行所发行
2003 年 3 月第 1 版第 1 次印刷
850mm×1168mm 1/16 · 63.75 印张 · 2 插页 · 1390 千字
定价：85.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话（010）68993821、88379646
封面无防伪标均为盗版

国外高校优秀教材审定委员会

主任委员：杨叔子

委员（按姓氏笔画为序）：

王先逵　王大康　白峰杉　史荣昌　朱孝禄
陆启韶　张润琦　张　策　张三慧　张福润
张延华　吴宗泽　吴　麒　宋心琦　李俊峰
佘远斌　陈文楷　陈立周　范　瑜　俞正光
赵汝嘉　翁海珊　龚光鲁　章栋恩　黄永畅
谭泽光

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，机械工业出版社拟于近期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

2002 年 3 月

此为试读，需要完整PDF请访问：www.ertongbook.com

序

自20世纪60年代以来，正像人们所看到的那样，电子技术正在持续不断地发展，近年来取得了突飞猛进的进步。电子技术的进步推动了计算机、通信、工业自动化、医学及生物科学等诸多学科的发展。特别是超大规模集成电路（VLSI）的问世，为电子技术的发展树立了里程碑。VLSI带来了电子技术设计方法上的革命，使数字电子技术具有了用硬件描述语言把计算机与通信相互沟通的能力，使计算机技术和通信技术相互联结，因此使计算机能够具有基于网络的通信功能，通信网络能够通过基于系统的计算机来实现。

从计算机性能上的变化，我们能够深刻地看到电子技术进步的作用。最新的统计资料表明，微处理器IC芯片每年以60%的速度增长，存储器IC芯片的容量每3年增加4倍，与这些相关的关键模块也相应地快速增加。世界范围内最大的工业是电子工业，超过了汽车工业和石油工业，电子产品的销售额以每年高出2万亿美元的速度增长，电子消费品一般不超过3年就被淘汰。因此，电子产品开发研制周期已缩短至6个月或不足半年。正是出于这些原因，本书的作者编写了《Digital Electronics with PLD Integration》，使在校的学生及从事电子技术设计的人员能够快速掌握和应用PLD的设计方法，使他们的设计能力和基本素质得以快速提高和加强。

本书内容不同于一般传统数字电子技术教材，而是以PLD为核心内容，以软件工具相辅助，介绍Altera公司的CPLD和Xilinx公司的EPGA的基本概念和原理。为了便于理解和掌握PLD，在每一章节前都以普通标准逻辑电路开始，每章都有紧密结合实际应用的设计实例，然后采用图形输入方法（包括Max+PlusⅡ9.2和Fundation series)的软件工具进行PLD的设计、仿真和下载到实验板上进行实验验证。这是一种即学即做方式。为了注重培养动手能力，还在每章之后安排了有关调试和检测方面的问题指导。除此之外，各章后面都附加了多选题、练习题、硬件实践题和自测评价题等，并且书后附有答案，帮助实践者提高能力和弄清概念。综上所述，本书很有特色，对学习和学会使用PLD很有价值。

陈文楷
北京工业大学
2002年10月

Preface

Analog to Digital

Since World War II, no branch of science has contributed more to the development of the modern world than electronics. It has stimulated dramatic advances in the fields of communication, computing, consumer products, industrial automation, tests and measurement, and health care. It has now become the largest single industry in the world, exceeding the automobile and oil industries, with annual sales of electronic systems greater than $2 trillion.

One of the most important trends in this huge industry has been a gradual shift from analog electronics to digital electronics. This movement began in the 1960s and is almost complete today. In fact, a recent statistic stated that, on average, 90% of the circuitry within electronic systems is now digital and only 10% is analog. This digitalization of the electronics industry is merging sectors that were once separate. For example, two of the largest sectors or branches of electronics are *computing* and *communications*. Being able to communicate with each other using a common digital language has enabled computers and communications to interlink, so that computers can now function within communication-based networks, and communications networks can now function through computer-based systems. Industry experts call this merging *convergence,* and predict that digital electronics will continue to unite the industry and stimulate progress in practically every field of human endeavor.

Needless to say, this course you are about to undertake in digital electronic concepts, terminology, components, circuits, applications, and testing and troubleshooting is an essential element in your study of electronics.

WHY USE PROGRAMMABLE LOGIC DEVICES?

There has been, and continues to be, an almost meteoric advancement in digital technology every year. This is evidently clear when we see the performance of personal computers, or PCs, advance in leaps and bounds. To quote a recently posted statistic, microprocessor ICs are improving at a rate of 60% per year, while memory ICs are quadrupling their capacity every three years. These, and other rapid changes in all the key building blocks of digital electronic systems, mean that consumer products are generally obsolete in less than three years.

To keep up with this fast pace, electronic companies have to design and manufacture new products in a cycle of typically less than six months. To meet this accelerated schedule, engineers and technicians have looked for shortcuts that enable them to construct a digital prototype circuit and evaluate its performance in a much more timely manner.

Constructing a Prototype Using Standard Logic Devices

To construct a circuit using standard logic devices, you would first need to insert all of the devices into a protoboard, and then connect them with a spaghetti-like maze of hookup wire. This standard logic prototyping method has the following disadvantages:

- Hookup wire cutting and stripping is time consuming.
- Wires can easily be inserted incorrectly, causing possible device damage and lengthy delays while the errors are isolated.
- A large and costly inventory of all standard logic ICs must be maintained.
- If the desired standard logic IC is not available, further delays will result.
- To modify or add to a working circuit, the wires and ICs have to be removed from the protoboard, and the new design rebuilt from scratch.

Constructing a Circuit Using Programmable Logic Devices

By using an inexpensive personal computer (PC), a Computer Aided Design (CAD) software program, and a single Programmable Logic Device (PLD), you can easily prototype a digital circuit.

The five-step process for creating a prototype using a PLD is: (1) create the circuit using the PC, (2) then compile it, (3) simulate it on the PC to see if it works as it should, (4) download it into the PLD, and finally (5) test it by applying inputs and monitoring outputs.

This PLD prototyping method has the following advantages:

- With manual wiring reduced to a minimum, prototypes can be constructed, tested, and modified at a much faster rate.
- Wiring errors can be avoided.
- You can experiment with many digital IC types without having to stock them in your supply cabinet.
- Circuit designs can be saved as electronic files within the PC and used again when needed.
- Since the PLD can be used over and over again, modifications can easily be made by altering the circuit in the PC, and then downloading the new design into the PLD.
- Larger and more complex projects can be undertaken now that the tedious manual procedures are automated.

In this text we will be examining this PLD prototyping method in detail to prepare you for industry. Referring to the following chapter outline, you can see that you are first introduced to the PLD method of prototyping in Chapter 4 *(Standard Logic versus Programmable Logic)*. From that point on, the PLD alternative is integrated into the traditional standard logic device topics.

CHAPTER OUTLINE

PART I Digital Basics

Chapter 1 Analog to Digital
Chapter 2 Number Systems and Codes
Chapter 3 Logic Gates
Chapter 4 Standard Logic versus Programmable Logic
Chapter 5 Digital IC Types
Chapter 6 Troubleshooting Logic Gates
Chapter 7 Logic Circuit Simplification

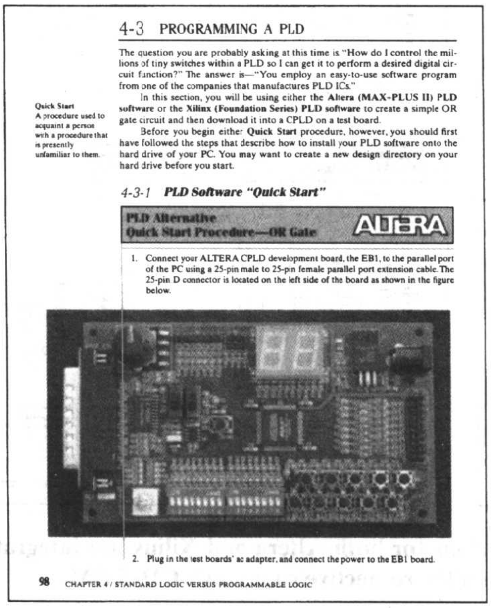
PART II Digital Circuits

Chapter 8 Decoders and Encoders
Chapter 9 Other Combinational Logic Circuits
Chapter 10 Set-Reset and Data-Type Flip-Flops
Chapter 11 *JK* Flip-Flop and Timer Circuits
Chapter 12 Registers
Chapter 13 Counters
Chapter 14 Arithmetic Operations and Circuits
Chapter 15 Semiconductor Memories
Chapter 16 Analog and Digital Signal Converter Circuits

PART III Digital Systems

Chapter 17 Introduction to Microprocessors

ILLUSTRATED TOUR OF TEXTBOOK FEATURES



4-3 PROGRAMMING A PLD

The question you are probably asking at this time is "How do I control the millions of tiny switches within a PLD so I can get it to perform a desired digital circuit function?" The answer is—"You employ an easy-to-use software program from one of the companies that manufactures PLD ICs."

In this section, you will be using either the **Altera (MAX-PLUS II) PLD software** or the **Xilinx (Foundation Series) PLD software** to create a simple OR gate circuit and then download it into a CPLD on a test board.

Before you begin either **Quick Start** procedure, however, you should first have followed the steps that describe how to install your PLD software onto the hard drive of your PC. You may want to create a new design directory on your hard drive before you start.

Quick Start
A procedure used to acquaint a person with a procedure that is presently unfamiliar to them.

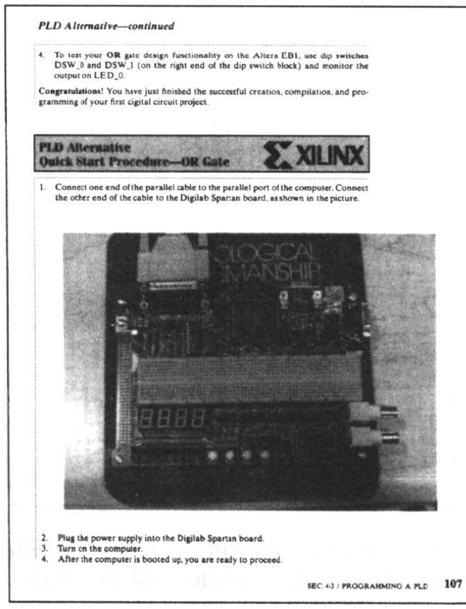
4-3-1 ***PLD Software "Quick Start"***

PLD Alternative
Quick Start Procedure—OR Gate ALTERA

1. Connect your ALTERA CPLD development board, the EB1, to the parallel port of the PC using a 25-pin male to 25-pin female parallel port extension cable. The 25-pin D connector is located on the left side of the board as shown in the figure below.

2. Plug in the test boards' ac adapter, and connect the power to the EB1 board.

98 CHAPTER 4 / STANDARD LOGIC VERSUS PROGRAMMABLE LOGIC



PLD Alternative—continued

4. To test your **OR** gate design functionality on the Altera EB1, use dip switches DSW_0 and DSW_1 (on the right end of the dip switch block) and monitor the output on LED_0.

Congratulations! You have just finished the successful creation, compilation, and programming of your first digital circuit project.

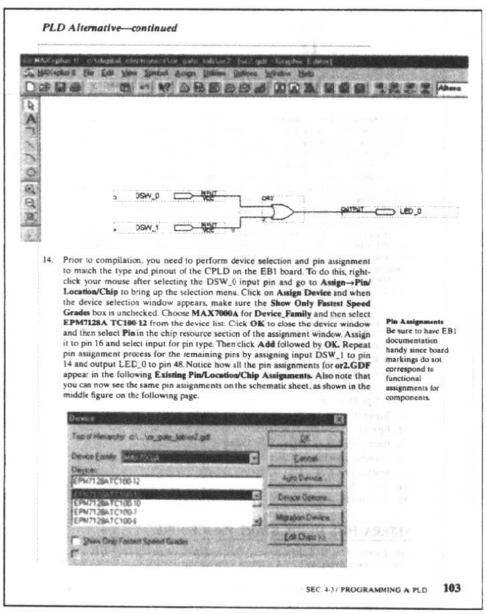
PLD Alternative
Quick Start Procedure—OR Gate XILINX

1. Connect one end of the parallel cable to the parallel port of the computer. Connect the other end of the cable to the Digilab Spartan board, as shown in the picture.

2. Plug the power supply into the Digilab Spartan board.
3. Turn on the computer.
4. After the computer is booted up, you are ready to proceed.

SEC 4-3 / PROGRAMMING A PLD 107

The textbook and its supplements support both PLD IC manufacturers, *Altera* and *Xilinx*. The two separate student lab manuals provide students with access to the full student version of the application software.

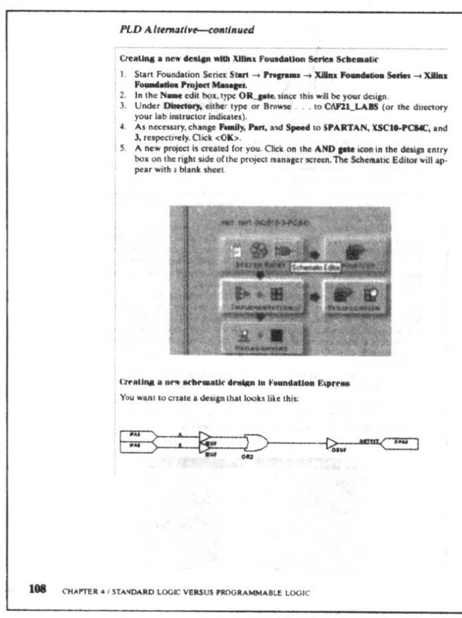


PLD Alternative—continued

14. Prior to compilation, you need to perform device selection and pin assignment to match the type and pinout of the CPLD on the EB1 board. To do this, right-click your mouse after selecting the DSW_0 input pin and go to **Assign→Pin/Location/Chip** to bring up the selection menu. Click on **Assign Device** and when the device selection window appears, make sure the **Show Only Fastest Speed Grades** box is unchecked. Choose **MAX7000A** for **Device_Family** and then select **EPM7128A TC100-12** from the device list. Click **OK** to close the device window and then select **Pin** in the chip resource section of the assignment window. Assign it to pin 16 and select input for pin type. Then click **Add** followed by **OK.** Repeat pin assignment process for the remaining pins by assigning input DSW_1 to pin 14 and output LED_0 to pin 48. Notice how all the pin assignments for **or2.GDF** appear in the following **Existing Pin/Location/Chip Assignments.** Also note that you can now see the same pin assignments on the schematic sheet, as shown in the middle figure on the following page.

Pin Assignments
Be sure to have EB1 documentation handy since board markings do not correspond to functional assignments for components.

SEC 4-3 / PROGRAMMING A PLD 103



PLD Alternative—continued

Creating a new design with Xilinx Foundation Series Schematic

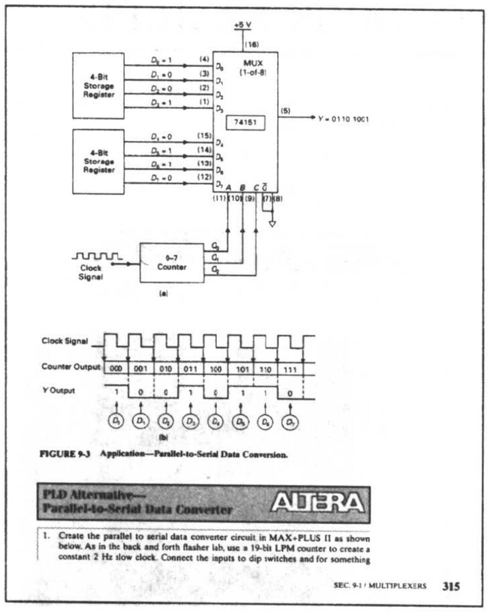
1. Start Foundation Series: **Start → Programs → Xilinx Foundation Series → Xilinx Foundation Project Manager.**
2. In the **Name** edit box, type **OR_gate**, since this will be your design.
3. Under **Directory,** either type or Browse . . . to **C:\F21_LABS** (or the directory your lab instructor indicates).
4. As necessary, change **Family, Part,** and **Speed** to **SPARTAN, XSC10-PC84C,** and **3,** respectively. Click <OK>.
5. A new project is created for you. Click on the **AND gate** icon in the design entry box on the right side of the project manager screen. The Schematic Editor will appear with a blank sheet.

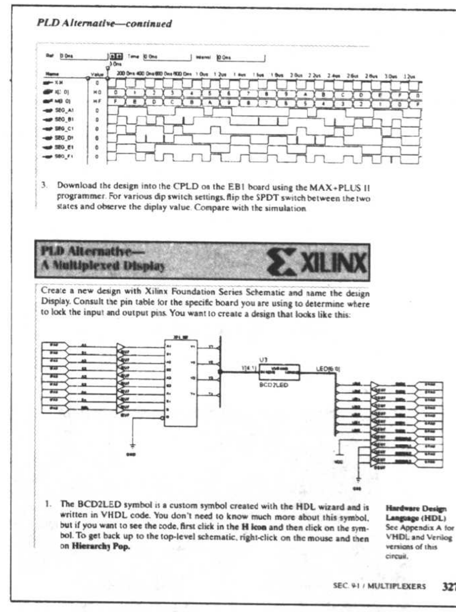
Creating a new schematic design in Foundation Express

You want to create a design that looks like this:

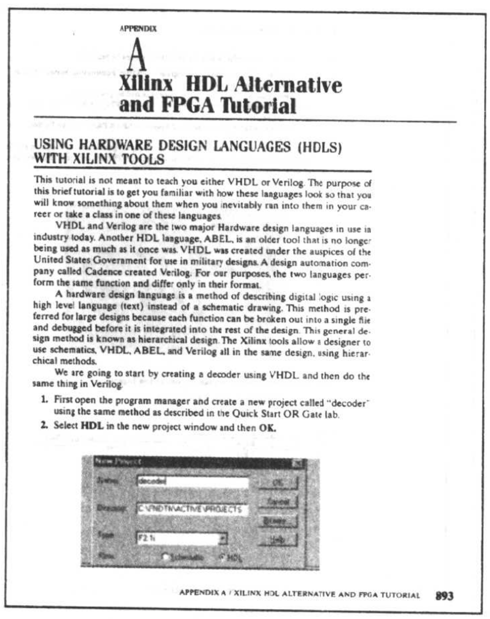
108 CHAPTER 4 / STANDARD LOGIC VERSUS PROGRAMMABLE LOGIC

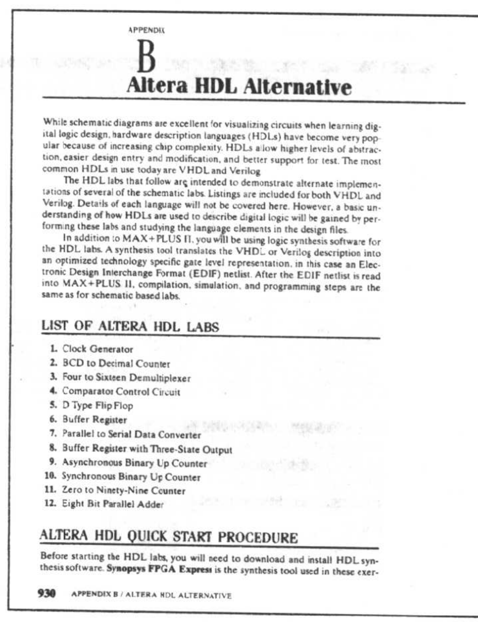
Chapter 4, *Standard Logic versus Programmable Logic*, introduces the student to the PLD concept, PLD devices, and the PLD software, with an easy-to-follow "Quick Start" procedure.





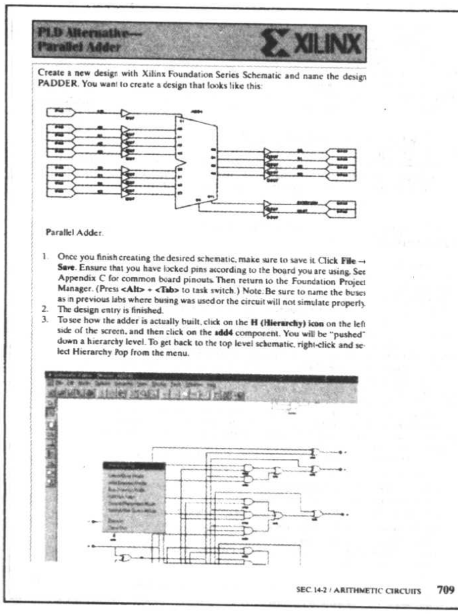
Over 80 *PLD Alternative Circuit Applications* for both Altera and Xilinx are integrated into the text and included within a data file on the respective company's CD-ROM.

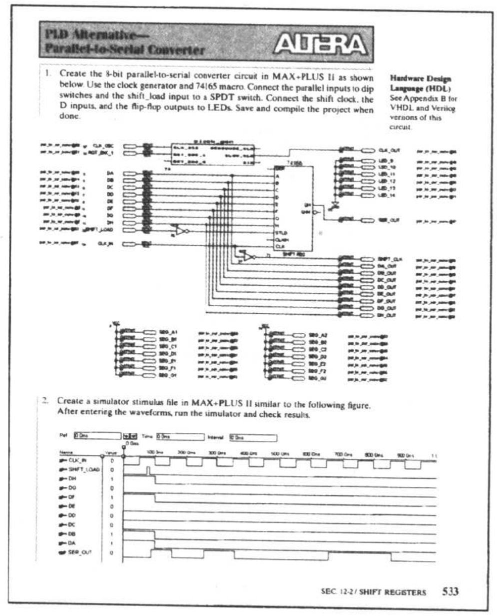




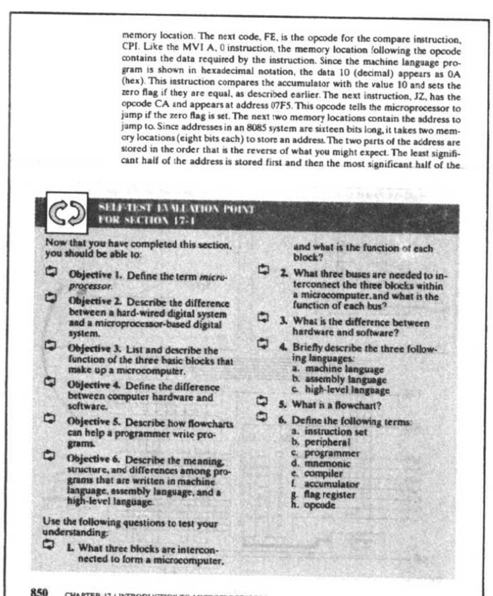
Many of the circuit applications can be implemented using a *Hardware Description Language (HDL)* detailed in an appendix. Both *VHDL* (VHSIC-HDL—very high speed integrated circuit hardware description language) and *Verilog HDL* are included.

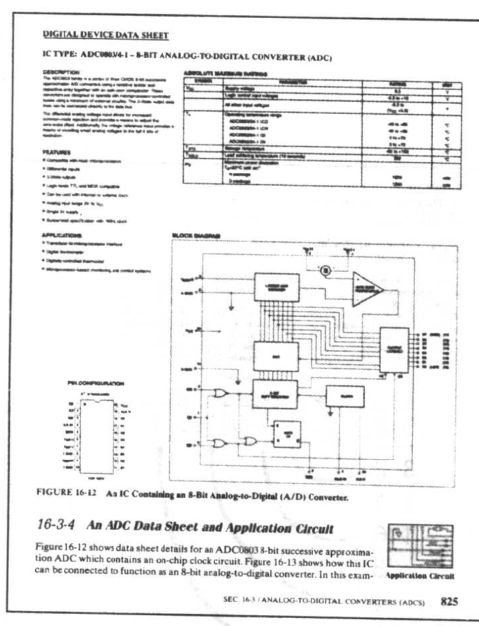
此为试读,需要完整PDF请访问: www.ertongbook.com





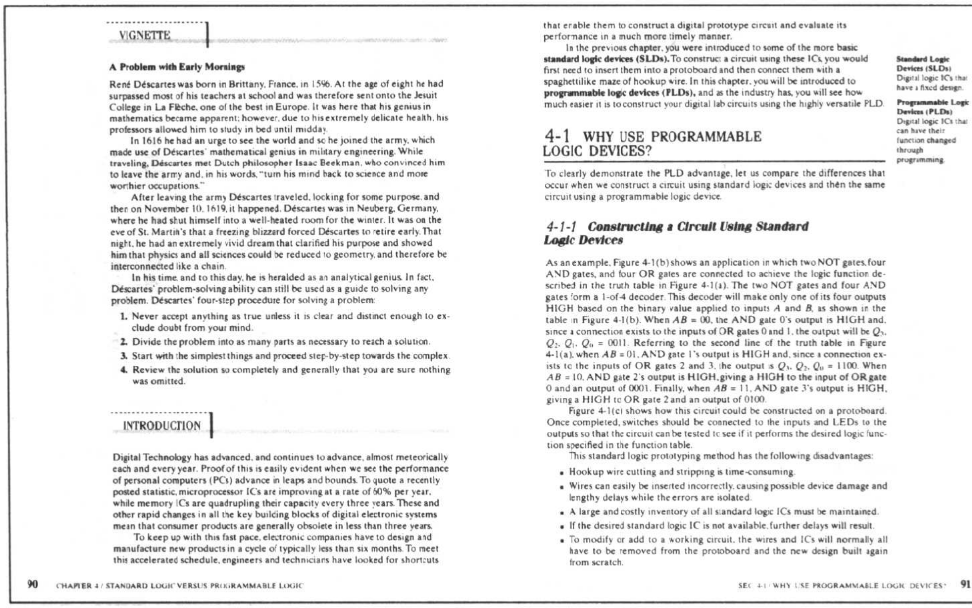
The text uses an *Application First Approach* in regard to the PLD application software. The initial "Quick Starts" take the student through the complete PLD prototyping procedure. The integrated "PLD Alternative Circuit Applications" in all of the following chapters give further examples for, and instruction on, the PLD software.





The section review breaks are now *Self-Test Evaluation Points,* with a list of objectives that should have been met up to that point, and a set of questions designed to test the students' level of comprehension. The *check-box* design invites students to take an active role in noting their understanding of the material.

A large number of *data sheets* are integrated into their appropriate positions, with *highlighted annotations* included to explain the meaning of key characteristics and symbols.



VIGNETTE

A Problem with Early Mornings

René Déscartes was born in Brittany, France, in 1596. At the age of eight he had surpassed most of his teachers at school and was therefore sent onto the Jesuit College in La Flèche, one of the best in Europe. It was here that his genius in mathematics became apparent; however, due to his extremely delicate health, his professors allowed him to study in bed until midday.

In 1616 he had an urge to see the world and so he joined the army, which made use of Déscartes' mathematical genius in military engineering. While traveling, Déscartes met Dutch philosopher Isaac Beekman, who convinced him to leave the army and, in his words, "turn his mind back to science and more worthier occupations."

After leaving the army Déscartes traveled, looking for some purpose, and then on November 10, 1619, it happened. Déscartes was in Neuberg, Germany, where he had shut himself into a well-heated room for the winter. It was on the eve of St. Martin's that a freezing blizzard forced Déscartes to retire early. That night, he had an extremely vivid dream that clarified his purpose and showed him that physics and all sciences could be reduced to geometry, and therefore be interconnected like a chain.

In his time, and to this day, he is heralded as an analytical genius. In fact, Déscartes' problem-solving ability can still be used as a guide to solving any problem. Déscartes' four-step procedure for solving a problem:

1. Never accept anything as true unless it is clear and distinct enough to exclude doubt from your mind.
2. Divide the problem into as many parts as necessary to reach a solution.
3. Start with the simplest things and proceed step-by-step towards the complex.
4. Review the solution so completely and generally that you are sure nothing was omitted.

INTRODUCTION

Digital Technology has advanced, and continues to advance, almost meteorically each and every year. Proof of this is easily evident when we see the performance of personal computers (PCs) advance in leaps and bounds. To quote a recently posted statistic, microprocessor ICs are improving at a rate of 60% per year, while memory ICs are quadrupling their capacity every three years. These and other rapid changes in all the key building blocks of digital electronic systems mean that consumer products are generally obsolete in less than three years.

To keep up with this fast pace, electronic companies have to design and manufacture new products in a cycle of typically less than six months. To meet this accelerated schedule, engineers and technicians have looked for shortcuts

90 CHAPTER 4 / STANDARD LOGIC VERSUS PROGRAMMABLE LOGIC

that enable them to construct a digital prototype circuit and evaluate its performance in a much more timely manner.

In the previous chapter, you were introduced to some of the more basic **standard logic devices (SLDs).** To construct a circuit using these ICs, you would first need to insert them into a protoboard and then connect them with a spaghettilike maze of hookup wire. In this chapter, you will be introduced to **programmable logic devices (PLDs),** and as the industry has, you will see how much easier it is to construct your digital lab circuits using the highly versatile PLD.

Standard Logic Devices (SLDs) Digital logic ICs that have a fixed design.

Programmable Logic Devices (PLDs) Digital logic ICs that can have their function changed through programming.

4-1 WHY USE PROGRAMMABLE LOGIC DEVICES?

To clearly demonstrate the PLD advantage, let us compare the differences that occur when we construct a circuit using standard logic devices and then the same circuit using a programmable logic device.

4-1-1 ***Constructing a Circuit Using Standard Logic Devices***

As an example, Figure 4-1(b) shows an application in which two NOT gates, four AND gates, and four OR gates are connected to achieve the logic function described in the truth table in Figure 4-1(a). The two NOT gates and four AND gates form a 1-of-4 decoder. This decoder will make only one of its four outputs HIGH based on the binary value applied to inputs A and B, as shown in the table in Figure 4-1(b). When $AB = 00$, the AND gate 0's output is HIGH and, since a connection exists to the inputs of OR gates 0 and 1, the output will be Q_3, Q_2, Q_1, Q_0 = 0011. Referring to the second line of the truth table in Figure 4-1(a), when $AB = 01$, AND gate 1's output is HIGH and, since a connection exists to the inputs of OR gates 2 and 3, the output is Q_3, Q_2, Q_0 = 1100. When $AB = 10$, AND gate 2's output is HIGH, giving a HIGH to the input of OR gate 0 and an output of 0001. Finally, when $AB = 11$, AND gate 3's output is HIGH, giving a HIGH to OR gate 2 and an output of 0100.

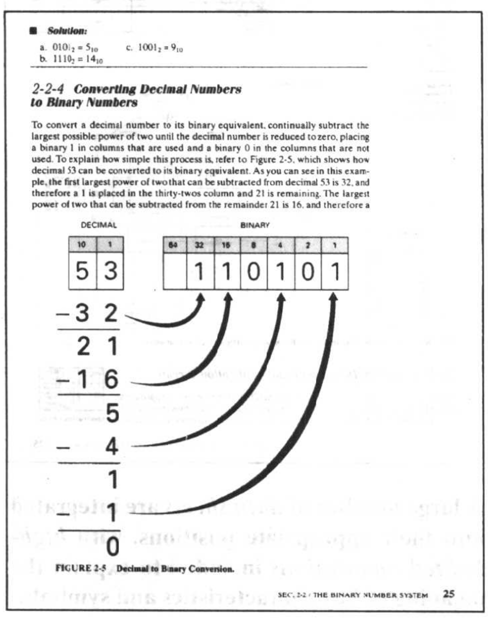
Figure 4-1(c) shows how this circuit could be constructed on a protoboard. Once completed, switches should be connected to the inputs and LEDs to the outputs so that the circuit can be tested to see if it performs the desired logic function specified in the function table.

This standard logic prototyping method has the following disadvantages:

- Hookup wire cutting and stripping is time-consuming.
- Wires can easily be inserted incorrectly, causing possible device damage and lengthy delays while the errors are isolated.
- A large and costly inventory of all standard logic ICs must be maintained.
- If the desired standard logic IC is not available, further delays will result.
- To modify or add to a working circuit, the wires and ICs will normally all have to be removed from the protoboard and the new design built again from scratch.

SEC. 4-1 / WHY USE PROGRAMMABLE LOGIC DEVICES? 91

***Chapter opening vignettes*, featuring electronic industry entrepreneurs, motivate students to read and understand chapter material, and *conversational introductions* review what has been previously covered and what is about to be covered.**



■ ***Solution:***

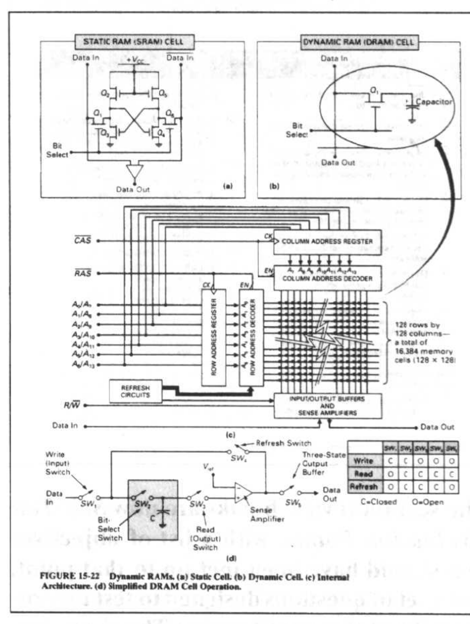
a. $0101_2 = 5_{10}$ c. $1001_2 = 9_{10}$
b. $1110_2 = 14_{10}$

2-2-4 ***Converting Decimal Numbers to Binary Numbers***

To convert a decimal number to its binary equivalent, continually subtract the largest possible power of two until the decimal number is reduced to zero, placing a binary 1 in columns that are used and a binary 0 in the columns that are not used. To explain how simple this process is, refer to Figure 2-5, which shows how decimal 53 can be converted to its binary equivalent. As you can see in this example, the first largest power of two that can be subtracted from decimal 53 is 32, and therefore a 1 is placed in the thirty-twos column and 21 is remaining. The largest power of two that can be subtracted from the remainder 21 is 16, and therefore a

FIGURE 2-5 Decimal to Binary Conversion.

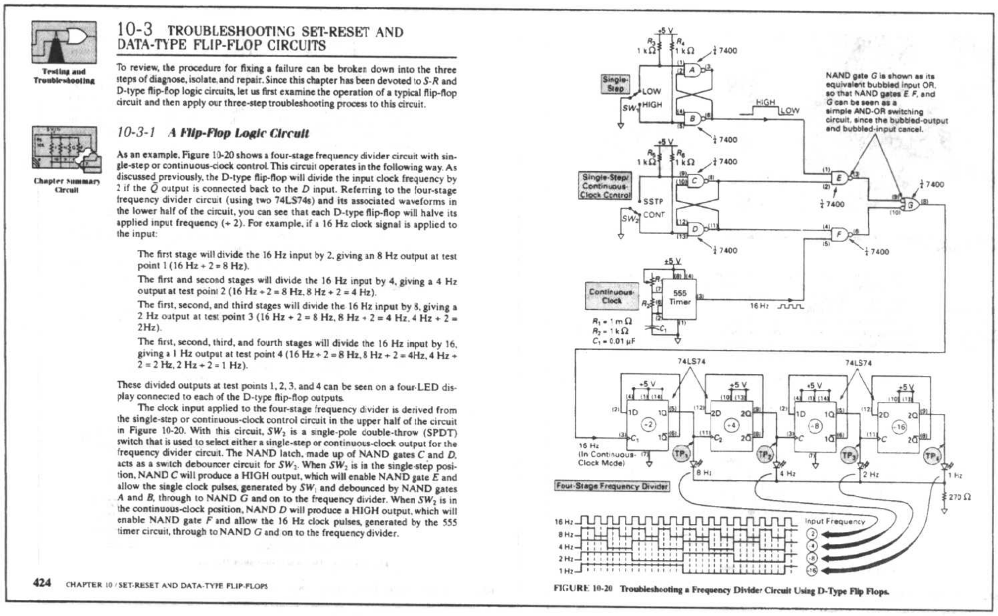
SEC. 2-2 / THE BINARY NUMBER SYSTEM 25



	SW_1	SW_2	SW_3	SW_4	SW_5
Write	C	C	O	O	O
Read	O	C	C	C	C
Refresh	O	C	C	C	O

FIGURE 15-22 Dynamic RAMs. (a) Static Cell. (b) Dynamic Cell. (c) Internal Architecture. (d) Simplified DRAM Cell Operation.

A *student-friendly writing style* coupled with *dynamic diagrams* enable the student to comfortably master the material. *Margin terms* highlight key terminology as it is introduced.



10-3 TROUBLESHOOTING SET-RESET AND DATA-TYPE FLIP-FLOP CIRCUITS

Testing and Troubleshooting

To review, the procedure for fixing a failure can be broken down into the three steps of diagnose, isolate, and repair. Since this chapter has been devoted to S-R and D-type flip-flop logic circuits, let us first examine the operation of a typical flip-flop circuit and then apply our three-step troubleshooting process to this circuit.

10-3-1 A Flip-Flop Logic Circuit

Chapter Summary Circuit

As an example, Figure 10-20 shows a four-stage frequency divider circuit with single-step or continuous-clock control. This circuit operates in the following way. As discussed previously, the D-type flip-flop will divide the input clock frequency by 2 if the $\bar{Q}$ output is connected back to the D input. Referring to the four-stage frequency divider circuit (using two 74LS74s) and its associated waveforms in the lower half of the circuit, you can see that each D-type flip-flop will halve its applied input frequency (÷ 2). For example, if a 16 Hz clock signal is applied to the input:

The first stage will divide the 16 Hz input by 2, giving an 8 Hz output at test point 1 (16 Hz ÷ 2 = 8 Hz).

The first and second stages will divide the 16 Hz input by 4, giving a 4 Hz output at test point 2 (16 Hz ÷ 2 = 8 Hz, 8 Hz ÷ 2 = 4 Hz).

The first, second, and third stages will divide the 16 Hz input by 8, giving a 2 Hz output at test point 3 (16 Hz ÷ 2 = 8 Hz, 8 Hz ÷ 2 = 4 Hz, 4 Hz ÷ 2 = 2 Hz).

The first, second, third, and fourth stages will divide the 16 Hz input by 16, giving a 1 Hz output at test point 4 (16 Hz ÷ 2 = 8 Hz, 8 Hz ÷ 2 = 4 Hz, 4 Hz ÷ 2 = 2 Hz, 2 Hz ÷ 2 = 1 Hz).

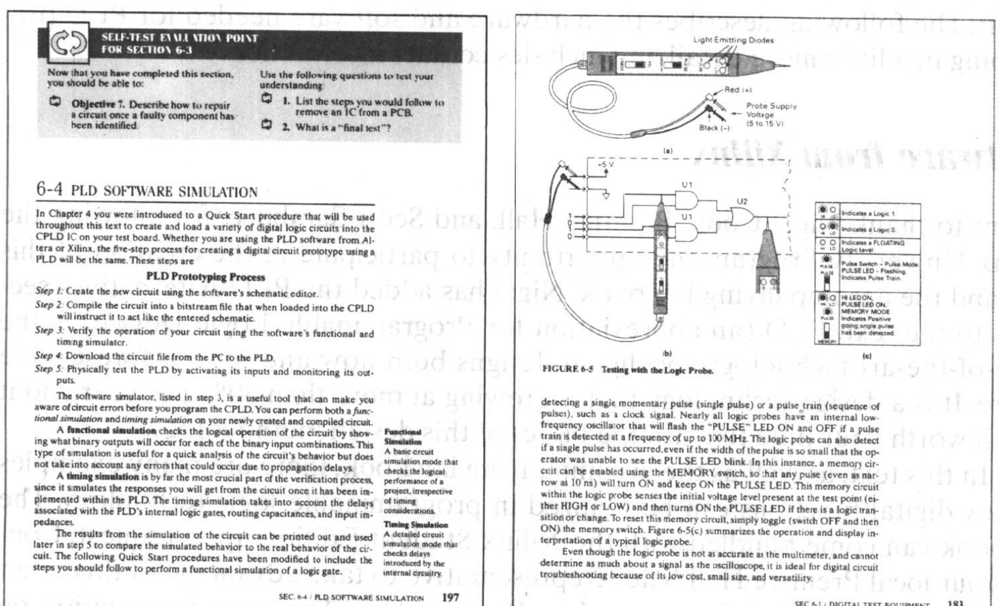
These divided outputs at test points 1, 2, 3, and 4 can be seen on a four-LED display connected to each of the D-type flip-flop outputs.

The clock input applied to the four-stage frequency divider is derived from the single-step or continuous-clock control circuit in the upper half of the circuit in Figure 10-20. With this circuit, SW_2 is a single-pole double-throw (SPDT) switch that is used to select either a single-step or continuous-clock output for the frequency divider circuit. The NAND latch, made up of NAND gates C and D, acts as a switch debouncer circuit for SW_2. When SW_2 is in the single-step position, NAND C will produce a HIGH output, which will enable NAND gate E and allow the single clock pulses, generated by SW_1 and debounced by NAND gates A and B, through to NAND G and on to the frequency divider. When SW_2 is in the continuous-clock position, NAND D will produce a HIGH output, which will enable NAND gate F and allow the 16 Hz clock pulses, generated by the 555 timer circuit, through to NAND G and on to the frequency divider.

424 CHAPTER 10 / SET-RESET AND DATA-TYPE FLIP-FLOPS

FIGURE 10-20 Troubleshooting a Frequency Divider Circuit Using D-Type Flip Flops.

At the end of each digital circuits chapter, an *application circuit* combines all of the functions discussed in the chapter, so the student can see how all of the building blocks can be put to practical use.



SELF-TEST EVALUATION POINT FOR SECTION 6-3

Now that you have completed this section, you should be able to:

Objective 7. Describe how to repair a circuit once a faulty component has been identified.

Use the following questions to test your understanding:

1. List the steps you would follow to remove an IC from a PCB.
2. What is a "final test"?

6-4 PLD SOFTWARE SIMULATION

In Chapter 4 you were introduced to a Quick Start procedure that will be used throughout this text to create and load a variety of digital logic circuits into the CPLD IC on your test board. Whether you are using the PLD software from Altera or Xilinx, the five-step process for creating a digital circuit prototype using a PLD will be the same. These steps are

PLD Prototyping Process

Step 1: Create the new circuit using the software's schematic editor.

Step 2: Compile the circuit into a bitstream file that when loaded into the CPLD will instruct it to act like the entered schematic.

Step 3: Verify the operation of your circuit using the software's functional and timing simulator.

Step 4: Download the circuit file from the PC to the PLD.

Step 5: Physically test the PLD by activating its inputs and monitoring its outputs.

The software simulator, listed in step 3, is a useful tool that can make you aware of circuit errors before you program the CPLD. You can perform both a *functional simulation* and *timing simulation* on your newly created and compiled circuit.

A **functional simulation** checks the logical operation of the circuit by showing what binary outputs will occur for each of the binary input combinations. This type of simulation is useful for a quick analysis of the circuit's behavior but does not take into account any errors that could occur due to propagation delays.

A **timing simulation** is by far the most crucial part of the verification process, since it simulates the responses you will get from the circuit once it has been implemented within the PLD. The timing simulation takes into account the delays associated with the PLD's internal logic gates, routing capacitances, and input impedances.

The results from the simulation of the circuit can be printed out and used later in step 5 to compare the simulated behavior to the real behavior of the circuit. The following Quick Start procedures have been modified to include the steps you should follow to perform a functional simulation of a logic gate.

Functional Simulation A basic circuit simulation mode that checks the logical performance of a project, irrespective of timing considerations.

Timing Simulation A detailed circuit simulation mode that checks delays introduced by the internal circuitry.

SEC. 6-4 / PLD SOFTWARE SIMULATION 197

FIGURE 6-5 Testing with the Logic Probe.

detecting a single momentary pulse (single pulse) or a pulse train (sequence of pulses), such as a clock signal. Nearly all logic probes have an internal low-frequency oscillator that will flash the "PULSE" LED ON and OFF if a pulse train is detected at a frequency of up to 100 MHz. The logic probe can also detect if a single pulse has occurred, even if the width of the pulse is so small that the operator was unable to see the PULSE LED blink. In this instance, a memory circuit can be enabled using the MEMORY switch, so that any pulse (even as narrow as 10 ns) will turn ON and keep ON the PULSE LED. This memory circuit within the logic probe senses the initial voltage level present at the test point (either HIGH or LOW) and then turns ON the PULSE LED if there is a logic transition or change. To reset this memory circuit, simply toggle (switch OFF and then ON) the memory switch. Figure 6-5(c) summarizes the operation and display interpretation of a typical logic probe.

Even though the logic probe is not as accurate as the multimeter, and cannot determine as much about a signal as the oscilloscope, it is ideal for digital circuit troubleshooting because of its low cost, small size, and versatility.

SEC. 6-1 / DIGITAL TEST EQUIPMENT 183

A strong *testing, test equipment, and troubleshooting* emphasis prepares the technician student for the working world. Extensive troubleshooting techniques and procedures are applied to all combined chapter application circuits, preparing the student for the work environment.

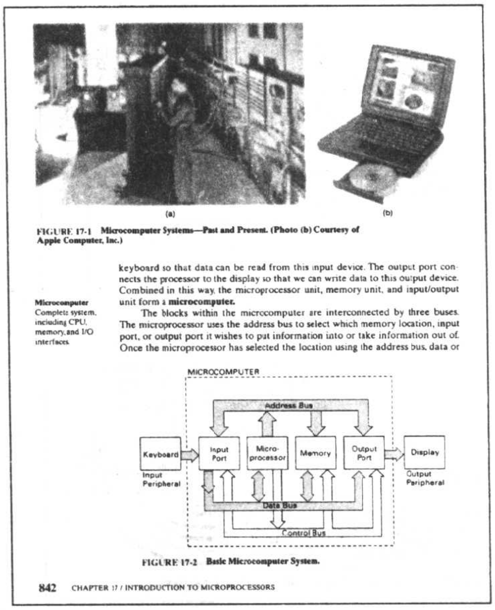

(a) (b)

FIGURE 17-1 Microcomputer Systems—Past and Present. (Photo (b) Courtesy of Apple Computer, Inc.)

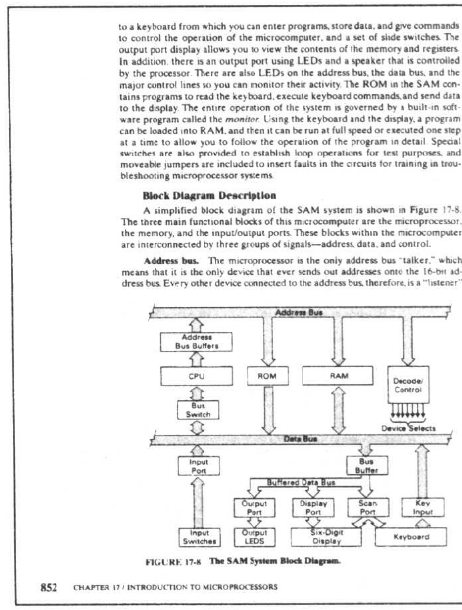
keyboard so that data can be read from this input device. The output port connects the processor to the display so that we can write data to this output device. Combined in this way, the microprocessor unit, memory unit, and input/output unit form a **microcomputer.**

Microcomputer Complete system, including CPU, memory, and I/O interfaces.

The blocks within the microcomputer are interconnected by three buses. The microprocessor uses the address bus to select which memory location, input port, or output port it wishes to put information into or take information out of. Once the microprocessor has selected the location using the address bus, data or

FIGURE 17-2 Basic Microcomputer System.

842 CHAPTER 17 / INTRODUCTION TO MICROPROCESSORS


to a keyboard from which you can enter programs, store data, and give commands to control the operation of the microcomputer, and a set of slide switches. The output port display allows you to view the contents of the memory and registers. In addition, there is an output port using LEDs and a speaker that is controlled by the processor. There are also LEDs on the address bus, the data bus, and the major control lines so you can monitor their activity. The ROM in the SAM contains programs to read the keyboard, execute keyboard commands, and send data to the display. The entire operation of the system is governed by a built-in software program called the *monitor.* Using the keyboard and the display, a program can be loaded into RAM, and then it can be run at full speed or executed one step at a time to allow you to follow the operation of the program in detail. Special switches are also provided to establish loop operations for test purposes, and moveable jumpers are included to insert faults in the circuits for training in troubleshooting microprocessor systems.

Block Diagram Description

A simplified block diagram of the SAM system is shown in Figure 17-8. The three main functional blocks of this microcomputer are the microprocessor, the memory, and the input/output ports. These blocks within the microcomputer are interconnected by three groups of signals—address, data, and control.

Address bus. The microprocessor is the only address bus "talker," which means that it is the only device that ever sends out addresses onto the 16-bit address bus. Every other device connected to the address bus, therefore, is a "listener"

FIGURE 17-8 The SAM System Block Diagram.

852 CHAPTER 17 / INTRODUCTION TO MICROPROCESSORS

The final chapter contains a detailed *component-level description of a microprocessor-based system*, including schematic diagram interpretation, troubleshooting procedures, and typical system faults.

HARDWARE AND SOFTWARE

By using an inexpensive personal computer, an application software program from Altera and/or Xilinx, and a PLD board, you can easily prototype a digital circuit. The following describes the hardware and software needed for PLD prototyping in a little more detail and includes contact information.

Software from Xilinx

I want to thank Nigel Cook, Prentice Hall, and Scott Sambucci for allowing the Xilinx University Program the opportunity to participate in the creation of this text and the accompanying lab book. Nigel has added the PLD Alternatives sections to the text. PLD (an abbreviation for Programmable Logic Device) is the state-of-the-art technology for digital designs both now and for the foreseeable future. It is a design technology that is growing at more than 30% per year and it is well worth your time to learn the basics of this design methodology.

In this text, as well as the accompanying lab book, there are many examples of how digital functions can be realized in programmable logic technology. The lab book can come bundled with the Xilinx Student Edition upon request. Contact your local Prentice Hall Sales Representative to take advantage of this offer. This software is sold and supported by Prentice Hall. For more information on this product, please visit www.prenhall.com.

Xilinx sponsors a special academic website at: www.xup.msu.edu. This website is meant for use by the academic community at large. Please use this website, as it has a wealth of resources for instructors and students alike. There are tutorials, reference designs, FAQs and much, much more. For more information on the Xilinx University Program, visit university.xilinx.com. This website explains what the Xilinx University Program is all about and has many resources, including third-party vendors that manufacture software and demo boards that can be used with Xilinx products.

Patrick Kane
Xilinx University Program Manager

Hardware for Xilinx

Every useful device created by humankind, from the first ancient lever to the most modern computer, began as an abstract idea that was reduced to a physical reality through application of a rigorous design process. The creation of a prototype, or early "working version" of the device, is central to an effective design process. A prototype allows a designer to interact with a new device early in the design process, so that initial design assumptions can be challenged and tested, and a deeper understanding of the design requirements can be forged before the final device is constructed. Recently, with the advent of high-density programmable logic devices called Field-Programmable Gate Arrays (or just FPGAs), it has become possible for digital designers to gain hands-on experience with complex circuits very early in the design cycle.

In an effort to bring useful FPGA technology into the hands of every circuit designer, the Digilent Company (www.digilent.cc) has introduced a circuit board designed to facilitate the creation of circuit prototypes. Called the Digilab board, it allows circuit designers to quickly and easily implement a wide variety of digital circuits right on the desktop, and without the need for any additional equipment. The board uses a high-capacity FPGA and a large collection of I/O devices to form a stand-alone design platform that can be used to create thousands of useful circuits, without the need for any additional components. The board is shipped with a power supply and programming cable—everything needed to begin creating circuits immediately. A complete set of reference materials, technical guidelines, and worked design problems are maintained at the manufacturer's website.

Software from Altera

The lab book provides students with contact information for obtaining Altera's MAX+PLUS II Student Edition programmable logic development software. MAX+PLUS II is a fully integrated design environment that offers unmatched flexibility and performance. The intuitive graphical interface is complemented by complete and instantly accessible on-line documentation, which makes learning and using MAX+PLUS II quick and easy. By entering the designs presented in the book or creating custom logic designs, students develop skills for prototyping digital systems using programmable logic devices.

For more information on Altera Corporation, its University Program, or its products, the reader is referred to the website www.altera.com.

Hardware for Altera—The SSEI FPGA Educational Board ONE

Key Features

- Reprogrammable Altera EPM7128 CPLD with 128 macrocells
- On-board Byteblaster circuit for easy programming through 25-pin D interface when connected to PC parallel port
- 33 binary user inputs consisting of 2 8-bit dip switches, 10 push-buttons, a 4-bit rotary hex switch, and a reset push-button
- 16 LEDs for visual checking, connected in parallel with 20-pin header for external I/O
- 2-digit seven-segment LED display
- 1 MHz oscillator configured for easy divide down in the CPLD
- 2-bit rotary encoder
- External SPI interface through 8-pin header
- External keypad interface through 7-pin header
- On-board 3.3V voltage regulator

Description

The FPGA Educational Board One (EB1) is intended to support a variety of digital logic experiments that might be encountered in an introductory digital electronics course. Designed around an Altera EPM7128 CPLD (Complex Programmable Logic Device), the board also includes an assortment of user inputs and visual indicators as shown in the block diagram on p. xvi. The flexibility of programmable internal routing and logic allows the external components to be soft-wired in a variety of interesting and instructive configurations. Flexible programmable clocking of the CPLD is accomplished by connecting the global clock input to an I/O instead of to the oscillator directly. The 2-bit rotary encoder can be used as a slow user-variable clock as well as an interface example. Wherever possible, the CPLD is protected from drive contention that may result from programming errors. Reprogramming is performed using Altera software running on a PC connected to the board through a parallel cable. A wall-mounted DC transformer supplies power.

此为试读，需要完整PDF请访问： www.ertongbook.com