国外电子与通信教材系列

专用集成电路

Application-Specific Integrated Circuits

英文版

[美] Michael John Sebastian Smith 著

Addison-Wesley

PHEI

電子工業出版社
Publishing House of Electronics Industry
www.phei.com.cn

国外电子与通信教材系列

专用集成电路

（英文版）

Application-Specific Integrated Circuits

[美] Michael John Sebastian Smith 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是一本有关专用集成电路（ASIC）的综合性和权威性书籍。书中叙述了VLSI系统设计的最新方法。利用商业化工具以及预先设计好的单元库已使得ASIC设计成为速度最快、成本最低而且错误最少的一种IC设计方法，因而ASIC和ASIC设计方法已迅速在工业界的各个应用领域得到推广。

本书介绍了半定制和可编程的ASIC。在对每种ASIC类型的数字逻辑设计与物理特性的基本原理进行描述后，讨论了ASIC逻辑设计——设计输入、逻辑综合、仿真以及测试，并进一步讲述了相应的物理设计——划分、平面布图规划、布局以及布线。此外，本书对在ASIC设计中需要了解的各方面知识以及必需的工作都有详尽叙述。

本书可作为大学高年级和研究生教材，也是ASIC领域工程技术人员的理想参考书。

版权贸易合同登记号：图字：01-2002-2740

图书在版编目（CIP）数据

专用集成电路 = Application-Specific Integrated Circuits/（美）史密斯（Smith, M. J. S.）著.
-北京：电子工业出版社，2003.1
（国外电子与通信教材系列）
ISBN 7-5053-8407-4

Ⅰ.专... Ⅱ.史 Ⅲ.集成电路-教材-英文 Ⅳ. TN4

中国版本图书馆CIP数据核字（2002）第105501号

责任编辑：周宏敏
印 刷 者：北京东光印刷厂
出版发行：电子工业出版社 http://www.phei.com.cn
北京市海淀区万寿路173信箱 邮编：100036
经 销：各地新华书店
开 本：787 × 980 1/16 印张：65.5 字数：1677千字
版 次：2003年1月第1版 2003年1月第1次印刷
定 价：86.00元

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿

中国工程院院士、清华大学教授

“国外电子与通信教材系列”出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套“国外计算机科学教材系列”，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于“十五”期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进“国外电子与通信教材系列”，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，“国外电子与通信教材系列”的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过“教育部高等学校电子信息科学与工程类专业教学指导委员会”的审核，并得到教育部高等教育司的批准，纳入了“教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书”。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

PREFACE

In 1988 I began to teach full-custom VLSI design. In 1990 I started teaching ASIC design instead, because my students found it easier to get jobs in this field. I wrote a proposal to The National Science Foundation (NSF) to use electronic distribution of teaching material. Dick Lyon helped me with preparing the first few CD-ROMs at Apple, but Chuck Seitz, Lynn Conway, and others explained to me that I was facing a problem that Carver Mead and Lynn had experienced in trying to get the concept of multichip wafers adopted. It was not until the publication of the Mead–Conway text that people accepted this new idea. It was suggested that I must generate interest using a conventional format before people would use my material in a new one (CD-ROM or the Internet). In 1992 I stopped writing papers and began writing this book—a result of my experiments in computer-based education. I have nearly finished this book twice. The first time was a copy of my notes. The second time was just before the second edition of Weste and Eshragian was published—a hard act to follow. In order to finish in 1997 I had to stop updating and including new ideas and material and now this book consists of three parts: Chapters 1–8 are an introduction to ASICs, 9–14 cover ASIC logical design, and 15–17 cover the physical design of ASICs.

The book is intended for a wide audience. It may be used in an undergraduate or graduate course. It is also intended for those in industry who are involved with ASICs. Another function of this book is an "ASIC Encyclopedia," and therefore I have kept the background material needed to a minimum. The book makes extensive use of industrial tools and examples. The examples in Chapters 2 and 3 use tools and libraries from MicroSim (PSpice), Meta Software (HSPICE), Compass Design Automation (standard-cell and gate-array libraries), and Tanner Research (L-Edit). The programmable ASIC design examples in Chapter 4–8 use tools from Compass, Synopsys, Actel, Altera, and Xilinx. The examples in Chapter 9 (covering low-level design entry) used tools from Exemplar, MINC, AMD, UC Berkeley, Compass, Capilano, Mentor Graphics Corporation, and Cadence Design Systems. The VHDL examples in Chapter 10 (VHDL) were checked using QuickVHDL from Mentor, V-System Plus from Model Technology, and Scout from Compass. The Verilog examples in Chapter 11 were checked using Verilog-XL from Cadence, V-System Plus, and VeriWell from Wellspring Solutions. The logic synthesis examples in

Chapter 12 were checked with the ASIC Synthesizer product family from Compass and tools from Mentor, Synopsys, and UC Berkeley. The simulation examples in Chapter 13 were checked with QuickVHDL, V-System/Plus, PSpice, Verilog-XL, DesignWorks from Capilano Computing, CompassSim, QSim, MixSim, and HSPICE. The test examples in Chapter 14 were checked using test software from Compass, Cadence, Mentor, Synopsys and Capilano's DesignWorks. The physical design examples in Chapters 15–17 were generated and tested using Preview, Gate Ensemble, and Cell Ensemble (Cadence) as well as ChipPlanner, ChipCompiler, and PathFinder (Compass). All these tools are installed at the University of Hawaii.

I wrote the text using FrameMaker. This allows me to project the text and figures using an LCD screen and an overhead projector. I used a succession of Apple Macintosh computers: a PowerBook 145, a 520, and lastly a 3400 with 144MB of RAM, which made it possible for me to create updates to the index in just under one minute. Equations are "live" in FrameMaker. Thus,

$$\text{book thickness} = \#\text{pages} \times 0.0015 \text{ in./page} \approx (1000)\ (1.5 \times 10^{-3}) = 1.5 \text{ in.}$$

can be updated in a lecture and the new result displayed. The circuit layouts are color EPS files with enhanced B&W PICT previews created using L-Edit from Tanner Research. All of the Verilog and VHDL code examples, compiler and simulation input/output, and the layout CIF that were used in the final version are included as conditional (hidden) text in the FrameMaker document, which is approximately 200MB and just over 6,000 pages (my original source material spans fourteen 560MB optical disks). Software can operate on the hidden text, allowing, for example, a choice of simulators to run the HDL code live in class. I converted draft versions of the VHDL and Verilog LRMs and related standards to FrameMaker and built hypertext links to my text, but copyright problems will have to be solved before this type of material may be published. I drew all the figures using FreeHand. They are "layered" allowing complex drawings to be built-up slowly or animated by turning layers on or off. This is difficult to utilize in book form, but can be done live in the classroom.

A course based on FPGAs can use Chapter 1 and Chapters 4–8. A course using commercial semicustom ASIC design tools may use Chapters 1–2 or Chapters 1–3 and then skip to Chapter 9 if you use schematic entry, Chapter 10 (if you use VHDL), or Chapter 11 (if you use Verilog) together with Chapter 12. All classes can use Chapters 13 and 14. FPGA-based classes may skim Chapters 15–17, but classes in semicustom design should cover these chapters. The chapter dependencies—Y (X) means Chapter Y depends on X—are approximately: 1, 2(1), 3(2), 4(2), 5(4), 6(5), 7(6), 8(7), 9(2), 10(2), 11(2), 12(10 or 11), 13(2), 14(13), 15(2), 16(15), 17(16).

I used the following references to help me with the orthography of complex terms, style, and punctuation while writing: *Merriam-Webster's Collegiate Dictionary,* 10th edition, 1996, Springfield, MA: Merriam-Webster, ISBN 0-87779-709-9, PE1628.M36; *The Chicago Manual of Style,* 14th edition, Chicago: University of

Chicago Press, 1993, ISBN 0-226-10389-7, Z253.U69; and *Merriam-Webster's Standard American Style Manual,* 1985, Springfield, MA: Merriam-Webster, ISBN 0-87779-133-3, PN147.W36. A particularly helpful book on technical writing is *BUGS in Writing* by Lyn Dupré, 1995, Reading, MA: Addison-Wesley, ISBN 0-201-60019-6, PE1408.D85 (Lyn's book grew from her unpublished work, Style SomeX, which I used).

The bibliography at the end of each chapter provides alternative sources if you cannot find what you are looking for. I have included the International Standard Book Number[1] (ISBN) and Library of Congress (LOC) Call Number for books, and the International Standard Serial Number[2] (ISSN) for journals (see the LOC information system, LOCIS, at `http://www.loc.gov`). I did not include references to material that I could not find myself (except where I have noted in the case of new or as yet unpublished books). The electronic references given in this text have (a last) access date of 4/19/97 and omit enclosing <> if the reference does not include spaces.

I receive a tremendous level of support and cooperation from industry in my work. I thank the following for help with this project: Cynthia Benn and Lyn Dupré for editing; Helen Goldstein, Peter Gordon, Susan London-Payne, Tracy Russ, and Juliet Silveri, all at Addison-Wesley; Matt Bowditch and Kim Arney at Argosy; Richard Lyon, Don North, William Rivard, Glen Stone, the managers of the Newton group, and many others at Apple Computer who provided financial support; Apple for providing support in the form of software and computers; Bill Becker, Fern Forcier, Donna Isidro, Mike Kliment, Paul McLellan, Tom Schaefer, Al Stein, Rich Talburt, Bill Walker, and others at Compass Design Automation and VLSI Technology for providing the opportunity for me to work on this book over many years and allowing me to test material inside these companies and on lecture tours they sponsored; Chuck Seitz at Caltech; Joseph Cavallaro, Bernie Chern, Jerry Dillion, Mike Foster, and Paul Hulina at the NSF; the NSF for financial support with a Presidential Young Investigator Award; Jim Rowson and Doug Fairbairn; Constantine Anagnostopolous, Pin Tschang and members of the ASIC design groups at Kodak for financial support; the disk-drive design group at Digital Equipment Corp. (Massachusetts), Hewlett-Packard, and Sun Microsystems for financial support; Ms. MOSIS and all of the staff at MOSIS who each have helped me at one point or another by providing silicon, technical support, and documentation; Bob Brodersen, Roger Howe, Randy Katz, and Ed Lee of UC Berkeley for help while I was visiting UCB; James Plummer of Stanford, for providing me with access to the Terman Engineering Library as a visiting scholar, as well as Abbas El Gamal and Paul Losleben, also at Stanford, for help on several occasions; Don Bouldin at University of Tennessee; Krzysztof Kozminski at MCNC for providing Uncle lay-

[1] A code that uniquely identifies a book, the tenth and last digit is a check digit.

[2] This number uniquely identifies a serial (a magazine, a journal, and so on). It is a seven-digit number with an eighth check digit (which may be the roman numeral X, the value ten).

out software; Gershom Kedem at Duke University for the public domain tools his group has written; Sue Drouin, José De Castro, and others at Mentor Graphics Corporation in Oregon for providing documentation and tools; Vahan Kasardjhan, Gail Grego, Michele Warthen, Steve Gardner, and others at the University Program at Cadence Design Systems in San Jose who helped with tools, documentation, and support; Karen Dorrington and the Cadence group in Massachusetts; Andy Haines, Tom Koppin, Sherri Mieth, Velma Miller, Robert Nalesnik, Mike Sarpa, Telle Whitney, and others at Actel for software, hardware, parts, and documentation; Peter Alfke, Leslie Baxter, Brad Fawcett, Chris Kingsley, Karlton Lau, Rick Mitchell, Scott Nance, and Richard Ravel at Xilinx for support, parts, software, and documentation; Greg Hedmann at NorComp for data on FPGAs; Anna Acevedo, Suzanne Bailey, Antje MacNaughton, Richard Terrell, and Altera for providing software, hardware programmers, parts, and documentation; the documentation group and executive management at LSI Logic for tools, libraries, and documentation; Toshiba, NEC, AT&T/NCR, Lucent, and Hitachi (for documentation); NEC for their visiting scholar program at UH; Fred Furtek, Oscar Naval, and Claire Pinkham at Concurrent Logic, Randy Fish at Crosspoint, and Gary Banta at Plus Logic—all for documentation; Paul Titchener and others at Comdisco (now part of Cadence Design Systems) for providing design tools; John Tanner and his staff at Tanner Research for providing their tools and documentation; Mahendra Jain and Nanci Magoun, who let me debug early prototypes at the IDEA conference organized by ASIC Technology and News; Exemplar for providing documentation on its tools; MINC for providing a copy of its FPGA software and documentation; Claudia Traver and Synopsys for tools and documentation; Mentor Graphics Corporation for providing its complete range of software; Alain Hanover and others at ViewLogic for providing tools; Mary Shepherd and Jerry Walker at IEEE for help with permissions; Meta Software for providing HSPICE; Chris Dewhurst and colleagues at Capilano Computing for its design tools; Greg Seltzer (Model Technology) and Charley Rowley for providing V-System Plus with online documentation prototypes; Farallon and Telebit for the software and hardware I used for early experiments with telelectures. Many research students at the University of Hawaii helped me throughout this project including: Chin Huang, Clem Portmann, Christeen Gray, Karlton Lau, Jon Otaguro, Moe Lwin, Troy Stockstad, Ron Jorgenson, Derwin Mattos, William Rivard, Wendy Ching, Anil Aggarwal, Sudhakar Jilla, Linda Xu, Angshuman Saha, Harish Pareek, Claude van Ham, Wen Huang, Kumar Vadhri, Yan Zhong, Yatin Acharya, and Barana Ranaweera. Each of the classes that used early versions of this text at the University of Hawaii at Manoa have also contributed by finding errors. The remaining errors are mine.

Links to figures, software, code, problem solutions, and other resources for this book may be found at:

`http://www.awl.com/cp/authors/smithm/asics/asics.html`.

Michael John Sebastian Smith
Palo Alto and Honolulu, 1997

目录概览

INTRODUCTION TO ASICs

1

An **ASIC** (pronounced "a-sick"; bold typeface defines a new term) is an **application-specific integrated circuit**—at least that is what the acronym stands for. Before we answer the question of what *that* means we first look at the evolution of the silicon chip or **integrated circuit (IC)**.

Figure 1.1(a) shows an IC package (this is a pin-grid array, or PGA, shown upside down; the pins will go through holes in a printed-circuit board). People often call the package a chip, but, as you can see in Figure 1.1(b), the silicon chip itself (more properly called a **die**) is mounted in the cavity under the sealed lid. A PGA package is usually made from a ceramic material, but plastic packages are also common.

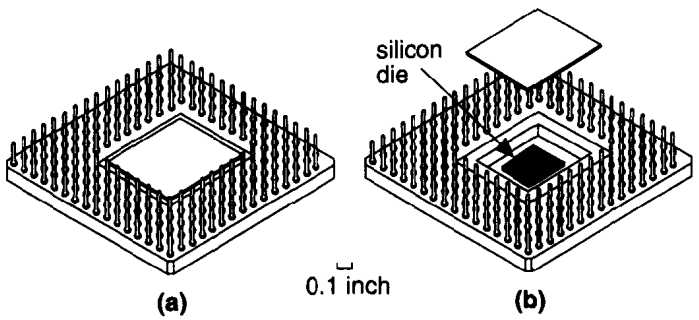

FIGURE 1.1 An integrated circuit (IC). (a) A pin-grid array (PGA) package. (b) The silicon die or chip is under the package lid.

The physical size of a silicon die varies from a few millimeters on a side to over 1 inch on a side, but instead we often measure the size of an IC by the number of logic gates or the number of transistors that the IC contains. As a unit of measure a **gate equivalent** corresponds to a two-input NAND gate (a circuit that performs the logic function, $F = \overline{A \cdot B}$). Often we just use the term *gates* instead of gate equivalents when we are measuring chip size—not to be confused with the gate terminal of a transistor. For example, a 100 k-gate IC contains the equivalent of 100,000 two-input NAND gates.

The semiconductor industry has evolved from the first ICs of the early 1970s and matured rapidly since then. Early **small-scale integration** (**SSI**) ICs contained a few (1 to 10) logic gates—NAND gates, NOR gates, and so on—amounting to a few tens of transistors. The era of **medium-scale integration** (**MSI**) increased the range of integrated logic available to counters and similar, larger scale, logic functions. The era of **large-scale integration** (**LSI**) packed even larger logic functions, such as the first microprocessors, into a single chip. The era of **very large scale integration** (**VLSI**) now offers 64-bit microprocessors, complete with cache memory and floating-point arithmetic units—well over a million transistors—on a single piece of silicon. As CMOS process technology improves, transistors continue to get smaller and ICs hold more and more transistors. Some people (especially in Japan) use the term **ultra large scale integration** (**ULSI**), but most people stop at the term VLSI; otherwise we have to start inventing new words.

The earliest ICs used **bipolar technology** and the majority of logic ICs used either **transistor–transistor logic** (**TTL**) or emitter-coupled logic (ECL). Although invented before the bipolar transistor, the **metal-oxide-silicon** (**MOS**) transistor was initially difficult to manufacture because of problems with the oxide interface. As these problems were gradually solved, metal-gate *n*-channel MOS (**nMOS** or **NMOS**) technology developed in the 1970s. At that time MOS technology required fewer masking steps, was denser, and consumed less power than equivalent bipolar ICs. This meant that, for a given performance, an MOS IC was cheaper than a bipolar IC and led to investment and growth of the MOS IC market.

By the early 1980s the aluminum gates of the transistors were replaced by polysilicon gates, but the name MOS remained. The introduction of polysilicon as a gate material was a major improvement in CMOS technology, making it easier to make two types of transistors, *n*-channel MOS and *p*-channel MOS transistors, on the same IC—a **complementary MOS** (**CMOS**, never cMOS) technology. The principal advantage of CMOS over NMOS is lower power consumption. Another advantage of a polysilicon gate was a simplification of the fabrication process, allowing devices to be scaled down in size.

There are four CMOS transistors in a two-input NAND gate (and a two-input NOR gate too), so to convert between gates and transistors, you multiply the number of gates by 4 to obtain the number of transistors. We can also measure an IC by the smallest **feature size** (roughly half the length of the smallest transistor) imprinted on the IC. Transistor dimensions are measured in microns (a micron, 1 μm, is a mil-

lionth of a meter). Thus we talk about a 0.5 μm IC or say an IC is built in (or with) a 0.5 μm process, meaning that the smallest transistors are 0.5 μm in length. We give a special label, λ or **lambda**, to this smallest feature size. Since lambda is equal to half of the smallest transistor length, $\lambda \approx 0.25$ μm in a 0.5 μm process. Many of the drawings in this book use a scale marked with lambda for the same reason we place a scale on a map.

A modern submicron CMOS process is now just as complicated as a submicron bipolar or BiCMOS (a combination of bipolar and CMOS) process. However, CMOS ICs have established a dominant position, are manufactured in much greater volume than any other technology, and therefore, because of the economy of scale, the cost of CMOS ICs is less than a bipolar or BiCMOS IC for the same function. Bipolar and BiCMOS ICs are still used for special needs. For example, bipolar technology is generally capable of handling higher voltages than CMOS. This makes bipolar and BiCMOS ICs useful in power electronics, cars, telephone circuits, and so on.

Some digital logic ICs and their analog counterparts (analog/digital converters, for example) are **standard parts**, or standard ICs. You can select standard ICs from catalogs and data books and buy them from distributors. Systems manufacturers and designers can use the same standard part in a variety of different **microelectronic systems** (systems that use microelectronics or ICs).

With the advent of VLSI in the 1980s engineers began to realize the advantages of designing an IC that was customized or tailored to a particular system or application rather than using standard ICs alone. Microelectronic system design then becomes a matter of defining the functions that you can implement using standard ICs and then implementing the remaining logic functions (sometimes called **glue logic**) with one or more **custom ICs**. As VLSI became possible you could build a system from a smaller number of components by combining many standard ICs into a few custom ICs. Building a microelectronic system with fewer ICs allows you to reduce cost and improve reliability.

Of course, there are many situations in which it is not appropriate to use a custom IC for each and every part of an microelectronic system. If you need a large amount of memory, for example, it is still best to use standard memory ICs, either **dynamic random-access memory** (**DRAM** or dRAM), or **static RAM** (**SRAM** or sRAM), in conjunction with custom ICs.

One of the first conferences to be devoted to this rapidly emerging segment of the IC industry was the *IEEE Custom Integrated Circuits Conference* (CICC), and the proceedings of this annual conference form a useful reference to the development of custom ICs. As different types of custom ICs began to evolve for different types of applications, these new ICs gave rise to a new term: application-specific IC, or ASIC. Now we have the *IEEE International ASIC Conference*, which tracks advances in ASICs separately from other types of custom ICs. Although the exact definition of an ASIC is difficult, we shall look at some examples to help clarify what people in the IC industry understand by the term.

Examples of ICs that are *not* ASICs include standard parts such as: memory chips sold as a commodity item—ROMs, DRAM, and SRAM; microprocessors; TTL or TTL-equivalent ICs at SSI, MSI, and LSI levels.

Examples of ICs that *are* ASICs include: a chip for a toy bear that talks; a chip for a satellite; a chip designed to handle the interface between memory and a microprocessor for a workstation CPU; and a chip containing a microprocessor as a cell together with other logic.

As a general rule, if you can find it in a data book, then it is probably not an ASIC, but there are some exceptions. For example, two ICs that might or might not be considered ASICs are a controller chip for a PC and a chip for a modem. Both of these examples are specific to an application (shades of an ASIC) but are sold to many different system vendors (shades of a standard part). ASICs such as these are sometimes called **application-specific standard products** (**ASSPs**).

Trying to decide which members of the huge IC family are application-specific is tricky—after all, every IC has an application. For example, people do not usually consider an application-specific microprocessor to be an ASIC. I shall describe how to design an ASIC that may include large cells such as microprocessors, but I shall not describe the design of the microprocessors themselves. Defining an ASIC by looking at the application can be confusing, so we shall look at a different way to categorize the IC family. The easiest way to recognize people is by their faces and physical characteristics: tall, short, thin. The easiest characteristics of ASICs to understand are physical ones too, and we shall look at these next. It is important to understand these differences because they affect such factors as the price of an ASIC and the way you design an ASIC.

1.1 Types of ASICs

ICs are made on a thin (a few hundred microns thick), circular silicon **wafer**, with each wafer holding hundreds of die (sometimes people use dies or dice for the plural of die). The transistors and wiring are made from many layers (usually between 10 and 15 distinct layers) built on top of one another. Each successive **mask layer** has a pattern that is defined using a **mask** similar to a glass photographic slide. The first half-dozen or so layers define the transistors. The last half-dozen or so layers define the metal wires between the transistors (the **interconnect**).

A **full-custom IC** includes some (possibly all) logic cells that are customized and all mask layers that are customized. A microprocessor is an example of a full-custom IC—designers spend many hours squeezing the most out of every last square micron of microprocessor chip space by hand. Customizing all of the IC features in this way allows designers to include analog circuits, optimized memory cells, or mechanical structures on an IC, for example. Full-custom ICs are the most expen-

sive to manufacture and to design. The **manufacturing lead time** (the time it takes just to make an IC—not including design time) is typically eight weeks for a full-custom IC. These specialized full-custom ICs are often intended for a specific application, so we might call some of them full-custom ASICs.

We shall discuss full-custom ASICs briefly next, but the members of the IC family that we are more interested in are **semicustom ASICs**, for which all of the logic cells are predesigned and some (possibly all) of the mask layers are customized. Using predesigned cells from a **cell library** makes our lives as designers much, much easier. There are two types of semicustom ASICs that we shall cover: standard-cell–based ASICs and gate-array–based ASICs. Following this we shall describe the **programmable ASICs**, for which all of the logic cells are predesigned and none of the mask layers are customized. There are two types of programmable ASICs: the programmable logic device and, the newest member of the ASIC family, the field-programmable gate array.

1.1.1 Full-Custom ASICs

In a **full-custom ASIC** an engineer designs some or all of the logic cells, circuits, or layout specifically for one ASIC. This means the designer abandons the approach of using pretested and precharacterized cells for all or part of that design. It makes sense to take this approach only if there are no suitable existing cell libraries available that can be used for the entire design. This might be because existing cell libraries are not fast enough, or the logic cells are not small enough or consume too much power. You may need to use full-custom design if the ASIC technology is new or so specialized that there are no existing cell libraries or because the ASIC is so specialized that some circuits must be custom designed. Fewer and fewer full-custom ICs are being designed because of the problems with these special parts of the ASIC. There is one growing member of this family, though, the mixed analog/digital ASIC, which we shall discuss next.

Bipolar technology has historically been used for precision analog functions. There are some fundamental reasons for this. In all integrated circuits the matching of component characteristics between chips is very poor, while the matching of characteristics between components on the same chip is excellent. Suppose we have transistors T1, T2, and T3 on an analog/digital ASIC. The three transistors are all the same size and are constructed in an identical fashion. Transistors T1 and T2 are located adjacent to each other and have the same orientation. Transistor T3 is the same size as T1 and T2 but is located on the other side of the chip from T1 and T2 and has a different orientation. ICs are made in batches called wafer lots. A **wafer lot** is a group of silicon wafers that are all processed together. Usually there are between 5 and 30 wafers in a lot. Each wafer can contain tens or hundreds of chips depending on the size of the IC and the wafer.