CAD/CAM TECHNIQUES

Michael F. Hordeski

CAD/CAM TECHNIQUES

Michael F. Hordeski

Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia

Library of Congress Cataloging in Publication Data

Hordeski, Michael F.
CAD/CAM techniques.

Bibliography: p.
Includes index.
1. CAD/CAM systems. I. Title. II. Title:
C.A.D./C.A.M. Techniques.
TS155.6.H67 1985 670'.28'54 84-24965
ISBN 0-8359-0620-5

CAD/CAM has more potential to radically increase productivity than any development since electricity.

The National Science Foundation of the United States

Production supervision/interior design:
Tally Morgan

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

PREFACE

This book on *Computer-Aided Design and Manufacturing Techniques* provides a complete overview of modern design and manufacturing methods which employ computers.

CAD/CAM will play an ever increasing role in design and production circles. It is likely to become one of the major growth areas of the 80s and 90s. International competition, inflation and price factors, the high cost of capital due to rising interest rates, the decreased availability of skilled labor, and the increased emphasis on quality are forcing manufacturers to automate much of their design and production. Service firms that support or supply these manufacturers must also automate many of their functions in order to compete.

The dramatic changes in computing power allow lower costs for equipment that does more. This provides an increased capacity for CAD/CAM techniques at all levels. It also makes a knowledge of CAD/CAM equipment and technology more critical. Manufacturers who do not utilize CAD/CAM techniques are expected to find it difficult to survive in the late 80s and almost impossible to compete in the 90s.

Business will continue to improve in many markets that had been depressed, but the companies that will do best are those that utilize automated design and manufacturing equipment and CAD/CAM techniques. CAD/CAM techniques are of interest to those who are involved in manufacturing or use automated equipment. This includes the users and suppliers of data-processing equipment, software, and services on both the domestic and international fronts.

Parts of the book are written in nontechnical language; these sections will be particularly useful to those involved in the overall management of systems and plants. Other sections, such as Chapters 5 and 6, give details of design in enough depth for practicing enginers and engineering faculty.

Chapter 1 introduces the complete field of CAD/CAM. It covers the major topics of the book.

Chapter 2 continues the topic of interactive graphics with an emphasis on the user interface. Chapter 3 approaches the same subject in terms of the technology and hardware used in present-day systems.

The topic of modeling is the subject of Chapter 4. Electronic circuits are one of the more mature areas in modeling, since a great deal of effort has gone into analytical techniques for computer-aided circuit analysis. Much of this work can be extended into the other areas of modeling and simulation which are discussed in this book.

Chapters 5 and 6 continue the treatment of computer-aided circuit analysis with a discussion of modern analysis techniques and the methods currently available for their application.

Chapter 7 is concerned with a different area of modeling—the geometric models that are used in mechanical design. Chapter 8 goes on to discuss the many mechanical design techniques that are available as tools.

Manufacturing applications are the subject of Chapter 9. Here we are concerned with such subjects as data processing and communications, factory-related software, the use of microcomputers in the factory, industrial networks, machine mobility, robots, unconventional materials, automated handling systems, vision systems, object recognition, interactive systems for plant modernization, and plant floor layout.

Chapter 10 explores the entire future realm of CAD/CAM technology and includes such topics as flexible manufacturing systems (FMS), computer-integrated manufacturing (CIM), automated test and inspection systems, automated storage and the new environment, automated factories, and the new technologies that may be used for interactive systems.

The first three chapters are common to any curriculum that treats the subject of CAD/CAM. The next three chapters are of particular interest in electrical engineering, and Chapters 7 through 9 contain material of interest in mechanical engineering and related disciplines. The final chapter offers material relating to the future which should be of interest to all groups. The exercise and review questions are designed to draw the reader into many of the basic topics and in many cases will require an extension of the material presented. The references listed in the bibliography allow additional research and study of many topics.

This book would not have been possible without the help of a number of others. For their support and assistance on this project I wish to thank Karen Bashara of Adage, Robert Duncan and Terence Binion of Catronix, Kevin Ryan, Joyce Anderson, and Paul Murphy of Lexidata, Bruce Gladstone of FutureNet, Dan Bowman and Ray Barger of Spectragraphics, Douglas Bombay of Engineering Automation Systems, William Ewer of Lasergraphics, Ralph Manildi of Nicolet Computer Graphics, Tom Lazear of T & W Systems, and a special thanks to the red-haired lady who helped in so many ways throughout the preparation of this book.

The present trends in computer automation are aimed at developing unified systems for directing the activities of interconnected groups of robots and other automatic machines. This is expected to pave the way for totally automated factories and offices. Total automation does not mean a factory or office without people but rather one automated to the fullest practical extent.

Most factory and office systems in the future will be made up of workers teamed up with robots and other automated machines. How well these systems operate will depend to a large extent upon the expertise that went into the selection and implementation of the technology. Thus, there is a great need for technical information and expertise in this area, for if computer-aided systems are to be successfully implemented in any situation, they must be fully understood.

We know that the face of business is changing quickly, and so must the management strategies that we have kept, in many instances, far too long. Management must understand the new technologies in order to beneficially employ them without serious disruption of business operations.

MICHAEL HORDESKI, P.E.
Atascadero, California

CONTENTS

INTRODUCTION TO CAD/CAM

1

Computer-aided design and computer-aided manufacturing, or CAD/CAM, can encompass the entire range of engineering activity. The use of computer aids may be also called computer-aided engineering, or CAE. Manufacturers can use many of these activities in an integrated process for the entire range of product design and development. In this process, product development time and cost are reduced substantially by a heavier reliance on computer simulation methods rather than prototype development and testing.

CAD/CAM systems allow most routine engineering tasks to be performed quickly, since they automate the repetitive functions, performing these tasks more quickly and accurately than would ever be possible by manual methods. Since the information is stored in computer memory instead of hard copy, the transfer of data tends to be quicker, more reliable, and less redundant.

CAD/CAM systems free one from these tedious, time-consuming chores that have little to do with ingenuity. Experience has shown that CAD/CAM speeds the design and manufacturing process, while reducing much of the paperwork and repetition that hampers one's productivity and creativity.

In a typical CAD/CAM system, the user interacts with the computer using a graphics terminal, designing and manufacturing a part with information stored in the computer database. With CAD the user constructs a geometric model, analyzes the structure, performs kinematic studies, and produces engineering drawings. Using a CAM system, the user can create instructions for machine tools, produce process plans for fabricating an assembly, program robots to handle tools and workpieces, and coordinate plant operation using a factory management system (Figure 1-1).

More realistic displays make it easier to generate truer representations. Programs allow one to change views, perspectives, and shadow placements. Displays can be animated or highlighted for instructional, promotional, or simulation purposes.

Complex products can be observed as they will appear in final form. Geometric flaws are revealed before the design is transformed into hardware.

CAD/CAM EVOLUTION

Design engineers have been using computers for many decades, but the use of CAD/CAM techniques has increased greatly in the last few years, owing to several related trends. One of the most important of these is the drastic reduction in the cost of computing. This cost reduction makes it practical to solve complex problems and to perform elaborate simulations that previously were beyond the reach and scope of many organizations.

Mass-produced microprocessors have played an important role in this dramatic cost reduction. They evolved from the interaction of technology and the market. Semiconductor manufacturing advanced to the point where a

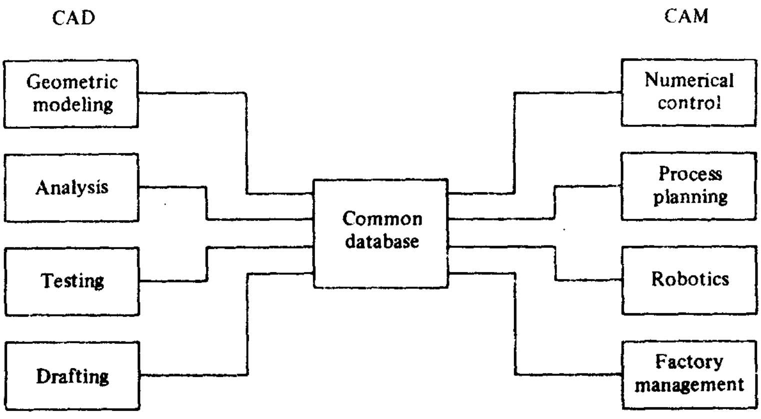

Figure 1-1 CAD/CAM functions.

computer could be fabricated on a chip (Figure 1-2). At the same time the volume of semiconductor products being sold became great enough to support the investment necessary to build and test microprocessors. As the market made microprocessors an attractive product, many quickly appeared. The *micro* part of the name evolved from the fact that the processor is fabricated on one chip, or a small number of chips. The name now has little to do with its capabilities.

The number of transistors that can be included on a 200-mil square chip of silicon provides microprocessors with instruction sets larger than those available in many early minicomputers. Microprocessors are now used in products ranging from simple entertainment devices to sophisticated instruments.

Consumer applications have been important to the industrial marketplace, since they make up a large part of the total market for microprocessors, which helps reduce prices and stimulates development. The effects of volume pricing and product competition are illustrated in the case of the hand-held calculator, which originally cost hundreds of dollars when introduced and today is sometimes given away as an advertising gift.

Microprocessors function in consumer and industrial applications both as controllers and as computers. When used as controllers, they are embedded in the equipment in such a way that their existence is not obvious to a user. The user of a high-speed printer, for example, is not aware that pushing the control buttons initiates the execution of a program in a microprocessor. The justification for the microprocessor in these applications is the equal or greater functional capability it provides at lower costs than the alternatives.

The low cost of mass-produced, general-purpose microprocessors affords flexible components which may be adapted through programming to

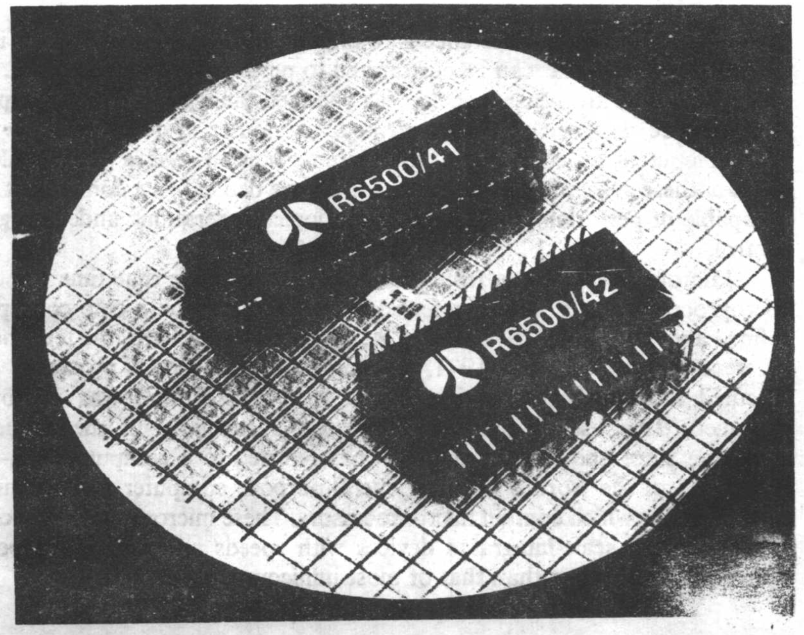

Figure 1-2 Intelligent peripheral controllers. Microprocessor devices such as these have made CAD/CAM hardware more affordable. (Courtesy: Rockwell International)

provide the different functions required by a wide variety of products.

The machine-language instructions required by any computer are a limiting factor in productivity. Higher-level languages, along with improved system software, have made the computer easier and simpler to use. Many high-level languages are currently available. The limitations of available software have been due in large measure to the recent emergence of microcomputers. As software developers can expect more satisfactory means of recovering their program-development investments, more substantial software support for microcomputers will be offered.

The other components of a computer, the peripherals, particularly those with electromechanical parts, initially provided a sharp contrast to the processors. But mass production has brought the price of peripherals down to the point that these devices represent costs similar to those of the processor. In recent years, less-expensive peripheral equipment has been designed specifically for use with microcomputers. The level of performance of these peripherals is such that it integrates well with the performance of most microcomputers currently on the market.

Microcomputers differ from larger computers in application and experience. The biggest difference is in philosophy: the microprocessor is an inexpensive device that can be customized to perform a specialized task. The existence of this inexpensive device has changed the way in which computer solutions are designed and implemented. Computer systems can now use microcomputers as components for system solutions. As a basic building block of these systems, a microprocessor can be defined as a component that is capable of performing arithmetical and logical operations under program control.

Microprocessors date from about 1971, and since that time the term microprocessor has generally referred to the computer-on-a-chip concept as defined above. Prior to 1971, and in isolated cases afterward, the term had a different meaning. Before 1971, a microprocessor was a processor that executed microinstructions; thus it was a processor that was microprogrammed. This use of the term appeared particularly in the academic literature. Since about 1971, the term has referred to a computer-on-a-chip device. All of the world's leading general-purpose computer manufacturers have products which contain microprocessors. These microprocessors include advanced large-scale-integrated devices with speeds and computing power equivalent to or better than that of most minicomputers of past years.

SOFTWARE CHARACTERISTICS

Another key to the emergence of the CAD/CAM industry was the creation of software which made it possible to perform the required analysis and simulations on this new generation of affordable computers. A *program* consists of the series of instructions which, when operated on by the computer, result in the performance of the required function. These instructions, when present in the processor, consist of patterns of digits which represent combinations of numbers or symbols. At this level the instructions are known as *machine code*. Programs can be written directly in this code, but the process is very tedious and difficult. Thus most programming is done using a language which can be converted into machine code by means of another program, such as a compiler.

The *low-level languages* are the simplest form of language. They consist of instructions which have an exact equivalent in machine code. This simplification of programming is achieved by making the instructions mnemonic, such as using the letters ADD for an addition instruction. The areas of storage are given labels, which are referred to within the program, both for addressing the data-storage areas and for the program instructions. The actual numerical addresses of the storage areas are inserted when the program is converted into machine code.

Low-level languages may also incorporate certain software facilities, such as the ability to call into the program standard subroutines and housekeeping packages for input and output operations. Since low-level languages have a one-to-one relationship with the processor's machine code, they can usually be converted only into the machine code for a particular computer.

High-level languages are more powerful than low-level languages, since they enable the programmer to use single instructions which involve large numbers of machine-code instructions. This has the effect of simplifying the programming, since fewer instructions need to be coded in order to produce the required result. Various kinds of programs are used on computer systems. The programs written to do some kind of useful task, such as plotting data or making calculations, are called *application programs* or *source programs.* These are generally written in a high-level language such as FORTRAN or BASIC. The FORTRAN or BASIC programs must be translated into machine language to actually control the computer circuits. This translation takes place using either an interpreter or a compiler.

The *compiler* or *interpreter* operates on the high-level language and produces the version of the program called the *object program.* The object program is constructed of machine code so that it will run directly on the computer.

FORTRAN

As a computer language, FORTRAN has had a great influence as well as staying power. FORTRAN's basic form and style have changed little in thirty years. The language has acquired new features and capabilities. It has been standardized twice, in 1966 and 1977, and we can expect to see other standardizations in the future.

Besides winning its own popularity, FORTRAN inspired BASIC, which is essentially a streamlined and simplified version of the FORTRAN language. Although it can be used as a general-purpose programming language, FORTRAN's main use is in scientific and engineering applications, and this is the area where it is used today.

Its special capabilities for scientific calculations along with the thousands of subroutines that have been developed in FORTRAN are primarily responsible for its continued use. The existence of many FORTRAN programs that can be used and adapted for scientific or technical purposes means that the language is still very useful, no matter how elegant and efficient the newer languages become.

FORTRAN has provisions for the standard programming constructs such as looping, branching, subroutines, functions, and the assignment of values to variable names. It supports a number of data types, including:

1. Integer.

2. Single- and double-precision floating-point (real) numbers.
3. Complex numbers.
4. Logical (Boolean) expressions.
5. Strings.

The only structure available for organizing data is the *array*. The variables in FORTRAN may have names of up to six characters, beginning with a letter. Variable names beginning with the letters I through N usually represent integers.

The FORTRAN program or subroutine consists of the following:

1. A header statement in which the program is named.
2. A series of specification statements in which data types may be specified and array dimensions declared.
3. A list of executable statements that makes up the body of the program.

FORTRAN programs use a rigid format that dates back to the earlier years when programs were entered statement by statement on 80-column punched cards. Spaces were reserved for labels in the first five columns, and a character in the sixth column indicated that a statement was continued from the line above. The actual program statements were in columns 7 through 72. Blanks are ignored, so B OD is the same as the variable BOD. The letter C or an asterisk in column 1 indicates a comment to be ignored by the compiler.

The line numbers in FORTRAN programs are used only where required to control loops, conditionals, or GOTO statements. Line numbers are used as shown below:

```
DO   50 I = 1,20
       .
       .
       .
       .
       .
50   CONTINUE
     GOTO 200
200  IF (A) 10,20,30
```

The arithmetic IF statement in FORTRAN branches to:

1. The first line number if the value of A is negative.
2. The second line number if it is zero.
3. The third line number if it is positive.