A PEARSON AUSTRALIA CUSTOM BOOK

7/22

ENGR10003
Engineering Systems Design 2

Introduction to Digital Systems, Programming and Mechanics

This custom book is compiled from:

MASTERING MATLAB
HANSELMAN & LITTLEFIELD

ENGINEERING MECHANICS STATICS
12TH EDITION IN SI UNITS
HIBBELER

ENGINEERING MECHANICS DYNAMICS
12TH EDITION IN SI UNITS
HIBBELER

DIGITAL SYSTEMS
PRINCIPLES AND APPLICATIONS
11TH EDITION
TOCCI, WIDMER & MOSS

THE UNIVERSITY OF MELBOURNE

Pearson Australia
Unit 4, Level 3
14 Aquatic Drive
Frenchs Forest, Sydney NSW 2086
Ph: 02 9454 2200
www.pearson.com.au

Copyright © 2012 This Custom Book Edition, Pearson Australia (a division of Pearson Australia Group Pty Ltd).
Copyright © 2010 by Prentice Hall Pearson Education South Asia Pte Ltd for *Engineering Mechanics Statics* 12th edn in SI Units by Hibbeler.
Copyright © 2010 by Prentice Hall Pearson Education South Asia Pte Ltd for *Engineering Mechanics Dynamics* 12th edn in SI Units by Hibbeler.
Copyright © 2011 by Pearson Education Inc., Upper Saddle River, New Jersey for *Digital Systems: Principles and Applications* 11th edn. by Tocci, Widmer & Moss.
Copyright © 2012 by Pearson Education Inc., Upper Saddle River, New Jersey for *Mastering MATLAB* by Hanselman & Littlefield.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the written permission of the publisher.

Managing Editor: Jill Gillies
Consultant: Claire Sadler
Project Manager: Aida Reyes Cruz
Project Editor: Barbara Honor

ISBN: 978 1 4860 0166 8

Printed in Malaysia (CTP-PPSB)

WAYS LEARNING PEARSON

TABLE OF CONTENTS

I. Digital Systems **1**

Introductory Concepts 2
Tocci *et al*, **Chapter 1**

Number Systems and Codes 32
Tocci *et al*, **Chapter 2**

Describing Logic Circuits 64
Tocci *et al*, **Chapter 3**

Combinational Logic Circuits 128
Tocci *et al*, **Chapter 4**

II. Programming with MATLAB **219**

Getting Started 221
Hanselman & Littlefield, **Chapter 1**

Basic Features 224
Hanselman & Littlefield, **Chapter 2**

The MATLAB Desktop 242
Hanselman & Littlefield, **Chapter 3**

Script M-files 251
Hanselman & Littlefield, **Chapter 4**

Multidimensional Arrays 261
Hanselman & Littlefield, **Chapter 6**

Cell Arrays and Structures 279
Hanselman & Littlefield, **Chapter 8**

Character Strings 315
Hanselman & Littlefield, **Chapter 9**

III. Mechanics: Statics **345**

General Principles 346
Hibbeler (Statics), **Chapter 1**

Force Vectors 360
Hibbeler(Statics), **Chapter 2**

Equilibrium of a Particle 428
Hibbeler (Statics), **Chapter 3**

Force Systems Resultants 460
Hibbeler (Statics), **Chapter 4**

Equilibrium of a Rigid Body 542
Hibbeler (Statics), **Chapter 5**

Structural Analysis 606
Hibbeler (Statics), **Chapter 6**

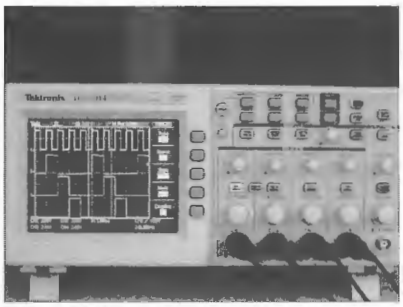
IV. Mechanics: Dynamics **673**
Kinematics of a Particle 674
Hibbeler, **Chapter 12**

Kinetics of a Particle: Force and Acceleration 726
Hibbeler, **Chapter 13**

HOW TO USE THIS BOOK

Navigation Bar – Use this information to navigate through the book. The page numbers here run continuously from beginning to end.

FIGURE 1-3 An oscilloscope displaying a 4-trace timing diagram.



1-2 NUMERICAL REPRESENTATIONS

In science, technology, business, and, in fact, most other fields of endeavor, we are constantly dealing with *quantities*. Quantities are measured, monitored, recorded, manipulated arithmetically, observed, or in some other way utilized in most physical systems. It is important when dealing with various quantities that we be able to represent their values efficiently and accurately. There are basically two ways of representing the numerical value of quantities: *analog* and *digital*.

Analog Representations

In **analog representation** a quantity is represented by a continuously variable, proportional indicator. An example is an automobile speedometer from the classic muscle cars of the 1960s and 1970s. The deflection of the needle is proportional to the speed of the car and follows any changes that occur as the vehicle speeds up or slows down. On older cars, a flexible mechanical shaft connected the transmission to the speedometer on the dash board. It is interesting to note that on newer cars, the analog representation is usually preferred even though speed is now measured digitally.

Thermometers before the digital revolution used analog representation to measure temperature, and many are still in use today. Mercury thermometers use a column of mercury whose height is proportional to temperature. These devices are being phased out of the market because of environmental concerns, but nonetheless they are an excellent example of analog representation. Another example is an outdoor thermometer on which the position of the pointer rotates around a dial as a metal coil expands and contracts with temperature changes. The position of the pointer is proportional to the temperature. Regardless of how small the change in temperature, there will be a proportional change in the indication.

In these two examples the physical quantities (speed and temperature) are being coupled to an indicator by purely mechanical means. In electrical analog systems, the physical quantity that is being measured or processed is converted to a proportional voltage or current (electrical signal). This

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

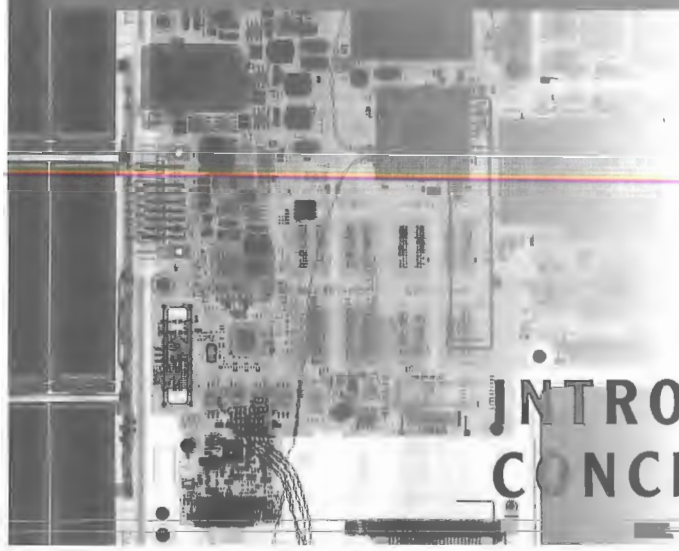
Reference Bar – This information tells you where the content was sourced. When referencing in your assignments you need to quote these original details including page numbers.

I. Digital Systems

The content in this section is compiled from:

Digital Systems: Principles and Applications
11th edition
Tocci, Widmer & Moss

CHAPTER 1



INTRODUCTORY CONCEPTS

OUTLINE

1-1 Introduction to Digital 1s and 0s
1-2 Numerical Representations
1-3 Digital and Analog Systems
1-4 Digital Number Systems
1-5 Representing Binary Quantities
1-6 Digital Circuits/Logic Circuits
1-7 Parallel and Serial Transmission
1-8 Memory
1-9 Digital Computers

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

OBJECTIVES

Upon completion of this chapter, you will be able to:

- Distinguish between analog and digital representations.
- Describe how information can be represented using just two states (1s and 0s).
- Cite the advantages and drawbacks of digital techniques compared with analog.
- Describe the purpose of analog-to-digital converters (ADCs) and digital-to-analog converters (DACs).
- Recognize the basic characteristics of the binary number system.
- Convert a binary number to its decimal equivalent.
- Count in the binary number system.
- Identify typical digital signals.
- Identify a timing diagram.
- State the differences between parallel and serial transmission.
- Describe the property of memory.
- Describe the major parts of a digital computer and understand their functions.
- Distinguish among microcomputers, microprocessors, and microcontrollers.

INTRODUCTION

In today's world, the term *digital* has become part of our everyday vocabulary because of the dramatic way that digital circuits and digital techniques have become so widely used in almost all areas of life: computers, automation, robots, medical science and technology, transportation, telecommunications, entertainment, space exploration, and on and on. You are about to begin an exciting educational journey in which you will discover the fundamental principles, concepts, and operations that are common to all digital systems, from the simplest on/off switch to the most complex computer.

We introduce the underlying concepts through a device that has become familiar to nearly everyone: the cellular telephone. This amazing device is made up of many of the fundamental building blocks common to all *digital systems*. In order to break down the complexity of a cell phone, we will first discuss the evolution of telecommunications technology. Your familiarity with the past and present communications systems will help you to understand the role that each building block plays, learn the terms that define each block, the nature of the signals that connect the blocks, and prepare you to expand your horizons by seeing how these blocks can be used in other

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

systems. Terms that are important to the study of digital systems are listed at the end of each chapter with full definitions contained in the glossary.

1-1 INTRODUCTION TO DIGITAL 1s AND 0s

It seems appropriate to begin a text named *Digital Systems* by considering one of the largest electronic systems in the world: the world-wide telecommunications system. Today, a very large part of this system falls in the category of "digital systems." Amazingly enough, it all started out as a simple digital system that used only two states to represent information. This concept is fundamental and easier to understand by first studying early technology.

The telegraph system, depicted in Figure 1-1, was a simple electromechanical system that revolutionized communication. It consisted of a battery, a code key (normally open, momentary contact switch), a very long telegraph wire, and on the other end an electromagnetic "clacker." When the telegrapher pressed down on the key, it completed the circuit by connecting the positive battery terminal to the wire. The negative battery terminal was connected to a rod that was driven down in the ground. Current from the battery flowed through the telegraph wire to the electromagnetic coil at the receiving station and then back to a rod driven into the ground. Electric current flowing through the coil caused a magnetic field to attract a metal plate that made a "clack" noise. The plate stayed in this position until the code key was released (interrupting the circuit) and a spring returned the plate to its original position, making a different "click" noise. Notice the two states of the system: code key and clacker down; code key and clacker up.

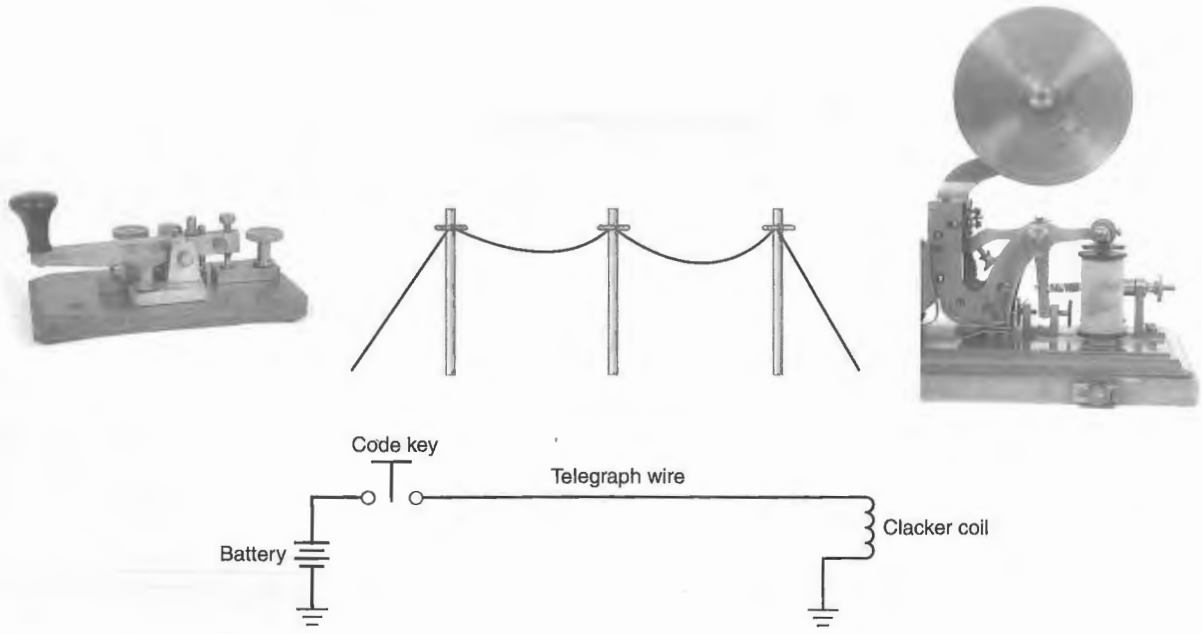


FIGURE 1-1 A telegraph system.

The telegraph system used two distinct "symbols" to transmit any word or number: short and long electric pulses, which represent the dots and dashes of Morse code. As we will see, this describes a *digital representation* of

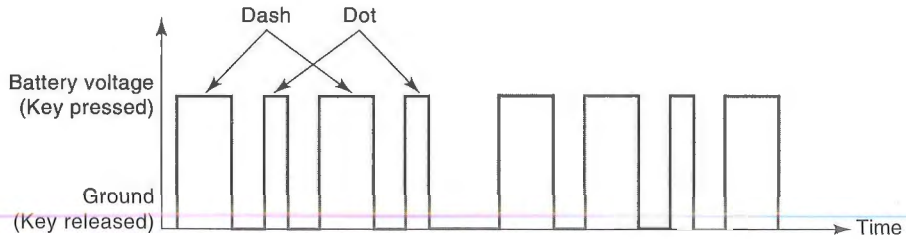


FIGURE 1-2 Timing diagram of a telegraph line.

information. What did the information look like? Figure 1-2 shows a graph of the voltage pulses on the telegraph wire as the telegrapher presses and releases the code key (current flow through the wire could have been graphed as well and would be of the same waveshape as voltage). Notice the nature of the pulses. The electric signal is either on or off at all times. This relates to modern digital systems that use electrical signals to represent 1s and 0s. The information is **encoded** in the width (length of time) of each pulse and the sequence of pulses. Pulse waveforms like this are used extensively to describe the activity in digital systems. Since the x axis is time, these graphs are referred to as **timing diagrams**. A timing diagram shows which state (1 or 0) the system is in at any point in time, and it also shows the exact time when a change in state occurs.

Timing diagrams are used to describe the detailed operation of a digital system. The transitions between the low voltage level (when the code key is released) and high voltage level (when the code key is pressed) on this timing diagram are drawn as crisp, vertical lines. Certainly, in a primitive telegraph system, the waveforms were not this perfect due to things like contact arcing, long wires, and natural effects of electromagnetic coils. In fact, even in modern digital systems, transitions are not truly instantaneous (vertical lines). In many situations, however, the transition times are so short compared to the times between transitions that we can reasonably show them on the diagram as vertical lines. We will encounter situations later where it will be necessary to show the transitions more accurately on an expanded time scale.

Timing diagrams are used extensively to show how digital signals change with time and especially to show the relationship between two or more digital signals in the same circuit or system. By displaying one or more digital signals using test instruments such as an oscilloscope, shown in Figure 1-3, we can compare the actual signals to the system's expected operation. A logic analyzer is another instrument that is able to display and analyze many concurrent digital signals as a timing diagram. Mastery of these instruments is an important part of testing and troubleshooting procedures used in digital systems.

REVIEW QUESTIONS*

1. How many fundamental states are there in a digital system? 2
2. What do we call a graph that shows changes between two states (1s and 0s) with respect to time?

*Answers to review questions are found at the end of the chapter in which they occur.

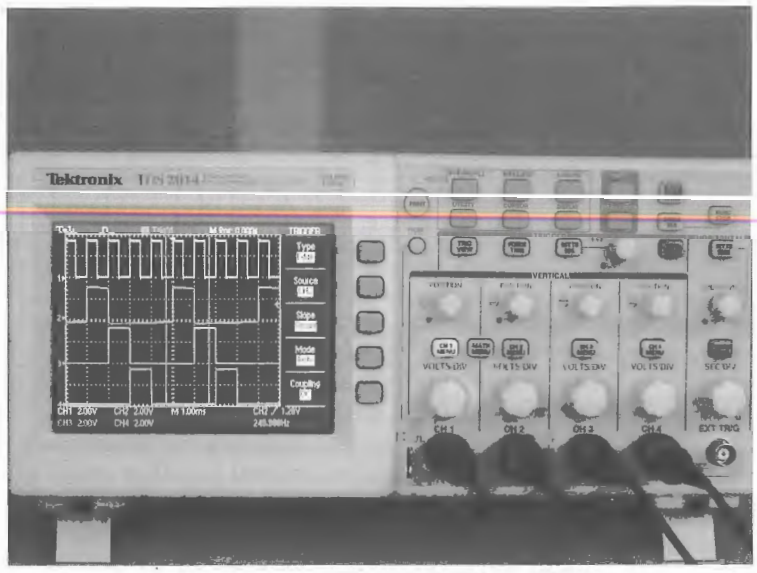


FIGURE 1-3 An oscilloscope displaying a 4-trace timing diagram.

1-2 NUMERICAL REPRESENTATIONS

In science, technology, business, and, in fact, most other fields of endeavor, we are constantly dealing with *quantities.* Quantities are measured, monitored, recorded, manipulated arithmetically, observed, or in some other way utilized in most physical systems. It is important when dealing with various quantities that we be able to represent their values efficiently and accurately. There are basically two ways of representing the numerical value of quantities: *analog* and *digital.*

Analog Representations

In **analog representation** a quantity is represented by a continuously variable, proportional indicator. An example is an automobile speedometer from the classic muscle cars of the 1960s and 1970s. The deflection of the needle is proportional to the speed of the car and follows any changes that occur as the vehicle speeds up or slows down. On older cars, a flexible mechanical shaft connected the transmission to the speedometer on the dash board. It is interesting to note that on newer cars, the analog representation is usually preferred even though speed is now measured digitally.

Thermometers before the digital revolution used analog representation to measure temperature, and many are still in use today. Mercury thermometers use a column of mercury whose height is proportional to temperature. These devices are being phased out of the market because of environmental concerns, but nonetheless they are an excellent example of analog representation. Another example is an outdoor thermometer on which the position of the pointer rotates around a dial as a metal coil expands and contracts with temperature changes. The position of the pointer is proportional to the temperature. Regardless of how small the change in temperature, there will be a proportional change in the indication.

In these two examples the physical quantities (speed and temperature) are being coupled to an indicator by purely mechanical means. In electrical analog systems, the physical quantity that is being measured or processed is converted to a proportional voltage or current (electrical signal). This

voltage or current is then used by the system for display, processing, or control purposes.

Sound is an example of a physical quantity that can be represented by an electrical analog signal. In 1875, Alexander Graham Bell figured out how to change his voice into a continuously variable electrical signal, send it through a wire, and change it back to sound energy at the other end. Today, the device that converts sound energy to an analog voltage signal is known as a microphone. Figure 1-4 shows what the analog voice signal looks like. The information is contained in the tones and inflections of this everchanging voice signal and is encoded by the spoken language of the speaker. Tones are measured by the **frequency** (f) which tells us how many cycles of the wave happen in a certain amount of time (cycles per second). The horizontal axis (time) gives us an indication of time for each cycle, known as the **period** (T) of the wave (seconds per cycle). The loudness is measured by the **amplitude** of the wave in units specified on the vertical axis (e.g., volts). Changes in amplitude represent

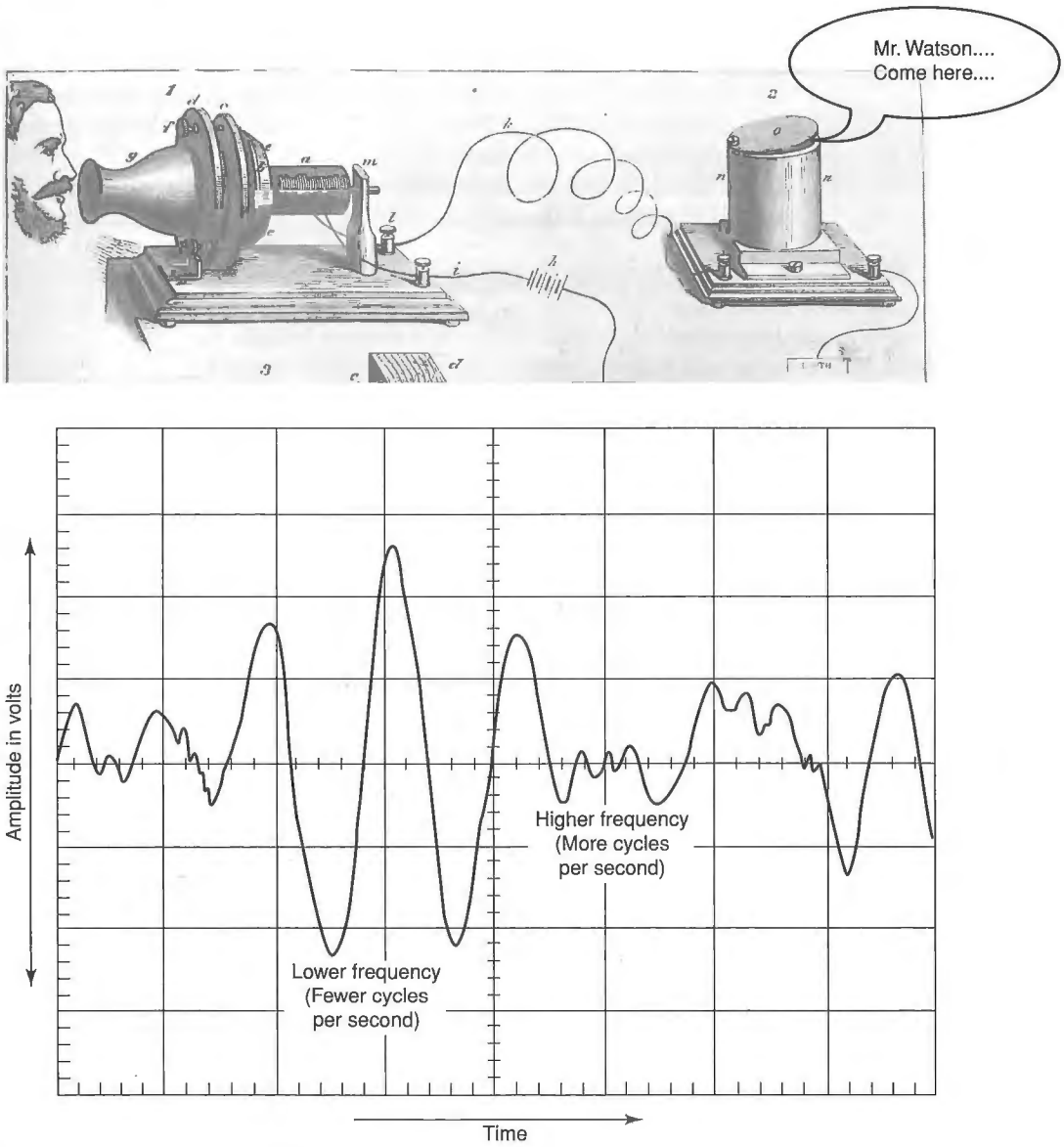


FIGURE 1-4 An audio wave.

inflection or emphasis of certain sounds. In other words, higher amplitudes result in louder sounds and lower amplitudes result in quieter sounds.

No matter how they are represented, analog quantities have an important characteristic: *they can vary over a continuous range of values.* The automobile speed can have *any* value between zero and, say, 100 mph. Similarly, the temperature indicated by an analog thermometer can have any value from −20° to 120° Fahrenheit, and the analog voltage signal produced by a microphone can have any value between zero and its maximum positive or negative output.

Digital Representations

In **digital representation** the quantities are represented not by continuously variable indicators but by symbols called *digits.* As an example, consider a digital indoor/outdoor thermometer. It has four digits and can measure changes of 0.1°. The actual temperature gradually increases from, say, 72.0 to 72.1 but the digital representation changes suddenly from 72.0 to 72.1. In other words, this digital representation of outdoor temperature changes in *discrete* steps, as compared with the analog representation of temperature provided by a fluid column or metal coil thermometer, where the reading changes continuously.

The major difference between analog and digital quantities, then, can be simply stated as follows:

analog ≡ continuous
digital ≡ discrete (step by step)

Because of the discrete nature of digital representations, there is no ambiguity when reading the value of a digital quantity, whereas the value of an analog quantity is often open to interpretation. In practice, when we take a measurement of an analog quantity, we always "round" to a convenient level of precision. In other words, we digitize the quantity. The digital representation is the result of assigning a number of limited precision to a continuously variable quantity.

The world around us is full of physical variables that are constantly changing. If we can measure these variables and represent them as a digital quantity, then we can record, arithmetically manipulate, or in some other way use these quantities to control things.

EXAMPLE 1-1

Which of the following involve analog quantities and which involve digital quantities?

(a) Elevation using a ladder
(b) Elevation using a ramp
(c) Current flowing from an electrical outlet through a motor
(d) Height of a child measured by a yard stick ruler
(e) Height of a child measured by putting a mark on the wall
(f) Volume of sand in a bucket
(g) Volume of water in a bucket

Solution

(a) Digital
(b) Analog
(c) Analog
(d) Digital: measured to nearest 1/8 inch
(e) Analog
(f) Digital: can only increase/decrease by discrete grains of sand
(g) Analog: (unless you want to get to the nano technology level!)

REVIEW QUESTIONS

1. What were the two binary states of a telegraph system?
2. How was telegraph information encoded using these two states?
3. What property of an audio waveform affects loudness?
4. What property of an audio waveform affects the pitch of a tone?
5. Which method of representing quantities involves discrete steps?
6. Which method of representing quantities is continuously variable?

1-3 DIGITAL AND ANALOG SYSTEMS

A **digital system** is a combination of devices designed to manipulate logical information or physical quantities that are represented in digital form; that is, the quantities can take on only discrete values. These devices are most often electronic, but they can also be mechanical, magnetic, or pneumatic. Some of the more familiar digital systems include digital computers and calculators, digital audio and video equipment, and the telecommunications system.

An **analog system** contains devices that manipulate physical quantities that are represented in analog form. In an analog system, the quantities can vary over a continuous range of values. For example, the amplitude of the output signal to the speaker in a radio receiver can have any value between zero and its maximum limit.

Mixed Systems: Digital and Analog

It is common to see both digital and analog techniques employed within the same system. This way the system is able to profit from the advantages of each. In these hybrid or mixed signal systems, one of the important parts of the design phase involves determining what parts of the system are to be analog and what parts are to be digital. The trend today in most systems is to digitize the signals as early as possible and convert them back to analog as late as possible in the signal flow path of the system.

To look at systems that use both digital and analog techniques, let's revisit the evolution of the telephone. As we do so, we will discover that there are many ways to represent information in systems. Bell realized that to be useful, the telephones needed to be networked as shown in Figure 1-5. His solution was to put a hand-cranked electrical generator on each telephone. When a person turned the crank, it produced a voltage that would cause a bell to ring on every telephone connected to the network. Stopping the crank

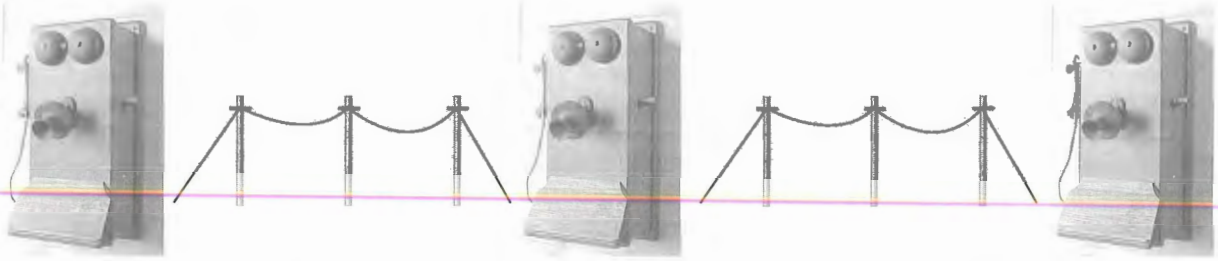


FIGURE 1-5 A "party-line" phone network.

caused the bells to stop ringing. Each person on the network was assigned a unique code of long and short rings (like digital pulses). The calling party *encoded* a person's identification by the way he cranked his telephone. The receiving party mentally **decoded** the ring patterns to know when to complete the connection by picking up his telephone receiver. The signaling (rings) used digital representation, but voice communication over this connection was purely analog.

The rotary-dial telephone shown in Figure 1-6 came next and used a more sophisticated series of pulses, which represented the ten digits of the decimal number system. To dial a number, a person would stick his finger in a numbered hole, rotate the dial to the stop arm, and let it go. A spring returned the dial to its original position while turning a cam shaft that opened and closed switch contacts, producing pulses. A sequence of pulses represented each number on the dial. For example, nine pulses represented the number 9, two pulses represented the number 2, and ten pulses represented the number 0. Electromechanical switching mechanisms interpreted (decoded) the pulses and made the connection to wires that went to the correct telephone on the network and caused it to ring until someone answered. Notice that in these systems a different type of digital code was used. The encoding and decoding were done by electromechanical machines to automatically make a connection for analog communication.

FIGURE 1-6 A rotary-dial telephone.

Touch-tone telephones came next. A different complex audible signal made up by combining two different sinusoidal frequencies represented each digit of a phone number. These familiar touch-tone sounds are referred to as Dual Tone, Multiple Frequency (DTMF). Electronic circuits were able to recognize each "touch tone," translate the tones into a sequence of digits, and make the right connections to cause a single phone to ring. Notice that in this example, digital switching information is sent using analog tone signals but each tone is made up of two distinct frequencies. Analog and digital representations of information have always worked together in electronic communication.

THE CELL PHONE Consider the cell phone in Figure 1-7 or pick up your own phone and marvel at what it does. Perhaps right now it seems impossible that you could understand how such a complex system operates. Inside a cell phone are some very sophisticated electronics, most of which are covered by the metal shielding on the circuit board. All complex systems (e.g., laptops, HDTV, and automotive controls) are made up of the same basic building blocks. By learning the basic building blocks and using the technology available today, you can design and miniaturize digital systems like the one shown in this cell phone.

FIGURE 1-7 A cell phone disassembled.

The cell phone is a mixed signal system, meaning that it has digital and analog components and uses both types of signals. Chapter 11 will explain how this is done. Your voice is picked up by an analog microphone, but is soon converted to a digital signal. The digital audio signal and lots of other digital information such as phone numbers, global position coordinates, text messages, and so on, are combined with a very high frequency radio wave and sent to a cell tower. Your phone also receives an analog radio signal, separates the digital information, converts the digital audio signal back to analog, and applies it to your speaker.

The keypad is the most obvious input to a cell phone. Each key is in one of two positions or states: pressed or not pressed. A system with only two states is a binary system. Of course, each key must be represented in a unique way to the digital system so the values (numbers on the keys) can be easily distinguished from one another. The **binary number system** uses groups of **binary digits** or **bits** (1s and 0s) to represent decimal numbers in a digital system. We will begin our study of binary numbers later in this chapter. How does a cell phone translate a specific button press to a unique binary code? You will learn the *encoding process* for keypads in Chapter 9.

The phone in Figure 1-7 also has a switch that recognizes whether the phone is flipped open (1) or is closed (0). This open/closed sensor also is used

REVIEW QUESTIONS

1. Name three ways that information is encoded using a binary digital system.
2. What was digital and what was analog in early telephone systems?
3. What is digital and what is analog in modern cell phone systems?
4. What are the advantages of digital techniques over analog?
5. What is the chief limitation to the use of digital techniques?

1-4 DIGITAL NUMBER SYSTEMS

Many number systems are in use in digital technology. The most common are the decimal, binary, and hexadecimal systems. Humans operate using decimal numbers, digital systems operate using binary numbers, and **hexadecimal** is a number system that makes it easier for humans to deal with binary numbers. All three of these number systems are defined and function in the exact same way. Let's start by examining the decimal system. Because it is so familiar, we rarely stop to think about how this number system actually works. Examining its characteristics fully will help you to understand the other systems better.

Decimal System

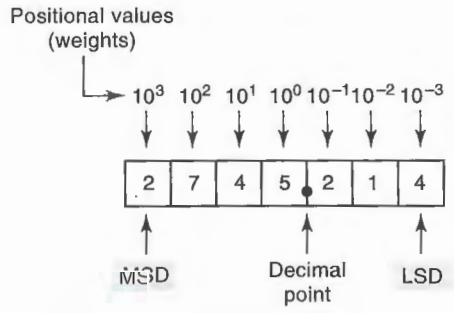
The **decimal system** is composed of *10* numerals or symbols. These 10 symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; using these symbols as *digits* of a number, we can express any quantity. The decimal system, also called the *base-10* system because it has 10 digits, has evolved naturally as a result of the fact that people have 10 fingers. In fact, the word *digit* is derived from the Latin word for "finger."

The decimal system is a *positional-value system* in which the value of a digit depends on its position. For example, consider the decimal number 453. We know that the digit 4 actually represents 4 *hundreds,* the 5 represents 5 *tens,* and the 3 represents 3 *units.* In essence, the 4 carries the most weight of the three digits; it is referred to as the *most significant digit (MSD).* The 3 carries the least weight and is called the *least significant digit (LSD).*

Consider another example, 27.35. This number is actually equal to 2 tens plus 7 units plus 3 tenths plus 5 hundredths, or $2 \times 10 + 7 \times 1 + 3 \times 0.1 + 5 \times 0.01$. The decimal point is used to separate the integer and fractional parts of the number.

More rigorously, the various positions relative to the decimal point carry weights that can be expressed as powers of 10. This is illustrated in Figure 1-9,

FIGURE 1-9 Decimal position values as powers of 10.



Custom Publishing exclusively for The University of Melbourne

where the number 2745.214 is represented. The decimal point separates the positive powers of 10 from the negative powers. The number 2745.214 is thus equal to

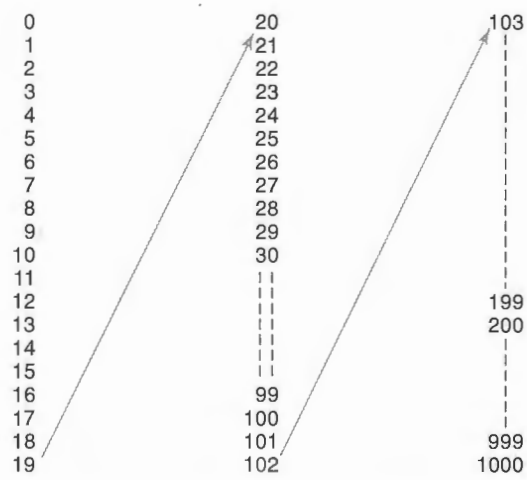
$$(2 \times 10^{+3}) + (7 \times 10^{+2}) + (4 \times 10^{1}) + (5 \times 10^{0}) + (2 \times 10^{-1}) + (1 \times 10^{-2}) + (4 \times 10^{-3})$$

In general, any number is simply the sum of the products of each digit value and its positional value.

Decimal Counting

When counting in the decimal system, we start with 0 in the units position and take each symbol (digit) in progression until we reach 9. Then we add a 1 to the next higher position and start over with 0 in the first position (see Figure 1-10). This process continues until the count of 99 is reached. Then we add a 1 to the third position and start over with 0s in the first two positions. The same pattern is followed continuously as high as we wish to count.

FIGURE 1-10 Decimal counting.



It is important to note that in decimal counting, the units position (LSD) changes upward with each step in the count, the tens position changes upward every 10 steps in the count, the hundreds position changes upward every 100 steps in the count, and so on.

Another characteristic of the decimal system is that using only two decimal places, we can count through $10^2 = 100$ different numbers (0 to 99).* With three places we can count through 1000 numbers (0 to 999), and so on. In general, with N places or digits, we can count through 10^N different numbers, starting with and including zero. The largest number will always be $10^N - 1$.

Binary System

Unfortunately, the decimal number system does not lend itself to convenient implementation in digital systems. For example, it is very difficult to design electronic equipment so that it can work with 10 different voltage levels (each one representing one decimal character, 0 through 9). On the other hand, it is very easy to design simple, accurate electronic circuits that operate with only two voltage levels. For this reason, almost every digital system

*Zero is counted as a number.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

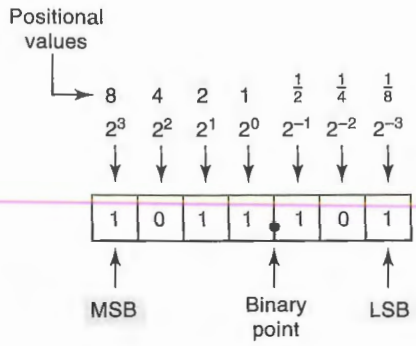


FIGURE 1-11 Binary position values as powers of 2.

uses the binary (base-2) number system as the basic number system of its operations. Other number systems are often used to interpret or represent binary quantities for the convenience of the people who work with and use these digital systems.

In the binary system there are only two symbols or possible digit values, 0 and 1. Even so, this base-2 system can be used to represent any quantity that can be represented in decimal or other number systems. In general though, it will take a greater number of binary digits to express a given quantity.

All of the statements made earlier concerning the decimal system are equally applicable to the binary system. The binary system is also a positional-value system, wherein each binary digit has its own value or weight expressed as a power of 2. This is illustrated in Figure 1-11. Here, places to the left of the *binary point* (counterpart of the decimal point) are positive powers of 2, and places to the right are negative powers of 2. The number 1011.101 is shown represented in the figure. To find its equivalent in the decimal system, we simply take the sum of the products of each digit value (0 or 1) and its positional value:

$$\begin{aligned} 1011.101_2 &= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &\quad + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) \\ &= 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125 \\ &= 11.625_{10} \end{aligned}$$

Notice in the preceding operation that subscripts (2 and 10) were used to indicate the base in which the particular number is expressed. This convention is used to avoid confusion whenever more than one number system is being employed.

In the binary system, the term *binary digit* is often abbreviated to the term *bit*, which we will use from now on. Thus, in the number expressed in Figure 1-11 there are four bits to the left of the binary point, representing the integer part of the number, and three bits to the right of the binary point, representing the fractional part. The most significant bit (MSB) is the leftmost bit (largest weight). The least significant bit (LSB) is the rightmost bit (smallest weight). These are indicated in Figure 1-11. Here, the MSB has a weight of 2^3; the LSB has a weight of 2^{-3}. The weights of each digit increase by a factor of 2 as the position moves from right to left.

Binary Counting

When we deal with binary numbers, we will usually be restricted to a specific number of bits. This restriction is based on the circuitry used to represent these binary numbers. Let's use four-bit binary numbers to illustrate the method for counting in binary.

The sequence (shown in Figure 1-12) begins with all bits at 0; this is called the *zero count*. For each successive count, the units (2^0) position *toggles*; that is, it changes from one binary value to the other. Each time the units bit changes from a 1 to a 0, the twos (2^1) position will toggle (change states). Each time the twos position changes from 1 to 0, the fours (2^2) position will toggle (change states). Likewise, each time the fours position goes from 1 to 0, the eights (2^3) position toggles. This same process would be continued for the higher-order bit positions if the binary number had more than four bits.

The binary counting sequence has an important characteristic, as shown in Figure 1-12. The units bit (LSB) changes either from 0 to 1 or 1 to 0 with *each* count. The second bit (twos position) stays at 0 for two counts, then at 1 for two counts, then at 0 for two counts, and so on. The third bit (fours position) stays at 0 for four counts, then at 1 for four counts, and so on. The fourth bit (eights position) stays at 0 for eight counts, then at 1 for eight counts. If we wanted to count further, we would add more places, and this pattern would continue with 0s and 1s alternating in groups of 2^{N-1}. For example, using a fifth binary place, the fifth bit would alternate sixteen 0s, then sixteen 1s, and so on.

As we saw for the decimal system, it is also true for the binary system that by using N bits or places, we can go through 2^N counts. For example, with two bits we can go through $2^2 = 4$ counts (00_2 through 11_2); with four bits we can go through $2^4 = 16$ counts (0000_2 through 1111_2); and so on. The last count will always be all 1s and is equal to $2^N - 1$ in the decimal system. For example, using four bits, the last count is $1111_2 = 2^4 - 1 = 15_{10}$.

FIGURE 1-12 Binary counting sequence.

Weights → $2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	Decimal equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15
			LSB	

EXAMPLE 1-2

What is the largest number that can be represented using eight bits?

Solution

$2^N - 1 = 2^8 - 1 = 255_{10} = 11111111_2$.

This has been a brief introduction of the binary number system and its relation to the decimal system. We will spend much more time on these two systems and several others in the next chapter.

REVIEW QUESTIONS

1. What is the decimal equivalent of 1101011_2?
2. What is the next binary number following 10111_2 in the counting sequence?
3. What is the largest decimal value that can be represented using 12 bits?

1-5 REPRESENTING BINARY QUANTITIES

Recall that a cell phone digitized the analog voice signal and actually transmitted the information as a bunch of 1s and 0s. While the details of how this is done will be saved for Chapter 11, we will now look at the concept of representing an analog signal as a sequence of digital numbers. From the previous section we know that any quantity can be represented by a binary number, just as easily as it can be represented by a decimal number. Suppose you are doing a science experiment that requires you to keep a record of temperature changes over a long period of time. You know that the temperature of the air is an analog quantity: continuously variable. However, you also know that temperature usually changes rather slowly, so instead of trying to continuously plot the current temperature, you decide to take a measurement at the top of every hour. The experiment is running from early spring to late fall, so you determine that the temperature will never be below 0 and it will never be above 127°. On a given day, you record the data as shown in Figure 1-13.

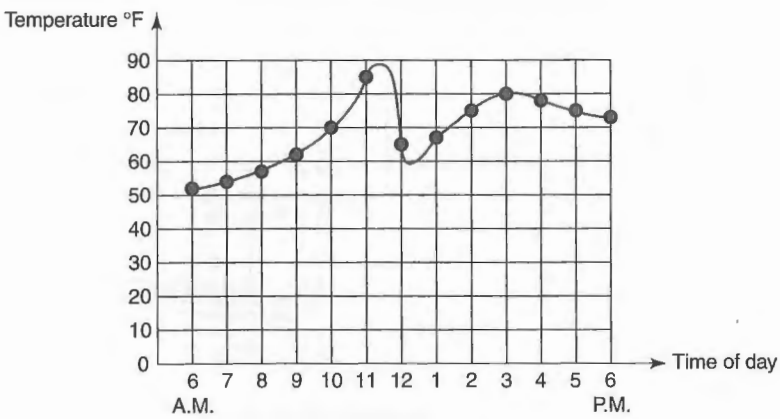


FIGURE 1-13 Temperature data sampled hourly. Red line indicates analog.

The actual analog temperature signal is shown in red. The data you have recorded is digital because you assigned a discrete integer number to the quantity each hour. The list of numbers is now your digital signal. The data can be stored, transmitted, and eventually plotted to show a very good approximation of the actual analog signal. To store the data in a digital system, each temperature reading would be converted to its binary equivalent value.

Now, imagine what the graph would look like over the entire experiment. The timeline (x-axis) would span several months, and the hourly "dots" would be very close together. During early spring there may be wide swings in temperature within a single day. On the graph this looks like high amplitude, high frequency changes. During the summer, the temperature may not vary more than 10 degrees for more than a week. This looks like low amplitude, low frequency changes. The point is that the graph of temperature may

ENGR10005
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

look strikingly similar to the analog waveform of a person's voice on a telephone wire, like we considered earlier in the chapter. The major difference is the time scale because temperature changes occur at a much lower frequency than the sound waves that your vocal cords produce. The key concepts are:

- Analog signals can be converted to digital by taking measurements or "samples" of the continuously varying signal at regular intervals.
- The appropriate time between samples depends on the maximum rate of change of the analog signal.
- The quantities that are measured are stored as binary numbers.

Electrical Representation of 1s and 0s

Earlier we identified one advantage of digital systems over analog systems as being the fact that the exact value of voltage or current is not important; the importance is the range in which these values fall. In other words, digital systems must simply represent these two states in a way that circuits can recognize a 1 from a 0. Think back to the telegraph example. The 1s were represented by a voltage on the telegraph wire. How much voltage was necessary to represent a 1? The answer is contained in the technology of the day. The battery voltage had to be sufficient to assure that enough current could flow through miles of wire and through the clacker's electromagnetic coil to make the clacker "clack." The voltage on the wire decreased as the batteries discharged and varied depending on which end of the wire it was measured, but it represented a 1 as long as it activated the clacker.

Electronic digital circuits have gone through many changes as technology has advanced, but these principles of representing 1s and 0s remain the same. Figure 1-14(a) shows the typical representation of the two states of a digital signal. Notice that a *higher* range of voltages represent a valid 1 and a *lower* range of voltages represent a valid 0. The terms HIGH and LOW will often be used to describe the two states of a digital system instead of "1" and "0", respectively. In between these two valid ranges there is a range of voltages that are considered invalid. They are neither 1s nor 0s. The specific voltage ranges are defined for a given technology. For example, the voltage ranges for the telegraph were drastically different than those used in our cell phones. In the evolution of digital systems, various technologies such as electromechanical switches (relays), vacuum tubes, bipolar transistors, and MOSFET transistors have been used to implement digital logic circuits: each with their own characteristic definition of how to represent a 1 and a 0.

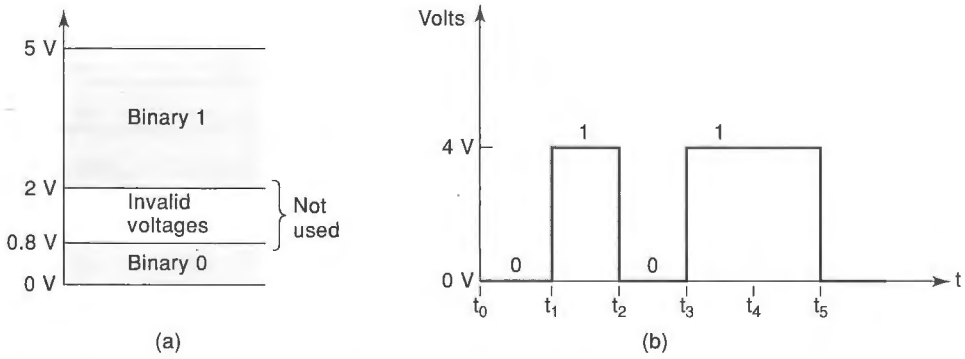


FIGURE 1-14 (a) Typical voltage assignments in digital system; (b) typical digital signal timing diagram.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

Recall that we discussed the concept of using a graph of voltage versus time to depict information travelling across a telegraph wire in Section 1-1. We called this a timing diagram. Figure 1-14(b) represents a typical timing diagram for the voltage ranges defined in part (a). The time axis is labeled at specific points in time, $t_1, t_2, \ldots t_5$. Notice that the HIGH voltage level between t_1 and t_2 is at 4 V. In digital systems, the exact value of a voltage is not important. A HIGH voltage of 3.7 V or 4.3 V would represent the exact same information. Likewise, a LOW voltage of 0.3 V represents the same information as 0 V. This points out a significant difference between analog and digital systems. In an analog system, the exact voltage is important. For example, if the analog voltage coming from a sensor is proportional to temperature, then 3.7 V would represent a different value of temperature than 4.3 V. In other words, the voltage carries significant information in the analog system. Circuits that can preserve exact voltages are much more complicated than digital circuits that simply need to recognize a voltage in one of two ranges.

REVIEW QUESTIONS

1. How is an analog signal represented in a digital system?
2. How are 1s and 0s represented electrically?

1-6 DIGITAL CIRCUITS/LOGIC CIRCUITS

Digital circuits are designed to produce output voltages that fall within the prescribed 0 and 1 voltage ranges such as those defined in Figure 1-14. Likewise, digital circuits are designed to respond predictably to input voltages that are within the defined 0 and 1 ranges. What this means is that a digital circuit will respond in the same way to all input voltages that fall within the allowed 0 range; similarly, it will not distinguish between input voltages that lie within the allowed 1 range.

To illustrate, Figure 1-15 represents a typical digital circuit with input v_i and output v_o. The output is shown for two different input signal waveforms. Note that v_o is the same for both cases because the two input waveforms, while differing in their exact voltage levels, are at the same binary levels.

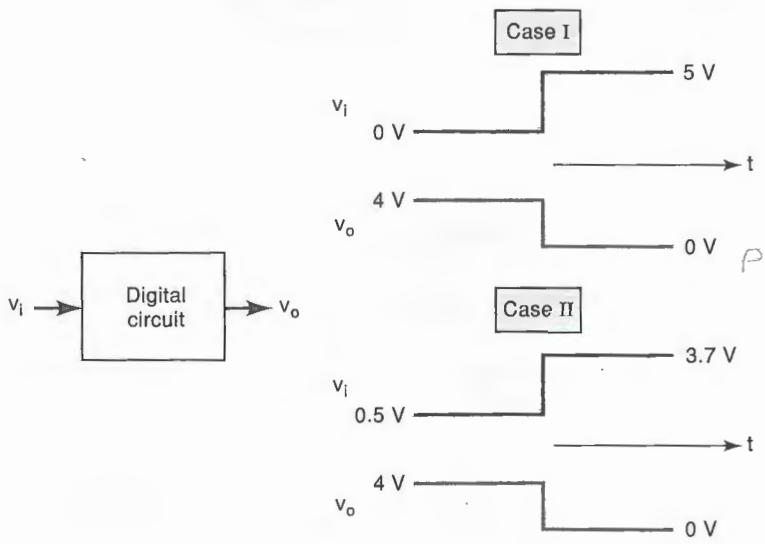


FIGURE 1-15 A digital circuit responds to an input's binary level (0 or 1) and not to its actual voltage.

Logic Circuits

The manner in which a digital circuit responds to an input is referred to as the circuit's *logic.* Each type of digital circuit obeys a certain set of logic rules. For this reason, digital circuits are also called **logic circuits.** We will use both terms interchangeably throughout the text. In Chapter 3, we will see more clearly what is meant by a circuit's "logic."

We will be studying all the types of logic circuits that are currently used in digital systems. Initially, our attention will be focused only on the logical operation that these circuits perform—that is, the relationship between the circuit inputs and outputs. We will defer any discussion of the internal circuit operation of these logic circuits until after we have developed an understanding of their logical operation.

Digital Integrated Circuits

Digital circuits of today's technology are primarily implemented using very sophisticated integrated circuits (ICs) that are electronically configured or tailor-made for their application. Many technologies of the past are completely obsolete. For example, the vacuum tube logic circuits would never be used today for a number of reasons such as: too big, too much power, and the vacuum tubes are very hard to find. Occasionally, it makes sense to use a mature technology where it is economical and the parts will be available over the life of a product. For example, most of the ICs that made up digital systems in the 1970s are no longer being manufactured but are still available on the market from large left-over inventories. These devices will, on rare occasion, be used in a new product and they are still used for laboratory instruction for digital courses in high school and college. Throughout this text we will try to provide enough information about the range of technologies to allow you to learn using simple devices from the past and yet introduce you to the fundamentals necessary to use the tools of the future.

Today the most common technology used to implement digital circuits (including the vast majority of computer hardware) is **CMOS**, which stands for **Complementary Metal-Oxide-Semiconductor**. Other technologies have been relegated to much smaller niches in the marketplace. Prior to the advancement of CMOS technology, bipolar transistor technology was king and had a profound influence on digital systems. The major logic family that sprung from bipolar technology is referred to as **TTL (Transistor/Transistor Logic)**. You will learn about the various IC technologies, their characteristics, and relative advantages and disadvantages in Chapter 8.

REVIEW QUESTIONS

1. *True or false:* The exact value of an input voltage is critical for a digital circuit.
2. Can a digital circuit produce the same output voltage for different input voltage values?
3. A digital circuit is also referred to as a _____ circuit.
4. A graph that shows how one or more digital signals change with time is called a _____.

1-7 PARALLEL AND SERIAL TRANSMISSION

One of the most common operations that occur in any digital system is the transmission of information from one place to another. The information can be transmitted over a distance as small as a fraction of an inch on the same circuit board, or over a distance of many miles when an operator at a computer terminal is communicating with a computer in another city. The information that is transmitted is in binary form and is generally represented as voltages at the outputs of a sending circuit that are connected to the inputs of a receiving circuit. Figure 1-16 illustrates the two basic methods for digital information transmission: **parallel** and **serial**.

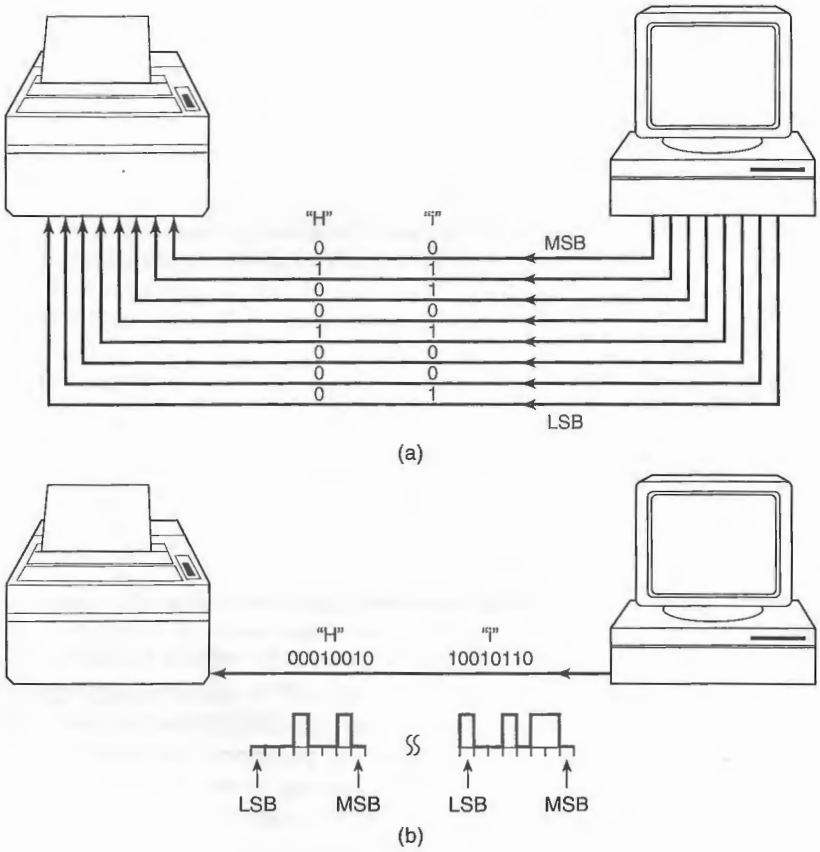


FIGURE 1-16 (a) Parallel transmission uses one connecting line per bit, and all bits are transmitted simultaneously; (b) serial transmission uses only one signal line, and the individual bits are transmitted serially (one at a time).

Figure 1-16(a) demonstrates parallel transmission of data from a computer to a printer. Parallel printer interfaces were standard in personal computers before the USB (Universal Serial Bus). In this scenario, assume we are trying to print the word "Hi" on the printer. The binary code for "H" is 01001000 and the binary code for "i" is 01101001. Each character (the "H" and the "i") are made up of eight bits. Using parallel transmission, all eight bits are sent simultaneously over eight wires. The "H" is sent first, followed by the "i."

Figure 1-16(b) demonstrates serial transmission such as is employed on your computer when using a USB port to send data to a printer. Although the details of the data format are much more complicated for a USB port than we show here, the data are sent one bit at a time over a single wire.

ENGR10005 – ...
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

The bits are shown in the diagram as though they were actually moving down the wire in the order shown. The least significant bit of "H" is sent first and the most significant bit of "i" is sent last. Of course, in reality, only one bit can be on the wire at any point in time and time is usually drawn on a graph starting at the left and advancing to the right. This produces a graph of logic bits versus time of the serial transmission called a timing diagram. Notice that in this presentation, the least significant bit is shown on the left because it was sent first.

The principal trade-off between parallel and serial representations is one of speed versus circuit simplicity. The transmission of binary data from one part of a digital system to another can be done more quickly using parallel representation because all the bits are transmitted simultaneously, while serial representation transmits one bit at a time. On the other hand, parallel requires more signal lines connected between the sender and the receiver of the binary data than does serial. In other words, parallel is faster, and serial requires fewer signal lines. This comparison between parallel and serial methods for representing binary information will be encountered many times in discussions throughout the text.

REVIEW QUESTION

1. Describe the relative advantages of parallel and serial transmission of binary data.

1-8 MEMORY

When an input signal is applied to most devices or circuits, the output somehow changes in response to the input, and when the input signal is removed, the output returns to its original state. These circuits do not exhibit the property of *memory* because their outputs revert back to normal. In digital circuitry certain types of devices and circuits do have memory. When an input is applied to such a circuit, the output will change its state, but it will remain in the new state even after the input is removed. This property of retaining its response to a momentary input is called **memory**. Figure 1-17 illustrates nonmemory and memory operations.

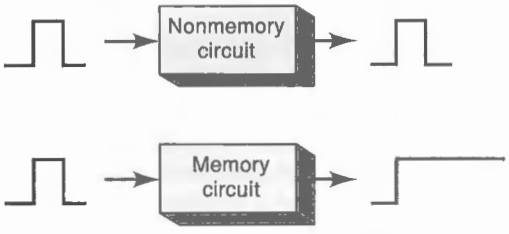


FIGURE 1-17 Comparison of nonmemory and memory operation.

Memory devices and circuits play an important role in digital systems because they provide a means for storing binary numbers either temporarily or permanently, with the ability to change the stored information at any time. As we shall see, the various memory elements include magnetic and optical types and those that utilize electronic latching circuits (called *latches* and *flip-flops*).

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

1-9 DIGITAL COMPUTERS

Digital techniques have found their way into innumerable areas of technology, but the area of automatic **digital computers** is by far the most notable and most extensive. Although digital computers affect some part of all of our lives, it is doubtful that many of us know exactly what a computer does. In simplest terms, *a computer is a system of hardware that performs arithmetic operations, manipulates data (usually in binary form), and makes decisions.*

For the most part, human beings can do whatever computers can do, but computers can do it with much greater speed and accuracy, in spite of the fact that computers perform all their calculations and operations one step at a time. For example, a human being can take a list of 10 numbers and find their sum all in one operation by listing the numbers one over the other and adding them column by column. A computer, on the other hand, can add numbers only two at a time, so that adding this same list of numbers will take nine actual addition steps. Of course, the fact that the computer requires only a few nanoseconds per step makes up for this apparent inefficiency.

A computer is faster and more accurate than people are, but unlike most of us, it must be given a complete set of instructions that tell it *exactly* what to do at each step of its operation. This set of instructions, called a **program**, is prepared by one or more persons for each job the computer is to do. Programs are placed in the computer's memory unit in binary-coded form, with each instruction having a unique code. The computer takes these instruction codes from memory *one at a time* and performs the operation called for by the code.

Major Parts of a Computer

There are several types of computer systems, but each can be broken down into the same functional units. Each unit performs specific functions, and all units function together to carry out the instructions given in the program. Figure 1-18 shows the five major functional parts of a digital computer and their interaction. The solid lines with arrows represent the flow of data and information. The dashed lines with arrows represent the flow of timing and control signals.

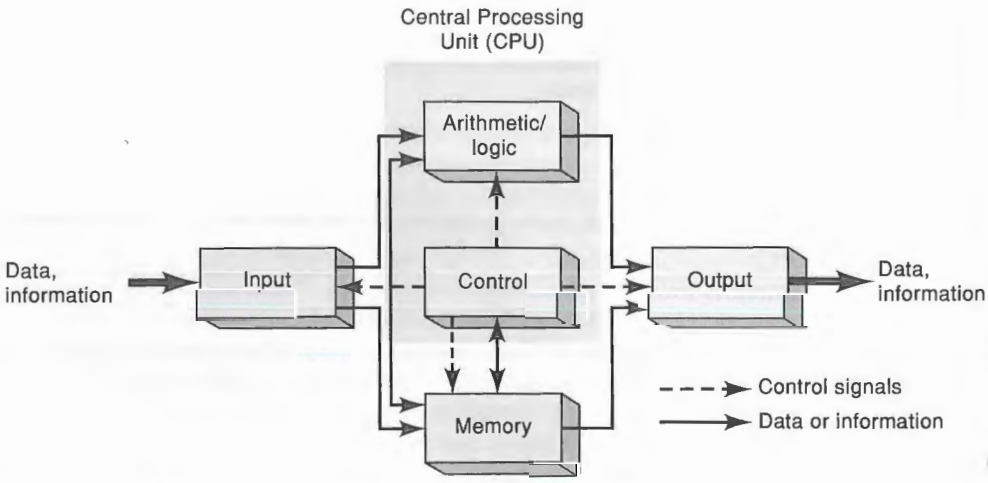


FIGURE 1-18 Functional diagram of a digital computer.

The major functions of each unit are:

1. **Input unit.** Through this unit, a complete set of instructions and data is fed into the computer system and into the memory unit, to be stored until needed. The information typically enters the input unit from a keyboard, a disk, or various sensors (in the case of a process control computer).
2. **Memory unit.** The memory stores the instructions and data received from the input unit. It stores the results of arithmetic operations received from the arithmetic unit. It also supplies information to the output unit.
3. **Control unit.** This unit takes instructions from the memory unit one at a time and interprets them. It then sends appropriate signals to all the other units to cause the specific instruction to be executed.
4. **Arithmetic/logic unit.** All arithmetic calculations and logical decisions are performed in this unit, which can then send results to the memory unit to be stored.
5. **Output unit.** This unit takes data from the memory unit and prints out, displays, or otherwise presents the information to the operator (or process, in the case of a process control computer).

As the diagram in Figure 1-18 shows, the control and arithmetic/logic units are often considered as one unit, called the **central processing unit (CPU).** The CPU contains all of the circuitry for fetching and interpreting instructions and for controlling and performing the various operations called for by the instructions.

Types of Computers

All computers are made up of the basic units described above, but they can differ as to physical size, operating speed, memory capacity, and computational power, as well as other characteristics. Computer systems are configured in many and various ways today, with many common characteristics and distinguishing differences. Large computer systems that are permanently installed in multiple cabinets are used by corporations and universities for information technology support. Desktop personal computers are used in our homes and offices to run useful application programs that enhance our lives and provide communication with other computers. Portable computers are found in PDAs and specialized computers are found in video game systems. The most prevalent form of computers can be found performing dedicated routine tasks in appliances and systems all around us.

Today, all of these systems utilize technology that has evolved from the invention of the **microprocessor.** The microprocessor is essentially a central processing unit (CPU) in an integrated circuit that can be connected to the other blocks of a computer system. Computers that use a microprocessor as their CPU are usually referred to as **microcomputers.** The general-purpose microcomputers (e.g., PCs, PDAs, etc.) perform a variety of tasks in a wide range of applications depending on the software (programs) they are running. Contrast these with the dedicated computers that are doing things such as operating your car's engine, controlling your car's antilock braking system, or running your microwave oven. These computers cannot be programmed by the user, but simply perform their intended control task: they are referred to as **microcontrollers.** Since these microcontrollers are an integral part of a bigger system and serve a dedicated purpose, they also are called *embedded controllers.* Microcontrollers generally have all the elements of a complete computer (CPU, memory, and input/output ports), all contained

digital **multiplexers** and **demultiplexers** which are covered in detail in Chapter 9.

Digital Progress Today and Tomorrow

Let's step back and review the reason we have been discussing cell phones. This little cell phone is made up of fundamental digital and analog circuits that are equally useful in other modern electronic systems such as Blu-Ray video, computers of all sizes, broadband internet service hardware, appliances, automotive systems, and video games, just to name a few. There are many needs in the world that digital technology can meet. Your cell phone is constantly managing the power usage from the battery so it can last for days between charges. Very soon our homes and businesses will be managing their power usage automatically to conserve energy just like your phone. An entire new digital industry will spring up to control appliances and outlets for this purpose. The FCC is discontinuing the analog transmission of the NTSC format television signal and is supporting only the transmission of the new digital format. The HDTV revolution is well under way. Wireless WIFI networks allow us to use computers on line in restaurants, airports, and all over campus. Your GPS system picks up microwave signals from space to tell you exactly where you are on a map of the world and help you navigate to your destination. Farmers already let their tractors drive themselves through the fields. How soon will the car drive itself? You will be able to become one of the pioneers on these new frontiers of technology.

REVIEW QUESTIONS

1. Explain how a digital circuit that has memory differs from one that does not.
2. Name the five major functional units of a computer.
3. Which two units make up the CPU?
4. An IC chip that contains a CPU is called a _____.
5. Name three fundamental digital circuits that can be found in the cell phone system.

SUMMARY

1. The two basic ways of representing the numerical value of physical quantities are analog (continuous) and digital (discrete).
2. Most quantities in the real world are analog, but digital techniques are generally superior to analog techniques, and most of the predicted advances will be in the digital realm.
3. The binary number system (0 and 1) is the basic system used in digital technology.
4. Digital or logic circuits operate on voltages that fall in prescribed ranges that represent either a binary 0 or a binary 1.
5. The two basic ways to transfer digital information are parallel—all bits simultaneously—and serial—one bit at a time.
6. The main parts of all computers are the input, control, memory, arithmetic/logic, and output units.

ENGR10000
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

7. The combination of the arithmetic/logic unit and the control unit makes up the CPU (central processing unit).
8. A microcomputer usually has a CPU that is on a single chip called a *microprocessor.*
9. A microcontroller is a microcomputer especially designed for dedicated (not general-purpose) control applications.

IMPORTANT TERMS†

encode
timing diagram
analog representation
digital representation
frequency (f)
period (T)
amplitude
digital system
analog system
decode
binary number system
binary digit
analog-to-digital converter (ADC)
digital-to-analog converter (DAC)
decimal system
hexadecimal
binary system
bit
digital circuits/logic circuits
CMOS
TTL
parallel transmission
serial transmission
memory
digital computer
program
input unit
memory unit
control unit
arithmetic/logic unit
output unit
central processing unit (CPU)
microprocessor
microcomputer
microcontroller
counter
decoder
cascading
multiplexer
demultiplexer

PROBLEMS

SECTION 1-3

1-1.* Which of the following are analog quantities, and which are digital?
(a) Number of atoms in a sample of material
(b) Altitude of an aircraft
(c) Pressure in a bicycle tire
(d) Current through a speaker
(e) Timer setting on a microwave oven

1-2. Which of the following are analog quantities, and which are digital?
(a) Width of a piece of lumber
(b) The amount of time before the oven buzzer goes off
(c) The time of day displayed on a quartz watch
(d) Altitude above sea level measured on a staircase
(e) Altitude above sea level measured on a ramp

SECTION 1-4

1-3.* Convert the following binary numbers to their equivalent decimal values.
(a) 11001_2
(b) 1001.1001_2
(c) 10011011001.10110_2

†These terms can be found in **boldface** type in the chapter and are defined in the Glossary at the end of the book. This applies to all chapters.

*Answers to problems marked with an asterisk can be found in the back of the text.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications,* (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.

1-4. Convert the following binary numbers to decimal.
   (a) 10011_2
   (b) 1100.0101
   (c) 10011100100.10010

1-5.* Using three bits, show the binary counting sequence from 000 to 111.

1-6. Using six bits, show the binary counting sequence from 000000 to 111111.

1-7.* What is the maximum number that we can count up to using 10 bits?

1-8. What is the maximum number that we can count up to using 14 bits?

1-9.* How many bits are needed to count up to a maximum of 511?

1-10. How many bits are needed to count up to a maximum of 63?

SECTION 1-5

1-11.* Draw the timing diagram for a digital signal that continuously alternates between 0.2 V (binary 0) for 2 ms and 4.4 V (binary 1) for 4 ms.

1-12. Draw the timing diagram for a signal that alternates between 0.3 V (binary 0) for 5 ms and 3.9 V (binary 1) for 2 ms.

SECTION 1-7

1-13.* Suppose that the decimal integer values from 0 to 15 are to be transmitted in binary.
   (a) How many lines will be needed if parallel representation is used?
   (b) How many will be needed if serial representation is used?

SECTIONS 1-8 AND 1-9

1-14. How is a microprocessor different from a microcomputer?

1-15. How is a microcontroller different from a microcomputer?

ANSWERS TO SECTION REVIEW QUESTIONS

SECTION 1-1

1. Two 2. A timing diagram

SECTION 1-2

1. Code key/clacker down (activated) and code key/clacker up (not activated)
2. Long and short intervals of holding the code key down represent dots and dashes.
3. Amplitude 4. Frequency 5. Digital 6. Analog

SECTION 1-3

1. Morse code (width and pattern of pulses), pulse dial code, binary numbers
2. Signaling and switching were digital; voice transmission was analog.
3. Signaling, switching and audio is digitally encoded; radio frequencies that carry the digital information are analog. The microphone changes sound to an analog signal and the speaker changes an analog signal into sound. 4. Easier to design; easier to store information; greater accuracy and precision; programmability; less affected by noise; higher degree of integration 5. Real-world physical quantities are analog. Digital processing takes time.

SECTION 1-4

1. 107_{10} 2. 11000_2 3. 4095_{10}

SECTION 1-5

1. A sequence of binary numbers, representing the signal's value measured at regular intervals 2. As a voltage that is within an acceptable range of HIGH or LOW values

SECTION 1-6

1. False 2. Yes, provided that the two input voltages are within the same logic level range 3. Logic 4. Timing diagram

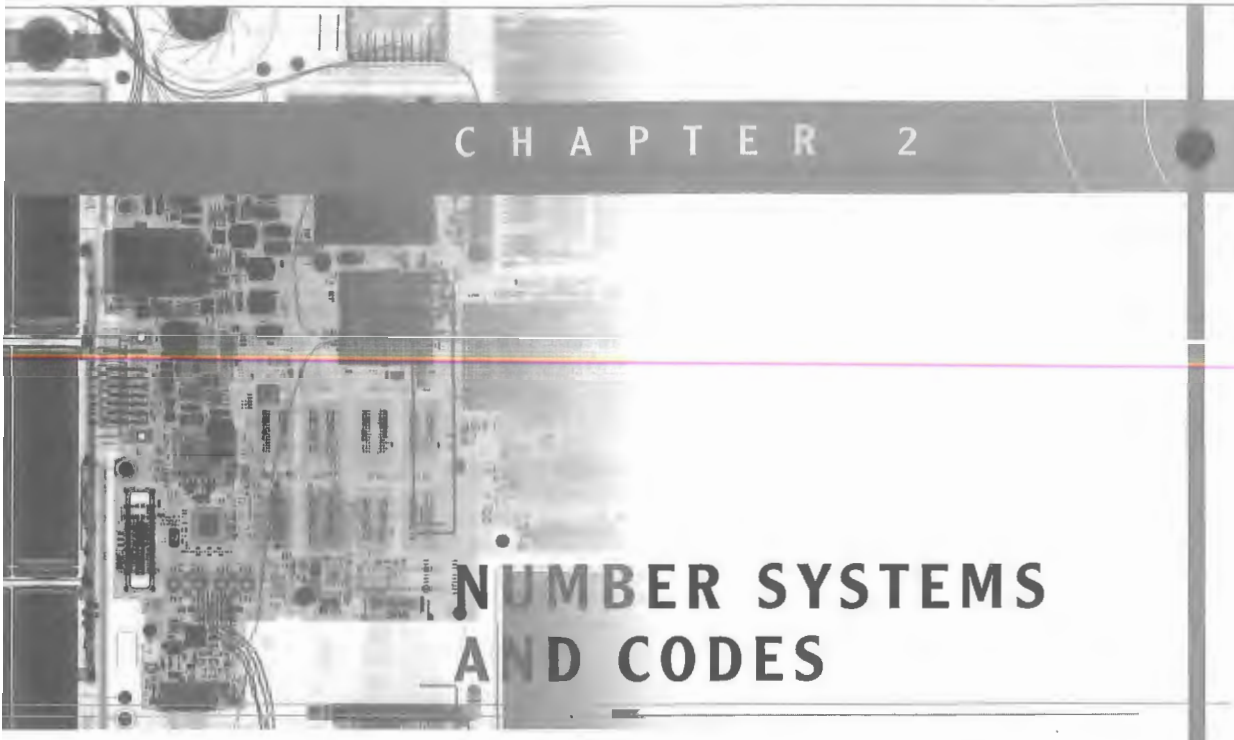
SECTION 1-7

1. Parallel is faster; serial requires only one signal line.

SECTION 1-9

1. One that has memory will have its output changed and *remain* changed in response to a momentary change in the input signal. 2. Input, output, memory, arithmetic/logic, control 3. Control and arithmetic/logic 4. Microprocessor 5. Counter, decoder, multiplexer, demultiplexer

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Introductory Concepts. In *Digital Systems: Principles and Applications*, (11th Ed. , pp. 2-31). Upper Saddle River, NJ: Pearson Education Inc.



CHAPTER 2

NUMBER SYSTEMS AND CODES

OUTLINE

2-1 Binary-to-Decimal Conversions
2-2 Decimal-to-Binary Conversions
2-3 Hexadecimal Number System
2-4 BCD Code
2-5 The Gray Code
2-6 Putting It All Together
2-7 The Byte, Nibble, and Word
2-8 Alphanumeric Codes
2-9 Parity Method for Error Detection
2-10 Applications

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

OBJECTIVES

Upon completion of this chapter, you will be able to:

- Convert a number from one number system (decimal, binary, hexadecimal) to its equivalent in one of the other number systems.
- Cite the advantages of the hexadecimal number system.
- Count in hexadecimal.
- Represent decimal numbers using the BCD code; cite the pros and cons of using BCD.
- Explain the difference between BCD and straight binary.
- Explain the purpose of alphanumeric codes such as the ASCII code.
- Explain the parity method for error detection.
- Determine the parity bit to be attached to a digital data string.

INTRODUCTION

The binary number system is the most important one in digital systems, but several others are also important. The decimal system is important because it is universally used to represent quantities outside a digital system. This means that there will be situations where decimal values must be converted to binary values before they are entered into the digital system. For example, when you punch a decimal number into your hand calculator (or computer), the circuitry inside the machine converts the decimal number to a binary value.

Likewise, there will be situations where the binary values at the outputs of a digital system must be converted to decimal values for presentation to the outside world. For example, your calculator (or computer) uses binary numbers to calculate answers to a problem and then converts the answers to decimal digits before displaying them.

As you will see, it is not easy to simply look at a large binary number and convert it to its equivalent decimal value. It is very tedious to enter a long sequence of 1s and 0s on a keypad, or to write large binary numbers on a piece of paper. It is especially difficult to try to convey a binary quantity while speaking to someone. The hexadecimal (base-16) number system has become a standard way of communicating numeric values in digital systems. The great advantage is that hexadecimal numbers can be converted easily to and from binary. You will find that many advanced computer tools, which are designed to help software developers troubleshoot or debug their programs, use the hexadecimal number system to enter numbers that are stored in the computer as binary and display them again as hexadecimal.

Other methods of representing decimal quantities with binary-encoded digits have been devised that are not truly number systems but offer the ease of conversion between the binary code and the decimal number system. This is referred to as binary-coded decimal. Quantities and patterns

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

of bits might be represented by any of these methods in any given system and throughout the written material that supports the system, so it is very important that you are able to interpret values in any system and convert between any of these numeric representations. Other codes that use 1s and 0s to represent things such as alphanumeric characters will be covered because they are so common in digital systems.

2-1 BINARY-TO-DECIMAL CONVERSIONS

As explained in Chapter 1, the binary number system is a positional system where each binary digit (bit) carries a certain weight based on its position relative to the LSB. Any binary number can be converted to its decimal equivalent simply by summing together the weights of the various positions in the binary number that contain a 1. To illustrate, let's change 11011_2 to its decimal equivalent.

$$\begin{aligned} &1 \quad 1 \quad 0 \quad 1 \quad 1_2 \\ &2^4 + 2^3 + 0 + 2^1 + 2^0 = 16 + 8 + 2 + 1 \\ &\qquad\qquad\qquad\qquad\quad = 27_{10} \end{aligned}$$

Let's try another example with a greater number of bits:

$$\begin{aligned} &1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1_2 = \\ &2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 181_{10} \end{aligned}$$

Note that the procedure is to find the weights (i.e., powers of 2) for each bit position that contains a 1, and then to add them up. Also note that the MSB has a weight of 2^7 even though it is the eighth bit; this is because the LSB is the first bit and has a weight of 2^0.

Another method of binary-to-decimal conversion that avoids the addition of large numbers and keeping track of column weights is called the double-dabble method. The procedure is as follows:

1. Write down the left-most 1 in the binary number.
2. Double it and add the next bit to the right.
3. Write down the result under the next bit.
4. Continue with steps 2 and 3 until finished with the binary number.

Let's use the same binary numbers to verify this method.

$$\begin{array}{lllllll} \text{Given:} & 1 & 1 & 0 & 1 & 1_2 \\ \text{Results:} & 1 \times 2 = 2 & & & & \\ & & \underline{+1} & & & \\ & & 3 \times 2 = 6 & & & \\ & & & \underline{+0} & & \\ & & & 6 \times 2 = 12 & & \\ & & & & \underline{+1} & \\ & & & & 13 \times 2 = 26 & \\ & & & & & \underline{+1} \\ & & & & & 27_{10} \end{array}$$

Given:	1	0	1	1	0	1	0	1_2
Results:	1 →	2 →	5 →	11 →	22 →	45 →	90 →	181_{10}

REVIEW QUESTIONS

1. Convert 100011011011_2 to its decimal equivalent by adding the products of digits and weights.
2. What is the weight of the MSB of a 16-bit number?
3. Repeat the conversion in Question 1 using the double-dabble method.

2-2 DECIMAL-TO-BINARY CONVERSIONS

There are two ways to convert a decimal *whole* number to its equivalent binary-system representation. The first method is the reverse of the first process described in Section 2-1. The decimal number is simply expressed as a sum of powers of 2, and then 1s and 0s are written in the appropriate bit positions. To illustrate:

$$\begin{aligned} 45_{10} = 32 + 8 + 4 + 1 &= 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 \\ &= 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1_2 \end{aligned}$$

Note that a 0 is placed in the 2^1 and 2^4 positions, since all positions must be accounted for. Another example is the following:

$$\begin{aligned} 76_{10} = 64 + 8 + 4 &= 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 0 \\ &= 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0_2 \end{aligned}$$

Another method for converting decimal integers uses repeated division by 2. The conversion, illustrated below for 25_{10}, requires repeatedly dividing the decimal number by 2 and writing down the remainder after each division until a quotient of 0 is obtained. Note that the binary result is obtained by writing the first remainder as the LSB and the last remainder as the MSB. This process, diagrammed in the flowchart of Figure 2-1, can also be used to convert from decimal to any other number system, as we shall see.

$$\frac{25}{2} = 12 + \text{remainder of } 1 \quad \text{LSB}$$

$$\frac{12}{2} = 6 + \text{remainder of } 0$$

$$\frac{6}{2} = 3 + \text{remainder of } 0$$

$$\frac{3}{2} = 1 + \text{remainder of } 1$$

$$\frac{1}{2} = 0 + \text{remainder of } 1 \quad \text{MSB}$$

$$25_{10} = 1 \quad 1 \quad 0 \quad 0 \quad 1_2$$

CALCULATOR HINT:

If you use a calculator to perform the divisions by 2, you can tell whether the remainder is 0 or 1 by whether or not the result has a fractional part. For instance, 25/2 would produce 12.5. Since there is a fractional part (the .5), the remainder is a 1. If there were no fractional part, such as 12/2 = 6, then the remainder would be 0. Example 2-1 illustrates this.

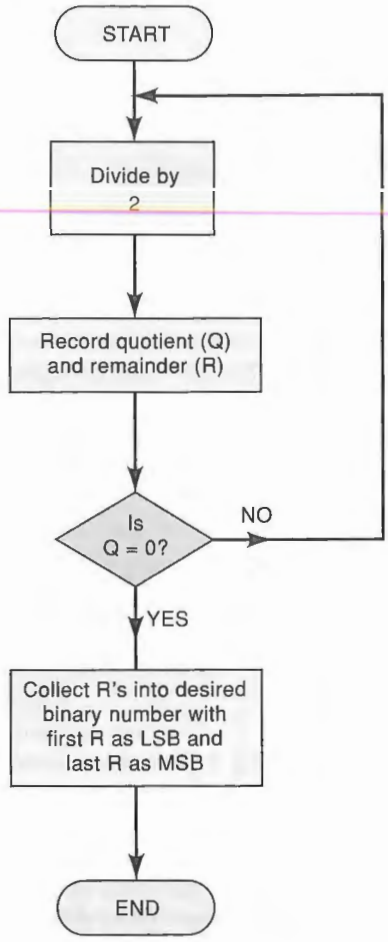


FIGURE 2-1 Flowchart for repeated-division method of decimal-to-binary conversion of integers. The same process can be used to convert a decimal integer to any other number system.

EXAMPLE 2-1

Convert 37_{10} to binary. Try to do it on your own before you look at the solution.

Solution

$$\frac{37}{2} = 18.5 \longrightarrow \text{remainder of 1 (LSB)}$$

$$\frac{18}{2} = 9.0 \longrightarrow \quad 0$$

$$\frac{9}{2} = 4.5 \longrightarrow \quad 1$$

$$\frac{4}{2} = 2.0 \longrightarrow \quad 0$$

$$\frac{2}{2} = 1.0 \longrightarrow \quad 0$$

$$\frac{1}{2} = 0.5 \longrightarrow \quad 1 \text{ (MSB)}$$

Thus, $37^{10} = 100101_2$.

Counting Range

Recall that using N bits, we can count through 2^N different decimal numbers ranging from 0 to $2^N - 1$. For example, for $N = 4$, we can count from 0000_2 to 1111_2, which is 0_{10} to 15_{10}, for a total of 16 different numbers. Here, the largest decimal value is $2^4 - 1 = 15$, and there are 2^4 different numbers.

In general, then, we can state:

Using N bits, we can represent decimal numbers ranging from 0 to $2^N - 1$, a total of 2^N different numbers.

EXAMPLE 2-2

(a) What is the total range of decimal values that can be represented in eight bits?

(b) How many bits are needed to represent decimal values ranging from 0 to 12,500?

Solution

(a) Here we have $N = 8$. Thus, we can represent decimal numbers from 0 to $2^8 - 1 = 255$. We can verify this by checking to see that 11111111_2 converts to 255_{10}.

(b) With 13 bits, we can count from decimal 0 to $2^{13} - 1 = 8191$. With 14 bits, we can count from 0 to $2^{14} - 1 = 16{,}383$. Clearly, 13 bits aren't enough, but 14 bits will get us up beyond 12,500. Thus, the required number of bits is **14**.

REVIEW QUESTIONS

1. Convert 83_{10} to binary using both methods.
2. Convert 729_{10} to binary using both methods. Check your answer by converting back to decimal.
3. How many bits are required to count up to decimal 1 million?

2-3 HEXADECIMAL NUMBER SYSTEM

The **hexadecimal number system** uses base 16. Thus, it has 16 possible digit symbols. It uses the digits 0 through 9 plus the letters A, B, C, D, E, and F as the 16 digit symbols. The digit positions are weighted as powers of 16 as shown below, rather than as powers of 10 as in the decimal system.

16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}

Hexadecimal point

Table 2-1 shows the relationships among hexadecimal, decimal, and binary. Note that each hexadecimal digit represents a group of four binary digits. It is important to remember that hex (abbreviation for "hexadecimal") digits A through F are equivalent to the decimal values 10 through 15.

TABLE 2-1

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Hex-to-Decimal Conversion

A hex number can be converted to its decimal equivalent by using the fact that each hex digit position has a weight that is a power of 16. The LSD has a weight of $16^0 = 1$; the next higher digit position has a weight of $16^1 = 16$; the next has a weight of $16^2 = 256$; and so on. The conversion process is demonstrated in the examples below.

CALCULATOR HINT:

You can use the y^x calculator function to evaluate the powers of 16.

$$\begin{aligned} 356_{16} &= 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 \\ &= 768 + 80 + 6 \\ &= 854_{10} \end{aligned}$$

$$\begin{aligned} 2AF_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 512 + 160 + 15 \\ &= 687_{10} \end{aligned}$$

Note that in the second example, the value 10 was substituted for A and the value 15 for F in the conversion to decimal.

For practice, verify that $1BC2_{16}$ is equal to 7106_{10}.

Decimal-to-Hex Conversion

Recall that we did decimal-to-binary conversion using repeated division by 2. Likewise, decimal-to-hex conversion can be done using repeated division by 16 (Figure 2-1). The following example contains two illustrations of this conversion.

2-3

(a) Convert 423_{10} to hex.

Solution

$$\frac{423}{16} = 26 + \text{remainder of } 7$$

$$\frac{26}{16} = 1 + \text{remainder of } 10$$

$$\frac{1}{16} = 0 + \text{remainder of } 1$$

$$423_{10} = 1A7_{16}$$

(b) Convert 214_{10} to hex.

Solution

$$\frac{214}{16} = 13 + \text{remainder of } 6$$

$$\frac{13}{16} = 0 + \text{remainder of } 13$$

$$214_{10} = D6_{16}$$

Again note that the remainders of the division processes form the digits of the hex number. Also note that any remainders that are greater than 9 are represented by the letters A through F.

CALCULATOR HINT:

If a calculator is used to perform the divisions in the conversion process, the results will include a decimal fraction instead of a remainder. The remainder can be obtained by multiplying the fraction by 16. To illustrate, in Example 2-3(b), the calculator would have produced

$$\frac{214}{16} = 13.375$$

The remainder becomes $(0.375) \times 16 = 6$.

Hex-to-Binary Conversion

The hexadecimal number system is used primarily as a "shorthand" method for representing binary numbers. It is a relatively simple matter to convert a hex number to binary. *Each* hex digit is converted to its four-bit binary equivalent (Table 2-1). This is illustrated below for $9F2_{16}$.

$$\begin{aligned} 9F2_{16} &= \quad 9 \qquad\qquad F \qquad\qquad 2 \\ &\qquad\;\downarrow \qquad\qquad\;\downarrow \qquad\qquad\;\downarrow \\ &= 1\;\;0\;\;0\;\;1 \quad 1\;\;1\;\;1\;\;1 \quad 0\;\;0\;\;1\;\;0 \\ &= 100111110010_2 \end{aligned}$$

For practice, verify that $BA6_{16} = 101110100110_2$.

Binary-to-Hex Conversion

Conversion from binary to hex is just the reverse of the process above. The binary number is grouped into groups of *four* bits, and each group is converted to its equivalent hex digit. Zeros (shown shaded) are added, as needed, to complete a four-bit group.

$$1\,1\,1\,0\,1\,0\,0\,1\,1\,0_2 = \underbrace{0\,0\,1\,1}_{3}\ \underbrace{1\,0\,1\,0}_{A}\ \underbrace{0\,1\,1\,0}_{6}$$
$$= 3A6_{16}$$

To perform these conversions between hex and binary, it is necessary to know the four-bit binary numbers (0000 through 1111) and their equivalent hex digits. Once these are mastered, the conversions can be performed quickly without the need for any calculations. This is why hex is so useful in representing large binary numbers.

For practice, verify that $101011111_2 = 15F_{16}$.

Counting in Hexadecimal

When counting in hex, each digit position can be incremented (increased by 1) from 0 to F. Once a digit position reaches the value F, it is reset to 0, and the next digit position is incremented. This is illustrated in the following hex counting sequences:

(a) 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42

(b) 6F8, 6F9, 6FA, 6FB, 6FC, 6FD, 6FE, 6FF, 700

Note that when there is a 9 in a digit position, it becomes an A when it is incremented.

With N hex digit positions, we can count from decimal 0 to $16^N - 1$, for a total of 16^N different values. For example, with three hex digits, we can count from 000_{16} to FFF_{16}, which is 0_{10} to 4095_{10}, for a total of $4096 = 16^3$ different values.

Usefulness of Hex

Hex is often used in a digital system as sort of a "shorthand" way to represent strings of bits. In computer work, strings as long as 64 bits are not uncommon. These binary strings do not always represent a numerical value, but—as you will find out—can be some type of code that conveys nonnumerical information. When dealing with a large number of bits, it is more convenient and less error-prone to write the binary numbers in hex and, as we have seen, it is relatively easy to convert back and forth between binary and hex. To illustrate the advantage of hex representation of a binary string, suppose you had in front of you a printout of the contents of 50 memory locations, each of which was a 16-bit number, and you were checking it against a list. Would you rather check 50 numbers like this one: 0110111001100111, or 50 numbers like this one: 6E67? And which one would you be more apt to read incorrectly? It is important to keep in mind, though, that digital circuits all work in binary. Hex is simply used as a convenience for the humans involved. You should memorize the 4-bit binary pattern for each hexadecimal digit. Only then will you realize the usefulness of this tool in digital systems.

ENGR1000 A publication publishing exclusively for The University of Melbourne page 41

EXAMPLE 2-4

Convert decimal 378 to a 16-bit binary number by first converting to hexadecimal.

Solution

$$\frac{378}{16} = 23 + \text{remainder of } 10_{10} = A_{16}$$

$$\frac{23}{16} = 1 + \text{remainder of } 7$$

$$\frac{1}{16} = 0 + \text{remainder of } 1$$

Thus, $378_{10} = 17A_{16}$. This hex value can be converted easily to binary 000101111010. Finally, we can express 378_{10} as a 16-bit number by adding four leading 0s:

$$378_{10} = 0000 \quad 0001 \quad 0111 \quad 1010_2$$

EXAMPLE 2-5

Convert $B2F_{16}$ to decimal.

Solution

$$\begin{aligned} B2F_{16} &= B \times 16^2 + 2 \times 16^1 + F \times 16^0 \\ &= 11 \times 256 + 2 \times 16 + 15 \\ &= 2863_{10} \end{aligned}$$

Summary of Conversions

Right now, your head is probably spinning as you try to keep straight all of these different conversions from one number system to another. You probably realize that many of these conversions can be done *automatically* on your calculator just by pressing a key, but it is important for you to master these conversions so that you understand the process. Besides, what happens if your calculator battery dies at a crucial time and you have no handy replacement? The following summary should help you, but nothing beats practice, practice, practice!

1. When converting from binary [or hex] to decimal, use the method of taking the weighted sum of each digit position or follow the double-dabble procedure.
2. When converting from decimal to binary [or hex], use the method of repeatedly dividing by 2 [or 16] and collecting remainders (Figure 2-1).
3. When converting from binary to hex, group the bits in groups of four, and convert each group into the correct hex digit.
4. When converting from hex to binary, convert each digit into its four-bit equivalent.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

REVIEW QUESTIONS

1. Convert $24CE_{16}$ to decimal.
2. Convert 3117_{10} to hex, then from hex to binary.
3. Convert $1001\,0111\,1011\,0101_2$ to hex.
4. Write the next four numbers in this hex counting sequence: E9A, E9B, E9C, E9D, ____, ____, ____, ____.
5. Convert 3527 to binary_{16}.
6. What range of decimal values can be represented by a four-digit hex number?

2-4 BCD CODE

When numbers, letters, or words are represented by a special group of symbols, we say that they are being encoded, and the group of symbols is called a *code*. Probably one of the most familiar codes is the Morse code, where a series of dots and dashes represents letters of the alphabet.

We have seen that any decimal number can be represented by an equivalent binary number. The group of 0s and 1s in the binary number can be thought of as a code representing the decimal number. When a decimal number is represented by its equivalent binary number, we call it **straight binary coding**.

Digital systems all use some form of binary numbers for their internal operation, but the external world is decimal in nature. This means that conversions between the decimal and binary systems are being performed often. We have seen that the conversions between decimal and binary can become long and complicated for large numbers. For this reason, a means of encoding decimal numbers that combines some features of both the decimal and the binary systems is used in certain situations.

Binary-Coded-Decimal Code

If *each* digit of a decimal number is represented by its binary equivalent, the result is a code called **binary-coded-decimal** (hereafter abbreviated BCD). Since a decimal digit can be as large as 9, four bits are required to code each digit (the binary code for 9 is 1001).

To illustrate the BCD code, take a decimal number such as 874. Each *digit* is changed to its binary equivalent as follows:

8	7	4	(decimal)
↓	↓	↓	
1000	0111	0100	(BCD)

As another example, let us change 943 to its BCD-code representation:

9	4	3	(decimal)
↓	↓	↓	
1001	0100	0011	(BCD)

Once again, each decimal digit is changed to its straight binary equivalent. Note that four bits are *always* used for each digit.

The BCD code, then, represents each digit of the decimal number by a four-bit binary number. Clearly only the four-bit binary numbers from 0000

through 1001 are used. The BCD code does not use the numbers 1010, 1011, 1100, 1101, 1110, and 1111. In other words, only 10 of the 16 possible four-bit binary code groups are used. If any of the "forbidden" four-bit numbers ever occurs in a machine using the BCD code, it is usually an indication that an error has occurred.

EXAMPLE 2-6

Convert 0110100000111001 (BCD) to its decimal equivalent.

Solution

Divide the BCD number into four-bit groups and convert each to decimal.

$$\underbrace{0110}_{6}\ \underbrace{1000}_{8}\ \underbrace{0011}_{3}\ \underbrace{1001}_{9}$$

EXAMPLE 2-7

Convert the BCD number 011111000001 to its decimal equivalent.

Solution

$$\underbrace{0111}_{7}\ \underset{\downarrow}{1100}\ \underbrace{0001}_{1}$$

The forbidden code group indicates an error in the BCD number

Comparison of BCD and Binary

It is important to realize that BCD is not another number system like binary, decimal, and hexadecimal. In fact, it is the decimal system with each digit encoded in its binary equivalent. It is also important to understand that a BCD number is *not* the same as a straight binary number. A straight binary number takes the *complete* decimal number and represents it in binary; the BCD code converts *each* decimal *digit* to binary individually. To illustrate, take the number 137 and compare its straight binary and BCD codes:

$$137_{10} = 10001001_2 \quad \text{(binary)}$$
$$137_{10} = 0001\ 0011\ 0111 \quad \text{(BCD)}$$

The BCD code requires 12 bits, while the straight binary code requires only eight bits to represent 137. BCD requires more bits than straight binary to represent decimal numbers of more than one digit because BCD does not use all possible four-bit groups, as pointed out earlier, and is therefore somewhat inefficient.

The main advantage of the BCD code is the relative ease of converting to and from decimal. Only the four-bit code groups for the decimal digits 0 through 9 need to be remembered. This ease of conversion is especially important from a hardware standpoint because in a digital system, it is the logic circuits that perform the conversions to and from decimal.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

EXAMPLE 2-8

An automatic teller machine (ATM) at the bank allows you to enter the amount of cash you wish to withdraw in decimal by pressing decimal digit symbol keys. Does the computer convert this decimal number into straight binary or BCD? Explain.

Solution

The number that represents your balance (all the money you have in the bank) is stored as a straight binary number. When the withdrawn amount is entered, it must be subtracted from the balance. Since arithmetic must be performed on the numbers, both values (the balance and the withdrawn cash) must be in straight binary. It converts the decimal entry to straight binary.

EXAMPLE 2-9

Your cell phone allows you to enter/store a 10-decimal-digit phone number. Does the cell phone store the phone number in straight binary or BCD? Explain.

Solution

A phone number is a combination of many decimal digits. It is not necessary to mathematically combine the digits (i.e., you never add two phone numbers together). The phone just needs to store them in the sequence they were entered and retrieve them one at a time when you press *send*. Therefore, they will be stored as BCD digits in the memory of the computer in your cell phone.

REVIEW QUESTIONS

1. Represent the decimal value 178 by its straight binary equivalent. Then encode the same decimal number using BCD.
2. How many bits are required to represent an eight-digit decimal number in BCD?
3. What is an advantage of encoding a decimal number in BCD rather than in straight binary? What is a disadvantage?

2-5 THE GRAY CODE

Digital systems operate at very fast speeds and respond to changes that occur in the digital inputs. Just as in life, when multiple input conditions are changing at the same time, the situation can be misinterpreted and cause an erroneous reaction. When you look at the bits in a binary count sequence, it is clear that there are often several bits that must change states at the same time. For example, consider when the three-bit binary number for 3 changes to 4: all three bits must change state.

In order to reduce the likelihood of a digital circuit misinterpreting a changing input, the **Gray code** has been developed as a way to represent a sequence of numbers. The unique aspect of the Gray code is that only one bit ever changes between two successive numbers in the sequence. Table 2-2 shows the translation between three-bit binary and Gray code values. To

ENGR10003 – ENGINEERING SYSTEMS
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

TABLE 2-2 Three-bit binary and Gray code equivalents.

B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

convert binary to Gray, simply start on the most significant bit and use it as the Gray MSB as shown in Figure 2-2(a). Now compare the MSB binary with the next binary bit (B1). If they are the same, then G1 = 0. If they are different, then G1 = 1. G0 can be found by comparing B1 with B0.

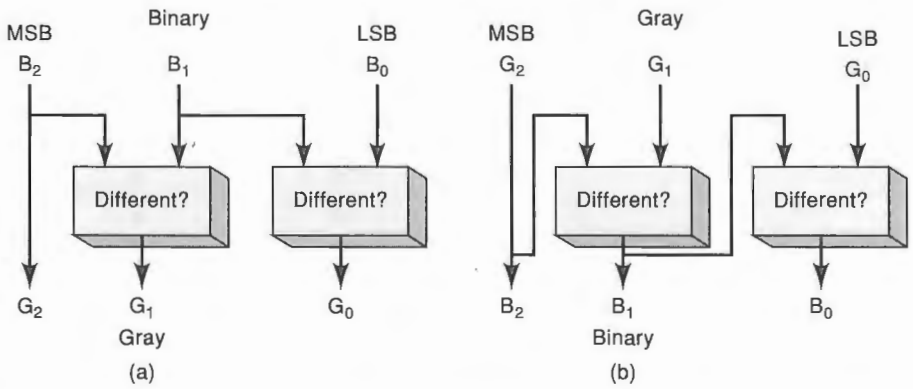


FIGURE 2-2 Converting (a) binary to Gray and (b) Gray to binary.

Conversion from Gray code back into binary is shown in Figure 2-2(b). Note that the MSB in Gray is always the same as the MSB in binary. The next binary bit is found by comparing the *binary* bit to the left with the *corresponding Gray code bit*. Similar bits produce a 0 and differing bits produce a 1. The most common application of the Gray code is in shaft position encoders as shown in Figure 2-3. These devices produce a binary value that represents the position of a rotating mechanical shaft. A practical shaft encoder would use many more bits than just three and divide the rotation into many more segments than eight, so that it could detect much smaller increments of rotation.

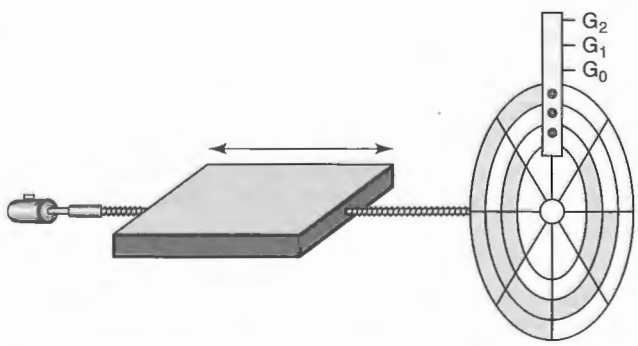


FIGURE 2-3 An eight-position, three-bit shaft encoder.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

REVIEW QUESTIONS

1. Convert the number 0101 (binary) to its Gray code equivalent.
2. Convert 0101 (Gray code) to its binary number equivalent.

2-6 PUTTING IT ALL TOGETHER

Table 2-3 gives the representation of the decimal numbers 1 through 15 in the binary and hex number systems and also in the BCD and Gray codes. Examine it carefully and make sure you understand how it was obtained. Especially note how the BCD representation always uses four bits for each decimal digit.

TABLE 2-3

Decimal	Binary	Hexadecimal	BCD	GRAY
0	0	0	0000	0000
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

2-7 THE BYTE, NIBBLE, AND WORD

Bytes

Most microcomputers handle and store binary data and information in groups of eight bits, so a special name is given to a string of eight bits: it is called a **byte**. A byte always consists of eight bits, and it can represent any of numerous types of data or information. The following examples will illustrate.

EXAMPLE 2-10

How many bytes are in a 32-bit string (a string of 32 bits)?

Solution

32/8 = 4, so there are **four** bytes in a 32-bit string.

EXAMPLE 2-11

What is the largest decimal value that can be represented in binary using two bytes?

Solution

Two bytes is 16 bits, so the largest binary value will be equivalent to decimal $2^{16} - 1 = 65{,}535$.

EXAMPLE 2-12

How many bytes are needed to represent the decimal value 846,569 in BCD?

Solution

Each decimal digit converts to a four-bit BCD code. Thus, a six-digit decimal number requires 24 bits. These 24 bits are equal to three bytes. This is diagrammed below.

$$\begin{array}{ccccccc} & 8 & 4 & 6 & 5 & 6 & 9 \quad \text{(decimal)} \\ \underbrace{1000\ 0100}_{\text{byte1}} & & \underbrace{0110\ 0101}_{\text{byte2}} & & \underbrace{0110\ 1001}_{\text{byte3}} & & \text{(BCD)} \end{array}$$

Nibbles

Binary numbers are often broken down into groups of four bits, as we have seen with BCD codes and hexadecimal number conversions. In the early days of digital systems, a term caught on to describe a group of four bits. Because it is half as big as a byte, it was named a **nibble**. The following examples illustrate the use of this term.

EXAMPLE 2-13

How many nibbles are in a byte?

Solution

2

EXAMPLE 2-14

What is the hex value of the least significant nibble of the binary number 10010101?

Solution

1001 0101

The least significant nibble is 0101 = 5.

Words

Bits, nibbles, and bytes are terms that represent a fixed number of binary digits. As systems have grown over the years, their capacity (appetite?) for

Solution

Locate each character (including the space) in Table 2-4 and record its ASCII code.

i	69	0110	1001
f	66	0110	0110
space	20	0010	0000
(	28	0010	1000
x	78	0111	1000
>	3E	0011	1110
3	33	0011	0011
)	29	0010	1001

Note that a 0 was added to the leftmost bit of each ASCII code because the codes must be stored as bytes (eight bits). This adding of an extra bit is called *padding with 0s.*

REVIEW QUESTIONS

1. Encode the following message in ASCII code using the hex representation: "COST = $72."
2. The following padded ASCII-coded message is stored in successive memory locations in a computer:

01010011 01010100 01001111 01010000

What is the message?

2-9 PARITY METHOD FOR ERROR DETECTION

The movement of binary data and codes from one location to another is the most frequent operation performed in digital systems. Here are just a few examples:

- The transmission of digitized voice over a microwave link
- The storage of data in and retrieval of data from external memory devices such as magnetic and optical disk
- The transmission of digital data from a computer to a remote computer over telephone lines (i.e., using a modem). This is one of the major ways of sending and receiving information on the Internet.

Whenever information is transmitted from one device (the transmitter) to another device (the receiver), there is a possibility that errors can occur such that the receiver does not receive the identical information that was sent by the transmitter. The major cause of any transmission errors is *electrical noise,* which consists of spurious fluctuations in voltage or current that are present in all electronic systems to varying degrees. Figure 2-4 is a simple illustration of a type of transmission error.

The transmitter sends a relatively noise-free serial digital signal over a signal line to a receiver. However, by the time the signal reaches the receiver,

ENGR10003 – ENGINEERING
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

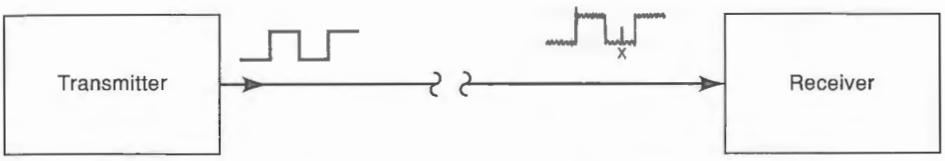


FIGURE 2-4 Example of noise causing an error in the transmission of digital data.

it contains a certain degree of noise superimposed on the original signal. Occasionally, the noise is large enough in amplitude that it will alter the logic level of the signal, as it does at point *x*. When this occurs, the receiver may incorrectly interpret that bit as a logic 1, which is not what the transmitter has sent.

Most modern digital equipment is designed to be relatively error-free, and the probability of errors such as the one shown in Figure 2-4 is very low. However, we must realize that digital systems often transmit thousands, even millions, of bits per second, so that even a very low rate of occurrence of errors can produce an occasional error that might prove to be bothersome, if not disastrous. For this reason, many digital systems employ some method for detection (and sometimes correction) of errors. One of the simplest and most widely used schemes for error detection is the **parity method**.

Parity Bit

A **parity bit** is an extra bit that is attached to a code group that is being transferred from one location to another. The parity bit is made either 0 or 1, depending on the number of 1s that are contained in the code group. Two different methods are used.

In the *even-parity* method, the value of the parity bit is chosen so that the total number of 1s in the code group (including the parity bit) is an *even* number. For example, suppose that the group is 1000011. This is the ASCII character "C." The code group has *three* 1s. Therefore, we will add a parity bit of 1 to make the total number of 1s an even number. The *new* code group, *including the parity bit,* thus becomes

1 1 0 0 0 0 1 1
↑ added parity bit*

If the code group contains an even number of 1s to begin with, the parity bit is given a value of 0. For example, if the code group were 1000001 (the ASCII code for "A"), the assigned parity bit would be 0, so that the new code, *including the parity bit,* would be 01000001.

The *odd-parity* method is used in exactly the same way except that the parity bit is chosen so the total number of 1s (including the parity bit) is an *odd* number. For example, for the code group 1000001, the assigned parity bit would be a 1. For the code group 1000011, the parity bit would be a 0.

Regardless of whether even parity or odd parity is used, the parity bit becomes an actual part of the code word. For example, adding a parity bit to

*The parity bit can be placed at either end of the code group, but it is usually placed to the left of the MSB.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications,* (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

the seven-bit ASCII code produces an eight-bit code. Thus, the parity bit is treated just like any other bit in the code.

The parity bit is issued to detect any *single-bit* errors that occur during the transmission of a code from one location to another. For example, suppose that the character "A" is being transmitted and *odd* parity is being used. The transmitted code would be

1 1 0 0 0 0 0 1

When the receiver circuit receives this code, it will check that the code contains an odd number of 1s (including the parity bit). If so, the receiver will assume that the code has been correctly received. Now, suppose that because of some noise or malfunction the receiver actually receives the following code:

1 1 0 0 0 0 0 0

The receiver will find that this code has an *even* number of 1s. This tells the receiver that there must be an error in the code because presumably the transmitter and receiver have agreed to use odd parity. There is no way, however, that the receiver can tell which bit is in error because it does not know what the code is supposed to be.

It should be apparent that this parity method would not work if *two* bits were in error, because two errors would not change the "oddness" or "evenness" of the number of 1s in the code. In practice, the parity method is used only in situations where the probability of a single error is very low and the probability of double errors is essentially zero.

When the parity method is being used, the transmitter and the receiver must have agreement, in advance, as to whether odd or even parity is being used. There is no advantage of one over the other, although even parity seems to be used more often. The transmitter must attach an appropriate parity bit to each unit of information that it transmits. For example, if the transmitter is sending ASCII-coded data, it will attach a parity bit to each seven-bit ASCII code group. When the receiver examines the data that it has received from the transmitter, it checks each code group to see that the total number of 1s (including the parity bit) is consistent with the agreed-upon type of parity. This is often called *checking the parity* of the data. In the event that it detects an error, the receiver may send a message back to the transmitter asking it to retransmit the last set of data. The exact procedure that is followed when an error is detected depends on the particular system.

EXAMPLE 2-17

Computers often communicate with other remote computers over telephone lines. For example, this is how dial-up communication over the internet takes place. When one computer is transmitting a message to another, the information is usually encoded in ASCII. What actual bit strings would a computer transmit to send the message HELLO, using ASCII with even parity?

Solution

First, look up the ASCII codes for each character in the message. Then for each code, count the number of 1s. If it is an even number, attach a 0 as the

MSB. If it is an odd number, attach a 1. Thus, the resulting eight-bit codes (bytes) will all have an even number of 1s (including parity).

	attached even-parity bits ↓	
H =	0	1 0 0 1 0 0 0
E =	1	1 0 0 0 1 0 1
L =	1	1 0 0 1 1 0 0
L =	1	1 0 0 1 1 0 0
O =	1	1 0 0 1 1 1 1

Error Correction

Error detection is beneficial because the system that receives a datum containing an error knows it has received "damaged goods." Wouldn't it be great if somehow the receiver could also know which bit was wrong? If a binary bit is wrong, then the correct value is simply its complement. Several methods have been developed to accomplish this. In each case, it requires that several bits of "error detection/correction codes" be applied to each transmitted packet of information. As the packet is received, a digital circuit can detect if errors have occurred (even multiple errors) and correct them. This technology is used for massive transfer of high speed data in such applications as magnetic disk drives, flash drives, CD, DVD, Blu-ray Disc, digital television, and broadband Internet networks.

REVIEW QUESTIONS

1. Attach an odd-parity bit to the ASCII code for the $ symbol, and express the result in hexadecimal.
2. Attach an even-parity bit to the BCD code for decimal 69.
3. Why can't the parity method detect a double error in transmitted data?

2-10 APPLICATIONS

Here are several applications that will serve as a review of some of the concepts covered in this chapter. These applications should give a sense of how the various number systems and codes are used in the digital world. More applications are presented in the end-of-chapter problems.

APPLICATION 2-1

A typical CD-ROM can store 650 megabytes of digital data. Since mega $= 2^{20}$, how many bits of data can a CD-ROM hold?

Solution

Remember that a byte is eight bits. Therefore, 650 megabytes is $650 \times 2^{20} \times 8 =$ **5,452,595,200 bits.**

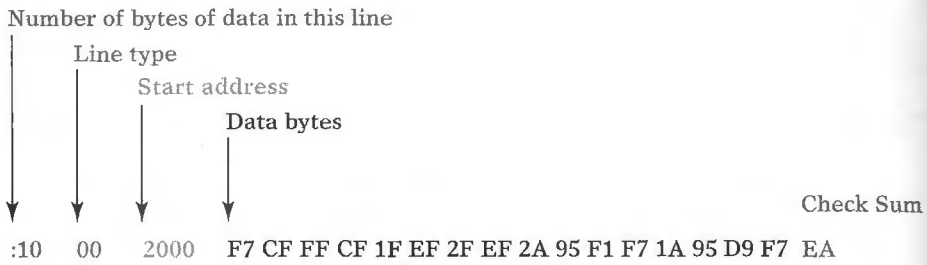
APPLICATION 2-2

In order to program many microcontrollers, the binary instructions are stored in a file on a personal computer in a special way known as Intel Hex

Format. The hexadecimal information is encoded into ASCII characters so it can be displayed easily on the PC screen, printed, and easily transmitted one character at a time over a standard PC's serial COM port. One line of an Intel Hex Format file is shown below:

:10002000F7CFFFCF1FEF2FEF2A95F1F71A95D9F7EA

Intel hex format:



The first character sent is the ASCII code for a colon, followed by a 1. Each has an even parity bit appended as the most significant bit. A test instrument captures the binary bit pattern as it goes across the cable to the microcontroller.

(a) What should the binary bit pattern (including parity) look like? (MSB – LSB)

(b) The value 10, following the colon, represents the total hexadecimal number of bytes that are to be loaded into the micro's memory. What is the decimal number of bytes being loaded?

(c) The number 0020 is a four-digit hex value representing the address where the first byte is to be stored. What is the biggest address possible? How many bits would it take to represent this address?

(d) The value of the first data byte is F7. What is the value (in binary) of the least significant nibble of this byte?

FFFF 1111 1111 1111 1111 16 bits

Solution

(a) ASCII codes are 3A (for :) and 31 (for 1) 00111010 10110001
even parity bit

(b) 10 hex = $1 \times 16 + 0 \times 1 = 16$ decimal bytes

(c) FFFF is the biggest possible value. Each hex digit is 4 bits, so we need 16 bits.

(d) The least significant nibble (4 bits) is represented by hex 7. In binary this would be 0111.

APPLICATION 2-3

A small process-control computer uses hexadecimal codes to represent its 16-bit memory addresses.

(a) How many hex digits are required?

(b) What is the range of addresses in hex?

(c) How many memory locations are there?

ENGR100 A publica... publishing exclusively for The University of Melbourne page 55

Solution

(a) Since 4 bits convert to a single hex digit, 16/4 = 4 hex digits are needed.

(b) The binary range is 0000000000000000_2 to 1111111111111111_2. In hex, this becomes 0000_{16} to $FFFF_{16}$.

(c) With 4 hex digits, the total number of addresses is 16^4 = **65,536.**

SECTION 2-4

Numbers are entered into a microcontroller-based system in BCD, but stored in straight binary. As a programmer, you must decide whether you need a one-byte or two-byte storage location.

(a) How many bytes do you need if the system takes a two-digit decimal entry?

(b) What if you needed to be able to enter three digits?

Solution

(a) With two digits you can enter values up to 99 ($1001\ 1001_{BCD}$). In binary this value is 01100011, which will fit into an eight-bit memory location. Thus you can use a single byte.

(b) Three digits can represent up to 999 (1001 1001 1001). In binary this value is 1111100111 (10 bits). Thus you cannot use a single byte; you need two bytes.

APPLICATION 2-5

When ASCII characters must be transmitted between two independent systems (such as between a computer and a modem), there must be a way of telling the receiver when a new character is coming in. There is often a need to detect errors in the transmission as well. The method of transfer is called asynchronous data communication. The normal resting state of the transmission line is logic 1. When the transmitter sends an ASCII character, it must be "framed" so the receiver knows where the data begins and ends. The first bit must always be a start bit (logic 0). Next the ASCII code is sent LSB first and MSB last. After the MSB, a parity bit is appended to check for transmission errors. Finally, the transmission is ended by sending a stop bit (logic 1). A typical asynchronous transmission of a seven-bit ASCII code for the pound sign # (23 Hex) with even parity is shown in Figure 2-5.

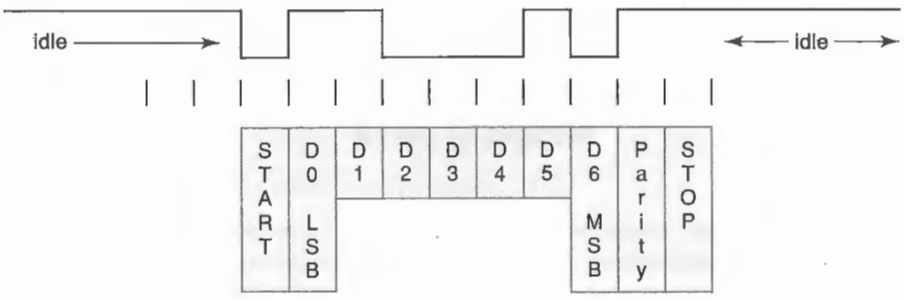


FIGURE 2-5
Asynchronous serial data with even parity.

APPLICATION 2-6

Your PC encounters an error when running an application. The dialog box reports information about the addresses that it could not read or write. What number system is used to report the address area?

Solution

These numbers will normally be reported in hexadecimal. Rather than using the subscript 16 as we have done in this text, other methods may be used to indicate hexadecimal (e.g., attaching a 0x prefix to the number).

SUMMARY

1. The hexadecimal number system is used in digital systems and computers as an efficient way of representing binary quantities.
2. In conversions between hex and binary, each hex digit corresponds to four bits.
3. The repeated-division method is used to convert decimal numbers to binary or hexadecimal.
4. Using an N-bit binary number, we can represent decimal values from 0 to $2^N - 1$.
5. The BCD code for a decimal number is formed by converting each digit of the decimal number to its four-bit binary equivalent.
6. The Gray code defines a sequence of bit patterns in which only one bit changes between successive patterns in the sequence.
7. A byte is a string of eight bits. A nibble is four bits. The word size depends on the system.
8. An alphanumeric code is one that uses groups of bits to represent all of the various characters and functions that are part of a typical computer's keyboard. The ASCII code is the most widely used alphanumeric code.
9. The parity method for error detection attaches a special parity bit to each transmitted group of bits.

IMPORTANT TERMS

hexadecimal number system
straight binary coding
binary-coded-decimal (BCD) code
Gray code
byte
nibble
word
word size
alphanumeric code
American Standard Code for Information Interchange (ASCII)
parity method
parity bit

PROBLEMS

SECTIONS 2-1 AND 2-2

2-1. Convert these binary numbers to decimal.

(a)* 10110	(e)* 11111111	(i)* 100110
(b) 10010101	(f) 01101111	(j) 1101
(c)* 100100001001	(g)* 1111010111	(k)* 111011
(d) 01101011	(h) 11011111	(l) 1010101

*Answers to problems marked with an asterisk can be found in the back of the text.

2-2. Convert the following decimal values to binary.

(a)*37	(e)*77	(i)* 511
(b) 13	(f) 390	(j) 25
(c)*189	(g)*205	(k) 52
(d) 1000	(h) 2133	(l) 47

2-3. What is the largest decimal value that can be represented by (a)* an eight-bit binary number? (b) A 16-bit number?

SECTION 2-4

2-4. Convert each hex number to its decimal equivalent.

(a)*743	(e)*165	(i) E71
(b) 36	(f) ABCD	(j) 89
(c)*37FD	(g)*7FF	(k) 58
(d) 2000	(h) 1204	(l) 72

2-5. Convert each of the following decimal numbers to hex.

(a)*59	(e)*771	(i) 29
(b) 372	(f) 2313	(j) 33
(c)*919	(g)*65,536	(k) 100
(d) 1024	(h) 255	(l) 200

2-6. Convert each of the hex values from Problem 2-4 to binary.

2-7. Convert the binary numbers in Problem 2-1 to hex.

2-8. List the hex numbers in sequence from 195_{16} to 180_{16}.

2-9.* When a large decimal number is to be converted to binary, it is sometimes easier to convert it first to hex, and then from hex to binary. Try this procedure for 2133_{10} and compare it with the procedure used in Problem 2-2(h).

2-10. How many hex digits are required to represent decimal numbers up to 20,000? up to 40,000?

2-11. Convert these hex values to decimal.

(a)*92	(e)*000F	(i) 19
(b) 1A6	(f) 55	(j) 42
(c)*37FD	(g)*2C0	(k) CA
(d) ABCD	(h) 7FF	(l) F1

2-12. Convert these decimal values to hex.

(a)*75	(e)*7245	(i) 95
(b) 314	(f) 498	(j) 89
(c)*2048	(g)*25,619	(k) 128
(d) 24	(h) 4095	(l) 256

2-13. Take each four-bit binary number in the order they are written and write the equivalent hex digit without performing a calculation by hand or by calculator.

(a) 1001	(e) 1111	(i) 1011	(m) 0001
(b) 1101	(f) 0010	(j) 1100	(n) 0101
(c) 1000	(g) 1010	(k) 0011	(o) 0111
(d) 0000	(h) 1001	(l) 0100	(p) 0110

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

2-14. Take each hex digit and write its four-bit binary value without performing any calculations by hand or by calculator.

(a) 6	(e) 4	(i) 9	(m) 0
(b) 7	(f) 3	(j) A	(n) 8
(c) 5	(g) C	(k) 2	(o) D
(d) 1	(h) B	(l) F	(p) 9

2-15. What is the largest value that can be represented by three hex digits?

2-16.* Convert the hex values in Problem 2-11 to binary.

2-17.* List the hex numbers in sequence from 280 to 2A0.

2-18. How many hex digits are required to represent decimal numbers up to 1 million? 4 million?

SECTION 2-4

2-19. Encode these decimal numbers in BCD.

(a)*47	(e)*13	(i)* 72
(b) 962	(f) 529	(j) 38
(c)*187	(g)*89,627	(k)*61
(d) 6727	(h) 1024	(l) 90

2-20. How many bits are required to represent the decimal numbers in the range from 0 to 999 using (a) straight binary code? (b) Using BCD code?

2-21. The following numbers are in BCD. Convert them to decimal.

(a)*1001011101010010	(f) 010101010101
(b) 000110000100	(g) 10111
(c)*011010010101	(h) 010110
(d) 0111011101110101	(i) 1110101
(e)*010010010010	

SECTION 2-7

2-22.*(a) How many bits are contained in eight bytes?

(b) What is the largest hex number that can be represented in four bytes?

(c) What is the largest BCD-encoded decimal value that can be represented in three bytes?

2-23. (a) Refer to Table 2-4. What is the most significant nibble of the ASCII code for the letter X?

(b) How many nibbles can be stored in a 16-bit word?

(c) How many bytes does it take to make up a 24-bit word?

SECTIONS 2-8 AND 2-9

2-24. Represent the statement "X = 3 × Y" in ASCII code. Attach an odd-parity bit.

2-25.* Attach an *even*-parity bit to each of the ASCII codes for Problem 2-24, and give the results in hex.

2-26. The following bytes (shown in hex) represent a person's name as it would be stored in a computer's memory. Each byte is a padded ASCII code. Determine the name of each person.

(a)*42 45 4E 20 53 4D 49 54 48

(b) 4A 6F 65 20 47 72 65 65 6E

ENGR10003 – ENGINEERING SCIENCE
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

2-27. Convert the following decimal numbers to BCD code and then attach an *odd* parity bit.

(a)*74 (c)*8884 (e)*165 (g) 11
(b) 38 (d) 275 (f) 9201 (h) 51

2-28.*In a certain digital system, the decimal numbers from 000 through 999 are represented in BCD code. An *odd*-parity bit is also included at the end of each code group. Examine each of the code groups below, and assume that each one has just been transferred from one location to another. Some of the groups contain errors. Assume that *no more than* two errors have occurred for each group. Determine which of the code groups have a single error and which of them *definitely* have a double error. (*Hint:* Remember that this is a BCD code.)

(a) 1001010110000
msb lsb parity bit
(b) 0100011101100
(c) 0111110000011
(d) 1000011000101

2-29. Suppose that the receiver received the following data from the transmitter of Example 2-17:

0 1 0 0 1 0 0 0
1 1 0 0 0 1 0 1
1 1 0 0 1 1 0 0
1 1 0 0 1 0 0 0
1 1 0 0 1 1 0 0

What errors can the receiver determine in these received data?

DRILL QUESTIONS

2-30.*Perform each of the following conversions. For some of them, you may want to try several methods to see which one works best for you. For example, a binary-to-decimal conversion may be done directly, or it may be done as a binary-to-hex conversion followed by a hex-to-decimal conversion.

(a) 1417_{10} = _____ $_2$
(b) 255_{10} = _____ $_2$
(c) 11010001_2 = _____ $_{10}$
(d) 1110101000100111_2 = _____ $_{10}$
(e) 2497_{10} = _____ $_{16}$
(f) 511_{10} = _____ (BCD)
(g) 235_{16} = _____ $_{10}$
(h) 4316_{10} = _____ $_{16}$
(i) $7A9_{16}$ = _____ $_{10}$
(j) $3E1C_{16}$ = _____ $_{10}$
(k) 1600_{10} = _____ $_{16}$
(l) $38{,}187_{10}$ = _____ $_{16}$
(m) 865_{10} = _____ (BCD)
(n) 100101000111 (BCD) = _____ $_{10}$

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Number Systems and Codes. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 32-63). Upper Saddle River, NJ: Pearson Education Inc.

(o) 465_{16} = _____ 2

(p) $B34_{16}$ = _____ 2

(q) 01110100 (BCD) = _____ 2

(r) 111010_2 = _____ (BCD)

2-31.*Represent the decimal value 37 in each of the following ways.

(a) straight binary

(b) BCD

(c) hex

(d) ASCII (i.e., treat each digit as a character)

2-32.*Fill in the blanks with the correct word or words.

(a) Conversion from decimal to _____ requires repeated division by 16.

(b) Conversion from decimal to binary requires repeated division by _____.

(c) In the BCD code, each _____ is converted to its four-bit binary equivalent.

(d) The _____ code has the characteristic that only one bit changes in going from one step to the next.

(e) A transmitter attaches a _____ to a code group to allow the receiver to detect _____.

(f) The _____ code is the most common alphanumeric code used in computer systems.

(g) _____ is often used as a convenient way to represent large binary numbers.

(h) A string of eight bits is called a _____.

2-33. Write the binary number that results when each of the following numbers is incremented by one.

(a)* 0111 (b) 010011 (c) 1011 (d) 1111

2-34. Decrement each binary number by one.

(a)* 1100 (b) 101000 (c) 1110 (d) 1001 0000

2-35. Write the number that results when each of the following is incremented.

(a)* 7779_{16} (c)* $OFFF_{16}$ (e)* $9FF_{16}$ (g) F_{16}

(b) 9999_{16} (d) 2000_{16} (f) $100A_{16}$ (h) FE_{16}

2-36.*Repeat Problem 2-35 for the decrement operation.

CHALLENGING EXERCISES

2-37.*In a microcomputer, the *addresses* of memory locations are binary numbers that identify each memory circuit where a byte is stored. The number of bits that make up an address depends on how many memory locations there are. Since the number of bits can be very large, the addresses are often specified in hex instead of binary.

(a) If a microcomputer uses a 20-bit address, how many different memory locations are there?

(b) How many hex digits are needed to represent the address of a memory location?

(c) What is the hex address of the 256th memory location? (*Note:* The first address is always 0.)

(d) The computer program is stored in the lowest 2 kbyte block of memory. Give the start and end address of this block.

2-38. In an audio CD, the audio voltage signal is typically sampled about 44,000 times per second, and the value of each sample is recorded on the CD surface as a binary number. In other words, each recorded binary number represents a single voltage point on the audio signal waveform.

(a) If the binary numbers are six bits in length, how many different voltage values can be represented by a single binary number? Repeat for eight bits and ten bits.

(b) If ten-bit numbers are used, how many bits will be recorded on the CD in 1 second?

(c) If a CD can typically store 5 billion bits, how many seconds of audio can be recorded when ten-bit numbers are used?

2-39.*A black-and-white digital camera lays a fine grid over an image and then measures and records a binary number representing the level of gray it sees in each cell of the grid. For example, if four-bit numbers are used, the value of black is set to 0000 and the value of white to 1111, and any level of gray is somewhere between 0000 and 1111. If six-bit numbers are used, black is 000000, white is 111111, and all grays are between the two.

Suppose we wanted to distinguish among 254 different levels of gray within each cell of the grid. How many bits would we need to use to represent these levels?

2-40. A 3-Megapixel digital camera stores an eight-bit number for the brightness of each of the primary colors (red, green, blue) found in each picture element (pixel). If every bit is stored (no data compression), how many pictures can be stored on a 128-Megabyte memory card? (Note: In digital systems, Mega means 2^{20}.)

2-41. Construct a table showing the binary, hex, and BCD representations of all decimal numbers from 0 to 15. Compare your table with Table 2-3.

ANSWERS TO SECTION REVIEW QUESTIONS

SECTION 2-1

1. 2267 2. 32768 3. 2267

SECTION 2-2

1. 1010011 2. 1011011001 3. 20 bits

SECTION 2-3

1. 9422 2. C2D; 110000101101 3. 97B5 4. E9E, E9F, EA0, EA1
5. 11010100100111 6. 0 to 65,535

SECTION 2-4

1. 10110010_2; 000101111000 (BCD) 2. 32 3. Advantage: ease of conversion. Disadvantage: BCD requires more bits.

SECTION 2-5

1. 0111 2. 0110

SECTION 2-7

1. One 2. 9999 3. One 4. One

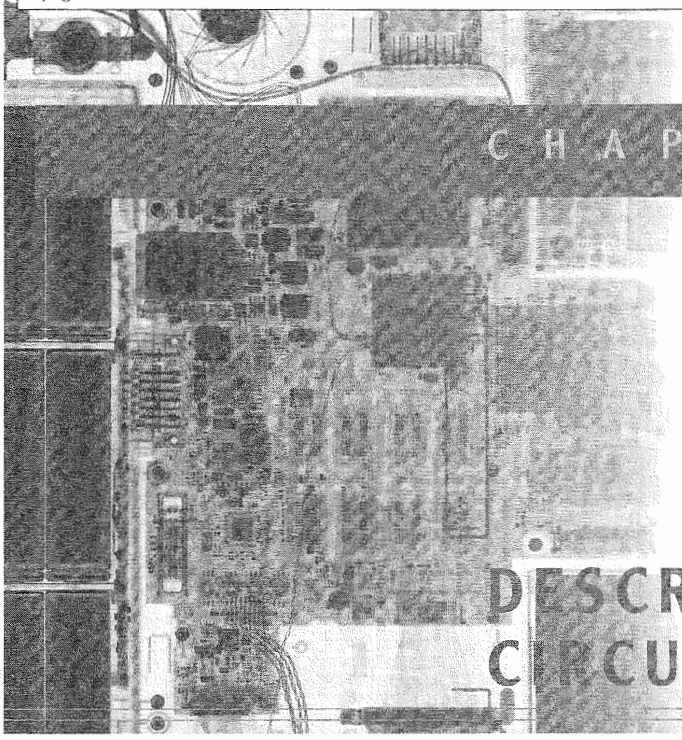
SECTION 2-8

1. 43, 4F, 53, 54, 20, 3D, 20, 24, 37, 32 2. STOP

SECTION 2-9

1. A4 2. 001101001 3. Two errors in the data would not change the oddness or evenness of the number of 1s in the data.

This page is intentionally blank.



CHAPTER 3

DESCRIBING LOGIC CIRCUITS

OUTLINE

3-1 Boolean Constants and Variables
3-2 Truth Tables
3-3 OR Operation with OR Gates
3-4 AND Operation with AND Gates
3-5 NOT Operation
3-6 Describing Logic Circuits Algebraically
3-7 Evaluating Logic-Circuit Outputs
3-8 Implementing Circuits from Boolean Expressions
3-9 NOR Gates and NAND Gates
3-10 Boolean Theorems
3-11 DeMorgan's Theorems
3-12 Universality of NAND Gates and NOR Gates
3-13 Alternate Logic-Gate Representations
3-14 Which Gate Representation to Use
3-15 Propagation Delay
3-16 Summary of Methods to Describe Logic Circuits
3-17 Description Languages Versus Programming Languages
3-18 Implementing Logic Circuits with PLDs
3-19 HDL Format and Syntax
3-20 Intermediate Signals

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Describing Logic Circuits. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 64-127). Upper Saddle River, NJ: Pearson Education Inc.

OBJECTIVES

Upon completion of this chapter, you will be able to:

- Perform the three basic logic operations.
- Describe the operation of and construct the truth tables for the AND, NAND, OR, and NOR gates, and the NOT (INVERTER) circuit.
- Draw timing diagrams for the various logic-circuit gates.
- Write the Boolean expression for the logic gates and combinations of logic gates.
- Implement logic circuits using basic AND, OR, and NOT gates.
- Use Boolean algebra to simplify complex logic circuits.
- Use DeMorgan's theorems to simplify logic expressions.
- Use either of the universal gates (NAND or NOR) to implement a circuit represented by a Boolean expression.
- Explain the advantages of constructing a logic-circuit diagram using the alternate gate symbols versus the standard logic-gate symbols.
- Describe the concept of active-LOW and active-HIGH logic signals.
- Describe and measure propagation delay time.
- Use several methods to describe the operation of logic circuits.
- Interpret simple circuits defined by a hardware description language (HDL).
- Explain the difference between an HDL and a computer programming language.
- Create an HDL file for a simple logic gate.
- Create an HDL file for combinational circuits with intermediate variables.

INTRODUCTION

Chapters 1 and 2 introduced the concepts of logic levels and logic circuits. In logic, only two possible conditions exist for any input or output: true and false. The binary number system uses only two digits, 1 and 0, so it is perfect for representing logical relationships. Digital logic circuits use predefined voltage ranges to represent these binary states. Using these concepts, we can create circuits made of little more than processed beach sand and wire that make consistent, intelligent, logical decisions. It is vitally important that we have a method to describe the logical decisions made by these circuits. In other words, we must describe how they operate. In this chapter, we will discover many ways to describe their operation. Each description method is important because all these methods commonly appear in

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Describing Logic Circuits. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 64-127). Upper Saddle River, NJ: Pearson Education Inc.

technical literature and system documentation and are used in conjunction with modern design and development tools.

Life is full of examples of circumstances that are in one state or another. For example, a creature is either alive or dead, a light is either on or off, a door is locked or unlocked, and it is either raining or it is not. In 1854, a mathematician named George Boole wrote *An Investigation of the Laws of Thought,* in which he described the way we make logical decisions based on true or false circumstances. The methods he described are referred to today as Boolean logic, and the system of using symbols and operators to describe these decisions is called Boolean algebra. In the same way we use symbols such as x and y to represent unknown numerical values in regular algebra, Boolean algebra uses symbols to represent a logical expression that has one of two possible values: true or false. The logical expression might be *door is closed, button is pressed,* or *fuel is low.* Writing these expressions is very tedious, and so we tend to substitute symbols such as A, B, and C.

The main purpose of these logical expressions is to describe the relationship between a logic circuit's output (the decision) and its inputs (the circumstances). In this chapter, we will study the most basic logic circuits—*logic gates*—which are the fundamental building blocks from which all other logic circuits and digital systems are constructed. We will see how the operation of the different logic gates and the more complex circuits formed from combinations of logic gates can be described and analyzed using Boolean algebra. We will also get a glimpse of how Boolean algebra can be used to simplify a circuit's Boolean expression so that the circuit can be rebuilt using fewer logic gates and/or fewer connections. Much more will be done with circuit simplification in Chapter 4.

Boolean algebra is not only used as a tool for analysis and simplification of logic systems. It can also be used as a tool to create a logic circuit that will produce the desired input/output relationship. This process is often called synthesis of logic circuits as opposed to analysis. Other techniques have been used in the analysis, synthesis, and documentation of logic systems and circuits including truth tables, schematic symbols, timing diagrams, and—last but by no means least—language. To categorize these methods, we could say that Boolean algebra is a mathematic tool, truth tables are data organizational tools, schematic symbols are drawing tools, timing diagrams are graphing tools, and language is the universal description tool.

Today, any of these tools can be used to provide input to computers. The computers can be used to simplify and translate between these various forms of description and ultimately provide an output in the form necessary to implement a digital system. To take advantage of the powerful benefits of computer software, we must first fully understand the acceptable ways for describing these systems in terms the computer can understand. This chapter will lay the groundwork for further study of these vital tools for synthesis and analysis of digital systems.

Clearly the tools described here are invaluable tools in describing, analyzing, designing, and implementing digital circuits. The student who expects to work in the digital field must work hard at understanding and becoming comfortable with Boolean algebra (believe us, it's much, much easier than conventional algebra) and all the other tools. Do *all* of the examples, exercises, and problems, even the ones your instructor doesn't assign. When those run out, make up your own. The time you spend will be well worth it because you will see your skills improve and your confidence grow.

3-1 BOOLEAN CONSTANTS AND VARIABLES

Boolean algebra differs in a major way from ordinary algebra because Boolean constants and variables are allowed to have only two possible values, 0 or 1. A Boolean variable is a quantity that may, at different times, be equal to either 0 or 1. Boolean variables are often used to represent the voltage level present on a wire or at the input/output terminals of a circuit. For example, in a certain digital system, the Boolean value of 0 might be assigned to any voltage in the range from 0 to 0.8 V, while the Boolean value of 1 might be assigned to any voltage in the range 2 to 5 V.*

Thus, Boolean 0 and 1 do not represent actual numbers but instead represent the state of a voltage variable, or what is called its **logic level**. A voltage in a digital circuit is said to be at the logic 0 level or the logic 1 level, depending on its actual numerical value. In digital logic, several other terms are used synonymously with 0 and 1. Some of the more common ones are shown in Table 3-1. We will use the 0/1 and LOW/HIGH designations most of the time.

TABLE 3-1

Logic 0	Logic 1
False	True
Off	On
LOW	HIGH
No	Yes
Open switch	Closed switch

As we said in the introduction, **Boolean algebra** is a means for expressing the relationship between a logic circuit's inputs and outputs. The inputs are considered logic variables whose logic levels at any time determine the output levels. In all our work to follow, we shall use letter symbols to represent logic variables. For example, the letter A might represent a certain digital circuit input or output, and at any time we must have either $A = 0$ or $A = 1$: if not one, then the other.

Because only two values are possible, Boolean algebra is relatively easy to work with compared with ordinary algebra. In Boolean algebra, there are no fractions, decimals, negative numbers, square roots, cube roots, logarithms, imaginary numbers, and so on. In fact, in Boolean algebra there are only *three* basic operations: *OR, AND,* and *NOT.*

These basic operations are called *logic operations.* Digital circuits called *logic gates* can be constructed from diodes, transistors, and resistors connected so that the circuit output is the result of a basic logic operation *(OR, AND, NOT)* performed on the inputs. We will be using Boolean algebra first to describe and analyze these basic logic gates, then later to analyze and design combinations of logic gates connected as logic circuits.

3-2 TRUTH TABLES

A **truth table** is a means for describing how a logic circuit's output depends on the logic levels present at the circuit's inputs. Figure 3-1(a) illustrates a truth table for one type of two-input logic circuit. The table lists all possible

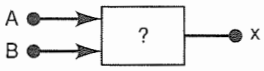
*Voltages between 0.8 and 2 V are undefined (neither 0 nor 1) and should not occur under normal circumstances.

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Describing Logic Circuits. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 64-127). Upper Saddle River, NJ: Pearson Education Inc.

FIGURE 3-1 Example truth tables for (a) two-input, (b) three-input, and (c) four-input circuits.

Inputs → A, B; Output → x

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



(a)

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

combinations of logic levels present at inputs A and B, along with the corresponding output level x. The first entry in the table shows that when A and B are both at the 0 level, the output x is at the 1 level or, equivalently, in the 1 state. The second entry shows that when input B is changed to the 1 state, so that $A = 0$ and $B = 1$, the output x becomes a 0. In a similar way, the table shows what happens to the output state for any set of input conditions.

Figures 3-1(b) and (c) show samples of truth tables for three- and four-input logic circuits. Again, each table lists all possible combinations of input logic levels on the left, with the resultant logic level for output x on the right. Of course, the actual values for x will depend on the type of logic circuit.

Note that there are 4 table entries for the two-input truth table, 8 entries for a three-input truth table, and 16 entries for the four-input truth table. The number of input combinations will equal 2^N for an N-input truth table. Also note that the list of all possible input combinations follows the binary counting sequence, and so it is an easy matter to write down all of the combinations without missing any.

REVIEW QUESTIONS

1. What is the output state of the four-input circuit represented in Figure 3-1(c) when all inputs except B are 1?
2. Repeat question 1 for the following input conditions: $A = 1, B = 0, C = 1, D = 0$.
3. How many table entries are needed for a five-input circuit?

3-3 OR OPERATION WITH OR GATES

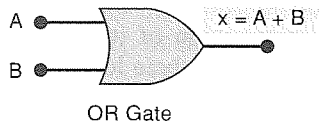
The **OR operation** is the first of the three basic Boolean operations to be learned. An example can be found in the kitchen oven. The light inside the oven should turn on if either the *oven light switch is on* OR if the *door is opened*. The letter A could be used to represent the *oven light switch is on* (true or false) and B could represent *door is opened* (true or false). The letter x could represent the *light is on* (true or false). The truth table in Figure 3-2(a) shows what happens when two logic inputs, A and B, are combined using the OR operation to produce the output x. The table shows that x is a logic 1 for every

FIGURE 3-2 (a) Truth table defining the OR operation; (b) circuit symbol for a two-input OR gate.

OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

(a)



(b)

combination of input levels where one *or* more inputs are 1. The only case where x is a 0 is when both inputs are 0.

The Boolean expression for the OR operation is

$$x = A + B$$

In this expression, the + sign does not stand for ordinary addition; it stands for the OR operation. The OR operation is similar to ordinary addition except for the case where A and B are both 1; the OR operation produces $1 + 1 = 1$, not $1 + 1 = 2$. In Boolean algebra, 1 is as high as we go, so we can never have a result greater than 1. The same holds true for combining three inputs using the OR operation. Here we have $x = A + B + C$. If we consider the case where all three inputs are 1, we have

$$x = 1 + 1 + 1 = 1$$

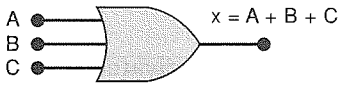
The expression $x = A + B$ is read as "x equals A OR B," which means that x will be 1 when A or B or both are 1. Likewise, the expression $x = A + B + C$ is read as "x equals A OR B OR C," which means that x will be 1 when A or B or C or any combination of them are 1. To describe this circuit in the English language we could say that *x is true (1) WHEN A is true (1) OR B is true (1) OR C is true (1).*

OR Gate

In digital circuitry, an **OR gate*** is a circuit that has two or more inputs and whose output is equal to the OR combination of the inputs. Figure 3-2(b) is the logic symbol for a two-input OR gate. The inputs A and B are logic voltage levels, and the output x is a logic voltage level whose value is the result of the OR operation on A and B; that is, $x = A + B$. In other words, the OR gate operates so that its output is HIGH (logic 1) if either input *A or B or both* are at a logic 1 level. The OR gate output will be LOW (logic 0) only if all its inputs are at logic 0.

This same idea can be extended to more than two inputs. Figure 3-3 shows a three-input OR gate and its truth table. Examination of this truth table shows again that the output will be 1 for every case where one or more inputs are 1. This general principle is the same for OR gates with any number of inputs.

FIGURE 3-3 Symbol and truth table for a three-input OR gate.



A	B	C	x = A + B + C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

*The term *gate* comes from the inhibit/enable operation discussed in Chapter 4.

Using the language of Boolean algebra, the output x can be expressed as $x = A + B + C$, where again it must be emphasized that the + represents the OR operation. The output of any OR gate, then, can be expressed as the OR combination of its various inputs. We will put this to use when we subsequently analyze logic circuits.

Summary of the OR Operation

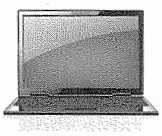
The important points to remember concerning the OR operation and OR gates are:

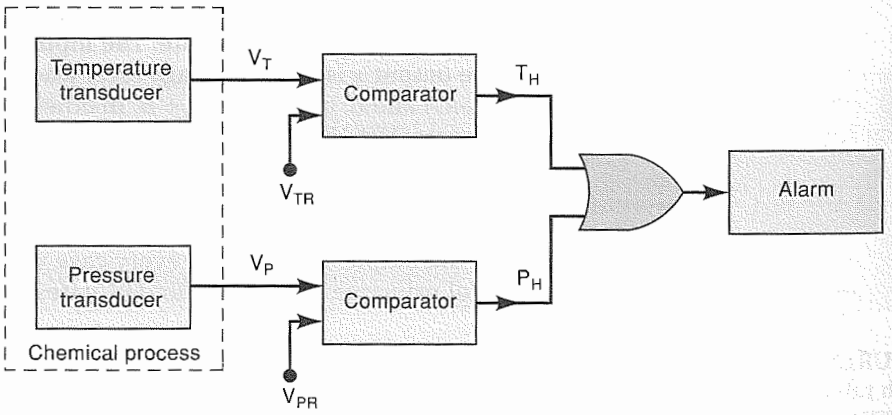
1. The OR operation produces a result (output) of 1 whenever *any* input is a 1. Otherwise the output is 0.
2. An OR gate is a logic circuit that performs an OR operation on the circuit's inputs.
3. The expression $x = A + B$ is read as "x equals A OR B."

EXAMPLE 3-1

In many industrial control systems, it is required to activate an output function whenever any one of several inputs is activated. For example, in a chemical process it may be desired that an alarm be activated whenever the process temperature exceeds a maximum value *or* whenever the pressure goes above a certain limit. Figure 3-4 is a block diagram of this situation. The temperature transducer circuit produces an output voltage proportional to the process temperature. This voltage, V_T, is compared with a temperature reference voltage, V_{TR}, in a voltage comparator circuit. The comparator output, T_H, is normally a low voltage (logic 0), but it switches to a high voltage (logic 1) when V_T exceeds V_{TR}, indicating that the process temperature is too high. A similar arrangement is used for the pressure measurement, so that its associated comparator output, P_H, goes from LOW to HIGH when the pressure is too high. What is the purpose of the OR gate?

FIGURE 3-4 Example of the use of an OR gate in an alarm system.





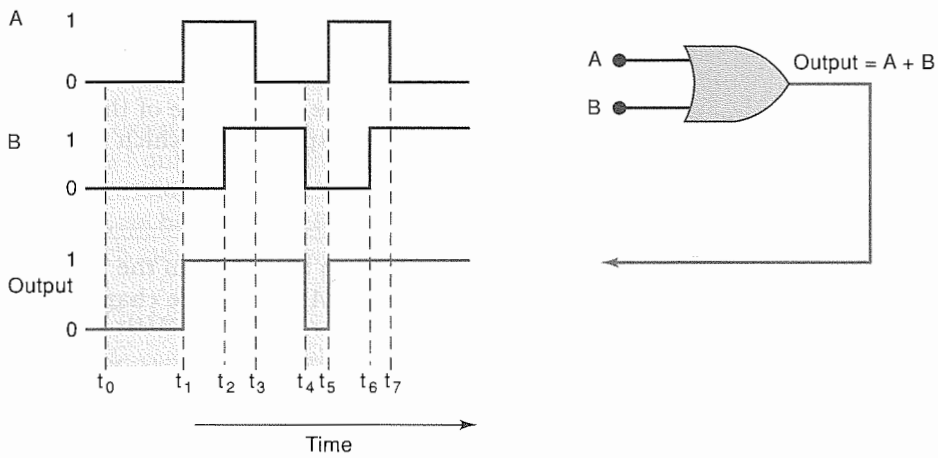
Solution

Since we want the alarm to be activated when either temperature *or* pressure is too high, it should be apparent that the two comparator outputs can be fed to a two-input OR gate. The OR gate output thus goes HIGH (1) for either alarm condition and will activate the alarm. This same idea can obviously be extended to situations with more than two process variables.

EXAMPLE 3-2

Determine the OR gate output in Figure 3-5. The OR gate inputs *A* and *B* are varying according to the timing diagrams shown. For example, *A* starts out LOW at time t_0, goes HIGH at t_1, back to LOW at t_3, and so on.

FIGURE 3-5 Example 3-2.



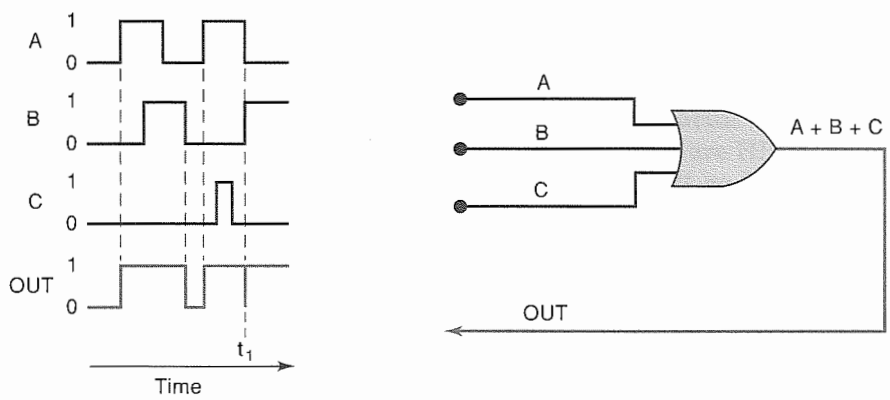
Solution

The OR gate output will be HIGH whenever *any* input is HIGH. Between time t_0 and t_1, both inputs are LOW, so OUTPUT = LOW. At t_1, input *A* goes HIGH while *B* remains LOW. This causes OUTPUT to go HIGH at t_1 and stay HIGH until t_4 because, during this interval, one or both inputs are HIGH. At t_4, input *B* goes from 1 to 0 so that now both inputs are LOW, and this drives OUTPUT back to LOW. At t_5, *A* goes HIGH, sending OUTPUT back HIGH, where it stays for the rest of the shown time span.

EXAMPLE 3-3A

For the situation depicted in Figure 3-6, determine the waveform at the OR gate output.

FIGURE 3-6 Examples 3-3A and B.



Solution

The three OR gate inputs *A*, *B*, and *C* are varying, as shown by their waveform diagrams. The OR gate output is determined by realizing that it will be

HIGH whenever *any* of the three inputs is at a HIGH level. Using this reasoning, the OR output waveform is as shown in the figure. Particular attention should be paid to what occurs at time t_1. The diagram shows that, at that instant of time, input A is going from HIGH to LOW while input B is going from LOW to HIGH. Since these inputs are making their transitions at approximately the same time, and since these transitions take a certain amount of time, there is a short interval when these OR gate inputs are both in the undefined range between 0 and 1. When this occurs, the OR gate output also becomes a value in this range, as evidenced by the glitch or spike on the output waveform at t_1. The occurrence of this glitch and its size (amplitude and width) depend on the speed with which the input transitions occur.

EXAMPLE 3-3B

What would happen to the glitch in the output in Figure 3-6 if input C sat in the HIGH state while A and B were changing at time t_1?

Solution

With the C input HIGH at t_1, the OR gate output will remain HIGH, regardless of what is occurring at the other inputs, because any HIGH input will keep an OR gate output HIGH. Therefore, the glitch will not appear in the output.

REVIEW QUESTIONS

1. What is the only set of input conditions that will produce a LOW output for any OR gate?
2. Write the Boolean expression for a six-input OR gate.
3. If the A input in Figure 3-6 is permanently kept at the 1 level, what will the resultant output waveform be?

3-4 AND OPERATION WITH AND GATES

The **AND operation** is the second basic Boolean operation. As an example of the use of AND logic, consider a typical clothes dryer. It is drying clothes (heating, tumbling) only if the *timer is set above zero* AND the *door is closed*. Let's assign A to represent *timer is set* (T/F), B to represent *door is closed* (T/F), and x can represent the *heater and motor are on* (T/F). The truth table in Figure 3-7(a) shows what happens when two logic inputs, A and B, are combined using the AND operation to produce output x. The table shows that x is a logic 1 only when both A and B are at the logic 1 level. For any case where one of the inputs is 0, the output is 0.

The Boolean expression for the AND operation is

$$x = A \cdot B$$

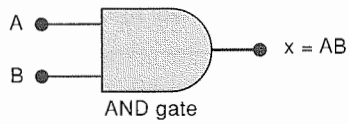
In this expression, the $\cdot$ sign stands for the Boolean AND operation and not the multiplication operation. However, the AND operation on Boolean variables operates the same as ordinary multiplication, as examination of the truth table shows, so we can think of them as being the same. This characteristic can be helpful when evaluating logic expressions that contain AND operations.

FIGURE 3-7 (a) Truth table for the AND operation; (b) AND gate symbol.

AND

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)

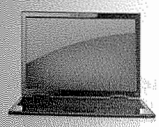
The expression $x = A \cdot B$ is read as "x equals A AND B," which means that x will be 1 only when A and B are both 1. The $\cdot$ sign is usually omitted so that the expression simply becomes $x = AB$. For the case when three inputs are ANDed, we have $x = A \cdot B \cdot C = ABC$. This is read as "$x$ equals A AND B AND C," which means that x will be 1 only when A and B and C are all 1.

AND Gate

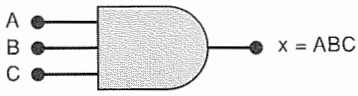
The logic symbol for a two-input **AND gate** is shown in Figure 3-7(b). The AND gate output is equal to the AND product of the logic inputs; that is, $x = AB$. In other words, the AND gate is a circuit that operates so that its output is HIGH only when all its inputs are HIGH. For all other cases, the AND gate output is LOW.

This same operation is characteristic of AND gates with more than two inputs. For example, a three-input AND gate and its accompanying truth table are shown in Figure 3-8. Once again, note that the gate output is 1 only for the case where $A = B = C = 1$. The expression for the output is $x = ABC$. For a four-input AND gate, the output is $x = ABCD$, and so on.

FIGURE 3-8 Truth table and symbol for a three-input AND gate.



A	B	C	x = ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



Note the difference between the symbols for the AND gate and the OR gate. Whenever you see the AND symbol on a logic-circuit diagram, it tells you that the output will go HIGH *only* when *all* inputs are HIGH. Whenever you see the OR symbol, it means that the output will go HIGH when *any* input is HIGH.

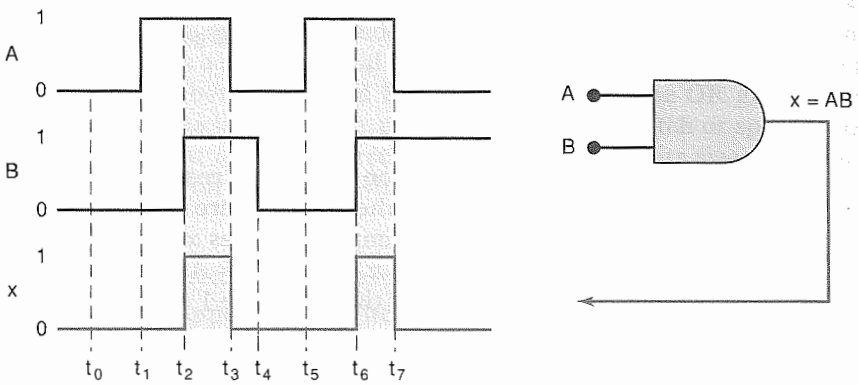
Summary of the AND Operation

1. The AND operation is performed the same as ordinary multiplication of 1s and 0s.
2. An AND gate is a logic circuit that performs the AND operation on the circuit's inputs.
3. An AND gate output will be 1 *only* for the case when *all* inputs are 1; for all other cases, the output will be 0.
4. The expression $x = AB$ is read as "x equals A AND B."

EXAMPLE 3-4

Determine the output x from the AND gate in Figure 3-9 for the given input waveforms.

FIGURE 3-9 Example 3-4.



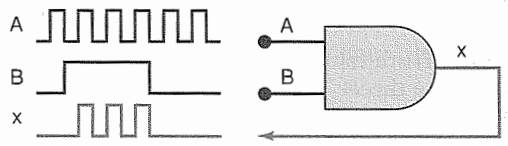
Solution

The output of an AND gate is determined by realizing that it will be HIGH only when all inputs are HIGH at the same time. For the input waveforms given, this condition is met only during intervals $t_2 - t_3$ and $t_6 - t_7$. At all other times, one or more of the inputs are 0, thereby producing a LOW output. Note that input level changes that occur while the other input is LOW have no effect on the output.

EXAMPLE 3-5A

Determine the output waveform for the AND gate shown in Figure 3-10.

FIGURE 3-10 Examples 3-5A and B.



Solution

The output x will be at 1 only when A and B are both HIGH at the same time. Using this fact, we can determine the x waveform as shown in the figure.

Notice that the x waveform is 0 whenever B is 0, regardless of the signal at A. Also notice that whenever B is 1, the x waveform is the same as A. Thus, we can think of the B input as a *control* input whose logic level determines whether or not the A waveform gets through to the x output. In this situation, the AND gate is used as an *inhibit circuit*. We can say that $B = 0$ is the inhibit condition producing a 0 output. Conversely, $B = 1$ is the *enable* condition, which enables A to reach the output. This inhibit operation is an important application of AND gates, which will be encountered later.

EXAMPLE 3-5B

What will happen to the x output waveform in Figure 3-10 if the B input is kept at the 0 level?

Solution

With B kept LOW, the x output will also stay LOW. This can be reasoned in two different ways. First, with $B = 0$ we have $x = A \cdot B = A \cdot 0 = 0$ because

anything multiplied (ANDed) by 0 will be 0. Another way to look at it is that an AND gate requires that all inputs be HIGH for the output to be HIGH, and this cannot happen if B is kept LOW.

REVIEW QUESTIONS

1. What is the only input combination that will produce a HIGH at the output of a five-input AND gate?
2. What logic level should be applied to the second input of a two-input AND gate if the logic signal at the first input is to be inhibited (prevented) from reaching the output?
3. *True or false:* An AND gate output will always differ from an OR gate output for the same input conditions.

3-5 NOT OPERATION

The **NOT operation** is unlike the OR and AND operations because it can be performed on a single input variable. For example, if the variable A is subjected to the NOT operation, the result x can be expressed as

$$x = \overline{A}$$

where the overbar represents the NOT operation. This expression is read as "x equals NOT A" or "x equals the *inverse* of A" or "x equals the *complement* of A." Each of these is in common usage, and all indicate that the logic value of $x = \overline{A}$ *is opposite to* the logic value of A. The truth table in Figure 3-11(a) clarifies this for the two cases $A = 0$ and $A = 1$. That is,

$$0 = \overline{1} \quad \text{because 0 is not 1}$$

and

$$1 = \overline{0} \quad \text{because 1 is not 0}$$

The NOT operation is also referred to as **inversion** or **complementation**, and these terms will be used interchangeably throughout the book. Although we will always use the overbar indicator to represent inversion, it is important to mention that another indicator for inversion is the prime symbol (′). That is,

$$A' = \overline{A}$$

Both should be recognized as indicating the inversion operation.

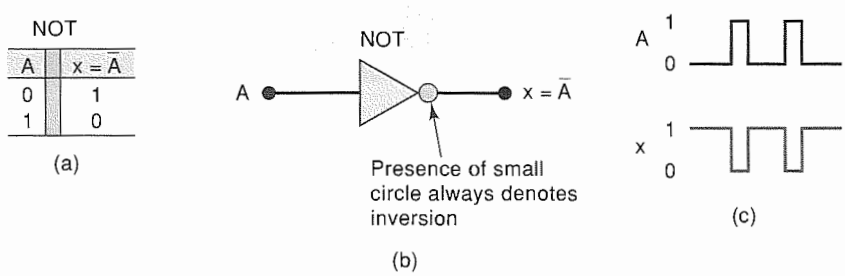


FIGURE 3-11 (a) Truth table; (b) symbol for the INVERTER (NOT circuit); (c) sample waveforms.

NOT Circuit (INVERTER)

Figure 3-11(b) shows the symbol for a **NOT circuit**, which is more commonly called an **INVERTER**. This circuit *always* has only a single input, and its output logic level is always opposite to the logic level of this input. Figure 3-11(c) shows how the INVERTER affects an input signal. It inverts (complements) the input signal at all points on the waveform so that whenever the input = 0, output = 1, and vice versa.

APPLICATION 3-1

Figure 3-12 shows a typical application of the NOT gate. The push button is wired to produce a logic 1 (true) when it is pressed. Sometimes we want to know if the push button is not being pressed, and so this circuit provides an expression that is true when the button is not pressed.

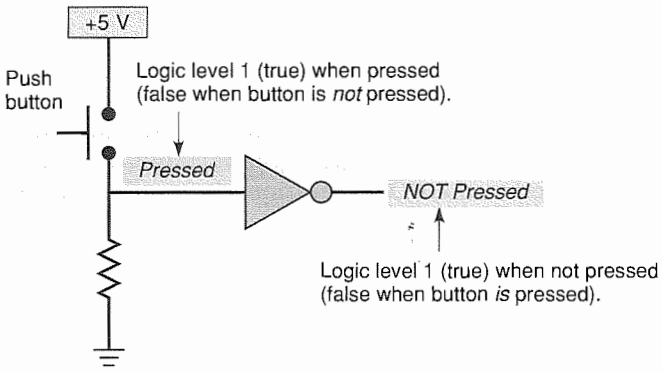


FIGURE 3-12 A NOT gate indicating a button is *not* pressed when its output is true.

Summary of Boolean Operations

The rules for the OR, AND, and NOT operations may be summarized as follows:

OR	*AND*	*NOT*
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\overline{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\overline{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

REVIEW QUESTIONS

1. The output of the INVERTER of Figure 3-11 is connected to the input of a second INVERTER. Determine the output level of the second INVERTER for each level of input A.
2. The output of the AND gate in Figure 3-7 is connected to the input of an INVERTER. Write the truth table showing the INVERTER output, y, for each combination of inputs A and B.

3-6 DESCRIBING LOGIC CIRCUITS ALGEBRAICALLY

Any logic circuit, no matter how complex, can be described completely using the three basic Boolean operations because the OR gate, AND gate, and NOT circuit are the basic building blocks of digital systems. For example, consider

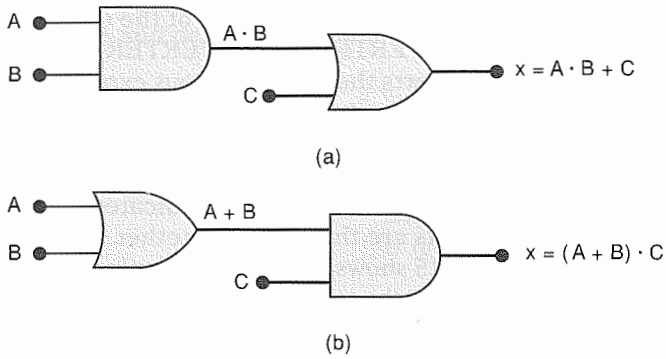


FIGURE 3-13 (a) Logic circuit with its Boolean expression; (b) logic circuit whose expression requires parentheses.

the circuit in Figure 3-13(a). This circuit has three inputs, A, B, and C, and a single output, x. Utilizing the Boolean expression for each gate, we can easily determine the expression for the output.

The expression for the AND gate output is written $A \cdot B$. This AND output is connected as an input to the OR gate along with C, another input. The OR gate operates on its inputs so that its output is the OR sum of the inputs. Thus, we can express the OR output as $x = A \cdot B + C$. (This final expression could also be written as $x = C + A \cdot B$ because it does not matter which term of the OR sum is written first.)

Operator Precedence

Occasionally, there may be confusion about which operation in an expression is performed first. The expression $A \cdot B + C$ can be interpreted in two different ways: (1) $A \cdot B$ is ORed with C, or (2) A is ANDed with the term $B + C$. To avoid this confusion, it will be understood that if an expression contains both AND and OR operations, the AND operations are performed first, unless there are *parentheses* in the expression, in which case the operation inside the parentheses is to be performed first. This is the same rule that is used in ordinary algebra to determine the order of operations.

To illustrate further, consider the circuit in Figure 3-13(b). The expression for the OR gate output is simply $A + B$. This output serves as an input to the AND gate along with another input, C. Thus, we express the output of the AND gate as $x = (A + B) \cdot C$. Note the use of parentheses here to indicate that A and B are ORed *first,* before their OR sum is ANDed with C. Without the parentheses it would be interpreted *incorrectly,* because $A + B \cdot C$ means that A is ORed with the product $B \cdot C$.

Circuits Containing INVERTERs

Whenever an INVERTER is present in a logic-circuit diagram, its output expression is simply equal to the input expression with a bar over it. Figure 3-14 shows two examples using INVERTERs. In Figure 3-14(a), input A is fed through an INVERTER, whose output is therefore $\overline{A}$. The INVERTER output is fed to an OR gate together with B, so that the *OR* output is equal to $\overline{A} + B$. Note that the bar is over the A alone, indicating that A is first inverted and then ORed with B.

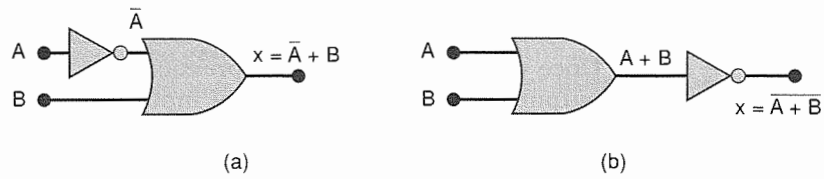


FIGURE 3-14 Circuits using INVERTERs.

In Figure 3-14(b), the output of the OR gate is equal to $A + B$ and is fed through an INVERTER. The INVERTER output is therefore equal to $(\overline{A + B})$ because it inverts the *complete* input expression. Note that the bar covers the entire expression $(A + B)$. This is important because, as will be shown later, the expressions $(\overline{A} + \overline{B})$ and $(\overline{A} + \overline{B})$ are *not* equivalent. The expression $(\overline{A + B})$ means that A is ORed with B and then their OR sum is inverted, whereas the expression $(\overline{A} + \overline{B})$ indicates that A is inverted and B is inverted and the results are then ORed together.

Figure 3-15 shows two more examples, which should be studied carefully. Note especially the use of *two* separate sets of parentheses in Figure 3-15(b). Also notice in Figure 3-15(a) that the input variable A is connected as an input to two different gates.

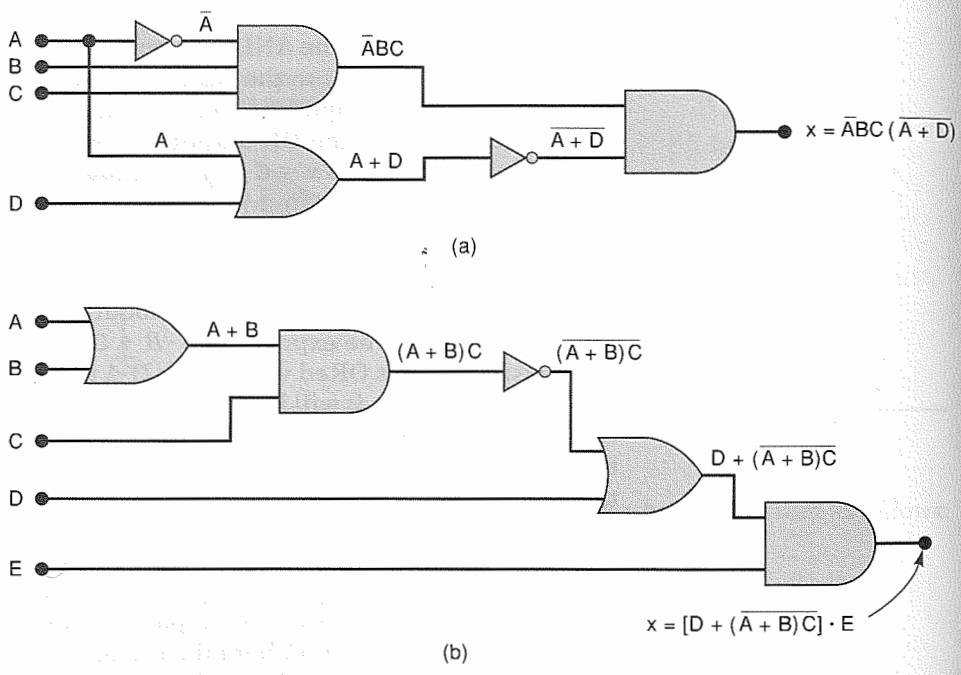


FIGURE 3-15 More examples.

REVIEW QUESTIONS

1. In Figure 3-15(a), change each AND gate to an OR gate, and change the OR gate to an AND gate. Then write the expression for output x.
2. In Figure 3-15(b), change each AND gate to an OR gate, and each OR gate to an AND gate. Then write the expression for x.

3-7 EVALUATING LOGIC-CIRCUIT OUTPUTS

Once we have the Boolean expression for a circuit output, we can obtain the output logic level for any set of input levels. For example, suppose that we want to know the logic level of the output x for the circuit in Figure 3-15(a) for the case where $A = 0, B = 1, C = 1$, and $D = 1$. As in ordinary algebra, the

value of x can be found by "plugging" the values of the variables into the expression and performing the indicated operations as follows:

$$\begin{aligned} x &= \overline{A}BC(\overline{A + D}) \\ &= \overline{0} \cdot 1 \cdot 1 \cdot (\overline{0 + 1}) \\ &= 1 \cdot 1 \cdot 1 \cdot (\overline{0 + 1}) \\ &= 1 \cdot 1 \cdot 1 \cdot (\overline{1}) \\ &= 1 \cdot 1 \cdot 1 \cdot 0 \\ &= 0 \end{aligned}$$

As another illustration, let us evaluate the output of the circuit in Figure 3-15(b) for $A = 0, B = 0, C = 1, D = 1$, and $E = 1$.

$$\begin{aligned} x &= [D + \overline{(A + B)C}] \cdot E \\ &= [1 + \overline{(0 + 0) \cdot 1}] \cdot 1 \\ &= [1 + \overline{0 \cdot 1}] \cdot 1 \\ &= [1 + \overline{0}] \cdot 1 \\ &= [1 + 1] \cdot 1 \\ &= 1 \cdot 1 \\ &= 1 \end{aligned}$$

In general, the following rules must always be followed when evaluating a Boolean expression:

1. First, perform all inversions of single terms; that is, $\overline{0} = 1$ or $\overline{1} = 0$.
2. Then perform all operations within parentheses.
3. Perform an AND operation before an OR operation unless parentheses indicate otherwise.
4. If an expression has a bar over it, perform the operations inside the expression first and then invert the result.

For practice, determine the outputs of both circuits in Figure 3-15 for the case where all inputs are 1. The answers are $x = 0$ and $x = 1$, respectively.

Analysis Using a Table

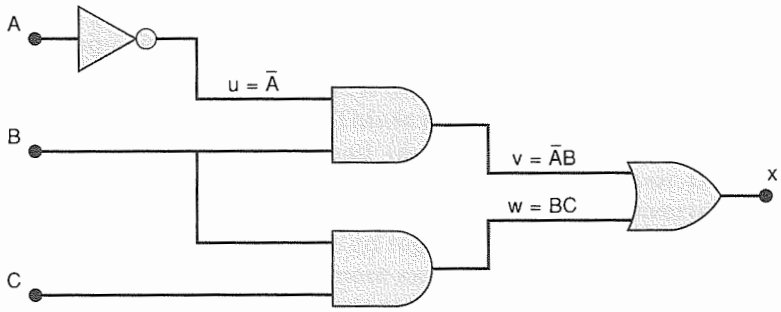
Whenever you have a circuit made up of multiple logic gates and you want to know how it works, the best way to analyze it is to use a truth table. The advantages of this method are:

It allows you to analyze one gate or logic combination at a time.

It allows you to easily double-check your work.

When you are done, you have a table that is of tremendous benefit in troubleshooting the logic circuit.

Recall that a truth table lists all the possible input combinations in numerical order. For each possible input combination, you can determine the logic state at every point (node) in the logic circuit including the output. For example refer to Figure 3-16(a). There are several intermediate nodes in this circuit that are neither inputs nor outputs to the circuit. They are simply connections between one gate's output and another gate's input. In this diagram they have been labeled u, v, and w. The first step after listing all the input combinations is to create a column in the truth table for each intermediate signal (node) as shown in Figure 3-16(b). Node u has been filled in as the complement of A.



(a)

A	B	C	u= Ā	v= ĀB	w= BC	x= v+w
0	0	0	1			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	0			

(b)

A	B	C	u= Ā	v= ĀB	w= BC	x= v+w
0	0	0	1	0		
0	0	1	1	0		
0	1	0	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	0	1	0	0		
1	1	0	0	0		
1	1	1	0	0		

(c)

A	B	C	u= Ā	v= ĀB	w= BC	x= v+w
0	0	0	1	0	0	
0	0	1	1	0	0	
0	1	0	1	1	0	
0	1	1	1	1	1	
1	0	0	0	0	0	
1	0	1	0	0	0	
1	1	0	0	0	0	
1	1	1	0	0	1	

(d)

A	B	C	u= Ā	v= ĀB	w= BC	x= v+w
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

(e)

FIGURE 3-16 Analysis of a logic circuit using truth tables.

The next step is to fill in the values for column v as shown in Figure 3-16(c). From the diagram we can see that $v = \overline{A}B$. The node v should be HIGH when $\overline{A}$ (node u) is HIGH AND B is HIGH. This occurs whenever A is LOW AND B is HIGH. The third step is to predict the values at node w which is the logical product of BC. This column is HIGH whenever B is HIGH AND C is HIGH as shown in Figure 3-16(d). The final step is to logically combine columns v and w to predict the output x. Since $x = v + w$, the x output will be HIGH when v is HIGH OR w is HIGH as shown in Figure 3-16(e).

If you built this circuit and it was not producing the correct output for x under all conditions, this table could be used to find the trouble. The general procedure is to test the circuit under each combination of inputs. If any input combination produces an incorrect output (i.e., a fault), compare the actual logic state of each intermediate node in the circuit with the correct theoretical value in the table while applying that input condition. If the logic state for an intermediate node is *correct*, the problem must be farther to the right of that node. If the logic state for an intermediate node is *incorrect*, the problem must be to the left of that node (or that node is shorted to something). Detailed troubleshooting procedures and possible circuit faults will be covered more extensively in Chapter 4.

EXAMPLE 3-6

Analyze the operation of Figure 3-15(a) by creating a table showing the logic state at each node of the circuit.

Solution

Fill in the column for t by entering a 1 for all entries where $A = 0$ and $B = 1$ and $C = 1$.

Fill in the column for u by entering a 1 for all entries where $A = 1$ or $D = 1$.

Fill in the column for v by complementing all entries in column u.

Fill in the column for x by entering a 1 for all entries where $t = 1$ and $v = 1$.

A	B	C	D	$t = \overline{A}BC$	$u = A + D$	$v = \overline{A + D}$	$x = tv$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0

REVIEW QUESTIONS

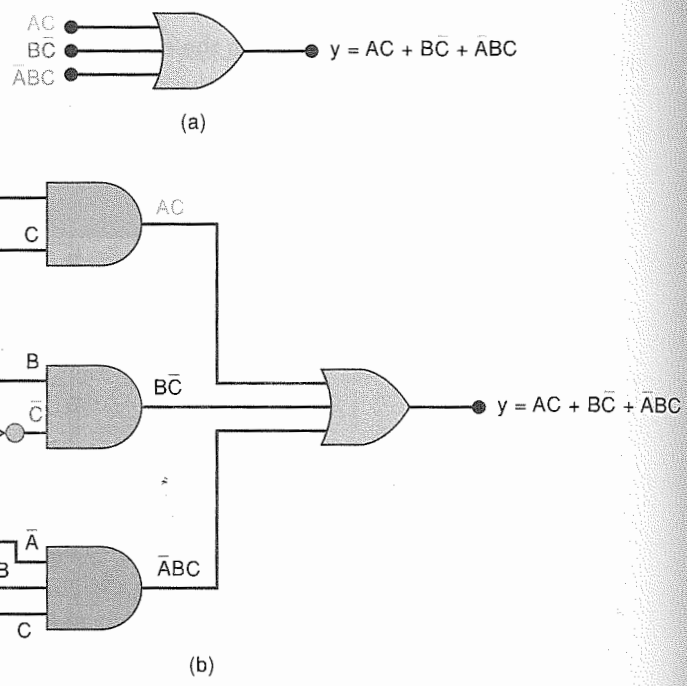
1. Use the expression for x to determine the output of the circuit in Figure 3-15(a) for the conditions $A = 0, B = 1, C = 1$, and $D = 0$.
2. Use the expression for x to determine the output of the circuit in Figure 3-15(b) for the conditions $A = B = E = 1, C = D = 0$.
3. Determine the answers to Questions 1 and 2 by finding the logic levels present at each gate output using a table as in Figure 3-16.

3-8 IMPLEMENTING CIRCUITS FROM BOOLEAN EXPRESSIONS

When the operation of a circuit is defined by a Boolean expression, we can draw a logic-circuit diagram directly from that expression. For example, if we needed a circuit that was defined by $x = A \cdot B \cdot C$, we would immediately know that all that was needed was a three-input AND gate. If we needed a circuit that was defined by $x = A + \overline{B}$, we would use a two-input OR gate with an INVERTER on one of the inputs. The same reasoning used for these simple cases can be extended to more complex circuits.

Suppose that we wanted to construct a circuit whose output is $y = AC + B\overline{C} + \overline{A}BC$. This Boolean expression contains three terms (AC, $B\overline{C}$, $\overline{A}BC$), which are ORed together. This tells us that a three-input OR gate is required with inputs that are equal to AC, $B\overline{C}$, and $\overline{A}BC$. This is illustrated in Figure 3-17(a), where a three-input OR gate is drawn with inputs labeled as AC, $B\overline{C}$, and $\overline{A}BC$.

FIGURE 3-17
Constructing a logic circuit from a Boolean expression.



Each OR gate input is an AND product term, which means that an AND gate with appropriate inputs can be used to generate each of these terms. This is shown in Figure 3-17(b), which is the final circuit diagram. Note the use of INVERTERs to produce the $\overline{A}$ and $\overline{C}$ terms required in the expression.

This same general approach can always be followed, although we shall find that there are some clever, more efficient techniques that can be employed. For now, however, this straightforward method will be used to minimize the number of new items that are to be learned.

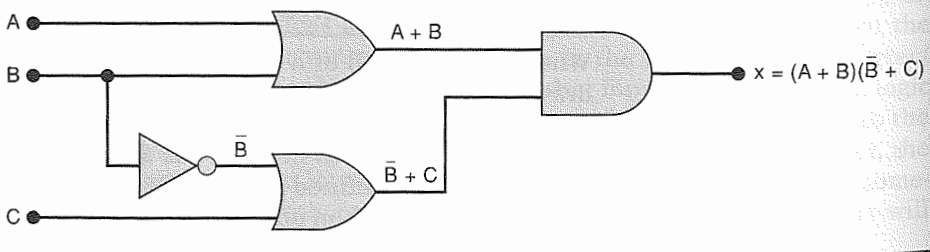
EXAMPLE 3-7

Draw the circuit diagram to implement the expression $x = (A + B)(\overline{B} + C)$.

Solution

This expression shows that the terms $A + B$ and $\overline{B} + C$ are inputs to an AND gate, and each of these two terms is generated from a separate OR gate. The result is drawn in Figure 3-18.

FIGURE 3-18
Example 3-7.



REVIEW QUESTIONS

1. Draw the circuit diagram that implements the expression $x = \overline{A}BC(\overline{A + D})$ using gates with no more than three inputs.
2. Draw the circuit diagram for the expression $y = AC + B\overline{C} + \overline{A}BC$.
3. Draw the circuit diagram for $x = [D + (\overline{A + B})\overline{C}] \cdot E$.

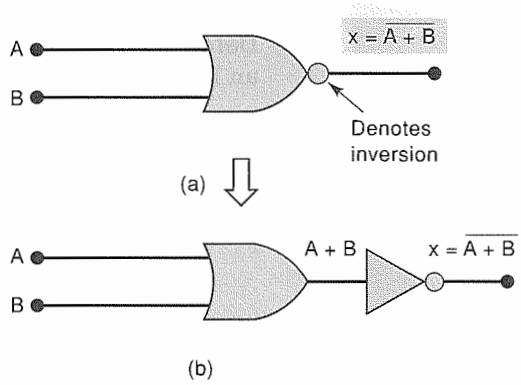
3-9 NOR GATES AND NAND GATES

Two other types of logic gates, NOR gates and NAND gates, are widely used in digital circuits. These gates actually combine the basic AND, OR, and NOT operations, so it is a relatively simple matter to write their Boolean expressions.

NOR Gate

The symbol for a two-input **NOR gate** is shown in Figure 3-19(a). It is the same as the OR gate symbol except that it has a small circle on the output. The small circle represents the inversion operation. Thus, the NOR gate operates like an OR gate followed by an INVERTER, so that the circuits in Figure 3-19(a) and (b) are equivalent, and the output expression for the NOR gate is $x = \overline{A + B}$.

FIGURE 3-19 (a) NOR symbol; (b) equivalent circuit; (c) truth table.



A	B	OR: A + B	NOR: A + B
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

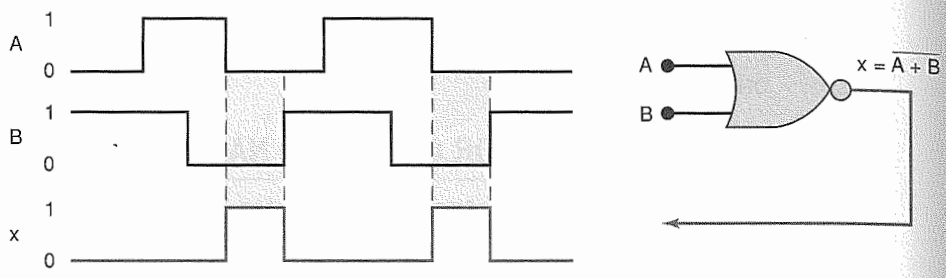
(c)

The truth table in Figure 3-19(c) shows that the NOR gate output is the exact inverse of the OR gate output for all possible input conditions. An OR gate output goes HIGH when any input is HIGH; the NOR gate output goes LOW when any input is HIGH. This same operation can be extended to NOR gates with more than two inputs.

EXAMPLE 3-8

Determine the waveform at the output of a NOR gate for the input waveforms shown in Figure 3-20.

FIGURE 3-20
Example 3-8.



Solution

One way to determine the NOR output waveform is to find first the OR output waveform and then invert it (change all 1s to 0s, and vice versa). Another way utilizes the fact that a NOR gate output will be HIGH *only* when all inputs are LOW. Thus, you can examine the input waveforms, find those time intervals where they are all LOW, and make the NOR output HIGH for those intervals. The NOR output will be LOW for all other time intervals. The resultant output waveform is shown in the figure.

EXAMPLE 3-9

Determine the Boolean expression for a three-input NOR gate followed by an INVERTER.

Solution

Refer to Figure 3-21, where the circuit diagram is shown. The expression at the NOR output is $\overline{(A + B + C)}$, which is then fed through an INVERTER to produce

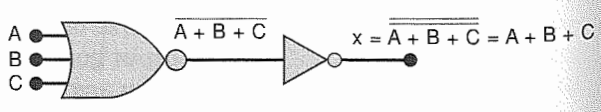
$$x = \overline{\overline{(A + B + C)}}$$

The presence of the double inversion signs indicates that the quantity $(A + B + C)$ has been inverted and then inverted again. It should be clear that this simply results in the expression $(A + B + C)$ being unchanged. That is,

$$x = \overline{\overline{(A + B + C)}} = (A + B + C)$$

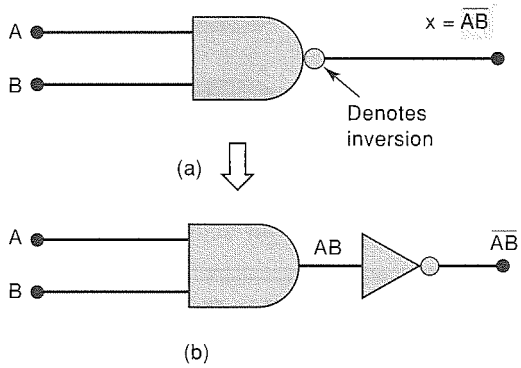
Whenever two inversion bars are over the same variable or quantity, they cancel each other out, as in the example above. However, in cases such as $\overline{A} + \overline{B}$ the inversion bars do not cancel. This is because the smaller inversion bars invert the single variables A and B, while the wide bar inverts the quantity $(\overline{A} + \overline{B})$. Thus, $\overline{A} + \overline{B} \neq A + B$. Similarly, $\overline{A}\,\overline{B} \neq AB$.

FIGURE 3-21 Example 3-9.



NAND Gate

The symbol for a two-input **NAND gate** is shown in Figure 3-22(a). It is the same as the AND gate symbol except for the small circle on its output. Once again, this small circle denotes the inversion operation. Thus, the NAND operates like an AND gate followed by an INVERTER, so that the circuits of Figure 3-22(a) and (b) are equivalent, and the output expression for the NAND gate is $x = \overline{AB}$.



A	B	AND: AB	NAND: $\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(c)

FIGURE 3-22 (a) NAND symbol; (b) equivalent circuit; (c) truth table.

The truth table in Figure 3-22(c) shows that the NAND gate output is the exact inverse of the AND gate for all possible input conditions. The AND output goes HIGH only when all inputs are HIGH, while the NAND output goes LOW only when all inputs are HIGH. This same characteristic is true of NAND gates having more than two inputs.

EXAMPLE 3-10

Determine the output waveform of a NAND gate having the input waveforms shown in Figure 3-23.

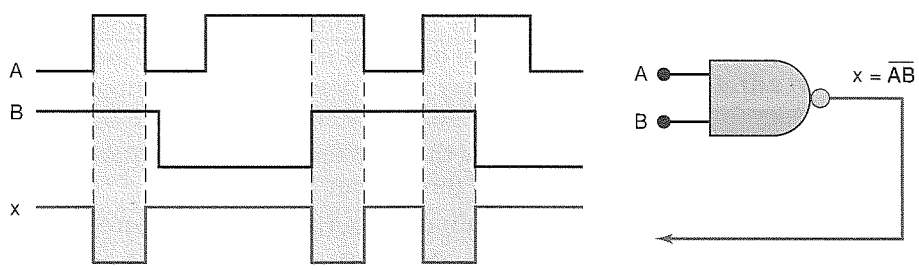


FIGURE 3-23 Example 3-10.

Solution

One way is to draw first the output waveform for an AND gate and then invert it. Another way utilizes the fact that a NAND output will be LOW only when all inputs are HIGH. Thus, you can find those time intervals during which the inputs are all HIGH, and make the NAND output LOW for those intervals. The output will be HIGH at all other times.

EXAMPLE 3-11

Implement the logic circuit that has the expression $x = \overline{AB \cdot (\overline{C + D})}$ using only NOR and NAND gates.

Solution

The $(\overline{C + D})$ term is the expression for the output of a NOR gate. This term is ANDed with A and B, and the result is inverted; this, of course, is the NAND operation. Thus, the circuit is implemented as shown in Figure 3-24. Note that the NAND gate first ANDs the A, B, and $(\overline{C + D})$ terms, and then it inverts the *complete* result.

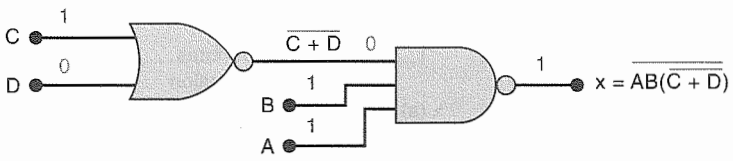


FIGURE 3-24 Examples 3-11 and 3-12.

EXAMPLE 3-12

Determine the output level in Figure 3-24 for $A = B = C = 1$ and $D = 0$.

Solution

In the first method we use the expression for x.

$$\begin{aligned} x &= \overline{AB(\overline{C + D})} \\ &= \overline{1 \cdot 1 \cdot (\overline{1 + 0})} \\ &= \overline{1 \cdot 1 \cdot (\overline{1})} \\ &= \overline{1 \cdot 1 \cdot 0} \\ &= \overline{0} = 1 \end{aligned}$$

In the second method, we write down the input logic levels on the circuit diagram (shown in color in Figure 3-24) and follow these levels through each gate to the final output. The NOR gate has inputs of 1 and 0 to produce an output of 0 (an OR would have produced an output of 1). The NAND gate thus has input levels of 0, 1, and 1 to produce an output of 1 (an AND would have produced an output of 0).

REVIEW QUESTIONS

1. What is the only set of input conditions that will produce a HIGH output from a three-input NOR gate?
2. Determine the output level in Figure 3-24 for $A = B = 1$, $C = D = 0$.
3. Change the NOR gate of Figure 3-24 to a NAND gate, and change the NAND to a NOR. What is the new expression for x?

3-10 BOOLEAN THEOREMS

We have seen how Boolean algebra can be used to help analyze a logic circuit and express its operation mathematically. We will continue our study of Boolean algebra by investigating the various **Boolean theorems** (rules) that

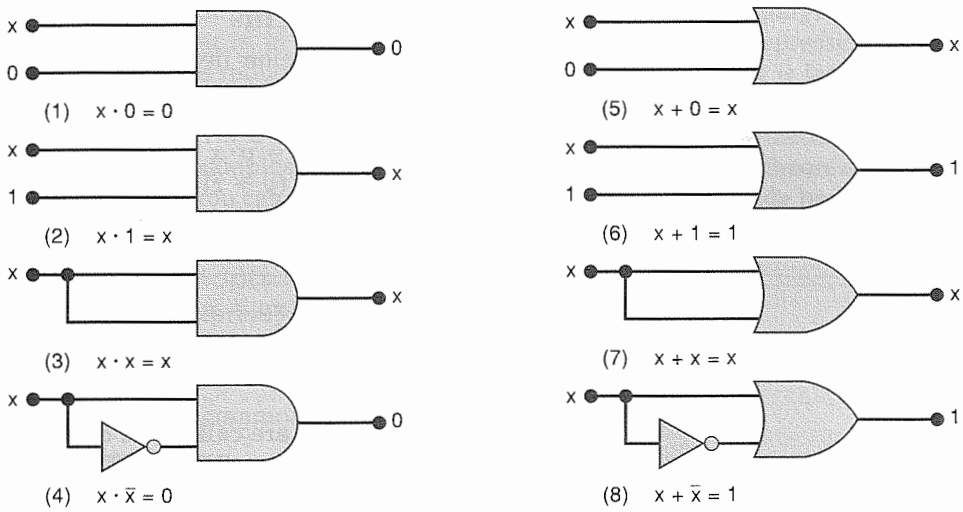


FIGURE 3-25 Single-variable theorems.

can help us to simplify logic expressions and logic circuits. The first group of theorems is given in Figure 3-25. In each theorem, x is a logic variable that can be either a 0 or a 1. Each theorem is accompanied by a logic-circuit diagram that demonstrates its validity.

Theorem (1) states that if any variable is ANDed with 0, the result must be 0. This is easy to remember because the AND operation is just like ordinary multiplication, where we know that anything multiplied by 0 is 0. We also know that the output of an AND gate will be 0 whenever any input is 0, regardless of the level on the other input.

Theorem (2) is also obvious by comparison with ordinary multiplication.

Theorem (3) can be proved by trying each case. If $x = 0$, then $0 \cdot 0 = 0$; if $x = 1$, then $1 \cdot 1 = 1$. Thus, $x \cdot x = x$.

Theorem (4) can be proved in the same manner. However, it can also be reasoned that at any time either x or its inverse $\bar{x}$ must be at the 0 level, and so their AND product always must be 0.

Theorem (5) is straightforward, since 0 *added* to anything does not affect its value, either in regular addition or in OR addition.

Theorem (6) states that if any variable is ORed with 1, the result will always be 1. We check this for both values of x: $0 + 1 = 1$ and $1 + 1 = 1$. Equivalently, we can remember that an OR gate output will be 1 when *any* input is 1, regardless of the value of the other input.

Theorem (7) can be proved by checking for both values of x: $0 + 0 = 0$ and $1 + 1 = 1$.

Theorem (8) can be proved similarly, or we can just reason that at any time either x or $\bar{x}$ must be at the 1 level so that we are always ORing a 0 and a 1, which always results in 1.

Before introducing any more theorems, we should point out that when theorems (1) through (8) are applied, the variable x may actually represent an expression containing more than one variable. For example, if we have $A\bar{B}(\overline{A\bar{B}})$, we can invoke theorem (4) by letting $x = A\bar{B}$. Thus, we can say that $A\bar{B}(\overline{A\bar{B}}) = 0$. The same idea can be applied to the use of any of these theorems.

Multivariable Theorems

The theorems presented below involve more than one variable:

(9) $x + y = y + x$
(10) $x \cdot y = y \cdot x$
(11) $x + (y + z) = (x + y) + z = x + y + z$
(12) $x(yz) = (xy)z = xyz$
(13*a*) $x(y + z) = xy + xz$
(13*b*) $(w + x)(y + z) = wy + xy + wz + xz$
(14) $x + xy = x$
(15*a*) $x + \overline{x}y = x + y$
(15*b*) $\overline{x} + xy = \overline{x} + y$

Theorems (9) and (10) are called the *commutative laws.* These laws indicate that the order in which we OR or AND two variables is unimportant; the result is the same.

Theorems (11) and (12) are the *associative laws,* which state that we can group the variables in an AND expression or OR expression any way we want.

Theorem (13) is the *distributive law,* which states that an expression can be expanded by multiplying term by term just the same as in ordinary algebra. This theorem also indicates that we can factor an expression. That is, if we have a sum of two (or more) terms, each of which contains a common variable, the common variable can be factored out just as in ordinary algebra. For example, if we have the expression $A\overline{B}C + \overline{A}\overline{B}\overline{C}$, we can factor out the $\overline{B}$ variable:

$$A\overline{B}C + \overline{A}\overline{B}\overline{C} = \overline{B}(AC + \overline{A}\overline{C})$$

As another example, consider the expression $ABC + ABD$. Here the two terms have the variables A and B in common, and so $A \cdot B$ can be factored out of both terms. That is,

$$ABC + ABD = AB(C + D)$$

Theorems (9) to (13) are easy to remember and use because they are identical to those of ordinary algebra. Theorems (14) and (15), on the other hand, do not have any counterparts in ordinary algebra. Each can be proved by trying all possible cases for x and y. This is illustrated (for theorem 14) by creating an analysis table for the equation $x + xy$ as follows:

x	y	xy	x + xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Notice that the value of the entire expression $(x + xy)$ is always the same as x.

Theorem (14) can also be proved by factoring and using theorems (6) and (2) as follows:

$$\begin{aligned} x + xy &= x(1 + y) \\ &= x \cdot 1 && \text{[using theorem (6)]} \\ &= x && \text{[using theorem (2)]} \end{aligned}$$

All of these Boolean theorems can be useful in simplifying a logic expression—that is, in reducing the number of terms in the expression. When this is done, the reduced expression will produce a circuit that is less complex than the one that the original expression would have produced. A good portion of the next chapter will be devoted to the process of circuit simplification. For now, the following examples will serve to illustrate how the Boolean theorems can be applied. **Note:** You can find all the Boolean theorems on the inside back cover.

EXAMPLE 3-13

Simplify the expression $y = A\overline{B}D + A\overline{B}\,\overline{D}$.

Solution

Factor out the common variables $A\overline{B}$ using theorem (13):

$$y = A\overline{B}(D + \overline{D})$$

Using theorem (8), the term in parentheses is equivalent to 1. Thus,

$$\begin{aligned} y &= A\overline{B} \cdot 1 \\ &= A\overline{B} \qquad \text{[using theorem (2)]} \end{aligned}$$

EXAMPLE 3-14

Simplify $z = (\overline{A} + B)(A + B)$.

Solution

The expression can be expanded by multiplying out the terms [theorem (13)]:

$$z = \overline{A} \cdot A + \overline{A} \cdot B + B \cdot A + B \cdot B$$

Invoking theorem (4), the term $\overline{A} \cdot A = 0$. Also, $B \cdot B = B$ [theorem (3)]:

$$z = 0 + \overline{A} \cdot B + B \cdot A + B = \overline{A}B + AB + B$$

Factoring out the variable B [theorem (13)], we have

$$z = B(\overline{A} + A + 1)$$

Finally, using theorems (2) and (6),

$$z = B$$

EXAMPLE 3-15

Simplify $x = ACD + \overline{A}BCD$.

Solution

Factoring out the common variables CD, we have

$$x = CD(A + \overline{A}B)$$

Utilizing theorem (15a), we can replace $A + \overline{A}B$ by $A + B$, so

$$\begin{aligned} x &= CD(A + B) \\ &= ACD + BCD \end{aligned}$$

REVIEW QUESTIONS

1. Use theorems (13) and (14) to simplify $y = A\overline{C} + AB\overline{C}$.
2. Use theorems (13) and (8) to simplify $y = \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$.
3. Use theorems (13) and (15b) to simplify $y = \overline{A}D + ABD$.

3-11 DEMORGAN'S THEOREMS

Two of the most important theorems of Boolean algebra were contributed by a great mathematician named DeMorgan. **DeMorgan's theorems** are extremely useful in simplifying expressions in which a product or sum of variables is inverted. The two theorems are:

$$(16)\quad \overline{(x + y)} = \overline{x} \cdot \overline{y}$$
$$(17)\quad \overline{(x \cdot y)} = \overline{x} + \overline{y}$$

Theorem (16) says that when the OR sum of two variables is inverted, this is the same as inverting each variable individually and then ANDing these inverted variables. Theorem (17) says that when the AND product of two variables is inverted, this is the same as inverting each variable individually and then ORing them. Each of DeMorgan's theorems can readily be proven by checking for all possible combinations of x and y. This will be left as an end-of-chapter exercise.

Although these theorems have been stated in terms of single variables x and y, they are equally valid for situations where x and/or y are expressions that contain more than one variable. For example, let's apply them to the expression $(A\overline{B} + C)$ as shown below:

$$\overline{(A\overline{B} + C)} = \overline{(A\overline{B})} \cdot \overline{C}$$

Note that we used theorem (16) and treated $A\overline{B}$ as x and C as y. The result can be further simplified because we have a product $A\overline{B}$ that is inverted. Using theorem (17), the expression becomes

$$\overline{A\overline{B}} \cdot \overline{C} = (\overline{A} + \overline{\overline{B}}) \cdot \overline{C}$$

Notice that we can replace $\overline{\overline{B}}$ by B, so that we finally have

$$(\overline{A} + B) \cdot \overline{C} = \overline{A}\,\overline{C} + B\overline{C}$$

This final result contains only inverter signs that invert a single variable.

EXAMPLE 3-16

Simplify the expression $z = \overline{(\overline{A} + C) \cdot (B + \overline{D})}$ to one having only single variables inverted.

Solution

Using theorem (17), and treating $(\overline{A} + C)$ as x and $(B + \overline{D})$ as y, we have

$$z = \overline{(\overline{A} + C)} + \overline{(B + \overline{D})}$$

We can think of this as breaking the large inverter sign down the middle and changing the AND sign (·) to an OR sign (+). Now the term $(\overline{\overline{A} + C})$ can be simplified by applying theorem (16). Likewise, $(\overline{B + \overline{D}})$ can be simplified:

$$\begin{aligned} z &= (\overline{\overline{A} + C}) + (\overline{B + \overline{D}}) \\ &= (\overline{\overline{A}} \cdot \overline{C}) + \overline{B} \cdot \overline{\overline{D}} \end{aligned}$$

Here we have broken the larger inverter signs down the middle and replaced the (+) with a (·). Canceling out the double inversions, we have finally

$$z = A\overline{C} + \overline{B}D$$

Example 3-16 points out that when using DeMorgan's theorems to reduce an expression, we may break an inverter sign at any point in the expression and change the operator sign at that point in the expression to its opposite (+ is changed to ·, and vice versa). This procedure is continued until the expression is reduced to one in which only single variables are inverted. Two more examples are given below.

Example 1

$$\begin{aligned} z &= \overline{A + \overline{B} \cdot C} \\ &= \overline{A} \cdot (\overline{\overline{B} \cdot C}) \\ &= \overline{A} \cdot (\overline{\overline{B}} + \overline{C}) \\ &= \overline{A} \cdot (B + \overline{C}) \end{aligned}$$

Example 2

$$\begin{aligned} \omega &= \overline{(A + BC) \cdot (D + EF)} \\ &= (\overline{A + BC}) + (\overline{D + EF}) \\ &= (\overline{A} \cdot \overline{BC}) + (\overline{D} \cdot \overline{EF}) \\ &= [\overline{A} \cdot (\overline{B} + \overline{C})] + [\overline{D} \cdot (\overline{E} + \overline{F})] \\ &= \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + \overline{D}\,\overline{E} + \overline{D}\,\overline{F} \end{aligned}$$

DeMorgan's theorems are easily extended to more than two variables. For example, it can be proved that

$$\begin{aligned} \overline{x + y + z} &= \overline{x} \cdot \overline{y} \cdot \overline{z} \\ \overline{x \cdot y \cdot z} &= \overline{x} + \overline{y} + \overline{z} \end{aligned}$$

Here, we see that the large inverter sign is broken at *two* points in the expression and the operator sign is changed to its opposite. This can be extended to any number of variables. Again, realize that the variables can themselves be expressions rather than single variables. Here is another example.

$$\begin{aligned} x &= \overline{\overline{AB} \cdot \overline{CD} \cdot \overline{EF}} \\ &= \overline{\overline{AB}} + \overline{\overline{CD}} + \overline{\overline{EF}} \\ &= AB + CD + EF \end{aligned}$$

Implications of DeMorgan's Theorems

Let us examine theorems (16) and (17) from the standpoint of logic circuits. First, consider theorem (16):

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

The left-hand side of the equation can be viewed as the output of a NOR gate whose inputs are x and y. The right-hand side of the equation, on the other hand,

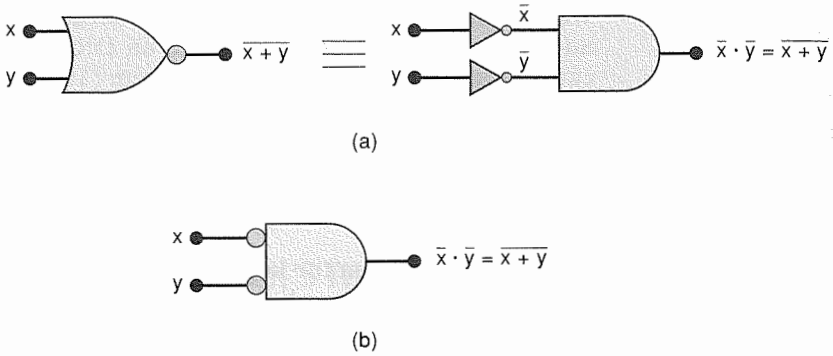


FIGURE 3-26
(a) Equivalent circuits implied by theorem (16); (b) alternative symbol for the NOR function.

is the result of first inverting both x and y and then putting them through an AND gate. These two representations are equivalent and are illustrated in Figure 3-26(a). What this means is that an AND gate with INVERTERs on each of its inputs is equivalent to a NOR gate. In fact, both representations are used to represent the NOR function. When the AND gate with inverted inputs is used to represent the NOR function, it is usually drawn as shown in Figure 3-26(b), where the small circles on the inputs represent the inversion operation.

Now consider theorem (17):

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

The left side of the equation can be implemented by a NAND gate with inputs x and y. The right side can be implemented by first inverting inputs x and y and then putting them through an OR gate. These two equivalent representations are shown in Figure 3-27(a). The OR gate with INVERTERs on each of its inputs is equivalent to the NAND gate. In fact, both representations are used to represent the NAND function. When the OR gate with inverted inputs is used to represent the NAND function, it is usually drawn as shown in Figure 3-27(b), where the circles again represent inversion.

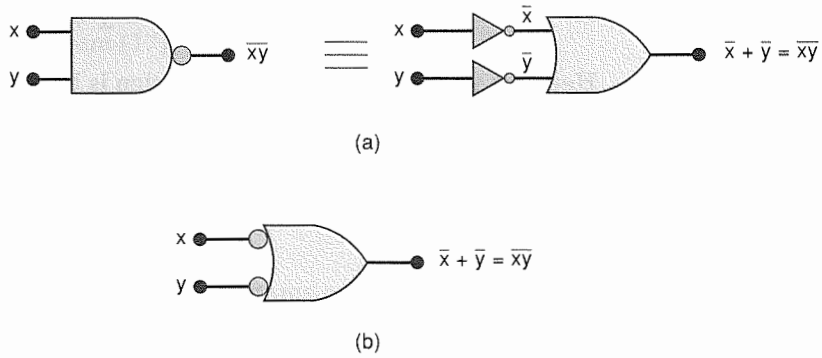


FIGURE 3-27
(a) Equivalent circuits implied by theorem (17); (b) alternative symbol for the NAND function.

EXAMPLE 3-17

Determine the output expression for the circuit of Figure 3-28 and simplify it using DeMorgan's theorems.

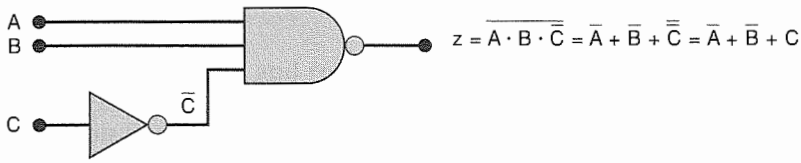


FIGURE 3-28
Example 3-17.

Solution

The expression for z is $z = \overline{AB\overline{C}}$. Use DeMorgan's theorem to break the large inversion sign:

$$z = \overline{A} + \overline{B} + \overline{\overline{C}}$$

Cancel the double inversions over C to obtain

$$z = \overline{A} + \overline{B} + C$$

REVIEW QUESTIONS

1. Use DeMorgan's theorems to convert the expression $z = \overline{(A + B) \cdot \overline{C}}$ to one that has only single-variable inversions.
2. Repeat question 1 for the expression $y = \overline{R\overline{S}T + \overline{Q}}$.
3. Implement a circuit having output expression $z = \overline{A}\,\overline{B}C$ using only a NOR gate and an INVERTER.
4. Use DeMorgan's theorems to convert $y = \overline{A + \overline{B} + \overline{C}D}$ to an expression containing only single-variable inversions.

3-12 UNIVERSALITY OF NAND GATES AND NOR GATES

All Boolean expressions consist of various combinations of the basic operations of OR, AND, and INVERT. Therefore, any expression can be implemented using combinations of OR gates, AND gates, and INVERTERs. It is possible, however, to implement any logic expression using *only* NAND gates and no other type of gate. This is because NAND gates, in the proper combination, can be used to perform each of the Boolean operations OR, AND, and INVERT. This is demonstrated in Figure 3-29.

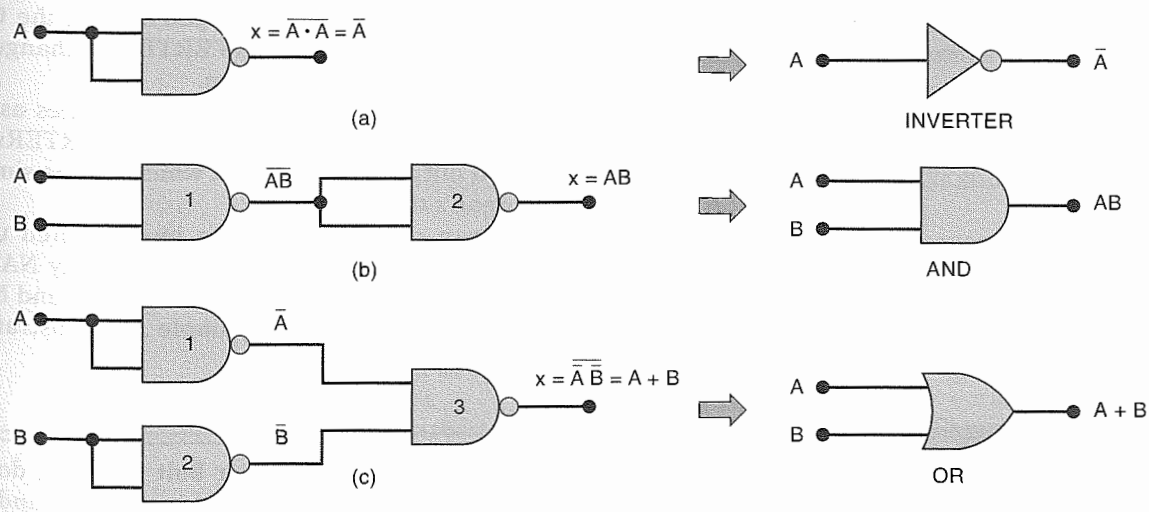


FIGURE 3-29 NAND gates can be used to implement any Boolean function.

REVIEW QUESTIONS

1. Use theorems (13) and (14) to simplify $y = A\overline{C} + AB\overline{C}$.
2. Use theorems (13) and (8) to simplify $y = \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$.
3. Use theorems (13) and (15b) to simplify $y = \overline{A}D + ABD$.

3-11 DEMORGAN'S THEOREMS

Two of the most important theorems of Boolean algebra were contributed by a great mathematician named DeMorgan. **DeMorgan's theorems** are extremely useful in simplifying expressions in which a product or sum of variables is inverted. The two theorems are:

$$(16) \quad \overline{(x + y)} = \overline{x} \cdot \overline{y}$$
$$(17) \quad \overline{(x \cdot y)} = \overline{x} + \overline{y}$$

Theorem (16) says that when the OR sum of two variables is inverted, this is the same as inverting each variable individually and then ANDing these inverted variables. Theorem (17) says that when the AND product of two variables is inverted, this is the same as inverting each variable individually and then ORing them. Each of DeMorgan's theorems can readily be proven by checking for all possible combinations of x and y. This will be left as an end-of-chapter exercise.

Although these theorems have been stated in terms of single variables x and y, they are equally valid for situations where x and/or y are expressions that contain more than one variable. For example, let's apply them to the expression $(A\overline{B} + C)$ as shown below:

$$\overline{(A\overline{B} + C)} = \overline{(A\overline{B})} \cdot \overline{C}$$

Note that we used theorem (16) and treated $A\overline{B}$ as x and C as y. The result can be further simplified because we have a product $A\overline{B}$ that is inverted. Using theorem (17), the expression becomes

$$\overline{A\overline{B}} \cdot \overline{C} = (\overline{A} + \overline{\overline{B}}) \cdot \overline{C}$$

Notice that we can replace $\overline{\overline{B}}$ by B, so that we finally have

$$(\overline{A} + B) \cdot \overline{C} = \overline{A}\,\overline{C} + B\overline{C}$$

This final result contains only inverter signs that invert a single variable.

EXAMPLE 3-16

Simplify the expression $z = \overline{(\overline{A} + C) \cdot (B + \overline{D})}$ to one having only single variables inverted.

Solution

Using theorem (17), and treating $(\overline{A} + C)$ as x and $(B + \overline{D})$ as y, we have

$$z = \overline{(\overline{A} + C)} + \overline{(B + \overline{D})}$$

We can think of this as breaking the large inverter sign down the middle and changing the AND sign (·) to an OR sign (+). Now the term $(\overline{\overline{A} + C})$ can be simplified by applying theorem (16). Likewise, $(B + \overline{D})$ can be simplified:

$$\begin{aligned} z &= (\overline{\overline{\overline{A} + C}}) + (\overline{B + \overline{D}}) \\ &= (\overline{\overline{A} \cdot \overline{C}}) + \overline{B} \cdot \overline{\overline{D}} \end{aligned}$$

Here we have broken the larger inverter signs down the middle and replaced the (+) with a (·). Canceling out the double inversions, we have finally

$$z = A\overline{C} + \overline{B}D$$

Example 3-16 points out that when using DeMorgan's theorems to reduce an expression, we may break an inverter sign at any point in the expression and change the operator sign at that point in the expression to its opposite (+ is changed to ·, and vice versa). This procedure is continued until the expression is reduced to one in which only single variables are inverted. Two more examples are given below.

Example 1

$$\begin{aligned} z &= \overline{A + \overline{B} \cdot C} \\ &= \overline{A} \cdot (\overline{\overline{B} \cdot C}) \\ &= \overline{A} \cdot (\overline{\overline{B}} + \overline{C}) \\ &= \overline{A} \cdot (B + \overline{C}) \end{aligned}$$

Example 2

$$\begin{aligned} \omega &= \overline{(A + BC) \cdot (D + EF)} \\ &= (\overline{A + BC}) + (\overline{D + EF}) \\ &= (\overline{A} \cdot \overline{BC}) + (\overline{D} \cdot \overline{EF}) \\ &= [\overline{A} \cdot (\overline{B} + \overline{C})] + [\overline{D} \cdot (\overline{E} + \overline{F})] \\ &= \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + \overline{D}\,\overline{E} + \overline{D}\,\overline{F} \end{aligned}$$

DeMorgan's theorems are easily extended to more than two variables. For example, it can be proved that

$$\begin{aligned} \overline{x + y + z} &= \overline{x} \cdot \overline{y} \cdot \overline{z} \\ \overline{x \cdot y \cdot z} &= \overline{x} + \overline{y} + \overline{z} \end{aligned}$$

Here, we see that the large inverter sign is broken at *two* points in the expression and the operator sign is changed to its opposite. This can be extended to any number of variables. Again, realize that the variables can themselves be expressions rather than single variables. Here is another example.

$$\begin{aligned} x &= \overline{\overline{AB} \cdot \overline{CD} \cdot \overline{EF}} \\ &= \overline{\overline{AB}} + \overline{\overline{CD}} + \overline{\overline{EF}} \\ &= AB + CD + EF \end{aligned}$$

Implications of DeMorgan's Theorems

Let us examine theorems (16) and (17) from the standpoint of logic circuits. First, consider theorem (16):

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

The left-hand side of the equation can be viewed as the output of a NOR gate whose inputs are x and y. The right-hand side of the equation, on the other hand,

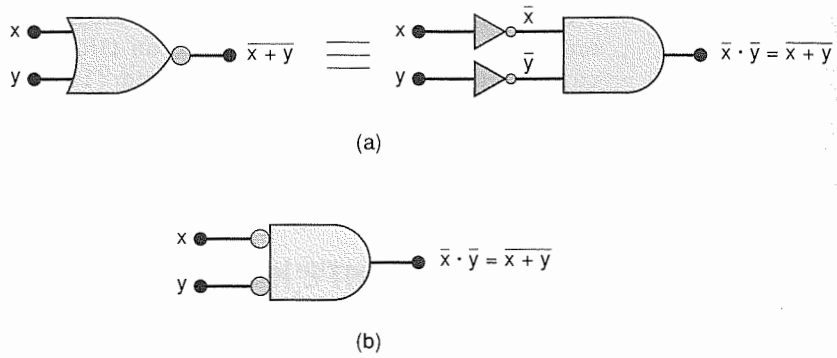


FIGURE 3-26
(a) Equivalent circuits implied by theorem (16); (b) alternative symbol for the NOR function.

is the result of first inverting both x and y and then putting them through an AND gate. These two representations are equivalent and are illustrated in Figure 3-26(a). What this means is that an AND gate with INVERTERs on each of its inputs is equivalent to a NOR gate. In fact, both representations are used to represent the NOR function. When the AND gate with inverted inputs is used to represent the NOR function, it is usually drawn as shown in Figure 3-26(b), where the small circles on the inputs represent the inversion operation.

Now consider theorem (17):

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

The left side of the equation can be implemented by a NAND gate with inputs x and y. The right side can be implemented by first inverting inputs x and y and then putting them through an OR gate. These two equivalent representations are shown in Figure 3-27(a). The OR gate with INVERTERs on each of its inputs is equivalent to the NAND gate. In fact, both representations are used to represent the NAND function. When the OR gate with inverted inputs is used to represent the NAND function, it is usually drawn as shown in Figure 3-27(b), where the circles again represent inversion.

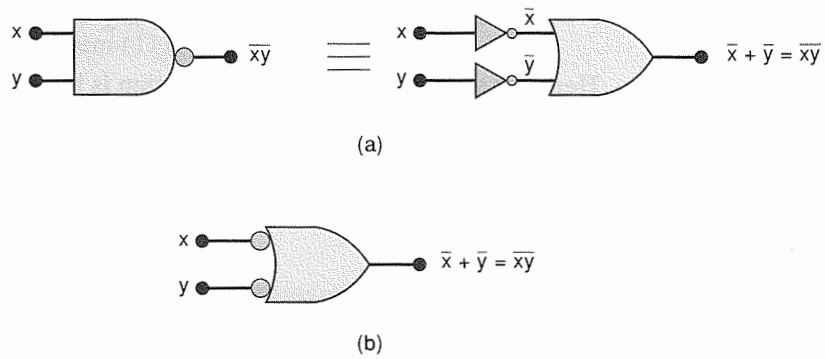


FIGURE 3-27
(a) Equivalent circuits implied by theorem (17); (b) alternative symbol for the NAND function.

EXAMPLE 3-17

Determine the output expression for the circuit of Figure 3-28 and simplify it using DeMorgan's theorems.

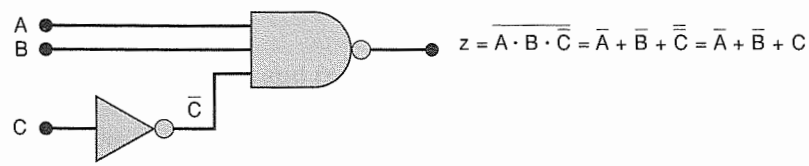


FIGURE 3-28
Example 3-17.

Solution

The expression for z is $z = \overline{AB\overline{C}}$. Use DeMorgan's theorem to break the large inversion sign:

$$z = \overline{A} + \overline{B} + \overline{\overline{C}}$$

Cancel the double inversions over C to obtain

$$z = \overline{A} + \overline{B} + C$$

REVIEW QUESTIONS

1. Use DeMorgan's theorems to convert the expression $z = \overline{(A + B) \cdot \overline{C}}$ to one that has only single-variable inversions.
2. Repeat question 1 for the expression $y = \overline{R\overline{S}T + \overline{Q}}$.
3. Implement a circuit having output expression $z = \overline{A}\,\overline{B}C$ using only a NOR gate and an INVERTER.
4. Use DeMorgan's theorems to convert $y = \overline{A + \overline{B} + \overline{C}D}$ to an expression containing only single-variable inversions.

3-12 UNIVERSALITY OF NAND GATES AND NOR GATES

All Boolean expressions consist of various combinations of the basic operations of OR, AND, and INVERT. Therefore, any expression can be implemented using combinations of OR gates, AND gates, and INVERTERs. It is possible, however, to implement any logic expression using *only* NAND gates and no other type of gate. This is because NAND gates, in the proper combination, can be used to perform each of the Boolean operations OR, AND, and INVERT. This is demonstrated in Figure 3-29.

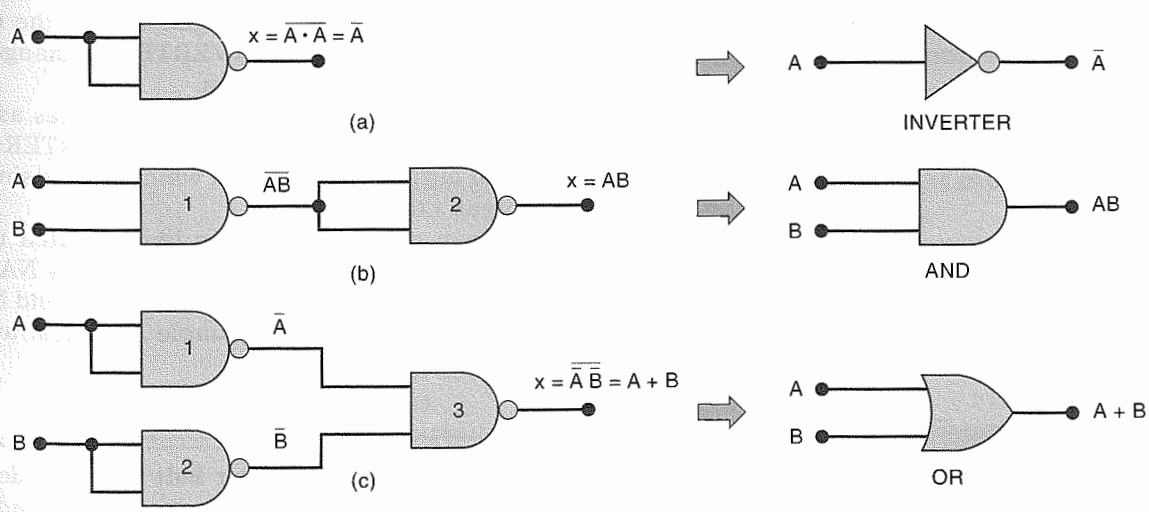


FIGURE 3-29 NAND gates can be used to implement any Boolean function.

First, in Figure 3-29(a), we have a two-input NAND gate whose inputs are purposely connected together so that the variable A is applied to both. In this configuration, the NAND simply acts as INVERTER because its output is $x = \overline{A \cdot A} = \overline{A}$.

In Figure 3-29(b), we have two NAND gates connected so that the AND operation is performed. NAND gate 2 is used as an INVERTER to change $\overline{AB}$ to $\overline{\overline{AB}} = AB$, which is the desired AND function.

The OR operation can be implemented using NAND gates connected as shown in Figure 3-29(c). Here NAND gates 1 and 2 are used as INVERTERs to invert the inputs, so that the final output is $x = \overline{\overline{A} \cdot \overline{B}}$, which can be simplified to $x = A + B$ using DeMorgan's theorem.

In a similar manner, it can be shown that NOR gates can be arranged to implement any of the Boolean operations. This is illustrated in Figure 3-30. Part (a) shows that a NOR gate with its inputs connected together behaves as an INVERTER because the output is $x = \overline{A + A} = \overline{A}$.

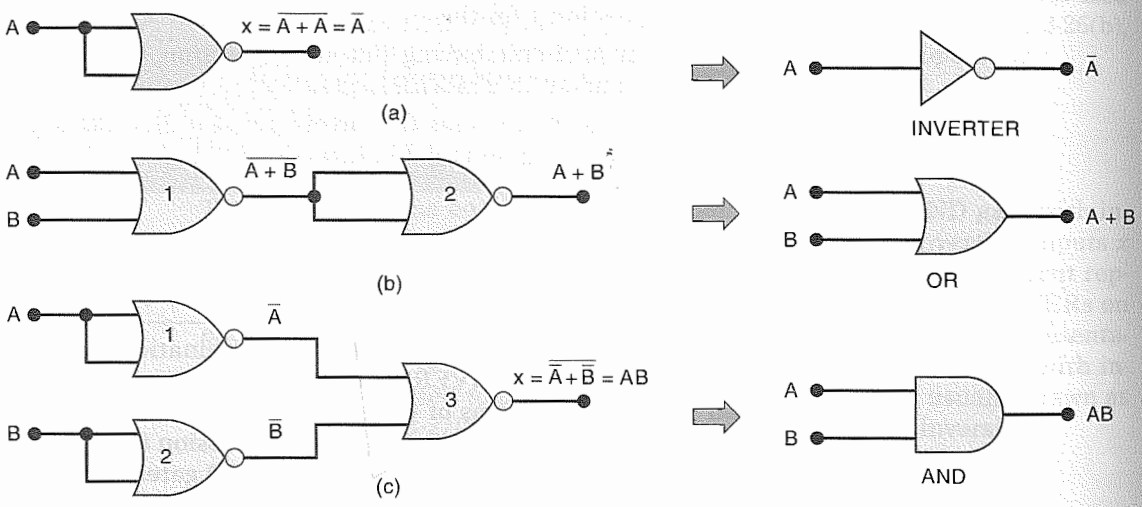


FIGURE 3-30 NOR gates can be used to implement any Boolean operation.

In Figure 3-30(b), two NOR gates are arranged so that the OR operation is performed. NOR gate 2 is used as an INVERTER to change $\overline{A + B}$ to $\overline{\overline{A + B}} = A + B$, which is the desired OR function.

The AND operation can be implemented with NOR gates as shown in Figure 3-30(c). Here, NOR gates 1 and 2 are used as INVERTERs to invert the inputs, so that the final output is $x = \overline{\overline{A} + \overline{B}}$, which can be simplified to $x = A \cdot B$ by use of DeMorgan's theorem.

Since any of the Boolean operations can be implemented using only NAND gates, any logic circuit can be constructed using only NAND gates. The same is true for NOR gates. This characteristic of NAND and NOR gates can be very useful in logic-circuit design, as Example 3-18 illustrates.

EXAMPLE 3-18

In a certain manufacturing process, a conveyor belt will shut down whenever specific conditions occur. These conditions are monitored and reflected

by the states of four logic signals as follows: signal A will be HIGH whenever the conveyor belt speed is too fast; signal B will be HIGH whenever the collection bin at the end of the belt is full; signal C will be HIGH when the belt tension is too high; signal D will be HIGH when the manual override is off.

A logic circuit is needed to generate a signal x that will go HIGH whenever conditions A and B exist simultaneously or whenever conditions C and D exist simultaneously. Clearly, the logic expression for x will be $x = AB + CD$. The circuit is to be implemented with a minimum number of ICs. The TTL integrated circuits shown in Figure 3-31 are available. Each IC is a *quad*, which means that it contains *four* identical gates on one chip.

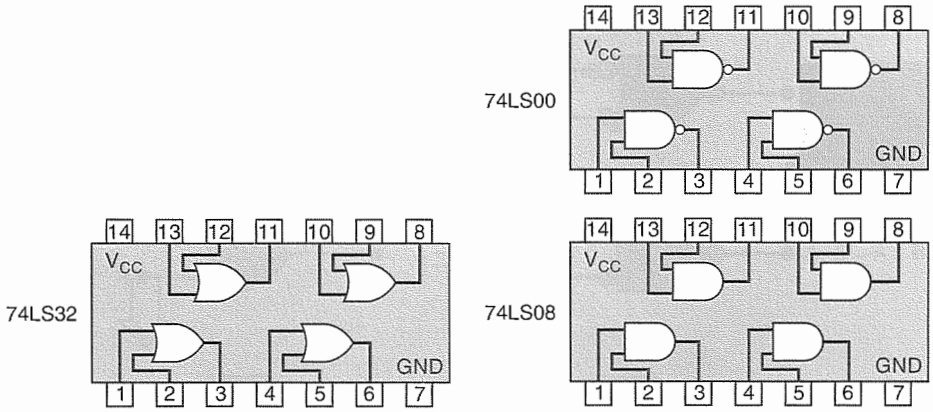


FIGURE 3-31 ICs available for Example 3-18.

Solution

The straightforward method for implementing the given expression uses two AND gates and an OR gate, as shown in Figure 3-32(a). This implementation uses two gates from the 74LS08 IC and a single gate from the 74LS32 IC. The numbers in parentheses at each input and output are the pin numbers of the respective IC. These are always shown on any logic-circuit wiring diagram. For our purposes, most logic diagrams will not show pin numbers unless they are needed in the description of circuit operation.

Another implementation can be accomplished by taking the circuit of Figure 3-32(a) and replacing each AND gate and OR gate by its equivalent NAND gate implementation from Figure 3-29. The result is shown in Figure 3-32(b).

At first glance, this new circuit looks as if it requires seven NAND gates. However, NAND gates 3 and 5 are connected as INVERTERs in series and can be eliminated from the circuit because they perform a double inversion of the signal out of NAND gate 1. Similarly, NAND gates 4 and 6 can be eliminated. The final circuit, after eliminating the double INVERTERs, is drawn in Figure 3-32(c).

This final circuit is more efficient than the one in Figure 3-32(a) because it uses three two-input NAND gates that can be implemented from one IC, the 74LS00.

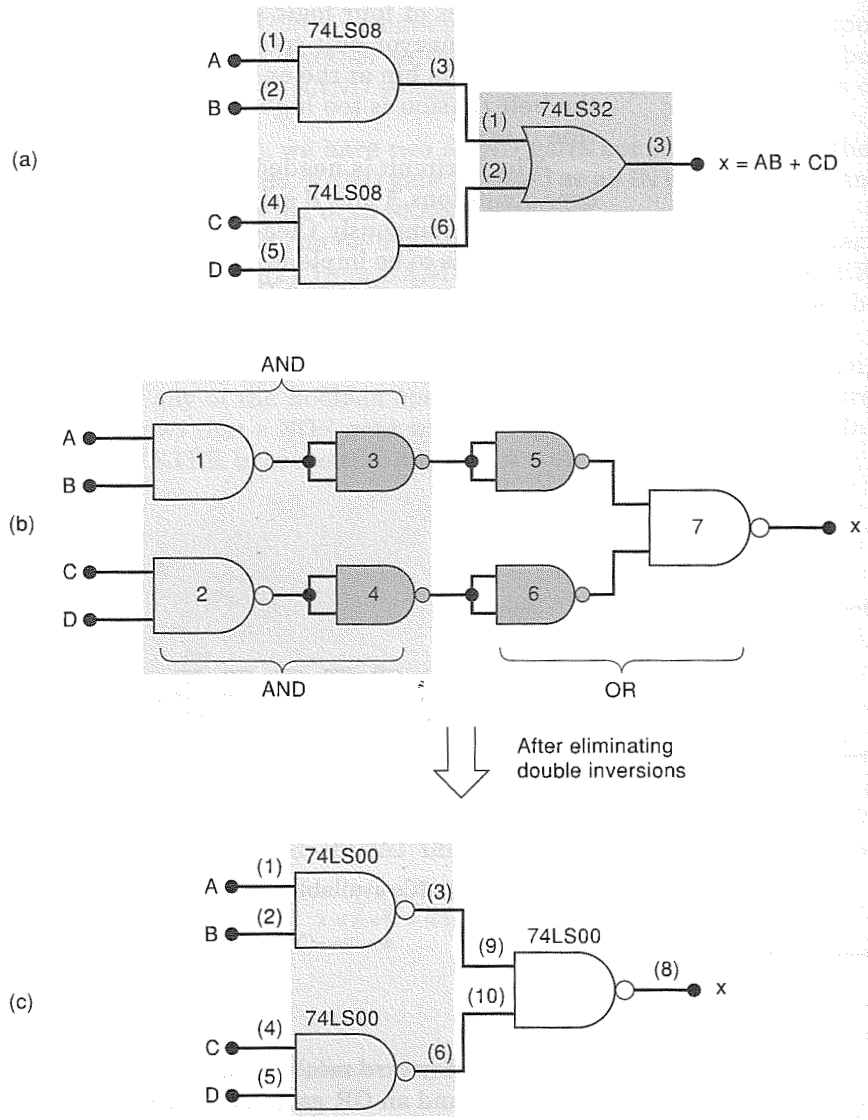


FIGURE 3-32 Possible implementations for Example 3-18.

REVIEW QUESTIONS

1. How many different ways do we now have to implement the inversion operation in a logic circuit?
2. Implement the expression $x = (A + B)(C + D)$ using OR and AND gates. Then implement the expression using only NOR gates by converting each OR and AND gate to its NOR implementation from Figure 3-30. Which circuit is more efficient?
3. Write the output expression for the circuit of Figure 3-32(c), and use DeMorgan's theorems to show that it is equivalent to the expression for the circuit of Figure 3-32(a).

3-13 ALTERNATE LOGIC-GATE REPRESENTATIONS

We have introduced the five basic logic gates (AND, OR, INVERTER, NAND, and NOR) and the standard symbols used to represent them on logic-circuit diagrams. Although you may find that some circuit diagrams still use these

standard symbols exclusively, it has become increasingly more common to find circuit diagrams that utilize **alternate logic symbols** *in addition* to the standard symbols.

Before discussing the reasons for using an alternate symbol for a logic gate, we will present the alternate symbols for each gate and show that they are equivalent to the standard symbols. Refer to Figure 3-33; the left side of the illustration shows the standard symbol for each logic gate, and the right side shows the alternate symbol. The alternate symbol for each gate is obtained from the standard symbol by doing the following:

1. Invert each input and output of the standard symbol. This is done by adding bubbles (small circles) on input and output lines that do not have bubbles and by removing bubbles that are already there.
2. Change the operation symbol from AND to OR, or from OR to AND. (In the special case of the INVERTER, the operation symbol is not changed.)

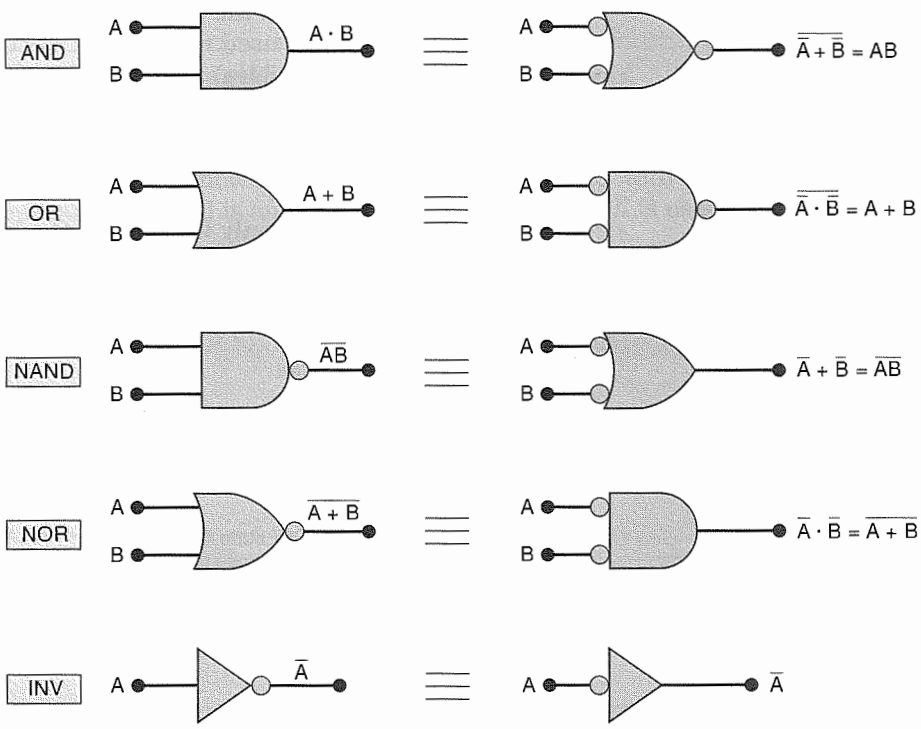


FIGURE 3-33 Standard and alternate symbols for various logic gates and inverter.

For example, the standard NAND symbol is an AND symbol with a bubble on its output. Following the steps outlined above, remove the bubble from the output, and add a bubble to each input. Then change the AND symbol to an OR symbol. The result is an OR symbol with bubbles on its inputs.

We can easily prove that this alternate symbol is equivalent to the standard symbol by using DeMorgan's theorems and recalling that the bubble represents an inversion operation. The output expression from the standard NAND symbol is $\overline{AB} = \bar{A} + \bar{B}$, which is the same as the output expression for the alternate symbol. This same procedure can be followed for each pair of symbols in Figure 3-33.

Several points should be stressed regarding the logic symbol equivalences:

1. The equivalences can be extended to gates with *any* number of inputs.
2. None of the standard symbols have bubbles on their inputs, and all the alternate symbols do.

3. The standard and alternate symbols for each gate represent the same physical circuit; *there is no difference in the circuits represented by the two symbols.*
4. NAND and NOR gates are inverting gates, and so both the standard and the alternate symbols for each will have a bubble on *either* the input or the output. AND and OR gates are *noninverting* gates, and so the alternate symbols for each will have bubbles on *both* inputs and output.

Logic-Symbol Interpretation

Each of the logic-gate symbols of Figure 3-33 provides a unique interpretation of how the gate operates. Before we can demonstrate these interpretations, we must first establish the concept of **active logic levels.**

When an input or output line on a logic circuit symbol has *no bubble* on it, that line is said to be **active-HIGH.** When an input or output line *does* have a *bubble* on it, that line is said to be **active-LOW.** The presence or absence of a bubble, then, determines the active-HIGH/active-LOW status of a circuit's inputs and output, and is used to interpret the circuit operation.

To illustrate, Figure 3-34(a) shows the standard symbol for a NAND gate. The standard symbol has a bubble on its output and no bubbles on its inputs. Thus, it has an active-LOW output and active-HIGH inputs. The logic operation represented by this symbol can therefore be interpreted as follows:

The output goes LOW only when *all* of the inputs are HIGH.

Note that this says that the output will go to its active state only when *all* of the inputs are in their active states. The word *all* is used because of the AND symbol.

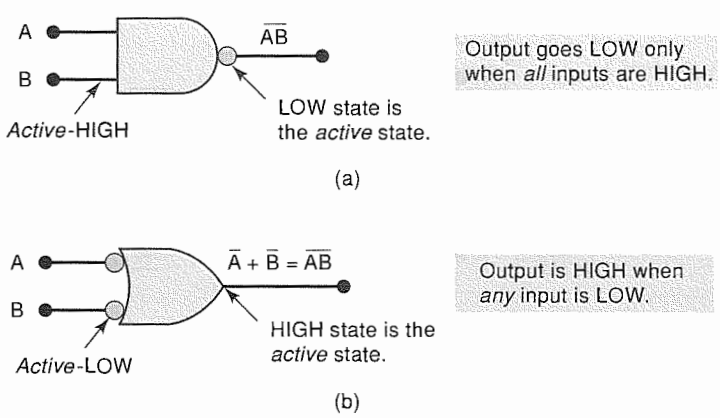


FIGURE 3-34 Interpretation of the two NAND gate symbols.

The alternate symbol for a NAND gate shown in Figure 3-34(b) has an active-HIGH output and active-LOW inputs, and so its operation can be stated as follows:

The output goes HIGH when *any input* is LOW.

This says that the output will be in its active state whenever *any* of the inputs is in its active state. The word *any* is used because of the OR symbol.

With a little thought, you can see that the two interpretations for the NAND symbols in Figure 3-34 are different ways of saying the same thing.

Summary

At this point you are probably wondering why there is a need to have two different symbols and interpretations for each logic gate. We hope the reasons will become clear after reading the next section. For now, let us summarize the important points concerning the logic-gate representations.

1. To obtain the alternate symbol for a logic gate, take the standard symbol and change its operation symbol (OR to AND, or AND to OR), and change the bubbles on both inputs and output (i.e., delete bubbles that are present, and add bubbles where there are none).
2. To interpret the logic-gate operation, first note which logic state, 0 or 1, is the active state for the inputs and which is the active state for the output. Then realize that the output's active state is produced by having *all* of the inputs in their active state (if an AND symbol is used) or by having *any* of the inputs in its active state (if an OR symbol is used).

EXAMPLE 3-19

Give the interpretation of the two OR gate symbols.

Solution

The results are shown in Figure 3-35. Note that the word *any* is used when the operation symbol is an OR symbol and the word *all* is used when it includes an AND symbol.

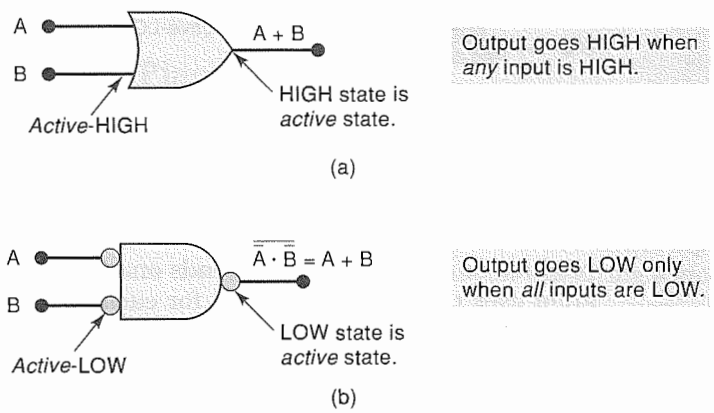


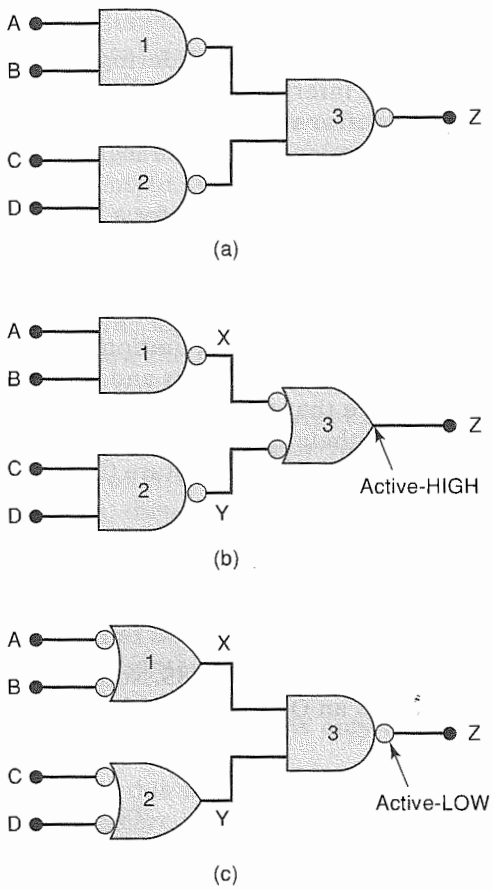
FIGURE 3-35 Interpretation of the two OR gate symbols.

REVIEW QUESTIONS

1. Write the interpretation of the operation performed by the standard NOR gate symbol in Figure 3-33.
2. Repeat question 1 for the alternate NOR gate symbol.
3. Repeat question 1 for the alternate AND gate symbol.
4. Repeat question 1 for the standard AND gate symbol.

3-14 WHICH GATE REPRESENTATION TO USE

Some logic-circuit designers and some textbooks use only the standard logic-gate symbols in their circuit schematics. While this practice is logically correct, it does nothing to make the circuit operation easier to follow. Proper use of the alternate gate symbols in the circuit diagram can make the circuit operation



A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

(d)

FIGURE 3-36 (a) Original circuit using standard NAND symbols; (b) equivalent representation where output Z is active-HIGH; (c) equivalent representation where output Z is active-LOW; (d) truth table.

much clearer. This can be illustrated by considering the example shown in Figure 3-36.

The circuit in Figure 3-36(a) contains three NAND gates connected to produce an output Z that depends on inputs A, B, C, and D. The circuit diagram uses the standard symbol for each of the NAND gates. While this diagram is logically correct, it does not facilitate an understanding of how the circuit functions. The circuit representations given in Figures 3-36(b) and (c), however, can be analyzed more easily to determine the circuit operation.

The representation of Figure 3-36(b) is obtained from the original circuit diagram by replacing NAND gate 3 with its alternate symbol. In this diagram, output Z is taken from a NAND gate symbol that has an active-HIGH output. Thus, we can say that Z will go HIGH when either X or Y is LOW. Now, since X and Y each appear at the output of NAND symbols having active-LOW outputs, we can say that X will go LOW only if $A = B = 1$, and Y will go LOW only if $C = D = 1$. Putting this all together, we can describe the circuit operation as follows:

> **Output Z will go HIGH whenever either $A = B = 1$ or $C = D = 1$ (or both).**

This description can be translated to truth-table form by setting $Z = 1$ for those cases where $A = B = 1$ and for those cases where $C = D = 1$. For all other cases, Z is made a 0. The resultant truth table is shown in Figure 3-36(d).

The representation of Figure 3-36(c) is obtained from the original circuit diagram by replacing NAND gates 1 and 2 by their alternate symbols. In this

equivalent representation, the Z output is taken from a NAND gate that has an active-LOW output. Thus, we can say that Z will go LOW only when $X = Y = 1$. Because X and Y are active-HIGH outputs, we can say that X will be HIGH when either A or B is LOW, and Y will be HIGH when either C or D is LOW. Putting this all together, we can describe the circuit operation as follows:

Output Z will go LOW only when A or B is LOW and C or D is LOW.

This description can be translated to truth-table form by making $Z = 0$ for all cases where at least one of the A or B inputs is LOW at the same time that at least one of the C or D inputs is LOW. For all other cases, Z is made a 1. The resultant truth table is the same as that obtained for the circuit diagram of Figure 3-36(b).

Which Circuit Diagram Should Be Used?

The answer to this question depends on the particular function being performed by the circuit output. If the circuit is being used to cause some action (e.g., turn on an LED or activate another logic circuit) when output Z goes to the 1 state, then we say that Z is to be active-HIGH, and the circuit diagram of Figure 3-36(b) should be used. On the other hand, if the circuit is being used to cause some action when Z goes to the 0 state, then Z is to be active-LOW, and the diagram of Figure 3-36(c) should be used.

Of course, there will be situations where *both* output states are used to produce different actions and either one can be considered to be the active state. For these cases, either circuit representation can be used.

Bubble Placement

Refer to the circuit representation of Figure 3-36(b) and note that the symbols for NAND gates 1 and 2 were chosen to have active-LOW outputs to match the active-LOW inputs of NAND gate 3. Refer to the circuit representation of Figure 3-36(c) and note that the symbols for NAND gates 1 and 2 were chosen to have active-HIGH outputs to match the active-HIGH inputs of NAND gate 3. This leads to the following general rule for preparing logic-circuit schematics:

Whenever possible, choose gate symbols so that bubble outputs are connected to bubble inputs, and nonbubble outputs to nonbubble inputs.

The following examples will show how this rule can be applied.

EXAMPLE 3-20

The logic circuit in Figure 3-37(a) is being used to activate an alarm when its output Z goes HIGH. Modify the circuit diagram so that it represents the circuit operation more effectively.

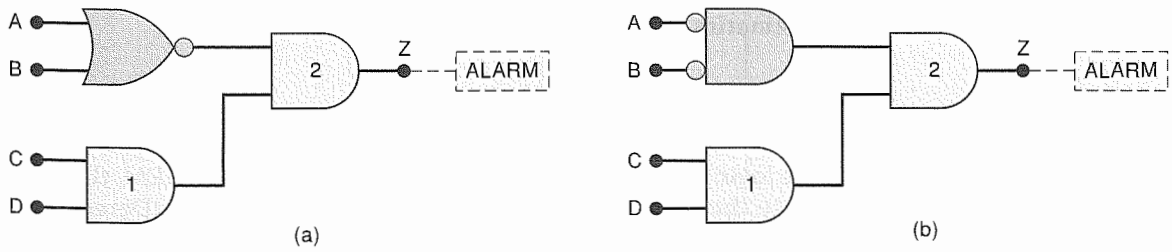


FIGURE 3-37 Example 3-20.

Solution

Because Z = 1 will activate the alarm, Z is to be active-HIGH. Thus, the AND gate 2 symbol does not have to be changed. The NOR gate symbol should be changed to the alternate symbol with a nonbubble (active-HIGH) output to match the nonbubble input of AND gate 2, as shown in Figure 3-37(b). Note that the circuit now has nonbubble outputs connected to the nonbubble inputs of gate 2.

EXAMPLE 3-21

When the output of the logic circuit in Figure 3-38(a) goes LOW, it activates another logic circuit. Modify the circuit diagram to represent the circuit operation more effectively.

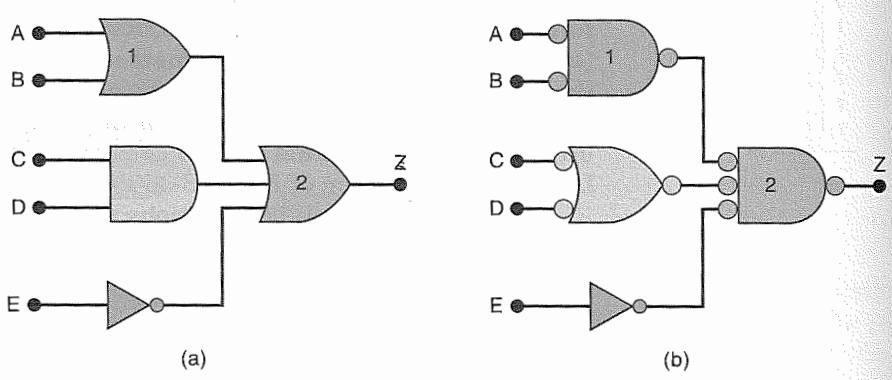


FIGURE 3-38
Example 3-21.

Solution

Because Z is to be active-LOW, the symbol for OR gate 2 must be changed to its alternate symbol, as shown in Figure 3-38(b). The new OR gate 2 symbol has bubble inputs, and so the AND gate and OR gate 1 symbols must be changed to bubbled outputs, as shown in Figure 3-38(b). The INVERTER already has a bubble output. Now the circuit has all bubble outputs connected to bubble inputs of gate 2.

Analyzing Circuits

When a logic-circuit schematic is drawn using the rules we followed in these examples, it is much easier for an engineer or technician (or student) to follow the signal flow through the circuit and to determine the input conditions that are needed to activate the output. This will be illustrated in the following examples—which, incidentally, use circuit diagrams taken from the logic schematics of an actual microcomputer.

EXAMPLE 3-22

The logic circuit in Figure 3-39 generates an output, *MEM*, that is used to activate the memory ICs in a particular microcomputer. Determine the input conditions necessary to activate *MEM*.

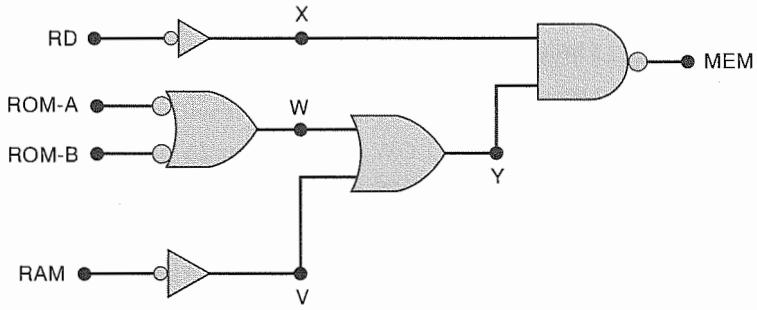


FIGURE 3-39
Example 3-22.

Solution

One way to do this would be to write the expression for *MEM* in terms of the inputs *RD*, *ROM-A*, *ROM-B*, and *RAM*, and to evaluate it for the 16 possible combinations of these inputs. While this method would work, it would require a lot more work than is necessary.

A more efficient method is to interpret the circuit diagram using the ideas we have been developing in the last two sections. These are the steps:

1. *MEM* is active-LOW, and it will go LOW only when *X* and *Y* are HIGH.
2. *X* will be HIGH only when *RD* = 0.
3. *Y* will be HIGH when either *W* or *V* is HIGH.
4. *V* will be HIGH when *RAM* = 0.
5. *W* will be HIGH when either *ROM-A* or *ROM-B* = 0.
6. Putting this all together, *MEM* will go LOW only when *RD* = 0 *and* at least one of the three inputs *ROM-A*, *ROM-B*, or *RAM* is LOW.

EXAMPLE 3-23

The logic circuit in Figure 3-40 is used to enable the liquid crystal display (LCD) of a handheld electronic device when the microcontroller is sending data to or receiving data from the LCD controller. The circuit will enable the display when *LCD* = 1. Determine the input conditions necessary to enable the LCD.

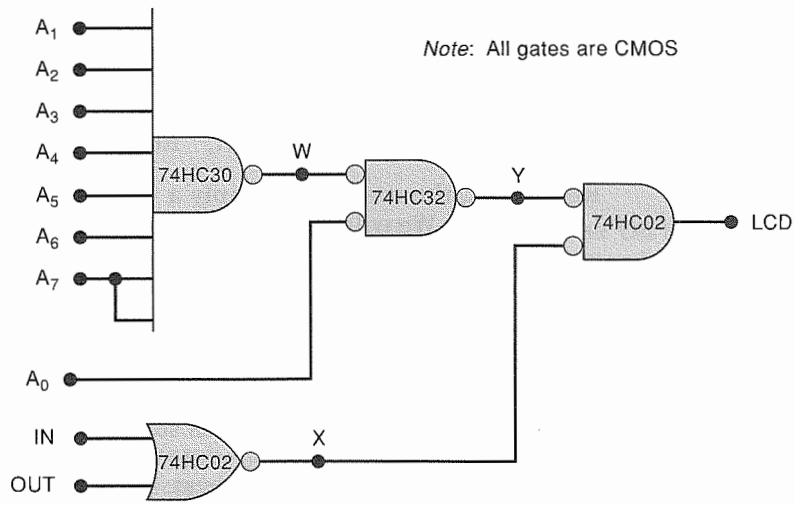


FIGURE 3-40
Example 3-23.

Solution

Once again, we will interpret the diagram in a step-by-step fashion:

1. *LCD* is active-HIGH, and it will go HIGH only when $X = Y = 0$.
2. X will be LOW when either *IN* or *OUT* is HIGH.
3. Y will be LOW only when $W = 0$ and $A_0 = 0$.
4. W will be LOW only when A_1 through A_7 are all HIGH.
5. Putting this all together, *LCD* will be HIGH when $A_1 = A_2 = A_3 = A_4 = A_5 = A_6 = A_7 = 1$ and $A_0 = 0$, and either *IN* or *OUT* or both are 1.

Note the strange symbol for the eight-input CMOS NAND gate (74HC30); also note that signal A_7 is connected to two of the NAND inputs.

Asserted Levels

We have been describing logic signals as being active-LOW or active-HIGH. For example, the output *MEM* in Figure 3-39 is active-LOW, and the output *LCD* in Figure 3-40 is active-HIGH because these are the output states that cause something to happen. Similarly, Figure 3-40 has active-HIGH inputs A_1 to A_7, and active-LOW input A_0.

When a logic signal is in its active state, it can be said to be **asserted**. For example, when we say that input A_0 is asserted, we are saying that it is in its active-LOW state. When a logic signal is not in its active state, it is said to be **unasserted**. Thus, when we say that *LCD* is unasserted, we mean that it is in its inactive state (low).

Clearly, the terms *asserted* and *unasserted* are synonymous with *active* and *inactive*, respectively:

asserted = active
unasserted = inactive

Both sets of terms are in common use in the digital field, so you should recognize both ways of describing a logic signal's active state.

Labeling Active-LOW Logic Signals

It has become common practice to use an overbar to label active-LOW signals. The overbar serves as another indication that the signal is active-LOW; of course, the absence of an overbar means that the signal is active-HIGH.

To illustrate, all of the signals in Figure 3-39 are active-LOW, and so they can be labeled as follows:

$$\overline{RD},\quad \overline{ROM\text{-}A},\quad \overline{ROM\text{-}B},\quad \overline{RAM},\quad \overline{MEM}$$

Remember, the overbar is simply a way to emphasize that these are active-LOW signals. We will employ this convention for labeling logic signals whenever appropriate.

Labeling Bistate Signals

Very often, an output signal will have two active states; that is, it will have one important function in the HIGH state and another in the LOW state. It is customary to label such signals so that both active states are apparent. A common example is the read/write signal, $RD/\overline{WR}$, which is interpreted as follows: when this signal is HIGH, the read operation (RD) is performed; when it is LOW, the write operation ($\overline{WR}$) is performed.

REVIEW QUESTIONS

1. Use the method of Examples 3-22 and 3-23 to determine the input conditions needed to activate the output of the circuit in Figure 3-37(b).
2. Repeat question 1 for the circuit of Figure 3-38(b).
3. How many NAND gates are shown in Figure 3-39?
4. How many NOR gates are shown in Figure 3-40?
5. What will be the output level in Figure 3-38(b) when all of the inputs are asserted?
6. What inputs are required to assert the alarm output in Figure 3-37(b)?
7. Which of the following signals is active-LOW: RD, $\overline{W}$, $R/\overline{W}$?

3-15 PROPAGATION DELAY

Propagation delay can be simply defined as the time it takes for a system to produce the appropriate output after it receives an input. Think about a typical vending machine. You put your money into the machine and press a button to make a selection. You do not receive the product immediately; it takes a little time for the product to be moved off the rack and dropped to the vending door. This is propagation delay. A biologic example can be found in our reflexes. From the time you see brake lights on the car in front of you until you get your foot on the brakes, there is a measurable delay or reaction time.

Real digital circuits have a measurable propagation delay time as well. The reasons will become clearer when you study the real characteristics of circuits and semiconductors (transistors) instead of just their idealized operation. Chapter 8 will provide more information about the inner workings of logic ICs. An AND gate, as shown in Figure 3-41(a), serves as an example that propagation delay does exist and that it can be measured.

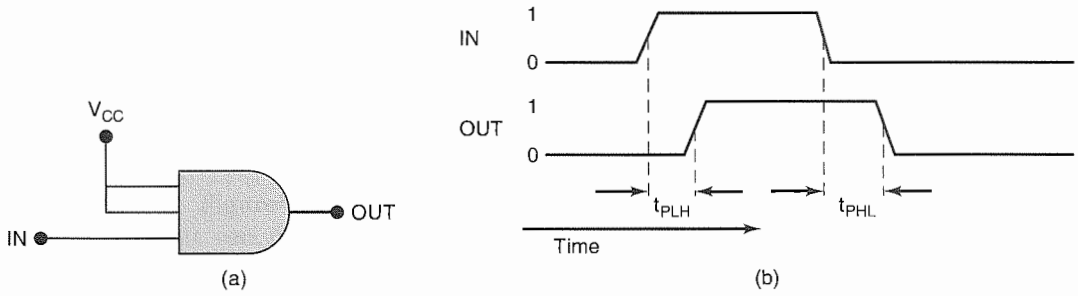


FIGURE 3-41 Measuring propagation delay in a logic gate.

When the IN signal goes HIGH, it causes the OUT signal to go HIGH a short time later. Likewise, when the IN signal goes LOW, it causes the OUT signal to go LOW a short time later. Two things are important to note from the timing diagram in Figure 3-41(b):

1. Transitions are not truly vertical (instantaneous) so we measure from the 50% point on the input to the 50% point on the output.
2. The time it takes to make the *output* go HIGH is not necessarily the same as the time to make the *output* go LOW. These delay times are called t_{PLH} (time propagation LOW to HIGH) and t_{PHL} (time propagation HIGH to LOW).

The speed of a logic circuit is related to this characteristic of propagation delay. Whatever part is chosen to implement the logic circuit will have a data sheet that states the value of propagation delay. This information is used to assure that the circuit can operate fast enough for the application.

REVIEW QUESTIONS

1. Why are the transitions not vertical when measuring propagation delay?
2. Where are time measurements taken when transitions are not vertical?
3. What is the parameter that measures the time after the input changes until the output can switch from HIGH to LOW?
4. What is the parameter that measures the time after the input changes until the output can switch from LOW to HIGH?

3-16 SUMMARY OF METHODS TO DESCRIBE LOGIC CIRCUITS

The topics we have covered so far in this chapter have all centered around just three simple logic functions that we refer to as AND, OR, and NOT. The concept is not new to anyone because we all use these logical functions every day as we make decisions. Here are some logical examples of how you might think: If it is raining OR the newspaper says that it could rain, then I will take my umbrella; If I get my paycheck today AND I make it to the bank, then I will have money to spend this evening; If I have a passing grade in lecture AND I have NOT failed in lab, then I will pass my digital class. At this point, you may be wondering why we have spent so much effort in describing such familiar concepts. The answer can be summed up in two key points:

1. We must be able to represent these logical decisions.
2. We must be able to combine these logic functions and implement a decision-making system.

We have learned how to represent each of the basic logic functions using:

Logical statements in our own language
Truth tables
Traditional graphic logic symbols
Boolean algebra expressions
Timing diagrams

EXAMPLE 3-24

The following English expression describes the way a logic circuit needs to operate in order to drive a seatbelt warning indicator in a car.

If the driver is present AND the driver is NOT buckled up AND the ignition switch is on, THEN turn on the warning light.

Describe the circuit using Boolean algebra, schematic diagrams with logic symbols, truth tables, and timing diagrams.

Solution

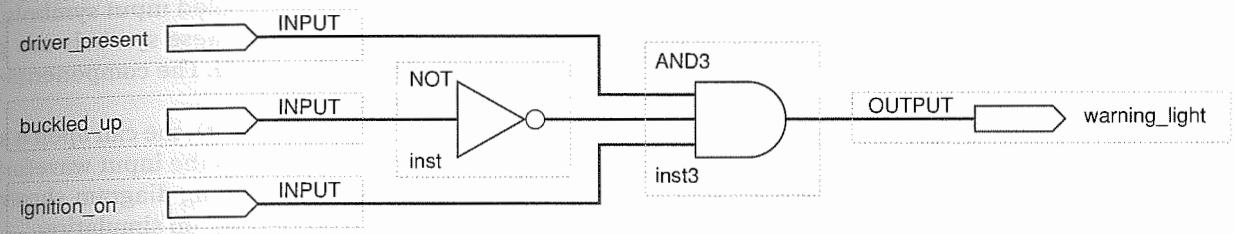
See Figure 3-42.

Boolean expression

warning_light = driver_present • $\overline{\text{buckled_up}}$ • ignition_on

(a)

Schematic diagram



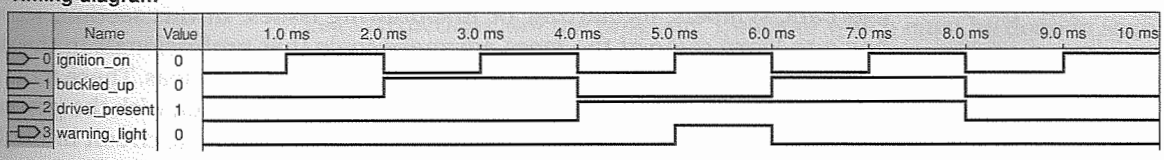
(b)

Truth table

driver_present	buckled_up	ignition_on	warning_light
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

(c)

Timing diagram



(d)

FIGURE 3-42 Methods of describing logic circuits: (a) Boolean expression; (b) schematic diagram; (c) truth table; (d) timing diagram.

Figure 3-42 shows four different ways of representing the logic circuit that was described in English as the problem statement of Example 3-24. There are many other ways in which we could represent the logic of this decision. As an example we could dream up an entirely new set of graphic symbols, or state the logical relationship in French or Japanese. Of course, we cannot cover all the possible ways of describing a logic circuit, but we must understand the most common methods to be able to communicate with others in this profession. Furthermore, certain situations are easier to describe using one method over another. In some cases, a picture is worth a thousand words, and in other cases words are concise enough and are more easily communicated to others. The important point here is that we need ways to describe and communicate the operation of digital systems.

Many tools have been developed to allow a designer to enter a circuit description into a computer for the purpose of documenting the circuit, simulating the circuit, and ultimately creating a functional circuit. The tool we recommend is from Altera Corporation, one of the top digital circuit suppliers in the world. Its Quartus II software is available free of charge and can be

downloaded from its web site. It is easy to learn how to use, especially through the use of the tutorials that are available to owners of this book at www.pearsonhighered.com/electronics. Quartus II offers a way for you to describe a circuit by drawing a logic diagram. The logic diagram in Figure 3-42(b) is a block description file (.bdf) that was generated using the Quartus II software. Notice that this diagram is made up of labeled input symbols, labeled output symbols, and logic gate symbols. All of these symbols are provided in a library of components included in Quartus II. The components are easily connected using a wire drawing tool.

After the designer draws a block description file (.bdf), she can open a simulation file in the form of a timing diagram. She creates the input waveforms, and the simulator draws the output waveform. The timing diagram shown in Figure 3-42(d) is an example of a Quartus II timing diagram simulation.

REVIEW QUESTIONS

1. Name five ways to describe the operation of logic circuits.
2. Name two tools available in Quartus II software.

3-17 DESCRIPTION LANGUAGES VERSUS PROGRAMMING LANGUAGES*

Recent trends in the field of digital systems are favoring text-based language description of digital circuits. You probably noticed that each description method in Figure 3-42 offers challenges to computer entry, whether it is due to overbars, symbols, formatting, or line-drawing issues. In this section, we will begin to learn some of the more advanced tools that professionals in the digital field use to describe the circuits that implement their ideas. These tools are referred to as **hardware description languages (HDLs)**. Even with the powerful computers we have today, it is not possible to describe a logic circuit in English prose and expect the computer to understand what you mean. Computers need a more rigidly defined language. We will focus on two languages in this text: **Altera hardware description language (AHDL)** and **very high speed integrated circuit (VHSIC) hardware description language (VHDL)**.

VHDL and AHDL

VHDL is not a new language. It was developed by the Department of Defense in the early 1980s as a concise way to document the designs in the very high speed integrated circuit (VHSIC) program. Appending HDL onto this acronym was too much, even for the military, and so the language was abbreviated to VHDL. Computer programs were developed to take the VHDL language files and simulate the operation of the circuits. With the growth of complex programmable logic devices in digital systems, VHDL has evolved into one of the primary high-level hardware description languages for designing and implementing digital circuits (synthesis). The language has been standardized by the IEEE, making it universally appealing for engineers as well as the makers of software tools that translate designs into the bit patterns used to program actual devices.

*All sections covering hardware description languages may by skipped without loss of continuity in Chapters 3–13.

AHDL is a language that the Altera Corporation developed to provide a convenient way to configure the logic devices that they offer. Altera was one of the first companies to introduce logic devices that can be reconfigured electronically. These devices are called **programmable logic devices (PLDs)**. Unlike VHDL, AHDL is not intended to be used as a universal language for describing any logic circuit. It is intended to be used for programming complex digital systems into Altera PLDs in a language that is generally perceived to be easier to learn yet very similar to VHDL. It also has features that take full advantage of the architecture of Altera devices. All of the examples in this text will use the Altera Quartus II software to develop both AHDL and VHDL design files. You will see the advantage of using Altera's development system for both languages when you program an actual device. The Altera system makes circuit development very easy and contains all the necessary tools to translate from the HDL design file to a file ready to load into an Altera PLD. It also allows you to develop building blocks using schematic entry, AHDL, VHDL, and other methods and then interconnect them to form a complete system.

Other HDLs are available that are more suitable for programming simple programmable logic devices. You will find any of these languages easy to use after learning the basics of AHDL or VHDL as covered in this text.

Computer Programming Languages

It is important to distinguish between hardware description languages intended to describe the hardware configuration of a circuit and programming languages that represent a sequence of instructions intended to be carried out by a computer to accomplish some task. In both cases, we use a *language* to *program* a device. However, computers are complex digital systems that are made up of logic circuits. Computers operate by following a laundry list of tasks (i.e., instructions, or "the program"), each of which must be done in sequential order. The speed of operation is determined by how fast the computer can execute each instruction. For example, if a computer were to respond to four different inputs, it would require at least four separate instructions (sequential tasks) to detect and identify which input changed state. A digital logic circuit, on the other hand, is limited in its speed only by how quickly the circuitry can change the outputs in response to changes in the inputs. It is monitoring all inputs **concurrently** (at the same time) and responding to any changes.

The following analogy will help you understand the difference between computer operation and digital logic circuit operation and the role of language elements used to describe what the systems do. Consider the challenge of describing what is done to an Indy 500 car during a pit stop. If a single person performed all the necessary tasks one at a time, he or she would need to be very fast at each task. This is the way a computer operates: one task at a time but very quickly. Of course, at Indy, there is an entire pit crew that swarms the car, and each member of the crew does his or her task while the others do theirs. All crew members operate concurrently, like the elements of a digital circuit. Now consider how you would describe to someone else what is being done to the Indy car during the pit stop using (1) the individual-mechanic approach or (2) the pit-crew approach. Wouldn't the two English language descriptions of what is being done sound very similar? As we will see, the languages used to describe digital hardware (HDL) are very similar to languages that describe computer programs (e.g., BASIC, C, JAVA), even though the resulting implementation operates quite differently. Knowledge of any of these computer programming languages is not necessary to understand HDL. The important thing is that when you have learned both an HDL and a computer language, you must understand their different roles in digital systems.

EXAMPLE 3-25

Compare the operation of a computer and a logic circuit in performing the simple logical operation of $y = AB$.

Solution

The logic circuit is a simple AND gate. The output y will be HIGH within approximately 10 nanoseconds of the point when A and B are HIGH simultaneously. Within approximately 10 nanoseconds after either input goes LOW, the output y will be LOW.

The computer must run a program of instructions that makes decisions. Suppose each instruction takes 20 ns (that's pretty fast!). Each shape in the flowchart shown in Figure 3-43 represents one instruction. Clearly, it will take a minimum of two or three instructions (40–60 ns) to respond to changes in the inputs.

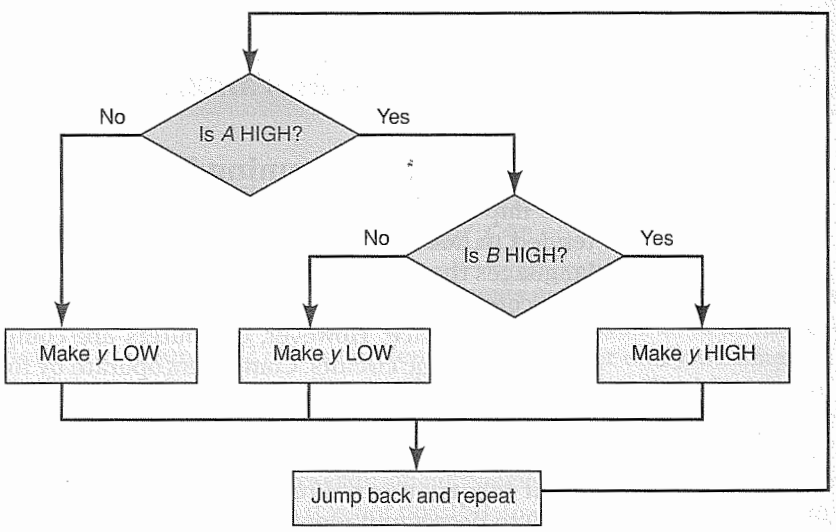


FIGURE 3-43 Decision process of a computer program.

REVIEW QUESTIONS

1. What does HDL stand for?
2. What is the purpose of an HDL?
3. What is the purpose of a computer programming language?
4. What is the key difference between HDL and computer programming languages?

3-18 IMPLEMENTING LOGIC CIRCUITS WITH PLDs

Many digital circuits today are implemented using programmable logic devices (PLDs). These devices are not like microcomputers or microcontrollers that "run" the program of instructions. Instead, they are configured electronically, and their internal circuits are "wired" together electronically to form a logic circuit. This programmable wiring can be thought of as thousands of connections that are either connected (1) or not connected (0). Figure 3-44 shows a small area of programmable connections. Each intersection between a row (horizontal wire) and a column (vertical wire) is a

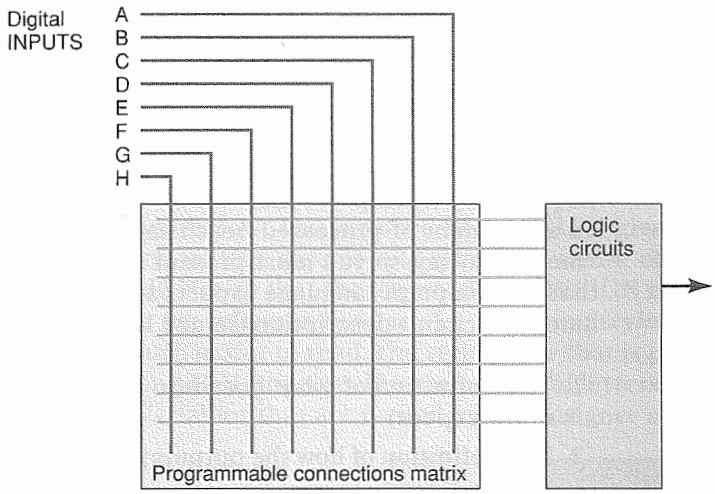


FIGURE 3-44 Configuring hardware connections with programmable logic devices.

programmable connection. You can imagine how difficult it would be to try to configure these devices by placing 1s and 0s in a grid manually (which is how they did it back in the 1970s).

The role of the hardware description language is to provide a concise and convenient way for the designer to describe the operation of the circuit in a format that a personal computer can handle and store conveniently. The computer runs a special software application called a **compiler** to translate from the hardware description language into the grid of 1s and 0s that can be loaded into the PLD. If a person can master the higher-level hardware description language, it actually makes programming the PLDs much easier than trying to use Boolean algebra, schematic drawings, or truth tables. In much the same way that you learned the English language, we will start by expressing simple things and gradually learn the more complicated aspects of these languages. Our objective is to learn enough of HDL to be able to communicate with others and perform simple tasks. A full understanding of all the details of these languages is beyond the scope of this text and can really be mastered only by regular use.

In the sections throughout this book that cover the HDLs, we will present both AHDL and VHDL in a format that allows you to skip over one language and concentrate on the other without missing important information. Of course, this setup means there will be some redundant information presented if you choose to read about both languages. We feel this redundancy is worth the extra effort to provide you with the flexibility of focusing on either of the two languages or learning both by comparing and contrasting similar examples. The recommended way to use the text is to focus on one language. It is true that the easiest way to become bilingual, and fluent in both languages, is to be raised in an environment where both languages are spoken routinely. It is also very easy, however, to confuse details, so we will keep the specific examples separate and independent. We hope this format provides you with the opportunity to learn one language now and then use this book as a reference later in your career should you need to pick up the second language.

REVIEW QUESTIONS

1. What does PLD stand for?
2. How are the circuits reconfigured electronically in a PLD?
3. What does a compiler do?

3-19 HDL FORMAT AND SYNTAX

Any language has its unique properties, similarities to other languages, and its proper syntax. When we study grammar in school, we learn conventions such as the order of words as elements in a sentence and proper punctuation. This is referred to as the **syntax** of language. A language designed to be interpreted by a computer must follow strict rules of syntax. A computer is just an assortment of processed beach sand and wire that has no idea what you "meant" to say, so you must present the instructions using the exact syntax that the computer language expects and understands. The basic format of any hardware circuit description (in any language) involves two vital elements:

1. The definition of what goes into it and what comes out of it (i.e., input/output specs)
2. The definition of how the outputs respond to the inputs (i.e., its operation)

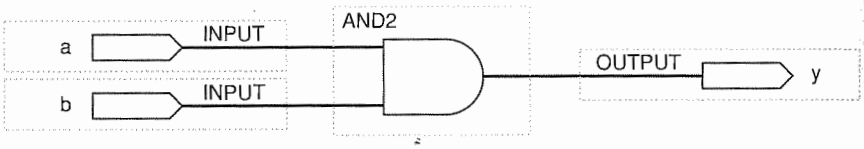


FIGURE 3-45 A schematic diagram description.

A circuit schematic diagram such as Figure 3-45 can be read and understood by a competent engineer or technician because both would understand the meaning of each symbol in the drawing. If you understand how each element works and how the elements are connected to each other, you can understand how the circuit operates. On the left side of the diagram is the set of inputs, and on the right is the set of outputs. The symbols in the middle define its operation. The text-based language must convey the same information. All HDLs use the format shown in Figure 3-46.

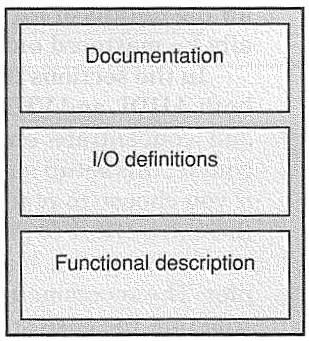


FIGURE 3-46 Format of HDL files.

In a text-based language, the circuit being described must be given a name. The inputs and outputs (sometimes called ports) must be assigned names and defined according to the nature of the port. Is it a single bit from a toggle switch? Or is it a four-bit number coming from a keypad? The text-based language must somehow convey the nature of these inputs and outputs. The **mode** of a port defines whether it is input, output, or both. The **type** refers to the number of bits and how those bits are grouped and interpreted. If the *type* of input is a single bit, then it can have only two possible values: 0 and 1. If the type of input is a four-bit binary number from a keypad, it can have any one of 16 different values (0000_2–1111_2). The type determines the range of possible values. The definition of the circuit's operation in a text-based

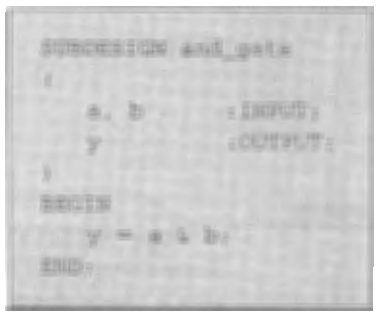
language is contained in a set of statements that follow the circuit input/output (I/O) definition. The following two sections describe the very simple circuit of Figure 3-45 and illustrate the critical elements of AHDL and VHDL.

BOOLEAN DESCRIPTION USING AHDL

AHDL

Refer to Figure 3-47. The keyword **SUBDESIGN** gives a name to the circuit block, which in this case is *and_gate*. The name of the file must also be and_gate.tdf. Notice that the keyword SUBDESIGN is capitalized. This is not required by the software, but use of a consistent style in capitalization makes the code much easier to read. The style guide that is provided with the Altera compiler for AHDL suggests the use of capital letters for the keywords in the language. Variables that are named by the designer should be lowercase.

FIGURE 3-47 Essential elements in AHDL.



The SUBDESIGN section defines the inputs and outputs of the logic circuit block. Something must enclose the circuit that we are trying to describe, much the same way that a block diagram encloses everything that makes up that part of the design. In AHDL, this input/output definition is enclosed in parentheses. The list of variables used for inputs to this block are separated by commas and followed by :INPUT;. In AHDL, the single-bit type is assumed unless the variable is designated as multiple bits. The single-output bit is declared with the mode :OUTPUT;. We will learn the proper way to describe other types of inputs, outputs, and variables as we need to use them.

The set of statements that describe the operation of the AHDL circuit are contained in the logic section between the keywords BEGIN and END. END must be followed by a semicolon, similar to the way you would always end a paragraph with a period. In this example, the operation of the hardware is described by a very simple Boolean algebra equation that states that the output (y) is assigned (=) the logic level produced by a AND b. This Boolean algebra equation is referred to as a **concurrent assignment statement**. Any statements (there is only one in this example) between BEGIN and END are evaluated constantly and concurrently. The order in which they are listed makes no difference. The basic Boolean operators are:

&	AND
#	OR
!	NOT
$	XOR

REVIEW QUESTIONS

1. What appears inside the parentheses ( ) after SUBDESIGN?
2. What appears between BEGIN and END?

VHDL

BOOLEAN DESCRIPTION USING VHDL

Refer to Figure 3-48. The keyword **ENTITY** gives a name to the circuit block, which in this case is and_gate. Notice that the keyword ENTITY is capitalized but and_gate is not. This is not required by the software, but use of a consistent style in capitalization makes the code much easier to read. The style guide provided with the Altera compiler for VHDL suggests using capital letters for the keywords in the language. Variables that are named by the designer should be lowercase.

FIGURE 3-48 Essential elements in VHDL.

```
ENTITY and_gate IS
PORT (   a, b    :IN BIT;
         y       :OUT BIT);
END and_gate;
ARCHITECTURE ckt OF and_gate IS
BEGIN
       y <= a AND b;
END ckt;
```

The ENTITY declaration can be thought of as a block description. Something must enclose the circuit we are trying to describe, much the same way a block diagram encloses everything that makes up that part of the design. In VHDL, the keyword PORT tells the compiler that we are defining inputs and outputs to this circuit block. The names used for inputs (separated by commas) are listed, ending with a colon and a description of the mode and type of input (:IN BIT;). In VHDL, the **BIT** description tells the compiler that each variable in the list is a single bit. We will learn the proper way to describe other types of inputs, outputs, and variables as we need to use them. The line containing END and_gate; terminates the ENTITY declaration.

The **ARCHITECTURE** declaration is used to describe the operation of everything inside the block. The designer makes up a name for this architectural description of the inner workings of the ENTITY block (*ckt* in this example). Every ENTITY must have at least one ARCHITECTURE associated with it. The words OF and IS are keywords in this declaration. The body of the architecture description is enclosed between the BEGIN and END keywords. END is followed by the name that has been assigned to this architecture. This line must be punctuated with a semicolon, similar to the way you end a paragraph with a period. Within the body (between BEGIN and END) is the description of the block's operation. In this example, the operation of the hardware is described by a very simple Boolean algebra equation that states that the output (y) is assigned (<=) the logic level produced by a AND b. This is referred to as a **concurrent assignment statement**, which means that all the statements (there is only one in this example) between BEGIN and END are evaluated constantly and concurrently. The order in which they are listed makes no difference.

REVIEW QUESTIONS

1. What is the role of the ENTITY declaration?
2. Which key section defines the operation of the circuit?
3. What is the assignment operator used to give a value to a logic signal?

3-20 INTERMEDIATE SIGNALS

In many designs, there is a need to define signal points "inside" the circuit block. They are points in the circuit that are neither inputs nor outputs for the block but may be useful as a reference point. It may be a signal that needs to be connected to many other places within the block. In an analog or digital schematic diagram, they would be called test points or *nodes*. In an HDL, they are referred to as **buried nodes** or **local signals**. Figure 3-49 shows a very simple circuit that uses an intermediate signal named *m*. In the HDL, these nodes (signals) are not defined with the inputs and outputs but rather in the section that describes the operation of the block. The inputs and outputs are available to other circuit blocks in the system, but these local signals are recognized only within this block.

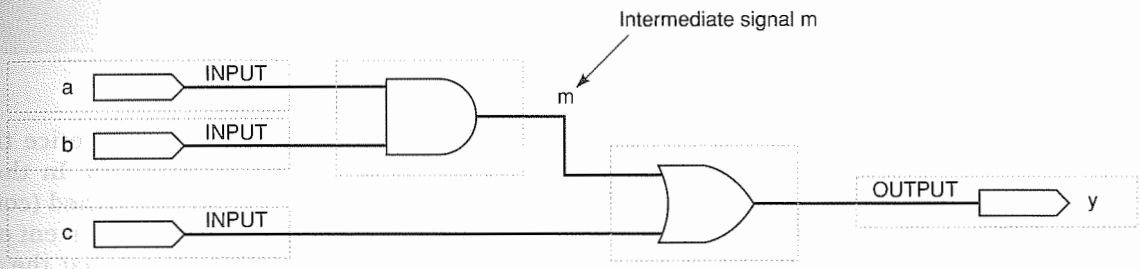


FIGURE 3-49 A logic circuit diagram with an intermediate variable.

In the example code that follows, notice the information at the top. The purpose of this information is strictly for documentation purposes. It is absolutely vital that the design is documented thoroughly. At a minimum, it should describe the project it is being used in, who wrote it, and the date. This information is often referred to as a header. We are keeping our headers brief to make this book a little lighter to carry to class, but remember: memory space is cheap and information is valuable. So don't be afraid to *document thoroughly!* There are also comments next to many of the statements in the code. These comments help the designer remember what she or he was trying to do and help any other person to understand what was intended.

AHDL BURIED NODES

The AHDL code that describes the circuit in Figure 3-49 is shown in Figure 3-50. The **comments** in AHDL can be enclosed between % characters, as you can see in the figure between lines 1 and 4. This section of the code allows the designer to write many lines of information that will be ignored by computer programs using this file but can be read by any person trying to decipher the code. Notice that the comments at the end of lines 7, 8, 11, 13, and 14 are preceded by two dashes (--). The text following the dashes is for documentation only. Either type of comment symbol may be used, but percent signs must be used in pairs to open and close a comment. Double dashes indicate a comment that extends to the end of the line.

In AHDL, local signals are declared in the VARIABLE section, which is placed between the SUBDESIGN section and the logic section. The intermediate signal *m* is defined on line 11, following the keyword **VARIABLE**. The

AHDL

FIGURE 3-50
Intermediate variables in AHDL described in Figure 3-49.

```
1   % Intermediate variables in AHDL (Figure 3-49)
2     Digital Systems 11th ed
3     NS Widmer
4     MAY 24, 2010          %
5   SUBDESIGN fig3_50
6   (
7     a,b,c      :INPUT;    -- define inputs to block
8     y          :OUTPUT;   -- define block output
9   )
10  VARIABLE
11    m        :NODE;       -- name an intermediate signal
12  BEGIN
13    m = a & b;            -- generate buried product term
14    y = m # c;            -- generate sum on output
15  END;
```

keyword **NODE** designates the nature of the variable. Notice that a colon separates the variable name from its node designation. In the hardware description on line 13, the intermediate variable is assigned (connected to) a value ($m = a$ & b;) and then m is used in the second statement on line 14 to assign (connect) a value to y ($y = m$ # c;). Remember that the assignment statements are concurrent and, thus, the order in which they are given does not matter. For human readability, it may seem more logical to assign values to intermediate variables before they are used in other assignment statements, as shown here.

REVIEW QUESTIONS

1. What is the designation used for intermediate variables?
2. Where are these variables declared?
3. Does it matter whether the m or y equation comes first?
4. What character is used to limit a block of comments?
5. What characters are used to comment a single line?

VHDL

VHDL LOCAL SIGNALS

The VHDL code that describes the circuit in Figure 3-49 is shown in Figure 3-51. The **comments** in VHDL follow two dashes (--). Typing two successive dashes allows the designer to write information from that point to the end of the line. The information following the two successive dashes will be ignored by computer programs using this file, but can be read by any person trying to decipher the code.

The intermediate signal m is defined on line 13 following the keyword SIGNAL. The keyword BIT designates the type of the signal. Notice that a colon separates the signal name from its type designation. In the hardware description on line 16, the intermediate signal is assigned (connected to) a

```
1   -- Intermediate variables in VHDL (Figure 3-49)
2   -- Digital Systems 11th ed
3   -- NS Widmer
4   -- MAY 24, 2010
5
6   ENTITY  fig3_51 IS
7   PORT( a, b, c   :IN BIT;     -- define inputs to block
8         y         :OUT BIT);   -- define block output
9   END fig3_51;
10
11  ARCHITECTURE ckt OF fig3_51 IS
12
13     SIGNAL m     :BIT;        -- same as intermediate signal
14
15  BEGIN
16        m <= a AND b;          -- generate buried product term
17        y <= m OR c;           -- generate sum on output
18  END ckt;
```

FIGURE 3-51 Intermediate signals in VHDL described in Figure 3-49.

value (m<=a AND b;) and then m is used in the statement on line 17 to assign (connect) a value to y (y<=m OR c;). Remember that the assignment statements are concurrent and, thus, the order in which they are given does not matter. For human readability, it may seem more logical to assign values to intermediate signals before they are used in other assignment statements, as shown here.

REVIEW QUESTIONS

1. What is the designation used for intermediate signals?
2. Where are these signals declared?
3. Does it matter whether the m or y equation comes first?
4. What characters are used to comment a single line?

SUMMARY

1. Boolean algebra is a mathematical tool used in the analysis and design of digital circuits.
2. The basic Boolean operations are the OR, AND, and NOT operations.
3. An OR gate produces a HIGH output when any input is HIGH. An AND gate produces a HIGH output only when all inputs are HIGH. A NOT circuit (INVERTER) produces an output that is the opposite logic level compared to the input.
4. A NOR gate is the same as an OR gate with its output connected to an INVERTER. A NAND gate is the same as an AND gate with its output connected to an INVERTER.

5. Boolean theorems and rules can be used to simplify the expression of a logic circuit and can lead to a simpler way of implementing the circuit.
6. NAND gates can be used to implement any of the basic Boolean operations. NOR gates can be used likewise.
7. Either standard or alternate symbols can be used for each logic gate, depending on whether the output is to be active-HIGH or active-LOW.
8. Propagation delay is the time between an input transition and the circuit's resulting response.
9. Hardware description languages have become an important method of describing digital circuits.
10. HDL code should always contain comments that document its vital characteristics so a person reading it later can understand what it does.
11. Every HDL circuit description contains a definition of the inputs and outputs, followed by a section that describes the circuit's operation.
12. In addition to inputs and outputs, intermediate connections that are buried within the circuit can be defined. These intermediate connections are called nodes or signals.

IMPORTANT TERMS

logic level
Boolean algebra
truth table
OR operation
OR gate
AND operation
AND gate
NOT operation
inversion (complementation)
NOT circuit (INVERTER)
NOR gate
NAND gate
Boolean theorems
DeMorgan's theorems
alternate logic symbols
active logic levels
active-HIGH
active-LOW
asserted
unasserted
propagation delay
hardware description languages (HDLs)
Altera hardware description language (AHDL)
very high speed integrated circuit (VHSIC) hardware description language (VHDL)
programmable logic devices (PLDs)
concurrent
compiler
syntax
mode
type
SUBDESIGN
concurrent assignment statement
ENTITY
BIT
ARCHITECTURE
buried nodes (local signals)
comments
VARIABLE
NODE

PROBLEMS

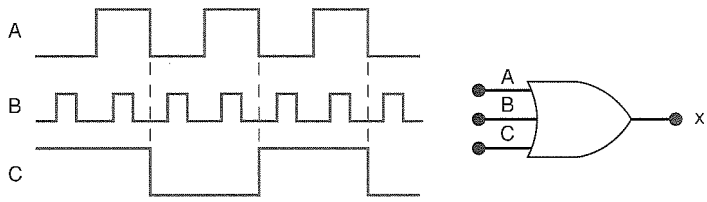
The color letters preceding some of the problems are used to indicate the nature or type of problem as follows:

B basic problem
T troubleshooting problem
D design or circuit-modification problem
N new concept or technique not covered in text
C challenging problem
H HDL problem

SECTION 3-3

B 3-1.* Draw the output waveform for the OR gate of Figure 3-52.

FIGURE 3-52



B 3-2. Suppose that the A input in Figure 3-52 is unintentionally shorted to ground (i.e., $A = 0$). Draw the resulting output waveform.

B 3-3.* Suppose that the A input in Figure 3-52 is unintentionally shorted to the +5 V supply line (i.e., $A = 1$). Draw the resulting output waveform.

C 3-4. Read the statements below concerning an OR gate. At first, they may appear to be valid, but after some thought you should realize that neither one is *always* true. Prove this by showing a specific example to refute each statement.

(a) If the output waveform from an OR gate is the same as the waveform at one of its inputs, the other input is being held permanently LOW.

(b) If the output waveform from an OR gate is always HIGH, one of its inputs is being held permanently HIGH.

B 3-5. How many different sets of input conditions will produce a HIGH output from a five-input OR gate?

SECTION 3-4

B 3-6. Change the OR gate in Figure 3-52 to an AND gate.

(a)* Draw the output waveform.

(b) Draw the output waveform if the A input is permanently shorted to ground.

(c) Draw the output waveform if A is permanently shorted to +5 V.

D 3-7.* Refer to Figure 3-4. Modify the circuit so that the alarm is to be activated only when the pressure and the temperature exceed their maximum limits at the same time.

B 3-8.* Change the OR gate in Figure 3-6 to an AND gate and draw the output waveform.

B 3-9. Suppose that you have an unknown two-input gate that is either an OR gate or an AND gate. What combination of input levels should you apply to the gate's inputs to determine which type of gate it is?

B 3-10. *True or false:* No matter how many inputs it has, an AND gate will produce a HIGH output for only one combination of input levels.

*Answers to problems marked with an asterisk can be found in the back of the text.

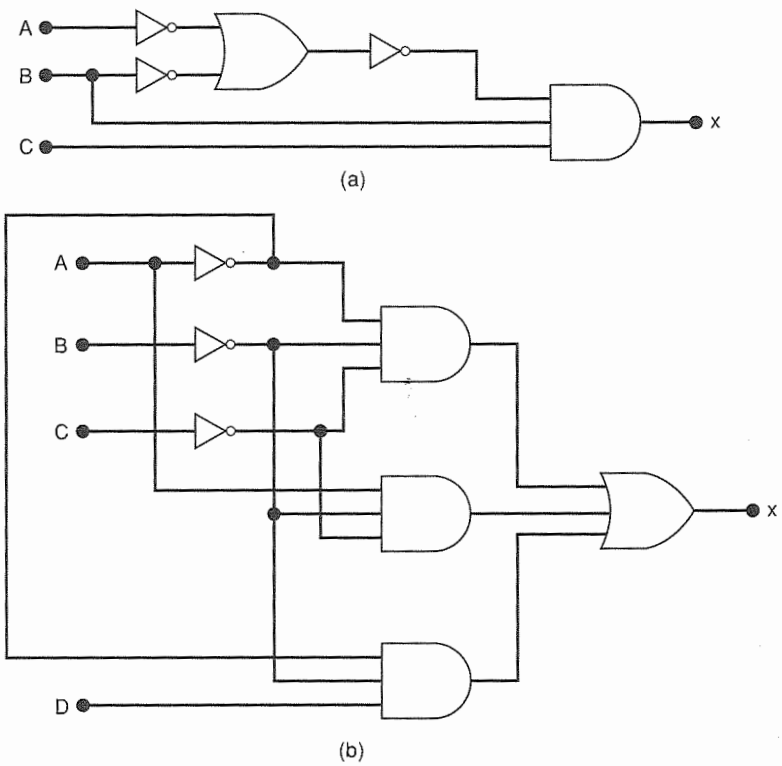
SECTIONS 3-5 TO 3-7

B 3-11. Apply the A waveform from Figure 3-23 to the input of an INVERTER. Draw the output waveform. Repeat for waveform B.

B 3-12. (a)* Write the Boolean expression for output x in Figure 3-53(a). Determine the value of x for all possible input conditions, and list the values in a truth table.

(b) Repeat for the circuit in Figure 3-53(b).

FIGURE 3-53



B 3-13* Create a complete analysis table for the circuit of Figure 3-15(b) by finding the logic levels present at each gate output for each of the 32 possible input combinations.

B 3-14. (a)* Change each OR to an AND, and each AND to an OR, in Figure 3-15(b). Then write the expression for the output.

(b) Complete an analysis table.

B 3-15. Create a complete analysis table for the circuit in Figure 3-15(a) by finding the logic levels present at each gate output for each of the 16 possible input combinations.

SECTION 3-8

B 3-16. For each of the following expressions, construct the corresponding logic circuit, using AND and OR gates and INVERTERs.

(a)* $x = \overline{AB(C + D)}$

(b)* $z = \overline{A + B + \overline{C}D\overline{E}} + \overline{B}C\overline{D}$

(c) $y = (\overline{M + N} + \overline{P}Q)$

(d) $x = \overline{W + P\overline{Q}}$

(e) $z = MN(P + \overline{N})$

(f) $x = (A + B)(\overline{A} + \overline{B})$

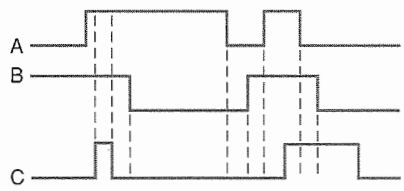
SECTION 3-9

B 3-17.* (a) Apply the input waveforms of Figure 3-54 to a NOR gate, and draw the output waveform.

(b) Repeat with C held permanently LOW.

(c) Repeat with C held HIGH.

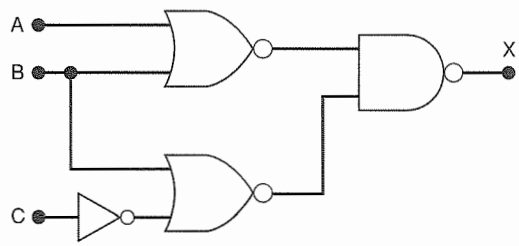
FIGURE 3-54



B 3-18. Repeat Problem 3-17 for a NAND gate.

C 3-19.* Write the expression for the output of Figure 3-55, and use it to determine the complete truth table. Then apply the waveforms of Figure 3-54 to the circuit inputs, and draw the resulting output waveform.

FIGURE 3-55



B 3-20. Determine the truth table for the circuit of Figure 3-24.

B 3-21. Modify the circuits that were constructed in Problem 3-16 so that NAND gates and NOR gates are used wherever appropriate.

SECTION 3-10

C 3-22. Prove theorems (15a) and (15b) by trying all possible cases.

B 3-23.* DRILL QUESTION

Complete each expression.

(a) $A + 1 =$ _____

(b) $A \cdot A =$ _____

(c) $B \cdot \overline{B} =$ _____

(d) $C + C =$ _____

(e) $x \cdot 0 =$ _____

(f) $D \cdot 1 =$ _____

(g) $D + 0 =$ _____

(h) $C + \overline{C} =$ _____

(i) $G + GF =$ _____

(j) $y + \overline{w}y =$ _____

C 3-24. (a)*Simplify the following expression using theorems (13b), (3), and (4):

$$x = (M + N)(\overline{M} + P)(\overline{N} + \overline{P})$$

(b) Simplify the following expression using theorems (13a), (8), and (6):

$$z = \overline{A}B\overline{C} + AB\overline{C} + B\overline{C}D$$

SECTIONS 3-11 AND 3-12

C 3-25. Prove DeMorgan's theorems by trying all possible cases.

B 3-26. Simplify each of the following expressions using DeMorgan's theorems.

(a)* $\overline{\overline{A}B\overline{C}}$	(d) $\overline{A + \overline{B}}$	(g)* $\overline{\overline{A(\overline{B} + \overline{C})D}}$
(b) $\overline{\overline{A} + \overline{B}C}$	(e)* $\overline{\overline{AB}}$	(h) $\overline{(M + \overline{N})(\overline{M} + N)}$
(c)* $\overline{AB\overline{CD}}$	(f) $\overline{\overline{A} + \overline{C} + \overline{D}}$	(i) $\overline{\overline{ABCD}}$

B 3-27.* Use DeMorgan's theorems to simplify the expression for the output of Figure 3-55.

C 3-28. Convert the circuit of Figure 3-53(b) to one using only NAND gates. Then write the output expression for the new circuit, simplify it using DeMorgan's theorems, and compare it with the expression for the original circuit.

C 3-29. Convert the circuit of Figure 3-53(a) to one using only NOR gates. Then write the expression for the new circuit, simplify it using DeMorgan's theorems, and compare it with the expression for the original circuit.

B 3-30. Show how a two-input NAND gate can be constructed from two-input NOR gates.

B 3-31. Show how a two-input NOR gate can be constructed from two-input NAND gates.

C 3-32. A jet aircraft employs a system for monitoring the rpm, pressure, and temperature values of its engines using sensors that operate as follows:

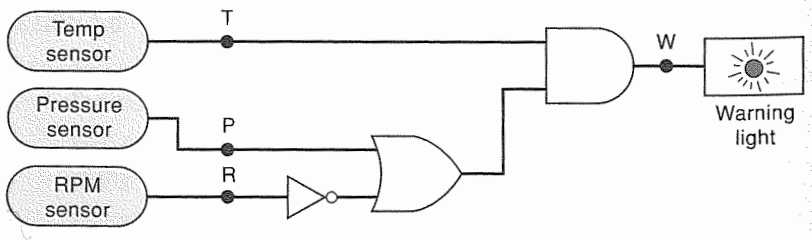
RPM sensor output = 0 only when speed < 4800 rpm
P sensor output = 0 only when pressure < 220 psi
T sensor output = 0 only when temperature < 200°F

Figure 3-56 shows the logic circuit that controls a cockpit warning light for certain combinations of engine conditions. Assume that a HIGH at output *W* activates the warning light.

(a)* Determine what engine conditions will give a warning to the pilot.

(b) Change this circuit to one using all NAND gates.

FIGURE 3-56



SECTIONS 3-13 AND 3-14

B 3-33.*For each statement below, draw the appropriate logic-gate symbol—standard or alternate—for the given operation.

(a) A HIGH output occurs only when all three inputs are LOW.

(b) A LOW output occurs when any of the four inputs is LOW.

(c) A LOW output occurs only when all eight inputs are HIGH.

B 3-34. Draw the standard representations for each of the basic logic gates. Then draw the alternate representations.

C 3-35. The circuit of Figure 3-55 is supposed to be a simple digital combination lock whose output will generate an active-LOW $\overline{UNLOCK}$ signal for only one combination of inputs.

(a)*Modify the circuit diagram so that it represents more effectively the circuit operation.

(b) Use the new circuit diagram to determine the input combination that will activate the output. Do this by working back from the output using the information given by the gate symbols, as was done in Examples 3-22 and 3-23. Compare the results with the truth table obtained in Problem 3-19.

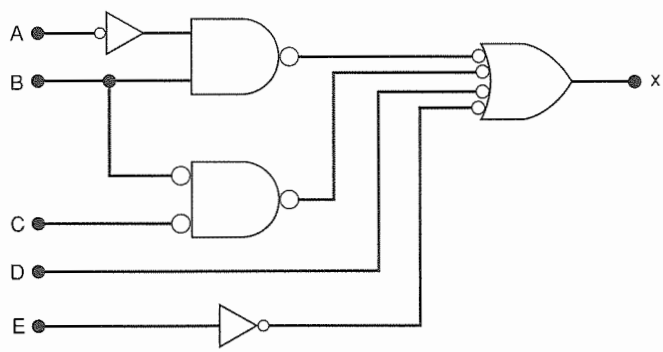
C 3-36. (a) Determine the input conditions needed to activate output Z in Figure 3-37(b). Do this by working back from the output, as was done in Examples 3-22 and 3-23.

(b) Assume that it is the LOW state of Z that is to activate the alarm. Change the circuit diagram to reflect this, and then use the revised diagram to determine the input conditions needed to activate the alarm.

D 3-37. Modify the circuit of Figure 3-40 so that $A_1 = 0$ is needed to produce $LCD = 1$ instead of $A_1 = 1$.

B 3-38.*Determine the input conditions needed to cause the output in Figure 3-57 to go to its active state.

FIGURE 3-57



B 3-39.*(a) What is the asserted state for the output of Figure 3-57?

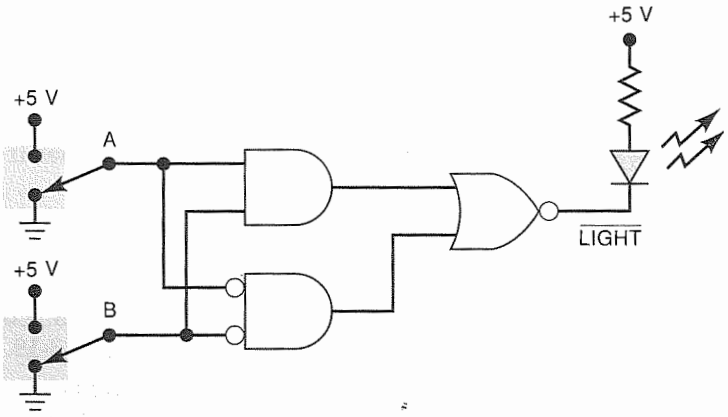
(b) What is the asserted state for the output of Figure 3-36(c)?

B 3-40. Use the results of Problem 3-38 to obtain the complete truth table for the circuit of Figure 3-57.

N 3-41.*Figure 3-58 shows an application of logic gates that simulates a two-way switch like the ones used in our homes to turn a light on or off

from two different switches. Here the light is an LED that will be ON (conducting) when the NOR gate output is LOW. Note that this output is labeled $\overline{LIGHT}$ to indicate that it is active-LOW. Determine the input conditions needed to turn on the LED. Then verify that the circuit operates as a two-way switch using switches *A* and *B*. (In Chapter 4, you will learn how to design circuits like this one to produce a given relationship between inputs and outputs.)

FIGURE 3-58



SECTION 3-15

B 3-42. A 7406 TTL inverter has a maximum t_{PLH} of 15 ns and a t_{PHL} of 23 ns. A positive pulse that lasts 100 ns is applied to the input.

(a) Draw the input and output waveforms. Scale the X-axis such that the end time is 200 ns.

(b) Label t_{PLH} and t_{PHL} on the graph.

(c) What is the pulse width of the output if worst case propagation delays occur?

SECTION 3-17
HDL DRILL QUESTIONS

H 3-43.* *True or false:*

(a) VHDL is a computer programming language.

(b) VHDL can accomplish the same thing as AHDL.

(c) AHDL is an IEEE standard language.

(d) Each intersection in a switch matrix can be programmed as an open or short circuit between a row and column wire.

(e) The first item that appears at the top of an HDL listing is the functional description.

(f) The type of an object indicates if it is an input or an output.

(g) The mode of an object determines if it is an input or an output.

(h) Buried nodes are nodes that have been deleted and will never be used again.

(i) Local signals are another name for intermediate variables.

(j) The header is a block of comments that document vital information about the project.

SECTION 3-18

B 3-44. Redraw the programmable connection matrix from Figure 3-44. Label the output signals (horizontal lines) from the connection matrix (from top row to bottom row) as follows: AAABADHE. Draw an X in the appropriate intersections to short-circuit a row to a column and create these connections to the logic circuit.

H 3-45.* Write the HDL code in the language of your choice that will produce the following output functions:

$$X = A + B$$
$$Y = \overline{AB}$$
$$Z = A + B + C$$

H 3-46. Write the HDL code in the language of your choice that will implement the logic circuit of Figure 3-39.

(a) Use a single Boolean equation.

(b) Use the intermediate variables V, W, X, and Y.

MICROCOMPUTER APPLICATION

C 3-47.* Refer to Figure 3-40 in Example 3-23. Inputs A_7 through A_0 are *address* inputs that are supplied to this circuit from outputs of the microprocessor chip in a microcomputer. The eight-bit address code A_7 to A_0 selects which device the microprocessor wants to activate. In Example 3-23, the required address code to activate the LCD was A_7 through A_0 = $11111110_2 = FE_{16}$.

Modify the circuit so that the microprocessor must supply an address code of $4A_{16}$ to activate the LCD.

CHALLENGING EXERCISES

C 3-48. Show how $x = AB\overline{C}$ can be implemented with one two-input NOR and one two-input NAND gate.

C 3-49.* Implement $y = ABCD$ using only two-input NAND gates.

ANSWERS TO SECTION REVIEW QUESTIONS

SECTION 3-2

1. $x = 1$ 2. $x = 0$ 3. 32

SECTION 3-3

1. All inputs LOW 2. $x = A + B + C + D + E + F$ 3. Constant HIGH

SECTION 3-4

1. All five inputs = 1 2. A LOW input will keep the output LOW. 3. False; see truth table of each gate.

SECTION 3-5

1. Output of second INVERTER will be the same as input A. 2. y will be LOW only for $A = B = 1$.

SECTION 3-6

1. $x = \overline{A} + B + C + \overline{AD}$ 2. $x = D(\overline{AB + C}) + E$

SECTION 3-7

1. $x = 1$ 2. $x = 1$ 3. $x = 1$ for both.

SECTION 3-8

1. See Figure 3-15(a). 2. See Figure 3-17(b). 3. See Figure 3-15(b).

SECTION 3-9

1. All inputs LOW. 2. $x = 0$ 3. $x = \overline{A + B + \overline{CD}}$

SECTION 3-10

1. $y = A\overline{C}$ 2. $y = \overline{A}\,\overline{B}\,\overline{D}$ 3. $y = \overline{A}D + BD$

SECTION 3-11

1. $z = \overline{A}\overline{B} + C$ 2. $y = (\overline{R} + S + \overline{T})Q$ 3. Same as Figure 3-28 except NAND is replaced by NOR. 4. $y = \overline{A}B(C + \overline{D})$

SECTION 3-12

1. Three. 2. NOR circuit is more efficient because it can be implemented with only three NOR gates. 3. $x = (\overline{AB})(\overline{CD}) = \overline{\overline{AB}} + (\overline{\overline{CD}}) = AB + CD$

SECTION 3-13

1. Output goes LOW when any input is HIGH. 2. Output goes HIGH only when all inputs are LOW. 3. Output goes LOW when any input is LOW. 4. Output goes HIGH only when all inputs are HIGH.

SECTION 3-14

1. Z will go HIGH when $A = B = 0$ and $C = D = 1$. 2. Z will go LOW when $A = B = 0, E = 1$, and either C or D or both are 0. 3. Two 4. Two 5. LOW 6. $A = B = 0, C = D = 1$ 7. $\overline{W}$

SECTION 3-15

1. The time scale is in nanoseconds and it takes a finite amount of time to change states. 2. From 50% point on the input to 50% point on the output 3. t_{PHL} 4. t_{PLH}

SECTION 3-16

1. Boolean equation, truth table, logic diagram, timing diagram, language.
2. Schematic entry of .bdf files and simulation using timing diagrams

SECTION 3-17

1. Hardware description language 2. To describe a digital circuit and its operation. 3. To give a computer a sequential list of tasks. 4. HDL describes concurrent hardware circuits; computer instructions execute one at a time.

SECTION 3-18

1. Programmable logic device 2. By making and breaking connections in a switching matrix 3. It translates HDL code into a pattern of bits to configure the switching matrix.

SECTION 3-19

AHDL

1. The input and output definitions. 2. The description of how it operates.

VHDL

1. To give a name to the circuit and define its inputs and outputs. 2. The ARCHITECTURE description. 3. <=

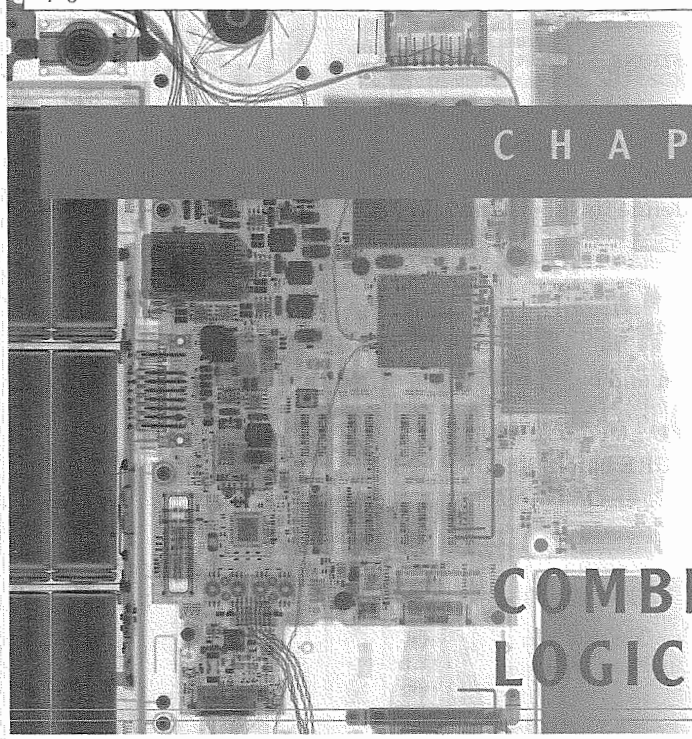
SECTION 3-20

AHDL

1. NODE 2. After the I/O definition and before BEGIN. 3. No 4. % 5. --

VHDL

1. SIGNAL 2. Inside ARCHITECTURE before BEGIN. 3. No 4. --



CHAPTER 4

COMBINATIONAL LOGIC CIRCUITS

OUTLINE

4-1 Sum-of-Products Form
4-2 Simplifying Logic Circuits
4-3 Algebraic Simplification
4-4 Designing Combinational Logic Circuits
4-5 Karnaugh Map Method
4-6 Exclusive-OR and Exclusive-NOR Circuits
4-7 Parity Generator and Checker
4-8 Enable/Disable Circuits
4-9 Basic Characteristics of Digital ICs
4-10 Troubleshooting Digital Systems
4-11 Internal Digital IC Faults
4-12 External Faults
4-13 Troubleshooting Case Study
4-14 Programmable Logic Devices
4-15 Representing Data in HDL
4-16 Truth Tables Using HDL
4-17 Decision Control Structures in HDL

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Combinational Logic Circuits. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 128-217). Upper Saddle River, NJ: Pearson Education Inc.

OBJECTIVES

Upon completion of this chapter, you will be able to:

- Convert a logic expression into a sum-of-products expression.
- Perform the necessary steps to reduce a sum-of-products expression to its simplest form.
- Use Boolean algebra and the Karnaugh map as tools to simplify and design logic circuits.
- Explain the operation of both exclusive-OR and exclusive-NOR circuits.
- Design simple logic circuits without the help of a truth table.
- Describe how to implement enable circuits.
- Cite the basic characteristics of TTL and CMOS digital ICs.
- Use the basic troubleshooting rules of digital systems.
- Deduce from observed results the faults of malfunctioning combinational logic circuits.
- Describe the fundamental idea of programmable logic devices (PLDs).
- Describe the steps involved in programming a PLD to perform a simple combinational logic function.
- Describe hierarchical design methods.
- Identify proper data types for single-bit, bit array, and numeric value variables.
- Describe logic circuits using HDL control structures IF/ELSE, IF/ELSIF, and CASE.
- Select the appropriate HDL control structure for a given problem.

INTRODUCTION

In Chapter 3, we studied the operation of all the basic logic gates, and we used Boolean algebra to describe and analyze circuits that were made up of combinations of logic gates. These circuits can be classified as *combinational* logic circuits because, at any time, the logic level at the output depends on the combination of logic levels present at the inputs. A combinational circuit has no *memory* characteristic, so its output depends *only* on the current value of its inputs.

In this chapter, we will continue our study of combinational circuits. To start, we will go further into the simplification of logic circuits. Two methods will be used: one uses Boolean algebra theorems; the other uses a *mapping* technique. In addition, we will study simple techniques for designing

Tocci, RJ., Widmer, NS. and Moss, GL. (2011). Combinational Logic Circuits. In *Digital Systems: Principles and Applications*, (11th Ed., pp. 128-217). Upper Saddle River, NJ: Pearson Education Inc.

combinational logic circuits to satisfy a given set of requirements. A complete study of logic-circuit design is not one of our objectives, but the methods we introduce will provide a good introduction to logic design.

A good portion of this chapter is devoted to the topic of *troubleshooting*. This term has been adopted as a general description of the process of isolating a problem or fault in any system and identifying a way of fixing it. The analytical skills and efficient methods of troubleshooting are equally applicable to any system whether it is a plumbing problem, a problem with your car, a health issue, or a digital circuit. Digital systems, implemented using TTL-integrated circuits, have for decades provided an exceptional vehicle for the study of efficient, systematic troubleshooting methods. As with any system, the practical characteristics of the pieces that make up the system must be understood in order to effectively analyze its normal operation, locate the trouble, and propose a remedy. We will present some basic characteristics and typical failure modes of logic ICs in the TTL and CMOS families that are still commonly used for laboratory instruction in introductory digital courses and take advantage of this technology to teach some fundamental troubleshooting principles.

In the last sections of this chapter, we will extend our knowledge of programmable logic devices and hardware description languages. The concept of programmable hardware connections will be reinforced, and we will provide more details regarding the role of the development system. You will learn the steps followed in the design and development of digital systems today. Enough information will be provided to allow you to choose the correct types of data objects for use in simple projects to be presented later in this text. Finally, several control structures will be explained, along with some instruction regarding their appropriate use.

4-1 SUM-OF-PRODUCTS FORM

The methods of logic-circuit simplification and design that we will study require the logic expression to be in a **sum-of-products (SOP)** form. Some examples of this form are:

1. $ABC + \overline{A}B\overline{C}$
2. $AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$
3. $\overline{A}B + C\overline{D} + EF + GK + H\overline{L}$

Each of these sum-of-products expressions consists of two or more AND terms (products) that are ORed together. Each AND term consists of one or more variables *individually* appearing in either complemented or uncomplemented form. For example, in the sum-of-products expression $ABC + \overline{A}B\overline{C}$, the first AND product contains the variables A, B, and C in their uncomplemented (not inverted) form. The second AND term contains A and C in their complemented (inverted) form. Note that in a sum-of-products expression, one inversion sign *cannot* cover more than one variable in a term (e.g., we cannot have $\overline{ABC}$ or $\overline{RST}$).

Product-of-Sums

Another general form for logic expressions is sometimes used in logic-circuit design. Called the **product-of-sums (POS)** form, it consists of two or more OR terms (sums) that are ANDed together. Each OR term contains one or more

variables in complemented or uncomplemented form. Here are some product-of-sum expressions:

1. $(A + \overline{B} + C)(A + C)$
2. $(A + \overline{B})(\overline{C} + D)F$
3. $(A + C)(B + \overline{D})(\overline{B} + C)(A + \overline{D} + \overline{E})$

The methods of circuit simplification and design that we will be using are based on the sum-of-products (SOP) form, so we will not be doing much with the product-of-sums (POS) form. It will, however, occur from time to time in some logic circuits that have a particular structure.

REVIEW QUESTIONS

1. Which of the following expressions is in SOP form?
   (a) $AB + CD + E$
   (b) $AB(C + D)$
   (c) $(A + B)(C + D + F)$
   (d) $\overline{MN} + PQ$
2. Repeat question 1 for the POS form.

4-2 SIMPLIFYING LOGIC CIRCUITS

Once the expression for a logic circuit has been obtained, we may be able to reduce it to a simpler form containing fewer terms or fewer variables in one or more terms. The new expression can then be used to implement a circuit that is equivalent to the original circuit but that contains fewer gates and connections.

To illustrate, the circuit of Figure 4-1(a) can be simplified to produce the circuit of Figure 4-1(b). Both circuits perform the same logic, so it should be obvious that the simpler circuit is more desirable because it contains fewer gates and will therefore be smaller and cheaper than the original. Furthermore, the circuit reliability will improve because there are fewer interconnections that can be potential circuit faults.

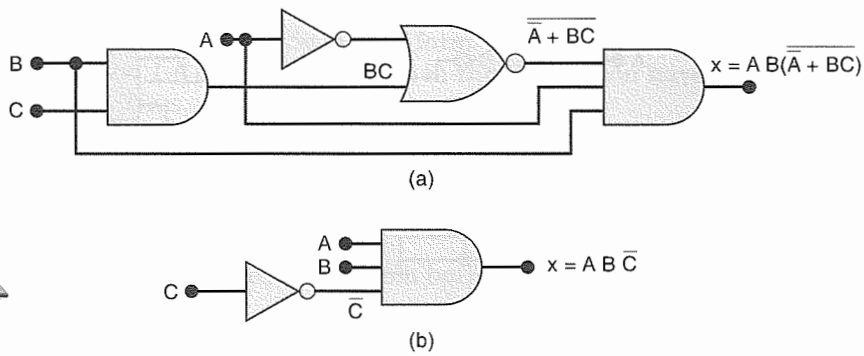


FIGURE 4-1 It is often possible to simplify a logic circuit such as that in part (a) to produce a more efficient implementation, shown in (b).

Another strategic advantage of simplifying logic circuits involves the operational speed of circuits. Recall from previous discussions that logic gates are subject to propagation delay. If practical logic circuits are configured such that logical changes in the inputs must propagate through many layers of gates in order to determine the output, they cannot possibly operate as fast as circuits with fewer layers of gates. For example, compare the circuits

of Figure 4-1(a) and (b). In 4-1(a) the longest path a signal must travel involves three gates. In Figure 4-1(b), the longest signal path (C) only involves two gates. Working toward a common form such as SOP or POS assures similar propagation delay for all signals in the system and helps determine the maximum operating speed of the system.

In subsequent sections, we will study two methods for simplifying logic circuits. One method will utilize the Boolean algebra theorems and, as we shall see, is greatly dependent on inspiration and experience. The other method (Karnaugh mapping) is a systematic, step-by-step approach. Some instructors may wish to skip over this latter method because it is somewhat mechanical and probably does not contribute to a better understanding of Boolean algebra. This can be done without affecting the continuity or clarity of the rest of the text.

4-3 ALGEBRAIC SIMPLIFICATION

We can use the Boolean algebra theorems that we studied in Chapter 3 to help us simplify the expression for a logic circuit. Unfortunately, it is not always obvious which theorems should be applied to produce the simplest result. Furthermore, there is no easy way to tell whether the simplified expression is in its simplest form or whether it could have been simplified further. Thus, algebraic simplification often becomes a process of trial and error. With experience, however, one can become adept at obtaining reasonably good results.

The examples that follow will illustrate many of the ways in which the Boolean theorems can be applied in trying to simplify an expression. You should notice that these examples contain two essential steps:

1. The original expression is put into SOP form by repeated application of DeMorgan's theorems and multiplication of terms.
2. Once the original expression is in SOP form, the product terms are checked for common factors, and factoring is performed wherever possible. The factoring should result in the elimination of one or more terms.

EXAMPLE 4-1

Simplify the logic circuit shown in Figure 4-2(a).

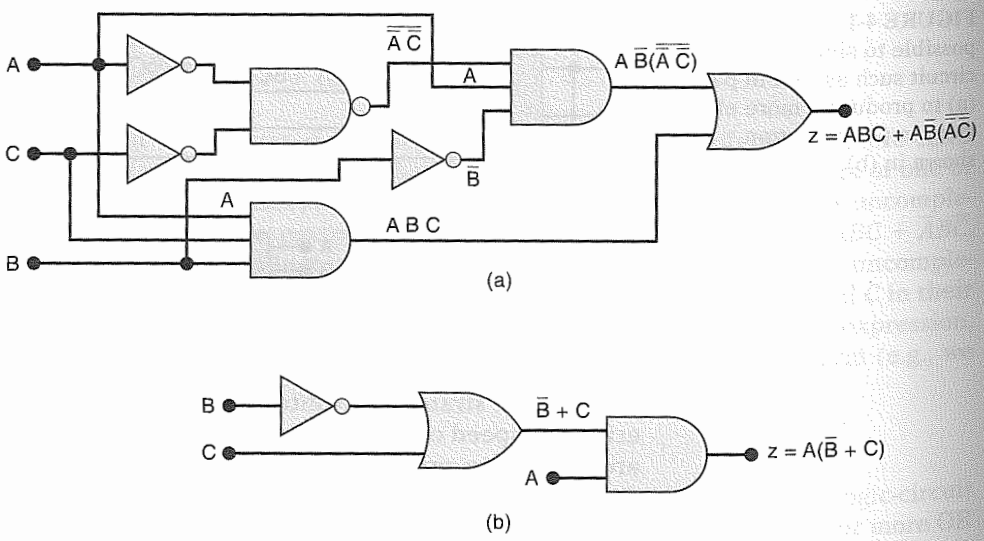


FIGURE 4-2 Example 4-1.

Solution

The first step is to determine the expression for the output using the method presented in Section 3-6. The result is

$$z = ABC + A\overline{B} \cdot (\overline{\overline{A}\,\overline{C}})$$

Once the expression is determined, it is usually a good idea to break down all large inverter signs using DeMorgan's theorems and then multiply out all terms.

$$\begin{aligned} z &= ABC + A\overline{B}(\overline{\overline{A}} + \overline{\overline{C}}) && \text{[theorem (17)]} \\ &= ABC + A\overline{B}(A + C) && \text{[cancel double inversions]} \\ &= ABC + A\overline{B}A + A\overline{B}C && \text{[multiply out]} \\ &= ABC + A\overline{B} + A\overline{B}C && [A \cdot A = A] \end{aligned}$$

With the expression now in SOP form, we should look for common variables among the various terms with the intention of factoring. The first and third terms above have AC in common, which can be factored out:

$$z = AC(B + \overline{B}) + A\overline{B}$$

Since $B + \overline{B} = 1$, then

$$\begin{aligned} z &= AC(1) + A\overline{B} \\ &= AC + A\overline{B} \end{aligned}$$

We can now factor out A, which results in

$$z = A(C + \overline{B})$$

This result can be simplified no further. Its circuit implementation is shown in Figure 4-2(b). It is obvious that the circuit in Figure 4-2(b) is a great deal simpler than the original circuit in Figure 4-2(a).

EXAMPLE 4-2

Simplify the expression $z = A\overline{B}\,\overline{C} + A\overline{B}C + ABC$.

Solution

The expression is already in SOP form.

Method 1: The first two terms in the expression have the product $A\overline{B}$ in common. Thus,

$$\begin{aligned} z &= A\overline{B}(\overline{C} + C) + ABC \\ &= A\overline{B}(1) + ABC \\ &= A\overline{B} + ABC \end{aligned}$$

We can factor the variable A from both terms:

$$z = A(\overline{B} + BC)$$

Invoking theorem (15b):

$$z = A(\overline{B} + C)$$

Method 2: The original expression is $z = A\overline{B}\,\overline{C} + A\overline{B}C + ABC$. The first two terms have $A\overline{B}$ in common. The last two terms have AC in common. How do we know whether to factor $A\overline{B}$ from the first two terms or AC from the

last two terms? Actually, we can do both by using the $A\overline{B}C$ term *twice*. In other words, we can rewrite the expression as:

$$z = A\overline{B}\,\overline{C} + A\overline{B}C + A\overline{B}C + ABC$$

where we have added an extra term $A\overline{B}C$. This is valid and will not change the value of the expression because $A\overline{B}C + A\overline{B}C = A\overline{B}C$ [theorem (7)]. Now we can factor $A\overline{B}$ from the first two terms and AC from the last two terms:

$$\begin{aligned} z &= A\overline{B}(C + \overline{C}) + AC(\overline{B} + B) \\ &= A\overline{B}\cdot 1 + AC\cdot 1 \\ &= A\overline{B} + AC = A(\overline{B} + C) \end{aligned}$$

Of course, this is the same result obtained with method 1. This trick of using the same term twice can always be used. In fact, the same term can be used more than twice if necessary.

EXAMPLE 4-3

Simplify $z = \overline{A}C(\overline{\overline{A}BD}) + \overline{A}B\overline{C}\,\overline{D} + A\overline{B}C$.

Solution

First, use DeMorgan's theorem on the first term:

$$z = \overline{A}C(A + \overline{B} + \overline{D}) + \overline{A}B\overline{C}\,\overline{D} + A\overline{B}C \qquad \text{(step 1)}$$

Multiplying out yields

$$z = \overline{A}CA + \overline{A}C\overline{B} + \overline{A}C\overline{D} + \overline{A}B\overline{C}\,\overline{D} + A\overline{B}C \qquad (2)$$

Because $\overline{A}\cdot A = 0$, the first term is eliminated:

$$z = \overline{A}\,\overline{B}C + \overline{A}C\overline{D} + \overline{A}B\overline{C}\,\overline{D} + A\overline{B}C \qquad (3)$$

This is the desired SOP form. Now we must look for common factors among the various product terms. The idea is to check for the largest common factor between any two or more product terms. For example, the first and last terms have the common factor $\overline{B}C$, and the second and third terms share the common factor $\overline{A}\,\overline{D}$. We can factor these out as follows:

$$z = \overline{B}C(\overline{A} + A) + \overline{A}\,\overline{D}(C + B\overline{C}) \qquad (4)$$

Now, because $\overline{A} + A = 1$, and $C + B\overline{C} = C + B$ [theorem (15a)], we have

$$z = \overline{B}C + \overline{A}\,\overline{D}(B + C) \qquad (5)$$

This same result could have been reached with other choices for the factoring. For example, we could have factored C from the first, second, and fourth product terms in step 3 to obtain

$$z = C(\overline{A}\,\overline{B} + \overline{A}\,\overline{D} + A\overline{B}) + \overline{A}B\overline{C}\,\overline{D}$$

The expression inside the parentheses can be factored further:

$$z = C(\overline{B}[\overline{A} + A] + \overline{A}\,\overline{D}) + \overline{A}B\overline{C}\,\overline{D}$$

Because $\overline{A} + A = 1$, this becomes

$$z = C(\overline{B} + \overline{A}\,\overline{D}) + \overline{A}B\overline{C}\,\overline{D}$$

Multiplying out yields

$$z = \overline{B}C + \overline{A}C\overline{D} + \overline{A}B\overline{C}\,\overline{D}$$

Now we can factor $\overline{A}\,\overline{D}$ from the second and third terms to get

$$z = \overline{B}C + \overline{A}\,\overline{D}(C + B\overline{C})$$

If we use theorem (15a), the expression in parentheses becomes $B + C$. Thus, we finally have

$$z = \overline{B}C + \overline{A}\,\overline{D}(B + C)$$

This is the same result that we obtained earlier, but it took us many more steps. This illustrates why you should look for the largest common factors: it will generally lead to the final expression in the fewest steps.

Example 4-3 illustrates the frustration often encountered in Boolean simplification. Because we have arrived at the same equation (which appears irreducible) by two different methods, it might seem reasonable to conclude that this final equation is the simplest form. In fact, the simplest form of this equation is

$$z = \overline{A}B\overline{D} + \overline{B}C$$

But there is no apparent way to reduce step (5) to reach this simpler version. In this case, we missed an operation earlier in the process that could have led to the simpler form. The question is, "How could we have known that we missed a step?" Later in this chapter, we will examine a mapping technique that will always lead to the simplest SOP form.

EXAMPLE 4-4

Simplify the expression $x = (\overline{A} + B)(A + B + D)\overline{D}$.

Solution

The expression can be put into sum-of-products form by multiplying out all the terms. The result is

$$x = \overline{A}A\overline{D} + \overline{A}B\overline{D} + \overline{A}D\overline{D} + BA\overline{D} + BB\overline{D} + BD\overline{D}$$

The first term can be eliminated because $\overline{A}A = 0$. Likewise, the third and sixth terms can be eliminated because $D\overline{D} = 0$. The fifth term can be simplified to $B\overline{D}$ because $BB = B$. This gives us

$$x = \overline{A}B\overline{D} + AB\overline{D} + B\overline{D}$$

We can factor $B\overline{D}$ from each term to obtain

$$x = B\overline{D}(\overline{A} + A + 1)$$

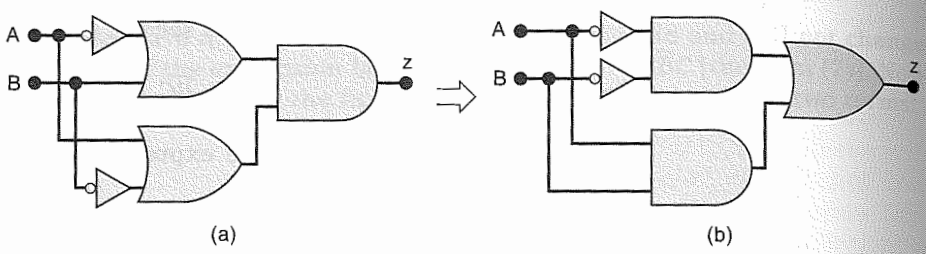
Clearly, the term inside the parentheses is always 1, so we finally have

$$x = B\overline{D}$$

EXAMPLE 4-5

Simplify the circuit of Figure 4-3(a).

FIGURE 4-3 Example 4-5.



Solution

The expression for output z is

$$z = (\overline{A} + B)(A + \overline{B})$$

Multiplying out to get the sum-of-products form, we obtain

$$z = \overline{A}A + \overline{A}\,\overline{B} + BA + B\overline{B}$$

We can eliminate $\overline{A}A = 0$ and $B\overline{B} = 0$ to end up with

$$z = \overline{A}\,\overline{B} + AB$$

This expression is implemented in Figure 4-3(b), and if we compare it with the original circuit, we see that both circuits contain the same number of gates and connections. In this case, the simplification process produced an equivalent, but not simpler, circuit.

EXAMPLE 4-6

Simplify $x = A\overline{B}C + \overline{A}BD + \overline{C}\,\overline{D}$.

Solution

You can try, but you will not be able to simplify this expression any further.

REVIEW QUESTIONS

1. State which of the following expressions are *not* in the sum-of-products form:
   (a) $RS\overline{T} + \overline{R}S\overline{T} + \overline{T}$
   (b) $A\overline{CD} + \overline{A}CD$
   (c) $MN\overline{P} + (M + \overline{N})P$
   (d) $AB + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}D$
2. Simplify the circuit in Figure 4-1(a) to arrive at the circuit of Figure 4-1(b).
3. Change each AND gate in Figure 4-1(a) to a NAND gate. Determine the new expression for x and simplify it.

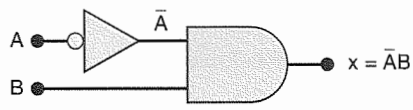
4-4 DESIGNING COMBINATIONAL LOGIC CIRCUITS

When the desired output level of a logic circuit is given for all possible input conditions, the results can be conveniently displayed in a truth table. The Boolean expression for the required circuit can then be derived from the truth table. For example, consider Figure 4-4(a), where a truth table is shown for a circuit that has two inputs, A and B, and output x. The table shows that output x is to be at the 1 level *only* for the case where $A = 0$ and $B = 1$. It now remains to determine what logic circuit will produce this desired operation. It should be apparent that one possible solution is that shown in Figure 4-4(b). Here an AND gate is used with inputs $\overline{A}$ and B, so that $x = \overline{A} \cdot B$. Obviously x will be 1 *only if* both inputs to the AND gate are 1, namely, $\overline{A} = 1$ (which means that $A = 0$) and $B = 1$. For all other values of A and B, the output x will be 0.

FIGURE 4-4 Circuit that produces a 1 output only for the $A = 0, B = 1$ condition.

A	B	x
0	0	0
0	1	1
1	0	0
1	1	0

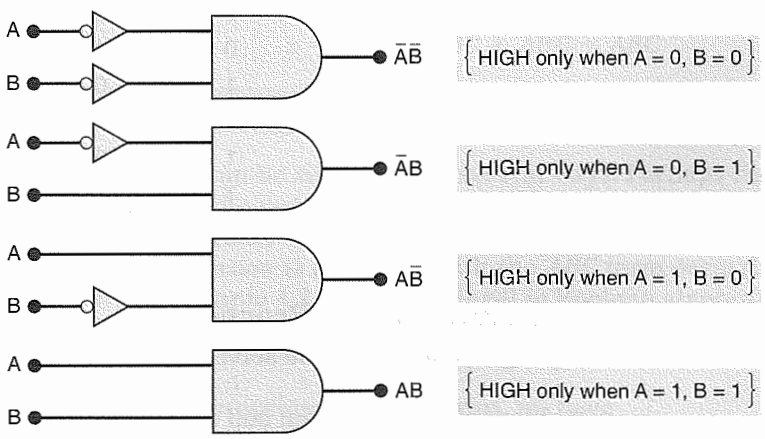
(a)



(b)

A similar approach can be used for the other input conditions. For instance, if x were to be HIGH only for the $A = 1, B = 0$ condition, the resulting circuit would be an AND gate with inputs A and $\overline{B}$. In other words, for any of the four possible input conditions, we can generate a HIGH x output by using an AND gate with appropriate inputs to generate the required AND product. The four different cases are shown in Figure 4-5. Each of the AND gates shown generates an output that is HIGH *only* for one given input condition and the output is LOW for all other conditions. It should be noted that the AND inputs are inverted or not inverted depending on the values that the variables have for the given condition. If the variable is LOW for the given condition, it is inverted before entering the AND gate.

FIGURE 4-5 An AND gate with appropriate inputs can be used to produce a HIGH output for a specific set of input levels.



Let us now consider the case shown in Figure 4-6(a), where we have a truth table that indicates that the output x is to be 1 for two different cases: $A = 0, B = 1$ and $A = 1, B = 0$. How can this be implemented? We know that

the AND term $\overline{A} \cdot B$ will generate a 1 only for the $A = 0, B = 1$ condition, and the AND term $A \cdot \overline{B}$ will generate a 1 for the $A = 1, B = 0$ condition. Because $x = 1$ for *either* condition, it should be clear that these terms should be ORed together to produce the desired output, x. This implementation is shown in Figure 4-6(b), where the resulting expression for the output is $x = \overline{A}B + A\overline{B}$.

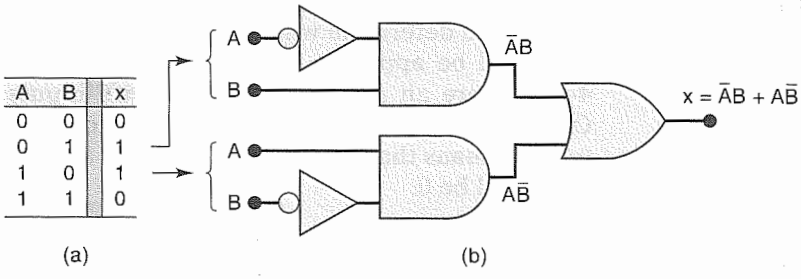


FIGURE 4-6 Each set of input conditions that is to produce a 1 output is implemented by a separate AND gate. The AND outputs are ORed to produce the final output.

In this example, an AND term is generated for each case in the table where the output $x = 1$. The AND gate outputs are then ORed together to produce the total output x, which will be 1 when either AND term is 1. This same procedure can be extended to examples with more than two inputs. Consider the truth table for a three-input circuit (Table 4-1). Here there are three cases where the output $x = 1$. The required AND term for each of these cases is shown. Again, note that for each case where a variable is 0, it appears inverted in the AND term. The sum-of-products expression for x is obtained by ORing the three AND terms.

$$x = \overline{A}B\overline{C} + \overline{A}BC + ABC$$

TABLE 4-1

A	B	C	x	
0	0	0	0	
0	0	1	0	
0	1	0	1	$\rightarrow \overline{A}B\overline{C}$
0	1	1	1	$\rightarrow \overline{A}BC$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	$\rightarrow ABC$

Complete Design Procedure

Any logic problem can be solved using the following step-by-step procedure.

1. Interpret the problem and set up a truth table to describe its operation.
2. Write the AND (product) term for each case where the output is 1.
3. Write the sum-of-products (SOP) expression for the output.
4. Simplify the output expression if possible.
5. Implement the circuit for the final, simplified expression.

The following example illustrates the complete design procedure.

EXAMPLE 4-7

Design a logic circuit that has three inputs, A, B, and C, and whose output will be HIGH only when a majority of the inputs are HIGH.

Solution

Step 1. Set up the truth table.

On the basis of the problem statement, the output x should be 1 whenever two or more inputs are 1; for all other cases, the output should be 0 (Table 4-2).

TABLE 4-2

A	B	C	x	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \overline{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\overline{B}C$
1	1	0	1	$\rightarrow AB\overline{C}$
1	1	1	1	$\rightarrow ABC$

Step 2. Write the AND term for each case where the output is a 1.

There are four such cases. The AND terms are shown next to the truth table (Table 4-2). Again note that each AND term contains each input variable in either inverted or noninverted form.

Step 3. Write the sum-of-products expression for the output.

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 4. Simplify the output expression.

This expression can be simplified in several ways. Perhaps the quickest way is to realize that the last term ABC has two variables in common with each of the other terms. Thus, we can use the ABC term to pair with each of the other terms. The expression is rewritten with the ABC term occurring three times (recall from Example 4-2 that this is legal in Boolean algebra):

$$x = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

Factoring the appropriate pairs of terms, we have

$$x = BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

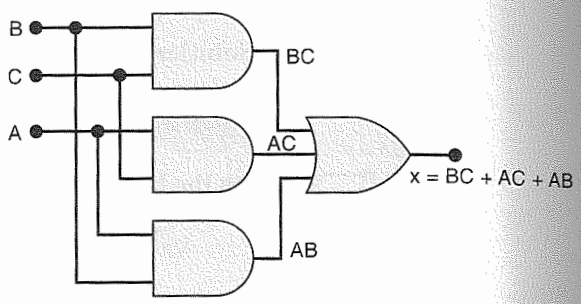
Each term in parentheses is equal to 1, so we have

$$x = BC + AC + AB$$

Step 5. Implement the circuit for the final expression.

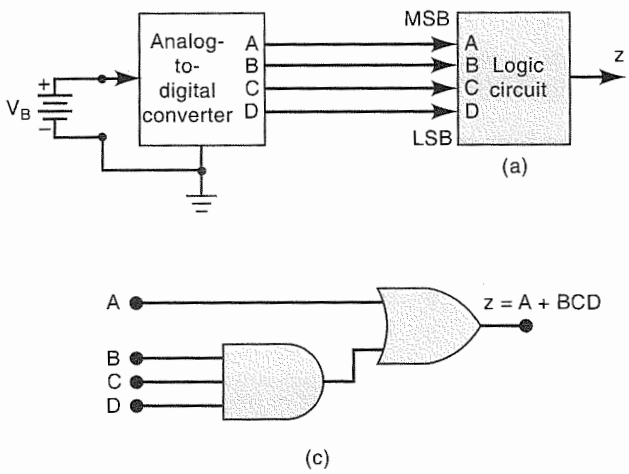
This expression is implemented in Figure 4-7. Since the expression is in SOP form, the circuit consists of a group of AND gates working into a single OR gate.

FIGURE 4-7 Example 4-7.



EXAMPLE 4-8

Refer to Figure 4-8(a), where an analog-to-digital converter is monitoring the dc voltage (V_B) of a 12-V storage battery on an orbiting spaceship. The converter's output is a four-bit binary number, *ABCD*, corresponding to the battery voltage in steps of 1 V, with *A* as the MSB. The converter's binary outputs are fed to a logic circuit that is to produce a HIGH output as long as the binary value is greater than $0110_2 = 6_{10}$; that is, the battery voltage is greater than 6 V. Design this logic circuit.



	A	B	C	D	z
(0)	0	0	0	0	0
(1)	0	0	0	1	0
(2)	0	0	1	0	0
(3)	0	0	1	1	0
(4)	0	1	0	0	0
(5)	0	1	0	1	0
(6)	0	1	1	0	0
(7)	0	1	1	1	1→ $\bar{A}BCD$
(8)	1	0	0	0	1→ $A\bar{B}\bar{C}\bar{D}$
(9)	1	0	0	1	1→ $A\bar{B}\bar{C}D$
(10)	1	0	1	0	1→ $A\bar{B}C\bar{D}$
(11)	1	0	1	1	1→ $A\bar{B}CD$
(12)	1	1	0	0	1→ $AB\bar{C}\bar{D}$
(13)	1	1	0	1	1→ $AB\bar{C}D$
(14)	1	1	1	0	1→ $ABC\bar{D}$
(15)	1	1	1	1	1→ $ABCD$

(b)

FIGURE 4-8 Example 4-8.

Solution

The truth table is shown in Figure 4-8(b). For each case in the truth table, we have indicated the decimal equivalent of the binary number represented by the *ABCD* combination.

The output *z* is set equal to 1 for all those cases where the binary number is greater than 0110. For all other cases, *z* is set equal to 0. This truth table gives us the following sum-of-products expression:

$$z = \bar{A}BCD + A\bar{B}\,\bar{C}\,\bar{D} + A\bar{B}\,\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + AB\bar{C}\,\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$$

Simplification of this expression will be a formidable task, but with a little care it can be accomplished. The step-by-step process involves factoring and eliminating terms of the form $A + \overline{A}$:

$$\begin{aligned} z &= \overline{A}BCD + A\overline{B}\,\overline{C}(\overline{D} + D) + A\overline{B}C(\overline{D} + D) + AB\overline{C}(\overline{D} + D) + ABC(\overline{D} + D) \\ &= \overline{A}BCD + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC \\ &= \overline{A}BCD + A\overline{B}(\overline{C} + C) + AB(\overline{C} + C) \\ &= \overline{A}BCD + A\overline{B} + AB \\ &= \overline{A}BCD + A(\overline{B} + B) \\ &= \overline{A}BCD + A \end{aligned}$$

This can be reduced further by invoking theorem (15a), which says that $x + \overline{x}y = x + y$. In this case $x = A$ and $y = BCD$. Thus,

$$z = \overline{A}BCD + A = BCD + A$$

This final expression is implemented in Figure 4-8(c).

As this example demonstrates, the algebraic simplification method can be quite lengthy when the original expression contains a large number of terms. This is a limitation that is not shared by the Karnaugh mapping method, as we will see later.

EXAMPLE 4-9

Refer to Figure 4-9(a). In a simple copy machine, a stop signal, S, is to be generated to stop the machine operation and energize an indicator light whenever either of the following conditions exists: (1) there is no paper in the paper feeder tray; or (2) the two microswitches in the paper path are

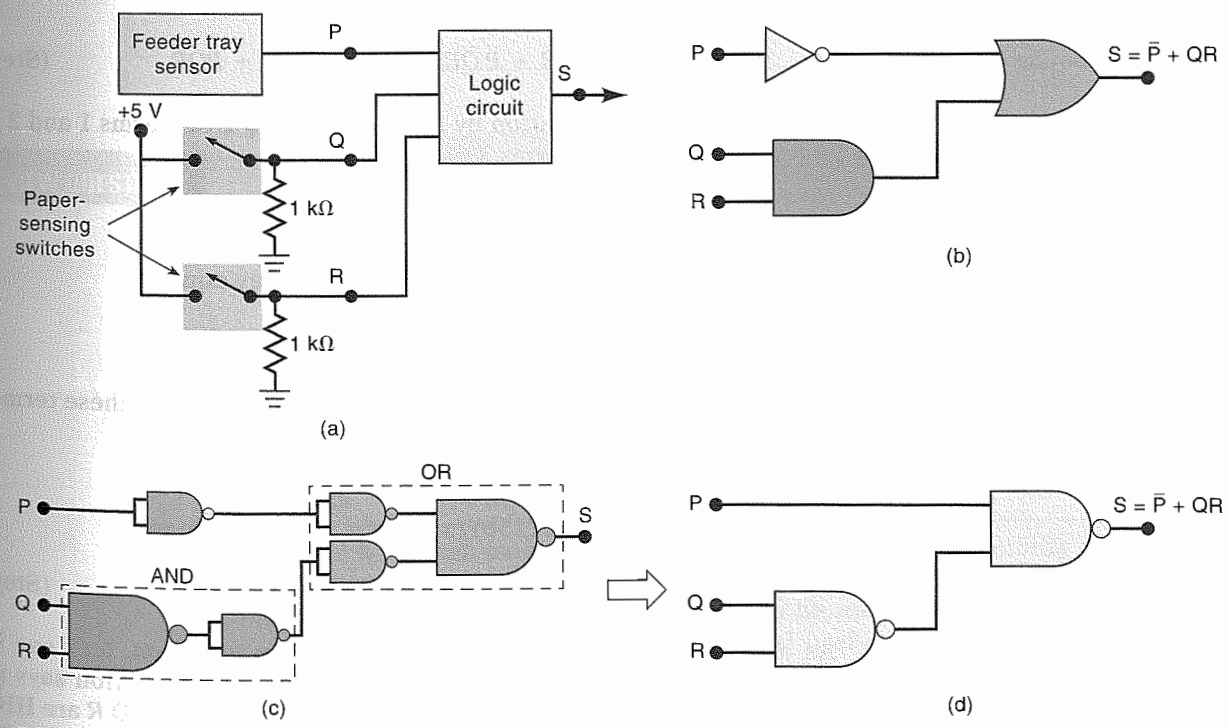


FIGURE 4-9 Example 4-9.

activated, indicating a jam in the paper path. The presence of paper in the feeder tray is indicated by a HIGH at logic signal P. Each of the microswitches produces a logic signal (Q and R) that goes HIGH whenever paper is passing over the switch to activate it. Design the logic circuit to produce a HIGH at output signal S for the stated conditions, and implement it using the 74HC00 CMOS quad two-input NAND chip.

Solution

We will use the five-step process used in Example 4-7. The truth table is shown in Table 4-3. The S output will be a logic 1 whenever $P = 0$ because this indicates no paper in the feeder tray. S will also be a 1 for the two cases where Q and R are both 1, indicating a paper jam. As the table shows, there are five different input conditions that produce a HIGH output. (Step 1)

TABLE 4-3

P	Q	R	S	
0	0	0	1	$\overline{P}\,\overline{Q}\,\overline{R}$
0	0	1	1	$\overline{P}\,\overline{Q}R$
0	1	0	1	$\overline{P}Q\overline{R}$
0	1	1	1	$\overline{P}QR$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	PQR

The AND terms for each of these cases are shown. (Step 2)
The sum-of-products expression becomes

$$S = \overline{P}\,\overline{Q}\,\overline{R} + \overline{P}\,\overline{Q}R + \overline{P}Q\overline{R} + \overline{P}QR + PQR \qquad \text{(Step 3)}$$

We can begin the simplification by factoring out $\overline{P}\,\overline{Q}$ from terms 1 and 2 and by factoring out $\overline{P}Q$ from terms 3 and 4:

$$S = \overline{P}\,\overline{Q}(\overline{R} + R) + \overline{P}Q(\overline{R} + R) + PQR$$

Now we can eliminate the $\overline{R} + R$ terms because they equal 1:

$$S = \overline{P}\,\overline{Q} + \overline{P}Q + PQR$$

Factoring $\overline{P}$ from terms 1 and 2 allows us to eliminate Q from these terms:

$$S = \overline{P} + PQR$$

Here we can apply theorem (15b) ($\overline{x} + xy = \overline{x} + y$) to obtain

$$S = \overline{P} + QR \qquad \text{(Step 4)}$$

As a double check of this simplified Boolean equation, let's see if it matches the truth table that we started out with. This equation says that the output S will be HIGH whenever P is LOW OR both Q AND R are HIGH. Look at Table 4-3 and observe that the output is HIGH for all four cases

when P is LOW. S is also HIGH when Q AND R are both HIGH, regardless of the state of P. This agrees with the equation.

The AND/OR implementation for this circuit is shown in Figure 4-9(b). (Step 5)

To implement this circuit using the 74HC00 quad two-input NAND chip, we must convert each gate and the INVERTER by substituting their NAND-gate equivalents (per Section 3-12). This is shown in Figure 4-9(c). Clearly, we can eliminate two pairs of double inverters to produce the NAND-gate implementation shown in Figure 4-9(d).

The final wired-up circuit is obtained by connecting two of the NAND gates on the 74HC00 chip. This CMOS chip has the same gate configuration and pin numbers as the TTL 74LS00 chip of Figure 3-31. Figure 4-10 shows the wired-up circuit with pin numbers, including the +5 V and GROUND pins. It also includes an output driver transistor and LED to indicate the state of output S.

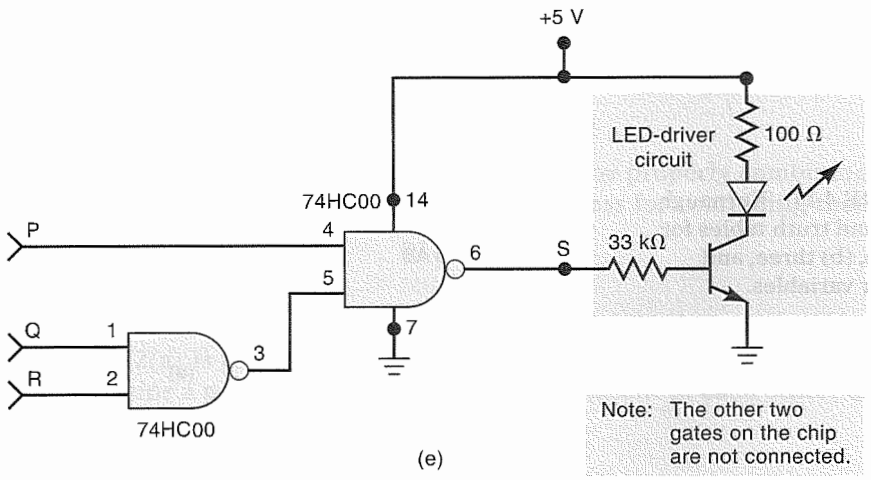


FIGURE 4-10 Circuit of Figure 4-9(d) implemented using 74HC00 NAND chip.

REVIEW QUESTIONS

1. Write the sum-of-products expression for a circuit with four inputs and an output that is to be HIGH only when input A is LOW at the same time that exactly two other inputs are LOW.
2. Implement the expression of question 1 using all four-input NAND gates. How many are required?

4-5 KARNAUGH MAP METHOD

The **Karnaugh map (K map)** is a graphical tool used to simplify a logic equation or to convert a truth table to its corresponding logic circuit in a simple, orderly process. Although a K map can be used for problems involving any number of input variables, its practical usefulness is limited to five or six variables. The following discussion will be limited to problems with up to four inputs because even five- and six-input problems are too involved and are best done by a computer program.

Karnaugh Map Format

The K map, like a truth table, is a means for showing the relationship between logic inputs and the desired output. Figure 4-11 shows three examples of K maps for two, three, and four variables, together with the corresponding truth tables. These examples illustrate the following important points:

1. The truth table gives the value of output X for each combination of input values. The K map gives the same information in a different format. Each case in the truth table corresponds to a square in the K map. For example, in Figure 4-11(a), the $A = 0, B = 0$ condition in the truth table corresponds to the $\overline{A}\,\overline{B}$ square in the K map. Because the truth table shows $X = 1$ for this case, a 1 is placed in the $\overline{A}\,\overline{B}$ square in the K map. Similarly, the $A = 1$, $B = 1$ condition in the truth table corresponds to the AB square of the K map. Because $X = 1$ for this case, a 1 is placed in the AB square. All other squares are filled with 0s. This same idea is used in the three- and four-variable maps shown in the figure.
2. The K-map squares are labeled so that horizontally adjacent squares differ only in one variable. For example, the upper left-hand square in the four-variable map is $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$, while the square immediately to its right is $\overline{A}\,\overline{B}\,\overline{C}D$ (only the D variable is different). Similarly, vertically adjacent

FIGURE 4-11 Karnaugh maps and truth tables for (a) two, (b) three, and (c) four variables.

A	B	X
0	0	1 → $\overline{A}\,\overline{B}$
0	1	0
1	0	0
1	1	1 → AB

$$\left\{ X = \overline{A}\,\overline{B} + AB \right\}$$

	$\overline{B}$	B
$\overline{A}$	1	0
A	0	1

(a)

A	B	C	X
0	0	0	1 → $\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	1 → $\overline{A}\,\overline{B}C$
0	1	0	1 → $\overline{A}B\overline{C}$
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1 → $AB\overline{C}$
1	1	1	0

$$\left\{ \begin{aligned} X = {} & \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C \\ & + \overline{A}B\overline{C} + AB\overline{C} \end{aligned} \right\}$$

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	1	1
$\overline{A}B$	1	0
AB	1	0
$A\overline{B}$	0	0

(b)

A	B	C	D	X
0	0	0	0	0
0	0	0	1	1 → $\overline{A}\,\overline{B}\,\overline{C}D$
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1 → $\overline{A}B\overline{C}D$
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1 → $AB\overline{C}D$
1	1	1	0	0
1	1	1	1	1 → ABCD

$$\left\{ \begin{aligned} X = {} & \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}B\overline{C}D \\ & + AB\overline{C}D + ABCD \end{aligned} \right\}$$

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	0	0
$\overline{A}B$	0	1	0	0
AB	0	1	1	0
$A\overline{B}$	0	0	0	0

(c)

squares differ only in one variable. For example, the upper left-hand square is $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$, while the square directly below it is $\overline{A}B\overline{C}\,\overline{D}$ (only the B variable is different).

Note that each square in the top row is considered to be adjacent to a corresponding square in the bottom row. For example, the $\overline{A}\,\overline{B}CD$ square in the top row is adjacent to the $A\overline{B}CD$ square in the bottom row because they differ only in the A variable. You can think of the top of the map as being wrapped around to touch the bottom of the map. Similarly, squares in the leftmost column are adjacent to corresponding squares in the rightmost column.

3. In order for vertically and horizontally adjacent squares to differ in only one variable, the top-to-bottom labeling must be done in the order shown: $\overline{A}\,\overline{B}$, $\overline{A}B$, AB, $A\overline{B}$. The same is true of the left-to-right labeling: $\overline{C}\,\overline{D}$, $\overline{C}D$, CD, $C\overline{D}$.
4. Once a K map has been filled with 0s and 1s, the sum-of-products expression for the output X can be obtained by ORing together those squares that contain a 1. In the three-variable map of Figure 4-11(b), the $\overline{A}\,\overline{B}\,\overline{C}$, $\overline{A}\,\overline{B}C$, $\overline{A}B\overline{C}$, and $AB\overline{C}$ squares contain a 1, so that $X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C}$.

Looping

The expression for output X can be simplified by properly combining those squares in the K map that contain 1s. The process for combining these 1s is called **looping**.

Looping Groups of Two (Pairs)

Figure 4-12(a) is the K map for a particular three-variable truth table. This map contains a pair of 1s that are vertically adjacent to each other; the first

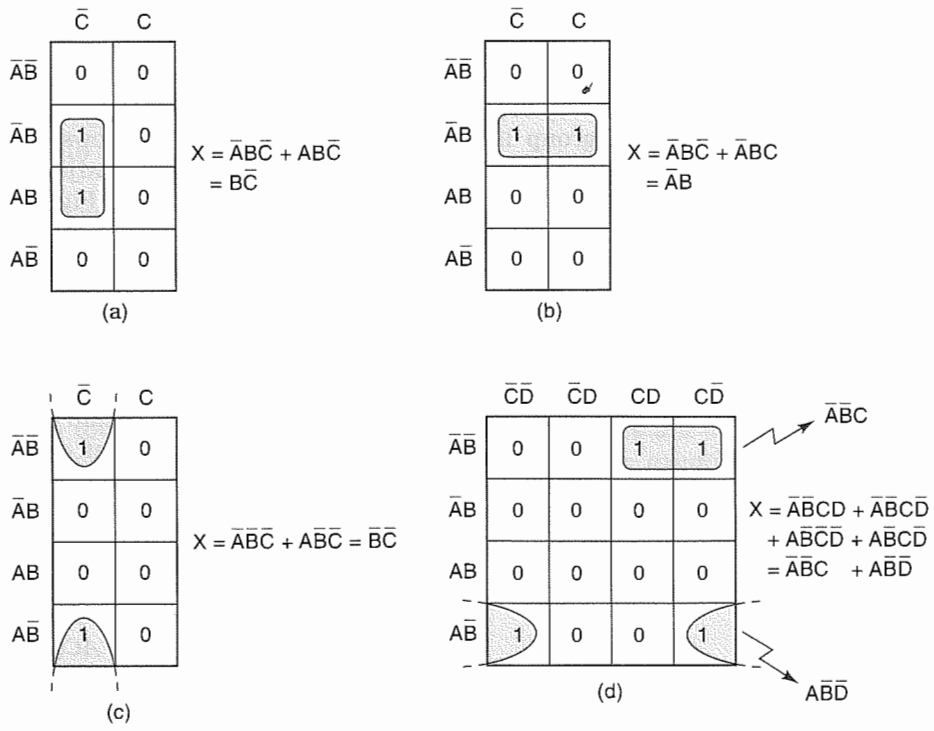


FIGURE 4-12 Examples of looping pairs of adjacent 1s.

represents $\overline{A}B\overline{C}$, and the second represents $AB\overline{C}$. Note that in these two terms only the A variable appears in both normal and complemented (inverted) form, while B and $\overline{C}$ remain unchanged. These two terms can be looped (combined) to give a resultant that eliminates the A variable because it appears in both uncomplemented and complemented forms. This is easily proved as follows:

$$\begin{aligned} X &= \overline{A}B\overline{C} + AB\overline{C} \\ &= B\overline{C}(\overline{A} + A) \\ &= B\overline{C}(1) = B\overline{C} \end{aligned}$$

This same principle holds true for any pair of vertically or horizontally adjacent 1s. Figure 4-12(b) shows an example of two horizontally adjacent 1s. These two can be looped and the C variable eliminated because it appears in both its uncomplemented and complemented forms to give a resultant of $X = \overline{A}B$.

Another example is shown in Figure 4-12(c). In a K map, the top row and bottom row of squares are considered to be adjacent. Thus, the two 1s in this map can be looped to provide a resultant of $\overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} = \overline{B}\,\overline{C}$.

Figure 4-12(d) shows a K map that has two pairs of 1s that can be looped. The two 1s in the top row are horizontally adjacent. The two 1s in the bottom row are also adjacent because, in a K map, the leftmost column and the rightmost column of squares are considered to be adjacent. When the top pair of 1s is looped, the D variable is eliminated (because it appears as both D and $\overline{D}$) to give the term $\overline{A}\,\overline{B}C$. Looping the bottom pair eliminates the C variable to give the term $A\overline{B}\,\overline{D}$. These two terms are ORed to give the final result for X.

To summarize:

Looping a pair of adjacent 1s in a K map eliminates the variable that appears in complemented and uncomplemented form.

Looping Groups of Four (Quads)

A K map may contain a group of four 1s that are adjacent to each other. This group is called a *quad.* Figure 4-13 shows several examples of quads. In Figure 4-13(a), the four 1s are vertically adjacent, and in Figure 4-13(b), they are horizontally adjacent. The K map in Figure 4-13(c) contains four 1s in a square, and they are considered adjacent to each other. The four 1s in Figure 4-13(d) are also adjacent, as are those in Figure 4-13(e), because, as pointed out earlier, the top and bottom rows are considered to be adjacent to each other, as are the leftmost and rightmost columns.

When a quad is looped, the resultant term will contain only the variables that do not change form for all the squares in the quad. For example, in Figure 4-13(a), the four squares that contain a 1 are $\overline{A}\,\overline{B}C$, $\overline{A}BC$, ABC, and $A\overline{B}C$. Examination of these terms reveals that only the variable C remains unchanged (both A and B appear in complemented and uncomplemented form). Thus, the resultant expression for X is simply $X = C$. This can be proved as follows:

$$\begin{aligned} X &= \overline{A}\,\overline{B}C + \overline{A}BC + ABC + A\overline{B}C \\ &= \overline{A}C(\overline{B} + B) + AC(B + \overline{B}) \\ &= \overline{A}C + AC \\ &= C(\overline{A} + A) = C \end{aligned}$$

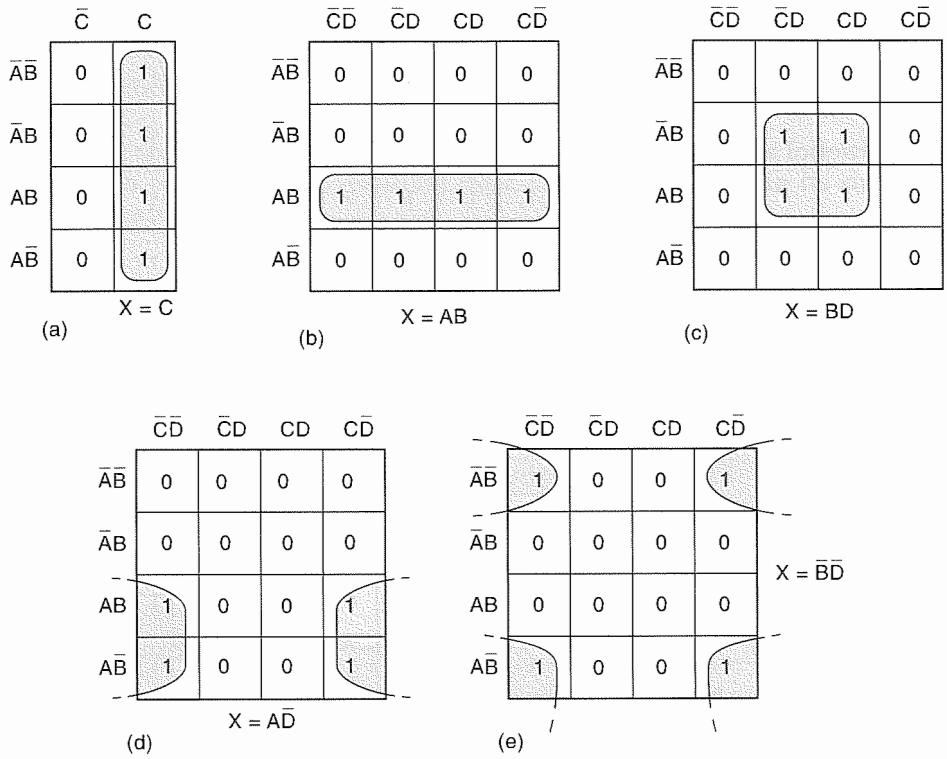


FIGURE 4-13 Examples of looping groups of four 1s (quads).

As another example, consider Figure 4-13(d), where the four squares containing 1s are $AB\bar{C}\bar{D}$, $A\bar{B}\bar{C}\bar{D}$, $ABC\bar{D}$, and $A\bar{B}C\bar{D}$. Examination of these terms indicates that only the variables A and $\bar{D}$ remain unchanged, so that the simplified expression for X is

$$X = A\bar{D}$$

This can be proved in the same manner that was used above. The reader should check each of the other cases in Figure 4-13 to verify the indicated expressions for X.

To summarize:

Looping a quad of adjacent 1s eliminates the two variables that appear in both complemented and uncomplemented form.

Looping Groups of Eight (Octets)

A group of eight 1s that are adjacent to one another is called an *octet*. Several examples of octets are shown in Figure 4-14. When an octet is looped in a four-variable map, three of the four variables are eliminated because only one variable remains unchanged. For example, examination of the eight looped squares in Figure 4-14(a) shows that only the variable B is in the same form for all eight squares: the other variables appear in complemented and uncomplemented form. Thus, for this map, $X = B$. The reader can verify the results for the other examples in Figure 4-14.

To summarize:

Looping an octet of adjacent 1s eliminates the three variables that appear in both complemented and uncomplemented form.

Step 5 Squares 6, 7, 10, and 11 form a quad. This quad is looped (loop 6, 7, 10, 11). Note that square 11 is used again, even though it was part of loop 11, 15.

Step 6 All 1s have already been looped.

Step 7 Each loop generates a term in the expression for X. Loop 4 is simply $\overline{A}\,\overline{B}C\overline{D}$. Loop 11, 15 is ACD (the B variable is eliminated). Loop 6, 7, 10, 11 is BD (A and C are eliminated).

EXAMPLE 4-11

Consider the K map in Figure 4-15(b). Once again, we can assume that step 1 has already been performed. Simplify.

Solution

Step 2 There are no isolated 1s.

Step 3 The 1 in square 3 is adjacent *only* to the 1 in square 7. Looping this pair (loop 3, 7) produces the term $\overline{A}CD$.

Step 4 There are no octets.

Step 5 There are two quads. Squares 5, 6, 7, and 8 form one quad. Looping this quad produces the term $\overline{A}B$. The second quad is made up of squares 5, 6, 9, and 10. This quad is looped because it contains two squares that have not been looped previously. Looping this quad produces $B\overline{C}$.

Step 6 All 1s have already been looped.

Step 7 The terms generated by the three loops are ORed together to obtain the expression for X.

EXAMPLE 4-12

Consider the K map in Figure 4-15(c). Simplify.

Solution

Step 2 There are no isolated 1s.

Step 3 The 1 in square 2 is adjacent only to the 1 in square 6. This pair is looped to produce $\overline{A}\,\overline{C}D$. Similarly, square 9 is adjacent only to square 10. Looping this pair produces $AB\overline{C}$. Likewise, loop 7, 8 and loop 11, 15 produce the terms $\overline{A}BC$ and ACD, respectively.

Step 4 There are no octets.

Step 5 There is one quad formed by squares 6, 7, 10, and 11. This quad, however, is *not* looped because all the 1s in the quad have been included in other loops.

Step 6 All 1s have already been looped.

Step 7 The expression for X is shown in the figure.

EXAMPLE 4-13

Consider the two K map loopings in Figure 4-16. Is one better than the other?

Solution

Step 2 There are no isolated 1s.

Step 3 There are no 1s that are adjacent to only one other 1.

Step 4 There are no octets.

Step 5 There are no quads.

Steps 6 and 7 There are many possible pairs. The looping must use the minimum number of loops to account for all the 1s. For this map, there are

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	0	0
$\overline{A}B$	0	1	1	1
AB	0	0	0	1
$A\overline{B}$	1	1	0	1

$X = \overline{A}\overline{C}D + \overline{A}BC + A\overline{B}\overline{C} + AC\overline{D}$

(a)

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	0	0
$\overline{A}B$	0	1	1	1
AB	0	0	0	1
$A\overline{B}$	1	1	0	1

$X = \overline{A}BD + BC\overline{D} + \overline{B}\overline{C}D + A\overline{B}\overline{D}$

(b)

FIGURE 4-16 The same K map with two equally good solutions.

two possible loopings, which require only four looped pairs. Figure 4-16(a) shows one solution and its resultant expression. Figure 4-16(b) shows the other. Note that both expressions are of the same complexity, and so neither is better than the other.

Filling a K Map from an Output Expression

When the desired output is presented as a Boolean expression instead of a truth table, the K map can be filled by using the following steps:

1. Get the expression into SOP form if it is not already in that form.
2. For each product term in the SOP expression, place a 1 in each K-map square whose label contains the same combination of input variables. Place a 0 in all other squares.

The following example illustrates this procedure.

EXAMPLE 4-14

Use a K map to simplify $y = \overline{C}(\overline{A}\,\overline{B}\,\overline{D} + D) + A\overline{B}C + \overline{D}$.

Solution

1. Multiply out the first term to get $y = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{C}D + A\overline{B}C + \overline{D}$, which is now in SOP form.
2. For the $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$ term, simply put a 1 in the $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$ square of the K map (Figure 4-17). For the $\overline{C}D$ term, place a 1 in all squares with $\overline{C}D$ in their labels, that is, $\overline{A}\,\overline{B}\,\overline{C}D$, $\overline{A}B\overline{C}D$, $AB\overline{C}D$, $A\overline{B}\,\overline{C}D$. For the $A\overline{B}C$ term, place a 1 in all squares that have an $A\overline{B}C$ in their labels, that is, $A\overline{B}C\overline{D}$, $A\overline{B}CD$. For the $\overline{D}$ term, place a 1 in all squares that have a $\overline{D}$ in their labels, that is, all squares in the leftmost and rightmost columns.

FIGURE 4-17 Example 4-14.

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	1	1	0	1
$\overline{A}B$	1	1	0	1
AB	1	1	0	1
$A\overline{B}$	1	1	1	1

$y = A\overline{B} + \overline{C} + \overline{D}$

The K map is now filled and can be looped for simplification. Verify that proper looping produces $y = A\overline{B} + \overline{C} + \overline{D}$.

Don't-Care Conditions

Some logic circuits can be designed so that there are certain input conditions for which there are no specified output levels, usually because these input conditions will never occur. In other words, there will be certain combinations of input levels where we "don't care" whether the output is 1 or 0. This is illustrated in the truth table of Figure 4-18(a).

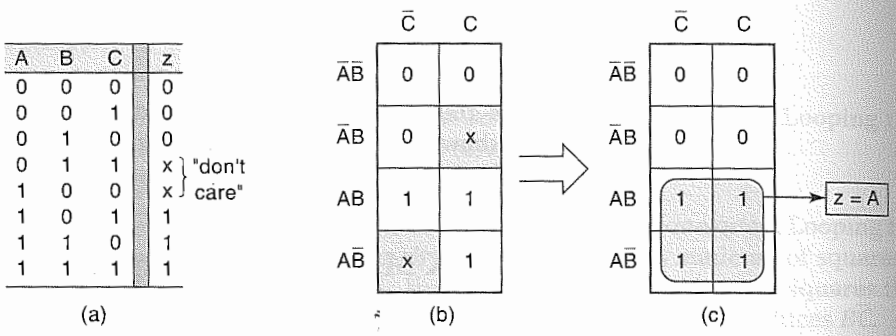


FIGURE 4-18 "Don't-care" conditions should be changed to 0 or 1 to produce K-map looping that yields the simplest expression.

Here the output z is not specified as either 0 or 1 for the conditions $A, B, C = 1, 0, 0$ and $A, B, C = 0, 1, 1$. Instead, an x is shown for these conditions. The x represents the **don't-care condition**. A don't-care condition can come about for several reasons, the most common being that in some situations certain input combinations can never occur, and so there is no specified output for these conditions.

A circuit designer is free to make the output for any don't-care condition either a 0 or a 1 to produce the simplest output expression. For example, the K map for this truth table is shown in Figure 4-18(b) with an x placed in the $A\overline{B}\overline{C}$ and $\overline{A}BC$ squares. The designer here would be wise to change the x in the $A\overline{B}\overline{C}$ square to a 1 and the x in the $\overline{A}BC$ square to a 0 because this would produce a quad that can be looped to produce $z = A$, as shown in Figure 4-18(c).

Whenever don't-care conditions occur, we must decide which x to change to 0 and which to 1 to produce the best K-map looping (i.e., the simplest expression). This decision is not always an easy one. Several end-of-chapter problems will provide practice in dealing with don't-care cases. Here's another example.

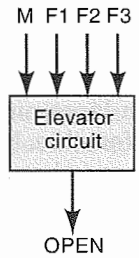
EXAMPLE 4-15

Let's design a logic circuit that controls an elevator door in a three-story building. The circuit in Figure 4-19(a) has four inputs. M is a logic signal that indicates when the elevator is moving ($M = 1$) or stopped ($M = 0$). $F1$, $F2$, and $F3$ are floor indicator signals that are normally LOW, and they go HIGH only when the elevator is positioned at the level of that particular floor. For example, when the elevator is lined up level with the second floor, $F2 = 1$ and $F1 = F3 = 0$. The circuit output is the *OPEN* signal, which is normally LOW and will go HIGH when the elevator door is to be opened.

We can fill in the truth table for the *OPEN* output [Figure 4-19(b)] as follows:

1. Because the elevator cannot be lined up with more than one floor at a time, only one of the floor inputs can be HIGH at any given time. This means that all those cases in the truth table where more than one floor

FIGURE 4-19 Example 4-15.



(a)

M	F1	F2	F3	OPEN
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	X
0	1	0	0	1
0	1	0	1	X
0	1	1	0	X
0	1	1	1	X
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	X
1	1	0	0	0
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

(b)

	$\overline{F2}\,\overline{F3}$	$\overline{F2}F3$	$F2F3$	$F2\overline{F3}$
$\overline{M}\,\overline{F1}$	0	1	X	1
$\overline{M}F1$	1	X	X	X
$MF1$	0	X	X	X
$M\overline{F1}$	0	0	X	0

(c)

	$\overline{F2}\,\overline{F3}$	$\overline{F2}F3$	$F2F3$	$F2\overline{F3}$
$\overline{M}\,\overline{F1}$	0	1	1	1
$\overline{M}F1$	1	1	1	1
$MF1$	0	0	0	0
$M\overline{F1}$	0	0	0	0

$OPEN = \overline{M}\,(F1 + F2 + F3)$

(d)

input is a 1 are don't-care conditions. We can place an x in the *OPEN* output column for those eight cases where more than one F input is 1.

2. Looking at the other eight cases, when $M = 1$ the elevator is moving, so *OPEN* must be a 0 because we do not want the elevator door to open. When $M = 0$ (elevator stopped) we want $OPEN = 1$ provided that one of the floor inputs is 1. When $M = 0$ and all floor inputs are 0, the elevator is stopped but is not properly lined up with any floor, so we want $OPEN = 0$ to keep the door closed.

The truth table is now complete and we can transfer its information to the K map in Figure 4-19(c). The map has only three 1s, but it has eight don't-cares. By changing four of these don't-care squares to 1s, we can produce quad loopings that contain the original 1s [Figure 4-19(d)]. This is the best we can do as far as minimizing the output expression. Verify that the loopings produce the *OPEN* output expression shown.

Summary

The K-map process has several advantages over the algebraic method. K mapping is a more orderly process with well-defined steps compared with the trial-and-error process sometimes used in algebraic simplification. K mapping usually requires fewer steps, especially for expressions containing many terms, and it always produces a minimum expression.

Nevertheless, some instructors prefer the algebraic method because it requires a thorough knowledge of Boolean algebra and is not simply a mechanical procedure. Each method has its advantages, and although most logic designers are adept at both, being proficient in one method is all that is necessary to produce acceptable results.

There are other, more complex techniques that designers use to minimize logic circuits with more than four inputs. These techniques are especially suited for circuits with large numbers of inputs where algebraic and K-mapping methods are not feasible. Most of these techniques can be translated into a computer program that will perform the minimization from input data that supply the truth table or the unsimplified expression.

REVIEW QUESTIONS

1. Use K mapping to obtain the expression of Example 4-7.
2. Use K mapping to obtain the expression of Example 4-8. This should emphasize the advantage of K mapping for expressions containing many terms.
3. Obtain the expression of Example 4-9 using a K map.
4. What is a don't-care condition?

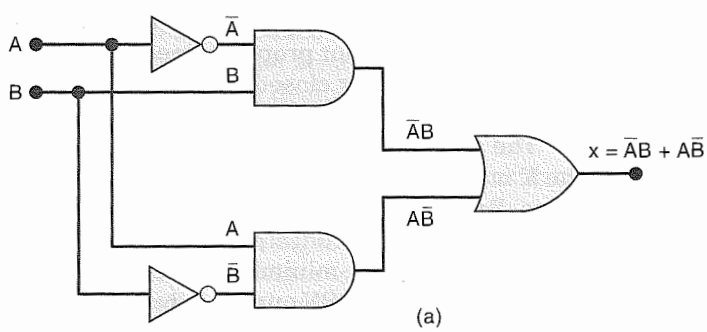
4-6 EXCLUSIVE-OR AND EXCLUSIVE-NOR CIRCUITS

Two special logic circuits that occur quite often in digital systems are the *exclusive-OR* and *exclusive-NOR* circuits.

Exclusive-OR

Consider the logic circuit of Figure 4-20(a). The output expression of this circuit is

$$x = \overline{A}B + A\overline{B}$$



A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

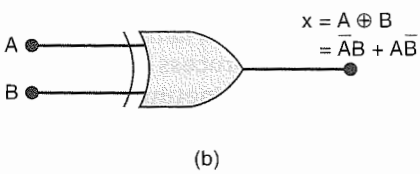


FIGURE 4-20
(a) Exclusive-OR circuit and truth table; (b) traditional XOR gate symbol.

The accompanying truth table shows that $x = 1$ for two cases: $A = 0, B = 1$ (the $\overline{A}B$ term) and $A = 1, B = 0$ (the $A\overline{B}$ term). In other words:

This circuit produces a HIGH output whenever the two inputs are at opposite levels.

This is the **exclusive-OR** circuit, which will hereafter be abbreviated **XOR**.

This particular combination of logic gates occurs quite often and is very useful in certain applications. In fact, the XOR circuit has been given a symbol of its own, shown in Figure 4-20(b). This symbol is assumed to contain all of the logic contained in the XOR circuit and therefore has the same logic expression and truth table. This XOR circuit is commonly referred to as an XOR *gate,* and we consider it as another type of logic gate.

An XOR gate has only *two* inputs; there are no three-input or four-input XOR gates. The two inputs are combined so that $x = \overline{A}B + A\overline{B}$. A shorthand way that is sometimes used to indicate the XOR output expression is

$$x = A \oplus B$$

where the symbol $\oplus$ represents the XOR gate operation.

The characteristics of an XOR gate are summarized as follows:

1. It has only two inputs and its output is

$$x = \overline{A}B + A\overline{B} = A \oplus B$$

2. Its output is *HIGH* only when the two inputs are at *different* levels.

Several ICs are available that contain XOR gates. Those listed below are *quad* XOR chips containing four XOR gates.

74LS86	Quad XOR (TTL family)
74C86	Quad XOR (CMOS family)
74HC86	Quad XOR (high-speed CMOS)

Exclusive-NOR

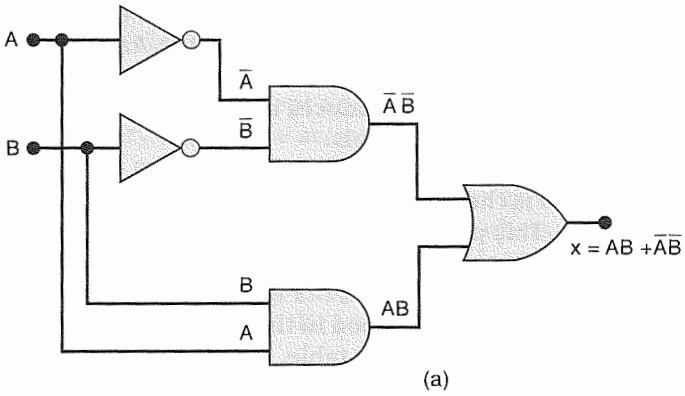
The **exclusive-NOR** circuit (abbreviated **XNOR**) operates completely opposite to the XOR circuit. Figure 4-21(a) shows an XNOR circuit and its accompanying truth table. The output expression is

$$x = AB + \overline{A}\,\overline{B}$$

which indicates along with the truth table that x will be 1 for two cases: $A = B = 1$ (the AB term) and $A = B = 0$ (the $\overline{A}\,\overline{B}$ term). In other words:

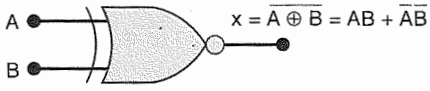
The XNOR produces a HIGH output whenever the two inputs are at the same level.

It should be apparent that the output of the XNOR circuit is the exact inverse of the output of the XOR circuit. The traditional symbol for an XNOR gate is obtained by simply adding a small circle at the output of the XOR symbol [Figure 4-21(b)].



(a)

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1



(b)

FIGURE 4-21
(a) Exclusive-NOR circuit; (b) traditional symbol for XNOR gate.

The XNOR gate also has *only two* inputs, and it combines them so that its output is

$$x = AB + \overline{A}\,\overline{B}$$

A shorthand way to indicate the output expression of the XNOR is

$$x = \overline{A \oplus B}$$

which is simply the inverse of the XOR operation. The XNOR gate is summarized as follows:

1. It has only two inputs and its output is

$$x = AB + \overline{A}\,\overline{B} = \overline{A \oplus B}$$

2. Its output is HIGH only when the two inputs are at the *same* level.

Several ICs are available that contain XNOR gates. Those listed below are quad XNOR chips containing four XNOR gates.

74LS266	Quad XNOR (TTL family)
74C266	Quad XNOR (CMOS)
74HC266	Quad XNOR (high-speed CMOS)

Each of these XNOR chips, however, has special output circuitry that limits its use to special types of applications. Very often, a logic designer will obtain the XNOR function simply by connecting the output of an XOR to an INVERTER.

EXAMPLE 4-16

Determine the output waveform for the input waveforms given in Figure 4-22.

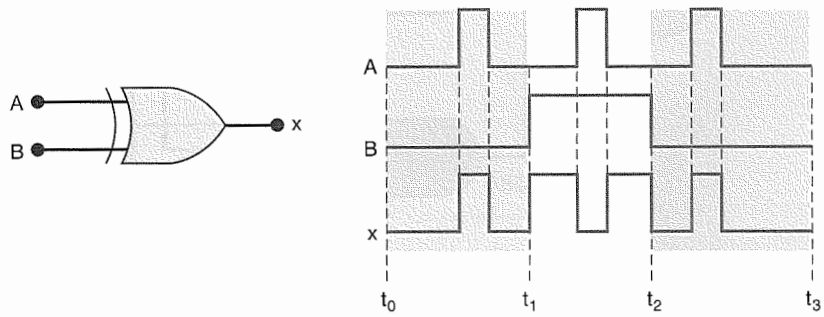


FIGURE 4-22
Example 4-16.

Solution

The output waveform is obtained using the fact that the XOR output will go HIGH only when its inputs are at different levels. The resulting output waveform reveals several interesting points:

1. The x waveform matches the A input waveform during those time intervals when $B = 0$. This occurs during the time intervals t_0 to t_1 and t_2 to t_3.
2. The x waveform is the *inverse* of the A input waveform during those time intervals when $B = 1$. This occurs during the interval t_1 to t_2.
3. These observations show that an XOR gate can be used as a *controlled INVERTER*; that is, one of its inputs can be used to control whether or not the signal at the other input will be inverted. This property will be useful in certain applications.

EXAMPLE 4-17

The notation x_1x_0 represents a two-bit binary number that can have any value (00, 01, 10, or 11); for example, when $x_1 = 1$ and $x_0 = 0$, the binary number is 10, and so on. Similarly, y_1y_0 represents another two-bit binary number. Design a logic circuit, using x_1, x_0, y_1, and y_0 inputs, whose output will be HIGH only when the two binary numbers x_1x_0 and y_1y_0 are *equal*.

Solution

The first step is to construct a truth table for the 16 input conditions (Table 4-4). The output z must be HIGH whenever the x_1x_0 values match the y_1y_0 values; that is, whenever $x_1 = y_1$ and $x_0 = y_0$. The table shows that there are four such cases. We could now continue with the normal procedure, which would be to obtain a sum-of-products expression for z, attempt to simplify it, and then implement the result. However, the nature of this problem makes it ideally suited for implementation using XNOR gates, and a little thought will produce a simple solution with minimum work. Refer to Figure 4-23; in this logic diagram, x_1 and y_1 are fed to one XNOR gate, and x_0 and y_0 are fed to another XNOR gate. The output of each XNOR will be HIGH only when its inputs are equal. Thus, for $x_0 = y_0$ and $x_1 = y_1$, both XNOR outputs will be HIGH. This is the condition we are

TABLE 4-4

x_1	x_0	y_1	y_0	z (Output)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

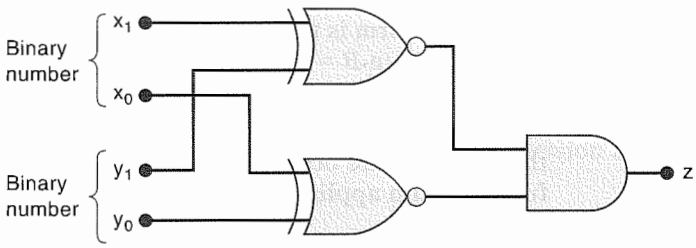
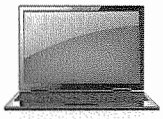


FIGURE 4-23 Circuit for detecting equality of two two-bit binary numbers.



looking for because it means that the two two-bit numbers are equal. The AND gate output will be HIGH only for this case, thereby producing the desired output.

EXAMPLE 4-18

When simplifying the expression for the output of a combinational logic circuit, you may encounter the XOR or XNOR operations as you are factoring. This will often lead to the use of XOR or XNOR gates in the implementation of the final circuit. To illustrate, simplify the circuit of Figure 4-24(a).

Solution

The unsimplified expression for the circuit is obtained as

$$z = ABCD + A\overline{B}\,\overline{C}D + \overline{A}\,\overline{D}$$

We can factor AD from the first two terms:

$$z = AD(BC + \overline{B}\,\overline{C}) + \overline{A}\,\overline{D}$$

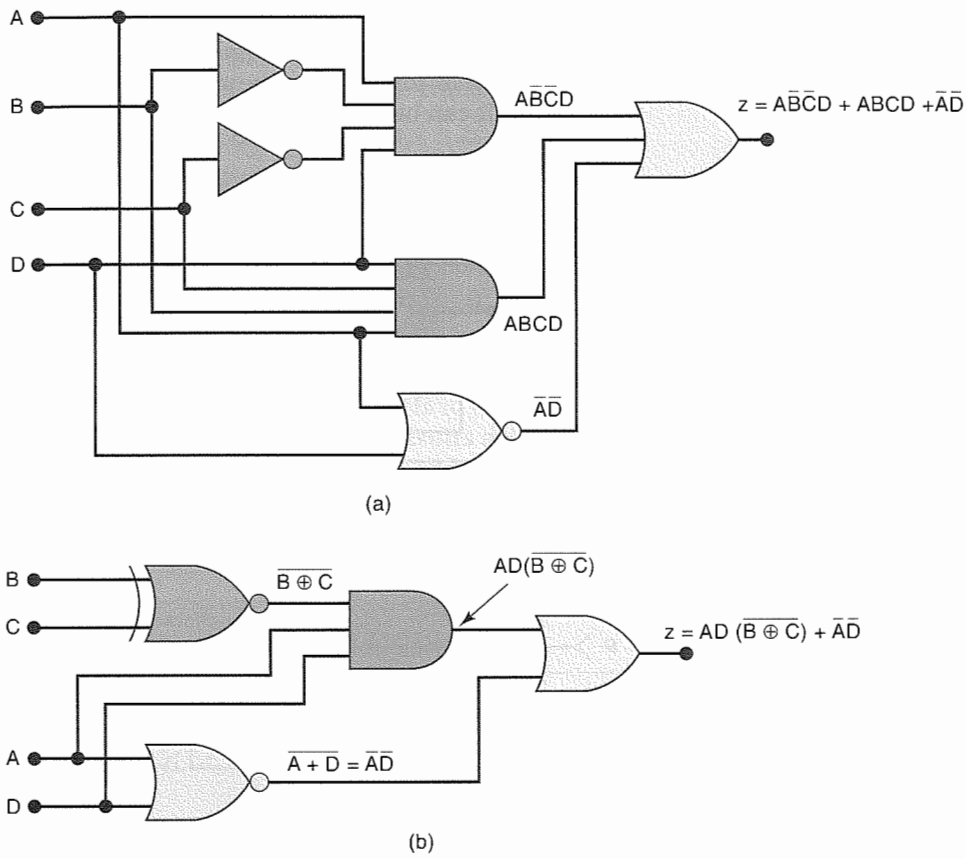


FIGURE 4-24 Example 4-18, showing how an XNOR gate may be used to simplify circuit implementation.

At first glance, you might think that the expression in parentheses can be replaced by 1. But that would be true only if it were $BC + \overline{B}\overline{C}$. You should recognize the expression in parentheses as the XNOR combination of B and C. This fact can be used to reimplement the circuit as shown in Figure 4-24(b). This circuit is much simpler than the original because it uses gates with fewer inputs and two INVERTERs have been eliminated.

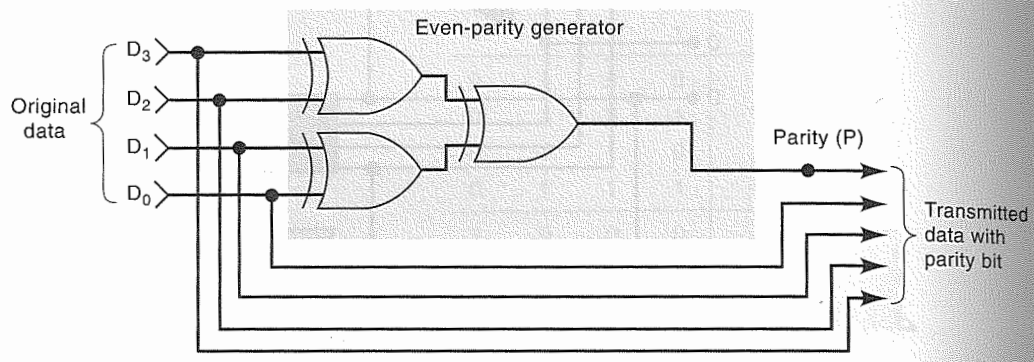
REVIEW QUESTIONS

1. Use Boolean algebra to prove that the XNOR output expression is the exact inverse of the XOR output expression.
2. What is the output of an XNOR gate when a logic signal and its exact inverse are connected to its inputs?
3. A logic designer needs an INVERTER, and all that is available is one XOR gate from a 74HC86 chip. Does he need another chip?

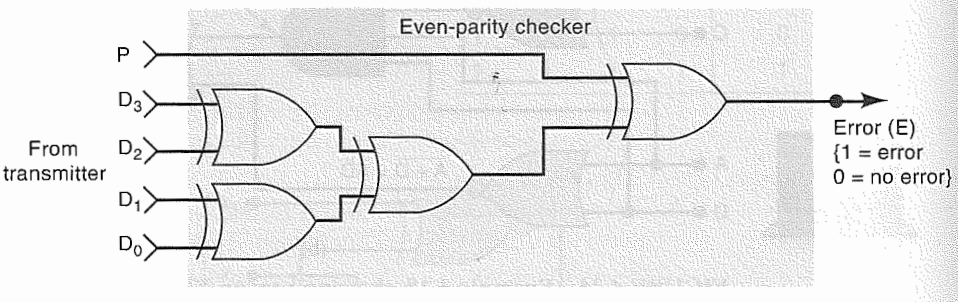
4-7 PARITY GENERATOR AND CHECKER

In Chapter 2, we saw that a transmitter can attach a parity bit to a set of data bits before transmitting the data bits to a receiver. We also saw how this allows the receiver to detect any single-bit errors that may have occurred during the

transmission. Figure 4-25 shows an example of one type of logic circuitry that is used for **parity generation** and **parity checking**. This particular example uses a group of four bits as the data to be transmitted, and it uses an even-parity bit. It can be readily adapted to use odd parity and any number of bits.



(a)



(b)

FIGURE 4-25 XOR gates used to implement (a) the parity generator and (b) the parity checker for an even-parity system.

In Figure 4-25(a), the set of data to be transmitted is applied to the parity-generator circuit, which produces the even-parity bit, P, at its output. This parity bit is transmitted to the receiver along with the original data bits, making a total of five bits. In Figure 4-25(b), these five bits (data + parity) enter the receiver's parity-checker circuit, which produces an error output, E, that indicates whether or not a single-bit error has occurred.

It should not be too surprising that both of these circuits employ XOR gates when we consider that a single XOR gate operates so that it produces a 1 output if an odd number of its inputs are 1, and a 0 output if an even number of its inputs are 1.

EXAMPLE 4-19

Determine the parity generator's output for each of the following sets of input data, $D_3D_2D_1D_0$: (a) 0111; (b) 1001; (c) 0000; (d) 0100. Refer to Figure 4-25(a).

Solution

For each case, apply the data levels to the parity-generator inputs and trace them through each gate to the P output. The results are: (a) 1; (b) 0; (c) 0; and (d) 1. Note that P is a 1 only when the original data contain an odd number of 1s. Thus, the total number of 1s sent to the receiver (data + parity) will be even.

EXAMPLE 4-20

Determine the parity checker's output [see Figure 4-25(b)] for each of the following sets of data from the transmitter:

	P	D_3	D_2	D_1	D_0
(a)	0	1	0	1	0
(b)	1	1	1	1	0
(c)	1	1	1	1	1
(d)	1	0	0	0	0

Solution

For each case, apply these levels to the parity-checker inputs and trace them through to the E output. The results are: (a) 0; (b) 0; (c) 1; (d) 1. Note that a 1 is produced at E only when an odd number of 1s appears in the inputs to the parity checker. This indicates that an error has occurred because even parity is being used.

4-8 ENABLE/DISABLE CIRCUITS

Each of the basic logic gates can be used to control the passage of an input logic signal through to the output. This is depicted in Figure 4-26, where a logic signal, A, is applied to one input of each of the basic logic gates. The

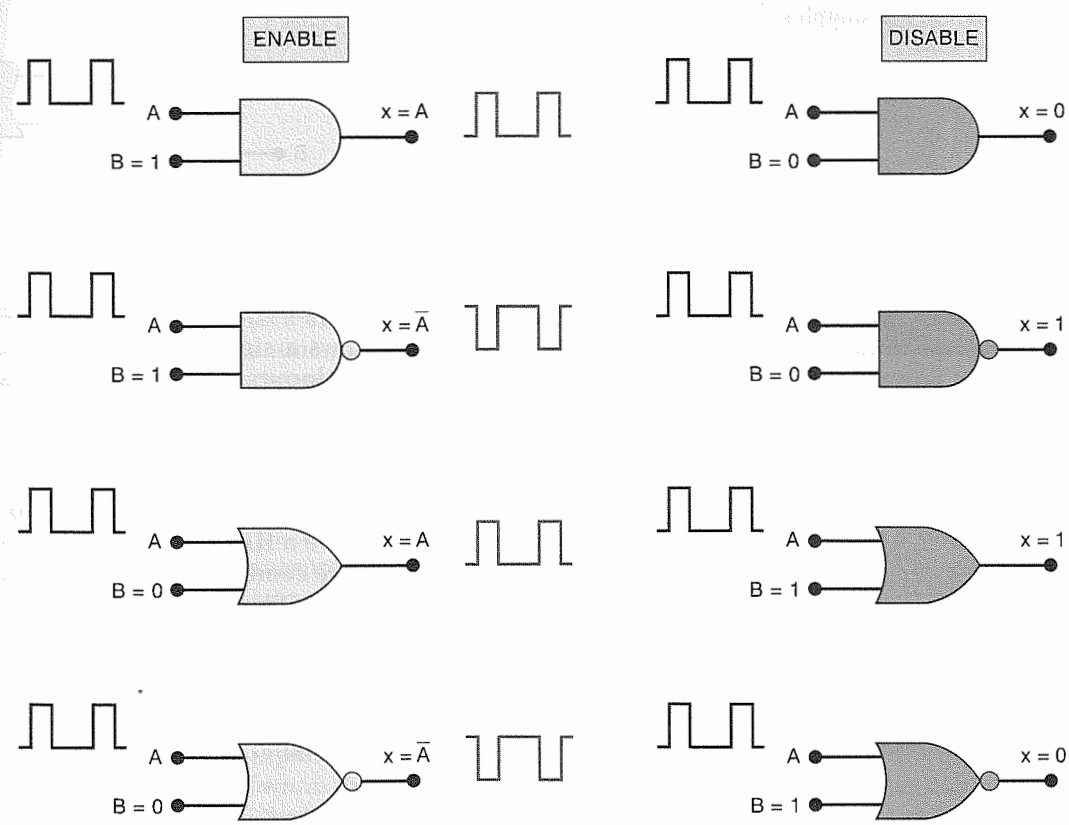


FIGURE 4-26 Four basic gates can either enable or disable the passage of an input signal, A, under control of the logic level at control input B.

other input of each gate is the control input, B. The logic level at this control input will determine whether the input signal is **enabled** to reach the output or **disabled** from reaching the output. This controlling action is why these circuits came to be called *gates*.

Examine Figure 4-26 and you should notice that when the noninverting gates (AND, OR) are enabled, the output will follow the A signal exactly. Conversely, when the inverting gates (NAND, NOR) are enabled, the output will be the exact inverse of the A signal.

Also notice in the figure that AND and NOR gates produce a constant LOW output when they are in the disabled condition. Conversely, the NAND and OR gates produce a constant HIGH output in the disabled condition.

There will be many situations in digital-circuit design where the passage of a logic signal is to be enabled or disabled, depending on conditions present at one or more control inputs. Several are shown in the following examples.

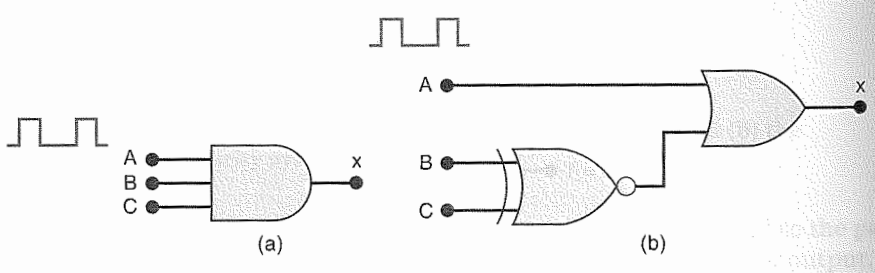
EXAMPLE 4-21

Design a logic circuit that will allow a signal to pass to the output only when control inputs B and C are both HIGH; otherwise, the output will stay LOW.

Solution

An AND gate should be used because the signal is to be passed without inversion, and the disable output condition is a LOW. Because the enable condition must occur only when $B = C = 1$, a three-input AND gate is used, as shown in Figure 4-27(a).

FIGURE 4-27 Examples 4-21 and 4-22.



EXAMPLE 4-22

Design a logic circuit that allows a signal to pass to the output only when one, but not both, of the control inputs are HIGH; otherwise, the output will stay HIGH.

Solution

The result is drawn in Figure 4-27(b). An OR gate is used because we want the output disable condition to be a HIGH, and we do not want to invert the signal. Control inputs B and C are combined in an XNOR gate. When B and C are different, the XNOR sends a LOW to enable the OR gate. When B and C are the same, the XNOR sends a HIGH to disable the OR gate.

EXAMPLE 4-23

Design a logic circuit with input signal A, control input B, and outputs X and Y to operate as follows:

1. When $B = 1$, output X will follow input A, and output Y will be 0.
2. When $B = 0$, output X will be 0, and output Y will follow input A.

Solution

The two outputs will be 0 when they are disabled and will follow the input signal when they are enabled. Thus, an AND gate should be used for each output. Because X is to be enabled when $B = 1$, its AND gate must be controlled by B, as shown in Figure 4-28. Because Y is to be enabled when $B = 0$, its AND gate is controlled by $\overline{B}$. The circuit in Figure 4-28 is called a *pulse-steering circuit* because it steers the input pulse to one output or the other, depending on B.

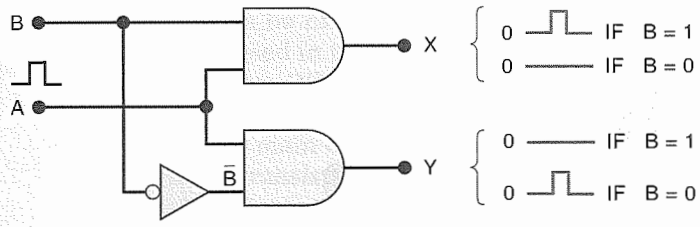
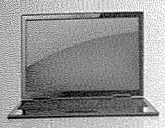


FIGURE 4-28 Example 4-23.



REVIEW QUESTIONS

1. Design a logic circuit with three inputs A, B, C and an output that goes LOW only when A is HIGH while B and C are different.
2. Which logic gates produce a 1 output in the disabled state?
3. Which logic gates pass the inverse of the input signal when they are enabled?

4-9 BASIC CHARACTERISTICS OF DIGITAL ICs

Digital ICs are a collection of resistors, diodes, and transistors fabricated on a single piece of semiconductor material (usually silicon) called a *substrate*, which is commonly referred to as a *chip*. The chip is enclosed in a protective plastic or ceramic package from which pins extend for connecting the IC to other devices. One of the more common types of package is the **dual-in-line package (DIP)**, shown in Figure 4-29(a), so called because it contains two parallel rows of pins. The pins are numbered counterclockwise when viewed from the top of the package with respect to an identifying notch or dot at one end of the package [see Figure 4-29(b)]. The DIP shown here is a 14-pin package that measures 0.75 in. by 0.25 in.; 16-, 20-, 24-, 28-, 40-, and 64-pin packages are also used.

Figure 4-29(c) shows that the actual silicon chip is much smaller than the DIP; typically, it might be a 0.05-in. square. The silicon chip is connected to the pins of the DIP by very fine wires (1-mil diameter).

The DIP is probably the most common digital IC package found in older digital equipment, but other types are becoming more and more popular. The IC shown in Figure 4-29(d) is only one of the many packages common to modern digital circuits. This particular package uses J-shaped leads that curl under the IC. We will take a look at some of these other types of IC packages in Chapter 8.

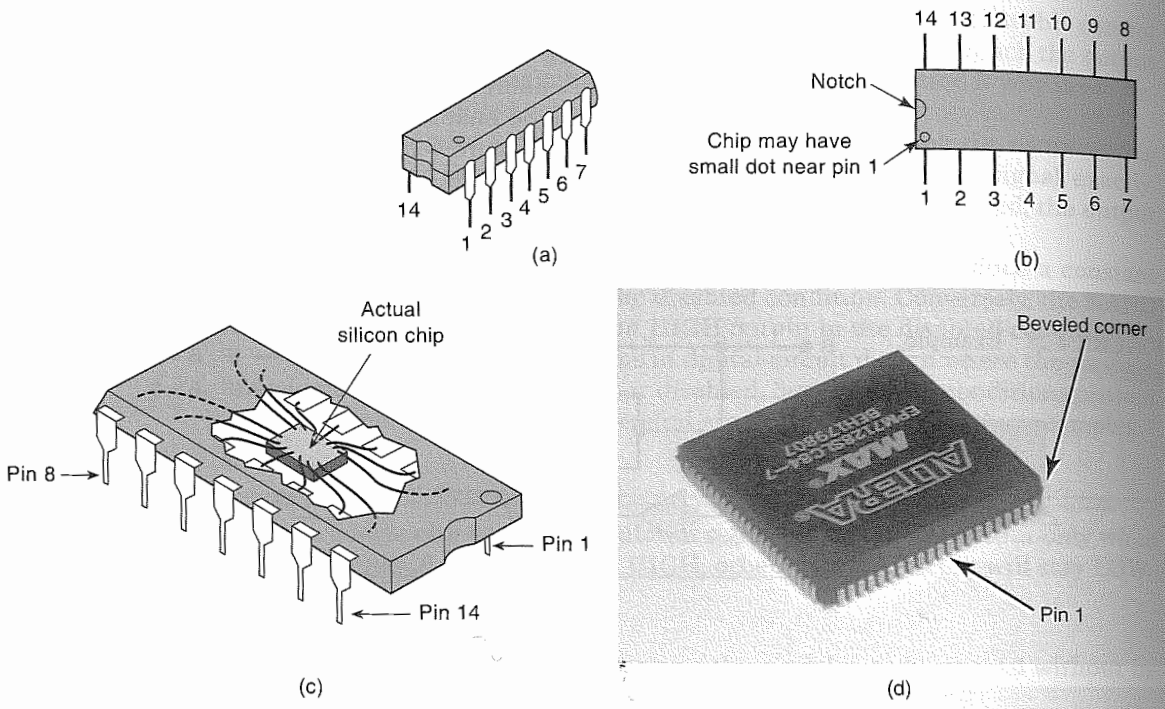


FIGURE 4-29 (a) Dual-in-line package (DIP); (b) top view; (c) actual silicon chip is much smaller than the protective package; (d) PLCC package.

Digital ICs are often categorized according to their circuit complexity as measured by the number of equivalent logic gates on the substrate. The levels of complexity that are commonly defined are shown in Table 4-5.

TABLE 4-5

Complexity	Gates per Chip
Small-scale integration (SSI)	Fewer than 12
Medium-scale integration (MSI)	12 to 99
Large-scale integration (LSI)	100 to 9999
Very large-scale integration (VLSI)	10,000 to 99,999
Ultra large-scale integration (ULSI)	100,000 to 999,999
Giga-scale integration (GSI)	1,000,000 or more

All of the specific ICs referred to in Chapter 3 and this chapter are **SSI** chips having a small number of gates. In digital systems, medium-scale integration (**MSI**) and large-scale integration devices (**LSI, VLSI, ULSI, GSI**) can perform most of the functions that once required several circuit boards full of SSI devices. However, SSI chips are still used as the "interface," or "glue," between these more complex chips. The small-scale ICs also offer an excellent way to learn the basic building blocks of digital systems. Consequently, many laboratory-based courses use these ICs to build and test small projects.

The industrial world of digital electronics has now turned to programmable logic devices (PLDs) to implement a digital system of any significant size. Some simple PLDs are available in DIP packages, but the more complex

programmable logic devices require many more pins than are available in DIPs. Larger integrated circuits that may need to be removed from a circuit and replaced are typically manufactured in a plastic leaded chip carrier (PLCC) package. Figure 4-29(d) shows the Altera EPM 7128SLC84 in a PLCC package, which is a very popular PLD used in many educational laboratories. The key features of this chip are more pins, closer spacing, and pins around the entire periphery. Notice that pin 1 is not "on the corner" like the DIP but rather at the middle of the top of the package.

Bipolar and Unipolar Digital ICs

Digital ICs can also be categorized according to the principal type of electronic component used in their circuitry. *Bipolar ICs* are made using the bipolar junction transistor (NPN and PNP) as their main circuit element. *Unipolar ICs* use the unipolar field-effect transistor (P-channel and N-channel MOSFETs) as their main element.

The **transistor-transistor logic (TTL)** family has been the major family of bipolar digital ICs for over 40 years. The standard 74 series was the first series of TTL ICs. It is no longer used in new designs, having been replaced by several higher-performance TTL series, but its basic circuit arrangement forms the foundation for all the TTL series ICs. This circuit arrangement is shown in Figure 4-30(a) for the standard TTL INVERTER. Notice that the circuit contains several bipolar transistors as the main circuit element.

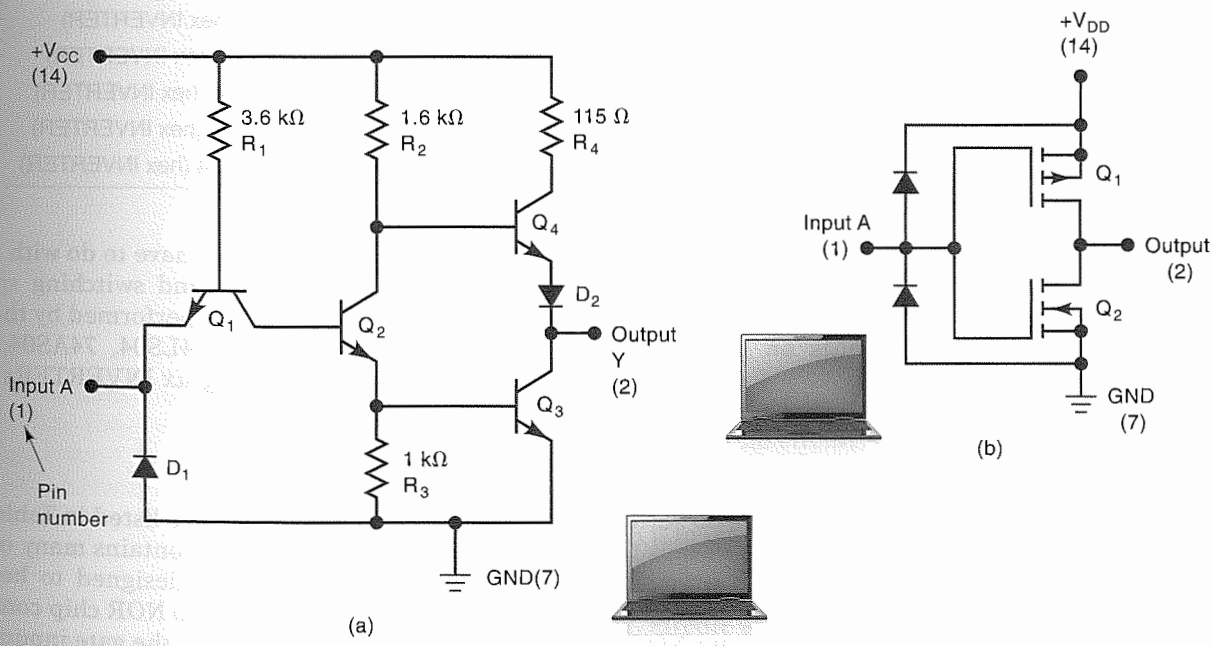


FIGURE 4-30 (a) TTL INVERTER circuit; (b) CMOS INVERTER circuit. Pin numbers are given in parentheses.

TTL was the leading IC family in the SSI and MSI categories up until approximately 1990. Since then, its leading position has been challenged by CMOS technology, which has gradually displaced TTL from that position. The **complementary metal-oxide semiconductor (CMOS)** family belongs to the class of unipolar digital ICs because it uses P- and N-channel MOSFETs as the main circuit elements. Figure 4-30(b) is a standard CMOS INVERTER circuit. If we compare the TTL and CMOS circuits in Figure 4-30, it is apparent

that the CMOS version uses fewer components. This is one of the main advantages of CMOS over TTL.

Because of the simplicity and compactness as well as some other superior attributes of CMOS, the modern large-scale ICs are manufactured primarily using CMOS technology. Teaching laboratories that use SSI and MSI devices often use TTL due to its durability, although some use CMOS as well. Chapter 8 will provide a comprehensive study of the circuitry and characteristics of TTL and CMOS. For now, we need to look at only a few of their basic characteristics so that we can talk about troubleshooting simple combinational circuits.

TTL Family

The TTL logic family actually consists of several subfamilies or series. Table 4-6 lists the name of each TTL series together with the prefix designation used to identify different ICs as being part of that series. For example, ICs that are part of the standard TTL series have an identification number that starts with 74. The 7402, 7438, and 74123 are all ICs in this series. Likewise, ICs that are part of the low-power Schottky TTL series have an identification number that starts with 74LS. The 74LS02, 74LS38, and 74LS123 are examples of devices in the 74LS series.

TABLE 4-6 Various series within the TTL logic family.

TTL Series	Prefix	Example IC
Standard TTL	74	7404 (hex INVERTER)
Schottky TTL	74S	74S04 (hex INVERTER)
Low-power Schottky TTL	74LS	74LS04 (hex INVERTER)
Advanced Schottky TTL	74AS	74AS04 (hex INVERTER)
Advanced low-power Schottky TTL	74ALS	74ALS04 (hex INVERTER)

The principal differences in the various TTL series have to do with their electrical characteristics such as power dissipation and switching speed. They do not differ in the pin layout or logic operations performed by the circuits on the chip. For example, the 7404, 74S04, 74LS04, 74AS04, and 74ALS04 are all hex-INVERTER ICs, each containing *six* INVERTERs on a single chip.

CMOS Family

Several CMOS series are available, and some of these are listed in Table 4-7. The 4000 series is the oldest CMOS series. This series contains many of the same logic functions as the TTL family but was not designed to be *pin-compatible* with TTL devices. For example, the 4001 quad NOR chip contains four two-input NOR gates, as does the TTL 7402 chip, but the gate inputs and outputs on the CMOS chip will not have the same pin numbers as the corresponding signals on the TTL chip.

The 74C, 74HC, 74HCT, 74AC, and 74ACT series are newer CMOS series. The first three are pin-compatible with correspondingly numbered TTL devices. For example, the 74C02, 74HC02, and 74HCT02 have the same pin layout as the 7402, 74LS02, and so on. The 74HC and 74HCT series operate at a higher speed than 74C devices. The 74HCT series is designed to be *electrically compatible* with TTL devices; that is, a 74HCT integrated circuit can be connected directly to TTL devices without any interfacing circuitry. The 74AC and 74ACT series are advanced-performance ICs. Neither is pin-compatible with

TABLE 4-7 Various series within the CMOS logic family.

CMOS Series	Prefix	Example IC
Metal-gate CMOS	40	4001 (quad NOR gates)
Metal-gate, pin-compatible with TTL	74C	74C02 (quad NOR gates)
Silicon-gate, pin-compatible with TTL, high-speed	74HC	74HC02 (quad NOR gates)
Silicon-gate, high-speed, pin-compatible and electrically compatible with TTL	74HCT	74HCT02 (quad NOR gates)
Advanced-performance CMOS, not pin-compatible or electrically compatible with TTL	74AC	74AC02 (quad NOR)
Advanced-performance CMOS, not pin-compatible with TTL, but electrically compatible with TTL	74ACT	74ACT02 (quad NOR)

TTL. The 74ACT devices are electrically compatible with TTL. We explore the various TTL and CMOS series in greater detail in Chapter 8 as well as the latest low-voltage technologies used in modern ICs.

Power and Ground

To use digital ICs, it is necessary to make the proper connections to the IC pins. The most important connections are *dc power* and *ground*. These are required for the circuits on the chip to operate correctly. In Figure 4-30, you can see that both the TTL and the CMOS circuits have a dc power supply voltage connected to one of their pins, and ground to another. The power supply pin is labeled V_{CC} for the TTL circuit, and V_{DD} for the CMOS circuit. Many of the newer CMOS integrated circuits that are designed to be compatible with TTL integrated circuits also use the V_{CC} designation as their power pin.

If either the power or the ground connection is not made to the IC, the logic gates on the chip will not respond properly to the logic inputs, and the gates will not produce the expected output logic levels.

Logic-Level Voltage Ranges

For TTL devices, V_{CC} is nominally +5 V. For CMOS integrated circuits, V_{DD} can range from +3 to +18 V, although +5 V is most often used when CMOS integrated circuits are used in the same circuit with TTL integrated circuits.

For standard TTL devices, the acceptable input voltage ranges for the logic 0 and logic 1 levels are defined as shown in Figure 4-31(a). A logic 0 is any voltage in the range from 0 to 0.8 V; a logic 1 is any voltage from 2 to 5 V. Voltages that are not in either of these ranges are said to be **indeterminate** and should not be used as inputs to any TTL device. The IC manufacturers cannot guarantee how a TTL circuit will respond to input levels that are in the indeterminate range (between 0.8 and 2.0 V).

The logic input voltage ranges for CMOS integrated circuits operating with V_{DD} = +5 V are shown in Figure 4-31(b). Voltages between 0 and 1.5 V are defined as a logic 0, and voltages from 3.5 to 5 V are defined as a logic 1. The indeterminate range includes voltages between 1.5 and 3.5 V.

Unconnected (Floating) Inputs

What happens when the input to a digital IC is left unconnected? An unconnected input is often called a **floating** input. The answer to this question will be different for TTL and CMOS.

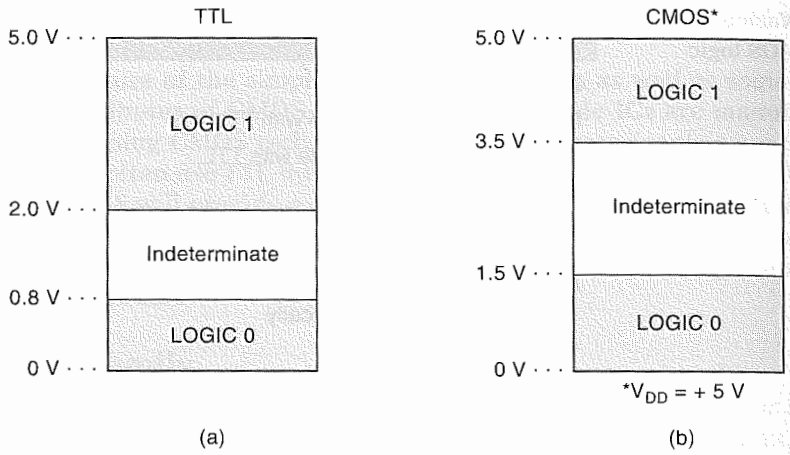


FIGURE 4-31 Logic-level input voltage ranges for (a) TTL and (b) CMOS digital ICs.

A floating TTL input acts just like a logic 1. In other words, the IC will respond as if the input had a logic HIGH level applied to it. This characteristic is often used when testing a TTL circuit. A lazy technician might leave certain inputs unconnected instead of connecting them to a logic HIGH. Although this is logically correct, it is not a recommended practice, especially in final circuit designs, because the floating TTL input is extremely susceptible to picking up noise signals that will probably adversely affect the device's operation.

A floating input on some TTL gates will measure a dc level of between 1.4 and 1.8 V when checked with a VOM or an oscilloscope. Even though this is in the indeterminate range for TTL, it will produce the same response as a logic 1. Being aware of this characteristic of a floating TTL input can be valuable when troubleshooting TTL circuits.

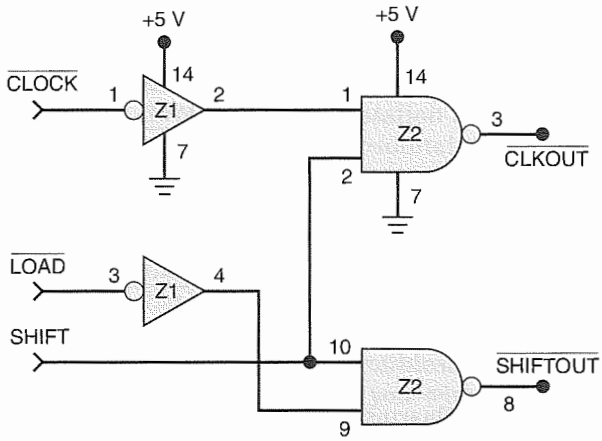
If a CMOS input is left floating, it may have disastrous results. The IC may become overheated and eventually destroy itself. For this reason all inputs to a CMOS integrated circuit must be connected to a LOW or a HIGH level or to the output of another IC. A floating CMOS input will not measure as a specific dc voltage but will fluctuate randomly as it picks up noise. Thus, it does not act as logic 1 or logic 0, and so its effect on the output is unpredictable. Sometimes the output will oscillate as a result of the noise picked up by the floating input.

Many of the more complex CMOS ICs have circuitry built into the inputs, which reduces the likelihood of any destructive reaction to an open input. With this circuitry, it is not necessary to ground each unused pin on these large ICs when experimenting. It is still good practice, however, to tie unused inputs to HIGH or LOW (whichever is appropriate) in the final circuit implementation.

Logic-Circuit Connection Diagrams

A connection diagram shows *all* electrical connections, pin numbers, IC numbers, component values, signal names, and power supply voltages. Figure 4-32 shows a typical connection diagram for a simple logic circuit. Examine it carefully and note the following important points:

1. The circuit uses logic gates from two different ICs. The two INVERTERs are part of a 74HC04 chip that has been given the designation Z1. The 74HC04 contains six INVERTERs; two of them are used in this circuit, and each is labeled as being part of chip Z1. Similarly, the two NAND gates are part of a 74HC00 chip that contains four NAND gates. All of the gates on this chip are designated with the label Z2. By numbering each



IC	Type
Z1	74HC04 hex inverter
Z2	74HC00 quad nand

FIGURE 4-32 Typical logic-circuit connection diagram.

gate as Z1, Z2, Z3, and so on, we can keep track of which gate is part of which chip. This is especially valuable in more complex circuits containing many ICs with several gates per chip.

2. Each gate input and output pin number is indicated on the diagram. These pin numbers and the IC labels are used to reference easily any point in the circuit. For example, Z1 pin 2 refers to the output pin of the top INVERTER. Similarly, we can say that Z1 pin 4 is connected to Z2 pin 9.
3. The power and ground connections to each IC (not each gate) are shown on the diagram. For example, Z1 pin 14 is connected to +5 V, and Z1 pin 7 is connected to ground. These connections provide power to *all* of the six INVERTERs that are part of Z1.
4. For the circuit contained in Figure 4-32, the signals that are inputs are on the left. The signals that are outputs are on the right. The bar over the signal name indicates that the signal is active when LOW. The bubbles are positioned on the diagram symbols also to indicate the active-LOW state. Each signal in this case is obviously a single bit.
5. Signals are defined graphically in Figure 4-32 as inputs and outputs, and the relationship between them (the operation of the circuit) is described graphically using interconnected logic symbols.

Manufacturers of electronic equipment generally supply detailed schematics that use a format similar to that in Figure 4-32. These connection diagrams are a virtual necessity when troubleshooting a faulty circuit. We have chosen to identify individual ICs as Z1, Z2, Z3, and so on. Other designations that are commonly used are IC1, IC2, IC3, and so on, and U1, U2, U3, and so on.

In Chapter 3 we introduced the graphic entry tools of Quartus II software from Altera. An example of a logic diagram drawn using Altera software is shown in Figure 4-33. A circuit like this is not intended to be implemented using SSI or MSI logic ICs. That is why there are no pin numbers or chip designations on the logic symbols, only instance numbers. Altera's software will translate a graphic description of the logic function into a binary file that is used to configure logic circuits inside one of Altera's many digital ICs. These configurable or programmable logic circuits will be described in detail later in this chapter. Also, notice that a common convention in naming input and output signals is to use the N suffix rather than the overbar to indicate that the signal is active-LOW. For example, the LOADN input is a signal that will be LOW in order to perform the LOAD function. Quartus II software is available free from Altera (www.altera.com), and a step-by-step tutorial (Quartus Tutorial 1 - Schematic.pdf) is available to owners of this book on www.pearsonhighered.com/electronics.

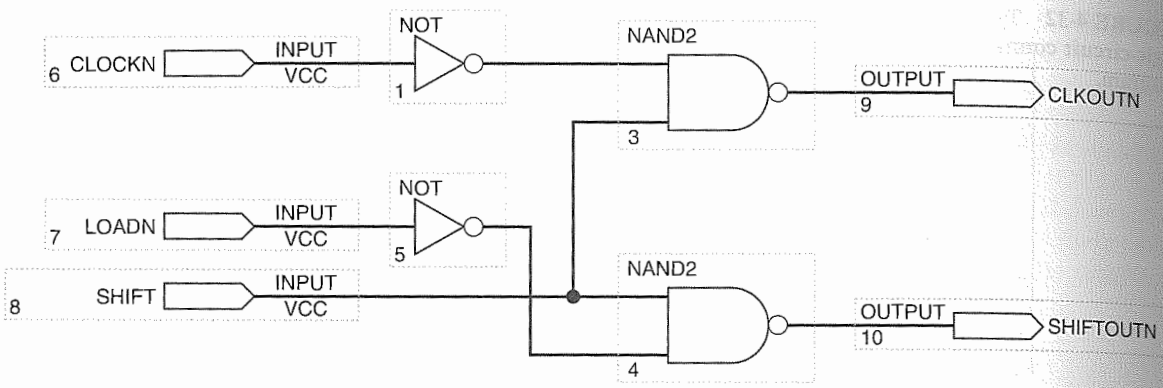


FIGURE 4-33 Logic diagram using Quartus II schematic capture.

REVIEW QUESTIONS

1. Which type of transistor is used in (a) TTL and (b) CMOS?
2. Name the six common categories of digital ICs according to complexity.
3. *True or false:* A 74S74 chip will contain the same logic and pin layout as the 74LS74.
4. *True or false:* A 74HC74 chip will contain the same logic and pin layout as the 74AS74.
5. Which CMOS series are not pin-compatible with TTL?
6. What is the acceptable input voltage range of a logic 0 for TTL? What is it for a logic 1?
7. Repeat question 6 for CMOS operating at $V_{DD} = 5$ V.
8. How does a TTL integrated circuit respond to a floating input?
9. How does a CMOS integrated circuit respond to a floating input?
10. Which CMOS series can be connected directly to TTL with no interfacing circuitry?
11. What is the purpose of pin numbers on a logic circuit connection diagram?
12. What are the key similarities of graphic design files used for programmable logic and traditional logic circuit connection diagrams?

4-10 TROUBLESHOOTING DIGITAL SYSTEMS

There are three basic steps in fixing a digital circuit or system that has a fault (failure):

1. *Fault detection.* Observe the circuit/system operation and compare it with the expected correct operation.
2. *Fault isolation.* Perform tests and make measurements to isolate the fault.
3. *Fault correction.* Replace the faulty component, repair the faulty connection, remove the short, and so on.

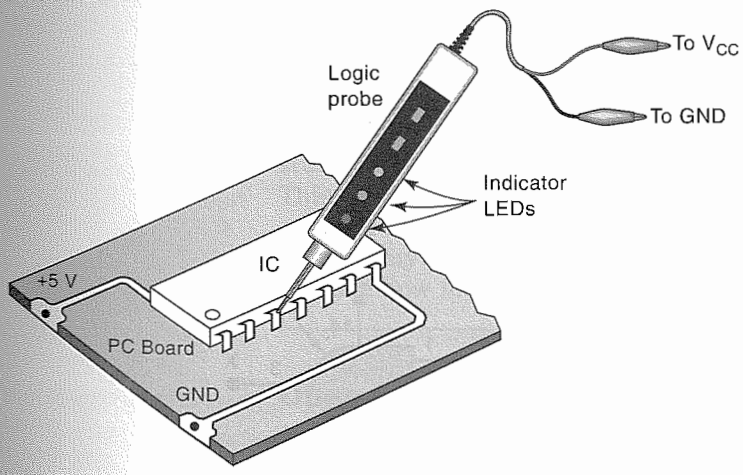
Although these steps may seem relatively apparent and straightforward, the actual troubleshooting procedure that is followed is highly dependent on the

type and complexity of the circuitry, and on the kinds of troubleshooting tools and documentation that are available.

Good troubleshooting techniques can be learned only in a laboratory environment through experimentation and actual troubleshooting of faulty circuits and systems. There is absolutely no better way to become an effective troubleshooter than to do as much troubleshooting as possible, and no amount of textbook reading can provide that kind of experience. We can, however, help you to develop the analytical skills that are the most essential part of effective troubleshooting. We will describe the types of faults that are common to systems that are made primarily from digital ICs and we will tell you how to recognize them. We will then present typical case studies to illustrate the analytical processes involved in troubleshooting. In addition, there will be end-of-chapter problems to provide you with the opportunity to go through these analytical processes to reach conclusions about faulty digital circuits.

For the troubleshooting discussions and exercises we will be doing in this book, it will be assumed that the troubleshooting technician has the basic troubleshooting tools available: *logic probe, oscilloscope, logic pulser.* Of course, the most important and effective troubleshooting tool is the technician's brain, and that's the tool we are hoping to develop by presenting troubleshooting principles and techniques, examples and problems, here and in the following chapters.

In the next three sections on troubleshooting, we will use only our brain and a **logic probe** such as the one illustrated in Figure 4-34. The logic probe has a pointy metal tip that is touched to the specific point you want to test. Here, it is shown probing (touching) pin 3 of an IC. It can also be touched to a printed circuit board trace, an uninsulated wire, a connector pin, a lead on a discrete component such as a transistor, or any other conducting point in a circuit. The logic level that is present at the probe tip will be indicated by the status of the indicator LEDs in the probe. The four possibilities are given in the table of Figure 4-34. Note that an *indeterminate* logic level produces no indicator light. This includes the condition where the probe tip is touched to a point in a circuit that is open or floating—that is, not connected to any source of voltage. This type of probe also offers a yellow LED to indicate the presence of a pulse train. Any transitions (LOW to HIGH or HIGH to LOW) will cause the yellow LED to flash on for a fraction of a second and then turn off. If the transitions are occurring frequently, the LED will continue to flash



LEDs			
Red	Green	Yellow	Logic Condition
OFF	ON	OFF	LOW
ON	OFF	OFF	HIGH
OFF	OFF	OFF	INDETERMINATE*
X	X	FLASHING	PULSING

* Includes open or floating condition

FIGURE 4-34 A logic probe is used to monitor the logic level activity at an IC pin or any other accessible point in a logic circuit.

at around 3 Hz. By observing the green and red LEDs along with the flashing yellow, you can tell whether the signal is mostly HIGH or mostly LOW.

4-11 INTERNAL DIGITAL IC FAULTS

The most common internal failures of digital ICs are:

1. Malfunction in the internal circuitry
2. Inputs or outputs shorted to ground or V_{CC}
3. Inputs or outputs open-circuited
4. Short between two pins (other than ground or V_{CC})

We will now describe each of these types of failure.

Malfunction in Internal Circuitry

This is usually caused by one of the internal components failing completely or operating outside its specifications. When this happens, the IC outputs do not respond properly to the IC inputs. There is no way to predict what the outputs will do because it depends on what internal component has failed. Examples of this type of failure would be a base-emitter short in transistor Q_4 or an extremely large resistance value for R_2 in the TTL INVERTER of Figure 4-30(a). This type of internal IC failure is not as common as the other three.

Input Internally Shorted to Ground or Supply

This type of internal failure will cause an IC input to be stuck in the LOW or HIGH state. Figure 4-35(a) shows input pin 2 of a NAND gate shorted to ground within the IC. This will cause pin 2 always to be in the LOW state. If this input pin is being driven by a logic signal B, it will effectively short B to ground. Thus, this type of fault will affect the output of the device that is generating the B signal.

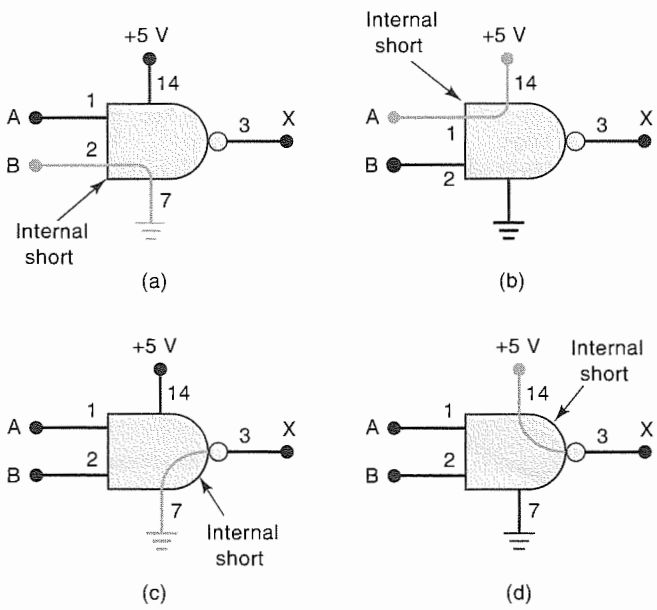


FIGURE 4-35 (a) IC input internally shorted to ground; (b) IC input internally shorted to supply voltage. These two types of failures force the input signal at the shorted pin to stay in the same state. (c) IC output internally shorted to ground; (d) output internally shorted to supply voltage. These two failures do not affect signals at the IC inputs.

Similarly, an IC input pin could be internally shorted to +5 V, as in Figure 4-35(b). This would keep that pin stuck in the HIGH state. If this input pin is being driven by a logic signal *A*, it will effectively short *A* to +5 V.

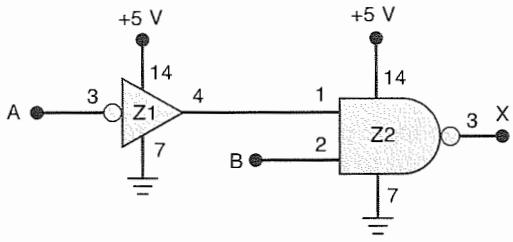
Output Internally Shorted to Ground or Supply

This type of internal failure will cause the output pin to be stuck in the LOW or HIGH state. Figure 4-35(c) shows pin 3 of the NAND gate shorted to ground within the IC. This output is stuck LOW, and it will not respond to the conditions applied to input pins 1 and 2; in other words, logic inputs *A* and *B* will have no effect on output *X*.

An IC output pin can also be shorted to +5 V within the IC, as shown in Figure 4-35(d). This forces the output pin 3 to be stuck HIGH regardless of the state of the signals at the input pins. Note that this type of failure has no effect on the logic signals at the IC inputs.

EXAMPLE 4-24

Refer to the circuit of Figure 4-36. A technician uses a logic probe to determine the conditions at the various IC pins. The results are recorded in the figure. Examine these results and determine if the circuit is working properly. If not, suggest some of the possible faults.



Pin	Condition
Z1-3	Pulsing
Z1-4	LOW
Z2-1	LOW
Z2-2	HIGH
Z2-3	HIGH

FIGURE 4-36
Example 4-24.

Solution

Output pin 4 of the INVERTER should be pulsing because its input is pulsing. The recorded results, however, show that pin 4 is stuck LOW. Because this is connected to Z2 pin 1, this keeps the NAND output HIGH. From our preceding discussion, we can list three possible faults that could produce this operation.

First, there could be an internal component failure in the INVERTER that prevents it from responding properly to its input. Second, pin 4 of the INVERTER could be shorted to ground internal to Z1, thereby keeping it stuck LOW. Third, pin 1 of Z2 could be shorted to ground internal to Z2. This would prevent the INVERTER output pin from changing.

In addition to these possible faults, there can be external shorts to ground anywhere in the conducting path between Z1 pin 4 and Z2 pin 1. We will see how to go about isolating the actual fault in Section 4-13.

Open-Circuited Input or Output

Sometimes the very fine conducting wire that connects an IC pin to the IC's internal circuitry will break, producing an open circuit. Figure 4-37 in Example 4-25 shows this for an input (pin 13) and an output (pin 6). If a signal is applied to pin 13, it will not reach the NAND-1 gate input and so will not have an effect

on the NAND-1 output. The open gate input will be in the floating state. As stated earlier, TTL devices will respond as if this floating input is a logic 1, and CMOS devices will respond erratically and may even become damaged from overheating.

The open at the NAND-4 output prevents the signal from reaching IC pin 6, so there will be no stable voltage present at that pin. If this pin is connected to the input of another IC, it will produce a floating condition at that input.

EXAMPLE 4-25

What would a logic probe indicate at pin 13 and at pin 6 of Figure 4-37?

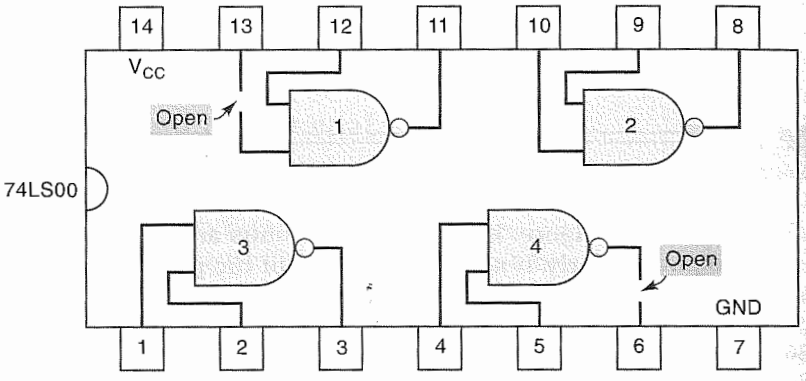


FIGURE 4-37 An IC with an internally open input will not respond to signals applied to that input pin. An internally open output will produce an unpredictable voltage at that output pin.

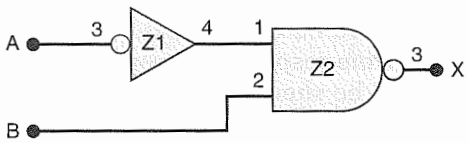
Solution

At pin 13, the logic probe will indicate the logic level of the external signal that is connected to pin 13 (which is not shown in the diagram). At pin 6, the logic probe will show no LED lit for an indeterminate logic level because the NAND output level never makes it to pin 6.

EXAMPLE 4-26

Refer to the circuit of Figure 4-38 and the recorded logic probe indications. What are some of the possible faults that could produce the recorded results? Assume that the ICs are TTL.

FIGURE 4-38 Example 4-26.



Pin	Condition
Z1-3	HIGH
Z1-4	LOW
Z2-1	LOW
Z2-2	Pulsing
Z2-3	Pulsing

Note: V_{CC} and ground connections to each IC are not shown

Solution

Examination of the recorded results indicates that the INVERTER appears to be working properly, but the NAND output is inconsistent with its inputs. The NAND output should be HIGH because its input pin 1 is LOW. This LOW should prevent the NAND gate from responding to the pulses at pin 2. It is probable that this LOW is not reaching the internal NAND gate circuitry

because of an internal open. Because the IC is TTL, this open circuit would produce the same effect as a logic HIGH at pin 1. If the IC had been CMOS, the internal open circuit at pin 1 might have produced an indeterminate output and possible overheating and destruction of the chip.

From our earlier statement regarding open TTL inputs, you might have expected that the voltage of pin 1 of Z2 would be 1.4 to 1.8 V and should have been registered as indeterminate by the logic probe. This would have been true if the open circuit had been *external* to the NAND chip. There is no open circuit between Z1 pin 4 and Z2 pin 1, and so the voltage at Z1 pin 4 is reaching Z2 pin 1, but it becomes disconnected *inside* the NAND chip.

Short Between Two Pins

An internal short between two pins of an IC will force the logic signals at those pins always to be identical. Whenever two signals that are supposed to be different show the same logic-level variations, there is a good possibility that the signals are shorted together.

Consider the circuit in Figure 4-39, where pins 5 and 6 of the NOR gate are internally shorted together. The short causes the two INVERTER output pins to be connected together so that the signals at Z1 pin 2 and Z1 pin 4 must be identical, even when the two INVERTER input signals are trying to produce different outputs. To illustrate, consider the input waveforms shown in the diagram. Even though these input waveforms are different, the waveforms at outputs Z1-2 and Z1-4 are the same.

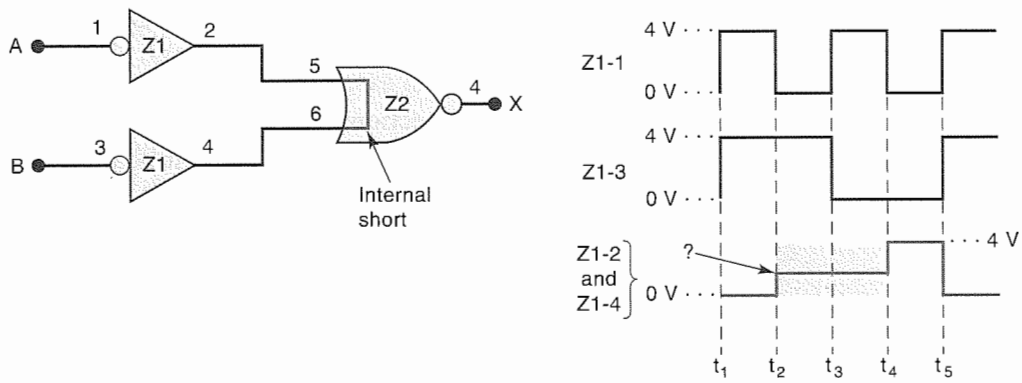


FIGURE 4-39 When two input pins are internally shorted, the signals driving these pins are forced to be identical, and usually a signal with three distinct levels results.

During the interval t_1 to t_2, both INVERTERs have a HIGH input and both are trying to produce a LOW output, so that their being shorted together makes no difference. During the interval t_4 to t_5, both INVERTERs have a LOW input and are trying to produce a HIGH output, so that again their being shorted has no effect. However, during the intervals t_2 to t_3 and t_3 to t_4, one INVERTER is trying to produce a HIGH output while the other is trying to produce a LOW output. This is called signal **contention** because the two signals are "fighting" each other. When this happens, the actual voltage level that appears at the shorted outputs will depend on the internal IC circuitry. For TTL devices, it will usually be a voltage in the high end of the logic 0 range (i.e., close to 0.8 V), although it may also be in the indeterminate range. For CMOS devices, it will often be a voltage in the indeterminate range.

Whenever you see a waveform like the Z1-2, Z1-4 signal in Figure 4-39 with three different levels, you should suspect that two output signals may be shorted together.

REVIEW QUESTIONS

1. List the different internal digital IC faults.
2. Which internal IC fault can produce signals that show three different voltage levels?
3. What would a logic probe indicate at Z1-2 and Z1-4 of Figure 4-39 if $A = 0$ and $B = 1$?
4. What is signal contention?

4-12 EXTERNAL FAULTS

We have seen how to recognize the effects of various faults internal to digital ICs. Many more things can go wrong external to the ICs; we will describe the most common ones in this section.

Open Signal Lines

This category includes any fault that produces a break or discontinuity in the conducting path such that a voltage level or signal is prevented from going from one point to another. Some of the causes of open signal lines are:

1. Broken wire
2. Poor solder connection; loose wire-wrap connection
3. Crack or cut trace on a printed circuit board (some of these are hairline cracks that are difficult to see without a magnifying glass)
4. Bent or broken pin on an IC
5. Faulty IC socket such that the IC pin does not make good contact with the socket

This type of circuit fault can often be detected by a careful visual inspection and then verified by disconnecting power from the circuit and checking for continuity (i.e., a low-resistance path) with an ohmmeter between the two points in question.

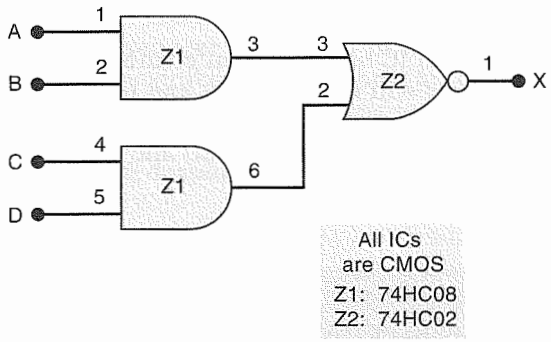
EXAMPLE 4-27

Consider the CMOS circuit of Figure 4-40 and the accompanying logic probe indications. What is the most probable circuit fault?

Solution

The indeterminate level at the NOR gate output is probably due to the indeterminate input at pin 2. Because there is a LOW at Z1-6, this LOW should also be at Z2-2. Clearly, the LOW from Z1-6 is not reaching Z2-2, and there must be an open circuit in the signal path between these two points. The location of this open circuit can be determined by starting at Z1-6 with the logic probe and tracing the LOW level along the signal path toward Z2-2 until it changes into an indeterminate level.

FIGURE 4-40 Example 4-27.



Pin	Condition
Z1-1	Pulsing
Z1-2	HIGH
Z1-3	Pulsing
Z1-4	LOW
Z1-5	Pulsing
Z1-6	LOW
Z2-3	Pulsing
Z2-2	Indeterminate
Z2-1	Indeterminate

Shorted Signal Lines

This type of fault has the same effect as an internal short between IC pins. It causes two signals to be exactly the same (signal contention). A signal line may be shorted to ground or V_{CC} rather than to another signal line. In those cases, the signal will be forced to the LOW or the HIGH state. The main causes for unexpected shorts between two points in a circuit are as follows:

1. *Sloppy wiring.* An example of this is stripping too much insulation from ends of wires that are in close proximity.
2. *Solder bridges.* These are splashes of solder that short two or more points together. They commonly occur between points that are very close together, such as adjacent pins on a chip.
3. *Incomplete etching.* The copper between adjacent conducting paths on a printed circuit board is not completely etched away.

Again, a careful visual inspection can very often uncover this type of fault, and an ohmmeter check can verify that the two points in the circuit are shorted together.

Faulty Power Supply

All digital systems have one or more dc power supplies that supply the V_{CC} and V_{DD} voltages required by the chips. A faulty power supply or one that is overloaded (supplying more than its rated amount of current) will provide poorly regulated supply voltages to the ICs, and the ICs either will not operate or will operate erratically.

A power supply may go out of regulation because of a fault in its internal circuitry, or because the circuits that it is powering are drawing more current than the supply is designed for. This can happen if a chip or a component has a fault that causes it to draw much more current than normal.

It is good troubleshooting practice to check the voltage levels at each power supply in the system to see that they are within their specified ranges. It is also a good idea to check them on an oscilloscope to verify that there is no significant amount of ac ripple on the dc levels and to verify that the voltage levels stay regulated during the system operation.

One of the most common signs of a faulty power supply is one or more chips operating erratically or not at all. Some ICs are more tolerant of power supply variations and may operate properly, while others do not. You should always check the power and ground levels at each IC that appears to be operating incorrectly.

Output Loading

When a digital IC has its output connected to too many IC inputs, its output current rating will be exceeded, and the output voltage can fall into the indeterminate range. This effect is called *loading* the output signal (actually it's overloading the output signal) and is usually the result of poor design or an incorrect connection.

REVIEW QUESTIONS

1. What are the most common types of external faults?
2. List some of the causes of signal-path open circuits.
3. What symptoms are caused by a faulty power supply?
4. How might loading affect an IC output voltage level?

4-13 TROUBLESHOOTING CASE STUDY

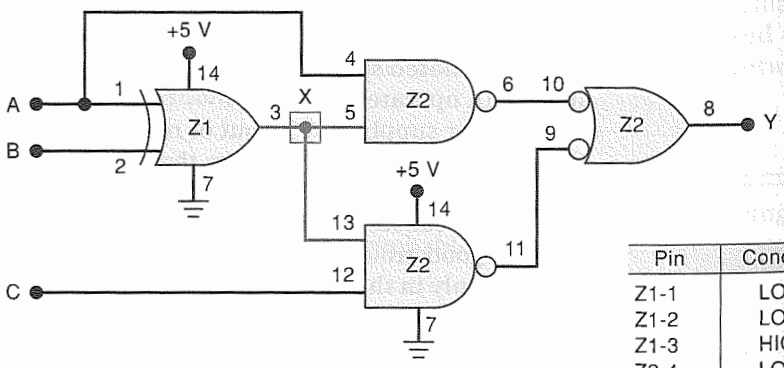
The following example will illustrate the analytical processes involved in troubleshooting digital circuits. Although the example is a fairly simple combinational logic circuit, the reasoning and the troubleshooting procedures used can be applied to the more complex digital circuits that we encounter in subsequent chapters.

EXAMPLE 4-28

Consider the circuit of Figure 4-41. Output Y is supposed to go HIGH for either of the following conditions:

1. $A = 1, B = 0$ regardless of the level on C
2. $A = 0, B = 1, C = 1$

You may wish to verify these results for yourself.



ICs are TTL
Z1: 74LS86
Z2: 74LS00

Pin	Condition
Z1-1	LOW
Z1-2	LOW
Z1-3	HIGH
Z2-4	LOW
Z2-5	HIGH
Z2-6,10	HIGH
Z2-13	HIGH
Z2-12	HIGH
Z2-9,11	LOW
Z2-8	HIGH

FIGURE 4-41 Example 4-28.

When the circuit is tested, the technician observes that output Y goes HIGH whenever A is HIGH or C is HIGH, regardless of the level at B. She takes logic probe measurements for the condition where $A = B = 0, C = 1$ and comes up with the indications recorded in Figure 4-41.

Examine the recorded levels and list the possible causes for the malfunction. Then develop a step-by-step procedure to determine the exact fault.

Solution

All of the NAND gate outputs are correct for the levels present at their inputs. The XOR gate, however, should be producing a LOW at output pin 3 because both of its inputs are at the same LOW level. It appears that Z1-3 is stuck HIGH, even though its inputs should produce a LOW. There are several possible causes for this:

1. An internal component failure in Z1 that prevents its output from going LOW
2. An external short to V_{CC} from any point along the conductors connected to node X (shaded in the diagram of the figure)
3. Pin 3 of Z1 internally shorted to V_{CC}
4. Pin 5 of Z2 internally shorted to V_{CC}
5. Pin 13 of Z2 internally shorted to V_{CC}

All of these possibilities except for the first one will short node X (and every IC pin connected to it) directly to V_{CC}.

The following procedure can be used to isolate the fault. This procedure is not the only approach that can be used and, as we stated earlier, the actual troubleshooting procedure that a technician uses is very dependent on what test equipment is available.

1. Check the V_{CC} and ground levels at the appropriate pins of Z1. Although it is unlikely that the absence of either of these might cause Z1-3 to stay HIGH, it is a good idea to make this check on any IC that is producing an incorrect output.
2. Turn off power to the circuit and use an ohmmeter to check for a short (resistance less than 1 Ω) between node X and any point connected to V_{CC} (such as Z1-14 or Z2-14). If no short is indicated, the last four possibilities in our list can be eliminated. This means that it is very likely that Z1 has an internal failure and should be replaced.
3. If step 2 shows that there is a short from node X to V_{CC}, perform a thorough visual examination of the circuit board and look for solder bridges, unetched copper slivers, uninsulated wires touching each other, and any other possible cause of an external short to V_{CC}. A likely spot for a solder bridge would be between adjacent pins 13 and 14 of Z2. Pin 14 is connected to V_{CC}, and pin 13 to node X. If an external short is found, remove it and perform an ohmmeter check to verify that node X is no longer shorted to V_{CC}.
4. If step 3 does not reveal an external short, the three possibilities that remain are internal shorts to V_{CC} at Z1-3, Z2-13, or Z2-5. One of these is shorting node X to V_{CC}.

To determine which of these IC pins is the culprit, we should disconnect each of them from node X *one at a time* and recheck for a short to V_{CC} after each disconnection. When the pin that is internally shorted to V_{CC} is disconnected, node X will no longer be shorted to V_{CC}.

Output Loading

When a digital IC has its output connected to too many IC inputs, its output current rating will be exceeded, and the output voltage can fall into the indeterminate range. This effect is called *loading* the output signal (actually it's overloading the output signal) and is usually the result of poor design or an incorrect connection.

REVIEW QUESTIONS

1. What are the most common types of external faults?
2. List some of the causes of signal-path open circuits.
3. What symptoms are caused by a faulty power supply?
4. How might loading affect an IC output voltage level?

4-13 TROUBLESHOOTING CASE STUDY

The following example will illustrate the analytical processes involved in troubleshooting digital circuits. Although the example is a fairly simple combinational logic circuit, the reasoning and the troubleshooting procedures used can be applied to the more complex digital circuits that we encounter in subsequent chapters.

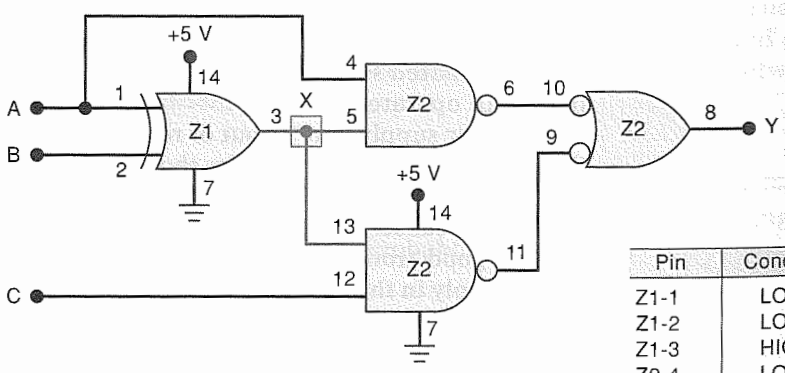
EXAMPLE 4-28

Consider the circuit of Figure 4-41. Output Y is supposed to go HIGH for either of the following conditions:

1. $A = 1, B = 0$ regardless of the level on C
2. $A = 0, B = 1, C = 1$

You may wish to verify these results for yourself.

FIGURE 4-41 Example 4-28.



ICs are TTL
Z1: 74LS86
Z2: 74LS00

Pin	Condition
Z1-1	LOW
Z1-2	LOW
Z1-3	HIGH
Z2-4	LOW
Z2-5	HIGH
Z2-6,10	HIGH
Z2-13	HIGH
Z2-12	HIGH
Z2-9,11	LOW
Z2-8	HIGH

When the circuit is tested, the technician observes that output Y goes HIGH whenever A is HIGH or C is HIGH, regardless of the level at B. She takes logic probe measurements for the condition where $A = B = 0, C = 1$ and comes up with the indications recorded in Figure 4-41.

Examine the recorded levels and list the possible causes for the malfunction. Then develop a step-by-step procedure to determine the exact fault.

Solution

All of the NAND gate outputs are correct for the levels present at their inputs. The XOR gate, however, should be producing a LOW at output pin 3 because both of its inputs are at the same LOW level. It appears that Z1-3 is stuck HIGH, even though its inputs should produce a LOW. There are several possible causes for this:

1. An internal component failure in Z1 that prevents its output from going LOW
2. An external short to V_{CC} from any point along the conductors connected to node X (shaded in the diagram of the figure)
3. Pin 3 of Z1 internally shorted to V_{CC}
4. Pin 5 of Z2 internally shorted to V_{CC}
5. Pin 13 of Z2 internally shorted to V_{CC}

All of these possibilities except for the first one will short node X (and every IC pin connected to it) directly to V_{CC}.

The following procedure can be used to isolate the fault. This procedure is not the only approach that can be used and, as we stated earlier, the actual troubleshooting procedure that a technician uses is very dependent on what test equipment is available.

1. Check the V_{CC} and ground levels at the appropriate pins of Z1. Although it is unlikely that the absence of either of these might cause Z1-3 to stay HIGH, it is a good idea to make this check on any IC that is producing an incorrect output.
2. Turn off power to the circuit and use an ohmmeter to check for a short (resistance less than 1 Ω) between node X and any point connected to V_{CC} (such as Z1-14 or Z2-14). If no short is indicated, the last four possibilities in our list can be eliminated. This means that it is very likely that Z1 has an internal failure and should be replaced.
3. If step 2 shows that there is a short from node X to V_{CC}, perform a thorough visual examination of the circuit board and look for solder bridges, unetched copper slivers, uninsulated wires touching each other, and any other possible cause of an external short to V_{CC}. A likely spot for a solder bridge would be between adjacent pins 13 and 14 of Z2. Pin 14 is connected to V_{CC}, and pin 13 to node X. If an external short is found, remove it and perform an ohmmeter check to verify that node X is no longer shorted to V_{CC}.
4. If step 3 does not reveal an external short, the three possibilities that remain are internal shorts to V_{CC} at Z1-3, Z2-13, or Z2-5. One of these is shorting node X to V_{CC}.

To determine which of these IC pins is the culprit, we should disconnect each of them from node X *one at a time* and recheck for a short to V_{CC} after each disconnection. When the pin that is internally shorted to V_{CC} is disconnected, node X will no longer be shorted to V_{CC}.

The process of disconnecting each suspected pin from node X can be easy or difficult depending on how the circuit is constructed. If the ICs are in sockets, all you need to do is to pull the IC from its socket, bend out the suspected pin, and reinsert the IC into its socket. If the ICs are soldered into a printed circuit board, you will have to cut the trace that is connected to the pin and repair the cut trace when you are finished.

Example 4-28, although fairly simple, shows you the kinds of thinking that a troubleshooter must employ to isolate a fault. You will have the opportunity to begin developing your own troubleshooting skills by working on many end-of-chapter problems that have been designated with a **T** for troubleshooting.

4-14 PROGRAMMABLE LOGIC DEVICES*

In the previous sections, we briefly considered the class of ICs known as programmable logic devices. In Chapter 3, we introduced the concept of describing a circuit's operation using a hardware description language. In this section, we will explore these topics further and become prepared to use the tools of the trade to develop and implement digital systems using PLDs. Of course, it is impossible to understand all the complex details of how a PLD works before grasping the fundamentals of digital circuits. As we examine new fundamental concepts, we will expand our knowledge of the PLDs and the programming methods. The material is presented in such a way that anyone who is not interested in PLDs can easily skip over these sections without loss of continuity in the coverage of the basic principles.

Let's review the process we covered earlier of designing combinational digital circuits. The input devices are identified and assigned an algebraic name like *A*, *B*, *C*, or *LOAD, SHIFT, CLOCK*. Likewise, output devices are given names like *X*, *Z*, or *CLOCK_OUT, SHIFT_OUT*. Then a truth table is constructed that lists all the possible input combinations and identifies the required state of the outputs under each input condition. The truth table is one way of describing how the circuit is to operate. Another way to describe the circuit's operation is the Boolean expression. From this point the designer must find the simplest algebraic relationship and select digital ICs that can be wired together to implement the circuit. You have probably experienced that these last steps are the most tedious, time consuming, and prone to errors.

Programmable logic devices allow most of these tedious steps to be automated by a computer and PLD *development software*. Using programmable logic improves the efficiency of the design and development process. Consequently, most modern digital systems are implemented in this way. The job of the circuit designer is to identify inputs and outputs, specify the logical relationship in the most convenient manner, and select a programmable device that is capable of implementing the circuit at the lowest cost. The concept behind programmable logic devices is simple: put lots of logic gates in a single IC and control the interconnection of these gates electronically.

PLD Hardware

Recall from Chapter 3 that many digital circuits today are implemented using programmable logic devices (PLDs). These devices are configured electronically and their internal circuits are "wired" together electronically to

*All sections covering PLDs may be skipped without loss of continuity.

form a logic circuit. This programmable wiring can be thought of as thousands of connections that are either connected (1) or not connected (0). It is very tedious to try to configure these devices by manually placing 1s and 0s in a grid, so the next logical question is, "How do we control the interconnection of gates in a PLD electronically?"

A common method of connecting one of many signals entering a network to one of many signal lines exiting the network is a switching matrix. Refer back to Figure 3-44, where this concept was introduced. A matrix is simply a grid of conductors (wires) arranged in rows and columns. Input signals are connected to the columns of the matrix, and the outputs are connected to the rows of the matrix. At each intersection of a row and a column is a switch that can electrically connect that row to that column. The switches that connect rows to columns can be mechanical switches, fusible links, electromagnetic switches (relays), or transistors. This is the general structure used in many applications and will be explored further when we study memory devices in Chapter 12.

PLDs also use a switch matrix that is often referred to as a programmable array. By deciding which intersections are connected and which ones are not, we can "program" the way the inputs are connected to the outputs of the array. In Figure 4-42, a programmable array is used to select the inputs for each AND gate. Notice that in this simple matrix, we can produce any logical product combination of variables A, B at any of the AND gate outputs. A matrix or programmable array such as the one shown in the figure can also be used to connect the AND outputs to OR gates. The details of various PLD architectures will be covered thoroughly in Chapter 13.

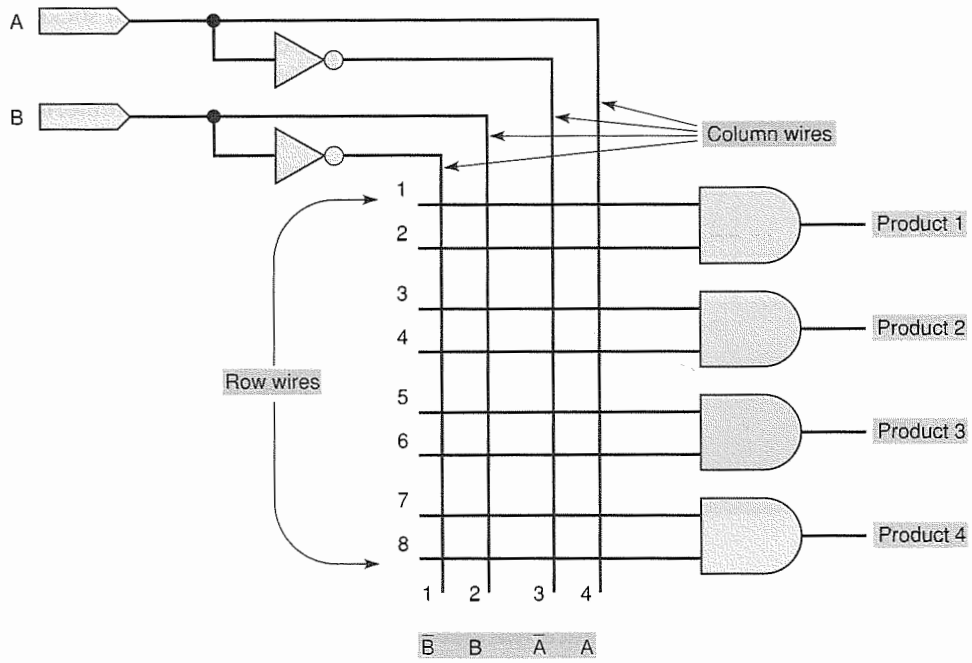


FIGURE 4-42 A programmable array for selecting inputs as product terms.

Programming a PLD

There are two ways to "program" a PLD IC. Programming means making the actual connections in the array. In other words, it means determining which

of those connections are supposed to be open (0) and which are supposed to be closed (1). One method involves removing the PLD IC chip from its circuit board. The chip is then placed in a special fixture called a **programmer**, shown in Figure 4-43. Most modern programmers are connected to a personal computer that is running software containing libraries of information about the many types of programmable devices available.

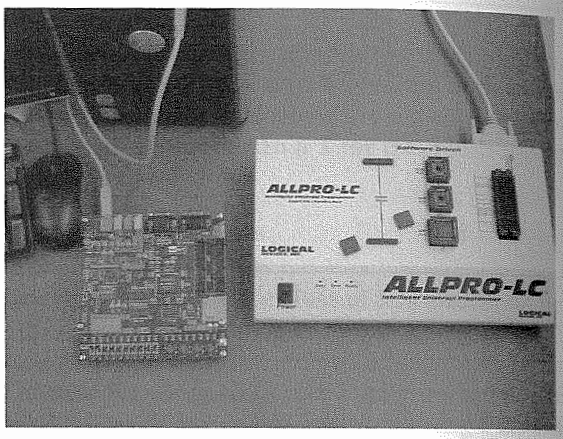


FIGURE 4-43 A universal IC programmer (right) and an in-system programmed (ISP) logic IC on a development board (left) interfaced through a USB port.

The programming software is invoked (called up and executed) on the PC to establish communication with the programmer. This software allows the user to set up the programmer for the type of device that is to be programmed, check if the device is blank, read the state of any programmable connection in the device, and provide instructions for the user to program a chip. Ultimately, the part is placed into a special socket that allows you to drop the chip in and then clamp the contacts onto the pins. This is called a **zero insertion force (ZIF)** socket. *Universal programmers* that can program any type of programmable device are available from numerous manufacturers.

Fortunately, as programmable parts began to proliferate, manufacturers saw the need to standardize pin assignments and programming methods. As a result, the Joint Electronic Device Engineering Council **(JEDEC)** was formed. One of the results was JEDEC standard 3, a format for transferring programming data for PLDs, independent of the PLD manufacturer or programming software. Pin assignments for various IC packages were also standardized, making universal programmers less complicated. Consequently, programming fixtures can program numerous types of PLDs. The software that allows the designer to specify a configuration for a PLD simply needs to produce an output file that conforms to the JEDEC standards. Then this JEDEC file can be loaded into any JEDEC-compatible PLD programmer that is capable of programming the desired type of PLD.

The more common method used today is referred to as in-system programming **(ISP)**. As its name implies, the chip does not need to be removed from its circuit for storage of the programming information. A standard interface has been developed by the Joint Test Action Group **(JTAG)**. The interface was developed to allow ICs to be tested without actually connecting test equipment to every pin of the IC. It also allows for internal programming. Four pins on the IC are used as a portal to store data and retrieve information about the inner condition of the IC. Many ICs, including PLDs and microcontrollers, are manufactured today to include the JTAG interface. An interface cable connects the four JTAG pins on the IC to an output port (typically USB) of a personal computer. Software running on the PC establishes contact with the IC and loads the information in the proper format.

Development Software

We have examined several methods of describing logic circuits now, including schematic capture, logic equations, truth tables, and HDL. We also described the fundamental methods of storing 1s and 0s into a PLD IC to connect the logic circuits in the desired way. The biggest challenge in getting a PLD programmed is converting from any form of description into the array of 1s and 0s. Fortunately, this task is accomplished quite easily by a computer running the development software. The development software that we will be referring to and using for examples is produced by Altera. This software allows the designer to enter a circuit description in any one of the many ways we have been discussing: graphic design files (schematics), AHDL, and VHDL. It also allows the use of another HDL, called Verilog, and the option of describing the circuit with timing diagrams. Circuit blocks described by any of these methods can also be "connected" together to implement a much larger digital system, as shown in Figure 4-44. Any logic diagram found in this text can be redrawn using the schematic entry tools in the Altera software to create a graphic design file. We will not focus on graphic design entry in this text because it is quite straightforward to pick up these skills in the laboratory. We will focus our examples on the methods that allow us to use HDL as an alternate means of describing a circuit. For more information on the Altera software, visit the Altera web site (http://www.altera.com).

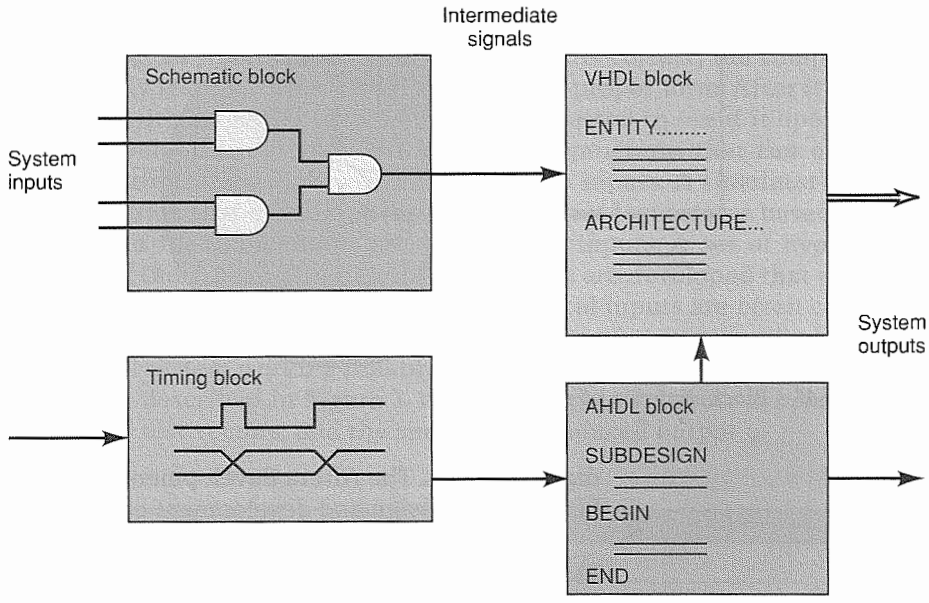


FIGURE 4-44 Combining blocks that were developed using different description methods.

This concept of using building blocks of circuits is called **hierarchical design**. Small, useful logic circuits can be defined in whatever manner is most convenient (graphic, HDL, timing, etc.) and then combined with other circuits to form a large section of a project. Sections can be combined and connected with other sections to form the whole system. Figure 4-45 shows the hierarchical structure of a DVD player using a block diagram. The outer box encloses the entire system. The dashed lines identify each major subsection, and each subsection contains individual circuits. Although it is not shown in this diagram, each circuit may be made up of smaller building blocks of common

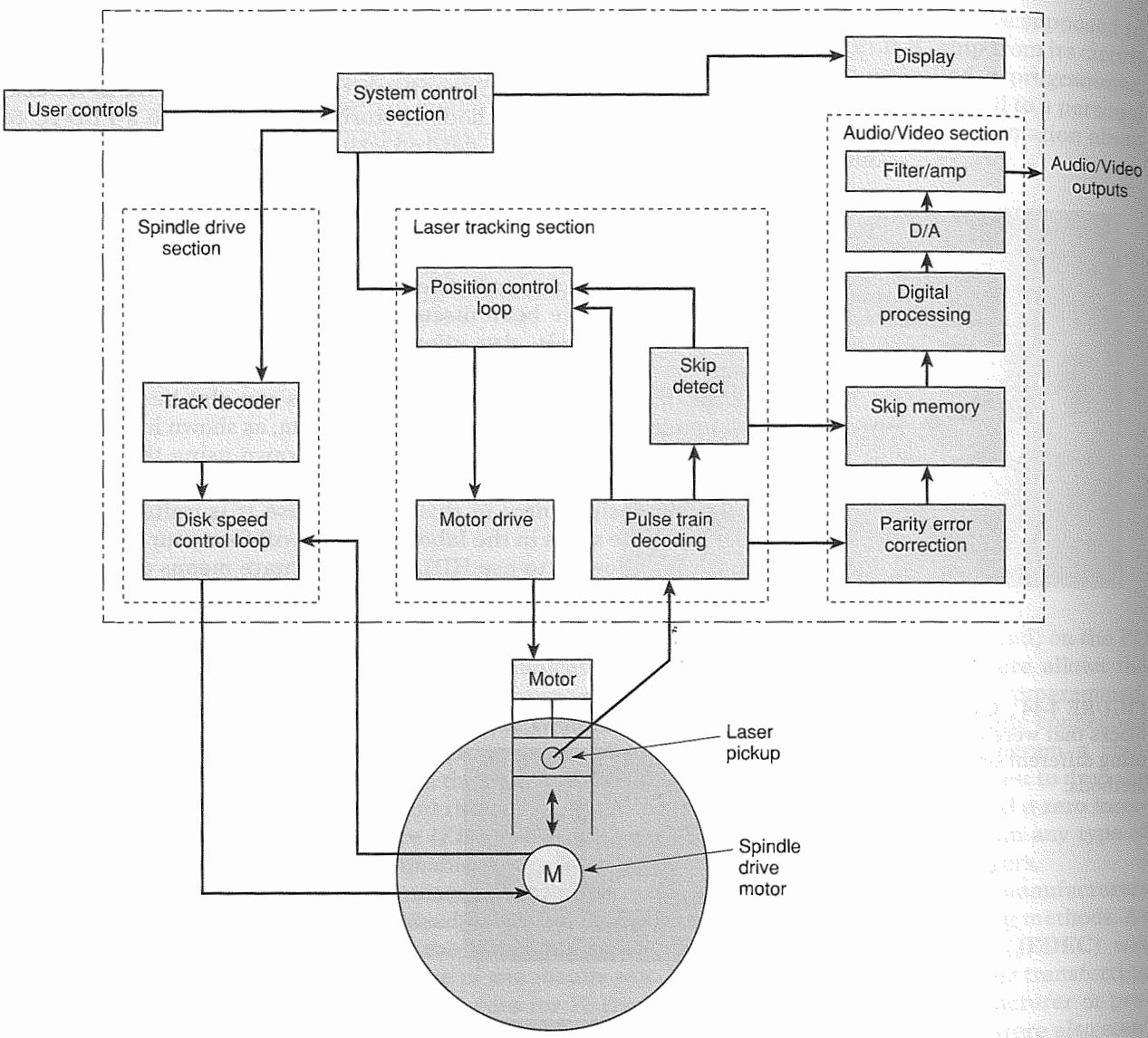


FIGURE 4-45 Block diagram of a DVD player.

digital circuits. The Altera development software makes this type of modular, hierarchical design and development easy to accomplish.

Design and Development Process

Another way you might see the hierarchy of a system like the DVD player just described is shown in Figure 4-46. The top level represents the entire system. It is made up of three subsections, each of which in turn is made up of the smaller circuits shown. Notice that this diagram does not show how the signals flow throughout the system but clearly identifies the various levels of the hierarchical structure of the project.

This type of diagram has led to the name for one of the most common methods of design: **top-down**. With this design approach, you start with the overall description of the entire system, such as the top box in Figure 4-46. Then you define several subsections that will make up the system. The subsections are further refined into individual circuits connected together.

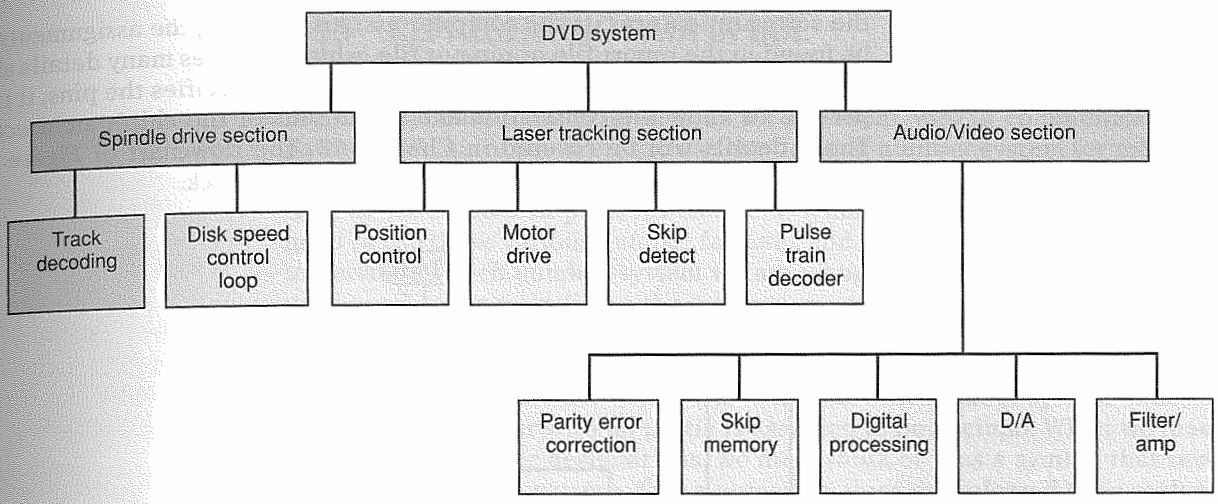


FIGURE 4-46 An organizational hierarchy chart.

Every one of these hierarchy levels has defined inputs, outputs, and behavior. Each can be tested individually before it is connected to the others.

After defining the blocks from the top down, the system is built from the bottom up. Each block in this system design has a design file that describes it. The lowest level blocks must be designed by opening a design file and writing a description of its operation. The designed block is then compiled using the development tools. The compiling process determines if you have made errors in your syntax. Until your syntax is correct, the computer cannot possibly translate your description into its proper form. After it has been compiled with no syntax errors, it should be tested to see if it operates correctly. Development systems offer simulator programs that run on the PC and simulate the way your circuit responds to inputs. A simulator is a computer program that calculates the correct output logic states based on a description of the logic circuit and the current inputs. A set of hypothetical inputs and their corresponding correct outputs are developed that will prove the block works as expected. These hypothetical inputs are often called **test vectors.** Thorough testing during simulation greatly increases the likelihood of the final system working reliably. Figure 4-47 shows the simulation file for the circuit described in Figure 3-13(a) of Chapter 3. Inputs a, b, and c were entered as test vectors, and the simulation produced output y.

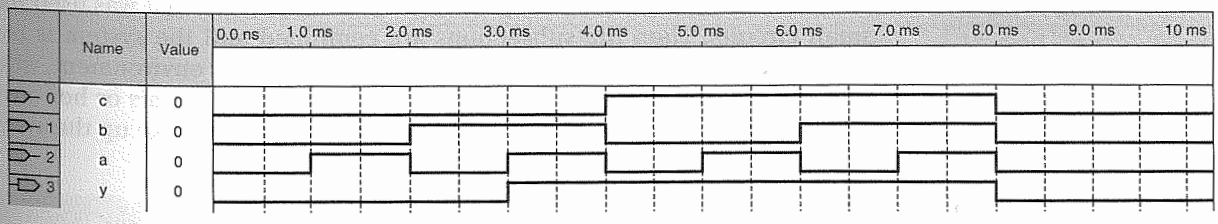


FIGURE 4-47 A timing simulation of a circuit described in HDL.

When the designer is satisfied that the design works, the design can be verified by actually programming a chip and testing. For a complex PLD, the designer can either let the development system assign pins and then lay out the final circuit board accordingly, or specify the pins for each signal using

the software features. If the compiler assigns the pins, the assignments can be found in the report file or pin-out file, which provides many details about the implementation of the design. If the designer specifies the pins, it is important to know the constraints and limitations of the chip's architecture. These details will be covered in Chapter 13. The flowchart of Figure 4-48 summarizes the design process for designing each block.

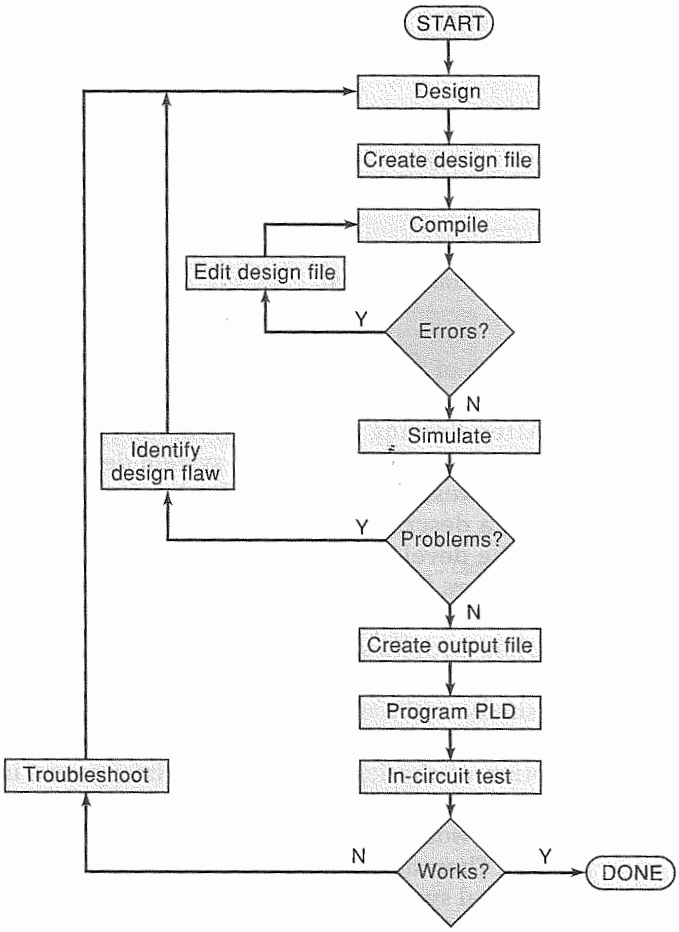


FIGURE 4-48 PLD development cycle flowchart.

After each circuit in a subsection has been tested, all can be combined and the subsection can be tested following the same process that was used for the small circuits. Then the subsections are combined and the system is tested. This approach lends itself very well to a typical project environment, where a team of people are working together, each responsible for his or her own circuits and sections that will ultimately come together to make up the system.

REVIEW QUESTIONS

1. What is actually being "programmed" in a PLD?
2. What bits (column, row) in Figure 4-42 must be connected to make Product 1 = AB?
3. What bits (column, row) in Figure 4-42 must be connected to make Product 3 = $A\overline{B}$?

4-15 REPRESENTING DATA IN HDL

Numeric data can be represented in various ways. We have studied the use of the hexadecimal number system as a convenient way to communicate bit patterns. We naturally prefer to use the decimal number system for numeric data, but computers and digital systems can operate only on binary information, as we studied in previous chapters. When we write in HDL, we often need to use these various number formats, and the computer must be able to understand which number system we are using. So far in this text, we have used a subscript to indicate the number system. For example, 101_2 was binary, 101_{16} was hexadecimal, and 101_{10} was decimal. Every programming language and HDL has its own unique way of identifying the various number systems, generally done with a prefix to indicate the number system. In most languages, a number with no prefix is assumed to be decimal. When we read one of these number designations, we must think of it as a symbol that represents a binary bit pattern. These numeric values are referred to as scalars or **literals**. Table 4-8 summarizes the methods of specifying values in binary, hex, and decimal for AHDL and VHDL.

TABLE 4-8 Designating number systems in HDL.

Number System	AHDL	VHDL	Bit Pattern	Decimal Equivalent
Binary	B"101"	B"101"	101	5
Hexadecimal	H"101"	X"101"	100000001	257
Decimal	101	101	1100101	101

EXAMPLE 4-29

Express the following bit pattern's numeric value in binary, hex, and decimal using AHDL and VHDL notation:

11001

Solution

Binary is designated the same in both AHDL and VHDL: **B "11001"**.
Converting the binary to hex, we have 19_{16}.
In AHDL: **H "19"**
In VHDL: **X "19"**
Converting the binary to decimal, we have 25_{10}.
Decimal is designated the same in both AHDL and VHDL: 25.

Bit Arrays/Bit Vectors

In Chapter 3, we declared names for inputs to and outputs of a very simple logic circuit. These were defined as bits, or single binary digits. What if we want to represent an input, output, or signal that is made up of several bits? In an HDL, we must define the type of the signal and its range of acceptable values.

To understand the concepts used in HDLs, let's first consider some conventions for describing bits of binary words in common digital systems. Suppose we have an eight-bit number representing the current temperature, and the number is coming into our digital system through an input port that we have named P1, as shown in Figure 4-49. We can refer to the individual bits of this port as P1 bit 0 for the least significant bit, on up to P1 bit 7 for the most significant bit.

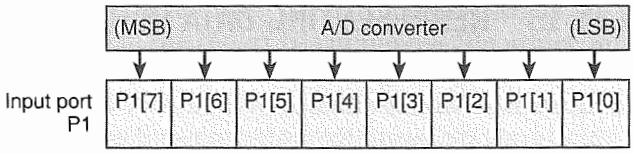


FIGURE 4-49 Bit array notation.

We can also describe this port by saying that it is named P1, with bits numbered 7 down to 0. The terms **bit array** and **bit vector** are often used to describe this type of data structure. It simply means that the overall data structure (eight-bit port) has a name (P1) and that each individual element (bit) has a unique **index** number (0–7) to describe that bit's position (and possibly its numeric weight) in the overall structure. The HDLs and computer programming languages take advantage of this notation. For example, the third bit from the right is designated as P1[2], and it can be connected to another signal bit by using an assignment operator.

EXAMPLE 4-30

Assume there is an eight-bit array named P1, as shown in Figure 4-49, and another four-bit array is named P5.

(a) Write the bit designation for the most significant bit of P1.
(b) Write the bit designation for the least significant bit of P5.
(c) Write an expression that causes the least significant bit of P5 to drive the most significant bit of P1.

Solution

(a) The name of the port is P1 and the most significant bit is bit 7. The proper designation for P1 bit 7 is P1[7].
(b) The name of the port is P5 and the least significant bit is bit 0. The proper designation for P5 bit 0 is P5[0].
(c) The driving signal (source) is placed on the right side of the assignment operator, and the driven signal (destination) is placed on the left: P1[7] = P5[0];.

AHDL

AHDL BIT ARRAY DECLARATIONS

In AHDL, port *p1* of Figure 4-49 is defined as an eight-bit input port, and the value on this port can be referred to using any number system, such as hex, binary, decimal, etc. The syntax for AHDL uses a name for the bit vector followed by the range of index designations, which are enclosed in square brackets. This declaration is included in the SUBDESIGN section. For example, to declare an eight-bit input port called *p1*, you would write

```
p1[7..0] :INPUT; --define an 8-bit input port
```

EXAMPLE 4-31

Declare a four-bit input named *keypad* using AHDL.

Solution

```
keypad[3..0] :INPUT;
```

Intermediate variables can also be declared as an array of bits. As with single bits, they are declared just after the I/O declarations in SUBDESIGN. As an example, the eight-bit temperature port *p1* can be assigned (connected) to a node named *temp*, as follows:

```
VARIABLE temp[7..0] :NODE;
BEGIN
    temp[] = p1[];
END;
```

Notice that the input port *p1* has the data applied to it, and it is driving the signal wires named *temp*. Think of the term on the right of the equals sign as the source of the data and the term on the left as the destination. The empty brackets [] mean that each of the correspanding bits in the two arrays are being connected. Individual bits can also be "connected" by specifying the bits inside the brackets. For example, to connect only the least significant bit of p1 to the LSB of temp, the statement would be temp[0] = p1[0];.

VHDL BIT VECTOR DECLARATIONS

In VHDL, port *p1* of Figure 4-49 is defined as an eight-bit input port, and the value on this port can be referred to using only binary literals. The syntax for VHDL uses a name for the bit vector followed by the mode (:IN), the type (**BIT_VECTOR**), and the range of index designations, which are enclosed in parentheses. This declaration is included in the ENTITY section. For example, to declare an eight-bit input port called *p1*, you would write

```
PORT (p1 :IN BIT_VECTOR (7 DOWNTO 0);
```

EXAMPLE 4-32

Declare a four-bit input named *keypad* using VHDL.

Solution

```
PORT(keypad :IN BIT_VECTOR (3 DOWNTO 0);
```

Intermediate signals can also be declared as an array of bits. As with single bits, they are declared just inside the ARCHITECTURE definition. As an example, the eight-bit temperature on port *p1* can be assigned (connected) to a signal named *temp*, as follows:

```
SIGNAL      temp :BIT_VECTOR (7 DOWNTO 0);
BEGIN
    temp <= p1;
END;
```

Notice that the input port *p1* has the data applied to it, and it is driving the signal wires named *temp*. No elements in the bit vector are specified, which means that all the bits are being connected. Individual bits can also be "connected" using signal assignments and by specifying the bit numbers inside parentheses. For example, to connect only the least significant bit of *p1* to the LSB of *temp*, the statement would be temp(0) <= p1(0);.

VHDL

VHDL is very particular regarding the definitions of each type of the data. The type "bit_vector" describes an array of individual bits. This is interpreted differently than an eight-bit binary number (called a scaler quantity), which has the type **integer**. Unfortunately, VHDL does not allow us to assign an integer value to a BIT_VECTOR signal directly. Data can be represented by any of the types shown in Table 4-9, but data assignments and other operations must be done between objects of the same type. For example, the compiler will not allow you to take a number from a keypad declared as an integer and connect it to four LEDs that are declared as BIT_VECTOR outputs. Notice in Table 4-9, under Possible Values, that individual BIT and STD_LOGIC data **objects** (e.g., signals, variables, inputs, and outputs) are designated by single quotes, whereas values assigned to BIT_VECTOR and STD_LOGIC_VECTOR types are strings of valid bit values enclosed in double quotes.

TABLE 4-9 Common VHDL data types.

Data Type	Sample Declaration	Possible Values	Use
BIT	y :OUT BIT;	'0' '1'	y <= '0';
STD_LOGIC	driver :STD_LOGIC	'0' '1' 'z' 'x' '-'	driver <= 'z';
BIT_VECTOR	bcd_data :BIT_VECTOR (3 DOWNTO 0);	"0101" "1001" "0000"	digit <= bcd_data;
STD_LOGIC_VECTOR	dbus :STD_LOGIC_VECTOR (3 DOWNTO 0);	"0Z1X"	IF rd = '0' THEN dbus <= "zzzz";
INTEGER	SIGNAL z:INTEGER RANGE –32 TO 31;	–32.. –2, –1,0,1,2 . . . 31	IF z > 5 THEN . . .

VHDL also offers some standardized data types that are necessary when using logic functions that are contained in the **libraries**. As you might have guessed, libraries are simply collections of little pieces of VHDL code that can be used in your hardware descriptions without reinventing the wheel. These libraries offer convenient functions, called **macrofunctions** or **maxplus2** functions, like many of the standard TTL devices that are described throughout this text. Rather than writing a new description of a familiar TTL device, we can simply pull its macrofunction out of the library and use it in our system. Of course, you need to get signals into and out of these macrofunctions, and the types of the signals in your code must match the types in the functions (which someone else wrote). This means that everyone must use the same standard data types.

When VHDL was standardized through the IEEE, many data types were created at the same time. The two that we will use in this text are **STD_LOGIC**, which is an expansion of the BIT type, and **STD_LOGIC_VECTOR**, which is an expansion of the BIT_VECTOR. As you recall, BIT type can have values of only '0' and '1'. The standard logic types are defined in the IEEE library and have a broader range of possible values than their built-in counterparts. The possible values for a STD_LOGIC type or for any element in a STD_LOGIC_VECTOR are given in Table 4-10. The names of these categories will make much more sense after we study the characteristics of logic circuits in Chapter 8. For now, we will show examples using values of only '1' and '0'.

TABLE 4-10 STD_LOGIC values.

'1'	Logic 1 (just like BIT type)
'0'	Logic 0 (just like BIT type)
'z'	High impedance*
'-'	don't care (just like you used in your K maps)
'U'	Uninitialized
'X'	Unknown
'W'	Weak unknown
'L'	Weak '0'
'H'	Weak '1'

*We will study tristate logic in Chapter 8.

REVIEW QUESTIONS

1. How would you declare a six-bit input array named push_buttons in (a) AHDL or (b) VHDL?
2. What statement would you use to take the MSB from the array in question 1 and put it on a single-bit output port named *z*? Use (a) AHDL or (b) VHDL.
3. In VHDL, what is the IEEE standard type that is equivalent to the BIT type?
4. In VHDL, what is the IEEE standard type that is equivalent to the BIT_VECTOR type?

4-16 TRUTH TABLES USING HDL

We have learned that a truth table is another way of expressing the operation of a circuit block. It relates the output of the circuit to every possible combination of its inputs. As we saw in Section 4-4, a truth table is the starting point for a designer to define how the circuit should operate. Then a Boolean expression is derived from the truth table and simplified using K maps or Boolean algebra. Finally the circuit is implemented from the final Boolean equation. Wouldn't it be great if we could go from the truth table directly to the final circuit without all those steps? We can do exactly that by entering the truth table using HDL.

AHDL

TRUTH TABLES USING AHDL

The code in Figure 4-50 uses AHDL to implement a circuit and uses a truth table to describe its operation. The truth table for this design was presented in Example 4-7. The key point of this example is the use of the TABLE keyword in AHDL. It allows the designer to specify the operation of the circuit just like you would fill out a truth table. On the first line after TABLE, the input variables (*a*,*b*,*c*) are listed exactly like you would create a column heading on a truth table. By including the three binary variables in parentheses, we tell the compiler that we want to use these three bits as a group and to refer to them as a three-bit binary number or bit pattern. The specific values for this bit pattern are listed below the group and are referred to as binary literals. The special operator (=>) is used in truth tables to separate the inputs from the output (*y*).

FIGURE 4-50 AHDL design file for Figure 4-7.

```
SUBDESIGN fig4_50
(
      a,b,c :INPUT;              --a is most significant
      y     :OUTPUT;             --define block output
)
BEGIN
      TABLE
            (a,b,c)              =>    y;     --column headings
            (0,0,0)              =>    0;
            (0,0,1)              =>    0;
            (0,1,0)              =>    0;
            (0,1,1)              =>    1;
            (1,0,0)              =>    0;
            (1,0,1)              =>    1;
            (1,1,0)              =>    1;
            (1,1,1)              =>    1;
      END TABLE;
END;
```

The TABLE in Figure 4-50 is intended to show the relationship between the HDL code and a truth table. A more common way of representing the input data heading is to use a variable bit array to represent the value on *a*, *b*, *c*. This method involves a declaration of the bit array on the line before BEGIN, such as:

```
VARIABLE in_bits[2..0]     :NODE;
```

Just before the TABLE keyword, the input bits can be assigned to the array, *inbits[ ]*:

```
in_bits[] = (a,b,c);
```

Grouping three independent bits in order like this is referred to as **concatenating**, and it is often done to connect individual bits to a bit array. The table heading on the input bit sets can be represented by *in_bits[ ]*, in this case. Note that as we list the possible combinations of the inputs, we have several options. We can make up a group of 1s and 0s in parentheses, as shown in Figure 4-50, or we can represent the same bit pattern using the equivalent binary, hex, or decimal number. It is up to the designer to decide which format is most appropriate depending on what the input variables represent.

VHDL

TRUTH TABLES USING VHDL: SELECTED SIGNAL ASSIGNMENT

The code in Figure 4-51 uses VHDL to implement a circuit using a **selected signal assignment** to describe its operation. It allows the designer to specify the operation of the circuit, just like you would fill out a truth table. The truth table for this design was presented in Example 4-7. The primary point of this example is the use of the WITH signal_name SELECT statement in VHDL. A secondary point presented here shows how to put the data into

```
ENTITY fig4_51 IS
PORT(
      a,b,c :IN BIT;          --a is most significant
      y     :OUT BIT);
END fig4_51;

ARCHITECTURE truth OF fig4_51 IS
      SIGNAL in_bits :BIT_VECTOR(2 DOWNTO 0);
      BEGIN
      in_bits <= a & b & c;     --concatenate input bits into bit_vector
            WITH in_bits SELECT
            y    <=      '0'  WHEN "000",       --Truth Table
                         '0'  WHEN "001",
                         '0'  WHEN "010",
                         '1'  WHEN "011",
                         '0'  WHEN "100",
                         '1'  WHEN "101",
                         '1'  WHEN "110",
                         '1'  WHEN "111";
END truth;
```

FIGURE 4-51 VHDL design file for Figure 4-7.

a format that can be used conveniently with the selected signal assignment. Notice that the inputs are defined in the ENTITY declaration as three independent bits *a*, *b*, and *c*. Nothing in this declaration makes one of these more significant than another. The order in which they are listed does not matter. We want to compare the current value of these bits with each of the possible combinations that could be present. If we drew out a truth table, we would decide which bit to place on the left (MSB) and which to place on the right (LSB). This is accomplished in VHDL by **concatenating** (connecting in order) the bit variables to form a bit vector. The concatenation operator is "&". A signal is declared as a BIT_VECTOR to receive the ordered set of input bits and is used to compare the input's value with the string literals contained in quotes. The output (*y*) is assigned (<=) a bit value ('0' or '1') WHEN *in_bits* contains the value listed in double quotes.

VHDL is very strict in the way it allows us to assign and compare objects such as signals, variables, constants, and literals. The output *y* is a BIT, and so it must be assigned a value of '0' or '1'. The SIGNAL *in_bits* is a three-bit BIT_VECTOR, so it must be compared with a three-bit string literal value. VHDL will not allow *in_bits* (a BIT_VECTOR) to be compared with a hex number like X "5" or a decimal number like 3. These scalar quantities would be valid for assignment or comparison with integers.

EXAMPLE 4-33

Declare three signals in VHDL that are single bits named *too_hot*, *too_cold*, and *just_right*. Combine (concatenate) these three bits into a three-bit signal called *temp_status*, with hot on the left and cold on the right.

Solution

1. Declare signals first in Architecture.

```
SIGNAL too_hot, too_cold, just_right :BIT;
SIGNAL temp_status :BIT_VECTOR (2 DOWNTO 0);
```

2. Write concurrent assignment statements between BEGIN and END.

```
temp_status <= too_hot & just_right & too_cold;
```

REVIEW QUESTIONS

1. How would you concatenate three bits x, y, and z into a three-bit array named *omega*? Use AHDL or VHDL.
2. How are truth tables implemented in AHDL?
3. How are truth tables implemented in VHDL?

4-17 DECISION CONTROL STRUCTURES IN HDL

In this section, we will examine methods that allow us to tell the digital system how to make "logical" decisions in much the same way that we make decisions every day. In Chapter 3, we learned that concurrent assignment statements are evaluated such that the order in which they are written has no effect on the circuit being described. When using **decision control structures**, the order in which we ask the questions does matter. To summarize this concept in the terms used in HDL documentation, statements that can be written in any sequence are called **concurrent**, and statements that are evaluated in the sequence in which they are written are called **sequential**. The sequence of sequential statements affects the circuit's operation.

The examples we have considered so far involve several individual bits. Many digital systems require inputs that represent a numeric value. Refer back to Example 4-8, in which the purpose of the logic circuit is to monitor the battery voltage measured by an A/D converter. The digital value is represented by a four-bit number coming from the A/D into the logic circuit. These inputs are not independent binary variables but rather four binary digits of a number representing battery voltage. We need to give the data the correct type that will allow us to use it as a number.

IF/ELSE

Truth tables are great for listing all the possible combinations of independent variables, but there are better ways to handle numeric data. As an example, when a person leaves for school or work in the morning, she must make a logical decision about wearing a coat. Let's assume she decides this issue based only on the current temperature. How many of us would reason as follows?

I will wear a coat if the temperature is 0.
I will wear a coat if the temperature is 1.
I will wear a coat if the temperature is 2. . . .
I will wear a coat if the temperature is 55.

I will *not* wear a coat if the temperature is 56.
I will *not* wear a coat if the temperature is 57.
I will *not* wear a coat if the temperature is 58. . . .
I will *not* wear a coat if the temperature is 99.

This method is similar to the truth table approach of describing the decision. For every possible input, she decides what the output should be. Of course, what she would really do is decide as follows:

I will wear a coat if the temperature is less than 56 degrees.
Otherwise, I will *not* wear a coat.

An HDL gives us the power to describe logic circuits using this type of reasoning. First, we must describe the inputs as a *number within a given range,* and then we can write statements that decide what to do to the outputs based on the *value* of the incoming number. In most computer programming languages, as well as HDLs, these types of decisions are made using an IF/THEN/ELSE control structure. Whenever the decision is between doing something and doing nothing, an **IF/THEN** construct is used. The keyword IF is followed by a statement that is true or false. IF it is true, THEN do whatever is specified. In the event that the statement is false, no action is taken. Figure 4-52(a) shows graphically how this decision works. The diamond shape represents the decision being made by evaluating the statement contained within the diamond. Every decision has two possible outcomes: true or false. In this example, if the statement is false, no action is taken.

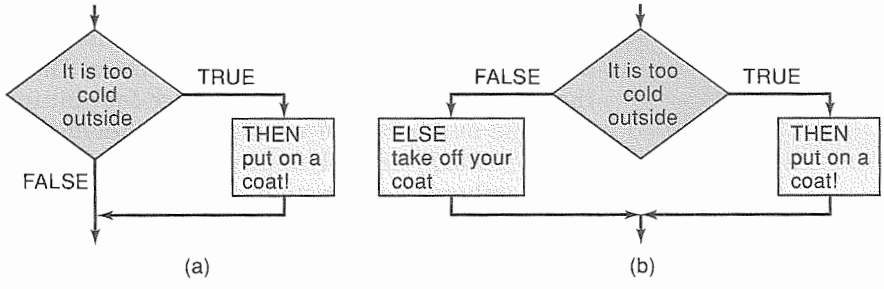


FIGURE 4-52 Logical flow of (a) IF/THEN and (b) IF/THEN/ELSE constructs.

In some cases it is not enough only to decide to act or not to act, but rather we must choose between two different actions. For example, in our analogy about the decision to wear a coat or not using IF/THEN, it is assumed that the person is not initially wearing a coat when making the decision (because it was the beginning of the day). If she already has a coat on when making this decision (say at mid-day), there is no provision for her to take the coat off if it is too warm.

When decisions demand two possible actions, the **IF/THEN/ELSE** control structure is used, as shown in Figure 4-52(b). Here again, the statement is evaluated as true or false. The difference is that, when the statement is false, a different action is performed. One of the two actions must occur with this construct. We can state it verbally as, "IF the statement is true, THEN do this. ELSE do that." In our coat analogy, this control structure would work, regardless of whether the person's coat was on or off initially.

Example 4-8 gave a simple example of a logic circuit that has as its input a numeric value representing battery voltage from an A/D converter. The inputs A, B, C, D are actually binary digits in a four-bit number, with A being the MSB and D being the LSB. Figure 4-53 shows the same circuit with the

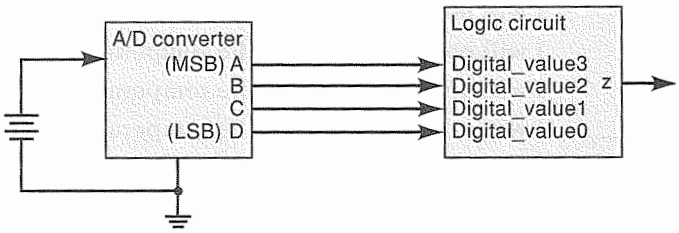


FIGURE 4-53 Logic circuit similar to Example 4-8.

inputs labeled as a four-bit number called *digital_value.* The relationship between bits is as follows:

A	*digital_value[3]*	digital value bit 3 (MSB)
B	*digital_value[2]*	digital value bit 2
C	*digital_value[1]*	digital value bit 1
D	*digital_value[0]*	digital value bit 0 (LSB)

The input can be treated as a decimal number between 0 and 15 if we specify the correct type of the input variable.

AHDL

IF/THEN/ELSE USING AHDL

In AHDL, the inputs can be specified as a binary number made up of multiple bits by assigning a variable name followed by a list of the bit positions, as shown in Figure 4-54. The name is *digital_value,* and the bit positions range from 3 down to 0. Notice how simple the code becomes using this method along with an IF/ELSE construct. The IF is followed by a statement that refers to the value of the entire four-bit input variable and compares it with the number 6. Of course, 6 is a decimal form of a scalar quantity and *digital_value[ ]* actually represents a binary number. The compiler can interpret numbers in any system, so it creates a logic circuit that compares the binary value of *digital_value* with the binary number for 6 and decides if this statement is true or false. If it is true, THEN the next statement (z = VCC) is used to assign *z* a value. Notice that in AHDL, we must use VCC for a logic 1 and GND for a logic 0 when assigning a logic level to a single bit. When *digital_value* is 6 or less, it follows the statement after ELSE (z = GND). The END IF; terminates the control structure.

FIGURE 4-54 AHDL version.

```
SUBDESIGN fig4_54
(
    digital_value[3..0]    :INPUT;    -- define inputs to block
    z                      :OUTPUT;   -- define block output
)
BEGIN
    IF digital_value[] > 6 THEN
            z = VCC;                  -- output a 1
    ELSE    z = GND;                  -- output a 0
    END IF;
END;
```

IF/THEN/ELSE USING VHDL

In VHDL, the critical issue is the declaration of the type of inputs. Refer to Figure 4-55. The input is treated as a single variable called *digital_value*. Because its type is declared as INTEGER, the compiler knows to treat it as a number. By specifying a range of 0 to 15, the compiler knows it is a four-bit number. Notice that RANGE does not specify the index number of a bit vector but rather the limits of the numeric value of the integer. Integers are treated differently than bit arrays (BIT_VECTOR) in VHDL. An integer can be compared with other numbers using inequality operators. A BIT_VECTOR cannot be used with inequality operators.

FIGURE 4-55 VHDL version.

```
ENTITY fig4_55 IS
PORT( digital_value :IN INTEGER RANGE 0 TO 15; -- 4-bit input
      z             :OUT BIT);
END fig4_55;

ARCHITECTURE Decision OF fig4_55 IS

BEGIN
   PROCESS (digital_value)
      BEGIN
         IF (digital_value > 4) THEN
            z <= '1';
         ELSE
            z <= '0';
      END IF;
END PROCESS ;
END Decision;
```

To use the IF/THEN/ELSE control structure, VHDL requires that the code be put inside a "PROCESS." The statements that occur within a process are *sequential,* meaning that the order in which they are written affects the operation of the circuit. The keyword **PROCESS** is followed by a list of variables called a **sensitivity list**, which is a list of variables to which the process code must respond. Whenever *digital_value* changes, it causes the process code to be reevaluated. Even though we know *digital_value* is really a four-bit binary number, the compiler will evaluate it as a number between the equivalent decimal values of 0 and 15. IF the statement in parentheses is true, THEN the next statement is applied (*z* is assigned a value of logic 1). If this statement is not true, the logic follows the ELSE clause and assigns a value of 0 to *z*. The END IF; terminates the control structure, and the END PROCESS; terminates the evaluation of the sequential statements.

ELSIF

We often need to choose among many possible actions, depending on the situation. The IF construct chooses whether to perform a set of actions or not. The IF/ELSE construct selects one out of two possible actions. By combining

IF and ELSE decisions, we can create a control structure referred to as **ELSIF**, which chooses one of many possible outcomes. The decision structure is shown graphically in Figure 4-56.

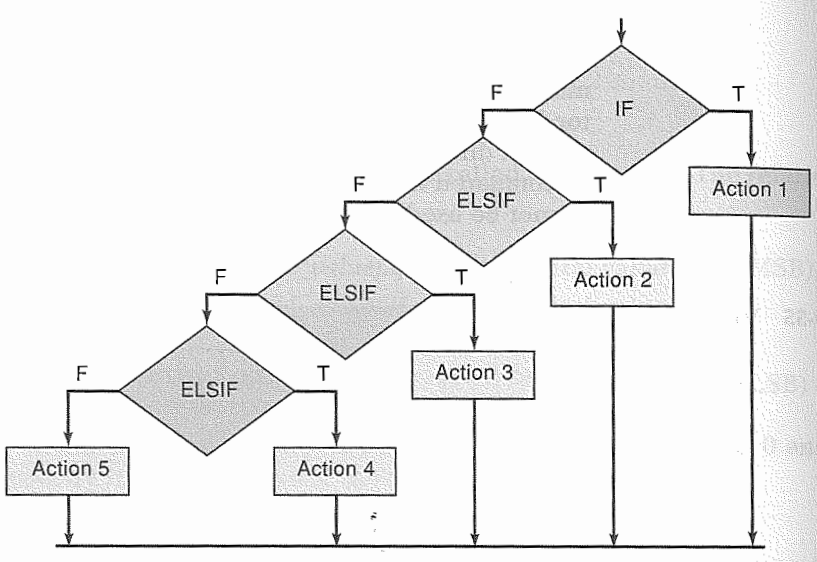


FIGURE 4-56 Flowchart for multiple decisions using IF/ELSIF.

Notice that as each condition is evaluated, it either performs an action if true or goes on to evaluate the next condition. Each action is associated with one condition, and there is no chance to select more than one action. Note also that the conditions used to decide the appropriate action can be any expression that evaluates as true or false. This fact allows the designer to use the inequality operators to choose an action based on a range of input values. As an example of this application, let's consider the temperature-measuring system that uses an A/D converter, as described in Figure 4-57. Suppose that we want to indicate when the temperature is in a certain range, which we will refer to as Too Cold, Just Right, and Too Hot.

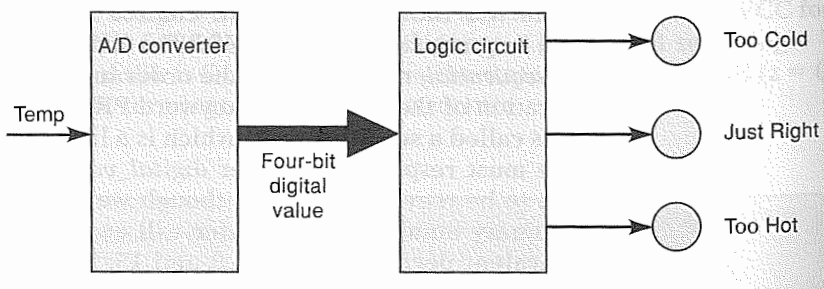


FIGURE 4-57 Temperature range indicator circuit.

The relationship between the digital values for temperature and the categories is

Digital Values	*Category*
0000–1000	Too Cold
1001–1010	Just Right
1011–1111	Too Hot

We can express the decision-making process for this logic circuit as follows:

IF the digital value is less than or equal to 8, THEN light only the Too Cold indicator.

ELSE IF the digital value is greater than 8 AND less than 11, THEN light only the Just Right indicator.

ELSE light only the Too Hot indicator.

AHDL

ELSIF USING AHDL

The AHDL code in Figure 4-58 defines the inputs as a four-bit binary number. The outputs are three individual bits that drive the three range indicators. This example uses an intermediate variable (*status*) that allows us to assign a bit pattern representing the three conditions of *too_cold, just_right,* and *too_hot.* The sequential section of the code uses the IF, ELSIF, ELSE to identify the range in which the temperature lies and assigns the correct bit pattern to status. In the last statement, the bits of *status* are connected to the actual output port bits. These bits have been ordered in a group that relates to the bit patterns assigned to *status[ ].* This could also have been written as three concurrent statements: too_cold = status[2]; just_right = status[1]; too_hot = status[0];.

```
SUBDESIGN fig4_58
(
   digital_value[3..0]            :INPUT; --define inputs to block
   too_cold, just_right, too_hot :OUTPUT;--define outputs
)
VARIABLE
status[2..0]    :NODE;--holds state of too_cold, just_right, too_hot
BEGIN
   IF       digital_value[] <= 8 THEN status[] = b"100";
   ELSIF    digital_value[] > 8 AND digital_value[] < 11 THEN
         status[] = b"010";
   ELSE  status[] = b"001";
   END IF;
   (too_cold, just_right, too_hot) = status[]; -- update output bits
END;
```

FIGURE 4-58 Temperature range example in AHDL using ELSIF.

VHDL

ELSIF USING VHDL

The VHDL code in Figure 4-59 defines the inputs as a four-bit integer. The outputs are three individual bits that drive the three range indicators. This example uses an intermediate signal (*status*) that allows us to assign a bit pattern representing all three conditions of *too_cold, just_right,* and *too_hot.* The process section of the code uses the IF, ELSIF, and ELSE to identify the range in which the temperature lies and assigns the correct bit pattern to status. In the last three statements, each bit of *status* is connected to the correct output port bit.

```
ENTITY fig4_59 IS
PORT(digital_value:IN INTEGER RANGE 0 TO 15;     -- declare 4-bit input
      too_cold, just_right, too_hot :OUT BIT);
END fig4_59 ;

ARCHITECTURE howhot OF fig4_59 IS
SIGNAL status   :BIT_VECTOR (2 downto 0);
BEGIN
   PROCESS (digital_value)
      BEGIN
         IF (digital_value <= 8) THEN status <= "100";
         ELSIF (digital_value > 8 AND digital_value < 11) THEN
               status <= "010";
         ELSE  status <= "001";
         END IF;
      END PROCESS ;
   too_cold     <= status(2);      -- assign status bits to output
   just_right   <= status(1);
   too_hot      <= status(0);
END howhot;
```

FIGURE 4-59 Temperature range example in VHDL using ELSIF.

CASE

One more important control structure is useful for choosing actions based on current conditions. It is called by various names, depending on the programming language, but it nearly always involves the word **CASE**. This construct determines the value of an expression or object and then goes through a list of possible values (cases) for the expression or object being evaluated. Each case has a list of actions that should take place. A CASE construct is different from an IF/ELSIF because a case correlates one unique value of an object with a set of actions. Recall that an IF/ELSIF correlates a set of actions with a true statement. There can be only one match for a CASE statement. An IF/ELSIF can have more than one statement that is true, but will THEN perform the action associated with the first true statement it evaluates.

Another important point in the HDL examples of Figures 4-60 through 4-64 is the need to combine several independent variables into a set of bits, called a bit vector. Recall that this action of linking several bits in a particular order is called *concatenation*. It allows us to consider the bit pattern as an ordered group.

AHDL

CASE USING AHDL

The AHDL example in Figure 4-60 demonstrates a case construct implementing the circuit of Figure 4-9 (see also Table 4-3). It uses individual bits as its inputs. In the first statement after BEGIN, these bits are concatenated and assigned to the intermediate variable called *status*. The CASE statement evaluates the variable *status* and finds the bit pattern (following the keyword WHEN) that matches the value of *status*. It then performs the action described following =>. In this example, it simply assigns logic 0 to the output for each of the three specified cases. All *other* cases result in a logic 1 on the output.

FIGURE 4-60 Figure 4-9 represented in AHDL.

```
SUBDESIGN fig4_60
{
   p, q, r          :INPUT;        -- define inputs to block
   a                :OUTPUT;       -- define outputs
}
VARIABLE
  status[2..0]      :NODE;
BEGIN
   status[]= (p, q, r);  -- link input bits in order
   CASE status[] IS
      WHEN b"100"       => a = GND;
      WHEN b"101"       => a = GND;
      WHEN b"110"       => a = GND;
      WHEN OTHERS       => a = VCC;
   END CASE;
END;
```

CASE USING VHDL

The VHDL example in Figure 4-61 demonstrates the case construct implementing the circuit of Figure 4-9 (see also Table 4-3). It uses individual bits as its inputs. In the first statement after BEGIN, these bits are concatenated and assigned to the intermediate variable called *status* using the & operator. The CASE statement evaluates the variable status and finds the bit pattern (following the keyword WHEN) that matches the value of *status*. It then performs the action described following =>. In this simple example, it merely assigns logic 0 to the output for each of the three specified cases. All *other* cases result in a logic 1 on the output.

FIGURE 4-61 Figure 4-9 represented in VHDL.

```
ENTITY fig4_61 IS
PORT( p, q, r      :IN bit;              --declare 3 bits input
      a            :OUT BIT);
END fig4_61;

ARCHITECTURE copy OF fig4_61 IS
SIGNAL status      :BIT_VECTOR (2 downto 0);
BEGIN
   status <= p & q & r;                  --link bits in order
   PROCESS (status)
      BEGIN
         CASE status IS
            WHEN "100" => a <= '0';
            WHEN "101" => a <= '0';
            WHEN "110" => a <= '0';
            WHEN OTHERS => a <= '1';
         END CASE;
      END PROCESS;
END copy;
```

VHDL

EXAMPLE 4-34

A coin detector in a vending machine accepts quarters, dimes, and nickels and activates the corresponding digital signal (Q, D, N) only when the correct coin is present. It is physically impossible for multiple coins to be present at the same time. A digital circuit must use the Q, D, and N signals as inputs and produce a binary number representing the value of the coin as shown in Figure 4-62. Write the AHDL and VHDL code.

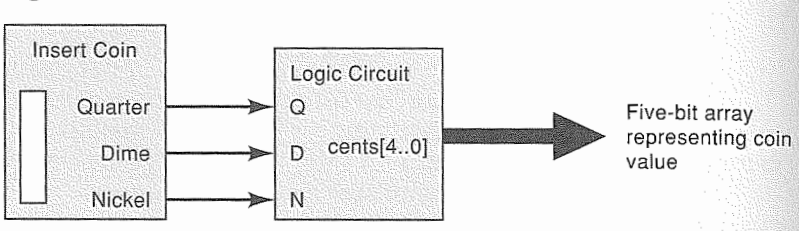


FIGURE 4-62 A coin detector circuit for a vending machine.

Solution

This is an ideal application of the CASE construct to describe the correct operation. The outputs must be declared as five-bit numbers in order to represent up to 25 cents. Figure 4-63 shows the AHDL solution and Figure 4-64 shows the VHDL solution.

AHDL

```
SUBDESIGN   fig4_63
(
   q, d, n          :INPUT;      -- define quarter, dime, nickel
   cents[4..0]      :OUTPUT;     -- define binary value of coins
)
BEGIN
   CASE (q, d, n) IS             -- group coins in an ordered set
      WHEN b"001" => cents[] = 5;
      WHEN b"010" => cents[] = 10;
      WHEN b"100" => cents[] = 25;
      WHEN others => cents[] = 0;
   END CASE;
END;
```

FIGURE 4-63 An AHDL coin detector.

VHDL

```
ENTITY   fig4_64 IS
PORT( q, d, n:IN BIT;                        -- quarter, dime, nickel
      cents  :OUT INTEGER RANGE 0 TO 25);  -- binary value of coins
END fig4_64;
ARCHITECTURE detector of fig4_64 IS
   SIGNAL   coins :BIT_VECTOR(2 DOWNTO 0);-- group the coin sensors
   BEGIN
      coins <= (q & d & n);                  -- assign sensors to group
      PROCESS (coins)
         BEGIN
            CASE (coins) IS
               WHEN "001"   => cents <= 5;
               WHEN "010"   => cents <= 10;
               WHEN "100"   => cents <= 25;
               WHEN others => cents <= 0;
            END CASE;
         END PROCESS;
   END detector;
```

FIGURE 4-64 A VHDL coin detector.

REVIEW QUESTIONS

1. Which control structure decides to do or not to do?
2. Which control structure decides to do this or to do that?
3. Which control structure(s) decides which one of several different actions to take?
4. Declare an input named *count* that can represent a numeric quantity as big as 205. Use AHDL or VHDL.

SUMMARY

1. The two general forms for logic expressions are the sum-of-products form and the product-of-sums form.
2. One approach to the design of a combinatorial logic circuit is to (1) construct its truth table, (2) convert the truth table to a sum-of-products expression, (3) simplify the expression using Boolean algebra or K mapping, (4) implement the final expression.
3. The K map is a graphical method for representing a circuit's truth table and generating a simplified expression for the circuit output.
4. An exclusive-OR circuit has the expression $x = A\overline{B} + \overline{A}B$. Its output x will be HIGH only when inputs A and B are at opposite logic levels.
5. An exclusive-NOR circuit has the expression $x = \overline{A}\,\overline{B} + AB$. Its output x will be HIGH only when inputs A and B are at the same logic level.
6. Each of the basic gates (AND, OR, NAND, NOR) can be used to enable or disable the passage of an input signal to its output.
7. The main digital IC families are the TTL and CMOS families. Digital ICs are available in a wide range of complexities (gates per chip), from the basic to the high-complexity logic functions.
8. To perform basic troubleshooting requires—at a minimum—an understanding of circuit operation, a knowledge of the types of possible faults, a complete logic-circuit connection diagram, and a logic probe.
9. A programmable logic device (PLD) is an IC that contains a large number of logic gates whose interconnections can be programmed by the user to generate the desired logic relationship between inputs and outputs.
10. To program a PLD, you need a development system that consists of a computer, PLD development software, and a programmer fixture that does the actual programming of the PLD chip.
11. The Altera system allows convenient hierarchical design techniques using any form of hardware description.
12. The type of data objects must be specified so that the HDL compiler knows the range of numbers to be represented.
13. Truth tables can be entered directly into the source file using the features of HDL.
14. Logical control structures such as IF, ELSE, and CASE can be used to describe the operation of a logic circuit, making the code and the problem's solution much more straightforward.

IMPORTANT TERMS

sum-of-products (SOP)
product-of-sums (POS)
Karnaugh map (K map)
looping
don't-care condition
exclusive-OR (XOR)
exclusive-NOR (XNOR)
parity generator
parity checker
enable/disable
dual-in-line package (DIP)
SSI, MSI, LSI, VLSI, ULSI, GSI
transistor-transistor logic (TTL)
complementary metal-oxide-semiconductor (CMOS)
indeterminate
floating
logic probe
contention
programmer
ZIF socket
JEDEC
ISP
JTAG
hierarchical design
top-down
test vectors
literals
bit array
bit vector
index
BIT_VECTOR
integer
objects
libraries
macrofunction
maxplus2
STD_LOGIC
STD_LOGIC_VECTOR
concatenate
selected signal assignment
decision control structure
concurrent
sequential
IF/THEN
ELSE
PROCESS
sensitivity list
ELSIF
CASE

PROBLEMS

SECTIONS 4-2 AND 4-3

B 4-1.* Simplify the following expressions using Boolean algebra.

(a) $x = ABC + \overline{A}C$

(b) $y = (Q + R)(\overline{Q} + \overline{R})$

(c) $w = ABC + A\overline{B}C + \overline{A}$

(d) $q = \overline{RST(R + S + T)}$

(e) $x = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + ABC + A\overline{B}\,\overline{C} + A\overline{B}C$

(f) $z = (B + \overline{C})(\overline{B} + C) + \overline{\overline{A} + B + \overline{C}}$

(g) $y = \overline{(C + D)} + \overline{A}C\overline{D} + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}CD + AC\overline{D}$

(h) $x = AB(\overline{\overline{C}\overline{D}}) + \overline{A}BD + \overline{B}\,\overline{C}\,\overline{D}$

B 4-2. Simplify the circuit of Figure 4-65 using Boolean algebra.

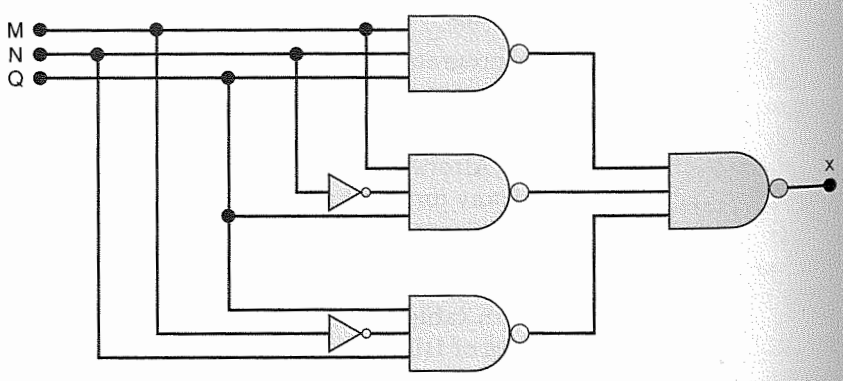


FIGURE 4-65 Problems 4-2 and 4-3.

*Answers to problems marked with an asterisk can be found in the back of the text.

B 4-3.* Change each gate in Problem 4-2 to a NOR gate, and simplify the circuit using Boolean algebra.

SECTION 4-4

B, D 4-4.* Design the logic circuit corresponding to the truth table shown in Table 4-11.

TABLE 4-11

A	B	C	x
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

B, D 4-5. Design a logic circuit whose output is HIGH *only* when a majority of inputs A, B, and C are LOW.

D 4-6. A manufacturing plant needs to have a horn sound to signal quitting time. The horn should be activated when either of the following conditions is met:

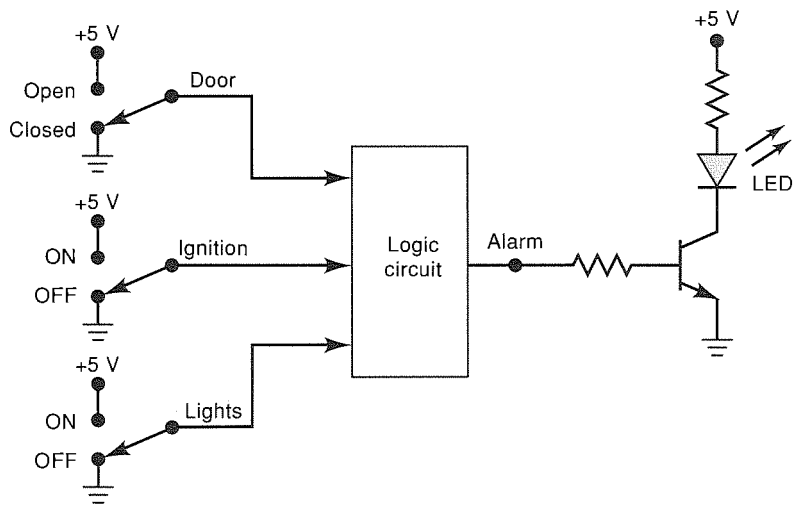
1. It's after 5 o'clock and all machines are shut down.
2. It's Friday, the production run for the day is complete, and all machines are shut down.

Design a logic circuit that will control the horn. (*Hint:* Use four logic input variables to represent the various conditions; for example, input A will be HIGH only when the time of day is 5 o'clock or later.)

D 4-7.* A four-bit binary number is represented as $A_3A_2A_1A_0$, where A_3, A_2, A_1, and A_0 represent the individual bits and A_0 is equal to the LSB. Design a logic circuit that will produce a HIGH output whenever the binary number is greater than 0010 and less than 1000.

D 4-8. Figure 4-66 shows a diagram for an automobile alarm circuit used to detect certain undesirable conditions. The three switches are used to

FIGURE 4-66 Problem 4-8.



indicate the status of the door by the driver's seat, the ignition, and the headlights, respectively. Design the logic circuit with these three switches as inputs so that the alarm will be activated whenever either of the following conditions exists:

- The headlights are on while the ignition is off.
- The door is open while the ignition is on.

4-9.* Implement the circuit of Problem 4-4 using all NAND gates.

4-10. Implement the circuit of Problem 4-5 using all NAND gates.

SECTION 4-5

B 4-11. Determine the minimum expression for each K map in Figure 4-67. Pay particular attention to step 5 for the map in (a).

FIGURE 4-67 Problem 4-11.

(a)*

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	1	1	0	0
AB	0	0	0	1
$A\bar{B}$	0	0	1	1

(b)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	1	1
$\bar{A}B$	1	0	0	1
AB	0	0	0	0
$A\bar{B}$	1	0	1	1

(c)

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	0	0
AB	1	0
$A\bar{B}$	1	X

B 4-12. For the truth table below, create a 2 × 2 K map, group terms, and simplify. Then look at the truth table again to see if the expression is true for every entry in the table.

A	B	y
0	0	1
0	1	1
1	0	0
1	1	0

B 4-13. Starting with the truth table in Table 4-11, use a K map to find the simplest SOP equation.

B 4-14. Simplify the expression in (a)* Problem 4-1(e) using a K map. (b) Problem 4-1(g) using a K map. (c)* Problem 4-1(h) using a K map.

B 4-15.* Obtain the output expression for Problem 4-7 using a K map.

C, D 4-16. Figure 4-68 shows a *BCD counter* that produces a four-bit output representing the BCD code for the number of pulses that have been applied to the counter input. For example, after four pulses have occurred, the counter outputs are $DCBA = 0100_2 = 4_{10}$. The counter resets to 0000 on the tenth pulse and starts counting over again. In other words, the *DCBA* outputs will never represent a number greater than $1001_2 = 9_{10}$.

(a)* Design the logic circuit that produces a HIGH output whenever the count is 2, 3, or 9. Use K mapping and take advantage of the don't-care conditions.

(b) Repeat for $x = 1$ when $DCBA = 3, 4, 5, 8$.

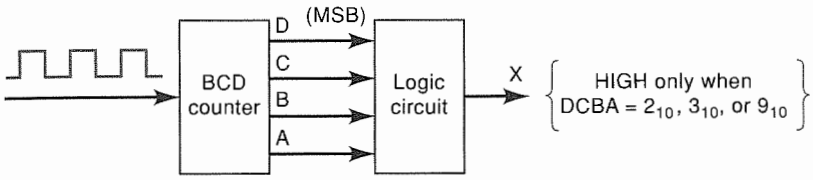


FIGURE 4-68 Problem 4-16.

D 4-17.* Figure 4-69 shows four switches that are part of the control circuitry in a copy machine. The switches are at various points along the path of the copy paper as the paper passes through the machine. Each switch is normally open, and as the paper passes over a switch, the switch closes. It is impossible for switches SW1 and SW4 to be closed at the same time. Design the logic circuit to produce a HIGH output whenever *two or more* switches are closed at the same time. Use K mapping and take advantage of the don't-care conditions.

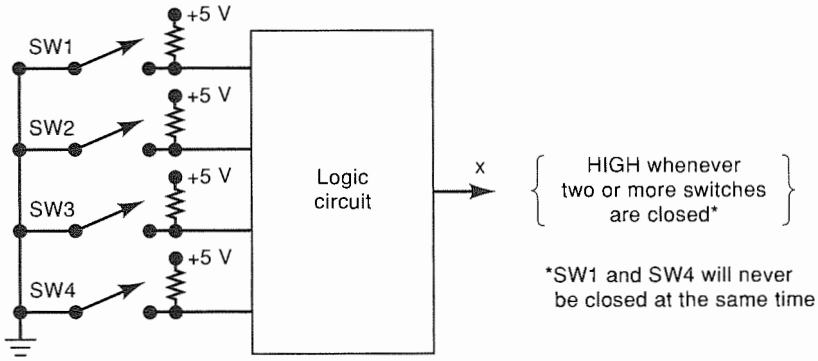


FIGURE 4-69 Problem 4-17.

B 4-18.* Example 4-3 demonstrated algebraic simplification. Step 3 resulted in the SOP equation $z = \overline{A}\,\overline{B}C + \overline{A}C\overline{D} + \overline{A}B\overline{C}\,\overline{D} + A\overline{B}C$. Use a K map to prove that this equation can be simplified further than the answer shown in the example.

C 4-19. Use Boolean algebra to arrive at the same result obtained by the K map method of Problem 4-18.

SECTION 4-6

B 4-20. (a) Determine the output waveform for the circuit of Figure 4-70.
(b) Repeat with the *B* input held LOW.
(c) Repeat with *B* held HIGH.

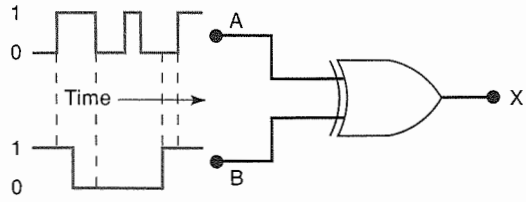
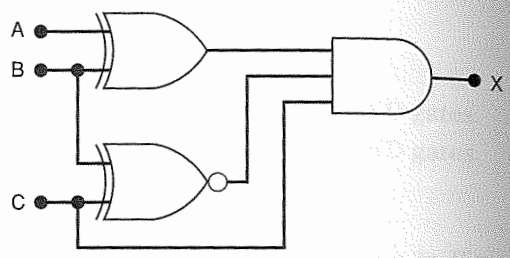


FIGURE 4-70 Problem 4-20.

B 4-21.* Determine the input conditions needed to produce $x = 1$ in Figure 4-71.

FIGURE 4-71 Problem 4-21.



B 4-22. Design a circuit that produces a HIGH out only when all three inputs are the same level.

(a) Use a truth table and K map to produce the SOP solution.

(b) Use two-input XOR and other gates to find a solution. (*Hint:* Recall the transitive property from algebra. . . if $a = b$ and $b = c$ then $a = c$.)

B 4-23.* A 7486 chip contains four XOR gates. Show how to make an XNOR gate using only a 7486 chip. *Hint:* See Example 4-16.

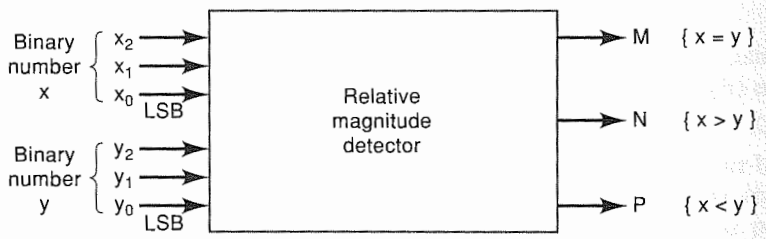
B 4-24.* Modify the circuit of Figure 4-23 to compare two four-bit numbers and produce a HIGH output when the two numbers match exactly.

B 4-25. Figure 4-72 represents a *relative-magnitude detector* that takes two three-bit binary numbers, $x_2x_1x_0$ and $y_2y_1y_0$, and determines whether they are equal and, if not, which one is larger. There are three outputs, defined as follows:

1. $M = 1$ only if the two input numbers are equal.
2. $N = 1$ only if $x_2x_1x_0$ is greater than $y_2y_1y_0$.
3. $P = 1$ only if $y_2y_1y_0$ is greater than $x_2x_1x_0$.

Design the logic circuitry for this detector. The circuit has *six* inputs and *three* outputs and is therefore much too complex to handle using the truth-table approach. Refer to Example 4-17 as a hint about how you might start to solve this problem.

FIGURE 4-72 Problem 4-25.



MORE DESIGN PROBLEMS

C, D 4-26.* Figure 4-73 represents a multiplier circuit that takes two-bit binary numbers, x_1x_0 and y_1y_0, and produces an output binary number $z_3z_2z_1z_0$ that is equal to the arithmetic product of the two input numbers. Design the logic circuit for the multiplier. (*Hint:* The logic circuit will have four inputs and four outputs.)

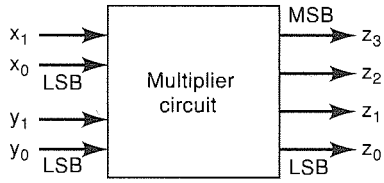


FIGURE 4-73 Problem 4-26.

D 4-27. A BCD code is being transmitted to a remote receiver. The bits are A_3, A_2, A_1, and A_0, with A_3 as the MSB. The receiver circuitry includes a *BCD error detector* circuit that examines the received code to see if it is a legal BCD code (i.e., ≤1001). Design this circuit to produce a HIGH for any error condition.

D 4-28.* Design a logic circuit whose output is HIGH whenever A and B are both HIGH as long as C and D are either both LOW or both HIGH. Try to do this without using a truth table. Then check your result by constructing a truth table from your circuit to see if it agrees with the problem statement.

D 4-29. Four large tanks at a chemical plant contain different liquids being heated. Liquid-level sensors are being used to detect whenever the level in tank A or tank B rises above a predetermined level. Temperature sensors in tanks C and D detect when the temperature in either of these tanks drops below a prescribed temperature limit. Assume that the liquid-level sensor outputs A and B are LOW when the level is satisfactory and HIGH when the level is too high. Also, the temperature-sensor outputs C and D are LOW when the temperature is satisfactory and HIGH when the temperature is too low. Design a logic circuit that will detect whenever the level in tank A or tank B is too high at the same time that the temperature in either tank C or tank D is too low.

C, D 4-30.* Figure 4-74 shows the intersection of a main highway with a secondary access road. Vehicle-detection sensors are placed along lanes C and D (main road) and lanes A and B (access road). These sensor

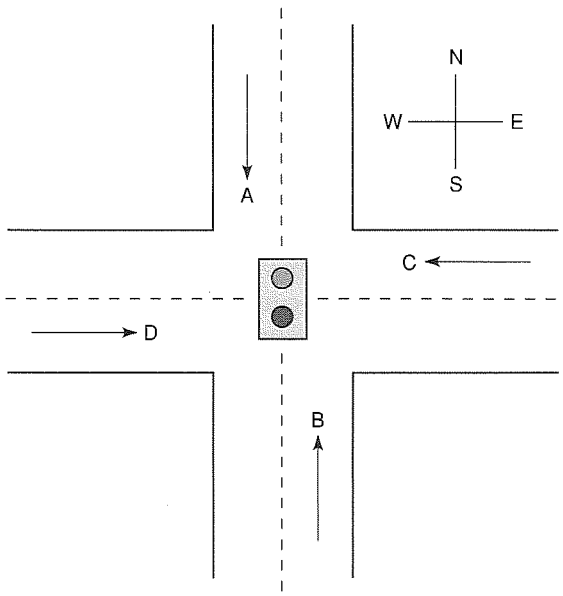


FIGURE 4-74 Problem 4-30.

outputs are LOW (0) when no vehicle is present and HIGH (1) when a vehicle is present. The intersection traffic light is to be controlled according to the following logic:

1. The east-west (E-W) traffic light will be green whenever *both* lanes *C* and *D* are occupied.
2. The E-W light will be green whenever *either* *C* or *D* is occupied but lanes *A* and *B* are not *both* occupied.
3. The north-south (N-S) light will be green whenever *both* lanes *A* and *B* are occupied but *C* and *D* are not *both* occupied.
4. The N-S light will also be green when *either* *A* or *B* is occupied while *C* and *D* are *both* vacant.
5. The E-W light will be green when *no* vehicles are present.

Using the sensor outputs *A*, *B*, *C*, and *D* as inputs, design a logic circuit to control the traffic light. There should be two outputs, N-S and E-W, that go HIGH when the corresponding light is to be *green*. Simplify the circuit as much as possible and show *all* steps.

SECTION 4-7

D 4-31. Redesign the parity generator and checker of Figure 4-25 to (a) operate using odd parity. (*Hint:* What is the relationship between an odd-parity bit and an even-parity bit for the same set of data bits?) (b) Operate on eight data bits.

SECTION 4-8

B 4-32. (a) Under what conditions will an OR gate allow a logic signal to pass through to its output unchanged?

(b) Repeat (a) for an AND gate.

(c) Repeat for a NAND gate.

B (d) Repeat for a NOR gate.

4-33.* (a) Can an INVERTER be used as an enable/disable circuit? Explain.

(b) Can an XOR gate be used as an enable/disable circuit? Explain.

D 4-34. Design a logic circuit that will allow input signal *A* to pass through to the output only when control input *B* is LOW while control input *C* is HIGH; otherwise, the output is LOW.

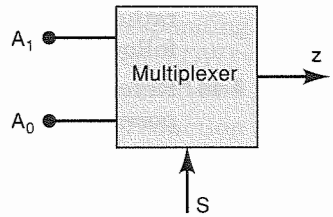
D 4-35.* Design a circuit that will *disable* the passage of an input signal only when control inputs *B*, *C*, and *D* are all HIGH; the output is to be HIGH in the disabled condition.

D 4-36. Design a logic circuit that controls the passage of a signal *A* according to the following requirements:

1. Output *X* will equal *A* when control inputs *B* and *C* are the same.
2. *X* will remain HIGH when *B* and *C* are different.

D 4-37. Design a logic circuit that has two signal inputs, A_1 and A_0, and a control input *S* so that it functions according to the requirements given in Figure 4-75. (This type of circuit is called a *multiplexer* and will be covered in Chapter 9.)

FIGURE 4-75 Problem 4-37.



S	z
0	$= A_0$
1	$= A_1$

D 4-38.* Use K mapping to design a circuit to meet the requirements of Example 4-17. Compare this circuit with the solution in Figure 4-23. This points out that the K-map method cannot take advantage of the XOR and XNOR gate logic. The designer must be able to determine when these gates are applicable.

SECTIONS 4-9 TO 4-13

T* 4-39. (a) A technician testing a logic circuit sees that the output of a particular INVERTER is stuck LOW while its input is pulsing. List as many possible reasons as you can for this faulty operation.

(b) Repeat part (a) for the case where the INVERTER output is stuck at an indeterminate logic level.

T 4-40.* The signals shown in Figure 4-76 are applied to the inputs of the circuit of Figure 4-32. Suppose that there is an internal open circuit at Z1-4.

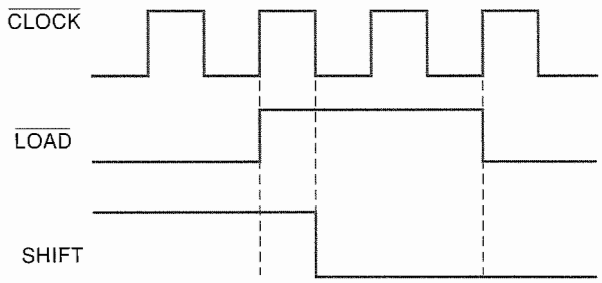
(a) What will a logic probe indicate at Z1-4?

(b) What dc voltage reading would you expect at Z1-4? (Remember that the ICs are TTL.)

(c) Sketch what you think the $\overline{CLKOUT}$ and $\overline{SHIFTOUT}$ signals will look like.

(d) Instead of the open at Z1-4, suppose that pins 9 and 10 of Z2 are internally shorted. Sketch the probable signals at Z2-10, $\overline{CLOCKOUT}$, and $\overline{SHIFTOUT}$.

FIGURE 4-76 Problem 4-40.



T 4-41. Assume that the ICs in Figure 4-32 are CMOS. Describe how the circuit operation would be affected by an open circuit in the conductor connecting Z2-2 and Z2-10.

T 4-42. In Example 4-24, we listed three possible faults for the situation of Figure 4-36. What procedure would you follow to determine which of the faults is the actual one?

T 4-43.* Refer to the circuit of Figure 4-38. Assume that the devices are CMOS. Also assume that the logic probe indication at Z2-3 is "indeterminate"

*Recall that T indicates a troubleshooting exercise.

rather than "pulsing." List the possible faults, and write a procedure to follow to determine the actual fault.

T 4-44.* Refer to the logic circuit of Figure 4-41. Recall that output Y is supposed to be HIGH for either of the following conditions:

1. $A = 1, B = 0$, regardless of C
2. $A = 0, B = 1, C = 1$

When testing the circuit, the technician observes that Y goes HIGH only for the first condition but stays LOW for all other input conditions. Consider the following list of possible faults. For each one, write yes or no to indicate whether or not it could be the actual fault. Explain your reasoning for each no response.

(a) An internal short to ground at Z2-13
(b) An open circuit in the connection to Z2-13
(c) An internal short to V_{CC} at Z2-11
(d) An open circuit in the V_{CC} connection to Z2
(e) An internal open circuit at Z2-9
(f) An open in the connection from Z2-11 to Z2-9
(g) A solder bridge between pins 6 and 7 of Z2

T 4-45. Develop a procedure for isolating the fault that is causing the malfunction described in Problem 4-44.

T 4-46.* Assume that the gates in Figure 4-41 are all CMOS. When the technician tests the circuit, he finds that it operates correctly except for the following conditions:

1. $A = 1, B = 0, C = 0$
2. $A = 0, B = 1, C = 1$

For these conditions, the logic probe indicates indeterminate levels at Z2-6, Z2-11, and Z2-8. What do you think is the probable fault in the circuit? Explain your reasoning.

T 4-47. Figure 4-77 is a combinational logic circuit that operates an alarm in a car whenever the driver and/or passenger seats are occupied and the seatbelts are not fastened when the car is started. The active-HIGH signals *DRIV* and *PASS* indicate the presence of the driver and passenger, respectively, and are taken from pressure-actuated switches in the seats. The signal *IGN* is active-HIGH when the ignition switch is on. The signal $\overline{BELTD}$ is active-LOW and indicates that the driver's seatbelt is

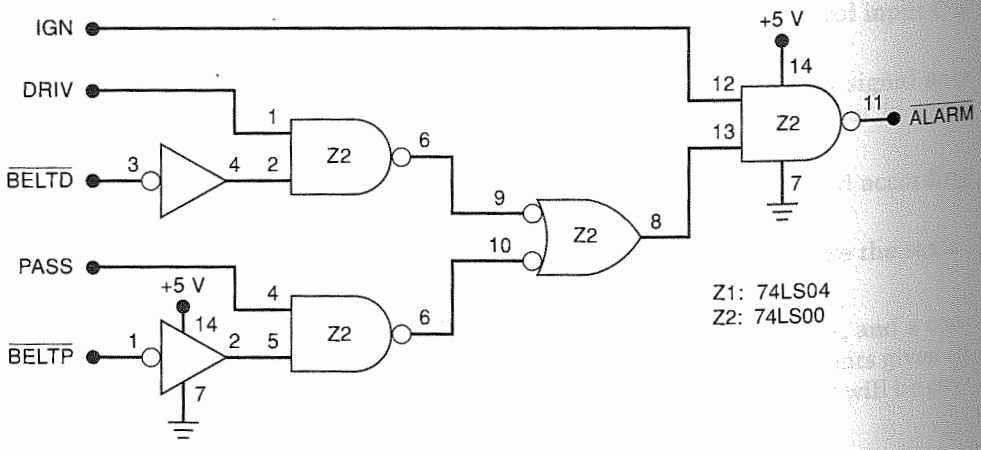


FIGURE 4-77 Problems 4-47, 4-48, and 4-49.

unfastened; $\overline{BELTP}$ is the corresponding signal for the passenger seatbelt. The alarm will be activated (LOW) whenever the car is started and either of the front seats is occupied and its seatbelt is not fastened.

(a) Verify that the circuit will function as described.

(b) Describe how this alarm system would operate if Z1-2 were internally shorted to ground.

(c) Describe how it would operate if there were an open connection from Z2-6 to Z2-10.

T 4-48.* Suppose that the system of Figure 4-77 is functioning so that the alarm is activated as soon as the driver and/or passenger are seated and the car is started, regardless of the status of the seatbelts. What are the possible faults? What procedure would you follow to find the actual fault?

T 4-49.* Suppose that the alarm system of Figure 4-77 is operating so that the alarm goes on continuously as soon as the car is started, regardless of the state of the other inputs. List the possible faults and write a procedure to isolate the fault.

DRILL QUESTIONS ON PLDs (50 THROUGH 55)

4-50.* *True or false:*

(a) Top-down design begins with an overall description of the entire system and it specifications.

(b) A JEDEC file can be used as the input file for a programmer.

(c) If an input file compiles with no errors, it means the PLD circuit will work correctly.

(d) A compiler can interpret code in spite of syntax errors.

(e) Test vectors are used to simulate and test a device.

H, B 4-51. What are the % characters used for in the AHDL design file?

H, B 4-52. How are comments indicated in a VHDL design file?

B 4-53. What is a ZIF socket?

B 4-54.* Name three entry modes used to input a circuit description into PLD development software.

B 4-55. What do JEDEC and HDL stand for?

SECTION 4-15

H, B 4-56. Declare the following data objects in AHDL or VHDL.

(a)* An array of eight output bits named *gadgets.*

(b) A single-output bit named *buzzer.*

(c) A 16-bit numeric input port named *altitude.*

(d) A single, intermediate bit within a hardware description file named *wire2.*

H, B 4-57. Express the following literal numbers in hex, binary, and decimal using the syntax of AHDL or VHDL.

(a)* 152_{10}

(b) 1001010100_2

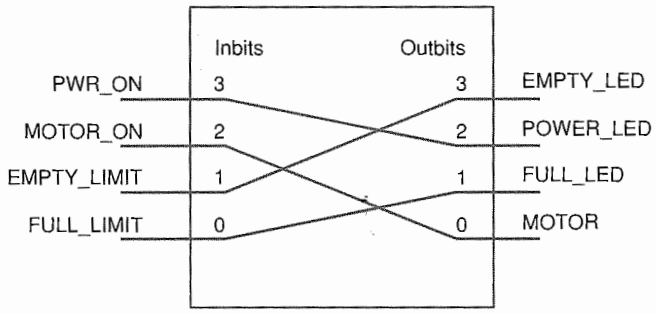
(c) $3C4_{16}$

H, B 4-58.* The following similar I/O definition is given for AHDL and VHDL. Write four concurrent assignment statements that will connect the inputs to the outputs as shown in Figure 4-78.

FIGURE 4-78 Problem 4-58.

```
SUBDESIGN hw
(
    inbits[3..0]    :INPUT;
    outbits[3..0]   :OUTPUT;
)
```

```
ENTITY hw IS
PORT (
    inbits      :IN BIT_VECTOR (3 downto 0);
    outbits     :OUT BIT_VECTOR (3 downto 0)
    );
END hw;
```



SECTION 4-16

H, D 4-59. Modify the AHDL truth table of Figure 4-50 to implement $AB + A\overline{C} + \overline{A}B$.

H, D 4-60.* Modify the AHDL design in Figure 4-54 so that $z = 1$ only when the digital value is less than 1010_2.

H, D 4-61. Modify the VHDL truth table of Figure 4-51 to implement $AB + A\overline{C} + \overline{A}B$.

H, D 4-62.* Modify the VHDL design in Figure 4-55 so that $z = 1$ only when the digital value is less than 1010_2.

H, B 4-63. Modify the code of (a) Figure 4-54 or (b) Figure 4-55 such that the output z is LOW only when digital_value is between 6 and 11 (inclusive).

H, D 4-64. Modify (a) the AHDL design in Figure 4-60 to implement Table 4-1. (b) the VHDL design in Figure 4-61 to implement Table 4-1.

H, D 4-65.* Write the hardware description design file Boolean equation to implement Example 4-9.

4-66. Write the hardware description design file Boolean equation to implement a four-bit parity generator as shown in Figure 4-25(a).

DRILL QUESTION

B 4-67. Define each of the following terms.

(a) Karnaugh map

(b) Sum-of-products form

(c) Parity generator

(d) Octet

(e) Enable circuit
(f) Don't-care condition
(g) Floating input
(h) Indeterminate voltage level
(i) Contention
(j) PLD
(k) TTL
(l) CMOS

MICROCOMPUTER APPLICATIONS

C 4-68. In a microcomputer, the microprocessor unit (MPU) is always communicating with one of the following: (1) random-access memory (RAM), which stores programs and data that can be readily changed; (2) read-only memory (ROM), which stores programs and data that never change; and (3) external input/output (I/O) devices such as keyboards, video displays, printers, and disk drives. As it is executing a program, the MPU will generate an address code that selects which type of device (RAM, ROM, or I/O) it wants to communicate with. Figure 4-79 shows a typical arrangement where the MPU outputs an eight-bit address code A_{15} through A_8. Actually, the MPU outputs a 16-bit address code, but the low-order bits A_7 through A_0 are not used in the device selection process. The address code is applied to a logic circuit that uses it to generate the device select signals: $\overline{RAM}$, $\overline{ROM}$, and $\overline{I/O}$.

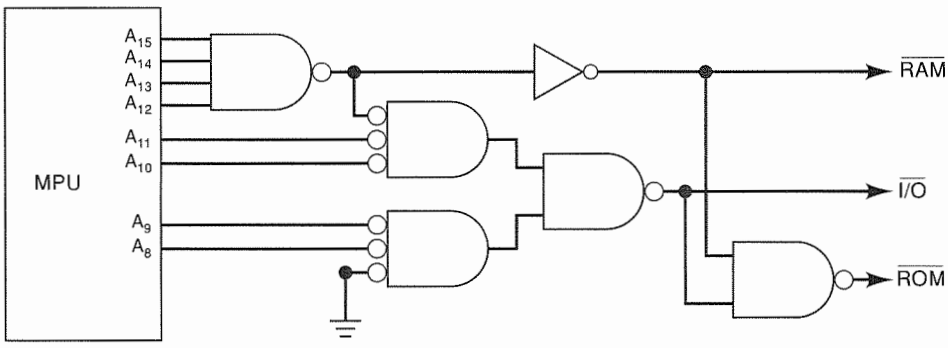


FIGURE 4-79 Problems 4-68 and 4-69.

Analyze this circuit and determine the following.

(a)*The range of addresses A_{15} through A_8 that will activate $\overline{RAM}$
(b) The range of addresses that activate $\overline{I/O}$
(c) The range of addresses that activate $\overline{ROM}$

Express the addresses in binary and hexadecimal. For example, the answer to (a) is A_{15} to A_8 = 00000000_2 to 11101111_2 = 00_{16} to EF_{16}.

C, D 4-69. In some microcomputers, the MPU can be *disabled* for short periods of time while another device controls the RAM, ROM, and I/O. During these intervals, a special control signal ($\overline{DMA}$) is activated by the MPU and is used to disable (deactivate) the device select logic so that the $\overline{RAM}$, $\overline{ROM}$, and $\overline{I/O}$ are all in their inactive state. Modify the circuit of Figure 4-79 so that $\overline{RAM}$, $\overline{ROM}$, and $\overline{I/O}$ will be deactivated whenever the signal $\overline{DMA}$ is active, regardless of the state of the address code.

ANSWERS TO SECTION REVIEW QUESTIONS

SECTION 4-1

1. Only (a) 2. Only (c)

SECTION 4-3

1. Expression (b) is not in sum-of-products form because of the inversion sign over both the C and D variables (i.e., the $A\overline{CD}$ term). Expression (c) is not in sum-of-products form because of the $(M + \overline{N})P$ term. 3. $x = \overline{A} + \overline{B} + \overline{C}$

SECTION 4-4

1. $x = \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}C\overline{D} + \overline{A}B\overline{C}\,\overline{D}$ 2. Eight: four inverters for A, B, C, and D plus four for SOP.

SECTION 4-5

1. $x = AB + AC + BC$ 2. $z = A + BCD$ 3. $S = \overline{P} + QR$ 4. An input condition for which there is no specific required output condition; i.e., we are free to make it 0 or 1.

SECTION 4-6

2. A constant LOW 3. No; the available XOR gate can be used as an INVERTER by connecting one of its inputs to a constant HIGH (see Example 4-16).

SECTION 4-8

1. $x = \overline{A(B \oplus C)}$ 2. OR, NAND 3. NAND, NOR

SECTION 4-9

1. (a) bipolar junction transistors (b) unipolar metal oxide semiconductor field effect transistors (MOSFETS). 2. SSI, MSI, LSI, VLSI, ULSI, GSI 3. True 4. True 5. 40, 74AC, 74ACT series 6. 0 to 0.8 V; 2.0 to 5.0 V 7. 0 to 1.5 V; 3.5 to 5.0 V 8. As if the input were HIGH 9. Unpredictably; it may overheat and be destroyed. 10. 74HCT and 74ACT 11. They describe exactly how to interconnect the chips for laying out the circuit and troubleshooting. 12. Inputs and outputs are defined, and logical relationships are described.

SECTION 4-11

1. Open inputs or outputs; inputs or outputs shorted to V_{CC}; inputs or outputs shorted to ground; pins shorted together; internal circuit failures 2. Pins shorted together 3. For TTL, a LOW; for CMOS, indeterminate 4. Two or more outputs connected together

SECTION 4-12

1. Open signal lines; shorted signal lines; faulty power supply; output loading
2. Broken wires; poor solder connections; cracks or cuts in PC board; bent or broken IC pins; faulty IC sockets 3. ICs operating erratically or not at all 4. Logic level indeterminate

SECTION 4-14

1. Electrically controlled connections are being programmed as open or closed.
2. (4, 1) (2, 2) or (2, 1) (4, 2) 3. (4, 5) (1, 6) or (4, 6) (1, 5)

SECTION 4-15

1. (a) push_buttons[5..0] :INPUT; (b) push_buttons :IN BIT_VECTOR (5 DOWNTO 0).
2. (a) z = push_buttons[5]; (b) z <= push_buttons(5); 3. STD_LOGIC
4. STD_LOGIC_VECTOR

SECTION 4-16

1. (AHDL) omega[] = (x, y, z); (VHDL) omega <= x & y & z; 2. Using the keyword TABLE 3. Using selected signal assignments

SECTION 4-17

1. IF/THEN 2. IF/THEN/ELSE 3. CASE or IF/ELSIF
4. (AHDL) count[7..0] :INPUT; (VHDL) count :IN INTEGER RANGE 0 TO 205

II. Programming with MATLAB

The content in this section is compiled from:

Mastering MATLAB
Hanselman & Littlefield

1

Getting Started

1.1 INTRODUCTION

This text assumes that you have some familiarity with matrices and computer programming. Matrices and arrays, in general, are at the heart of MATLAB, since all data in MATLAB are stored as arrays. Besides offering common matrix algebra operations, MATLAB offers array operations that allow you to quickly manipulate sets of data in a wide variety of ways. In addition to its matrix orientation, MATLAB offers programming features that are similar to those of other computer programming languages. Finally, MATLAB offers graphical user interface (GUI) tools that enable you to use MATLAB as an application-development tool. This combination of array data structures, programming features, and GUI tools makes MATLAB an extremely powerful tool for solving problems in many fields. In this text, each of these aspects of MATLAB is discussed in detail. To facilitate learning, detailed examples are presented.

1.2 TYPOGRAPHICAL CONVENTIONS

The following conventions are used throughout this book:

Bold italics	New terms or important facts
Boxed text	Important terms and facts
Bold Initial Caps	Keyboard key names, menu names, and menu items
`Constant width`	Computer user input, function and file names, commands, and screen displays

Hanselman, D. and Littlefield, B. (2012). Getting Started. In *Mastering MATLAB®*, (pp. 1-3). Upper Saddle River, NJ: Pearson Education Inc.

`Boxed constant width`	Contents of a script, function, or data file
`Constant width italics`	User input that is to be replaced and not taken literally, such as `» help` *`functionname`*
Italics	Window names, object names, book titles, toolbox names, company names, example text, and mathematical notations

1.3 WHAT'S NEW IN MATLAB

The previous edition of this text, *Mastering MATLAB 7*, was based on MATLAB version 7.0. Over time, MATLAB has evolved through a number of versions. As of Release 2011a of the MATLAB/Simulink software suite, MATLAB has progressed to version 7.12.

The latest version is yet another evolutionary change in MATLAB. The *Command* window remains the primary user interface, *Figure* windows are used to display graphical information and to create graphical user interfaces (GUIs), and a text editor/debugger is provided for writing, editing, and debugging MATLAB code. The *MATLAB desktop* coordinates the position and visibility of a number of other windows, such as *Workspace*, *Editor*, *Help*, and *Command History*.

Under the surface, several numerical and operational changes have been made to the software. In addition to having added or modified several internal numerical algorithms, MATLAB 7 and above supports mathematical operations on data types other than double-precision arrays, which have always been at the heart of MATLAB. Perhaps even more importantly, the MATLAB command interpreter now includes acceleration features, collectively called *The MATLAB JIT-accelerator*. This accelerator, which first appeared in MATLAB 6.5, dramatically increases the speed of loop operations by interpreting and executing code within a loop as a whole, rather than line by line, as was done previously. This acceleration eliminates the need to *vectorize* some code, which requires you to create and maximize the use of arrays and array mathematics to achieve optimal performance. To make use of the JIT-accelerator, loop operation code must follow specific guidelines, which are covered in this text. When these guidelines are not followed, loop operation code is interpreted at the much slower line-by-line rate.

In sum, version 7.12 represents an evolutionary change in MATLAB. The basic operation of MATLAB and its capabilities haven't changed in any dramatic way. In almost all cases, MATLAB code written for MATLAB 6.0–7.11 will run without change in MATLAB 7.12 and above. For the most part, the new and changed features of MATLAB can increase your productivity in solving problems with MATLAB.

1.4 WHAT'S IN *MASTERING MATLAB*

MATLAB documentation, in both hard copy and electronic format, exceeds 5000 pages of information and help text. Given this exhaustive documentation and help, *Mastering MATLAB* does not attempt to be a comprehensive tutorial or reference.

which just isn't possible in a single book containing less than 1000 pages. *Mastering MATLAB* does not even attempt to document all functions within the base product of MATLAB. This too is impossible, as there are more than 1500 operators and functions that are part of the base product of MATLAB. In light of this documentation set, the goals of *Mastering MATLAB* include the following: (1) introduce MATLAB to the novice user, (2) illustrate all key features and capabilities of MATLAB, and (3) demonstrate by example how to write efficient MATLAB code.

This text is intended to be a valuable resource when the MATLAB documentation set is unavailable. It follows the rule of providing 80 percent of the information needed in 20 percent of the space required to cover everything. The book is also intended to be a valuable resource when the MATLAB documentation set is available. In this case, it supplies numerous examples of efficient MATLAB coding that demonstrate how the many features of MATLAB come together to solve real problems.

Equally important is what's not included in *Mastering MATLAB*. The book does not discuss all of the windows, menus, menu items, submenus, and dialog boxes of the MATLAB user interface. While these user-interface aspects of MATLAB are very important, there is no room for them in this text. Moreover, they are difficult to cover effectively in a text and are best learned by hands-on exploration. Rather, this text focuses on the mathematical, programming, and graphical features that facilitate problem solution.

In addition to updating the second edition to reflect changes and additions to MATLAB since version 7.0, revisions include a major expansion of Chapter 32 ("Examples, Examples, Examples"), consolidation and revision of some of the other chapters, and dropping the specialized GUI and API (application programming interface) chapters. For those interested in building graphical user interfaces, the MATLAB online documentation is quite extensive. In addition to the MATLAB documentation, a number of specialized texts on integrating MATLAB with C, FORTRAN, Java, and Microsoft Windows applications are also available.

This book was written using information about MATLAB version 7.12. As MATLAB evolves, some things are bound to change. As a result, there may be isolated areas in this book where information about new features is missing and, worse yet, where information is incorrect. The authors have no control over MATLAB. We also cannot rewrite the text to reflect minor MATLAB releases. We are thankful that the makers of MATLAB are very careful when introducing new features and when changing old features. Historically, old features are grandfathered for one major release, and sometimes for additional releases. As a result, even though the book reflects MATLAB version 7.12, it will undoubtedly be useful for all future MATLAB versions as well.

To support this text, the authors maintain the Mastering MATLAB website at *http://www.masteringmatlab.com*. At the site, you can find errata for the text, as well as MATLAB script M-files for creating all of the figures in the book. The authors also encourage constructive feedback about the text at the e-mail address *masteringmatlab@gmail.com*.

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

2

Basic Features

Running MATLAB creates one or more windows on your computer monitor. One of these windows, entitled *MATLAB*, is commonly called the ***MATLAB desktop***. This window is the primary graphical user interface for MATLAB. Within the *MATLAB* window, there is a window called the ***Command*** window, which is the primary place wherein you interact with MATLAB. The prompt >> is displayed in the *Command* window, and when the *Command* window is active, a blinking cursor appears to the right of the prompt. This cursor and prompt signify that MATLAB is waiting to perform a mathematical operation.

2.1 SIMPLE MATH

Just like a calculator, MATLAB can do basic math. Consider the following simple example: Mary goes to the office supply store and buys four erasers at 25 cents each, six memo pads at 52 cents each, and two rolls of tape at 99 cents each. How many items did Mary buy, and how much did they cost?

To solve this problem with a calculator, you enter

```
4 + 6 + 2 = 12 items
4 × 25 + 6 × 52 + 2 × 99 = 610 cents
```

In MATLAB, this problem can be solved in a number of different ways. First, the calculator approach can be taken:

```
>> 4+6+2
ans =
    12
```


Hanselman, D. and Littlefield, B. (2012). Basic Features. In *Mastering MATLAB®*, (pp. 4-21). Upper Saddle River, NJ: Pearson Education Inc.

```
>> 4*25 + 6*52 + 2*99
ans =
    610
```

Note that MATLAB doesn't care about spaces, for the most part, and that multiplication takes precedence over addition. Note also that MATLAB calls the result `ans`, which is short for *answer* for both computations.

As an alternative, the problem can be solved by storing information in ***MATLAB variables***:

```
>> erasers = 4
erasers =
     4
>> pads = 6
pads =
     6
>> tape = 2;
>> items = erasers + pads + tape
items =
    12
>> cost = erasers*25 + pads*52 + tape*99
cost =
   610
```

Here, we created three MATLAB variables—`erasers`, `pads`, and `tape`—to store the number of each item. After entering each statement, MATLAB displayed the results, except in the case of `tape`. The semicolon at the end of the line tells MATLAB to evaluate the line, but not to display the answer. Finally, rather than calling the results `ans`, we told MATLAB to call the number of items purchased `items` and the total price paid `cost`. At each step, MATLAB remembered past information. Because MATLAB remembers things, let's ask what the average cost per item was:

```
>> average_cost = cost/items
average_cost =
    50.833
```

Since the term *average cost* is two words and MATLAB variable names must be one word, an underscore was used to create the single MATLAB variable `average_cost`.

In all, MATLAB offers the following basic arithmetic operations:

Operation	Symbol	Example
Addition	+	3 + 22
Subtraction	–	54.4 – 16.5
Multiplication	*	3.14 * 6
Division	/ or \	19.54/7 or 7\19.54
Exponentiation	^	2^8

The order in which these operations are evaluated in a given expression is determined by the usual rules of precedence, summarized as follows:

> Expressions are evaluated from left to right, with the exponentiation operation having the highest precedence, followed by multiplication and division, having equal precedence, and then by addition and subtraction, having equal precedence.
>
> Parentheses can be used to alter this ordering, in which case these rules of precedence are applied within each set of parentheses, by starting with the innermost set and proceeding outward.

More information about precedence rules can be displayed by typing `help precedence` at the MATLAB prompt.

2.2 THE MATLAB WORKSPACE

As you work in the *Command* window, MATLAB remembers the commands you enter, as well as the values of any variables you create. These commands and variables are said to reside in the ***MATLAB workspace*** or ***base workspace***, and they can be recalled whenever you wish. For example, to check the value of `tape`, all you have to do is ask MATLAB for it by entering its name at the prompt:

```
>> tape
tape =
     2
```

If you can't remember the name of a variable, you can ask MATLAB for a list of the variables it knows by using the MATLAB command `who`:

```
>> who
Your variables are:
cost          items    tape
average_cost  erasers  pads
```

Note that MATLAB doesn't tell you the value of all of the variables; it merely gives you their names. To find their values, you must enter their names individually at the MATLAB prompt.

To recall previous commands, MATLAB responds to the **Cursor** keys on your keyboard. For example, pressing the ↑ key once recalls the most recent command to the MATLAB prompt. Repeatedly pressing ↑ scrolls back through prior commands, one at a time. In a similar manner, pressing the ↓ key scrolls forward through commands. By pressing the → or ← keys, you can maneuver within a given command at the MATLAB prompt, thereby editing the command in much the same way that you edit text in a word-processing program. Other standard editing keys, such as **Delete** or **Backspace**, **Home**, and **End**, perform their commonly assigned tasks. The **Tab** key is useful for variable-name completion. Once a scrolled command is acceptable, pressing the **Return** key with the cursor *anywhere* in the command tells MATLAB to process it. Finally, and perhaps most useful, the **Escape** key erases the current command at the prompt. For those of you familiar with the EMACS editor, MATLAB also accepts common EMACS editing control-character sequences, such as **Control-U** to erase the current command.

2.3 ABOUT VARIABLES

Like any other computer language, MATLAB has rules about variable names. Earlier, it was noted that variable names must be a single word containing no spaces. More specifically, MATLAB variable-naming rules are listed as follows.

Variable-Naming Rules	Comments/Examples
Variable names are case sensitive.	`Cost`, `cost`, `CoSt`, and `COST` are all different MATLAB variables.
Variable names can contain up to `namelengthmax` characters (63 as of version 7.12). Any characters beyond the 63rd are ignored.	`Howabouthisvariablename`
Variable names must start with a letter, followed by any number of letters, digits, or underscores.	`how_about_this_variable_nameX51483`
Punctuation characters are not allowed, because many of them have special meanings in MATLAB.	`a_b_c_d_e`

There are some specific exceptions to these naming rules. MATLAB has several names that cannot be used for variables. These names are ***keywords*** and form a ***reserved word list*** for MATLAB:

Reserved Word List
`for end if while function return elseif case otherwise classdef switch continue else try catch global persistent break parfor spmd`

This list is returned as an output of the `iskeyword` function. MATLAB will report an error if you try to use a reserved word as a variable. However, you can use words similar to keywords by capitalizing one or more letters. The function `isvarname('teststring')` returns True (1) if the character-string argument `'teststring'` is a valid variable name; otherwise, it returns False (0).

In addition, just as your calculator stores constants such as π, MATLAB has a number of special variables:

Special Variables	Description
`ans`	Default variable name used for results
`beep`	Makes computer sound a beep
`computer`	Computer type
`version`	MATLAB version string
`ver`	MATLAB, computer OS, and Java version information
`pi`	Ratio of the circumference of a circle to its diameter
`eps`	Smallest number that, when added to 1, creates a number greater than 1 on the computer
`inf`	Stands for infinity (e.g., `1/0`)
`NaN` or `nan`	Stands for Not-a-Number (e.g., `0/0`)
`i` or `j`	Stands for $\sqrt{-1}$
`nargin`	Number of function input arguments
`nargout`	Number of function output arguments
`intmin`	Largest usable negative integer
`intmax`	Largest usable positive integer
`realmin`	Smallest usable positive real number
`realmax`	Largest usable positive real number
`bitmax`	Largest usable positive integer stored in double-precision format
`varargin`	Variable number of function input arguments
`varargout`	Variable number of function output arguments

A publication from Pearson Custom Publishing exclusively for The University of Melbourne

If you reuse a variable (such as tape in the earlier example) or assign a value to one of the special variables from the preceding list, the prior value of the variable is overwritten and lost. However, any other expressions computed with the prior values do not change. Consider the following example:

```
>> erasers = 4;
>> pads = 6;
>> tape = 2;
>> items = erasers + pads + tape
items =
    12
>> erasers = 6
erasers =
     6
>> items
items =
    12
```

Here, using the first example again, we found the number of items Mary purchased. Afterward, we changed the number of erasers to six, overwriting its prior value of 4. In doing so, the value of items has not changed. Unlike a common spreadsheet program, MATLAB does not recalculate the number of items based on the new value of erasers. ***When MATLAB performs a calculation, it does so using the values that it knows at the time the requested command is evaluated.*** In the preceding example, if you wish to recalculate the number of items, the total cost, and the average cost, it is necessary to recall the appropriate MATLAB commands and ask MATLAB to evaluate them again.

The special variables given earlier follow this guideline, also, with the exception of the fact that the special values can be restored. When you start MATLAB, the variables have their original values; when you change their values, the original special values are lost. To restore a special value, all you have to do is ***clear*** the overwritten value. For example,

```
>> pi
ans =
      3.1416
>> pi = 1.23e-4
pi =
      0.000123
```

Hanselman, D. and Littlefield, B. (2012). Basic Features. In *Mastering MATLAB®*, (pp. 4-21). Upper Saddle River, NJ: Pearson Education Inc.

```
>> clear pi
>> pi
ans =
      3.1416
```

shows that `pi` has the special value of `3.1416`, to five significant digits; is overwritten with the value of `1.23e-4`; and then, after being cleared using the `clear` function has its original special value once again.

2.4 COMMENTS, PUNCTUATION, AND ABORTING EXECUTION

As we saw earlier, placing a semicolon at the end of a command suppresses printing of the computed results. This feature is especially useful for suppressing the results of intermediate calculations. For instance,

```
>> erasers
erasers =
     6
>> items = erasers + pads + tape;
>> cost = erasers*25 + pads*52 + tape*99;
>> average_cost = cost/items
average_cost =
     47.143
```

displays the average cost of the items that Mary bought when she purchased six erasers, rather than the original four. The intermediate results `items` and `cost` were not printed, because semicolons appear at the ends of the commands defining them.

In addition to semicolons, MATLAB uses other punctuation symbols. For example, all text after a percent sign (%) is taken as a comment statement:

```
>> tape = 2 % number of rolls of tape purchased
```

The variable `tape` is given the value of 2, and MATLAB simply ignores the percent sign and all text following it.

If they are separated by commas or semicolons, multiple commands can be placed on one line:

```
>> erasers = 6, pads = 6; tape = 2
erasers =
     6
tape =
     2
```

Commas tell MATLAB to display results; semicolons suppress printing.

Sometimes, expressions or commands are so long that it is convenient to continue them onto additional lines. In MATLAB, statement continuation is denoted by three periods in succession, as shown in the following code:

```
>> average_cost = cost/items % command as done earlier
average_cost =
       47.143
>> average_cost = cost/...% command with valid continuation
items
average_cost =
       47.143
>> average_cost = cost...% command with valid continuation
/items
average_cost =
       47.143
>> average_cost = cost...command with valid continuation (no % needed)
/items
average_cost =
       47.143
>> average_cost = cost/it...% command with INvalid continuation
ems
??? ems
    |
Error: Missing MATLAB operator.
```

Note that statement continuation works if the three periods appear between variable names and mathematical operators, but not in the middle of a variable name. That is, variable names cannot be split between two lines. Furthermore, all text after the three periods is considered to be a comment, so no percent symbol is needed. In addition, since comment lines are ignored, they cannot be continued either, as the following example shows:

```
>> % Comments cannot be continued...
>> either
??? Undefined function or variable 'either'.
```

In this case, the . . . in the comment line is part of the comment and is not processed by MATLAB.

Finally, MATLAB processing can be interrupted at any time by pressing **Control-C** (i.e., pressing the **Ctrl** and **C** keys simultaneously).

2.5 COMPLEX NUMBERS

One of the most powerful features of MATLAB is that it does not require any special handling for complex numbers. Complex numbers are formed in MATLAB in several ways. Examples of complex numbers include the following:

```
>> c1 = 1-2i % the appended i signifies the imaginary part
c1 =
   1.0000 - 2.0000i
>> c1 = 1-2j % j also works
c1 =
   1.0000 - 2.0000i
>> c1 = complex(1,-2) % a function that creates complex numbers
c1 =
   1.0000 - 2.0000i
>> c2 = 3*(2-sqrt(-1)*3)
c2 =
   6.0000 - 9.0000i
>> c3 = sqrt(-2)
c3 =
        0 + 1.4142i
>> c4 = 6+sin(.5)*1i
c4 =
   6.0000 + 0.4794i
>> c5 = 6+sin(.5)*1j
c5 =
   6.0000 + 0.4794i
```

In the last two examples, multiplication by `1i` and `1j` are used to form the imaginary part. Multiplication by `1i` or `1j` is required in these cases, since `sin(.5)i` and `sin(.5)j` have no meaning in MATLAB. Termination with the characters `i` and `j`, as shown in the first two examples above, works only with numbers, and not with expressions.

Some programming languages require special handling for complex numbers, wherever they appear. In MATLAB, no special handling is required. Mathematical operations on complex numbers are written the same way as those on real numbers:

```
>> c6 = (c1+c2)/c3 % from the above data
c6 =
   -7.7782 - 4.9497i
>> c6r = real(c6)
c6r =
     -7.7782
>> c6i = imag(c6)
c6i =
     -4.9497
>> check_it_out = 1i^2 % sqrt(-1) squared must be -1!
check_it_out =
   -1
```

In general, operations on complex numbers lead to complex number results. However, in the last case, MATLAB is smart enough to drop the zero imaginary part of the result. In addition, the foregoing shows that the functions `real` and `imag` extract the real and imaginary parts of a complex number, respectively.

As a final example of complex arithmetic, consider the Euler identity that relates the polar form of a complex number to its rectangular form $M\angle\theta = Me^{j\theta} = a + bj$, where the polar form is given by a magnitude M and an angle θ, and the rectangular form is given by $a + bj$. The relationships among these forms are $M = \sqrt{a^2 + b^2}$, $\theta = \tan^{-1}(b/a)$, $a = M\cos(\theta)$, $b = M\sin(\theta)$.

In MATLAB, the conversion between polar and rectangular forms makes use of the functions `real`, `imag`, `abs`, and `angle`:

```
>> c1
c1 =
   1.0000 - 2.0000i
>> mag_c1 = abs(c1) % magnitude
mag_c1 =
     2.2361
>> angle_c1 = angle(c1) % angle in radians
angle_c1 =
   -1.1071
>> deg_c1 = angle_c1*180/pi % angle in degrees
```

```
deg_c1 =
   -63.4349
>> real_c1 = real(c1) % real part
real_c1 =
    1
>> imag_c1 = imag(c1) % imaginary part
imag_c1 =
   -2
```

The MATLAB function abs computes the magnitude of complex numbers or the absolute value of real numbers, depending on which one you assign it to compute. Likewise, the MATLAB function angle computes the angle of a complex number in radians. ***MATLAB does not natively perform trigonometric operations with units of degrees; however, basic trigonometric functions supporting angles in degrees are provided in MATLAB.***

2.6 FLOATING-POINT ARITHMETIC

In almost all cases, numerical values in MATLAB are represented in double-precision arithmetic using a binary (base 2) representation internally. It is the most common representation used by computers and is a native format for numerical coprocessors. Because of this representation, not all numbers can be represented exactly. There are limiting values that can be represented, and there is a recognizable lower limit for addition.

The largest positive real number that can be represented is

```
>> format long % tell MATLAB to display more precision
>> realmax
ans =
   1.797693134862316e+308
```

The smallest positive number that can be represented is

```
>> realmin
ans =
   2.225073858507201e-308
```

The smallest number that can be added to 1 to produce a number larger than 1 in double precision is

```
>> eps
ans =
   2.220446049250313e-16
```

Generalizing this concept, `eps(x)` is the smallest increment that can be added to `x` to produce a number larger than `x`, as the following examples show:

```
> eps(1) % same as eps by itself
ans =
   2.220446049250313e-16
>> eps(10)
ans =
   1.776356839400251e-15
>> eps(1e10)
ans =
   1.907348632812500e-06
```

As the magnitude of a number increases, the distance between values that can be represented in finite precision (which is what `eps(x)` returns) increases as well.

The consequences of the limitations of finite-precision arithmetic are sometimes strange. For example, as shown here, addition is not *exactly* commutative:

```
>> 0.42 - 0.5 + 0.08
ans =
   -1.387778780781446e-17
>> 0.08 - 0.5 + 0.42 % rearrange order
ans =
    0
>> 0.08 + 0.42 - 0.5 % rearrange order again
ans =
    0
```

All three of these results should be zero, but they are not. In each case, the arithmetic was performed from left to right. The issue here is that not all of the numbers can be represented exactly in double-precision arithmetic. In fact, only `0.5` has an exact representation. When numbers cannot be represented exactly, they are approximated with as much precision as possible—leading to inevitable errors in computed results. For the most part, these errors are minor; otherwise, double-precision arithmetic wouldn't be used in modern computers. In practice, the problems with double-precision

arithmetic occur most often when asking MATLAB to compare two numbers for equality or inequality. Clearly, within MATLAB, 0.42 − 0.5 + 0.08 does not equal 0.08 − 0.5 + 0.42, even though our brains can do the exact arithmetic to show that it does.

A second drawback of finite-precision arithmetic appears in function evaluation. Not only is finite-precision arithmetic unable to always represent function arguments exactly, but also most functions cannot themselves be represented exactly, as the following code shows:

```
>> sin(0)
ans =
    0
>> sin(pi)
ans =
   1.224646799147353e-16
```

Here, both results should be zero, but sin(π) is not. It is interesting to note that the error here and the error in the previous example are both less than eps.

Finally, when using double-precision, floating-point arithmetic to represent integers, only integers up to a limit can be represented *exactly*. The limiting value is $2^{53} - 1$, which is represented by MATLAB as follows:

```
>> bitmax
ans =
   9.007199254740991e+15
```

2.7 MATHEMATICAL FUNCTIONS

Lists of the common functions that MATLAB support are shown in the tables at the end of this chapter. Most of these functions are used in the same way you would write them mathematically:

```
>> x = sqrt(2)/2
x =
   0.7071
>> y = asin(x)
y =
   0.7854
>> y_deg = y*180/pi
y_deg =
   45.0000
```

These commands find the angle where the sine function has a value of $\sqrt{2}/2$. Note, again, that MATLAB uses radians, not degrees, in trigonometric functions. Other examples include the following:

```
>> y = sqrt(3^2 + 4^2) % show 3-4-5 right triangle relationship
y =
   5
>> y = rem(23,4) % remainder function, 23/4 has a remainder of 3
y =
   3
>> x = 2.6, y1 = fix(x), y2 = floor(x), y3 = ceil(x), y4 = round(x)
x =
   2.6000
y1 =
   2
y2 =
   2
y3 =
   3
y4 =
   3
```

Trigonometric Function	Description
acos	Inverse cosine returning radians
acosd	Inverse cosine returning degrees
acosh	Inverse hyperbolic cosine returning radians
acot	Inverse cotangent returning radians
acotd	Inverse cotangent returning degrees
acoth	Inverse hyperbolic cotangent returning radians
acsc	Inverse cosecant returning radians
acscd	Inverse cosecant returning degrees
acsch	Inverse hyperbolic cosecant returning radians
asec	Inverse secant returning radians
asecd	Inverse secant returning degrees

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

Trigonometric Function	Description
asech	Inverse hyperbolic secant returning radians
asin	Inverse sine returning radians
asind	Inverse sine returning degrees
asinh	Inverse hyperbolic sine returning radians
atan	Inverse tangent returning radians
atand	Inverse tangent returning degrees
atanh	Inverse hyperbolic tangent returning radians
atan2	Four-quadrant inverse tangent returning radians
cos	Cosine returning radians
cosd	Cosine of argument in degrees
cosh	Hyperbolic cosine returning radians
cot	Cotangent returning radians
cotd	Cotangent of argument in degrees
coth	Hyperbolic cotangent returning radians
csc	Cosecant returning radians
cscd	Cosecant of argument in degrees
csch	Hyperbolic cosecant returning radians
hypot	Square root of sum of squares
sec	Secant returning radians
secd	Secant of argument in degrees
sech	Hyperbolic secant returning radians
sin	Sine returning radians
sind	Sine returning degrees
sinh	Hyperbolic sine returning radians
tan	Tangent returning radians
tand	Tangent returning degrees
tanh	Hyperbolic tangent returning radians

ENGINEERING SYSTEMS DESIGN 2
Publication from Pearson Custom Publishing exclusively for The University of Melbourne

Exponential Function	Description
`^`	Power
`exp`	Exponential
`expm1`	Exponential minus 1 [i.e., $\exp(x) - 1$]
`log`	Natural logarithm
`log10`	Base 10 logarithm
`log1p`	Natural logarithm of $x + 1$ [i.e., $\log(x + 1)$]
`log2`	Base 2 logarithm and floating-point number dissection
`nthroot`	n^{th} real root of real numbers
`pow2`	Base 2 power and floating-point number scaling
`reallog`	Natural logarithm limited to real nonnegative values
`realpow`	Power limited to real-valued arguments
`realsqrt`	Square root limited to real-valued values
`sqrt`	Square root
`nextpow2`	Next higher power of 2

Complex Function	Description
`abs`	Absolute value or magnitude
`angle`	Phase angle in radians
`conj`	Complex conjugate
`imag`	Imaginary part
`real`	Real part
`unwrap`	Unwraps phase angle
`isreal`	True for real values
`cplxpair`	Sorts vector into complex conjugate pairs
`complex`	Forms complex number from real and imaginary parts
`sign`	Signum function

Hanselman, D. and Littlefield, B. (2012). Basic Features. In *Mastering MATLAB®*, (pp. 4-21). Upper Saddle River, NJ: Pearson Education Inc.

Rounding and Remainder Function	Description
`fix`	Rounds toward zero
`floor`	Rounds toward negative infinity
`ceil`	Rounds toward positive infinity
`round`	Rounds toward nearest integer
`mod`	Modulus or signed remainder
`rem`	Remainder after division
`idivide`	Integer division with rounding option
`sign`	Signum function

Coordinate Transformation Function	Description
`cart2sph`	Cartesian to spherical
`cart2pol`	Cartesian to cylindrical or polar
`pol2cart`	Cylindrical or polar to Cartesian
`sph2cart`	Spherical to Cartesian

Number Theoretic Function	Description
`factor`	Prime factors
`isprime`	True for prime numbers
`primes`	Generates list of prime numbers
`gcd`	Greatest common divisor
`lcm`	Least common multiple
`rat`	Rational approximation
`rats`	Rational output
`perms`	All possible combinations
`nchoosek`	All combinations of N elements taken K at a time
`factorial`	Factorial function

Specialized Function	Description
airy	Airy function
besselj	Bessel function of the first kind
bessely	Bessel function of the second kind
besselh	Bessel function of the third kind
besseli	Modified Bessel function of the first kind
besselk	Modified Bessel function of the second kind
beta	Beta function
betainc	Incomplete beta function
betaincinv	Inverse incomplete beta function
betaln	Logarithm of beta function
ellipj	Jacobi elliptic function
ellipke	Complete elliptic integral
erf	Error function
erfc	Complementary error function
erfcinv	Inverse complementary error function
erfcx	Scaled complementary error function
erfinv	Inverse error function
expint	Exponential error function
gamma	Gamma function
gammainc	Incomplete gamma function
gammaincinv	Inverse incomplete gamma function
gammaln	Logarithm of gamma function
legendre	Associated Legendre functions
psi	Psi (polygamma) function
cross	Vector cross product
dot	Vector dot product

ENGR10003 – ENGINEERING A
A publication from Pearson Custom Publishing exclusively for the U

3

The MATLAB Desktop

As stated in the preceding chapter, running MATLAB creates one or more windows on your computer monitor. One of these windows, entitled *MATLAB*, is commonly called the *MATLAB* desktop. This window contains or manages all other windows that are part of MATLAB. Depending on how you set up MATLAB, some windows associated with the desktop may or may not be visible, and some may or may not reside within (i.e., be ***docked*** in) the *MATLAB* window. Management of the *MATLAB* desktop window and its associated windows is not discussed in this text. If you are familiar with other window-based programs, manipulation of windows in MATLAB will be familiar to you. The *MATLAB* desktop menu items change, depending on which window is active. In addition, there are many helpful contextual menus (which are accessed by pressing the right mouse button over an item). If you are not familiar with window-based programs, it may be beneficial to seek general assistance in this area from other sources. In any case, to find out which windows are associated with the desktop, investigate the items on the **Desktop** menu within the *MATLAB* desktop window.

3.1 MATLAB WINDOWS

The windows used in MATLAB include the (1) *Command*, (2) *Command History*, (3) *Current Folder* (browser), (4) *Workspace* (browser), (5) *Help* (browser), (6) *Profiler*, (7) *File Exchange*, (8) *Editor*, (9) *Figures*, (10) *Web Browser*, (11) *Variable Editor*, and (12) *Comparison Tool*. The following table gives an overview of the purpose of each of these windows:

Hanselman, D. and Littlefield, B. (2012). The MATLAB Desktop. In *Mastering MATLAB®*, (pp. 22-30). Upper Saddle River, NJ: Pearson Education Inc.

exclusively for The University of Melbourne

Window	Description
Command	Issues commands to MATLAB for processing
Command History	Running history of prior commands issued in the *Command* window
Current Folder	GUI for directory and file manipulation in MATLAB
Workspace	GUI for viewing, editing, loading, and saving MATLAB variables
Help	GUI for finding and viewing online documentation
Profiler	Tool for optimizing M-file performance
File Exchange	GUI for accessing the MATLAB File Exchange
Editor	Text editor for creating, editing, and debugging M-files
Figures	Creates, views, and modifies plots and other figures
Web Browser	MATLAB HTML viewer
Variable Editor	Tool for viewing and editing arrays in table format
Comparison Tool	GUI for comparing text files line by line

3.2 MANAGING THE MATLAB WORKSPACE

Within the *MATLAB* desktop, actions taken in all windows support computations performed in the *Command* window. As a result, the rest of this chapter provides more detailed information about the *Command* window.

The data and variables created in the *Command* window reside in what is called the ***MATLAB workspace*** or ***base workspace***. In addition to viewing variables in the *Workspace* window, you can see what variable names exist in the MATLAB workspace by issuing the command who:

```
>> who
Your variables are:
angle_c1        c4            cost      pads
ans             c5            deg_c1    real_c1
average_cost    c6            erasers   tape
c1              c6i           imag_c1
c2              c6r           items
c3              check_it_out  mag_c1
```

Hanselman, D. and Littlefield, B. (2012). The MATLAB Desktop. In *Mastering MATLAB®*, (pp. 22-30). Upper Saddle River, NJ: Pearson Education Inc.

The variables you see may differ from those just listed, depending on what you've asked MATLAB to do since you opened the program. For more detailed information, use the command `whos`:

```
>> whos
  Name            Size      Bytes      Class       Attributes
  angle_c1        1x1           8      double
  ans             1x1           8      double
  average_cost    1x1           8      double
  c1              1x1          16      double      (complex)
  c2              1x1          16      double      (complex)
  c3              1x1          16      double      (complex)
  c4              1x1          16      double      (complex)
  c5              1x1          16      double      (complex)
  c6              1x1          16      double      (complex)
  c6i             1x1           8      double
  c6r             1x1           8      double
  check_it_out    1x1           8      double
  cost            1x1           8      double
  deg_c1          1x1           8      double
  erasers         1x1           8      double
  imag_c1         1x1           8      double
  items           1x1           8      double
  mag_c1          1x1           8      double
  pads            1x1           8      double
  real_c1         1x1           8      double
  tape            1x1           8      double
```

Here, each variable is listed, along with its size, the number of bytes used, its class, and other attributes, if any. Since MATLAB is array oriented, all of the variables belong to the class of double-precision arrays, even though all of the variables are scalars. Later, as other data types or classes are introduced, the information in this last column will become more useful. The preceding list is also displayed in the *Workspace* window, which can be viewed by typing `workspace` at the MATLAB prompt or by choosing the **Workspace** menu item on the **Desktop** menu of the *MATLAB* desktop.

The command `clear` deletes variables from the MATLAB workspace. For example,

```
>> clear real_c1 imag_c1 c*
>> who
Your variables are:
angle_c1        deg_c1      mag_c1
ans             erasers     pads
average_cost    items       tape
```

deletes the variables `real_c1`, `imag_c1`, and all variables starting with the letter c. Other options for the `clear` function can be identified by asking for information with the command `help` or `helpwin`:

```
>> help clear
 CLEAR Clear variables and functions from memory.
    CLEAR removes all variables from the workspace.
    CLEAR VARIABLES does the same thing.
    CLEAR GLOBAL removes all global variables.
    CLEAR FUNCTIONS removes all compiled M- and MEX-functions.

    CLEAR ALL removes all variables, globals, functions and MEX links.
    CLEAR ALL at the command prompt also removes the Java packages import
    list.

    CLEAR IMPORT removes the Java packages import list at the command
    prompt. It cannot be used in a function.

    CLEAR CLASSES is the same as CLEAR ALL except that class definitions
    are also cleared. If any objects exist outside the workspace (say in
    userdata or persistent in a locked m-file) a warning will be issued and
    the class definition will not be cleared. CLEAR CLASSES must be used if
    the number or names of fields in a class are changed.

    CLEAR JAVA is the same as CLEAR ALL except that java classes on the
    dynamic java path (defined using JAVACLASSPATH) are also cleared.

    CLEAR VAR1 VAR2 ... clears the variables specified. The wildcard
    character '*' can be used to clear variables that match a pattern. For
```

ENGR10003 – ENGINEERING SYSTEMS DESIGN 1
A publication from Pearson Custom Publishing exclusively for the University of Melbourne

instance, CLEAR X* clears all the variables in the current workspace that start with X.

CLEAR -REGEXP PAT1 PAT2 can be used to match all patterns using regular expressions. This option only clears variables. For more information on using regular expressions, type "doc regexp" at the command prompt.

If X is global, CLEAR X removes X from the current workspace, but leaves it accessible to any functions declaring it global.
CLEAR GLOBAL X completely removes the global variable X.
CLEAR GLOBAL -REGEXP PAT removes global variables that match regular expression patterns.
Note that to clear specific global variables, the GLOBAL option must come first. Otherwise, all global variables will be cleared.

CLEAR FUN clears the function specified. If FUN has been locked by MLOCK it will remain in memory. Use a partial path (see PARTIALPATH) to distinguish between different overloaded versions of FUN. For instance, 'clear inline/display' clears only the INLINE method for DISPLAY, leaving any other implementations in memory.

CLEAR ALL, CLEAR FUN, or CLEAR FUNCTIONS also have the side effect of removing debugging breakpoints and reinitializing persistent variables since the breakpoints for a function and persistent variables are cleared whenever the m-file changes or is cleared.

Use the functional form of CLEAR, such as CLEAR('name'), when the variable name or function name is stored in a string.

Examples for pattern matching:

```
    clear a*                % Clear variables starting with "a"
    clear -regexp ^b\d{3}$  % Clear variables starting with "b" and
                            %    followed by 3 digits
    clear -regexp \d        % Clear variables containing any digits
```

See also clearvars, who, whos, mlock, munlock, persistent.

ENGINE ... exclusively for The University of Melbourne


```
Reference page in Help browser
doc clear
```

Obviously, the `clear` command does more than just delete variables. Its other uses will become apparent as you become familiar with more of MATLAB's features.

3.3 MEMORY MANAGEMENT

MATLAB allocates memory for variables as they are created and for M-file functions as they are used. Depending on the computer on which the program is installed, it is possible for MATLAB to run out of memory, making it impossible to do any further work. When you eliminate variables by using the `clear` command, MATLAB frees up the memory used by the variables. Over time, however, it is possible for memory to become fragmented, leaving MATLAB's memory space populated by variables surrounded by numerous small fragments of free memory. Since MATLAB always stores variables in contiguous chunks of memory, these fragments of free memory may not be reusable. To alleviate this problem, the `pack` command performs memory garbage collection. The command saves all MATLAB workspace variables to disk, clears all variables from the workspace, and then reloads the variables back into the workspace. On completion, all fragments of free memory are consolidated into one large, usable block. Depending on how much memory is allocated to MATLAB on your computer, how long you've been running a particular MATLAB session, and how many variables you've created, you may or may not ever need to use the `pack` command.

3.4 NUMBER DISPLAY FORMATS

When MATLAB displays numerical results, it follows several rules. By default, if a result is an integer, MATLAB displays it as an integer. Likewise, when a result is a real number, MATLAB displays it with approximately four digits to the right of the decimal point. If the significant digits in the result are outside of this range, MATLAB displays the result in scientific notation, similar to the display of a scientific calculator. You can override this default behavior by specifying a different numerical format. From the *Command* window **File** menu, choose the **Preferences** menu. Alternatively, type the appropriate MATLAB `format` command at the prompt. With the special variable `pi`, the numerical display formats produced by different `format` selections are as follows:

MATLAB Command	`pi`	**Comments**
`format short`	`3.1416`	5 digits
`format long`	`3.141592653589793`	16 digits

Hanselman, D. and Littlefield, B. (2012). The MATLAB Desktop. In *Mastering MATLAB®*, (pp. 22-30). Upper Saddle River, NJ: Pearson Education Inc.

`format short e` `format shorte`	`3.1416e+00`	5 digits plus a 2-digit exponent
`format long e` `format longe`	`3.141592653589793e+00`	16 digits plus a 2-digit exponent
`format short eng` `format shorteng`	`3.1416e+000`	At least 5 digits plus a 3-digit exponent that is a multiple of three
`format long eng` `format longeng`	`3.14159265358979e+000`	15 significant digits plus a 3-digit exponent that is a multiple of three
`format short g`	`3.1416`	Best of `format short` or `format short e` with 5 digits
`format long g`	`3.14159265358979`	Best of `format long` or `format long e` with 15 digits
`format hex`	`400921fb54442d18`	Hexadecimal, floating point
`format bank`	`3.14`	2 decimal digits
`format +`	`+`	Positive (+), negative (–), or blank ( )
`format rat`	`355/113`	Rational approximation

`format debug`	`Structure address = 1214830`
	`m = 1`
	`n = 1`
	`pr = 11d60d0`
	`pi = 0`
	`3.1416`
		Internal storage information in addition to `short g`

Note: In MATLAB, the internal representation of a number does not change when different display formats are chosen; only the display changes. All calculations using double-precision numbers are performed using double-precision arithmetic.

3.5 SYSTEM INFORMATION

MATLAB provides a number of commands that provide information about the computer in use, as well as about the MATLAB version in use. The command `computer` returns a character string identifying the computer in use:

```
>> computer
ans =
GLNX86
```

In this case, the computer is a PC that is running 32-bit *Linux*. The command `version` returns a character string identifying the MATLAB version:

```
>> version
ans =
7.12.0.635 (R2011a)
```

Note that the command `ver` returns information about MATLAB, as well as installed toolboxes:

```
>> ver
-----------------------------------------------------------------------------
MATLAB Version 7.12.0.635 (R2011a)
MATLAB License Number: 123456
Operating System: Linux 2.6.38.8-35.fc15.i686 #1 SMP Wed Jul 6 14:46:26 UTC 2011 i686
Java VM Version: Java 1.6.0_17-b04 with Sun Microsystems Inc. Java HotSpot(TM) Client
VM mixed mode
-----------------------------------------------------------------------------
MATLAB                                Version 7.12          (R2011a)
Mastering MATLAB Toolbox              Version 6.0
```

MATLAB licensing information can be found by using the commands `license` and `hostid`:

```
>> license
ans =
123456

>> hostid
   'no_file'
```

Of course, your results from entering these commands will most likely be different from those shown above since your computer and MATLAB version differ from those used to produce this text. The `hostid` command returns information about your license server. In this example, the license is node-locked and does not require a separate license server.

3.6 THE MATLAB SEARCH PATH

MATLAB uses a ***search path*** to find information stored in files on disk. MATLAB's files are organized into numerous directories and subdirectories.

> The list of all directories where MATLAB's files are found is called the MATLAB search path or simply MATLAB path.

Use of the MATLAB search path is described next. When you enter `cow` at the MATLAB prompt by typing `>> cow`, MATLAB does the following:

1. It checks to see if `cow` is a ***variable*** in the MATLAB workspace.
2. If not, checks to see if `cow` is a built-in function.
3. If not, searches for a file named `cow.m` in the ***current directory***.
4. If none exists, checks to see if `cow.m` exists anywhere on the ***MATLAB search path*** by searching in the order in which the path is specified.
5. If `cow` isn't found at this point, MATLAB reports an error.

MATLAB takes appropriate action according to this search strategy. If `cow` is a variable, MATLAB uses the variable. If `cow` is a built-in function or if `cow.m` is a file either in the current directory or anywhere on the MATLAB search path, the built-in function is executed or the file `cow.m` is opened, and MATLAB acts on what is found in the file. As is documented in Chapters 4 and 12, MATLAB has two basic file types that are of common use; both are simple text files containing MATLAB commands. (See Chapters 4 and 12 for further information regarding these ***M-files***.)

In reality, the search procedure is more complicated than just described because of advanced features in MATLAB. For the most part, however, this search procedure provides sufficient detail for basic MATLAB work. (More detailed information regarding the MATLAB search path can be found in Chapter 12.)

When MATLAB starts up, it defines a default MATLAB search path that points to all of the directories in which MATLAB stores its files. This search path can be displayed and modified in several ways. The easiest way is to use the ***Path Browser***, which is a graphical user interface designed for viewing and modifying the MATLAB search path. The path browser is made available by choosing **Set Path . . .** from the **File** menu on the *MATLAB* desktop window. Since the MATLAB search path already points to all directories where MATLAB stores its files, the primary purpose for accessing the path browser is to add your own MATLAB file storage directories to the search path.

To display the MATLAB search path in the *Command* window, MATLAB provides the functions `path` and `matlabpath`. In addition, the features of the path browser can be duplicated in the *Command* window by using the functions `path`, `addpath`, and `rmpath`. For more information regarding these functions, see the online documentation.

4

Script M-files

For simple problems, entering your requests at the MATLAB prompt in the *Command* window is fast and efficient. However, as the number of commands increases, or when you wish to change the value of one or more variables and reevaluate a number of commands, typing at the MATLAB prompt quickly becomes tedious. MATLAB provides a logical solution to this problem. It allows you to place MATLAB commands in a simple text file and then tells MATLAB to open the file and evaluate the commands exactly as it would have if you had typed the commands at the MATLAB prompt. These files are called ***script files*** or ***M-files***. The term *script* signifies that MATLAB simply reads from the script found in the file. The term *M-file* means that script filenames must end with the extension `'.m'`, as in, for example, `example1.m`.

4.1 SCRIPT M-FILE USE

To create a script M-file, click on the blank page icon on the MATLAB desktop toolbar, or choose **New** from the **File** menu and select **M-file**. This procedure brings up a text editor window wherein you can enter MATLAB commands. The following script M-file shows the commands from an example considered earlier:

```
% script M-file example1.m
  erasers = 4; % number of each item
  pads = 6;
  tape = 2;
```


Hanselman, D. and Littlefield, B. (2012). Script M-files. In *Mastering MATLAB®*, (pp. 31-39). Upper Saddle River, NJ: Pearson Education Inc.

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

3.6 THE MATLAB SEARCH PATH

MATLAB uses a ***search path*** to find information stored in files on disk. MATLAB's files are organized into numerous directories and subdirectories.

> The list of all directories where MATLAB's files are found is called the MATLAB search path or simply MATLAB path.

Use of the MATLAB search path is described next. When you enter `cow` at the MATLAB prompt by typing `>> cow`, MATLAB does the following:

1. It checks to see if `cow` is a ***variable*** in the MATLAB workspace.
2. If not, checks to see if `cow` is a built-in function.
3. If not, searches for a file named `cow.m` in the ***current directory***.
4. If none exists, checks to see if `cow.m` exists anywhere on the ***MATLAB search path*** by searching in the order in which the path is specified.
5. If `cow` isn't found at this point, MATLAB reports an error.

MATLAB takes appropriate action according to this search strategy. If `cow` is a variable, MATLAB uses the variable. If `cow` is a built-in function or if `cow.m` is a file either in the current directory or anywhere on the MATLAB search path, the built-in function is executed or the file `cow.m` is opened, and MATLAB acts on what is found in the file. As is documented in Chapters 4 and 12, MATLAB has two basic file types that are of common use; both are simple text files containing MATLAB commands. (See Chapters 4 and 12 for further information regarding these ***M-files***.)

In reality, the search procedure is more complicated than just described because of advanced features in MATLAB. For the most part, however, this search procedure provides sufficient detail for basic MATLAB work. (More detailed information regarding the MATLAB search path can be found in Chapter 12.)

When MATLAB starts up, it defines a default MATLAB search path that points to all of the directories in which MATLAB stores its files. This search path can be displayed and modified in several ways. The easiest way is to use the ***Path Browser***, which is a graphical user interface designed for viewing and modifying the MATLAB search path. The path browser is made available by choosing **Set Path . . .** from the **File** menu on the *MATLAB* desktop window. Since the MATLAB search path already points to all directories where MATLAB stores its files, the primary purpose for accessing the path browser is to add your own MATLAB file storage directories to the search path.

To display the MATLAB search path in the *Command* window, MATLAB provides the functions `path` and `matlabpath`. In addition, the features of the path browser can be duplicated in the *Command* window by using the functions `path`, `addpath`, and `rmpath`. For more information regarding these functions, see the online documentation.

4

Script M-files

For simple problems, entering your requests at the MATLAB prompt in the *Command* window is fast and efficient. However, as the number of commands increases, or when you wish to change the value of one or more variables and reevaluate a number of commands, typing at the MATLAB prompt quickly becomes tedious. MATLAB provides a logical solution to this problem. It allows you to place MATLAB commands in a simple text file and then tells MATLAB to open the file and evaluate the commands exactly as it would have if you had typed the commands at the MATLAB prompt. These files are called ***script files*** or ***M-files***. The term *script* signifies that MATLAB simply reads from the script found in the file. The term *M-file* means that script filenames must end with the extension `'.m'`, as in, for example, `example1.m`.

4.1 SCRIPT M-FILE USE

To create a script M-file, click on the blank page icon on the MATLAB desktop toolbar, or choose **New** from the **File** menu and select **M-file**. This procedure brings up a text editor window wherein you can enter MATLAB commands. The following script M-file shows the commands from an example considered earlier:

```
% script M-file example1.m
  erasers = 4; % number of each item
  pads = 6;
  tape = 2;
```


Hanselman, D. and Littlefield, B. (2012). Script M-files. In *Mastering MATLAB®*, (pp. 31-39). Upper Saddle River, NJ: Pearson Education Inc.

```
items = erasers + pads + tape
cost = erasers*25 + pads*52 + tape*99
average_cost = cost/items
```

This file can be saved to disk and executed immediately by (1) choosing **Save and Run** from the **Debug** menu, (2) pressing the **Save and Run** button on the *Editor* toolbar, or (3) simply pressing the function key **F5**. Alternatively, you can save this file as the M-file `example1.m` on your disk by choosing **Save** from the **File** menu; then, at the MATLAB prompt, it's just a matter of typing the name of the script file without the `.m` extension:

```
>> example1
items =
    12
cost =
  610
average_cost =
      50.833
```

When MATLAB interprets the `example1` statement, it follows the search path described in Chapter 3. In brief, MATLAB prioritizes current MATLAB variables ahead of M-file names. If `example1` is not a current MATLAB variable or a built-in function name, MATLAB opens the file `example1.m` (if it can find it) and evaluates the commands found there just as if they had been entered directly at the *Command* window prompt. As a result, commands within the M-file have access to all of the variables in the MATLAB workspace, and all of the variables created by the M-file become part of the workspace. Normally, the M-file commands are not displayed as they are evaluated. The `echo on` command tells MATLAB to display, or echo, commands to the *Command* window as they are read and evaluated. You can probably guess what the `echo off` command does. Similarly, the command `echo` by itself toggles the echo state.

This ability to create script M-files makes it simple to answer "what-if?" questions. For example, you could repeatedly open the `example1.m` M-file, change the number of `erasers`, `pads`, or `tape`, and then save and execute the file.

The utility of MATLAB comments is readily apparent when you use script files, as shown in `example1.m`. Comments allow you to document the commands found in a script file, so that you do not forget the commands when you view them in the future. In addition, the use of semicolons at the ends of lines to suppress the display of results allows you to control script-file output, so that only important results are shown.

Because of the utility of script files, MATLAB provides several functions that are particularly helpful when used in M-files:

Function	Description
`beep`	Makes computer sound a beep
`disp(variablename)`	Displays results without identifying variable names
`echo`	Controls *Command* window echoing of script file contents as they are executed
`input`	Prompts user for input
`keyboard`	Temporarily gives control to keyboard. (Type `return` to return control to the executing script M-file.)
`pause or pause(n)`	Pauses until user presses any keyboard key, or pauses for `n` seconds and then continues
`waitforbuttonpress`	Pauses until user presses any keyboard key or any mouse button over a figure window

When a MATLAB command is not terminated by a semicolon, the results are displayed in the *Command* window, with the variable name identified. For a prettier display, it is sometimes convenient to suppress the variable name. In MATLAB, this is accomplished with the command `disp`:

```
>> items
items =
    12
>> disp(items)
    12
```

Rather than repeatedly edit a script file for computations for a variety of cases, you can employ the `input` command to prompt for input as a script file is executed. For example, reconsider the `example1.m` script file, with the following modifications:

```
% script M-file example1.m

erasers = 4;  % Number of each item
pads = 6;
tape = input('Enter the number of rolls of tape purchased > ');
items = erasers + pads + tape
```

```
cost = erasers*25 + pads*52 + tape*99
average_cost = cost/items
```

Running this script M-file produces this result:

```
>> example1
Enter the number of rolls of tape purchased > 3
items =
    13
cost =
   709
average_cost =
      54.538
```

In response to the prompt, the number 3 was entered and the **Return** or **Enter** key was pressed. The remaining commands were evaluated as before. The function `input` accepts any valid MATLAB expression for input. For example, running the script file again and providing different input gives the following result:

```
>> example1
Enter the number of rolls of tape purchased > round(sqrt(13))-1
items =
    13
cost =
   709
average_cost =
      54.538
```

In this case, the number of rolls of tape was set equal to the result of evaluating the expression

`round(sqrt(13))-1.`

To see the effect of the `echo` command, add it to the script file and execute it:

```
% script M-file example1.m
echo on
```

```
erasers = 4;  % Number of each item
pads = 6;
tape = input('Enter the number of rolls of tape purchased > ');
items = erasers + pads + tape
cost = erasers*25 + pads*52 + tape*99
average_cost = cost/items
echo off
```

```
>> example1
erasers = 4; % Number of each item
pads = 6;
tape = input('Enter the number of rolls of tape purchased > ');
Enter the number of rolls of tape purchased > 2
items = erasers + pads + tape
items =
    12
cost = erasers*25 + pads*52 + tape*99
cost =
   610
average_cost = cost/items
average_cost =
       50.833
echo off
```

As you can see, the echo command made the result much harder to read. On the other hand, the echo command can be very helpful when debugging more complicated script files.

4.2 BLOCK COMMENTS AND CODE CELLS

In prior versions of MATLAB, comments were line oriented. That is, comments began with an initial percent sign and continued to the end of the current line. To continue the comments on the next line required another initial percent sign.

Therefore, a block of comments would all start with initial percent signs, as shown here:

```
% This is an example of multiple line comments
% in an M-file. Each line requires an initial % sign, or MATLAB
% assumes the line contains code to be executed.
```

While the block of comments is visually simple, it can become cumbersome to manage later, when comment text is augmented or edited. In this case, the percent signs must remain at the beginning of the lines, and the comment text must flow after the initial percent signs on each line. In the past, to make it less cumbersome, the MATLAB editor included commands for adding or removing the initial percent signs on a highlighted block of lines. Now, MATLAB 7 and above supports *block comments* through the use of the syntax %{ and %}. These symbols mark the beginning and end, respectively, of a block of text to be treated as comments by MATLAB. For example, using block comment syntax on the previous example produces

```
%{
This is an example of multiple line comments
in an M-file. Each line requires an initial % sign or MATLAB
assumes the line contains code to be executed.
(Now lines can be added and edited as desired without having to
place percent signs at the beginning of each line.)
%}
```

In addition to their utility in composing multiline comments, block comments will allow you to rapidly turn on and off the interpretation and execution of any number of lines of code in an M-file. This feature is particularly helpful in creating and debugging large M-files. Simply adding %{ before and %} after a block of MATLAB code turns the enclosed code into comments that are not processed by MATLAB. When this feature is used, different sections of a script file can be executed at different times during the editing and debugging processes.

In the past, the MATLAB editor offered commands for executing a block of highlighted code in an editor window. In MATLAB 7 and above, the editor now supports the selective execution of M-file code through the use of ***code cells***. A code cell is simply a block of M-file code that starts with a comment line containing two percent signs followed by a space (i.e., % %). The code cell continues to the end of the M-file or to the beginning of another code cell. From within

ENGINEERING SYSTEMS
Custom Publishing exclusively for The University of Melbourne

the MATLAB editor, cells can be created, individually executed, and sequentially executed, thereby enabling effective M-file debugging. The **Cell** menu in the *Editor* window facilitates these operations. It is important to note that the syntax for code cells is interpreted by the editor, not by the MATLAB command interpreter. As a result, when an M-file is executed after its name has been entered in the *Command* window, code-cell syntax is ignored, and all executable lines in the file are processed.

4.3 SETTING EXECUTION TIME

When the name of an M-file is typed at the *Command* window prompt, the M-file is executed immediately. In some circumstances (such as large programming projects, or situations wherein data to be manipulated becomes available only over a long period of time while MATLAB is running), it is convenient to specify execution times for M-files. In MATLAB, this capability is accomplished by using ***timer objects***. A timer object is created using the function `timer`. For example,

```
>> my_timer = timer('TimerFcn','MfileName','StartDelay',100)
```

creates the timer object stored in the variable `my_timer`, which executes the M-file *`MfileName`* 100 seconds after the timer has been started by using the `start` function:

```
>> start(my_timer) % start the timer in the variable my_timer
```

In general terms, timer function syntax has the form

```
>> t=timer('PropertyName1',PropertyValue1,'PropertyName2',PropertyValue2,...)
```

where the arguments identify property names as character strings paired with corresponding property values.

Neither MATLAB nor the computer operating system is held up while a timer is running, but not executing code. MATLAB and other programs can be used during this idle time. When the timer object initiates code execution, the timer takes control of MATLAB and executes the code, just as if the code had been typed at the MATLAB prompt. When this execution ends, MATLAB returns control to the *Command* window prompt.

The timer object has many more features. For example, *`'MfileName'`* can be any statement executable at the MATLAB prompt. It can be a script M-file, a function handle or function M-file, or a set of MATLAB commands. You also can specify that timer code be executed on a periodic basis or executed a specified number of times. You can specify four different M-files or code sequences to be executed under different conditions. For example,

Hanselman, D. and Littlefield, B. (2012). Script M-files. In *Mastering MATLAB®*, (pp. 31-39). Upper Saddle River, NJ: Pearson Education Inc.

```
>> my_timer = timer('TimerFcn','Mfile1',...
                    'StartFcn','Mfile2',...
                    'StopFcn,, 'Mfile3',...
                    'ErrorFcn','Mfile4');
```

creates a timer function that executes (1) `'Mfile1'` as the primary timer code that may be repeatedly executed, (2) `'Mfile2'` when the timer is started with the `start` function, (3) `'Mfile3'` when the timer is stopped with the `stop` function, and (4) `'Mfile4'` if a MATLAB error occurs while any of these functions are executing. (See the MATLAB documentation for more complete information about timer objects.)

4.4 STARTUP AND FINISH

When MATLAB starts up, it executes two script M-files, `matlabrc.m` and `startup.m`. The first, `matlabrc.m`, comes with MATLAB and generally should not be modified. The commands in this M-file set the default *Figure* window size and placement, as well as a number of other default features. The default MATLAB search path is set by retrieving the script file `pathdef.m` from `matlabrc.m`. The *Path Browser* and *Command* window functions for editing the MATLAB search path maintain the file `pathdef.m`, so there is no need to edit it with a text editor.

Commands in `matlabrc.m` check for the existence of the script M-file `startup.m` on the MATLAB search path. If the M-file exists, the commands in it are executed. This optional M-file `startup.m` typically contains commands that add personal default features to MATLAB. For example, it is common to put one or more `addpath` or `path` commands in `startup.m` to append additional directories to the MATLAB search path. Similarly, the default number display format can be changed (e.g., `format compact`). Since `startup.m` is a standard script M-file, there are no restrictions as to what commands can be placed in it. (However, it's probably not wise to include the command `quit` in `startup.m`!) On single-user installations, `startup.m` is commonly stored in the `toolbox/local` subdirectory on the MATLAB path. On network installations, a convenient location for your `startup.m` file is the default directory where you start MATLAB sessions.

When you terminate MATLAB via the **Exit MATLAB** item on the **File** menu in the *MATLAB* desktop window (or by typing `exit` or `quit` at the MATLAB prompt), MATLAB searches the MATLAB path for a script M-file named `finish.m`. If one is found, the commands in it are executed before MATLAB terminates. For example, the following `finish.m` prompts the user for confirmation using a dialog box before quitting, and the command `quit cancel` provides a way to cancel quitting:

```
%FINISH Confirm Desire for Quitting MATLAB

question = 'Are You Sure You Want To Quit?';
button = questdlg(question,'Exit Request','Yes','No','No');

switch button
case 'No'
   quit cancel; % how to cancel quitting!
end
% 'Yes' lets script and MATLAB end.
```

6

Multidimensional Arrays

In the previous chapter, 1- and 2-D arrays and their manipulation were illustrated. Since MATLAB version 5 appeared several years ago, MATLAB has added support for arrays of arbitrary dimensions. For the most part, MATLAB supports multidimensional arrays (i.e., *n*-D arrays) by using the same functions and addressing techniques that apply to 1- and 2-D arrays. In general, the third dimension is numbered by ***pages***, while higher dimensions have no generic name. Thus, a 3-D array has rows, columns, and pages. Each page contains a 2-D array of rows and columns. In addition, just as all of the columns of a 2-D array must have the same number of rows and vice versa, all of the pages of a 3-D array must have the same number of rows and columns. One way to visualize 3-D arrays is to think of the residential listings (white pages) in a phone book. Each page has the same number of columns and the same number of names (rows) in each column. The stack of all of the pages forms a 3-D array of names and phone numbers.

Even though there is no limit to the number of dimensions, 3-D arrays are predominately used as examples in this chapter, because they are easily visualized and displayed.

6.1 ARRAY CONSTRUCTION

Multidimensional arrays can be created in several ways:

```
>> A = zeros(4,3,2)
A(:,:,1) =
  0    0    0
  0    0    0
```


Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

```
  0     0     0
  0     0     0
A(:,:,2) =
  0     0     0
  0     0     0
  0     0     0
  0     0     0
```

This is an array of zeros, having four rows, three columns, and two pages. The first page is displayed first, followed by the second page. The other common array-generation functions `ones`, `rand`, and `randn` work the same way, simply by adding dimensions to the input arguments.

Direct indexing also works, as in the following example:

```
>> A = zeros(2,3) % start with a 2-D array
A =
     0     0     0
     0     0     0
>> A(:,:,2) = ones(2,3) % add a second page to go 3-D!
A(:,:,1) =
     0     0     0
     0     0     0
A(:,:,2) =
     1     1     1
     1     1     1
>> A(:,:,3) = 4 % add a third page by scalar expansion
A(:,:,1) =
     0     0     0
     0     0     0
A(:,:,2) =
     1     1     1
     1     1     1
A(:,:,3) =
     4     4     4
     4     4     4
```

This approach starts with a 2-D array, which is the first page of a 3-D array. Then additional pages are added by straightforward array addressing.

The functions `reshape` and `repmat` can also be used to create *n*-D arrays. For example, in the code

```
>> B = reshape(A,2,9) % 2-D data, stack pages side-by-side
B =
     0     0     0     1     1     1     4     4     4
     0     0     0     1     1     1     4     4     4

>> B = [A(:,:,1) A(:,:,2) A(:,:,3)] % equivalent to above
B =
     0     0     0     1     1     1     4     4     4
     0     0     0     1     1     1     4     4     4

>> reshape(B,2,3,3) % recreate A
ans(:,:,1) =
     0     0     0
     0     0     0
ans(:,:,2) =
     1     1     1
     1     1     1
ans(:,:,3) =
     4     4     4
     4     4     4

>> reshape(B,[2 3 3]) % alternative to reshape(B,2,3,3)
ans(:,:,1) =
     0     0     0
     0     0     0
ans(:,:,2) =
     1     1     1
     1     1     1
ans(:,:,3) =
     4     4     4
     4     4     4
```

`reshape` can change any dimensional array into any other dimensional array.

The code

```
>> C = ones(2,3) % new data
C =
     1     1     1
     1     1     1

>> repmat(C,1,1,3) % this form not allowed above 2-D!
??? Error using ==> repmat
Too many input arguments.

>> repmat(C,[1 1 3]) % this is how to do it
ans(:,:,1) =
     1     1     1
     1     1     1
ans(:,:,2) =
     1     1     1
     1     1     1
ans(:,:,3) =
     1     1     1
     1     1     1
```

replicates C once in the *row* dimension, once in the *column* dimension, and three times in the *page* dimension.

The cat function creates *n*-D arrays from lower dimensional arrays:

```
>> a = zeros(2); % new data
>> b = ones(2);
>> c = repmat(2,2,2);
>> D = cat(3,a,b,c) % conCATenate a,b,c along the 3rd dimension
D(:,:,1) =
     0     0
     0     0
D(:,:,2) =
     1     1
     1     1
```

```
D(:,:,3) =
     2     2
     2     2
>> D = cat(4,a,b,c) % try the 4th dimension!
D(:,:,1,1) =
     0     0
     0     0
D(:,:,1,2) =
     1     1
     1     1
D(:,:,1,3) =
     2     2
     2     2
>> D(:,1,:,:) % look at elements in column 1
ans(:,:,1,1) =
     0
     0
ans(:,:,1,2) =
     1
     1
ans(:,:,1,3) =
     2
     2
>> size(D)
ans =
     2     2     1     3
```

Note that D has two rows, two columns, one page, and three fourth-dimension parts.

6.2 ARRAY MATHEMATICS AND MANIPULATION

As additional dimensions are created, array mathematics and manipulation become more cumbersome. Scalar–array arithmetic remains straightforward, but array–array arithmetic requires that the two arrays have the same size in all dimensions. Since scalar–array and array–array arithmetic remains unchanged from the 2-D case presented in the previous chapter, further illustrations are not presented here.

MATLAB provides several functions for the manipulation of *n*-D arrays. The function `squeeze` eliminates ***singleton dimensions***; that is, it eliminates dimensions of size 1. For example, in the code

```
>> E = squeeze(D) % squeeze dimension 4 down to dimension 3
E(:,:,1) =
     0     0
     0     0
E(:,:,2) =
     1     1
     1     1
E(:,:,3) =
     2     2
     2     2
>> size(E)
ans =
     2     2     3
```

`E` contains the same data as `D`, but has two rows, two columns, and three pages.

How about a 3-D vector? This case is shown as follows:

```
>> v(1,1,:) = 1:6 % a vector along the page dimension
v(:,:,1) =
     1
v(:,:,2) =
     2
v(:,:,3) =
     3
v(:,:,4) =
     4
v(:,:,5) =
     5
v(:,:,6) =
     6
>> squeeze(v) % squeeze it into a column vector
```

```
ans =
     1
     2
     3
     4
     5
     6
>> v(:) % this always creates a column vector
ans =
     1
     2
     3
     4
     5
     6
```

The function `reshape` allows you to change the row, column, page, and higher order dimensions, without changing the total number of elements:

```
>> F = cat(3,2+zeros(2,4),ones(2,4),zeros(2,4)) % new 3-D array
F(:,:,1) =
     2     2     2     2
     2     2     2     2
F(:,:,2) =
     1     1     1     1
     1     1     1     1
F(:,:,3) =
     0     0     0     0
     0     0     0     0
>> G = reshape(F,[3 2 4]) % change it to 3 rows, 2 columns, 4 pages
G(:,:,1) =
     2     2
     2     2
     2     2
```

Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

```
G(:,:,2) =
     2     1
     2     1
     1     1
G(:,:,3) =
     1     1
     1     0
     1     0
G(:,:,4) =
     0     0
     0     0
     0     0

>> H = reshape(F,[4 3 2]) % or 4 rows, 3 columns, 2 pages
H(:,:,1) =
     2     2     1
     2     2     1
     2     2     1
     2     2     1
H(:,:,2) =
     1     0     0
     1     0     0
     1     0     0
     1     0     0

>> K = reshape(F,2,12) % 2 rows, 12 columns, 1 page
K =
     2     2     2     2     1     1     1     1     0     0     0     0
     2     2     2     2     1     1     1     1     0     0     0     0
```

Reshaping can be confusing until you become comfortable with visualizing arrays in *n*-D space. In addition, some reshaping requests make more practical sense than others. For example, G in the previously displayed code has little practical value, whereas K is much more practical, because it stacks the pages of F side by side as additional columns.

The reshaping process follows the same pattern as that for 2-D arrays. Data are gathered first by rows, followed by columns, then by pages, and so on into higher dimensions. That is, all the rows in the first column are gathered, then all the rows in the second column, and so on. Thus, when the first page has been gathered, MATLAB moves on to the second page and starts over with all the rows in the first column.

The order in which array elements are gathered is the order in which the [redacted] functions `sub2ind` and `ind2sub` consider single-index addressing:

```
>> sub2ind(size(F),1,1,1) % 1st row, 1st column, 1st page is element 1
ans =
     1
>> sub2ind(size(F),1,2,1) % 1st row, 2nd column, 1st page is element 3
ans =
     3

>> sub2ind(size(F),1,2,3) % 1st row, 2nd column, 3rd page is element 19
ans =
     19
>> [r,c,p] = ind2sub(size(F),19) % inverse of above
r =
     1
c =
     2
p =
     3
```

The *n*-D equivalent to `flipud` and `fliplr` is `flipdim`:

```
>> M = reshape(1:18,2,3,3) % new data
M(:,:,1) =
     1     3     5
     2     4     6
M(:,:,2) =
     7     9    11
     8    10    12
M(:,:,3) =
    13    15    17
    14    16    18

>> flipdim(M,1) % flip row order
```

Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

```
ans(:,:,1)
     2     4     6
     1     3     5
ans(:,:,2) =
     8    10    12
     7     9    11
ans(:,:,3) =
    14    16    18
    13    15    17
>> flipdim(M,2) % flip column order
ans(:,:,1) =
     5     3     1
     6     4     2
ans(:,:,2) =
    11     9     7
    12    10     8
ans(:,:,3) =
    17    15    13
    18    16    14

>> flipdim(M,3) % flip page order
ans(:,:,1) =
    13    15    17
    14    16    18
ans(:,:,2) =
     7     9    11
     8    10    12
ans(:,:,3) =
     1     3     5
     2     4     6
```

The function `shiftdim` shifts the dimensions of an array. That is, if an array has `r` rows, `c` columns, and `p` pages, a shift by one dimension creates an array with `c` rows, `p` columns, and `r` pages, as shown in the following example:

```
>> M % recall data
M(:,:,1) =
     1     3     5
     2     4     6
M(:,:,2) =
     7     9    11
     8    10    12
M(:,:,3) =
    13    15    17
    14    16    18
>> shiftdim(M,1) % shift one dimension
ans(:,:,1) =
     1     7    13
     3     9    15
     5    11    17
ans(:,:,2) =
     2     8    14
     4    10    16
     6    12    18
```

Shifting dimensions by 1 causes the first row on page 1 to become the first column on page 1, the second row on page 1 to become the first column on page 2, and so on.

In the code

```
>> shiftdim(M,2) % shift two dimensions
ans(:,:,1) =
     1     2
     7     8
    13    14
ans(:,:,2) =
     3     4
     9    10
    15    16
```

```
ans(:,:,3) =
     5     6
    11    12
    17    18
```

the first column on page 1 of M becomes the first row on page 1, the first column on page 2 becomes the second row on page 1, and so on. If you are like the authors of this text, shifting dimensions is not immediately intuitive. For the 3-D case, it helps if you visualize M forming a rectangular box with page 1 on the front, followed by page 2 behind page 1, and then page 3 forming the back of the box. Hence, shifting dimensions is equivalent to rotating the box so that a different side faces you.

The function `shiftdim` also accepts negative shifts. In this case, the array is pushed into higher dimensions, leaving singleton dimensions behind, as in this example

```
>> M % recall data
M(:,:,1) =
     1     3     5
     2     4     6
M(:,:,2) =
     7     9    11
     8    10    12
M(:,:,3) =
    13    15    17
    14    16    18
>> size(M) % M has 2 rows, 3 columns, and 3 pages
ans =
     2     3     3

>> shiftdim(M,-1) % shift dimensions out by 1
ans(:,:,1,1) =
     1     2
ans(:,:,2,1) =
     3     4
ans(:,:,3,1) =
     5     6
ans(:,:,1,2) =
     7     8
```

```
ans(:,:,2,2) =
     9    10
ans(:,:,3,2) =
    11    12
ans(:,:,1,3) =
    13    14
ans(:,:,2,3) =
    15    16
ans(:,:,3,3) =
    17    18
>> size(ans)
ans =
     1     2     3     3
```

The result now has four dimensions. The reader is left to figure out the correspondence between the original data and the shifted result.

In 2-D arrays, the transpose operator swapped rows and columns, converting an `r`-by-`c` array into a `c`-by-`r` array. The functions `permute` and `ipermute` are the *n*-D equivalents of the transpose operator. By itself, `permute` is a generalization of the function `shiftdim`. In the code

```
>> M % recall data
M(:,:,1) =
     1     3     5
     2     4     6
M(:,:,2) =
     7     9    11
     8    10    12
M(:,:,3) =
    13    15    17
    14    16    18
>> permute(M,[2 3 1]) % same as shiftdim(M,1)
ans(:,:,1) =
     1     7    13
     3     9    15
     5    11    17
```

Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

```
ans(:,:,2) =
     2     8    14
     4    10    16
     6    12    18

>> shiftdim(M,1)
ans(:,:,1) =
     1     7    13
     3     9    15
     5    11    17
ans(:,:,2) =
     2     8    14
     4    10    16
     6    12    18
```

`[2 3 1]` instructs the function to make the second dimension the first, the third dimension the second, and the first dimension the third.

Now, in the simpler example

```
>> permute(M,[2 1 3])
ans(:,:,1) =
     1     2
     3     4
     5     6
ans(:,:,2) =
     7     8
     9    10
    11    12
ans(:,:,3) =
    13    14
    15    16
    17    18
```

`[2 1 3]` instructs `permute` to transpose the rows and columns, but to leave the third dimension alone. As a result, each page in the result is a conventional transpose of the original data.

The second argument to `permute`, called `ORDER`, must be a permutation of the dimensions of the array, passed as the first argument; otherwise, the requested permutation doesn't make sense, as shown in the following example:

Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

```
>> permute(M,[2 1 1])
??? Error using ==> permute
ORDER cannot contain repeated permutation indices.
>> permute(M,[2 1 4])
??? Error using ==> permute
ORDER contains an invalid permutation index.
```

The function `permute` can also be used to push an array into higher dimensions. For example, `shiftdim(M,-1)`, shown earlier, is equivalent to

```
>> permute(M,[4 1 2 3])
ans(:,:,1,1) =
     1     2
ans(:,:,2,1) =
     3     4
ans(:,:,3,1) =
     5     6
ans(:,:,1,2) =
     7     8
ans(:,:,2,2) =
     9    10
ans(:,:,3,2) =
    11    12
ans(:,:,1,3) =
    13    14
ans(:,:,2,3) =
    15    16
ans(:,:,3,3) =
    17    18
```

An array always has a unit dimension beyond its size; for example, a 2-D array has one page. That is, all dimensions past the nonunity size of an array are singletons. As a result, in the preceding example, the singleton fourth dimension in `M` is made the first dimension of the result shown.

For 2-D arrays, issuing the transpose operator a second time returns the array to its original form. Because of the added generality of *n*-D arrays, the function `ipermute` is used to undo the actions performed by `permute`:

```
>> M % recall data
M(:,:,1) =
     1     3     5
     2     4     6
M(:,:,2) =
     7     9    11
     8    10    12
M(:,:,3) =
    13    15    17
    14    16    18
>> permute(M,[3 2 1]) % sample permutation
ans(:,:,1) =
     1     3     5
     7     9    11
    13    15    17
ans(:,:,2) =
     2     4     6
     8    10    12
    14    16    18
>> ipermute(ans,[3 2 1]) % back to original data
ans(:,:,1) =
     1     3     5
     2     4     6
ans(:,:,2) =
     7     9    11
     8    10    12
ans(:,:,3) =
    13    15    17
    14    16    18
```

6.3 ARRAY SIZE

As demonstrated in the prior chapter and earlier in this chapter, the function `size` returns the size of an array along each of its dimensions. The functionality of `size` is unchanged from the features demonstrated in the last chapter. In addition, the function `numel` remains unchanged. Consider the following example

```
>> size(M) % return array of dimensions
ans =
     2     3     3
>> numel(M) % number of elements
ans =
    18
>> [r,c,p] = size(M) % return individual variables
r =
     2
c =
     3
p =
     3
>> r = size(M,1) % return just rows
ans =
     2
>> c = size(M,2) % return just columns
c =
     3
>> p = size(M,3) % return just pages
p =
     3
>> v = size(M,4) % default for all higher dimensions
v =
     1
```

When the number of dimensions is unknown or variable, the function `ndims` is useful:

```
>> ndims(M)
ans =
     3
>> ndims(M(:,:,1)) % just the 2-D first page of M
ans =
     2
```

In this last example, `M(:,:,1)` is a 2-D array, because it has only one page. That is, it has a singleton third dimension. The function `ndims` is equivalent to the following simple code fragment:

```
>> length(size(M))
ans =
     3
```

The following table summarizes the functions illustrated in this chapter:

n-D Function	Description

`ones(r,c,...)`
`zeros(r,c,...)`
`rand(r,c,...)`
`randn(r,c,...)`
	Basic *n*-D array creation

`reshape(B,2,3,3)` `reshape(B,[2 3 3])`	Reshapes array into arbitrary dimensions
`repmat(C,[1 1 3])`	Replicates array into arbitrary dimensions
`cat(3,a,b,c)`	Concatenates array along a specified dimension
`squeeze(D)`	Eliminates dimensions of size equal to 1 (i.e., singleton dimensions)
`sub2ind(size(F),1,1,1)` `[r,c,p]=ind2sub(size(F),19)`	Subscript to single-index conversion, and single-index to subscript conversion
`flipdim(M,1)`	Flips order along a given dimension; *n*-D equivalent to `flipud` and `fliplr`
`shiftdim(M,2)`	Shifts dimensions; circular shift for positive second argument; push-out for negative second argument
`permute(M,[2 1 3])` `ipermute(M,[2 1 3])`	Arbitrary permutation and inverse of dimensions; generalization of transpose operator to *n*-D arrays

`size(M)`
`[r,c,p]=size(M)`
`r=size(M,1)`
`c=size(M,2)`
`p=size(M,3)`
	Size of *n*-D array along its dimensions

`ndims(M)`	Number of dimensions in an array
`numel(M)`	Number of elements in an array

Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

8

Cell Arrays and Structures

MATLAB version 5 introduced two container data types called ***cell arrays*** and ***structures***. These data types allow the grouping of dissimilar, but related, arrays into a single variable. Data management then becomes easier, since groups of related data can be organized and accessed through a cell array or structure. Because cell arrays and structures are *containers* for other data types, mathematical operations on them are not defined. The *contents* of a cell or structure must be addressed to perform mathematical operations.

One way to visualize cell arrays is to imagine a collection of post office boxes covering a wall at the post office. The collection of boxes is the cell array, with each box being one cell in the array. The contents of each post office box are different, just as the contents of each cell in a cell array are different types or sizes of MATLAB data, such as character strings or numerical arrays of varying dimensions. Just as each post office box is identified by a number, each cell in a cell array is indexed by a number. When you send mail to a post office box, you identify the box number you want it put in. When you put data in a particular cell, you identify the cell number you want them put in. The number is also used to identify which box or cell to take data out of.

Structures are almost identical to cell arrays, except that the individual post office boxes (or data-storage locations) are not identified by number. Instead, they are identified by name. Continuing the post office box analogy, the collection of post office boxes is the structure, and each box is identified by its owner's name. To send mail to a post office box, you identify the name of the box you want it put in. To place data in a particular structure element, you identify the name (i.e., field) of the structure element to put the data in.

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

In this last example, `M(:,:,1)` is a 2-D array, because it has only one page. That is, it has a singleton third dimension. The function `ndims` is equivalent to the following simple code fragment:

```
>> length(size(M))
ans =
     3
```

The following table summarizes the functions illustrated in this chapter:

n-D Function	Description

`ones(r,c,...)`
`zeros(r,c,...)`
`rand(r,c,...)`
`randn(r,c,...)`
	Basic *n*-D array creation

`reshape(B,2,3,3)` `reshape(B,[2 3 3])`	Reshapes array into arbitrary dimensions
`repmat(C,[1 1 3])`	Replicates array into arbitrary dimensions
`cat(3,a,b,c)`	Concatenates array along a specified dimensio[redacted]
`squeeze(D)`	Eliminates dimensions of size equal to 1 (i.e., singleton dimensions)
`sub2ind(size(F),1,1,1)` `[r,c,p]=ind2sub(size(F),19)`	Subscript to single-index conversion, and single-index to subscript conversion
`flipdim(M,1)`	Flips order along a given dimension; *n*-D equivalent to `flipud` and `fliplr`
`shiftdim(M,2)`	Shifts dimensions; circular shift for positive second argument; push-out for negative second argument
`permute(M,[2 1 3])` `ipermute(M,[2 1 3])`	Arbitrary permutation and inverse of dimensio[redacted] generalization of transpose operator to *n*-D arrays

`size(M)`
`[r,c,p]=size(M)`
`r=size(M,1)`
`c=size(M,2)`
`p=size(M,3)`
	Size of *n*-D array along its dimensions

`ndims(M)`	Number of dimensions in an array
`numel(M)`	Number of elements in an array

Hanselman, D. and Littlefield, B. (2012). Miltidimensional Arrays. In *Mastering MATLAB®*, (pp. 101-118). Upper Saddle River, NJ: Pearson Education Inc.

8

Cell Arrays and Structures

MATLAB version 5 introduced two container data types called ***cell arrays*** and ***structures***. These data types allow the grouping of dissimilar, but related, arrays into a single variable. Data management then becomes easier, since groups of related data can be organized and accessed through a cell array or structure. Because cell arrays and structures are *containers* for other data types, mathematical operations on them are not defined. The *contents* of a cell or structure must be addressed to perform mathematical operations.

One way to visualize cell arrays is to imagine a collection of post office boxes covering a wall at the post office. The collection of boxes is the cell array, with each box being one cell in the array. The contents of each post office box are different, just as the contents of each cell in a cell array are different types or sizes of MATLAB data, such as character strings or numerical arrays of varying dimensions. Just as each post office box is identified by a number, each cell in a cell array is indexed by a number. When you send mail to a post office box, you identify the box number you want it put in. When you put data in a particular cell, you identify the cell number you want them put in. The number is also used to identify which box or cell to take data out of.

Structures are almost identical to cell arrays, except that the individual post office boxes (or data-storage locations) are not identified by number. Instead, they are identified by name. Continuing the post office box analogy, the collection of post office boxes is the structure, and each box is identified by its owner's name. To send mail to a post office box, you identify the name of the box you want it put in. To place data in a particular structure element, you identify the name (i.e., field) of the structure element to put the data in.

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

8.1 CELL ARRAY CREATION

Cell arrays are MATLAB arrays whose elements are cells. Each cell in a cell array can hold any MATLAB data type, including numerical arrays, character strings, symbolic objects, other cell arrays, and structures. For example, one cell of a cell array might contain a numerical array; another, an array of character strings; and another, a vector of complex values. Cell arrays can be created with any number of dimensions, just as numerical arrays can. However, in most cases, cell arrays are created as a simple vector of cells.

Cell arrays can be created by using assignment statements or by preallocating the array with the `cell` function and then assigning data to the cells. If you have trouble with these examples, it is very likely that you have another variable of the same name in the workspace. If any of the examples that follow give unexpected results, clear the array from the workspace and try again.

> If you assign a cell to an existing variable that is not a cell, MATLAB will stop and report an error.

Like other kinds of arrays, cell arrays can be built by assigning data to individual cells, one at a time. There are two different ways to access cells. If you use standard array syntax to index the array, you must enclose the cell contents in curly braces, `{ }`. These curly braces construct a cell array. They act like a call to the `cell` function.

```
>> clear A % make sure A is not being used
>> A(1,1) = { [1 2 3; 4 5 6; 7 8 9] };
>> A(1,2) = { 2+3i };                % semicolons suppress display
>> A(2,1) = { 'A character string' };
>> A(2,2) = { 12:-2:0 } % no semicolon, so display requested
A =
            [3x3 double]    [2.0000+ 3.0000i]
    'A character string'         [1x7 double]
```

Note that MATLAB shows that A is a 2-by-2 cell array, but it does not show the contents of all cells. Basically, MATLAB shows the cell contents if they do not take up much space, but only describes the cell contents if they do take up significant space. The curly braces on the right side of the equal sign indicate that the right-hand expression returns a cell, rather than numerical values. This is called ***cell indexing***. Alternatively, the following statements create the same cell array:

```
>> A{1,1} = [1 2 3; 4 5 6; 7 8 9];
>> A{1,2} = 2+3i;
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

```
>> A{2,1} = 'A character string';
>> A{2,2} = 12:-2:0
A =
            [3x3 double]    [2.0000+ 3.0000i]
    'A character string'          [1x7 double]
```

Here, the curly braces appear on the left-hand side of the equal sign. The term `A{1,1}` indicates that `A` is a cell array and that the contents of the first row and first column of the cell array are on the right-hand side of the equal sign. This is called ***content addressing***. Both methods can be used interchangeably.

As the example demonstrates, curly braces `{ }` are used to access or specify the contents of cells, whereas parentheses `( )` are used to identify cells, but not their contents. To apply the post office box analogy, curly braces are used to look at the contents of post office boxes, whereas parentheses are used to identify post office boxes without looking inside at their contents.

> The two commands `A(i,j) = {x};` and `A{i,j} = x;` both tell MATLAB to store the content of variable `x` in the `(i,j)` element of the cell array `A`. The notation `A(i,j)` is called **cell indexing**, and the notation `A{i,j}` is called **content addressing**. That is, curly braces `{ }` access the contents of cells, whereas parentheses `( )` identify cells without looking at their contents.

The `celldisp` function forces MATLAB to display the contents of cells in the usual manner:

```
>> celldisp(A)
A{1,1} =
    1    2    3
    4    5    6
    7    8    9
A{2,1} =
    A character string
A{1,2} =
    2.0000+  3.0000i
A{2,2} =
    12    10    8    6    4    2    0
```

If a cell array has many elements, `celldisp` can produce a lot of output to the *Command* window. As an alternative, requesting the contents of the cell by using content addressing displays the contents of a single cell. This is different from cell indexing, which identifies the cell, but not its contents:

```
>> A{2,2} % content addressing
ans =
    12    10    8    6    4    2    0
>> A(2,2) % cell indexing
ans =
    [1x7 double]

>> A{1,:} % address contents of the first row
ans =
    1    2    3
    4    5    6
    7    8    9
ans =
    2.0000 + 3.0000i
>> A(1,:)
ans =
     [3x3 double]    [2.0000+ 3.0000i]
```

Note that the contents of all of the cells shown are generically named ans, because the cells store data that do not have associated variable names.

Square brackets were used in previous chapters to create numerical arrays. Curly braces work the same way for cells. In the following example, commas separate columns, and semicolons separate rows:

```
>> B = { [1 2], 'John Smith'; 2+3i, 5 }
B =
          [1x2 double]    'John Smith'
     [2.0000+ 3.0000i]    [         5]
```

When dealing with numerical arrays, it is common to preallocate an array with zeros and then fill the array as needed. The same can be done with cell arrays. The `cell` function creates a cell array and fills it with empty numerical matrices, `[]`:

```
>> C = cell(2,3)
C =
    []    []    []
    []    []    []
```

Once the cell array has been defined, both cell indexing and content addressing can be used to populate the cells. In the code

```
>> C(1,1) = 'This doesn''t work'
??? Conversion to cell from char is not possible.
```

the left-hand side uses cell indexing. As a result, the right-hand side must be a cell, but it is not, because it lacks curly braces surrounding its contents. This problem can be resolved by addressing the cell correctly:

```
>> C(1,1) = { 'This does work' }
C =
    'This does work'    []    []
                  []    []    []
>> C{2,3} = 'This works too'
C =
    'This does work'    []                    []
                  []    []     'This works too'
```

Because curly braces appear on the left-hand side of the last statement, MATLAB makes the character string the contents of the addressed cell. Once again, this is an example of content addressing, whereas the prior statement is an example of cell indexing.

8.2 CELL ARRAY MANIPULATION

The numerical array-manipulation techniques presented in preceding chapters apply to cell arrays as well. In a sense, cell array manipulation is just a natural extension of these techniques to a different type of array. If you assign data to a cell outside of the dimensions of the current cell array, MATLAB automatically expands the array and fills the intervening cells with the empty numerical array [].

Square brackets are used to combine cell arrays into larger cell arrays, just as they are used to construct larger numerical arrays:

```
>> A % recall prior cell arrays
A =
            [3x3 double]    [2.0000+ 3.0000i]
    'A character string'         [1x7 double]
>> B
B =
            [1x2 double]    'John Smith'
        [2.0000+ 3.0000i]    [          5]
```

```
>> C = [A;B]
C =
            [3x3 double]    [2.0000+ 3.0000i]
    'A character string'         [1x7 double]
            [1x2 double]    'John Smith'
   [     2.0000+ 3.0000i]    [             5]
```

A subset of cells can be extracted to create a new cell array by conventional array-addressing techniques:

```
>> D = C([1 3],:) % first and third rows
D =
   [3x3 double]     [2.0000+ 3.0000i]
   [1x2 double]     'John Smith'
```

An entire row or column of a cell array can be deleted by using the empty array:

```
>> C(3,:) = []
C =
            [3x3 double]     [2.0000+ 3.0000i]
    'A character string'          [1x7 double]
   [     2.0000+ 3.0000i]      [             5]
```

Note that curly braces do not appear in either of the preceding expressio[redacted] because we are working with the cell array itself, not the contents of the cells. Once again, curly braces are used to address the contents of cells, whereas parenthe[redacted] are used to identify the cells without regard for their content.

The `reshape` function can be used to change the configuration of a cell array, but cannot be used to add or remove cells. For example, in the code

```
>> X = cell(3,4);
>> size(X) % size of the cell array, not the contents
ans =
    3    4

>> Y = reshape(X,6,2);
>> size(Y)
ans =
    6    2
```

the function `reshape` naturally reshapes any array, without regard to its type. Similarly, the `size` function returns the size of any array type.

The function `repmat` also works, even though its name implies that it replicates matrices. The function `repmat` was created when matrices were MATLAB's only data type. If MATLAB had been written after the introduction of cell arrays and other array types, perhaps this function would have been called `reparray`. The following example is illustrative:

```
>> Y % recall data
Y =
   []    []
   []    []
   []    []
   []    []
   []    []
   []    []
>> Z = repmat(Y,1,3)
Z =
   []    []    []    []    []    []
   []    []    []    []    []    []
   []    []    []    []    []    []
   []    []    []    []    []    []
   []    []    []    []    []    []
   []    []    []    []    []    []
```

8.3 RETRIEVING CELL ARRAY CONTENT

To retrieve the contents of a cell in a cell array, you must apply content addressing, which involves the use of curly braces. For instance, in the code

```
>> B % recall cell array
B =
        [1x2 double]      'John Smith'
    [2.0000+ 3.0000i]      [        5]
>> x = B{2,2} % content addressing uses { }
x =
     5
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

the variable x now contains the numerical value 5, making x a numerical array (a scalar, in this case). The `class` function can be used to confirm that this is true, as in the following example:

```
>> class(x) % return argument's data type
ans =
     double
```

Because numerical arrays are double precision, `class` returns a character string that identifies x as being `double`. If cell indexing had been used mistakenly, the result would be different:

```
>> y = B(2,2)    % cell indexing uses ( )
y =
    [5]
>> y = B(4)      % same as above using single index
y =
    [5]

>> class(y)      % y is not a double, but a cell!
ans =
cell
>> class(y{1})   % but the contents of y is a double!
ans =
double
```

By now, you are either bored with this distinction between cell indexing and content addressing, or your head is spinning from the confusion.

There are also other functions for testing variable types. The following examples return logical results, where True = 1 and False = 0:

```
>> iscell(y)        % yes, y is a cell
ans =
     1
>> iscell(y{1})     % contents of y is NOT a cell
ans =
     0
```

```
>> isa(y,'cell')   % yes, y is a cell
ans =
     1
>> isdouble(y{1})  % this function doesn't exist
??? Undefined function or method 'isdouble' for input arguments of
type 'double'.
>> isnumeric(y{1}) % contents of y is numerical
ans =
     1
> isfloat(y{1}) % contents of y is floating point
ans =
     1
>> isa(y{1},'double')  % contents of y is a double
ans =
     1
>> isa(y{1},'numeric') % contents of y is also numeric
ans =
     1
>> isa(y{1},'cell')    % contents of y is NOT a cell
ans =
     0
```

While you can display the contents of more than one cell array at a time, it is not possible to assign more than one at a time to a single variable by using a standard assignment statement. Consider the following code:

```
>> B{:,2}
ans =
John Smith
ans =
     5
>> d = B{:,2}
d =
John Smith
>> class(d)
```

```
ans =
char
```

If you think about it, this behavior makes sense. How can two pieces of data be assigned to a single variable? In the preceding example, d is assigned the conten[redacted] of the first cell referenced by B{:,2}, that is, B{1,2}. The cell content referenc[redacted] by B{2,2} is superfluous and is silently dropped. Extracting data from multipl[redacted] cells at a time simply cannot be done so casually.

When addressing the contents of a single cell, it is possible to further addre[redacted] a subset of the contents by simply appending the desired subscript range:

```
>> A % recall prior data
A =
          [3x3 double]    [2.0000+ 3.0000i]
   'A character string'         [1x7 double]
>> celldisp(A) % display contents
A{1,1} =
     1    2    3
     4    5    6
     7    8    9
A{2,1} =
A character string
A{1,2} =
    2.0000 + 3.0000i
A{2,2} =
    12   10    8    6    4    2    0
>> A{1,1}(3,:)  % third row of 3-by-3 array
ans =
     7    8    9
>> A{4}(2:5)    % second through fifth elements of A{2,2}
ans =
    10    8    6    4
>> A{1,2}(2)    % second element doesn't exist
??? Index exceeds matrix dimensions.
>> A{2,1}(3:11) % extract part of the character string
ans =
character
```

COMMA-SEPARATED LISTS

To extract the contents of more than one cell at a time, MATLAB provides comma–separated list syntax. This syntax applies in any place where variables or constants appear in a list separated by commas. For example, comma-separated lists appear in array construction:

```
>> a = ones(2,3);
>> b = zeros(2,1);
>> c = (3:4)';
>> d = [a,b,c] % same as [a b c]
d =
 1   1   1   0   3
 1   1   1   0   4
```

They also appear in function input and output argument lists:

```
>> d = cat(2,a,b,c)
d =
    1   1   1   0   3
    1   1   1   0   4
>> [m,n] = size(d)
m =
  2
n =
  5
```

Comma–separated list syntax is implemented as follows: Placing a content-addressed cell array in locations where comma-separated lists appear causes MATLAB to extract the contents of the addressed cell arrays and place them sequentially, separated by commas. In the code

```
>> F = {a b c} % create a cell array
F =
    [2x3 double]    [2x1 double]    [2x1 double]
>> d = cat(2,F{:}) % same as cat(2,a,b,c)
d =
    1   1   1   0   3
    1   1   1   0   4
```

```
>> d = cat(2,F(:)) % not content addressing
d =
     [2x3 double]
     [2x1 double]
     [2x1 double]
```

`cat(2,F{:})` is interpreted by MATLAB as `cat(2,F{1},F{2},F{3})`, which is equal to `cat(2,a,b,c)`. That is, comma–separated list syntax dictates that `F{:}` is interpreted as a listing of all addressed parts of `F{:}`, separated by commas. Note that content addressing must be used. Cell indexing as shown in the second case in the preceding example, acts on the cells, not on their contents. As a result, there is no comma-separated list in the second case, `cat(2,F(:))`. Consider this example:

```
>> d = [F{:}]
d =
     1   1   1   0   3
     1   1   1   0   4
>> d = [F{1},F{2},F{3}] % what is implied by the above
d =
     1   1   1   0   3
     1   1   1   0   4
>> e = [F{2:3}] % can also content address any subset
e =
     0   3
     0   4
>> e = [F{2},F{3}] % what is implied by the above
e =
     0   3
     0   4
```

At first, comma–separated list syntax may seem strange. However, once you become familiar with it, you will recognize its power. (For more information, see the online help for `lists`, or search for the phrase `comma separated list`.)

Given comma–separated list syntax, MATLAB provides a way to extract the contents of numerous cells into separate variables. In the code

```
>> celldisp(F) % recall data
F{1} =
```

```
     1     1     1
     1     1     1
F{2} =
     0
     0
F{3} =
     3
     4
>> [r,s,t] = F{:} % extract the contents of F into separate
variables
r =
     1     1     1
     1     1     1
s =
     0
     0
t =
     3
     4
```

the variables r, s, and t are numerical variables, with `r = F{1}`, `s = F{2}`, `t = F{3}`.

Comma-separated lists can be used to assign the contents of multiple unrelated variables in one statement. If you have ever used playing cards, you know that "to deal" means to pass out the cards in some organized way. The function `deal` does just that; it passes out the contents of a list of variables to another set of variables. In the code

```
>> a,b,c % recall data
a =
     1     1     1
     1     1     1
b =
     0
     0
c =
     3
     4
```

```
>> [r,s,t] = deal(a,b,c) % deal out contents of a, b, and c
r =
     1     1     1
     1     1     1
s =
     0
     0
t =
     3
     4
```

the variables r, s, and t are numerical variables, with r = a, s = b, t = c. So, the function deal is, in reality, pretty simple. It just assigns the contents of the first input argument to the first output argument, the second to the second, and so on. Despite its simplicity, deal is a useful tool for extracting data from multiple variables with one statement when content addressing is not easily accomplished.

Because the output of deal is also a comma-separated list, deal can be used to assign the contents of multiple types of variables in one statement:

```
>> d = eye(2);
>> e = 'This is a character string.';
>> f = {[1 2 3]};

>> [x,y,z] = deal(d,e,f)
x =
     1     0
     0     1
y =
This is a character string.
z =
    [1x3 double]

>> class(x)
ans =
double
>> class(y)
```

```
ans =
char
>> class(z)
ans =
cell
```

8.5 CELL FUNCTIONS

Besides the functions `celldisp`, `cell`, `iscell`, `deal`, and `isa`, there are several other functions that are useful when dealing with cell arrays. The function `cellfun` provides a way to apply certain functions to all cells in a cell array, thereby eliminating the need to apply them to each cell individually:

```
>> A % recall data
A =
            [3x3 double]    [2.0000+ 3.0000i]
    'A character string'         [1x7 double]

>> cellfun('isreal',A) % True=1 where not complex
ans =
    1    0
    1    1
>> cellfun('length',A) % length of contents
ans =
    3    1
   18    7
>> cellfun('prodofsize',A) % number of elements in each cell
ans =
    9    1
   18    7
>> cellfun('isclass',A,'char') % True for character strings
ans =
    0    0
    1    0
```

The function `cellfun` offers other options as well. For further information, see online help.

Another function that is sometimes useful is `num2cell`. This function takes an array of any type (not just numbers as the function name suggests) and fills a cell array with its components:

```
>> a = rand(3,6) % new numerical data
a =
    0.7266    0.2679    0.6833    0.6288    0.6072    0.5751
    0.4120    0.4399    0.2126    0.1338    0.6299    0.4514
    0.7446    0.9334    0.8392    0.2071    0.3705    0.0439
>> c = num2cell(a) % c{i,j}=a(i,j)
c =
    [0.7266]    [0.2679]    [0.6833]    [0.6288]    [0.6072]    [0.5751]
    [0.4120]    [0.4399]    [0.2126]    [0.1338]    [0.6299]    [0.4514]
    [0.7446]    [0.9334]    [0.8392]    [0.2071]    [0.3705]    [0.0439]
>> d = num2cell(a,1) % d{i}=a(:,i)
d =
  Columns 1 through 4
    [3x1 double]    [3x1 double]    [3x1 double]    [3x1 double]
  Columns 5 through 6
    [3x1 double]    [3x1 double]
>> e = num2cell(a,2) % e{i}=a(i,:)
e =
    [1x6 double]
    [1x6 double]
    [1x6 double]
```

With numerical data as input, `num2cell(a)` isn't useful in many applications, but packing larger pieces of an array into cells is often more useful, as illustrated in the last two cases.

A similar function is `mat2cell`. This function breaks up a two-dimensional matrix of any type into a cell array of adjacent matrices. The arguments are an input matrix, a vector of the distribution of rows, and a vector of the distribution of columns. The distribution vectors must sum to the size of the input matrix. Consider the following code:

```
>> M=[1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15; 16 17 18 19 20; 21 22 23 24 25];
>> size(M)
```

```
ans =
     5     5

>> C = mat2cell(M,[1 2 2],[2 3]) % Break up matrix into six pieces
C =
    [1x2 double] [1x3 double]
    [2x2 double] [2x3 double]
    [2x2 double] [2x3 double]
>> celldisp(C)
C{1,1} =
     1     2
C{2,1} =
     6     7
    11    12
C{3,1} =
    16    17
    21    22
C{1,2} =
     3     4     5
C{2,2} =
     8     9    10
    13    14    15
C{3,2} =
    18    19    20
    23    24    25
```

The corresponding function is `cell2mat`:

```
>> N = cell2mat(C)
N =
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
    16    17    18    19    20
    21    22    23    24    25
```

8.6 CELL ARRAYS OF STRINGS

Although a rigorous discussion of character strings doesn't occur until the next chapter, the use of character strings and their storage in cell arrays are described in this section. If the concept of a character string is foreign to you, read the next chapter and then come back to this material.

In MATLAB, character strings are formed by enclosing characters within quotes. For example, in the code

```
>> s = 'Rarely is the question asked: is our children learning?'
s =
Rarely is the question asked: is our children learning?
```

the variable s contains a string of characters. In many applications, groups of character strings are associated with each other. When this occurs, it is convenient to group them into a cell array rather than store them in individual variables. In the code

```
>> cs = {'My answer is bring them on.'
s
'I understand small business growth. I was one.'}
cs =
    'My answer is bring them on.'
    'Rarely is the question asked: is our children learning?'
    'I understand small business growth. I was one.'

>> size(cs) % a column cell array
ans =
    3   1
>> iscell(cs) % yes, it is a cell array
ans =
    1
```

the cell array cs has three cells, each containing a character string. In MATLAB this is simply called a ***cell array of strings***. Because cell arrays are commonly used to store sets of character strings, MATLAB provides several functions to support them, like the one in the following example:

```
>> iscellstr(cs)
ans =
 1
```

The function `iscellstr` returns logical True (1) if all the cells in its cell array argument contain character strings; otherwise, it returns logical False (0).

Before cell arrays were introduced in MATLAB 5, groups of character strings were stored in character-string arrays—2-D arrays just like the numerical arrays discussed earlier—with each string occupying a separate row in the array. Each character in the ***string array*** occupies its own location and is indexed just like a numerical array.

MATLAB provides functions to convert a cell array of strings to a string array and vice versa:

```
>> cs % recall previous cell array of strings
cs =
    'My answer is bring them on.'
    'Rarely is the question asked: is our children learning?'
    'I understand small business growth. I was one.'
>> sa = char(cs) % convert to a string array
sa =
My answer is bring them on.
Rarely is the question asked: is our children learning?
I understand small business growth. I was one.
>> ischar(sa) % True for string array
ans =
    1
>> iscell(sa) % True for cell array
ans =
    0
>> size(sa) % size of string array
ans =
    3    34
>> size(cs) % size of cell array
ans =
    3     1
>> cst = cellstr(sa) % convert back to cell array
cst =
    'My answer is bring them on.'
    'Rarely is the question asked: is our children learning?'
    'I understand small business growth. I was one.'
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

```
>> iscell(cst)          % True for cell array
ans =
    1
>> isequal(cs,cst)      % True for equal variables
ans =
    1
>> isequal(cs,sa)       % cell array not equal to string array
ans =
    0
```

So, the MATLAB functions `char` and `cellstr` are inverses of each other. In addition, since a string array must have the same number of columns in each row, blank spaces are added to rows, as necessary, to make the string array rectangular.

8.7 STRUCTURE CREATION

Structures are like cell arrays in that they allow you to group collections of dissimilar data into a single variable. However, instead of addressing elements by number, structure elements are addressed by names called ***fields***. Like cell arrays, structures can have any number of dimensions, but a simple scalar or vector array is the most common type.

Whereas cell arrays use curly braces to access data, structures use dot notation to access data in fields. Creating a structure can be as simple as assigning data to individual fields. For example, in the code

```
>> circle.radius = 2.5;          % semicolon, no display
>> circle.center = [0 1];
>> circle.linestyle = '--';
>> circle.color = 'red'          % no semicolon, so display
circle =
       radius: 2.5
       center: [0 1]
    linestyle: '--'
        color: 'red'
```

the data are stored in a structure variable called `circle`. The case-sensitive fields are entitled `radius`, `center`, `linestyle`, and `color`. Structure field names have the same restrictions as variable names: They can contain up to 63 characters, and they must begin with a letter. For example, in the statement

Exclusively for The University of Melbourne


```
>> size(circle)
ans =
     1     1
>> whos
  Name            Size          Bytes    Class Attributes

  ans             1x2              16    double
  circle          1x1             530    struct
```

circle is a scalar structure, since size says that it is a 1-by-1 array. The whos command shows its size, the fact that it is a structure array, and the fact that it uses 530 bytes of memory. If there is more than one circle, it can be stored as a second element in the circle variable. For example, in the code

```
>> circle(2).radius = 3.4;
>> circle(2).color = 'green';
>> circle(2).linestyle = ':';
>> circle(2).center = [2.3 -1.2]
circle =
1x2 struct array with fields:
    radius
    center
    linestyle
    color
```

circle is a structure array having two elements. The (2) appears immediately after the variable name, because it is the variable that is having an element added to it. The .fieldname suffix identifies the field where data is to be placed. Note that the structure fields are filled in a different order this time and that the size of the data differs between the two elements. That is, the color fields are 'red' and 'green'. There are no restrictions on what can be placed in fields from one array element to the next. For instance, in the MATLAB statement

```
>> circle(2).radius = 'sqrt(2)'
circle =
1x2 struct array with fields:
    radius
    center
    linestyle
    color
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

```
>> circle.radius % display radius contents
ans =
          2.5000
ans =
sqrt(2)
```

`circle(1).radius` holds numerical data and `circle(2).radius` holds a character string.

If the value of structures is not apparent to you, consider how the data would be stored without structures:

```
>> Cradius = [2.5 3.4]; % ignore sqrt(2) change above
>> Ccenter = [0 1 ; 2.3 -1.2];
>> Clinestyle = {'--' ':'}; % cell array of strings
>> Ccolor = {'red' 'green'};
```

Now, rather than having a single variable storing the information for two circles, there are four variables that must be indexed properly to extract data for each circle. It's easy to add another circle to the structure, but more cumbersome to add them to the four variables. Now, consider the following example:

```
>> circle(3).radius = 25.4;
>> circle(3).center = [-1 0];
>> circle(3).linestyle = '-.';
>> circle(3).color = 'blue'  % third circle added
circle =
1x3 struct array with fields:
    radius
    center
    linestyle
    color
>> Cradius(3) = 25.4
Cradius =
          2.5000       3.4000       25.4000
>> Ccenter(3,:) = [-1 0]
Ccenter =
               0       1.0000
          2.3000      -1.2000
         -1.0000            0
```

```
>> Cradius(3) = 25.4
Cradius =
          2.5000        3.4000        25.4000

>> Clinestyle{3} = '-.'
Clinestyle =
    '--'    ':'    '-.'
>> Ccolor(3) = {'blue'}
Ccolor =
  'red'    'green'    'blue'
```

The clarity provided by structures should be readily apparent. In addition, consider passing the circle data to a function. For example, as a structure, the data are passed simply as

```
myfunc(circle)
```

whereas the other approach requires

```
myfunc(Cradius,Ccenter,Clinestyle,Ccolor)
```

Suppose that, at some later date, you wanted to add another field to `circle`:

```
>> circle(1).filled = 'yes'
circle =
1x3 struct array with fields:
    radius
    center
    linestyle
    color
    filled
>> circle.filled % display all .filled fields
ans =
yes
ans =
     []
ans =
     []
```

Now, all of the elements of `circle` have the field `filled`. Those not assigned by default contain the empty array `[]`. The other `filled` fields are easily assigned:

```
>> circle(2).filled = 'no';
>> circle(3).filled = 'yes';
>> circle.filled
ans =
yes
ans =
no
ans =
yes
```

When structure creation by direct assignment isn't possible, MATLAB provides the function `struct`. For example, to re-create the previous structure, the following code is used:

```
>> values1 = {2.5 'sqrt(2)' 25.4}; % cell arrays with field data
>> values2 = {[0 1] [2.3 -1.2] [-1 0]};
>> values3 = {'--' ':' '-.'};
>> values4 = {'red' 'green' 'blue'};
>> values5 = {'yes' 'no' 'yes'};
>> CIRCLE = struct('radius',values1,'center',values2,...
            'linestyle',values3,'color',values4,'filled',values5)
CIRCLE =
1x3 struct array with fields:
    radius
    center
    linestyle
    color
    filled
>> isequal(circle,CIRCLE) % True since structures are equal
ans =
     1
```

usively for The University of Melbourne

8.8 STRUCTURE MANIPULATION

Structures are arrays and therefore can be combined and indexed like numerical arrays and cell arrays. When combining structure arrays, the only restriction is that the arrays combined must share the same fields. Thus, in the code

```
>> square.width = 5; % a new structure
>> square.height = 14;
>> square.center = zeros(1,2);
>> square.rotation = pi/4
square =
       width: 5
      height: 14
      center: [0 0]
    rotation: 0.7854
>> A = [circle CIRCLE]
A =
1x6 struct array with fields:
    radius
    center
    linestyle
    color
    filled
>> B = [circle square]
??? Error using ==> horzcat
CAT arguments are not consistent in structure field number.
```

the structures `circle` and `CIRCLE` are both 1 by 3 and share the same fields; hence, concatenating them produces a 1-by-6 structure array. However, the structures `square` and `circle` do not have the same fields and therefore cannot be concatenated.

It is also possible to address a subarray of a structure. For example, in the code

```
>> C = [circle(1:2) CIRCLE(3)]
C =
1x3 struct array with fields:
    radius
    center
    linestyle
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

```
    color
    filled
>> isequal(C,circle) % True since equal
ans =
     1
```

C and `circle` are equal because `CIRCLE(3) = circle(3)`. Once again, basic array addressing and concatenation apply to structure arrays as well. For concatena[redacted] field names must match exactly.

Although they are not as useful in this case, structures can also be manipula[redacted] with the functions `reshape` and `repmat`:

```
>> Aa = reshape(A,3,2)
Aa =
3x2 struct array with fields:
    radius
    center
    linestyle
    color
    filled
>> Aaa = reshape(A,1,2,3)
Aaa =
1x2x3 struct array with fields:
    radius
    center
    linestyle
    color
    filled
```

Since there is seldom a practical reason to have anything other than a structu[redacted] array with vector orientation, `reshape` is seldom needed or used with structur[redacted] However, it is often convenient to create a structure array with default data in all fields of all array elements. The function `repmat` performs this task with ease:

```
>> S = repmat(square,3,1)
S =
3x1 struct array with fields:
    width
    height
```

```
    center
    rotation
>> S.width % look at all width fields
ans =
     5
ans =
     5
ans =
     5
```

All three elements of the structure S contain the data originally assigned to the structure square. At this point, the structure fields can be modified, as needed, to describe different squares.

8.9 RETRIEVING STRUCTURE CONTENT

When you know the names of the fields associated with a structure array, retrieving the data in a particular structure element and field simply requires identifying them. For example, in the code

```
>> rad2 = circle(2).radius
rad2 =
sqrt(2)

>> circle(1).radius
ans =
          2.5000

>> area1 = pi*circle(1).radius^2
area1 =
         19.6350
```

circle(1).radius identifies the value 2.5, which is used to compute the area of the first circle.

When the contents of a field are an array, it is also possible to retrieve a subset of that field by appending an array index to the structure request:

```
>> circle(1).filled            % the entire field
ans =
yes
```

```
>> circle(1).filled(1)          % first element of field
ans =
y
>> circle(1).filled(2:end)      % rest of field
ans =
es
```

As is true for cell arrays, retrieving the contents of more than one structu array element and field cannot be accomplished by direct addressing. For exampl the command

```
>> col = circle.color
col =
red
```

attempted to extract three pieces of data and store them in one variable, but only stored one. However, MATLAB solves this problem the same way it does for cell arrays by using comma-separated lists.

When the structure field to be accessed is stored in a character string MATLAB provides *dynamic addressing* to gain access to the data:

```
>> fldstr = 'color';   % store desired field in variable

>> circle.(fldstr)     % get all color fields
ans =
red
ans =
green
ans =
blue

>> fldstr = 'radius';
>> area=pi*circle(1).(fldstr)^2 % compute area of first circle
area =
      19.6350
```

Dynamic addressing is similar to using subscripts to identify specific elements in an array. Both methods use parentheses to hold the chosen elements. For exampl `A(r,c)` uses the information in variables `r` and `c` to identify the desired rows and

ENGI
Pe
...ing exclusively for The University of Melbourne

columns of array A, whereas `circle.(fldstr)` uses the information in variable `fldstr` to identify the desired field in the structure `circle`.

8.10 COMMA-SEPARATED LISTS (AGAIN)

To extract the contents of more than one structure array element at a time, MATLAB provides comma–separated list syntax. This syntax applies in any place where variables or constants appear in a list separated by commas. For example, it appears in array construction:

```
>> a = ones(2,3);
>> b = zeros(2,1);
>> c = (3:4)';
>> d = [a,b,c] % same as [a b c]
d =
 1     1     1     0     3
 1     1     1     0     4
```

It also appears in function input and output argument lists:

```
>> d = cat(2,a,b,c)
d =
    1     1     1     0     3
    1     1     1     0     4
>> [m,n] = size(d)
m =
 2
n =
 5
```

Building on this idea, comma–separated list syntax is implemented as follows: Placing a structure array with appended field name in locations where comma-separated lists appear causes MATLAB to extract the contents of the addressed fields and place them sequentially, separated by commas. In the code

```
>> cent = cat(1,circle.center) % comma-separated list syntax
cent =
                 0       1.0000
            2.3000      -1.2000
           -1.0000            0
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

```
>> cent = cat(1,circle(1).center,circle(2).center,circle(3).center)
cent =
                 0       1.0000
            2.3000      -1.2000
           -1.0000            0
>> some = cat(1,circle(2:end).center)
some =
            2.3000      -1.2000
           -1.0000            0
```

the `cat` function concatenates the circle centers into rows of the numerical array `cent`. According to comma–separated list syntax, the first two statements in the preceding example are identical. The third statement shows that you can index and extract a subarray as well.

Since the `color` fields of the structure `circle` are character strings of different lengths, they cannot be extracted into a string array, but they can be extracted into a cell array, as in the following example:

```
>> circle.color
ans =
red
ans =
green
ans =
blue

>> col = cat(1,circle.color) % elements have different lengths
??? Error using ==> cat
CAT arguments dimensions are not consistent.
>> col = [circle.color] % no error but not much use!
col =
redgreenblue
>> col = {circle.color} % cell array of strings
col =
    'red' 'green' 'blue'

>> col = char(col) % if needed, convert to string array
col =
```

```
red
green
blue
```

MATLAB does not provide tools for extracting all the fields of a single structure array element. For instance, there is no way to retrieve the `radius`, `center`, `linestyle`, `color`, and `filled` fields of the first circle in one statement. In a sense, there is no need to have such a function, because each of these fields can be directly addressed and used in computations, such as `area1 = pi*circle(1).radius^2`.

Given comma–separated list syntax, you can extract the contents of numerous structure elements into separate variables, as in the following example:

```
>> [c1,c2,c3] = circle.color % get all colors
c1 =
red
c2 =
green
c3 =
blue
>> [rad1,rad3] = circle ([1 3]).radius % 1st and 3rd radius
rad1 =
          2.5000
rad3 =
         25.4000
```

Because the output of `deal` is also a comma-separated list, `deal` can be used to assign the contents of multiple structure array elements, with a single field in one statement. For example, consider the following code:

```
>> [circle.radius] = deal(5,14,83)
circle =
1x3 struct array with fields:
    radius
    center
    linestyle
    color
    filled
```

```
>> circle.radius % confirm assignments
ans =
     5
ans =
    14
ans =
    83
>> [triangle(:).type] = deal('right','isosceles','unknown')
??? Error using ==> deal at 38
The number of outputs should match the number of inputs.
>> [triangle(1:3).type] = deal('right','isosceles','unknown')
triangle =
1x3 struct array with fields:
    type
>> triangle.type
ans =
right
ans =
isosceles
ans =
unknown
```

In the first statement, the structure `circle` already existed and has three elements. Therefore, the output argument was expanded into three elements. In the second statement containing `deal`, the structure `triangle` didn't exist. Because it wasn't possible to determine how many elements to create in a comma-separated list, an error was returned. However, in the last statement containing `deal`, the number of elements was given explicitly, and the `type` property of the newly created structure `triangle` was populated with the given data.

8.11 STRUCTURE FUNCTIONS

In the *Command* window, it is easy to identify the field names of a given structure by simply entering the structure name at the MATLAB prompt:

```
>> circle
circle =
```

Exclusively for The University of Melbourne


```
1x3 struct array with fields:
    radius
    center
    linestyle
    color
    filled
>> square
square =
        width: 5
       height: 14
       center: [0 0]
     rotation: 0.7854
```

When a structure is passed to a function—for example, `myfunc(circle)`—the function internally must know or have some way to obtain the field names of the structure. (Writing functions in MATLAB is covered in Chapter 12.) In MATLAB, the function `fieldnames` provides this information:

```
>> fieldnames(circle)
ans =
    'radius'
    'center'
    'linestyle'
    'color'
    'filled'
```

The output of `fieldnames` is a cell array of strings identifying the fields that are associated with the input structure.

It is also possible to guess the field names and ask whether they exist by using the logical function `isfield`:

```
>> isfield(circle,'color') % True
ans =
     1
>> isfield(circle,'width') % False
ans =
     0
```

Hanselman, D. and Littlefield, B. (2012). Cell Arrays and Structures. In *Mastering MATLAB®*, (pp. 129-164). Upper Saddle River, NJ: Pearson Education Inc.

If you don't know whether a variable is a structure or not, the functions `class` and `isstruct` are helpful, as the following example shows:

```
>> class(square)       % ask for the class of variable square
ans =
struct
>> isstruct(circle)    % True for structures
ans =
     1
>> d = pi;
>> isstruct(d) % False for doubles
ans =
     0
```

When the field names of a structure are known, the function `rmfield` allows you to remove one or more fields from the structure. For example, the code

```
>> fnames = fieldnames(circle)
fnames =
       'radius'
       'center'
       'linestyle'
       'color'
       'filled'
```

stores in `fnames` the field names of `circle` in a cell array. Arbitrarily choosing the last field name, we can remove the `filled` field from the structure by calling `rmfield`. This code:

```
>> circle2 = rmfield(circle,fnames{5})
circle2 =
1x3 struct array with fields:
    radius
    center
    linestyle
    color
```

removes the field `filled` from `circle` and assigns the result to a new structure `circle2`. Similarly,

```
>> circle3 = rmfield(circle,fnames(1:3))
circle3 =
1x3 struct array with fields:
    'color'
    'filled'
```

removes the fields `radius`, `center`, and `linestyle` from `circle`, and assigns the result to a new structure `circle3`.

In some cases, it is beneficial to rearrange the order in which the field names of a structure are presented. The function `orderfields` performs this task and places the fields in ASCII order:

```
>> circleA = orderfields(circle)
circleA =
1x3 struct array with fields:
    center
    color
    filled
    linestyle
    radius
```

Other orderings are possible as well, as in the following code:

```
>> circleB = orderfields(circleA,CIRCLE) % match fields of structure CIRCLE
circleB =
1x3 struct array with fields:
    radius
    center
    linestyle
    color
    filled
>> circleC = orderfields(circle,[2 5 1 4 3]) % provide permutation vector
circleC =
1x3 struct array with fields:
    center
    filled
    radius
```

```
    color
    linestyle
>> circleD = orderfields(circle,fnames(end.-1.1)) % reverse original order
circleD =
1x3 struct array with fields:
    filled
    color
    linestyle
    center
    radius
```

Finally, given the similarity between cell arrays and structures and applying the post office box analogy, it's not hard to believe that MATLAB provides the functions `cell2struct` and `struct2cell` to convert cell arrays to structures and back in an organized fashion. (More detailed explanations of these functions can be found in the online documentation.)

8.12 SUMMARY

MATLAB supports arrays with an unlimited number of dimensions. Cell arrays and structures can store any array type, including cell arrays and structures. So, it is possible to have a cell in a cell array that contains a structure with a field that contains another structure that has a field containing a cell array, of which one cell contains another cell array. Needless to say, there comes a point where the power of cell arrays and structures becomes indecipherable and of little practical use. Thus, to end this chapter, try to decipher the following legal MATLAB statements:

```
>> one(2).three(4).five = {circle}
one =
1x2 struct array with fields:
    three
>> test = {{{circle}}}
test =
    {1x1 cell}
```

How many structures are involved in the first statement? What is the total number of structure elements, including empty arrays, created by the statement? Can the structure `circle` be extracted from the second statement with a single MATLAB statement?

9

Character Strings

MATLAB's true power is in its ability to crunch numbers. However, there are times when it is desirable to manipulate text, such as when putting labels and titles on plots. In MATLAB, text is referred to as character strings, or simply strings. Character strings represent another variable class or data type in MATLAB.

9.1 STRING CONSTRUCTION

Character strings in MATLAB are special numerical arrays of ASCII values that are displayed as their character-string representation, as in the following code:

```
>> t = 'How about this character string?'
t =
How about this character string?
>> size(t)
ans =
     1    32
>> whos
  Name          Size          Bytes     Class Attributes

  ans           1x2              16     double
  t             1x32             64     char
```


Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

```
    color
    linestyle
>> circleD = orderfields(circle,fnames(end:-1:1)) % reverse original order
circleD =
1x3 struct array with fields:
    filled
    color
    linestyle
    center
    radius
```

Finally, given the similarity between cell arrays and structures and applying the post office box analogy, it's not hard to believe that MATLAB provides the functions `cell2struct` and `struct2cell` to convert cell arrays to structures and back in an organized fashion. (More detailed explanations of these functions can be found in the online documentation.)

8.12 SUMMARY

MATLAB supports arrays with an unlimited number of dimensions. Cell arrays and structures can store any array type, including cell arrays and structures. So, it is possible to have a cell in a cell array that contains a structure with a field that contains another structure that has a field containing a cell array, of which one cell contains another cell array. Needless to say, there comes a point where the power of cell arrays and structures becomes indecipherable and of little practical use. Thus, to end this chapter, try to decipher the following legal MATLAB statements:

```
>> one(2).three(4).five = {circle}
one =
1x2 struct array with fields:
    three
>> test = {{{circle}}}
test =
    {1x1 cell}
```

How many structures are involved in the first statement? What is the total number of structure elements, including empty arrays, created by the statement? Can the structure `circle` be extracted from the second statement with a single MATLAB statement?

9

Character Strings

MATLAB's true power is in its ability to crunch numbers. However, there are times when it is desirable to manipulate text, such as when putting labels and titles on plots. In MATLAB, text is referred to as character strings, or simply strings. Character strings represent another variable class or data type in MATLAB.

9.1 STRING CONSTRUCTION

Character strings in MATLAB are special numerical arrays of ASCII values that are displayed as their character-string representation, as in the following code:

```
>> t = 'How about this character string?'
t =
How about this character string?
>> size(t)
ans =
     1    32
>> whos
  Name          Size          Bytes     Class Attributes

  ans           1x2              16     double
  t             1x32             64     char
```


Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

A character string is simply text surrounded by single quotes. Each character in a string is one element in an array that requires 2 bytes per character for storage, which is different from the 8 bytes per element that is required for numerical or double arrays.

To see the underlying ASCII representation of a character string, you need only perform some arithmetic operation on the string or use the dedicated function `double`, as in the following code:

```
>> u = double(t)
u =
  Columns 1 through 12
      72   111   119    32    97    98   111   117   116    32   116   104
  Columns 13 through 24
     105   115    32    99   104    97   114    97    99   116   101   114
  Columns 25 through 32
      32   115   116   114   105   110   103    63


>> abs(t)
ans =
  Columns 1 through 12
      72   111   119    32    97    98   111   117   116    32   116   104
  Columns 13 through 24
     105   115    32    99   104    97   114    97    99   116   101   114
  Columns 25 through 32
      32   115   116   114   105   110   103    63
```

The function `char` performs the inverse transformation:

```
>> char(u)
ans =
How about this character string?
```

Numerical values less than 0 produce a warning message when converted to character; values greater than 255 simply address characters in the font beyond `char(255)`:

```
>> a = double('a')
a =
    97
```

Exclusively for The University of Melbourne


```
>> char(a)
ans =
a
>> char(a+256) % adding 256 does change the result
ans =
š
>> char(a-256) % negative value produces a blank character
Warning: Out of range or non-integer values truncated during conversion to
character.
ans =
```

Since strings are arrays, they can be manipulated with all the array-manipulation tools available in MATLAB. For example, in the code

```
>> u = t(16:24)
u =
character
```

strings are addressed just as arrays are. Here, elements 16 through 24 contain the word `character`.

```
>> u = t(24:-1:16)
u =
retcarahc
```

The result here is the word `character` spelled backwards.

Using the transpose operator changes the word `character` to a column:

```
>> u = t(16:24)'
u =
c
h
a
r
a
c
t
```

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

```
e
r
```

Single quotes within a character string are symbolized by two consecutive quotes:

```
>> v = 'I can''t find the manual!'
v =
I can't find the manual!
```

Also, string concatenation follows directly from array concatenation:

```
>> u = 'If a woodchuck could chuck wood,';
>> v = 'how much wood could a woodchuck chuck?';


>> w = [u v]
w =
If a woodchuck could chuck wood, how much wood could a woodchuck chuck?
```

The function `disp` allows you to display a string without printing its variable name:

```
>> disp(u)
If a woodchuck could chuck wood,
```

Note that the `u =` statement is suppressed. This feature is useful for displaying help text within a script file.

Like other arrays, character strings can have multiple rows, but each row must have an equal number of columns. Therefore, blanks are explicitly required to make all rows the same length, as in the following code:

```
>> v = ['Character strings having more than'
        'one row must have the same number '
        'of columns just like arrays!      ']
v =
Character strings having more than
one row must have the same number
of columns just like arrays!
```

The functions `char` and `strvcat` create multiple-row string arrays from individual strings of varying lengths:

```
>> legends = char('Wilt','Russel','Kareem','Bird','Magic','Jordan')
legends =
Wilt
Russel
Kareem
Bird
Magic
Jordan
>> legends = strvcat('Wilt','Russel','Kareem','Bird','Magic','Jordan')
legends =
Wilt
Russel
Kareem
Bird
Magic
Jordan
>> size(legends)
ans =
     6     6
```

The only difference between `char` and `strvcat` is that `strvcat` ignores empty string inputs, whereas `char` inserts blank rows for empty strings, as in the following example:

```
>> char('one','','two','three')
ans =
one

two
three

>> strvcat('one','','two','three')
ans =
one
two
three
```

Horizontal concatenation of string arrays that have the same number of rows is accomplished by the function `strcat`. Padded blanks are ignored. The following code is illustrative:

```
>> a = char('apples','bananas')
a =
apples
bananas
>> b = char('oranges','grapefruit')
b =
oranges
grapefruit
>> strcat(a,b)
ans =
applesoranges
bananasgrapefruit
```

Once a string array is created with padded blanks, the function `deblank` is useful for eliminating the extra blanks from individual rows extracted from the array:

```
>> c = legends(4,:)
c =
Bird

>> size(c)
ans =
     1     6

>> c = deblank(legends(4,:))
c =
Bird
>> size(c)
ans =
     1     4
```

g exclusively for The University of Melbourne

9.2 NUMBERS TO STRINGS TO NUMBERS

There are numerous contexts in which it is desirable to convert numerical results to character strings and to extract numerical data from character strings. MATLAB provides the functions `int2str`, `num2str`, `mat2str`, `sprintf`, and `fprintf` for converting numerical results to character strings. Examples of the first three functions are in the following code:

```
>> int2str(eye(3)) % convert integer arrays
ans =
1  0  0
0  1  0
0  0  1
>> size(ans) % it's a character array, not a numerical matrix
ans =
     3      7

>> num2str(rand(2,4)) % convert noninteger arrays
ans =
0.95013    0.60684    0.8913    0.45647
0.23114    0.48598    0.7621    0.01850
>> size(ans) % again it is a character array
ans =

     2     43

>> mat2str(pi*eye(2)) % convert to MATLAB input syntax form!
ans =
[3.14159265358979 0;0 3.14159265358979]
>> size(ans)
ans =
     1     39
```

The following code illustrates the two functions `sprintf` and `fprintf`:

```
>> fprintf('%.4g\n',sqrt(2)) % display in Command window
1.414
```

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

```
>> sprintf('%.4g',sqrt(2)) % create character string
ans =
1.414

>> size(ans)
ans =
    1     5
```

sprintf and fprintf are general-purpose conversion functions that closely resemble their ANSI C language counterparts. As a result, these two functions offer the most flexibility. Normally fprintf is used to convert numerical results to ASCII format and append the converted results to a data file. However, if no file identifier is provided as the first argument to fprintf or if a file identifier of one is used, the resulting output is displayed in the *Command* window. The functions sprintf and fprintf are identical, except that sprintf simply creates a character array that can be displayed, passed to a function, or modified like any other character array. Because sprintf and fprintf are nearly identical, consider the usage of sprintf in the following example:

```
>> radius = sqrt(2);
>> area = pi * radius^2;

>>s = sprintf('A circle of radius %.5g has an area of %.5g.',radius,area)
s =
A circle of radius 1.4142 has an area of 6.2832.
```

Here, %.5g, the format specification for the variable radius, indicates that five significant digits in general-conversion format are desired. The most common usage of sprintf is to create a character string for the purposes of annotating a graph, displaying numerical values in a graphical user interface, or creating a sequence of data file names. A rudimentary example of this last usage is as follows:

```
>> i = 3;
>> fname = sprintf('mydata%.0f.dat',i)
fname =
mydata3.dat
```

In the past, the functions int2str and num2str were nothing more than a simple call to sprintf, with %.0f and %.4g as format specifiers, respectively. In MATLAB 5, int2str and num2str were enhanced to work with numerical arrays. As a result of these functions' former simplicity, in prior versions of MATLAB it was common to use the following code:

```
s = ['A circle of radius' num2str(radius)' has an area of' ...
num2str(area) '.']
s =
A circle of radius 1.4142 has an area of 6.2832.
```

Even though we obtain the same result when using `int2str` and `num2str` as we do when using `sprintf`, these functions require more computational effort, are more prone to typographical errors (such as missing spaces or single quotes), and require more effort to read. As a result, it is suggested that usage of `int2str` and `num2str` be limited to the conversion of arrays, as illustrated earlier in this section. In almost all other cases, it is more productive to use `sprintf` directly.

The help text for `sprintf` concisely describes its use:

```
>> help sprintf
SPRINTF Write formatted data to string.
   STR = SPRINTF(FORMAT, A,...) applies the FORMAT to all elements of
   array A and any additional array arguments in column order, and returns
   the results to string STR.

   [STR, ERRMSG] = SPRINTF(FORMAT, A,...) returns an error message when
   the operation is unsuccessful. Otherwise, ERRMSG is empty.

   SPRINTF is the same as FPRINTF except that it returns the data in a
   MATLAB string rather than writing to a file.

   FORMAT is a string that describes the format of the output fields, and
   can include combinations of the following:

   * Conversion specifications, which include a % character, a
     conversion character (such as d, i, o, u, x, f, e, g, c, or s),
     and optional flags, width, and precision fields. For more
     details, type "doc sprintf" at the command prompt.
   * Literal text to print.
   * Escape characters, including:
       \b    Backspace              '' Single quotation mark
       \f    Form feed              %% Percent character
       \n    New line               \\ Backslash
       \r    Carriage return        \xN Hexadecimal number N
       \t    Horizontal tab         \N Octal number N%
```

where \n is a line termination character on all platforms.

Notes:

If you apply an integer or string conversion to a numeric value that contains a fraction, MATLAB overrides the specified conversion, and uses %e.

Numeric conversions print only the real component of complex numbers.

Examples

```
sprintf('%0.5g',(1+sqrt(5))/2)          % 1.618
sprintf('%0.5g',1/eps)                  % 4.5036e+15
sprintf('%15.5f',1/eps)                 % 4503599627370496.00000
sprintf('%d',round(pi))                 % 3
sprintf('%s','hello')                   % hello
sprintf('The array is %dx%d.',2,3)      % The array is 2x3.
```

See also fprintf, sscanf, num2str, int2str, char.

Reference page in Help browser

doc sprintf

The following table shows how pi is displayed under a variety of conversion specifications:

Command	Result
sprintf('%.0e',pi)	3e+00
sprintf('%.1e',pi)	3.1e+00
sprintf('%.3e',pi)	3.142e+00
sprintf('%.5e',pi)	3.14159e+00
sprintf('%.10e',pi)	3.1415926536e+00
sprintf('%.0f',pi)	3
sprintf('%.1f',pi)	3.1
sprintf('%.3f',pi)	3.142
sprintf('%.5f',pi)	3.14159
sprintf('%.10f',pi)	3.1415926536
sprintf('%.0g',pi)	3

sprintf('%.1g',pi)	3
sprintf('%.3g',pi)	3.14
sprintf('%.5g',pi)	3.1416
sprintf('%.10g',pi)	3.141592654
sprintf('%8.0g',pi)	3
sprintf('%8.1g',pi)	3
sprintf('%8.3g',pi)	3.14
sprintf('%8.5g',pi)	3.1416
sprintf('%8.10g',pi)	3.141592654

In the preceding table, the format specifier e signifies exponential notation, f signifies fixed-point notation, and g signifies the use of e or f, whichever is shorter. Note that for the e and f formats, the number to the right of the decimal point indicates how many digits to the right of the decimal point are to be displayed. On the other hand, in the g format, the number to the right of the decimal specifies the total number of digits that are to be displayed. In addition, note that in the last five entries, a width of eight characters is specified for the result, and the result is right justified. In the very last case, the 8 is ignored because more than eight digits were specified.

Although it is not as common, sometimes it is necessary to convert or extract a numerical value from a character string. The MATLAB functions str2num, sscanf, and str2double provide this capability:

```
>> s = num2str(pi*eye(2)) % create string data
s =
3.1416          0
     0     3.1416

>> ischar(s) % True for string
ans =
     1

>> m = str2num(s) % convert string to number
m =
      3.1416          0
           0     3.1416
```

```
>> isdouble(m) % Oops, this function doesn't exist
??? Undefined function or method 'isdouble' for input arguments of
type 'double'.

>> isnumeric(m) % True for numbers
ans =
     1
>> isfloat(m) % True for floating point numbers
ans =
     1
>> pi*eye(2) - m % accuracy is lost
ans =
-7.3464e-06            0
          0  -7.3464e-06
```

The function `str2num` can contain expressions, but not variables, in the workspace:

```
>> x = pi; % create a variable

>> ss = '[sqrt(2) j; exp(1) 2*pi-x]' % string with variable x
ss =
[sqrt(2) j; exp(1) 2*pi-x]

>> str2num(ss) % conversion fails because of x
ans =
     []

>> ss = '[sqrt(2) j; exp(1) 2*pi-6]' % replace x with 6
ss =
[sqrt(2) j; exp(1) 2*pi-6]

>> str2num(ss) % now it works
ans =
 1.4142                        0 +         1i
 2.7183                  0.28319
```

```
>> class(ans) % yes, its a double
ans =
double
```

The function `sscanf` is the counterpart to `sprintf`. The function `sscanf` reads data from a string under format control:

```
>> v = version       % get MATLAB version as a string
v =
7.11.0.584 (R2010b)

>> sscanf(v,'%f')    % get floating point numbers
ans =
          7.11
             0
         0.584

>> sscanf(v,'%f',1) % get just one floating point number
ans =
    7.11

>> sscanf(v,'%d')    % get an integer
ans =
    7

>> sscanf(v,'%s')    % get a string
ans =
7.11.0.584(R2010b)
```

You can specify the format under which `sscanf` operates; this capability makes `sscanf` a very powerful and flexible function. (See the online help text for `sscanf` for more thorough information about the capabilities of this function.)

When the conversion to a single double-precision value is required, the function `str2double` is useful. While the function `str2num` performs this task as well, `str2double` is generally quicker because of its more limited scope:

```
>> str2double('Inf') % It does convert infinity
ans =
Inf
```

```
>> class(ans)
ans =
double

>> str2double('34.6 - 23.2j') % complex numbers work
ans =
       34.6 -        23.2i

>> str2double('pi') % variables and expressions don't work
ans =
 NaN
```

9.3 STRING EVALUATION

There are some applications in which it is convenient to evaluate a character string as if it were a MATLAB expression. This is a generalization of what `str2num` and `str2double` do. These functions are limited to extracting numerical values from strings. The MATLAB functions `eval` and `evalc` bring in the entire MATLAB interpreter to evaluate any string that conforms to MATLAB syntax. For example in the code

```
>> funs = char('ceil','fix','floor','round')
funs =
ceil
fix
floor
round
>> [deblank(funs(1,:)) '(pi)'] % display string to evaluate
ans =
ceil(pi)
>> f = eval([deblank(funs(1,:)) '(pi)'])
f =
     4
>> class(f) % data type of output is numeric double
ans =
double
>> fc = evalc([deblank(funs(1,:)) '(pi)']) % try evalc
```

```
fc =

ans =
     4
>> class(fc) % output of evalc is a character string
ans =
char
>> size(fc)
ans =
     1    13
```

the function `eval` uses the MATLAB command interpreter to evaluate a character-string input and returns the results into its output argument. The function `evalc` also evaluates an input string, but its output is the character-string representation of the output. In other words, it returns the results seen in the *Command* window.

If at all possible, for the most part, the use of `eval` and `evalc` should be avoided. Since these functions bring in the entire MATLAB interpreter to evaluate a string expression, they incur significant overhead. In addition, the functions cannot be compiled by the MATLAB compiler, which is a MATLAB add-on product that converts MATLAB code into executable C code.

9.4 STRING FUNCTIONS

MATLAB provides a variety of string-related functions, some of which have been discussed already. The following table briefly describes many of the string functions in MATLAB:

Function	Description
`char(S1,S2,...)`	Creates character array from strings or cell arrays
`double(S)`	Converts string to ASCII representation
`cellstr(S)`	Creates cell array of strings from character array
`blanks(n)`	Creates string of `n` blanks
`deblank(S)`	Removes trailing blanks
`eval(S), evalc(S)`	Evaluates string expression with MATLAB interpreter
`ischar(S)`	True for string array
`iscellstr(C)`	True for cell array of strings

Function	Description
`isletter(S)`	True for letters of the alphabet
`isspace(S)`	True for white-space characters
`isstrprop(S,'property')`	True for elements that have specified property
`strcat(S1,S2,...)`	Concatenates strings horizontally
`strvcat(S1,S2,...)`	Concatenates strings vertically, ignoring blanks
`strcmp(S1,S2)`	True if strings are equal
`strncmp(S1,S2,n)`	True if `n` characters of strings are equal
`strcmpi(S1,S2)`	True if strings are equal, ignoring case
`strncmpi(S1,S2,n)`	True if `n` characters of strings are equal, ignoring case
`strtrim(S1)`	Trims leading and trailing white space
`findstr(S1,S2)`	Finds shorter string within longer string (depreciate[redacted] use `strfind`)
`strfind(S1,S2)`	Finds `S2` string in `S1` string
`strjust(S1,type)`	Justifies string array left, right, or center
`strmatch(S1,S2)`	Finds indices of matching strings
`strrep(S1,S2,S3)`	Replaces occurrences of `S2` in `S1` with `S3`
`strtok(S1,D)`	Finds tokens in string given delimiters
`upper(S)`	Converts to uppercase
`lower(S)`	Converts to lowercase
`num2str(x)`	Converts number to string
`int2str(k)`	Converts integer to string
`mat2str(X)`	Converts matrix to string for `eval`
`str2double(S)`	Converts string to double-precision value
`str2num(S)`	Converts string array to numerical array
`sprintf(S)`	Creates string under format control
`sscanf(S)`	Reads string under format control
`textscan(S)`	Reads formatted data from a string

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

Now consider some examples of the usage of some of the functions that were defined in the preceding table. For instance, the function `strfind` returns the starting indices of one string within another:

```
>> b = 'Peter Piper picked a peck of pickled peppers';

>> strfind(b,' ')    % find indices of spaces
ans =
     6    12    19    21    26    29    37
>> strfind(b,'p')    % find the letter p
ans =
     9    13    22    30    38    40    41

>> find(b=='p')      % for single character searches find works too
ans =
     9    13    22    30    38    40    41
>> strfind(b,'cow')  % cow does not exist
ans =
     []
>> strfind(b,'pick') % find the string pick
ans =
    13    30
```

Note that `strfind` is case sensitive and returns the empty matrix when no match is found. `strfind` does not work on string arrays with multiple rows.

Tests on character strings include the following:

```
>> c = 'a2 : b_c'
c =
a2 : b_c
>> ischar(c) % it is a character string
ans =
     1
>> isletter(c) % where are the letters?
ans =
     1     0     0     0     0     1     0     1
```

```
>> isspace(c) % where are the spaces?
ans =
     0     0     1     0     1     0     0     0

>> isstrprop(c,'wspace') % same as above
ans =
     0     0     1     0     1     0     0     0

>> isstrprop(c,'digit') % True for digits
ans =
     0     1     0     0     0     0     0     0
```

To illustrate string comparison, consider the situation where a user types (perhaps into an editable text `uicontrol`) a string that must match, at least in part, one of a list of strings. The function `strmatch` provides this capability:

```
>> S = char('apple','banana','peach','mango','pineapple')
S =
apple
banana
peach
mango
pineapple
>> strmatch('pe',S) % pe is in 3rd row
ans =
     3
>> strmatch('p',S) % p is in 3rd and 5th rows
ans =
     3
     5
>> strmatch('banana',S) % banana is in 2nd row
ans =
     2
>> strmatch('Banana',S) % but Banana is nowhere
ans =
     Empty matrix: 0-by-1
```

Pearson Custom Publishing exclusively for The University of Melbourne


```
>> strmatch(lower('Banana'),S) % changing B to b finds banana
ans =
     2
```

9.5 CELL ARRAYS OF STRINGS

The fact that all rows in string arrays must have the same number of columns is sometimes cumbersome, especially when the nonblank portions vary significantly from row to row. This issue of cumbersomeness is eliminated by using cell arrays. All data forms can be placed in cell arrays, but their most frequent use is with character strings. A cell array is simply a data type that allows you to name a group of data of various sizes and types, as in the following example:

```
>> C = {'How';'about';'this for a';'cell array of strings?'}
C =
    'How'
    'about'
    'this for a'
    'cell array of strings?'

>> size(C)
ans =
     4     1
```

Note that curly brackets { } are used to create cell arrays and that the quotes around each string are displayed. In this example, the cell array C has four rows and one column. However, each element of the cell array contains a character string of different length.

Cell arrays are addressed just as other arrays are:

```
>> C(2:3)
ans =
    'about'
    'this for a'
>> C([4 3 2 1])
ans =
    'cell array of strings?'
    'this for a'
```

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

```
    'about'
    'How'
>> C(1)
ans =
    'How'
```

Here, the results are still cell arrays. That is, `C(indices)` addresses given cells, but not the contents of those cells. To retrieve the contents of a particular cell, use curly brackets, as in the following example:

```
>> s = C{4}
s =
cell array of strings?
>> size(s)
ans =
     1     22
```

To extract more than one cell, the function `deal` is useful:

```
>> [a,b,c,d] = deal(C{:})
a =
How
b =
about
c =
this for a
d =
cell array of strings?
```

Here, `C{:}` denotes all the cells as a list. That is, it's the same as

```
>> [a,b,c,d] = deal(C{1},C{2},C{3},C{4})
a =
How
b =
about
c =
this for a
```

```
d =
cell array of strings?
>> [a,b,c,d] = C{:} % since C{:} is a list, deal is optional
a =
How
b =
about
c =
this for a
d =
cell array of strings?
```

Partial cell array contents can also be dealt, as in the following code:

```
>> [a,b] = deal(C{2:2:4}) % get 2nd and 4th cell contents
a =
about
b =
cell array of strings?

>> [a,b] = C{2:2:4} % the right hand side is a list, so no deal
a =
about
b =
cell array of strings?
```

A subset of the contents of a particular cell array can be addressed as well:

```
>> C{4}(1:10) % 4th cell, elements 1 through 10
ans =
cell array
```

The multipurpose function `char` converts the contents of a cell array to a conventional string array, as in the following example:

```
>> s = char(C)
s =
```

```
How
about
this for a
cell array of strings?
>> size(s) % result is a standard string array with blanks
ans =
     4    22
>> ss = char(C(1:2)) % naturally you can convert subsets
ss =
How
about
>> size(ss) % result is a standard string array with blanks
ans =
     2     5
```

The inverse conversion is performed by the function `cellstr`, which deblanks the strings as well:

```
>> cellstr(s)
ans =
    'How'
    'about'
    'this for a'
    'cell array of strings?'
```

You can test whether a particular variable is a cell array of strings by using the function `iscellstr`:

```
>> iscellstr(C)   % True for cell arrays of strings
ans =
     1
>> ischar(C)     % True for string arrays not cell arrays of strings
ans =
     0
>> ischar(C{3})  % Contents of 3rd cell is a string array
ans =
     1
```

ENGINEERING SYSTEMS DESIGN 2
Pearson Custom Publishing exclusively for The University of Melbourne


```
>> iscellstr(C(3))        % but 3rd cell itself is a cell
ans =
     1
>> ischar(C(3))           % and not a string array
ans =
     0

>> class(C)               % get data type or class string
ans =
cell
>> class(s)               % get data type or class string
ans =
char
```

Most of the string functions in MATLAB work with either string arrays or cell arrays of strings. In particular, examples include the functions `deblank`, `strcat`, `strcmp`, `strncmp`, `strcmpi`, `strncmpi`, `strmatch`, and `strrep`. (Further information regarding cell arrays in general can be found in Chapter 8.)

9.6 SEARCHING USING REGULAR EXPRESSIONS

The standard string functions, like `strfind`, are useful to search for specific sequences of characters within strings and to replace specific character sequences. More generally, you may want to find such *patterns* in strings as repeated characters, all uppercase letters, any capitalized words, or all dollar amounts (i.e., strings of digits preceded by a dollar sign and containing a decimal point). MATLAB supports ***regular expressions***, which is a powerful tool to search for character strings within character strings. A regular expression is a formula for matching strings that follow some pattern. The Unix world has used regular expressions in many of its tools (such as `grep`, `awk`, `sed`, and `vi`) for years. Programmers who use Perl and other languages make extensive use of regular expressions. MATLAB's implementation of regular expressions is very comprehensive and has many features including some called *extended regular expressions*. And, if you are familiar with regular expressions, MATLAB's implementation will seem to you to follow naturally. If you are new to regular expressions, you can make use of the simpler features immediately and the more complex features later on.

Here are a few simple rules to get you started. A formula string, or *expression*, that describes the criteria for identifying matching portions of a character string is created. The expression consists of characters and optional modifiers that define the criteria to be used for the substring match. The simplest expression is a string of literal characters, as in the following code:

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

```
>> str = 'Peter Piper picked a peck of pickled peppers.'; % create a string

>> regexp(str,'pe')      % return the indices of substrings matching 'pe'
ans =
     9    22    38    41
```

Here, we found the letter combination `'pe'` in the words `Piper` and `peck` and twice in the word `peppers`.

Character classes are used to match a specific type of character—such as a letter, a number, or a white-space character—or to match a specific set of characters. The most useful character class is a period (`.`), which represents *any single character*. Another useful character class is a list of characters, or range of characters, within square brackets (`[]`) that matches *any member of the list or range of characters in the set*. For example, consider searching for a sequence of three characters consisting of a `p`, followed by any single character, followed by the letter `c` or the letter `r`:

```
>> regexpi(str,'p.[cr]')
ans =
     9    13    22    30    41
>> regexp(str,'p.[cr]','match')        % list the substring matches
ans =
     'per'    'pic'    'pec'    'pic'    'per'
```

You could also use a range of characters to find all uppercase letters in the string:

```
>> regexp(str,'[A-Z]')                 % match any uppercase letter
ans =
     1     7
```

The character expressions in the following table match a single character with the indicated characteristics:

Character Expression	Description and Usage
`.`	Any single character (including white space)
`[abcd35]`	Any single character in the set enclosed by square brackets `[]`
`[a-zA-Z]`	Any single character in the range `a-z` or `A-Z`; the dash `-` between characters defines an inclusive range of characters
`[^aeiou]`	Any single character NOT in the set of lowercase vowels; the `^` used as the first character in the set negates the sense of the set

\s	Any white-space character (space, tab, form feed, new line, or carriage return), equivalent to the set [\t\f\n\r]
\S	Any nonwhite-space character: [^ \t\f\n\r]
\w	Any "word character" (an upper- or lowercase letter, a digit, or an underscore): [a-zA-Z_0-9]
\W	Any character that is not a "word character": [^a-zA-Z_0-9]
\d	Any numeric digit: [0-9]
\D	Any nondigit character: [^0-9]
\xN or \x{N}	The character with hexadecimal value N
\oN or \o{N}	The character with octal value N
\a	The alarm, bell, or beep character: \o007 or \x07
\b	The backspace character: \o010 or \x08
\t	The horizontal tab character: \o011 or \x09
\n	The line-feed or newline character: \o012 or \x0A
\v	The vertical tab character: \o013 or \x0B
\f	The form-feed character: \o014 or \x0C
\r	The carriage-return character: \o015 or \x0D
\e	The escape character: \o033 or \x1B
\	Use a backslash to force a literal match on the next character. This enables matching on characters that have special meaning in regular expressions; for example, the characters \.*\?\\ represent a literal period, asterisk, question mark, and backslash, respectively.

The character expressions in the preceding table can be modified by using regular-expression *modifiers* or *quantifiers*. Here is an example of using the \w class, along with a modifier, to find all of the words in the string:

```
>> regexp(str,'\w+','match')          % find all individual words
ans =
    'Peter'   'Piper'   'picked'   'a'   'peck'   'of'
    'pickled' 'peppers'
```

This example shows how the + modifier changes the sense of the expression. Without the modifier, the \w expression matches any single "word character." Adding the

modifier changes the expression to match all words (groups of one or more "word characters") in the string.

Character modifiers are listed in the following table:

Modifier	Description and Usage
?	Matches the preceding element zero or one time
*	Matches the preceding element zero or more times
+	Matches the preceding element one or more times
{n}	Matches the preceding element exactly n times
{n,}	Matches the preceding element at least n times
{n,m}	Matches the preceding element at least n times, but no more than m times

Character modifiers are considered *greedy* (i.e., they match the longest string they can). For example, in the code

```
>> regexp(str,'p.*p','match')
ans =
        'per picked a peck of pickled pepp'
```

the modifier matches a lowercase p, followed by zero or more of any charact[redacted], followed by a lowercase p. This behavior can be changed by using the quantif[redacted] expressions listed in the following table:

Quantifier	Description and Usage
q	Matches as much of the string as possible (a *greedy quantifier*). This is the default. Here, q represents one of the modifiers in the previous table (e.g., ?, *, +, {n,m}).
q+	Matches as many components of the string as possible, without rescanning any portions of the string should the initial match fail (a *possessive quantifier*). Again, q represents one of the modifiers in the previous table (e.g., ?, *, +, {n,m}).
q?	Matches as little of the string as possible, while scanning the string (a *lazy quantifier*). Again, q represents one of the modifiers in the previous table (e.g., ?, *, +, {n,m}).

The following code matches shorter strings bracketed by lowercase p charact[redacted]

```
>> regexp(str,'p.*?p','match')
ans =
        'per p' 'peck of p' 'pep'
```

ENGINEERING SYSTEMS
from Pearson Custom Publishing exclusively for The University of Melbourne

Notice that the string `'picked a p'` was not returned. Lazy quantifiers cause the string to be scanned from beginning to end. If a match is found, scanning continues, beginning with the next character in the string; previous portions are not rescanned.

Parentheses can be used to group patterns. For example, the expression in the following code matches a p, followed by one or more characters that are not the letter i:

```
>> regexp(str,'p[^i]+','match')
ans =
        'per p' 'peck of p' 'peppers'
```

The modifier applies to the preceding character. The sense of the expression changes slightly when parentheses are added. The expression in the following example matches one or more sequences of a p followed by a character that is not the letter i:

```
>> regexp(str,'(p[^i])+','match')
ans =
        'pe' 'pe' 'pepp'
```

The modifier now applies to the two-character sequence within the parentheses.

Matches can be made conditional to the context of a regular-expression match. MATLAB supports both *logical* operators and *lookaround* operators. Lookaround operators can be used to condition a match only if the string is preceded or followed by the presence or absence of another string match. The next example matches all words preceded by words ending in d:

```
>> regexp(str,'(?<=d\s)\w+','match')
ans =
        'a' 'peppers'
```

The expression matches groups of word characters only if preceded by the character d, followed by a white-space character.

Lookaround operators are listed in the following table:

Lookaround Operator	Description and Usage
`p(?=q)`	Lookahead. Matches pattern p only if followed by a match for pattern q
`p(?!q)`	Negative lookahead. Matches pattern p only if not followed by a match for pattern q
`(?<=q)p`	Lookbehind. Matches pattern p only if preceded by a match for pattern q
`(?<!q)p`	Negative lookbehind. Matches pattern p only if not preceded by a match for pattern q

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

Logical operators are also available. The following example matches words containing the string 'ip' or the string 'ck':

```
>> regexp(str,'\w*(ip|ck)\w*','match')
ans =
    'Piper' 'picked' 'peck' 'pickled'
```

Logical operators are listed in the following table:

Logical Operator	Description and Usage
p|q	Matches the pattern p or the pattern q
^p	Matches the pattern p only if it occurs at the beginning of the string
p$	Matches the pattern p only if it occurs at the end of the string
\<p	Matches the pattern p only if it occurs at the beginning of a word
p\>	Matches the pattern p only if it occurs at the end of a word
\<p\>	Matches the pattern p only if it exactly matches a word

Repeated sequences can be matched by using tokens. When expressions contain parentheses, the string matching the expression within the parentheses (a *token*) is stored and can be reused in the expression. Up to 255 tokens can be used in an expression. Tokens can be referenced by using the syntax \d, where d is the index of the desired token. For example, \1 represents the first token, \2 represents the second token, and so on. The following example matches any doubled word character:

```
>> regexp(str,'(\w)\1','match')
ans =
    'pp'
```

The next example matches any character followed by zero or more word characters, followed by a white-space character, followed by the same initial character, followed by zero or more word characters, followed by the same white-space character:

```
>> regexp(str,'(.)\w*(\s)\1\w*\2','match')
ans =
    'Peter Piper'
```

Notice that the string 'pickled peppers' did not match, since there is no space after the string 'peppers'.

Hanselman, D. and Littlefield, B. (2012). Character Strings. In *Mastering MATLAB®*, (pp. 165-194). Upper Saddle River, NJ: Pearson Education Inc.

The following table lists the token expressions available in MATLAB:

Expression	Description and Usage
(p)	Captures all characters matched by expression p in a token
(?:p)	Groups all characters matched by expression p, but do not save in a token
(?>p)	Groups atomically, but do not save in a token
(?#A Comment)	Inserts a comment into an expression (The comment is ignored.)
\N	Matches the Nth token in this expression (\1 is the first token, \2 is the second token, etc.)
$N	Inserts a match for the Nth token in a replacement string (regexprep function only)
(?<name>p)	Captures all characters matched by the pattern p in a token and assigns a name to the token
\k<name>	Matches the token referred to by name
$<name>	Inserts a match for the named token in a replacement string (regexprep function only)
(?(T)p)	If token T is generated (i.e., the match for token T was successful), then matches pattern p. This is an IF/THEN construction. The token can be a named token or a positional token.
(?(T)p|q)	If token T is generated, then matches pattern p; otherwise, matches pattern q. This is an IF/THEN/ELSE construction. The token can be a named token or a positional token.

There are four different regular-expression functions in MATLAB. The preceding examples used `regexp`. The `regexpi` function ignores case when matching, whereas the `regexprep` function replaces strings by using regular expressions. The `regexptranslate` function can help to generate regular expressions from character strings.

The following example searches for words beginning with the string `'pi'`, reverses the words, and returns the resulting string:

```
>> regexprep(str,'(pi\w*)(.*)(pi\w*)','$3$2$1')
ans =
     'Peter Piper pickled a peck of picked peppers'
```

This example generates three tokens: Token 1 is the string `'picked'`, token 2 is `'a peck of'`, and token 3 is `'pickled'`. The replacement string `'$3$2$1'` instructs `regexprep` to remove the part of the original string matched by the first argument,

and replace it with a string consisting of the values of the tokens, in the reverse order. The rest of the original string is returned unchanged. Numbered tokens are referenced in the search pattern by using the \N notation. The $N notation is used to reference the tokens in the replacement string.

The regular-expression functions are listed in the following table:

Function	Description and Usage
`regexp`	Searches for substrings by using regular expressions
`regexpi`	Searches for substrings by using regular expressions, ignoring case
`regexprep`	Searches and replaces substrings by using regular expressions
`regexptranslate`	Translates a wildcard string into a regular expression

There are many options available for applying and modifying the regular expression functions. One option is the `'match'` argument illustrated earlier. The `regexprep` function is normally case sensitive and replaces all matches it finds. Options are available to change that default behavior.

All four functions operate on cell arrays of strings, as well as on individual strings. One pattern can be used to search many strings, or different patterns can be used for different strings. Any or all of the input parameters can be a cell array of strings.

Dynamic regular expressions are available for the more adventurous or those more familiar with the usage of regular expressions. Dynamic regular expressions use independent regular expression matching or MATLAB command output to modify a regular expression dynamically. They allow you to make the pattern you want to match dependent on the string you are searching.

Dynamic operators available in MATLAB are listed in the following table:

Dynamic Operator	Description and Usage
`(??p)`	Parses p as a separate regular expression and includes the resulting string in the match expression
`(??@cmd)`	Executes the MATLAB command `cmd` and includes the string returned in the match expression
`(?@cmd)`	Executes the MATLAB command `cmd` but throws away any string returned by the command (often used to debug regular expressions. e.g., `disp()`)
`${cmd}`	Executes the MATLAB command `cmd` and includes the string returned in the replacement expression

For additional examples and more thorough information about these extremely powerful and complex capabilities, see the MATLAB help documentation.

III. Mechanics: Statics

The content in this section is compiled from:

Engineering Mechanics: Statics
12th edition
Hibbeler

The design of this rocket and gantry structure requires a basic knowledge of both statics and dynamics, which form the subject matter of engineering mechanics.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

General Principles

1

CHAPTER OBJECTIVES

- To provide an introduction to the basic quantities and idealizations of mechanics.
- To give a statement of Newton's Laws of Motion and Gravitation.
- To review the principles for applying the SI system of units.
- To examine the standard procedures for performing numerical calculations.
- To present a general guide for solving problems.

1.1 Mechanics

Mechanics is a branch of the physical sciences that is concerned with the state of rest or motion of bodies that are subjected to the action of forces. In general, this subject can be subdivided into three branches: *rigid-body mechanics, deformable-body mechanics*, and *fluid mechanics*. In this book, we will study rigid-body mechanics since it is a basic requirement for the study of the mechanics of deformable bodies and the mechanics of fluids. Furthermore, rigid-body mechanics is essential for the design and analysis of many types of structural members, mechanical components, or electrical devices encountered in engineering.

Rigid-body mechanics is divided into two areas: statics and dynamics. *Statics* deals with the equilibrium of bodies, that is, those that are either at rest or move with a constant velocity; whereas *dynamics* is concerned with the accelerated motion of bodies. We can consider statics as a special case of dynamics, in which the acceleration is zero; however, statics deserves separate treatment in engineering education since many objects are designed with the intention that they remain in equilibrium.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

ENGR1000
A publication from Pearson Custom Publishing

Historical Development. The subject of statics developed very early in history because its principles can be formulated simply from measurements of geometry and force. For example, the writings of Archimedes (287–212 B.C.) deal with the principle of the lever. Studies of the pulley, inclined plane, and wrench are also recorded in ancient writings—at times when the requirements for engineering were limited primarily to building construction.

Since the principles of dynamics depend on an accurate measurement of time, this subject developed much later. Galileo Galilei (1564–1642) was one of the first major contributors to this field. His work consisted of experiments using pendulums and falling bodies. The most significant contributions in dynamics, however, were made by Isaac Newton (1642–1727), who is noted for his formulation of the three fundamental laws of motion and the law of universal gravitational attraction. Shortly after these laws were postulated, important techniques for their application were developed by such notables as Euler, D'Alembert, Lagrange, and others.

1.2 Fundamental Concepts

Before we begin our study of engineering mechanics, it is important to understand the meaning of certain fundamental concepts and principles.

Basic Quantities. The following four quantities are used throughout mechanics.

Length. *Length* is used to locate the position of a point in space and thereby describe the size of a physical system. Once a standard unit of length is defined, one can then use it to define distances and geometric properties of a body as multiples of this unit.

Time. *Time* is conceived as a succession of events. Although the principles of statics are time independent, this quantity plays an important role in the study of dynamics.

Mass. *Mass* is a measure of a quantity of matter that is used to compare the action of one body with that of another. This property manifests itself as a gravitational attraction between two bodies and provides a measure of the resistance of matter to a change in velocity.

Force. In general, *force* is considered as a "push" or "pull" exerted by one body on another. This interaction can occur when there is direct contact between the bodies, such as a person pushing on a wall, or it can occur through a distance when the bodies are physically separated. Examples of the latter type include gravitational, electrical, and magnetic forces. In any case, a force is completely characterized by its magnitude, direction, and point of application.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

Pearson Custom Publishing exclusively for The University of Melbourne

Idealizations. Models or idealizations are used in mechanics in order to simplify application of the theory. Here, we will consider three important idealizations.

Particle. A *particle* has a mass, but a size that can be neglected. For example, the size of the earth is insignificant compared to the size of its orbit, and therefore the earth can be modeled as a particle when studying its orbital motion. When a body is idealized as a particle, the principles of mechanics reduce to a rather simplified form since the geometry of the body *will not be involved* in the analysis of the problem.

Rigid Body. A *rigid body* can be considered as a el and the ground. n of a large number of particles in which all the particles remain at a fixed distance from one another, both before and after applying a load. This model is important because the material properties of any body that is assumed to be rigid will not have to be considered when studying the effects of forces acting on the body. In most cases, the actual deformations occurring in structures, machines, mechanisms, and the like are relatively small, and the rigid-body assumption is suitable for analysis.

Concentrated Force. A *concentrated force* represents the effect of a loading which is assumed to act at a point on a body. We can represent a load by a concentrated force, provided the area over which the load is applied is very small compared to the overall size of the body. An example would be the contact force between a wheel and the ground.

Three forces act on the hook at A. Since these forces all meet at a point, then for any force analysis, we can assume the hook to be represented as a particle.

Steel is a common engineering material that does not deform very much under load. Therefore, we can consider this railroad wheel to be a rigid body acted upon by the concentrated force of the rail.

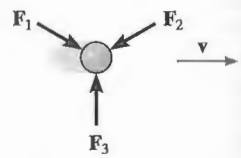
Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

ENGR1000
A publication from Pearson Custom Publishing

1

Newton's Three Laws of Motion. Engineering mechanics is formulated on the basis of Newton's three laws of motion, the validity of which is based on experimental observation. These laws apply to the motion of a particle as measured from a *nonaccelerating* reference frame. They may be briefly stated as follows.

First Law. A particle originally at rest, or moving in a straight line with constant velocity, tends to remain in this state provided the particle is not subjected to an unbalanced force, Fig. 1–1*a*.



Equilibrium

(a)

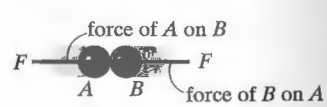
Second Law. A particle acted upon by an *unbalanced force* **F** experiences an acceleration **a** that has the same direction as the force and a magnitude that is directly proportional to the force, Fig. 1–1*b*.* If **F** is applied to a particle of mass m, this law may be expressed mathematically as

$$\mathbf{F} = m\mathbf{a} \tag{1–1}$$

Accelerated motion

(b)

Third Law. The mutual forces of action and reaction between two particles are equal, opposite, and collinear, Fig. 1–1*c*.



Action – reaction

(c)

Fig. 1–1

*Stated another way, the unbalanced force acting on the particle is proportional to the time rate of change of the particle's linear momentum.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

Newton's Law of Gravitational Attraction. Shortly after formulating his three laws of motion, Newton postulated a law governing the gravitational attraction between any two particles. Stated mathematically,

$$F = G\frac{m_1 m_2}{r^2} \tag{1–2}$$

where

F = force of gravitation between the two particles

G = universal constant of gravitation; according to experimental evidence, $G = 66.73(10^{-12})\ \mathrm{m^3/(kg \cdot s^2)}$

m_1, m_2 = mass of each of the two particles

r = distance between the two particles

Weight. According to Eq. 1–2, any two particles or bodies have a mutual attractive (gravitational) force acting between them. In the case of a particle located at or near the surface of the earth, however, the only gravitational force having any sizable magnitude is that between the earth and the particle. Consequently, this force, termed the *weight*, will be the only gravitational force considered in our study of mechanics.

From Eq. 1–2, we can develop an approximate expression for finding the weight W of a particle having a mass $m_1 = m$. If we assume the earth to be a nonrotating sphere of constant density and having a mass $m_2 = M_e$, then if r is the distance between the earth's center and the particle, we have

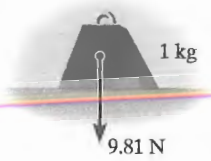
$$W = G\frac{mM_e}{r^2}$$

Letting $g = GM_e/r^2$ yields

$$\boxed{W = mg} \tag{1–3}$$

By comparison with $\mathbf{F} = m\mathbf{a}$, we can see that g is the acceleration due to gravity. Since it depends on r, then the weight of a body is *not* an absolute quantity. Instead, its magnitude is determined from where the measurement was made. For most engineering calculations, however, g is determined at sea level and at a latitude of 45°, which is considered the "standard location."

The astronaut is weightless, for all practical purposes, since she is far removed from the gravitational field of the earth.



(a)

Fig. 1–2

1.3 Units of Measurement

The four basic quantities—length, time, mass, and force—are not all independent from one another; in fact, they are *related* by Newton's second law of motion, $\mathbf{F} = m\mathbf{a}$. Because of this, the *units* used to measure these quantities cannot *all* be selected arbitrarily. The equality $\mathbf{F} = m\mathbf{a}$ is maintained only if three of the four units, called *base units*, are *defined* and the fourth unit is then *derived* from the equation.

SI Units. The International System of units, abbreviated SI after the French "Système International d'Unités," is a modern version of the metric system which has received worldwide recognition. As shown in Table 1–1, the SI system defines length in meters (m), time in seconds (s), and mass in kilograms (kg). The unit of force, called a newton (N), is *derived* from $\mathbf{F} = m\mathbf{a}$. Thus, 1 newton is equal to a force required to give 1 kilogram of mass an acceleration of 1 m/s^2 (N = kg · m/s^2).

If the weight of a body located at the "standard location" is to be determined in newtons, then Eq. 1–3 must be applied. Here measurements give $g = 9.806\,65$ m/s^2; however, for calculations, the value $g = 9.81$ m/s^2 will be used. Thus,

$$W = mg \qquad (g = 9.81 \text{ m/s}^2) \tag{1–4}$$

Therefore, a body of mass 1 kg has a weight of 9.81 N, a 2-kg body weighs 19.62 N, and so on, Fig. 1–2.

TABLE 1–1 Systems of Units

Name	Length	Time	Mass	Force
International System of Units	meter	second	kilogram	newton
SI	m	s	kg	N $\left(\frac{\text{kg}\cdot\text{m}}{\text{s}^2}\right)$

1.4 The International System of Units

The SI system of units is used extensively in this book since it is intended to become the worldwide standard for measurement. Therefore, we will now present some of the rules for its use and some of its terminology relevant to engineering mechanics.

Prefixes. When a numerical quantity is either very large or very small, the units used to define its size may be modified by using a prefix. Some of the prefixes used in the SI system are shown in Table 1–2. Each represents a multiple or submultiple of a unit which, if applied successively, moves the decimal point of a numerical quantity to every third place.* For example, 4 000 000 N = 4 000 kN (kilo-newton) = 4 MN (mega-newton), or 0.005 m = 5 mm (milli-meter). Notice that the SI system does not include the multiple deca (10) or the submultiple centi (0.01), which form part of the metric system. Except for some volume and area measurements, the use of these prefixes is to be avoided in science and engineering.

TABLE 1–2 Prefixes

	Exponential Form	Prefix	SI Symbol
Multiple			
1 000 000 000	10^9	giga	G
1 000 000	10^6	mega	M
1 000	10^3	kilo	k
Submultiple			
0.001	10^{-3}	milli	m
0.000 001	10^{-6}	micro	μ
0.000 000 001	10^{-9}	nano	n

* The kilogram is the only base unit that is defined with a prefix.

Rules for Use. Here are a few of the important rules that describe the proper use of the various SI symbols:

- Quantities defined by several units which are multiples of one another are separated by a *dot* to avoid confusion with prefix notation, as indicated by $N = kg \cdot m/s^2 = kg \cdot m \cdot s^{-2}$. Also, $m \cdot s$ (meter-second) whereas ms (milli-second).
- The exponential power on a unit having a prefix refers to both the unit *and* its prefix. For example, $\mu N^2 = (\mu N)^2 = \mu N \cdot \mu N$. Likewise, mm^2 represents $(mm)^2 = mm \cdot mm$.
- With the exception of the base unit the kilogram, in general avoid the use of a prefix in the denominator of composite units. For example, do not write N/mm, but rather kN/m; also, m/mg should be written as Mm/kg.
- When performing calculations, represent the numbers in terms of their *base or derived units* by converting all prefixes to powers of 10. The final result should then be expressed using a *single prefix*. Also after calculation, it is best to keep numerical values between 0.1 and 1000; otherwise, a suitable prefix should be chosen. For example,

$$\begin{aligned}(50\text{ kN})(60\text{ nm}) &= [50(10^3)\text{ N}][60(10^{-9})\text{ m}] \\ &= 3000(10^{-6})\text{ N}\cdot\text{m} = 3(10^{-3})\text{ N}\cdot\text{m} = 3\text{ mN}\cdot\text{m}\end{aligned}$$

1.5 Numerical Calculations

Computers are often used in engineering for advanced design and analysis.

Numerical work in engineering practice is most often performed by using handheld calculators and computers. It is important, however, that the answers to any problem be reported with both justifiable accuracy and appropriate significant figures. In this section, we will discuss these topics together with some other important aspects involved in all engineering calculations.

Dimensional Homogeneity. The terms of any equation used to describe a physical process must be *dimensionally homogeneous*, that is, each term must be expressed in the same units. Provided this is the case, all the terms of an equation can then be combined if numerical values are substituted for the variables. Consider, for example, the equation $s = vt + \frac{1}{2}at^2$, where, in SI units, s is the position in meters, m, t is time in seconds, s, v is velocity in m/s and a is acceleration in m/s^2. Regardless of how this equation is evaluated, it maintains its dimensional homogeneity. In the form stated, each of the three terms is expressed in meters [m,(m/s) s, $(m/s^2)\,s^2$,] or solving for a, $a = 2s/t^2 - 2v/t$, the terms are each expressed in units of m/s^2 [m/s^2, m/s^2, (m/s)/s].

Keep in mind that problems in mechanics always involve the solution of dimensionally homogeneous equations; so, this fact can then be used as a partial check for algebraic manipulations of an equation.

Significant Figures. The number of significant figures contained in any number determines the accuracy of the number. For instance, the number 4981 contains four significant figures. However, if zeros occur at the end of a whole number, it may be unclear as to how many significant figures the number represents. For example, 23 400 might have three (234), four (2340), or five (23 400) significant figures. To avoid these ambiguities, we will use *engineering notation* to report a result. This requires that numbers be rounded off to the appropriate number of significant digits and then expressed in multiples of (10^3), such as (10^3), (10^6), or (10^{-9}). For instance, if 23 400 has five significant figures, it is written as $23.400(10^3)$, but if it has only three significant figures, it is written as $23.4(10^3)$.

If zeros occur at the beginning of a number that is less than one, then the zeros are not significant. For example, 0.00821 has three significant figures. Using engineering notation, this number is expressed as $8.21(10^{-3})$. Likewise, 0.000582 can be expressed as $0.582(10^{-3})$ or $582(10^{-6})$.

Rounding Off Numbers. Rounding off a number is necessary so that the accuracy of the result will be the same as that of the problem data. As a general rule, any numerical figure ending in five or greater is rounded up and a number less than five is rounded down. The rules for rounding off numbers are best illustrated by examples. Suppose the number 3.5587 is to be rounded off to *three* significant figures. Because the fourth digit (8) is *greater than* 5, the third number is rounded up to 3.56. Likewise 0.5896 becomes 0.590 and 9.3866 becomes 9.39. If we round off 1.341 to three significant figures, because the fourth digit (1) is *less than* 5, then we get 1.34. Likewise, 0.3762 becomes 0.376 and 9.871 becomes 9.87. There is a special case for any number that has a 5 with zeroes following it. As a general rule, if the digit preceding the 5 is an *even number*, then this digit is *not* rounded up. If the digit preceding the 5 is an *odd number*, then it is rounded up. For example, 75.25 rounded off to three significant digits becomes 75.2, 0.1275 becomes 0.128, and 0.2555 becomes 0.256.

Calculations. When a sequence of calculations is performed, it is best to store the intermediate results in the calculator. In other words, do not round off calculations until expressing the final result. This procedure maintains precision throughout the series of steps to the final solution. In this text, we will generally round off the answers to three significant figures since most of the data in engineering mechanics, such as geometry and loads, may be reliably measured to this accuracy.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

1.6 General Procedure for Analysis

When solving problems, do the work as neatly as possible. Being neat will stimulate clear and orderly thinking, and vice versa.

The most effective way of learning the principles of engineering mechanics is to *solve problems*. To be successful at this, it is important to always present the work in a *logical* and *orderly manner*, as suggested by the following sequence of steps:

- Read the problem carefully and try to correlate the actual physical situation with the theory studied.
- Tabulate the problem data and draw any necessary diagrams.
- Apply the relevant principles, generally in mathematical form. When writing any equations, be sure they are dimensionally homogeneous.
- Solve the necessary equations, and report the answer with no more than three significant figures.
- Study the answer with technical judgment and common sense to determine whether or not it seems reasonable.

Important Points

- Statics is the study of bodies that are at rest or move with constant velocity.
- A particle has a mass but a size that can be neglected.
- A rigid body does not deform under load.
- Concentrated forces are assumed to act at a point on a body.
- Newton's three laws of motion should be memorized.
- Mass is measure of a quantity of matter that does not change from one location to another.
- Weight refers to the gravitational attraction of the earth on a body or quantity of mass. Its magnitude depends upon the elevation at which the mass is located.
- In the SI system, the unit of force, the newton, is a derived unit. The meter, second, and kilogram are base units.
- Prefixes G, M, k, m, μ, and n are used to represent large and small numerical quantities. Their exponential size should be known along with the rules for using the SI units.
- Perform numerical calculations with several significant figures and then report the final answer to three significant figures.
- Algebraic manipulations of an equation can be checked in part by verifying that the equation remains dimensionally homogeneous.
- Know the rules for rounding off numbers.

EXAMPLE 1.1

Convert 2 km/h to m/s.

SOLUTION

Since 1 km = 1000 m and 1 h = 3600 s, the factors of conversion are arranged in the following order, so that a cancellation of the units can be applied:

$$2\ \text{km/h} = \frac{2\ \cancel{\text{km}}}{\cancel{\text{h}}}\left(\frac{1000\ \text{m}}{\cancel{\text{km}}}\right)\left(\frac{1\ \cancel{\text{h}}}{3600\ \text{s}}\right)$$

$$= \frac{2000\ \text{m}}{3600\ \text{s}} = 0.556\ \text{m/s} \qquad \textit{Ans.}$$

NOTE: Remember to round off the final answer to three significant figures.

EXAMPLE 1.2

Evaluate each of the following and express with SI units having an appropriate prefix: (a) (50 mN)(6 GN), (b) (400 mm)(0.6 MN)^2, (c) 45 MN^3/900 Gg.

SOLUTION

First convert each number to base units, perform the indicated operations, then choose an appropriate prefix.

Part (a)

$$(50\ \text{mN})(6\ \text{GN}) = [50(10^{-3})\ \text{N}][6(10^9)\ \text{N}]$$

$$= 300(10^6)\ \text{N}^2$$

$$= 300(10^6)\ \cancel{\text{N}^2}\left(\frac{1\ \text{kN}}{10^3\ \cancel{\text{N}}}\right)\left(\frac{1\ \text{kN}}{10^3\ \cancel{\text{N}}}\right)$$

$$= 300\ \text{kN}^2 \qquad \textit{Ans.}$$

NOTE: Keep in mind the convention $\text{kN}^2 = (\text{kN})^2 = 10^6\ \text{N}^2$.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 1.2 (Continued)

Part (b)

$$
\begin{aligned}
(400\text{ mm})(0.6\text{ MN})^2 &= [400(10^{-3})\text{ m}][0.6(10^6)\text{ N}]^2 \\
&= [400(10^{-3})\text{ m}][0.36(10^{12})\text{ N}^2] \\
&= 144(10^9)\text{ m}\cdot\text{N}^2 \\
&= 144\text{ Gm}\cdot\text{N}^2 \qquad \textit{Ans.}
\end{aligned}
$$

We can also write

$$
\begin{aligned}
144(10^9)\text{ m}\cdot\text{N}^2 &= 144(10^9)\text{ m}\cdot\cancel{\text{N}^2}\left(\frac{1\text{ MN}}{10^6\,\cancel{\text{N}}}\right)\left(\frac{1\text{ MN}}{10^6\,\cancel{\text{N}}}\right) \\
&= 0.144\text{ m}\cdot\text{MN}^2 \qquad \textit{Ans.}
\end{aligned}
$$

Part (c)

$$
\begin{aligned}
\frac{45\text{ MN}^3}{900\text{ Gg}} &= \frac{45(10^6\text{ N})^3}{900(10^6)\text{ kg}} \\
&= 50(10^9)\,\cancel{\text{N}^3}/\text{kg} \\
&= 50(10^9)\,\cancel{\text{N}^3}\left(\frac{1\text{ kN}}{10^3\,\cancel{\text{N}}}\right)^3\frac{1}{\text{kg}} \\
&= 50\text{ kN}^3/\text{kg} \qquad \textit{Ans.}
\end{aligned}
$$

MasteringENGINEERING™
Customized coaching is available!

PROBLEMS

1

1–1. Round off the following numbers to three significant figures: (a) 4.65735 m, (b) 55.578 s, (c) 4555 N, and (d) 2768 kg.

1–2. Represent each of the following combinations of units in the correct SI form using an appropriate prefix: (a) μMN, (b) N/μm, (c) MN/ks^2, and (d) kN/ms.

1–3. Represent each of the following quantities in the correct SI form using an appropriate prefix: (a) 0.000431 kg, (b) 35.3(10^3) N, and (c) 0.00532 km.

***1–4.** Represent each of the following combinations of units in the correct SI form: (a) Mg/ms, (b) N/mm, and (c) mN/(kg · μs).

1–5. Represent each of the following combinations of units in the correct SI form using an appropriate prefix: (a) kN/μs, (b) Mg/mN, and (c) MN/(kg · ms).

Video Solutions are available for Problems 1-3 to 1-5.

1–6. Represent each of the following to three significant figures and express each answer in SI units using an appropriate prefix: (a) 45 320 kN, (b) 568(10^5) mm, and (c) 0.005 63 mg.

1–7. A rocket has a mass of 3.65(10^6) kg on earth. Specify its weight in SI units. If the rocket is on the moon, where the acceleration due to gravity is g_m = 1.62 m/s^2, determine to three significant figures its weight in SI units and its mass in SI units.

***1–8.** If a car is traveling at 88 km/h, determine its speed in meters per second.

1–9. The *pascal* (Pa) is actually a very small unit of pressure. Given (1 Pa = 1 N/m^2) and atmospheric pressure at sea level is 101.325 kN/m^2. How many pascals is this?

1–10. What is the weight in newtons of an object that has a mass of: (a) 10 kg, (b) 0.5 g, and (c) 4.50 Mg? Express the result to three significant figures. Use an appropriate prefix.

1–11. Evaluate each of the following to three significant figures and express each answer in SI units using an appropriate prefix: (a) 354 mg(45 km)/(0.0356 kN), (b) (0.004 53 Mg)(201 ms), and (c) 435 MN/23.2 mm.

***1–12.** The specific weight (wt./vol.) of brass is 85 kN/m^3. Determine its density (mass/vol.) in SI units. Use an appropriate prefix.

***1–13.** Two particles have a mass of 8 kg and 12 kg, respectively. If they are 800 mm apart, determine the force of gravity acting between them. Compare this result with the weight of each particle.

1–14. Determine the mass in kilograms of an object that has a weight of (a) 20 mN, (b) 150 kN, and (c) 60 MN. Express the answer to three significant figures.

1–15. Evaluate each of the following to three significant figures and express each answer in SI units using an appropriate prefix: (a) (200 kN)^2, (b) (0.005 mm)^2, and (c) (400 m)^3.

1–16. Using the base units of the SI system, show that Eq. 1–2 is a dimensionally homogeneous equation which gives F in newtons. Determine to three significant figures the gravitational force acting between two spheres that are touching each other. The mass of each sphere is 200 kg and the radius is 300 mm.

***1–17.** Evaluate each of the following to three significant figures and express each answer in SI units using an appropriate prefix: (a) (0.631 Mm)/(8.60 kg)^2, and (b) (35 mm)^2(48 kg)^3.

1–18. Evaluate (204 mm)(0.00457 kg)/(34.6 N) to three significant figures and express the answer in SI units using an appropriate prefix.

Hibbeler, RC. (2012). General Principles. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 3-16). Singapore: Pearson Education South Asia Pte Ltd.

ENGR10003 – ENGINE[redacted]
A publication from Pearson Custom Publishing exclusively for The [redacted]

This bridge tower is stabilized by cables that exert forces at the points of connection. In this chapter we will show how to express these forces as Cartesian vectors and then determine the resultant force.

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

Force Vectors

2

CHAPTER OBJECTIVES

- To show how to add forces and resolve them into components using the Parallelogram Law.
- To express force and position in Cartesian vector form and explain how to determine the vector's magnitude and direction.
- To introduce the dot product in order to determine the angle between two vectors or the projection of one vector onto another.

2.1 Scalars and Vectors

All physical quantities in engineering mechanics are measured using either scalars or vectors.

Scalar. A *scalar* is any positive or negative physical quantity that can be completely specified by its *magnitude*. Examples of scalar quantities include length, mass, and time.

Vector. A *vector* is any physical quantity that requires both a *magnitude* and a *direction* for its complete description. Examples of vectors encountered in statics are force, position, and moment. A vector is shown graphically by an arrow. The length of the arrow represents the *magnitude* of the vector, and the angle θ between the vector and a fixed axis defines the *direction of its line of action*. The head or tip of the arrow indicates the *sense of direction* of the vector, Fig. 2–1.

In print, vector quantities are represented by bold face letters such as **A**, and its magnitude of the vector is italicized, A. For handwritten work, it is often convenient to denote a vector quantity by simply drawing an arrow on top of it, $\vec{A}$.

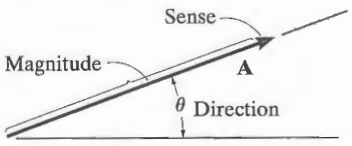


Fig. 2–1

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

This bridge tower is stabilized by cables that exert forces at the points of connection. In this chapter we will show how to express these forces as Cartesian vectors and then determine the resultant force.

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

Force Vectors

2

CHAPTER OBJECTIVES

- To show how to add forces and resolve them into components using the Parallelogram Law.
- To express force and position in Cartesian vector form and explain how to determine the vector's magnitude and direction.
- To introduce the dot product in order to determine the angle between two vectors or the projection of one vector onto another.

2.1 Scalars and Vectors

All physical quantities in engineering mechanics are measured using either scalars or vectors.

Scalar. A *scalar* is any positive or negative physical quantity that can be completely specified by its *magnitude*. Examples of scalar quantities include length, mass, and time.

Vector. A *vector* is any physical quantity that requires both a *magnitude* and a *direction* for its complete description. Examples of vectors encountered in statics are force, position, and moment. A vector is shown graphically by an arrow. The length of the arrow represents the *magnitude* of the vector, and the angle θ between the vector and a fixed axis defines the *direction of its line of action*. The head or tip of the arrow indicates the *sense of direction* of the vector, Fig. 2–1.

In print, vector quantities are represented by bold face letters such as **A**, and its magnitude of the vector is italicized, A. For handwritten work, it is often convenient to denote a vector quantity by simply drawing an arrow on top of it, $\vec{A}$.

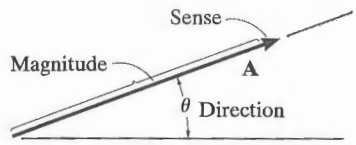
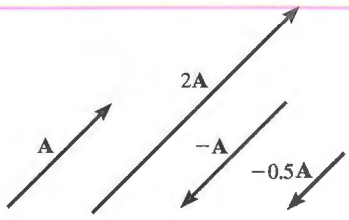


Fig. 2–1

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

2.2 Vector Operations



Scalar multiplication and division

Fig. 2–2

Multiplication and Division of a Vector by a Scalar. If a vector is multiplied by a positive scalar, its magnitude is increased by that amount. When multiplied by a negative scalar it will also change the directional sense of the vector. Graphic examples of these operations are shown in Fig. 2–2.

Vector Addition. All vector quantities obey the *parallelogram law of addition*. To illustrate, the two "*component*" *vectors* **A** and **B** in Fig. 2–3*a* are added to form a "*resultant*" *vector* $\mathbf{R} = \mathbf{A} + \mathbf{B}$ using the following procedure:

- First join the tails of the components at a point so that it makes them concurrent, Fig. 2–3*b*.
- From the head of **B**, draw a line parallel to **A**. Draw another line from the head of **A** that is parallel to **B**. These two lines intersect at point *P* to form the adjacent sides of a parallelogram.
- The diagonal of this parallelogram that extends to *P* forms **R**, which then represents the resultant vector $\mathbf{R} = \mathbf{A} + \mathbf{B}$, Fig. 2–3*c*.

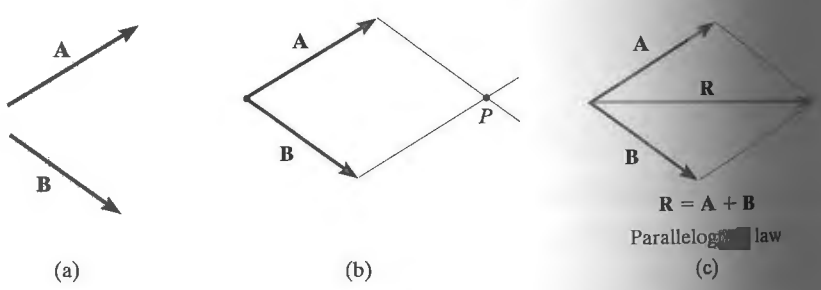


Fig. 2–3

We can also add **B** to **A**, Fig. 2–4*a*, using the *triangle rule*, which is a special case of the parallelogram law, whereby vector **B** is added to vector **A** in a "head-to-tail" fashion, i.e., by connecting the head of **A** to the tail of **B**, Fig. 2–4*b*. The resultant **R** extends from the tail of **A** to the head of **B**. In a similar manner, **R** can also be obtained by adding **A** to **B**, Fig. 2–4*c*. By comparison, it is seen that vector addition is commutative; in other words, the vectors can be added in either order, i.e., $\mathbf{R} = \mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.

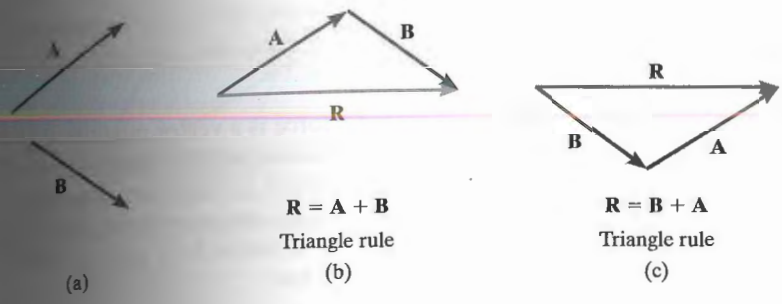


Fig. 2–4

As a special case, if the two vectors **A** and **B** are *collinear*, i.e., both have the same line of action, the parallelogram law reduces to an *algebraic* or *scalar addition* $R = A + B$, as shown in Fig. 2–5.

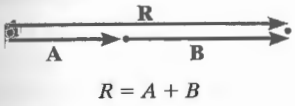


Fig. 2–5

Vector Subtraction. The resultant of the *difference* between two vectors **A** and **B** of the same type may be expressed as

$$\mathbf{R}' = \mathbf{A} - \mathbf{B} = \mathbf{A} + (-\mathbf{B})$$

This vector sum is shown graphically in Fig. 2–6. Subtraction is therefore defined as a special case of addition, so the rules of vector addition also apply to vector subtraction.

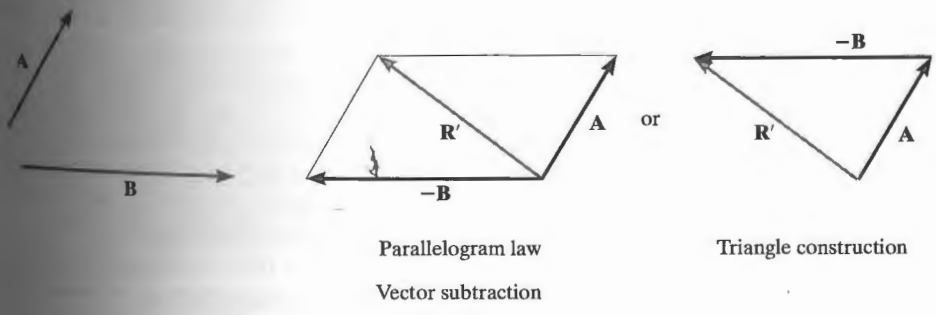
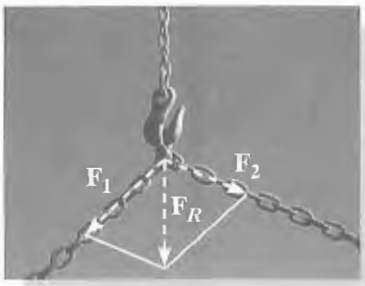


Fig. 2–6

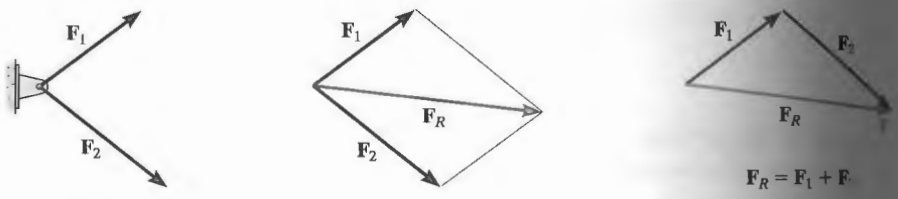
2.3 Vector Addition of Forces

Experimental evidence has shown that a force is a vector quantity since it has a specified magnitude, direction, and sense and it adds according to the parallelogram law. Two common problems in statics involve either finding the resultant force, knowing its components, or resolving a known force into two components. We will now describe how each of these problems is solved using the parallelogram law.



The parallelogram law must be used to determine the resultant of the two forces acting on the hook.

Finding a Resultant Force. The two component forces $\mathbf{F}_1$ and $\mathbf{F}_2$ acting on the pin in Fig. 2–7*a* can be added together to form the resultant force $\mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2$, as shown in Fig. 2–7*b*. From this construction or using the triangle rule, Fig. 2–7*c*, we can apply the law of cosines or the law of sines to the triangle in order to obtain the magnitude of the resultant force and its direction.



(a) (b) (c)

Fig. 2–7

Using the parallelogram law force **F** caused by the vertical member can be resolved into components acting along the suspension cables *u* and *v*.

Finding the Components of a Force. Sometimes it is necessary to resolve a force into two *components* in order to study its pulling or pushing effect in two specific directions. For example, in Fig. 2–8*a*, **F** is to be resolved into two components along the two members, defined by the *u* and *v* axes. In order to determine the magnitude of each component, a parallelogram is constructed first, by drawing lines starting from the tip of **F**, one line parallel to *u*, and the other line parallel to *v*. These lines then intersect with the *v* and *u* axes, forming a parallelogram. The force components $\mathbf{F}_u$ and $\mathbf{F}_v$ are then established by simply joining the tail of **F** to the intersection points on the *u* and *v* axes, Fig. 2–8*b*. This parallelogram can then be reduced to a triangle, which represents the triangle rule, Fig. 2–8*c*. From this, the law of sines can then be applied to determine the unknown magnitudes of the components.

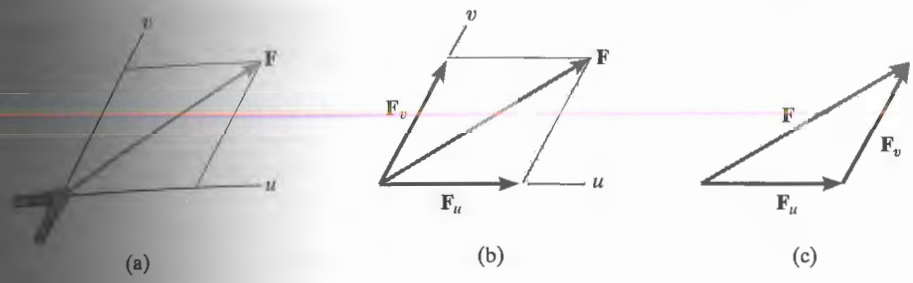


Fig. 2–8

Addition of Several Forces. If more than two forces are to be added, successive applications of the parallelogram law can be carried out in order to obtain the resultant force. For example, if three forces $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$ act at a point O, Fig. 2–9, the resultant of any two of the forces is found, say, $\mathbf{F}_1 + \mathbf{F}_2$—and then this resultant is added to the third force, yielding the resultant of all three forces; i.e., $\mathbf{F}_R = (\mathbf{F}_1 + \mathbf{F}_2) + \mathbf{F}_3$. Using the parallelogram law to add more than two forces, as shown here, often requires extensive geometric and trigonometric calculation to determine the numerical values for the magnitude and direction of the resultant. Instead, problems of this type are easily solved by using the "rectangular-component method," which is explained in Sec. 2.4.

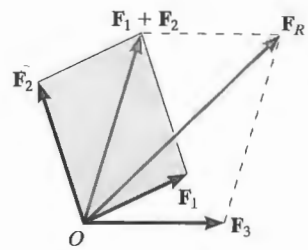
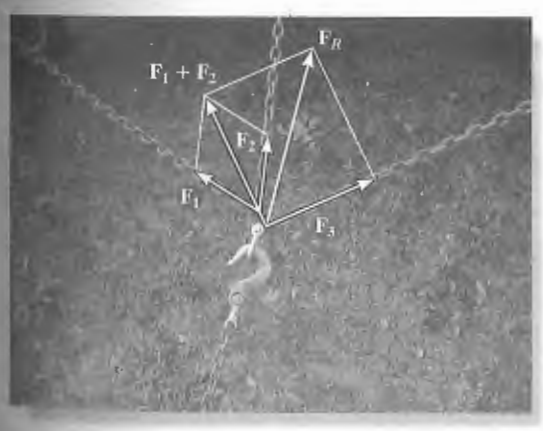
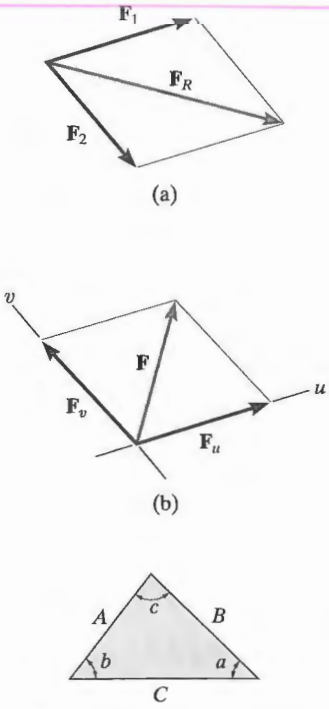


Fig. 2–9



The resultant force $\mathbf{F}_R$ on the hook requires the addition of $\mathbf{F}_1 + \mathbf{F}_2$, then this resultant is added to $\mathbf{F}_3$.

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.



Cosine law:

$$C = \sqrt{A^2 + B^2 - 2AB\cos c}$$

Sine law:

$$\frac{A}{\sin a} = \frac{B}{\sin b} = \frac{C}{\sin c}$$

(c)

Fig. 2–10

Procedure for Analysis

Problems that involve the addition of two forces can be solved as follows:

Parallelogram Law.

- Two "component" forces $\mathbf{F}_1$ and $\mathbf{F}_2$ in Fig. 2–10*a* add according to the parallelogram law, yielding a *resultant* force $\mathbf{F}_R$ that forms the diagonal of the parallelogram.
- If a force $\mathbf{F}$ is to be resolved into *components* along two axes u and v, Fig. 2–10*b*, then start at the head of force $\mathbf{F}$ and construct lines parallel to the axes, thereby forming the parallelogram. The sides of the parallelogram represent the components, $\mathbf{F}_u$ and $\mathbf{F}_v$.
- Label all the known and unknown force magnitudes and the angles on the sketch and identify the two unknowns as the magnitude and direction of $\mathbf{F}_R$, or the magnitudes of its components.

Trigonometry.

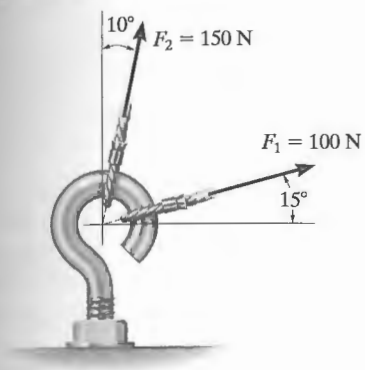
- Redraw a half portion of the parallelogram to illustrate the triangular head-to-tail addition of the components.
- From this triangle, the magnitude of the resultant force can be determined using the law of cosines, and its direction is determined from the law of sines. The magnitudes of two force components are determined from the law of sines. The formulas are given in Fig. 2–10*c*.

Important Points

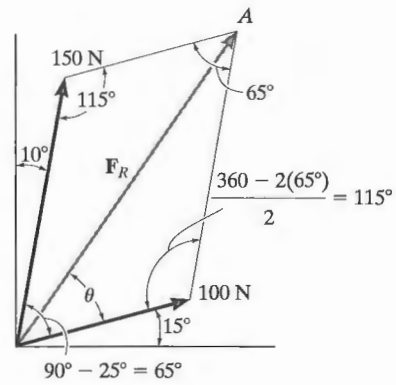
- A scalar is a positive or negative number.
- A vector is a quantity that has a magnitude, direction and sense.
- Multiplication or division of a vector by a scalar will change the magnitude of the vector. The sense of the vector will change if the scalar is negative.
- As a special case, if the vectors are collinear, the resultant is formed by an algebraic or scalar addition.

EXAMPLE 2.1

The screw eye in Fig. 2–11*a* is subjected to two forces, $\mathbf{F}_1$ and $\mathbf{F}_2$. Determine the magnitude and direction of the resultant force.



(a)



(b)

SOLUTION

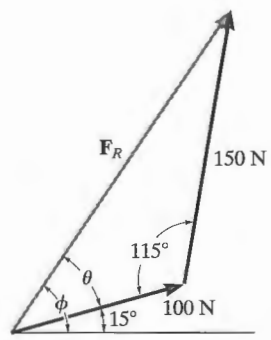
Parallelogram Law. The parallelogram is formed by drawing a line from the head of $\mathbf{F}_1$ that is parallel to $\mathbf{F}_2$, and another line from the head of $\mathbf{F}_2$ that is parallel to $\mathbf{F}_1$. The resultant force $\mathbf{F}_R$ extends to where these lines intersect at point A, Fig. 2–11*b*. The two unknowns are the magnitude of $\mathbf{F}_R$ and the angle θ (theta).

Trigonometry. From the parallelogram, the vector triangle is constructed, Fig. 2–11*c*. Using the law of cosines

$$F_R = \sqrt{(100\text{ N})^2 + (150\text{ N})^2 - 2(100\text{ N})(150\text{ N})\cos 115°}$$

$$= \sqrt{10\,000 + 22\,500 - 30\,000(-0.4226)} = 212.6\text{ N}$$

$$= 213\text{ N} \qquad \textit{Ans.}$$



(c)

Fig. 2–11

Applying the law of sines to determine θ,

$$\frac{150\text{ N}}{\sin\theta} = \frac{212.6\text{ N}}{\sin 115°} \qquad \sin\theta = \frac{150\text{ N}}{212.6\text{ N}}(\sin 115°)$$

$$\theta = 39.8°$$

Thus, the direction ϕ (phi) of $\mathbf{F}_R$, measured from the horizontal, is

$$\phi = 39.8° + 15.0° = 54.8° \qquad \textit{Ans.}$$

NOTE: The results seem reasonable, since Fig. 2–11*b* shows $\mathbf{F}_R$ to have a magnitude larger than its components and a direction that is between them.

EXAMPLE 2.2

Resolve the horizontal 600-N force in Fig. 2–12*a* into components acting along the u and v axes and determine the magnitudes of these components.

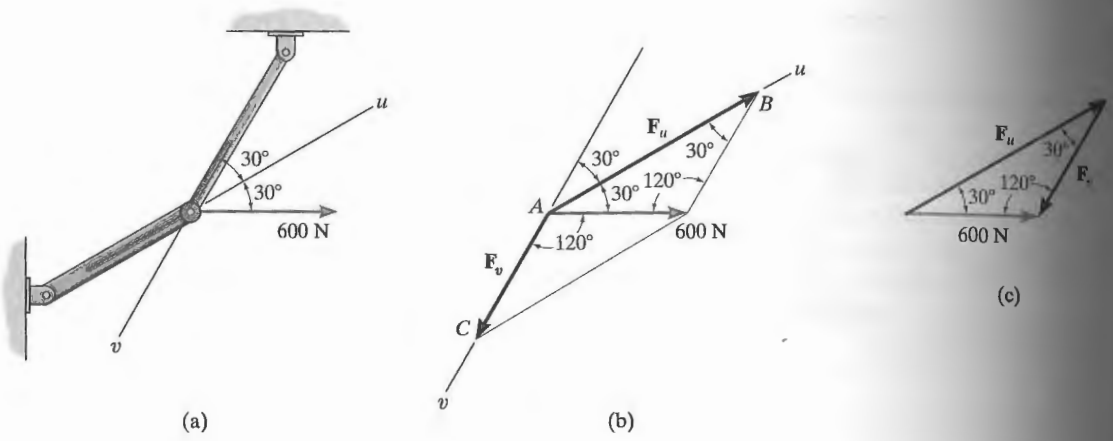


Fig. 2–12

SOLUTION

The parallelogram is constructed by extending a line from the *head* of the 600-N force parallel to the v axis until it intersects the u axis at point B, Fig. 2–12*b*. The arrow from A to B represents $\mathbf{F}_u$. Similarly, the line extended from the head of the 600-N force drawn parallel to the u axis intersects the v axis at point C, which gives $\mathbf{F}_v$.

The vector addition using the triangle rule is shown in Fig. 2–12*c*. The two unknowns are the magnitudes of $\mathbf{F}_u$ and $\mathbf{F}_v$. Applying the law of sines,

$$\frac{F_u}{\sin 120^\circ} = \frac{600\text{ N}}{\sin 30^\circ}$$

$$F_u = 1039\text{ N} \qquad \textit{Ans.}$$

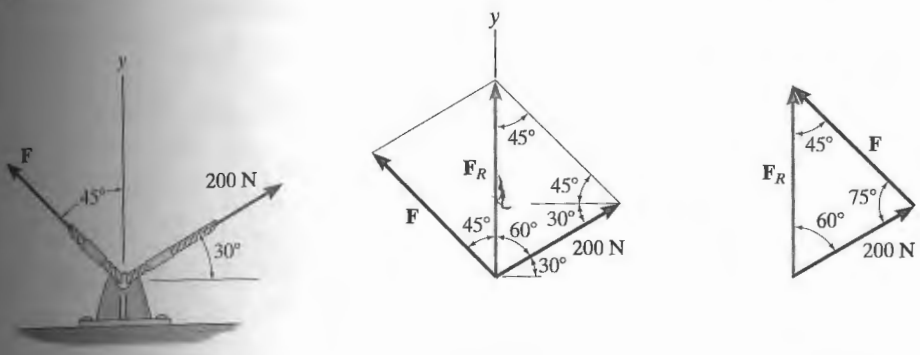
$$\frac{F_v}{\sin 30^\circ} = \frac{600\text{ N}}{\sin 30^\circ}$$

$$F_v = 600\text{ N} \qquad \textit{Ans.}$$

NOTE: The result for F_u shows that sometimes a component can have a greater magnitude than the resultant.

2.3

[redacted]e the magnitude of the component force **F** in Fig. 2–13*a* and the [redacted]gnitude of the resultant force $\mathbf{F}_R$ if $\mathbf{F}_R$ is directed along the positive *y* axis.



(a) (b) (c)

Fig. 2–13

SOLUTION

The parallelogram law of addition is shown in Fig. 2–13*b*, and the triangle rule is shown in Fig. 2–13*c*. The magnitudes of $\mathbf{F}_R$ and **F** are the two [redacted]knowns. They can be determined by applying the law of sines.

$$\frac{F}{\sin 60^\circ} = \frac{200\text{ N}}{\sin 45^\circ}$$

$$F = 245\text{ N} \qquad \textit{Ans.}$$

$$\frac{F_R}{\sin 75^\circ} = \frac{200\text{ N}}{\sin 45^\circ}$$

$$F_R = 273\text{ N} \qquad \textit{Ans.}$$

EXAMPLE 2.4

It is required that the resultant force acting on the eyebolt in Fig. 2–14*a* be directed along the positive x axis and that $\mathbf{F}_2$ have a *minimum* magnitude. Determine this magnitude, the angle θ, and the corresponding resultant force.

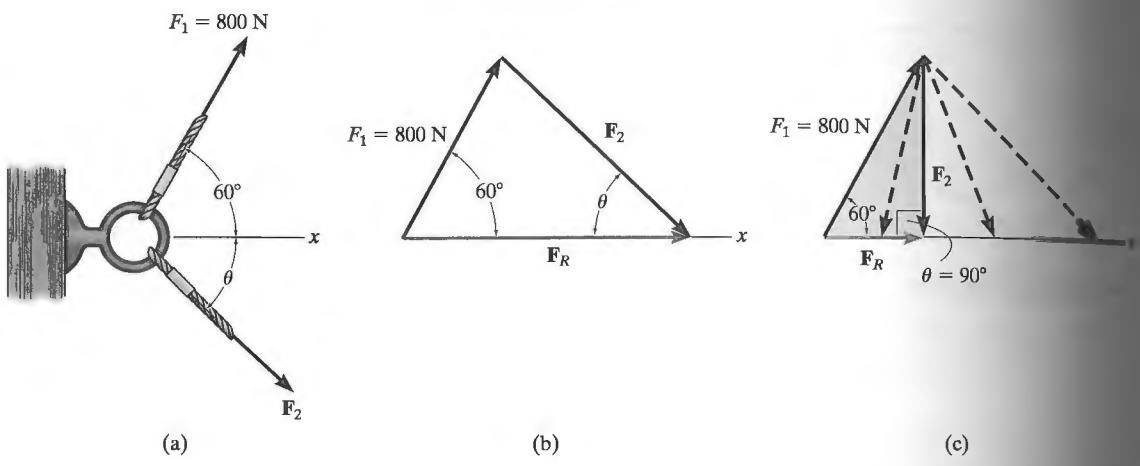


Fig. 2–14

SOLUTION

The triangle rule for $\mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2$ is shown in Fig. 2–14*b*. Since the magnitudes (lengths) of $\mathbf{F}_R$ and $\mathbf{F}_2$ are not specified, then $\mathbf{F}_2$ can actually be any vector that has its head touching the line of action of $\mathbf{F}_R$, Fig. 2–14*c*. However, as shown, the magnitude of $\mathbf{F}_2$ is a *minimum* or the shortest length when its line of action is *perpendicular* to the line of action of $\mathbf{F}_R$, that is, when

$$\theta = 90° \qquad \textit{Ans.}$$

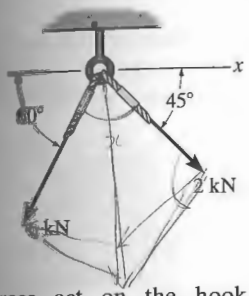
Since the vector addition now forms a right triangle, the two unknown magnitudes can be obtained by trigonometry.

$$F_R = (800\text{ N})\cos 60° = 400\text{ N} \qquad \textit{Ans.}$$

$$F_2 = (800\text{ N})\sin 60° = 693\text{ N} \qquad \textit{Ans.}$$

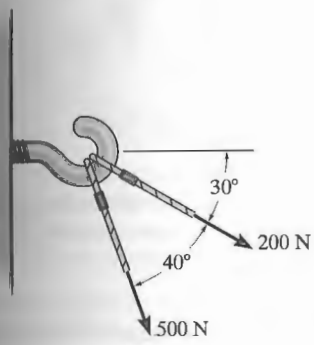
FUNDAMENTAL PROBLEMS*

F2–1. Determine the magnitude of the resultant force acting on the screw eye and its direction measured clockwise from the x axis.



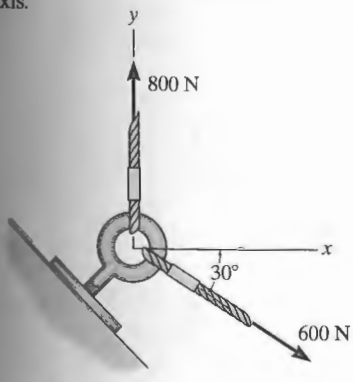
F2–1

F2–2. Two forces act on the hook. Determine the magnitude of the resultant force.



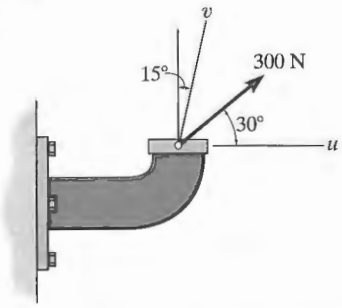
F2–2

F2–3. Determine the magnitude of the resultant force and its direction measured counterclockwise from the positive x axis.



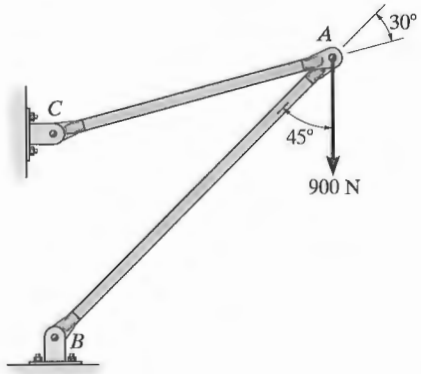
F2–3

F2–4. Resolve the 300-N force into components along the u and v axes, and determine the magnitude of each of these components.



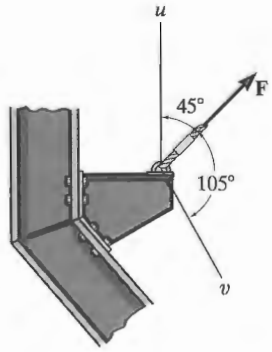
F2–4

F2–5. The force $F = 900$ N acts on the frame. Resolve this force into components acting along members AB and AC, and determine the magnitude of each component.



F2–5

F2–6. If force **F** is to have a component along the u axis of $F_u = 6$ kN, determine the magnitude of **F** and the magnitude of its component $\mathbf{F}_v$ along the v axis.



F2–6

*Partial solutions and answers to all Fundamental Problems are given in the back of the book.

MasteringENGINEERING
Customized coaching is available

PROBLEMS

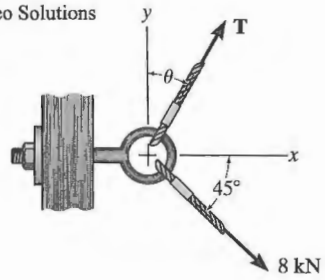
2

•2–1. If $\theta = 30°$ and $T = 6$ kN, determine the magnitude of the resultant force acting on the eyebolt and its direction measured clockwise from the positive x axis.

2–2. If $\theta = 60°$ and $T = 5$ kN, determine the magnitude of the resultant force acting on the eyebolt and its direction measured clockwise from the positive x axis.

2–3. If the magnitude of the resultant force is to be 9 kN directed along the positive x axis, determine the magnitude of force **T** acting on the eyebolt and its angle θ.

Video Solutions

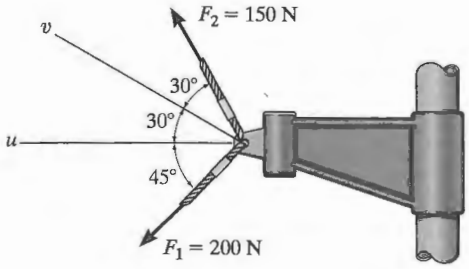


Probs. 2–1/2/3

***2–4.** Determine the magnitude of the resultant force acting on the bracket and its direction measured counterclockwise from the positive u axis.

•2–5. Resolve $\mathbf{F}_1$ into components along the u and v axes, and determine the magnitudes of these components.

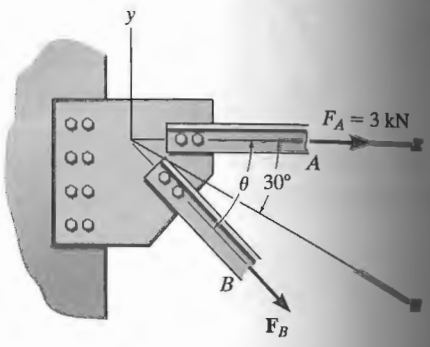
2–6. Resolve $\mathbf{F}_2$ into components along the u and v axes, and determine the magnitudes of these components.



Probs. 2–4/5/6

2–7. If $F_B = 2$ kN and the resultant force acts along the positive u axis, determine the magnitude of the resultant force and the angle θ.

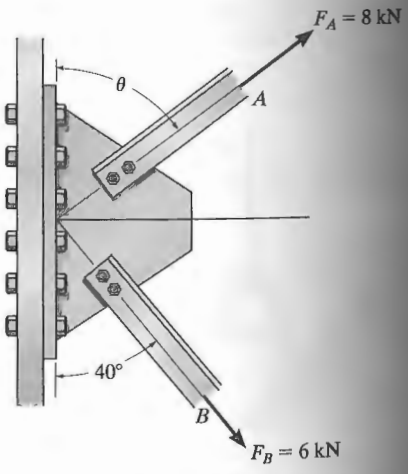
***2–8.** If the resultant force is required to act along the positive u axis and have a magnitude of 5 kN, determine the required magnitude of $\mathbf{F}_B$ and its direction θ.



Probs. 2–7/8

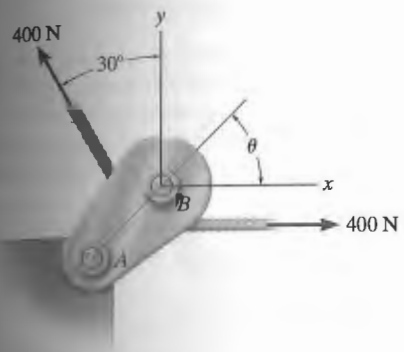
•2–9. The plate is subjected to the two forces at A and B as shown. If $\theta = 60°$, determine the magnitude of the resultant of these two forces and its direction measured clockwise from the horizontal.

2–10. Determine the angle of θ for connecting member A to the plate so that the resultant force of $\mathbf{F}_A$ and $\mathbf{F}_B$ is directed horizontally to the right. Also, what is the magnitude of the resultant force?



Probs. 2–9/10

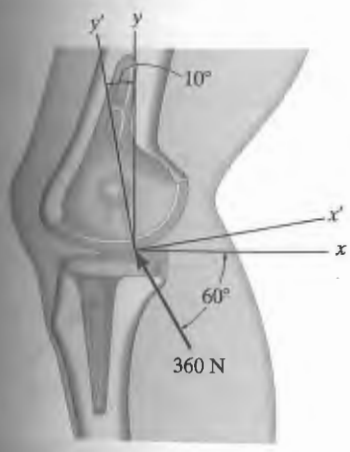
2–11. If the tension in the cable is 400 N, determine the magnitude and direction of the resultant force acting on the pulley. This angle is the same angle θ of line AB on the tailboard block.



Prob. 2–11

2–12. The device is used for surgical replacement of the knee joint. If the force acting along the leg is 360 N, determine its components along the x and y' axes.

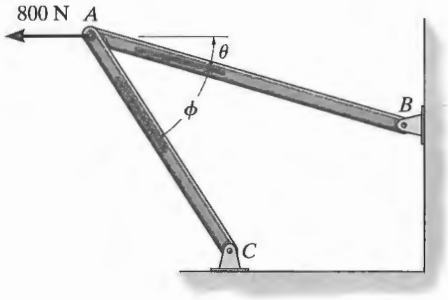
•2–13. The device is used for surgical replacement of the knee joint. If the force acting along the leg is 360 N, determine its components along the x' and y axes.



Probs. 2–12/13

2–14. Determine the design angle θ ($0° \leq \theta \leq 90°$) for strut AB so that the 800-N horizontal force has a component of 1000 N directed from A towards C. What is the component of force acting along member AB? Take $\phi = 40°$.

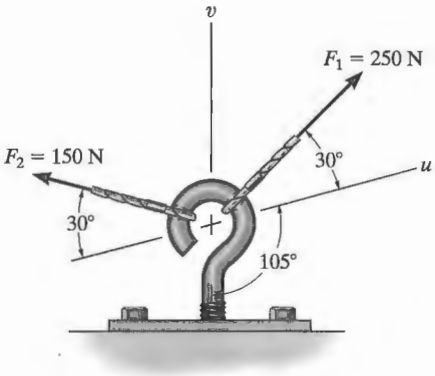
2–15. Determine the design angle ϕ ($0° \leq \phi \leq 90°$) between struts AB and AC so that the 800-N horizontal force has a component of 1200 N which acts up to the left, in the same direction as from B towards A. Take $\theta = 30°$.



Probs. 2–14/15

***2–16.** Resolve $\mathbf{F}_1$ into components along the u and v axes and determine the magnitudes of these components.

•2–17. Resolve $\mathbf{F}_2$ into components along the u and v axes and determine the magnitudes of these components.

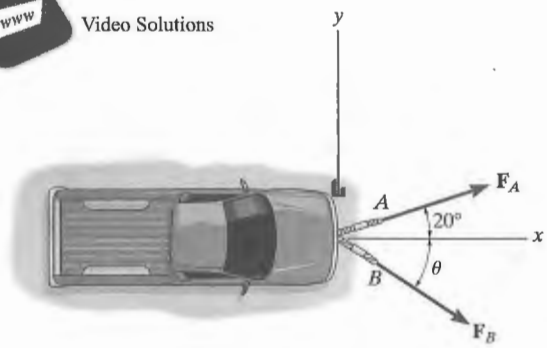


Probs. 2–16/17

2–18. The truck is to be towed using two ropes. Determine the magnitudes of forces $\mathbf{F}_A$ and $\mathbf{F}_B$ acting on each rope in order to develop a resultant force of 950 N directed along the positive x axis. Set $\theta = 50°$.

2–19. The truck is to be towed using two ropes. If the resultant force is to be 950 N, directed along the positive x axis, determine the magnitudes of forces $\mathbf{F}_A$ and $\mathbf{F}_B$ acting on each rope and the angle θ of $\mathbf{F}_B$ so that the magnitude of $\mathbf{F}_B$ is a *minimum*. $\mathbf{F}_A$ acts at 20° from the x axis as shown.

Video Solutions

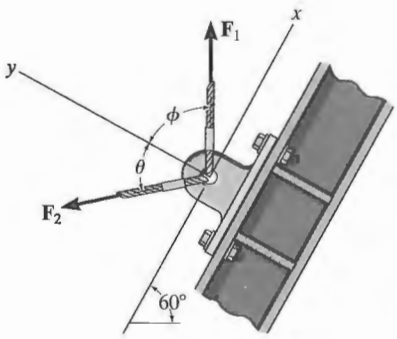


Prob. 2–18/19

***2–20.** If $\phi = 45°$, $F_1 = 5$ kN, and the resultant force is 6 kN directed along the positive y axis, determine the required magnitude of $\mathbf{F}_2$ and its direction θ.

•2–21. If $\phi = 30°$ and the resultant force is to be 6 kN directed along the positive y axis, determine the magnitudes of $\mathbf{F}_1$ and $\mathbf{F}_2$ and the angle θ if F_2 is required to be a minimum.

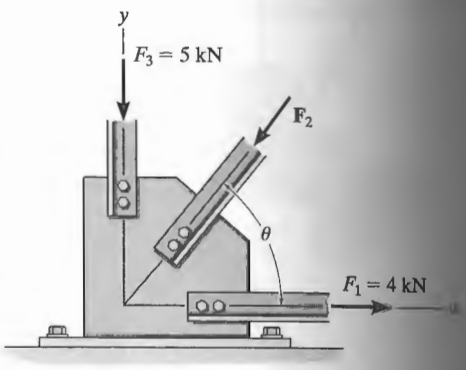
2–22. If $\phi = 30°$, $F_1 = 5$ kN, and the resultant force is to be directed along the positive y axis, determine the magnitude of the resultant force if F_2 is to be a minimum. Also, what is F_2 and the angle θ?



Probs. 2–20/21/22

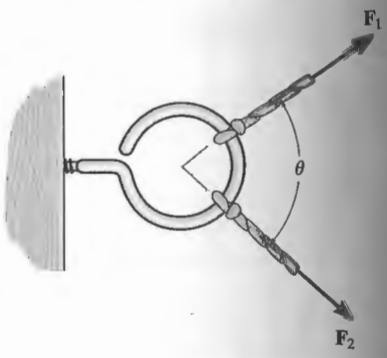
2–23. If $\theta = 30°$ and $F_2 = 6$ kN, determine the magnitude of the resultant force acting on the plate and its direction measured clockwise from the positive x axis.

***2–24.** If the resultant force $\mathbf{F}_R$ is directed along a line measured 75° clockwise from the positive x axis and the magnitude of $\mathbf{F}_2$ is to be a minimum, determine the magnitudes of $\mathbf{F}_R$ and $\mathbf{F}_2$ and the angle $\theta \leq 90°$.



Probs. 2–23/24

•2–25. Two forces $\mathbf{F}_1$ and $\mathbf{F}_2$ act on the screw eye. If their lines of action are at an angle θ apart and the magnitude of each force is $F_1 = F_2 = F$, determine the magnitude of the resultant force $\mathbf{F}_R$ and the angle between $\mathbf{F}_R$ and $\mathbf{F}_1$.

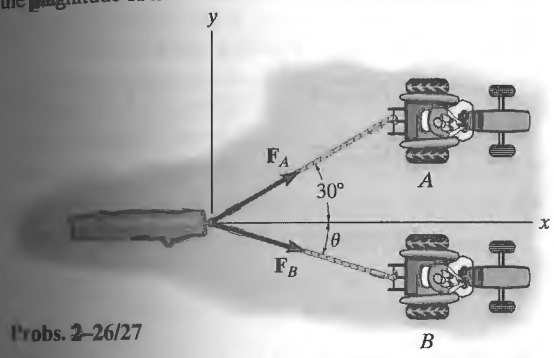


Prob. 2–25

2-26. The log is being towed by two tractors A and B. Determine the magnitudes of the two towing forces $\mathbf{F}_A$ and $\mathbf{F}_B$ if it is required that the resultant force have a magnitude $F_R = 10$ kN and be directed along the x axis. Set $\theta = 15°$.

Video Solutions

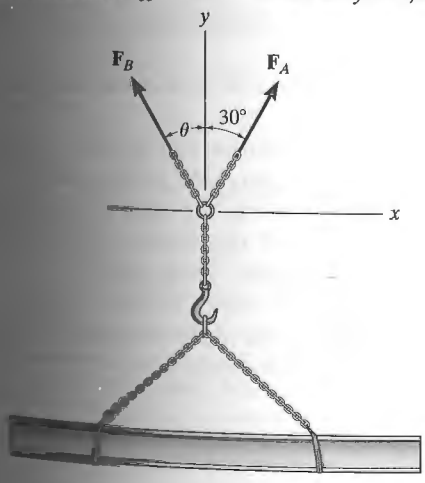
2-27. The resultant $\mathbf{F}_R$ of the two forces acting on the log is to be directed along the positive x axis and have a magnitude of 10 kN. Determine the angle θ of the cable, attached to B such that the magnitude of force $\mathbf{F}_B$ in this cable is a minimum. What is the magnitude of the force in each cable for this situation?



Probs. 2-26/27

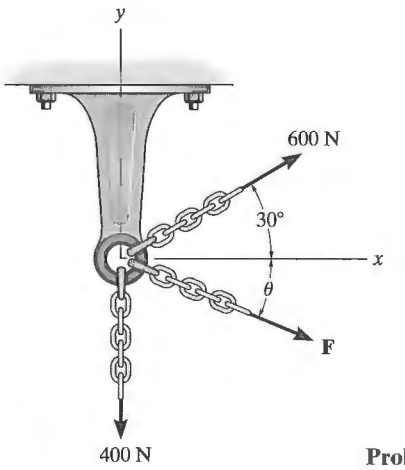
•2-28. The beam is to be hoisted using two chains. Determine the magnitudes of forces $\mathbf{F}_A$ and $\mathbf{F}_B$ acting on each chain in order to develop a resultant force of 600 N directed along the positive y axis. Set $\theta = 45°$.

•2-29. The beam is to be hoisted using two chains. If the resultant force is to be 600 N directed along the positive y axis, determine the magnitudes of forces $\mathbf{F}_A$ and $\mathbf{F}_B$ acting on each chain and the angle θ of $\mathbf{F}_B$ so that the magnitude of $\mathbf{F}_B$ is a *minimum*. $\mathbf{F}_A$ acts at 30° from the y axis, as shown.



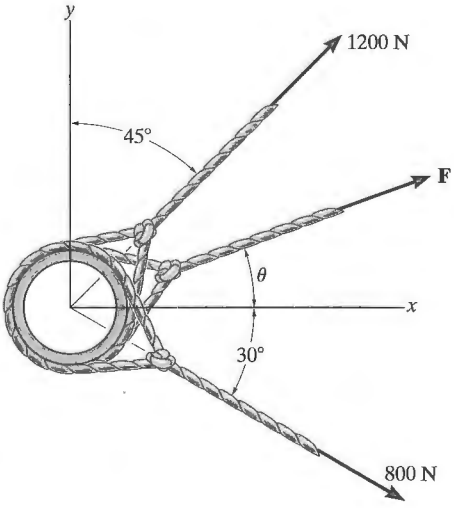
Probs. 2-28/29

2-30. Three chains act on the bracket such that they create a resultant force having a magnitude of 1000 N. If two of the chains are subjected to known forces, as shown, determine the angle θ of the third chain measured clockwise from the positive x axis, so that the magnitude of force $\mathbf{F}$ in this chain is a *minimum*. All forces lie in the x–y plane. What is the magnitude of $\mathbf{F}$? *Hint*: First find the resultant of the two known forces. Force $\mathbf{F}$ acts in this direction.



Prob. 2-30

2-31. Three cables pull on the pipe such that they create a resultant force having a magnitude of 1800 N. If two of the cables are subjected to known forces, as shown in the figure, determine the angle θ of the third cable so that the magnitude of force $\mathbf{F}$ in this cable is a *minimum*. All forces lie in the x–y plane. What is the magnitude of $\mathbf{F}$? *Hint*: First find the resultant of the two known forces.



Prob. 2-31

2.4 Addition of a System of Coplanar Forces

When a force is resolved into two components along the x and y axes, the components are then called *rectangular components*. For analytical work we can represent these components in one of two ways, using either scalar notation or Cartesian vector notation.

Scalar Notation. The rectangular components of force $\mathbf{F}$ shown in Fig. 2–15*a* are found using the parallelogram law, so that $\mathbf{F} = \mathbf{F}_x + \mathbf{F}_y$. Because these components form a right triangle, their magnitudes can be determined from

$$F_x = F\cos\theta \qquad \text{and} \qquad F_y = F\sin\theta$$

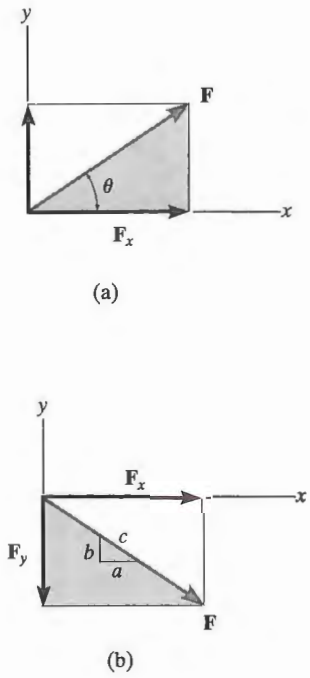


Fig. 2–15

Instead of using the angle θ, however, the direction of $\mathbf{F}$ can also be defined using a small "slope" triangle, such as shown in Fig. 2–15*b*. Since this triangle and the larger shaded triangle are similar, the proportional length of the sides gives

$$\frac{F_x}{F} = \frac{a}{c}$$

or

$$F_x = F\left(\frac{a}{c}\right)$$

and

$$\frac{F_y}{F} = \frac{b}{c}$$

or

$$F_y = -F\left(\frac{b}{c}\right)$$

Here the y component is a negative scalar since $\mathbf{F}_y$ is directed along the negative y axis.

It is important to keep in mind that this positive and negative scalar notation is to be used only for computational purposes, not for graphical representations in figures. Throughout the book, the *head of a vector arrow* in any figure indicates the sense of the vector *graphically*; algebraic signs are not used for this purpose. Thus, the vectors in Figs. 2–15*a* and 2–15*b* are designated by using boldface (vector) notation.* Whenever italic symbols are written near vector arrows in figures, they indicate the *magnitude* of the vector, which is *always* a *positive* quantity.

*Negative signs are used only in figures with boldface notation when showing equal but opposite pairs of vectors, as in Fig. 2–2.

Cartesian Vector Notation. It is also possible to represent the x and y components of a force in terms of Cartesian unit vectors **i** and **j**. Each of these unit vectors has a dimensionless magnitude of one, and so they can be used to designate the *directions* of the x and y axes, respectively, Fig. 2–16.*

Since the *magnitude* of each component of **F** is *always a positive quantity*, which is represented by the (positive) scalars F_x and F_y, then we can express **F** as a *Cartesian vector*,

$$\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j}$$

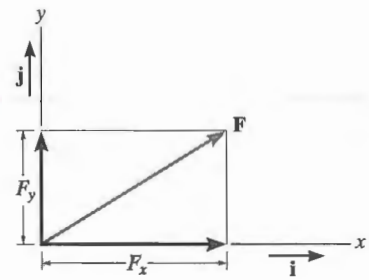


Fig. 2–16

Coplanar Force Resultants. We can use either of the two methods just described to determine the resultant of several *coplanar forces*. To do this, each force is first resolved into its x and y components, and then the respective components are added using *scalar algebra* since they are collinear. The resultant force is then formed by adding the resultant components using the parallelogram law. For example, consider the three concurrent forces in Fig. 2–17*a*, which have x and y components shown in Fig. 2–17*b*. Using *Cartesian vector notation*, each force is first represented as a Cartesian vector, i.e.,

$$\mathbf{F}_1 = F_{1x}\mathbf{i} + F_{1y}\mathbf{j}$$
$$\mathbf{F}_2 = -F_{2x}\mathbf{i} + F_{2y}\mathbf{j}$$
$$\mathbf{F}_3 = F_{3x}\mathbf{i} - F_{3y}\mathbf{j}$$

The vector resultant is therefore

$$\begin{aligned}\mathbf{F}_R &= \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3\\ &= F_{1x}\mathbf{i} + F_{1y}\mathbf{j} - F_{2x}\mathbf{i} + F_{2y}\mathbf{j} + F_{3x}\mathbf{i} - F_{3y}\mathbf{j}\\ &= (F_{1x} - F_{2x} + F_{3x})\mathbf{i} + (F_{1y} + F_{2y} - F_{3y})\mathbf{j}\\ &= (F_{Rx})\mathbf{i} + (F_{Ry})\mathbf{j}\end{aligned}$$

If *scalar notation* is used, then we have

$$(\overset{+}{\rightarrow}) \qquad F_{Rx} = F_{1x} - F_{2x} + F_{3x}$$
$$(+\uparrow) \qquad F_{Ry} = F_{1y} + F_{2y} - F_{3y}$$

These are the *same* results as the **i** and **j** components of $\mathbf{F}_R$ determined above.

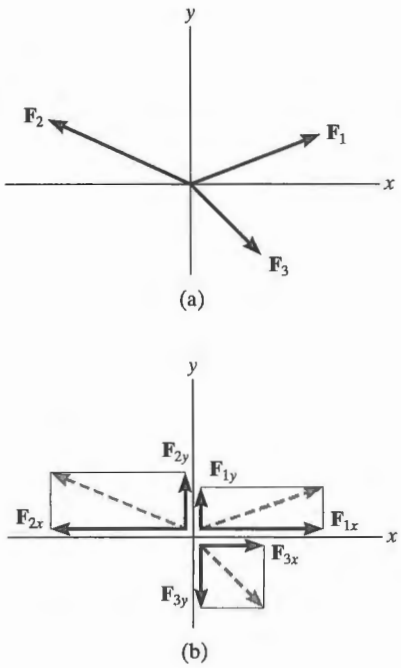
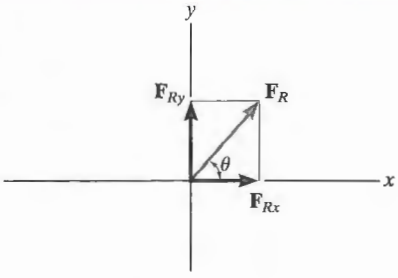


Fig. 2–17

*For handwritten work, unit vectors are usually indicated using a circumflex, e.g., $\hat{i}$ and $\hat{j}$. These vectors have a dimensionless magnitude of unity, and their sense (or arrowhead) will be described analytically by a plus or minus sign, depending on whether they are pointing along the positive or negative x or y axis.

We can represent the components of the resultant force of any number of coplanar forces symbolically by the algebraic sum of the x and y components of all the forces, i.e.,

$$\boxed{\begin{aligned} F_{Rx} &= \Sigma F_x \\ F_{Ry} &= \Sigma F_y \end{aligned}} \qquad (2\text{–}1)$$



(c)

Fig. 2–17

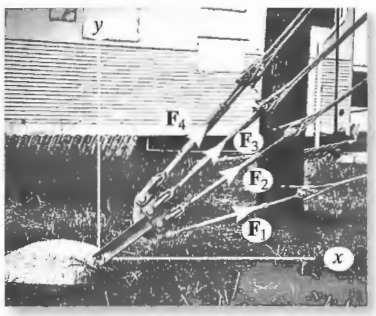
Once these components are determined, they may be sketched along the x and y axes with their proper sense of direction, and the resultant force can be determined from vector addition, as shown in Fig. 2–17. From this sketch, the magnitude of $\mathbf{F}_R$ is then found from the Pythagorean theorem; that is,

$$F_R = \sqrt{F_{Rx}^2 + F_{Ry}^2}$$

Also, the angle θ, which specifies the direction of the resultant force, is determined from trigonometry:

$$\theta = \tan^{-1}\left|\frac{F_{Ry}}{F_{Rx}}\right|$$

The above concepts are illustrated numerically in the examples which follow.



The resultant force of the four cable forces acting on the supporting bracket can be determined by adding algebraically the separate x and y components of each cable force. This resultant FR produces the same pulling effect on the bracket as all four cables.

Important Points

- The resultant of several coplanar forces can easily be determined if an x, y coordinate system is established and the forces are resolved along the axes.
- The direction of each force is specified by the angle its line of action makes with one of the axes, or by a sloped triangle.
- The orientation of the x and y axes is arbitrary, and their positive direction can be specified by the Cartesian unit vectors **i** and **j**.
- The x and y components of the *resultant force* are simply the algebraic addition of the components of all the coplanar forces.
- The magnitude of the resultant force is determined from the Pythagorean theorem, and when the components are sketched on the x and y axes, the direction can be determined from trigonometry.

exclusively for The University of Melbourne

EXAMPLE 2.5

Determine the x and y components of $\mathbf{F}_1$ and $\mathbf{F}_2$ acting on the boom shown in Fig. 2–18a. Express each force as a Cartesian vector.

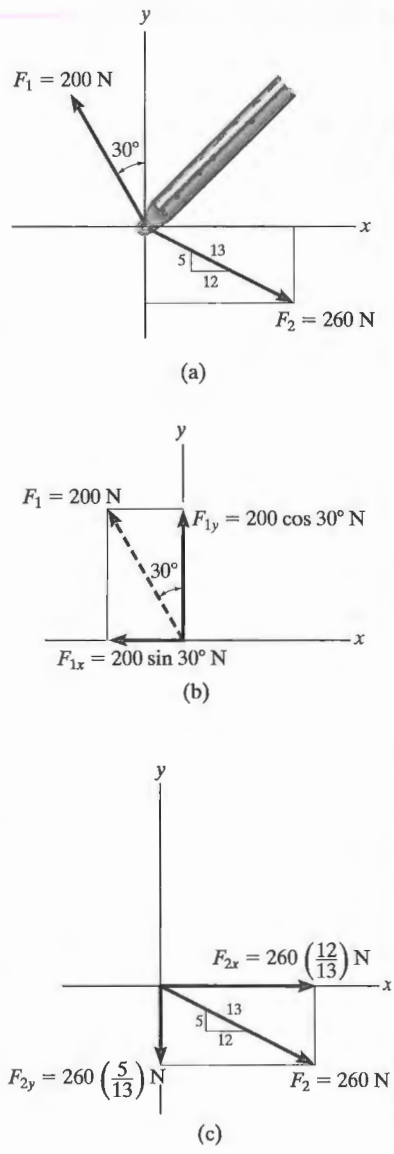


Fig. 2–18

SOLUTION

Scalar Notation. By the parallelogram law, $\mathbf{F}_1$ is resolved into x and y components, Fig. 2–18b. Since $\mathbf{F}_{1x}$ acts in the $-x$ direction, and $\mathbf{F}_{1y}$ acts in the $+y$ direction, we have

$$F_{1x} = -200 \sin 30° \text{ N} = -100 \text{ N} = 100 \text{ N} \leftarrow \qquad \textit{Ans.}$$

$$F_{1y} = 200 \cos 30° \text{ N} = 173 \text{ N} = 173 \text{ N} \uparrow \qquad \textit{Ans.}$$

The force $\mathbf{F}_2$ is resolved into its x and y components as shown in Fig. 2–18c. Here the *slope* of the line of action for the force is indicated. From this "slope triangle" we could obtain the angle θ, e.g., $\theta = \tan^{-1}(\frac{5}{12})$, and then proceed to determine the magnitudes of the components in the same manner as for $\mathbf{F}_1$. The easier method, however, consists of using proportional parts of similar triangles, i.e.,

$$\frac{F_{2x}}{260 \text{ N}} = \frac{12}{13} \qquad F_{2x} = 260 \text{ N}\left(\frac{12}{13}\right) = 240 \text{ N}$$

Similarly,

$$F_{2y} = 260 \text{ N}\left(\frac{5}{13}\right) = 100 \text{ N}$$

Notice how the magnitude of the *horizontal component*, F_{2x}, was obtained by multiplying the force magnitude by the ratio of the *horizontal leg* of the slope triangle divided by the hypotenuse; whereas the magnitude of the *vertical component*, F_{2y}, was obtained by multiplying the force magnitude by the ratio of the *vertical leg* divided by the hypotenuse. Hence,

$$F_{2x} = 240 \text{ N} = 240 \text{ N} \rightarrow \qquad \textit{Ans.}$$

$$F_{2y} = -100 \text{ N} = 100 \text{ N} \downarrow \qquad \textit{Ans.}$$

Cartesian Vector Notation. Having determined the magnitudes and directions of the components of each force, we can express each force as a Cartesian vector.

$$\mathbf{F}_1 = \{-100\mathbf{i} + 173\mathbf{j}\} \text{ N} \qquad \textit{Ans.}$$

$$\mathbf{F}_2 = \{240\mathbf{i} - 100\mathbf{j}\} \text{ N} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 2.6

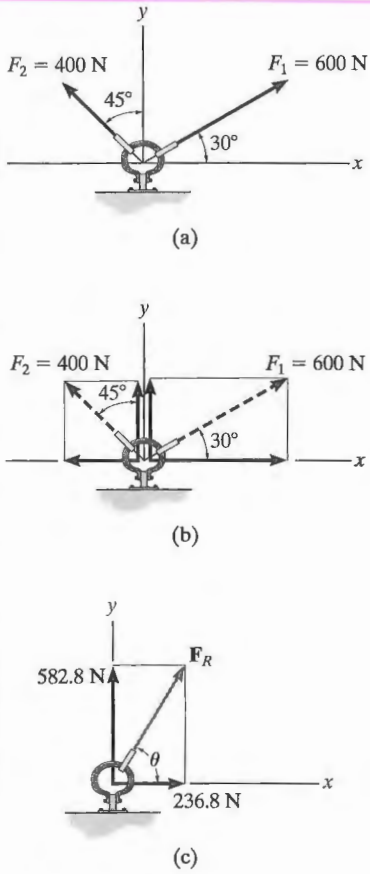


Fig. 2–19

The link in Fig. 2–19*a* is subjected to two forces $\mathbf{F}_1$ and $\mathbf{F}_2$. Determine the magnitude and direction of the resultant force.

SOLUTION I

Scalar Notation. First we resolve each force into its x and y components, Fig. 2–19*b*, then we sum these components algebraically.

$$\overset{+}{\rightarrow} F_{Rx} = \Sigma F_x; \qquad F_{Rx} = 600 \cos 30° \text{ N} - 400 \sin 45° \text{ N}$$
$$= 236.8 \text{ N} \rightarrow$$

$$+\uparrow F_{Ry} = \Sigma F_y; \qquad F_{Ry} = 600 \sin 30° \text{ N} + 400 \cos 45° \text{ N}$$
$$= 582.8 \text{ N} \uparrow$$

The resultant force, shown in Fig. 2–18*c*, has a *magnitude* of

$$F_R = \sqrt{(236.8 \text{ N})^2 + (582.8 \text{ N})^2}$$
$$= 629 \text{ N} \qquad \textit{Ans.}$$

From the vector addition,

$$\theta = \tan^{-1}\left(\frac{582.8 \text{ N}}{236.8 \text{ N}}\right) = 67.9° \qquad \textit{Ans.}$$

SOLUTION II

Cartesian Vector Notation. From Fig. 2–19*b*, each force is first expressed as a Cartesian vector.

$$\mathbf{F}_1 = \{600 \cos 30°\mathbf{i} + 600 \sin 30°\mathbf{j}\} \text{ N}$$
$$\mathbf{F}_2 = \{-400 \sin 45°\mathbf{i} + 400 \cos 45°\mathbf{j}\} \text{ N}$$

Then,

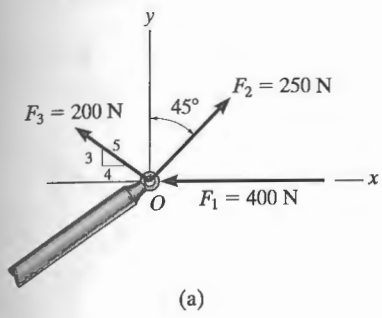
$$\mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2 = (600 \cos 30° \text{ N} - 400 \sin 45° \text{ N})\mathbf{i}$$
$$+ (600 \sin 30° \text{ N} + 400 \cos 45° \text{ N})\mathbf{j}$$
$$= \{236.8\mathbf{i} + 582.8\mathbf{j}\} \text{ N}$$

The magnitude and direction of $\mathbf{F}_R$ are determined in the same manner as before.

NOTE: Comparing the two methods of solution, notice that the use of scalar notation is more efficient since the components can be found *directly*, without first having to express each force as a Cartesian vector before adding the components. Later, however, we will show that Cartesian vector analysis is very beneficial for solving three-dimensional problems.

EXAMPLE 2.7

The end of the boom O in Fig. 2–20a is subjected to three concurrent and coplanar forces. Determine the magnitude and direction of the resultant force.



(a)

SOLUTION

Each force is resolved into its x and y components, Fig. 2–20b. Summing the x components, we have

$$\xrightarrow{+} F_{Rx} = \Sigma F_x; \qquad F_{Rx} = -400\text{ N} + 250\sin 45°\text{ N} - 200\left(\tfrac{4}{5}\right)\text{N}$$

$$= -383.2\text{ N} = 383.2\text{ N}\leftarrow$$

The negative sign indicates that F_{Rx} acts to the left, i.e., in the negative x direction, as noted by the small arrow. Obviously, this occurs because F_1 and F_3 in Fig. 2–20b contribute a greater pull to the left than F_2 which pulls to the right. Summing the y components yields

$$+\uparrow F_{Ry} = \Sigma F_y; \qquad F_{Ry} = 250\cos 45°\text{ N} + 200\left(\tfrac{3}{5}\right)\text{N}$$

$$= 296.8\text{ N}\uparrow$$

The resultant force, shown in Fig. 2–20c, has a *magnitude* of

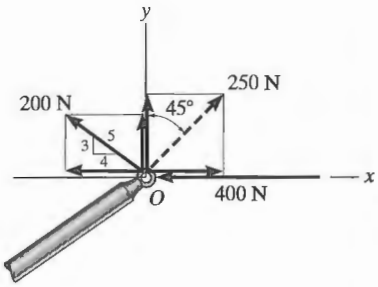
$$F_R = \sqrt{(-383.2\text{ N})^2 + (296.8\text{ N})^2}$$

$$= 485\text{ N} \qquad \textit{Ans.}$$

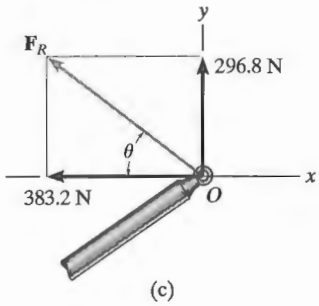
From the vector addition in Fig. 2–20c, the direction angle θ is

$$\theta = \tan^{-1}\left(\frac{296.8}{383.2}\right) = 37.8° \qquad \textit{Ans.}$$

NOTE: Application of this method is more convenient, compared to using two applications of the parallelogram law, first to add $\mathbf{F}_1$ and $\mathbf{F}_2$ then adding $\mathbf{F}_3$ to this resultant.



(b)

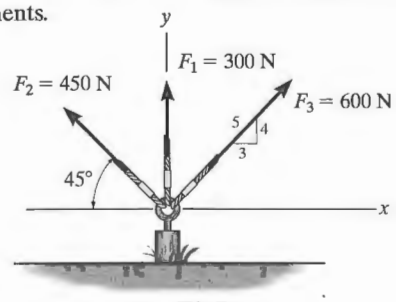


(c)

Fig. 2–20

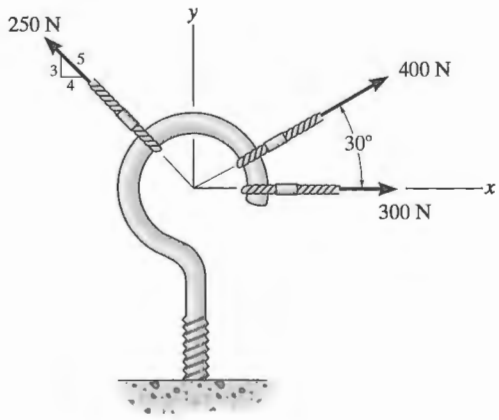
FUNDAMENTAL PROBLEMS

F2–7. Resolve each force acting on the post into its x and y components.



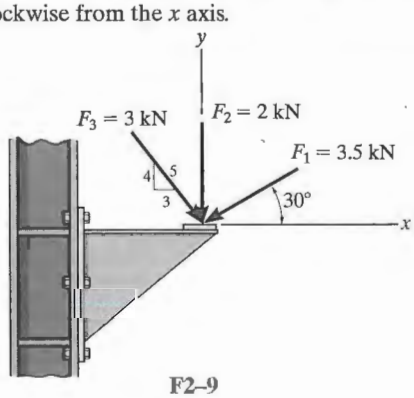
F2–7

F2–8. Determine the magnitude and direction of the resultant force.



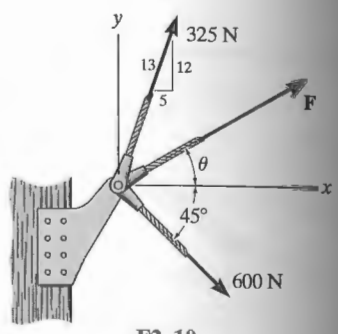
F2–8

F2–9. Determine the magnitude of the resultant force acting on the corbel and its direction θ measured counterclockwise from the x axis.



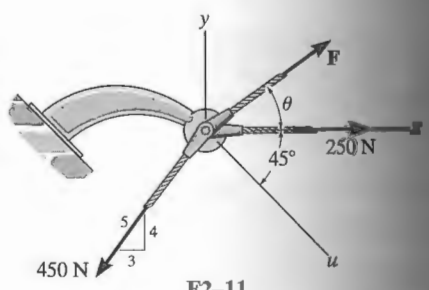
F2–9

F2–10. If the resultant force acting on the bracket is to be 750 N directed along the positive x axis, determine the magnitude of F and its direction θ.



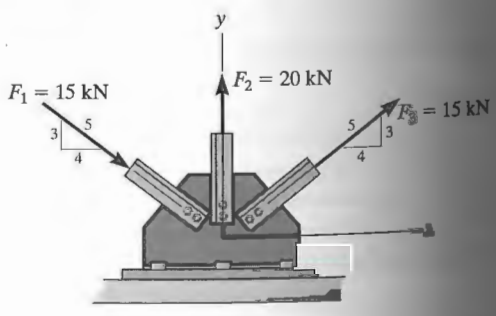
F2–10

F2–11. If the magnitude of the resultant force acting on the bracket is to be 400 N directed along the u axis determine the magnitude of F and its direction θ.



F2–11

F2–12. Determine the magnitude of the resultant force and its direction θ measured counterclockwise from the positive x axis.



F2–12

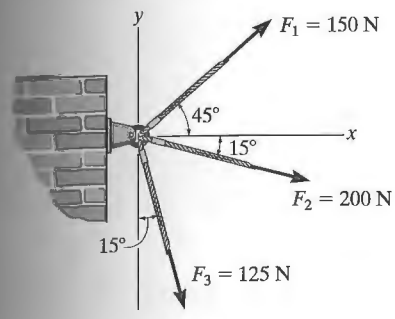
Exclusively for The University of Melbourne

MASTERING ENGINEERING
Customized coaching is available!

PROBLEMS

*2–32. Determine the magnitude of the resultant force acting on the pin and its direction measured clockwise from the positive x axis.

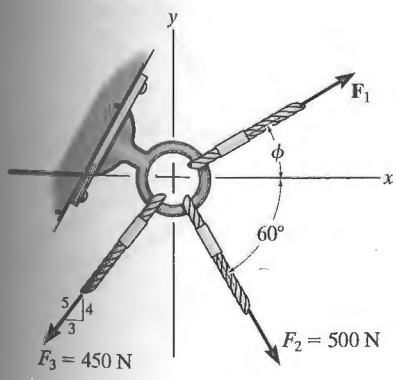
Video Solutions



Prob. 2–32

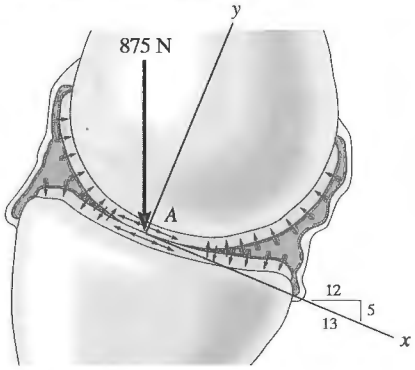
•2–33. If $F_1 = 600$ N and $\phi = 30°$, determine the magnitude of the resultant force acting on the eyebolt and its direction measured clockwise from the positive x axis.

2–34. If the magnitude of the resultant force acting on the eyebolt is 600 N and its direction measured clockwise from the positive x axis is $\theta = 30°$, determine the magnitude of $\mathbf{F}_1$ and the angle ϕ.



Probs. 2–33/34

2–35. The contact point between the femur and tibia bones of the leg is at A. If a vertical force of 875 N is applied at this point, determine the components along the x and y axes. Note that the y component represents the normal force on the load-bearing region of the bones. Both the x and y components of this force cause synovial fluid to be squeezed out of the bearing space.

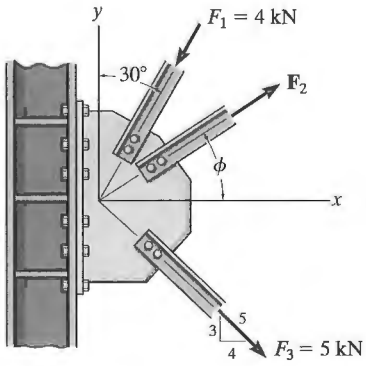


Prob. 2–35

*2–36. If $\phi = 30°$ and $F_2 = 3$ kN, determine the magnitude of the resultant force acting on the plate and its direction θ measured clockwise from the positive x axis.

•2–37. If the magnitude for the resultant force acting on the plate is required to be 6 kN and its direction measured clockwise from the positive x axis is $\theta = 30°$, determine the magnitude of $\mathbf{F}_2$ and its direction ϕ.

2–38. If $\phi = 30°$ and the resultant force acting on the gusset plate is directed along the positive x axis, determine the magnitudes of $\mathbf{F}_2$ and the resultant force.



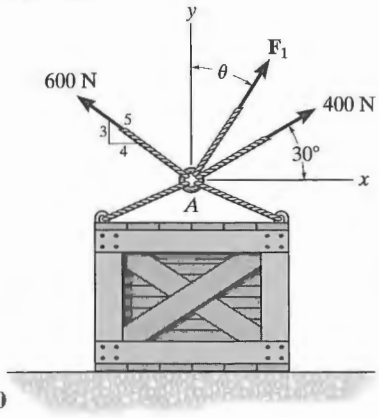
Probs. 2–36/37/38

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

2–39. Determine the magnitude of $\mathbf{F}_1$ and its direction θ so that the resultant force is directed vertically upward and has a magnitude of 800 N.

Video Solutions

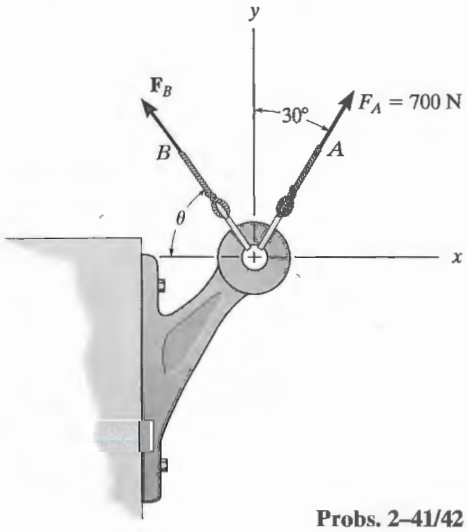
***2–40.** Determine the magnitude and direction measured counterclockwise from the positive x axis of the resultant force of the three forces acting on the ring A. Take $F_1 = 500$ N and $\theta = 20°$.



Probs. 2–39/40

•2–41. Determine the magnitude and direction θ of $\mathbf{F}_B$ so that the resultant force is directed along the positive y axis and has a magnitude of 1500 N.

2–42. Determine the magnitude and angle measured counterclockwise from the positive y axis of the resultant force acting on the bracket if $F_B = 600$ N and $\theta = 20°$.

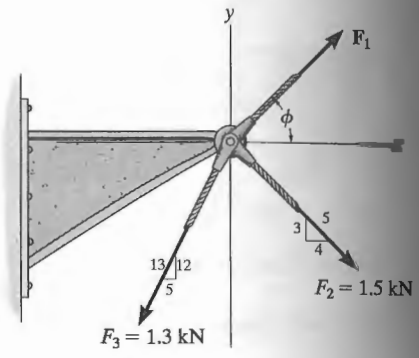


Probs. 2–41/42

2–43. If $\phi = 30°$ and $F_1 = 1.25$ kN, determine the magnitude of the resultant force acting on the bracket and its direction measured clockwise from the positive x axis.

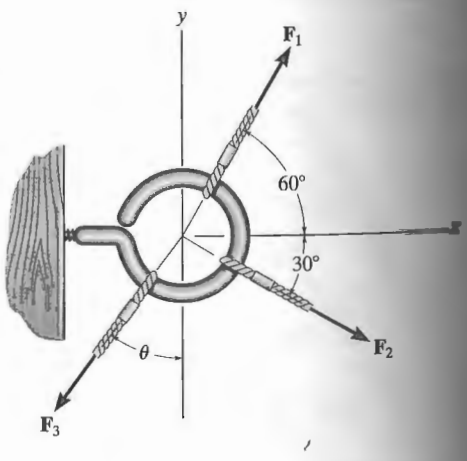
***2–44.** If the magnitude of the resultant force acting on the bracket is 2 kN directed along the positive x axis, determine the magnitude of $\mathbf{F}_1$ and its direction ϕ.

•2–45. If the resultant force acting on the bracket is to be directed along the positive x axis and the magnitude of $\mathbf{F}_1$ is required to be a minimum, determine the magnitude of the resultant force and $\mathbf{F}_1$.



Probs. 2–43/44/45

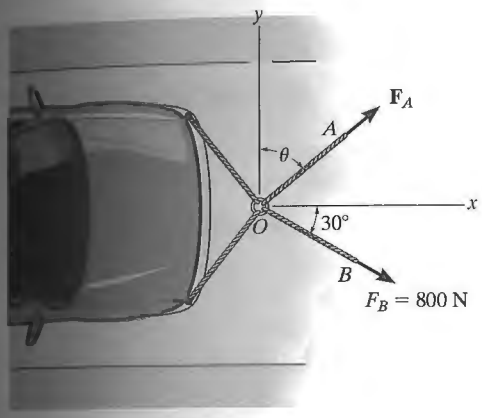
2–46. The three concurrent forces acting on the screw eye produce a resultant force $\mathbf{F}_R = 0$. If $F_2 = \frac{2}{3} F_1$ and $\mathbf{F}_1$ is to be 90° from $\mathbf{F}_2$ as shown, determine the required magnitude of $\mathbf{F}_3$ expressed in terms of F_1 and the angle θ.



Prob. 2–46

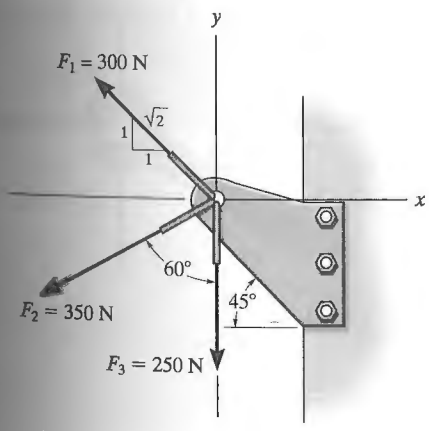
2–47. Determine the magnitude of $\mathbf{F}_A$ and its direction θ so that the resultant force is directed along the positive x axis and has a magnitude of 1250 N.

***2–48.** Determine the magnitude and direction measured counterclockwise from the positive x axis of the resultant force acting on the ring at O if $F_A = 750$ N and $\theta = 45°$.



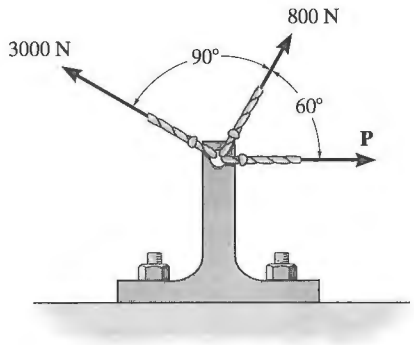
Probs. 2–47/48

•2–49. Determine the magnitude of the resultant force and its direction measured counterclockwise from the positive x axis.



Prob. 2–49

2–50. The three forces are applied to the bracket. Determine the range of values for the magnitude of force $\mathbf{P}$ so that the resultant of the three forces does not exceed 2400 N.

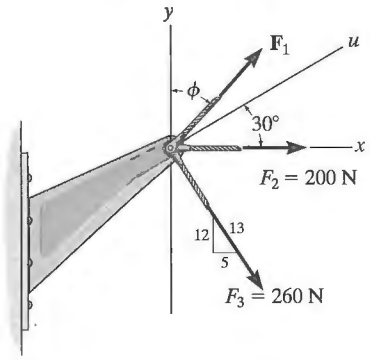


Prob. 2–50

2–51. If $F_1 = 150$ N and $\phi = 30°$, determine the magnitude of the resultant force acting on the bracket and its direction measured clockwise from the positive x axis.

***2–52.** If the magnitude of the resultant force acting on the bracket is to be 450 N directed along the positive u axis, determine the magnitude of $\mathbf{F}_1$ and its direction ϕ.

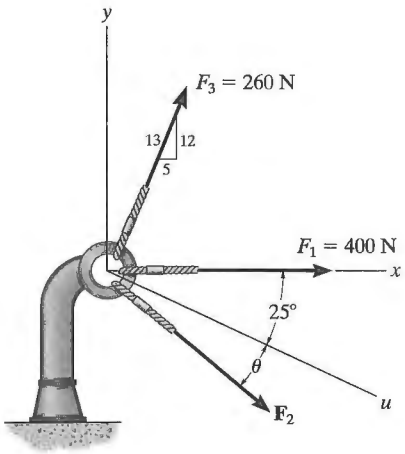
•2–53. If the resultant force acting on the bracket is required to be a minimum, determine the magnitudes of $\mathbf{F}_1$ and the resultant force. Set $\phi = 30°$.



Probs. 2–51/52/53

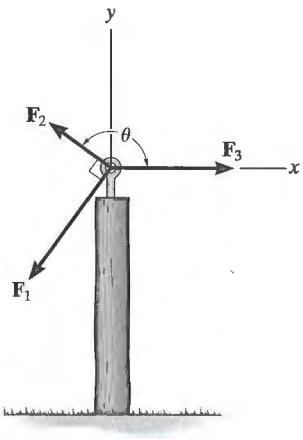
2–54. Three forces act on the bracket. Determine the magnitude and direction θ of $\mathbf{F}_2$ so that the resultant force is directed along the positive u axis and has a magnitude of 250 N.

2–55. If $F_2 = 750$ N and $\theta = 55°$, determine the magnitude and direction measured clockwise from the positive x axis of the resultant force of the three forces acting on the bracket.



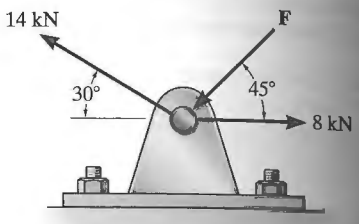
Probs. 2–54/55

***2–56.** The three concurrent forces acting on the post produce a resultant force $\mathbf{F}_R = \mathbf{0}$. If $F_2 = \frac{1}{2}F_1$, and $\mathbf{F}_1$ is to be 90° from $\mathbf{F}_2$ as shown, determine the required magnitude of F_3 expressed in terms of F_1 and the angle θ.



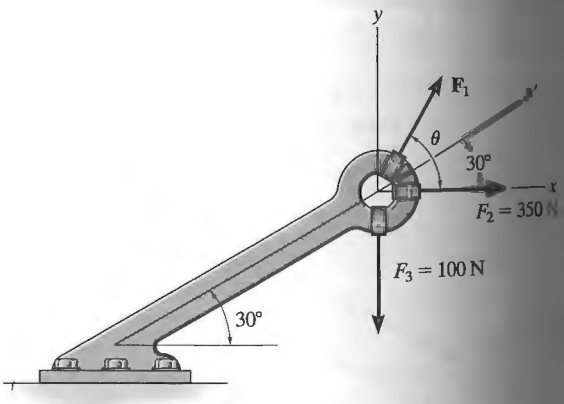
Prob. 2–56

•2–57. Determine the magnitude of force $\mathbf{F}$ so that the resultant force of the three forces is as small as possible. What is the magnitude of this smallest resultant force?



Prob. 2–57

2–58. Express each of the three forces acting on the bracket in Cartesian vector form with respect to the x and y axes. Determine the magnitude and direction θ of $\mathbf{F}_1$ so that the resultant force is directed along the positive x' axis and has a magnitude of $F_R = 600$ N.



Prob. 2–58

exclusively for The University of Melbourne

2.5 Cartesian Vectors

The operations of vector algebra, when applied to solving problems in [redacted]*sions*, are greatly simplified if the vectors are first represented in Cartesian vector form. In this section we will present a general method for doing this; then in the next section we will use this method for finding the resultant force of a system of concurrent forces.

Right-Handed Coordinate System. We will use a right-handed [redacted]ordinate system to develop the theory of vector algebra that follows. A [redacted]tangular coordinate system is said to be *right-handed* if the thumb of the right hand points in the direction of the positive z axis when the [redacted]ht-hand fingers are curled about this axis and directed from the positive x towards the positive y axis, Fig. 2–21.

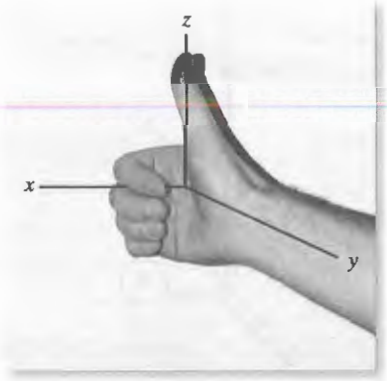


Fig. 2–21

Rectangular [redacted]omponents of a Vector. A vector **A** may have one, two, or three rectangular components along the x, y, z coordinate axes, depending on how the vector is oriented relative to the axes. In general, though, when **A** is directed within an octant of the x, y, z frame, Fig. 2–22, then by two successive applications of the parallelogram law, we may resolve the vector into components as $\mathbf{A} = \mathbf{A}' + \mathbf{A}_z$ and then $\mathbf{A}' = \mathbf{A}_x + \mathbf{A}_y$. Combining these equations, to eliminate $\mathbf{A}'$, **A** is represented by the vector sum of its *three* rectangular components,

$$\mathbf{A} = \mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z \tag{2–2}$$

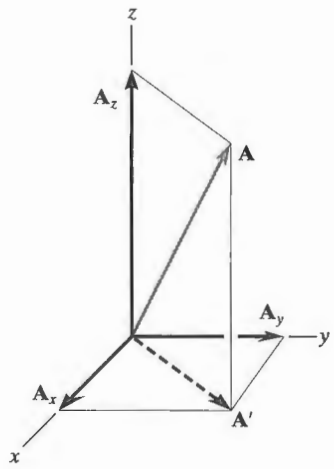


Fig. 2–22

Cartesian Unit Vectors. In three dimensions, the set of Cartesian unit vectors, **i**, **j**, **k**, is used to designate the directions of the x, y, z axes, respectively. As stated in Sec. 2.4, the *sense* (or arrowhead) of these vectors will be represented analytically by a plus or minus sign, depending on whether they are directed along the positive or negative x, y or z axes. The positive Cartesian unit vectors are shown in Fig. 2–23.

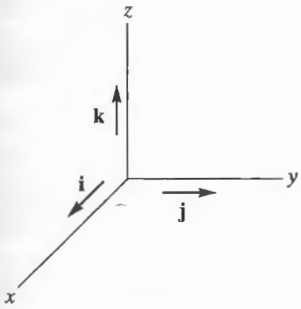


Fig. 2–23

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

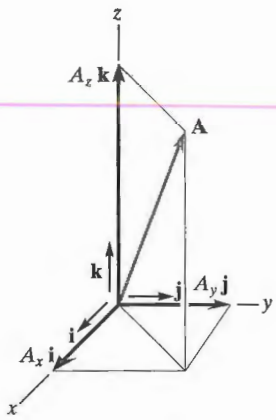


Fig. 2–24

Cartesian Vector Representation. Since the three components of **A** in Eq. 2–2 act in the positive **i**, **j**, and **k** directions, Fig. 2–24, we can write **A** in Cartesian vector form as

$$\boxed{\mathbf{A} = A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}} \tag{2–3}$$

There is a distinct advantage to writing vectors in this manner. Separating the *magnitude* and *direction* of each *component vector* will simplify the operations of vector algebra, particularly in three dimensions.

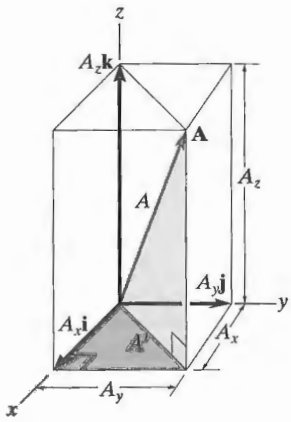


Fig. 2–25

Magnitude of a Cartesian Vector. It is always possible to obtain the magnitude of **A** provided it is expressed in Cartesian vector form. As shown in Fig. 2–25, from the blue right triangle, $A = \sqrt{A'^2 + A_z^2}$, and from the gray right triangle, $A' = \sqrt{A_x^2 + A_y^2}$. Combining these equations to eliminate A', yields

$$\boxed{A = \sqrt{A_x^2 + A_y^2 + A_z^2}} \tag{2–4}$$

Hence, the magnitude of **A** *is equal to the positive square root of the sum of the squares of its components.*

Direction of a Cartesian Vector. We will define the *direction* of **A** by the *coordinate direction angles* α (alpha), β (beta), and γ (gamma), measured between the *tail* of **A** and the *positive* x, y, z axes provided they are located at the tail of **A**, Fig. 2–26. Note that regardless of where **A** is directed, each of these angles will be between 0° and 180°.

To determine α, β, and γ, consider the projection of **A** onto the x, y, z axes, Fig. 2–27. Referring to the blue colored right triangles shown in each figure, we have

$$\boxed{\cos\alpha = \frac{A_x}{A} \qquad \cos\beta = \frac{A_y}{A} \qquad \cos\gamma = \frac{A_z}{A}} \tag{2–5}$$

These numbers are known as the *direction cosines* of **A**. Once they have been obtained, the coordinate direction angles α, β, γ can then be determined from the inverse cosines.

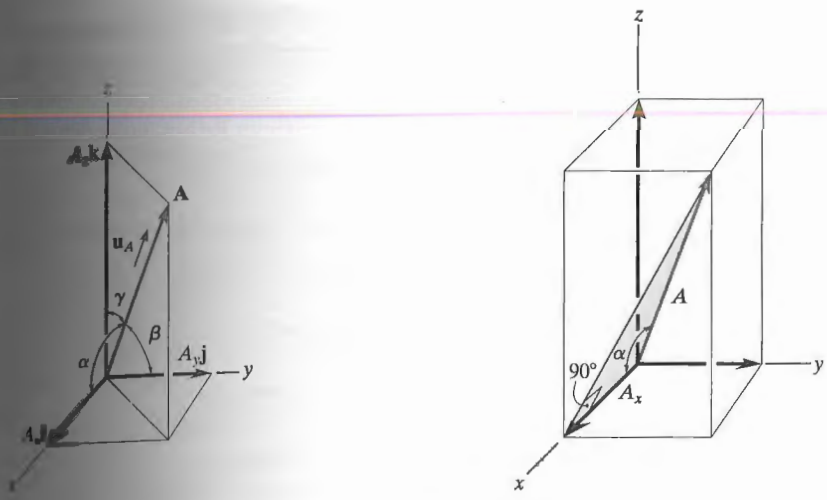


Fig. 2–26

An easy way of obtaining these direction cosines is to form a unit vector $\mathbf{u}_A$ in the direction of **A**, Fig. 2–26. If **A** is expressed in Cartesian vector form, $\mathbf{A} = A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}$, then $\mathbf{u}_A$ will have a magnitude of one and be dimensionless provided **A** is divided by its magnitude, i.e.,

$$\boxed{\frac{\mathbf{A}}{A} = \frac{A_x}{A}\mathbf{i} + \frac{A_y}{A}\mathbf{j} + \frac{A_z}{A}\mathbf{k}} \qquad (2\text{–}6)$$

where $A = \sqrt{A_x^2 + A_y^2 + A_z^2}$. By comparison with Eqs. 2–7, it is seen that *the* **i**, **j**, **k** *components of* $\mathbf{u}_A$ *represent the direction cosines of* **A**, i.e.,

$$\mathbf{u}_A = \cos\alpha\,\mathbf{i} + \cos\beta\,\mathbf{j} + \cos\gamma\,\mathbf{k} \qquad (2\text{–}7)$$

Since the magnitude of a vector is equal to the positive square root of the sum of the squares of the magnitudes of its components, and $\mathbf{u}_A$ has a magnitude of one, then from the above equation an important relation between the direction cosines can be formulated as

$$\boxed{\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1} \qquad (2\text{–}8)$$

Here we can see that if only *two* of the coordinate angles are known, the third angle can be found using this equation.

Finally, if the magnitude and coordinate direction angles of **A** are known, then **A** may be expressed in Cartesian vector form as

$$\boxed{\begin{aligned}\mathbf{A} &= A\mathbf{u}_A \\ &= A\cos\alpha\,\mathbf{i} + A\cos\beta\,\mathbf{j} + A\cos\gamma\,\mathbf{k} \\ &= A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}\end{aligned}} \qquad (2\text{–}9)$$

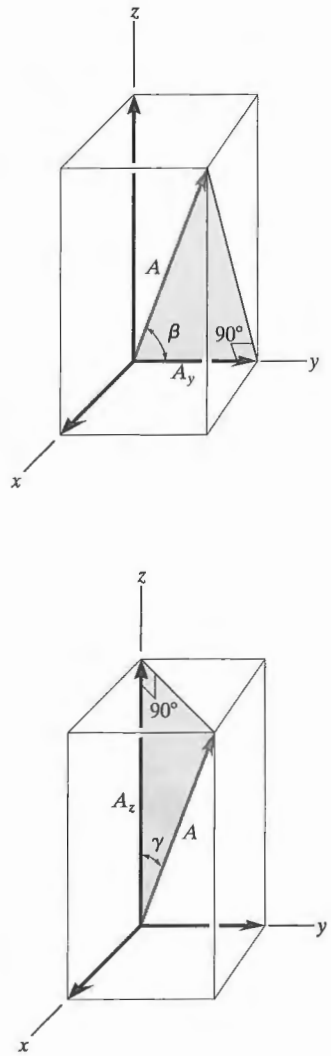


Fig 2–27

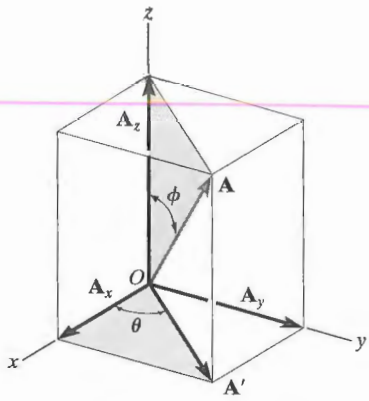


Fig. 2–28

Sometimes, the direction of **A** can be specified using two angles θ and ϕ (phi), such as shown in Fig. 2–28. The components of **A** can then be determined by applying trigonometry first to the blue right triangle, which yields

$$A_z = A \cos \phi$$

and

$$A' = A \sin \phi$$

Now applying trigonometry to the other shaded right triangle

$$A_x = A' \cos \theta = A \sin \phi \cos \theta$$

$$A_y = A' \sin \theta = A \sin \phi \sin \theta$$

Therefore **A** written in Cartesian vector form becomes

$$\mathbf{A} = A \sin \phi \cos \theta\, \mathbf{i} + A \sin \phi \sin \theta\, \mathbf{j} + A \cos \phi\, \mathbf{k}$$

You should not memorize this equation, rather it is important to understand how the components were determined using trigonometry.

2.6 Addition of Cartesian Vectors

The addition (or subtraction) of two or more vectors are greatly simplified if the vectors are expressed in terms of their Cartesian components. For example, if $\mathbf{A} = A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}$ and $\mathbf{B} = B_x\mathbf{i} + B_y\mathbf{j} + B_z\mathbf{k}$, Fig. 2–29, then the resultant vector, **R**, has components which are the scalar sums of the **i**, **j**, **k** components of **A** and **B**, i.e.,

$$\mathbf{R} = \mathbf{A} + \mathbf{B} = (A_x + B_x)\mathbf{i} + (A_y + B_y)\mathbf{j} + (A_z + B_z)\mathbf{k}$$

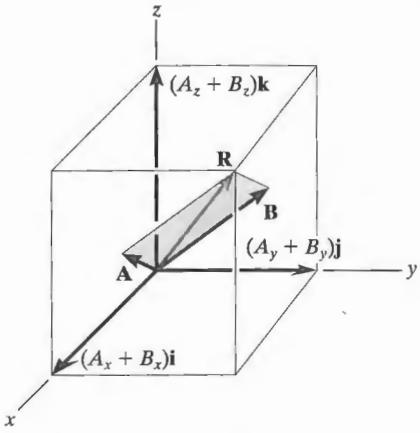


Fig. 2–29

If this is generalized and applied to a system of several concurrent forces, then the force resultant is the vector sum of all the forces in the system and can be written as

$$\mathbf{F}_R = \Sigma \mathbf{F} = \Sigma F_x\mathbf{i} + \Sigma F_y\mathbf{j} + \Sigma F_z\mathbf{k} \tag{2–10}$$

Here ΣF_x, ΣF_y, and ΣF_z represent the algebraic sums of the respective x, y, z or **i**, **j**, **k** components of each force in the system.

Important Points

- Cartesian vector analysis is often used to solve problems in three dimensions.
- The positive directions of the x, y, z axes are defined by the Cartesian unit vectors **i**, **j**, **k**, respectively.
- The *magnitude* of a Cartesian vector is $A = \sqrt{A_x^2 + A_y^2 + A_z^2}$.
- The *direction* of a Cartesian vector is specified using coordinate direction angles α, β, γ which the tail of the vector makes with the positive x, y, z axes, respectively. The components of the unit vector $\mathbf{u}_A = \mathbf{A}/A$ represent the direction cosines of α, β, γ. Only two of the angles α, β, γ have to be specified. The third angle is determined from the relationship $\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$.
- Sometimes the direction of a vector is defined using the two angles θ and ϕ as in Fig. 2–28. In this case, the vector components are obtained by vector resolution using trigonometry.
- To find the *resultant* of a concurrent force system, express each force as a Cartesian vector and add the **i**, **j**, **k** components of all the forces in the system.

The resultant force acting on the bow the ship can be determined by first representing each rope force as a Cartesian vector and then summing the **i**, **j**, and **k** components.

2

EXAMPLE 2.8

Express the force **F** shown in Fig. 2–30 as a Cartesian vector.

SOLUTION

Since only two coordinate direction angles are specified, the third angle α must be determined from Eq. 2–8; i.e.,

$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$$
$$\cos^2\alpha + \cos^2 60° + \cos^2 45° = 1$$
$$\cos\alpha = \sqrt{1 - (0.5)^2 - (0.707)^2} = \pm 0.5$$

Hence, two possibilities exist, namely,

$$\alpha = \cos^{-1}(0.5) = 60° \qquad \text{or} \qquad \alpha = \cos^{-1}(-0.5) = 120°$$

By inspection, it is necessary that $\alpha = 60°$, since $\mathbf{F}_x$ must be in the $+x$ direction.

Using Eq. 2–9, with $F = 200$ N, we have

$$\mathbf{F} = F\cos\alpha\mathbf{i} + F\cos\beta\mathbf{j} + F\cos\gamma\mathbf{k}$$
$$= (200\cos 60°\text{ N})\mathbf{i} + (200\cos 60°\text{ N})\mathbf{j} + (200\cos 45°\text{ N})\mathbf{k}$$
$$= \{100.0\mathbf{i} + 100.0\mathbf{j} + 141.4\mathbf{k}\}\text{ N} \qquad \textit{Ans.}$$

Show that indeed the magnitude of $F = 200$ N.

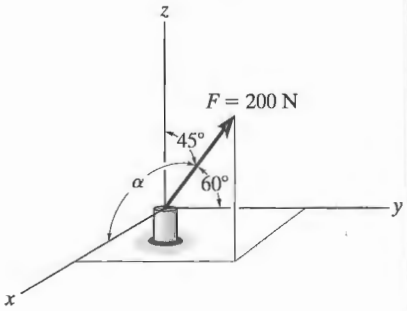


Fig. 2–30

EXAMPLE 2.9

Determine the magnitude and the coordinate direction angles of the resultant force acting on the ring in Fig. 2–31*a*.

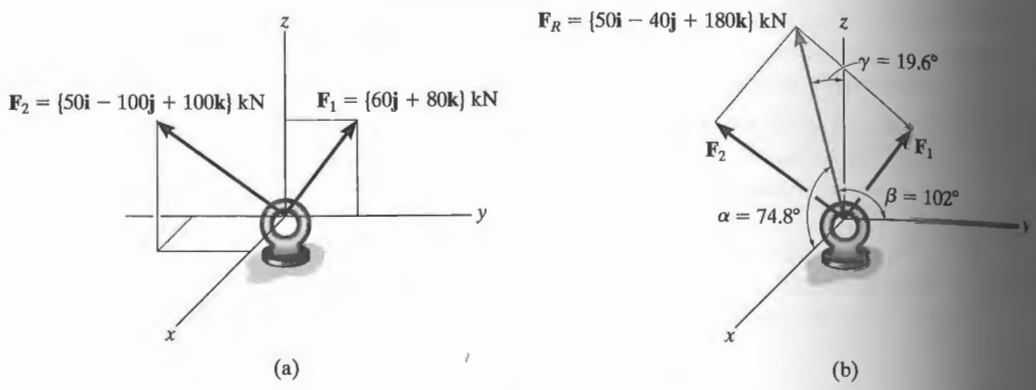


Fig. 2–31

SOLUTION

Since each force is represented in Cartesian vector form, the resultant force, shown in Fig. 2–31*b*, is

$$\mathbf{F}_R = \Sigma\mathbf{F} = \mathbf{F}_1 + \mathbf{F}_2 = \{60\mathbf{j} + 80\mathbf{k}\}\text{ kN} + \{50\mathbf{i} - 100\mathbf{j} + 100\mathbf{k}\}\text{ kN}$$
$$= \{50\mathbf{i} - 40\mathbf{j} + 180\mathbf{k}\}\text{ kN}$$

The magnitude of $\mathbf{F}_R$ is

$$F_R = \sqrt{(50\text{ kN})^2 + (-40\text{ kN})^2 + (180\text{ kN})^2} = 191.0\text{ kN}$$
$$= 191\text{ kN} \qquad \textit{Ans.}$$

The coordinate direction angles α, β, γ are determined from the components of the unit vector acting in the direction of $\mathbf{F}_R$.

$$\mathbf{u}_{F_R} = \frac{\mathbf{F}_R}{F_R} = \frac{50}{191.0}\mathbf{i} - \frac{40}{191.0}\mathbf{j} + \frac{180}{191.0}\mathbf{k}$$
$$= 0.2617\mathbf{i} - 0.2094\mathbf{j} + 0.9422\mathbf{k}$$

so that

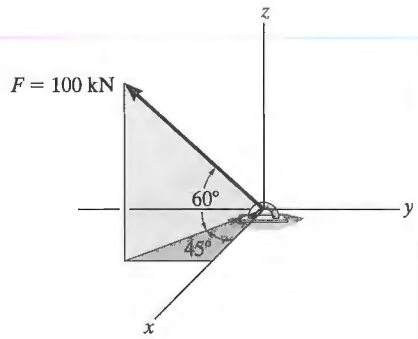
$$\cos\alpha = 0.2617 \qquad \alpha = 74.8° \qquad \textit{Ans.}$$
$$\cos\beta = -0.2094 \qquad \beta = 102° \qquad \textit{Ans.}$$
$$\cos\gamma = 0.9422 \qquad \gamma = 19.6° \qquad \textit{Ans.}$$

These angles are shown in Fig. 2–31*b*.

NOTE: In particular, notice that $\beta > 90°$ since the $\mathbf{j}$ component of $\mathbf{u}_{F_R}$ is negative. This seems reasonable considering how $\mathbf{F}_1$ and $\mathbf{F}_2$ add according to the parallelogram law.

EXAMPLE 10

Express the force **F** shown in Fig. 2–32*a* as a Cartesian vector.



(a)

SOLUTION

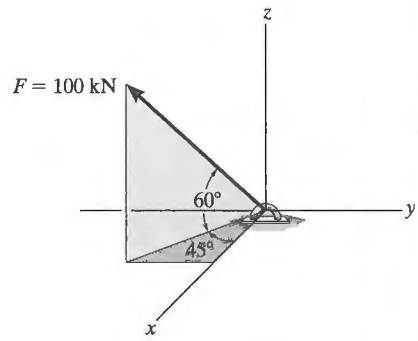
The angles of 60° and 45° defining the direction of **F** are *not* coordinate direction angles. Two successive applications of the parallelogram law are needed to resolve **F** into its *x, y, z* components. First $\mathbf{F} = \mathbf{F}' + \mathbf{F}_z$, then $\mathbf{F}' = \mathbf{F}_x + \mathbf{F}_y$, Fig. 2–32*b*. By trigonometry, the magnitudes of the components are

$$F_z = 100 \sin 60° \text{ kN} = 86.6 \text{ kN}$$

$$F' = 100 \cos 60° \text{ kN} = 50 \text{ kN}$$

$$F_x = F' \cos 45° = 50 \cos 45° \text{ kN} = 35.4 \text{ kN}$$

$$F_y = F' \sin 45° = 50 \sin 45° \text{ kN} = 35.4 \text{ kN}$$



(a)

Realizing that $\mathbf{F}_y$ has a direction defined by $-\mathbf{j}$, we have

$$\mathbf{F} = \{35.4\mathbf{i} - 35.4\mathbf{j} + 86.6\mathbf{k}\} \text{ kN} \qquad \textit{Ans.}$$

To show that the magnitude of this vector is indeed 100 kN, apply Eq. 2–4,

$$F = \sqrt{F_x^2 + F_y^2 + F_z^2}$$

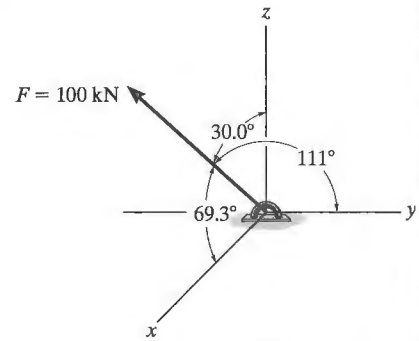
$$= \sqrt{(35.4)^2 + (-35.4)^2 + (86.6)^2} = 100 \text{ kN}$$

If needed, the coordinate direction angles of **F** can be determined from the components of the unit vector acting in the direction of **F**. Hence,

$$\mathbf{u} = \frac{\mathbf{F}}{F} = \frac{F_x}{F}\mathbf{i} + \frac{F_y}{F}\mathbf{j} + \frac{F_z}{F}\mathbf{k}$$

$$= \frac{35.4}{100}\mathbf{i} - \frac{35.4}{100}\mathbf{j} + \frac{86.6}{100}\mathbf{k}$$

$$= 0.354\mathbf{i} - 0.354\mathbf{j} + 0.866\mathbf{k}$$



(c)

Fig. 2–32

so that

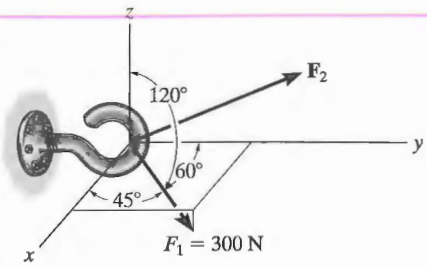
$$\alpha = \cos^{-1}(0.354) = 69.3°$$

$$\beta = \cos^{-1}(-0.354) = 111°$$

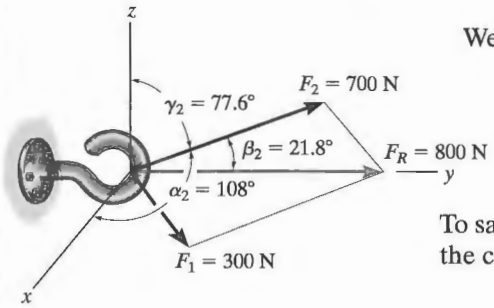
$$\gamma = \cos^{-1}(0.866) = 30.0°$$

These results are shown in Fig. 2–32*c*.

EXAMPLE 2.11



(a)



(b)

Fig. 2–33

Two forces act on the hook shown in Fig. 2–33*a*. Specify the magnitude of $\mathbf{F}_2$ and its coordinate direction angles of $\mathbf{F}_2$ that the resultant force $\mathbf{F}_R$ acts along the positive y axis and has a magnitude of 800 N.

SOLUTION

To solve this problem, the resultant force $\mathbf{F}_R$ and its two components, $\mathbf{F}_1$ and $\mathbf{F}_2$, will each be expressed in Cartesian vector form. Then, as shown in Fig. 2–33*a*, it is necessary that $\mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2$.

Applying Eq. 2–9,

$$\begin{aligned}\mathbf{F}_1 &= F_1 \cos\alpha_1 \mathbf{i} + F_1 \cos\beta_1 \mathbf{j} + F_1 \cos\gamma_1 \mathbf{k}\\ &= 300 \cos 45° \,\mathbf{i} + 300 \cos 60° \,\mathbf{j} + 300 \cos 120° \,\mathbf{k}\\ &= \{212.1\mathbf{i} + 150\mathbf{j} - 150\mathbf{k}\}\text{ N}\\ \mathbf{F}_2 &= F_{2x}\mathbf{i} + F_{2y}\mathbf{j} + F_{2z}\mathbf{k}\end{aligned}$$

Since $\mathbf{F}_R$ has a magnitude of 800 N and acts in the $+\mathbf{j}$ direction,

$$\mathbf{F}_R = (800\text{ N})(+\mathbf{j}) = \{800\mathbf{j}\}\text{ N}$$

We require

$$\begin{aligned}\mathbf{F}_R &= \mathbf{F}_1 + \mathbf{F}_2\\ 800\mathbf{j} &= 212.1\mathbf{i} + 150\mathbf{j} - 150\mathbf{k} + F_{2x}\mathbf{i} + F_{2y}\mathbf{j} + F_{2z}\mathbf{k}\\ 800\mathbf{j} &= (212.1 + F_{2x})\mathbf{i} + (150 + F_{2y})\mathbf{j} + (-150 + F_{2z})\mathbf{k}\end{aligned}$$

To satisfy this equation, the $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components of $\mathbf{F}_R$ must be equal to the corresponding $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components of $(\mathbf{F}_1 + \mathbf{F}_2)$. Hence,

$$\begin{aligned}0 &= 212.1 + F_{2x} & F_{2x} &= -212.1\text{ N}\\ 800 &= 150 + F_{2y} & F_{2y} &= 650\text{ N}\\ 0 &= -150 + F_{2z} & F_{2z} &= 150\text{ N}\end{aligned}$$

The magnitude of $\mathbf{F}_2$ is thus

$$\begin{aligned}F_2 &= \sqrt{(-212.1\text{ N})^2 + (650\text{ N})^2 + (150\text{ N})^2}\\ &= 700\text{ N}\end{aligned}$$ *Ans.*

We can use Eq. 2–9 to determine α_2, β_2, and γ_2.

$$\cos\alpha_2 = \frac{-212.1}{700}; \qquad \alpha_2 = 108°$$ *Ans.*

$$\cos\beta_2 = \frac{650}{700}; \qquad \beta_2 = 21.8°$$ *Ans.*

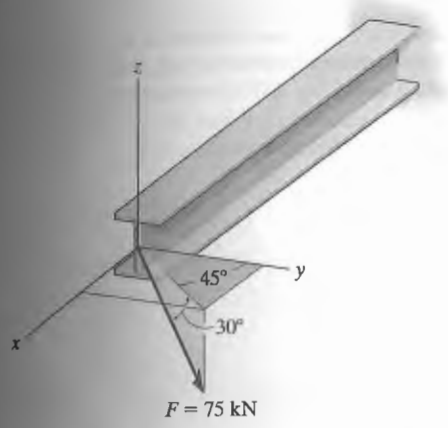
$$\cos\gamma_2 = \frac{150}{700}; \qquad \gamma_2 = 77.6°$$ *Ans.*

These results are shown in Fig. 2–32*b*.

Exclusively for The University of Melbourne

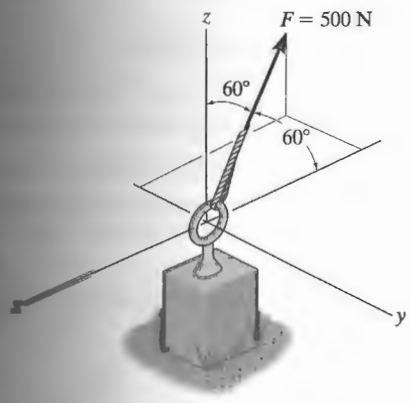
FUNDAMENTAL PROBLEMS

F2–13. Determine the coordinate direction angles of the force.



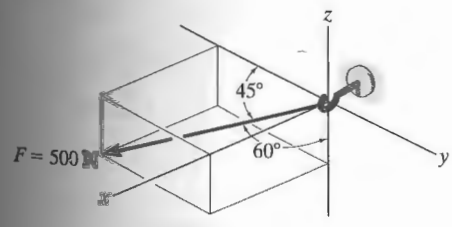
F2–13

F2–14. Express the force as a Cartesian vector.



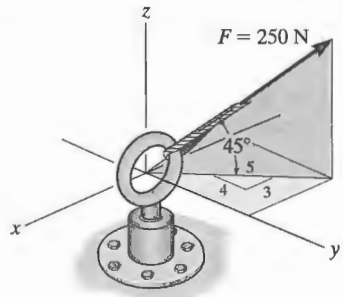
F2–14

F2–15. Express the force as a Cartesian vector.



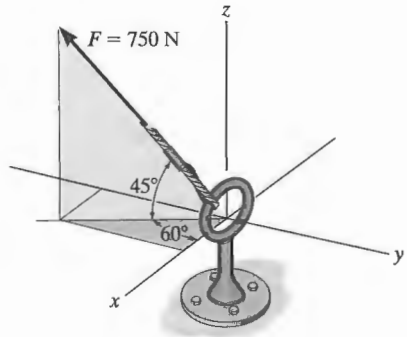
F2–15

F2–16. Express the force as a Cartesian vector.



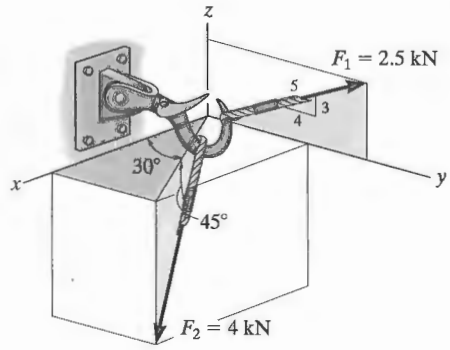
F2–16

F2–17. Express the force as a Cartesian vector.



F2–17

F2–18. Determine the resultant force acting on the hook.

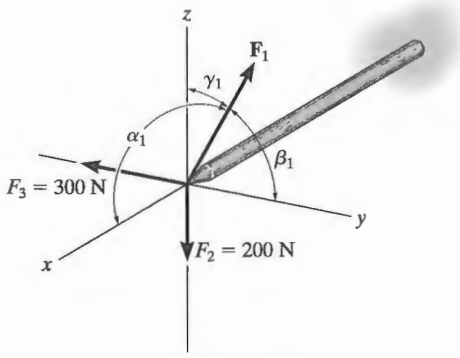


F2–18

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

2–74. The mast is subjected to the three forces shown. Determine the coordinate direction angles $\alpha_1, \beta_1, \gamma_1$ of $\mathbf{F}_1$ so that the resultant force acting on the mast is $\mathbf{F}_R = \{350\mathbf{i}\}$ N.

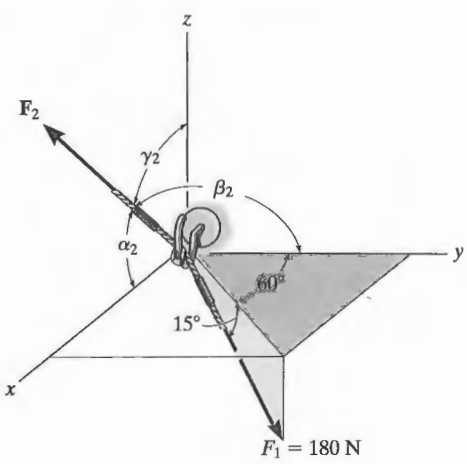
2–75. The mast is subjected to the three forces shown. Determine the coordinate direction angles $\alpha_1, \beta_1, \gamma_1$ of $\mathbf{F}_1$ so that the resultant force acting on the mast is zero.



Probs. 2–74/75

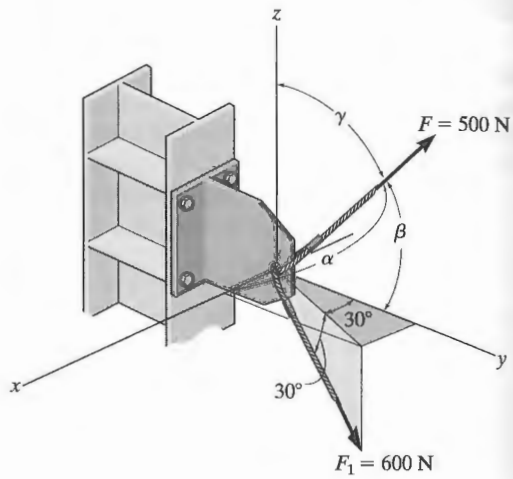
***2–76.** Determine the magnitude and coordinate direction angles of $\mathbf{F}_2$ so that the resultant of the two forces acts along the positive x axis and has a magnitude of 500 N.

•2–77. Determine the magnitude and coordinate direction angles of $\mathbf{F}_2$ so that the resultant of the two forces is zero.



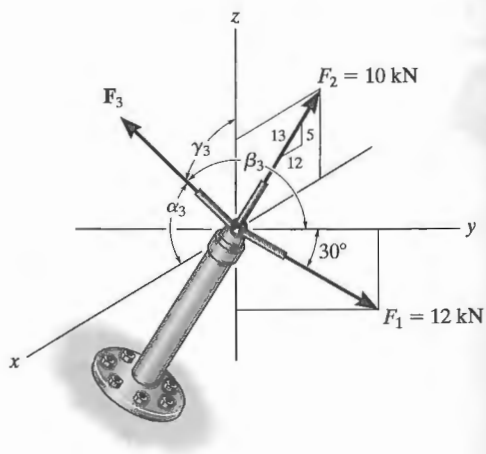
Probs. 2–76/77

2–78. If the resultant force acting on the bracket is directed along the positive y axis, determine the magnitude of the resultant force and the coordinate direction angles of $\mathbf{F}$ so that $\beta < 90°$.



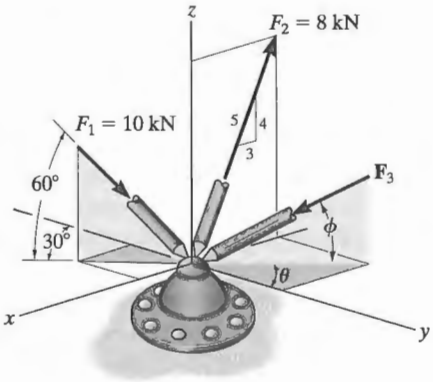
Prob. 2–78

2–79. Specify the magnitude of $\mathbf{F}_3$ and its coordinate direction angles $\alpha_3, \beta_3, \gamma_3$ so that the resultant force $\mathbf{F}_R = \{9\mathbf{j}\}$ kN.



Prob. 2–79

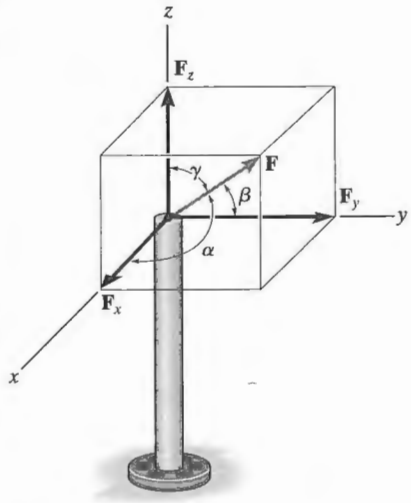
*2–80. If $F_3 = 9$ kN, $\theta = 30°$, and $\phi = 45°$, determine the magnitude and coordinate direction angles of the resultant force acting on the ball-and-socket joint.



Prob. 2–80

•2–81. The pole is subjected to the force **F**, which has components acting along the *x, y, z* axes as shown. If the magnitude of **F** is 3 kN, $\beta = 30°$, and $\gamma = 75°$, determine the magnitudes of its three components.

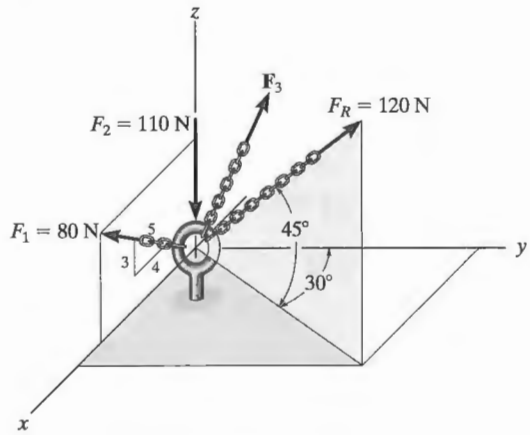
2–82. The pole is subjected to the force **F** which has components $F_x = 1.5$ kN and $F_z = 1.25$ kN. If $\beta = 75°$, determine the magnitudes of **F** and $\mathbf{F}_y$.



Probs. 2–81/82

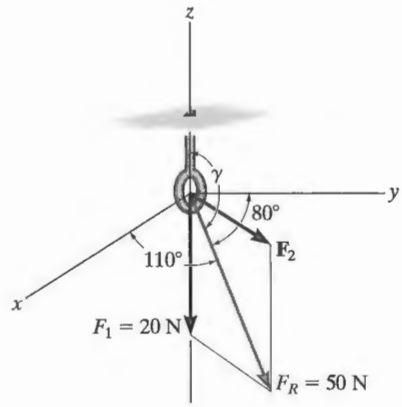
2–83. Three forces act on the ring. If the resultant force $\mathbf{F}_R$ has a magnitude and direction as shown, determine the magnitude and the coordinate direction angles of force $\mathbf{F}_3$.

*2–84. Determine the coordinate direction angles of $\mathbf{F}_1$ and $\mathbf{F}_R$.



Probs. 2–83/84

•2–85. Two forces $\mathbf{F}_1$ and $\mathbf{F}_2$ act on the bolt. If the resultant force $\mathbf{F}_R$ has a magnitude of 50 N and coordinate direction angles $\alpha = 110°$ and $\beta = 80°$, as shown, determine the magnitude of $\mathbf{F}_2$ and its coordinate direction angles.



Prob. 2–85

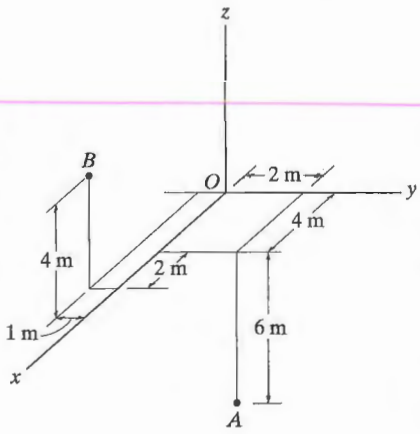


Fig. 2–34

2.7 Position Vectors

In this section we will introduce the concept of a position vector. It will be shown that this vector is of importance in formulating a Cartesian force vector directed between two points in space.

x, y, z Coordinates. Throughout the book we will use a *right-handed* coordinate system to reference the location of points in space. We will also use the convention followed in many technical books which requires the positive z axis to be directed *upward* (the zenith direction) so that it measures the height of an object or the altitude of a point. The x, y axes then lie in the horizontal plane, Fig. 2–34. Points in space are located relative to the origin of coordinates, O, by successive measurements along the x, y, z axes. For example, the coordinates of point A are obtained by starting at O and measuring $x_A = +4$ m along the x axis, then $y_A = +2$ m along the y axis, and finally $z_A = -6$ m along the z axis. Thus, $A(4\text{ m}, 2\text{ m}, -6\text{ m})$. In a similar manner, measurements along the x, y, z axes from O to B yield the coordinates of B, i.e., $B(6\text{ m}, -1\text{ m}, 4\text{ m})$.

Position Vector. A *position vector* $\mathbf{r}$ is defined as a fixed vector which locates a point in space relative to another point. For example, if $\mathbf{r}$ extends from the origin of coordinates, O, to point $P(x, y, z)$, Fig. 2–35*a*, then $\mathbf{r}$ can be expressed in Cartesian vector form as

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

Note how the head-to-tail vector addition of the three components yields vector $\mathbf{r}$, Fig. 2–35*b*. Starting at the origin O, one "travels" x in the $+\mathbf{i}$ direction, then y in the $+\mathbf{j}$ direction, and finally z in the $+\mathbf{k}$ direction to arrive at point $P(x, y, z)$.

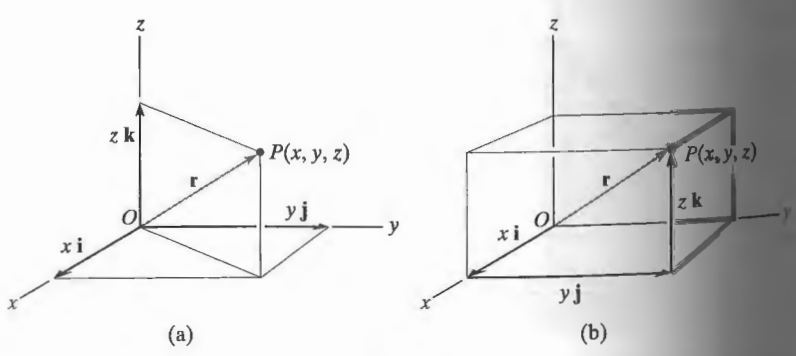


Fig. 2–35

Exclusively for The University of Melbourne

In the more general case, the position vector may be directed from point A to point B in space, Fig. 2–36*a*. This vector is also designated by the symbol **r**. As a matter of convention, we will *sometimes* refer to this vector with *two subscripts* to indicate from and to the point where it is directed. Thus, **r** can also be designated as $\mathbf{r}_{AB}$. Also, note that $\mathbf{r}_A$ and $\mathbf{r}_B$ in Fig. 2–36*a* are referenced with only one subscript since they extend from the origin of coordinates.

From Fig. 2–36*a*, by the head-to-tail vector addition, using the triangle rule, we require

$$\mathbf{r}_A + \mathbf{r} = \mathbf{r}_B$$

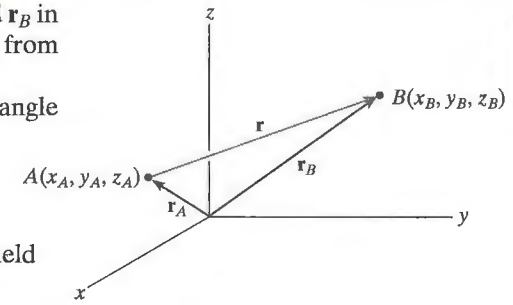
Solving for **r** and expressing $\mathbf{r}_A$ and $\mathbf{r}_B$ in Cartesian vector form yield

$$\mathbf{r} = \mathbf{r}_B - \mathbf{r}_A = (x_B\mathbf{i} + y_B\mathbf{j} + z_B\mathbf{k}) - (x_A\mathbf{i} + y_A\mathbf{j} + z_A\mathbf{k})$$

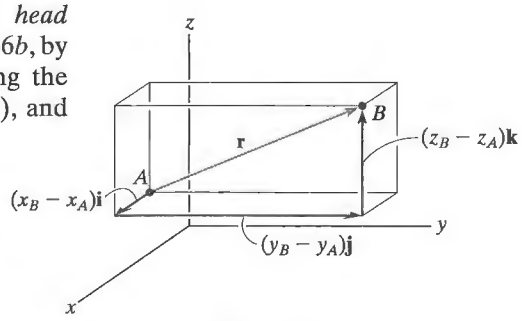
or

$$\boxed{\mathbf{r} = (x_B - x_A)\mathbf{i} + (y_B - y_A)\mathbf{j} + (z_B - z_A)\mathbf{k}} \qquad (2\text{–}11)$$

Thus, the **i**, **j**, **k** *components of the position vector* **r** *may be formed by taking the coordinates of the tail of the vector* $A(x_A, y_A, z_A)$ *and subtracting them from the corresponding coordinates of the head* $B(x_B, y_B, z_B)$. We can also form these components *directly*, Fig. 2–36*b*, by starting at A and moving through a distance of $(x_B - x_A)$ along the positive x axis (+**i**), then $(y_B - y_A)$ along the positive y axis (+**j**), and finally $(z_B - z_A)$ along the positive z axis (+**k**) to get to B.



(a)



(b)

Fig. 2–36

If an x, y, z coordinate system is established, then the coordinates of points A and B can be determined. From this, the position vector **r** acting along the cable can be formulated. Its magnitude represents the length of the cable, and its unit vector, $\mathbf{u} = \mathbf{r}/r$, gives the direction defined by α, β, γ.

EXAMPLE 2.12

An elastic rubber band is attached to points A and B as shown in Fig. 2–37a. Determine its length and its direction measured from A toward B.

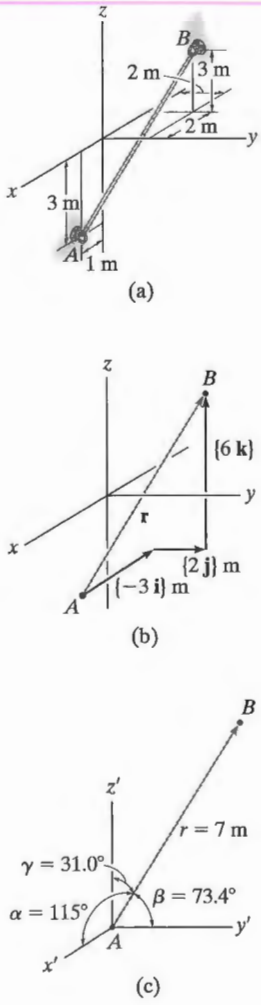


Fig. 2–37

SOLUTION

We first establish a position vector from A to B, Fig. 2–37b. In accordance with Eq. 2–11, the coordinates of the tail $A(1\text{ m}, 0, -3\text{ m})$ are subtracted from the coordinates of the head $B(-2\text{ m}, 2\text{ m}, 3\text{ m})$, which yields

$$\mathbf{r} = [-2\text{ m} - 1\text{ m}]\mathbf{i} + [2\text{ m} - 0]\mathbf{j} + [3\text{ m} - (-3\text{ m})]\mathbf{k}$$
$$= \{-3\mathbf{i} + 2\mathbf{j} + 6\mathbf{k}\}\text{ m}$$

These components of $\mathbf{r}$ can also be determined *directly* by realizing that they represent the direction and distance one must travel along each axis in order to move from A to B, i.e., along the x axis $\{-3\mathbf{i}\}$ m, along the y axis $\{2\mathbf{j}\}$ m, and finally along the z axis $\{6\mathbf{k}\}$ m.

The length of the rubber band is therefore

$$r = \sqrt{(-3\text{ m})^2 + (2\text{ m})^2 + (6\text{ m})^2} = 7\text{ m} \qquad \textit{Ans.}$$

Formulating a unit vector in the direction of $\mathbf{r}$, we have

$$\mathbf{u} = \frac{\mathbf{r}}{r} = -\frac{3}{7}\mathbf{i} + \frac{2}{7}\mathbf{j} + \frac{6}{7}\mathbf{k}$$

The components of this unit vector give the coordinate direction angles

$$\alpha = \cos^{-1}\left(-\frac{3}{7}\right) = 115° \qquad \textit{Ans.}$$

$$\beta = \cos^{-1}\left(\frac{2}{7}\right) = 73.4° \qquad \textit{Ans.}$$

$$\gamma = \cos^{-1}\left(\frac{6}{7}\right) = 31.0° \qquad \textit{Ans.}$$

NOTE: These angles are measured from the *positive axes* of a localized coordinate system placed at the tail of $\mathbf{r}$, as shown in Fig. 2–37c.

2.8 Force Vector Directed Along a Line

Quite often in three-dimensional statics problems, the direction of a force is specified by two points through which its line of action passes. Such a situation is shown in Fig. 2–38, where the force **F** is directed along the cord AB. We can formulate **F** as a Cartesian vector by realizing that it has the *same direction* and *sense* as the position vector **r** directed from point A to point B on the cord. This common direction is specified by the *unit vector* $\mathbf{u} = \mathbf{r}/r$. Hence,

$$\mathbf{F} = F\mathbf{u} = F\left(\frac{\mathbf{r}}{r}\right) = F\left(\frac{(x_B - x_A)\mathbf{i} + (y_B - y_A)\mathbf{j} + (z_B - z_A)\mathbf{k}}{\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}}\right)$$

Although we have represented **F** symbolically in Fig. 2–38, note that it has *units of force*, unlike **r**, which has units of length.

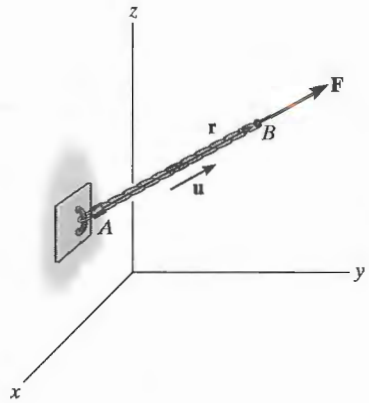
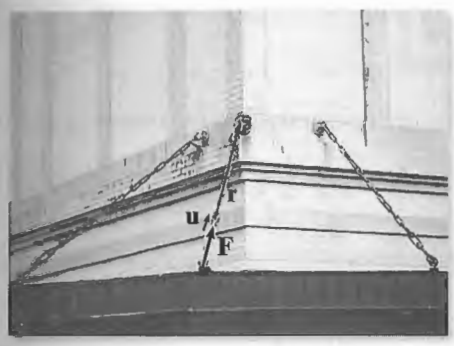


Fig. 2–38



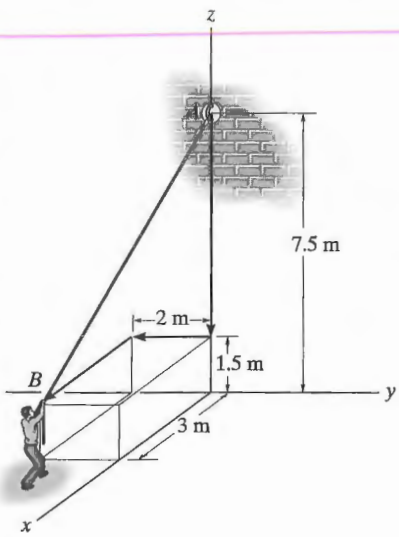
The force **F** acting along the chain can be represented as a Cartesian vector by establishing x, y, z axes and first forming a position vector **r** along the length of the chain. Then, the corresponding unit vector $\mathbf{u} = \mathbf{r}/r$ that defines the direction of both the chain and the force can be determined. Finally, the magnitude of the force is combined with its direction, $\mathbf{F} = F\mathbf{u}$.

Important Points

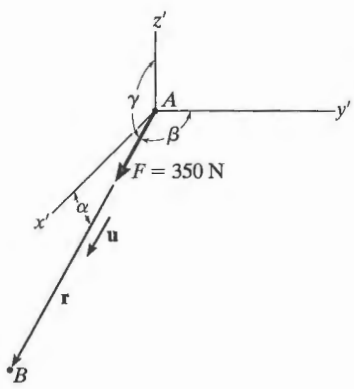
- A position vector locates one point in space relative to another point.
- The easiest way to formulate the components of a position vector is to determine the distance and direction that must be traveled along the x, y, z directions—going from the tail to the head of the vector.
- A force **F** acting in the direction of a position vector **r** can be represented in Cartesian form if the unit vector **u** of the position vector is determined and it is multiplied by the magnitude of the force, i.e., $\mathbf{F} = F\mathbf{u} = F(\mathbf{r}/r)$.

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 2.13



(a)



(b)

Fig. 2–39

The man shown in Fig. 2–39*a* pulls on the cord with a force of 350 N. Represent this force acting on the support A as a Cartesian vector and determine its direction.

SOLUTION

Force **F** is shown in Fig. 2–39*b*. The *direction* of this vector, **u**, is determined from the position vector **r**, which extends from A to B. Rather than using the coordinates of the end points of the cord, **r** can be determined *directly* by noting in Fig. 2–39*a* that one must travel from A {–6**k**} m, then {–2**j**} m, and finally {3**i**} m to get to B. Thus,

$$\mathbf{r} = \{3\mathbf{i} - 2\mathbf{j} - 6\mathbf{k}\}\ \text{m}$$

The magnitude of **r**, which represents the *length* of cord AB, is

$$r = \sqrt{(3\ \text{m})^2 + (-2\ \text{m})^2 + (-6\ \text{m})^2} = 7\ \text{m}$$

Forming the unit vector that defines the direction and sense of both **r** and **F**, we have

$$\mathbf{u} = \frac{\mathbf{r}}{r} = \frac{3}{7}\mathbf{i} - \frac{2}{7}\mathbf{j} - \frac{6}{7}\mathbf{k}$$

Since **F**, has a *magnitude* of 350 N and a *direction* specified by **u**, then

$$\mathbf{F} = F\mathbf{u} = 350\ \text{N}\left(\frac{3}{7}\mathbf{i} - \frac{2}{7}\mathbf{j} - \frac{6}{7}\mathbf{k}\right)$$

$$= \{150\mathbf{i} - 100\mathbf{j} - 300\mathbf{k}\}\ \text{N} \qquad \textit{Ans.}$$

The coordinate direction angles are measured between **r** (or **F**) and the *positive axes* of a localized coordinate system with origin placed at A, Fig. 2–39*b*. From the components of the unit vector:

$$\alpha = \cos^{-1}\left(\frac{3}{7}\right) = 64.6° \qquad \textit{Ans.}$$

$$\beta = \cos^{-1}\left(\frac{-2}{7}\right) = 107° \qquad \textit{Ans.}$$

$$\gamma = \cos^{-1}\left(\frac{-6}{7}\right) = 149° \qquad \textit{Ans.}$$

NOTE: These results make sense when compared with the angles identified in Fig. 2–39*b*.

EXAMPLE 2.14

The force in Fig. 2–40*a* acts on the hook. Express it as a Cartesian vector.

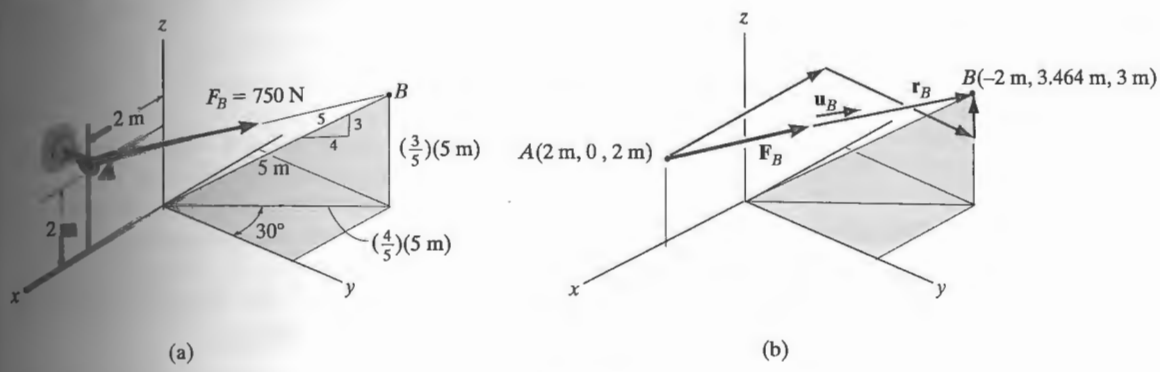


Fig. 2–40

SOLUTION

As shown in Fig. 2–40*b*, the coordinates for points A and B are

$$A(2\text{ m}, 0, 2\text{ m})$$

and

$$B\left[-\left(\frac{4}{5}\right)5\sin 30^\circ\text{ m}, \left(\frac{4}{5}\right)5\cos 30^\circ\text{ m}, \left(\frac{3}{5}\right)5\text{ m}\right]$$

or

$$B(-2\text{ m}, 3.464\text{ m}, 3\text{ m})$$

Therefore, to go from A to B, one must travel $\{4\mathbf{i}\}$ m, then $\{3.464\,\mathbf{j}\}$ m, and finally $\{1\,\mathbf{k}\}$ m. Thus,

$$\mathbf{u}_B = \left(\frac{\mathbf{r}_B}{r_B}\right) = \frac{\{-4\mathbf{i} + 3.464\mathbf{j} + 1\mathbf{k}\}\text{ m}}{\sqrt{(-4\text{ m})^2 + (3.464\text{ m})^2 + (1\text{ m})^2}}$$

$$= -0.7428\mathbf{i} + 0.6433\mathbf{j} + 0.1857\mathbf{k}$$

Force $\mathbf{F}_B$ expressed as a Cartesian vector becomes

$$\mathbf{F}_B = F_B\mathbf{u}_B = (750\text{ N})(-0.7428\mathbf{i} + 0.6433\mathbf{j} + 0.1857\mathbf{k})$$

$$= \{-557\mathbf{i} + 482\mathbf{j} + 139\mathbf{k}\}\text{ N} \qquad \textit{Ans.}$$

EXAMPLE 2.15

The roof is supported by cables as shown in the photo. If the cables exert forces $F_{AB} = 100\text{ N}$ and $F_{AC} = 120\text{ N}$ on the wall hook at A as shown in Fig. 2–41a, determine the resultant force acting at A. Express the result as a Cartesian vector.

SOLUTION

The resultant force $\mathbf{F}_R$ is shown graphically in Fig. 2–41b. We can express this force as a Cartesian vector by first formulating $\mathbf{F}_{AB}$ and $\mathbf{F}_{AC}$ as Cartesian vectors and then adding their components. The directions of $\mathbf{F}_{AB}$ and $\mathbf{F}_{AC}$ are specified by forming unit vectors $\mathbf{u}_{AB}$ and $\mathbf{u}_{AC}$ along the cables. These unit vectors are obtained from the associated position vectors $\mathbf{r}_{AB}$ and $\mathbf{r}_{AC}$. With reference to Fig. 2–41a, to go from A to B, we must travel $\{-4\mathbf{k}\}$ m and, then $\{-4\mathbf{i}\}$ m . Thus,

$$\mathbf{r}_{AB} = \{4\mathbf{i} - 4\mathbf{k}\}\text{ m}$$

$$r_{AB} = \sqrt{(4\text{ m})^2 + (-4\text{ m})^2} = 5.66\text{ m}$$

$$\mathbf{F}_{AB} = \mathbf{F}_{AB}\left(\frac{\mathbf{r}_{AB}}{r_{AB}}\right) = (100\text{ N})\left(\frac{4}{5.66}\mathbf{i} - \frac{4}{5.66}\mathbf{k}\right)$$

$$\mathbf{F}_{AB} = \{70.7\mathbf{i} - 70.7\mathbf{k}\}\text{ N}$$

To go from A to C, we must travel $\{-4\mathbf{k}\}$ m , then $\{2\mathbf{j}\}$ m, and finally $\{4\mathbf{i}\}$. Thus,

$$\mathbf{r}_{AC} = \{4\mathbf{i} + 2\mathbf{j} - 4\mathbf{k}\}\text{ m}$$

$$r_{AC} = \sqrt{(4\text{ m})^2 + (2\text{ m})^2 + (-4\text{ m})^2} = 6\text{ m}$$

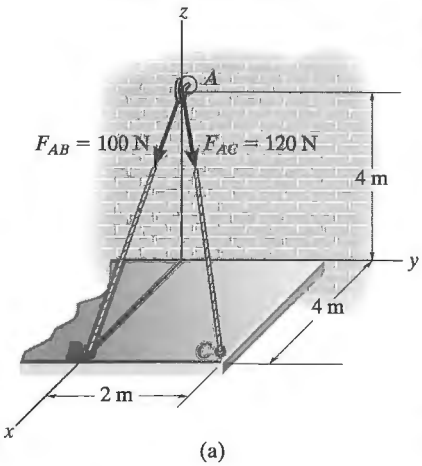
$$\mathbf{F}_{AC} = \mathbf{F}_{AC}\left(\frac{\mathbf{r}_{AC}}{r_{AC}}\right) = (120\text{ N})\left(\frac{4}{6}\mathbf{i} + \frac{2}{6}\mathbf{j} - \frac{4}{6}\mathbf{k}\right)$$

$$= \{80\mathbf{i} + 40\mathbf{j} - 80\mathbf{k}\}\text{ N}$$

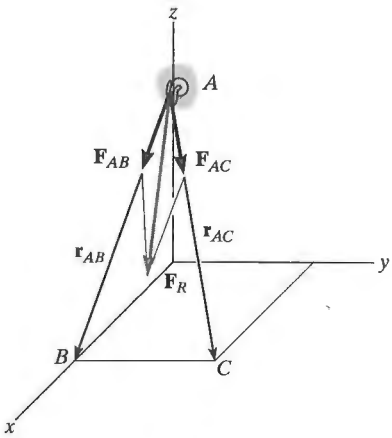
The resultant force is therefore

$$\mathbf{F}_R = \mathbf{F}_{AB} + \mathbf{F}_{AC} = \{70.7\mathbf{i} - 70.7\mathbf{k}\}\text{ N} + \{80\mathbf{i} + 40\mathbf{j} - 80\mathbf{k}\}\text{ N}$$

$$= \{151\mathbf{i} + 40\mathbf{j} - 151\mathbf{k}\}\text{ N} \qquad \textit{Ans.}$$



(a)

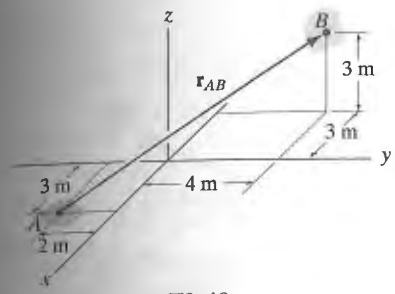


(b)

Fig. 2–41

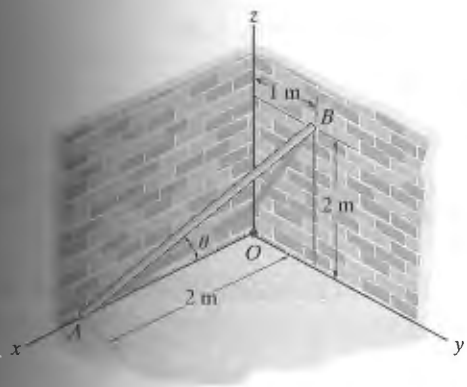
FUNDAMENTAL PROBLEMS

F2–19. Express the position vector $\mathbf{r}_{AB}$ in Cartesian vector form, then determine its magnitude and coordinate direction angles.



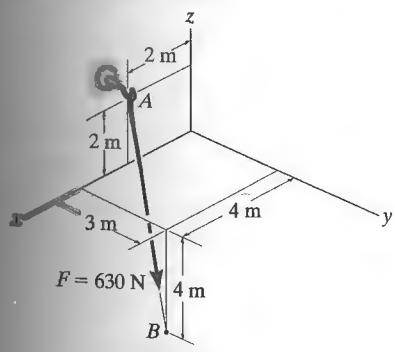
F2–19

F2–20. Determine the length of the rod and the position vector directed from A to B. What is the angle θ?



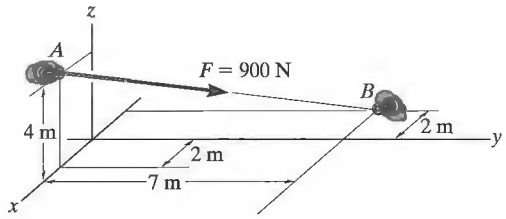
F2–20

F2–21. Express the force as a Cartesian vector.



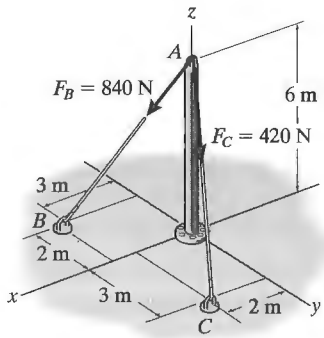
F2–21

F2–22. Express the force as a Cartesian vector.



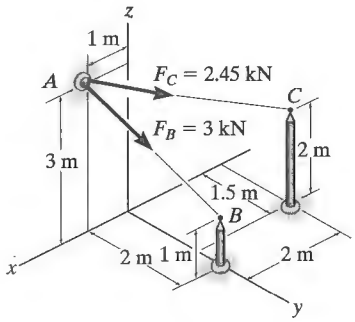
F2–22

F2–23. Determine the magnitude of the resultant force at A.



F2–23

F2–24. Determine the resultant force at A.



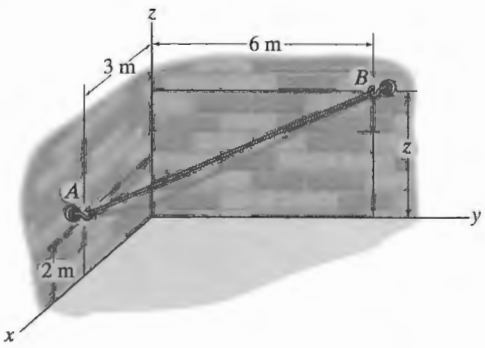
F2–24

MasteringENGINEERING
Customized coaching is available

PROBLEMS

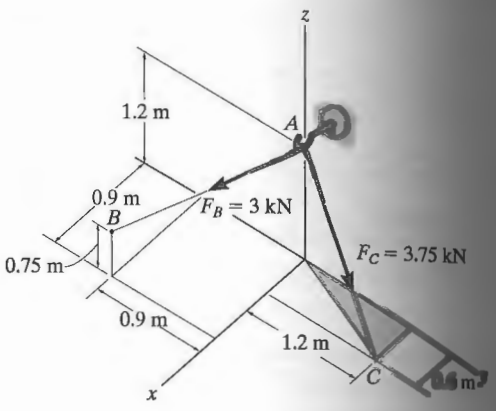
2–86. Determine the position vector **r** directed from point A to point B and the length of cord AB. Take $z = 4$ m.

2–87. If the cord AB is 7.5 m long, determine the coordinate position $+z$ of point B.



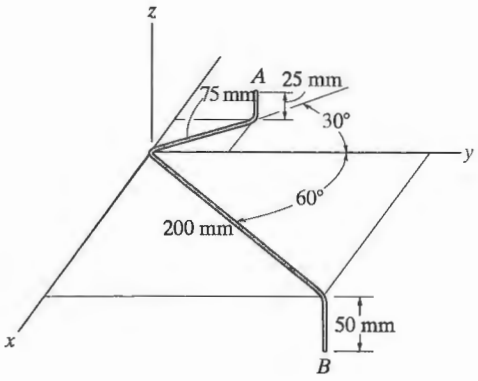
Probs. 2–86/87

•2–89. Determine the magnitude and coordinate direction angles of the resultant force acting at A.



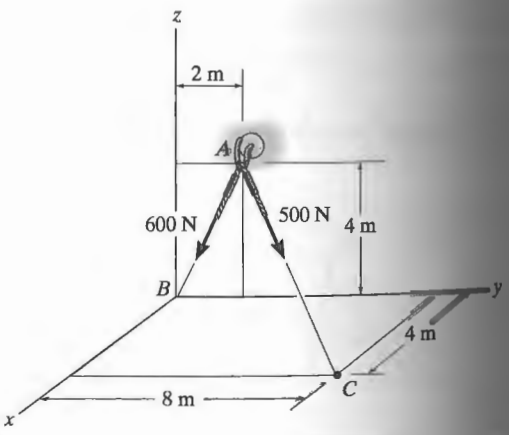
Prob. 2–89

***2–88.** Determine the distance between the end points A and B on the wire by first formulating a position vector from A to B and then determining its magnitude.



Prob. 2–88

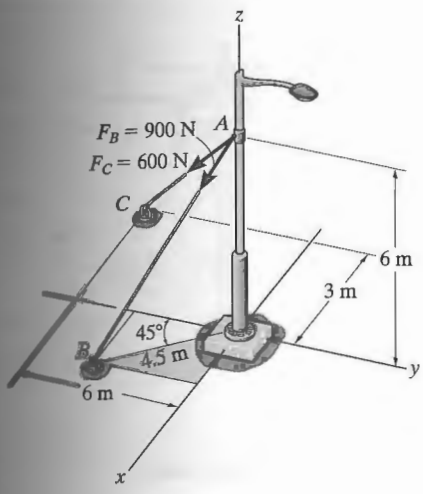
2–90. Determine the magnitude and coordinate direction angles of the resultant force.



Prob. 2–90

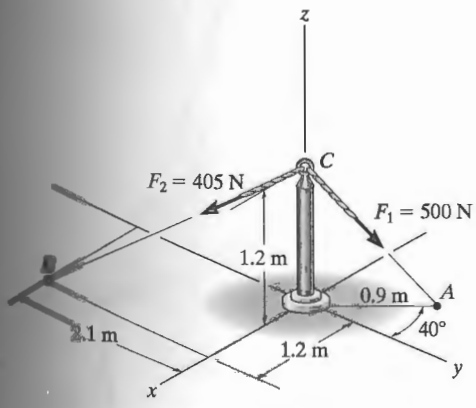
Exclusively for The University of Melbourne

2–91. Determine the magnitude and coordinate direction angles of the resultant force acting at A.



Prob. 2–91

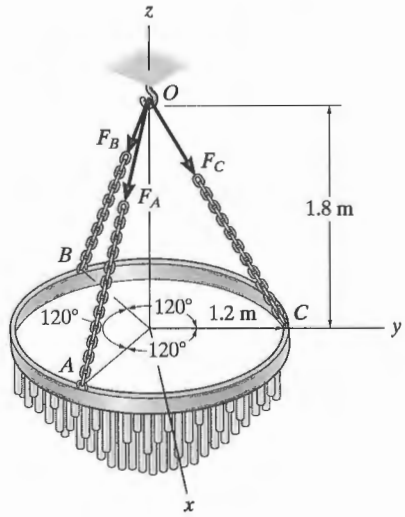
2–92. Determine the magnitude and coordinate direction angles of the resultant force.



Prob. 2–92

•2–93. The chandelier is supported by three chains which are concurrent at point O. If the force in each chain has a magnitude of 300 N, express each force as a Cartesian vector and determine the magnitude and coordinate direction angles of the resultant force.

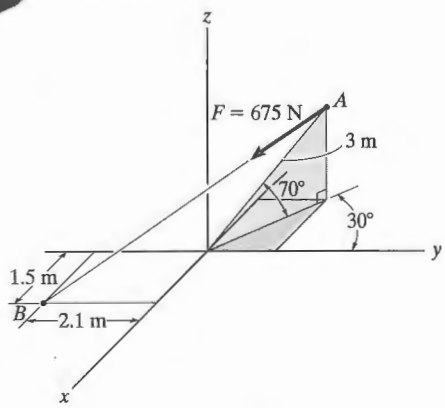
2–94. The chandelier is supported by three chains which are concurrent at point O. If the resultant force at O has a magnitude of 650 N and is directed along the negative z axis, determine the force in each chain.



Probs. 2–93/94

2–95. Express force **F** as a Cartesian vector; then determine its coordinate direction angles.

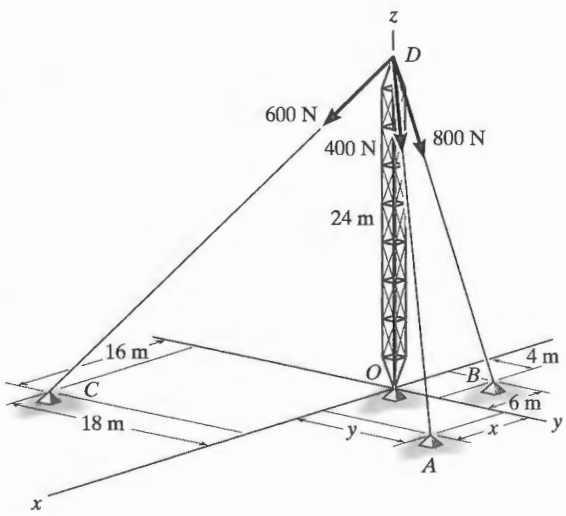
Video Solutions



Prob. 2–95

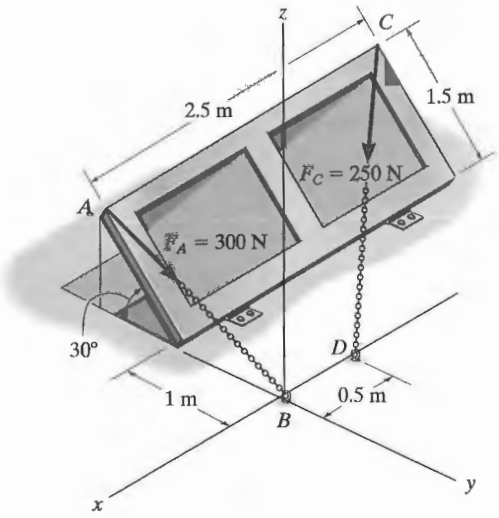
Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

***2–96.** The tower is held in place by three cables. If the force of each cable acting on the tower is shown, determine the magnitude and coordinate direction angles α, β, γ of the resultant force. Take $x = 20$ m, $y = 15$ m.



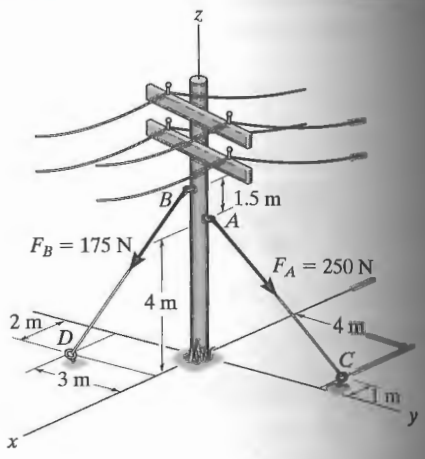
Prob. 2–96

•2–97. The door is held opened by means of two chains. If the tension in AB and CD is $F_A = 300$ N and $F_C = 250$ N, respectively, express each of these forces in Cartesian vector form.



Prob. 2–97

2–98. The guy wires are used to support the telephone pole. Represent the force in each wire in Cartesian vector form. Neglect the diameter of the pole.

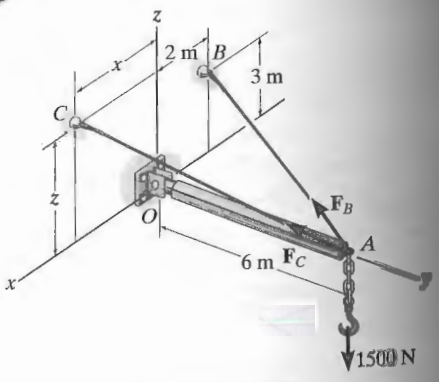


Prob. 2–98

2–99. Two cables are used to secure the overhang boom in position and support the 1500-N load. If the resultant force is directed along the boom from point A towards O, determine the magnitudes of the resultant force and forces $\mathbf{F}_B$ and $\mathbf{F}_C$. Set $x = 3$ m and $z = 2$ m.

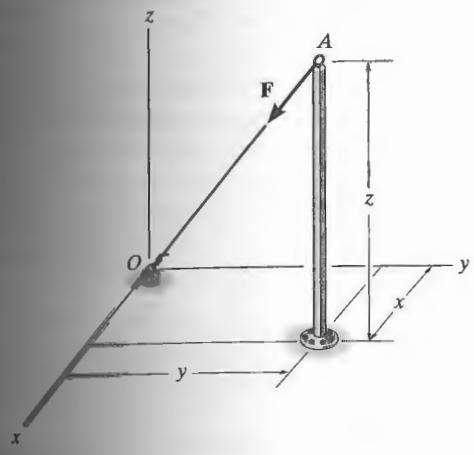
***2–100.** Two cables are used to secure the overhang boom in position and support the 1500-N load. If the resultant force is directed along the boom from point A towards O, determine the values of x and z for the coordinates of point C and the magnitude of the resultant force. Set $F_B = 1610$ N and $F_C = 2400$ N.

Video Solutions



Probs. 2–99/100

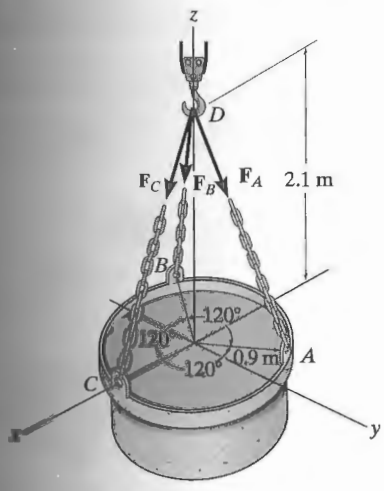
•2–101. The cable AO exerts a force on the top of the pole of $\mathbf{F} = \{-120\mathbf{i} - 90\mathbf{j} - 80\mathbf{k}\}$ N. If the cable has a length of 1.02 m, determine the height z of the pole and the location (x, y) of its base.



Prob. 2–101

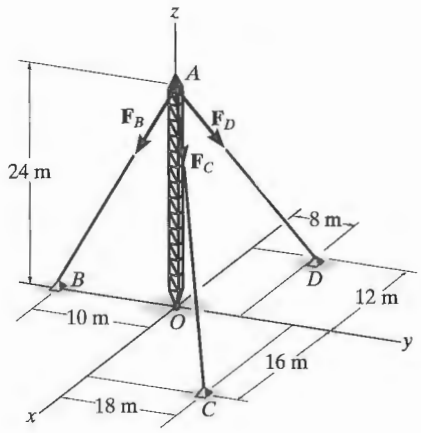
2–102. If the force in each chain has a magnitude of 2.25 kN, determine the magnitude and coordinate direction angles of the resultant force.

2–103. If the resultant of the three forces is $\mathbf{F}_R = \{-4.5\mathbf{k}\}$ kN, determine the magnitude of the force in each chain.



Probs. 2–102/103

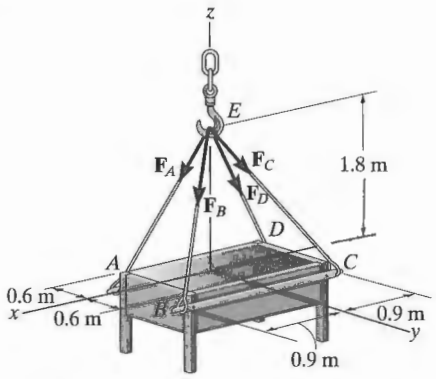
*2–104. The antenna tower is supported by three cables. If the forces of these cables acting on the antenna are $F_B = 520$ N, $F_C = 680$ N, and $F_D = 560$ N, determine the magnitude and coordinate direction angles of the resultant force acting at A.



Prob. 2–104

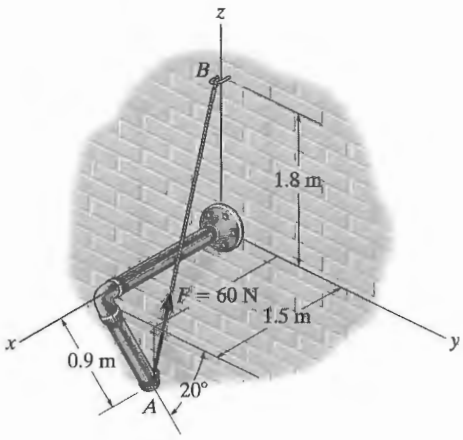
•2–105. If the force in each cable tied to the bin is 350 N, determine the magnitude and coordinate direction angles of the resultant force.

2–106. If the resultant of the four forces is $\mathbf{F}_R = \{-1.8\mathbf{k}\}$ kN, determine the tension developed in each cable. Due to symmetry, the tension in the four cables is the same.



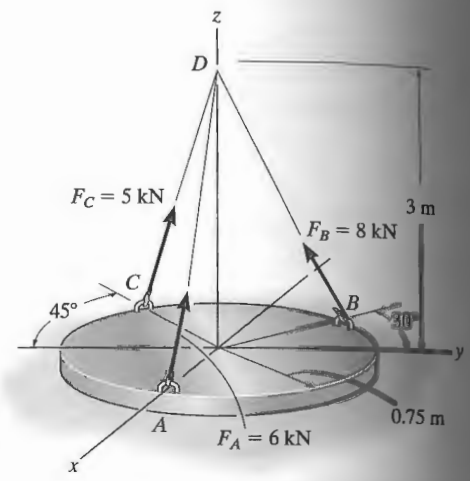
Probs. 2–105/106

2–107. The pipe is supported at its end by a cord AB. If the cord exerts a force of $F = 60$ N on the pipe at A, express this force as a Cartesian vector.



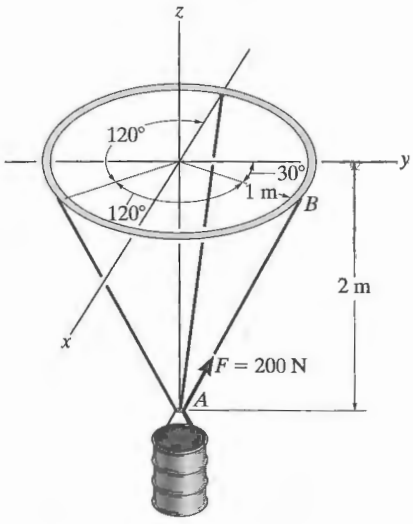
Prob. 2–107

•2–109. The cylindrical plate is subjected to the three cable forces which are concurrent at point D. Express each force which the cables exert on the plate as a Cartesian vector and determine the magnitude and coordinate direction angles of the resultant force.



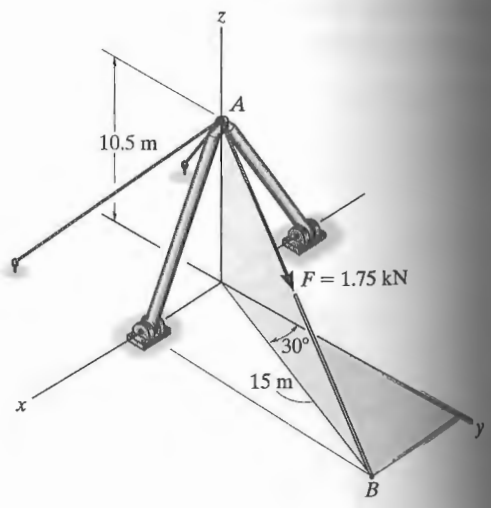
Prob. 2–109

***2–108.** The load at A creates a force of 200 N in wire AB. Express this force as a Cartesian vector, acting on A and directed towards B.



Prob. 2–108

2–110. The cable attached to the shear-leg derrick exerts a force on the derrick of $F = 1.75$ kN. Express this force as a Cartesian vector.



Prob. 2–110

Exclusively for The University of Melbourne

2.9 Dot Product

Occasionally in statics one has to find the angle between two lines or the components of a force parallel and perpendicular to a line. In two dimensions, these problems can readily be solved by trigonometry since the geometry is easy to visualize. In three dimensions, however, this is often difficult, and consequently vector methods should be employed for the solution. The dot product, which defines a particular method for "multiplying" two vectors, will be is used to solve the above-mentioned problems.

The *dot product* of vectors **A** and **B**, written $\mathbf{A} \cdot \mathbf{B}$, and read "**A** dot **B**" is defined as the product of the magnitudes of **A** and **B** and the cosine of the angle θ between their tails, Fig. 2–42. Expressed in equation form,

$$\boxed{\mathbf{A} \cdot \mathbf{B} = AB \cos \theta} \qquad (2\text{–}12)$$

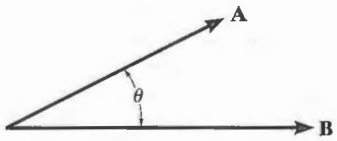


Fig. 2–42

where $0° \leq \theta \leq 180°$. The dot product is often referred to as the *scalar product* of vectors since the result is a *scalar* and not a vector.

Laws of Operation.

1. Commutative law: $\mathbf{A} \cdot \mathbf{B} = \mathbf{B} \cdot \mathbf{A}$
2. Multiplication by a scalar: $a(\mathbf{A} \cdot \mathbf{B}) = (a\mathbf{A}) \cdot \mathbf{B} = \mathbf{A} \cdot (a\mathbf{B})$
3. Distributive law: $\mathbf{A} \cdot (\mathbf{B} + \mathbf{D}) = (\mathbf{A} \cdot \mathbf{B}) + (\mathbf{A} \cdot \mathbf{D})$

It is easy to prove the first and second laws by using Eq. 2–12. The proof of the distributive law is left as an exercise (see Prob. 2–111).

Cartesian Vector Formulation. Equation 2–12 must be used to find the dot product for any two Cartesian unit vectors. For example, $\mathbf{i} \cdot \mathbf{i} = (1)(1) \cos 0° = 1$ and $\mathbf{i} \cdot \mathbf{j} = (1)(1) \cos 90° = 0$. If we want to find the dot product of two general vectors **A** and **B** that are expressed in Cartesian vector form, then we have

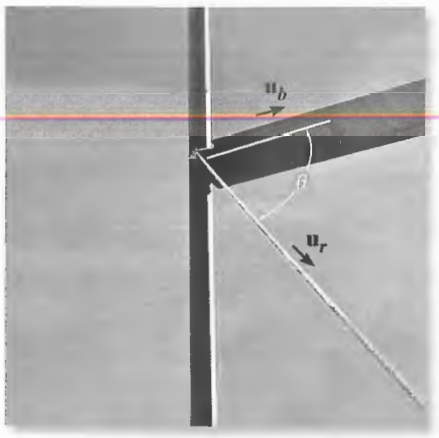
$$\begin{aligned}
\mathbf{A} \cdot \mathbf{B} &= (A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}) \cdot (B_x\mathbf{i} + B_y\mathbf{j} + B_z\mathbf{k}) \\
&= A_xB_x(\mathbf{i} \cdot \mathbf{i}) + A_xB_y(\mathbf{i} \cdot \mathbf{j}) + A_xB_z(\mathbf{i} \cdot \mathbf{k}) \\
&+ A_yB_x(\mathbf{j} \cdot \mathbf{i}) + (A_yB_y(\mathbf{j} \cdot \mathbf{j}) + A_yB_z(\mathbf{j} \cdot \mathbf{k}) \\
&+ A_zB_x(\mathbf{k} \cdot \mathbf{i}) + A_zB_y(\mathbf{k} \cdot \mathbf{j}) + A_zB_z(\mathbf{k} \cdot \mathbf{k})
\end{aligned}$$

Carrying out the dot-product operations, the final result becomes

$$\boxed{\mathbf{A} \cdot \mathbf{B} = A_xB_x + A_yB_y + A_zB_z} \qquad (2\text{–}13)$$

Thus, to determine the dot product of two Cartesian vectors, multiply their corresponding x, y, z components and sum these products algebraically. Note that the result will be either a positive or negative *scalar*.

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.



The angle θ between the rope and the beam can be determined by formulating unit vectors along the beam and rope and then using the dot product $\mathbf{u}_b \cdot \mathbf{u}_r = (1)(1)\cos\theta$.

Applications. The dot product has two important applications in mechanics.

- ***The angle formed between two vectors or intersecting lines.*** The angle θ between the tails of vectors **A** and **B** in Fig. 2–42 can be determined from Eq. 2–12 and written as

$$\theta = \cos^{-1}\left(\frac{\mathbf{A}\cdot\mathbf{B}}{AB}\right) \qquad 0° \le \theta \le 180°$$

Here $\mathbf{A}\cdot\mathbf{B}$ is found from Eq. 2–13. In particular, notice that if $\mathbf{A}\cdot\mathbf{B} = 0, \theta = \cos^{-1} 0 = 90°$ so that **A** will be *perpendicular* to **B**.

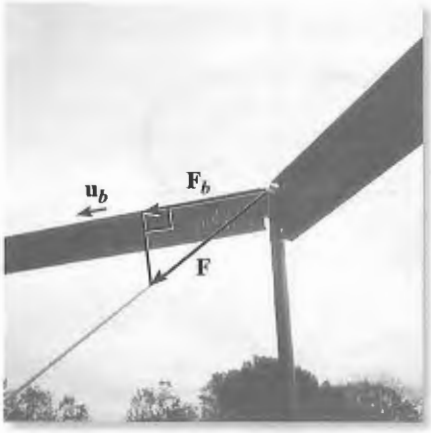
- ***The components of a vector parallel and perpendicular to a line.*** The component of vector **A** parallel to or collinear with the line aa' in Fig. 2–43 is defined by A_a where $A_a = A\cos\theta$. This component is sometimes referred to as the *projection* of **A** onto the line, since a *right angle* is formed in the construction. If the *direction* of the line is specified by the unit vector $\mathbf{u}_a$, then since $u_a = 1$, we can determine the magnitude of $\mathbf{A}_a$ directly from the dot product (Eq. 2–12), i.e.,

$$A_a = A\cos\theta = \mathbf{A}\cdot\mathbf{u}_a$$

Hence, the scalar projection of **A** *along a line is determined from the dot product of* **A** *and the unit vector* $\mathbf{u}_a$ *which defines the direction of the line.* Notice that if this result is positive, then $\mathbf{A}_a$ has a directional sense which is the same as $\mathbf{u}_a$, whereas if A_a is a negative scalar, then $\mathbf{A}_a$ has the opposite sense of direction to $\mathbf{u}_a$.

The component $\mathbf{A}_a$ represented as a *vector* is therefore

$$\mathbf{A}_a = A_a\,\mathbf{u}_a$$



The projection of the cable force **F** along the beam can be determined by first finding the unit vector $\mathbf{u}_b$ that defines this direction. Then apply the dot product, $F_b = \mathbf{F}\cdot\mathbf{u}_b$.

The component of **A** that is perpendicular to line aa can also be obtained, Fig. 2–43. Since $\mathbf{A} = \mathbf{A}_a + \mathbf{A}_\perp$, then $\mathbf{A}_\perp = \mathbf{A} - \mathbf{A}_a$. There are two possible ways of obtaining $A_\perp$. One way would be to determine θ from the dot product, $\theta = \cos^{-1}(\mathbf{A}\cdot\mathbf{u}_A/A)$, then $A_\perp = A\sin\theta$. Alternatively, if A_a is known, then by Pythagorean's theorem we can also write $A_\perp = \sqrt{A^2 - A_a{}^2}$.

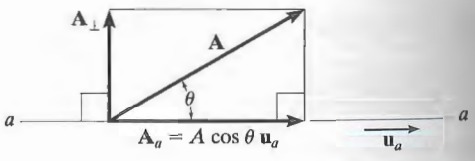


Fig. 2–43

Important Points

- The dot product is used to determine the angle between two vectors or the projection of a vector in a specified direction.
- If vectors **A** and **B** are expressed in Cartesian vector form, the dot product is determined by multiplying the respective x, y, z scalar components and algebraically adding the results, i.e., $\mathbf{A} \cdot \mathbf{B} = A_x B_x + A_y B_y + A_z B_z$.
- From the definition of the dot product, the angle formed between the tails of vectors **A** and **B** is $\theta = \cos^{-1}(\mathbf{A} \cdot \mathbf{B}/AB)$.
- The magnitude of the projection of vector **A** along a line aa whose direction is specified by $\mathbf{u}_a$ is determined from the dot product $A_a = \mathbf{A} \cdot \mathbf{u}_a$.

EXAMPLE 2.16

Determine the magnitudes of the projection of the force **F** in Fig. 2–44 onto the u and v axes.

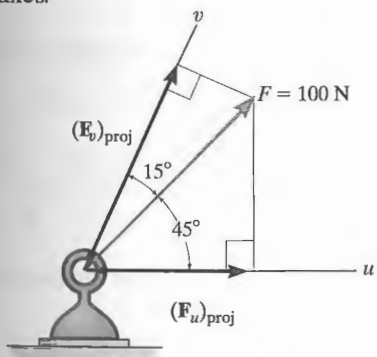


Fig. 2–44

SOLUTION

Projections of Force. The graphical representation of the *projections* is shown in Fig. 2–44. From this figure, the magnitudes of the projections of **F** onto the u and v axes can be obtained by trigonometry:

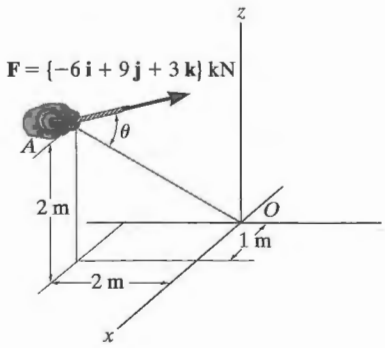
$$(F_u)_{\text{proj}} = (100\text{ N})\cos 45° = 70.7\text{ N} \qquad \textit{Ans.}$$

$$(F_v)_{\text{proj}} = (100\text{ N})\cos 15° = 96.6\text{ N} \qquad \textit{Ans.}$$

NOTE: These projections are not equal to the magnitudes of the components of force **F** along the u and v axes found from the parallelogram law. They will only be equal if the u and v axes are *perpendicular* to one another.

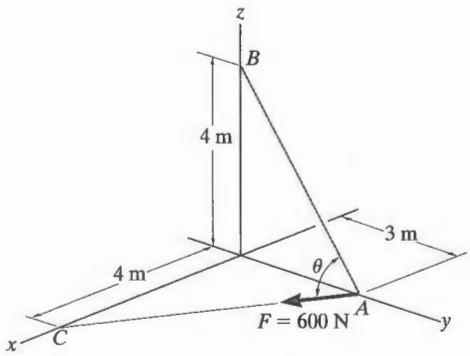
FUNDAMENTAL PROBLEMS

F2–25. Determine the angle θ between the force and the line AO.



F2–25

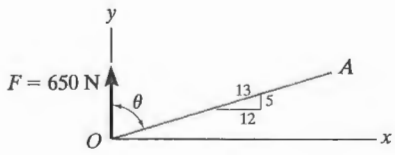
F2–26. Determine the angle θ between the force and the line AB.



F2–26

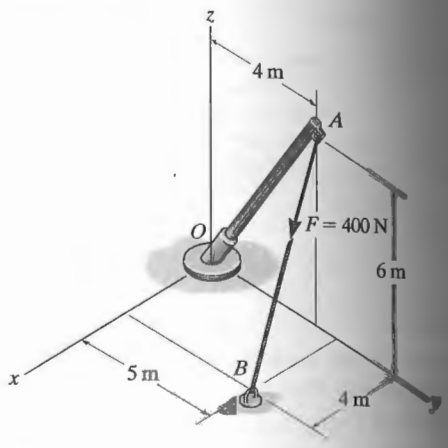
F2–27. Determine the angle θ between the force and the line OA.

F2–28. Determine the component of projection of the force along the line OA.



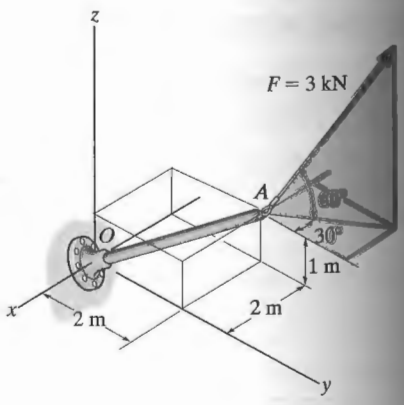
F2–27/28

F2–29. Find the magnitude of the projected component of the force along the pipe.



F2–29

F2–30. Determine the components of the force acting parallel and perpendicular to the axis of the pole.



F2–30

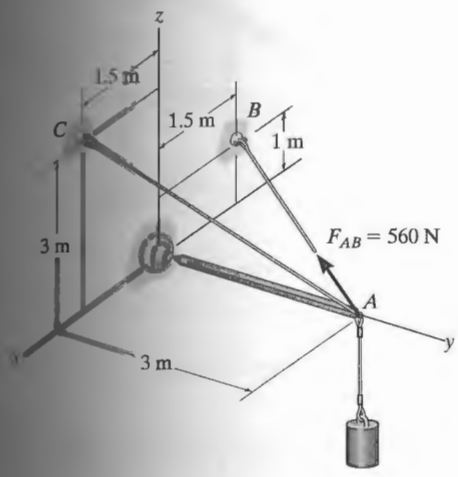
Pearson Custom Publishing exclusively for The University of Melbourne

ENGINEERING
Customized coaching is available!

PROBLEMS

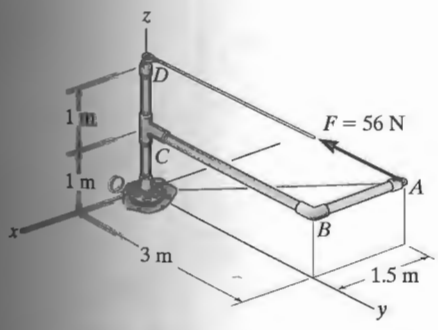
2–111. Given the three vectors A, B, and D, show that $\mathbf{A}\cdot(\mathbf{B}+\mathbf{D}) = (\mathbf{A}\cdot\mathbf{B}) + (\mathbf{A}\cdot\mathbf{D})$.

•2–112. Determine the projected component of the force $F_{AB} = 560$ N acting along cable AC. Express the result as a Cartesian vector.



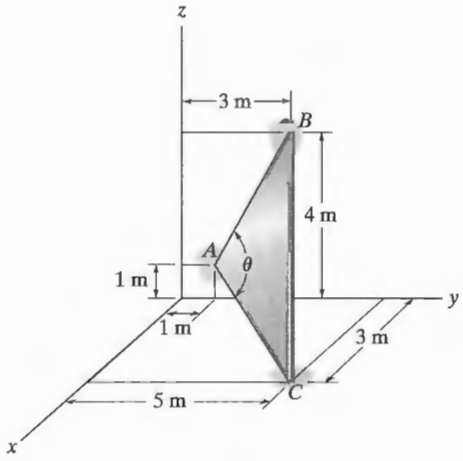
Prob. 2–112

•2–113. Determine the magnitudes of the components of force $F = 56$ N acting along and perpendicular to line AO.



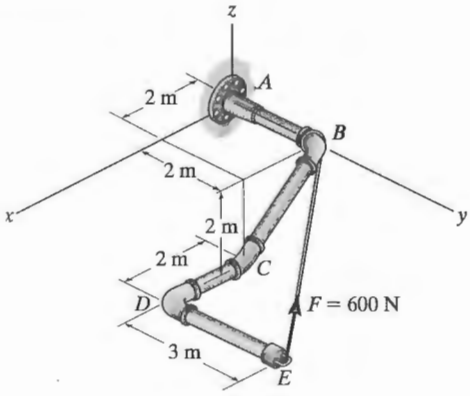
Prob. 2–113

2–114. Determine the length of side BC of the triangular plate. Solve the problem by finding the magnitude of $\mathbf{r}_{BC}$; then check the result by first finding θ, r_{AB}, and r_{AC} and then using the cosine law.



Prob. 2–114

2–115. Determine the magnitudes of the components of $F = 600$ N acting along and perpendicular to segment DE of the pipe assembly.

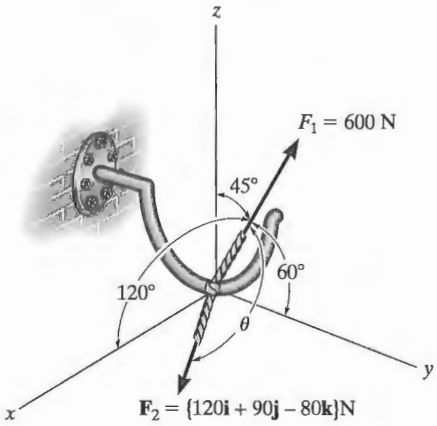


Prob. 2–115

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

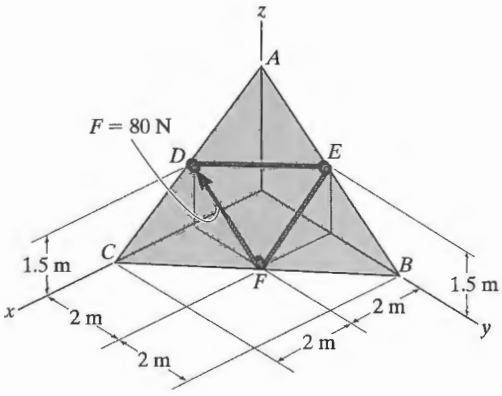
***2–116.** Two forces act on the hook. Determine the angle θ between them. Also, what are the projections of $\mathbf{F}_1$ and $\mathbf{F}_2$ along the y axis?

•2–117. Two forces act on the hook. Determine the magnitude of the projection of $\mathbf{F}_2$ along $\mathbf{F}_1$.



Probs. 2–116/117

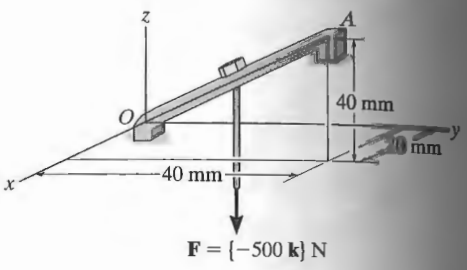
2–118. Determine the projection of force $F = 80$ N along line BC. Express the result as a Cartesian vector.



Prob. 2–118

2–119. The clamp is used on a jig. If the vertical force acting on the bolt is $\mathbf{F} = \{-500\mathbf{k}\}$ N, determine the magnitudes of its components $\mathbf{F}_1$ and $\mathbf{F}_2$ which act along the OA axis and perpendicular to it.

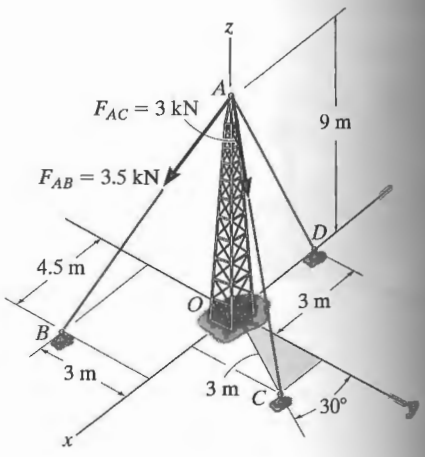
Video Solutions



Prob. 2–119

***2–120.** Determine the magnitude of the projected component of force $\mathbf{F}_{AB}$ acting along the z axis.

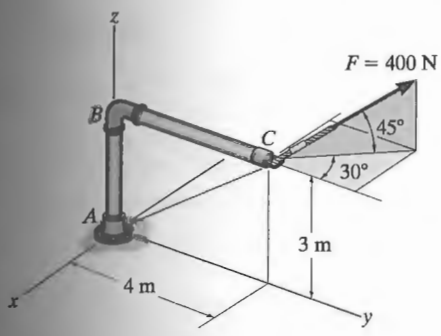
•2–121. Determine the magnitude of the projected component of force $\mathbf{F}_{AC}$ acting along the z axis.



Probs. 2–120/121

2–122. Determine the projection of force $F = 400$ N acting along line AC of the pipe assembly. Express the result as a Cartesian vector.

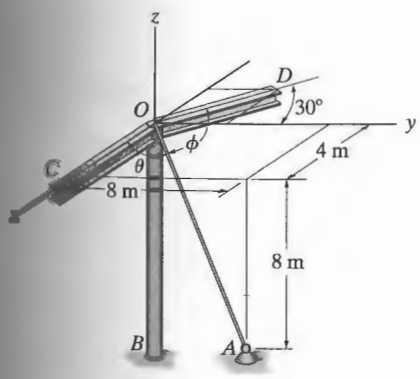
2–123. Determine the magnitudes of the components of force $F = 400$ N acting parallel and perpendicular to segment BC of the pipe assembly.



Probs. 2–122/123

***2–124.** Cable OA is used to support column OB. Determine the angle θ it makes with beam OC.

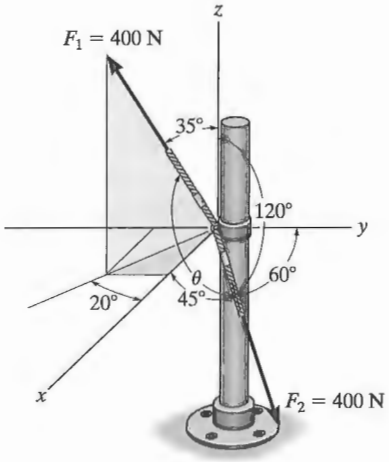
***2–125.** Cable OA is used to support column OB. Determine the angle ϕ it makes with beam OD.



Probs. 2–124/125

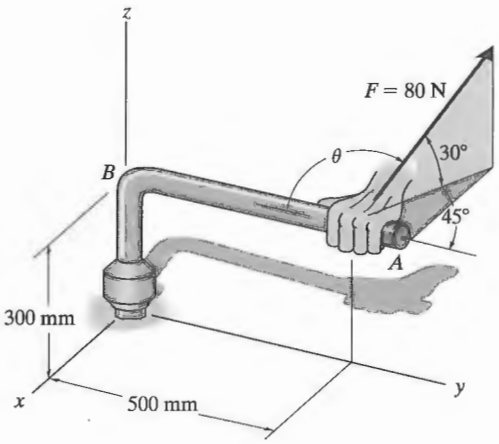
2–126. The cables each exert a force of 400 N on the post. Determine the magnitude of the projected component of $\mathbf{F}_1$ along the line of action of $\mathbf{F}_2$.

2–127. Determine the angle θ between the two cables attached to the post.



Probs. 2–126/127

***2–128.** A force of $F = 80$ N is applied to the handle of the wrench. Determine the angle θ between the tail of the force and the handle AB.

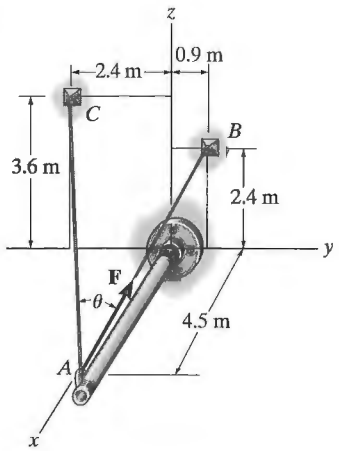


Prob. 2–128

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

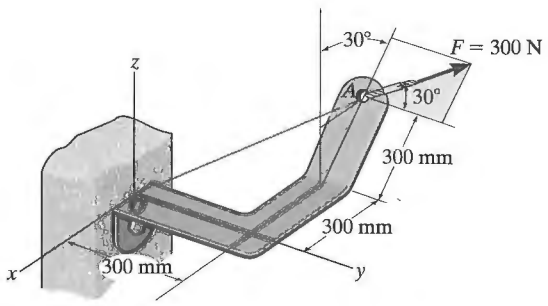
•**2–129.** Determine the angle θ between cables AB and AC.

2–130. If **F** has a magnitude of 250 N, determine the magnitude of its projected components acting along the x axis and along cable AC.



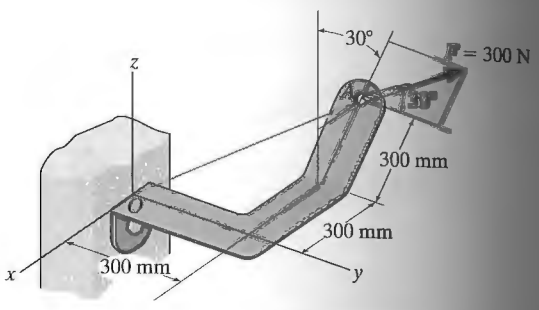
Probs. 2–129/130

2–131. Determine the magnitudes of the projected components of the force $F = 300$ N acting along the x and y axes.



Prob. 2–131

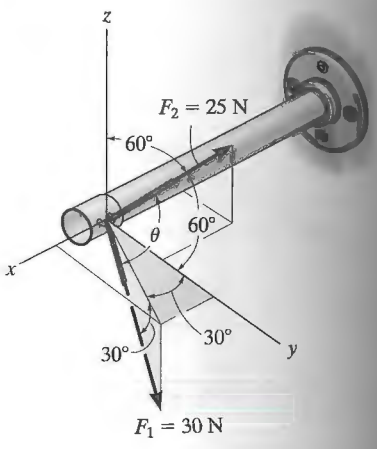
*2–132. Determine the magnitude of the projected component of the force $F = 300$ N acting along line OA.



Prob. 2–132

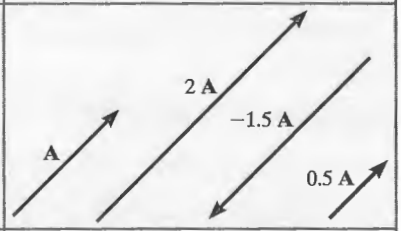
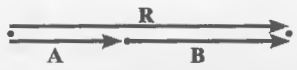
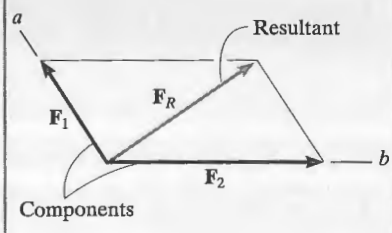
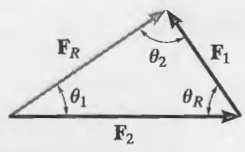
•**2–133.** Two cables exert forces on the pipe. Determine the magnitude of the projected component of $\mathbf{F}_1$ along the line of action of $\mathbf{F}_2$.

2–134. Determine the angle θ between the two cables attached to the pipe.



Probs. 2–133/134

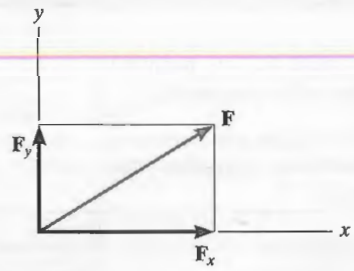
CHAPTER REVIEW

A [redacted] is a positive or negative number; e.g., mass and temperature. A vector has a magnitude and direction, where the arrowhead represents the sense of the vector.		
Multiplication or division of a vector by a scalar will change only the magnitude of the vector. If the scalar is negative, the sense of the vector will change so that it acts in the opposite sense.		
If vectors are collinear, the resultant is simply the algebraic or scalar addition.	$R = A + B$	
Parallelogram Law Two forces add according to the parallelogram law. The *components* form the sides of the parallelogram and the *resultant* is the diagonal. To find the components of a force along any two axes, extend lines from the head of the force, parallel to the axes, to form the components. To obtain the components or the resultant, show how the forces add by tip-to-tail using the triangle rule, and then use the law of cosines and the law of sines to calculate their values.	$F_R = \sqrt{F_1{}^2 + F_2{}^2 - 2\,F_1F_2\cos\theta_R}$ $\dfrac{F_1}{\sin\theta_1} = \dfrac{F_2}{\sin\theta_2} = \dfrac{F_R}{\sin\theta_R}$	 

2

Rectangular Components: Two Dimensions

Vectors $\mathbf{F}_x$ and $\mathbf{F}_y$ are rectangular components of $\mathbf{F}$.



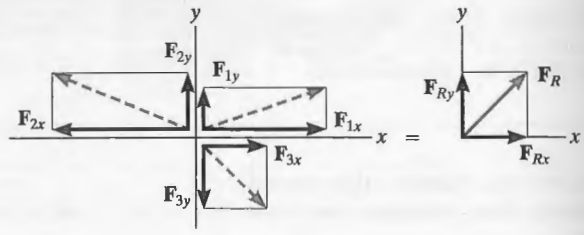
The resultant force is determined from the algebraic sum of its components.

$$F_{Rx} = \Sigma F_x$$

$$F_{Ry} = \Sigma F_y$$

$$F_R = \sqrt{(F_{Rx})^2 + (F_{Ry})^2}$$

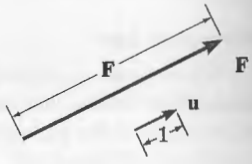
$$\theta = \tan^{-1}\left|\frac{F_{Ry}}{F_{Rx}}\right|$$



Cartesian Vectors

The unit vector $\mathbf{u}$ has a length of one, no units, and it points in the direction of the vector $\mathbf{F}$.

$$\mathbf{u} = \frac{\mathbf{F}}{F}$$



A force can be resolved into its Cartesian components along the x, y, z axes so that $\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k}$.

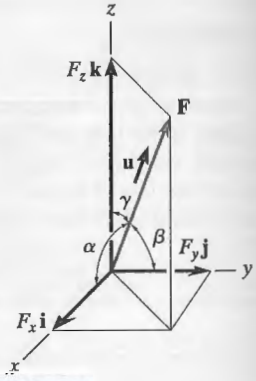
The magnitude of $\mathbf{F}$ is determined from the positive square root of the sum of the squares of its components.

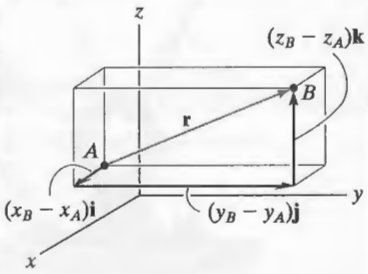
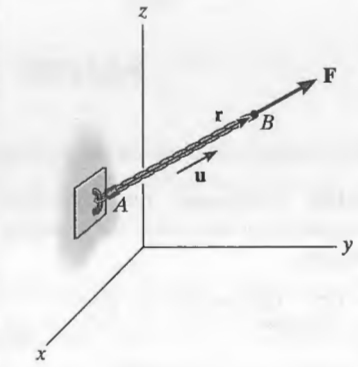
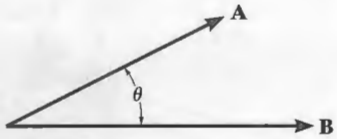
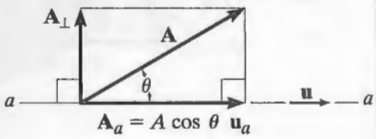
$$F = \sqrt{F_x^2 + F_y^2 + F_z^2}$$

The coordinate direction angles α, β, γ are determined by formulating a unit vector in the direction of $\mathbf{F}$. The x, y, z components of $\mathbf{u}$ represent $\cos\alpha, \cos\beta, \cos\gamma$.

$$\mathbf{u} = \frac{\mathbf{F}}{F} = \frac{\mathbf{F_x}}{\mathbf{F}}\mathbf{i} + \frac{F_y}{F}\mathbf{j} + \frac{F_z}{F}\mathbf{k}$$

$$\mathbf{u} = \cos\alpha\,\mathbf{i} + \cos\beta\,\mathbf{j} + \cos\gamma\,\mathbf{k}$$



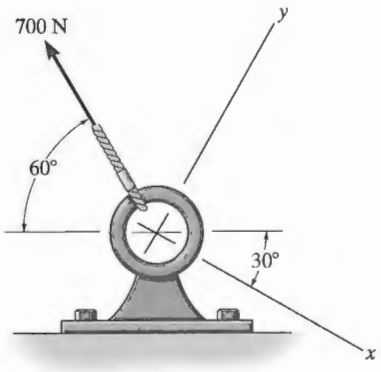
The coordinate direction angles are related so that only two of the three angles are independent of one another.	$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$	
To find the resultant of a concurrent force system, express each force as a Cartesian vector and add the **i**, **j**, **k** components of all the forces in the system.	$\mathbf{F}_R = \Sigma\mathbf{F} = \Sigma F_x\mathbf{i} + \Sigma F_y\mathbf{j} + \Sigma F_z\mathbf{k}$	
Position and Force Vectors A position vector locates one point in space relative to another. The easiest way to formulate the components of a position vector is to determine the distance and direction that one must travel along the x, y, and z directions—going from the tail to the head of the vector.	$\mathbf{r} = (x_B - x_A)\mathbf{i} + (y_B - y_A)\mathbf{j} + (z_B - z_A)\mathbf{k}$	
If the line of action of a force passes through points A and B, then the force acts in the same direction as the position vector **r**, which is defined by the unit vector u. The force can then be expressed as a Cartesian vector.	$\mathbf{F} = F\mathbf{u} = F\left(\frac{\mathbf{r}}{r}\right)$	
Dot Product The dot product between two vectors **A** and **B** yields a scalar. If **A** and **B** are expressed in Cartesian vector form, then the dot product is the sum of the products of their x, y, and z components	$\mathbf{A}\cdot\mathbf{B} = AB\cos\theta = A_xB_x + A_yB_y + A_zB_z$	
The dot product can be used to determine the angle between **A** and **B**.	$\theta = \cos^{-1}\left(\frac{\mathbf{A}\cdot\mathbf{B}}{AB}\right)$	
The dot product is also used to determine the projected component of a vector **A** onto an axis aa defined by its unit vector $\mathbf{u}_a$.	$\mathbf{A}_a = A\cos\theta\,\mathbf{u}_a = (\mathbf{A}\cdot\mathbf{u}_a)\mathbf{u}_a$	

Hibbeler, RC. (2012). Force Vectors. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 17-84). Singapore: Pearson Education South Asia Pte Ltd.

MasteringENGINEERING
Customized coaching is available

REVIEW PROBLEMS

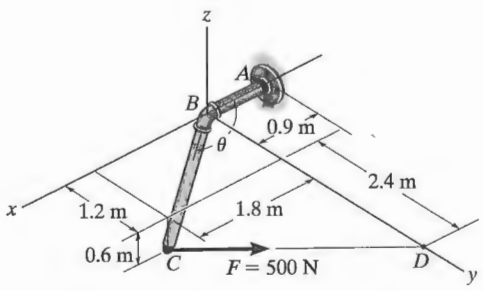
2–135. Determine the x and y components of the 700-N force.



Prob. 2–135

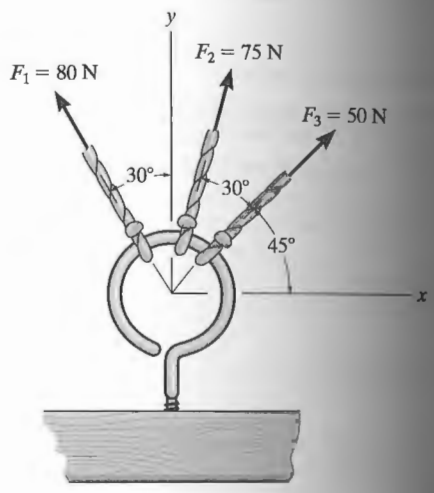
***2–136.** Determine the magnitude of the projected component of the 500-N force acting along the axis BC of the pipe.

•2–137. Determine the angle θ between pipe segments BA and BC.



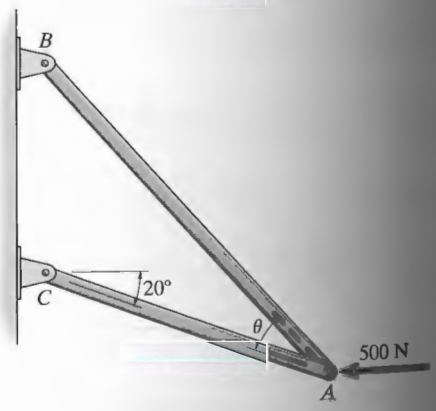
Probs. 2–136/137

2–138. Determine the magnitude and direction of the resultant $\mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3$ of the three forces by first finding the resultant $\mathbf{F}' = \mathbf{F}_1 + \mathbf{F}_3$ and then forming $\mathbf{F}_R = \mathbf{F}' + \mathbf{F}_2$. Specify its direction measured counterclockwise from the positive x axis.



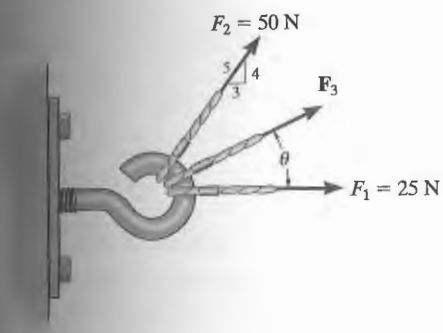
Prob. 2–138

2–139. Determine the design angle θ ($\theta < 90°$) between the two struts so that the 500-N horizontal force has a component of 600 N directed from A toward C. What is the component of force acting along member BA?



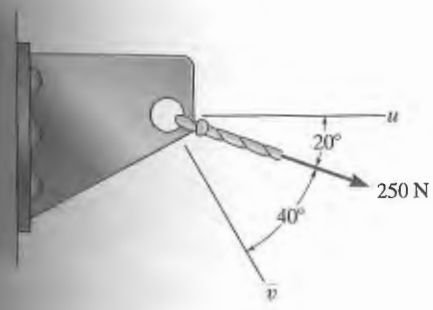
Prob. 2–139

•2–140. Determine the magnitude and direction of the *smallest* force $\mathbf{F}_3$ so that the resultant force of all three forces has a magnitude of 100 N.



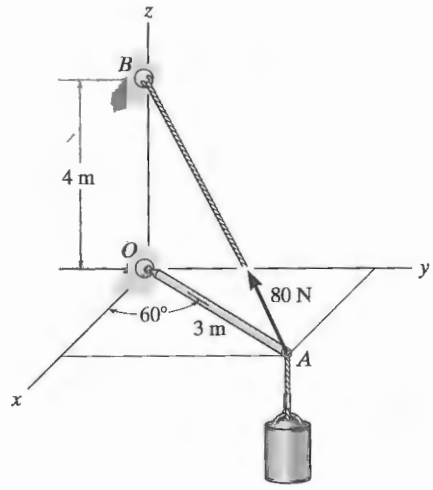
Prob. 2–140

•2–141. Resolve the 250-N force into components acting along the u and v axes and determine the magnitudes of these components.



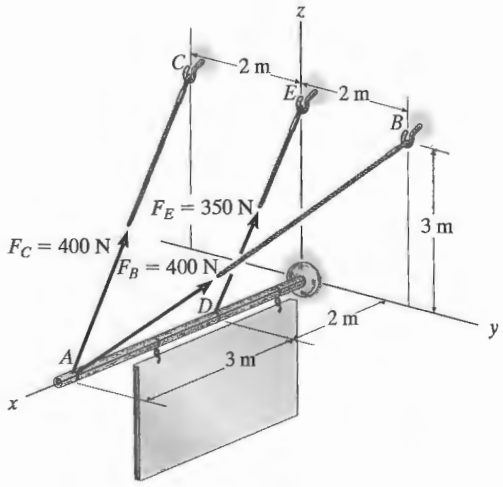
Prob. 2–141

2–142. Cable AB exerts a force of 80 N on the end of the 3-m-long boom OA. Determine the magnitude of the projection of this force along the boom.

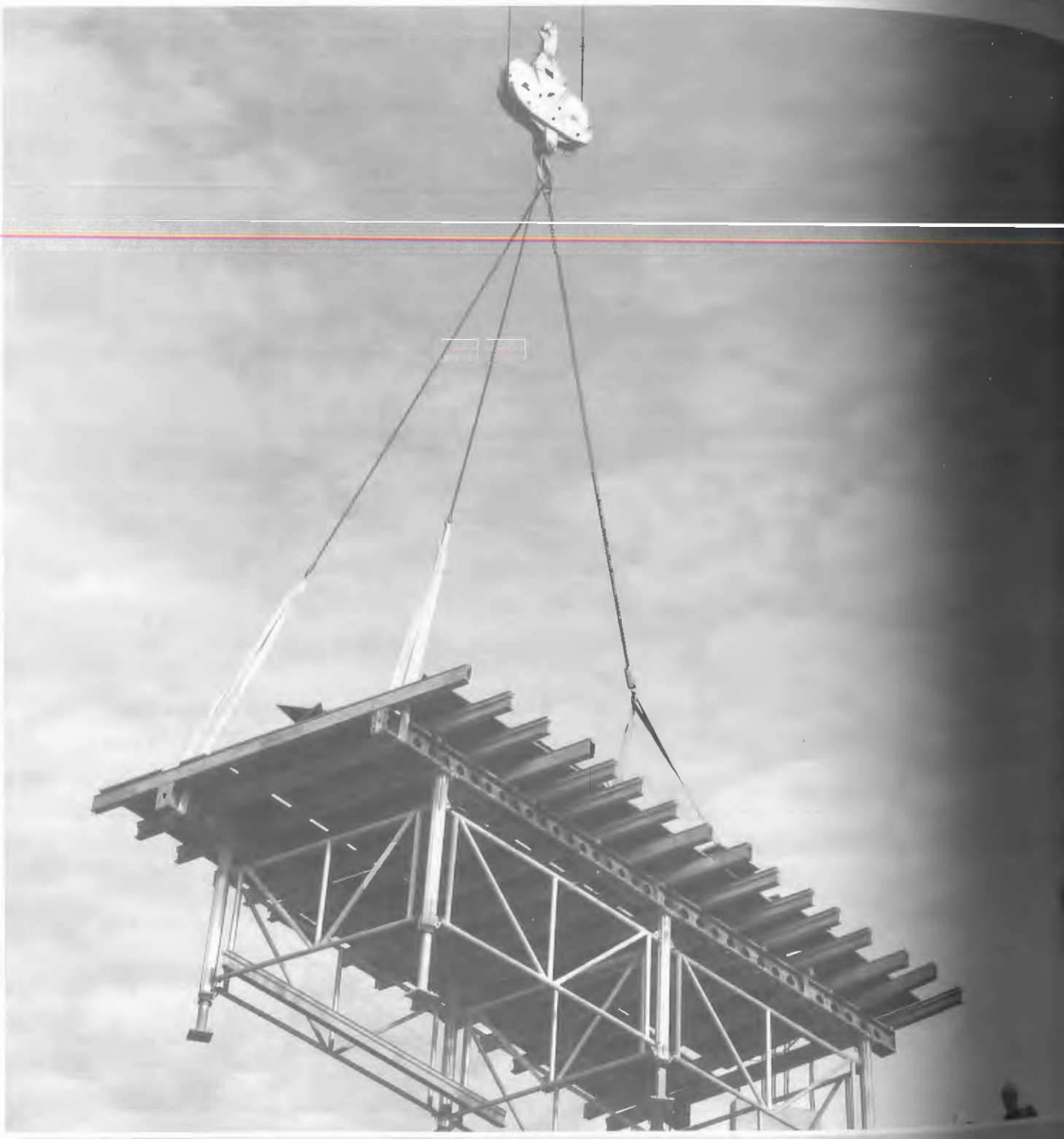


Prob. 2–142

2–143. The three supporting cables exert the forces shown on the sign. Represent each force as a Cartesian vector.



Prob. 2–143



Whenever cables are used for hoisting loads, they must be selected so that they do not fail when they are placed at their points of attachment. In this chapter, we will show how to calculate cable loadings for such cases.

Hibbeler, RC. (2012). Equilibrium of a Particle. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 85-116). Singapore: Pearson Education South Asia Pte Ltd.

Equilibrium of a Particle

3

CHAPTER OBJECTIVES

- To introduce the concept of the free-body diagram for a particle.
- To show how to solve particle equilibrium problems using the equations of equilibrium.

3.1 Condition for the Equilibrium of a Particle

A particle is said to be in *equilibrium* if it remains at rest if originally at rest, or has a constant velocity if originally in motion. Most often, however, the term "equilibrium" or, more specifically, "static equilibrium" is used to describe an object at rest. To maintain equilibrium, it is *necessary* to satisfy Newton's first law of motion, which requires the *resultant force* acting on a particle to be equal to *zero*. This condition may be stated mathematically as

$$\Sigma \mathbf{F} = \mathbf{0} \tag{3–1}$$

where $\Sigma \mathbf{F}$ is the vector *sum of all the forces* acting on the particle.

Not only is Eq. 3–1 a necessary condition for equilibrium, it is also a *sufficient* condition. This follows from Newton's second law of motion, which can be written as $\Sigma \mathbf{F} = m\mathbf{a}$. Since the force system satisfies Eq. 3–1, then $m\mathbf{a} = \mathbf{0}$, and therefore the particle's acceleration $\mathbf{a} = \mathbf{0}$. Consequently, the particle indeed moves with constant velocity or remains at rest.

Hibbeler, RC. (2012). Equilibrium of a Particle. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 85-116). Singapore: Pearson Education South Asia Pte Ltd.

3.2 The Free-Body Diagram

To apply the equation of equilibrium, we must account for *all* the known and unknown forces ($\Sigma \mathbf{F}$) which act on the particle. The best way to do this is to think of the particle as isolated and "free" from its surroundings. A drawing that shows the particle with *all* the forces that act on it is called a *free-body diagram (FBD)*.

Before presenting a formal procedure as to how to draw a free-body diagram, we will first consider two types of connections often encountered in particle equilibrium problems.

Springs. If a *linearly elastic spring* (or cord) of undeformed length l_o is used to support a particle, the length of the spring will change in direct proportion to the force $\mathbf{F}$ acting on it, Fig. 3–1. A characteristic that defines the "elasticity" of a spring is the *spring constant* or *stiffness* k.

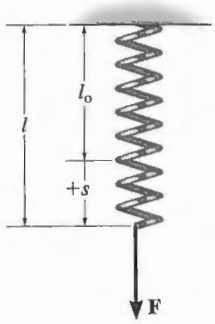


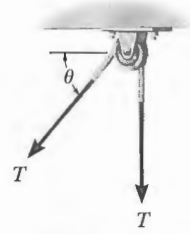
Fig. 3–1

The magnitude of force exerted on a linearly elastic spring which has a stiffness k and is deformed (elongated or compressed) a distance $s = l - l_o$, measured from its *unloaded* position, is

$$\boxed{F = ks} \tag{3–2}$$

If s is positive, causing an elongation, then $\mathbf{F}$ must pull on the spring; whereas if s is negative, causing a shortening, then $\mathbf{F}$ must push on it. For example, if the spring in Fig. 3–1 has an unstretched length of 0.8 m and a stiffness $k = 500$ N/m and it is stretched to a length of 1 m, so that $s = l - l_o = 1\text{ m} - 0.8\text{ m} = 0.2\text{ m}$, then a force $F = ks = 500\text{ N/m}(0.2\text{ m}) = 100\text{ N}$ is needed.

Cables and Pulleys. Unless otherwise stated, throughout this book, except in Sec. 7.4, all cables (or cords) will be assumed to have negligible weight and they cannot stretch. Also, a cable can support *only* a tension or "pulling" force, and this force always acts in the direction of the cable. In Chapter 5, it will be shown that the tension force developed in a *continuous cable* which passes over a frictionless pulley must have a *constant* magnitude to keep the cable in equilibrium. Hence, for any angle θ, shown in Fig. 3–2, the cable is subjected to a constant tension T throughout its length.



Cable is in tension

Fig. 3–2

Pearson Custom Publishing exclusively for The University of Melbourne

Procedure for Drawing a Free-Body Diagram

Since we must account for *all the forces acting on the particle* when applying the equations of equilibrium, the importance of first drawing a free-body diagram cannot be overemphasized. To construct a free-body diagram, the following three steps are necessary.

Draw Outlined Shape.

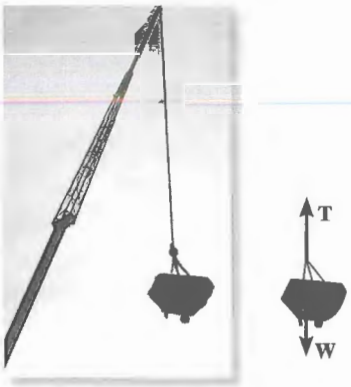
Imagine the particle to be *isolated* or cut "free" from its surroundings by drawing its outlined shape.

Show All Forces.

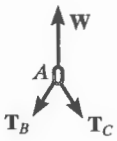
Indicate on this sketch *all* the forces that act *on the particle*. These forces can be *active forces*, which tend to set the particle in motion, or they can be *reactive forces* which are the result of the constraints or supports that tend to prevent motion. To account for all these forces, it may be helpful to trace around the particle's boundary, carefully noting each force acting on it.

Identify Each Force.

The forces that are *known* should be labeled with their proper magnitudes and directions. Letters are used to represent the magnitudes and directions of forces that are unknown.



The bucket is held in equilibrium by the cable, and instinctively we know that the force in the cable must equal the weight of the bucket. By drawing a free-body diagram of the bucket we can understand why this is so. This diagram shows that there are only two forces *acting on the bucket*, namely, its weight **W** and the force **T** of the cable. For equilibrium, the resultant of these forces must be equal to zero, and so $T = W$.

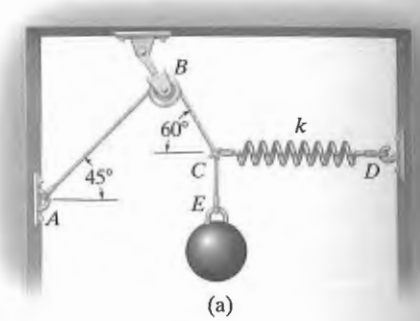


The spool has a weight W and is suspended from the crane boom. If we wish to obtain the forces in cables AB and AC, then we should consider the free-body diagram of the ring at A. Here the cables AD exert a resultant force of **W** on the ring and the condition of equilibrium is used to obtain $\mathbf{T}_B$ and $\mathbf{T}_C$.

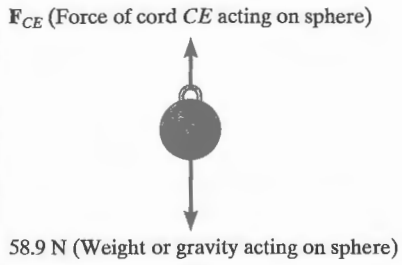
Hibbeler, RC. (2012). Equilibrium of a Particle. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 85-116). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 3.1

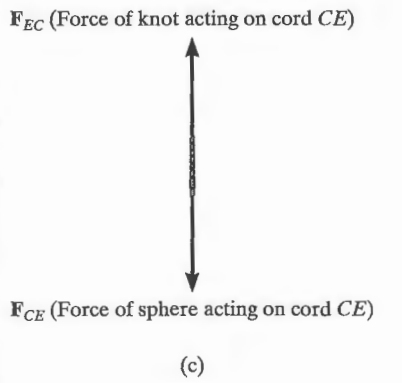
The sphere in Fig. 3–3*a* has a mass of 6 kg and is supported as shown. Draw a free-body diagram of the sphere, the cord *CE*, and the knot at *C*.



(a)



(b)



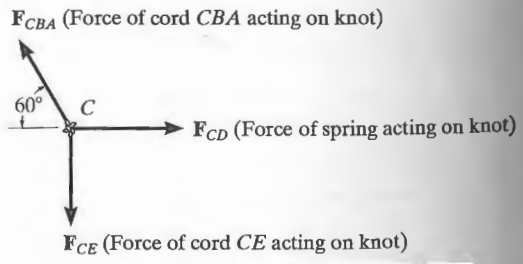
(c)

SOLUTION

Sphere. By inspection, there are only two forces acting on the sphere, namely, its weight, 6 kg (9.81 m/s^2) = 58.9 N, and the force of cord *CE*. The free-body diagram is shown in Fig. 3–3*b*.

Cord CE. When the cord *CE* is isolated from its surroundings, its free-body diagram shows only two forces acting on it, namely, the force of the sphere and the force of the knot, Fig. 3–3*c*. Notice that $\mathbf{F}_{CE}$ shown here is equal but opposite to that shown in Fig 3–3*b*, a consequence of Newton's third law of action–reaction. Also, $\mathbf{F}_{CE}$ and $\mathbf{F}_{EC}$ pull on the cord and keep it in tension so that it doesn't collapse. For equilibrium, $F_{CE} = F_{EC}$.

Knot. The knot at *C* is subjected to three forces, Fig. 3–3*d*. They are caused by the cords *CBA* and *CE* and the spring *CD*. As required, the free-body diagram shows all these forces labeled with their magnitudes and directions. It is important to recognize that the weight of the sphere does not directly act on the knot. Instead, the cord *CE* subjects the knot to this force.



(d)

Fig. 3–3

3.3 Coplanar Force Systems

If a particle is subjected to a system of coplanar forces that lie in the x–y plane as in Fig. 3–4, then each force can be resolved into its **i** and **j** components. For equilibrium, these forces must sum to produce a zero force resultant, i.e.,

$$\Sigma \mathbf{F} = \mathbf{0}$$
$$\Sigma F_x \mathbf{i} + \Sigma F_y \mathbf{j} = \mathbf{0}$$

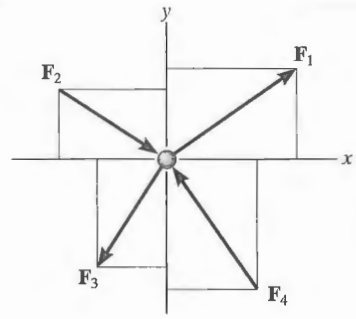


Fig. 3–4

For this vector equation to be satisfied, the force's x and y components must both be equal to zero. Hence,

$$\boxed{\begin{aligned}\Sigma F_x &= 0\\ \Sigma F_y &= 0\end{aligned}} \qquad (3\text{–}3)$$

These two equations can be solved for at most two unknowns, generally represented as angles and magnitudes of forces shown on the particle's free-body diagram.

When applying each of the two equations of equilibrium, we must account for the sense of direction of any component by using an *algebraic sign* which corresponds to the arrowhead direction of the component along the x or y axis. It is important to note that if a force has an *unknown magnitude*, then the arrowhead sense of the force on the free-body diagram can be *assumed*. Then if the *solution* yields a *negative scalar*, this indicates that the sense of the force is opposite to that which was assumed.

For example, consider the free-body diagram of the particle subjected to the two forces shown in Fig. 3–5. Here it is *assumed* that the *unknown force* **F** acts to the right to maintain equilibrium. Applying the equation of equilibrium along the x axis, we have

$$\xrightarrow{+} \Sigma F_x = 0; \qquad +F + 10\,\text{N} = 0$$

Both terms are "positive" since both forces act in the positive x direction. When this equation is solved, $F = -10\,\text{N}$. Here the *negative sign* indicates that **F** must act to the left to hold the particle in equilibrium, Fig. 3–5. Notice that if the $+x$ axis in Fig. 3–5 were directed to the left, both terms in the above equation would be negative, but again, after solving, $F = -10\,\text{N}$, indicating that **F** would be directed to the left.

Fig. 3–5

3

Procedure for Analysis

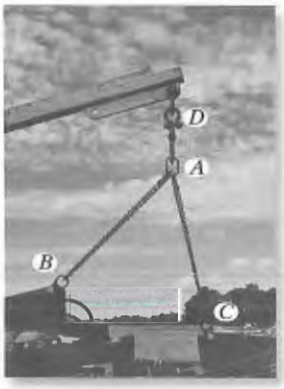
Coplanar force equilibrium problems for a particle can be solved using the following procedure.

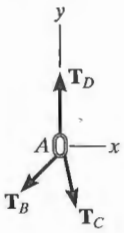
Free-Body Diagram.

- Establish the x, y axes in any suitable orientation.
- Label all the known and unknown force magnitudes and directions on the diagram.
- The sense of a force having an unknown magnitude can be assumed.

Equations of Equilibrium.

- Apply the equations of equilibrium, $\Sigma F_x = 0$ and $\Sigma F_y = 0$.
- Components are positive if they are directed along a positive axis, and negative if they are directed along a negative axis.
- If more than two unknowns exist and the problem involves a spring, apply $F = ks$ to relate the spring force to the deformation s of the spring.
- Since the magnitude of a force is always a positive quantity, then if the solution for a force yields a negative result, this indicates its sense is the reverse of that shown on the free-body diagram.



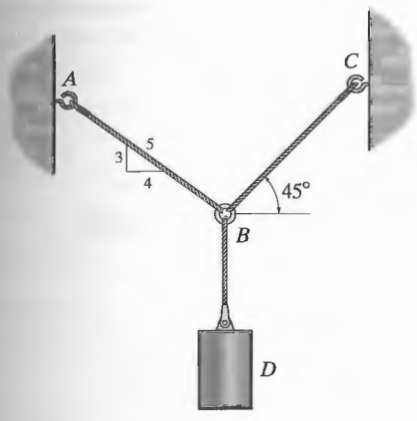


The chains exert three forces on the ring at A, as shown on its free-body diagram. The ring will not move, or will move with constant velocity, provided the summation of these forces along the x and along the y axis is zero. If one of the three forces is known, the magnitudes of the other two forces can be obtained from the two equations of equilibrium.

...ing exclusively for The University of Melbourne

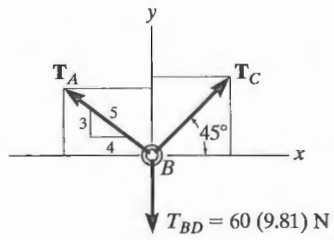
EXAMPLE 3.2

Determine the tension in cables BA and BC necessary to support the 60-kg cylinder in Fig. 3-6a.



(a)

(b)



(c)

Fig. 3–6

SOLUTION

Free-Body Diagram. Due to equilibrium, the weight of the cylinder causes the tension in cable BD to be $T_{BD} = 60(9.81)$ N, Fig. 3-6b. The forces in cables BA and BC can be determined by investigating the equilibrium of ring B. Its free-body diagram is shown in Fig. 3-6c. The magnitudes of $\mathbf{T}_A$ and $\mathbf{T}_C$ are unknown, but their directions are known.

Equations of Equilibrium. Applying the equations of equilibrium along the x and y axes, we have

$$\overset{+}{\rightarrow} \Sigma F_x = 0; \qquad T_C \cos 45° - \left(\tfrac{4}{5}\right)T_A = 0 \qquad (1)$$

$$+\uparrow \Sigma F_y = 0; \quad T_C \sin 45° + \left(\tfrac{3}{5}\right)T_A - 60(9.81)\text{ N} = 0 \qquad (2)$$

Equation (1) can be written as $T_A = 0.8839T_C$. Substituting this into Eq. (2) yields

$$T_C \sin 45° + \left(\tfrac{3}{5}\right)(0.8839T_C) - 60(9.81)\text{ N} = 0$$

So that

$$T_C = 475.66\text{ N} = 476\text{ N} \qquad \textit{Ans.}$$

Substituting this result into either Eq. (1) or Eq. (2), we get

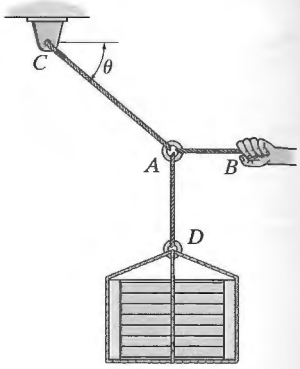
$$T_A = 420\text{ N} \qquad \textit{Ans.}$$

NOTE: The accuracy of these results, of course, depends on the accuracy of the data, i.e., measurements of geometry and loads. For most engineering work involving a problem such as this, the data as measured to three significant figures would be sufficient.

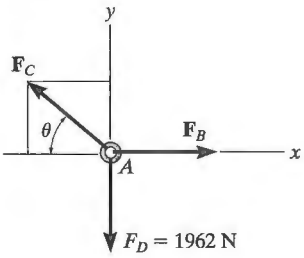
Hibbeler, RC. (2012). Equilibrium of a Particle. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 85-116). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 3.3

The 200-kg crate in Fig. 3-7*a* is suspended using the ropes *AB* and *AC*. Each rope can withstand a maximum force of 10 kN before it breaks. If *AB* always remains horizontal, determine the smallest angle θ to which the crate can be suspended before one of the ropes breaks.



(a)



(b)

Fig. 3–7

SOLUTION

Free-Body Diagram. We will study the equilibrium of ring *A*. There are three forces acting on it, Fig. 3-7*b*. The magnitude of $\mathbf{F}_D$ is equal to the weight of the crate, i.e., $F_D = 200\,(9.81)\text{ N} = 1962\text{ N} < 10\text{ kN}$.

Equations of Equilibrium. Applying the equations of equilibrium along the *x* and *y* axes,

$$\xrightarrow{+} \Sigma F_x = 0; \qquad -F_C \cos\theta + F_B = 0; \quad F_C = \frac{F_B}{\cos\theta} \tag{1}$$

$$+\uparrow \Sigma F_y = 0; \qquad F_C \sin\theta - 1962\text{ N} = 0 \tag{2}$$

From Eq. (1), F_C is always greater than F_B since $\cos\theta \leq 1$. Therefore, rope *AC* will reach the maximum tensile force of 10 kN *before* rope *AB*. Substituting $F_C = 10$ kN into Eq. (2), we get

$$[10(10^3)\text{ N}]\sin\theta - 1962\text{ N} = 0$$

$$\theta = \sin^{-1}(0.1962) = 11.31° \qquad \textit{Ans.}$$

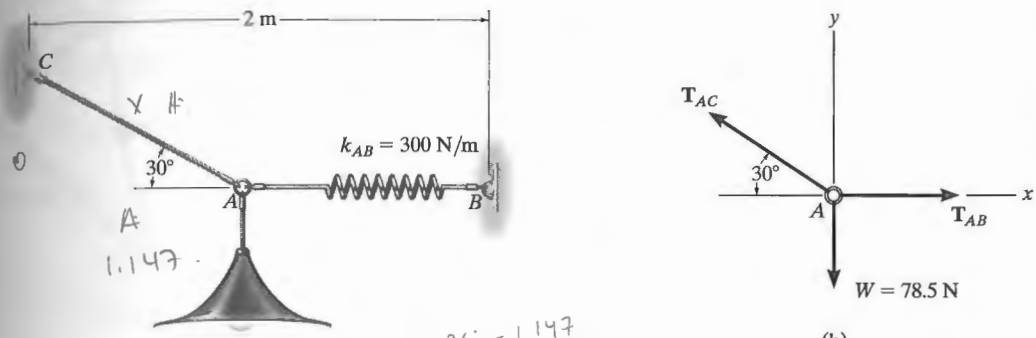
The force developed in rope *AB* can be obtained by substituting the values for θ and F_C into Eq. (1).

$$10(10^3)\text{ N} = \frac{F_B}{\cos 11.31°}$$

$$F_B = 9.81\text{ kN}$$

3.4

Determine the required length of cord AC in Fig. 3–8*a* so that the [redacted] can be suspended in the position shown. The *undeformed* length of spring AB is $l'_{AB} = 0.4$ m, and the spring has a stiffness of $k_{AB} = 300$ N/m.



(a) (b)

Fig. 3–8

$\cos 30° = 1.147$

$\Sigma F_x = 300N + T_A \cos 30° = 0$

$T_A = 5N = 8 \times 9.8 = 0$

SOLUTION

If the force in spring AB is known, the stretch of the spring can be found using $F = ks$. From the problem geometry, it is then possible to calculate the required length of AC.

Free-Body Diagram. The lamp has a weight $W = 8(9.81) = 78.5$ N and so the free-body diagram of the ring at A is shown in Fig. 3–8*b*.

Equations of Equilibrium. Using the x, y axes,

$\xrightarrow{+} \Sigma F_x = 0; \qquad T_{AB} - T_{AC} \cos 30° = 0$

$+\uparrow \Sigma F_y = 0; \qquad T_{AC} \sin 30° - 78.5 \text{ N} = 0$

Solving, we obtain

$$T_{AC} = 157.0 \text{ N}$$

$$T_{AB} = 135.9 \text{ N}$$

The stretch of spring AB is therefore

$T_{AB} = k_{AB} s_{AB}; \qquad 135.9 \text{ N} = 300 \text{ N/m}(s_{AB})$

$$s_{AB} = 0.453 \text{ m}$$

so the stretched length is

$$l_{AB} = l'_{AB} + s_{AB}$$

$$l_{AB} = 0.4 \text{ m} + 0.453 \text{ m} = 0.853 \text{ m}$$

The horizontal distance from C to B, Fig. 3–8*a*, requires

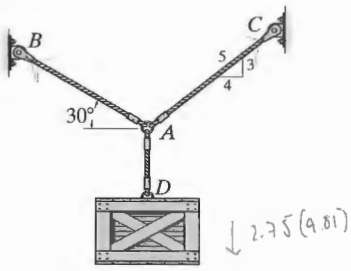
$$2 \text{ m} = l_{AC} \cos 30° + 0.853 \text{ m}$$

$$l_{AC} = 1.32 \text{ m} \qquad \textit{Ans.}$$

FUNDAMENTAL PROBLEMS

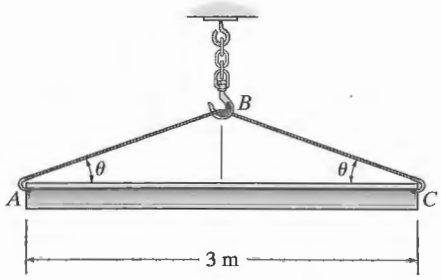
All problem solutions must include an FBD.

F3–1. The crate has a weight of 2.75 kN. Determine the force in each supporting cable.



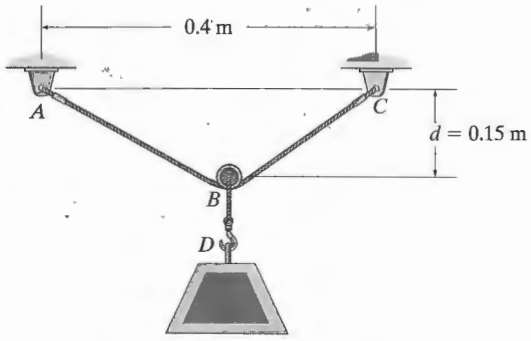
F3–1

F3–2. The beam has a weight of 3.5 kN. Determine the shortest cable ABC that can be used to lift it if the maximum force the cable can sustain is 7.5 kN.



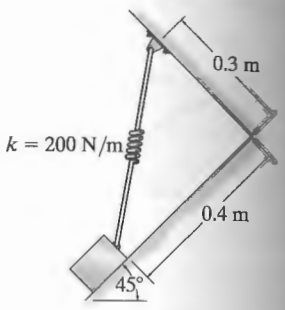
F3–2

F3–3. If the 5-kg block is suspended from the pulley B and the sag of the cord is $d = 0.15$ m, determine the force in cord ABC. Neglect the size of the pulley.



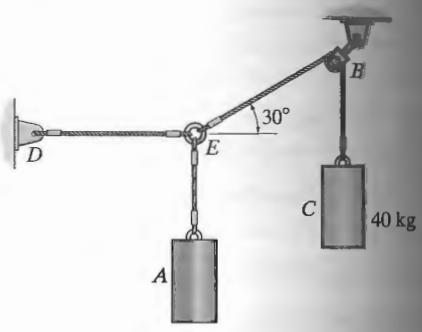
F3–3

F3–4. The block has a mass of 5 kg and rests on the smooth plane. Determine the unstretched length of the spring.



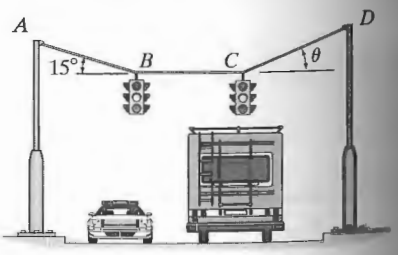
F3–4

F3–5. If the mass of cylinder C is 40 kg, determine the mass of cylinder A in order to hold the assembly in the position shown.



F3–5

F3–6. Determine the tension in cables AB, BC, and CD necessary to support the 10-kg and 15-kg traffic lights at B and C, respectively. Also, find the angle θ.



F3–6

exclusively for The University of Melbourne

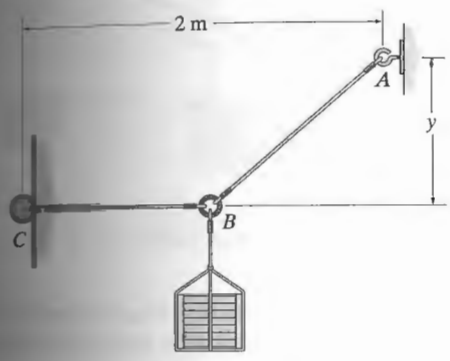
ENGINEERING
Customized coaching is available!

PROBLEMS

All problem solutions must include an FBD.

•3–1. Determine the force in each cord for equilibrium of the 200-kg crate. Cord BC remains horizontal due to the roller at C, and AB has a length of 1.5 m. Set $y = 0.75$ m.

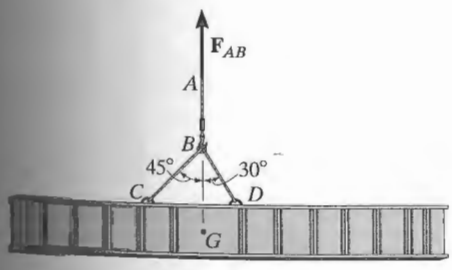
3–2. If the 1.5-m-long cord AB can withstand a maximum force of 3500 N, determine the force in cord BC and the distance y so that the 200-kg crate can be supported.



Probs. 3–1/2

3–3. If the mass of the girder is 3 Mg and its center of mass is located at point G, determine the tension developed in cables AB, BC, and BD for equilibrium.

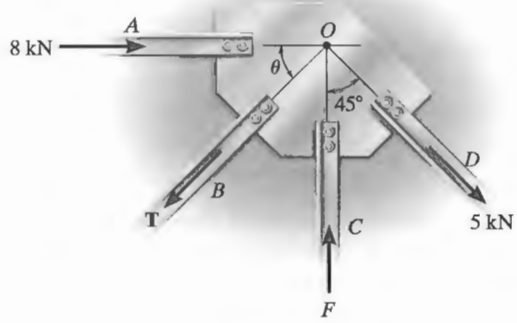
3–4. If cables BD and BC can withstand a maximum tensile force of 20 kN, determine the maximum mass of the girder that can be suspended from cable AB so that neither cable will fail. The center of mass of the girder is located at point G.



Probs. 3–3/4

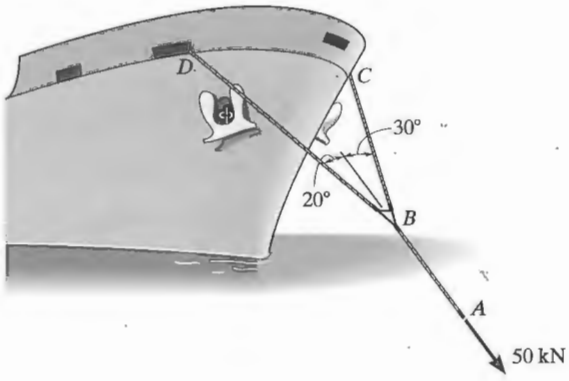
•3–5. The members of a truss are connected to the gusset plate. If the forces are concurrent at point O, determine the magnitudes of **F** and **T** for equilibrium. Take $\theta = 30°$.

3–6. The gusset plate is subjected to the forces of four members. Determine the force in member B and its proper orientation θ for equilibrium. The forces are concurrent at point O. Take $F = 12$ kN.



Probs. 3–5/6

3–7. The towing pendant AB is subjected to the force of 50 kN exerted by a tugboat. Determine the force in each of the bridles, BC and BD, if the ship is moving forward with constant velocity.

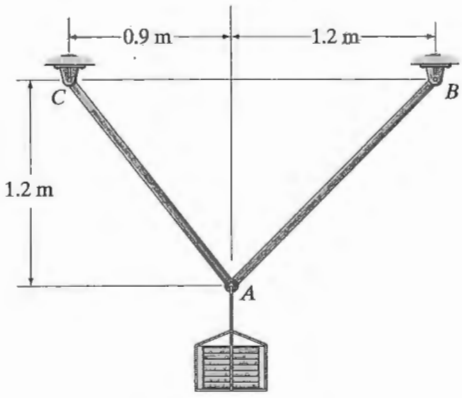


Prob. 3–7

Hibbeler, RC. (2012). Equilibrium of a Particle. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 85-116). Singapore: Pearson Education South Asia Pte Ltd.

***3–8.** Members AC and AB support the 100-kg crate. Determine the tensile force developed in each member.

•3–9. If members AC and AB can support a maximum tension of 1500 N and 1250 N, respectively, determine the largest weight of the crate that can be safely supported.

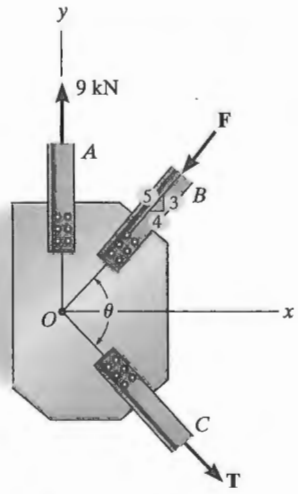


Probs. 3–8/9

3–10. The members of a truss are connected to the gusset plate. If the forces are concurrent at point O, determine the magnitudes of **F** and **T** for equilibrium. Take $\theta = 90°$.

Video Solutions

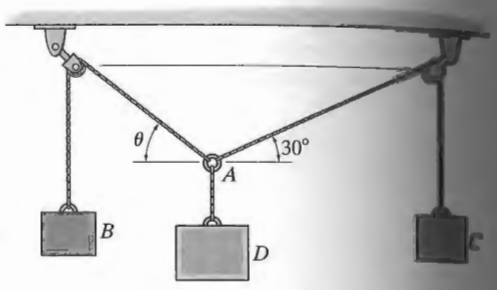
3–11. The gusset plate is subjected to the forces of three members. Determine the tension force in member C and its angle θ for equilibrium. The forces are concurrent at point O. Take $F = 8$ kN.



Probs. 3–10/11

***3–12.** If block B weighs 1 kN and block C weighs 0.5 kN, determine the required weight of block D and the angle θ for equilibrium.

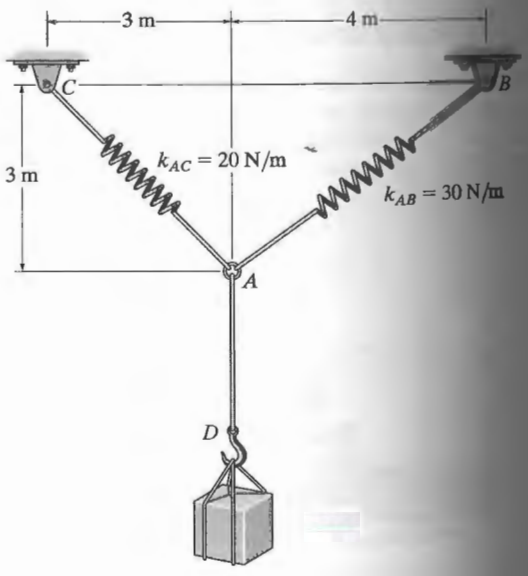
•3–13. If block D weighs 1.5 kN and block B weighs 1.375 kN, determine the required weight of block C and the angle θ for equilibrium.



Probs. 3–12/13

3–14. Determine the stretch in springs AC and AB for equilibrium of the 2-kg block. The springs are shown in the equilibrium position.

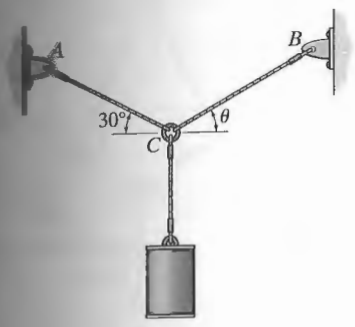
3–15. The unstretched length of spring AB is 3 m. If the block is held in the equilibrium position shown, determine the mass of the block at D.



Probs. 3–14/15

*3–16. Determine the tension developed in wires CA and CB required for equilibrium of the 10-kg cylinder. Take $\theta = 40°$.

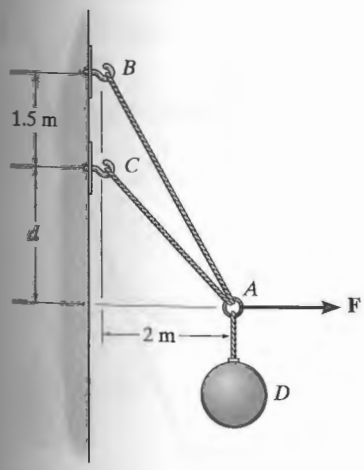
*3–17. If cable CB is subjected to a tension that is twice that of cable CA, determine the angle θ for equilibrium of the 10-kg cylinder. Also, what are the tensions in wires CA and CB?



Probs. 3–16/17

3–18. Determine the forces in cables AC and AB needed to hold the 20-kg ball D in equilibrium. Take $F = 300$ N and $d = 1$ m.

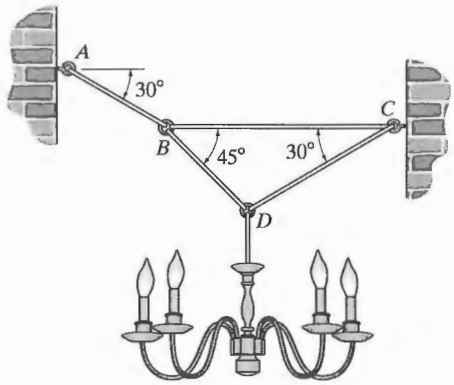
3–19. The ball D has a mass of 20 kg. If a force of $F = 100$ N is applied horizontally to the ring at A, determine the dimension d so that the force in cable AC is zero.



Probs. 3–18/19

*3–20. Determine the tension developed in each wire used to support the 50-kg chandelier.

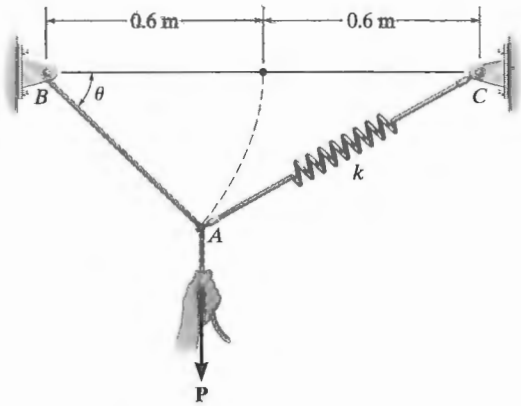
•3–21. If the tension developed in each of the four wires is not allowed to exceed 600 N, determine the maximum mass of the chandelier that can be supported.



Prob. 3–20/21

■3–22. A vertical force $P = 50$ N is applied to the ends of the 0.6 m cord AB and spring AC. If the spring has an unstretched length of 0.6 m, determine the angle θ for equilibrium. Take $k = 250$ N/m.

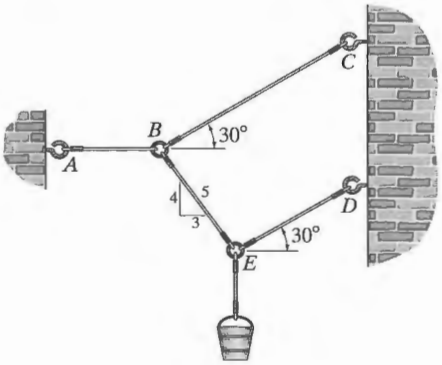
3–23. Determine the unstretched length of spring AC if a force $P = 400$ N causes the angle $\theta = 60°$ for equilibrium. Cord AB is 0.6 m long. Take $k = 850$ N/m.



Probs. 3–22/23

*3–24. If the bucket weighs 0.25 kN, determine the tension developed in each of the wires.

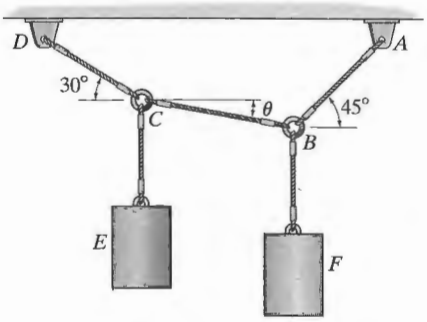
•3–25. Determine the maximum weight of the bucket that the wire system can support so that no single wire develops a tension exceeding 0.5 kN.



Probs. 3–24/25

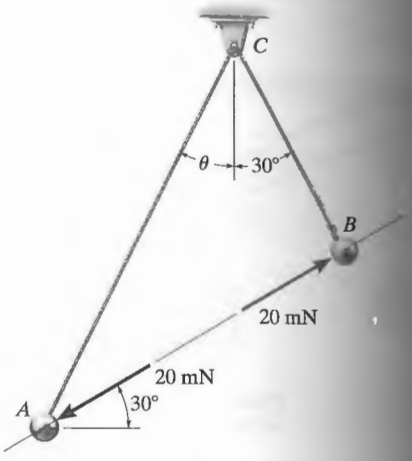
3–26. Determine the tensions developed in wires CD, CB, and BA and the angle θ required for equilibrium of the 15-kg cylinder E and the 30-kg cylinder F.

3–27. If cylinder E weighs 150 N and $\theta = 15°$, determine the weight of cylinder F.



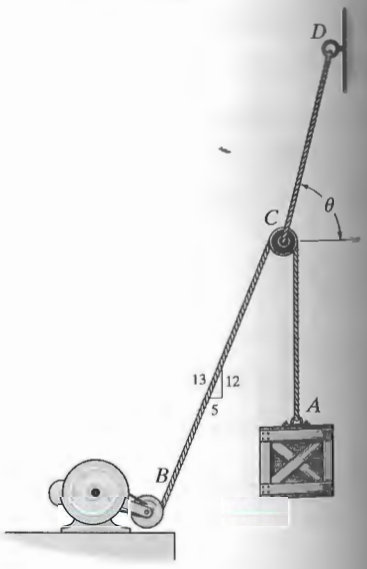
Probs. 3–26/27

*3–28. Two spheres A and B have an equal mass and are electrostatically charged such that the repulsive force acting between them has a magnitude of 20 mN and is directed along line AB. Determine the angle θ, the tension in cords AC and BC, and the mass m of each sphere.



Prob. 3–28

•3–29. The cords BCA and CD can each support a maximum load of 0.5 kN. Determine the maximum weight of the crate that can be hoisted at constant velocity and the angle θ for equilibrium. Neglect the size of the smooth pulley at C.

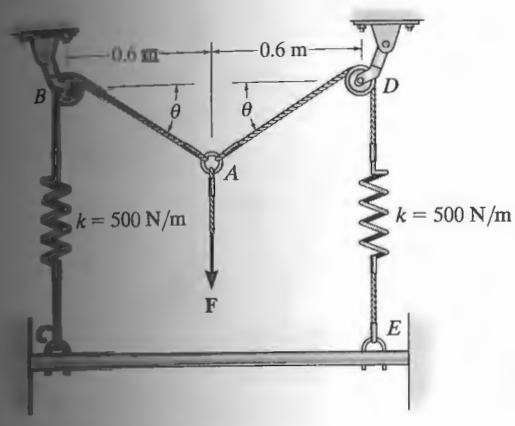


Prob. 3–29

Exclusively for The University of Melbourne

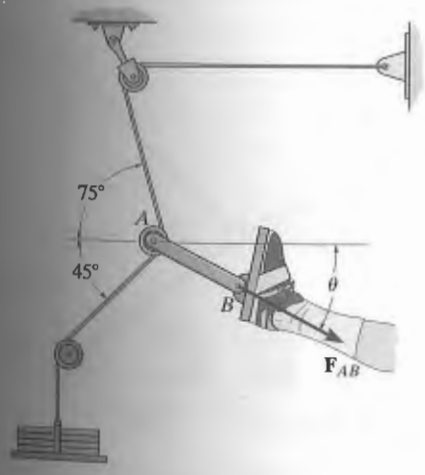
3-30. The springs on the rope assembly are originally stretched when $\theta = 0°$. Determine the tension in each rope when $F = 450$ N. Neglect the size of the pulleys at B and D.

3-31. The springs on the rope assembly are originally stretched 0.3 m when $\theta = 0°$. Determine the vertical force F that must be applied so that $\theta = 30°$.



Probs. 3–30/31

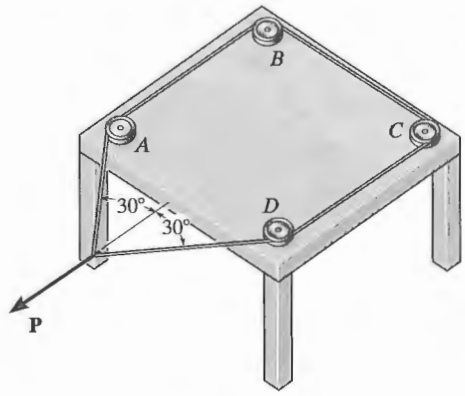
•3–32. Determine the magnitude and direction θ of the equilibrium force F_{AB} exerted along link AB by the tractive apparatus shown. The suspended mass is 10 kg. Neglect the size of the pulley at A.



Prob. 3–32

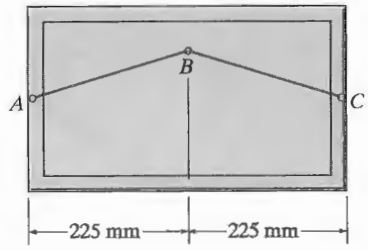
•3–33. The wire forms a loop and passes over the small pulleys at A, B, C, and D. If its end is subjected to a force of $P = 50$ N, determine the force in the wire and the magnitude of the resultant force that the wire exerts on each of the pulleys.

3–34. The wire forms a loop and passes over the small pulleys at A, B, C, and D. If the maximum *resultant force* that the wire can exert on each pulley is 120 N, determine the greatest force P that can be applied to the wire as shown.



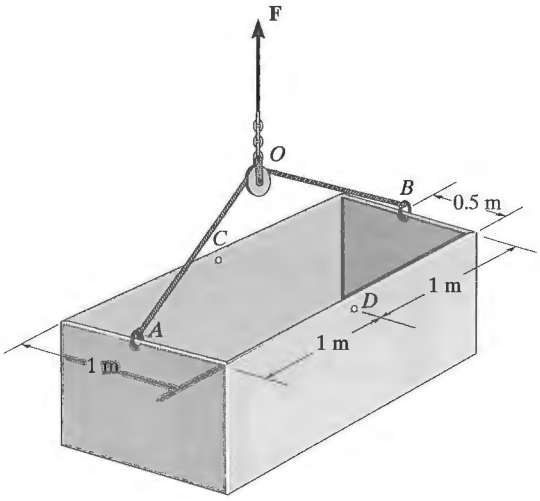
Probs. 3–33/34

3–35. The picture has a weight of 50 N and is to be hung over the smooth pin B. If a string is attached to the frame at points A and C, and the maximum force the string can support is 75 N, determine the shortest string that can be safely used.



Prob. 3–35

***3–36.** The 100-kg uniform tank is suspended by means of a 3-m-long cable, which is attached to the sides of the tank and passes over the small pulley located at O. If the cable can be attached at either points A and B or C and D, determine which attachment produces the least amount of tension in the cable. What is this tension?

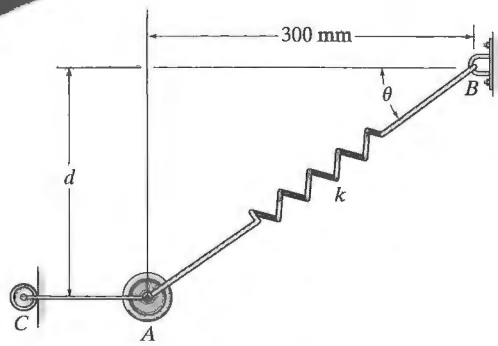


Prob. 3–36

•3–37. The 5-kg weight is supported by the cord AC and roller and by the spring that has a stiffness of $k = 2000$ N/m and an unstretched length of 300 mm. Determine the distance d to where the weight is located when it is in equilibrium.

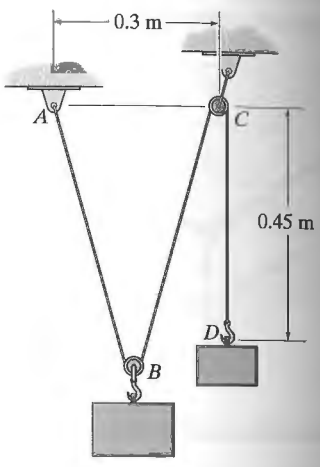
3–38. The 5-kg weight is supported by the cord AC and roller and by a spring. If the spring has an unstretched length of 200 mm and the weight is in equilibrium when $d = 100$ mm, determine the stiffness k of the spring.

Video Solutions



Probs. 3–37/38

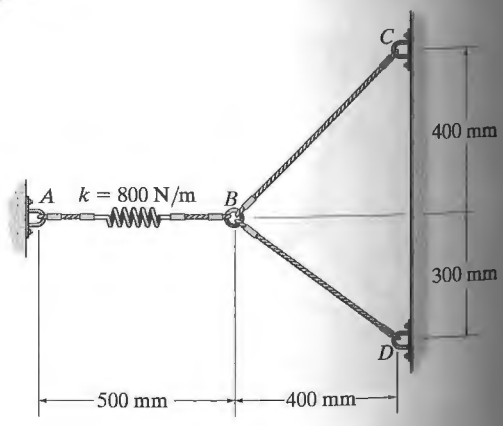
3–39. A "scale" is constructed with a 1.2-m-long cord and the 5-kg block D. The cord is fixed to a pin at A and passes over two *small* pulleys at B and C. Determine the weight of the suspended block at B if the system is in equilibrium.



Prob. 3–39

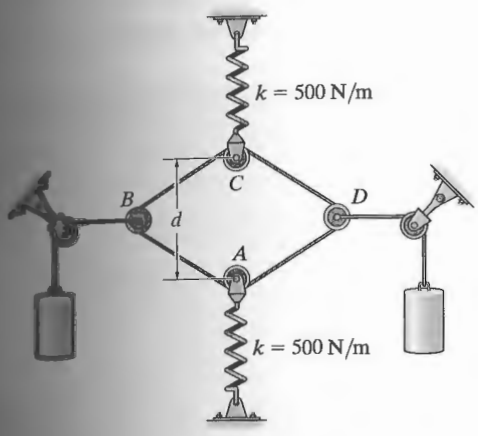
•*3–40. The spring has a stiffness of $k = 800$ N/m and an unstretched length of 200 mm. Determine the force in cables BC and BD when the spring is held in the position shown.

Video Solutions



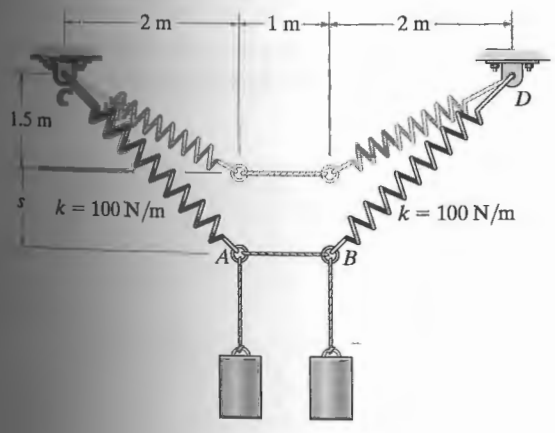
Prob. 3–40

•3–41. A continuous cable of total length 4 m is wrapped around the *small* pulleys at A, B, C, and D. If each spring is stretched 300 mm, determine the mass m of each block. Neglect the weight of the pulleys and cords. The springs are unstretched when $d = 2$ m.



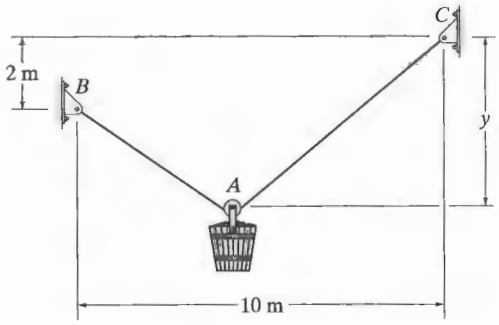
Prob. 3–41

3–42. Determine the mass of each of the two cylinders if they cause a sag of $s = 0.5$ m when suspended from the rings at A and B. Note that $s = 0$ when the cylinders are removed.



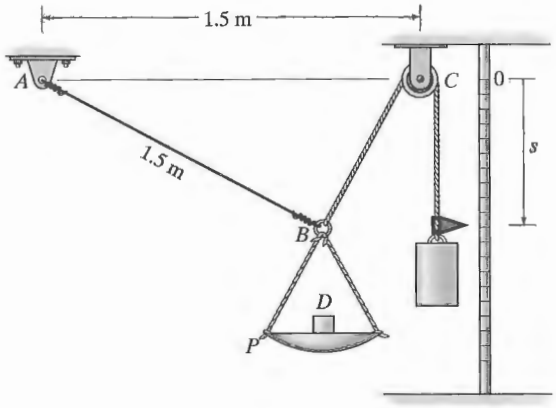
Prob. 3–42

3–43. The pail and its contents have a mass of 60 kg. If the cable BAC is 15 m long, determine the distance y to the pulley at A for equilibrium. Neglect the size of the pulley.



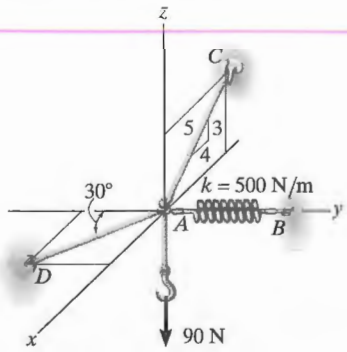
Prob. 3–43

•*3–44. A scale is constructed using the 10-kg mass, the 2-kg pan P, and the pulley and cord arrangement. Cord BCA is 2 m long. If $s = 0.75$ m, determine the mass D in the pan. Neglect the size of the pulley.

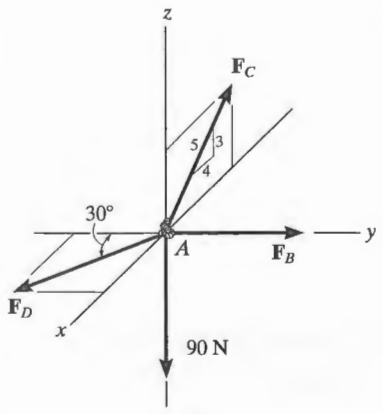


Prob. 3–44

EXAMPLE 3.5



(a)



(b)

Fig. 3–10

A 90-N load is suspended from the hook shown in Fig. 3–10*a*. If the load is supported by two cables and a spring having a stiffness $k = 500$ N/m, determine the force in the cables and the stretch of the spring for equilibrium. Cable AD lies in the x–y plane and cable AC lies in the x–z plane.

SOLUTION

The stretch of the spring can be determined once the force in the spring is determined.

Free-Body Diagram. The connection at A is chosen for the equilibrium analysis since the cable forces are concurrent at this point. The free-body diagram is shown in Fig. 3–10*b*.

Equations of Equilibrium. By inspection, each force can easily be resolved into its x, y, z components, and therefore the three scalar equations of equilibrium can be used. Considering components directed along each positive axis as "positive," we have

$$\Sigma F_x = 0; \qquad F_D \sin 30° - \left(\tfrac{4}{5}\right)F_C = 0 \qquad (1)$$

$$\Sigma F_y = 0; \qquad -F_D \cos 30° + F_B = 0 \qquad (2)$$

$$\Sigma F_z = 0; \qquad \left(\tfrac{3}{5}\right)F_C - 90\text{ N} = 0 \qquad (3)$$

Solving Eq. (3) for F_C, then Eq. (1) for F_D, and finally Eq. (2) for F_B, yields

$$F_C = 150\text{ N} \qquad \textit{Ans.}$$

$$F_D = 240\text{ N} \qquad \textit{Ans.}$$

$$F_B = 207.8\text{ N} \qquad \textit{Ans.}$$

The stretch of the spring is therefore

$$F_B = ks_{AB}$$

$$207.8\text{ N} = (500\text{ N/m})(s_{AB})$$

$$s_{AB} = 0.416\text{ m} \qquad \textit{Ans.}$$

NOTE: Since the results for all the cable forces are positive, each cable is in tension; that is, it pulls on point A as expected, Fig. 3–10*b*.

Exclusively for The University of Melbourne

EXAMPLE 3.6

The 10-kg lamp in Fig. 3-11*a* is suspended from the three equal-length [illegible]. Determine its smallest vertical distance s from the ceiling if the force developed in any cord is not allowed to exceed 50 N.

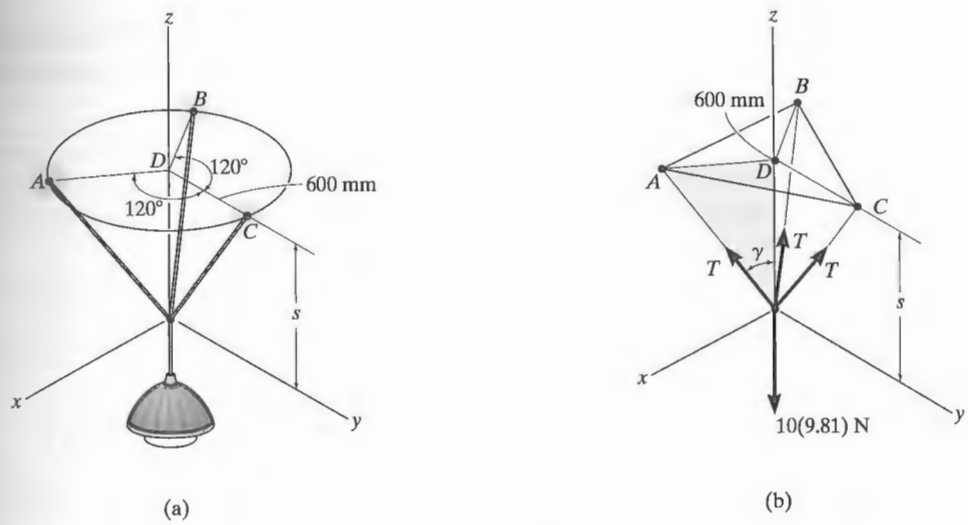


Fig. 3–11

SOLUTION

Free-Body Diagram. Due to symmetry, Fig. 3-11*b*, the distance $DA = DB = DC = 600$ mm. It follows that from $\Sigma F_x = 0$ and $\Sigma F_y = 0$, the tension T in each cord will be the same. Also, the angle between each cord and the z axis is γ.

Equation of Equilibrium. Applying the equilibrium equation along the z axis, with $T = 50$ N, we have

$$\Sigma F_z = 0; \qquad 3[(50\text{ N})\cos\gamma] - 10(9.81)\text{ N} = 0$$

$$\gamma = \cos^{-1}\frac{98.1}{150} = 49.16°$$

From the shaded triangle shown in Fig. 3-11*b*,

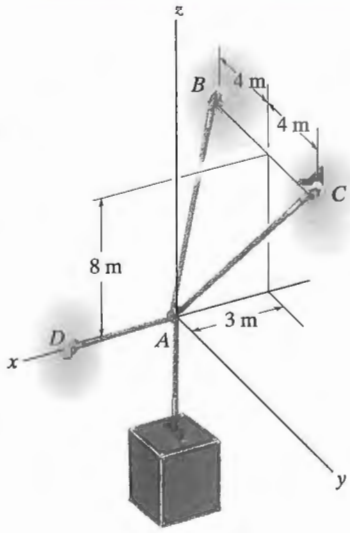
$$\tan 49.16° = \frac{600\text{ mm}}{s}$$

$$s = 519\text{ mm} \qquad \textit{Ans.}$$

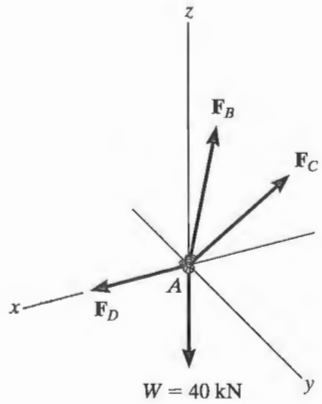
Hibbeler, RC. (2012). Equilibrium of a Particle. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 85-116). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 3.7

Determine the force in each cable used to support the 40-kN (≈ 4000-kg) crate shown in Fig. 3–12*a*.



(a)



(b)

Fig. 3–12

SOLUTION

Free-Body Diagram. As shown in Fig. 3–12*b*, the free-body diagram of point A is considered in order to "expose" the three unknown forces in the cables.

Equations of Equilibrium. First we will express each force in Cartesian vector form. Since the coordinates of points B and C are $B(-3\text{ m}, -4\text{ m}, 8\text{ m})$ and $C(-3\text{ m}, 4\text{ m}, 8\text{ m})$, we have

$$\mathbf{F}_B = F_B\left[\frac{-3\mathbf{i} - 4\mathbf{j} + 8\mathbf{k}}{\sqrt{(-3)^2 + (-4)^2 + (8)^2}}\right]$$

$$= -0.318F_B\mathbf{i} - 0.424F_B\mathbf{j} + 0.848F_B\mathbf{k}$$

$$\mathbf{F}_C = F_C\left[\frac{-3\mathbf{i} + 4\mathbf{j} + 8\mathbf{k}}{\sqrt{(-3)^2 + (4)^2 + (8)^2}}\right]$$

$$= -0.318F_C\mathbf{i} + 0.424F_C\mathbf{j} + 0.848F_C\mathbf{k}$$

$$\mathbf{F}_D = F_D\mathbf{i}$$

$$\mathbf{W} = \{-40\mathbf{k}\}\text{ kN}$$

Equilibrium requires

$$\Sigma\mathbf{F} = \mathbf{0}; \qquad \mathbf{F}_B + \mathbf{F}_C + \mathbf{F}_D + \mathbf{W} = \mathbf{0}$$

$$-0.318F_B\mathbf{i} - 0.424F_B\mathbf{j} + 0.848F_B\mathbf{k}$$

$$-0.318F_C\mathbf{i} + 0.424F_C\mathbf{j} + 0.848F_C\mathbf{k} + F_D\mathbf{i} - 40\mathbf{k} = \mathbf{0}$$

Equating the respective $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components to zero yields

$$\Sigma F_x = 0; \qquad -0.318F_B - 0.318F_C + F_D = 0 \qquad (1)$$

$$\Sigma F_y = 0; \qquad -0.424F_B + 0.424F_C = 0 \qquad (2)$$

$$\Sigma F_z = 0; \qquad 0.848F_B + 0.848F_C - 40 = 0 \qquad (3)$$

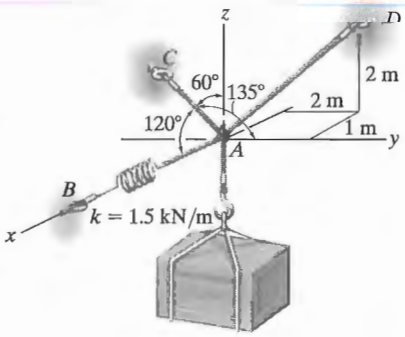
Equation (2) states that $F_B = F_C$. Thus, solving Eq. (3) for F_B and F_C and substituting the result into Eq. (1) to obtain F_D, we have

$$F_B = F_C = 23.6\text{ kN} \qquad \textit{Ans.}$$

$$F_D = 15.0\text{ kN} \qquad \textit{Ans.}$$

EXAMPLE 3.8

Determine the tension in each cord used to support the 100-kg crate shown in Fig. 3–13*a*.



(a)

SOLUTION

Free-Body Diagram. The force in each of the cords can be determined by investigating the equilibrium of point A. The free-body diagram is shown in Fig. 3–13*b*. The weight of the crate is $W = 100(9.81) = 981$ N.

Equations of Equilibrium. Each force on the free-body diagram is first expressed in Cartesian vector form. Using Eq. 2–9 for $\mathbf{F}_C$ and noting point $D(-1\text{ m}, 2\text{ m}, 2\text{ m})$ for $\mathbf{F}_D$, we have

$$\mathbf{F}_B = F_B\mathbf{i}$$

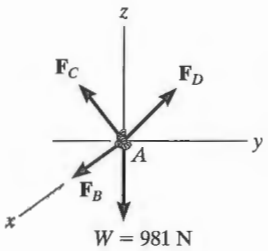
$$\mathbf{F}_C = F_C\cos 120°\mathbf{i} + F_C\cos 135°\mathbf{j} + F_C\cos 60°\mathbf{k}$$

$$= -0.5F_C\mathbf{i} - 0.707F_C\mathbf{j} + 0.5F_C\mathbf{k}$$

$$\mathbf{F}_D = F_D\left[\frac{-1\mathbf{i} + 2\mathbf{j} + 2\mathbf{k}}{\sqrt{(-1)^2 + (2)^2 + (2)^2}}\right]$$

$$= -0.333F_D\mathbf{i} + 0.667F_D\mathbf{j} + 0.667F_D\mathbf{k}$$

$$\mathbf{W} = \{-981\mathbf{k}\}\text{ N}$$



(b)

Fig. 3–13

Equilibrium requires

$$\Sigma\mathbf{F} = \mathbf{0}; \qquad \mathbf{F}_B + \mathbf{F}_C + \mathbf{F}_D + \mathbf{W} = \mathbf{0}$$

$$F_B\mathbf{i} - 0.5F_C\mathbf{i} - 0.707F_C\mathbf{j} + 0.5F_C\mathbf{k}$$

$$-0.333F_D\mathbf{i} + 0.667F_D\mathbf{j} + 0.667F_D\mathbf{k} - 981\mathbf{k} = \mathbf{0}$$

Equating the respective $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components to zero,

$$\Sigma F_x = 0; \qquad F_B - 0.5F_C - 0.333F_D = 0 \qquad (1)$$

$$\Sigma F_y = 0; \qquad -0.707F_C + 0.667F_D = 0 \qquad (2)$$

$$\Sigma F_z = 0; \qquad 0.5F_C + 0.667F_D - 981 = 0 \qquad (3)$$

Solving Eq. (2) for F_D in terms of F_C and substituting this into Eq. (3) yields F_C. F_D is then determined from Eq. (2). Finally, substituting the results into Eq. (1) gives F_B. Hence,

$$F_C = 813\text{ N} \qquad \textit{Ans.}$$

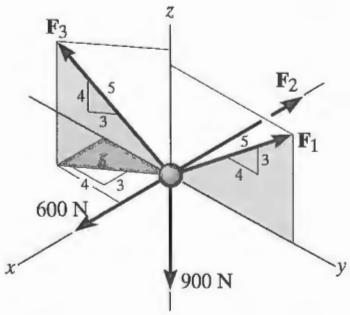
$$F_D = 862\text{ N} \qquad \textit{Ans.}$$

$$F_B = 694\text{ N} \qquad \textit{Ans.}$$

FUNDAMENTAL PROBLEMS

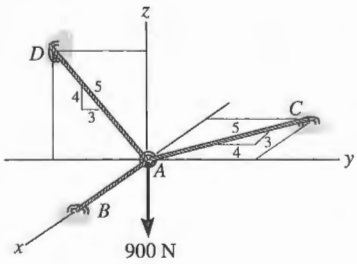
All problem solutions must include an FBD.

F3–7. Determine the magnitude of forces $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$, so that the particle is held in equilibrium.



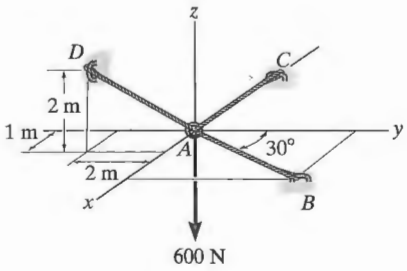
F3–7

F3–8. Determine the tension developed in cables AB, AC, and AD.



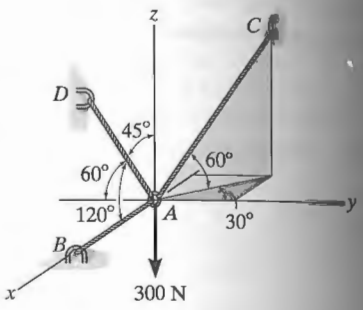
F3–8

F3–9. Determine the tension developed in cables AB, AC, and AD.



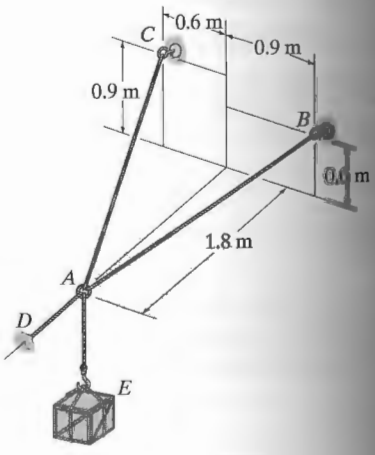
F3–9

F3–10. Determine the tension developed in cables AB, AC, and AD.



F3–10

F3–11. The 75-kg crate is supported by cables AB, AC and AD. Determine the tension in these wires.



F3–11

INEERING™
Customized coaching is available!

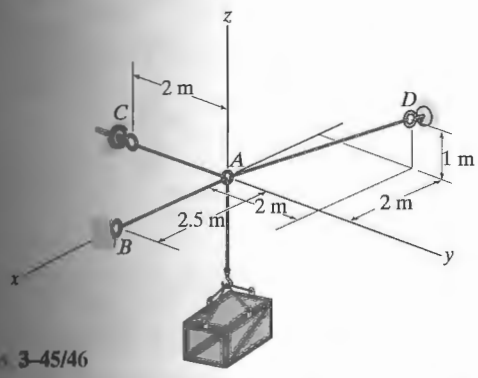
PROBLEMS

All problem solutions must include an FBD.

•3–45. Determine the tension in the cables in order to support the 100-kg crate in the equilibrium position shown.

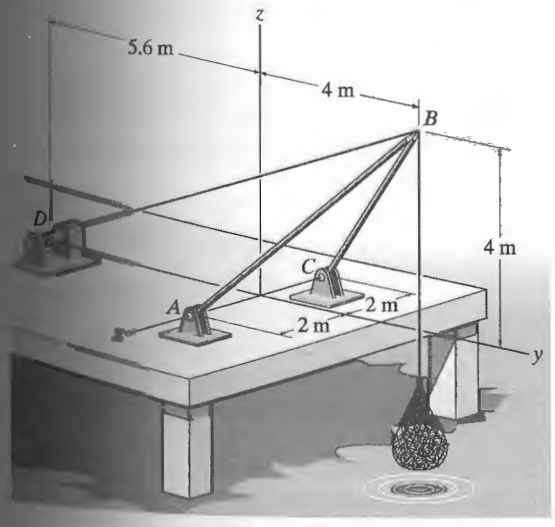
Video Solutions

3–46. Determine the maximum mass of the crate so that the tension developed in any cable does not exceeded 3 kN.



Probs. 3–45/46

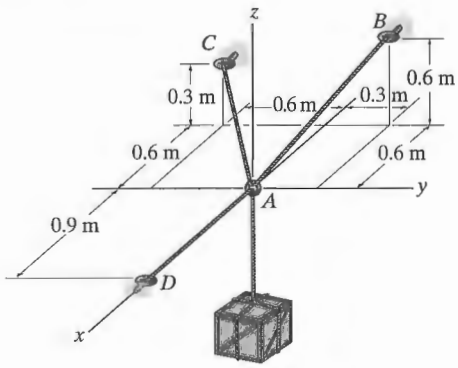
3–47. The shear leg derrick is used to haul the 200-kg net of fish onto the dock. Determine the compressive force along each of the legs AB and CB and the tension in the winch cable DB. Assume the force in each leg acts along its axis.



Prob. 3–47

***3–48.** Determine the tension developed in cables AB, AC, and AD required for equilibrium of the 150-kg crate.

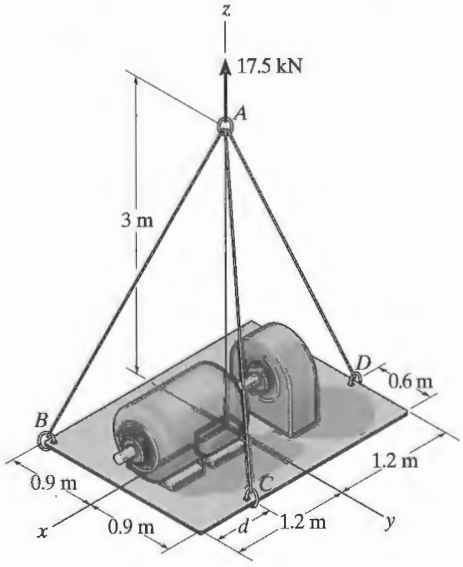
•3–49. Determine the maximum weight of the crate so that the tension developed in any cable does not exceed 2250 N.



Probs. 3–48/49

3–50. Determine the force in each cable needed to support the 17.5-kN (≈ 1750-kg) platform. Set $d = 0.6$ m.

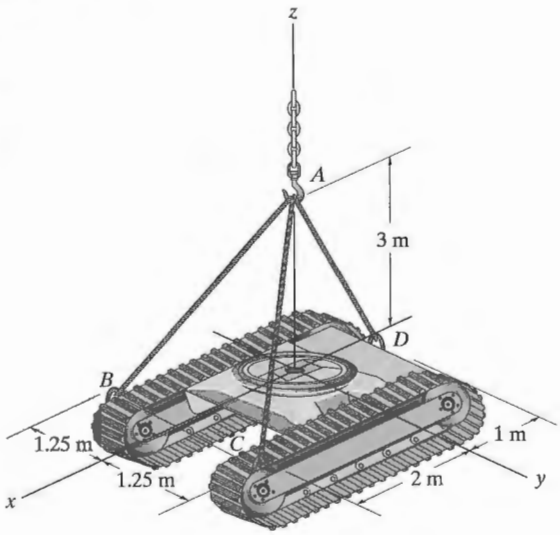
3–51. Determine the force in each cable needed to support the 17.5-kN (≈ 1750-kg) platform. Set $d = 1.2$ m.



Probs. 3–50/51

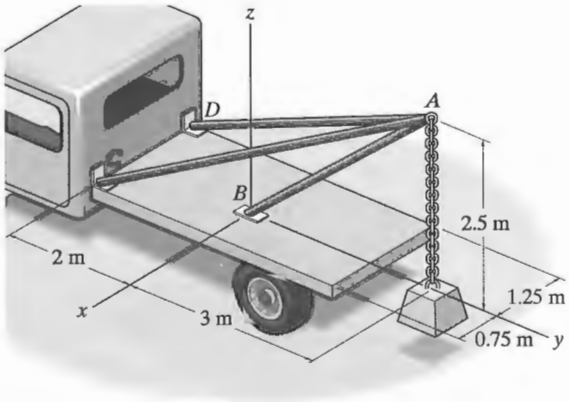
***3–52.** Determine the force in each of the three cables needed to lift the tractor which has a mass of 8 Mg.

Video Solutions



Prob. 3–52

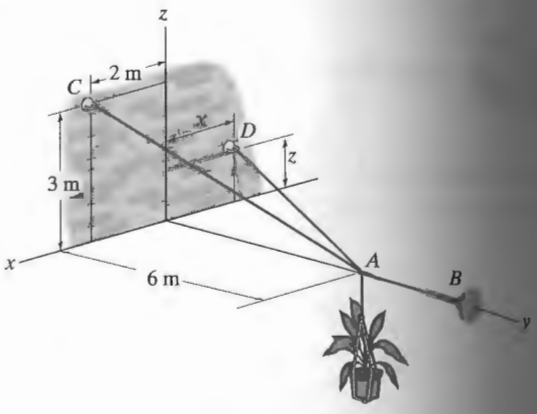
•3–53. Determine the force acting along the axis of each of the three struts needed to support the 500-kg block.



Prob. 3–53

3–54. If the mass of the flowerpot is 50 kg, determine the tension developed in each wire for equilibrium. Set $x = 1.5$ m and $z = 2$ m.

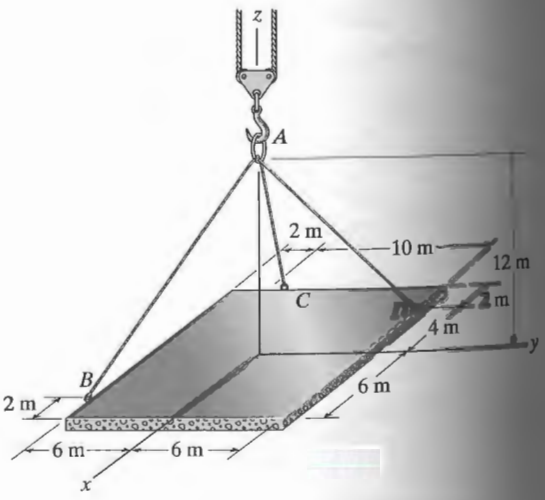
3–55. If the mass of the flowerpot is 50 kg, determine the tension developed in each wire for equilibrium. Set $x = 2$ m and $z = 1.5$ m.



Probs. 3–54/55

***3–56.** The ends of the three cables are attached to a ring at A and to the edge of a uniform 150-kg plate. Determine the tension in each of the cables for equilibrium.

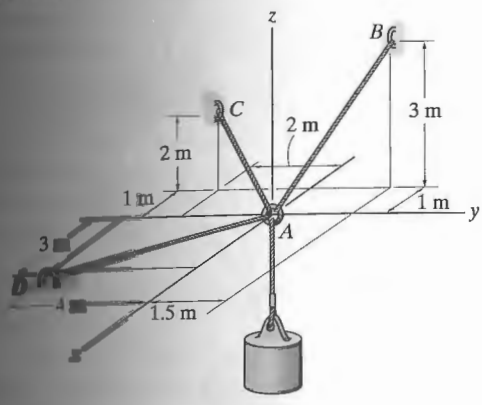
•3–57. The ends of the three cables are attached to a ring at A and to the edge of the uniform plate. Determine the largest mass the plate can have if each cable can support a maximum tension of 15 kN.



Probs. 3–56/57

3–58. Determine the tension developed in cables AB, AC, and AD required for equilibrium of the 75-kg cylinder.

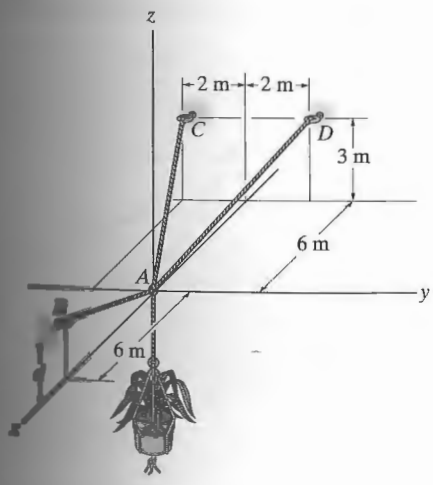
3–59. If each cable can withstand a maximum tension of 1000 N, determine the largest mass of the cylinder for equilibrium.



Probs. 3–58/59

3–60. The 50-kg pot is supported from A by the three cables. Determine the force acting in each cable for equilibrium. Take $d = 2.5$ m.

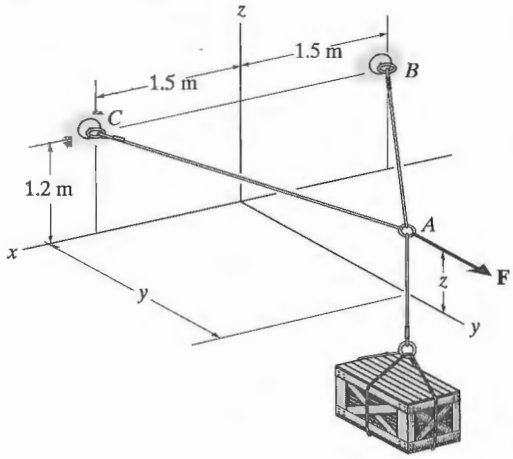
•3–61. Determine the height d of cable AB so that the force in cables AD and AC is one-half as great as the force in cable AB. What is the force in each cable for this case? The flower pot has a mass of 50 kg.



Probs. 3–60/61

3–62. A force of $F = 500$ N holds the 200-kg crate in equilibrium. Determine the coordinates $(0, y, z)$ of point A if the tension in cords AC and AB is 3500 N each.

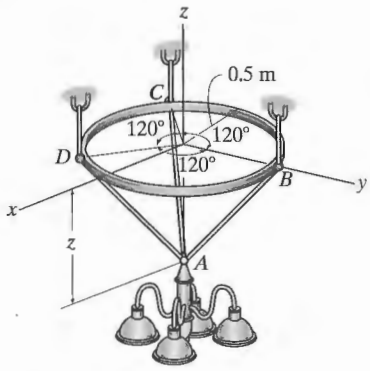
3–63. If the maximum allowable tension in cables AB and AC is 2500 N, determine the maximum height z to which the 100-kg crate can be lifted. What horizontal force F must be applied? Take $y = 2.4$ m.



Probs. 3–62/63

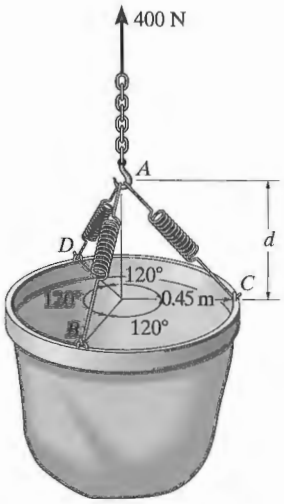
***3–64.** The thin ring can be adjusted vertically between three equally long cables from which the 100-kg chandelier is suspended. If the ring remains in the horizontal plane and $z = 600$ mm, determine the tension in each cable.

•3–65. The thin ring can be adjusted vertically between three equally long cables from which the 100-kg chandelier is suspended. If the ring remains in the horizontal plane and the tension in each cable is not allowed to exceed 1 kN, determine the smallest allowable distance z required for equilibrium.



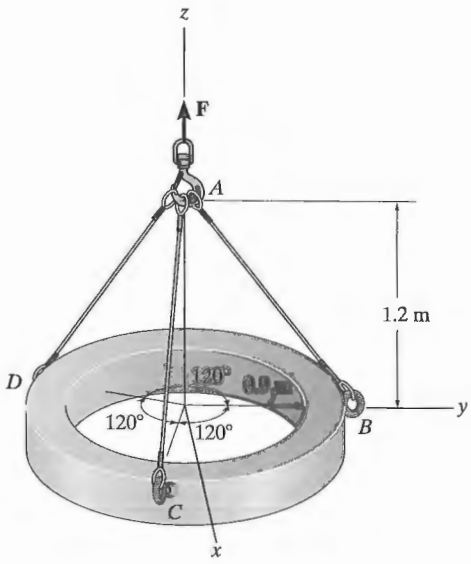
Probs. 3–64/65

3–66. The bucket has a weight of 400 N and is being hoisted using three springs, each having an unstretched length of $l_0 = 0.45$ m and stiffness of $k = 800$ N/m. Determine the vertical distance d from the rim to point A for equilibrium.



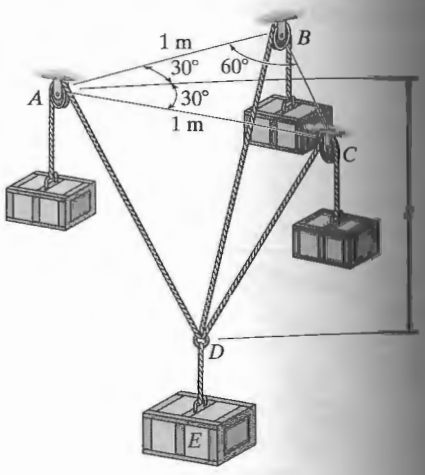
Prob. 3–66

3–67. Three cables are used to support a 450-kg ring. Determine the tension in each cable for equilibrium.



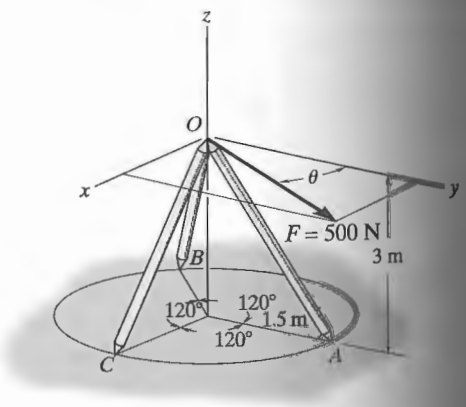
Prob. 3–67

***3–68.** The three outer blocks each have a mass of 2 kg and the central block E has a mass of 3 kg. Determine the sag s for equilibrium of the system.



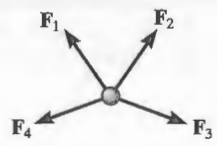
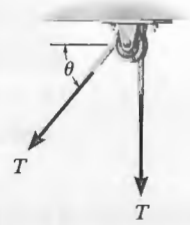
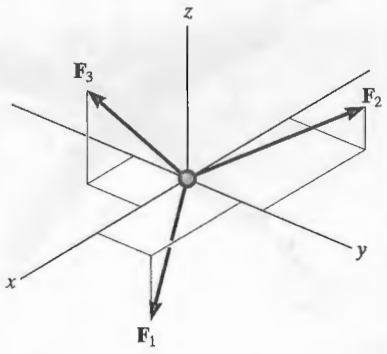
Prob. 3–68

•3–69. Determine the angle θ such that an equal force is developed in legs OB and OC. What is the force in each leg if the force is directed along the axis of each leg? The force $\mathbf{F}$ lies in the $x-y$ plane. The supports at A, B, C can exert forces in either direction along the attached legs.



Prob. 3–69

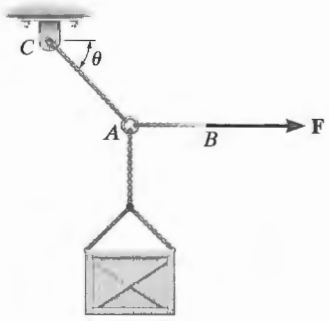
CHAPTER REVIEW

Particle Equilibrium When a particle is at rest or moves with constant velocity, it is in equilibrium. This requires that all the forces acting on the particle form a zero resultant force. In order to account for all the forces that act on a particle, it is necessary to draw its free-body diagram. This diagram is an outlined shape of the particle that shows all the forces listed with their known or unknown magnitudes and directions.	$\mathbf{F}_R = \Sigma\mathbf{F} = \mathbf{0}$	
Two Dimensions The two scalar equations of force equilibrium can be applied with reference to an established x, y coordinate system. The tensile force developed in a *continuous cable* that passes over a frictionless pulley must have a *constant* magnitude throughout the cable to keep the cable in equilibrium. If the problem involves a linearly elastic spring then the stretch or compression s of the spring can be related to the force applied to it.	$\Sigma F_x = 0$ $\Sigma F_y = 0$ $F = ks$	 Cable is in tension
Three Dimensions If the three-dimensional geometry is difficult to visualize, then the equilibrium equation should be applied using a Cartesian vector analysis. This requires first expressing each force on the free-body diagram as a Cartesian vector. When the forces are summed and set equal to zero, then the **i**, **j**, and **k** components are also zero.	$\Sigma\mathbf{F} = \mathbf{0}$ $\Sigma F_x = 0$ $\Sigma F_y = 0$ $\Sigma F_z = 0$	

MasteringENGINEERING
Customized coaching is available

REVIEW PROBLEMS

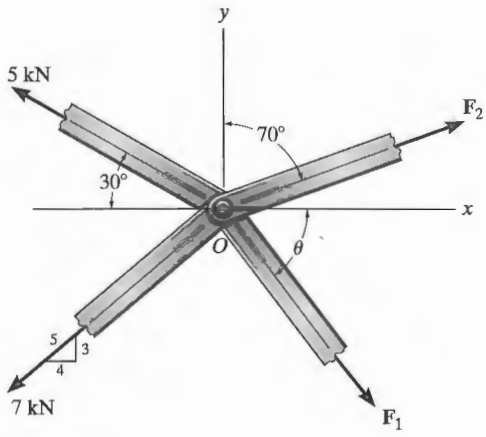
3–70. The 250-kg crate is hoisted using the ropes AB and AC. Each rope can withstand a maximum tension of 12.5 kN before it breaks. If AB always remains horizontal, determine the smallest angle θ to which the crate can be hoisted.



Prob. 3–70

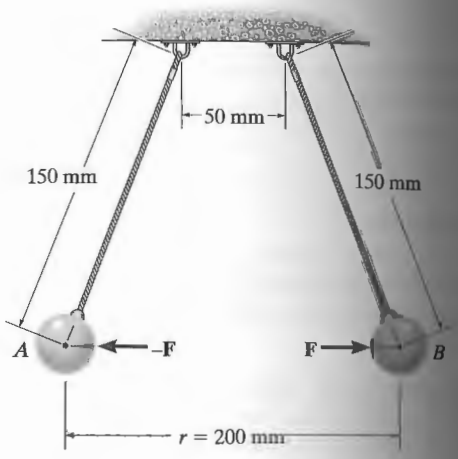
3–71. The members of a truss are pin connected at joint O. Determine the magnitude of $\mathbf{F}_1$ and its angle θ for equilibrium. Set $F_2 = 6$ kN.

***3–72.** The members of a truss are pin connected at joint O. Determine the magnitudes of $\mathbf{F}_1$ and $\mathbf{F}_2$ for equilibrium. Set $\theta = 60°$.



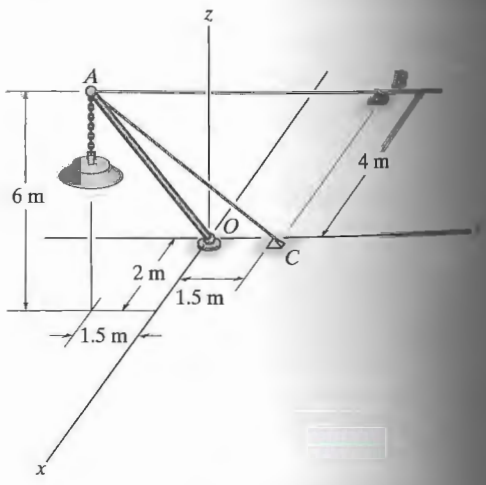
Prob. 3–71/72

•3–73. Two electrically charged pith balls, each having a mass of 0.15 g, are suspended from light threads of equal length. Determine the magnitude of the horizontal repulsive force, F, acting on each ball if the measured distance between them is $r = 200$ mm.



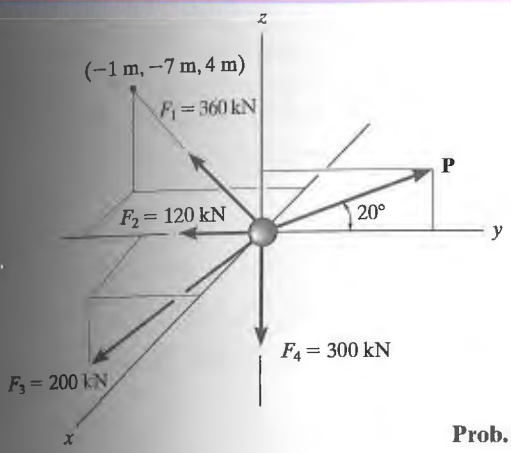
Prob. 3–73

3–74. The lamp has a mass of 15 kg and is supported by a pole AO and cables AB and AC. If the force in the pole acts along its axis, determine the forces in AO, AB, and AC for equilibrium.



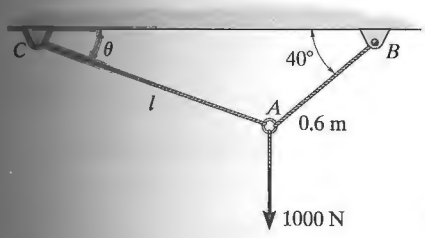
Prob. 3–74

3–75. Determine the magnitude of **P** and the coordinate direction angles of $\mathbf{F}_3$ required for equilibrium of the particle. Note that $\mathbf{F}_3$ acts in the octant shown.



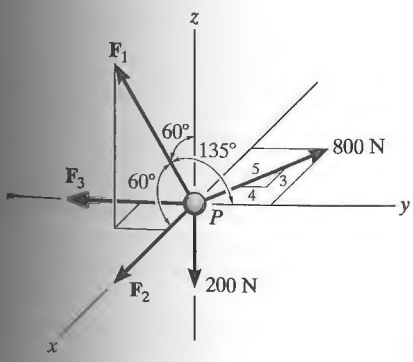
Prob. 3–75

*3–76. The ring of negligible size is subjected to a vertical force of 1000 N. Determine the longest length l of cord AC such that the tension acting in AC is 800 N. Also, what is the force acting in cord AB? *Hint:* Use the equilibrium condition to determine the required angle θ for attachment, then determine l using trigonometry applied to ΔABC.



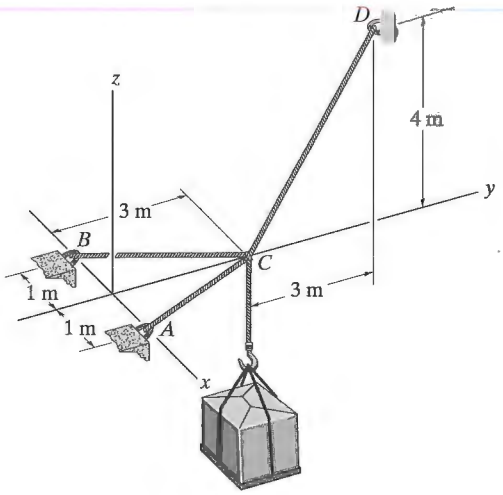
Prob. 3–76

*3–77. Determine the magnitudes of $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$ for equilibrium of the particle.



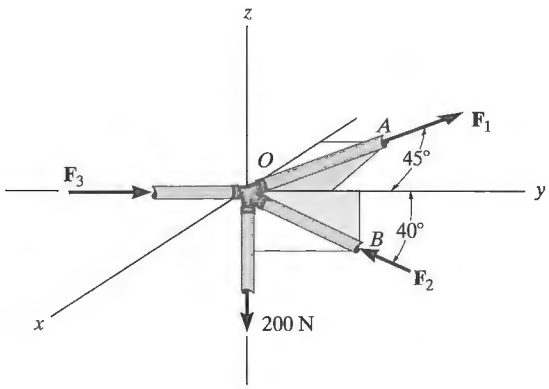
Prob. 3–77

3–78. Determine the force in each cable needed to support the 2.5-kN load.



Prob. 3–78

3–79. The joint of a space frame is subjected to four member forces. Member OA lies in the x–y plane and member OB lies in the y–z plane. Determine the forces acting in each of the members required for equilibrium of the joint.



Prob. 3–79

Application of forces to the handles of these wrenches will produce a tendency to rotate each wrench about its end. It is important to know how to calculate this effect and, in some cases, to be able to simplify this system to its resultants.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

Force System Resultants

4

CHAPTER OBJECTIVES

- To discuss the concept of the moment of a force and show how to calculate it in two and three dimensions.
- To provide a method for finding the moment of a force about a specified axis.
- To define the moment of a couple.
- To present methods for determining the resultants of nonconcurrent force systems.
- To indicate how to reduce a simple distributed loading to a resultant force having a specified location.

4.1 Moment of a Force—Scalar Formulation

When a force is applied to a body it will produce a tendency for the body to rotate about a point that is not on the line of action of the force. This tendency to rotate is sometimes called a *torque*, but most often it is called the moment of a force or simply the *moment*. For example, consider a wrench used to unscrew the bolt in Fig. 4–1*a*. If a force is applied to the handle of the wrench it will tend to turn the bolt about point O (or the z axis). The magnitude of the moment is directly proportional to the magnitude of $\mathbf{F}$ and the perpendicular distance or *moment arm* d. The larger the force or the longer the moment arm, the greater the moment or turning effect. Note that if the force $\mathbf{F}$ is applied at an angle $\theta \neq 90°$, Fig. 4–1*b*, then it will be more difficult to turn the bolt since the moment arm $d' = d \sin\theta$ will be smaller than d. If $\mathbf{F}$ is applied along the wrench, Fig. 4–1*c*, its moment arm will be zero since the line of action of $\mathbf{F}$ will intersect point O (the z axis). As a result, the moment of $\mathbf{F}$ about O is also zero and no turning can occur.

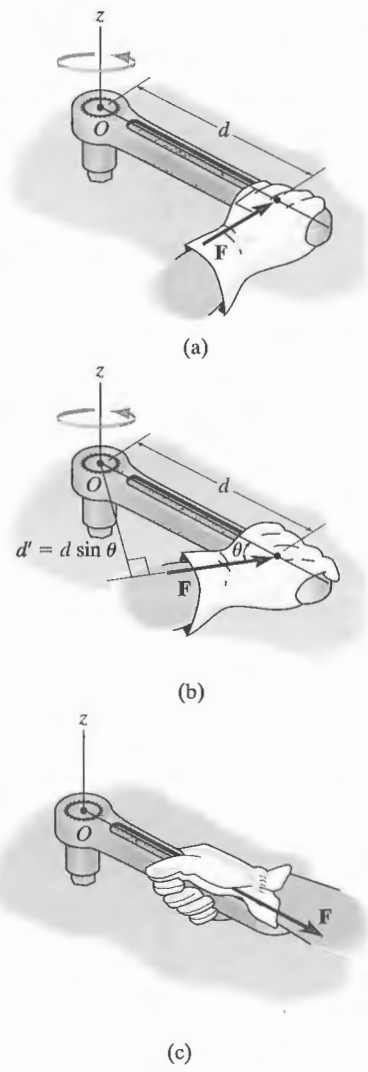


Fig. 4–1

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

Application of forces to the handles of these wrenches will produce a tendency to rotate each wrench about its end. It is important to know how to calculate this effect and, in some cases, to be able to simplify this system to its resultants.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

Force System Resultants

4

CHAPTER OBJECTIVES

- To discuss the concept of the moment of a force and show how to calculate it in two and three dimensions.
- To provide a method for finding the moment of a force about a specified axis.
- To define the moment of a couple.
- To present methods for determining the resultants of nonconcurrent force systems.
- To indicate how to reduce a simple distributed loading to a resultant force having a specified location.

4.1 Moment of a Force—Scalar Formulation

When a force is applied to a body it will produce a tendency for the body to rotate about a point that is not on the line of action of the force. This tendency to rotate is sometimes called a *torque*, but most often it is called the moment of a force or simply the *moment*. For example, consider a wrench used to unscrew the bolt in Fig. 4–1*a*. If a force is applied to the handle of the wrench it will tend to turn the bolt about point O (or the z axis). The magnitude of the moment is directly proportional to the magnitude of $\mathbf{F}$ and the perpendicular distance or *moment arm* d. The larger the force or the longer the moment arm, the greater the moment or turning effect. Note that if the force $\mathbf{F}$ is applied at an angle $\theta \neq 90°$, Fig. 4–1*b*, then it will be more difficult to turn the bolt since the moment arm $d' = d \sin\theta$ will be smaller than d. If $\mathbf{F}$ is applied along the wrench, Fig. 4–1*c*, its moment arm will be zero since the line of action of $\mathbf{F}$ will intersect point O (the z axis). As a result, the moment of $\mathbf{F}$ about O is also zero and no turning can occur.

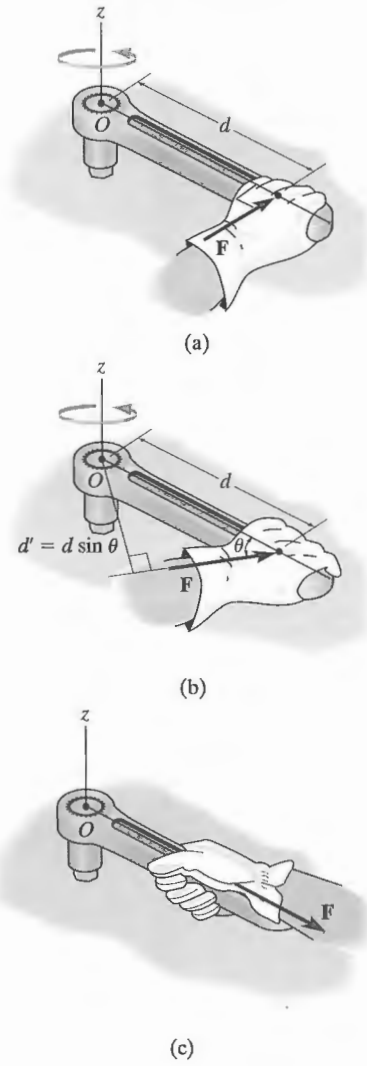


Fig. 4–1

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

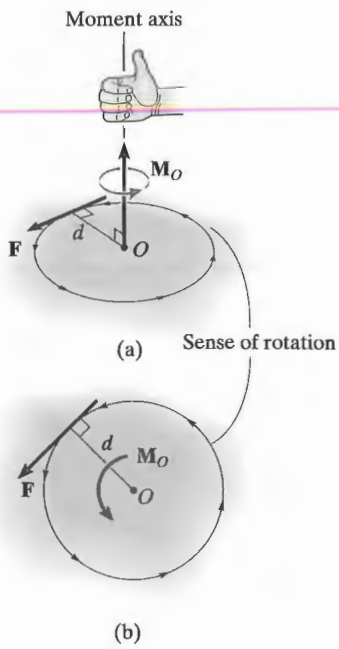


Fig. 4–2

We can generalize the above discussion and consider the force **F** and point O which lie in the shaded plane as shown in Fig. 4–2*a*. The moment $\mathbf{M}_O$ about point O, or about an axis passing through O and perpendicular to the plane, is a *vector quantity* since it has a specified magnitude and direction.

Magnitude. The magnitude of $\mathbf{M}_O$ is

$$\boxed{M_O = Fd} \tag{4–1}$$

where d is the *moment arm* or *perpendicular distance* from the axis at point O to the line of action of the force. Units of moment magnitude consist of force times distance, e.g., N · m or lb · ft.

Direction. The direction of $\mathbf{M}_O$ is defined by its *moment axis*, which is perpendicular to the plane that contains the force **F** and its moment arm d. The right-hand rule is used to establish the sense of direction of $\mathbf{M}_O$. According to this rule, the natural curl of the fingers of the right hand, as they are drawn towards the palm, represent the tendency for rotation caused by the moment. As this action is performed, the thumb of the right hand will give the directional sense of $\mathbf{M}_O$, Fig. 4–2*a*. Notice that the moment vector is represented three-dimensionally by a curl around an arrow. In two dimensions this vector is represented only by the curl as in Fig. 4–2*b*. Since in this case the moment will tend to cause a counterclockwise rotation, the moment vector is actually directed out of the page.

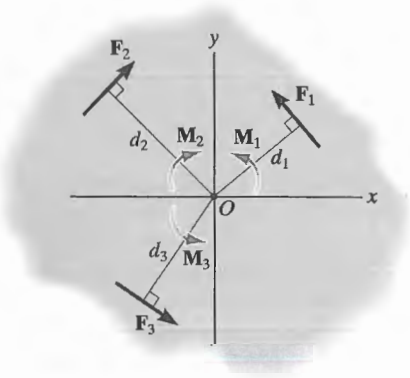


Fig. 4–3

Resultant Moment. For two-dimensional problems, where all the forces lie within the x–y plane, Fig. 4–3, the resultant moment $(\mathbf{M}_R)_O$ about point O (the z axis) can be determined by *finding the algebraic sum* of the moments caused by all the forces in the system. As a convention, we will generally consider *positive moments* as *counterclockwise* since they are directed along the positive z axis (out of the page). *Clockwise moments* will be *negative*. Doing this, the directional sense of each moment can be represented by a *plus or minus* sign. Using this sign convention, the resultant moment in Fig. 4–3 is therefore

$$\circlearrowleft + (M_R)_O = \Sigma Fd; \qquad (M_R)_O = F_1d_1 - F_2d_2 + F_3d_3$$

If the numerical result of this sum is a positive scalar, $(\mathbf{M}_R)_O$ will be a counterclockwise moment (out of the page); and if the result is negative, $(\mathbf{M}_R)_O$ will be a clockwise moment (into the page).

EXAMPLE 1

For each case illustrated in Fig. 4–4, determine the moment of the force about point O.

SOLUTION (SCALAR ANALYSIS)

The line of action of each force is extended as a dashed line in order to establish the moment arm d. Also illustrated is the tendency of rotation of the member as caused by the force. Furthermore, the orbit of the force about O is shown as a colored curl. Thus,

c/w = -ve

a/c/w = +ve.

Fig. 4–4a $M_O = (100\text{ N})(2\text{ m}) = 200\text{ N}\cdot\text{m}$ c/w *Ans.*

Fig. 4–4b $M_O = (50\text{ N})(0.75\text{ m}) = 37.5\text{ N}\cdot\text{m}$ c/w *Ans.*

Fig. 4–4c $M_O = (40\text{ kN})(4\text{ m} + 2\cos 30°\text{ m}) = 229\text{ kN}\cdot\text{m}$ c/w *Ans.*

Fig. 4–4d $M_O = (60\text{ kN})(1\sin 45°\text{ m}) = 42.4\text{ kN}\cdot\text{m}$ anti *Ans.*

Fig. 4–4e $M_O = (7\text{ kN})(4\text{ m} - 1\text{ m}) = 21.0\text{ kN}\cdot\text{m}$ a/c/w *Ans.*

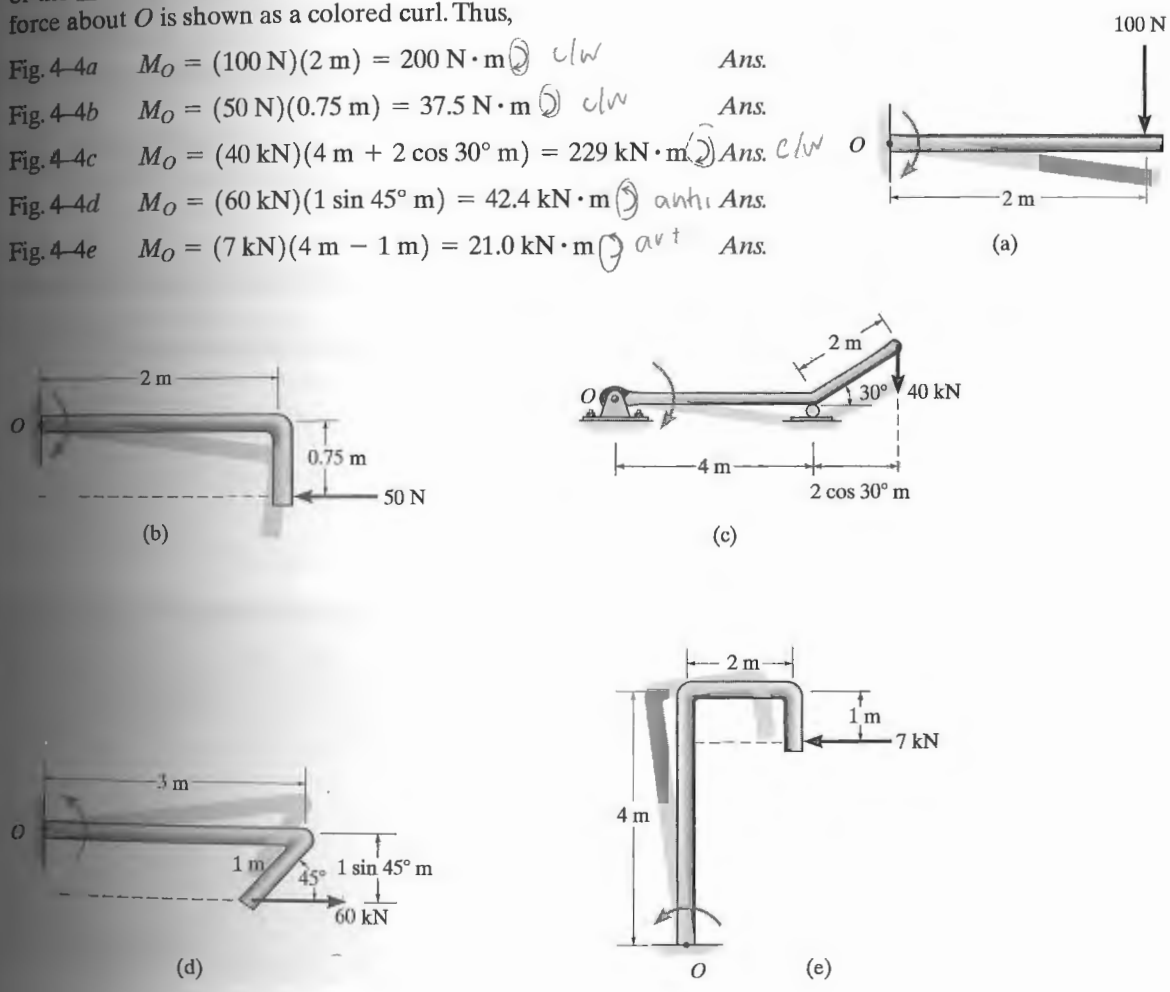


Fig. 4–4

EXAMPLE 4.2

$\vec{M} = -(50)(2) + 0 + (20)(3\sin 30) - (40)$

Determine the resultant moment of the four forces acting on the rod shown in Fig. 4–5 about point O.

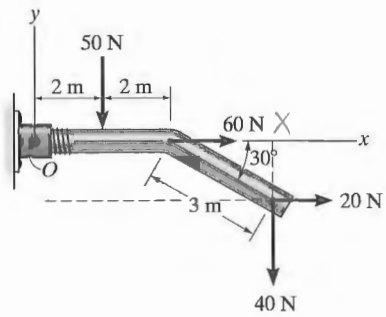


Fig. 4–5

SOLUTION

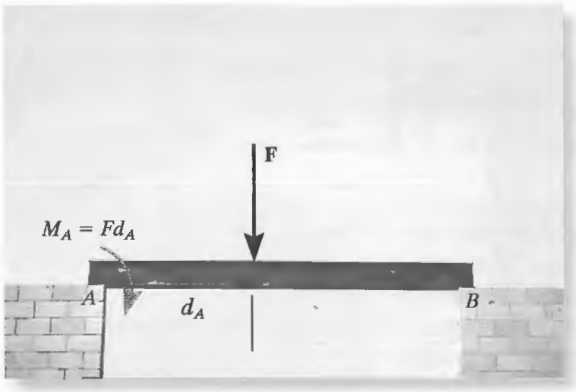
Assuming that positive moments act in the $+\mathbf{k}$ direction, i.e., counterclockwise, we have

$$\circlearrowleft + M_{R_O} = \Sigma Fd;$$

$$M_{R_O} = -50\text{ N}(2\text{ m}) + 60\text{ N}(0) + 20\text{ N}(3\sin 30^\circ\text{ m}) - 40\text{ N}(4\text{ m} + 3\cos 30^\circ\text{ m})$$

$$M_{R_O} = -334\text{ N}\cdot\text{m} = 334\text{ N}\cdot\text{m}\circlearrowright \qquad \textit{Ans.}$$

For this calculation, note how the moment-arm distances for the 20-N and 40-N forces are established from the extended (dashed) lines of action of each of these forces.



As illustrated by the example problems, the moment of a force does not always cause a rotation. For example, the force $\mathbf{F}$ tends to rotate the beam clockwise about its support at A with a moment $M_A = Fd_A$. The actual rotation would occur if the support at B were removed.

The ability to remove the nail will require the moment of $\mathbf{F}_H$ about point O to be larger than the moment of the force $\mathbf{F}_N$ about O that is needed to pull the nail out.

4.2 Cross Product

The moment of a force will be formulated using Cartesian vectors in the next section. Before doing this, however, it is first necessary to expand our knowledge of vector algebra and introduce the cross-product method of vector multiplication.

The *cross product* of two vectors **A** and **B** yields the vector **C**, which is written

$$\mathbf{C} = \mathbf{A} \times \mathbf{B} \tag{4–2}$$

and is read "**C** equals **A** cross **B**."

Magnitude. The *magnitude* of **C** is defined as the product of the magnitudes of **A** and **B** and the sine of the angle θ between their tails ($0° \leq \theta \leq 180°$). Thus, $C = AB \sin\theta$.

Direction. Vector **C** has a *direction* that is perpendicular to the plane containing **A** and **B** such that **C** is specified by the right-hand rule; i.e., curling the fingers of the right hand from vector **A** (cross) to vector **B**, the thumb points in the direction of **C**, as shown in Fig. 4–6.

Knowing both the magnitude and direction of **C**, we can write

$$\mathbf{C} = \mathbf{A} \times \mathbf{B} = (AB \sin\theta)\mathbf{u}_C \tag{4–3}$$

where the scalar $AB \sin\theta$ defines the *magnitude* of **C** and the unit vector $\mathbf{u}_C$ defines the *direction* of **C**. The terms of Eq. 4–3 are illustrated graphically in Fig. 4–6.

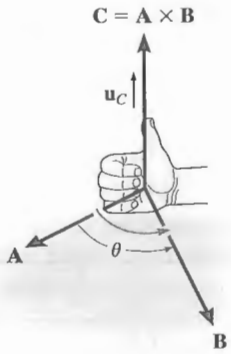


Fig. 4–6

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

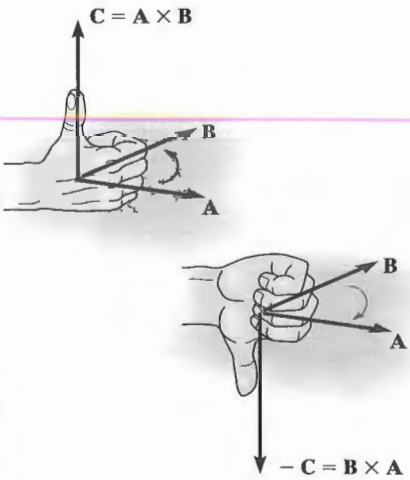


Fig. 4–7

Laws of Operation.

- The commutative law is *not* valid; i.e., $\mathbf{A} \times \mathbf{B} \neq \mathbf{B} \times \mathbf{A}$. Rather,

$$\mathbf{A} \times \mathbf{B} = -\mathbf{B} \times \mathbf{A}$$

This is shown in Fig. 4–7 by using the right-hand rule. The cross product $\mathbf{B} \times \mathbf{A}$ yields a vector that has the same magnitude but acts in the opposite direction to $\mathbf{C}$; i.e., $\mathbf{B} \times \mathbf{A} = -\mathbf{C}$.

- If the cross product is multiplied by a scalar a, it obeys the associative law;

$$a(\mathbf{A} \times \mathbf{B}) = (a\mathbf{A}) \times \mathbf{B} = \mathbf{A} \times (a\mathbf{B}) = (\mathbf{A} \times \mathbf{B})a$$

This property is easily shown since the magnitude of the resultant vector ($|a|AB \sin\theta$) and its direction are the same in each case.

- The vector cross product also obeys the distributive law of addition.

$$\mathbf{A} \times (\mathbf{B} + \mathbf{D}) = (\mathbf{A} \times \mathbf{B}) + (\mathbf{A} \times \mathbf{D})$$

- The proof of this identity is left as an exercise (see Prob. 4–1). It is important to note that *proper order* of the cross product must be maintained, since they are not commutative.

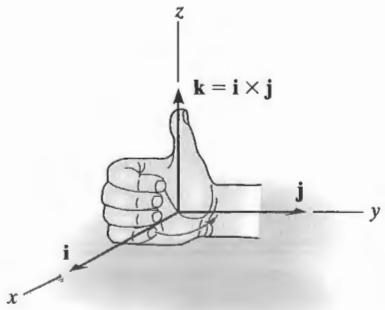


Fig. 4–8

Cartesian Vector Formulation. Equation 4–3 may be used to find the cross product of any pair of Cartesian unit vectors. For example, to find $\mathbf{i} \times \mathbf{j}$, the magnitude of the resultant vector is $(i)(j)(\sin 90°) = (1)(1)(1) = 1$, and its direction is determined using the right-hand rule. As shown in Fig. 4–8, the resultant vector points in the $+\mathbf{k}$ direction. Thus, $\mathbf{i} \times \mathbf{j} = (1)\mathbf{k}$. In a similar manner,

$$\begin{array}{lll} \mathbf{i} \times \mathbf{j} = \mathbf{k} & \mathbf{i} \times \mathbf{k} = -\mathbf{j} & \mathbf{i} \times \mathbf{i} = \mathbf{0} \\ \mathbf{j} \times \mathbf{k} = \mathbf{i} & \mathbf{j} \times \mathbf{i} = -\mathbf{k} & \mathbf{j} \times \mathbf{j} = \mathbf{0} \\ \mathbf{k} \times \mathbf{i} = \mathbf{j} & \mathbf{k} \times \mathbf{j} = -\mathbf{i} & \mathbf{k} \times \mathbf{k} = \mathbf{0} \end{array}$$

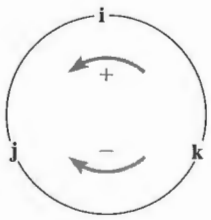


Fig. 4–9

These results should *not* be memorized; rather, it should be clearly understood how each is obtained by using the right-hand rule and the definition of the cross product. A simple scheme shown in Fig. 4–9 is helpful for obtaining the same results when the need arises. If the circle is constructed as shown, then "crossing" two unit vectors in a *counterclockwise* fashion around the circle yields the *positive* third unit vector; e.g., $\mathbf{k} \times \mathbf{i} = \mathbf{j}$. "Crossing" *clockwise*, a *negative* unit vector is obtained; e.g., $\mathbf{i} \times \mathbf{k} = -\mathbf{j}$.

Exclusively for The University of Melbourne

Let us now consider the cross product of two general vectors **A** and **B** which are expressed in Cartesian vector form. We have

$$\begin{aligned}\mathbf{A} \times \mathbf{B} &= (A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}) \times (B_x\mathbf{i} + B_y\mathbf{j} + B_z\mathbf{k}) \\ &= A_xB_x(\mathbf{i} \times \mathbf{i}) + A_xB_y(\mathbf{i} \times \mathbf{j}) + A_xB_z(\mathbf{i} \times \mathbf{k}) \\ &+ A_yB_x(\mathbf{j} \times \mathbf{i}) + A_yB_y(\mathbf{j} \times \mathbf{j}) + A_yB_z(\mathbf{j} \times \mathbf{k}) \\ &+ A_zB_x(\mathbf{k} \times \mathbf{i}) + A_zB_y(\mathbf{k} \times \mathbf{j}) + A_zB_z(\mathbf{k} \times \mathbf{k})\end{aligned}$$

Carrying out the cross-product operations and combining terms yields

$$\mathbf{A} \times \mathbf{B} = (A_yB_z - A_zB_y)\mathbf{i} - (A_xB_z - A_zB_x)\mathbf{j} + (A_xB_y - A_yB_x)\mathbf{k} \quad (4\text{–}4)$$

This equation may also be written in a more compact determinant form as

$$\boxed{\mathbf{A} \times \mathbf{B} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix}} \quad (4\text{–}5)$$

Thus, to find the cross product of any two Cartesian vectors **A** and **B**, it is necessary to expand a determinant whose first row of elements consists of the unit vectors **i**, **j**, and **k** and whose second and third rows represent the x, y, z components of the two vectors **A** and **B**, respectively.*

*A determinant having three rows and three columns can be expanded using three minors, each of which is multiplied by one of the three terms in the first row. There are four elements in each minor, for example,

$$\begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix}$$

By *definition*, this determinant notation represents the terms $(A_{11}A_{22} - A_{12}A_{21})$, which is simply the product of the two elements intersected by the arrow slanting downward to the right $(A_{11}A_{22})$ *minus* the product of the two elements intersected by the arrow slanting downward to the left $(A_{12}A_{21})$. For a 3×3 determinant, such as Eq. 4–5, the three minors can be generated in accordance with the following scheme:

For element **i**: $\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} = \mathbf{i}(A_yB_z - A_zB_y)$

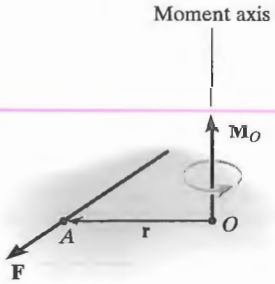
Remember the negative sign

For element **j**: $\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} = -\mathbf{j}(A_xB_z - A_zB_x)$

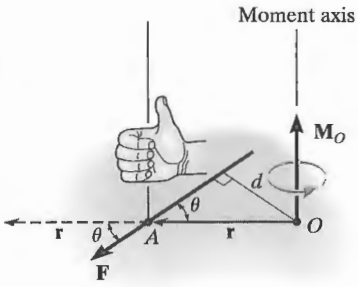
For element **k**: $\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} = \mathbf{k}(A_xB_y - A_yB_x)$

Adding the results and noting that the **j** element *must include the minus sign* yields the expanded form of $\mathbf{A} \times \mathbf{B}$ given by Eq. 4–4.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.



(a)



(b)

Fig. 4–10

4.3 Moment of a Force—Vector Formulation

The moment of a force **F** about point O, or actually about the moment axis passing through O and perpendicular to the plane containing O and **F**, Fig. 4–10*a*, can be expressed using the vector cross product, namely,

$$\mathbf{M}_O = \mathbf{r} \times \mathbf{F} \tag{4–6}$$

Here **r** represents a position vector directed *from O to any point* on the line of action of **F**. We will now show that indeed the moment $\mathbf{M}_O$, when determined by this cross product, has the proper magnitude and direction.

Magnitude. The magnitude of the cross product is defined from Eq. 4–3 as $M_O = rF \sin\theta$, where the angle θ is measured between the *tails* of **r** and **F**. To establish this angle, **r** must be treated as a sliding vector so that θ can be constructed properly, Fig. 4–10*b*. Since the moment arm $d = r \sin\theta$, then

$$M_O = rF \sin\theta = F(r \sin\theta) = Fd$$

which agrees with Eq. 4–1.

Direction. The direction and sense of $\mathbf{M}_O$ in Eq. 4–6 are determined by the right-hand rule as it applies to the cross product. Thus, sliding **r** to the dashed position and curling the right-hand fingers from **r** toward **F**, "**r** cross **F**," the thumb is directed upward or perpendicular to the plane containing **r** and **F** and this is in the *same direction* as $\mathbf{M}_O$, the moment of the force about point O, Fig. 4–10*b*. Note that the "curl" of the fingers, like the curl around the moment vector, indicates the sense of rotation caused by the force. Since the cross product does not obey the commutative law, the order of $\mathbf{r} \times \mathbf{F}$ must be maintained to produce the correct sense of direction for $\mathbf{M}_O$.

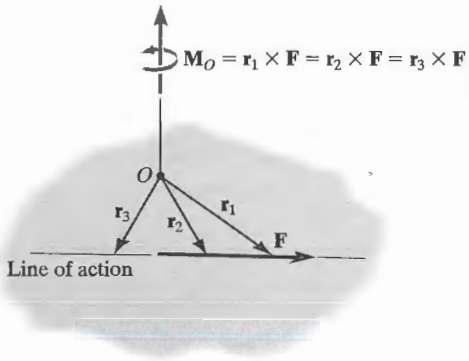


Fig. 4–11

Principle of Transmissibility. The cross product operation is often used in three dimensions since the perpendicular distance or moment arm from point O to the line of action of the force is not needed. In other words, we can use any position vector **r** measured from point O to any point on the line of action of the force **F**, Fig. 4–11. Thus,

$$\mathbf{M}_O = \mathbf{r}_1 \times \mathbf{F} = \mathbf{r}_2 \times \mathbf{F} = \mathbf{r}_3 \times \mathbf{F}$$

Since **F** can be applied at any point along its line of action and still create this *same moment* about point O, then **F** can be considered a *sliding vector*. This property is called the *principle of transmissibility* of a force.

DESIGN 2
ing exclusively for The University of Melbourne

Cartesian Vector Formulation. If we establish x, y, z coordinate axes, then the position vector $\mathbf{r}$ and force $\mathbf{F}$ can be expressed as Cartesian vectors, Fig. 4–12*a*. Applying Eq. 4–5 we have

$$\mathbf{M}_O = \mathbf{r} \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ r_x & r_y & r_z \\ F_x & F_y & F_z \end{vmatrix} \tag{4–7}$$

where

r_x, r_y, r_z represent the x, y, z components of the position vector drawn from point O to *any point* on the line of action of the force

F_x, F_y, F_z represent the x, y, z components of the force vector

If the determinant is expanded, then like Eq. 4–4 we have

$$\mathbf{M}_O = (r_yF_z - r_zF_y)\mathbf{i} - (r_xF_z - r_zF_x)\mathbf{j} + (r_xF_y - r_yF_x)\mathbf{k} \tag{4–8}$$

The physical meaning of these three moment components becomes evident by studying Fig. 4–12*b*. For example, the $\mathbf{i}$ component of $\mathbf{M}_O$ can be determined from the moments of $\mathbf{F}_x$, $\mathbf{F}_y$, and $\mathbf{F}_z$ about the x axis. The component $\mathbf{F}_x$ does *not* create a moment or tendency to cause turning about the x axis since this force is *parallel* to the x axis. The line of action of $\mathbf{F}_y$ passes through point B, and so the magnitude of the moment of $\mathbf{F}_y$ about point A on the x axis is r_zF_y. By the right-hand rule this component acts in the *negative* $\mathbf{i}$ direction. Likewise, $\mathbf{F}_z$ passes through point C and so it contributes a moment component of $r_yF_z\mathbf{i}$ about the axis. Thus, $(M_O)_x = (r_yF_z - r_zF_y)$ as shown in Eq. 4–8. As an exercise, establish the $\mathbf{j}$ and $\mathbf{k}$ components of $\mathbf{M}_O$ in this manner and show that indeed the expanded form of the determinant, Eq. 4–8, represents the moment of $\mathbf{F}$ about point O. Once $\mathbf{M}_O$ is determined, realize that it will always be *perpendicular* to the shaded plane containing vectors $\mathbf{r}$ and $\mathbf{F}$, Fig. 4–12*a*.

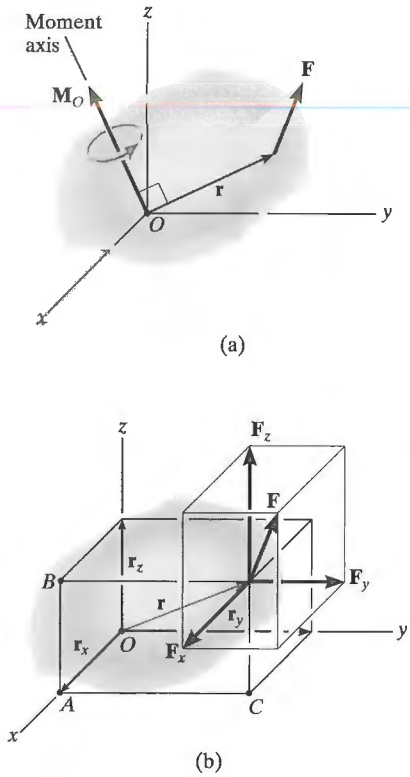


Fig. 4–12

Resultant Moment of a System of Forces. If a body is acted upon by a system of forces, Fig. 4–13, the resultant moment of the forces about point O can be determined by vector addition of the moment of each force. This resultant can be written symbolically as

$$\mathbf{M}_{R_O} = \Sigma(\mathbf{r} \times \mathbf{F}) \tag{4–9}$$

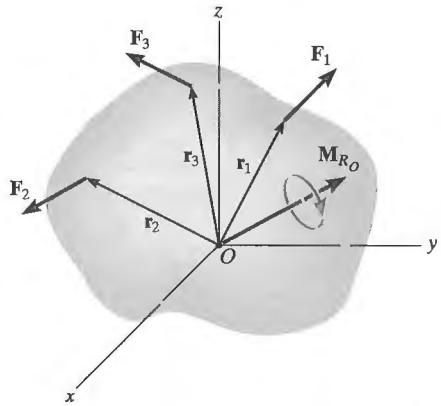
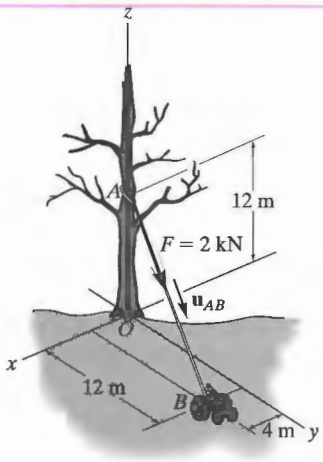


Fig. 4–13

EXAMPLE 4.3

Determine the moment produced by the force **F** in Fig. 4–14*a* about point *O*. Express the result as a Cartesian vector.



(a)

SOLUTION

As shown in Fig. 4–14*a*, either $\mathbf{r}_A$ or $\mathbf{r}_B$ can be used to determine the moment about point *O*. These position vectors are

$$\mathbf{r}_A = \{12\mathbf{k}\}\text{ m} \quad \text{and} \quad \mathbf{r}_B = \{4\mathbf{i} + 12\mathbf{j}\}\text{ m}$$

Force **F** expressed as a Cartesian vector is

$$\mathbf{F} = F\mathbf{u}_{AB} = 2\text{ kN}\left[\frac{\{4\mathbf{i} + 12\mathbf{j} - 12\mathbf{k}\}\text{ m}}{\sqrt{(4\text{ m})^2 + (12\text{ m})^2 + (-[redacted]}}\right]$$

$$= \{0.4588\mathbf{i} + 1.376\mathbf{j} - 1.376\mathbf{k}\}\text{ kN}$$

Thus

$$\mathbf{M}_O = \mathbf{r}_A \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & 12 \\ 0.4588 & 1.376 & -1.376 \end{vmatrix}$$

$$= [0(-1.376) - 12(1.376)]\mathbf{i} - [0(-1.376) - 12(0.458[redacted])]\mathbf{j} + [0(1.376) - 0(0.4588)]\mathbf{k}$$

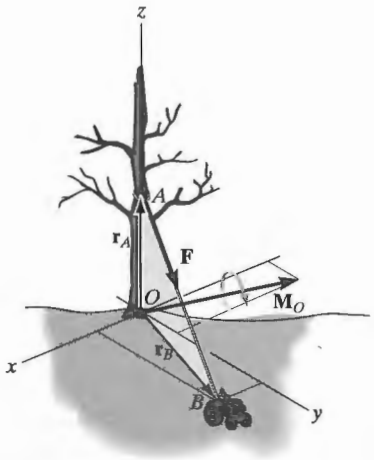
$$= \{-16.5\mathbf{i} + 5.51\mathbf{j}\}\text{ kN}\cdot\text{m} \qquad An$$

or

$$\mathbf{M}_O = \mathbf{r}_B \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 4 & 12 & 0 \\ 0.4588 & 1.376 & -1.376 \end{vmatrix}$$

$$= [12(-1.376) - 0(1.376)]\mathbf{i} - [4(-1.376) - 0(0.458[redacted])]\mathbf{j} + [4(1.376) - 12(0.4588)]\mathbf{k}$$

$$= \{-16.5\mathbf{i} + 5.51\mathbf{j}\}\text{ kN}\cdot\text{m} \qquad An$$



(b)

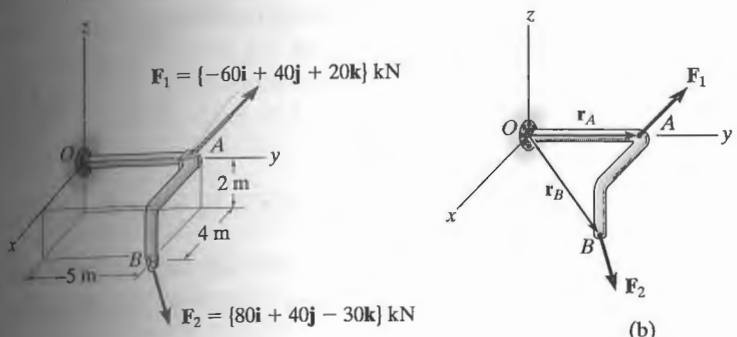
Fig. 4–14

NOTE: As shown in Fig. 4–14*b*, $\mathbf{M}_O$ acts perpendicular to the plane that contains **F**, $\mathbf{r}_A$, and $\mathbf{r}_B$. Had this problem been worked using $M_O = Fd$, notice the difficulty that would arise in obtaining the moment arm *d*.

Exclusively for The University of Melbourne

EXAMPLE 4.4

Two forces act on the rod shown in Fig. 4–15*a*. Determine the resultant moment they create about the flange at O. Express the result as a Cartesian vector.



SOLUTION

Position vectors are directed from point O to each force as shown in Fig. 4–15*b*. These vectors are

$$\mathbf{r}_A = \{5\mathbf{j}\}\text{ m}$$

$$\mathbf{r}_B = \{4\mathbf{i} + 5\mathbf{j} - 2\mathbf{k}\}\text{ m}$$

The resultant moment about O is therefore

$$\mathbf{M}_{R_O} = \Sigma(\mathbf{r} \times \mathbf{F})$$

$$= \mathbf{r}_A \times \mathbf{F}_1 + \mathbf{r}_B \times \mathbf{F}_3$$

$$= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 5 & 0 \\ -60 & 40 & 20 \end{vmatrix} + \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 4 & 5 & -2 \\ 80 & 40 & -30 \end{vmatrix}$$

$$= [5(20) - 0(40)]\mathbf{i} - [0]\mathbf{j} + [0(40) - (5)(-60)]\mathbf{k}$$

$$+ [5(-30) - (-2)(40)]\mathbf{i} - [4(-30) - (-2)(80)]\mathbf{j} + [4(40) - 5(80)]\mathbf{k}$$

$$= \{30\mathbf{i} - 40\mathbf{j} + 60\mathbf{k}\}\text{ kN}\cdot\text{m} \qquad \textit{Ans.}$$

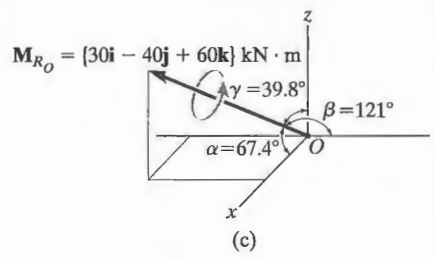


Fig. 4–15

NOTE: This result is shown in Fig. 4–15*c*. The coordinate direction angles were determined from the unit vector for $\mathbf{M}_{R_O}$. Realize that the two forces tend to cause the rod to rotate about the moment axis in the manner shown by the curl indicated on the moment vector.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

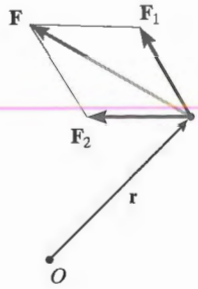


Fig 4–16

4.4 Principle of Moments

A concept often used in mechanics is the *principle of moments*, which is sometimes referred to as *Varignon's theorem* since it was originally developed by the French mathematician Varignon (1654–1722). It states that *the moment of a force about a point is equal to the sum of the moments of the components of the force about the point.* This theorem can be proven easily using the vector cross product since the cross product obeys the *distributive law*. For example, consider the moments of the force **F** and two of its components about point O. Fig. 4–16. Since $\mathbf{F} = \mathbf{F}_1 + \mathbf{F}_2$ we have

$$\mathbf{M}_O = \mathbf{r} \times \mathbf{F} = \mathbf{r} \times (\mathbf{F}_1 + \mathbf{F}_2) = \mathbf{r} \times \mathbf{F}_1 + \mathbf{r} \times \mathbf{F}_2$$

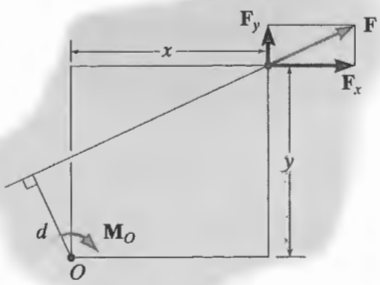


Fig. 4–17

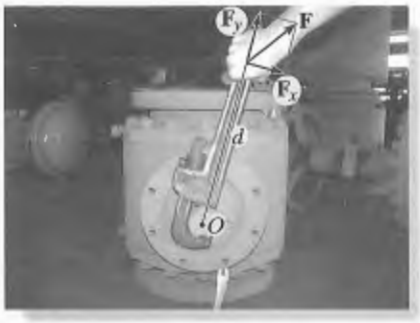
For two-dimensional problems, Fig. 4–17, we can use the principle of moments by resolving the force into its rectangular components and then determine the moment using a scalar analysis. Thus,

$$M_O = F_x y - F_y x$$

This method is generally easier than finding the same moment using $M_O = Fd$.

Important Points

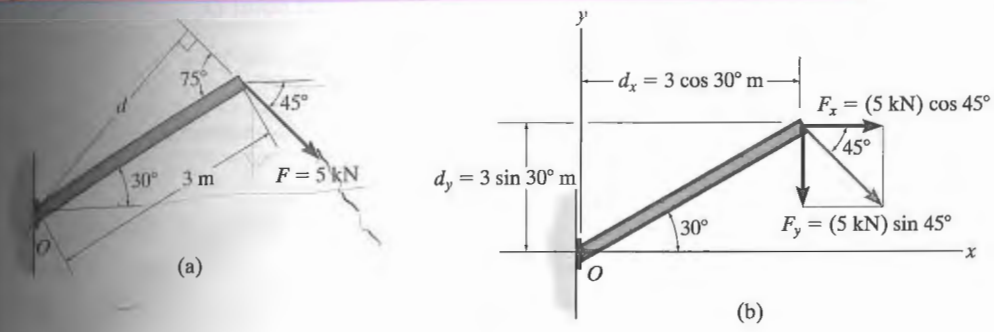
- The moment of a force creates the tendency of a body to turn about an axis passing through a specific point O.
- Using the right-hand rule, the sense of rotation is indicated by the curl of the fingers, and the thumb is directed along the moment axis, or line of action of the moment.
- The magnitude of the moment is determined from $M_O = Fd$ where d is called the moment arm, which represents the perpendicular or shortest distance from point O to the line of action of the force.
- In three dimensions the vector cross product is used to determine the moment, i.e., $\mathbf{M}_O = \mathbf{r} \times \mathbf{F}$. Remember that **r** is directed *from* point O *to any point* on the line of action of **F**.
- The principle of moments states that the moment of a force about a point is equal to the sum of the moments of the force's components about the point. This is a very convenient method to use in two dimensions.



The moment of the applied force **F** about point O is easy to determine if we use the principle of moments. It is simply $M_O = F_x d$.

EXAMPLE 4.5

Determine the moment of the force in Fig. 4–18*a* about point *O*.



SOLUTION I

The moment arm d in Fig. 4–18*a* can be found from trigonometry.

$$d = (3\text{ m})\sin 75° = 2.898\text{ m}$$

Thus,

$$M_O = Fd = (5\text{kN})(2.898\text{ m}) = 14.5\text{ kN}\cdot\text{m}\circlearrowleft \qquad \textit{Ans.}$$

Since the force tends to rotate or orbit clockwise about point O, the moment is directed into the page.

SOLUTION II

The x and y components of the force are indicated in Fig. 4–18*b*. Considering counterclockwise moments as positive, and applying the principle of moments, we have

$$\begin{aligned}\circlearrowleft + M_O &= -F_x d_y - F_y d_x \\ &= -(5\cos 45°\text{ kN})(3\sin 30°\text{ m}) - (5\sin 45°\text{ kN})(3\cos 30°\text{ m}) \\ &= -14.5\text{ kN}\cdot\text{m} = 14.5\text{ kN}\cdot\text{m}\circlearrowleft \qquad \textit{Ans.}\end{aligned}$$

SOLUTION III

The x and y axes can be set parallel and perpendicular to the rod's axis as shown in Fig. 4-18*c*. Here $\mathbf{F}_x$ produces no moment about point O since its line of action passes through this point. Therefore,

$$\begin{aligned}\circlearrowleft + M_O &= -F_y d_x \\ &= -(5\sin 75°\text{ kN})(3\text{ m}) \\ &= -14.5\text{ kN}\cdot\text{m} = 14.5\text{ kN}\cdot\text{m}\circlearrowleft \qquad \textit{Ans.}\end{aligned}$$

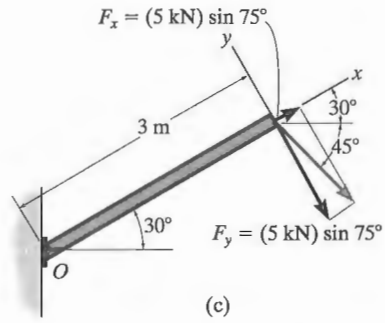
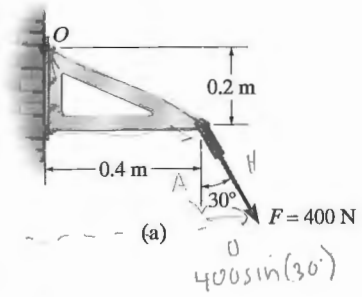


Fig. 4–18

EXAMPLE 4.6

Force **F** acts at the end of the angle bracket shown in Fig. 4–19*a*. Determine the moment of the force about point *O*.



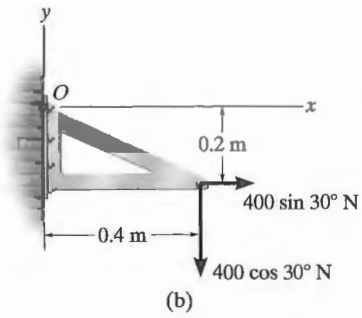
SOLUTION I *(SCALAR ANALYSIS)*

The force is resolved into its x and y components as shown in Fig. 4–19*b*, then

$$\circlearrowleft + M_O = 400 \sin 30° \text{ N}(0.2 \text{ m}) - 400 \cos 30° \text{ N}(0.4 \text{ m})$$
$$= -98.6 \text{ N} \cdot \text{m} = 98.6 \text{ N} \cdot \text{m} \circlearrowright$$

or

$$\mathbf{M}_O = \{-98.6\mathbf{k}\} \text{ N} \cdot \text{m} \qquad \textit{Ans.}$$



SOLUTION II *(VECTOR ANALYSIS)*

Using a Cartesian vector approach, the force and position vectors shown in Fig. 4–19*c* are

$$\mathbf{r} = \{0.4\mathbf{i} - 0.2\mathbf{j}\} \text{ m}$$
$$\mathbf{F} = \{400 \sin 30°\mathbf{i} - 400 \cos 30°\mathbf{j}\} \text{ N}$$
$$= \{200.0\mathbf{i} - 346.4\mathbf{j}\} \text{ N}$$

The moment is therefore

$$\mathbf{M}_O = \mathbf{r} \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0.4 & -0.2 & 0 \\ 200.0 & -346.4 & 0 \end{vmatrix}$$
$$= 0\mathbf{i} - 0\mathbf{j} + [0.4(-346.4) - (-0.2)(200.0)]\mathbf{k}$$
$$= \{-98.6\mathbf{k}\} \text{ N} \cdot \text{m} \qquad \textit{Ans.}$$

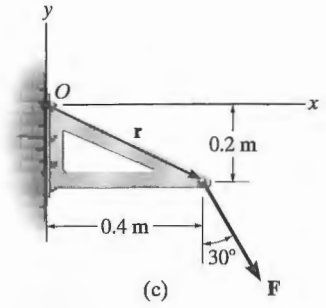


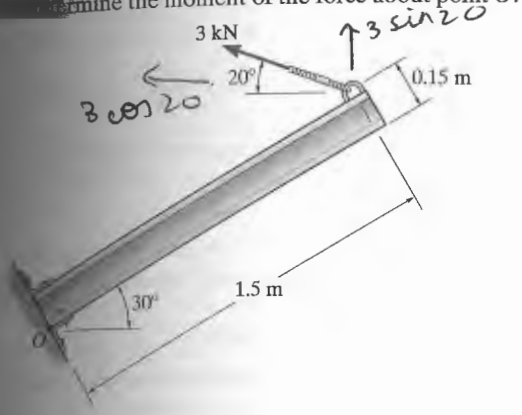
Fig. 4–19

NOTE: It is seen that the scalar analysis (Solution I) provides a more *convenient method* for analysis than Solution II since the direction of the moment and the moment arm for each component force are easy to establish. Hence, this method is generally recommended for solving problems displayed in two dimensions, whereas a Cartesian vector analysis is generally recommended only for solving three-dimensional problems.

Exclusively for The University of Melbourne

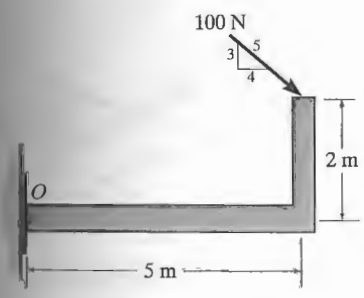
FUNDAMENTAL PROBLEMS

F4–1. Determine the moment of the force about point O.



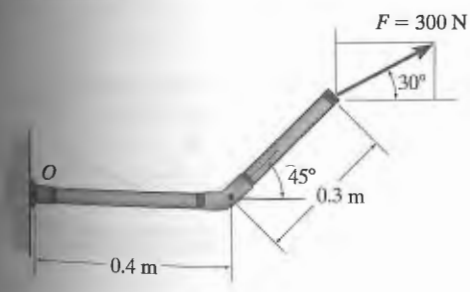
F4–1

F4–2. Determine the moment of the force about point O.



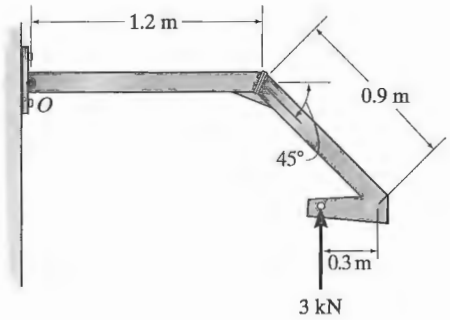
F4–2

F4–3. Determine the moment of the force about point O.



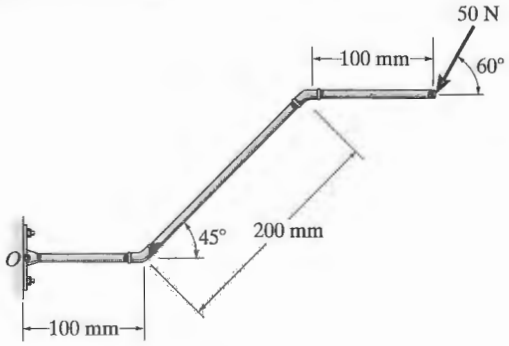
F4–3

F4–4. Determine the moment of the force about point O.



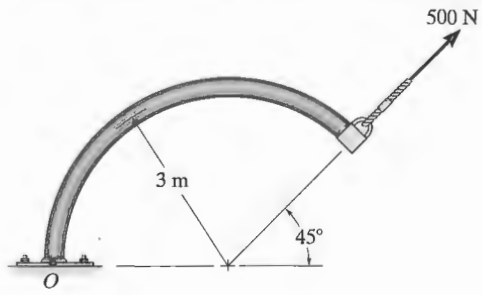
F4–4

F4–5. Determine the moment of the force about point O. Neglect the thickness of the member.



F4–5

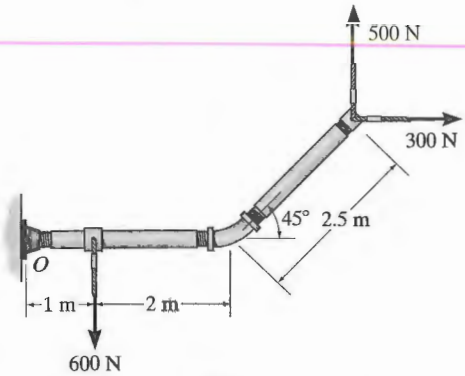
F4–6. Determine the moment of the force about point O.



F4–6

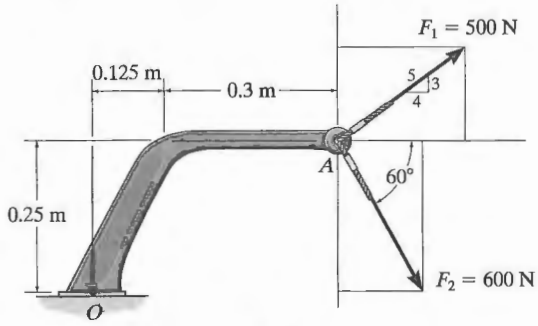
Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

F4–7. Determine the resultant moment produced by the forces about point O.



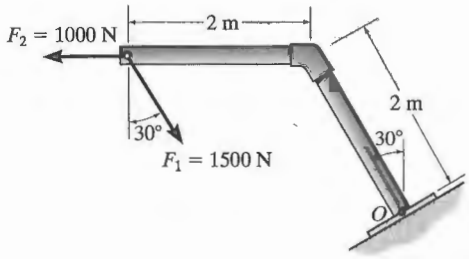
F4–7

F4–8. Determine the resultant moment produced by the forces about point O.



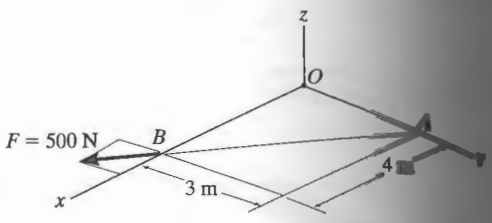
F4–8

F4–9. Determine the resultant moment produced by the forces about point O.



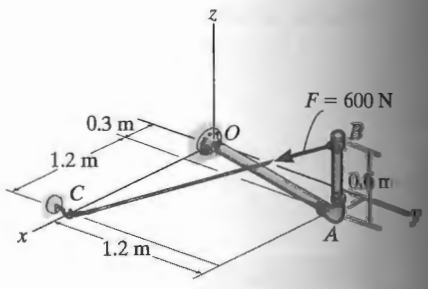
F4–9

F4–10. Determine the moment of force **F** about point O. Express the result as a Cartesian vector.



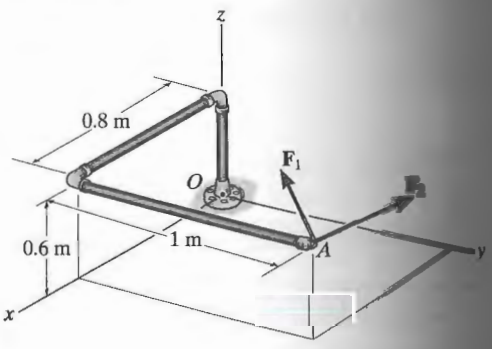
F 4–10

F4–11. Determine the moment of force **F** about point O. Express the result as a Cartesian vector.



F4–11

F4–12. If $\mathbf{F}_1 = \{100\mathbf{i} - 120\mathbf{j} + 75\mathbf{k}\}$ N and $\mathbf{F}_2 = \{-200\mathbf{i} + 250\mathbf{j} + 100\mathbf{k}\}$ N, determine the resultant moment produced by these forces about point O. Express the result as a Cartesian vector.



F4–12

ENGINEERING

Customized coaching is available!

PROBLEMS

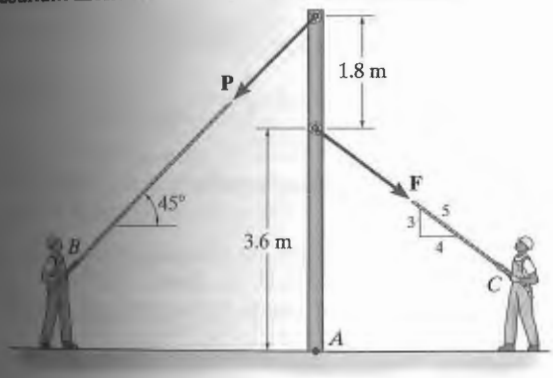
4-1. If **A**, **B**, and **D** are given vectors, prove the distributive law for the vector cross product, i.e., $\mathbf{A} \times (\mathbf{B} + \mathbf{D}) = (\mathbf{A} \times \mathbf{B}) + (\mathbf{A} \times \mathbf{D})$.

4-2. Prove the triple scalar product identity $\mathbf{A} \cdot \mathbf{B} \times \mathbf{C} = \mathbf{A} \times \mathbf{B} \cdot \mathbf{C}$.

4-3. Given the three nonzero vectors **A**, **B**, and **C**, show that if $\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C}) = 0$, the three vectors *must* lie in the same plane.

•4-4. Two men exert forces of $F = 400$ N and $P = 250$ N on the ropes. Determine the moment of each force about A. Which way will the pole rotate, clockwise or counterclockwise?

•4-5. If the man at B exerts a force of $P = 150$ N on his rope, determine the magnitude of the force **F** the man at C must exert to prevent the pole from rotating, i.e., so the resultant moment about A of both forces is zero.

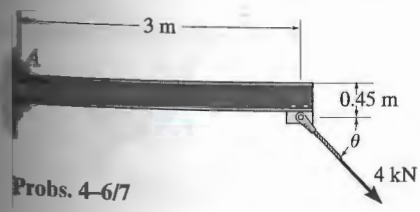


Probs. 4–4/5

4-6. If $\theta = 45°$, determine the moment produced by the 4 kN force about point A.

Video Solutions

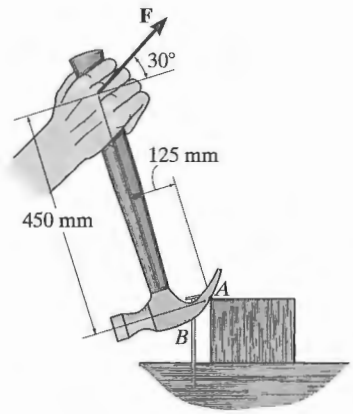
4-7. If the moment produced by the 4-kN force about point A is 10 kN · m clockwise, determine the angle θ, where $0 \le \theta \le 90°$.



Probs. 4–6/7

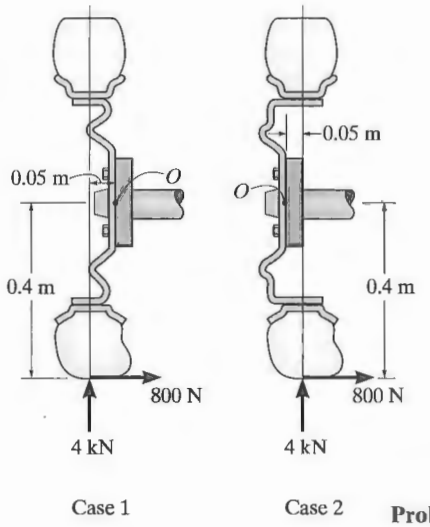
*4-8. The handle of the hammer is subjected to the force of $F = 100$ N. Determine the moment of this force about the point A.

•4-9. In order to pull out the nail at B, the force **F** exerted on the handle of the hammer must produce a clockwise moment of 60 N · m about point A. Determine the required magnitude of force **F**.



Probs. 4–8/9

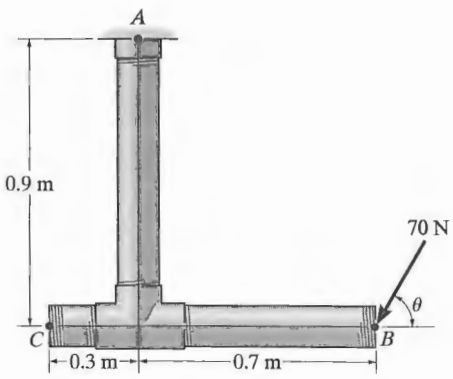
4-10. The hub of the wheel can be attached to the axle either with negative offset (left) or with positive offset (right). If the tire is subjected to both a normal and radial load as shown, determine the resultant moment of these loads about point O on the axle for both cases.



Prob. 4–10

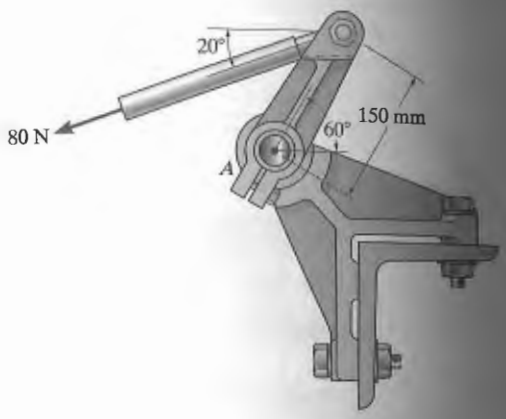
4–27. The 70-N force acts on the end of the pipe at B. Determine (a) the moment of this force about point A, and (b) the magnitude and direction of a horizontal force, applied at C, which produces the same moment. Take $\theta = 60°$.

***4–28.** The 70-N force acts on the end of the pipe at B. Determine the angles θ ($0° \leq \theta \leq 180°$) of the force that will produce maximum and minimum moments about point A. What are the magnitudes of these moments?



Probs. 4–27/28

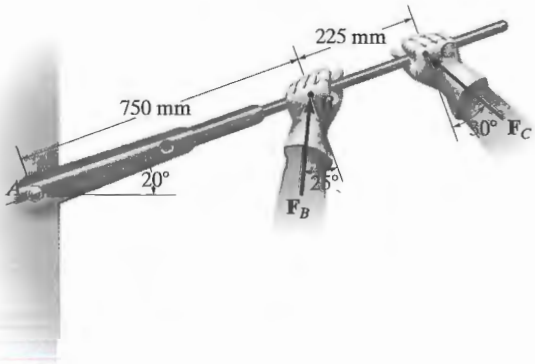
4–31. The rod on the power control mechan[redacted] for a business jet is subjected to a force of 80 N. Determine the moment of this force about the bearing at A.



Prob. 4–31

•4–29. Determine the moment of each force about the bolt located at A. Take $F_B = 200$ N, $F_C = 250$ N.

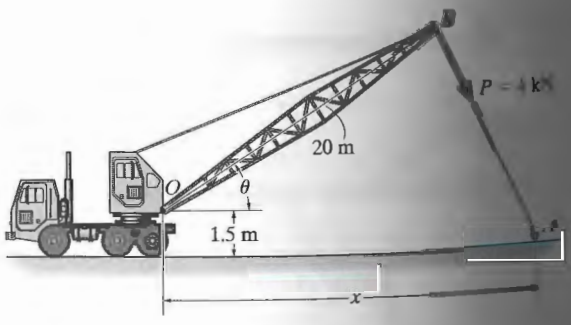
4–30. If $F_B = 150$ N and $F_C = 225$ N, determine the resultant moment about the bolt located at A.



Probs. 4–29/30

***4–32.** The towline exerts a force of $P = 4$ kN at the end of the 20-m-long crane boom. If $\theta = 30°$, determine the placement x of the hook at A so that this force creates a maximum moment about point O. What is this moment?

•4–33. The towline exerts a force of $P = 4$ kN at the end of the 20-m-long crane boom. If $x = 25$ m, determine the position θ of the boom so that this force creates a maximum moment about point O. What is this moment?



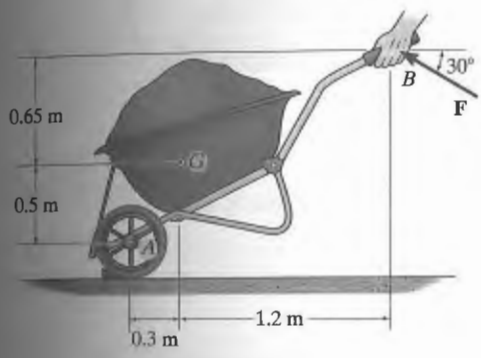
Probs. 4–32/33

Exclusively for The University of Melbourne

4–34. In order to hold the wheelbarrow in the position shown, force **F** must produce a counterclockwise moment of 200 N · m about the axle at A. Determine the required magnitude of force **F**.

4–35. The wheelbarrow and its contents have a mass of 50 kg and a center of mass at G. If the resultant moment produced by force **F** and the weight about point A is to be zero, determine the required magnitude of force **F**.

4–36. The wheelbarrow and its contents have a center of mass at G. If $F = 100$ N and the resultant moment produced by force **F** and the weight about the axle at A is zero, determine the mass of the wheelbarrow and its contents.

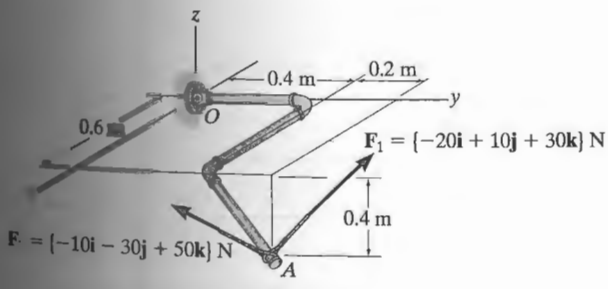


Prob. 4–34/35/36

•4–37. Determine the moment produced by $\mathbf{F}_1$ about point O. Express the result as a Cartesian vector.

4–38. Determine the moment produced by $\mathbf{F}_2$ about point O. Express the result as a Cartesian vector.

4–39. Determine the resultant moment produced by the two forces about point O. Express the result as a Cartesian vector.



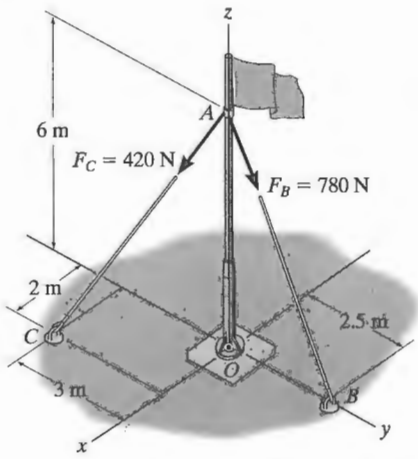
Probs. 4–37/38/39

***4–40.** Determine the moment produced by force $\mathbf{F}_B$ about point O. Express the result as a Cartesian vector.

•4–41. Determine the moment produced by force $\mathbf{F}_C$ about point O. Express the result as a Cartesian vector.

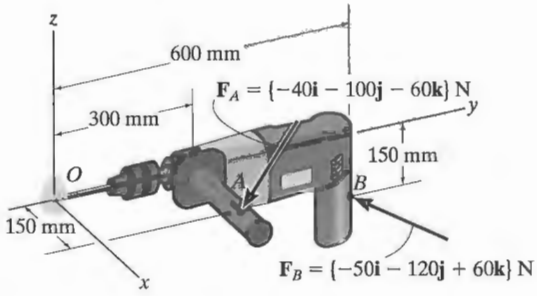
4–42. Determine the resultant moment produced by forces $\mathbf{F}_B$ and $\mathbf{F}_C$ about point O. Express the result as a Cartesian vector.

Video Solutions



Probs. 4–40/41/42

4–43. Determine the moment produced by each force about point O located on the drill bit. Express the results as Cartesian vectors.



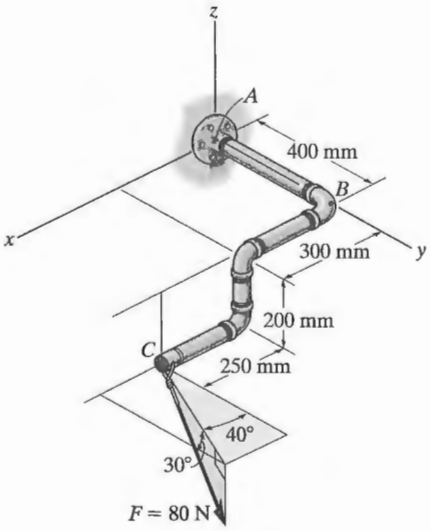
Prob. 4–43

***4–44.** A force of $\mathbf{F} = \{6\mathbf{i} - 2\mathbf{j} + 1\mathbf{k}\}$ kN produces a moment of $\mathbf{M}_O = \{4\mathbf{i} + 5\mathbf{j} - 14\mathbf{k}\}$ kN · m about the origin of coordinates, point O. If the force acts at a point having an x coordinate of $x = 1$ m, determine the y and z coordinates.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

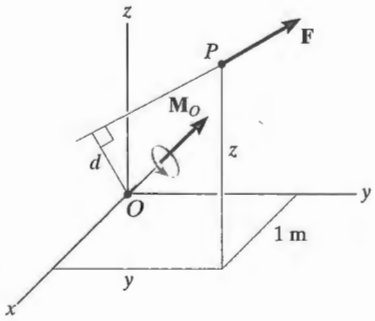
•**4–45.** The pipe assembly is subjected to the 80-N force. Determine the moment of this force about point A.

4–46. The pipe assembly is subjected to the 80-N force. Determine the moment of this force about point B.



Probs. 4–45/46

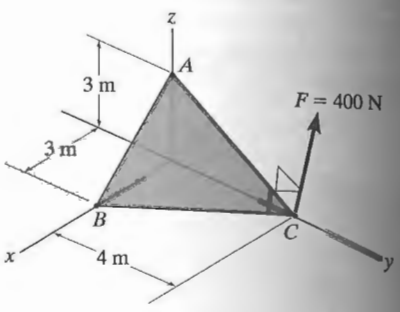
4–47. The force $\mathbf{F} = \{6\mathbf{i} + 8\mathbf{j} + 10\mathbf{k}\}$ N creates a moment about point O of $\mathbf{M}_O = \{-14\mathbf{i} + 8\mathbf{j} + 2\mathbf{k}\}$ N·m. If the force passes through a point having an x coordinate of 1 m, determine the y and z coordinates of the point. Also, realizing that $M_O = Fd$, determine the perpendicular distance d from point O to the line of action of $\mathbf{F}$.



Prob. 4–47

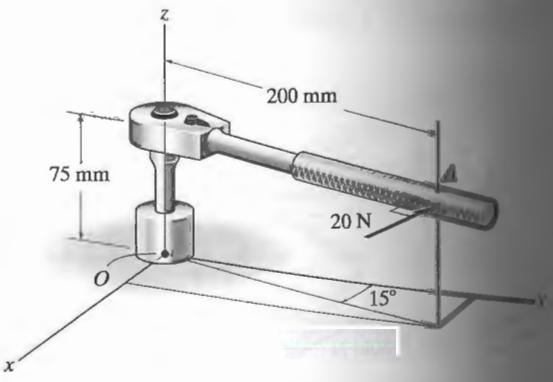
*4–48. Force $\mathbf{F}$ acts perpendicular to the inclined plane. Determine the moment produced by $\mathbf{F}$ about point A. Express the result as a Cartesian vector.

•4–49. Force $\mathbf{F}$ acts perpendicular to the inclined plane. Determine the moment produced by $\mathbf{F}$ about point B. Express the result as a Cartesian vector.



Probs. 4–48/49

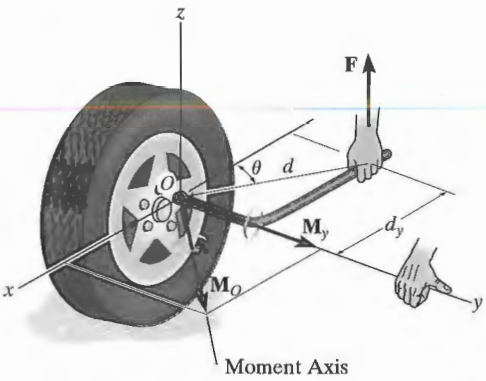
4–50. A 20-N horizontal force is applied perpendicular to the handle of the socket wrench. Determine the magnitude and the coordinate direction angles of the moment created by this force about point O.



Prob. 4–50

4.5 Moment of a Force about a Specified Axis

Sometimes, the moment produced by a force about a *specified axis* must be determined. For example, suppose the lug nut at O on the car tire in Fig. 4–20*a* needs to be loosened. The force applied to the wrench will create a tendency for the wrench and the nut to rotate about the *moment axis* passing through O; however, the nut can only rotate about the y axis. Therefore, to determine the turning effect, only the y component of the moment is needed, and the total moment produced is not important. To determine this component, we can use either a scalar or vector analysis.



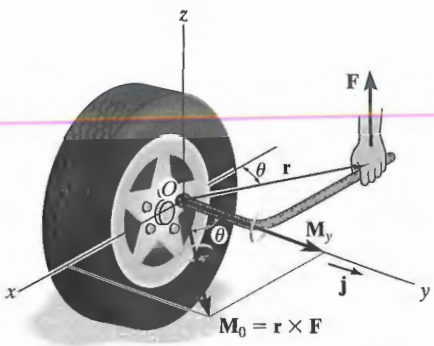
(a)

Fig. 4–20

Scalar Analysis. To use a scalar analysis in the case of the lug nut in Fig. 4–20*a*, the moment arm perpendicular distance from the axis to the line of action of the force is $d_y = d \cos\theta$. Thus, the moment of $\mathbf{F}$ about the y axis is $M_y = F\,d_y = F(d\cos\theta)$. According to the right-hand rule, $\mathbf{M}_y$ is directed along the positive y axis as shown in the figure. In general, for any axis a, the moment is

$$\boxed{M_a = Fd_a} \tag{4–10}$$

If large enough, the cable force $\mathbf{F}$ on the boom of this crane can cause the crane to topple over. To investigate this, the moment of the force must be calculated about an axis passing through the base of the legs at A and B.



(b)

Fig. 4–20

Vector Analysis. To find the moment of force **F** in Fig. 4–20*b* about the *y* axis using a vector analysis, we must first determine the moment of the force about *any point O* on the *y* axis by applying Eq. 4–7, $\mathbf{M}_O = \mathbf{r} \times \mathbf{F}$. The component $\mathbf{M}_y$ along the *y* axis is the *projection* of $\mathbf{M}_O$ onto the *y* axis. It can be found using the *dot product* discussed in Chapter 2, so that $M_y = \mathbf{j} \cdot \mathbf{M}_O = \mathbf{j} \cdot (\mathbf{r} \times \mathbf{F})$, where **j** is the unit vector for the *y* axis.

We can generalize this approach by letting $\mathbf{u}_a$ be the unit vector that specifies the direction of the *a* axis shown in Fig. 4–21. Then the moment of **F** about the axis is $M_a = \mathbf{u}_a \cdot (\mathbf{r} \times \mathbf{F})$. This combination is referred to as the *scalar triple product*. If the vectors are written in Cartesian form we have

$$M_a = [u_{a_x}\mathbf{i} + u_{a_y}\mathbf{j} + u_{a_z}\mathbf{k}] \cdot \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ r_x & r_y & r_z \\ F_x & F_y & F_z \end{vmatrix}$$

$$= u_{a_x}(r_yF_z - r_zF_y) - u_{a_y}(r_xF_z - r_zF_x) + u_{a_z}(r_xF_y - r_yF_x)$$

This result can also be written in the form of a determinant, making it easier to memorize.*

$$\boxed{M_a = \mathbf{u}_a \cdot (\mathbf{r} \times \mathbf{F}) = \begin{vmatrix} u_{a_x} & u_{a_y} & u_{a_z} \\ r_x & r_y & r_z \\ F_x & F_y & F_z \end{vmatrix}} \tag{4–11}$$

where

$u_{a_x}, u_{a_y}, u_{a_z}$ represent the *x*, *y*, *z* components of the unit vector defining the direction of the *a* axis

r_x, r_y, r_z represent the *x*, *y*, *z* components of the position vector extended from *any point O* on the *a* axis to *any point A* on the line of action of the force

F_x, F_y, F_z represent the *x*, *y*, *z* components of the force vector.

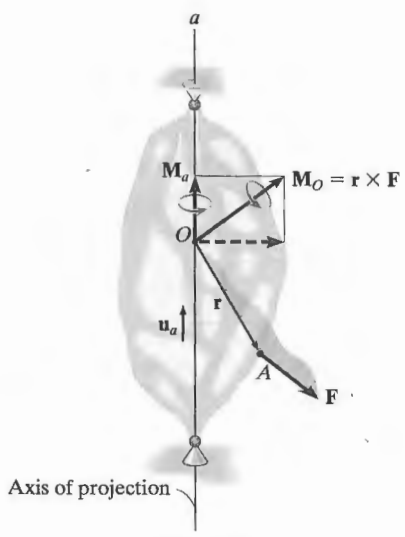


Fig. 4–21

When M_a is evaluated from Eq. 4–11, it will yield a positive or negative scalar. The sign of this scalar indicates the sense of direction of $\mathbf{M}_a$ along the *a* axis. If it is positive, then $\mathbf{M}_a$ will have the same sense as $\mathbf{u}_a$, whereas if it is negative, then $\mathbf{M}_a$ will act opposite to $\mathbf{u}_a$.

Once M_a is determined, we can then express $\mathbf{M}_a$ as a Cartesian vector, namely,

$$\mathbf{M}_a = M_a\mathbf{u}_a \tag{4–12}$$

The examples which follow illustrate numerical applications of the above concepts.

*Take a moment to expand this determinant, to show that it will yield the above result

Important Points

- The moment of a force about a specified axis can be determined provided the perpendicular distance d_a from the force line of action to the axis can be determined. $M_a = Fd_a$.
- If vector analysis is used, $M_a = \mathbf{u}_a \cdot (\mathbf{r} \times \mathbf{F})$, where $\mathbf{u}_a$ defines the direction of the axis and $\mathbf{r}$ is extended from *any point* on the axis to *any point* on the line of action of the force.
- If M_a is calculated as a negative scalar, then the sense of direction of $\mathbf{M}_a$ is opposite to $\mathbf{u}_a$.
- The moment $\mathbf{M}_a$ expressed as a Cartesian vector is determined from $\mathbf{M}_a = M_a\mathbf{u}_a$.

4

EXAMPLE 4.7

Determine the resultant moment of the three forces in Fig. 4–22 about the x axis, the y axis, and the z axis.

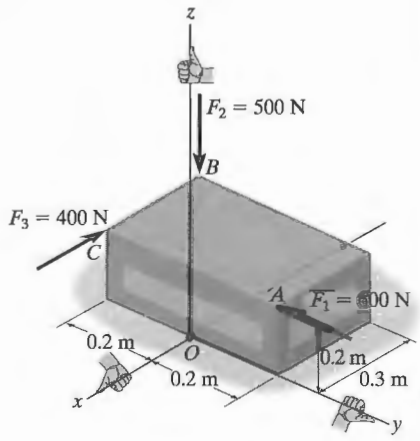


Fig. 4–22

SOLUTION

A force that is *parallel* to a coordinate axis or has a line of action that passes through the axis does *not* produce any moment or tendency for turning about that axis. Therefore, defining the positive direction of the moment of a force according to the right-hand rule, as shown in the figure, we have

$$M_x = (600\text{ N})(0.2\text{ m}) + (500\text{ N})(0.2\text{ m}) + 0 = 220\text{ N}\cdot\text{m} \qquad \textit{Ans.}$$

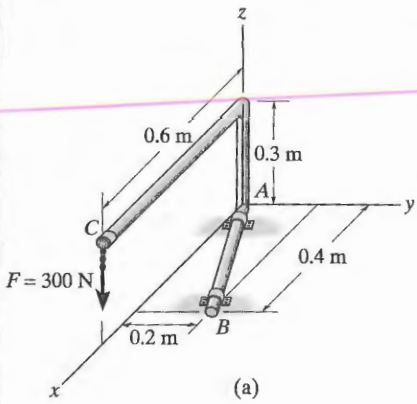
$$M_y = 0 - (500\text{ N})(0.3\text{ m}) - (400\text{ N})(0.2\text{ m}) = -230\text{ N}\cdot\text{m} \qquad \textit{Ans.}$$

$$M_z = 0 + 0 - (400\text{ N})(0.2\text{ m}) = -80\text{ N}\cdot\text{m} \qquad \textit{Ans.}$$

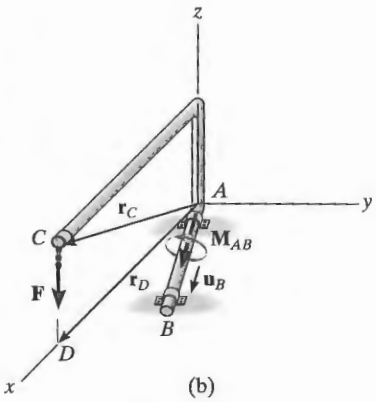
The negative signs indicate that $\mathbf{M}_y$ and $\mathbf{M}_z$ act in the $-y$ and $-z$ directions, respectively.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 4.8



(a)



(b)

Fig. 4–23

Determine the moment $\mathbf{M}_{AB}$ produced by the force $\mathbf{F}$ in Fig. 4–23*a*, which tends to rotate the rod about the AB axis.

SOLUTION

A vector analysis using $M_{AB} = \mathbf{u}_B \cdot (\mathbf{r} \times \mathbf{F})$ will be considered for the solution rather than trying to find the moment arm or perpendicular distance from the line of action of $\mathbf{F}$ to the AB axis. Each of the terms in the equation will now be identified.

Unit vector $\mathbf{u}_B$ defines the direction of the AB axis of the rod, Fig. 4–23*b*, where

$$\mathbf{u}_B = \frac{\mathbf{r}_B}{r_B} = \frac{\{0.4\mathbf{i} + 0.2\mathbf{j}\}\ \text{m}}{\sqrt{(0.4\ \text{m})^2 + (0.2\ \text{m})^2}} = 0.8944\mathbf{i} + 0.4472\mathbf{j}$$

Vector $\mathbf{r}$ is directed from *any point* on the AB axis to *any point* on the line of action of the force. For example, position vectors $\mathbf{r}_C$ and $\mathbf{r}_D$ are suitable, Fig. 4–23*b*. (Although not shown, $\mathbf{r}_{BC}$ or $\mathbf{r}_{BD}$ can also be used.) For simplicity, we choose $\mathbf{r}_D$, where

$$\mathbf{r}_D = \{0.6\mathbf{i}\}\ \text{m}$$

The force is

$$\mathbf{F} = \{-300\mathbf{k}\}\ \text{N}$$

Substituting these vectors into the determinant form and expanding, we have

$$M_{AB} = \mathbf{u}_B \cdot (\mathbf{r}_D \times \mathbf{F}) = \begin{vmatrix} 0.8944 & 0.4472 & 0 \\ 0.6 & 0 & 0 \\ 0 & 0 & -300 \end{vmatrix}$$

$$= 0.8944[0(-300) - 0(0)] - 0.4472[0.6(-300) - 0(0)] + 0[0.6(0) - 0(0)]$$

$$= 80.50\ \text{N} \cdot \text{m}$$

This positive result indicates that the sense of $\mathbf{M}_{AB}$ is in the same direction as $\mathbf{u}_B$.

Expressing $\mathbf{M}_{AB}$ as a Cartesian vector yields

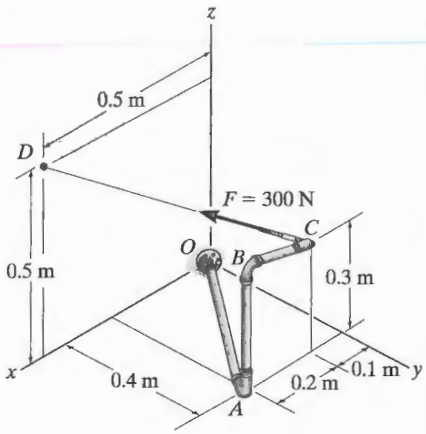
$$\mathbf{M}_{AB} = M_{AB}\mathbf{u}_B = (80.50\ \text{N} \cdot \text{m})(0.8944\mathbf{i} + 0.4472\mathbf{j})$$
$$= \{72.0\mathbf{i} + 36.0\mathbf{j}\}\ \text{N} \cdot \text{m} \qquad \textit{Ans.}$$

The result is shown in Fig. 4–23*b*.

NOTE: If axis AB is defined using a unit vector directed from B toward A, then in the above formulation $-\mathbf{u}_B$ would have to be used. This would lead to $M_{AB} = -80.50\ \text{N} \cdot \text{m}$. Consequently, $\mathbf{M}_{AB} = M_{AB}(-\mathbf{u}_B)$, and the same result would be obtained.

EXAMPLE 4.9

Determine the magnitude of the moment of force **F** about segment *OA* of the pipe assembly in Fig. 4–24*a*.



(a)

SOLUTION

The moment of **F** about the *OA* axis is determined from $M_{OA} = \mathbf{u}_{OA} \cdot (\mathbf{r} \times \mathbf{F})$, where **r** is a position vector extending from any point on the *OA* axis to any point on the line of action of **F**. As indicated in Fig. 4–24*b*, either $\mathbf{r}_{OD}$, $\mathbf{r}_{OC}$, $\mathbf{r}_{AD}$, or $\mathbf{r}_{AC}$ can be used; however, $\mathbf{r}_{OD}$ will be considered since it will simplify the calculation.

The unit vector $\mathbf{u}_{OA}$, which specifies the direction of the *OA* axis, is

$$\mathbf{u}_{OA} = \frac{\mathbf{r}_{OA}}{r_{OA}} = \frac{\{0.3\mathbf{i} + 0.4\mathbf{j}\}\ \text{m}}{\sqrt{(0.3\ \text{m})^2 + (0.4\ \text{m})^2}} = 0.6\mathbf{i} + 0.8\mathbf{j}$$

and the position vector $\mathbf{r}_{OD}$ is

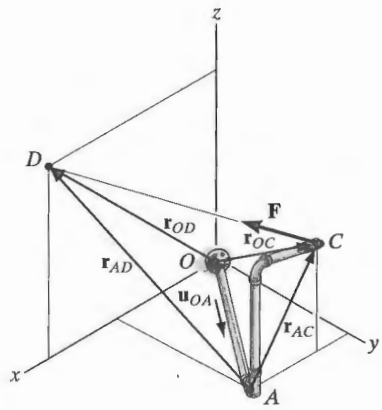
$$\mathbf{r}_{OD} = \{0.5\mathbf{i} + 0.5\mathbf{k}\}\ \text{m}$$

The force **F** expressed as a Cartesian vector is

$$\mathbf{F} = F\left(\frac{\mathbf{r}_{CD}}{r_{CD}}\right)$$

$$= (300\ \text{N})\left[\frac{\{0.4\mathbf{i} - 0.4\mathbf{j} + 0.2\mathbf{k}\}\ \text{m}}{\sqrt{(0.4\ \text{m})^2 + (-0.4\ \text{m})^2 + (0.2\ \text{m})^2}}\right]$$

$$= \{200\mathbf{i} - 200\mathbf{j} + 100\mathbf{k}\}\ \text{N}$$



(b)

Fig. 4–24

Therefore,

$$M_{OA} = \mathbf{u}_{OA} \cdot (\mathbf{r}_{OD} \times \mathbf{F})$$

$$= \begin{vmatrix} 0.6 & 0.8 & 0 \\ 0.5 & 0 & 0.5 \\ 200 & -200 & 100 \end{vmatrix}$$

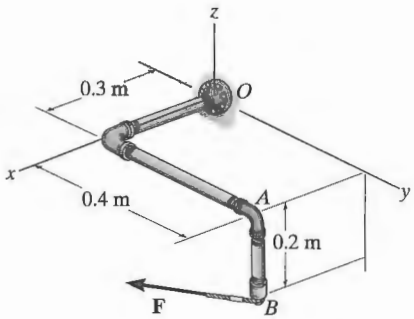
$$= 0.6[0(100) - (0.5)(-200)] - 0.8[0.5(100) - (0.5)(200)] + 0$$

$$= 100\ \text{N}\cdot\text{m} \qquad \textit{Ans.}$$

FUNDAMENTAL PROBLEMS

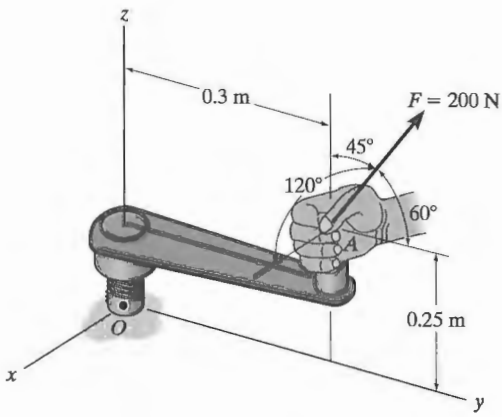
F4–13. Determine the magnitude of the moment of the force $\mathbf{F} = \{300\mathbf{i} - 200\mathbf{j} + 150\mathbf{k}\}$ N about the x axis. Express the result as a Cartesian vector.

F4–14. Determine the magnitude of the moment of the force $\mathbf{F} = \{300\mathbf{i} - 200\mathbf{j} + 150\mathbf{k}\}$ N about the OA axis. Express the result as a Cartesian vector.



F4–13/14

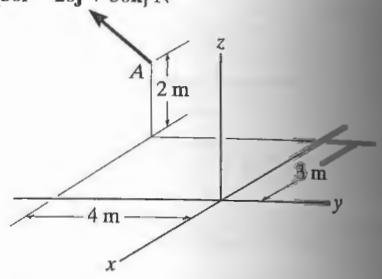
F4–15. Determine the magnitude of the moment of the 200-N force about the x axis.



F4–15

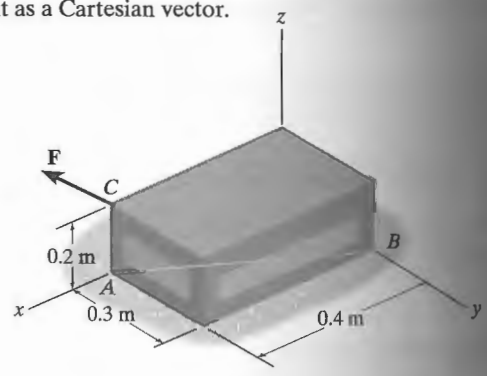
F4–16. Determine the magnitude of the moment of the force about the y axis.

$\mathbf{F} = \{30\mathbf{i} - 20\mathbf{j} + 50\mathbf{k}\}$ N



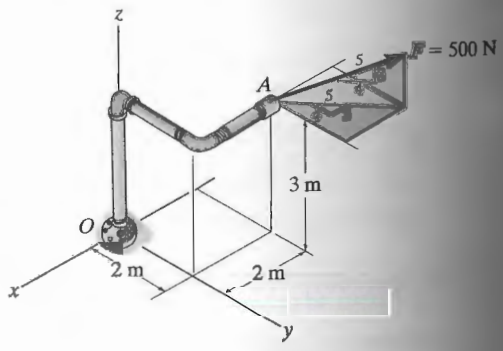
F4–16

F4–17. Determine the moment of the force $\mathbf{F} = \{50\mathbf{i} - 40\mathbf{j} + 20\mathbf{k}\}$ N about the AB axis. Express the result as a Cartesian vector.



F4–17

F4–18. Determine the moment of force $\mathbf{F}$ about the x, the y, and the z axes. Use a scalar analysis.



F4–18

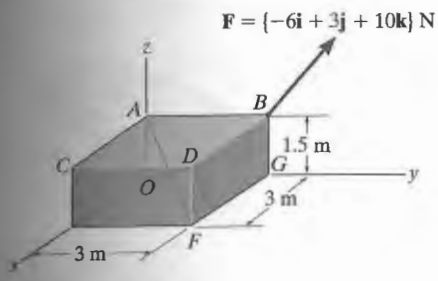
ing exclusively for The University of Melbourne

INEERING™
tomized coaching is available!

BLEMS

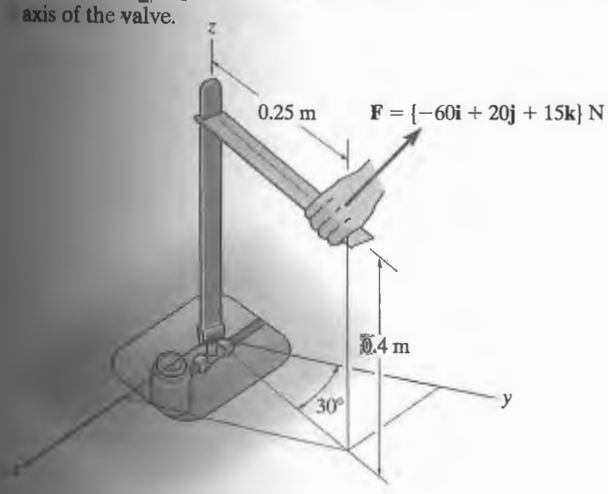
4–51. Determine the moment produced by force **F** about the diagonal AF of the rectangular block. Express the result as a Cartesian vector.

4–52. Determine the moment produced by force **F** about the diagonal OD of the rectangular block. Express the result as a Cartesian vector.



Probs. 4–51/52

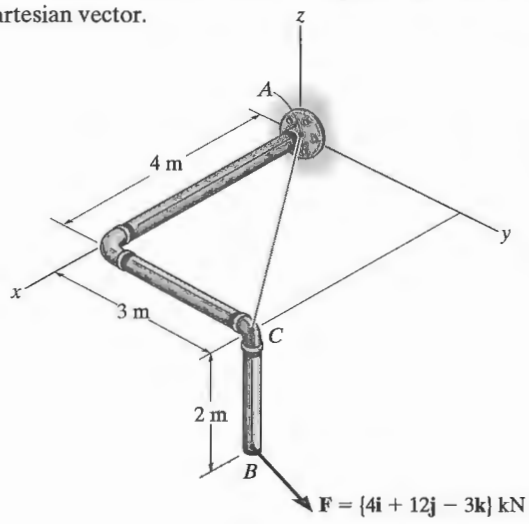
•4–53. The tool is used to shut off gas valves that are difficult to access. If the force **F** is applied to the handle, determine the component of the moment created about the axis of the valve.



Prob. 4–53

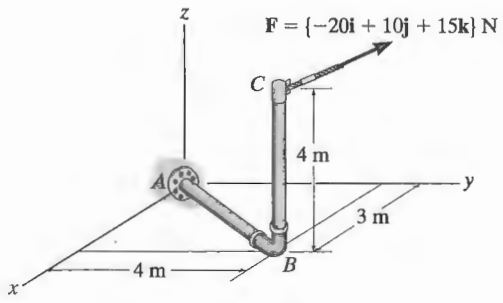
4–54. Determine the magnitude of the moments of the force **F** about the x, y, and z axes. Solve the problem (a) using a Cartesian vector approach and (b) using a scalar approach.

4–55. Determine the moment of the force **F** about an axis extending between A and C. Express the result as a Cartesian vector.



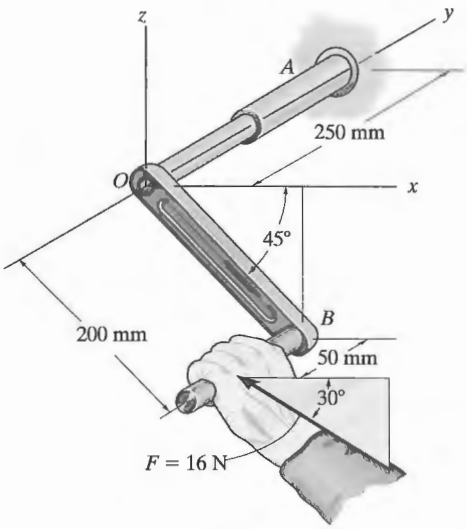
Probs. 4–54/55

***4–56.** Determine the moment produced by force **F** about segment AB of the pipe assembly. Express the result as a Cartesian vector.



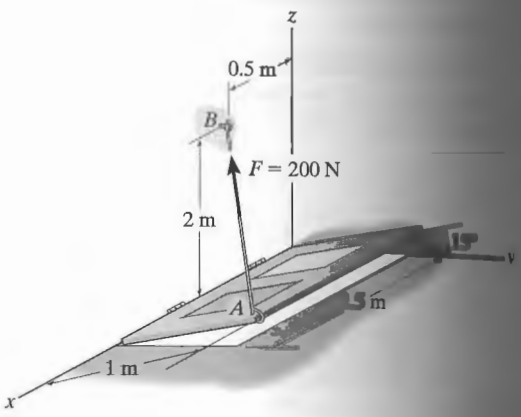
Prob. 4–56

•4–57. Determine the magnitude of the moment that the force **F** exerts about the y axis of the shaft. Solve the problem using a Cartesian vector approach and using a scalar approach.



Prob. 4–57

***4–60.** Determine the magnitude of the moment produced by the force of $F = 200$ N about the hinged axis (the x axis) of the door.

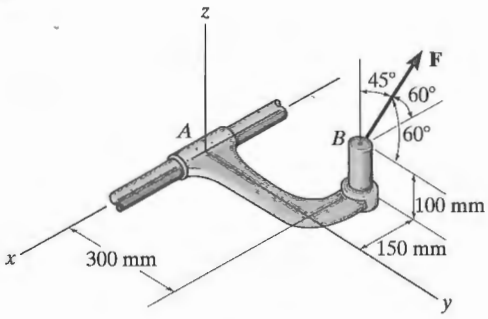


Prob. 4–60

4–58. If $F = 450$ N, determine the magnitude of the moment produced by this force about the x axis.

4–59. The friction at sleeve A can provide a maximum resisting moment of 125 N · m about the x axis. Determine the largest magnitude of force **F** that can be applied to the bracket so that the bracket will not turn.

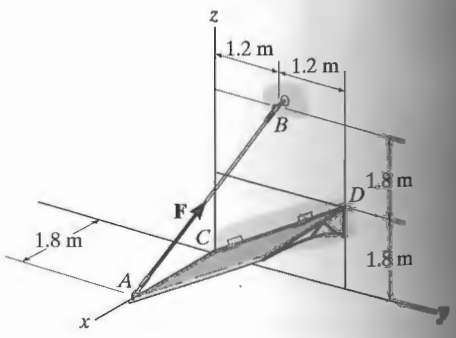
Video Solutions



Probs. 4–58/59

•4–61. If the tension in the cable is $F = 700$ N, determine the magnitude of the moment produced by this force about the hinged axis, CD, of the panel.

4–62. Determine the magnitude of force **F** in cable AB in order to produce a moment of 750 N · m about the hinged axis CD, which is needed to hold the panel in the position shown.



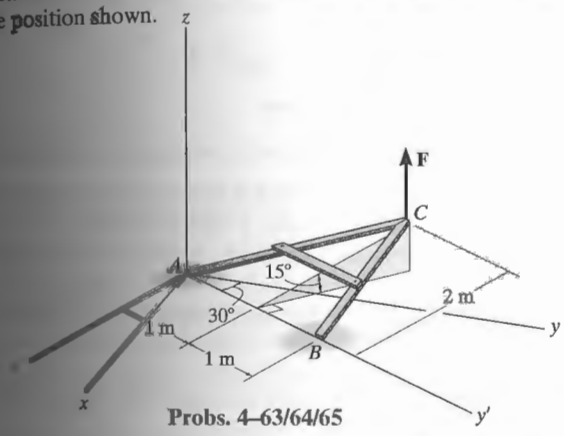
Probs. 4–61/62

4–63. The A-frame is being hoisted into an upright position by the vertical force of $F = 400$ N. Determine the moment of this force about the y' axis passing through points A and B when the frame is in the position shown.

•4–64. The A-frame is being hoisted into an upright position by the vertical force of $F = 400$ N. Determine the moment of this force about the x axis when the frame is in the position shown.

Video Solutions

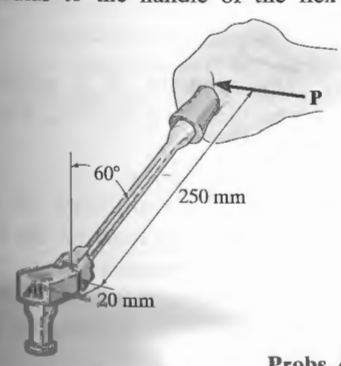
•4–65. The A-frame is being hoisted into an upright position by the vertical force of $F = 400$ N. Determine the moment of this force about the y axis when the frame is in the position shown.



Probs. 4–63/64/65

4–66. The flex-headed ratchet wrench is subjected to a force of $P = 80$ N, applied perpendicular to the handle as shown. Determine the moment or torque this imparts along the vertical axis of the bolt at A.

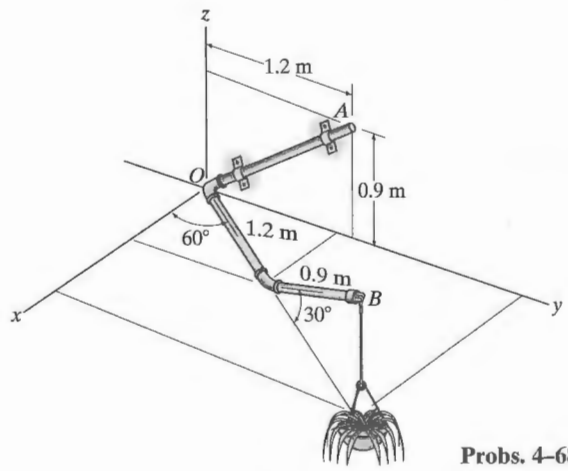
•4–67. If a torque or moment of $10\ \text{N}\cdot\text{m}$ is required to loosen the bolt at A, determine the force P that must be applied perpendicular to the handle of the flex-headed ratchet wrench.



Probs. 4–66/67

***4–68.** The pipe assembly is secured on the wall by the two brackets. If the flower pot has a weight of 250 N, determine the magnitude of the moment produced by the weight about the OA axis.

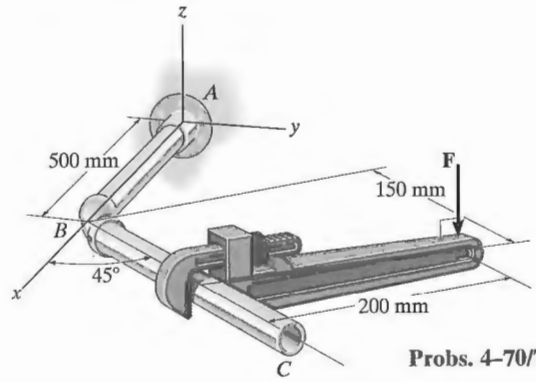
•4–69. The pipe assembly is secured on the wall by the two brackets. If the frictional force of both brackets can resist a maximum moment of $225\ \text{N}\cdot\text{m}$, determine the largest weight of the flower pot that can be supported by the assembly without causing it to rotate about the OA axis.



Probs. 4–68/69

4–70. A vertical force of $F = 60$ N is applied to the handle of the pipe wrench. Determine the moment that this force exerts along the axis AB (x axis) of the pipe assembly. Both the wrench and pipe assembly ABC lie in the x–y plane. *Suggestion:* Use a scalar analysis.

4–71. Determine the magnitude of the vertical force **F** acting on the handle of the wrench so that this force produces a component of moment along the AB axis (x axis) of the pipe assembly of $(M_A)_x = \{-5\mathbf{i}\}\ \text{N}\cdot\text{m}$. Both the pipe assembly ABC and the wrench lie in the x–y plane. *Suggestion:* Use a scalar analysis.



Probs. 4–70/71

4.6 Moment of a Couple

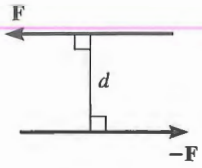


Fig. 4–25

A *couple* is defined as two parallel forces that have the same magnitude, but opposite directions, and are separated by a perpendicular distance d, Fig. 4–25. Since the resultant force is zero, the only effect of a couple is to produce a rotation or tendency of rotation in a specified direction. For example, imagine that you are driving a car with both hands on the steering wheel and you are making a turn. One hand will push up on the wheel while the other hand pulls down, which causes the steering wheel to rotate.

The moment produced by a couple is called a *couple moment*. We can determine its value by finding the sum of the moments of both couple forces about *any* arbitrary point. For example, in Fig. 4–26, position vectors $\mathbf{r}_A$ and $\mathbf{r}_B$ are directed from point O to points A and B lying on the line of action of $-\mathbf{F}$ and $\mathbf{F}$. The couple moment determined about O is therefore

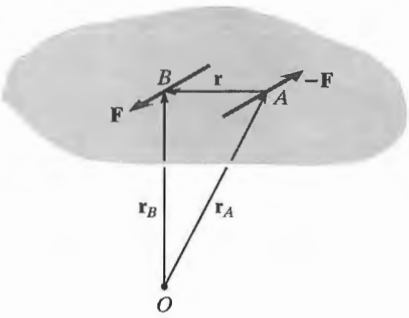


Fig. 4–26

$$\mathbf{M} = \mathbf{r}_B \times \mathbf{F} + \mathbf{r}_A \times -\mathbf{F} = (\mathbf{r}_B - \mathbf{r}_A) \times \mathbf{F}$$

However $\mathbf{r}_B = \mathbf{r}_A + \mathbf{r}$ or $\mathbf{r} = \mathbf{r}_B - \mathbf{r}_A$, so that

$$\mathbf{M} = \mathbf{r} \times \mathbf{F} \tag{4–13}$$

This result indicates that a couple moment is a *free vector*, i.e., it can act at *any point* since $\mathbf{M}$ depends *only* upon the position vector $\mathbf{r}$ directed *between* the forces and *not* the position vectors $\mathbf{r}_A$ and $\mathbf{r}_B$, directed from the arbitrary point O to the forces. This concept is unlike the moment of a force, which requires a definite point (or axis) about which moments are determined.

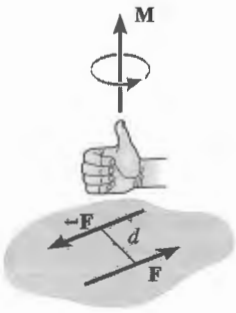


Fig. 4–27

Scalar Formulation. The moment of a couple, $\mathbf{M}$, Fig. 4–27, is defined as having a *magnitude* of

$$\boxed{M = Fd} \tag{4–14}$$

where F is the magnitude of one of the forces and d is the perpendicular distance or moment arm between the forces. The *direction* and sense of the couple moment are determined by the right-hand rule, where the thumb indicates this direction when the fingers are curled with the sense of rotation caused by the couple forces. In all cases, $\mathbf{M}$ will act perpendicular to the plane containing these forces.

Vector Formulation. The moment of a couple can also be expressed by the vector cross product using Eq. 4–13, i.e.,

$$\boxed{\mathbf{M} = \mathbf{r} \times \mathbf{F}} \tag{4–15}$$

Application of this equation is easily remembered if one thinks of taking the moments of both forces about a point lying on the line of action of one of the forces. For example, if moments are taken about point A in Fig. 4–26, the moment of $-\mathbf{F}$ is *zero* about this point, and the moment of $\mathbf{F}$ is defined from Eq. 4–15. Therefore, in the formulation $\mathbf{r}$ is crossed with the force $\mathbf{F}$ to which it is directed.

exclusively for The University of Melbourne

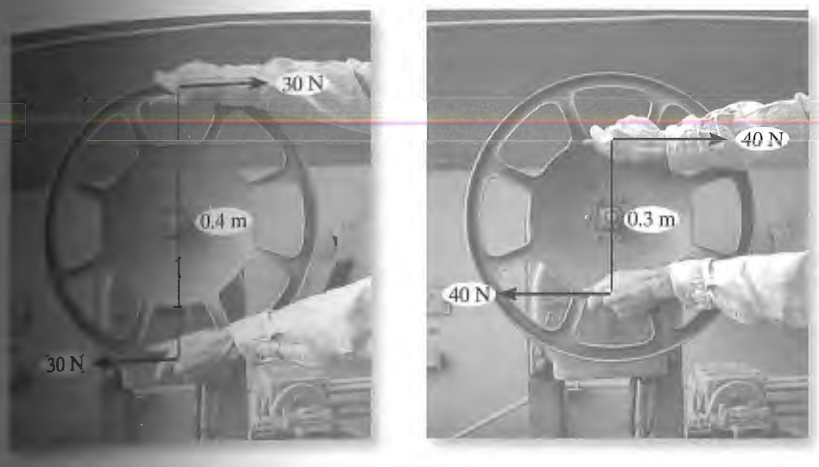


Fig. 4–28

Equivalent Couples. If two couples produce a moment with the *same magnitude and direction*, then these two couples are *equivalent*. For example, the two couples shown in Fig. 4–28 are *equivalent* because each couple moment has a magnitude of $M = 30\text{ N}(0.4\text{ m}) = 40\text{ N}(0.3\text{ m}) = 12\text{ N}\cdot\text{m}$, and each is directed into the plane of the page. Notice that larger forces are required in the second case to create the same turning effect because the hands are placed closer together. Also, if the wheel was connected to the shaft at a point other than at its center, then the wheel would still turn when each couple is applied since the $12\text{ N}\cdot\text{m}$ couple is a free vector.

Resultant Couple Moment. Since couple moments are vectors, their resultant can be determined by vector addition. For example, consider the couple moments $\mathbf{M}_1$ and $\mathbf{M}_2$ acting on the pipe in Fig. 4–29*a*. Since each couple moment is a free vector, we can join their tails at any arbitrary point and find the resultant couple moment, $\mathbf{M}_R = \mathbf{M}_1 + \mathbf{M}_2$ as shown in Fig. 4–29*b*.

If more than two couple moments act on the body, we may generalize this concept and write the vector resultant as

$$\mathbf{M}_R = \Sigma(\mathbf{r} \times \mathbf{F}) \tag{4–16}$$

These concepts are illustrated numerically in the examples that follow. In general, problems projected in two dimensions should be solved using a scalar analysis since the moment arms and force components are easy to determine.

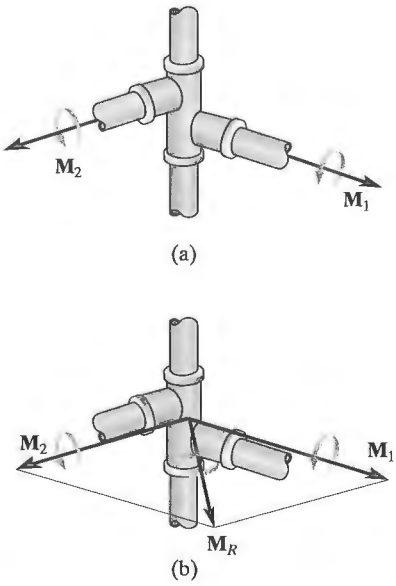
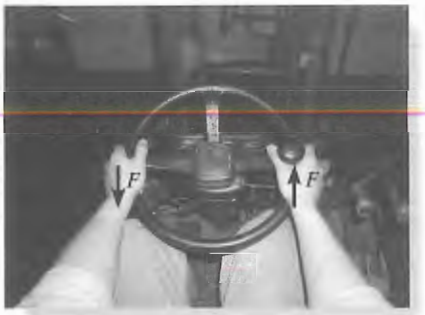


Fig. 4–29

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.



Steering wheels on vehicles have been made smaller than on older vehicles because power steering does not require the driver to apply a large couple moment to the rim of the wheel.

Important Points

- A couple moment is produced by two noncollinear forces that are equal in magnitude but opposite in direction. Its effect is to produce pure rotation, or tendency for rotation in a specified direction.
- A couple moment is a free vector, and as a result it causes the same rotational effect on a body regardless of where the couple moment is applied to the body.
- The moment of the two couple forces can be determined about *any point*. For convenience, this point is often chosen on the line of action of one of the forces in order to eliminate the moment of this force about the point.
- In three dimensions the couple moment is often determined using the vector formulation, $\mathbf{M} = \mathbf{r} \times \mathbf{F}$, where $\mathbf{r}$ is directed from *any point* on the line of action of one of the forces to *any point* on the line of action of the other force $\mathbf{F}$.
- A resultant couple moment is simply the vector sum of all the couple moments of the system.

EXAMPLE 4.10

Determine the resultant couple moment of the three couples acting on the plate in Fig. 4–30.

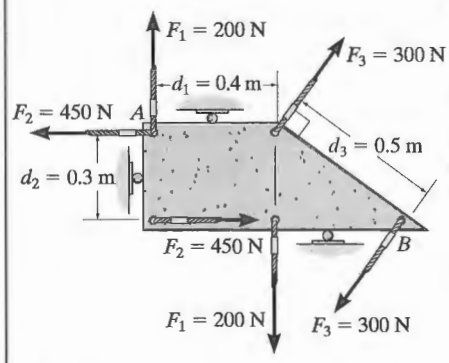


Fig. 4–30

SOLUTION

As shown the perpendicular distances between each pair of couple forces are $d_1 = 0.4$ m, $d_2 = 0.3$ m, and $d_3 = 0.5$ m. Considering counterclockwise couple moments as positive, we have

$$\circlearrowleft + M_R = \Sigma M;\ M_R = -F_1 d_1 + F_2 d_2 - F_3 d_3$$

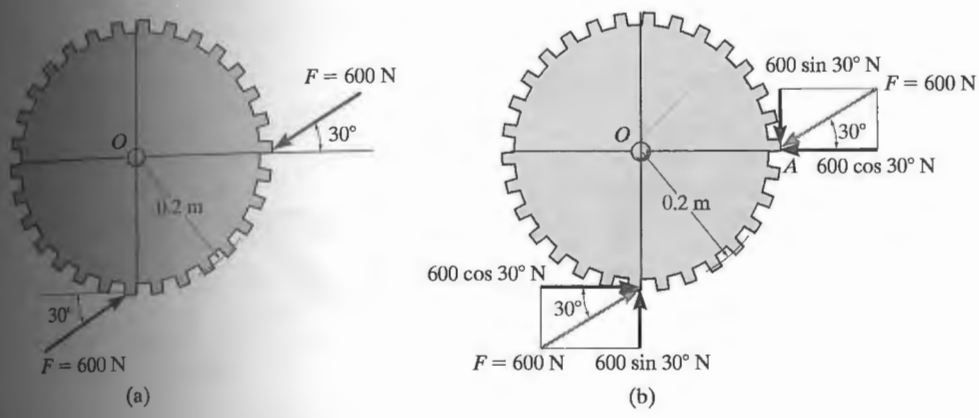
$$= (-200\text{ N})(0.4\text{ m}) + (450\text{ N})(0.3\text{ m}) - (300\text{ N})(0.5\text{ m})$$

$$= -95\text{ N}\cdot\text{m} = 95\text{ N}\cdot\text{m}\circlearrowright \qquad \textit{Ans.}$$

The negative sign indicates that $\mathbf{M}_R$ has a clockwise rotational sense.

EXAMPLE 1

Determine the magnitude and direction of the couple moment acting on the gear in Fig. 4–31*a*.



SOLUTION

The easiest solution requires resolving each force into its components as shown in Fig. 4–31*b*. The couple moment can be determined by summing the moments of these force components about any point, for example, the center O of the gear or point A. If we consider counterclockwise moments as positive, we have

$$\circlearrowleft + M = \Sigma M_O;\ M = (600\cos 30°\ \text{N})(0.2\ \text{m}) - (600\sin 30°\ \text{N})(0.2\ \text{m})$$
$$= 43.9\ \text{N}\cdot\text{m}\circlearrowleft \qquad \textit{Ans.}$$

or

$$\circlearrowleft + M = \Sigma M_A;\ M = (600\cos 30°\ \text{N})(0.2\ \text{m}) - (600\sin 30°\ \text{N})(0.2\ \text{m})$$
$$= 43.9\ \text{N}\cdot\text{m}\circlearrowleft \qquad \textit{Ans.}$$

This positive result indicates that **M** has a counterclockwise rotational sense, so it is directed outward, perpendicular to the page.

NOTE: The same result can also be obtained using $M = Fd$, where d is the perpendicular distance between the lines of action of the couple forces, Fig. 4–31*c*. However, the computation for d is more involved. Realize that the couple moment is a free vector and can act at any point on the gear and produce the same turning effect about point O.

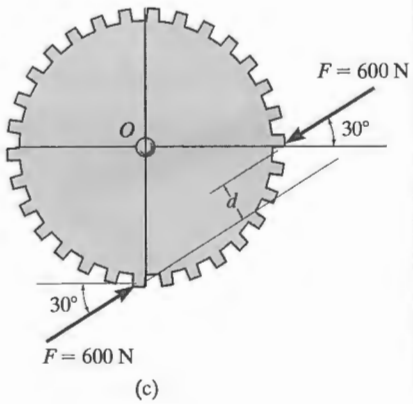


Fig. 4–31

EXAMPLE 4.12

Determine the couple moment acting on the pipe shown in Fig. 4–32*a*. Segment AB is directed 30° below the x–y plane.

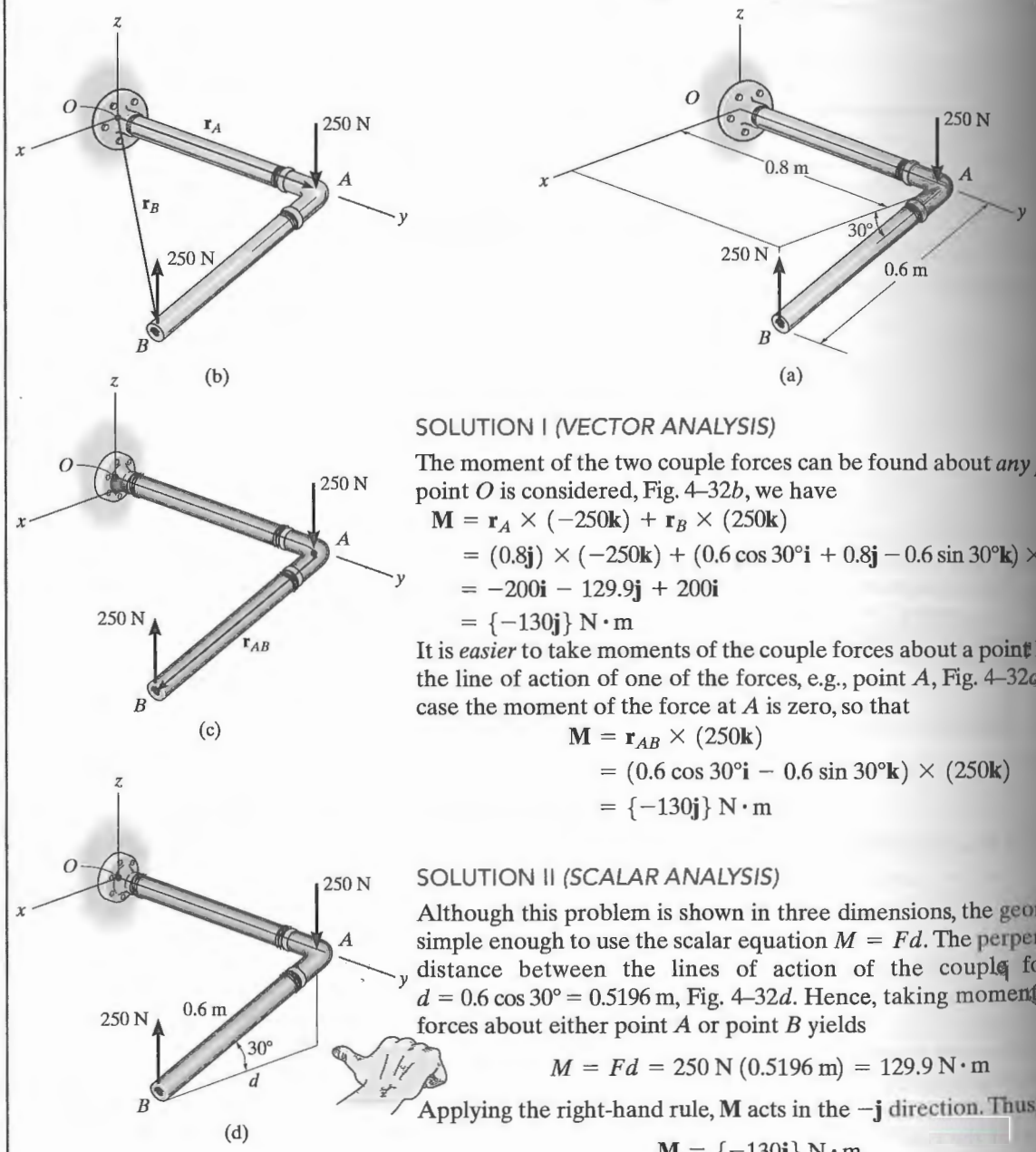


Fig. 4–32

SOLUTION I (*VECTOR ANALYSIS*)

The moment of the two couple forces can be found about *any point*. If point O is considered, Fig. 4–32*b*, we have

$$\begin{aligned}\mathbf{M} &= \mathbf{r}_A \times (-250\mathbf{k}) + \mathbf{r}_B \times (250\mathbf{k})\\ &= (0.8\mathbf{j}) \times (-250\mathbf{k}) + (0.6\cos 30°\mathbf{i} + 0.8\mathbf{j} - 0.6\sin 30°\mathbf{k}) \times (250\mathbf{k})\\ &= -200\mathbf{i} - 129.9\mathbf{j} + 200\mathbf{i}\\ &= \{-130\mathbf{j}\}\ \mathrm{N\cdot m}\end{aligned}$$ *Ans.*

It is *easier* to take moments of the couple forces about a point lying on the line of action of one of the forces, e.g., point A, Fig. 4–32*c*. In this case the moment of the force at A is zero, so that

$$\begin{aligned}\mathbf{M} &= \mathbf{r}_{AB} \times (250\mathbf{k})\\ &= (0.6\cos 30°\mathbf{i} - 0.6\sin 30°\mathbf{k}) \times (250\mathbf{k})\\ &= \{-130\mathbf{j}\}\ \mathrm{N\cdot m}\end{aligned}$$ *Ans.*

SOLUTION II (*SCALAR ANALYSIS*)

Although this problem is shown in three dimensions, the geometry is simple enough to use the scalar equation $M = Fd$. The perpendicular distance between the lines of action of the couple forces is $d = 0.6\cos 30° = 0.5196$ m, Fig. 4–32*d*. Hence, taking moments of the forces about either point A or point B yields

$$M = Fd = 250\ \mathrm{N}\,(0.5196\ \mathrm{m}) = 129.9\ \mathrm{N\cdot m}$$

Applying the right-hand rule, $\mathbf{M}$ acts in the $-\mathbf{j}$ direction. Thus,

$$\mathbf{M} = \{-130\mathbf{j}\}\ \mathrm{N\cdot m}$$ *Ans.*

ching exclusively for The University of Melbourne

EXAMPLE 4.13

Replace the two couples acting on the pipe column in Fig. 4–33*a* by a resultant couple moment.

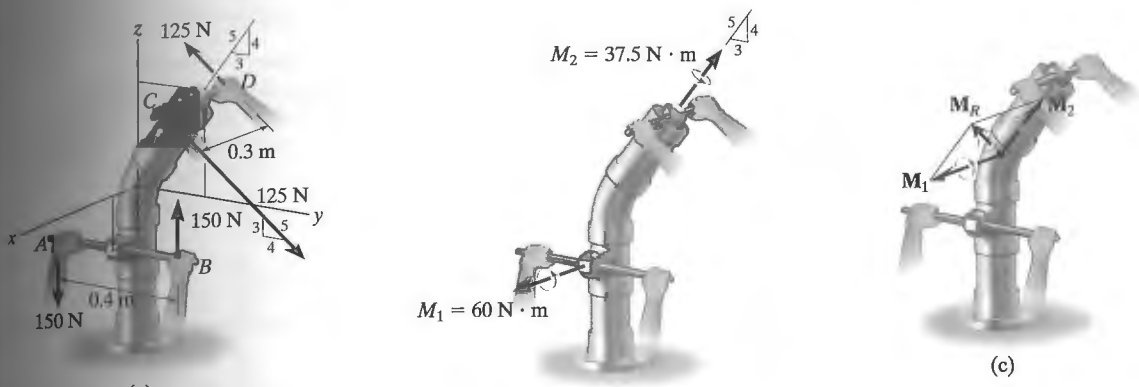


Fig. 4–33

SOLUTION (VECTOR ANALYSIS)

The couple moment $\mathbf{M}_1$, developed by the forces at A and B, can easily be determined from a scalar formulation.

$$M_1 = Fd = 150\,\text{N}(0.4\,\text{m}) = 60\,\text{N}\cdot\text{m}$$

By the right-hand rule, $\mathbf{M}_1$ acts in the $+\mathbf{i}$ direction, Fig. 4–33*b*. Hence,

$$\mathbf{M}_1 = \{60\mathbf{i}\}\,\text{N}\cdot\text{m}$$

Vector analysis will be used to determine $\mathbf{M}_2$, caused by forces at C and D. If moments are computed about point D, Fig. 4–33*a*, $\mathbf{M}_2 = \mathbf{r}_{DC} \times \mathbf{F}_C$, then

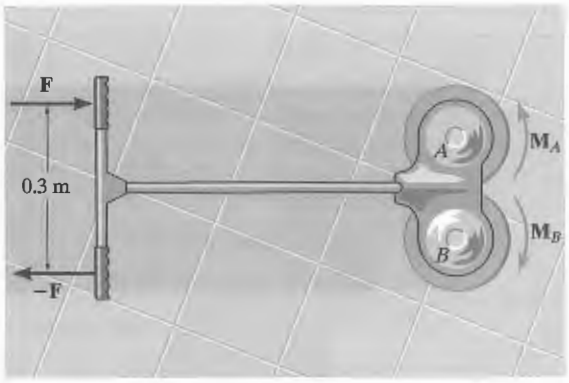
$$\begin{aligned}\mathbf{M}_2 &= \mathbf{r}_{DC} \times \mathbf{F}_C = (0.3\mathbf{i}) \times \left[125\left(\tfrac{4}{5}\right)\mathbf{j} - 125\left(\tfrac{3}{5}\right)\mathbf{k}\right]\\ &= (0.3\mathbf{i}) \times [100\mathbf{j} - 75\mathbf{k}] = 30(\mathbf{i} \times \mathbf{j}) - 22.5(\mathbf{i} \times \mathbf{k})\\ &= \{22.5\mathbf{j} + 30\mathbf{k}\}\,\text{N}\cdot\text{m}\end{aligned}$$

Since $\mathbf{M}_1$ and $\mathbf{M}_2$ are free vectors, they may be moved to some arbitrary point and added vectorially, Fig. 4–33*c*. The resultant couple moment becomes

$$\mathbf{M}_R = \mathbf{M}_1 + \mathbf{M}_2 = \{60\mathbf{i} + 22.5\mathbf{j} + 30\mathbf{k}\}\,\text{N}\cdot\text{m} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

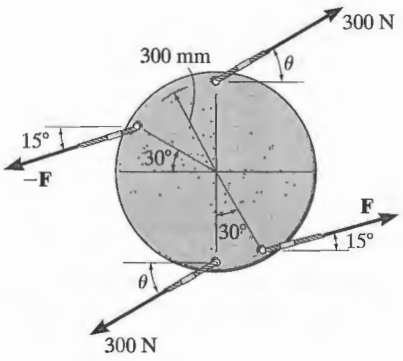
•4–77. The floor causes a couple moment of $M_A = 40\ \text{N}\cdot\text{m}$ and $M_B = 30\ \text{N}\cdot\text{m}$ on the brushes of the polishing machine. Determine the magnitude of the couple forces that must be developed by the operator on the handles so that the resultant couple moment on the polisher is zero. What is the magnitude of these forces if the brush at B suddenly stops so that $M_B = 0$?



Prob. 4–77

4–78. If $\theta = 30°$, determine the magnitude of force **F** so that the resultant couple moment is $100\ \text{N}\cdot\text{m}$, clockwise.

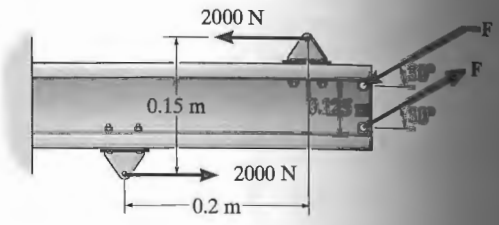
4–79. If $F = 200$ N, determine the required angle θ so that the resultant couple moment is zero.



Probs. 4–78/79

***4–80.** Two couples act on the beam. Determine the magnitude of **F** so that the resultant couple moment is $450\ \text{N}\cdot\text{m}$ counterclockwise. Where on the beam does the resultant couple moment act?

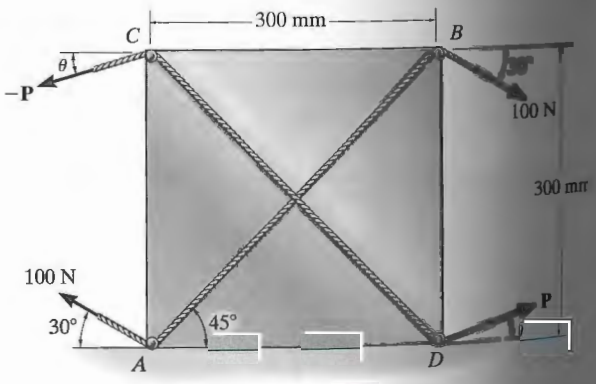
Video Solutions



Prob. 4–80

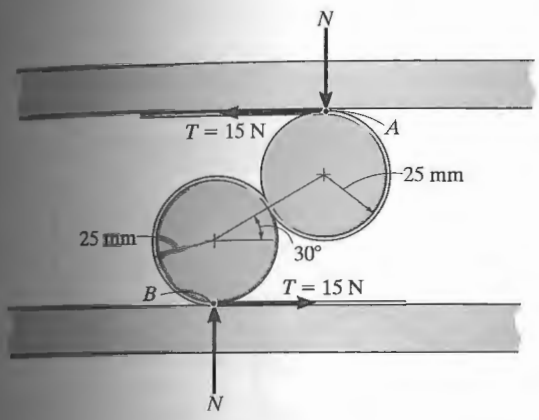
•4–81. The cord passing over the two small pegs A and B of the square board is subjected to a tension of 100 N. Determine the required tension P acting on the cord that passes over pegs C and D so that the resultant couple produced by the two couples is $15\ \text{N}\cdot\text{m}$ acting clockwise. Take $\theta = 15°$.

4–82. The cord passing over the two small pegs A and B of the board is subjected to a tension of 100 N. Determine the *minimum* tension P and the orientation θ of the cord passing over pegs C and D, so that the resultant couple moment produced by the two cords is $20\ \text{N}\cdot\text{m}$, clockwise.



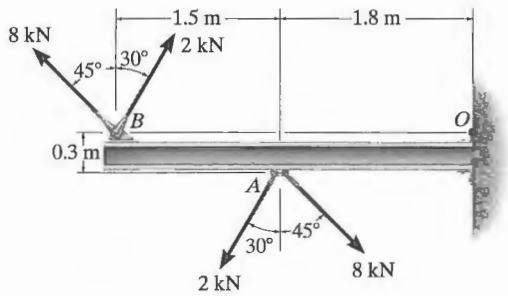
Probs. 4–81/82

4–83. A device called a rolamite is used in various ways to replace slipping motion with rolling motion. If the belt, which wraps between the rollers, is subjected to a tension of 15 N, determine the reactive forces N of the top and bottom plates on the rollers so that the resultant couple acting on the rollers is equal to zero.



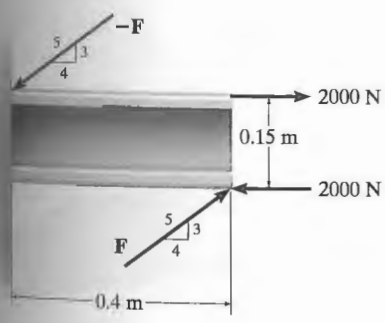
Prob. 4–83

•4–85. Determine the resultant couple moment acting on the beam. Solve the problem two ways: (a) sum moments about point O; and (b) sum moments about point A.



Prob. 4–85

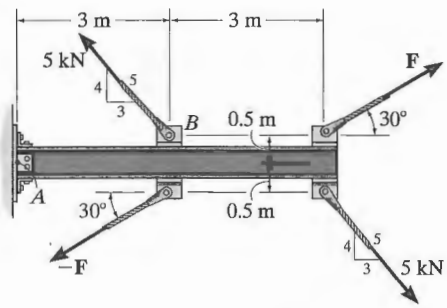
4–84. Two couples act on the beam as shown. Determine the magnitude of **F** so that the resultant couple moment is 300 N · m counterclockwise. Where on the beam does the resultant couple act?



Prob. 4–84

4–86. Two couples act on the cantilever beam. If $F = 6$ kN, determine the resultant couple moment.

4–87. Determine the required magnitude of force **F**, if the resultant couple moment on the beam is to be zero.

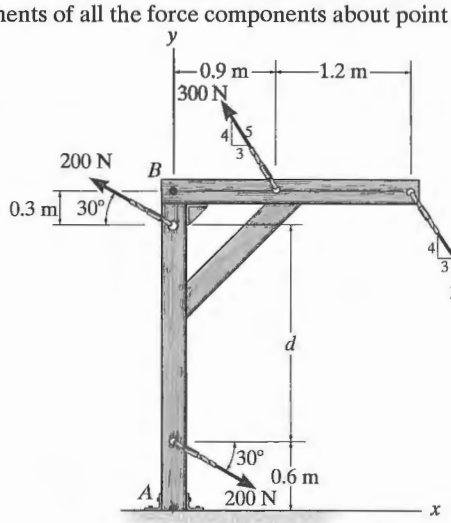


Probs. 4–86/87

***4–88.** Two couples act on the frame. If the resultant couple moment is to be zero, determine the distance d between the 200-N couple forces.

•4–89. Two couples act on the frame. If $d = 1.2$ m, determine the resultant couple moment. Compute the result by resolving each force into x and y components and (a) finding the moment of each couple (Eq. 4–13) and (b) summing the moments of all the force components about point A.

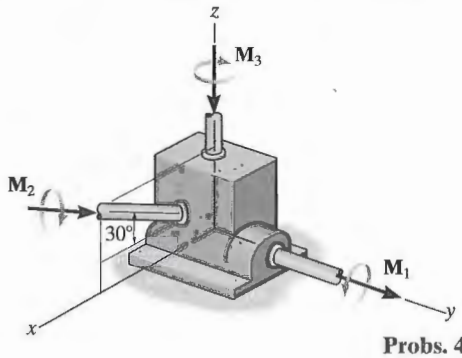
4–90. Two couples act on the frame. If $d = 1.2$ m, determine the resultant couple moment. Compute the result by resolving each force into x and y components and (a) finding the moment of each couple (Eq. 4–13) and (b) summing the moments of all the force components about point B.



Probs. 4–88/89/90

4–91. If $M_1 = 500$ N·m, $M_2 = 600$ N·m, and $M_3 = 450$ N·m, determine the magnitude and coordinate direction angles of the resultant couple moment.

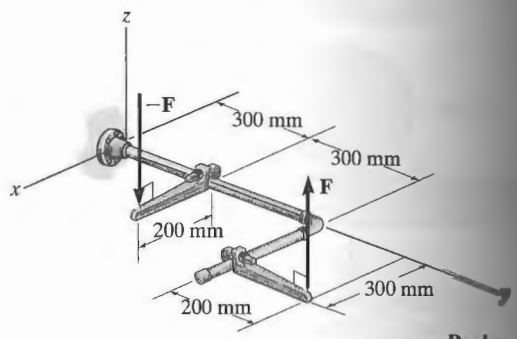
***4–92.** Determine the required magnitude of couple moments $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$ so that the resultant couple moment is $\mathbf{M}_R = \{-300\mathbf{i} + 450\mathbf{j} - 600\mathbf{k}\}$ N·m.



Probs. 4–91/92

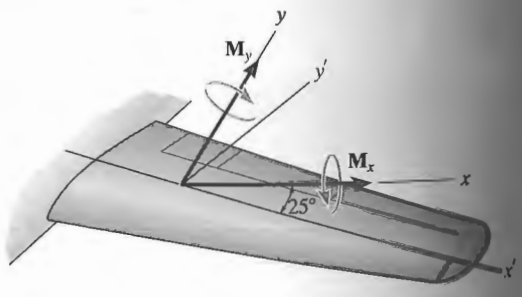
•4–93. If $F = 80$ N, determine the magnitude and coordinate direction angles of the couple moment. The pipe assembly lies in the x–y plane.

4–94. If the magnitude of the couple moment acting on the pipe assembly is 50 N·m, determine the magnitude of the couple forces applied to each wrench. The pipe assembly lies in the x–y plane.



Probs. 4–93/94

4–95. From load calculations it is determined that the wing is subjected to couple moments $M_x = 25.5$ kN·m and $M_y = 37.5$ kN·m. Determine the resultant couple moments created about the x' and y' axes. The axes all lie in the same horizontal plane.

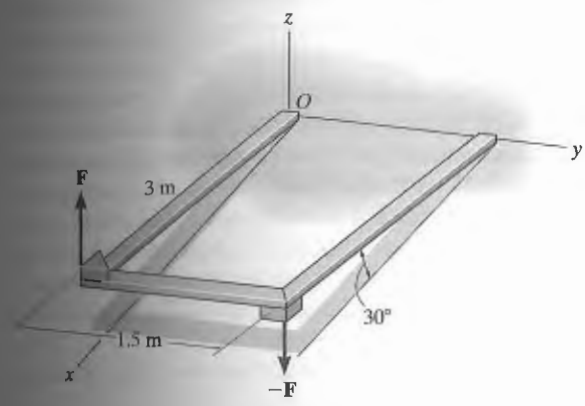


Prob. 4–95

[redacted] exclusively for The University of Melbourne

4–96. Express the moment of the couple acting on the f[redacted]rtesian vector form. The forces are applied perpendi[redacted]to the frame. What is the magnitude of the couple moment? Take $F = 50$ N.

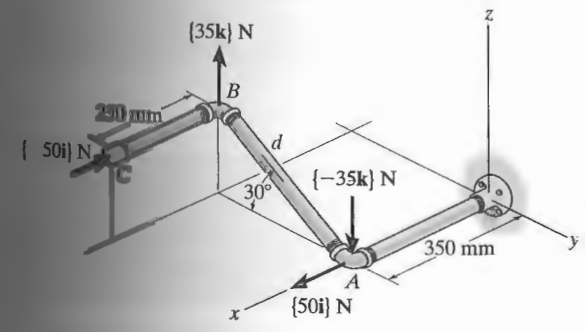
4–97. In order to turn over the frame, a couple moment is applied as shown. If the component of this couple moment along the x axis is $\mathbf{M}_x = \{-20\mathbf{i}\}$ N·m, determine the magnitude F of the couple forces.



Probs. 4–96/97

4–98. Determine the resultant couple moment of the two couples that act on the pipe assembly. The distance from A to B is $d = 400$ mm. Express the result as a Cartesian vector.

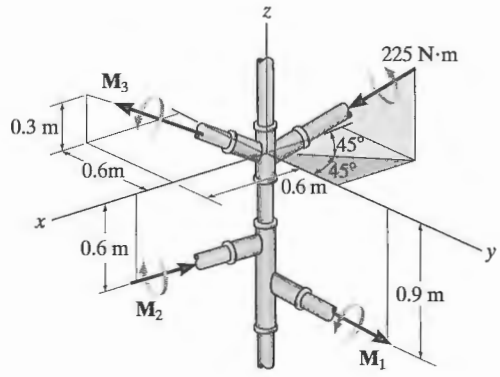
4–99. Determine the distance d between A and B so that the resultant couple moment has a magnitude of $M_R = 20$ N·m.



Probs. 4–98/99

***4–100.** If $M_1 = 270$ N·m, $M_2 = 135$ N·m, and $M_3 = 180$ N·m, determine the magnitude and coordinate direction angles of the resultant couple moment.

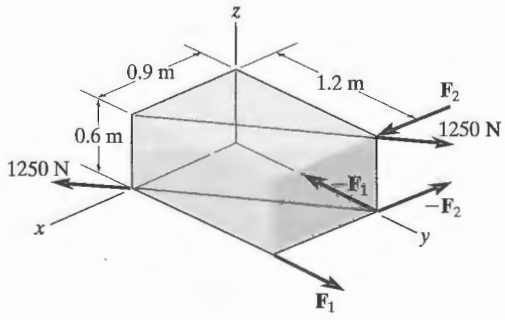
•4–101. Determine the magnitudes of couple moments $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$ so that the resultant couple moment is zero.



Probs. 4–100/101

4–102. If $F_1 = 500$ N and $F_2 = 1000$ N, determine the magnitude and coordinate direction angles of the resultant couple moment.

4–103. Determine the magnitude of couple forces $\mathbf{F}_1$ and $\mathbf{F}_2$ so that the resultant couple moment acting on the block is zero.



Probs. 4–102/103

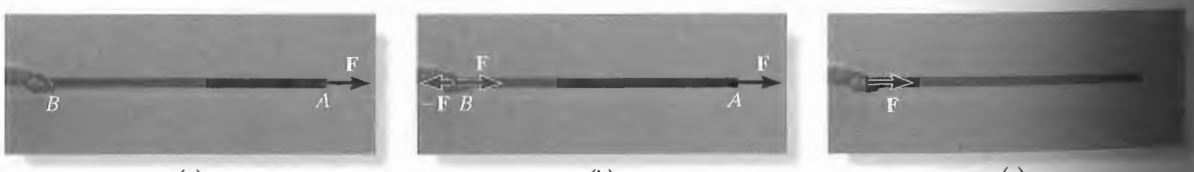
Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

4.7 Simplification of a Force and Couple System

Sometimes it is convenient to reduce a system of forces and couple moments acting on a body to a simpler form by replacing it with an *equivalent system*, consisting of a single resultant force acting at a specific point and a resultant couple moment. A system is equivalent if the *external effects* it produces on a body are the same as those caused by the original force and couple moment system. In this context, the external effects of a system refer to the *translating and rotating motion* of the body if the body is free to move, or it refers to the *reactive forces* at the supports if the body is held fixed.

For example, consider holding the stick in Fig. 4–34*a* which is subjected to the force **F** at point *A*. If we attach a pair of equal but opposite forces **F** and –**F** at point *B*, which is *on the line of action* of **F**, Fig. 4–34*b*, we observe that –**F** at *B* and **F** at *A* will cancel each other, leaving only **F** at *B*, Fig. 4–34*c*. Force **F** has now been moved from *A* to *B* without modifying its *external effects* on the stick; i.e., the reaction at the grip remains the same. This demonstrates the *principle of transmissibility*, which states that a force acting on a body (stick) is a *sliding vector* since it can be applied at any point along its line of action.

We can also use the above procedure to move a force to a point that is *not* on the line of action of the force. If **F** is applied perpendicular to the stick, as in Fig. 4–35*a*, then we can attach a pair of equal but opposite forces **F** and –**F** to *B*, Fig. 4–35*b*. Force **F** is now applied at *B*, and the other two forces **F** at *A* and –**F** at *B*, form a couple that produces the couple moment $M = Fd$, Fig. 4–35*c*. Therefore, the force **F** can be moved from *A* to *B* provided a couple moment **M** is added to maintain an equivalent system. This couple moment is determined by taking the moment of **F** about *B*. Since **M** is actually a *free vector*, it can act at any point on the stick. In both cases the systems are equivalent which causes a downward force **F** and clockwise couple moment $M = Fd$ to be felt at the grip.



(a) (b) (c)

Fig. 4–34

(a) (b) (c)

Fig. 4–35

System of Forces and Couple Moments. Using the above method, a system of several forces and couple moments acting on a body can be reduced to an equivalent single resultant force acting at a point O and a resultant couple moment. For example, in Fig. 4–36*a*, O is not on the line of action of $\mathbf{F}_1$, and so this force can be moved to point O provided a couple moment $\mathbf{M}_1 = \mathbf{r}_1 \times \mathbf{F}$ is added to the body. Similarly, the couple moment $\mathbf{M}_2 = \mathbf{r}_2 \times \mathbf{F}_2$ should be added to the body when we move $\mathbf{F}_2$ to point O. Finally, since the couple moment $\mathbf{M}$ is a free vector, it can just be moved to point O. By doing this, we obtain the equivalent system shown in Fig. 4–36*b*, which produces the same external effects (support reactions) on the body as that of the force and couple system shown in Fig. 4–36*a*. If we sum the forces and couple moments, we obtain the resultant force $\mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2$ and the resultant couple moment $(\mathbf{M}_R)_O = \mathbf{M} + \mathbf{M}_1 + \mathbf{M}_2$, Fig. 4–36*c*.

Notice that $\mathbf{F}_R$ is independent of the location of point O; however, $(\mathbf{M}_R)_O$ depends upon this location since the moments $\mathbf{M}_1$ and $\mathbf{M}_2$ are determined using the position vectors $\mathbf{r}_1$ and $\mathbf{r}_2$. Also note that $(\mathbf{M}_R)_O$ is a free vector and can act at *any point* on the body, although point O is generally chosen as its point of application.

We can generalize the above method of reducing a force and couple system to an equivalent resultant force $\mathbf{F}_R$ acting at point O and a resultant couple moment $(\mathbf{M}_R)_O$ by using the following two equations.

$$\boxed{\begin{aligned} \mathbf{F}_R &= \Sigma \mathbf{F} \\ (\mathbf{M}_R)_O &= \Sigma \mathbf{M}_O + \Sigma \mathbf{M} \end{aligned}} \qquad (4\text{–}17)$$

The first equation states that the resultant force of the system is equivalent to the sum of all the forces; and the second equation states that the resultant couple moment of the system is equivalent to the sum of all the couple moments $\Sigma \mathbf{M}$ plus the moments of all the forces $\Sigma \mathbf{M}_O$ about point O. If the force system lies in the x–y plane and any couple moments are perpendicular to this plane, then the above equations reduce to the following three scalar equations.

$$\boxed{\begin{aligned} (F_R)_x &= \Sigma F_x \\ (F_R)_y &= \Sigma F_y \\ (M_R)_O &= \Sigma M_O + \Sigma M \end{aligned}} \qquad (4\text{–}18)$$

Here the resultant force is determined from the vector sum of its two components $(F_R)_x$ and $(F_R)_y$.

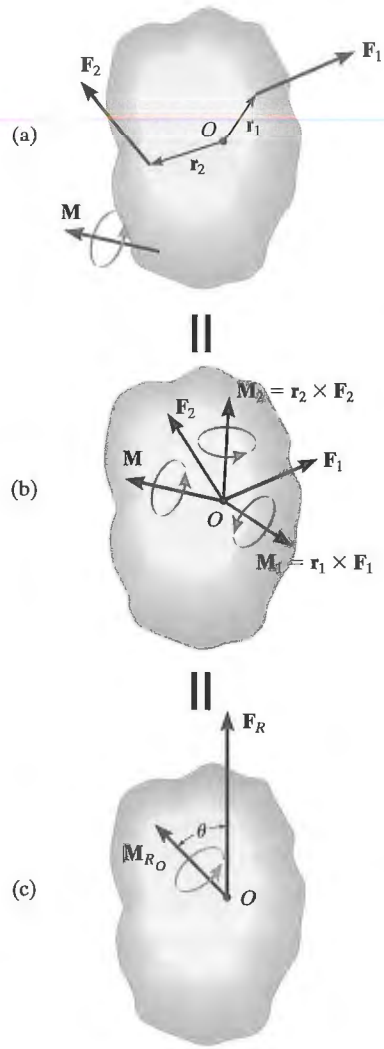


Fig. 4–36

ENGR1000
A publication from Pearson Custom Publishing

The weights of these traffic lights can be replaced by their equivalent resultant force $W_R = W_1 + W_2$ and a couple moment $(M_R)_O = W_1 d_1 + W_2 d_2$ at the support O. In both cases the support must provide the same resistance to translation and rotation in order to keep the member in the horizontal position.

4

Procedure for Analysis

The following points should be kept in mind when simplifying a force and couple moment system to an equivalent resultant force and couple system.

- Establish the coordinate axes with the origin located at point O and the axes having a selected orientation.

Force Summation.

- If the force system is *coplanar*, resolve each force into its x and y components. If a component is directed along the positive x or y axis, it represents a positive scalar; whereas if it is directed along the negative x or y axis, it is a negative scalar.
- In three dimensions, represent each force as a Cartesian vector before summing the forces.

Moment Summation.

- When determining the moments of a *coplanar* force system about point O, it is generally advantageous to use the principle of moments, i.e., determine the moments of the components of each force, rather than the moment of the force itself.
- In three dimensions use the vector cross product to determine the moment of each force about point O. Here the position vectors extend from O to any point on the line of action of the force.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

exclusively for The University of Melbourne

EXAMPLE 4.14

Replace the force and couple system shown in Fig. 4–37*a* by an equivalent resultant force and couple moment acting at point *O*.

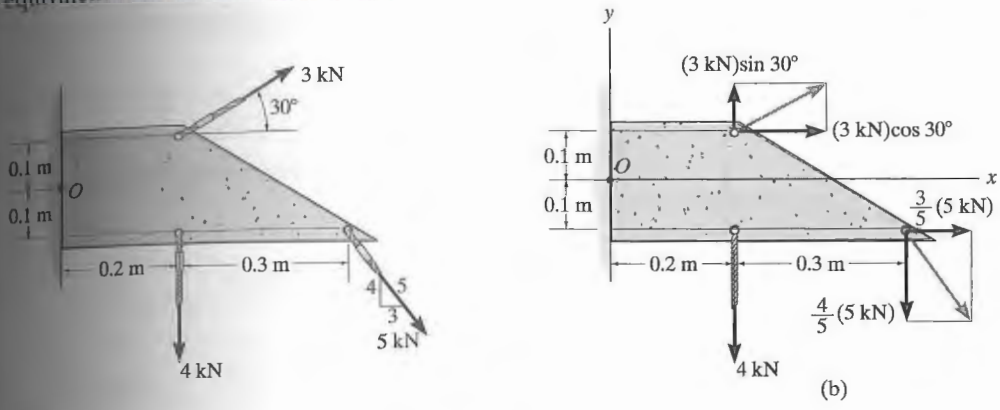


Fig. 4–37

SOLUTION

Force Summation. The 3 kN and 5 kN forces are resolved into their *x* and *y* components as shown in Fig. 4–37*b*. We have

$$\xrightarrow{+}(F_R)_x = \Sigma F_x; (F_R)_x = (3\text{ kN})\cos 30° + \left(\tfrac{3}{5}\right)(5\text{ kN}) = 5.598\text{ kN} \rightarrow$$

$$+\uparrow(F_R)_y = \Sigma F_y; (F_R)_y = (3\text{ kN})\sin 30° - \left(\tfrac{4}{5}\right)(5\text{ kN}) - 4\text{ kN} = -6.50\text{ kN} = 6.50\text{ kN}\downarrow$$

Using the Pythagorean theorem, Fig. 4–37c, the magnitude of $\mathbf{F}_R$ is

$$F_R = \sqrt{(F_{R_x})^2 + (F_{R_y})^2} = \sqrt{(5.598\text{ kN}^2) + (6.50\text{ kN}^2)} = 8.58\text{ kN} \quad \textit{Ans.}$$

Its direction θ is

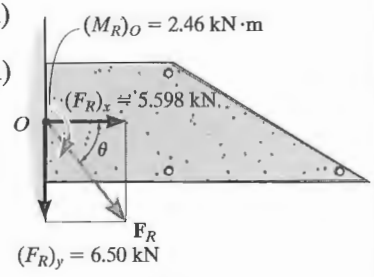
$$\theta = \tan^{-1}\left(\frac{(F_R)_y}{(F_R)_x}\right) = \tan^{-1}\left(\frac{6.50\text{ kN}}{5.598\text{ kN}}\right) = 49.3° \quad \textit{Ans.}$$

Moment Summation. The moments of 3 kN and 5 kN about point *O* will be determined using their *x* and *y* components. Referring to Fig. 4–37*b*, we have

$$\circlearrowleft + (M_R)_O = \Sigma M_O;$$

$$(M_R)_O = (3\text{ kN})\sin 30°(0.2\text{ m}) - (3\text{ kN})\cos 30°(0.1\text{ m}) + \left(\tfrac{3}{5}\right)(5\text{ kN})(0.1\text{ m})$$
$$- \left(\tfrac{4}{5}\right)(5\text{ kN})(0.5\text{ m}) - (4\text{ kN})(0.2\text{ m})$$
$$= -2.46\text{ kN}\cdot\text{m} = 2.46\text{ kN}\cdot\text{m}\circlearrowright \quad \textit{Ans.}$$

This clockwise moment is shown in Fig. 4–37*c*.



NOTE: Realize that the resultant force and couple moment in Fig. 4–37*c* will produce the same external effects or reactions at the supports as those produced by the force system, Fig 4–37*a*.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 4.15

Replace the force and couple system acting on the member in Fig. 4–38*a* by an equivalent resultant force and couple moment acting at point *O*.

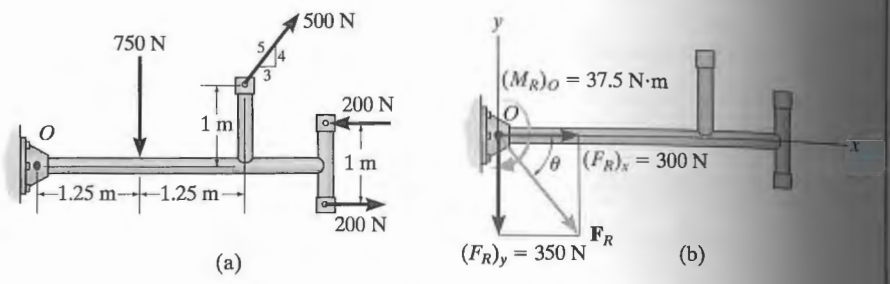


Fig. 4–38

SOLUTION

Force Summation. Since the couple forces of 200 N are equal but opposite, they produce a zero resultant force, and so it is not necessary to consider them in the force summation. The 500-N force is resolved into its x and y components, thus,

$$\overset{+}{\rightarrow} (F_R)_x = \Sigma F_x;\ (F_R)_x = \left(\tfrac{3}{5}\right)(500\text{ N}) = 300\text{ N} \rightarrow$$

$$+\uparrow (F_R)_y = \Sigma F_y;\ (F_R)_y = (500\text{ N})\left(\tfrac{4}{5}\right) - 750\text{ N} = -350\text{ N} = 350\text{ N}\downarrow$$

From Fig. 4–15*b*, the magnitude of $\mathbf{F}_R$ is

$$F_R = \sqrt{(F_R)_x^2 + (F_R)_y^2}$$

$$= \sqrt{(300\text{ N})^2 + (350\text{ N})^2} = 461\text{ N} \qquad \textit{Ans.}$$

And the angle θ is

$$\theta = \tan^{-1}\left(\frac{(F_R)_y}{(F_R)_x}\right) = \tan^{-1}\left(\frac{350\text{ N}}{300\text{ N}}\right) = 49.4^\circ \qquad \textit{Ans.}$$

Moment Summation. Since the couple moment is a free vector, it can act at any point on the member. Referring to Fig. 4–38*a*, we have

$$\circlearrowleft + (M_R)_O = \Sigma M_O + \Sigma M_c;$$

$$(M_R)_O = (500\text{ N})\left(\tfrac{4}{5}\right)(2.5\text{ m}) - (500\text{ N})\left(\tfrac{3}{5}\right)(1\text{ m})$$

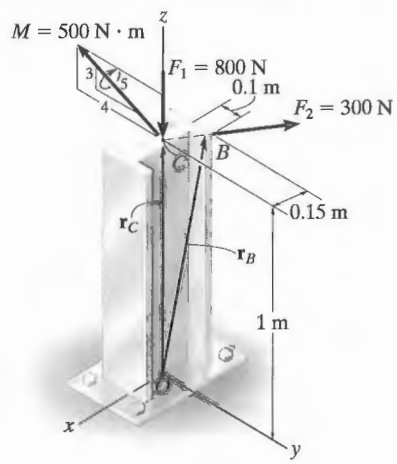
$$- (750\text{ N})(1.25\text{ m}) + 200\text{ N}\cdot\text{m}$$

$$= -37.5\text{ N}\cdot\text{m} = 37.5\text{ N}\cdot\text{m}\circlearrowright \qquad \textit{Ans.}$$

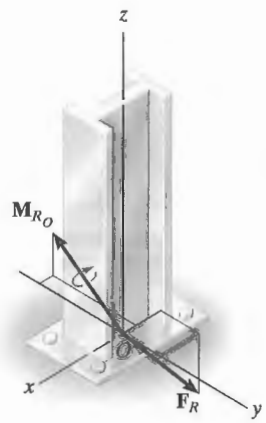
This clockwise moment is shown in Fig. 4–38*b*.

EXAMPLE 4.16

The structural member is subjected to a couple moment **M** and forces $\mathbf{F}_1$ and $\mathbf{F}_2$ in Fig. 4–39*a*. Replace this system by an equivalent resultant force and couple moment acting at its base, point *O*.



(a)



(b)

Fig. 4–39

SOLUTION (VECTOR ANALYSIS)

The [redacted] dimensional aspects of the problem can be simplified by using a Cartesian vector analysis. Expressing the forces and couple moment as Cartesian vectors, we have

$$\mathbf{F}_1 = \{-800\mathbf{k}\}\ \text{N}$$

$$\mathbf{F}_2 = (300\ \text{N})\mathbf{u}_{CB}$$

$$= (300\ \text{N})\left(\frac{\mathbf{r}_{CB}}{r_{CB}}\right)$$

$$= 300\ \text{N}\left[\frac{\{-0.15\mathbf{i} + 0.1\mathbf{j}\}\ \text{m}}{\sqrt{(-0.15\ \text{m})^2 + (0.1\ \text{m})^2}}\right] = \{-249.6\mathbf{i} + 166.4\mathbf{j}\}\ \text{N}$$

$$\mathbf{M} = -500\left(\tfrac{4}{5}\right)\mathbf{j} + 500\left(\tfrac{3}{5}\right)\mathbf{k} = \{-400\mathbf{j} + 300\mathbf{k}\}\ \text{N}\cdot\text{m}$$

Force Summation.

$$\mathbf{F}_R = \Sigma\mathbf{F}; \qquad \mathbf{F}_R = \mathbf{F}_1 + \mathbf{F}_2 = -800\mathbf{k} - 249.6\mathbf{i} + 166.4\mathbf{j}$$

$$= \{-250\mathbf{i} + 166\mathbf{j} - 800\mathbf{k}\}\ \text{N} \qquad \textit{Ans.}$$

Moment Summation.

$$\mathbf{M}_{R_O} = \Sigma\mathbf{M} + \Sigma\mathbf{M}_O$$

$$\mathbf{M}_{R_O} = \mathbf{M} + \mathbf{r}_C \times \mathbf{F}_1 + \mathbf{r}_B \times \mathbf{F}_2$$

$$\mathbf{M}_{R_O} = (-400\mathbf{j} + 300\mathbf{k}) + (1\mathbf{k}) \times (-800\mathbf{k}) + \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -0.15 & 0.1 & 1 \\ -249.6 & 166.4 & 0 \end{vmatrix}$$

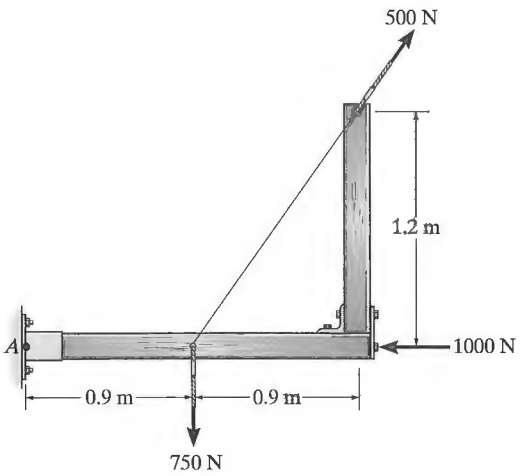
$$= (-400\mathbf{j} + 300\mathbf{k}) + (\mathbf{0}) + (-166.4\mathbf{i} - 249.6\mathbf{j})$$

$$= \{-166\mathbf{i} - 650\mathbf{j} + 300\mathbf{k}\}\ \text{N}\cdot\text{m} \qquad \textit{Ans.}$$

The results are shown in Fig. 4–39*b*.

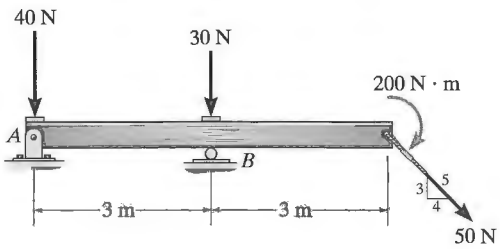
FUNDAMENTAL PROBLEMS

F4–25. Replace the loading system by an equivalent resultant force and couple moment acting at point A.



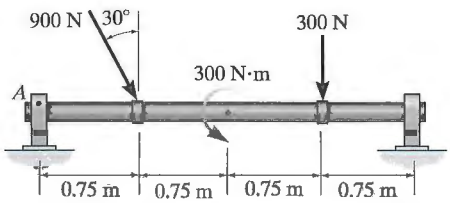
F4–25

F4–26. Replace the loading system by an equivalent resultant force and couple moment acting at point A.



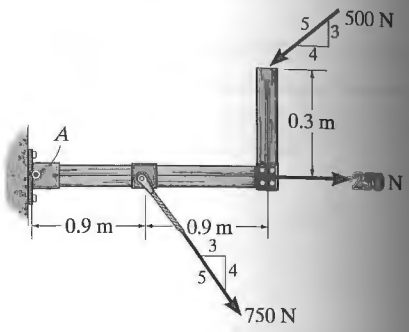
F4–26

F4–27. Replace the loading system by an equivalent resultant force and couple moment acting at point A.



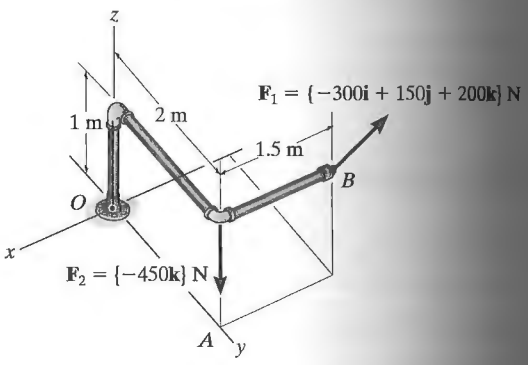
F4–27

F4–28. Replace the loading system by an equivalent resultant force and couple moment acting at point A.



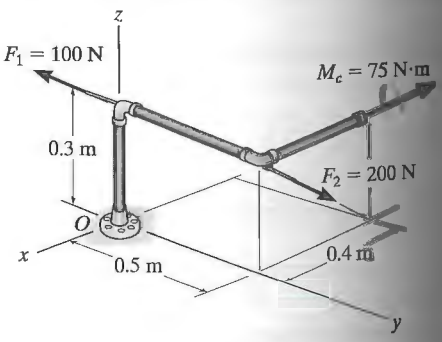
F4–28

F4–29. Replace the loading system by an equivalent resultant force and couple moment acting at point O.



F4–29

F4–30. Replace the loading system by an equivalent resultant force and couple moment acting at point O.



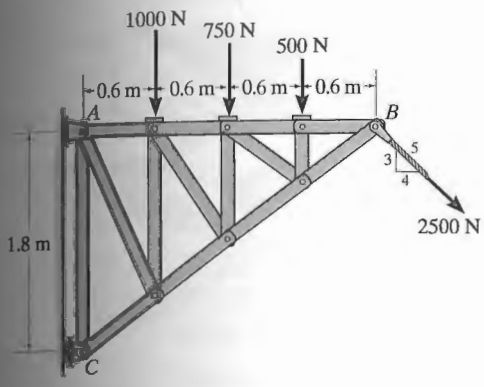
F4–30

...ing exclusively for The University of Melbourne

ENGINEERING
Customized coaching is available!

PROBLEMS

Replace the force system acting on the truss by a resultant force and couple moment at point C.

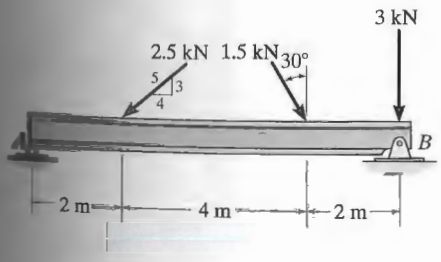


Prob. 4–104

•4–105. Replace the force system acting on the beam by an equivalent force and couple moment at point A.

Video Solutions

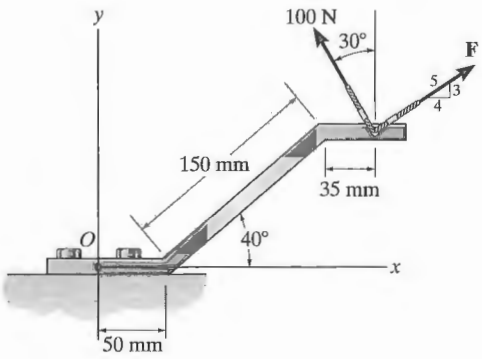
4–106. Replace the force system acting on the beam by an equivalent force and couple moment at point B.



Probs. 4–105/106

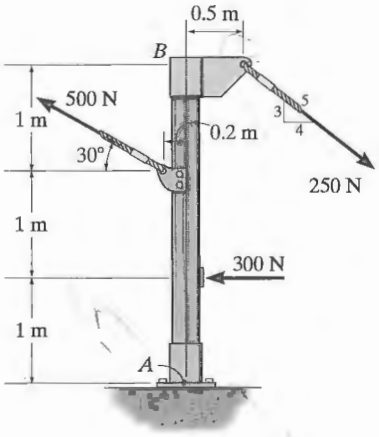
4–107. Replace the two forces by an equivalent resultant force and couple moment at point O. Set $F = 100$ N.

***4–108.** Replace the two forces by an equivalent resultant force and couple moment at point O. Set $F = 75$ N.



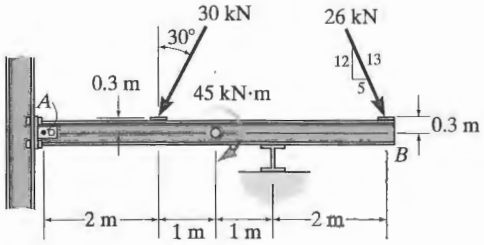
Probs. 4–107/108

•4–109. Replace the force system acting on the post by a resultant force and couple moment at point A.



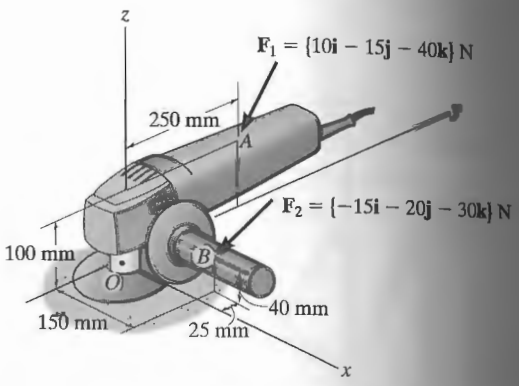
Prob. 4–109

4–110. Replace the force and couple moment system acting on the overhang beam by a resultant force and couple moment at point A.



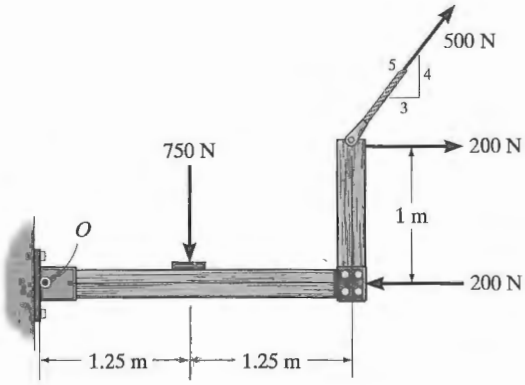
Prob. 4–110

***4–112.** Replace the two forces acting on the grinder by a resultant force and couple moment at point O. Express the results in Cartesian vector form.



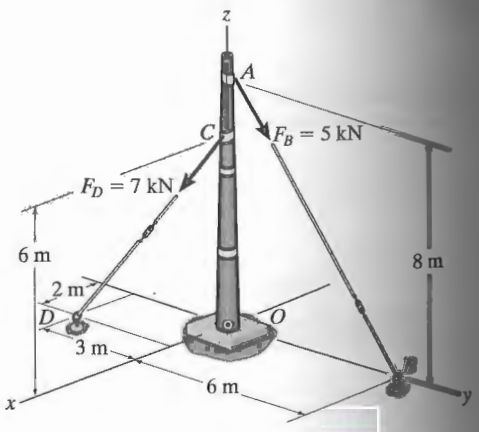
Prob. 4–112

4–111. Replace the force system by a resultant force and couple moment at point O.



Prob. 4–111

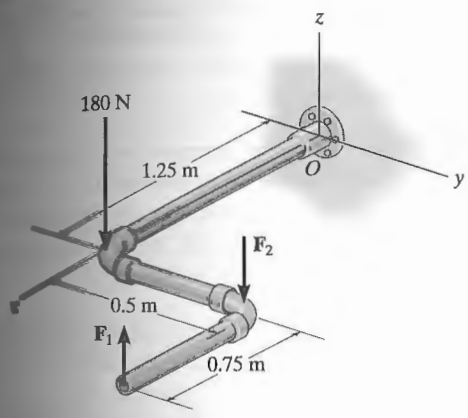
•4–113. Replace the two forces acting on the post by a resultant force and couple moment at point O. Express the results in Cartesian vector form.



Prob. 4–113

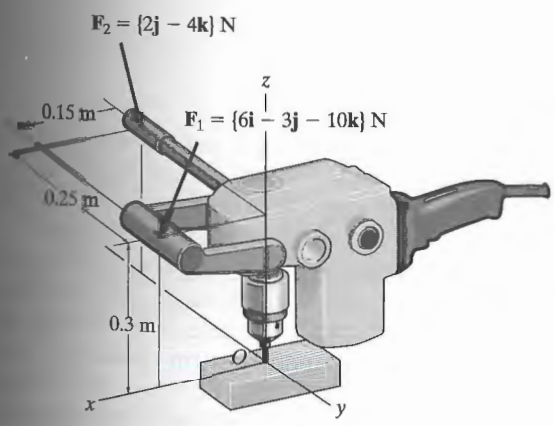
4–114. The three forces act on the pipe assembly. If $F_1 = 50$ N and $F_2 = 80$ N, replace this force system by an equivalent resultant force and couple moment acting at O. Express the results in Cartesian vector form.

Video Solutions



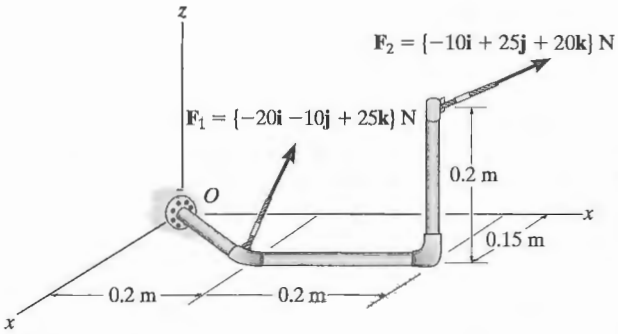
Prob. 4–114

4–115. Handle forces $\mathbf{F}_1$ and $\mathbf{F}_2$ are applied to the electric drill. Replace this force system by an equivalent resultant force and couple moment acting at point O. Express the results in Cartesian vector form.



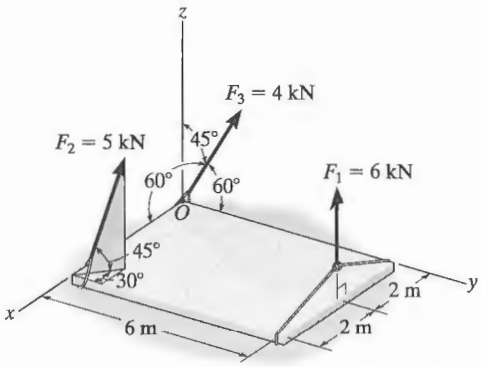
Prob. 4–115

***4–116.** Replace the force system acting on the pipe assembly by a resultant force and couple moment at point O. Express the results in Cartesian vector form.



Prob. 4–116

•4–117. The slab is to be hoisted using the three slings shown. Replace the system of forces acting on the slings by an equivalent force and couple moment at point O. The force $\mathbf{F}_1$ is vertical.



Prob. 4–117

ENGR1000
A publication from Pearson Custom Publishing

4.8 Further Simplification of a Force and Couple System

In the preceding section, we developed a way to reduce a force and couple moment system acting on a rigid body into an equivalent resultant force $\mathbf{F}_R$ acting at a specific point O and a resultant couple moment $(\mathbf{M}_R)_O$. The force system can be further reduced to an equivalent single resultant force provided the lines of action of $\mathbf{F}_R$ and $(\mathbf{M}_R)_O$ are *perpendicular* to each other. Because of this condition, only concurrent, coplanar and parallel force systems can be further simplified.

Concurrent Force System. Since a *concurrent force system* is one in which the lines of action of all the forces intersect at a common point O, Fig. 4–40*a*, then the force system produces no moment about this point. As a result, the equivalent system can be represented by a single resultant force $\mathbf{F}_R = \Sigma\mathbf{F}$ acting at O, Fig. 4–40*b*.

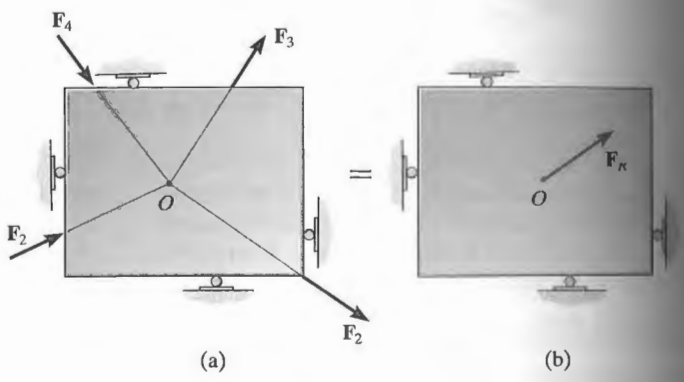


Fig. 4–40

Coplanar Force System. In the case of a *coplanar force system*, the lines of action of all the forces lie in the same plane, Fig. 4–41*a*, and so the resultant force $\mathbf{F}_R = \Sigma\mathbf{F}$ of this system also lies in this plane. Furthermore, the moment of each of the forces about any point O is directed perpendicular to this plane. Thus, the resultant moment $(\mathbf{M}_R)_O$ and resultant force $\mathbf{F}_R$ will be *mutually perpendicular*, Fig. 4–41*b*. The resultant moment can be replaced by moving the resultant force $\mathbf{F}_R$ a perpendicular or moment arm distance d away from point O such that $\mathbf{F}_R$ produces the *same moment* $(\mathbf{M}_R)_O$ about point O, Fig. 4–41*c*. This distance d can be determined from the scalar equation $(M_R)_O = F_R d = \Sigma M_O$ or $d = (M_R)_O/F_R$.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

Pearson Custom Publishing exclusively for The University of Melbourne

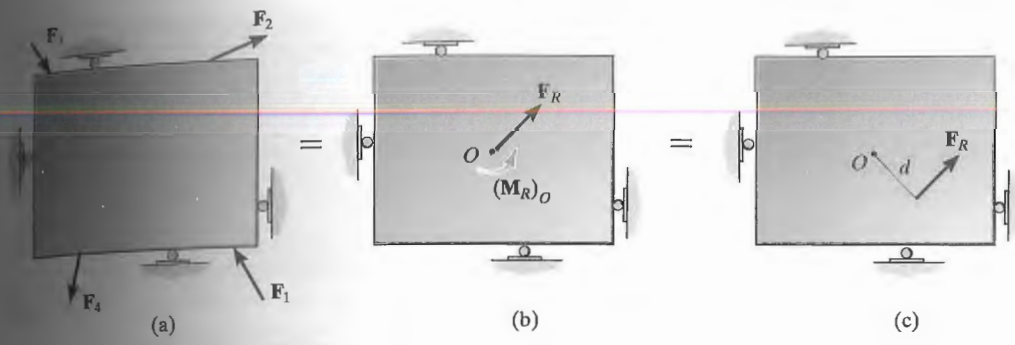


Fig. 4–41

Parallel Force System. The *parallel force system* shown in Fig. 4–42*a* consists of forces that are all parallel to the z axis. Thus, the resultant force $\mathbf{F}_R = \Sigma\mathbf{F}$ at point O must also be parallel to this axis, Fig. 4–42*b*. The moment produced by each force lies in the plane of the plate, and so the resultant couple moment, $(\mathbf{M}_R)_O$, will also lie in this plane, along the moment axis a since $\mathbf{F}_R$ and $(\mathbf{M}_R)_O$ are mutually perpendicular. As a result, the force system can be further reduced to an equivalent single resultant force $\mathbf{F}_R$, acting through point P located on the perpendicular b axis, Fig. 4–42*c*. The distance d along this axis from point O requires $(M_R)_O = F_R d = \Sigma M_O$ or $d = \Sigma M_O / F_R$.

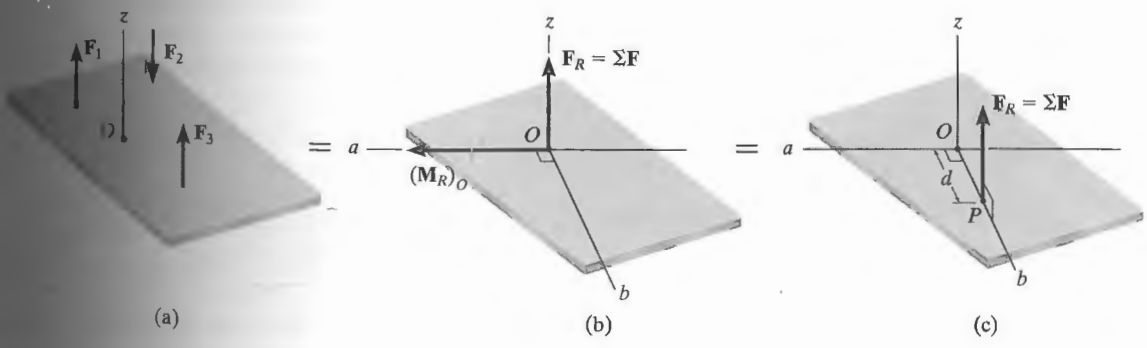


Fig. 4–42

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

4.8 Further Simplification of a Force and Couple System

In the preceding section, we developed a way to reduce a force and couple moment system acting on a rigid body into an equivalent resultant force $\mathbf{F}_R$ acting at a specific point O and a resultant couple moment $(\mathbf{M}_R)_O$. The force system can be further reduced to an equivalent single resultant force provided the lines of action of $\mathbf{F}_R$ and $(\mathbf{M}_R)_O$ are *perpendicular* to each other. Because of this condition, only concurrent, coplanar, and parallel force systems can be further simplified.

Concurrent Force System. Since a *concurrent force system* is one in which the lines of action of all the forces intersect at a common point O, Fig. 4–40*a*, then the force system produces no moment about this point. As a result, the equivalent system can be represented by a single resultant force $\mathbf{F}_R = \Sigma\mathbf{F}$ acting at O, Fig. 4–40*b*.

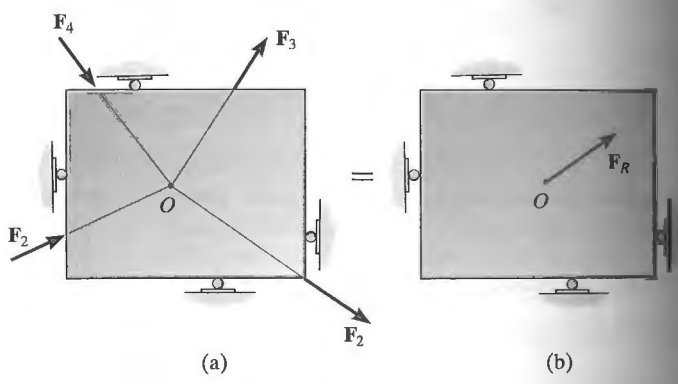


Fig. 4–40

Coplanar Force System. In the case of a *coplanar force system*, the lines of action of all the forces lie in the same plane, Fig. 4–41*a*, and so the resultant force $\mathbf{F}_R = \Sigma\mathbf{F}$ of this system also lies in this plane. Furthermore, the moment of each of the forces about any point O is directed perpendicular to this plane. Thus, the resultant moment $(\mathbf{M}_R)_O$ and resultant force $\mathbf{F}_R$ will be *mutually perpendicular*, Fig. 4–41*b*. The resultant moment can be replaced by moving the resultant force $\mathbf{F}_R$ a perpendicular or moment arm distance d away from point O such that $\mathbf{F}_R$ produces the *same moment* $(\mathbf{M}_R)_O$ about point O, Fig. 4–41*c*. This distance d can be determined from the scalar equation $(M_R)_O = F_R d = \Sigma M_O$ or $d = (M_R)_O/F_R$.

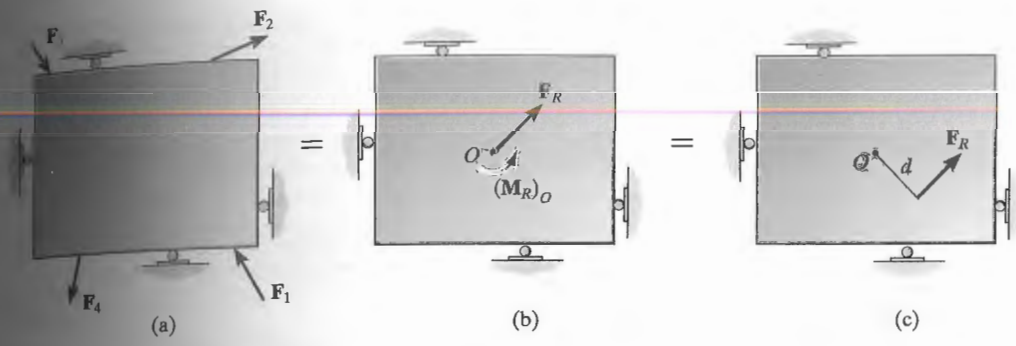


Fig. 4–41

Parallel Force System. The *parallel force system* shown in Fig. 4–42*a* consists of forces that are all parallel to the z axis. Thus, the resultant force $\mathbf{F}_R = \Sigma\mathbf{F}$ at point O must also be parallel to this axis, Fig. 4–42*b*. The moment produced by each force lies in the plane of the plate, and so the resultant couple moment, $(\mathbf{M}_R)_O$, will also lie in this plane, along the moment axis a since $\mathbf{F}_R$ and $(\mathbf{M}_R)_O$ are mutually perpendicular. As a result, the force system can be further reduced to an equivalent single resultant force $\mathbf{F}_R$, acting through point P located on the perpendicular b axis, Fig. 4–42*c*. The distance d along this axis from point O requires $(M_R)_O = F_R d = \Sigma M_O$ or $d = \Sigma M_O / F_R$.

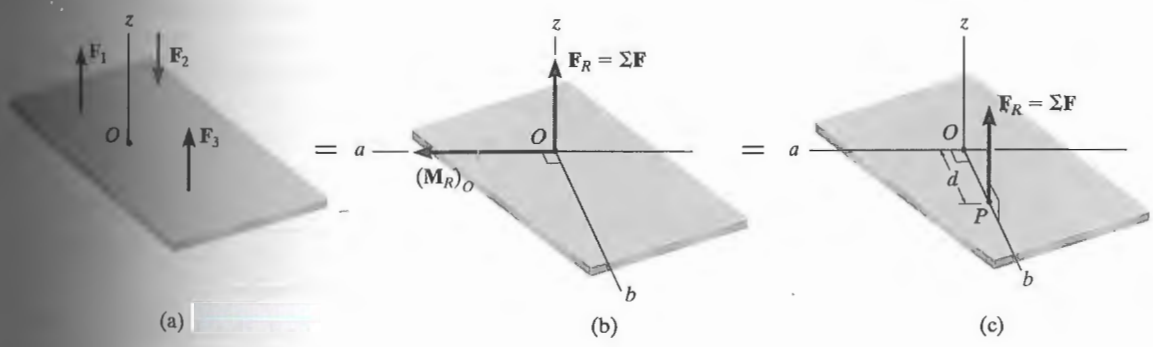


Fig. 4–42

EXAMPLE 4.17

Replace the force and couple moment system acting on the beam in Fig. 4–44*a* by an equivalent resultant force, and find where its line of action intersects the beam, measured from point O.

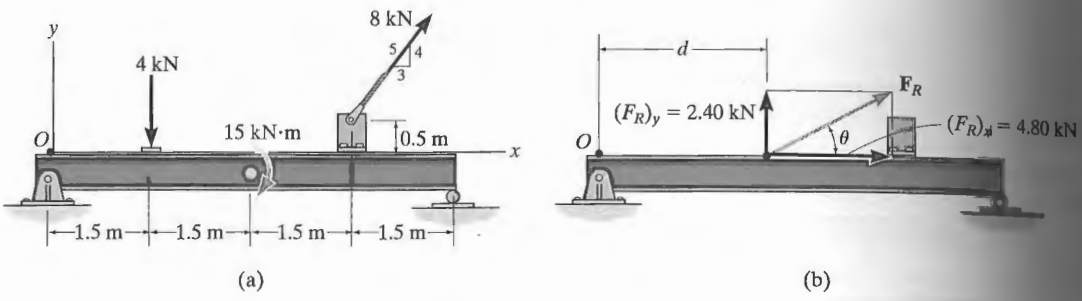


Fig. 4–44

SOLUTION

Force Summation. Summing the force components,

$$\overset{+}{\rightarrow}(F_R)_x = \Sigma F_x; \quad (F_R)_x = 8\text{ kN}\left(\tfrac{3}{5}\right) = 4.80\text{ kN} \rightarrow$$

$$+\uparrow(F_R)_y = \Sigma F_y; \quad (F_R)_y = -4\text{ kN} + 8\text{ kN}\left(\tfrac{4}{5}\right) = 2.40\text{ kN}\uparrow$$

From Fig. 4–44*b*, the magnitude of $\mathbf{F}_R$ is

$$F_R = \sqrt{(4.80\text{ kN})^2 + (2.40\text{ kN})^2} = 5.37\text{ kN} \qquad \textit{Ans}$$

The angle θ is

$$\theta = \tan^{-1}\left(\frac{2.40\text{ kN}}{4.80\text{ kN}}\right) = 26.6° \qquad \textit{Ans}$$

Moment Summation. We must equate the moment of $\mathbf{F}_R$ about point O in Fig. 4–44*b* to the sum of the moments of the force and couple moment system about point O in Fig. 4–44*a*. Since the line of action of $(\mathbf{F}_R)_x$ acts through point O, *only* $(\mathbf{F}_R)_y$ *produces a moment* about this point. Thus,

$$\circlearrowleft + (M_R)_O = \Sigma M_O; \quad 2.40\text{ kN}(d) = -(4\text{ kN})(1.5\text{ m}) - 15\text{ kN}\cdot\text{m}$$

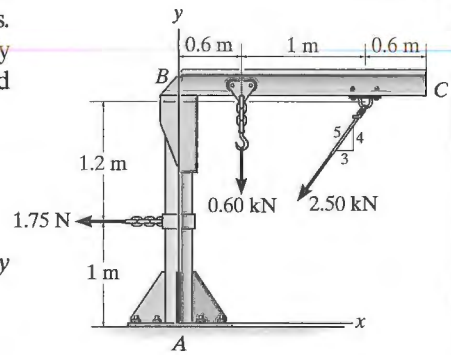
$$-[8\text{ kN}\left(\tfrac{3}{5}\right)](0.5\text{ m}) + [8\text{ kN}\left(\tfrac{4}{5}\right)](4.5\text{ m})$$

$$d = 2.25\text{ m} \qquad \textit{Ans}$$

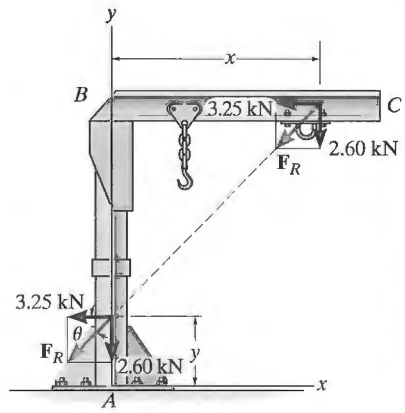
ing exclusively for The University of Melbourne

EXAMPLE 4.18

The jib crane shown in Fig. 4–45*a* is subjected to three coplanar forces. Replace this loading by an equivalent resultant force and specify where the resultant's line of action intersects the column AB and boom BC.



(a)



(b)

Fig. 4–45

SOLUTION

Force Summation. Resolving the 2.50-kN force into x and y components and summing the force components yield

$$\overset{+}{\rightarrow} F_{R_x} = \Sigma F_x;\ F_{R_x} = -2.50\text{ kN}\left(\tfrac{3}{5}\right) - 1.75\text{ kN} = -3.25\text{ kN} = 3.25\text{ kN}\leftarrow$$

$$+\uparrow F_{R_y} = \Sigma F_y;\ F_{R_y} = -2.50\text{ kN}\left(\tfrac{4}{5}\right) - 0.60\text{ kN} = -2.60\text{ kN} = 2.60\text{ kN}\downarrow$$

As shown by the vector addition in Fig. 4–45*b*,

$$F_R = \sqrt{(3.25\text{ kN})^2 + (2.60\text{ kN})^2} = 4.16\text{ kN} \qquad \textit{Ans.}$$

$$\theta = \tan^{-1}\left(\frac{2.60\text{ kN}}{3.25\text{ kN}}\right) = 38.7^\circ\ \theta\ \measuredangle \qquad \textit{Ans.}$$

Moment Summation. Moments will be summed about point A. Assuming the line of action of $\mathbf{F}_R$ *intersects* AB at a distance y from A, Fig. 4–45*b*, we have

$$\circlearrowleft + M_{R_A} = \Sigma M_A;\qquad 3.25\text{ kN}\,(y) + 2.60\text{ kN}\,(0)$$

$$= 1.75\text{ kN}(1\text{ m}) - 0.60\text{ kN}(0.6\text{ m}) + 2.50\text{ kN}\left(\tfrac{3}{5}\right)(2.2\text{ m}) - 2.50\text{ kN}\left(\tfrac{4}{5}\right)(1.6\text{ m})$$

$$y = 0.458\text{ m} \qquad \textit{Ans.}$$

By the principle of transmissibility, $\mathbf{F}_R$ can be placed at a distance x where it intersects BC, Fig. 4–45*b*. In this case we have

$$\circlearrowleft + M_{R_A} = \Sigma M_A;\qquad 3.25\text{ kN}\,(2.2\text{ m}) - 2.60\text{ kN}\,(x)$$

$$= 1.75\text{ kN}(1\text{ m}) - 0.60\text{ kN}(0.6\text{ m}) + 2.50\text{ kN}\left(\tfrac{3}{5}\right)(2.2\text{ m}) - 2.50\text{ kN}\left(\tfrac{4}{5}\right)(1.6\text{ m})$$

$$x = 2.177\text{ m} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 4.19

The slab in Fig. 4–46*a* is subjected to four parallel forces. Determine the magnitude and direction of a resultant force equivalent to the given force system and locate its point of application on the slab.

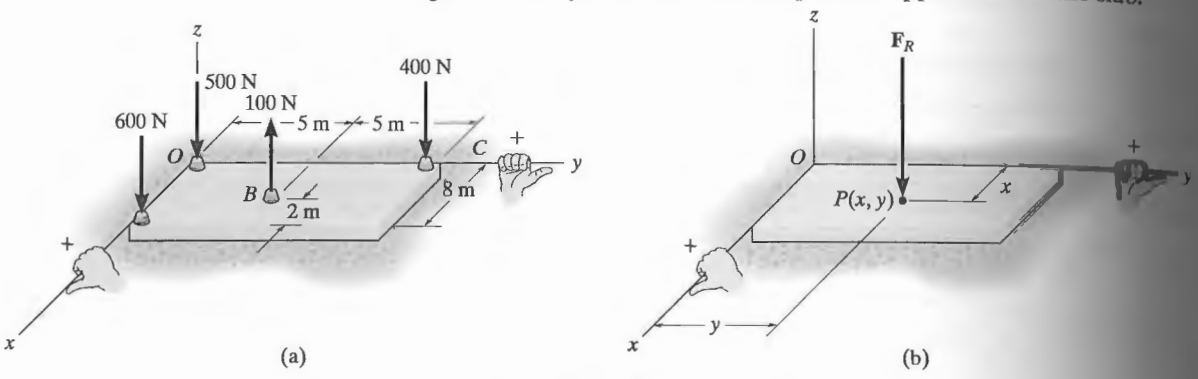


Fig. 4–46

SOLUTION *(SCALAR ANALYSIS)*

Force Summation. From Fig. 4–46*a*, the resultant force is

$$+\uparrow F_R = \Sigma F; \qquad -F_R = -600\text{ N} + 100\text{ N} - 400\text{ N} - 500\text{ N}$$

$$= -1400\text{ N} = 1400\text{ N}\downarrow \qquad \textit{Ans.}$$

Moment Summation. We require the moment about the x axis of the resultant force, Fig. 4–46*b*, to be equal to the sum of the moments about the x axis of all the forces in the system, Fig. 4–46*a*. The moment arms are determined from the y coordinates since these coordinates represent the *perpendicular distances* from the x axis to the lines of action of the forces. Using the right-hand rule, we have

$$(M_R)_x = \Sigma M_x;$$

$$-(1400\text{ N})y = 600\text{ N}(0) + 100\text{ N}(5\text{ m}) - 400\text{ N}(10\text{ m}) + 500\text{ N}(0)$$

$$-1400y = -3500 \qquad y = 2.50\text{ m} \qquad \textit{Ans.}$$

In a similar manner, a moment equation can be written about the y axis using moment arms defined by the x coordinates of each force.

$$(M_R)_y = \Sigma M_y;$$

$$(1400\text{ N})x = 600\text{ N}(8\text{ m}) - 100\text{ N}(6\text{ m}) + 400\text{ N}(0) + 500\text{ N}(0)$$

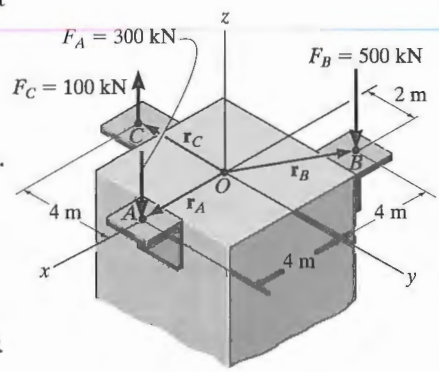
$$1400x = 4200$$

$$x = 3\text{ m} \qquad \textit{Ans.}$$

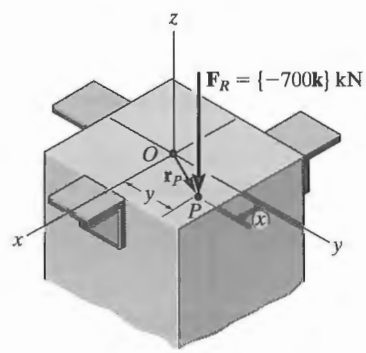
NOTE: A force of $F_R = 1400$ N placed at point $P(3.00\text{ m}, 2.50\text{ m})$ on the slab, Fig. 4–46*b*, is therefore equivalent to the parallel force system acting on the slab in Fig. 4–46*a*.

EXAMPLE 4.20

Replace the force system in Fig. 4–47*a* by an equivalent resultant force and specify its point of application on the pedestal.



(a)



(b)

Fig. 4–47

SOLUTION

Force Summation. Here we will demonstrate a vector analysis. Summing forces,

$$\mathbf{F}_R = \Sigma\mathbf{F}; \quad \mathbf{F}_R = \mathbf{F}_A + \mathbf{F}_B + \mathbf{F}_C$$
$$= \{-300\mathbf{k}\}\text{ kN} + \{-500\mathbf{k}\}\text{ kN} + \{100\mathbf{k}\}\text{ kN}$$
$$= \{-700\mathbf{k}\}\text{ kN} \qquad \textit{Ans.}$$

Location. Moments will be summed about point O. The resultant force $\mathbf{F}_R$ is assumed to act through point $P\,(x, y, 0)$, Fig. 4–47*b*. Thus

$$(\mathbf{M}_R)_O = \Sigma\mathbf{M}_O;$$
$$\mathbf{r}_P \times \mathbf{F}_R = (\mathbf{r}_A \times \mathbf{F}_A) + (\mathbf{r}_B \times \mathbf{F}_B) + (\mathbf{r}_C \times \mathbf{F}_C)$$
$$(x\mathbf{i} + y\mathbf{j}) \times (-700\mathbf{k}) = [(4\mathbf{i}) \times (-300\mathbf{k})]$$
$$+ [(-4\mathbf{i} + 2\mathbf{j}) \times (-500\mathbf{k})] + [(-4\mathbf{j}) \times (100\mathbf{k})]$$
$$-700x(\mathbf{i} \times \mathbf{k}) - 700y(\mathbf{j} \times \mathbf{k}) = -1200(\mathbf{i} \times \mathbf{k}) + 2000(\mathbf{i} \times \mathbf{k})$$
$$- 1000(\mathbf{j} \times \mathbf{k}) - 400(\mathbf{j} \times \mathbf{k})$$
$$700x\mathbf{j} - 700y\mathbf{i} = 1200\mathbf{j} - 2000\mathbf{j} - 1000\mathbf{i} - 400\mathbf{i}$$

Equating the **i** and **j** components,

$$-700y = -1400 \qquad (1)$$
$$y = 2\text{ m} \qquad \textit{Ans.}$$
$$700x = -800 \qquad (2)$$
$$x = -1.14\text{ m} \qquad \textit{Ans.}$$

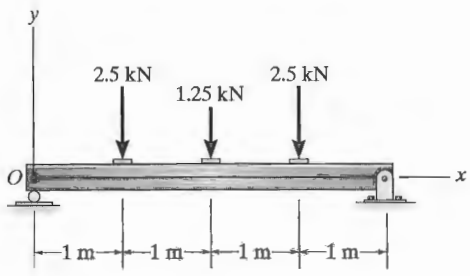
The negative sign indicates that the x coordinate of point P is negative.

NOTE: It is also possible to establish Eq. 1 and 2 directly by summing moments about the x and y axes. Using the right-hand rule, we have

$$(M_R)_x = \Sigma M_x; \qquad -700y = -100\text{ kN}(4\text{ m}) - 500\text{ kN}(2\text{ m})$$
$$(M_R)_y = \Sigma M_y; \qquad 700x = 300\text{ kN}(4\text{ m}) - 500\text{ kN}(4\text{ m})$$

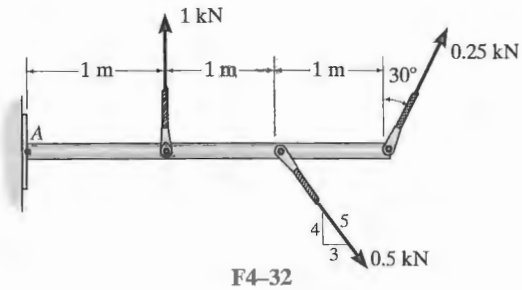
FUNDAMENTAL PROBLEMS

F4–31. Replace the loading system by an equivalent resultant force and specify where the resultant's line of action intersects the beam measured from O.



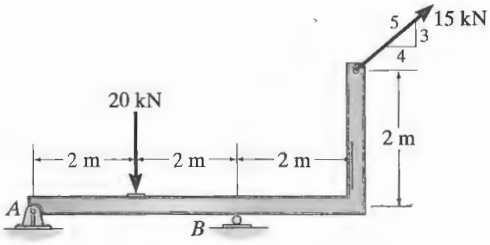
F4–31

F4–32. Replace the loading system by an equivalent resultant force and specify where the resultant's line of action intersects the member measured from A.



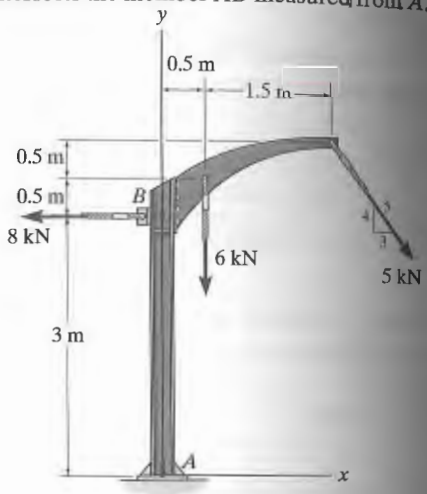
F4–32

F4–33. Replace the loading system by an equivalent resultant force and specify where the resultant's line of action intersects the member measured from A.



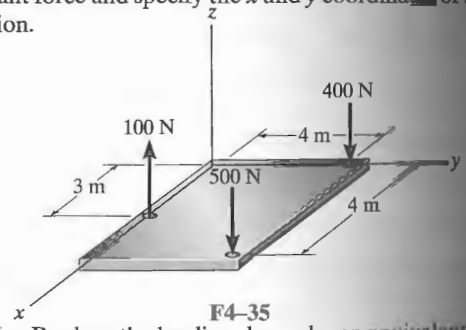
F4–33

F4–34. Replace the loading system by an equivalent resultant force and specify where the resultant's line of action intersects the member AB measured from A.



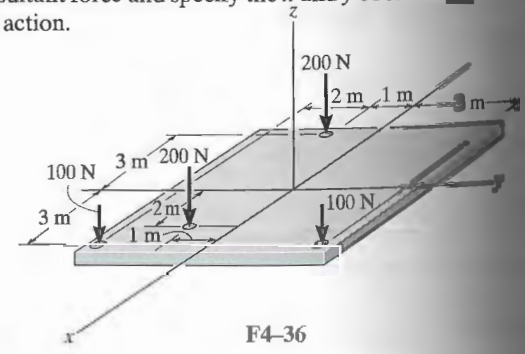
F4–34

F4–35. Replace the loading shown by an equivalent single resultant force and specify the x and y coordinates of its line of action.



F4–35

F4–36. Replace the loading shown by an equivalent single resultant force and specify the x and y coordinates of its line of action.



F4–36

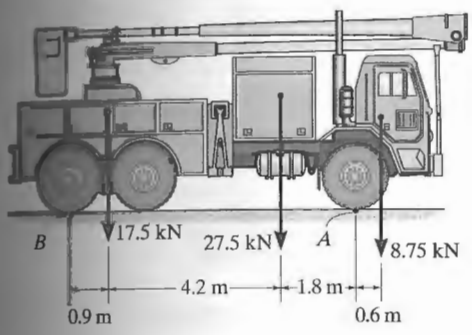
Pearson Custom Publishing exclusively for The University of Melbourne

ENGINEERING
Customized coaching is available!

PROBLEMS

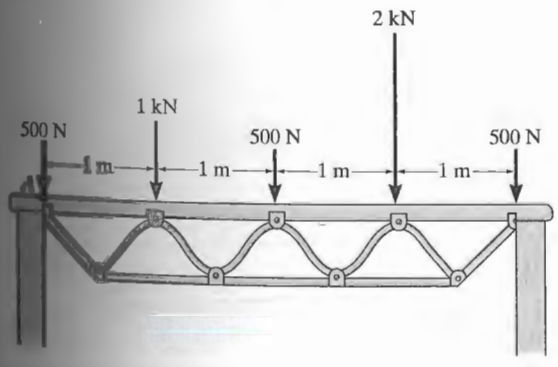
4–118. The weights of the various components of the truck are shown. Replace this system of forces by an equivalent resultant force and specify its location measured from B.

4–119. The weights of the various components of the truck are shown. Replace this system of forces by an equivalent resultant force and specify its location measured from point A.



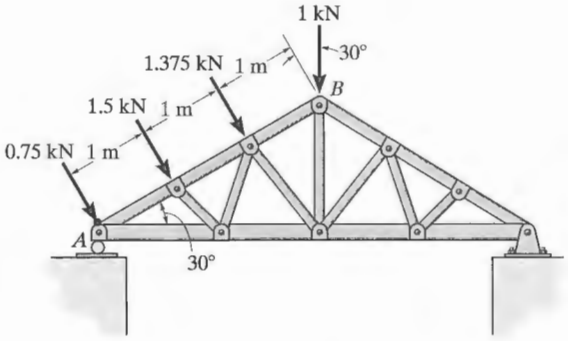
Probs. 4–118/119

4–120. The system of parallel forces acts on the top of the *Warren truss*. Determine the equivalent resultant force of the system and specify its location measured from point A.



Prob. 4–120

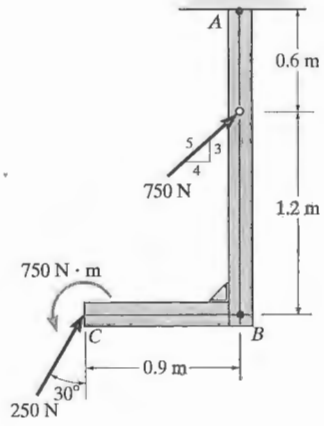
•**4–121.** The system of four forces acts on the roof truss. Determine the equivalent resultant force and specify its location along AB, measured from point A.



Prob. 4–121

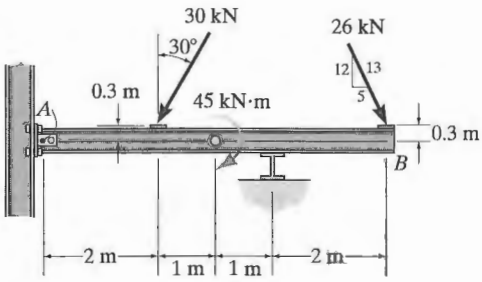
4–122. Replace the force and couple system acting on the frame by an equivalent resultant force and specify where the resultant's line of action intersects member AB, measured from A.

4–123. Replace the force and couple system acting on the frame by an equivalent resultant force and specify where the resultant's line of action intersects member BC, measured from B.



Probs. 4–122/123

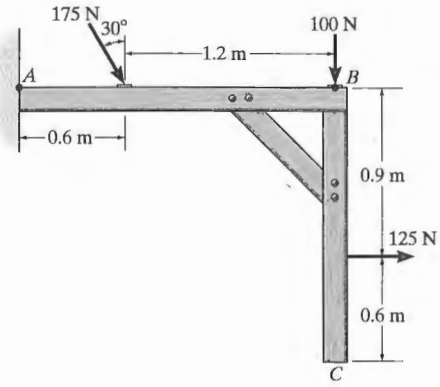
***4–124.** Replace the force and couple moment system acting on the overhang beam by a resultant force, and specify its location along AB measured from point A.



Prob. 4–124

•4–125. Replace the force system acting on the frame by an equivalent resultant force and specify where the resultant's line of action intersects member AB, measured from point A.

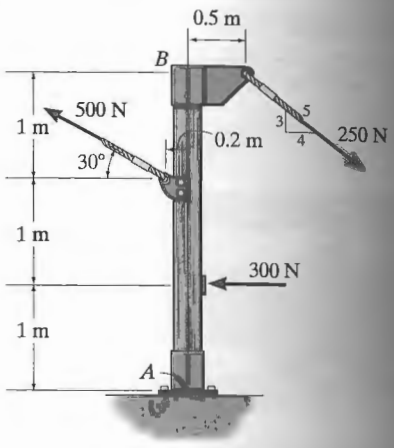
4–126. Replace the force system acting on the frame by an equivalent resultant force and specify where the resultant's line of action intersects member BC, measured from point B.



Probs. 4–125/126

4–127. Replace the force system acting on the post by a resultant force, and specify where its line of action intersects the post AB measured from point A.

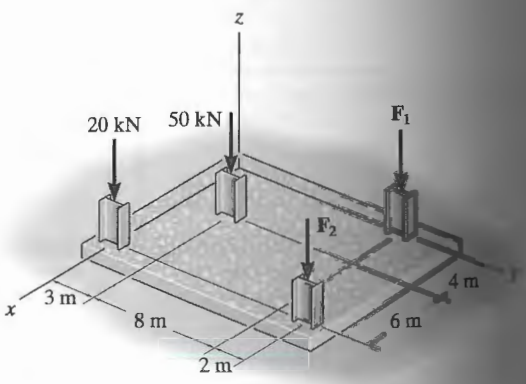
***4–128.** Replace the force system acting on the post by a resultant force, and specify where its line of action intersects the post AB measured from point B.



Probs. 4–127/128

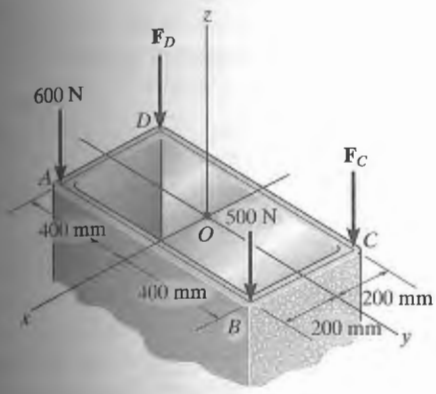
•4–129. The building slab is subjected to four parallel column loadings. Determine the equivalent resultant force and specify its location (x, y) on the slab. Take $F_1 = 30$ kN $F_2 = 40$ kN.

4–130. The building slab is subjected to four parallel column loadings. Determine the equivalent resultant force and specify its location (x, y) on the slab. Take $F_1 = 20$ kN, $F_2 = 50$ kN.



Probs. 4–129/130

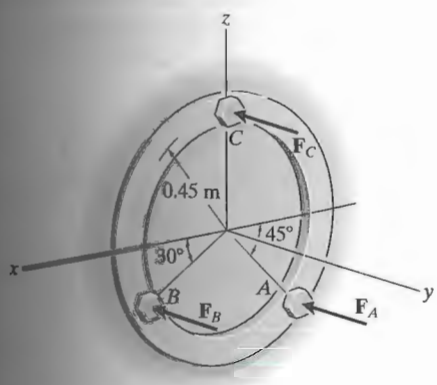
4–131. The tube supports the four parallel forces. Determine the magnitudes of forces $\mathbf{F}_C$ and $\mathbf{F}_D$ acting at C and D so that the equivalent resultant force of the force system acts through the midpoint O of the tube.



Prob. 4–131

4–132. Three parallel bolting forces act on the circular plate. Determine the resultant force, and specify its location (x, z) on the plate. $F_A = 1000$ N, $F_B = 500$ N, and $F_C = 2000$ N.

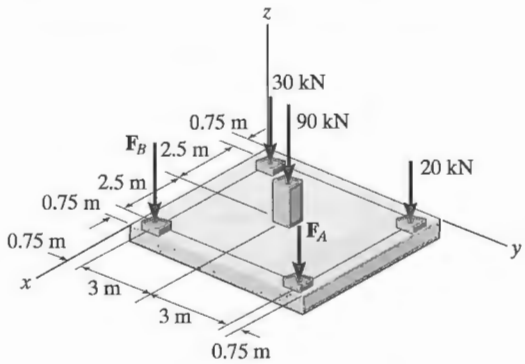
•4–133. The three parallel bolting forces act on the circular plate. If the force at A has a magnitude of $F_A = 1000$ N, determine the magnitudes of $\mathbf{F}_B$ and $\mathbf{F}_C$ so that the resultant force $\mathbf{F}_R$ of the system has a line of action that coincides with the y axis. *Hint:* This requires $\Sigma M_x = 0$ and $\Sigma M_z = 0$.



Probs. 4–132/133

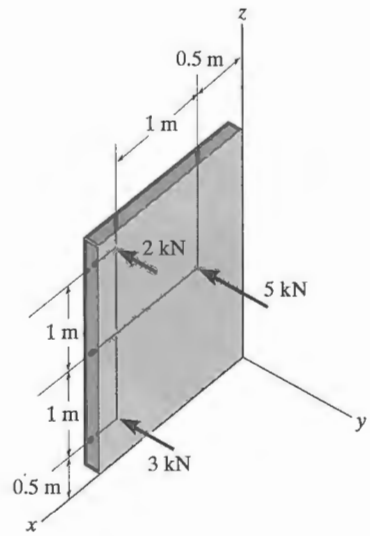
4–134. If $F_A = 40$ kN and $F_B = 35$ kN, determine the magnitude of the resultant force and specify the location of its point of application (x, y) on the slab.

4–135. If the resultant force is required to act at the center of the slab, determine the magnitude of the column loadings $\mathbf{F}_A$ and $\mathbf{F}_B$ and the magnitude of the resultant force.



Probs. 4–134/135

***4–136.** Replace the parallel force system acting on the plate by a resultant force and specify its location on the x–z plane.

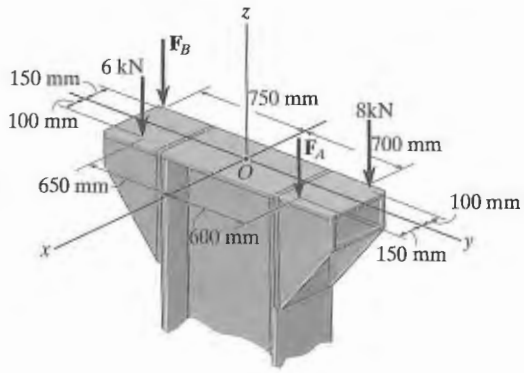


Prob. 4–136

•**4–137.** If $F_A = 7$ kN and $F_B = 5$ kN, represent the force system acting on the corbels by a resultant force, and specify its location on the x–y plane.

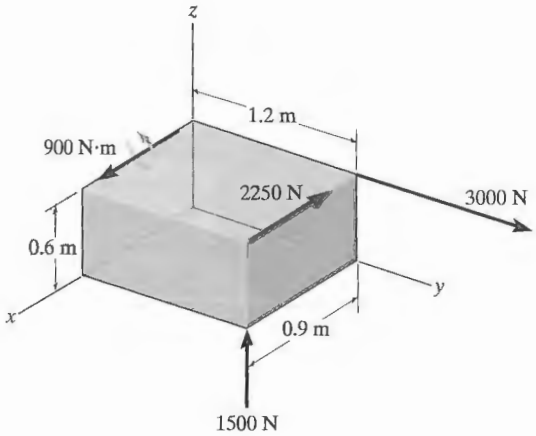
Video Solutions

4–138. Determine the magnitudes of $\mathbf{F}_A$ and $\mathbf{F}_B$ so that the resultant force passes through point O of the column.



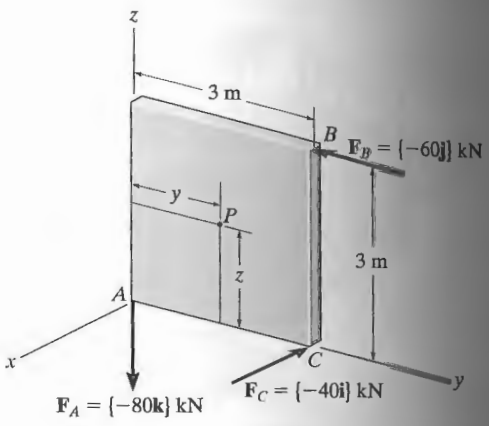
Probs. 4–137/138

4–139. Replace the force and couple moment system acting on the rectangular block by a wrench. Specify the magnitude of the force and couple moment of the wrench and where its line of action intersects the x–y plane.



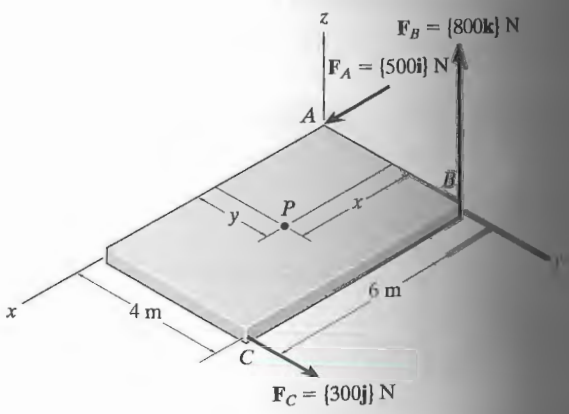
Prob. 4–139

**4–140.* Replace the three forces acting on the plate by a wrench. Specify the magnitude of the force and couple moment for the wrench and the point $P(y, z)$ where its line of action intersects the plate.



Prob. 4–140

•**4–141.** Replace the three forces acting on the plate by a wrench. Specify the magnitude of the force and couple moment for the wrench and the point $P(x, y)$ where its line of action intersects the plate.



Prob. 4–141

A publication from Pearson Custom Publishing exclusively for The University of Melbourne

4.9 Reduction of a Simple Distributed Loading

Sometimes, a body may be subjected to a loading that is distributed over its surface. For example, the pressure of the wind on the face of a sign, the pressure of water within a tank, or the weight of sand on the floor of a storage container, are all *distributed loadings*. The pressure exerted at each point on the surface indicates the intensity of the loading. It is measured using pascals Pa (or N/m^2) in SI units or lb/ft^2 in the U.S. Customary system.

Uniform Loading Along a Single Axis. The most common type of distributed loading encountered in engineering practice is generally uniform along a single axis.* For example, consider the beam (or plate) in Fig. 4–48*a* that has a constant width and is subjected to a pressure loading that varies only along the x axis. This loading can be described by the function $p = p(x)$ N/m^2. It contains only one variable x, and for this reason, we can also represent it as a *coplanar distributed load*. To do so, we multiply the loading function by the width b m of the beam, so that $w(x) = p(x)b$ N/m, Fig. 4-48*b*. Using the methods of Sec. 4.8, we can replace this coplanar parallel force system with a single equivalent resultant force $\mathbf{F}_R$ acting at a specific location on the beam, Fig. 4–48*c*.

Magnitude of Resultant Force. From Eq. 4–17 ($F_R = \Sigma F$), the magnitude of $\mathbf{F}_R$ is equivalent to the sum of all the forces in the system. In this case, integration must be used since there is an infinite number of parallel forces $d\mathbf{F}$ acting on the beam, Fig. 4–48*b*. Since $d\mathbf{F}$ is acting on an element of length dx, and $w(x)$ is a force per unit length, then $dF = w(x)\,dx = dA$. In other words, the magnitude of $d\mathbf{F}$ is determined from the colored differential *area* dA under the loading curve. For the entire length L,

$$+\downarrow F_R = \Sigma F; \qquad \boxed{F_R = \int_L w(x)\,dx = \int_A dA = A} \qquad (4\text{–}19)$$

Therefore, the magnitude of the resultant force is equal to the total area A under the loading diagram, Fig. 4–48*c*.

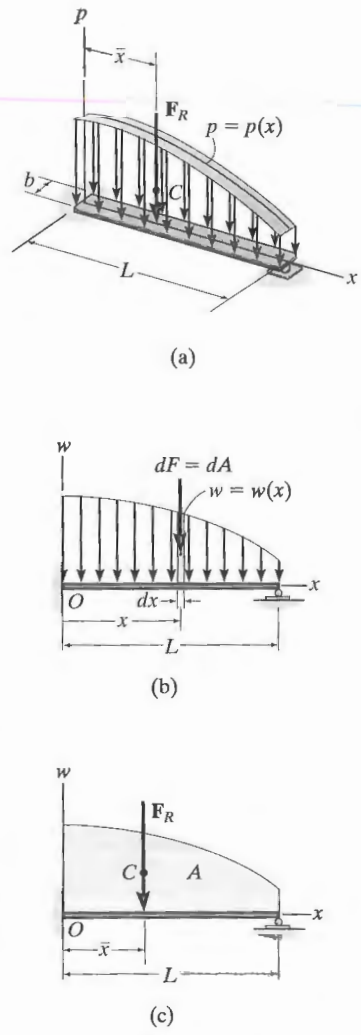
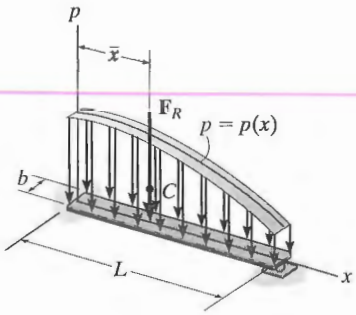


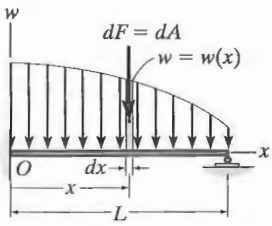
Fig. 4–48

The more general case of a nonuniform surface loading acting on a body is considered in Sec. 9.5.

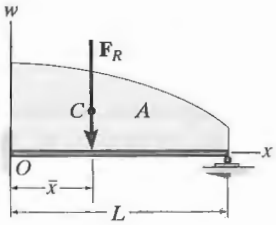
Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.



(a)



(b)



(c)

Location of Resultant Force. Applying Eq. 4–17 ($M_{R_O} = \Sigma M_O$) the location $\bar{x}$ of the line of action of $\mathbf{F}_R$ can be determined by equating the moments of the force resultant and the parallel force distribution about point O (the y axis). Since $d\mathbf{F}$ produces a moment of $x\,dF = xw(x)\,dx$ about O, Fig. 4–48*b*, then for the entire length, Fig. 4–48*c*,

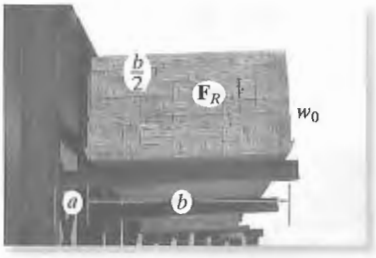
$$\circlearrowleft + (M_R)_O = \Sigma M_O; \qquad -\bar{x}F_R = -\int_L xw(x)\,dx$$

Solving for $\bar{x}$, using Eq. 4–19, we have

$$\bar{x} = \frac{\int_L xw(x)\,dx}{\int_L w(x)\,dx} = \frac{\int_A x\,dA}{\int_A dA} \tag{4–20}$$

This coordinate $\bar{x}$, locates the geometric center or *centroid* of the *area* under the distributed loading. *In other words, the resultant force has a line of action which passes through the centroid C (geometric center) of the area under the loading diagram*, Fig. 4–48*c*. Detailed treatment of the integration techniques for finding the location of the centroid for areas is given in Chapter 9. In many cases, however, the distributed-loading diagram is in the shape of a rectangle, triangle, or some other simple geometric form. The centroid location for such common shapes does not have to be determined from the above equation but can be obtained directly from the tabulation given on the inside back cover.

Once $\bar{x}$ is determined, $\mathbf{F}_R$ by symmetry passes through point $(\bar{x}, 0)$ on the surface of the beam, Fig. 4–48*a*. *Therefore, in this case, the resultant force has a magnitude equal to the volume under the loading curve $p = p(x)$ and a line of action which passes through the centroid (geometric center) of this volume.*



The beam supporting this stack of lumber is subjected to a uniform loading of w_0. The resultant force is therefore equal to the area under the loading diagram $F_R = w_0 b$. It acts through the centroid or geometric center of this area, $b/2$ from the support.

Important Points

- Coplanar distributed loadings are defined by using a loading function $w = w(x)$ that indicates the intensity of the loading along the length of a member. This intensity is measured in N/m or lb/ft.
- The external effects caused by a coplanar distributed load acting on a body can be represented by a single resultant force.
- This resultant force is equivalent to the *area* under the loading diagram, and has a line of action that passes through the *centroid* or geometric center of this area.

EXAMPLE 4.21

Determine the magnitude and location of the equivalent resultant force acting on the shaft in Fig. 4–49*a*.

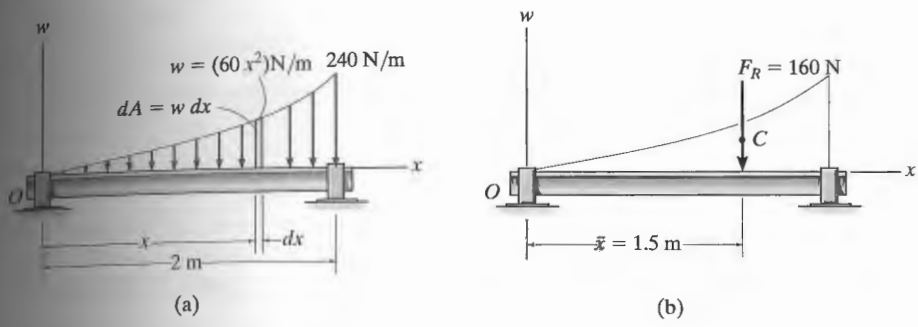


Fig. 4–49

SOLUTION

Since $w = w(x)$ is given, this problem will be solved by integration.

The differential element has an area $dA = w\,dx = 60x^2\,dx$. Applying Eq. 4–19,

$+\downarrow F_R = \Sigma F;$

$$F_R = \int_A dA = \int_0^{2\text{ m}} 60x^2\,dx = 60\left(\frac{x^3}{3}\right)\Bigg|_0^{2\text{ m}} = 60\left(\frac{2^3}{3} - \frac{0^3}{3}\right)$$

$$= 160\text{ N} \qquad \textit{Ans.}$$

The location $\bar{x}$ of $\mathbf{F}_R$ measured from O, Fig. 4–49*b*, is determined from Eq. 4–20.

$$\bar{x} = \frac{\int_A x\,dA}{\int_A dA} = \frac{\int_0^{2\text{ m}} x(60x^2)\,dx}{160\text{ N}} = \frac{60\left(\frac{x^4}{4}\right)\Big|_0^{2\text{ m}}}{160\text{ N}} = \frac{60\left(\frac{2^4}{4} - \frac{0^4}{4}\right)}{160\text{ N}}$$

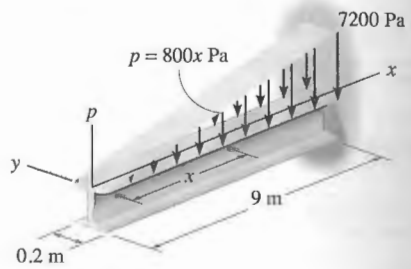
$$= 1.5\text{ m} \qquad \textit{Ans.}$$

NOTE: These results can be checked by using the table on the inside back cover, where it is shown that for an exparabolic area of length a, height b, and shape shown in Fig. 4–49*a*, we have

$$A = \frac{ab}{3} = \frac{2\text{ m}(240\text{ N/m})}{3} = 160\text{ N and } \bar{x} = \frac{3}{4}a = \frac{3}{4}(2\text{ m}) = 1.5\text{ m}$$

EXAMPLE 4.22

A distributed loading of $p = (800x)$ Pa acts over the top surface of the beam shown in Fig. 4–50*a*. Determine the magnitude and location of the equivalent resultant force.



(a)

SOLUTION

Since the loading intensity is uniform along the width of the beam (the y axis), the loading can be viewed in two dimensions as shown in Fig. 4–50*b*. Here

$$w = (800x \text{ N/m}^2)(0.2 \text{ m})$$
$$= (160x) \text{ N/m}$$

At $x = 9$ m, note that $w = 1440$ N/m. Although we may again apply Eqs. 4–19 and 4–20 as in the previous example, it is simpler to use the table on the inside back cover.

The magnitude of the resultant force is equivalent to the area of the triangle.

$$F_R = \tfrac{1}{2}(9 \text{ m})(1440 \text{ N/m}) = 6480 \text{ N} = 6.48 \text{ kN} \qquad \textit{Ans.}$$

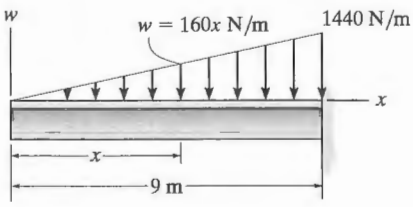
The line of action of $\mathbf{F}_R$ passes through the *centroid* C of this triangle. Hence,

$$\bar{x} = 9 \text{ m} - \tfrac{1}{3}(9 \text{ m}) = 6 \text{ m} \qquad \textit{Ans.}$$

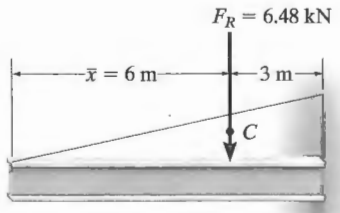
The results are shown in Fig. 4–50*c*.

NOTE: We may also view the resultant $\mathbf{F}_R$ as *acting* through the *centroid* of the *volume* of the loading diagram $p = p(x)$ in Fig. 4–50*a*. Hence $\mathbf{F}_R$ intersects the x–y plane at the point (6 m, 0). Furthermore, the magnitude of $\mathbf{F}_R$ is equal to the volume under the loading diagram; i.e.,

$$F_R = V = \tfrac{1}{2}(7200 \text{ N/m}^2)(9 \text{ m})(0.2 \text{ m}) = 6.48 \text{ kN} \qquad \textit{Ans.}$$



(b)



(c)

Fig. 4–50

from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 4.23

The granular material exerts the distributed loading on the beam as shown in Fig. 4–51*a*. Determine the magnitude and location of the equivalent resultant of this load.

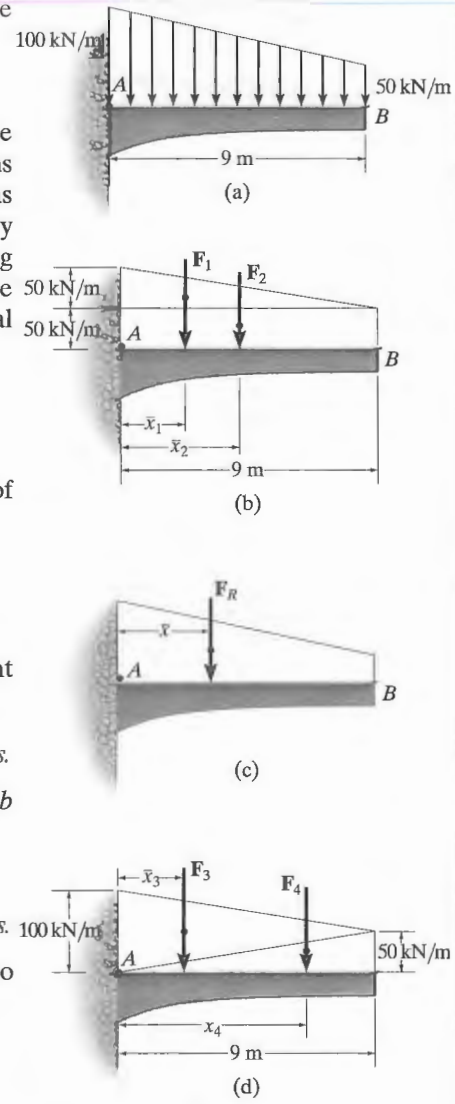


Fig. 4–51

SOLUTION

The area of the loading diagram is a *trapezoid*, and therefore the solution can be obtained directly from the area and centroid formulas for a trapezoid listed on the inside back cover. Since these formulas are not easily remembered, instead we will solve this problem by using "composite" areas. Here we will divide the trapezoidal loading into a rectangular and triangular loading as shown in Fig. 4–51*b*. The magnitude of the force represented by each of these loadings is equal to its associated *area*,

$$F_1 = \tfrac{1}{2}(9\text{ m})(50\text{ kN/m}) = 225\text{ kN}$$

$$F_2 = (9\text{ m})(50\text{ kN/m}) = 450\text{ kN}$$

The lines of action of these parallel forces act through the *centroid* of their associated areas and therefore intersect the beam at

$$\bar{x}_1 = \tfrac{1}{3}(9\text{ m}) = 3\text{ m}$$

$$\bar{x}_2 = \tfrac{1}{2}(9\text{ m}) = 4.5\text{ m}$$

The two parallel forces $\mathbf{F}_1$ and $\mathbf{F}_2$ can be reduced to a single resultant $\mathbf{F}_R$. The magnitude of $\mathbf{F}_R$ is

$$+\downarrow F_R = \Sigma F; \qquad F_R = 225 + 450 = 675\text{ kN} \qquad \textit{Ans.}$$

We can find the location of $\mathbf{F}_R$ with reference to point A, Fig. 4–51*b* and 4–51*c*. We require

$$\circlearrowleft + M_{R_A} = \Sigma M_A; \quad \bar{x}(675) = 3(225) + 4.5(450)$$

$$\bar{x} = 4\text{ m} \qquad \textit{Ans.}$$

NOTE: The trapezoidal area in Fig. 4–51*a* can also be divided into two triangular areas as shown in Fig. 4–51*d*. In this case

$$F_3 = \tfrac{1}{2}(9\text{ m})(100\text{ kN/m}) = 450\text{ kN}$$

$$F_4 = \tfrac{1}{2}(9\text{ m})(50\text{ kN/m}) = 225\text{ kN}$$

and

$$\bar{x}_3 = \tfrac{1}{3}(9\text{ m}) = 3\text{ m}$$

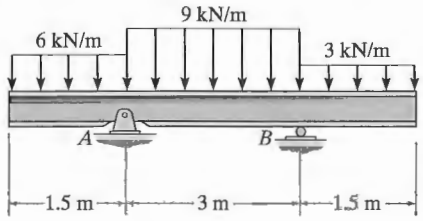
$$\bar{x}_4 = 9\text{ m} - \tfrac{1}{3}(9\text{ m}) = 6\text{ m}$$

NOTE: Using these results, show again that $F_R = 675$ kN and $\bar{x} = 4$ m

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

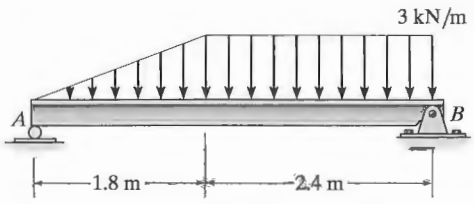
FUNDAMENTAL PROBLEMS

F4–37. Determine the resultant force and specify where it acts on the beam measured from A.



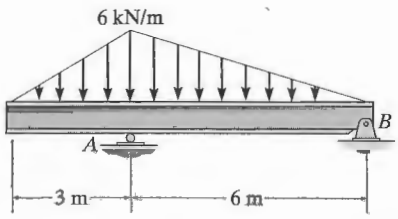
F4–37

F4–38. Determine the resultant force and specify where it acts on the beam measured from A.



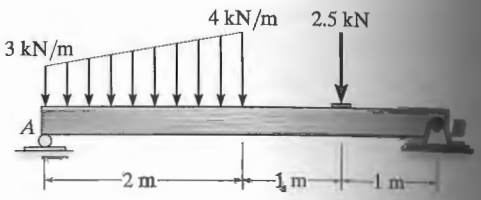
F4–38

F4–39. Determine the resultant force and specify where it acts on the beam measured from A.



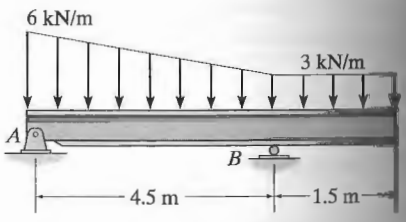
F4–39

F4–40. Determine the resultant force and specify where it acts on the beam measured from A.



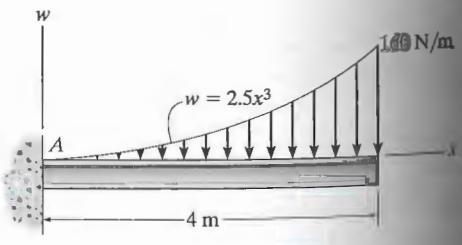
F4–40

F4–41. Determine the resultant force and specify where it acts on the beam measured from A.



F4–41

F4–42. Determine the resultant force and specify where it acts on the beam measured from A.

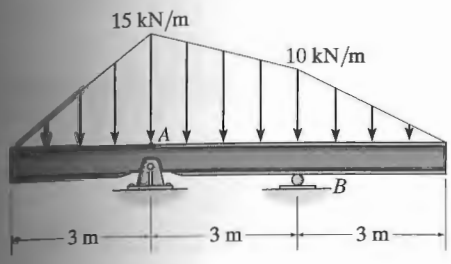


F4–42

ENGINEERING
Customized coaching is available!

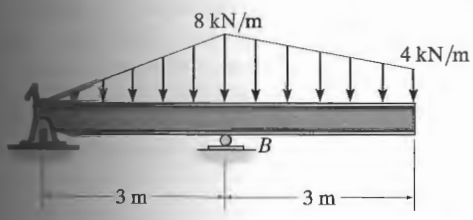
PROBLEMS

4–142. Replace the distributed loading with an equivalent resultant force, and specify its location on the beam measured from point A.



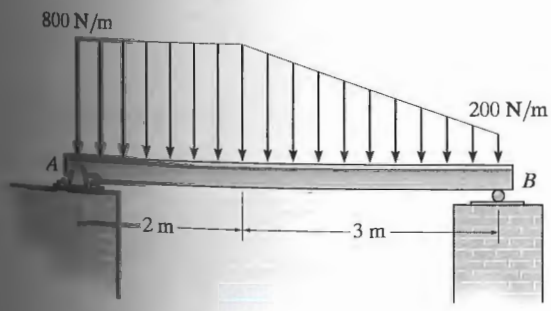
Prob. 4–142

4–143. Replace the distributed loading with an equivalent resultant force, and specify its location on the beam measured from point A.



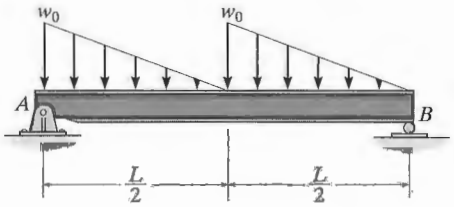
Prob. 4–143

4–144. Replace the distributed loading by an equivalent resultant force and specify its location, measured from point A.



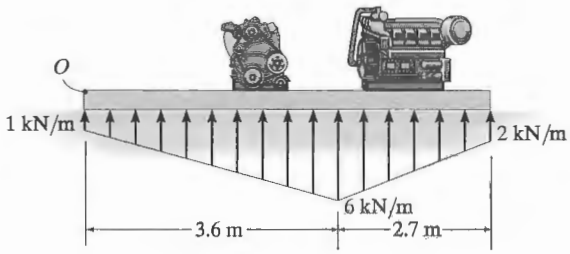
Prob. 4–144

•4–145. Replace the distributed loading with an equivalent resultant force, and specify its location on the beam measured from point A.



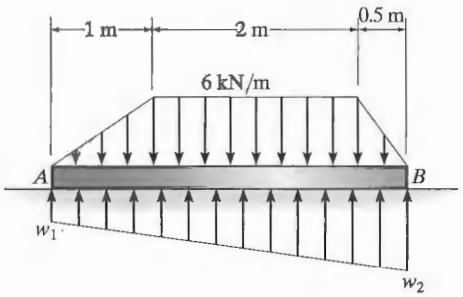
Prob. 4–145

4–146. The distribution of soil loading on the bottom of a building slab is shown. Replace this loading by an equivalent resultant force and specify its location, measured from point O.



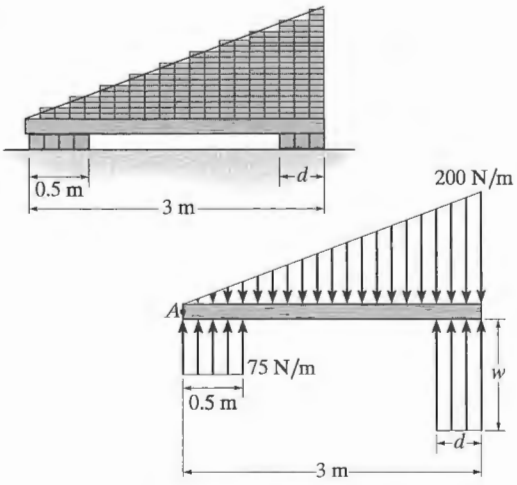
Prob. 4–146

4–147. Determine the intensities w_1 and w_2 of the distributed loading acting on the bottom of the slab so that this loading has an equivalent resultant force that is equal but opposite to the resultant of the distributed loading acting on the top of the plate.



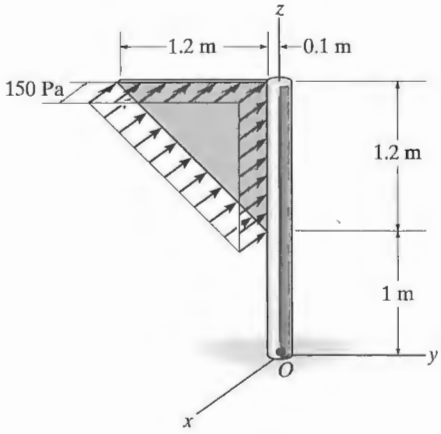
Prob. 4–147

***4–148.** The bricks on top of the beam and the supports at the bottom create the distributed loading shown in the second figure. Determine the required intensity w and dimension d of the right support so that the resultant force and couple moment about point A of the system are both zero.



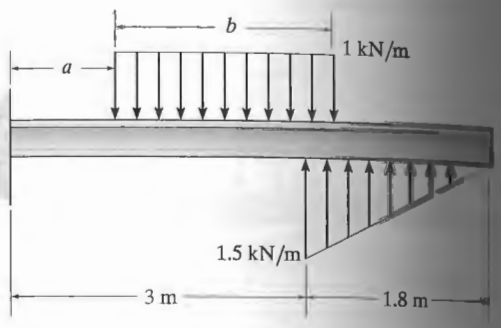
Prob. 4–148

•4–149. The wind pressure acting on a triangular sign is uniform. Replace this loading by an equivalent resultant force and couple moment at point O.



Prob. 4–149

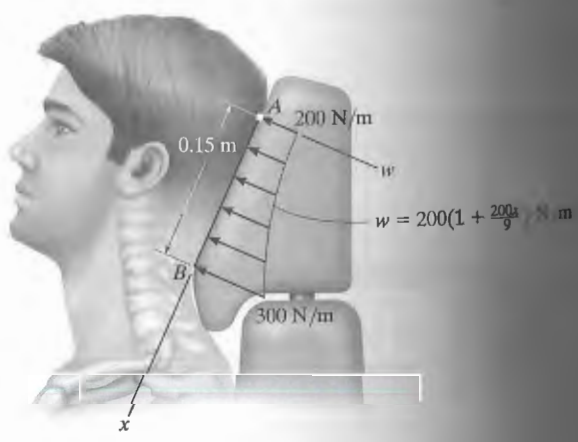
4–150. The beam is subjected to the distributed loading. Determine the length b of the uniform load and its position a on the beam such that the resultant force and couple moment acting on the beam are zero.



Prob. 4–150

4–151. Currently eighty-five percent of all neck injuries are caused by rear-end car collisions. To alleviate this problem, an automobile seat restraint has been developed that provides additional pressure contact with the cranium. During dynamic tests the distribution of load on the cranium has been plotted and shown to be parabolic. Determine the equivalent resultant force and its location, measured from point A.

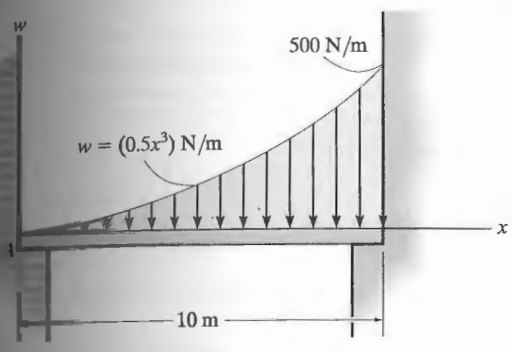
Video Solutions



Prob. 4–151

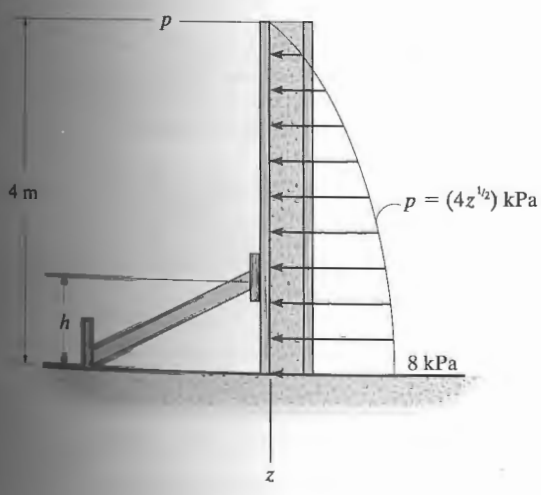
ENGI
m Pe
hing exclusively for The University of Melbourne

4–152. Wind has blown sand over a platform such that the intensity of the load can be approximated by the function $w = (0.5x^3)$ N/m. Simplify this distributed loading to an equivalent resultant force and specify its magnitude and location measured from A.



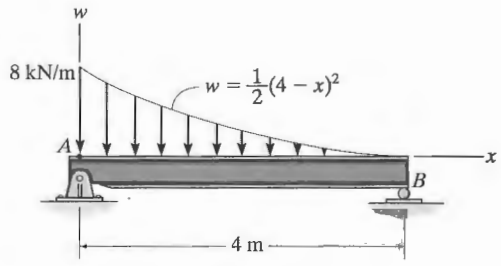
Prob. 4–152

•4–153. Wet concrete exerts a pressure distribution along the wall of the form. Determine the resultant force of this distribution and specify the height h where the bracing strut should be placed so that it lies through the line of action of the resultant force. The wall has a width of 5 m.



Prob. 4–153

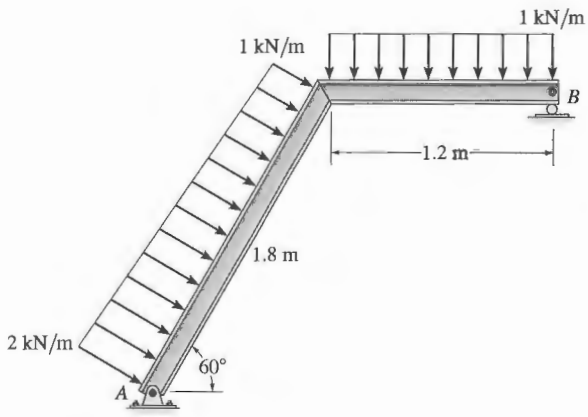
4–154. Replace the distributed loading with an equivalent resultant force, and specify its location on the beam measured from point A.



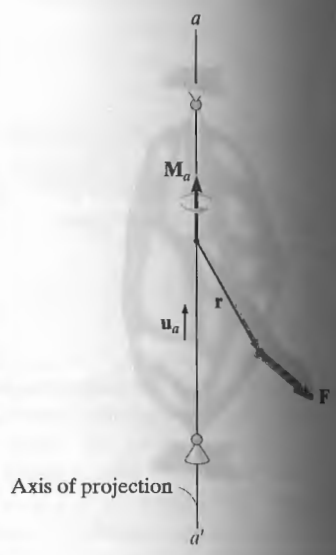
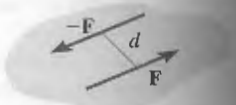
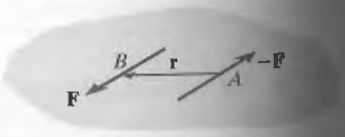
Prob. 4–154

4–155. Replace the loading by an equivalent resultant force and couple moment at point A.

***4–156.** Replace the loading by an equivalent resultant force and couple moment acting at point B.



Probs. 4–155/156

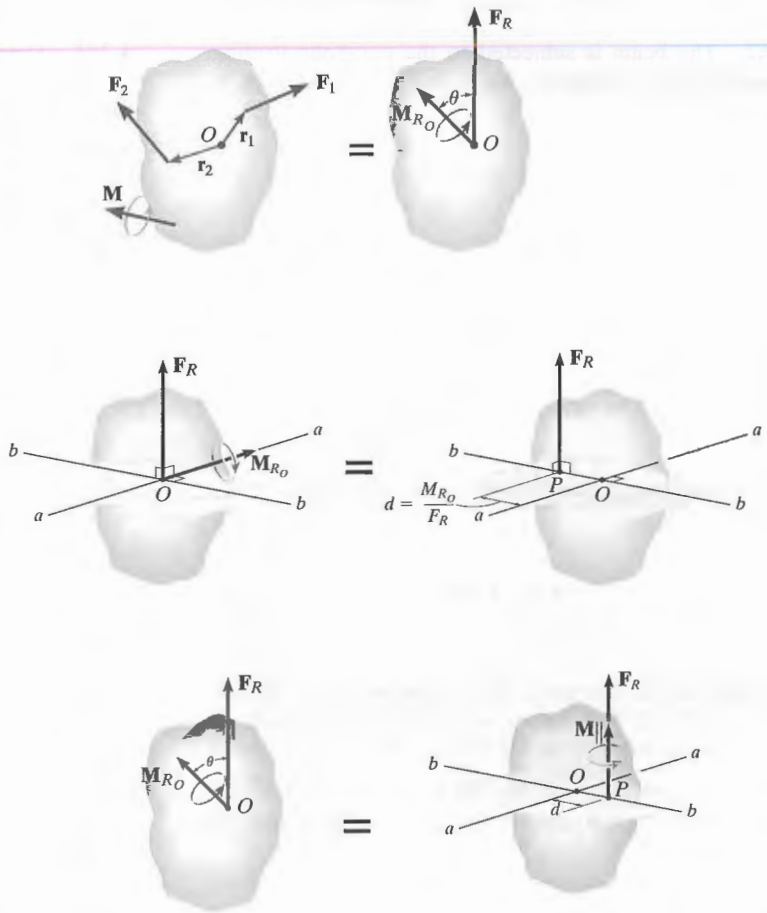
Moment about an Axis If the moment of a force **F** is to be determined about an arbitrary axis a, then the projection of the moment onto the axis must be obtained. Provided the distance d_a that is perpendicular to *both* the line of action of the force and the axis can be found, then the moment of the force about the axis can be determined from a scalar equation.	$M_a = Fd_a$	
Note that when the line of action of **F** intersects the axis then the moment of **F** about the axis is zero. Also, when the line of action of **F** is parallel to the axis, the moment of **F** about the axis is zero. In three dimensions, the scalar triple product should be used. Here $\mathbf{u}_a$ is the unit vector that specifies the direction of the axis, and **r** is a position vector that is directed from any point on the axis to any point on the line of action of the force. If M_a is calculated as a negative scalar, then the sense of direction of $\mathbf{M}_a$ is opposite to $\mathbf{u}_a$.	$M_a = \mathbf{u}_a \cdot (\mathbf{r} \times \mathbf{F}) = \begin{vmatrix} u_{a_x} & u_{a_y} & u_z \\ r_x & r_y & r_z \\ F_x & F_y & F_z \end{vmatrix}$	
Couple Moment A couple consists of two equal but opposite forces that act a perpendicular distance d apart. Couples tend to produce a rotation without translation. The magnitude of the couple moment is $M = Fd$, and its direction is established using the right-hand rule. If the vector cross product is used to determine the moment of a couple, then **r** extends from any point on the line of action of one of the forces to any point on the line of action of the other force **F** that is used in the cross product.	$M = Fd$ $\mathbf{M} = \mathbf{r} \times \mathbf{F}$	 

Simplification of a Force and Couple System

Any system of forces and couples can be reduced to a single resultant force and resultant couple moment acting at a point. The resultant force is the sum of all the forces in the system, $\mathbf{F}_R = \Sigma\mathbf{F}$, and the resultant couple moment is equal to the sum of all the moments of the forces about the point and couple moments. $\mathbf{M}_{R_O} = \Sigma\mathbf{M}_O + \Sigma\mathbf{M}$.

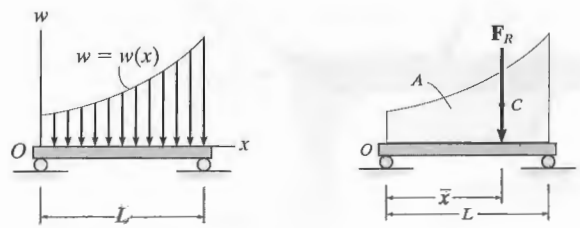
Further simplification to a single resultant force is possible provided the force system is concurrent, coplanar, or parallel. To find the location of the resultant force from a point, it is necessary to equate the moment of the resultant force about the point to the moment of the forces and couples in the system about the same point.

If the resultant force and couple moment at a point are not perpendicular to one another, then this system can be reduced to a wrench, which consists of the resultant force and collinear couple moment.



Coplanar Distributed Loading

A simple distributed loading can be represented by its resultant force, which is equivalent to the *area* under the loading curve. This resultant has a line of action that passes through the *centroid* or geometric center of the area or volume under the loading diagram.

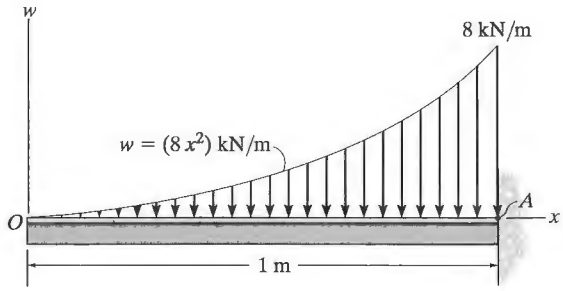


Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

MasteringENGINEERING
Customized coaching is availabl

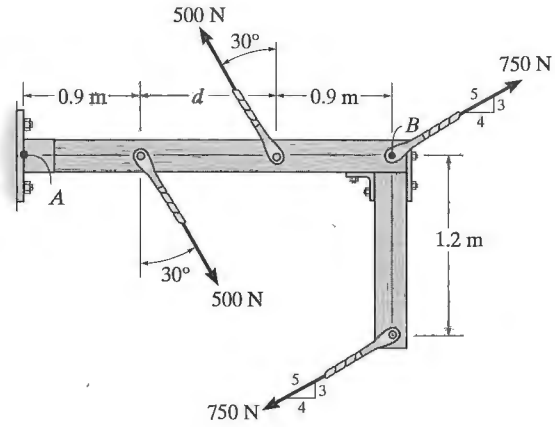
REVIEW PROBLEMS

4–162. The beam is subjected to the parabolic loading. Determine an equivalent force and couple system at point A.



Prob. 4–162

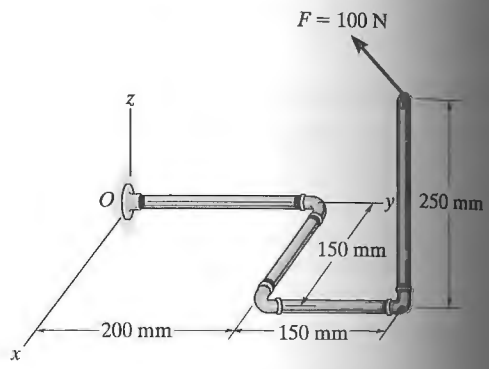
4–163. Two couples act on the frame. If the resultant couple moment is to be zero, determine the distance d between the 500-N couple forces.



Prob. 4–163

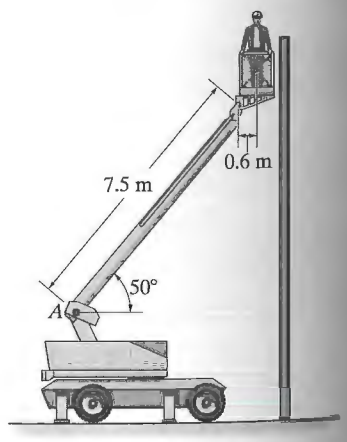
***4–164.** Determine the coordinate direction angles α, β, γ of **F**, which is applied to the end of the pipe assembly, so that the moment of **F** about O is zero.

•4–165. Determine the moment of the force **F** about point O. The force has coordinate direction angles of $\alpha = 60°$, $\beta = 120°$, $\gamma = 45°$. Express the result as a Cartesian vector.



Probs. 4–164/165

4–166. The snorkel boom lift is extended into the position shown. If the worker weighs 800 N (≈ 80 kg), determine the moment of this force about the connection at A.

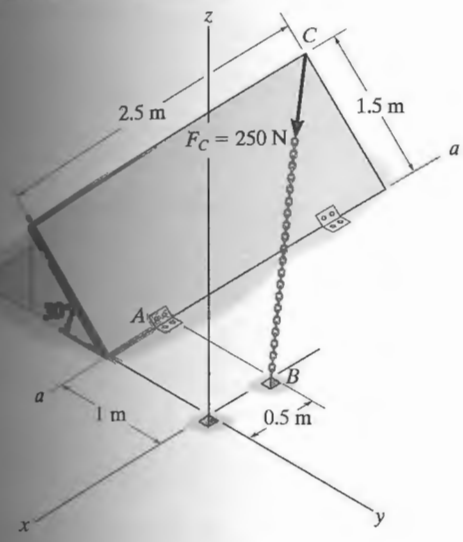


Prob. 4–166

exclusively for The University of Melbourne

4–167. Determine the moment of the force $\mathbf{F}_C$ about the hinge at A. Express the result as a Cartesian vector.

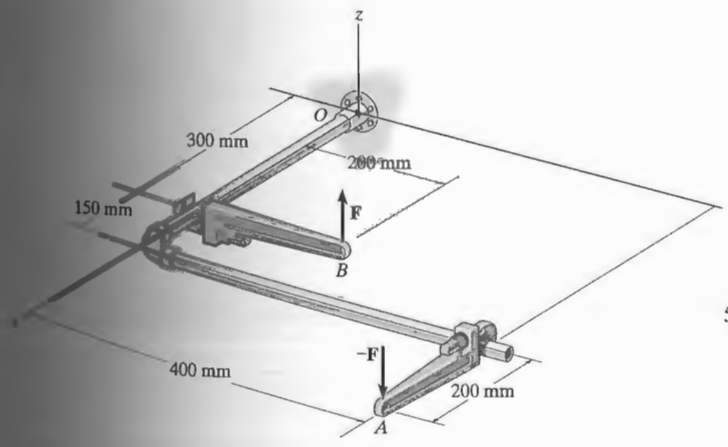
***4–168.** Determine the magnitude of the moment of the force $\mathbf{F}_C$ about the hinged axis aa of the door.



Probs. 4–167/168

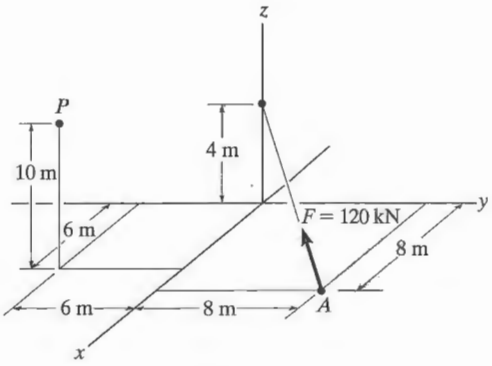
•4–169. Express the moment of the couple acting on the pipe assembly in Cartesian vector form. Solve the problem (a) using Eq. 4–13 and (b) summing the moment of each force about point O. Take $\mathbf{F} = \{25\mathbf{k}\}$ N.

4–170. If the couple moment acting on the pipe has a magnitude of 400 N · m, determine the magnitude F of the vertical force applied to each wrench.



Probs. 4–169/170

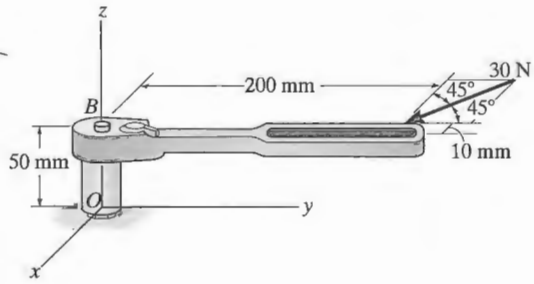
4–171. Replace the force at A by an equivalent resultant force and couple moment at point P. Express the results in Cartesian vector form.



Prob. 4–171

***4–172.** The horizontal 30-N force acts on the handle of the wrench. Determine the moment of this force about point O. Specify the coordinate direction angles α, β, γ of the moment axis.

•4–173. The horizontal 30-N force acts on the handle of the wrench. What is the magnitude of the moment of this force about the z axis?



Probs. 4–172/173

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 117-198). Singapore: Pearson Education South Asia Pte Ltd.

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for the University of

The crane is subjected to its weight and the load it supports. In order to calculate the support reactions on the crane, it is necessary to apply the principles of equilibrium.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

Equilibrium of a Rigid Body

5

CHAPTER OBJECTIVES

- To develop the equations of equilibrium for a rigid body.
- To introduce the concept of the free-body diagram for a rigid body.
- To show how to solve rigid-body equilibrium problems using the equations of equilibrium.

5.1 Conditions for Rigid-Body Equilibrium

In this section, we will develop both the necessary and sufficient conditions for the equilibrium of the rigid body in Fig. 5–1*a*. As shown, this body is subjected to an external force and couple moment system that is the result of the effects of gravitational, electrical, magnetic, or contact forces caused by adjacent bodies. The internal forces caused by interactions between particles within the body are not shown in this figure because these forces occur in equal but opposite collinear pairs and hence will cancel out, a consequence of Newton's third law.

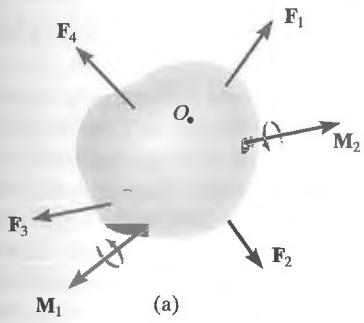


Fig. 5–1

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

The crane is subjected to its weight and the load it supports. In order to calculate the support reactions on the crane, it is necessary to apply the principles of equilibrium.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd

Equilibrium of a Rigid Body

5

CHAPTER OBJECTIVES

- To develop the equations of equilibrium for a rigid body.
- To introduce the concept of the free-body diagram for a rigid body.
- To show how to solve rigid-body equilibrium problems using the equations of equilibrium.

5.1 Conditions for Rigid-Body Equilibrium

In this section, we will develop both the necessary and sufficient conditions for the equilibrium of the rigid body in Fig. 5–1*a*. As shown, this body is subjected to an external force and couple moment system that is the result of the effects of gravitational, electrical, magnetic, or contact forces caused by adjacent bodies. The internal forces caused by interactions between particles within the body are not shown in this figure because these forces occur in equal but opposite collinear pairs and hence will cancel out, a consequence of Newton's third law.

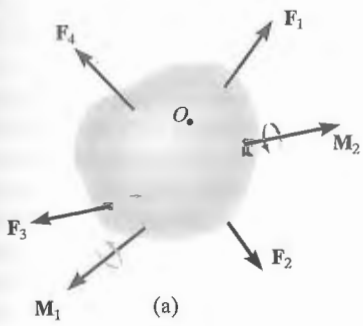
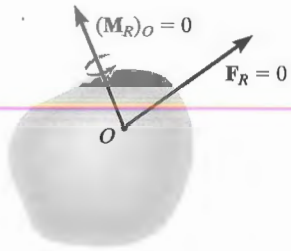
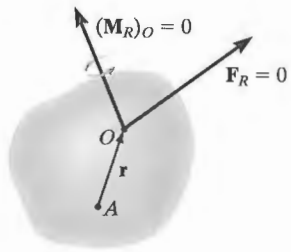


Fig. 5–1

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.



(b)



(c)

Fig. 5–1

Using the methods of the previous chapter, the force and couple moment system acting on a body can be reduced to an equivalent resultant force and resultant couple moment at any arbitrary point O on or off the body, Fig. 5–1*b*. If this resultant force and couple moment are both equal to zero, then the body is said to be in *equilibrium*. Mathematically, the equilibrium of a body is expressed as

$$\mathbf{F}_R = \Sigma\mathbf{F} = \mathbf{0}$$
$$(\mathbf{M}_R)_O = \Sigma\mathbf{M}_O = \mathbf{0} \qquad (5\text{–}1)$$

The first of these equations states that the sum of the forces acting on the body is equal to *zero*. The second equation states that the sum of the moments of all the forces in the system about point O, added to all the couple moments, is equal to *zero*. These two equations are not only necessary for equilibrium, they are also sufficient. To show this, consider summing moments about some other point, such as point A in Fig. 5–1*c*. We require

$$\Sigma\mathbf{M}_A = \mathbf{r} \times \mathbf{F}_R + (\mathbf{M}_R)_O = \mathbf{0}$$

Since $\mathbf{r} \neq \mathbf{0}$, this equation is satisfied only if Eqs. 5–1 are satisfied, namely $\mathbf{F}_R = \mathbf{0}$ and $(\mathbf{M}_R)_O = \mathbf{0}$.

When applying the equations of equilibrium, we will assume that the body remains rigid. In reality, however, all bodies deform when subjected to loads. Although this is the case, most engineering materials such as steel and concrete are very rigid and so their deformation is usually very small. Therefore, when applying the equations of equilibrium, we can generally assume that the body will remain *rigid* and *not deform* under the applied load without introducing any significant error. This way the direction of the applied forces and their moment arms with respect to a fixed reference remain unchanged before and after the body is loaded.

EQUILIBRIUM IN TWO DIMENSIONS

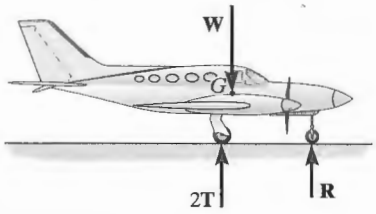


Fig. 5–2

In the first part of the chapter, we will consider the case where the force system acting on a rigid body lies in or may be projected onto a *single* plane and, furthermore, any couple moments acting on the body are directed perpendicular to this plane. This type of force and couple system is often referred to as a two-dimensional or *coplanar* force system. For example, the airplane in Fig. 5–2 has a plane of symmetry through its center axis, and so the loads acting on the airplane are symmetrical with respect to this plane. Thus, each of the two wing tires will support the same load **T**, which is represented on the side (two-dimensional view of the plane as 2**T**.

MS DESIGN 2 CHAPTER 5 Equilibrium of a Rigid Body
lishing exclusively for The University of Melbourne

5.2 Free-Body Diagrams

Successful application of the equations of equilibrium requires a complete specification of *all* the known and unknown external forces that act *on* the body. The best way to account for these forces is to draw a free-body diagram. This diagram is a sketch of the outlined shape of the body, which represents it as being *isolated* or "free" from its surroundings, i.e., a "free body." On this sketch it is necessary to show *all* the forces and couple moments that the surroundings exert *on the body* so that these effects can be accounted for when the equations of equilibrium are applied. *A thorough understanding of how to draw a free-body diagram is of primary importance for solving problems in mechanics.*

Support Reactions. Before presenting a formal procedure as to how to draw a free-body diagram, we will first consider the various types of reactions that occur at supports and points of contact between bodies subjected to coplanar force systems. As a general rule,

- If a support prevents the translation of a body in a given direction, then a force is developed on the body in that direction.
- If rotation is prevented, a couple moment is exerted on the body.

For example, let us consider three ways in which a horizontal member, such as a beam, is supported at its end. One method consists of a *roller* or cylinder, Fig. 5–3*a*. Since this support only prevents the beam from *translating* in the vertical direction, the roller will only exert a *force* on the beam in this direction, Fig. 5–3*b*.

The beam can be supported in a more restrictive manner by using a *pin*, Fig. 5–3*c*. The pin passes through a hole in the beam and two leaves which are fixed to the ground. Here the pin can prevent *translation* of the beam in *any direction* ϕ, Fig. 5–3*d*, and so the pin must exert a *force* **F** on the beam in this direction. For purposes of analysis, it is generally easier to represent this resultant force **F** by its two rectangular components $\mathbf{F}_x$ and $\mathbf{F}_y$, Fig. 5–3*e*. If F_x and F_y are known, then F and ϕ can be calculated.

The most restrictive way to support the beam would be to use a *fixed support* as shown in Fig. 5–3*f*. This support will prevent both *translation and rotation* of the beam. To do this a *force and couple moment* must be developed on the beam at its point of connection, Fig. 5–3*g*. As in the case of the pin, the force is usually represented by its rectangular components $\mathbf{F}_x$ and $\mathbf{F}_y$.

Table 5–1 lists other common types of supports for bodies subjected to coplanar force systems. (In all cases, the angle θ is assumed to be known.) Carefully study each of the symbols used to represent these supports and the types of reactions they exert on their contacting members.

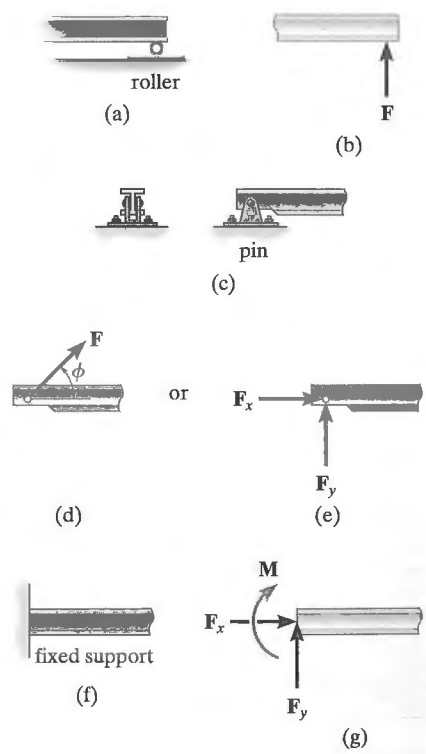
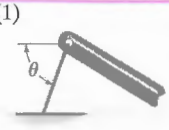
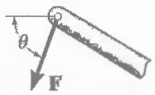
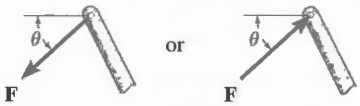
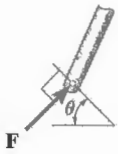
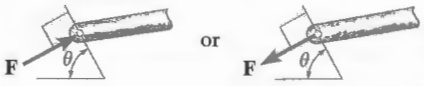
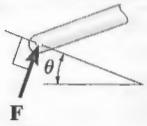
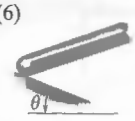
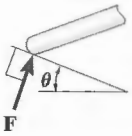
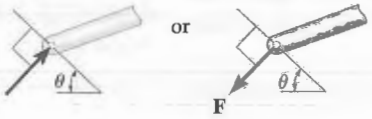


Fig. 5–3

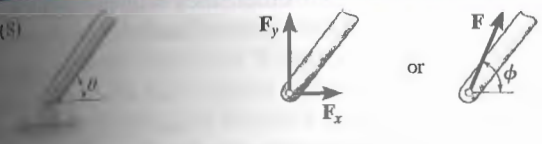
TABLE 5–1 Supports for Rigid Bodies Subjected to Two-Dimensional Force Systems

Types of Connection	Reaction	Number of Unknowns
(1)  cable		One unknown. The reaction is a tension force which acts away from the member in the direction of the cable.
(2) weightless link		One unknown. The reaction is a force which acts along the axis of the link.
(3) roller		One unknown. The reaction is a force which acts perpendicular to the surface at the point of contact.
(4) roller or pin in confined smooth slot		One unknown. The reaction is a force which acts perpendicular to the slot.
(5) rocker		One unknown. The reaction is a force which acts perpendicular to the surface at the point of contact.
(6)  smooth contacting surface		One unknown. The reaction is a force which acts perpendicular to the surface at the point of contact.
(7) member pin connected to collar on smooth rod		One unknown. The reaction is a force which acts perpendicular to the rod.

continued

Custom Publishing exclusively for The University of Melbourne

TABLE 5–1 Continued

Types of Connection	Reaction	Number of Unknowns
(8)  smooth pin or hinge		Two unknowns. The reactions are two components of force, or the magnitude and direction ϕ of the resultant force. Note that ϕ and θ are not necessarily equal [usually not, unless the rod shown is a link as in (2)].
(9) member fixed connected to collar on smooth rod		Two unknowns. The reactions are the couple moment and the force which acts perpendicular to the rod.
(10) fixed support		Three unknowns. The reactions are the couple moment and the two force components, or the couple moment and the magnitude and direction ϕ of the resultant force.

Typical examples of actual supports are shown in the following sequence of photos. The numbers refer to the connection types in Table 5–1.

The cable exerts a force on the bracket in the direction of the cable. (1)

The rocker support for this bridge girder allows horizontal movement so the bridge is free to expand and contract due to a change in temperature. (5)

This concrete girder rests on the ledge that is assumed to act as a smooth contacting surface. (6)

This utility building is pin supported at the top of the column. (8)

The floor beams of this building are welded together and thus form fixed connections. (10)

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

Internal Forces. As stated in Sec. 5.1, the internal forces that act between adjacent particles in a body always occur in collinear pairs such that they have the same magnitude and act in opposite directions (Newton's third law). Since these forces cancel each other, they will not create an *external effect* on the body. It is for this reason that the internal forces should not be included on the free-body diagram if the entire body is to be considered. For example, the engine shown in Fig. 5–4*a* has a free-body diagram shown in Fig. 5–4*b*. The internal forces between all its connected parts, such as the screws and bolts, will cancel out because they form equal and opposite collinear pairs. Only the external forces $\mathbf{T}_1$ and $\mathbf{T}_2$, exerted by the chains and the engine weight **W**, are shown on the free-body diagram.

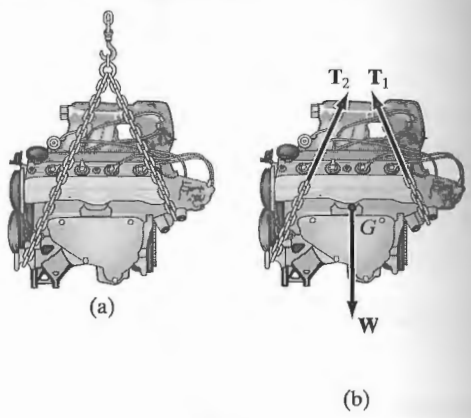


Fig. 5–4

Weight and the Center of Gravity. When a body is within a gravitational field, then each of its particles has a specified weight. It was shown in Sec. 4.8 that such a system of forces can be reduced to a single resultant force acting through a specified point. We refer to this force resultant as the *weight* **W** of the body and to the location of its point of application as the *center of gravity*. The methods used for its determination will be developed in Chapter 9.

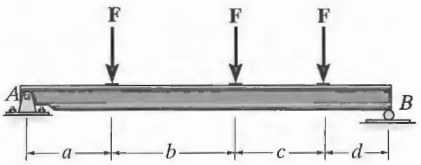
In the examples and problems that follow, if the weight of the body is important for the analysis, this force will be reported in the problem statement. Also, when the body is *uniform* or made from the same material, the center of gravity will be located at the body's *geometric center* or *centroid*; however, if the body consists of a nonuniform distribution of material, or has an unusual shape, then the location of its center of gravity G will be given.

Idealized Models. When an engineer performs a force analysis of any object, he or she considers a corresponding analytical or idealized model that gives results that approximate as closely as possible the actual situation. To do this, careful choices have to be made so that selection of the type of supports, the material behavior, and the object's dimensions can be justified. This way one can feel confident that any design or analysis will yield results which can be trusted. In complex

cases, this process may require developing several different models of the object that must be analyzed. In any case, this selection process requires both skill and experience.

The following two cases illustrate what is required to develop a proper model. In Fig. 5–5*a*, the steel beam is to be used to support the three roof joists of a building. For a force analysis, it is reasonable to assume the material (steel) is rigid since only very small deflections will occur when the beam is loaded. A bolted connection at *A* will allow for any slight rotation that occurs here when the load is applied, and so a *pin* can be considered for this support. At *B*, a *roller* can be considered since this support offers no resistance to horizontal movement. Building code is used to specify the roof loading *A* so that the joist loads **F** can be calculated. These forces will be larger than any actual loading on the beam since they account for extreme loading cases and for dynamic or vibrational effects. Finally, the weight of the beam is generally neglected when it is small compared to the load the beam supports. The idealized model of the beam is therefore shown with average dimensions *a, b, c,* and *d* in Fig. 5–5*b*.

(a)



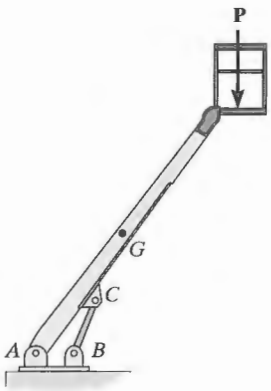
(b)

Fig. 5–5

As a second case, consider the lift boom in Fig. 5–6*a*. By inspection, it is supported by a pin at *A* and by the hydraulic cylinder *BC*, which can be approximated as a weightless link. The material can be assumed rigid, and with its density known, the weight of the boom and the location of its center of gravity *G* are determined. When a design loading **P** is specified, the idealized model shown in Fig. 5–6*b* can be used for a force analysis. Average dimensions (not shown) are used to specify the location of the loads and the supports.

Idealized models of specific objects will be given in some of the examples throughout the text. It should be realized, however, that each case represents the reduction of a practical situation using simplifying assumptions like the ones illustrated here.

(a)



(b)

Fig. 5–6

Procedure for Analysis

To construct a free-body diagram for a rigid body or any group of bodies considered as a single system, the following steps should be performed:

Draw Outlined Shape.

Imagine the body to be *isolated* or cut "free" from its constraints and connections and draw (sketch) its outlined shape.

Show All Forces and Couple Moments.

Identify all the known and unknown *external forces* and couple moments that *act on the body*. Those generally encountered are due to (1) applied loadings, (2) reactions occurring at the supports or at points of contact with other bodies (see Table 5–1), and (3) the weight of the body. To account for all these effects, it may help to trace over the boundary, carefully noting each force or couple moment acting on it.

Identify Each Loading and Give Dimensions.

The forces and couple moments that are known should be labeled with their proper magnitudes and directions. Letters are used to represent the magnitudes and direction angles of forces and couple moments that are unknown. Establish an x, y coordinate system so that these unknowns, A_x, A_y, etc., can be identified. Finally indicate the dimensions of the body necessary for calculating the moments of forces.

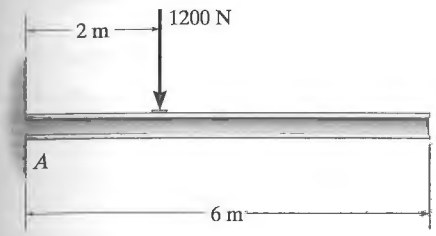
Important Points

- No equilibrium problem should be solved without *first drawing the free-body diagram*, so as to account for all the forces and couple moments that act on the body.
- If a support *prevents translation* of a body in a particular direction, then the support exerts a *force* on the body in that direction.
- If *rotation is prevented*, then the support exerts a *couple moment* on the body.
- Study Table 5–1.
- Internal forces are never shown on the free-body diagram since they occur in equal but opposite collinear pairs and therefore cancel out.
- The weight of a body is an external force, and its effect is represented by a single resultant force acting through the body's center of gravity G.
- *Couple moments* can be placed anywhere on the free-body diagram since they are *free vectors*. *Forces* can act at any point along their lines of action since they are *sliding vectors*.

ing exclusively for The University of Melbourne

EXAMPLE 5.1

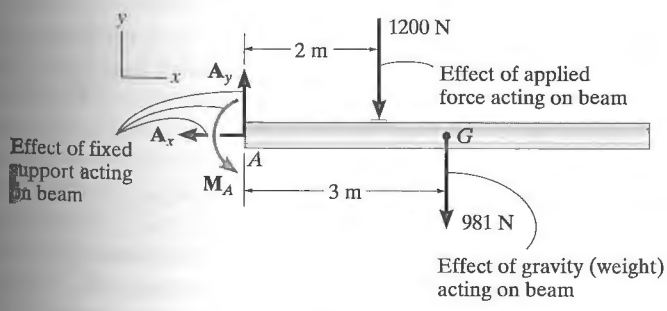
Draw the free-body diagram of the uniform beam shown in Fig. 5–7*a*. The beam has a mass of 100 kg.



(a)

SOLUTION

The free-body diagram of the beam is shown in Fig. 5–7*b*. Since the support at A is fixed, the wall exerts three reactions *on the beam*, denoted as $\mathbf{A}_x$, $\mathbf{A}_y$, and $\mathbf{M}_A$. The magnitudes of these reactions are *unknown*, and their sense has been *assumed*. The weight of the beam, $W = 100(9.81)\text{ N} = 981\text{ N}$, acts through the beam's center of gravity G, which is 3 m from A since the beam is uniform.



(b)

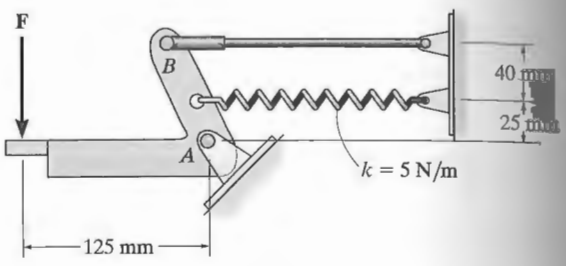
Fig. 5–7

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

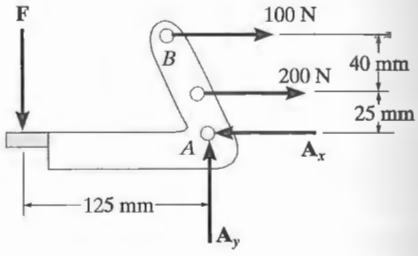
EXAMPLE 5.2

Draw the free-body diagram of the foot lever shown in Fig. 5–8*a*. The operator applies a vertical force to the pedal so that the spring is stretched 40 mm and the force in the short link at B is 100 N.

(a)



(b)



(c)

Fig. 5–8

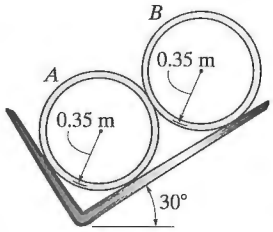
SOLUTION

By inspection of the photo the lever is loosely bolted to the frame at A. The rod at B is pinned at its ends and acts as a "short link." After making the proper measurements, the idealized model of the lever is shown in Fig. 5–8*b*. From this, the free-body diagram is shown in Fig. 5–8*c*. The pin support at A exerts force components $\mathbf{A}_x$ and $\mathbf{A}_y$ on the lever. The link at B exerts a force of 100 N, acting in the direction of the link. In addition the spring also exerts a horizontal force on the lever. If the stiffness is measured and found to be $k = 5\ \text{N/m}$, then since the stretch $s = 40$ mm, using Eq. 3–2, $F_s = ks = 5\ \text{N/m}\ (40\ \text{mm}) = 200$ N. Finally, the operator's shoe applies a vertical force of $\mathbf{F}$ on the pedal. The dimensions of the lever are also shown on the free-body diagram, since this information will be useful when computing the moments of the forces. As usual, the senses of the unknown forces at A have been assumed. The correct senses will become apparent after solving the equilibrium equations.

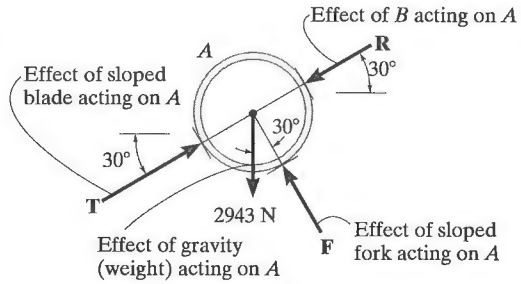
EXAMPLE 5.3

Two smooth pipes, each having a mass of 300 kg, are supported by the forked tines of the tractor in Fig. 5–9*a*. Draw the free-body diagrams for each pipe and both pipes together.

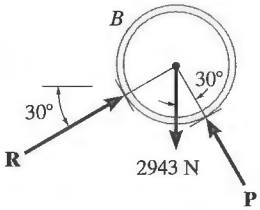
(a)



(b)



(c)



(d)

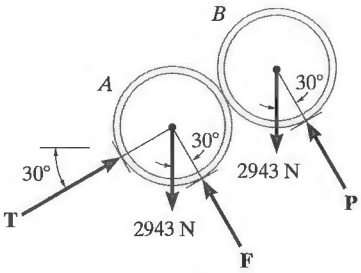
SOLUTION

The idealized model from which we must draw the free-body diagrams is shown in Fig. 5–9*b*. Here the pipes are identified, the dimensions have been added, and the physical situation reduced to its simplest form.

The free-body diagram for pipe A is shown in Fig. 5–9*c*. Its weight is $W = 300(9.81)\text{ N} = 2943\text{ N}$. Assuming all contacting surfaces are *smooth*, the reactive forces **T**, **F**, **R** act in a direction *normal* to the tangent at their surfaces of contact.

The free-body diagram of pipe B is shown in Fig. 5–9*d*. Can you identify each of the three forces acting *on this pipe*? In particular, note that **R**, representing the force of A on B, Fig. 5–9*d*, is equal and opposite to **R** representing the force of B on A, Fig. 5–9*c*. This is a consequence of Newton's third law of motion.

The free-body diagram of both pipes combined ("system") is shown in Fig. 5–9*e*. Here the contact force **R**, which acts between A and B, is considered as an *internal* force and hence is not shown on the free-body diagram. That is, it represents a pair of equal but opposite collinear forces which cancel each other.

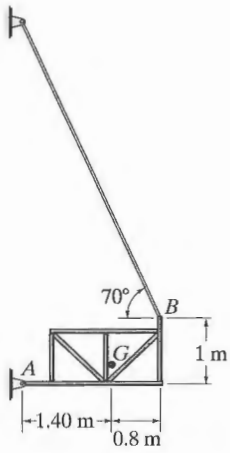


(e)

Fig. 5–9

EXAMPLE 5.4

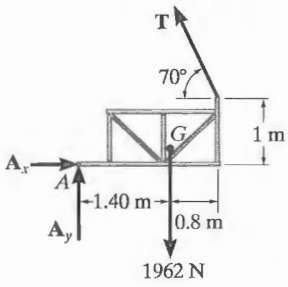
Draw the free-body diagram of the unloaded platform that is suspended off the edge of the oil rig shown in Fig. 5–10*a*. The platform has a mass of 200 kg.



(b)

(a)

Fig. 5–10



(c)

SOLUTION

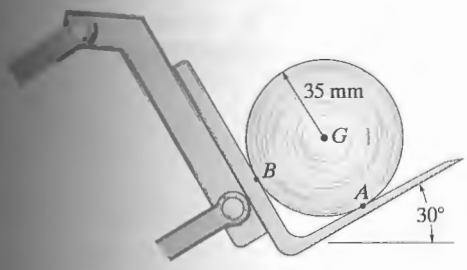
The idealized model of the platform will be considered in two dimensions because by observation the loading and the dimensions are all symmetrical about a vertical plane passing through its center, Fig. 5–10*b*. The connection at A is considered to be a pin, and the cable supports the platform at B. The direction of the cable and average dimensions of the platform are listed, and the center of gravity G has been determined. It is from this model that we have drawn the free-body diagram shown in Fig. 5–10*c*. The platform's weight is $200(9.81) = 1962$ N. The force components $\mathbf{A}_x$ and $\mathbf{A}_y$ along with the cable force $\mathbf{T}$ represent the reactions that *both* pins and *both* cables exert on the platform, Fig. 5–10*a*. Consequently, after the solution for these reactions, half their magnitude is developed at A and half is developed at B.

exclusively for The University of Melbourne

ENGINEERING™
mized coaching is available!

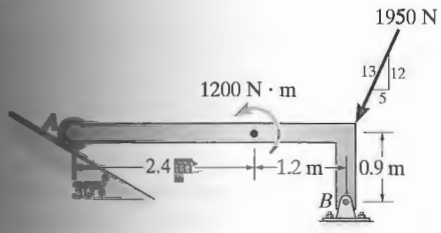
PROBLEMS

•5–1. Draw the free-body diagram of the 50-kg paper roll which has a center of mass at G and rests on the smooth blade of the paper hauler. Explain the significance of each force acting on the diagram. (See Fig. 5–7*b*.)



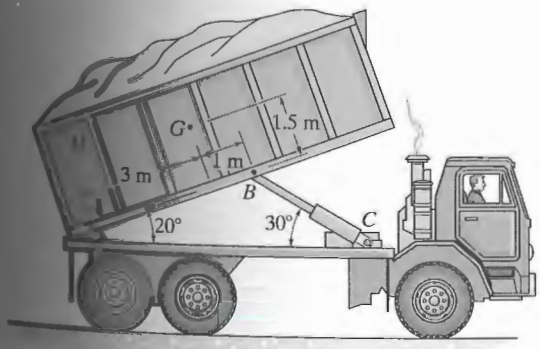
Prob. 5–1

5–2. Draw the free-body diagram of member AB, which is supported by a roller at A and a pin at B. Explain the significance of each force on the diagram. (See Fig. 5–7*b*.)



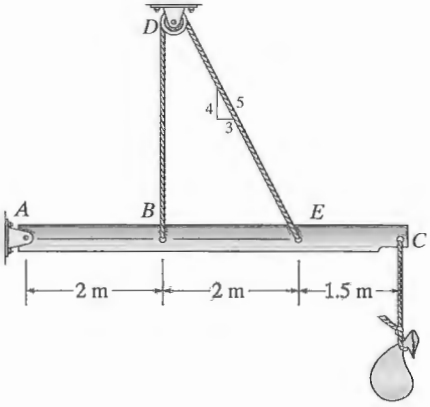
Prob. 5–2

5–3. Draw the free-body diagram of the dumpster D of the truck, which has a weight of 25 kN and a center of gravity at G. It is supported by a pin at A and a pin-connected hydraulic cylinder BC (short link). Explain the significance of each force on the diagram. (See Fig. 5–7*b*.)



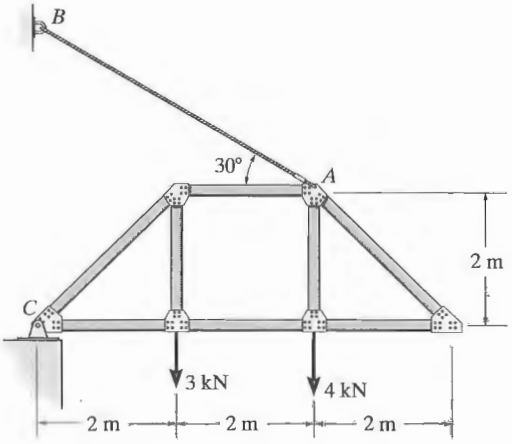
Prob. 5–3

*5–4. Draw the free-body diagram of the beam which supports the 80-kg load and is supported by the pin at A and a cable which wraps around the pulley at D. Explain the significance of each force on the diagram. (See Fig. 5–7*b*.)



Prob. 5–4

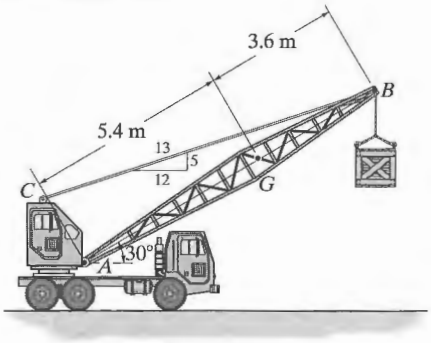
•5–5. Draw the free-body diagram of the truss that is supported by the cable AB and pin C. Explain the significance of each force acting on the diagram. (See Fig. 5–7*b*.)



Prob. 5–5

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

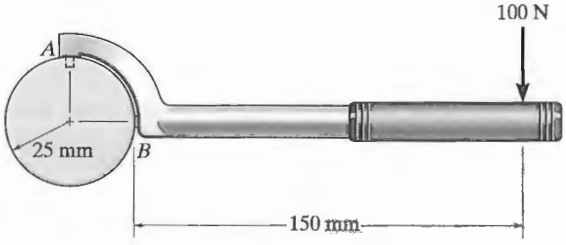
5–6. Draw the free-body diagram of the crane boom AB which has a weight of 3.25 kN and center of gravity at G. The boom is supported by a pin at A and cable BC. The load of 6.25 kN is suspended from a cable attached at B. Explain the significance of each force acting on the diagram. (See Fig. 5–7*b*.)



Prob. 5–6

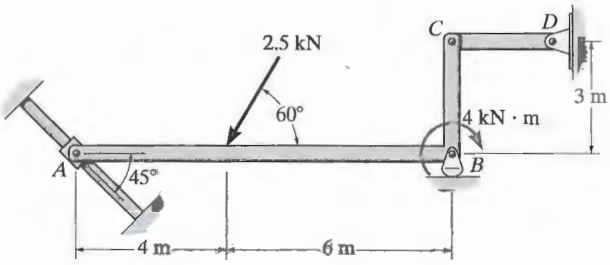
5

5–7. Draw the free-body diagram of the "spanner wrench" subjected to the 100-N force. The support at A can be considered a pin, and the surface of contact at B is smooth. Explain the significance of each force on the diagram. (See Fig. 5–7*b*.)



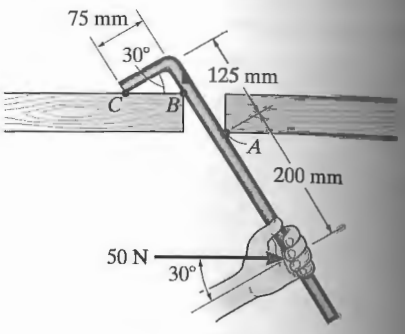
Prob. 5–7

***5–8.** Draw the free-body diagram of member ABC which is supported by a smooth collar at A, roller at B, and short link CD. Explain the significance of each force acting on the diagram. (See Fig. 5–7*b*.)



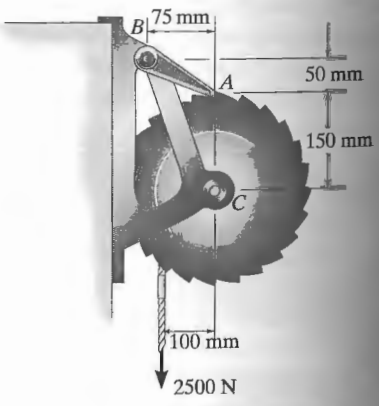
Prob. 5–8

•5–9. Draw the free-body diagram of the bar, which has a negligible thickness and smooth points of contact at A, B, and C. Explain the significance of each force on the diagram. (See Fig. 5–7*b*.)



Prob. 5–9

5–10. Draw the free-body diagram of the winch, which consists of a drum of radius 100 mm. It is pin-connected at its center C, and at its outer rim is a ratchet gear having a mean radius of 150 mm. The pawl AB serves as a two-force member (short link) and prevents the drum from rotating. Explain the significance of each force on the diagram. (See Fig. 5–7*b*.)



Prob. 5–10

CONCEPTUAL PROBLEMS

P5–1. Draw the free-body diagram of the uniform trash bucket which has a significant weight. It is pinned at A and rests against the smooth horizontal member at B. Show your result in side view. Label any necessary dimensions.

P5–1

P5–2. Draw the free-body diagram of the outrigger ABC used to support a backhoe. The top pin B is connected to the hydraulic cylinder, which can be considered to be a short link (two-force member), the bearing shoe at A is smooth, and the outrigger is pinned to the frame at C.

P5–2

P5–3. Draw the free-body diagram of the wing on the passenger plane. The weights of the engine and wing are significant. The tires at B are smooth.

P5–3

***P5–4.** Draw the free-body diagram of the wheel and member ABC used as part of the landing gear on a jet plane. The hydraulic cylinder AD acts as a two-force member, and there is a pin connection at B.

P5–4

EXAMPLE 5.5

Determine the horizontal and vertical components of reaction on the beam caused by the pin at B and the rocker at A as shown in Fig. 5–12a. Neglect the weight of the beam.

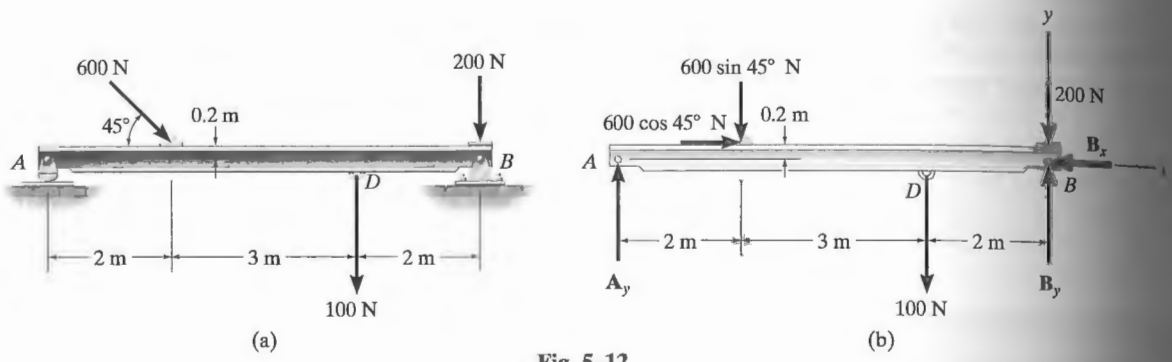


Fig. 5–12

5

SOLUTION

Free-Body Diagram. Identify each of the forces shown on the free-body diagram of the beam, Fig. 5–12b. (See Example 5.1.) For simplicity, the 600-N force is represented by its x and y components as shown in Fig. 5–12b.

Equations of Equilibrium. Summing forces in the x direction yields

$$\xrightarrow{+} \Sigma F_x = 0; \qquad 600 \cos 45° \text{ N} - B_x = 0$$

$$B_x = 424 \text{ N} \qquad \textit{Ans.}$$

A direct solution for $\mathbf{A}_y$ can be obtained by applying the moment equation $\Sigma M_B = 0$ about point B.

$$\circlearrowleft + \Sigma M_B = 0; \qquad 100 \text{ N}(2 \text{ m}) + (600 \sin 45° \text{ N})(5 \text{ m}) - (600 \cos 45° \text{ N})(0.2 \text{ m}) - A_y(7 \text{ m}) = 0$$

$$A_y = 319 \text{ N} \qquad \textit{Ans.}$$

Summing forces in the y direction, using this result, gives

$$+\uparrow \Sigma F_y = 0; \qquad 319 \text{ N} - 600 \sin 45° \text{ N} - 100 \text{ N} - 200 \text{ N} + B_y = 0$$

$$B_y = 405 \text{ N} \qquad \textit{Ans.}$$

NOTE: We can check this result by summing moments about point A.

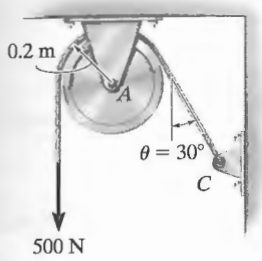
$$\circlearrowleft + \Sigma M_A = 0; \qquad -(600 \sin 45° \text{ N})(2 \text{ m}) - (600 \cos 45° \text{ N})(0.2 \text{ m}) - (100 \text{ N})(5 \text{ m}) - (200 \text{ N})(7 \text{ m}) + B_y(7 \text{ m}) = 0$$

$$B_y = 405 \text{ N} \qquad \textit{Ans.}$$

Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 5.6

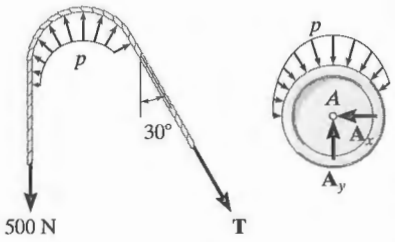
The cord shown in Fig. 5–13*a* supports a force of 500 N and wraps over the frictionless pulley. Determine the tension in the cord at C and the horizontal and vertical components of reaction at pin A.



(a)

SOLUTION

Free-Body Diagrams. The free-body diagrams of the cord and pulley are shown in Fig. 5–13*b*. Note that the principle of action, equal but opposite reaction must be carefully observed when drawing each of these diagrams: the cord exerts an unknown load distribution p on the pulley at the contact surface, whereas the pulley exerts an equal but opposite effect on the cord. For the solution, however, it is simpler to *combine* the free-body diagrams of the pulley and this portion of the cord, so that the distributed load becomes *internal* to this "system" and is therefore eliminated from the analysis, Fig. 5–13*c*.



(b)

Equations of Equilibrium. Summing moments about point A to eliminate $\mathbf{A}_x$ and $\mathbf{A}_y$, Fig. 5–13*c*, we have

$$\circlearrowleft + \Sigma M_A = 0; \qquad 500\text{ N}\,(0.2\text{ m}) - T(0.2\text{ m}) = 0$$

$$T = 500\text{ N} \qquad \textit{Ans.}$$

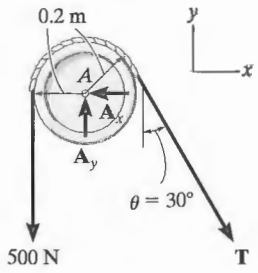
Using the result,

$$\xrightarrow{+} \Sigma F_x = 0; \qquad -A_x + 500 \sin 30^\circ\text{ N} = 0$$

$$A_x = 250\text{ N} \qquad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \qquad A_y - 500\text{ N} - 500 \cos 30^\circ\text{ N} = 0$$

$$A_y = 933\text{ N} \qquad \textit{Ans.}$$



(c)

Fig. 5–13

NOTE: It is seen that the tension remains *constant* as the cord passes over the pulley. (This of course is true for *any angle* θ at which the cord is directed and for *any radius* r of the pulley.)

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 5.7

The member shown in Fig. 5–14*a* is pin-connected at A and rests against a smooth support at B. Determine the horizontal and vertical components of reaction at the pin A.

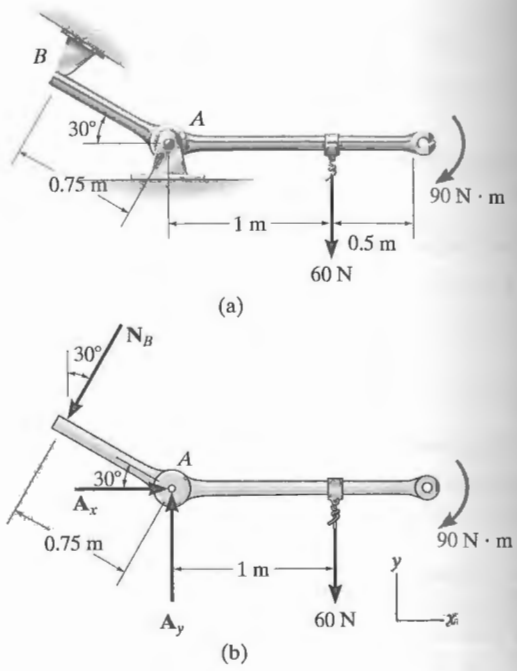


Fig. 5–14

SOLUTION

Free-Body Diagram. As shown in Fig. 5–14*b*, the reaction $\mathbf{N}_B$ is perpendicular to the member at B. Also, horizontal and vertical components of reaction are represented at A.

Equations of Equilibrium. Summing moments about A, we obtain a direct solution for N_B,

$$\circlearrowleft + \Sigma M_A = 0; \quad -90\ \text{N}\cdot\text{m} - 60\ \text{N}(1\ \text{m}) + N_B(0.75\ \text{m}) = 0$$

$$N_B = 200\ \text{N}$$

Using this result,

$$\xrightarrow{+} \Sigma F_x = 0; \qquad A_x - 200 \sin 30^\circ\ \text{N} = 0$$

$$A_x = 100\ \text{N} \qquad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \qquad A_y - 200 \cos 30^\circ\ \text{N} - 60\ \text{N} = 0$$

$$A_y = 233\ \text{N} \qquad \textit{Ans.}$$

EXAMPLE 5.8

The box wrench in Fig. 5–15*a* is used to tighten the bolt at A. If the wrench *does not* turn when the load is applied to the handle, determine the torque or moment applied to the bolt and the force of the wrench on the bolt.

SOLUTION

Free-Body Diagram. The free-body diagram for the wrench is shown in Fig. 5–15*b*. Since the bolt acts as a "fixed support," it exerts force components $\mathbf{A}_x$ and $\mathbf{A}_y$ and a moment $\mathbf{M}_A$ on the wrench at A.

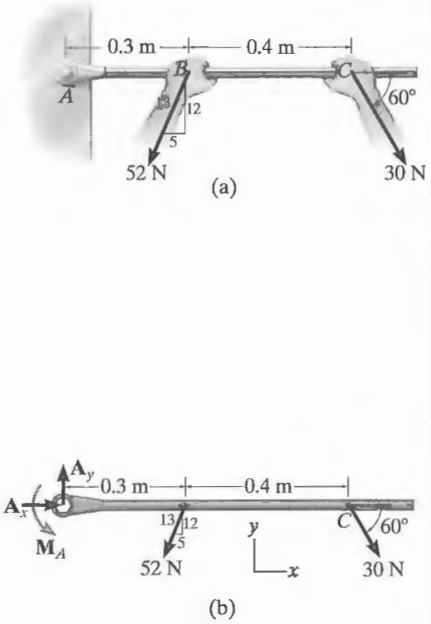


Fig. 5–15

Equations of Equilibrium.

$$\xrightarrow{+} \Sigma F_x = 0; \qquad A_x - 52\left(\tfrac{5}{13}\right)\text{N} + 30\cos 60^\circ\ \text{N} = 0$$

$$A_x = 5.00\ \text{N} \qquad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \qquad A_y - 52\left(\tfrac{12}{13}\right)\text{N} - 30\sin 60^\circ\ \text{N} = 0$$

$$A_y = 74.0\ \text{N} \qquad \textit{Ans.}$$

$$\circlearrowleft + \Sigma M_A = 0;\ M_A - \left[52\left(\tfrac{12}{13}\right)\text{N}\right](0.3\ \text{m}) - (30\sin 60^\circ\ \text{N})(0.7\ \text{m}) = 0$$

$$M_A = 32.6\ \text{N}\cdot\text{m} \qquad \textit{Ans.}$$

Note that $\mathbf{M}_A$ must be *included* in this moment summation. This couple moment is a free vector and represents the twisting resistance of the bolt on the wrench. By Newton's third law, the wrench exerts an equal but opposite moment or torque on the bolt. Furthermore, the resultant force on the wrench is

$$F_A = \sqrt{(5.00)^2 + (74.0)^2} = 74.1\ \text{N} \qquad \textit{Ans.}$$

NOTE: Although only *three* independent equilibrium equations can be written for a rigid body, it is a good practice to *check* the calculations using a fourth equilibrium equation. For example, the above computations may be verified in part by summing moments about point C:

$$\circlearrowleft + \Sigma M_C = 0;\ \left[52\left(\tfrac{12}{13}\right)\text{N}\right](0.4\ \text{m}) + 32.6\ \text{N}\cdot\text{m} - 74.0\ \text{N}(0.7\ \text{m}) = 0$$

$$19.2\ \text{N}\cdot\text{m} + 32.6\ \text{N}\cdot\text{m} - 51.8\ \text{N}\cdot\text{m} = 0$$

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 5.9

Determine the horizontal and vertical components of reaction on the member at the pin A, and the normal reaction at the roller B in Fig. 5–16*a*.

Please refer to the Companion CD for the animation: *Free-body Diagram for a Beam on Slanting Support*

SOLUTION

Free-Body Diagram. The free-body diagram is shown in Fig. 5–16*b*. The pin at A exerts two components of reaction on the member, $\mathbf{A}_x$ and $\mathbf{A}_y$.

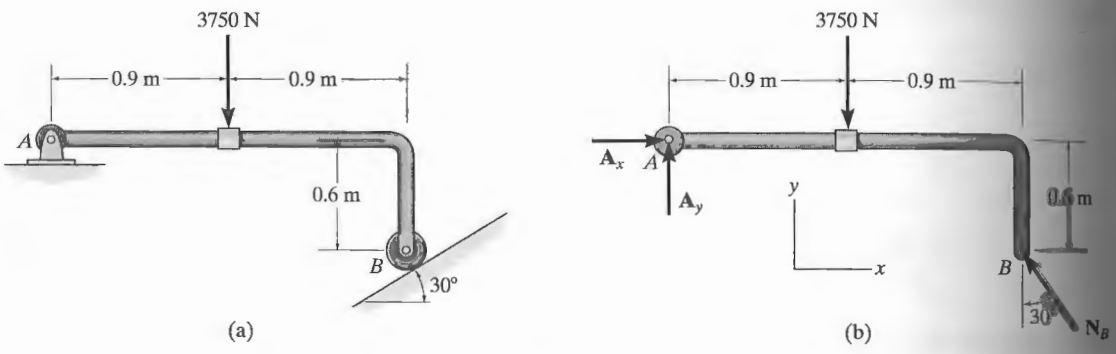


Fig. 5–16

Equations of Equilibrium. The reaction N_B can be obtained *directly* by summing moments about point A since $\mathbf{A}_x$ and $\mathbf{A}_y$ produce no moment about A.

$$\circlearrowleft + \Sigma M_A = 0;$$

$$[N_B \cos 30°](1.8 \text{ m}) - [N_B \sin 30°](0.6 \text{ m}) - 3750 \text{ N}(0.9 \text{ m}) = 0$$

$$N_B = 2681 \text{ N} \qquad \textit{Ans.}$$

Using this result,

$$\xrightarrow{+} \Sigma F_x = 0; \qquad A_x - (2681 \text{ N}) \sin 30° = 0$$

$$A_x = 1340.5 \text{ N} \qquad \textit{Ans.}$$

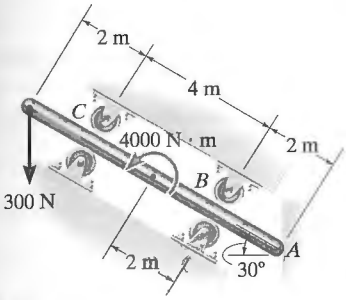
$$+\uparrow \Sigma F_y = 0; \qquad A_y + (2681 \text{ N}) \cos 30° - 3750 \text{ N} = 0$$

$$A_y = 1428.2 \text{ N} \qquad \textit{Ans.}$$

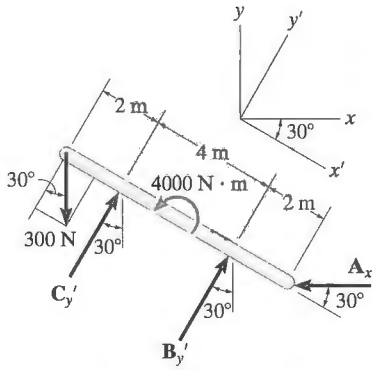
ENGR10003 – EN
A publication from Pe
lishing exclusively for The University of Melbourne

EX 5.10

The uniform smooth rod shown in Fig. 5–17*a* is subjected to a force couple moment. If the rod is supported at *A* by a smooth wall and and *C* either at the top or bottom by rollers, determine the reactions at these supports. Neglect the weight of the rod.



(a)



(b)

Fig. 5–17

SOLUTION

Free-Body Diagram. As shown in Fig. 5–17*b*, all the support reactions act normal to the surfaces of contact since these surfaces are smooth. The reactions at *B* and *C* are shown acting in the positive y' direction. This assumes that only the rollers located on the bottom of the rod are used for support.

Equations of Equilibrium. Using the x, y coordinate system in Fig. 5–17*b*, we have

$$\xrightarrow{+} \Sigma F_x = 0; \quad C_{y'} \sin 30° + B_{y'} \sin 30° - A_x = 0 \qquad (1)$$

$$+\uparrow \Sigma F_y = 0; \quad -300 \text{ N} + C_{y'} \cos 30° + B_{y'} \cos 30° = 0 \qquad (2)$$

$$\circlearrowleft + \Sigma M_A = 0; \quad -B_{y'}(2 \text{ m}) + 4000 \text{ N} \cdot \text{m} - C_{y'}(6 \text{ m}) + (300 \cos 30° \text{ N})(8 \text{ m}) = 0 \qquad (3)$$

When writing the moment equation, it should be noted that the line of action of the force component 300 sin 30° N passes through point *A*, and therefore this force is not included in the moment equation.

Solving Eqs. 2 and 3 simultaneously, we obtain

$$B_{y'} = -1000.0 \text{ N} = -1 \text{ kN} \qquad \textit{Ans.}$$

$$C_{y'} = 1346.4 \text{ N} = 1.35 \text{ kN} \qquad \textit{Ans.}$$

Since is a negative scalar, the sense of $\mathbf{B}_{y'}$ is opposite to that shown on the free-body diagram in Fig. 5–17*b*. Therefore, the top roller at *B* serves as the support rather than the bottom one. Retaining the negative sign for $B_{y'}$ (Why?) and substituting the results into Eq. 1, we obtain

$$1346.4 \sin 30° \text{ N} + (-1000.0 \sin 30° \text{ N}) - A_x = 0$$

$$A_x = 173 \text{ N} \qquad \textit{Ans.}$$

5

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 5.11

(a)

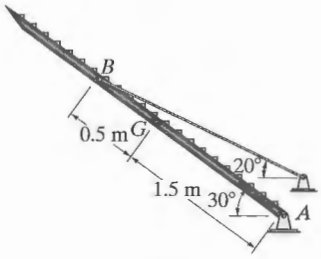
The uniform truck ramp shown in Fig. 5–18*a* has a weight of 1600 N and is pinned to the body of the truck at each side and held in the position shown by the two side cables. Determine the tension in the cables.

SOLUTION

The idealized model of the ramp, which indicates all necessary dimensions and supports, is shown in Fig. 5–18*b*. Here the center of gravity is located at the midpoint since the ramp is considered to be uniform.

Free-Body Diagram. Working from the idealized model, the ramp's free-body diagram is shown in Fig. 5–18*c*.

Equations of Equilibrium. Summing moments about point A will yield a direct solution for the cable tension. Using the principle of moments, there are several ways of determining the moment of $\mathbf{T}$ about A. If we use x and y components, with $\mathbf{T}$ applied at B, we have



(b)

$$\circlearrowleft + \Sigma M_A = 0; \qquad -T\cos 20°(2\sin 30°\text{ m}) + T\sin 20°(2\cos 30°\text{ m})$$

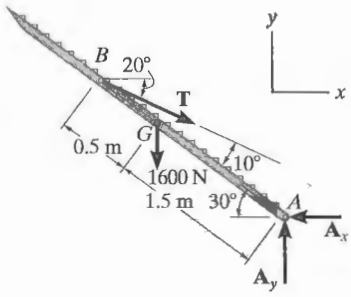
$$+ 1600\text{ N}\,(1.5\cos 30°\text{ m}) = 0$$

$$T = 5985\text{ N}$$

The simplest way to determine the moment of $\mathbf{T}$ about A is to resolve it into components along and perpendicular to the ramp at B. Then the moment of the component along the ramp will be zero about A, so that

$$\circlearrowleft + \Sigma M_A = 0; \qquad -T\sin 10°(2\text{ m}) + 1600\text{ N}\,(1.5\cos 30°\text{ m}) = 0$$

$$T = 5985\text{ N}$$



(c)

Fig. 5–18

Since there are two cables supporting the ramp,

$$T' = \frac{T}{2} = 2992.5\text{ N} \qquad \textit{Ans.}$$

NOTE: As an exercise, show that $A_x = 5624$ N and $A_y = 3647$ N.

5.12

Determine the support reactions on the member in Fig. 5–19*a*. The collar at *A* is fixed to the member and can slide vertically along the vertical shaft.

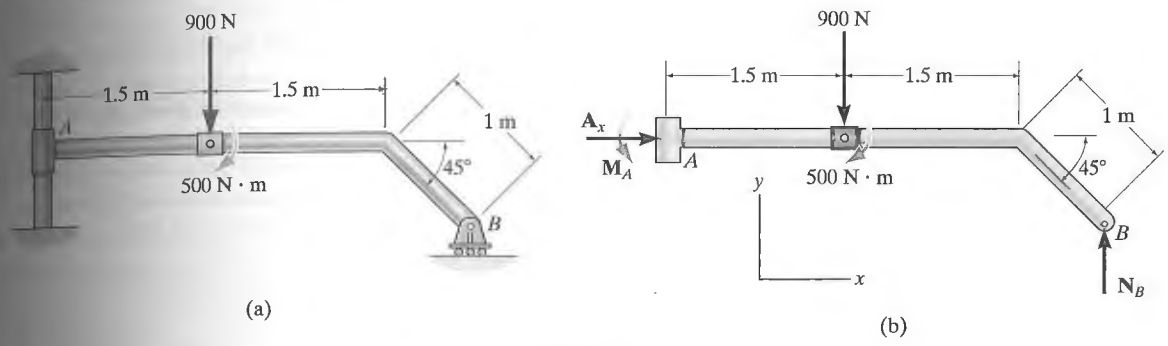


Fig. 5–19

5

SOLUTION

Free-Body Diagram. The free-body diagram of the member is shown in Fig. 5–19*b*. The collar exerts a horizontal force $\mathbf{A}_x$ and moment $\mathbf{M}_A$ on the member. The reaction $\mathbf{N}_B$ of the roller on the member is vertical.

Equations of Equilibrium. The forces A_x and N_B can be determined directly from the force equations of equilibrium.

$\xrightarrow{+} \Sigma F_x = 0;$ $\quad A_x = 0$ *Ans.*

$+\uparrow \Sigma F_y = 0;$ $\quad N_B - 900\text{ N} = 0$

$N_B = 900\text{ N}$ *Ans.*

The moment M_A can be determined by summing moments either about point *A* or point *B*.

$\circlearrowleft + \Sigma M_A = 0;$

$$M_A - 900\text{ N}(1.5\text{ m}) - 500\text{ N}\cdot\text{m} + 900\text{ N}[3\text{ m} + (1\text{ m})\cos 45°] = 0$$

$$M_A = -1486\text{ N}\cdot\text{m} = 1.49\text{ kN}\cdot\text{m}\circlearrowright$$ *Ans.*

or

$\circlearrowleft + \Sigma M_B = 0;$ $\quad M_A + 900\text{ N}[1.5\text{ m} + (1\text{ m})\cos 45°] - 500\text{ N}\cdot\text{m} = 0$

$$M_A = -1486\text{ N}\cdot\text{m} = 1.49\text{ kN}\cdot\text{m}\circlearrowright$$ *Ans.*

The negative sign indicates that $\mathbf{M}_A$ has the opposite sense of rotation to that shown on the free-body diagram.

The bucket link AB on the back-hoe is a typical example of a two-force member since it is pin connected at its ends and, provided its weight is neglected, only the pin forces on this member.

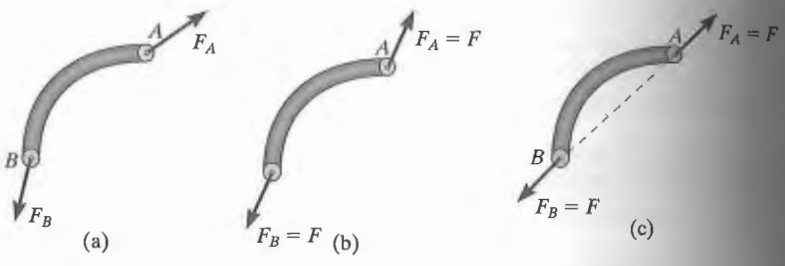
The link used for this railroad car brake is a three-force member. Since the force $\mathbf{F}_B$ in the tie rod at B and $\mathbf{F}_C$ from the link at C are parallel, then for equilibrium the resultant force $\mathbf{F}_A$ at pin A must also be parallel with these two forces.

The boom and bucket on this lift is a three-force member, provided its weight is neglected. Here the lines of action of the weight of the worker, $\mathbf{W}$, and the force of the two-force member (hydraulic cylinder) at B, $\mathbf{F}_B$, intersect at O. For moment equilibrium, the resultant force at pin A, $\mathbf{F}_A$, must also be directed towards O.

5.4 Two- and Three-Force Members

The solutions to some equilibrium problems can be simplified by recognizing members that are subjected to only two or three forces.

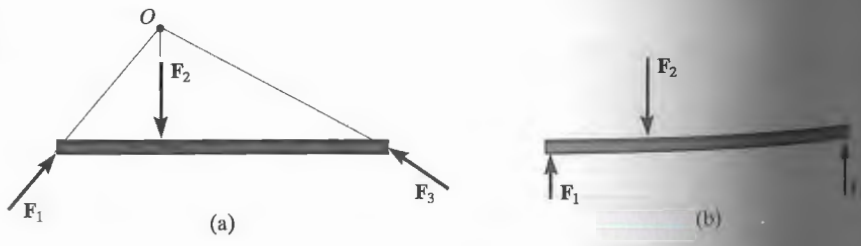
Two-Force Members As the name implies, a *two-force member* has forces applied at only two points on the member. An example of a two-force member is shown in Fig. 5–20*a*. To satisfy force equilibrium $\mathbf{F}_A$ and $\mathbf{F}_B$ must be equal in magnitude, $F_A = F_B = F$, but opposite in direction ($\Sigma\mathbf{F} = \mathbf{0}$), Fig. 5–20*b*. Furthermore, moment equilibrium requires that $\mathbf{F}_A$ and $\mathbf{F}_B$ share the same line of action, which can only happen if they are directed along the line joining points A and B ($\Sigma\mathbf{M}_A = \mathbf{0}$ or $\Sigma\mathbf{M}_B = \mathbf{0}$), Fig. 5–20*c*. Therefore, for any two-force member to be in equilibrium the two forces acting on the member *must have the same magnitude, act in opposite directions, and have the same line of action, directed along the line joining the two points where these forces act.*



Two-force member

Fig. 5–20

Three-Force Members If a member is subjected to only *three forces*, it is called a *three-force member*. Moment equilibrium can be satisfied only if the three forces form a *concurrent* or *parallel* force system. To illustrate, consider the member subjected to the three forces $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$, shown in Fig. 5–21*a*. If the lines of action of $\mathbf{F}_1$ and $\mathbf{F}_2$ intersect at point O, then the line of action of $\mathbf{F}_3$ must *also* pass through point O so that the forces satisfy $\Sigma\mathbf{M}_O = \mathbf{0}$. As a special case, if the three forces are all parallel, Fig. 5–21*b*, the location of the point of intersection, O, will approach infinity.



Three-force member

Fig. 5–21

ing exclusively for The University of Melbourne

EXAMPLE 5.13

The lever ABC is pin supported at A and connected to a short link BD as shown in Fig. 5–22a. If the weight of the members is negligible, determine the force of the pin on the lever at A.

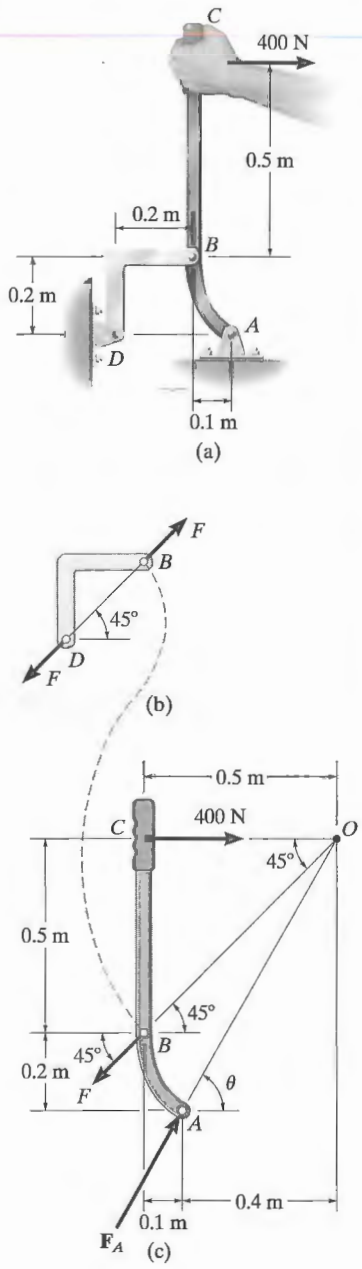


Fig. 5–22

SOLUTION

Free-Body Diagrams. As shown in Fig. 5–22b, the short link BD is a *two-force member*, so the *resultant forces* at pins D and B must be equal, opposite, and collinear. Although the magnitude of the force is unknown, the line of action is known since it passes through B and D.

Lever ABC is a *three-force member*, and therefore, in order to satisfy moment equilibrium, the three nonparallel forces acting on it must be concurrent at O, Fig. 5–22c. In particular, note that the force $\mathbf{F}$ on the lever at B is equal but opposite to the force $\mathbf{F}$ acting at B on the link. Why? The distance CO must be 0.5 m since the lines of action of $\mathbf{F}$ and the 400-N force are known.

Equations of Equilibrium. By requiring the force system to be concurrent at O, since $\Sigma M_O = 0$, the angle θ which defines the line of action of $\mathbf{F}_A$ can be determined from trigonometry,

$$\theta = \tan^{-1}\left(\frac{0.7}{0.4}\right) = 60.3°$$

Using the x, y axes and applying the force equilibrium equations,

$$\overset{+}{\rightarrow}\Sigma F_x = 0; \qquad F_A \cos 60.3° - F \cos 45° + 400\text{ N} = 0$$

$$+\uparrow\Sigma F_y = 0; \qquad F_A \sin 60.3° - F \sin 45° = 0$$

Solving, we get

$$F_A = 1.07\text{ kN} \qquad \textit{Ans.}$$

$$F = 1.32\text{ kN}$$

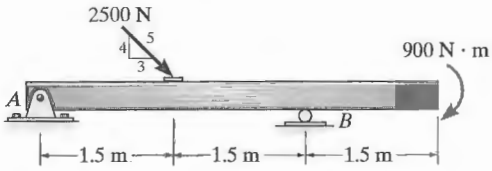
NOTE: We can also solve this problem by representing the force at A by its two components $\mathbf{A}_x$ and $\mathbf{A}_y$ and applying $\Sigma M_A = 0$, $\Sigma F_x = 0$, $\Sigma F_y = 0$ to the lever. Once A_x and A_y are determined, we can get F_A and θ.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

FUNDAMENTAL PROBLEMS

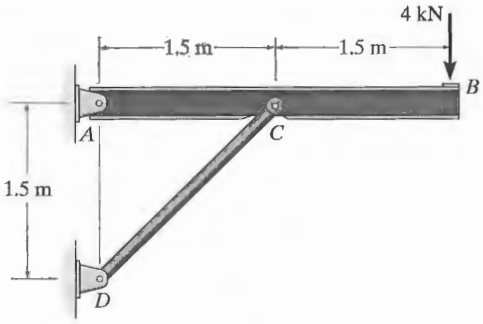
All problem solutions must include an FBD.

F5–1. Determine the horizontal and vertical components of reaction at the supports. Neglect the thickness of the beam.



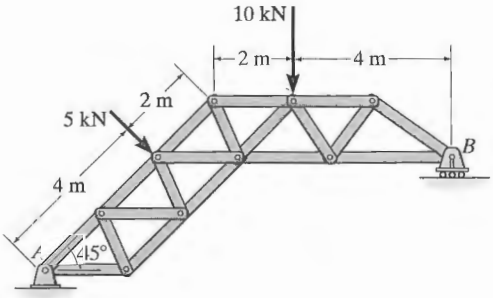
F5–1

F5–2. Determine the horizontal and vertical components of reaction at the pin A and the reaction on the beam at C.



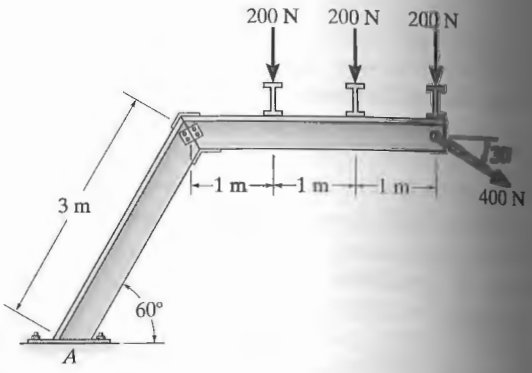
F5–22

F5–3. The truss is supported by a pin at A and a roller at B. Determine the support reactions.



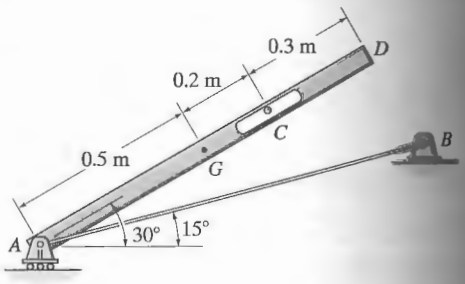
F5–3

F5–4. Determine the components of reaction at the fixed support A. Neglect the thickness of the beam.



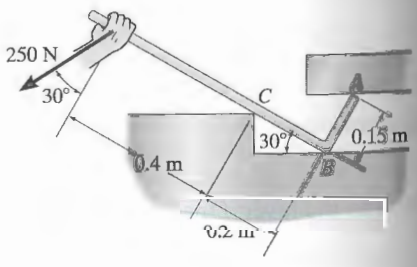
F5–4

F5–5. The 25-kg bar has a center of mass at G. If it is supported by a smooth peg at C, a roller at A, and cord AB, determine the reactions at these supports.



F5–5

F5–6. Determine the reactions at the smooth contact points A, B, and C on the bar.



F5–6

PROBLEMS

All problem solutions must include an FBD.

5–11. Determine the normal reactions at A and B in Prob. 5–1.

•5–12. Determine the tension in the cord and the horizontal and vertical components of reaction at support A of the beam in Prob. 5–4.

•5–13. Determine the horizontal and vertical components of reaction at C and the tension in the cable AB for the truss in Prob. 5–5.

5–14. Determine the horizontal and vertical components of reaction at A and the tension in cable BC on the boom in Prob. 5–6.

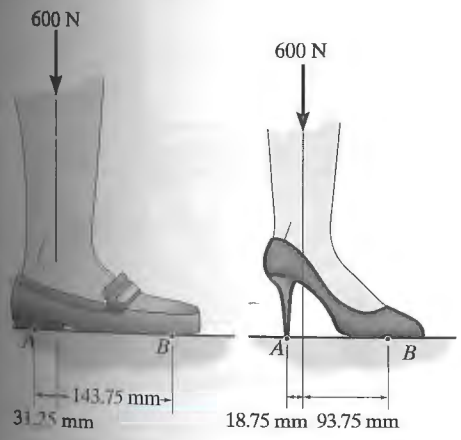
5–15. Determine the horizontal and vertical components of reaction at A and the normal reaction at B on the spanner wrench in Prob. 5–7.

5–16. Determine the normal reactions at A and B and the force in link CD acting on the member in Prob. 5–8.

•5–17. Determine the normal reactions at the points of contact at A, B, and C of the bar in Prob. 5–9.

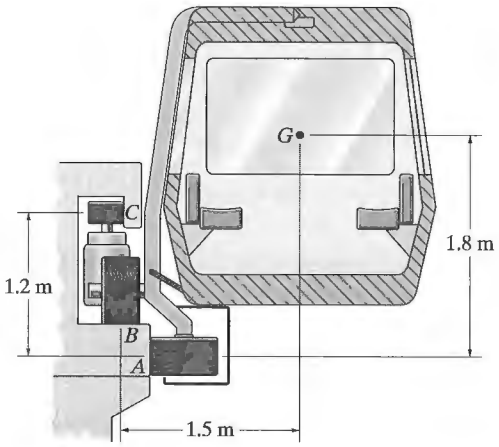
5–18. Determine the horizontal and vertical components of reaction at pin C and the force in the pawl of the winch in Prob. 5–10.

5–19. Compare the force exerted on the toe and heel of a 600-N woman when she is wearing regular shoes and stiletto heels. Assume all her weight is placed on one foot and the reactions occur at points A and B as shown.



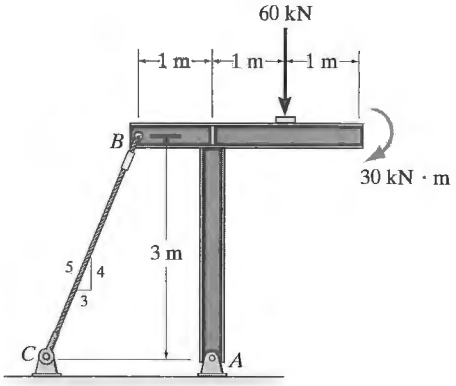
Prob. 5–19

***5–20.** The train car has a weight of 120 kN and a center of gravity at G. It is suspended from its front and rear on the track by six tires located at A, B, and C. Determine the normal reactions on these tires if the track is assumed to be a smooth surface and an equal portion of the load is supported at both the front and rear tires.



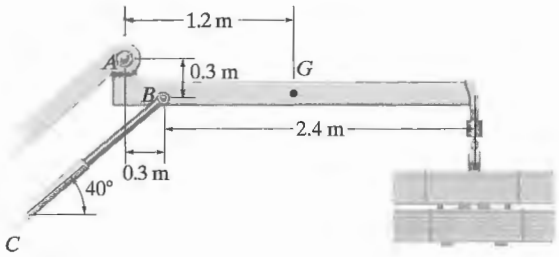
Prob. 5–20

•5–21. Determine the horizontal and vertical components of reaction at the pin A and the tension developed in cable BC used to support the steel frame.



Prob. 5–21

5–22. The articulated crane boom has a weight of 625 N and center of gravity at G. If it supports a load of 3000 N, determine the force acting at the pin A and the force in the hydraulic cylinder BC when the boom is in the position shown.



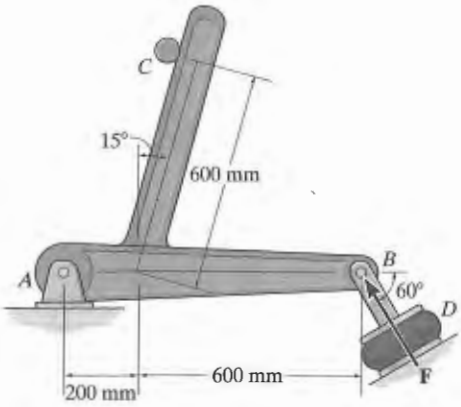
Prob. 5–22

5

5–23. The airstroke actuator at D is used to apply a force of $F = 200$ N on the member at B. Determine the horizontal and vertical components of reaction at the pin A and the force of the smooth shaft at C on the member.

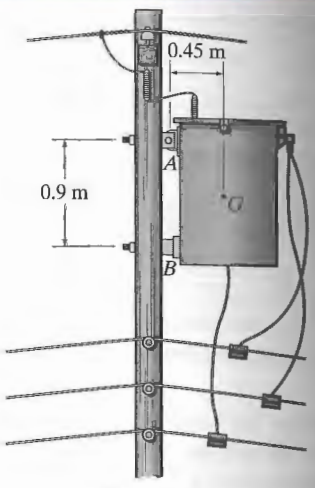
Video Solutions

***5–24.** The airstroke actuator at D is used to apply a force of $\mathbf{F}$ on the member at B. The normal reaction of the smooth shaft at C on the member is 300 N. Determine the magnitude of $\mathbf{F}$ and the horizontal and vertical components of reaction at pin A.



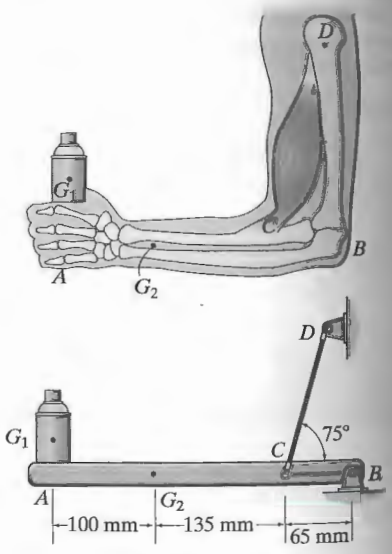
Probs. 5–23/24

•5–25. The 1500-N electrical transformer with center of gravity at G is supported by a pin at A and a smooth pad at B. Determine the horizontal and vertical components of reaction at the pin A and the reaction of the pad B on the transformer.



Prob. 5–25

5–26. A skeletal diagram of a hand holding a load is shown in the upper figure. If the load and the forearm have masses of 2 kg and 1.2 kg, respectively, and their centers of mass are located at G_1 and G_2, determine the force developed in the biceps CD and the horizontal and vertical components of reaction at the elbow joint B. The forearm supporting system can be modeled as the structural system shown in the lower figure.



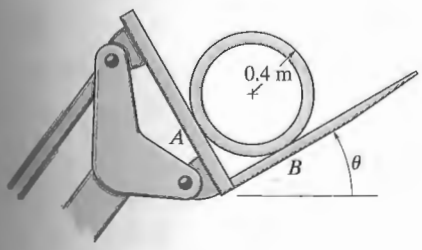
Prob. 5–26

ishing exclusively for The University of Melbourne

5–27. As an airplane's brakes are applied, the nose wheel exerts two forces on the end of the landing gear as shown. Determine the horizontal and vertical components of reaction at the pin C and the force in strut AB.

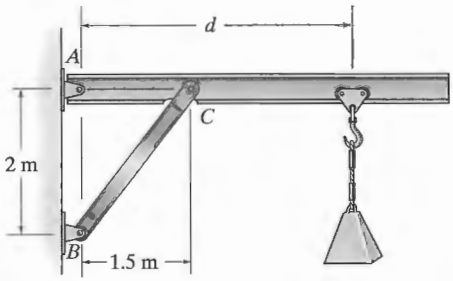
Prob. 5–27

5–28. The 1.4-Mg drainpipe is held in the tines of the fork lift. Determine the normal forces at A and B as functions of the blade angle θ and plot the results of force (vertical axis) versus θ (horizontal axis) for $0 \leq \theta \leq 90°$.



Prob. 5–28

•5–29. The mass of 700 kg is suspended from a trolley which moves along the crane rail from $d = 1.7$ m to $d = 3.5$ m. Determine the force along the pin-connected knee strut BC (short link) and the magnitude of force at pin A as a function of position d. Plot these results of F_{BC} and F_A (vertical axis) versus d (horizontal axis).



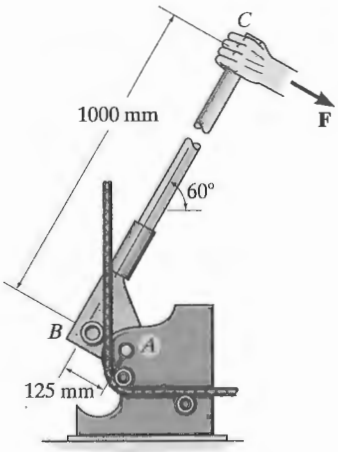
Prob. 5–29

5

5–30. If the force of $F = 500$ N is applied to the handle of the bar bender, determine the horizontal and vertical components of reaction at pin A and the reaction of the roller B on the smooth bar.

Video Solutions

5–31. If the force of the smooth roller at B on the bar bender is required to be 7.5 kN, determine the horizontal and vertical components of reaction at pin A and the required magnitude of force **F** applied to the handle.

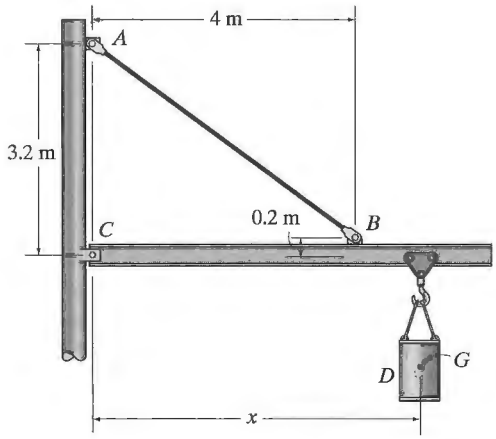


Probs. 5–30/31

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

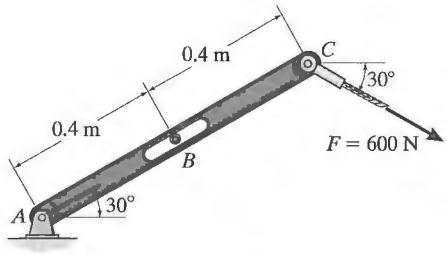
***5–32.** The jib crane is supported by a pin at C and rod AB. If the load has a mass of 2 Mg with its center of mass located at G, determine the horizontal and vertical components of reaction at the pin C and the force developed in rod AB on the crane when $x = 5$ m.

•5–33. The jib crane is supported by a pin at C and rod AB. The rod can withstand a maximum tension of 40 kN. If the load has a mass of 2 Mg, with its center of mass located at G, determine its maximum allowable distance x and the corresponding horizontal and vertical components of reaction at C.



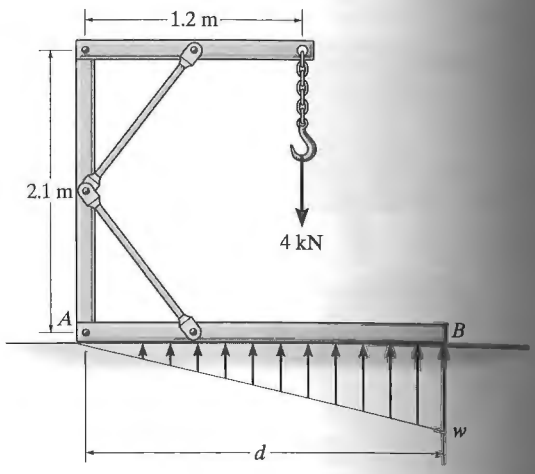
Probs. 5–32/33

5–34. Determine the horizontal and vertical components of reaction at the pin A and the normal force at the smooth peg B on the member.



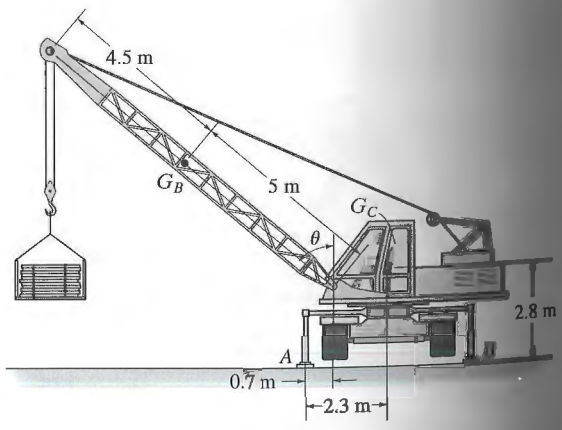
Prob. 5–34

5–35. The framework is supported by the member AB which rests on the smooth floor. When loaded, the pressure distribution on AB is linear as shown. Determine the length d of member AB and the intensity w for this case.



Prob. 5–35

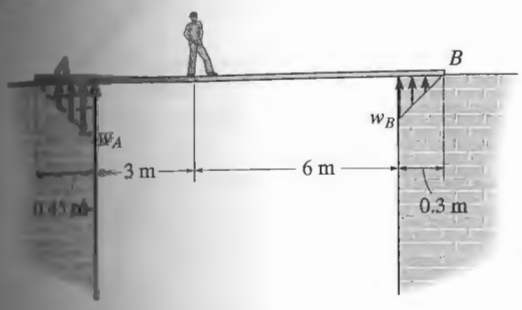
***5–36.** Outriggers A and B are used to stabilize the crane from overturning when lifting large loads. If the load to be lifted is 3 Mg, determine the *maximum* boom angle θ so that the crane does not overturn. The crane has a mass of 5 Mg and center of mass at G_C, whereas the boom has a mass of 0.6 Mg and center of mass at G_B.



Prob. 5–36

5–37. The wooden plank resting between the buildings deflects slightly when it supports the 50-kg boy. This deflection causes a triangular distribution of load at its ends, having maximum intensities of w_A and w_B. Determine w_A and w_B, each measured in N/m, when the boy is standing 3 m from one end as shown. Neglect the mass of the plank.

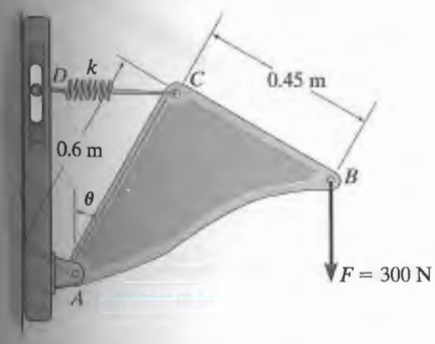
Video Solutions



Prob. 5–37

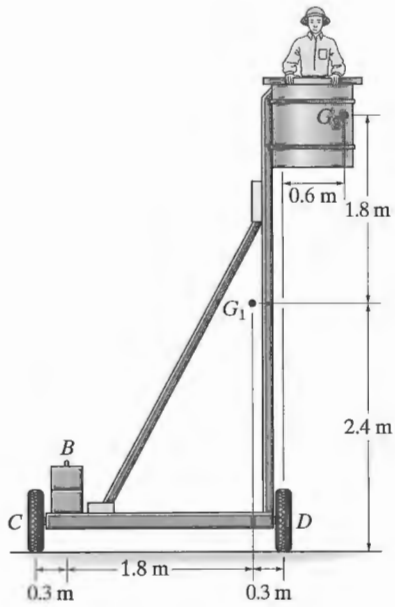
5–38. Spring CD remains in the horizontal position at all times due to the roller at D. If the spring is unstretched when $\theta = 0°$ and the bracket achieves its equilibrium position when $\theta = 30°$, determine the stiffness k of the spring and the horizontal and vertical components of reaction at pin A.

5–39. Spring CD remains in the horizontal position at all times due to the roller at D. If the spring is unstretched when $\theta = 0°$ and the stiffness is $k = 1.5$ kN/m, determine the smallest angle θ for equilibrium and the horizontal and vertical components of reaction at pin A.



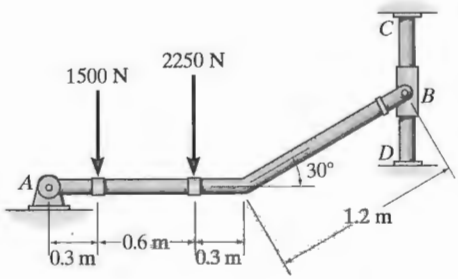
Probs. 5–38/39

***5–40.** The platform assembly has a weight of 1.25 kN and center of gravity at G_1. If it is intended to support a maximum load of 2 kN placed at point G_2, determine the smallest counterweight W that should be placed at B in order to prevent the platform from tipping over.



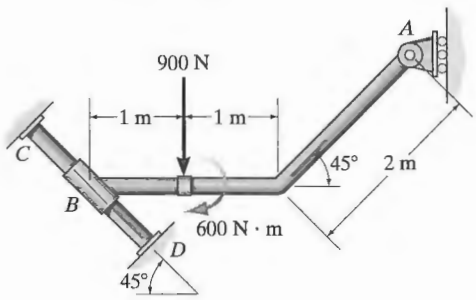
Prob. 5–40

•5–41. Determine the horizontal and vertical components of reaction at the pin A and the reaction of the smooth collar B on the rod.



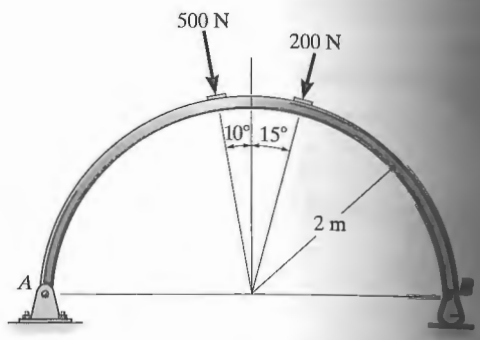
Prob. 5–41

5–42. Determine the support reactions of roller A and the smooth collar B on the rod. The collar is fixed to the rod AB, but is allowed to slide along rod CD.



Prob. 5–42

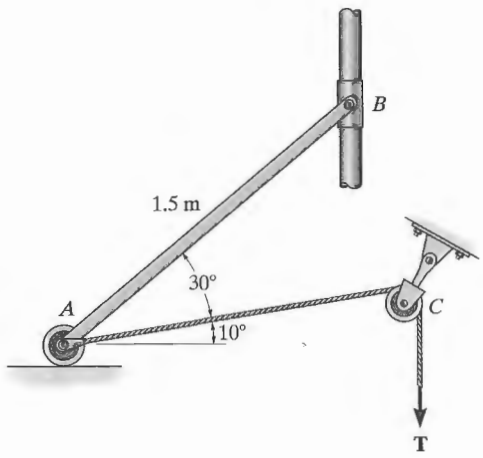
***5–44.** Determine the horizontal and vertical components of force at the pin A and the reaction at the rocker B of the curved beam.



Prob. 5–44

5

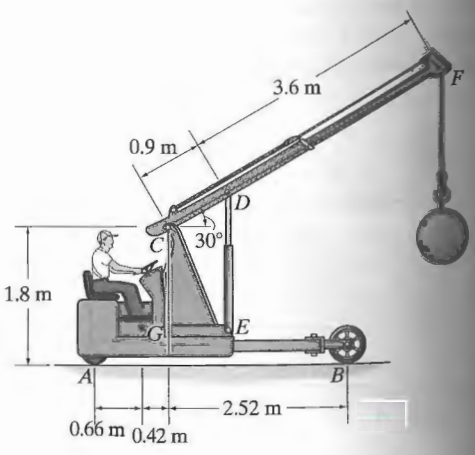
5–43. The uniform rod AB has a weight of 75 N. Determine the force in the cable when the rod is in the position shown.



Prob. 5–43

•5–45. The floor crane and the driver have a total weight of 12.5 kN with a center of gravity at G. If the crane is required to lift the 2.5-kN drum, determine the normal reaction on *both* the wheels at A and *both* the wheels at B when the boom is in the position shown.

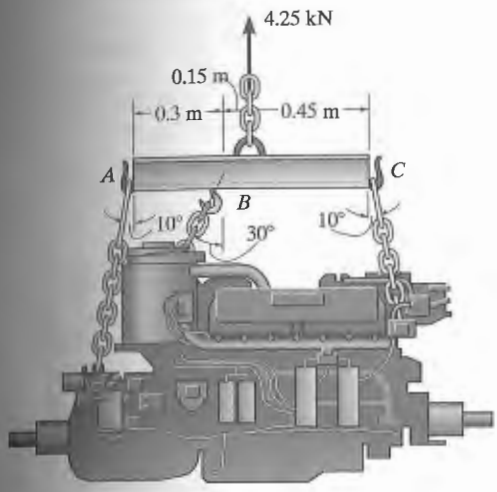
5–46. The floor crane and the driver have a total weight of 12.5 kN with a center of gravity at G. Determine the largest weight of the drum that can be lifted without causing the crane to overturn when its boom is in the position shown.



Probs. 5–45/46

Custom Publishing exclusively for The University of Melbourne

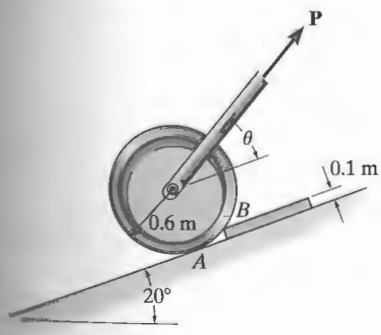
5–47. The motor has a weight of 4.25 kN. Determine the force that each of the chains exerts on the supporting hooks at A, B, and C. Neglect the size of the hooks and the thickness of the beam.



Prob. 5–47

5–48. Determine the force P needed to pull the 50-kg roller over the smooth step. Take $\theta = 60°$.

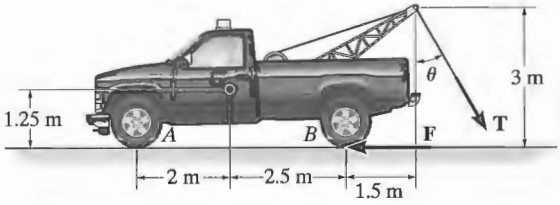
•5–49. Determine the magnitude and direction θ of the minimum force P needed to pull the 50-kg roller over the smooth step.



Probs. 5–48/49

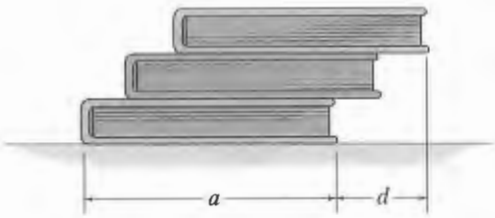
5–50. The winch cable on a tow truck is subjected to a force of $T = 6$ kN when the cable is directed at $\theta = 60°$. Determine the magnitudes of the total brake frictional force $\mathbf{F}$ for the rear set of wheels B and the total normal forces at *both* front wheels A and both rear wheels B for equilibrium. The truck has a total mass of 4 Mg and mass center at G.

5–51. Determine the minimum cable force T and critical angle θ which will cause the tow truck to start tipping, i.e., for the normal reaction at A to be zero. Assume that the truck is braked and will not slip at B. The truck has a total mass of 4 Mg and mass center at G.



Probs. 5–50/51

***5–52.** Three uniform books, each having a weight W and length a, are stacked as shown. Determine the maximum distance d that the top book can extend out from the bottom one so the stack does not topple over.



Prob. 5–52

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

CONCEPTUAL PROBLEMS

P5–5. The tie rod is used to support this overhang at the entrance of a building. If it is pin connected to the building wall at A and to the center of the overhang B, determine if the force in the rod will increase, decrease, or remain the same if (a) the support at A is moved to a lower position D, and (b) the support at B is moved to the outer position C. Explain your answer with an equilibrium analysis, using dimensions and loads. Assume the overhang is pin supported from the building wall.

P5–5

P5–6. The man attempts to pull the four wheeler up the incline and onto the truck bed. From the position shown, is it more effective to keep the rope attached at A, or would it be better to attach it to the axle of the front wheels at B? Draw a free-body diagram and do an equilibrium analysis to explain your answer.

P5–6

P5–7. Like all aircraft, this jet plane rests on three wheels. Why not use an additional wheel at the tail for better support? (Can you think of any other reason for not including this wheel?) If there was a fourth tail wheel, draw a free-body diagram of the plane from a side (2 D) view, and show why one would not be able to determine all the wheel reactions using the equations of equilibrium.

P5–7

***P5–8.** Where is the best place to arrange most of the logs in the wheelbarrow so that it minimizes the amount of force on the backbone of the person transporting the load? Do an equilibrium analysis to explain your answer.

P5–8

EQUILIBRIUM IN THREE DIMENSIONS

5.5 Free-Body Diagrams

The first step in solving three-dimensional equilibrium problems, as in the case of two dimensions, is to draw a free-body diagram. Before we can do this, however, it is first necessary to discuss the types of reactions that can occur at the supports.

Support Reactions. The reactive forces and couple moments acting at various types of supports and connections, when the members are viewed in three dimensions, are listed in Table 5–2. It is important to recognize the symbols used to represent each of these supports and to understand clearly how the forces and couple moments are developed. As in the two-dimensional case:

- A force is developed by a support that restricts the translation of its attached member.
- A couple moment is developed when rotation of the attached member is prevented.

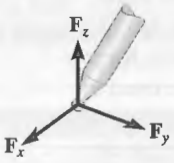
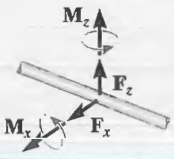
For example, in Table 5–2, item (4), the ball-and-socket joint prevents any translation of the connecting member; therefore, a force must act on the member at the point of connection. This force has three components having unknown magnitudes, Fx, Fy, Fz. Provided these components are known, one can obtain the magnitude of force, $F = \sqrt{F_x^2 + F_y^2 + F_z^2}$, and the force's orientation defined by its coordinate direction angles α, β, γ, Eqs. 2–7.* Since the connecting member is allowed to rotate freely about any axis, no couple moment is resisted by a ball-and-socket joint.

It should be noted that the *single* bearing supports in items (5) and (7), the *single* pin (8), and the *single* hinge (9) are shown to resist both force and couple-moment components. If, however, these supports are used in conjunction with *other* bearings, pins, or hinges to hold a rigid body in equilibrium and the supports are *properly aligned* when connected to the body, then the *force reactions* at these supports *alone* are adequate for supporting the body. In other words, the couple moments become redundant and are not shown on the free-body diagram. The reason for this should become clear after studying the examples which follow.

Typical examples of actual supports that are referenced to Table 5–2 are shown in the following sequence of photos.

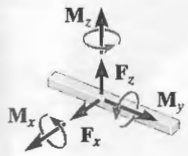
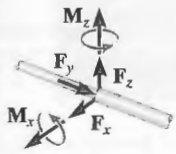
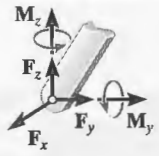
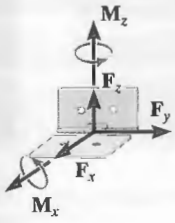
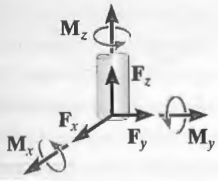
* The three unknowns may also be represented as an unknown force magnitude F and two unknown coordinate direction angles. The third direction angle is obtained using the identity $\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$, Eq. 2–8.

TABLE 5–2 Supports for Rigid Bodies Subjected to Three-Dimensional Force Systems

Types of Connection	Reaction	Number of Unknowns
(1) cable		One unknown. The reaction is a force which acts away from the member in the known direction of the cable.
(2) smooth surface support		One unknown. The reaction is a force which acts perpendicular to the surface at the point of contact.
(3) roller		One unknown. The reaction is a force which acts perpendicular to the surface at the point of contact.
(4) ball and socket		Three unknowns. The reactions are three rectangular force components.
(5) single journal bearing		Four unknowns. The reactions are two force and two couple-moment components which act perpendicular to the shaft. Note: The couple moments are generally not applied if the body is supported elsewhere. See the examples.

continued

Custom Publishing exclusively for The University of Melbourne

Types of Connection	Reaction	Number of Unknowns
(6) journal bearing with square shaft		Five unknowns. The reactions are two force and three couple-moment components. *Note*: The couple moments are generally not applied if the body is supported elsewhere. See the examples.
(7) single thrust bearing		Five unknowns. The reactions are three force and two couple-moment components. *Note*: The couple moments are generally not applied if the body is supported elsewhere. See the examples.
(8) single smooth pin		Five unknowns. The reactions are three force and two couple-moment components. *Note*: The couple moments are generally not applied if the body is supported elsewhere. See the examples.
(9) single hinge		Five unknowns. The reactions are three force and two couple-moment components. *Note*: The couple moments are generally not applied if the body is supported elsewhere. See the examples.
(10) fixed support		Six unknowns. The reactions are three force and three couple-moment components.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte. Ltd.

Free-Body Diagrams. The general procedure for establishing the free-body diagram of a rigid body has been outlined in Sec. 5.2

This ball-and-socket joint provides a connection for the housing of an earth grader to its frame. (4)

This journal bearing supports the end of the shaft. (5)

This thrust bearing is used to support the drive shaft on a machine. (7)

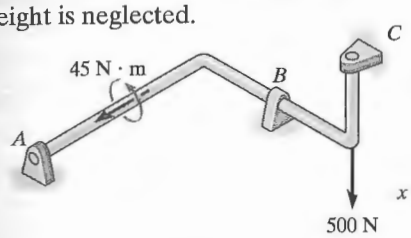
This pin is used to support the end of the strut used on a tractor. (8)

Essentially it requires first "isolating" the body by drawing its outlined shape. This is followed by a careful *labeling* of *all* the forces and couple moments with reference to an established x, y, z coordinate system. It is suggested to show the unknown components of reaction as acting on the free-body diagram in the *positive sense*. In this way, if any negative values are obtained, they will indicate that the components act in the negative coordinate directions.

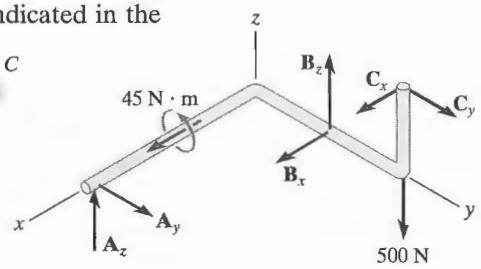
EXAMPLE 5.14

Consider the two rods and plate, along with their associated free-body diagrams shown in Fig. 5–23. The x, y, z axes are established on the diagram and the unknown reaction components are indicated in the *positive sense*. The weight is neglected.

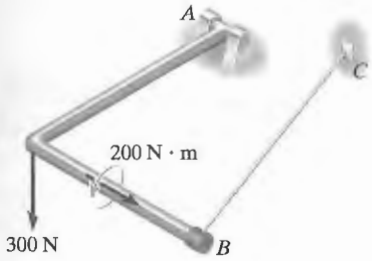
SOLUTION



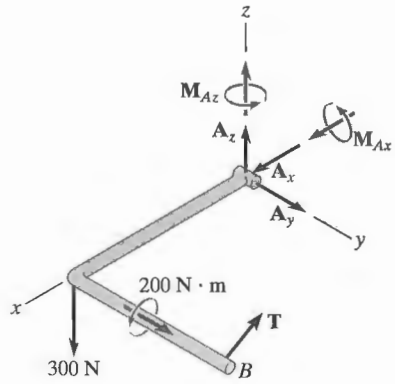
Properly aligned journal bearings at A, B, C.



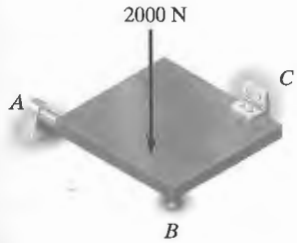
The force reactions developed by the bearings are sufficient for equilibrium since they prevent the shaft from rotating about each of the coordinate axes.



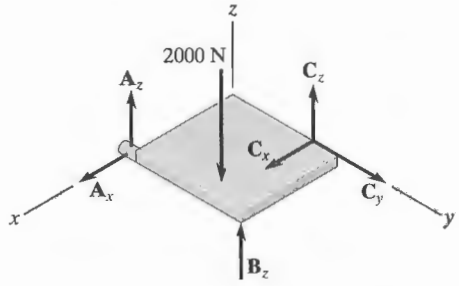
Pin at A and cable BC.



Moment components are developed by the pin on the rod to prevent rotation about the x and z axes.



Properly aligned journal bearing at A and hinge at C. Roller at B.



Only force reactions are developed by the bearing and hinge on the plate to prevent rotation about each coordinate axis. No moments at the hinge are developed.

Fig. 5–23

5.6 Equations of Equilibrium

As stated in Sec. 5.1, the conditions for equilibrium of a rigid body subjected to a three-dimensional force system require that both the *resultant* force and *resultant* couple moment acting on the body be equal to *zero*.

Vector Equations of Equilibrium. The two conditions for equilibrium of a rigid body may be expressed mathematically in vector form as

$$\boxed{\begin{aligned}\Sigma \mathbf{F} &= 0\\ \Sigma \mathbf{M}_O &= 0\end{aligned}} \tag{5–5}$$

where $\Sigma\mathbf{F}$ is the vector sum of all the external forces acting on the body and $\Sigma\mathbf{M}_O$ is the sum of the couple moments and the moments of all the forces about any point O located either on or off the body.

5

Scalar Equations of Equilibrium. If all the external forces and couple moments are expressed in Cartesian vector form and substituted into Eqs. 5–5, we have

$$\Sigma\mathbf{F} = \Sigma F_x\mathbf{i} + \Sigma F_y\mathbf{j} + \Sigma F_z\mathbf{k} = \mathbf{0}$$
$$\Sigma\mathbf{M}_O = \Sigma M_x\mathbf{i} + \Sigma M_y\mathbf{j} + \Sigma M_z\mathbf{k} = \mathbf{0}$$

Since the **i**, **j**, and **k** components are independent from one another, the above equations are satisfied provided

$$\boxed{\begin{aligned}\Sigma F_x &= 0\\ \Sigma F_y &= 0\\ \Sigma F_z &= 0\end{aligned}} \tag{5–6a}$$

and

$$\boxed{\begin{aligned}\Sigma M_x &= 0\\ \Sigma M_y &= 0\\ \Sigma M_z &= 0\end{aligned}} \tag{5–6b}$$

These *six scalar equilibrium equations* may be used to solve for at most six unknowns shown on the free-body diagram. Equations 5–6*a* require the sum of the external force components acting in the x, y, and z directions to be zero, and Eqs. 5–6*b* require the sum of the moment components about the x, y, and z axes to be zero.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

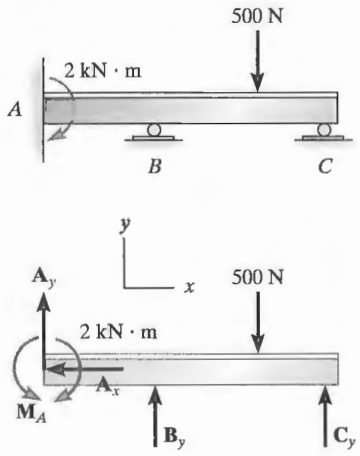
blishing exclusively for The University of Melbourne

5.7 Constraints and Statical Determinacy

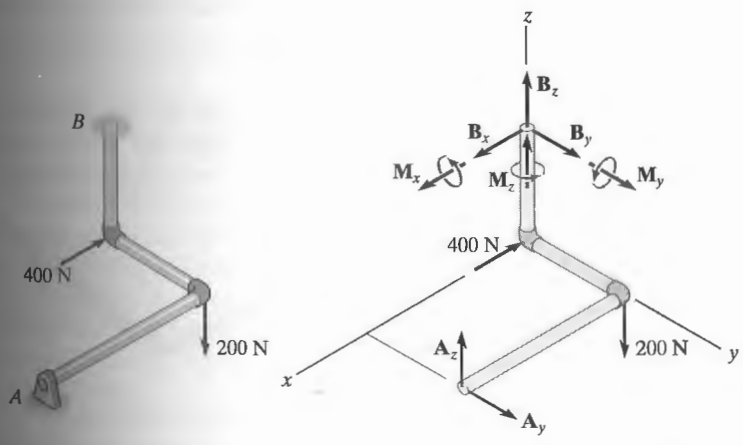
To ensure the equilibrium of a rigid body, it is not only necessary to satisfy the equations of equilibrium, but the body must also be properly held or constrained by its supports. Some bodies may have more supports than are necessary for equilibrium, whereas others may not have enough or the supports may be arranged in a particular manner that could cause the body to move. Each of these cases will now be discussed.

Redundant Constraints. When a body has redundant supports, that is, more supports than are necessary to hold it in equilibrium, it becomes statically indeterminate. *Statically indeterminate* means that there will be more unknown loadings on the body than equations of equilibrium available for their solution. For example, the beam in Fig. 5–24*a* and the pipe assembly in Fig. 5–24*b*, shown together with their free-body diagrams, are both statically indeterminate because of additional (or redundant) support reactions. For the beam there are five unknowns, M_A, A_x, A_y, B_y, and C_y, for which only three equilibrium equations can be written ($\Sigma F_x = 0$, $\Sigma F_y = 0$, and $\Sigma M_O = 0$, Eqs. 5–2). The pipe assembly has eight unknowns, for which only six equilibrium equations can be written, Eqs. 5–6.

The additional equations needed to solve statically indeterminate problems of the type shown in Fig. 5–24 are generally obtained from the deformation conditions at the points of support. These equations involve the physical properties of the body which are studied in subjects dealing with the mechanics of deformation, such as "mechanics of materials."*



(a)



(b)

Fig. 5–24

* See R. C. Hibbeler, *Mechanics of Materials*, 7th edition, Pearson Education/Prentice Hall, Inc.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

Improper Constraints. Having the same number of unknown reactive forces as available equations of equilibrium does not always guarantee that a body will be stable when subjected to a particular loading. For example, the pin support at A and the roller support at B for the beam in Fig. 5–25*a* are placed in such a way that the lines of action of the reactive forces are *concurrent* at point A. Consequently, the applied loading **P** will cause the beam to rotate slightly about A, and so the beam is improperly constrained, $\Sigma M_A \neq 0$.

In three dimensions, a body will be improperly constrained if the lines of action of all the reactive forces intersect a common axis. For example, the reactive forces at the ball-and-socket supports at A and B in Fig. 5–25*b* all intersect the axis passing through A and B. Since the moments of these forces about A and B are all zero, then the loading **P** will rotate the member about the AB axis, $\Sigma M_{AB} \neq 0$.

5

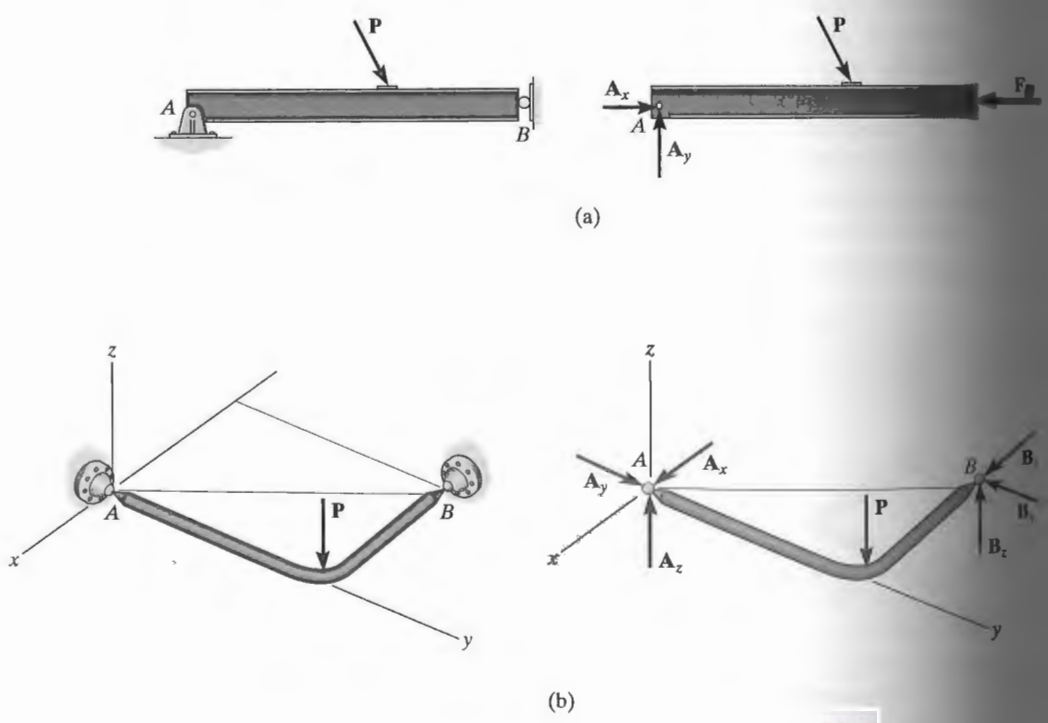
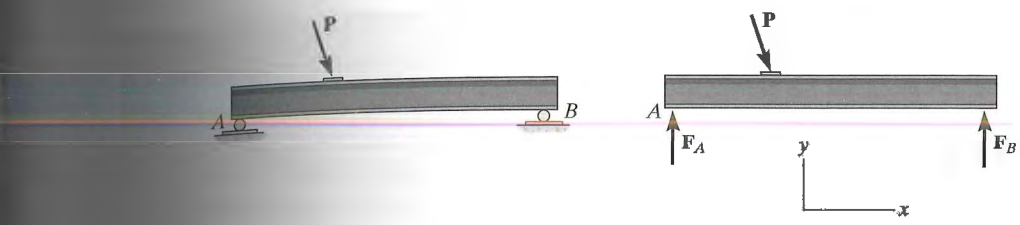
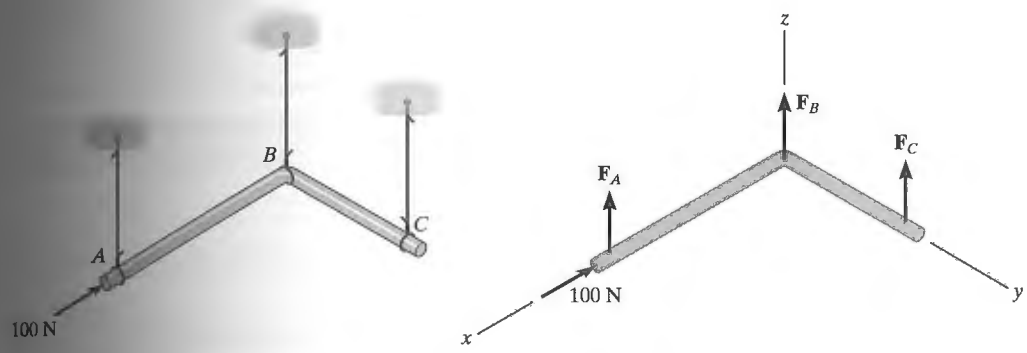


Fig. 5–25



(a)



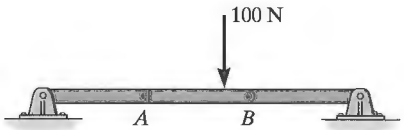
(b)

Fig. 5–26

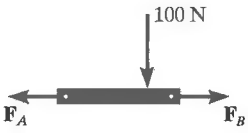
Another way in which improper constraining leads to instability occurs when the *reactive forces* are all *parallel*. Two- and three-dimensional examples of this are shown in Fig. 5–26. In both cases, the summation of forces along the x axis will not equal zero.

In some cases, a body may have *fewer* reactive forces than equations of equilibrium that must be satisfied. The body then becomes only *partially constrained*. For example, consider member AB in Fig. 5–27a with its corresponding free-body diagram in Fig. 5–27b. Here $\Sigma F_y = 0$ will not be satisfied for the loading conditions and therefore equilibrium will not be maintained.

To summarize these points, a body is considered *improperly constrained* if all the reactive forces intersect at a common point or pass through a common axis, or if all the reactive forces are parallel. In engineering practice, these situations should be avoided at all times since they will cause an unstable condition.



(a)



(b)

Fig. 5–27

Important Points

- Always draw the free-body diagram first when solving any equilibrium problem.
- If a support *prevents translation* of a body in a specific direction, then the support exerts a *force* on the body in that direction.
- If a support *prevents rotation about an axis*, then the support exerts a *couple moment* on the body about the axis.
- If a body is subjected to more unknown reactions than available equations of equilibrium, then the problem is *statically indeterminate*.
- A stable body requires that the lines of action of the reactive forces do not intersect a common axis and are not parallel to one another.

Procedure for Analysis

Three-dimensional equilibrium problems for a rigid body can be solved using the following procedure.

Free-Body Diagram.

- Draw an outlined shape of the body.
- Show all the forces and couple moments acting on the body.
- Establish the origin of the x, y, z axes at a convenient point and orient the axes so that they are parallel to as many of the external forces and moments as possible.
- Label all the loadings and specify their directions. In general, show all the *unknown* components having a *positive sense* along the x, y, z axes.
- Indicate the dimensions of the body necessary for computing the moments of forces.

Equations of Equilibrium.

- If the x, y, z force and moment components seem easy to determine, then apply the six scalar equations of equilibrium; otherwise use the vector equations.
- It is not necessary that the set of axes chosen for force summation coincide with the set of axes chosen for moment summation. Actually, an axis in any arbitrary direction may be chosen for summing forces and moments.
- Choose the direction of an axis for moment summation such that it intersects the lines of action of as many unknown forces as possible. Realize that the moments of forces passing through points on this axis and the moments of forces which are parallel to the axis will then be zero.
- If the solution of the equilibrium equations yields a negative scalar for a force or couple moment magnitude, it indicates that the sense is opposite to that assumed on the free-body diagram.

Custom Publishing exclusively for The University of Melbourne

5.15

The homogeneous plate shown in Fig. 5–28*a* has a mass of 100 kg and is subjected to a force and couple moment along its edges. If it is supported in the horizontal plane by a roller at A, a ball-and-socket joint at B, and a cord at C, determine the components of reaction at these supports.

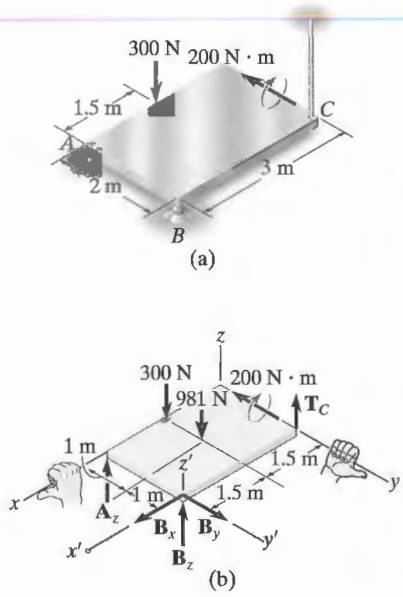


Fig. 5–28

SOLUTION (SCALAR ANALYSIS)

Free-Body Diagram. There are five unknown reactions acting on the plate, as shown in Fig. 5–28*b*. Each of these reactions is assumed to act in a positive coordinate direction.

Equations of Equilibrium. Since the three-dimensional geometry is rather simple, a *scalar analysis* provides a *direct solution* to this problem. A force summation along each axis yields

$$\Sigma F_x = 0; \qquad B_x = 0 \qquad \textit{Ans.}$$

$$\Sigma F_y = 0; \qquad B_y = 0 \qquad \textit{Ans.}$$

$$\Sigma F_z = 0; \qquad A_z + B_z + T_C - 300\text{ N} - 981\text{ N} = 0 \qquad (1)$$

Recall that the moment of a force about an axis is equal to the product of the force magnitude and the perpendicular distance (moment arm) from the line of action of the force to the axis. Also, forces that are parallel to an axis or pass through it create no moment about the axis. Hence, summing moments about the positive x and y axes, we have

$$\Sigma M_x = 0; \qquad T_C(2\text{ m}) - 981\text{ N}(1\text{ m}) + B_z(2\text{ m}) = 0 \qquad (2)$$

$$\Sigma M_y = 0;$$

$$300\text{ N}(1.5\text{ m}) + 981\text{ N}(1.5\text{ m}) - B_z(3\text{ m}) - A_z(3\text{ m}) - 200\text{ N}\cdot\text{m} = 0 \qquad (3)$$

The components of the force at B can be eliminated if moments are summed about the x' and y' axes. We obtain

$$\Sigma M_{x'} = 0; \qquad 981\text{ N}(1\text{ m}) + 300\text{ N}(2\text{ m}) - A_z(2\text{ m}) = 0 \qquad (4)$$

$$\Sigma M_{y'} = 0;$$

$$-300\text{ N}(1.5\text{ m}) - 981\text{ N}(1.5\text{ m}) - 200\text{ N}\cdot\text{m} + T_C(3\text{ m}) = 0 \qquad (5)$$

Solving Eqs. 1 through 3 or the more convenient Eqs. 1, 4, and 5 yields

$$A_z = 790\text{ N} \qquad B_z = -217\text{ N} \qquad T_C = 707\text{ N} \qquad \textit{Ans.}$$

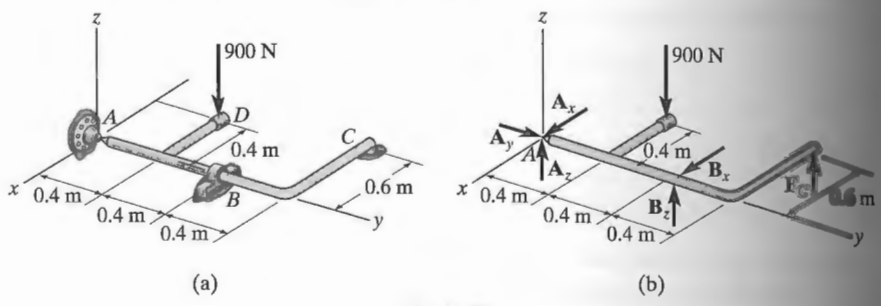
The negative sign indicates that $\mathbf{B}_z$ acts downward.

NOTE: The solution of this problem does not require a summation of moments about the z axis. The plate is partially constrained since the supports cannot prevent it from turning about the z axis if a force is applied to it in the x–y plane.

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 5.16

Determine the components of reaction that the ball and socket joint at A, the smooth journal bearing at B, and the roller support at C exert on the rod assembly in Fig. 5–29*a*.



(a) (b)

Fig. 5–29

SOLUTION

Free-Body Diagram. As shown on the free-body diagram, Fig. 5–29*b*, the reactive forces of the supports will prevent the assembly from rotating about each coordinate axis, and so the journal bearing at B only exerts reactive forces on the member.

Equations of Equilibrium. A direct solution for A_y can be obtained by summing forces along the y axis.

$$\Sigma F_y = 0; \qquad A_y = 0 \qquad \textit{Ans.}$$

The force F_C can be determined directly by summing moments about the y axis.

$$\Sigma M_y = 0; \qquad F_C(0.6\text{ m}) - 900\text{ N}(0.4\text{ m}) = 0$$
$$F_C = 600\text{ N} \qquad \textit{Ans.}$$

Using this result, B_z can be determined by summing moments about the x axis.

$$\Sigma M_x = 0; \qquad B_z(0.8\text{ m}) + 600\text{ N}(1.2\text{ m}) - 900\text{ N}(0.4\text{ m}) = 0$$
$$B_z = -450\text{ N} \qquad \textit{Ans.}$$

The negative sign indicates that $\mathbf{B}_z$ acts downward. The force B_x can be found by summing moments about the z axis.

$$\Sigma M_z = 0; \qquad -B_x(0.8\text{ m}) = 0 \quad B_x = 0 \qquad \textit{Ans.}$$

Thus,

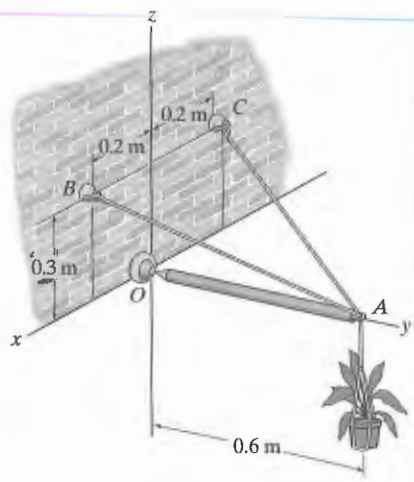
$$\Sigma F_x = 0; \qquad A_x + 0 = 0 \qquad A_x = 0 \qquad \textit{Ans.}$$

Finally, using the results of B_z and F_C.

$$\Sigma F_z = 0; \qquad A_z + (-450\text{ N}) + 600\text{ N} - 900\text{ N} = 0$$
$$A_z = 750\text{ N} \qquad \textit{Ans.}$$

PLE 5.17

The boom is used to support the 375-N (≈ 37.5 kg) flowerpot in Fig. 5–30*a*. Determine the tension developed in wires *AB* and *AC*.



(a)

SOLUTION

Free-Body Diagram. The free-body diagram of the boom is shown in Fig. 5–30*b*.

Equations of Equilibrium. We will use a vector analysis.

$$\mathbf{F}_{AB} = F_{AB}\left(\frac{\mathbf{r}_{AB}}{r_{AB}}\right) = F_{AB}\left(\frac{\{0.2\mathbf{i} - 0.6\mathbf{j} + 0.3\mathbf{k}\}\text{ m}}{\sqrt{(0.2\text{ m})^2 + (-0.6\text{ m})^2 + (0.3\text{ m})^2}}\right)$$

$$= \tfrac{2}{7}F_{AB}\mathbf{i} - \tfrac{6}{7}F_{AB}\mathbf{j} + \tfrac{3}{7}F_{AB}\mathbf{k}$$

$$\mathbf{F}_{AC} = F_{AC}\left(\frac{\mathbf{r}_{AC}}{r_{AC}}\right) = F_{AC}\left(\frac{\{-0.2\mathbf{i} - 0.6\mathbf{j} + 0.3\mathbf{k}\}\text{ m}}{\sqrt{(-0.2\text{ m})^2 + (-0.6\text{ m})^2 + (0.3\text{ m})^2}}\right)$$

$$= -\tfrac{2}{7}F_{AC}\mathbf{i} - \tfrac{6}{7}F_{AC}\mathbf{j} + \tfrac{3}{7}F_{AC}\mathbf{k}$$

We can eliminate the force reaction at *O* by writing the moment equation of equilibrium about point *O*.

$$\Sigma\mathbf{M}_O = \mathbf{0}; \qquad \mathbf{r}_A \times (\mathbf{F}_{AB} + \mathbf{F}_{AC} + \mathbf{W}) = \mathbf{0}$$

$$(0.6\mathbf{j}) \times \left[\left(\tfrac{2}{7}F_{AB}\mathbf{i} - \tfrac{6}{7}F_{AB}\mathbf{j} + \tfrac{3}{7}F_{AB}\mathbf{k}\right) + \left(-\tfrac{2}{7}F_{AC}\mathbf{i} - \tfrac{6}{7}F_{AC}\mathbf{j} + \tfrac{3}{7}F_{AC}\mathbf{k}\right) + (-375\mathbf{k})\right] = \mathbf{0}$$

$$\left(\tfrac{18}{7}F_{AB} + \tfrac{18}{7}F_{AC} - 2250\right)\mathbf{i} + \left(-\tfrac{12}{7}F_{AB} + \tfrac{12}{7}F_{AC}\right)\mathbf{k} = \mathbf{0}$$

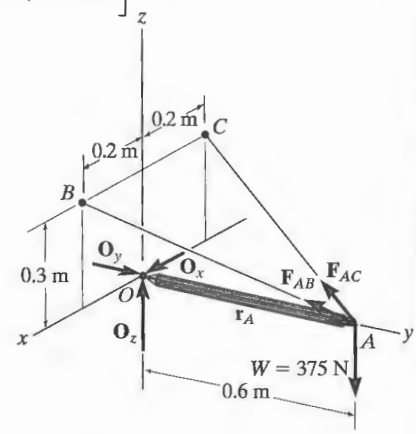
$$\Sigma M_x = 0; \qquad \tfrac{18}{7}F_{AB} + \tfrac{18}{7}F_{AC} - 2250 = 0 \qquad (1)$$

$$\Sigma M_y = 0; \qquad 0 = 0$$

$$\Sigma M_z = 0; \qquad -\tfrac{12}{7}F_{AB} + \tfrac{12}{7}F_{AC} = 0 \qquad (2)$$

Solving Eqs. (1) and (2) simultaneously,

$$F_{AB} = F_{AC} = 437.5\text{ N} \qquad \textit{Ans.}$$

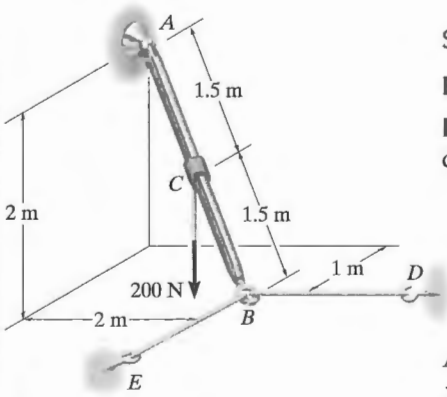


(b)

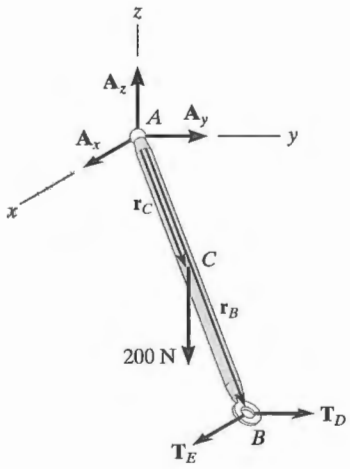
Fig. 5–30

EXAMPLE 5.18

Rod AB shown in Fig. 5–31a is subjected to the 200-N force. Determine the reactions at the ball-and-socket joint A and the tension in the cables BD and BE.



(a)



(b)

Fig. 5–31

SOLUTION (*VECTOR ANALYSIS*)

Free-Body Diagram. Fig. 5–31b.

Equations of Equilibrium. Representing each force on the free-body diagram in Cartesian vector form, we have

$$\mathbf{F}_A = A_x\mathbf{i} + A_y\mathbf{j} + A_z\mathbf{k}$$
$$\mathbf{T}_E = T_E\mathbf{i}$$
$$\mathbf{T}_D = T_D\mathbf{j}$$
$$\mathbf{F} = \{-200\mathbf{k}\}\text{ N}$$

Applying the force equation of equilibrium.

$$\Sigma\mathbf{F} = \mathbf{0}; \qquad \mathbf{F}_A + \mathbf{T}_E + \mathbf{T}_D + \mathbf{F} = \mathbf{0}$$
$$(A_x + T_E)\mathbf{i} + (A_y + T_D)\mathbf{j} + (A_z - 200)\mathbf{k} = \mathbf{0}$$

$$\Sigma F_x = 0; \qquad A_x + T_E = 0 \qquad (1)$$
$$\Sigma F_y = 0; \qquad A_y + T_D = 0 \qquad (2)$$
$$\Sigma F_z = 0; \qquad A_z - 200 = 0 \qquad (3)$$

Summing moments about point A yields

$$\Sigma\mathbf{M}_A = \mathbf{0}; \qquad \mathbf{r}_C \times \mathbf{F} + \mathbf{r}_B \times (\mathbf{T}_E + \mathbf{T}_D) = \mathbf{0}$$

Since $\mathbf{r}_C = \frac{1}{2}\mathbf{r}_B$, then

$$(0.5\mathbf{i} + 1\mathbf{j} - 1\mathbf{k}) \times (-200\mathbf{k}) + (1\mathbf{i} + 2\mathbf{j} - 2\mathbf{k}) \times (T_E\mathbf{i} + T_D\mathbf{j}) = \mathbf{0}$$

Expanding and rearranging terms gives

$$(2T_D - 200)\mathbf{i} + (-2T_E + 100)\mathbf{j} + (T_D - 2T_E)\mathbf{k} = \mathbf{0}$$

$$\Sigma M_x = 0; \qquad 2T_D - 200 = 0 \qquad (4)$$
$$\Sigma M_y = 0; \qquad -2T_E + 100 = 0 \qquad (5)$$
$$\Sigma M_z = 0; \qquad T_D - 2T_E = 0 \qquad (6)$$

Solving Eqs. 1 through 5, we get

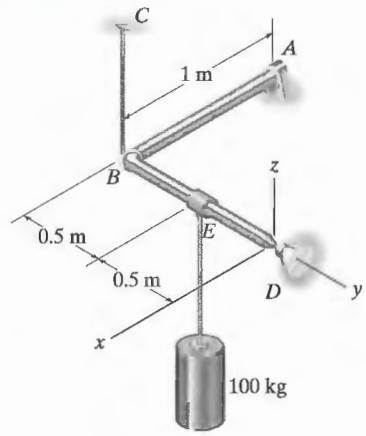
$$T_D = 100\text{ N} \qquad \textit{Ans.}$$
$$T_E = 50\text{ N} \qquad \textit{Ans.}$$
$$A_x = -50\text{ N} \qquad \textit{Ans.}$$
$$A_y = -100\text{ N} \qquad \textit{Ans.}$$
$$A_z = 200\text{ N} \qquad \textit{Ans.}$$

NOTE: The negative sign indicates that $\mathbf{A}_x$ and $\mathbf{A}_y$ have a sense which is opposite to that shown on the free-body diagram, Fig. 5–31b.

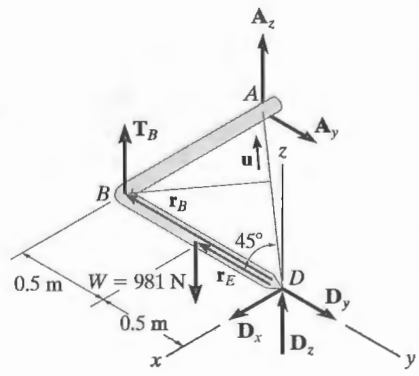
Custom Publishing exclusively for The University of Melbourne

EXAMPLE 5.19

The bent rod in Fig. 5–32*a* is supported at A by a journal bearing, at D by a ball-and-socket joint, and at B by means of cable BC. Using only *one equilibrium equation*, obtain a direct solution for the tension in cable BC. The bearing at A is capable of exerting force components only in the z and y directions since it is properly aligned on the shaft.



(a)



(b)

Fig. 5–32

SOLUTION (*VECTOR ANALYSIS*)

Free-Body Diagram. As shown in Fig. 5–32*b*, there are six unknowns.

Equations of Equilibrium. The cable tension $\mathbf{T}_B$ may be obtained *directly* by summing moments about an axis that passes through points D and A. Why? The direction of this axis is defined by the unit vector $\mathbf{u}$, where

$$\mathbf{u} = \frac{\mathbf{r}_{DA}}{r_{DA}} = -\frac{1}{\sqrt{2}}\mathbf{i} - \frac{1}{\sqrt{2}}\mathbf{j}$$
$$= -0.7071\mathbf{i} - 0.7071\mathbf{j}$$

Hence, the sum of the moments about this axis is zero provided

$$\Sigma M_{DA} = \mathbf{u}\cdot\Sigma(\mathbf{r}\times\mathbf{F}) = 0$$

Here $\mathbf{r}$ represents a position vector drawn from *any point* on the axis DA to any point on the line of action of force $\mathbf{F}$ (see Eq. 4–11). With reference to Fig. 5–32*b*, we can therefore write

$$\mathbf{u}\cdot(\mathbf{r}_B\times\mathbf{T}_B + \mathbf{r}_E\times\mathbf{W}) = \mathbf{0}$$

$$(-0.7071\mathbf{i} - 0.7071\mathbf{j})\cdot\big[(-1\mathbf{j})\times(T_B\mathbf{k}) + (-0.5\mathbf{j})\times(-981\mathbf{k})\big] = \mathbf{0}$$

$$(-0.7071\mathbf{i} - 0.7071\mathbf{j})\cdot[(-T_B + 490.5)\mathbf{i}] = \mathbf{0}$$

$$-0.7071(-T_B + 490.5) + 0 + 0 = 0$$

$$T_B = 490.5\text{ N} \qquad \textit{Ans.}$$

Since the moment arms from the axis to $\mathbf{T}_B$ and $\mathbf{W}$ are easy to obtain, we can also determine this result using a scalar analysis. As shown in Fig. 5–32*b*,

$$\Sigma M_{DA} = 0;\quad T_B(1\text{ m}\sin 45°) - 981\text{ N}(0.5\text{ m}\sin 45°) = 0$$

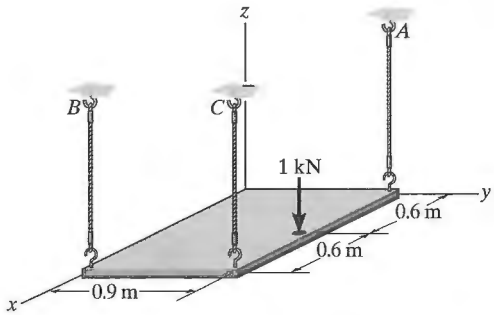
$$T_B = 490.5\text{ N} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

FUNDAMENTAL PROBLEMS

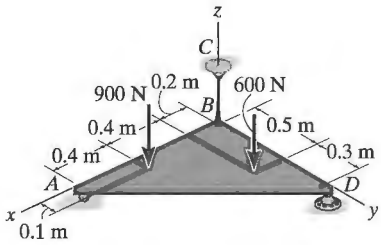
All problem solutions must include an FBD.

F5–7. The uniform plate has a weight of 2.5 kN. Determine the tension in each of the supporting cables.



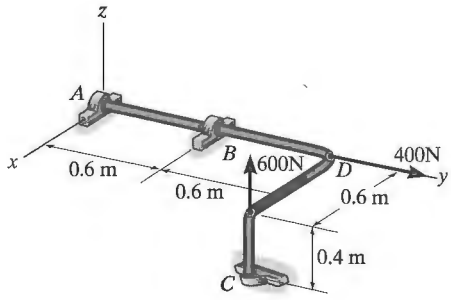
F5–7

F5–8. Determine the reactions at the roller support A, the ball-and-socket joint D, and the tension in cable BC for the plate.



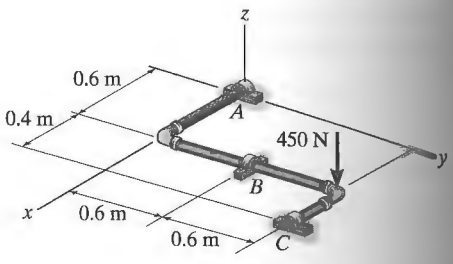
F5–8

F5–9. The rod is supported by smooth journal bearings at A, B and C and is subjected to the two forces. Determine the reactions at these supports.



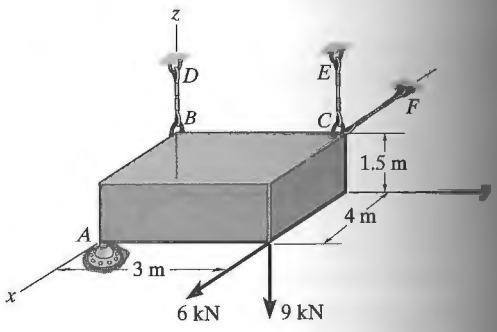
F5–9

F5–10. Determine the support reactions at the smooth journal bearings A, B, and C of the pipe assembly.



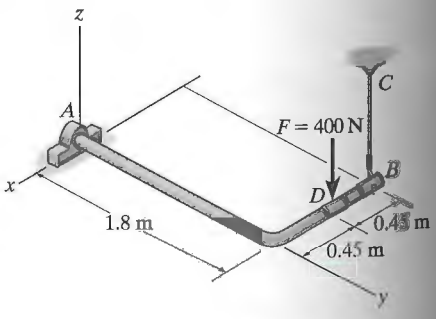
F5–10

F5–11. Determine the force developed in cords BD, CE, and CF and the reactions of the ball-and-sock joint A on the block.



F5–11

F5–12. Determine the components of reaction that the thrust bearing A and cable BC exert on the bar.



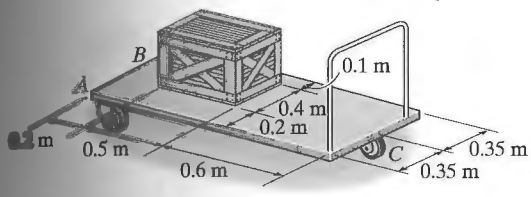
F5–12

MasteringENGINEERING™
Customized coaching is available!

PROBLEMS

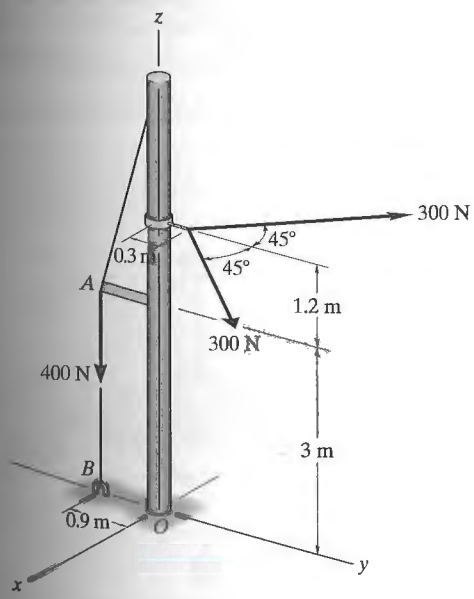
All problem solutions must include an FBD.

5–63. The cart supports the uniform crate having a mass of 85 kg. Determine the vertical reactions on the three casters at A, B, and C. The caster at B is not shown. Neglect the mass of the cart.



Prob. 5–63

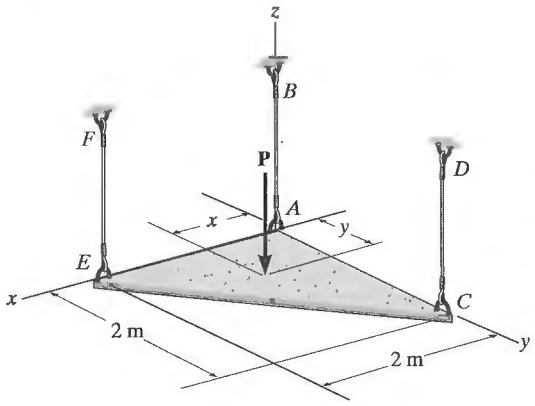
•5–64. The pole for a power line is subjected to the two cable forces of 300 N, each force lying in a plane parallel to the $x-y$ plane. If the tension in the guy wire AB is 400 N, determine the x, y, z components of reaction at the fixed base of the pole, O.



Prob. 5–64

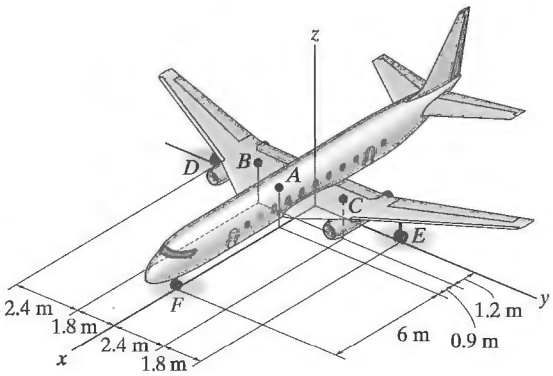
•5–65. If $P = 6$ kN, $x = 0.75$ m and $y = 1$ m, determine the tension developed in cables AB, CD, and EF. Neglect the weight of the plate.

5–66. Determine the location x and y of the point of application of force **P** so that the tension developed in cables AB, CD, and EF is the same. Neglect the weight of the plate.



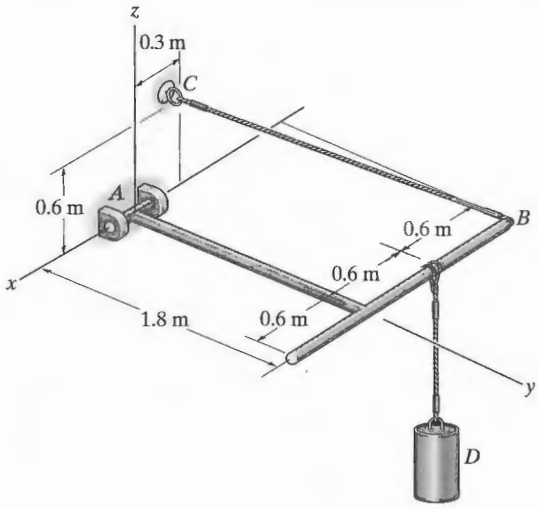
Probs. 5–65/66

5–67. Due to an unequal distribution of fuel in the wing tanks, the centers of gravity for the airplane fuselage A and wings B and C are located as shown. If these components have weights $W_A = 225$ kN, $W_B = 40$ kN, and $W_C = 30$ kN, determine the normal reactions of the wheels D, E, and F on the ground.



Prob. 5–67

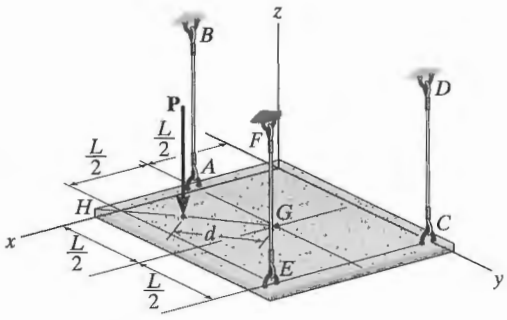
***5–76.** The member is supported by a pin at A and a cable BC. If the load at D is 1.5 kN, determine the x, y, z components of reaction at the pin A and the tension in cable BC.



Prob. 5–76

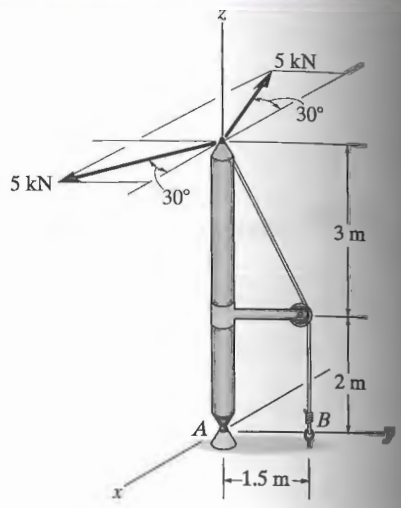
•5–77. The plate has a weight of W with a center of gravity at G. Determine the distance d along line GH where the vertical force $P = 0.75W$ will cause the tension in wire CD to become zero.

5–78. The plate has a weight of W with a center of gravity at G. Determine the tension developed in wires AB, CD, and EF if the force $P = 0.75W$ is applied at $d = L/2$.



Probs. 5–77/78

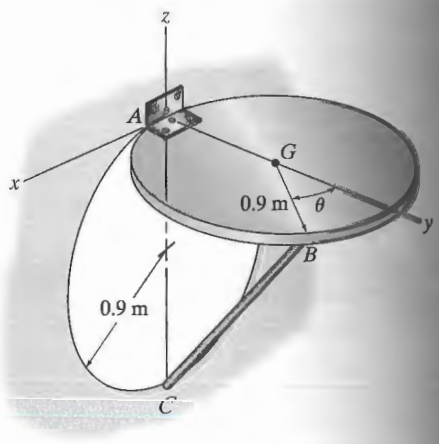
5–79. The boom is supported by a ball-and-socket joint at A and a guy wire at B. If the 5-kN loads lie in a plane which is parallel to the x–y plane, determine the x, y, z components of reaction at A and the tension in the cable at B.



Prob. 5–79

***5–80.** The circular door has a weight of 275 N and a center of gravity at G. Determine the x, y, z components of reaction at the hinge A and the force acting along strut CB needed to hold the door in equilibrium. Set $\theta = 45°$.

•5–81. The circular door has a weight of 275 N and a center of gravity at G. Determine the x, y, z components of reaction at the hinge A and the force acting along strut CB needed to hold the door in equilibrium. Set $\theta = 90°$.



Probs. 5–80/81

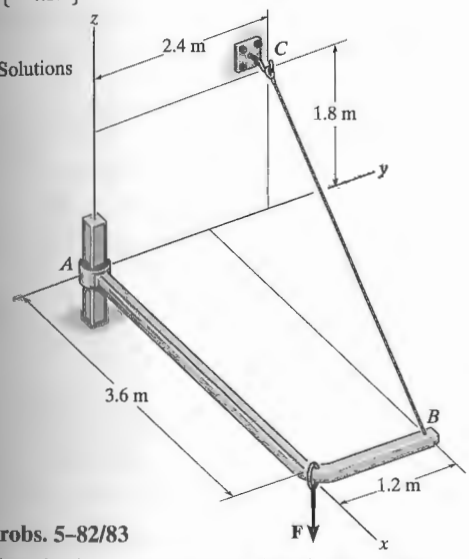
Publishing exclusively for The University of Melbourne

5–82. Member AB is supported at B by a cable and at A by a smooth fixed *square* rod which fits loosely through the square hole of the collar. If $\mathbf{F} = \{2\mathbf{i} - 4\mathbf{j} - 7.5\mathbf{k}\}$ kN, determine the x, y, z components of reaction at A and the tension in the cable.

Video Solutions

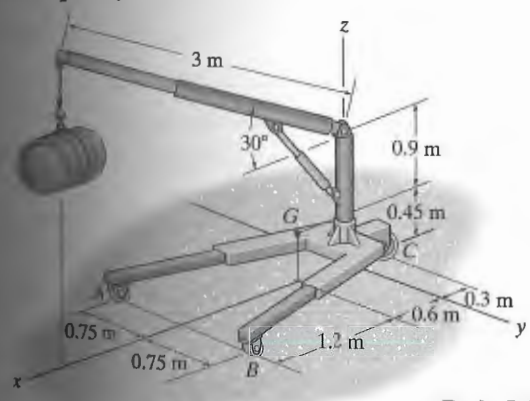
5–83. Member AB is supported at B by a cable and at A by a smooth fixed *square* rod which fits loosely through the square hole of the collar. Determine the tension in cable BC if the force $\mathbf{F} = \{-4.5\mathbf{k}\}$ kN.

Video Solutions



Probs. 5–82/83

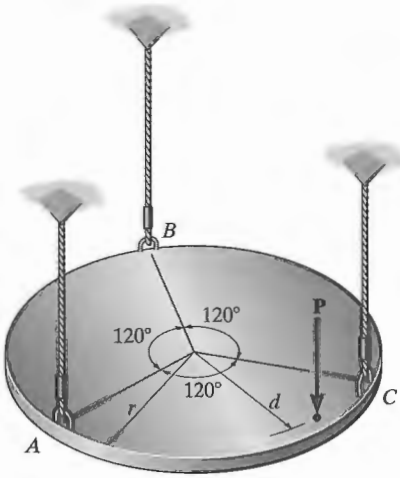
5–84. Determine the largest weight of the oil drum that the floor crane can support without overturning. Also, what are the vertical reactions at the smooth wheels A, B, and C for this case. The floor crane has a weight of 1.5 kN, with its center of gravity located at G.



Prob. 5–84

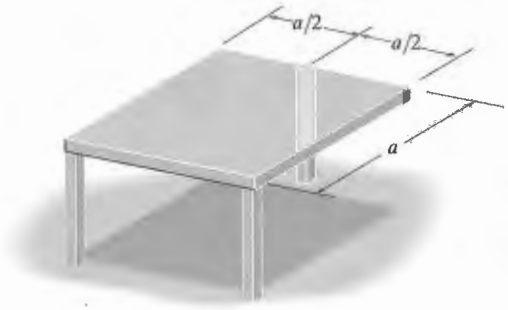
•5–85. The circular plate has a weight W and center of gravity at its center. If it is supported by three vertical cords tied to its edge, determine the largest distance d from the center to where any vertical force $\mathbf{P}$ can be applied so as not to cause the force in any one of the cables to become zero.

5–86. Solve Prob. 5–85 if the plate's weight W is neglected.



Probs. 5–85/86

5–87. A uniform square table having a weight W and sides a is supported by three vertical legs. Determine the smallest vertical force $\mathbf{P}$ that can be applied to its top that will cause it to tip over.



Prob. 5–87

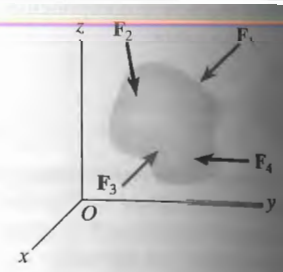
Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

CHAPTER REVIEW

Equilibrium

A body in equilibrium does not rotate but can translate with constant velocity, or it does not move at all.

$$\Sigma \mathbf{F} = \mathbf{0}$$
$$\Sigma \mathbf{M} = \mathbf{0}$$



Two Dimensions

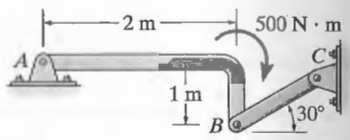
Before analyzing the equilibrium of a body, it is first necessary to draw its free-body diagram. This is an outlined shape of the body, which shows all the forces and couple moments that act on it.

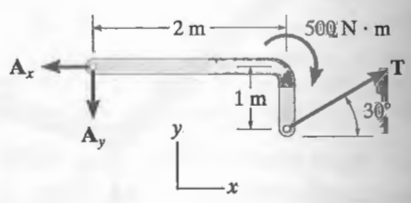
Couple moments can be placed anywhere on a free-body diagram since they are free vectors. Forces can act at any point along their line of action since they are sliding vectors.

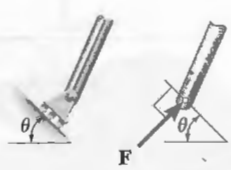
Angles used to resolve forces, and dimensions used to take moments of the forces, should also be shown on the free-body diagram.

Some common types of supports and their reactions are shown below in two dimensions.

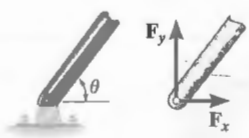
Remember that a support will exert a force on the body in a particular direction if it prevents translation of the body in that direction, and it will exert a couple moment on the body if it prevents rotation.



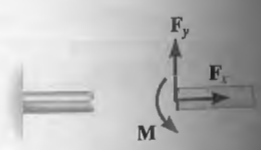




roller



smooth pin or hinge



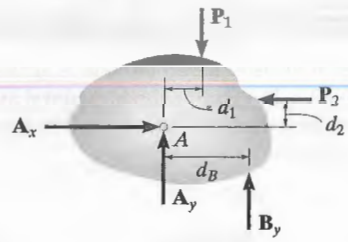
fixed support

The three scalar equations of equilibrium can be applied when solving problems in two dimensions, since the geometry is easy to visualize.

$$\Sigma F_x = 0$$
$$\Sigma F_y = 0$$
$$\Sigma M_O = 0$$

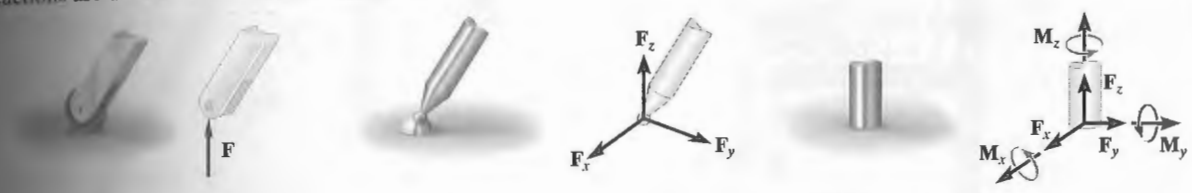
For the m[redacted]t solution, try to sum forces along an axis that will eliminate as many unknown forces as possible. Sum moments about a point A that passes through the line of action of as many unknown forces as possible.

$$\Sigma F_x = 0;$$
$$A_x - P_2 = 0 \qquad A_x = P_2$$
$$\Sigma M_A = 0;$$
$$P_2 d_2 + B_y d_B - P_1 d_1 = 0$$
$$B_y = \frac{P_1 d_1 - P_2 d_2}{d_B}$$



Three [redacted]mensions

Some common types of supports and their reactions are shown here in three dimensions.



roller ball and socket fixed support

In three [redacted]mensions, it is often advantageous to use a [redacted]artesian vector analysis when applying the equations of equilibrium. To do this, first express each known and unknown force and couple moment shown on the free-body diagram as a Cartesian vector. Then set the force [redacted]mmation equal to zero. Take moments about a point O that lies on the line of action of as many unknown force components as possible. From point O direct position vectors to each force, and then use the cross product to determine the moment of each force.

The six scalar equations of equilibrium are established by setting the respective **i**, **j**, and **k** [redacted]mponents of these force and moment summations equal to zero.

$$\Sigma \mathbf{F} = \mathbf{0}$$
$$\Sigma \mathbf{M}_O = \mathbf{0}$$

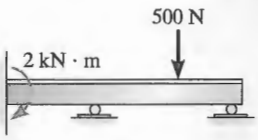
$$\Sigma F_x = 0$$
$$\Sigma F_y = 0$$
$$\Sigma F_z = 0$$

$$\Sigma M_x = 0$$
$$\Sigma M_y = 0$$
$$\Sigma M_z = 0$$

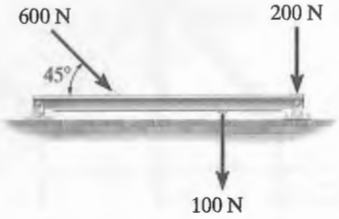
Determinacy and Stability

If a body is [redacted]upported by a minimum number of constraints to ensure equilibrium, then it is statically [redacted]terminate. If it has more constraints than required, then it is statically indeterminate.

To properly [redacted]onstrain the body, the reactions must not all be parallel to one another or concurrent.



Statically indeterminate, five reactions, three equilibrium equations



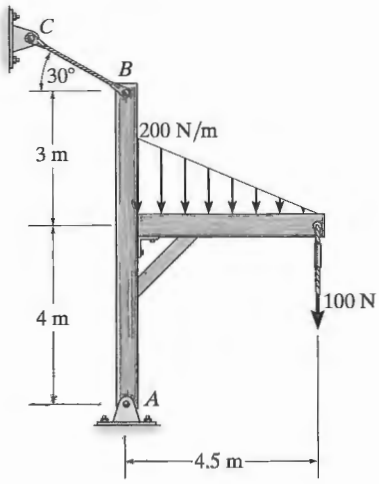
Proper constraint, statically determinate

ENGR10003
A publication from Pearson Custom Publishing

MasteringENGINEERING
Customized coaching is available!

REVIEW PROBLEMS

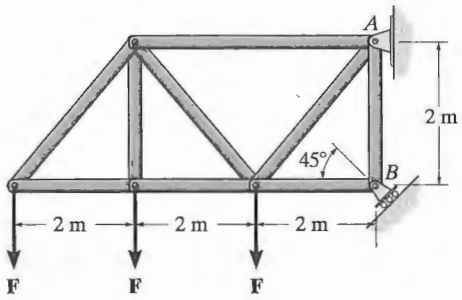
***5–88.** Determine the horizontal and vertical components of reaction at pin A and force in the cable BC. Neglect the thickness of the members.



Prob. 5–88

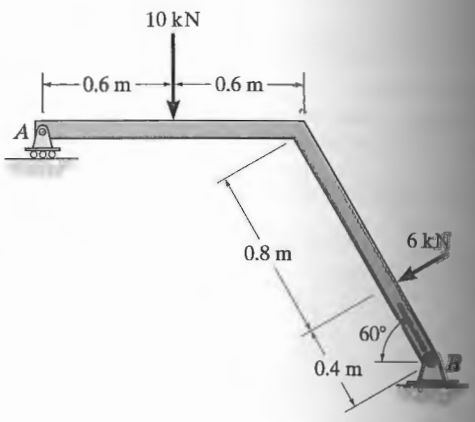
•5–89. Determine the horizontal and vertical components of reaction at pin A and the reaction at roller B required to support the truss. Set $F = 600$ N.

5–90. If the roller at B can sustain a maximum load of 3 kN, determine the largest magnitude of each of the three forces **F** that can be supported by the truss.



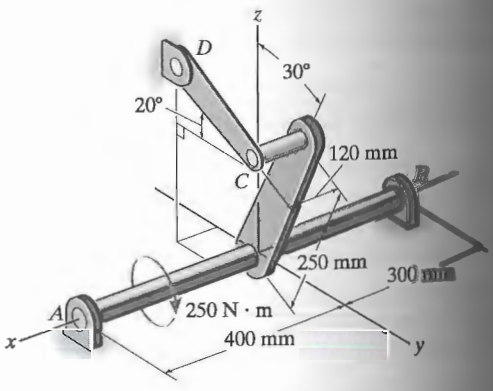
Probs. 5–89/90

5–91. Determine the normal reaction at the roller A and horizontal and vertical components at pin B for equilibrium of the member.



Prob. 5–91

***5–92.** The shaft assembly is supported by two smooth journal bearings A and B and a short link DC. If a couple moment is applied to the shaft as shown, determine the components of force reaction at the journal bearings and the force in the link. The link lies in a plane parallel to the y–z plane and the bearings are properly aligned on the shaft.

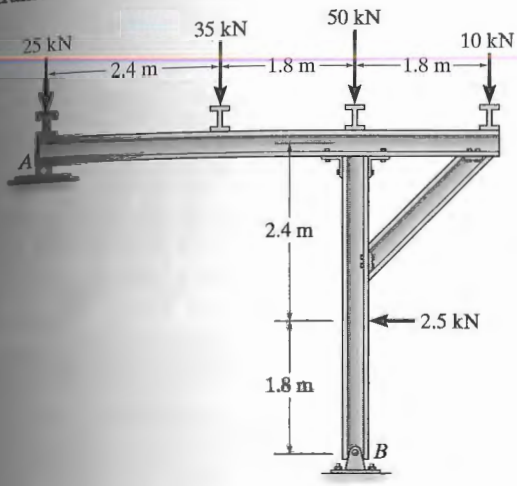


Prob. 5–92

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

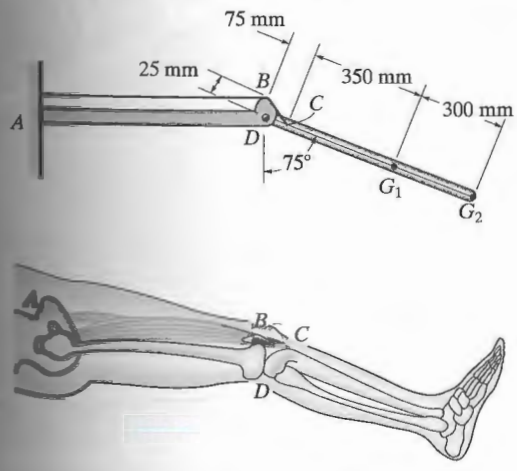
Publishing exclusively for The University of Melbourne

•5–93. Determine the reactions at the supports A and B of the frame.



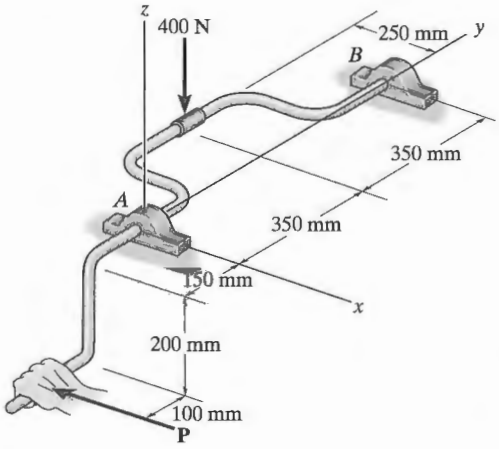
Prob. 5–93

5–94. A skeletal diagram of the lower leg is shown in the lower figure. Here it can be noted that this portion of the leg is lifted by the quadriceps muscle attached to the hip at A and to the patella bone at B. This bone slides freely over cartilage at the knee joint. The quadriceps is further extended and attached to the tibia at C. Using the mechanical system shown in the upper figure to model the lower leg, determine the tension in the quadriceps at C and the magnitude of the resultant force at the femur (pin), D, in order to hold the lower leg in the position shown. The lower leg has a mass of 3.2 kg and a mass center at G_1; the foot has a mass of 1.6 kg and a mass center at G_2.



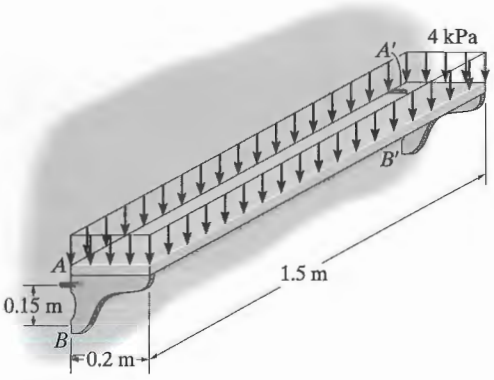
Prob. 5–94

5–95. A vertical force of 400 N acts on the crankshaft. Determine the horizontal equilibrium force P that must be applied to the handle and the x, y, z components of force at the smooth journal bearing A and the thrust bearing B. The bearings are properly aligned and exert only force reactions on the shaft.



Prob. 5–95

*5–96. The symmetrical shelf is subjected to a uniform load of 4 kPa. Support is provided by a bolt (or pin) located at each end A and A' and by the symmetrical brace arms, which bear against the smooth wall on both sides at B and B'. Determine the force resisted by each bolt at the wall and the normal force at B for equilibrium.



Prob. 5–96

Hibbeler, RC. (2012). Equilibrium of a Rigid Body. In *Engineering Mechanics Statics*, (12th Ed. In SI Units, pp. 199-262). Singapore: Pearson Education South Asia Pte Ltd.

The forces within the members of each truss bridge must be determined if the members are to be properly designed.

Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

Structural Analysis

6

HAPTER OBJECTIVES

- To show how to determine the forces in the members of a truss using the method of joints and the method of sections.
- To analyze the forces acting on the members of frames and machines composed of pin-connected members.

6.1 Simple Trusses

A *truss* is a structure composed of slender members joined together at their end points. The members commonly used in construction consist of wooden struts or metal bars. In particular, *planar* trusses lie in a single plane and are often used to support roofs and bridges. The truss shown in Fig. 6–1*a* is an example of a typical roof-supporting truss. In this figure, the roof load is transmitted to the truss *at the joints* by means of a series of *purlins*. Since this loading acts in the same plane as the truss, Fig. 6–1*b*, the analysis of the forces developed in the truss members will be two-dimensional.

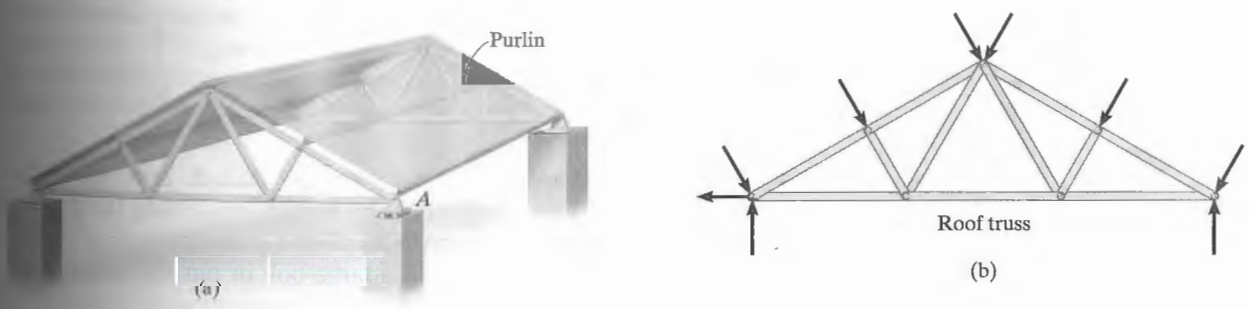


Fig. 6–1

Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

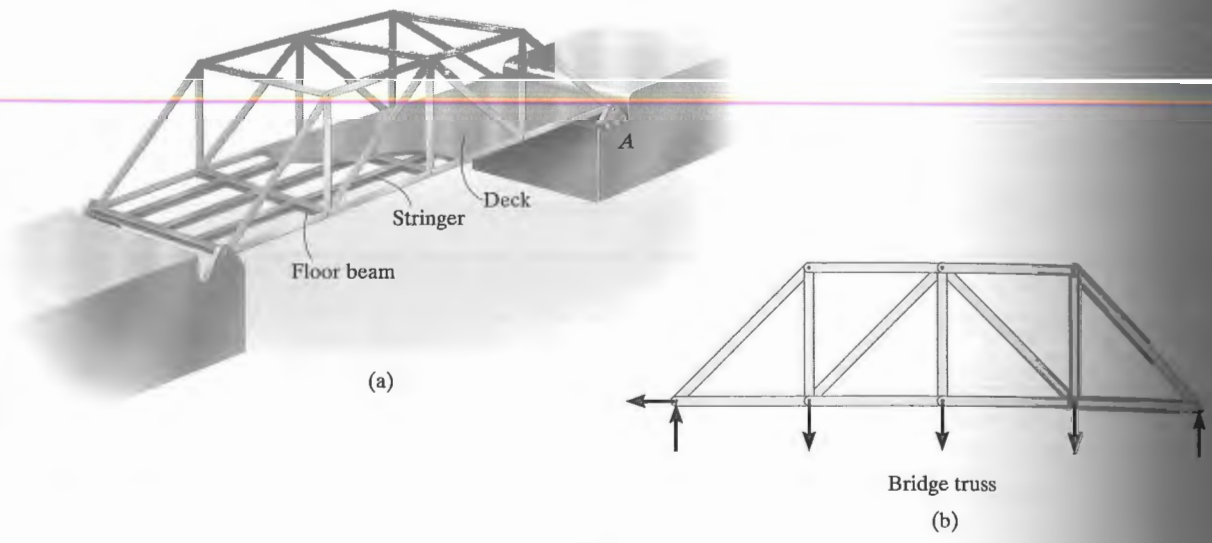


Fig. 6–2

In the case of a bridge, such as shown in Fig. 6–2*a*, the load on the deck is first transmitted to *stringers*, then to *floor beams*, and finally to the *joints* of the two supporting side trusses. Like the roof truss, the bridge truss loading is also coplanar, Fig. 6–2*b*.

When bridge or roof trusses extend over large distances, a rocker or roller is commonly used for supporting one end, for example, joint *A* in Figs. 6–1*a* and 6–2*a*. This type of support allows freedom for expansion or contraction of the members due to a change in temperature or application of loads.

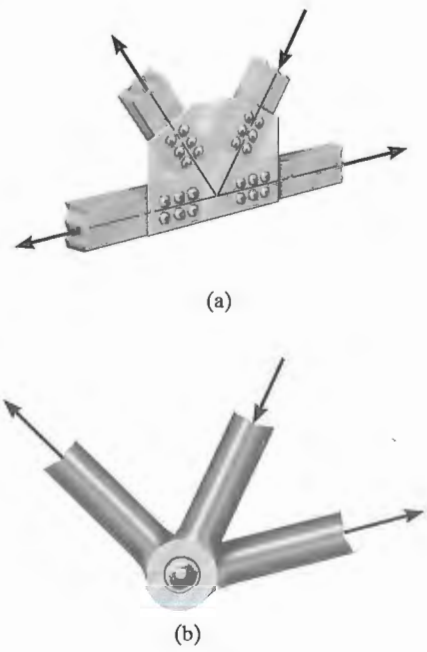


Fig. 6–3

Assumptions for Design. To design both the members and the connections of a truss, it is necessary first to determine the *force* developed in each member when the truss is subjected to a given loading. To do this we will make two important assumptions.

- ***All loadings are applied at the joints.*** In most situations such as for bridge and roof trusses, this assumption is true. Frequently the weight of the members is neglected because the force supported by each member is usually much larger than its weight. However, if the weight is to be included in the analysis, it is generally satisfactory to apply it as a vertical force, with half of its magnitude applied at each end of the member.
- ***The members are joined together by smooth pins.*** The joint connections are usually formed by bolting or welding the ends of the members to a common plate, called a *gusset plate*, as shown in Fig. 6–3*a*, or by simply passing a large bolt or pin through each of the members, Fig. 6–3*b*. We can assume these connections act as pins provided the center lines of the joining members are *concurrent*, as in Fig. 6–3.

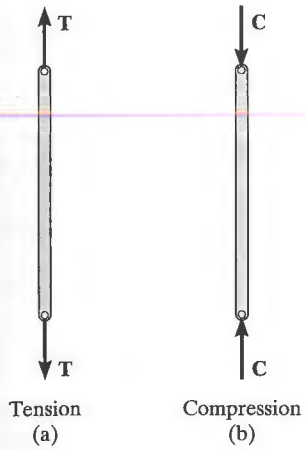


Fig. 6–4

Because of these two assumptions, *each truss member will act as a two-force member*, and therefore the force acting at each end of the member will be directed along the axis of the member. If the force tends to *elongate* the member, it is a *tensile force* (T), Fig. 6–4*a*; whereas if it tends to *shorten* the member, it is a *compressive force* (C), Fig. 6–4*b*. In the actual design of a truss it is important to state whether the nature of the force is tensile or compressive. Often, compression members must be made *thicker* than tension members because of the buckling or column effect that occurs when a member is in compression.

Simple Truss. If three members are pin connected at their ends they form a *triangular truss* that will be *rigid*, Fig. 6–5. Attaching two more members and connecting these members to a new joint *D* forms a larger truss, Fig. 6–6. This procedure can be repeated as many times as desired to form an even larger truss. If a truss can be constructed by expanding the basic triangular truss in this way, it is called a *simple truss*.

The use of metal gusset plates in the construction of these Warren trusses is clearly evident.

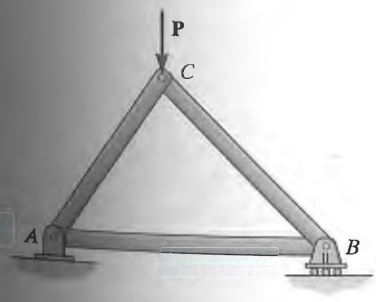


Fig. 6–5

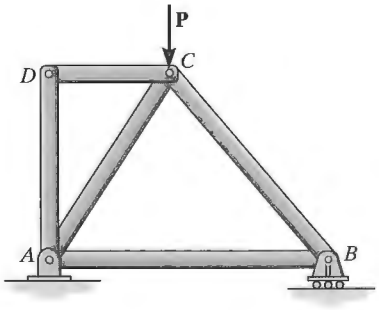


Fig. 6–6

6.2 The Method of Joints

Please refer to the Companion CD for the animation: *Free-body Diagram For an A-frame*

In order to analyze or design a truss, it is necessary to determine the force in each of its members. One way to do this is to use the method of joints. This method is based on the fact that if the entire truss is in equilibrium, then each of its joints is also in equilibrium. Therefore, if the free-body diagram of each joint is drawn, the force equilibrium equations can then be used to obtain the member forces acting on each joint. Since the members of a *plane truss* are straight two-force members lying in a single plane, each joint is subjected to a force system that is *coplanar and concurrent.* As a result, only $\Sigma F_x = 0$ and $\Sigma F_y = 0$ need to be satisfied for equilibrium.

For example, consider the pin at joint B of the truss in Fig. 6–7*a*. Three forces act on the pin, namely, the 500-N force and the forces exerted by members BA and BC. The free-body diagram of the pin is shown in Fig. 6–7*b*. Here, $\mathbf{F}_{BA}$ is "pulling" on the pin, which means that member BA is in *tension;* whereas $\mathbf{F}_{BC}$ is "pushing" on the pin, and consequently member BC is in *compression.* These effects are clearly demonstrated by isolating the joint with small segments of the member connected to the pin, Fig. 6–7*c*. The pushing or pulling on these small segments indicates the effect of the member being either in compression or tension.

When using the method of joints, always start at a joint having at least one known force and at most two unknown forces, as in Fig. 6–7*b*. In this way, application of $\Sigma F_x = 0$ and $\Sigma F_y = 0$ yields two algebraic equations which can be solved for the two unknowns. When applying these equations, the correct sense of an unknown member force can be determined using one of two possible methods.

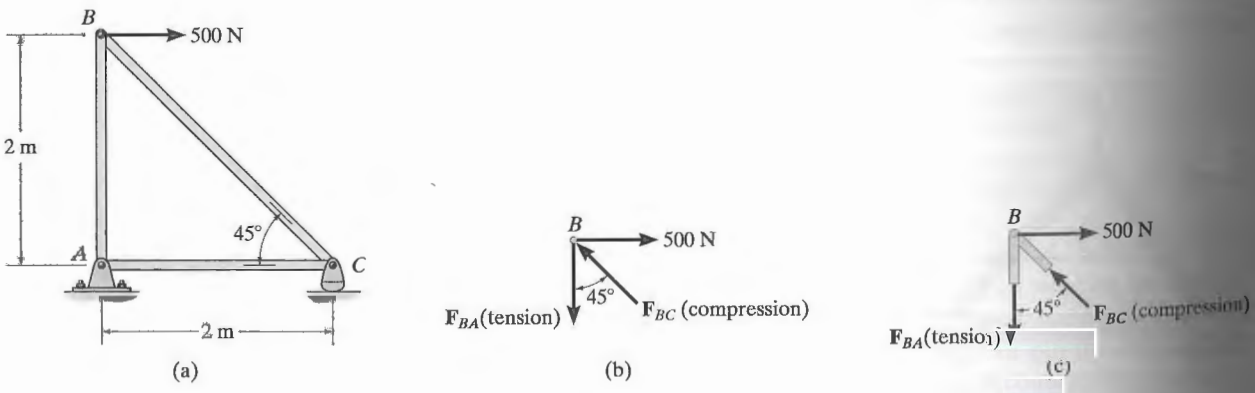


Fig. 6–7

- The *correct* sense of direction of an unknown member force can, in many cases, be determined "by inspection." For example, $\mathbf{F}_{BC}$ in Fig. 6–7*b* must push on the pin (compression) since its horizontal component, $F_{BC} \sin 45°$, must balance the 500-N force ($\Sigma F_x = 0$). Likewise, $\mathbf{F}_{BA}$ is a tensile force since it balances the vertical component, $F_{BC} \cos 45°$ ($\Sigma F_y = 0$). In more complicated cases, the sense of an unknown member force can be *assumed;* then, after applying the equilibrium equations, the assumed sense can be verified from the numerical results. A *positive* answer indicates that the sense is *correct*, whereas a *negative* answer indicates that the sense shown on the free-body diagram must be *reversed.*
- *Always assume* the *unknown member forces* acting on the joint's free-body diagram to be in *tension;* i.e., the forces "pull" on the pin. If this is done, then numerical solution of the equilibrium equations will yield *positive scalars for members in tension and negative scalars for members in compression.* Once an unknown member force is found, use its *correct* magnitude and sense (T or C) on subsequent joint free-body diagrams.

The forces in the members of this simple roof truss can be determined using the method of joints.

6

Procedure for Analysis

The following procedure provides a means for analyzing a truss using the method of joints.

- Draw the free-body diagram of a joint having at least one known force and at most two unknown forces. (If this joint is at one of the supports, then it may be necessary first to calculate the external reactions at the support.)
- Use one of the two methods described above for establishing the sense of an unknown force.
- Orient the x and y axes such that the forces on the free-body diagram can be easily resolved into their x and y components and then apply the two force equilibrium equations $\Sigma F_x = 0$ and $\Sigma F_y = 0$. Solve for the two unknown member forces and verify their correct sense.
- Using the calculated results, continue to analyze each of the other joints. Remember that a member in *compression* "pushes" on the joint and a member in *tension* "pulls" on the joint. Also, be sure to choose a joint having at most two unknowns and at least one known force.

EXAMPLE 6.1

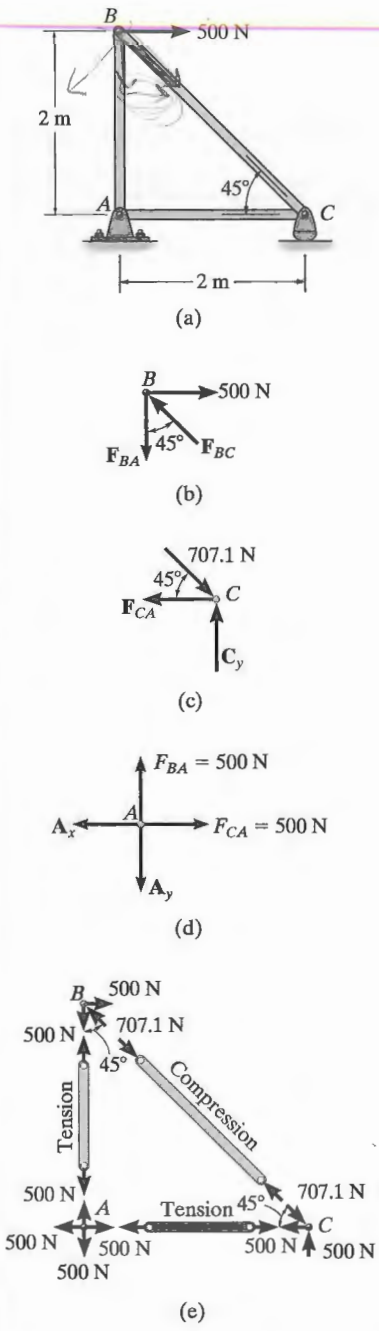


Fig. 6–8

Determine the force in each member of the truss shown in Fig. 6–8*a* and indicate whether the members are in tension or compression.

SOLUTION

Since we should have no more than two unknown forces at the joint and at least one known force acting there, we will begin our analysis at joint B.

Joint B. The free-body diagram of the joint at B is shown in Fig. 6–8*b*. Applying the equations of equilibrium, we have

$$\xrightarrow{+} \Sigma F_x = 0; \quad 500\text{ N} - F_{BC}\sin 45° = 0 \quad F_{BC} = 707.1\text{ N (C)} \quad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \quad F_{BC}\cos 45° - F_{BA} = 0 \quad F_{BA} = 500\text{ N (T)} \quad \textit{Ans.}$$

Since the force in member BC has been calculated, we can proceed to analyze joint C to determine the force in member CA and the support reaction at the rocker.

Joint C. From the free-body diagram of joint C, Fig. 6–8*c*, we have

$$\xrightarrow{+} \Sigma F_x = 0; \quad -F_{CA} + 707.1\cos 45°\text{ N} = 0 \quad F_{CA} = 500\text{ N (T)} \quad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \quad C_y - 707.1\sin 45°\text{ N} = 0 \quad C_y = 500\text{ N} \quad \textit{Ans.}$$

Joint A. Although it is not necessary, we can determine the components of the support reactions at joint A using the results of F_{CA} and F_{BA}. From the free-body diagram, Fig. 6–8*d*, we have

$$\xrightarrow{+} \Sigma F_x = 0; \quad 500\text{ N} - A_x = 0 \quad A_x = 500\text{ N}$$

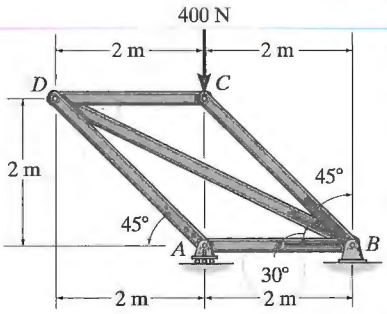
$$+\uparrow \Sigma F_y = 0; \quad 500\text{ N} - A_y = 0 \quad A_y = 500\text{ N}$$

NOTE: The results of the analysis are summarized in Fig. 6–8*e*. Note that the free-body diagram of each joint (or pin) shows the effects of all the connected members and external forces applied to the joint, whereas the free-body diagram of each member shows only the effects of the end joints on the member.

110003 – ENGINEERING ... from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 6.2

Determine the force in each member of the truss in Fig. 6–9*a* and indicate if the members are in tension or compression.



(a)

SOLUTION

Since joint C has one known and only two unknown forces acting on it, it is possible to start at this joint, then analyze joint D, and finally joint A. This way the support reactions will not have to be determined prior to starting the analysis.

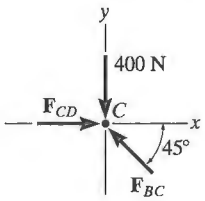
Joint C. By inspection of the force equilibrium, Fig. 6–9*b*, it can be seen that both members BC and CD must be in compression.

$$+\uparrow \Sigma F_y = 0; \qquad F_{BC} \sin 45° - 400\text{ N} = 0$$

$$F_{BC} = 565.69\text{ N} = 566\text{ N (C)} \qquad \textit{Ans.}$$

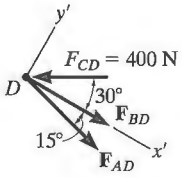
$$\xrightarrow{+} \Sigma F_x = 0; \qquad F_{CD} - (565.69\text{ N}) \cos 45° = 0$$

$$F_{CD} = 400\text{ N (C)} \qquad \textit{Ans.}$$



(b)

Joint D. Using the result $F_{CD} = 400$ N (C), the force in members BD and AD can be found by analyzing the equilibrium of joint D. We will assume $\mathbf{F}_{AD}$ and $\mathbf{F}_{BD}$ are both tensile forces, Fig. 6–9*c*. The x', y' coordinate system will be established so that the x' axis is directed along $\mathbf{F}_{BD}$. This way, we will eliminate the need to solve two equations simultaneously. Now $\mathbf{F}_{AD}$ can be obtained *directly* by applying $\Sigma F_{y'} = 0$.



(c)

$$+\nearrow \Sigma F_{y'} = 0; \qquad -F_{AD} \sin 15° - 400 \sin 30° = 0$$

$$F_{AD} = -772.74\text{ N} = 773\text{ N (C)} \qquad \textit{Ans.}$$

The negative sign indicates that $\mathbf{F}_{AD}$ is a compressive force. Using this result,

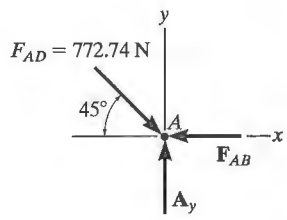
$$+\searrow \Sigma F_{x'} = 0; \qquad F_{BD} + (-772.74 \cos 15°) - 400 \cos 30° = 0$$

$$F_{BD} = 1092.82\text{ N} = 1.09\text{ kN (T)} \qquad \textit{Ans.}$$

Joint A. The force in member AB can be found by analyzing the equilibrium of joint A, Fig. 6–9*d*. We have

$$\xrightarrow{+} \Sigma F_x = 0; \qquad (772.74\text{ N}) \cos 45° - F_{AB} = 0$$

$$F_{AB} = 546.41\text{ N (C)} = 546\text{ N (C)} \qquad \textit{Ans.}$$



(d)

Fig. 6–9

Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 6.3

Determine the force in each member of the truss shown in Fig. 6–10*a*. Indicate whether the members are in tension or compression.

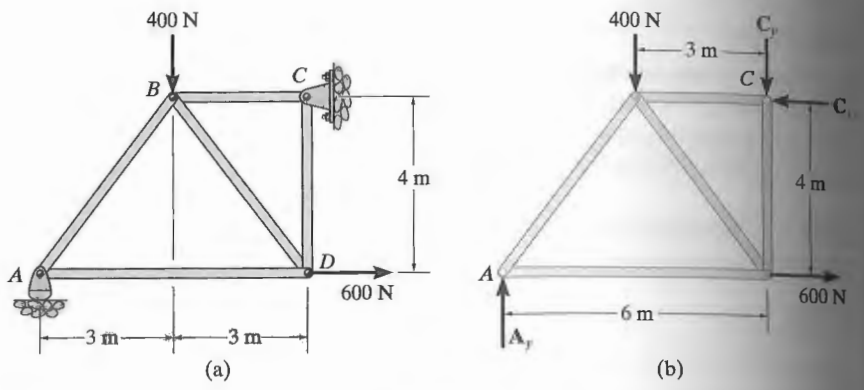


Fig. 6–10

6

SOLUTION

Support Reactions. No joint can be analyzed until the support reactions are determined, because each joint has more than three unknown forces acting on it. A free-body diagram of the entire truss is given in Fig. 6–10*b*. Applying the equations of equilibrium, we have

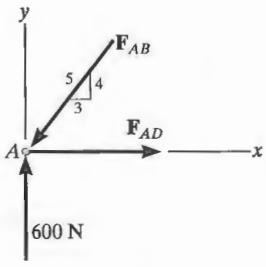
$$\overset{+}{\rightarrow} \Sigma F_x = 0; \qquad 600\text{ N} - C_x = 0 \qquad C_x = 600\text{ N}$$

$$\circlearrowleft + \Sigma M_C = 0; \qquad -A_y(6\text{ m}) + 400\text{ N}(3\text{ m}) + 600\text{ N}(4\text{ m}) = 0$$

$$A_y = 600\text{ N}$$

$$+\uparrow \Sigma F_y = 0; \qquad 600\text{ N} - 400\text{ N} - C_y = 0 \qquad C_y = 200\text{ N}$$

The analysis can now start at either joint A or C. The choice is arbitrary since there are one known and two unknown member forces acting on the pin at each of these joints.



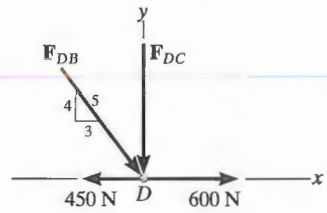
(c)

Joint A. (Fig. 6–10*c*). As shown on the free-body diagram, $\mathbf{F}_{AB}$ is assumed to be compressive and $\mathbf{F}_{AD}$ is tensile. Applying the equations of equilibrium, we have

$$+\uparrow \Sigma F_y = 0; \qquad 600\text{ N} - \tfrac{4}{5}F_{AB} = 0 \qquad F_{AB} = 750\text{ N} \quad (C) \qquad \textit{Ans.}$$

$$\overset{+}{\rightarrow} \Sigma F_x = 0; \qquad F_{AD} - \tfrac{3}{5}(750\text{ N}) = 0 \qquad F_{AD} = 450\text{ N} \quad (T) \qquad \textit{Ans.}$$

Joint D. (Fig. 6–10*d*). Using the result for F_{AD} and summing forces in the horizontal direction, Fig. 6–10*d*, we have



(d)

$$\xrightarrow{+} \Sigma F_x = 0; \qquad -450\text{ N} + \tfrac{3}{5}F_{DB} + 600\text{ N} = 0 \quad F_{DB} = -250\text{ N}$$

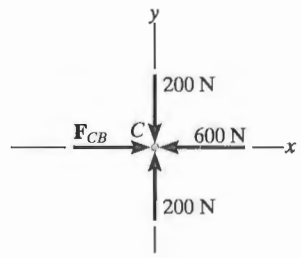
The negative sign indicates that $\mathbf{F}_{DB}$ acts in the *opposite sense* to that shown in Fig. 6–10*d*.* Hence,

$$F_{DB} = 250\text{ N (T)} \qquad \textit{Ans.}$$

To determine $\mathbf{F}_{DC}$, we can either correct the sense of $\mathbf{F}_{DB}$ on the free-body diagram, and then apply $\Sigma F_y = 0$, or apply this equation and retain the negative sign for F_{DB}, i.e.,

$$+\uparrow \Sigma F_y = 0; \qquad -F_{DC} - \tfrac{4}{5}(-250\text{ N}) = 0 \quad F_{DC} = 200\text{ N} \quad \text{(C)} \quad \textit{Ans.}$$

Joint C. (Fig. 6–10*e*).

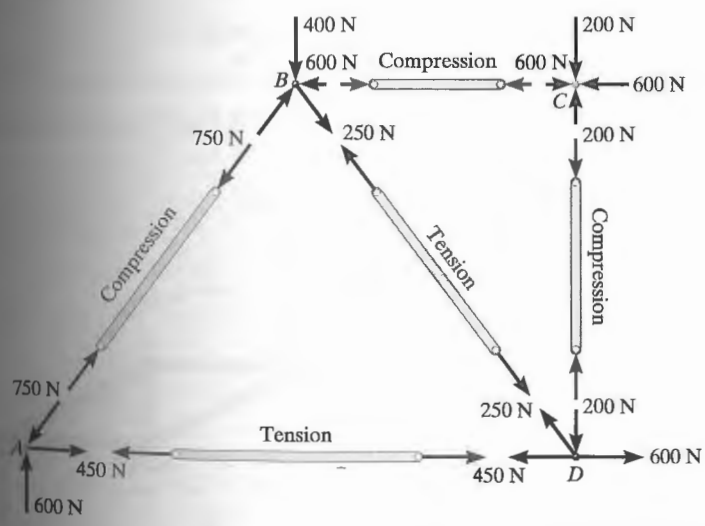


(e)

$$\xrightarrow{+} \Sigma F_x = 0; \qquad F_{CB} - 600\text{ N} = 0 \quad F_{CB} = 600\text{ N} \quad \text{(C)} \quad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \qquad 200\text{ N} - 200\text{ N} = 0 \quad \text{(check)}$$

NOTE: The analysis is summarized in Fig. 6–10*f*, which shows the free-body diagram for each joint and member.



(f)

*The proper sense could have been determined by inspection, prior to applying $\Sigma F_x = 0$.

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for The University of W

6.3 Zero-Force Members

Truss analysis using the method of joints is greatly simplified if we can first identify those members which support *no loading*. These *zero-force members* are used to increase the stability of the truss during construction and to provide added support if the loading is changed.

The zero-force members of a truss can generally be found *by inspection* of each of the joints. For example, consider the truss shown in Fig. 6–11*a*. If a free-body diagram of the pin at joint A is drawn, Fig. 6–11*b*, it is seen that members AB and AF are zero-force members. (We could not have come to this conclusion if we had considered the free-body diagrams of joints F or B simply because there are five unknowns at each of these joints.) In a similar manner, consider the free-body diagram of joint D, Fig. 6–11*c*. Here again it is seen that DC and DE are zero-force members. From these observations, we can conclude that *if only two members form a truss joint and no external load or support reaction is applied to the joint, the two members must be zero-force members.* The load on the truss in Fig. 6–11*a* is therefore supported by only five members as shown in Fig. 6–11*d*.

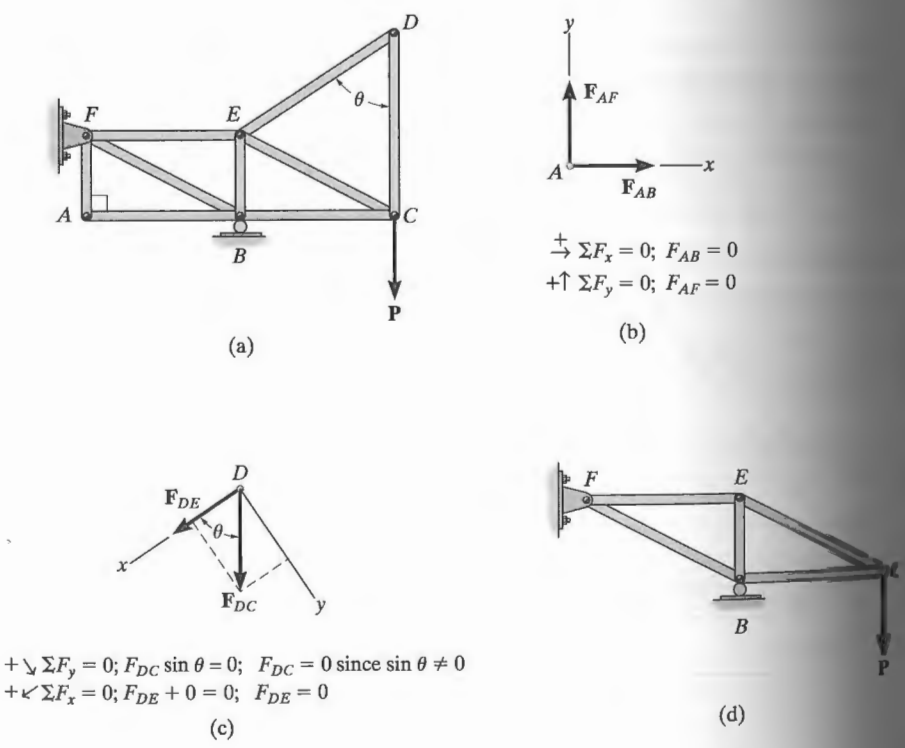


Fig. 6–11

from Pearson Custom Publishing exclusively for The University of Melbourne

Now consider the truss shown in Fig. 6–12*a*. The free-body diagram of the pin at joint *D* is shown in Fig. 6–12*b*. By orienting the *y* axis along members *DC* and *DE* and the *x* axis along member *DA*, it is seen that *DA* is a zero-force member. Note that this is also the case for member *CA*, Fig. 6–12*c*. In general then, *if three members form a truss joint for which two of the members are collinear, the third member is a zero-force member provided no external force or support reaction is applied to the joint.* The truss shown in Fig. 6–12*d* is therefore suitable for supporting the load **P**.

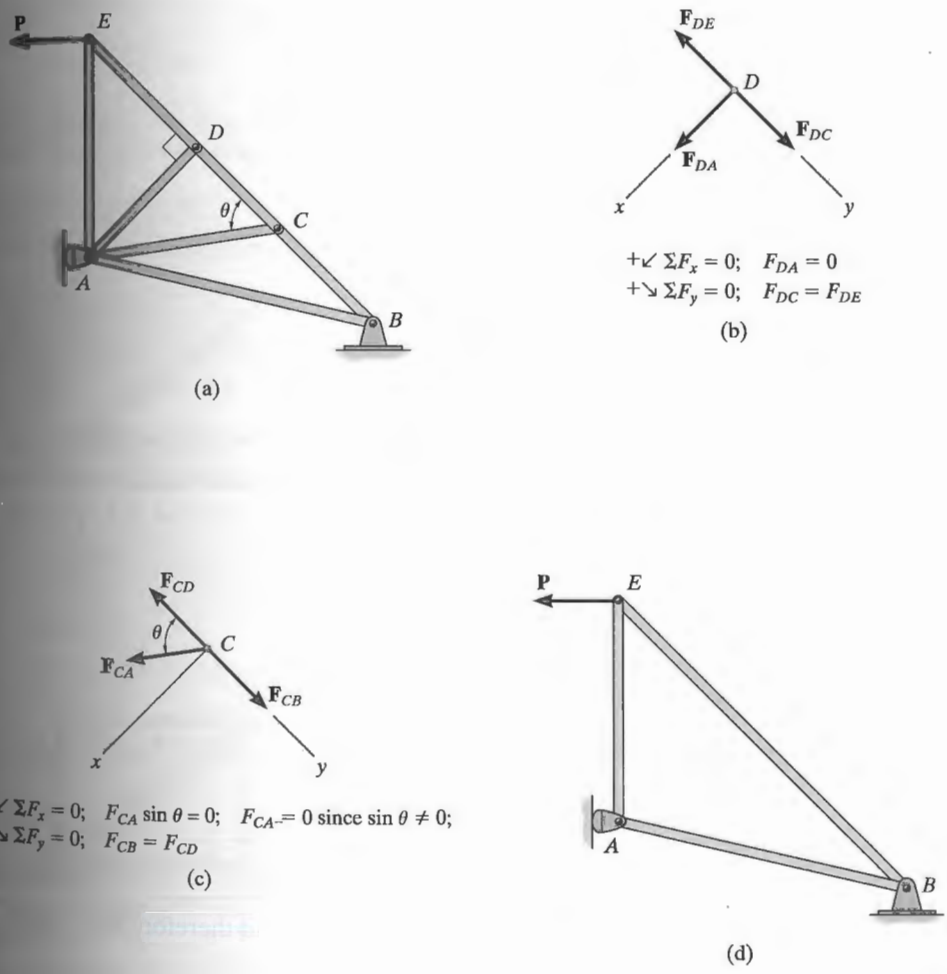
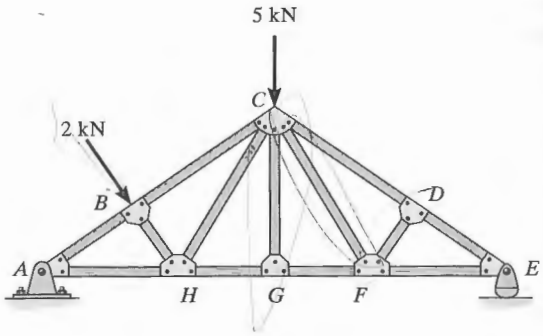


Fig. 6–12

Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 6.4

Using the method of joints, determine all the zero-force members of the *Fink roof truss* shown in Fig. 6–13*a*. Assume all joints are pin connected.



(a)

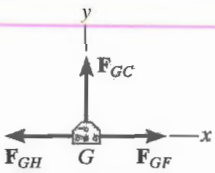
Fig. 6–13

SOLUTION

Look for joint geometries that have three members for which two are collinear. We have

Joint G. (Fig. 6–13*b*).

$+\uparrow \Sigma F_y = 0;$ $\quad F_{GC} = 0$ *Ans.*

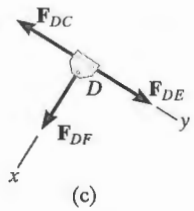


(b)

Realize that we could not conclude that GC is a zero-force member by considering joint C, where there are five unknowns. The fact that GC is a zero-force member means that the 5-kN load at C must be supported by members CB, CH, CF, and CD.

Joint D. (Fig. 6–13*c*).

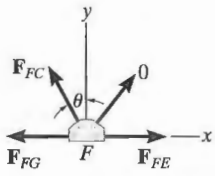
$+\swarrow \Sigma F_x = 0;$ $\quad F_{DF} = 0$ *Ans.*



(c)

Joint F. (Fig. 6–13*d*).

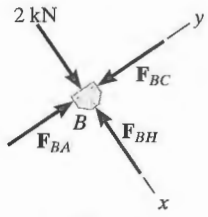
$+\uparrow \Sigma F_y = 0;$ $\quad F_{FC} \cos\theta = 0$ $\quad$ Since $\theta \neq 90°$, $\quad F_{FC} = 0$ *Ans.*



(d)

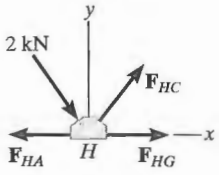
NOTE: If joint B is analyzed, Fig. 6–13*e*,

$+\searrow \Sigma F_x = 0;$ $\quad 2\text{ kN} - F_{BH} = 0 \quad F_{BH} = 2\text{ kN}$ (C)



(e)

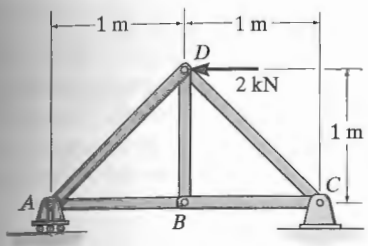
Also, F_{HC} must satisfy $\Sigma F_y = 0$, Fig. 6–13*f*, and therefore HC is *not* a zero-force member.



(f)

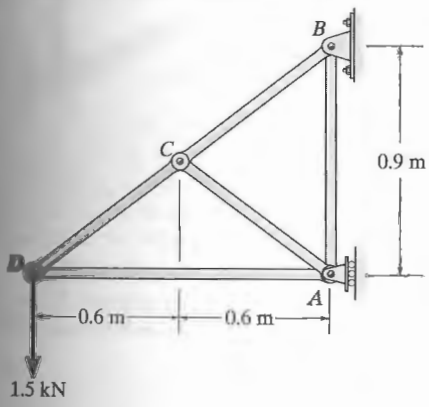
FUNDAMENTAL PROBLEMS

F6–1. Determine the force in each member of the truss. State if the members are in tension or compression.



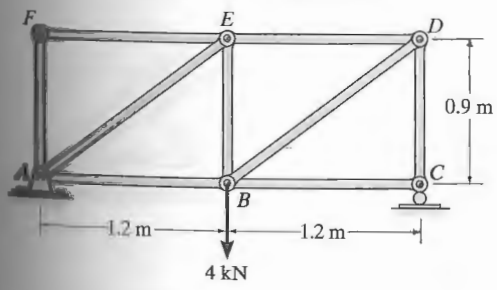
F6–1

F6–2. Determine the force in each member of the truss. State if the members are in tension or compression.



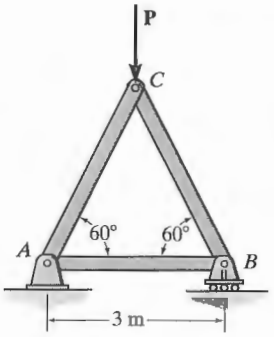
F6–2

F6–3. Determine the force in members AE and DC. State if the members are in tension or compression.



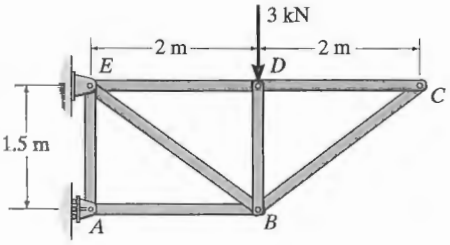
F6–3

F6–4. Determine the greatest load P that can be applied to the truss so that none of the members are subjected to a force exceeding either 2 kN in tension or 1.5 kN in compression.



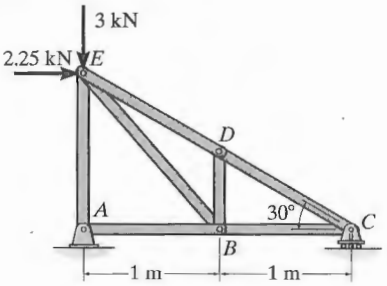
F6–4

F6–5. Identify the zero-force members in the truss.



F6–5

F6–6. Determine the force in each member of the truss. State if the members are in tension or compression.



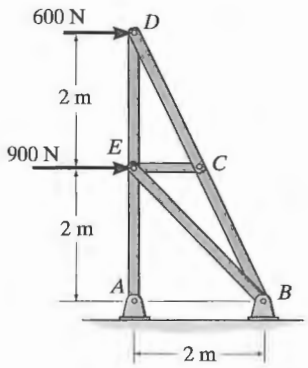
F6–6

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for The University of New South Wales

MasteringENGINEERING
Customized coaching is available!

PROBLEMS

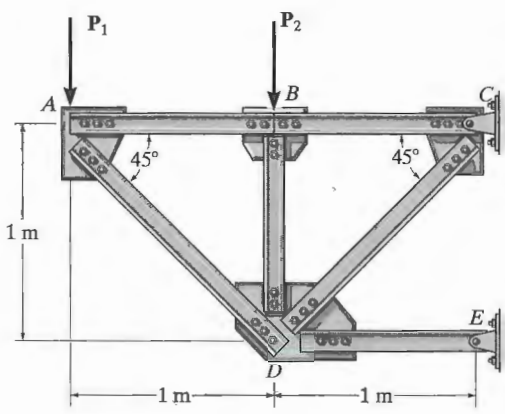
•6–1. Determine the force in each member of the truss, and state if the members are in tension or compression.



Prob. 6–1

6–2. The truss, used to support a balcony, is subjected to the loading shown. Approximate each joint as a pin and determine the force in each member. State whether the members are in tension or compression. Set $P_1 = 3$ kN, $P_2 = 2$ kN.

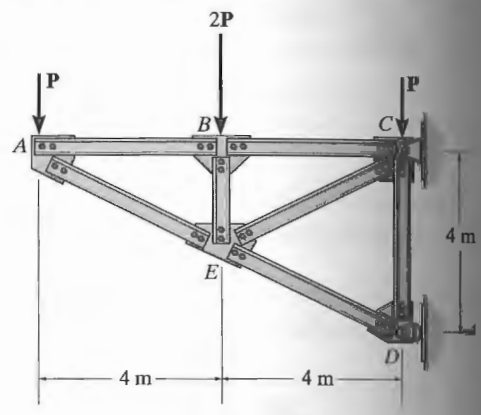
6–3. The truss, used to support a balcony, is subjected to the loading shown. Approximate each joint as a pin and determine the force in each member. State whether the members are in tension or compression. Set $P_1 = 4$ kN, $P_2 = 0$.



Probs. 6–2/3

*6–4. Determine the force in each member of the truss and state if the members are in tension or compression. Assume each joint as a pin. Set $P = 4$ kN.

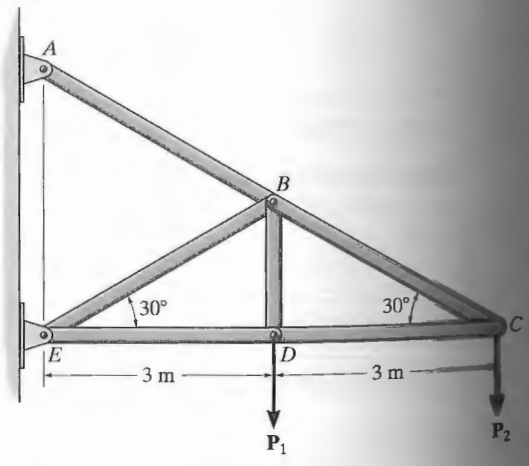
•6–5. Assume that each member of the truss is made of steel having a mass per length of 4 kg/m. Set $P = 0$, determine the force in each member, and indicate if the members are in tension or compression. Neglect the weight of the gusset plates and assume each joint is a pin. Solve the problem by assuming the weight of each member can be represented as a vertical force, half of which is applied at the end of each member.



Probs. 6–4/5

6–6. Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = 2$ kN and $P_2 = 1.5$ kN.

6–7. Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = P_2 = 4$ kN.



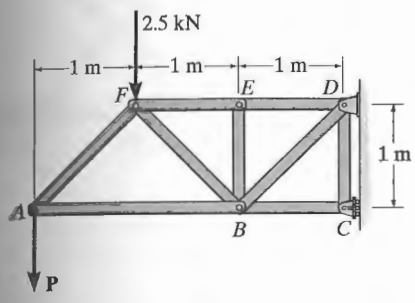
Probs. 6–6/7

Publishing exclusively for The University of Melbourne

6–8. Determine the force in each member of the truss, and state if the members are in tension or compression. Set $P = 4$ kN.

Video Solutions

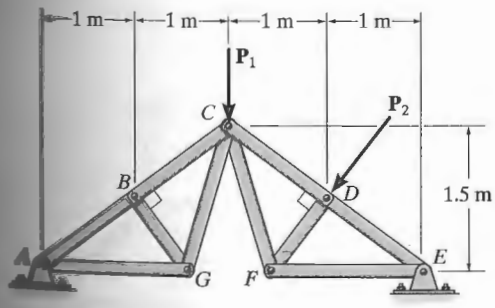
•6–9. Remove the 2.5-kN force and then determine the greatest force P that can be applied to the truss so that none of the members are subjected to a force exceeding either 4 kN in tension or 3 kN in compression.



Probs. 6–8/9

6–10. Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = 4$ kN, $P_2 = 0$.

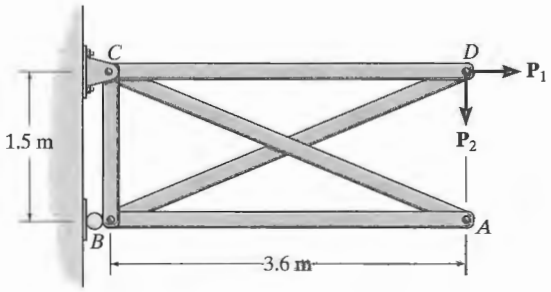
6–11. Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = 3$ kN, $P_2 = 2$ kN.



Probs. 6–10/11

***6–12.** Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = 1200$ N, $P_2 = 500$ N.

•6–13. Determine the largest load P_2 that can be applied to the truss so that the force in any member does not exceed 2.5 kN (T) or 1.75 kN (C). Take $P_1 = 0$.

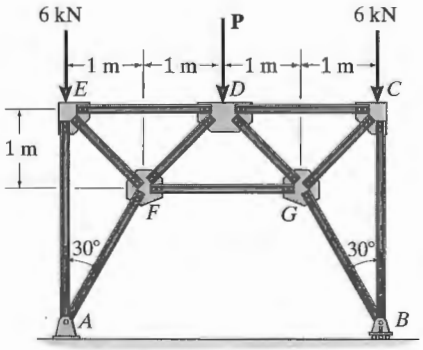


Probs. 6–12/13

6–14. Determine the force in each member of the truss, and state if the members are in tension or compression. Set $P = 12.5$ kN.

6–15. Remove the 6-kN forces and determine the greatest force P that can be applied to the truss so that none of the members are subjected to a force exceeding either 10 kN in tension or 7.5 kN in compression.

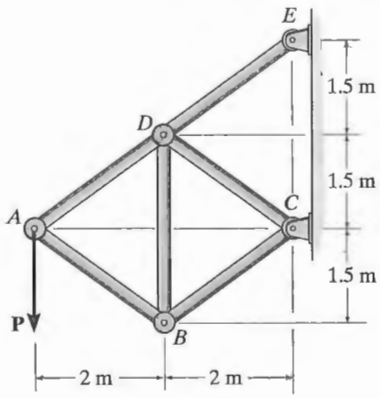
Video Solutions



Probs. 6–14/15

***6–16.** Determine the force in each member of the truss, and state if the members are in tension or compression. Set $P = 5$ kN.

•6–17. Determine the greatest force P that can be applied to the truss so that none of the members are subjected to a force exceeding either 2.5 kN in tension or 2 kN in compression.

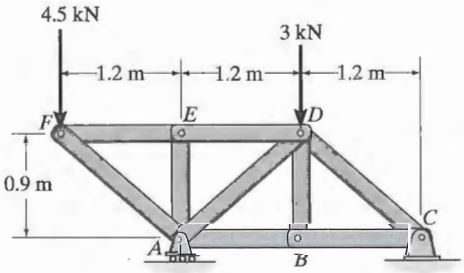


Probs. 6–16/17

6–18. Determine the force in each member of the truss, and state if the members are in tension or compression.

Video Solutions

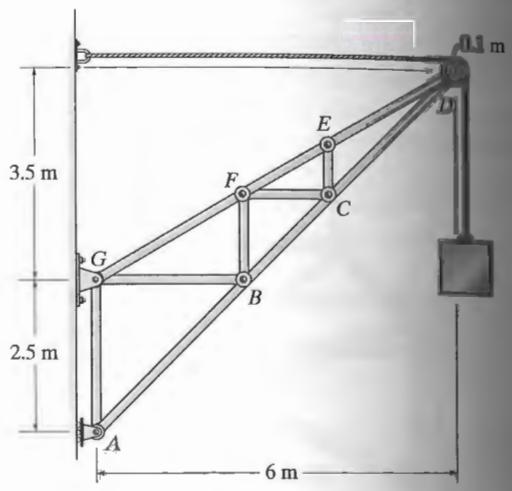
6–19. The truss is fabricated using members having a weight of 0.2 kN/m. Remove the external forces from the truss, and determine the force in each member due to the weight of the members. State whether the members are in tension or compression. Assume that the total force acting on a joint is the sum of half of the weight of every member connected to the joint.



Probs. 6–18/19

***6–20.** Determine the force in each member of the truss and state if the members are in tension or compression. The load has a mass of 40 kg.

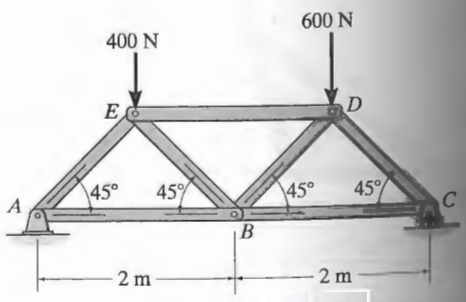
•6–21. Determine the largest mass m of the suspended block so that the force in any member does not exceed 30 kN (T) or 25 kN (C).



Probs. 6–20/21

6–22. Determine the force in each member of the truss, and state if the members are in tension or compression.

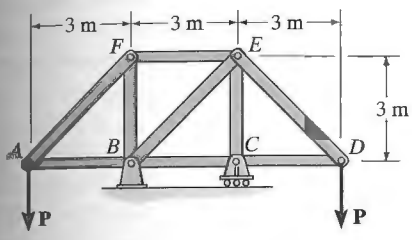
6–23. The truss is fabricated using uniform members having a mass of 5 kg/m. Remove the external forces from the truss, and determine the force in each member due to the weight of the truss. State whether the members are in tension or compression. Assume that the total force acting on a joint is the sum of half of the weight of every member connected to the joint.



Probs. 6–22/23

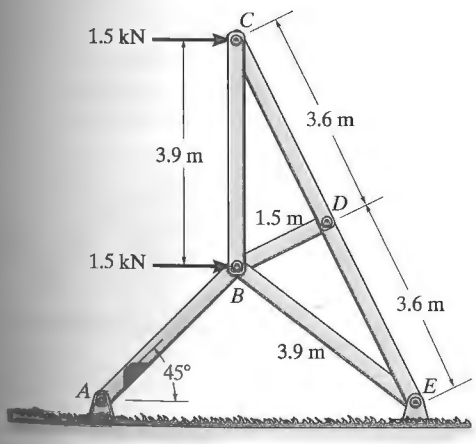
*6–24. Determine the force in each member of the truss, and state if the members are in tension or compression. Set $P = 4$ kN.

•6–25. Determine the greatest force P that can be applied to the truss so that none of the members are subjected to a force exceeding either 1.5 kN in tension or 1 kN in compression.



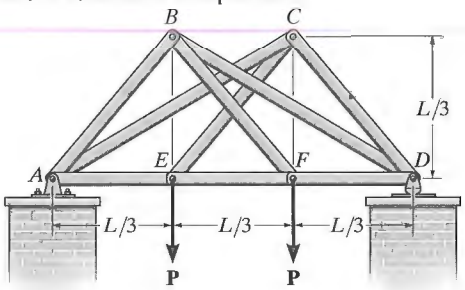
Probs. 6–24/25

6–26. A sign is subjected to a wind loading that exerts horizontal forces of 1.5 kN on joints B and C of one of the side supporting trusses. Determine the force in each member of the truss and state if the members are in tension or compression.



Prob. 6–26

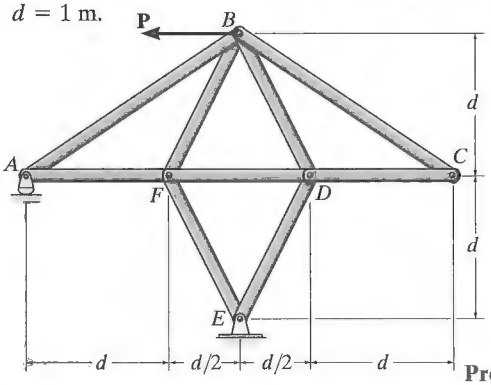
6–27. Determine the force in each member of the double scissors truss in terms of the load P and state if the members are in tension or compression.



Prob. 6–27

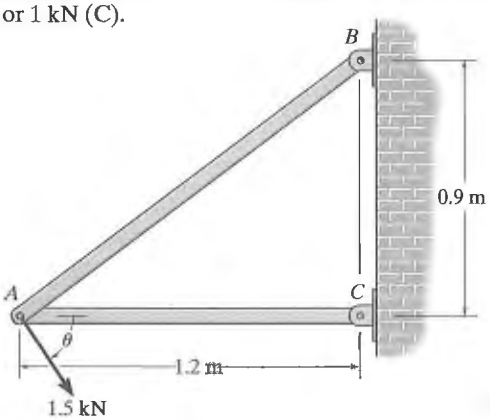
*6–28. Determine the force in each member of the truss in terms of the load P, and indicate whether the members are in tension or compression.

•6–29. If the maximum force that any member can support is 4 kN in tension and 3 kN in compression, determine the maximum force P that can be applied at joint B. Take $d = 1$ m.



Probs. 6–28/29

6–30. The two-member truss is subjected to the force of 1.5 kN. Determine the range of θ for application of the load so that the force in either member does not exceed 2 kN (T) or 1 kN (C).



Prob. 6–30

ENGR10003 – ENGINEERING SYSTEMS DESIGN 2
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

Simple trusses are often used in the construction of large cranes in order to reduce the weight of the boom and tower.

Procedure for Analysis

The forces in the members of a truss may be determined by the method of sections using the following procedure.

Free-Body Diagram.

- Make a decision on how to "cut" or section the truss through the members where forces are to be determined.
- Before isolating the appropriate section, it may first be necessary to determine the truss's support reactions. If this is done then the three equilibrium equations will be available to solve for member forces at the section.
- Draw the free-body diagram of that segment of the sectioned truss which has the least number of forces acting on it.
- Use one of the two methods described above for establishing the sense of the unknown member forces.

Equations of Equilibrium.

- Moments should be summed about a point that lies at the intersection of the lines of action of two unknown forces, so that the third unknown force can be determined directly from the moment equation.
- If two of the unknown forces are *parallel*, forces may be summed *perpendicular* to the direction of these unknowns to determine *directly* the third unknown force.

ion from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 6.5

Determine the force in members GE, GC, and BC of the truss shown in Fig. 6-16a. Indicate whether the members are in tension or compression.

Please refer to the Companion CD for the animation: *Free-body Diagram for a Truss*

SOLUTION

Section aa in Fig. 6–16a has been chosen since it cuts through the *three* members whose forces are to be determined. In order to use the method of sections, however, it is *first* necessary to determine the external reactions at A or D. Why? A free-body diagram of the entire truss is shown in Fig. 6–16b. Applying the equations of equilibrium, we have

$$\xrightarrow{+}\Sigma F_x = 0; \qquad 400\text{ N} - A_x = 0 \qquad A_x = 400\text{ N}$$

$$\circlearrowleft + \Sigma M_A = 0; \qquad -1200\text{ N}(8\text{ m}) - 400\text{ N}(3\text{ m}) + D_y(12\text{ m}) = 0$$

$$D_y = 900\text{ N}$$

$$+\uparrow \Sigma F_y = 0; \qquad A_y - 1200\text{ N} + 900\text{ N} = 0 \qquad A_y = 300\text{ N}$$

Free-Body Diagram. For the analysis the free-body diagram of the left portion of the sectioned truss will be used, since it involves the least number of forces, Fig. 6–16c.

Equations of Equilibrium. Summing moments about point G eliminates $\mathbf{F}_{GE}$ and $\mathbf{F}_{GC}$ and yields a direct solution for F_{BC}.

$$\circlearrowleft + \Sigma M_G = 0; \qquad -300\text{ N}(4\text{ m}) - 400\text{ N}(3\text{ m}) + F_{BC}(3\text{ m}) = 0$$

$$F_{BC} = 800\text{ N} \quad (\text{T}) \qquad \textit{Ans.}$$

In the same manner, by summing moments about point C we obtain a direct solution for F_{GE}.

$$\circlearrowleft + \Sigma M_C = 0; \qquad -300\text{ N}(8\text{ m}) + F_{GE}(3\text{ m}) = 0$$

$$F_{GE} = 800\text{ N} \quad (\text{C}) \qquad \textit{Ans.}$$

Since $\mathbf{F}_{BC}$ and $\mathbf{F}_{GE}$ have no vertical components, summing forces in the y direction directly yields F_{GC}, i.e.,

$$+\uparrow \Sigma F_y = 0; \qquad 300\text{ N} - \tfrac{3}{5}F_{GC} = 0$$

$$F_{GC} = 500\text{ N} \quad (\text{T}) \qquad \textit{Ans.}$$

NOTE: Here it is possible to tell, by inspection, the proper direction for each unknown member force. For example, $\Sigma M_C = 0$ requires $\mathbf{F}_{GE}$ to be *compressive* because it must balance the moment of the 300-N force about C.

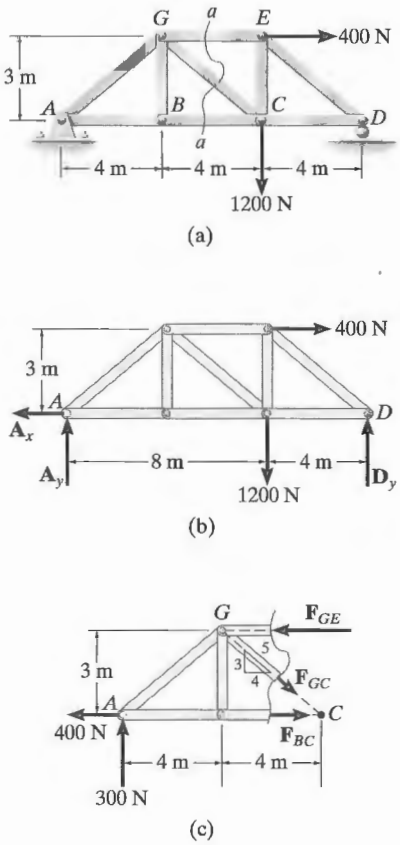


Fig. 6–16

EXAMPLE 6.6

Determine the force in member CF of the truss shown in Fig. 6–17a. Indicate whether the member is in tension or compression. Assume each member is pin connected.

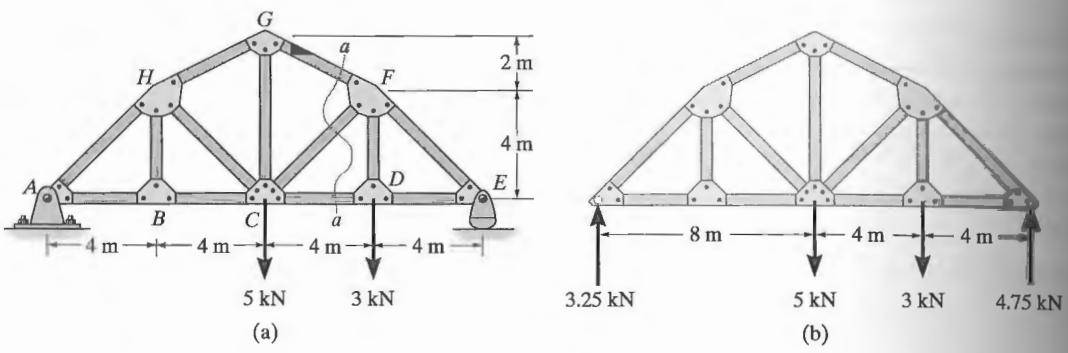
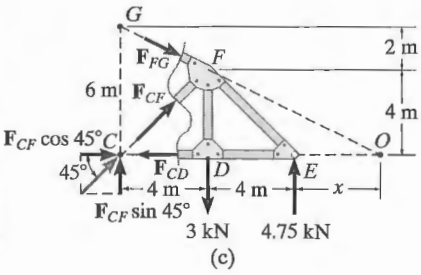


Fig. 6–17

SOLUTION

Free-Body Diagram. Section aa in Fig. 6–17a will be used since this section will "expose" the internal force in member CF as "external" on the free-body diagram of either the right or left portion of the truss. It is first necessary, however, to determine the support reactions on either the left or right side. Verify the results shown on the free-body diagram in Fig. 6–17b.



The free-body diagram of the right portion of the truss, which is the easiest to analyze, is shown in Fig. 6–17c. There are three unknowns, F_{FG}, F_{CF}, and F_{CD}.

Equations of Equilibrium. We will apply the moment equation about point O in order to eliminate the two unknowns F_{FG} and F_{CD}. The location of point O measured from E can be determined from proportional triangles, i.e., $4/(4 + x) = 6/(8 + x)$, $x = 4$ m. Or, stated in another manner, the slope of member GF has a drop of 2 m to a horizontal distance of 4 m. Since FD is 4 m, Fig. 6–17c, then from D to O the distance must be 8 m.

An easy way to determine the moment of $\mathbf{F}_{CF}$ about point O is to use the principle of transmissibility and slide $\mathbf{F}_{CF}$ to point C, and then resolve $\mathbf{F}_{CF}$ into its two rectangular components. We have

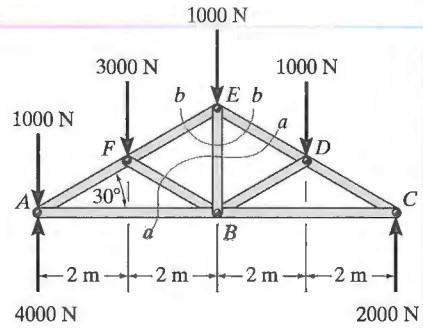
$$\circlearrowleft + \Sigma M_O = 0;$$

$$-F_{CF} \sin 45°(12 \text{ m}) + (3 \text{ kN})(8 \text{ m}) - (4.75 \text{ kN})(4 \text{ m}) = 0$$

$$F_{CF} = 0.589 \text{ kN} \quad \text{(C)} \qquad \textit{Ans.}$$

EXAMPLE 6.7

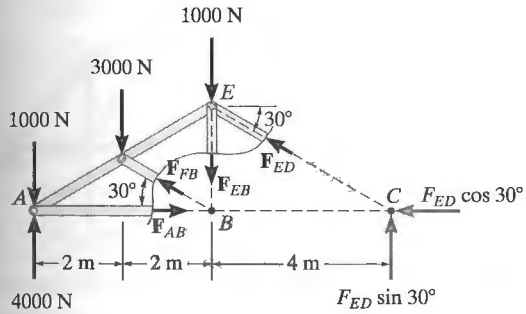
Determine the force in member EB of the roof truss shown in Fig. 6–18a. Indicate whether the member is in tension or compression.



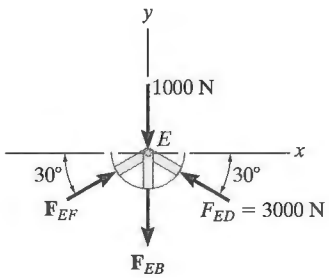
(a)

SOLUTION

Free-Body Diagrams. By the method of sections, any imaginary section that cuts through EB, Fig. 6–18a, will also have to cut through three other members for which the forces are unknown. For example, section aa cuts through ED, EB, FB, and AB. If a free-body diagram of the left side of this section is considered, Fig. 6–18b, it is possible to obtain $\mathbf{F}_{ED}$ by summing moments about B to eliminate the other three unknowns; however, $\mathbf{F}_{EB}$ cannot be determined from the remaining two equilibrium equations. One possible way of obtaining $\mathbf{F}_{EB}$ is first to determine $\mathbf{F}_{ED}$ from section aa, then use this result on section bb, Fig. 6–18a, which is shown in Fig. 6–18c. Here the force system is concurrent and our sectioned free-body diagram is the same as the free-body diagram for the joint at E.



(b)



(c)

Fig. 6–18

Equations of Equilibrium. In order to determine the moment of $\mathbf{F}_{ED}$ about point B, Fig. 6–18b, we will use the principle of transmissibility and slide the force to point C and then resolve it into its rectangular components as shown. Therefore,

$$\circlearrowleft + \Sigma M_B = 0; \qquad 1000\text{ N}(4\text{ m}) + 3000\text{ N}(2\text{ m}) - 4000\text{ N}(4\text{ m}) + F_{ED}\sin 30°(4\text{ m}) = 0$$

$$F_{ED} = 3000\text{ N} \quad (\text{C})$$

Considering now the free-body diagram of section bb, Fig. 6–18c, we have

$$\xrightarrow{+} \Sigma F_x = 0; \qquad F_{EF}\cos 30° - 3000\cos 30°\text{ N} = 0$$

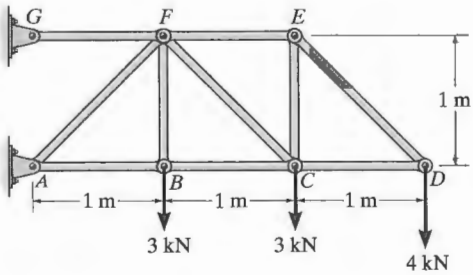
$$F_{EF} = 3000\text{ N} \quad (\text{C})$$

$$+\uparrow \Sigma F_y = 0; \qquad 2(3000\sin 30°\text{ N}) - 1000\text{ N} - F_{EB} = 0$$

$$F_{EB} = 2000\text{ N} \quad (\text{T}) \qquad \textit{Ans.}$$

FUNDAMENTAL PROBLEMS

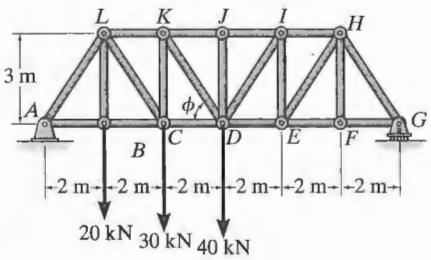
F6–7. Determine the force in members *BC*, *CF*, and *FE*. State if the members are in tension or compression.



F6–7

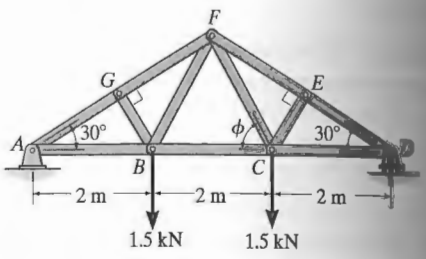
F6–8. Determine the force in members *LK*, *KC*, and *CD* of the Pratt truss. State if the members are in tension or compression.

F6–9. Determine the force in members *KJ*, *KD*, and *CD* of the Pratt truss. State if the members are in tension or compression.



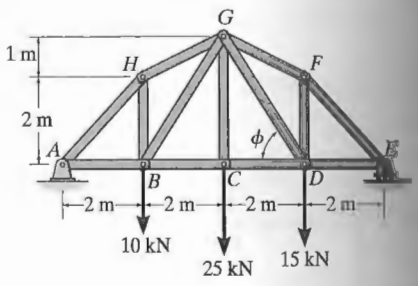
F6–8/9

F6–10. Determine the force in members *EF*, *CF*, and *BC* of the truss. State if the members are in tension or compression.



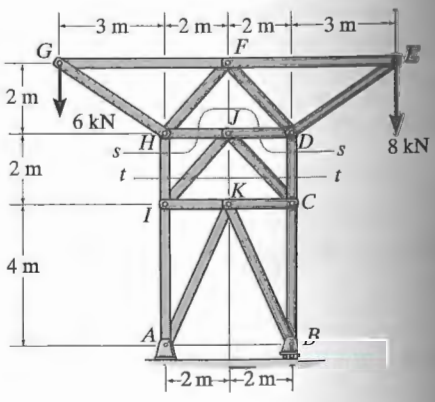
F6–10

F6–11. Determine the force in members *GF*, *GD*, and *CD* of the truss. State if the members are in tension or compression.



F6–11

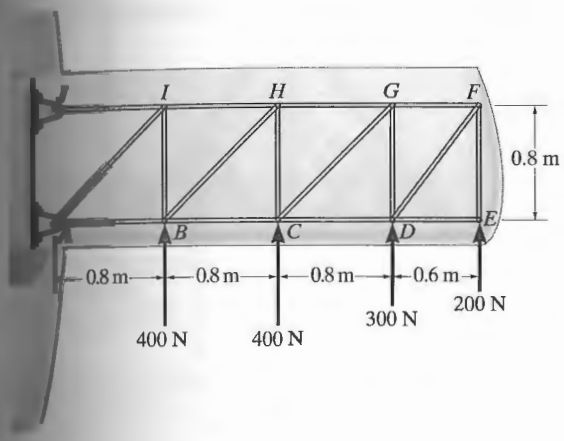
F6–12. Determine the force in members *DC*, *HI*, and *JI* of the truss. State if the members are in tension or compression.



F6–12

PROBLEMS

6–31. The internal drag truss for the wing of a light airplane is subjected to the forces shown. Determine the force in members *BC*, *BH*, and *HC*, and state if the members are in tension or compression.

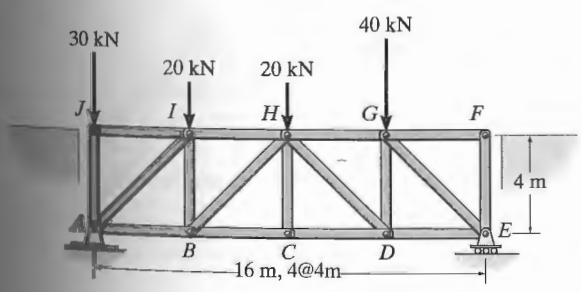


Prob. 6–31

***6–32.** The *Howe bridge truss* is subjected to the loading shown. Determine the force in members *HD*, *CD*, and *GD*, and state if the members are in tension or compression.

•6–33. The *Howe bridge truss* is subjected to the loading shown. Determine the force in members *HI*, *HB*, and *BC*, and state if the members are in tension or compression.

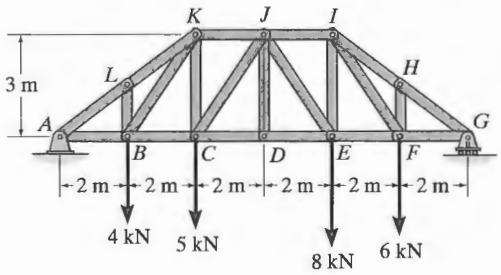
Video Solutions



Probs. 6–32/33

6–34. Determine the force in members *JK*, *CJ*, and *CD* of the truss, and state if the members are in tension or compression.

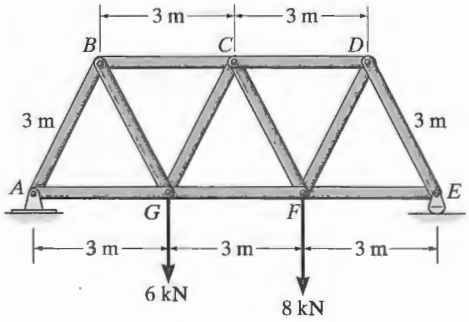
6–35. Determine the force in members *HI*, *FI*, and *EF* of the truss, and state if the members are in tension or compression.



Probs. 6–34/35

***6–36.** Determine the force in members *BC*, *CG*, and *GF* of the *Warren* truss. Indicate if the members are in tension or compression.

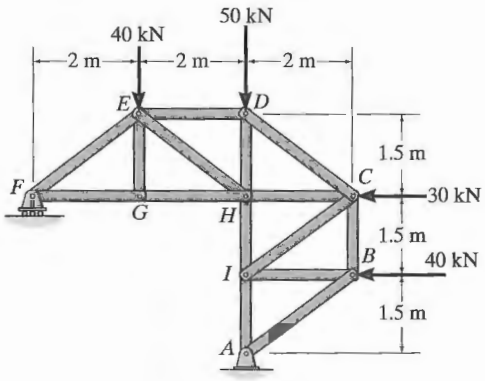
•6–37. Determine the force in members *CD*, *CF*, and *FG* of the *Warren truss*. Indicate if the members are in tension or compression.



Probs. 6–36/37

6–38. Determine the force in members DC, HC, and HI of the truss, and state if the members are in tension or compression.

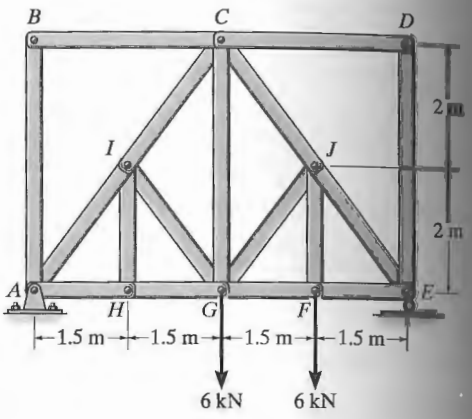
6–39. Determine the force in members ED, EH, and GH of the truss, and state if the members are in tension or compression.



Probs. 6–38/39

6–42. Determine the force in members IC and CG of the truss and state if these members are in tension or compression. Also, indicate all zero-force members.

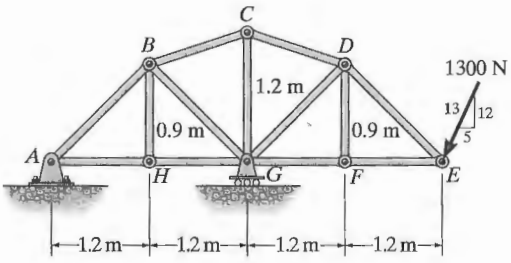
6–43. Determine the force in members JE and GF of the truss and state if these members are in tension or compression. Also, indicate all zero-force members.



Probs. 6–42/43

***6–40.** Determine the force in members GF, GD, and CD of the truss and state if the members are in tension or compression.

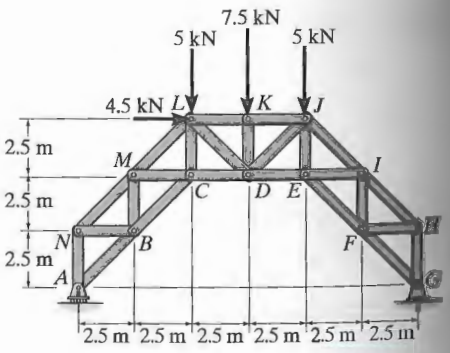
•6–41. Determine the force in members BG, BC, and HG of the truss and state if the members are in tension or compression.



Probs. 6–40/41

***6–44.** Determine the force in members JI, EF, EI, and JE of the truss, and state if the members are in tension or compression.

•6–45. Determine the force in members CD, LD, and KL of the truss, and state if the members are in tension or compression.

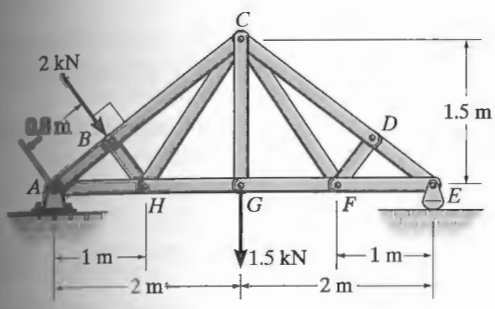


Probs. 6–44/45

6–46. Determine the force developed in members BC and CH of the roof truss and state if the members are in tension or compression.

Video Solutions

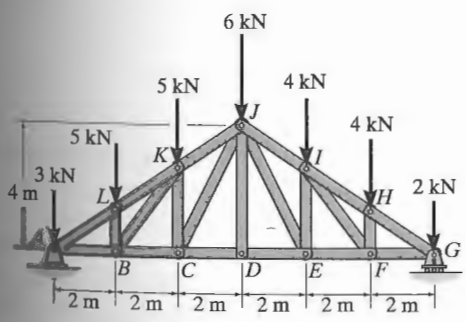
6–47. Determine the force in members CD and GF of the truss and state if the members are in tension or compression. Also indicate all zero-force members.



Probs. 6–46/47

***6–48.** Determine the force in members IJ, EJ, and CD of the *Howe* truss, and state if the members are in tension or compression.

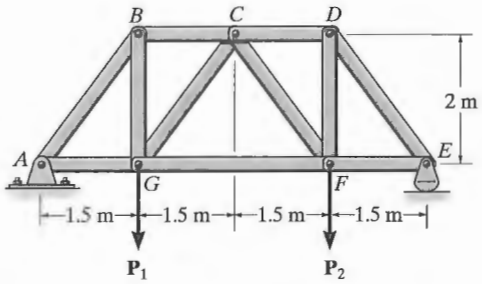
•6–49. Determine the force in members KJ, KC, and BC of the *Howe* truss, and state if the members are in tension or compression.



Probs. 6–48/49

6–50. Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = 20$ kN, $P_2 = 10$ kN.

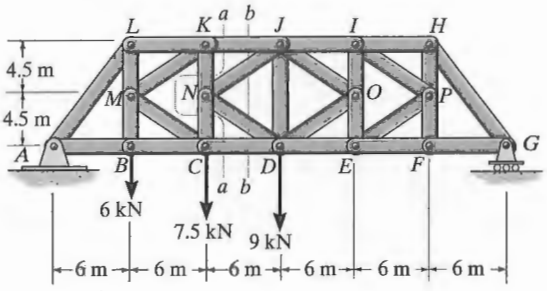
6–51. Determine the force in each member of the truss and state if the members are in tension or compression. Set $P_1 = 40$ kN, $P_2 = 20$ kN.



Probs. 6–50/51

***6–52.** Determine the force in members KJ, NJ, ND, and CD of the *K truss*. Indicate if the members are in tension or compression. *Hint:* Use sections *aa* and *bb*.

•6–53. Determine the force in members JI and DE of the *K truss*. Indicate if the members are in tension or compression.



Probs. 6–52/53

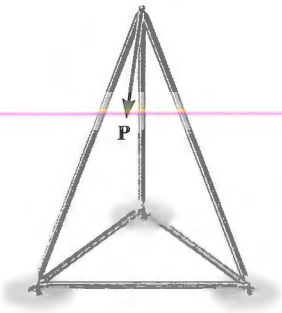
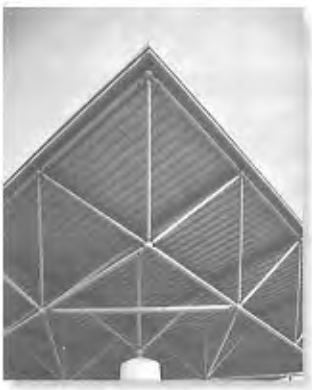


Fig. 6–19



Typical roof-supporting space truss. Notice the use of ball-and-socket joints for the connections.

For economic reasons, large electrical transmission towers are often constructed using space trusses.

*6.5 Space Trusses

A *space truss* consists of members joined together at their ends to form a stable three-dimensional structure. The simplest form of a space truss is a *tetrahedron*, formed by connecting six members together, as shown in Fig. 6–19. Any additional members added to this basic element would be redundant in supporting the force **P**. A *simple space truss* can be built from this basic tetrahedral element by adding three additional members and a joint, and continuing in this manner to form a system of multiconnected tetrahedrons.

Assumptions for Design The members of a space truss may be treated as two-force members provided the external loading is applied at the joints and the joints consist of ball-and-socket connections. These assumptions are justified if the welded or bolted connections of the joined members intersect at a common point and the weight of the members can be neglected. In cases where the weight of a member is to be included in the analysis, it is generally satisfactory to apply it as a vertical force, half of its magnitude applied at each end of the member.

Procedure for Analysis

Either the method of joints or the method of sections can be used to determine the forces developed in the members of a simple space truss.

Method of Joints.

If the forces in *all* the members of the truss are to be determined, then the method of joints is most suitable for the analysis. Here it is necessary to apply the three equilibrium equations $\Sigma F_x = 0$, $\Sigma F_y = 0$, $\Sigma F_z = 0$ to the forces acting at each joint. Remember that the solution of many simultaneous equations can be avoided if the force analysis begins at a joint having at least one known force and at most three unknown forces. Also, if the three-dimensional geometry of the force system at the joint is hard to visualize, it is recommended that a Cartesian vector analysis be used for the solution.

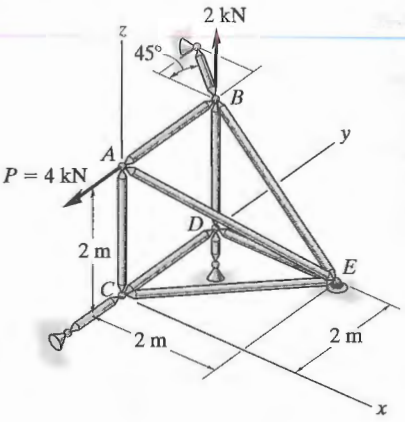
Method of Sections.

If only a *few* member forces are to be determined, the method of sections can be used. When an imaginary section is passed through a truss and the truss is separated into two parts, the force system acting on one of the segments must satisfy the *six* equilibrium equations $\Sigma F_x = 0$, $\Sigma F_y = 0$, $\Sigma F_z = 0$, $\Sigma M_x = 0$, $\Sigma M_y = 0$, $\Sigma M_z = 0$ (Eqs. 5–6). By proper choice of the section and axes for summing forces and moments, many of the unknown member forces in a space truss can be computed *directly*, using a single equilibrium equation.

from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 6.8

Determine the forces acting in the members of the space truss shown in Fig. 6–20*a*. Indicate whether the members are in tension or compression.



(a)

SOLUTION

Since there are one known force and three unknown forces acting at joint A, the force analysis of the truss will begin at this joint.

Joint A. (Fig. 6–20*b*). Expressing each force acting on the free-body diagram of joint A as a Cartesian vector, we have

$$\mathbf{P} = \{-4\mathbf{j}\}\text{ kN}, \qquad \mathbf{F}_{AB} = F_{AB}\mathbf{j}, \quad \mathbf{F}_{AC} = -F_{AC}\mathbf{k},$$

$$\mathbf{F}_{AE} = F_{AE}\left(\frac{\mathbf{r}_{AE}}{r_{AE}}\right) = F_{AE}(0.577\mathbf{i} + 0.577\mathbf{j} - 0.577\mathbf{k})$$

For equilibrium,

$$\Sigma\mathbf{F} = \mathbf{0}; \qquad \mathbf{P} + \mathbf{F}_{AB} + \mathbf{F}_{AC} + \mathbf{F}_{AE} = \mathbf{0}$$

$$-4\mathbf{j} + F_{AB}\mathbf{j} - F_{AC}\mathbf{k} + 0.577F_{AE}\mathbf{i} + 0.577F_{AE}\mathbf{j} - 0.577F_{AE}\mathbf{k} = \mathbf{0}$$

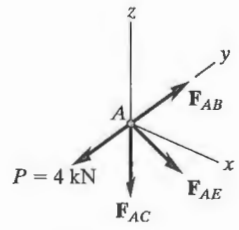
$$\Sigma F_x = 0; \qquad 0.577F_{AE} = 0$$

$$\Sigma F_y = 0; \qquad -4 + F_{AB} + 0.577F_{AE} = 0$$

$$\Sigma F_z = 0; \qquad -F_{AC} - 0.577F_{AE} = 0$$

$$F_{AC} = F_{AE} = 0 \qquad \textit{Ans.}$$

$$F_{AB} = 4\text{ kN} \quad (\text{T}) \qquad \textit{Ans.}$$



(b)

Since F_{AB} is known, joint B can be analyzed next.

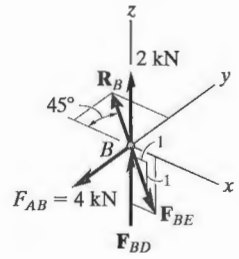
Joint B. (Fig. 6–20*c*).

$$\Sigma F_x = 0; \qquad -R_B\cos 45° + 0.707F_{BE} = 0$$

$$\Sigma F_y = 0; \qquad -4 + R_B\sin 45° = 0$$

$$\Sigma F_z = 0; \qquad 2 + F_{BD} - 0.707F_{BE} = 0$$

$$R_B = F_{BE} = 5.66\text{ kN} \quad (\text{T}), \qquad F_{BD} = 2\text{ kN} \quad (\text{C}) \qquad \textit{Ans.}$$



(c)

Fig. 6–20

The *scalar* equations of equilibrium may also be applied directly to the forces acting on the free-body diagrams of joints D and C since the force components are easily determined. Show that

$$F_{DE} = F_{DC} = F_{CE} = 0 \qquad \textit{Ans.}$$

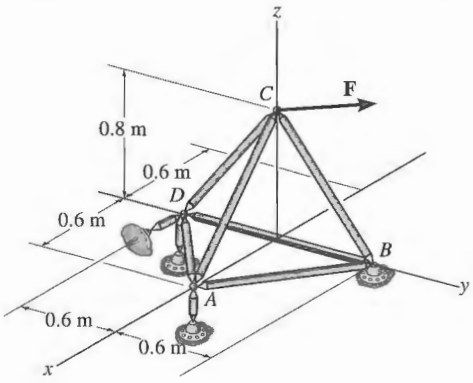
Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

Mastering ENGIN[redacted]ING
Customized coaching is available!

PROBLEMS

6–54. The space truss supports a force $\mathbf{F} = \{-500\mathbf{i} + 600\mathbf{j} + 400\mathbf{k}\}$ N. Determine the force in each member, and state if the members are in tension or compression.

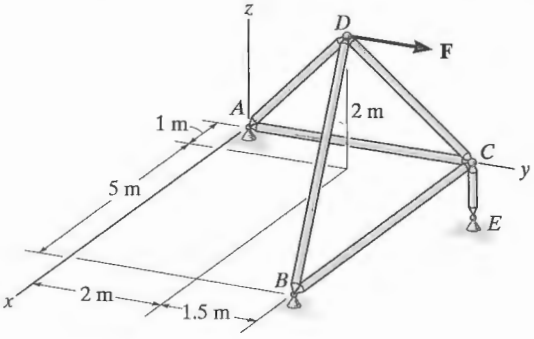
6–55. The space truss supports a force $\mathbf{F} = \{600\mathbf{i} + 450\mathbf{j} - 750\mathbf{k}\}$ N. Determine the force in each member, and state if the members are in tension or compression.



Probs. 6–54/55

***6–56.** Determine the force in each member of the space truss and state if the members are in tension or compression. The truss is supported by ball-and-socket joints at A, B, and E. Set $\mathbf{F} = \{800\mathbf{j}\}$ N. *Hint:* The support reaction at E acts along member EC. Why?

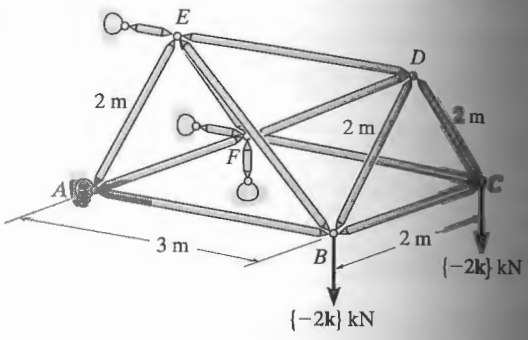
•6–57. Determine the force in each member of the space truss and state if the members are in tension or compression. The truss is supported by ball-and-socket joints at A, B, and E. Set $\mathbf{F} = \{-200\mathbf{i} + 400\mathbf{j}\}$ N. *Hint:* The support reaction at E acts along member EC. Why?



Probs. 6–56/57

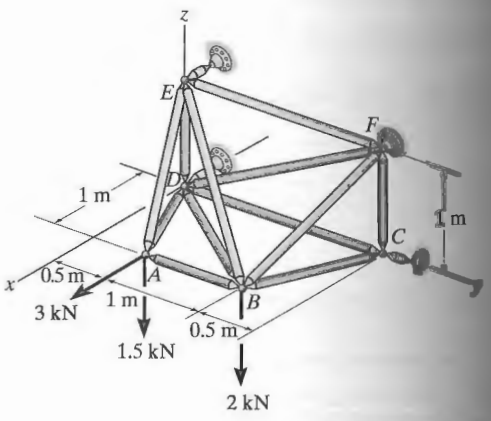
6–58. Determine the force in members BE, DE, and BC of the space truss and state if the members are in tension or compression.

6–59. Determine the force in members AB, CD, ED, and CF of the space truss and state if the members are in tension or compression.



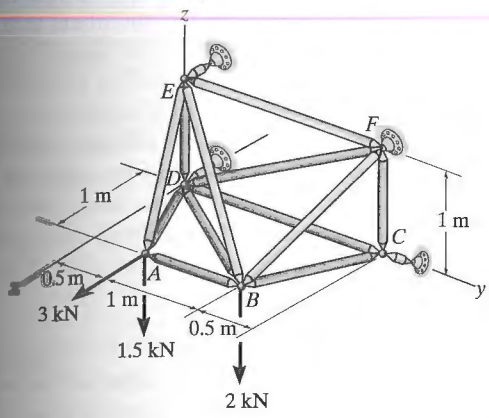
Probs. 6–58/59

***6–60.** Determine the force in the members AB, AE, BC, BF, BD, and BE of the space truss, and state if the members are in tension or compression.



Prob. 6–60

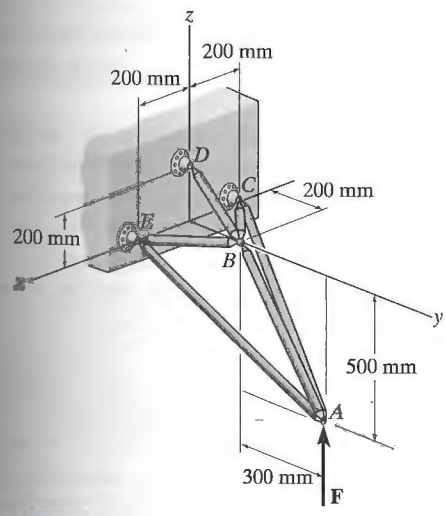
6–61. Determine the force in the members *EF*, *DF*, *CF*, and *CD* of the space truss, and state if the members are in tension or compression.



Prob. 6–61

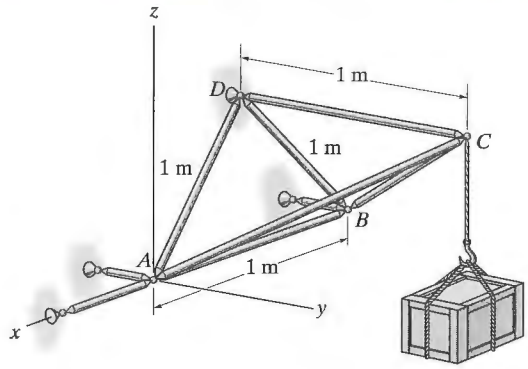
6–62. If the truss supports a force of $F = 200$ N, determine the force in each member and state if the members are in tension or compression.

6–63. If each member of the space truss can support a maximum force of 600 N in compression and 800 N in tension, determine the greatest force F the truss can support.



Probs. 6–62/63

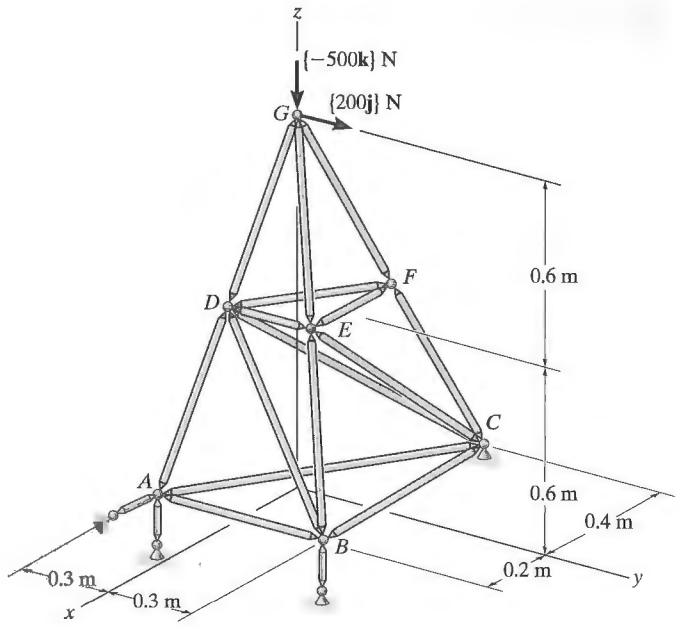
***6–64.** Determine the force developed in each member of the space truss and state if the members are in tension or compression. The crate has a weight of 750 N.



Prob. 6–64

•6–65. Determine the force in members *FE* and *ED* of the space truss and state if the members are in tension or compression. The truss is supported by a ball-and-socket joint at *C* and short links at *A* and *B*.

6–66. Determine the force in members *GD*, *GE*, and *FD* of the space truss and state if the members are in tension or compression.



Probs. 6–65/66

6.6 Frames and Machines

Frames and machines are two types of structures which are often composed of pin-connected *multiforce members*, i.e., members that are subjected to more than two forces. *Frames* are used to support loads, whereas *machines* contain moving parts and are designed to transmit and alter the effect of forces. Provided a frame or machine contains no more supports or members than are necessary to prevent its collapse, the forces acting at the joints and supports can be determined by applying the equations of equilibrium to each of its members. Once these forces are obtained, it is then possible to *design* the size of the members, connections, and supports using the theory of mechanics of materials and an appropriate engineering design code.

This large crane is a typical example of a framework.

Free-Body Diagrams. In order to determine the forces acting at the joints and supports of a frame or machine, the structure must be disassembled and the free-body diagrams of its parts must be drawn. The following important points *must* be observed:

- Isolate each part by drawing its *outlined shape*. Then show all the forces and/or couple moments that act on the part. Make sure to *label* or *identify* each known and unknown force and couple moment with reference to an established x, y coordinate system. Also, indicate any dimensions used for taking moments. Most often the equations of equilibrium are easier to apply if the forces are represented by their rectangular components. As usual, the sense of an unknown force or couple moment can be assumed.
- Identify all the two-force members in the structure and represent their free-body diagrams as having two equal but opposite collinear forces acting at their points of application. (See Sec. 5.4.) By recognizing the two-force members, we can avoid solving an unnecessary number of equilibrium equations.
- Forces common to any two *contacting* members act with equal magnitudes but opposite sense on the respective members. If the two members are treated as a *"system" of connected members*, then these forces are *"internal"* and are *not shown* on the *free-body diagram of the system*; however, if the free-body diagram of *each member* is drawn, the forces are *"external"* and *must* be shown on each of the free-body diagrams.

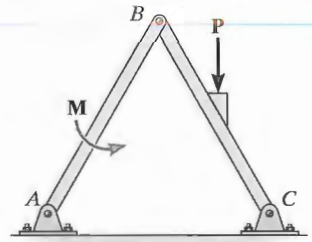
Common tools such as these pliers act as simple machines. Here the applied force on the handles creates a much larger force at the jaws.

The following examples graphically illustrate how to draw the free-body diagrams of a dismembered frame or machine. In all cases, the weight of the members is neglected.

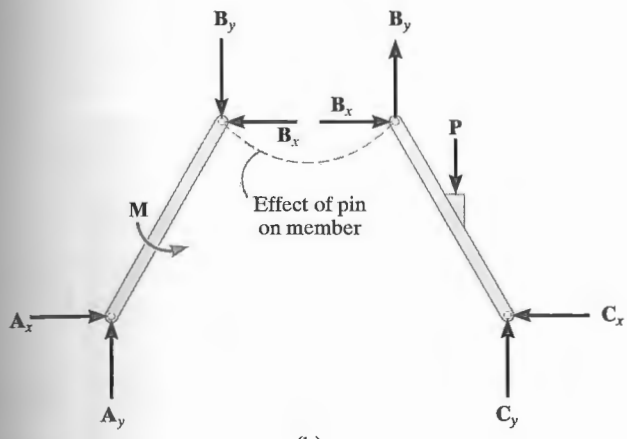
Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 6.9

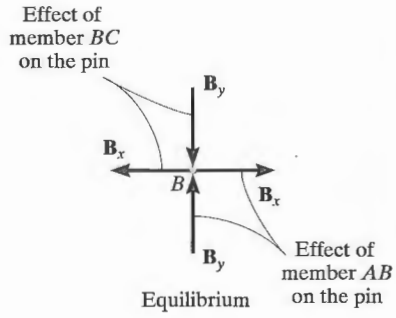
For the frame shown in Fig. 6–21*a*, draw the free-body diagram of (a) each member, (b) the pin at *B*, and (c) the two members connected together.



(a)



(b)



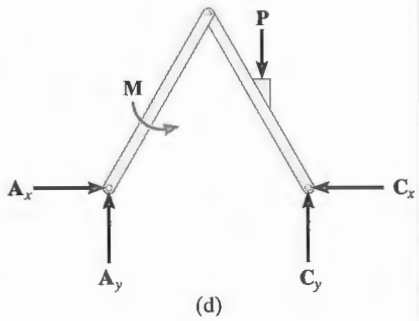
(c)

SOLUTION

Part (a). By inspection, members *BA* and *BC* are *not* two-force members. Instead, as shown on the free-body diagrams, Fig. 6–21*b*, *BC* is subjected to a force from the pins at *B* and *C* and the external force **P**. Likewise, *AB* is subjected to a force from the pins at *A* and *B* and the external couple moment **M**. The pin forces are represented by their *x* and *y* components.

Part (b). The pin at *B* is subjected to only *two forces*, i.e., the force of member *BC* and the force of member *AB*. For *equilibrium*, these forces or their respective components must be equal but opposite, Fig. 6–21*c*. Realize that Newton's third law is applied between the pin and its connected members, i.e., the effect of the pin on the two members, Fig. 6–21*b*, and the equal but opposite effect of the two members on the pin, Fig. 6–21*c*.

Part (c). The free-body diagram of both members connected together, yet removed from the supporting pins at *A* and *C*, is shown in Fig. 6–21*d*. The force components $\mathbf{B}_x$ and $\mathbf{B}_y$ are *not shown* on this diagram since they are *internal* forces (Fig. 6–21*b*) and therefore cancel out. Also, to be consistent when later applying the equilibrium equations, the unknown force components at *A* and *C* must act in the *same sense* as those shown in Fig. 6–21*b*.



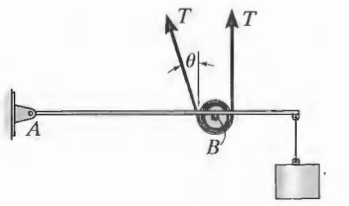
(d)

Fig. 6–21

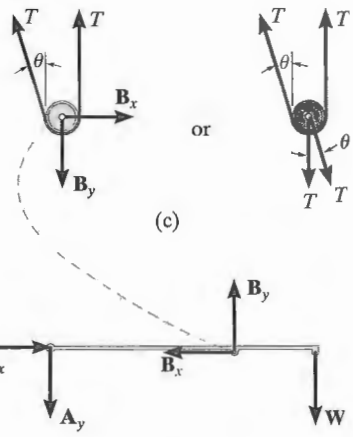
Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 6.10

A constant tension in the conveyor belt is maintained by using the device shown in Fig. 6–22*a*. Draw the free body diagrams of the frame and the cylinder that the belt surrounds. The suspended block has a weight of W.



(b)



(c)

(d)

(a)

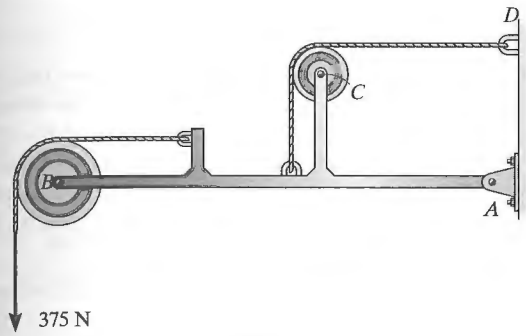
Fig. 6–22

SOLUTION

The idealized model of the device is shown in Fig. 6–22*b*. Here the angle θ is assumed to be known. From this model, the free-body diagrams of the cylinder and frame are shown in Figs. 6–22*c* and 6–22*d*, respectively. Note that the force that the pin at B exerts on the cylinder can be represented by either its horizontal and vertical components $\mathbf{B}_x$ and $\mathbf{B}_y$, which can be determined by using the force equations of equilibrium applied to the cylinder, or by the two components T, which provide equal but opposite moments on the cylinder and thus keep it from turning. Also, realize that once the pin reactions at A have been determined, half of their values act on each side of the frame since pin connections occur on each side, Fig. 6–22*a*.

EXAMPLE 6.11

For the frame shown in Fig. 6–23*a*, draw the free-body diagrams of (a) the entire frame including the pulleys and cords, (b) the frame without the pulleys and cords, and (c) each of the pulleys.



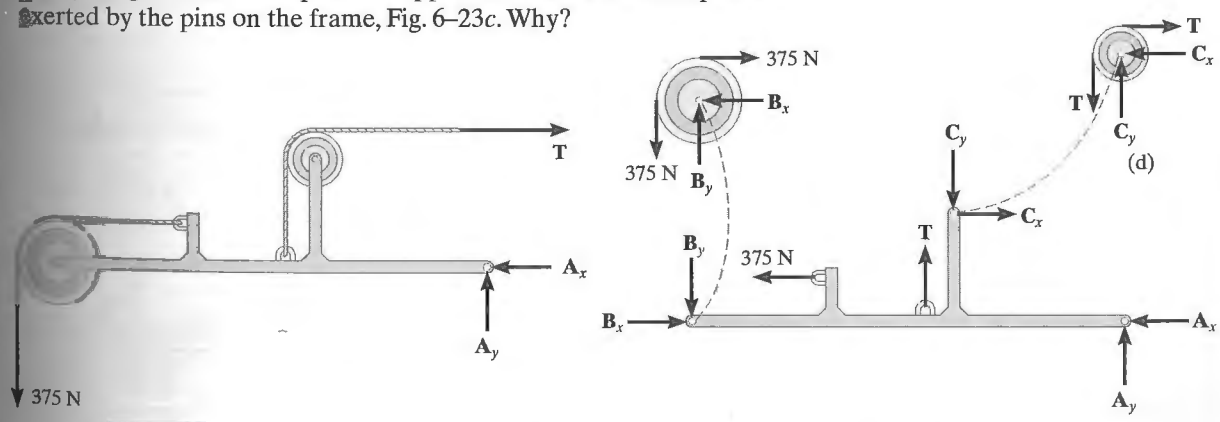
(a)

SOLUTION

Part (a). When the entire frame including the pulleys and cords is considered, the interactions at the points where the pulleys and cords are connected to the frame become pairs of *internal* forces which cancel each other and therefore are not shown on the free-body diagram, Fig. 6–23*b*.

Part (b). When the cords and pulleys are removed, their effect *on the frame* must be shown, Fig. 6–23*c*.

Part (c). The force components $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{C}_x$, $\mathbf{C}_y$ of the pins on the pulleys, Fig. 6–23*d*, are equal but opposite to the force components exerted by the pins on the frame, Fig. 6–23*c*. Why?

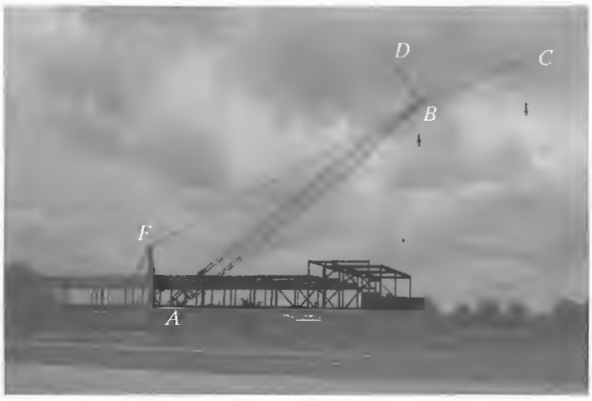


(b) (c)

Fig. 6–23

CONCEPTUAL PROBLEMS

P6–1. Draw the free-body diagrams of each of the crane boom segments AB, BC, and BD. Only the weights of AB and BC are significant. Assume A and B are pins.



P6–1

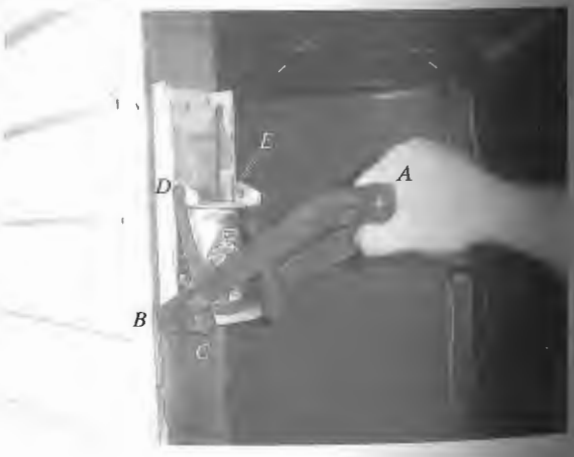
P6–2. Draw the free-body diagrams of the boom $ABCD$ and the stick $EDFGH$ of the backhoe. The weights of these two members are significant. Neglect the weights of all the other members, and assume all indicated points of connection are pins.

P6–2

P6–3. Draw the free-body diagrams of the boom $ABCDF$ and the stick FGH of the bucket lift. Neglect the weights of the member. The bucket weighs W. The two force members are BI, CE, DE and GE. Assume all indicated points of connection are pins.

P6–3

P6–4. To operate the can crusher, one pushes down on the lever arm ABC which rotates about the fixed pin at B. This moves the side links CD downward, which causes the guide plate E to also move downward and thereby crush the can. Draw the free-body diagrams of the lever, side link, and guide plate. Make up some reasonable numbers and do an equilibrium analysis to show how much an applied vertical force at the handle is magnified when it is transmitted to the can. Assume all points of connection are pins and the guides for the plate are smooth.



P6–4

Procedure for Analysis

The joint reactions on frames or machines (structures) composed of multiforce members can be determined using the following procedure.

Free-Body Diagram.

- Draw the free-body diagram of the entire frame or machine, a portion of it, or each of its members. The choice should be made so that it leads to the most direct solution of the problem.
- When the free-body diagram of a group of members of a frame or machine is drawn, the forces between the connected parts of this group are internal forces and are not shown on the free-body diagram of the group.
- Forces common to two members, which are in contact, act with equal magnitude but opposite sense on the respective free-body diagrams of the members.
- Two-force members, regardless of their shape, have equal but opposite collinear forces acting at the ends of the member.
- In many cases it is possible to tell by inspection the proper sense of the unknown forces acting on a member; however, if this seems difficult, the sense can be assumed.
- Remember that a couple moment is a free vector and can act at any point on the free-body diagram. Also, a force is a sliding vector and can act at any point along its line of action.

Equations of Equilibrium.

- Count the number of unknowns and compare it to the total number of equilibrium equations that are available. In two dimensions, there are three equilibrium equations that can be written for each member.
- Sum moments about a point that lies at the intersection of the lines of action of as many of the unknown forces as possible.
- If the solution of a force or couple moment magnitude is found to be negative, it means the sense of the force is the reverse of that shown on the free-body diagram.

EXAMPLE 6.14

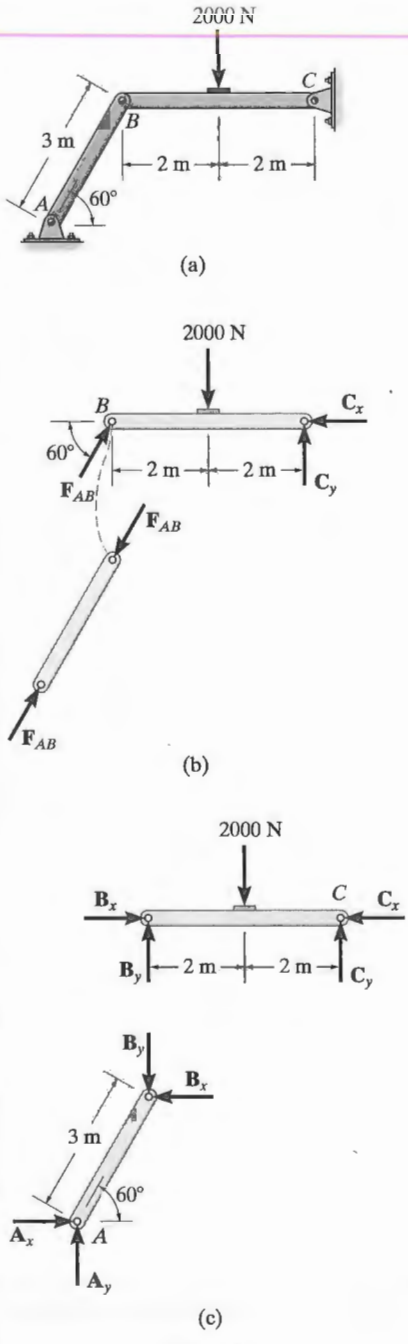


Fig. 6–26

Determine the horizontal and vertical components of force which the pin at C exerts on member BC of the frame in Fig. 6–26a.

SOLUTION I

Free-Body Diagrams. By inspection, it can be seen that AB is a two-force member. The free-body diagrams are shown in Fig. 6–26b.

Equations of Equilibrium. The *three unknowns* can be determined by applying the three equations of equilibrium to member CB.

$$\circlearrowleft + \Sigma M_C = 0;\ 2000\text{ N}(2\text{ m}) - (F_{AB}\sin 60°)(4\text{ m}) = 0 \quad F_{AB} = 1154.7\text{ N}$$

$$\xrightarrow{+} \Sigma F_x = 0;\ 1154.7\cos 60°\text{ N} - C_x = 0 \qquad C_x = 577\text{ N} \qquad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0;\ 1154.7\sin 60°\text{ N} - 2000\text{ N} + C_y = 0 \quad C_y = 1000\text{ N} \quad \textit{Ans.}$$

SOLUTION II

Free-Body Diagrams. If one does not recognize that AB is a two-force member, then more work is involved in solving this problem. The free-body diagrams are shown in Fig. 6–26c.

Equations of Equilibrium. The *six unknowns* are determined by applying the three equations of equilibrium to each member.

Member AB

$$\circlearrowleft + \Sigma M_A = 0;\ B_x(3\sin 60°\text{ m}) - B_y(3\cos 60°\text{ m}) = 0 \qquad (1)$$

$$\xrightarrow{+} \Sigma F_x = 0;\ A_x - B_x = 0 \qquad (2)$$

$$+\uparrow \Sigma F_y = 0;\ A_y - B_y = 0 \qquad (3)$$

Member BC

$$\circlearrowleft + \Sigma M_C = 0;\ 2000\text{ N}(2\text{ m}) - B_y(4\text{ m}) = 0 \qquad (4)$$

$$\xrightarrow{+} \Sigma F_x = 0;\ B_x - C_x = 0 \qquad (5)$$

$$+\uparrow \Sigma F_y = 0;\ B_y - 2000\text{ N} + C_y = 0 \qquad (6)$$

The results for C_x and C_y can be determined by solving these equations in the following sequence: 4, 1, 5, then 6. The results are

$$B_y = 1000\text{ N}$$

$$B_x = 577\text{ N}$$

$$C_x = 577\text{ N} \qquad \textit{Ans.}$$

$$C_y = 1000\text{ N} \qquad \textit{Ans.}$$

By comparison, Solution I is simpler since the requirement that F_{AB} in Fig. 6–26b be equal, opposite, and collinear at the ends of member AB automatically satisfies Eqs. 1, 2, and 3 above and therefore eliminates the need to write these equations. *As a result, save yourself some time and effort by always identifying the two-force members before starting the analysis!*

EXAMPLE 6.15

The compound beam shown in Fig. 6–27*a* is pin connected at *B*. Determine the components of reaction at its supports. Neglect its weight and thickness.

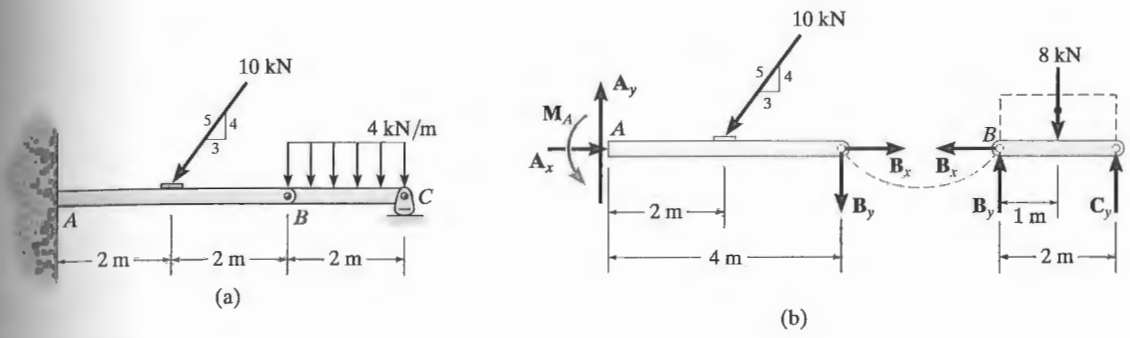


Fig. 6–27

SOLUTION

Free-Body Diagrams. By inspection, if we consider a free-body diagram of the *entire beam ABC*, there will be three unknown reactions at *A* and one at *C*. These four unknowns cannot all be obtained from the three available equations of equilibrium, and so for the solution it will become necessary to dismember the beam into its two segments, as shown in Fig. 6–27*b*.

Equations of Equilibrium. The six unknowns are determined as follows:

Segment BC

$$\overset{+}{\rightarrow}\Sigma F_x = 0; \qquad B_x = 0$$

$$\circlearrowleft + \Sigma M_B = 0; \qquad -8\text{ kN}(1\text{ m}) + C_y(2\text{ m}) = 0$$

$$+\uparrow \Sigma F_y = 0; \qquad B_y - 8\text{ kN} + C_y = 0$$

Segment AB

$$\overset{+}{\rightarrow}\Sigma F_x = 0; \qquad A_x - (10\text{ kN})\left(\tfrac{3}{5}\right) + B_x = 0$$

$$\circlearrowleft + \Sigma M_A = 0; \qquad M_A - (10\text{ kN})\left(\tfrac{4}{5}\right)(2\text{ m}) - B_y(4\text{ m}) = 0$$

$$+\uparrow \Sigma F_y = 0; \qquad A_y - (10\text{ kN})\left(\tfrac{4}{5}\right) - B_y = 0$$

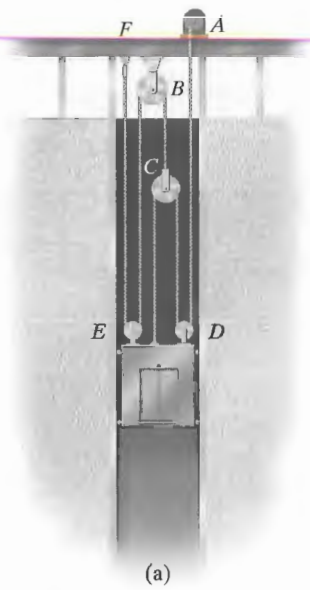
Solving each of these equations successively, using previously calculated results, we obtain

$A_x = 6\text{ kN}$ $\quad A_y = 12\text{ kN}$ $\quad M_A = 32\text{ kN}\cdot\text{m}$ *Ans.*

$B_x = 0$ $\quad B_y = 4\text{ kN}$

$C_y = 4\text{ kN}$ *Ans.*

EXAMPLE 6.16



A 500-kg elevator car in Fig. 6–28*a* is being hoisted by motor *A* using the pulley system shown. If the car is traveling with a constant speed, determine the force developed in the two cables. Neglect the mass of the cable and pulleys.

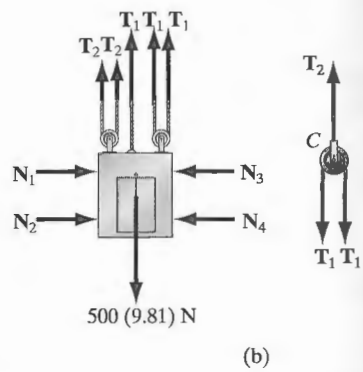


Fig. 6–28

SOLUTION

Free-Body Diagram. We can solve this problem using the free-body diagrams of the elevator car and pulley *C*, Fig. 6–28*b*. The tensile forces developed in the cables are denoted as T_1 and T_2.

Equations of Equilibrium. For pulley *C*,

$$+\uparrow \Sigma F_y = 0; \qquad T_2 - 2T_1 = 0 \qquad \text{or} \qquad T_2 = 2T_1 \tag{1}$$

For the elevator car,

$$+\uparrow \Sigma F_y = 0; \qquad 3T_1 + 2T_2 - 500(9.81)\text{ N} = 0 \tag{2}$$

Substituting Eq. (1) into Eq. (2) yields

$$3T_1 + 2(2T_1) - 500(9.81)\text{ N} = 0$$

$$T_1 = 700.71\text{ N} = 701\text{ N} \qquad \textit{Ans.}$$

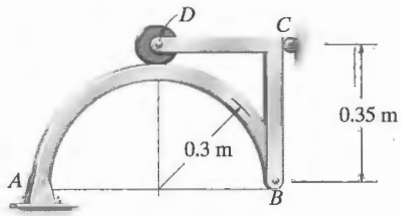
Substituting this result into Eq. (1),

$$T_2 = 2(700.71)\text{ N} = 1401\text{ N} = 1.40\text{ kN} \qquad \textit{Ans.}$$

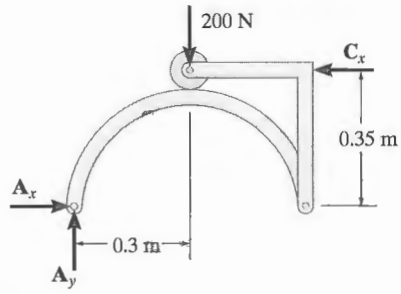
Custom Publishing exclusively for The University of Melbourne

EXAMPLE 6.17

The smooth disk shown in Fig. 6–29*a* is pinned at D and has a weight of 200 N. Neglecting the weights of the other members, determine the horizontal and vertical components of reaction at pins B and D.



(a)



SOLUTION

Free-Body Diagrams. The free-body diagrams of the entire frame and each of its members are shown in Fig. 6–29*b*.

Equations of Equilibrium. The eight unknowns can of course be obtained by applying the eight equilibrium equations to each member—three to member AB, three to member BCD, and two to the disk. (Moment equilibrium is automatically satisfied for the disk.) If this is done, however, all the results can be obtained only from a simultaneous solution of some of the equations. (Try it and find out.) To avoid this situation, it is best first to determine the three support reactions on the *entire* frame; then, using these results, the remaining five equilibrium equations can be applied to two other parts in order to solve successively for the other unknowns.

Entire Frame

$$\circlearrowleft + \Sigma M_A = 0; \quad -200\text{ N}(0.3\text{ m}) + C_x(0.35\text{ m}) = 0 \quad C_x = 171\text{ N}$$

$$\xrightarrow{+} \Sigma F_x = 0; \quad A_x - 171\text{ N} = 0 \quad A_x = 171\text{ N}$$

$$+\uparrow \Sigma F_y = 0; \quad A_y - 200\text{ N} = 0 \quad A_y = 200\text{ N}$$

Member AB

$$\xrightarrow{+} \Sigma F_x = 0; \quad 171\text{ N} - B_x = 0 \quad B_x = 171\text{ N} \quad \textit{Ans.}$$

$$\circlearrowleft + \Sigma M_B = 0; \quad -200\text{ N}(0.6\text{ m}) + N_D(0.3\text{ m}) = 0 \quad N_D = 400\text{ N}$$

$$+\uparrow \Sigma F_y = 0; \quad 200\text{ N} - 400\text{ N} + B_y = 0 \quad B_y = 200\text{ N} \quad \textit{Ans.}$$

Disk

$$\xrightarrow{+} \Sigma F_x = 0; \quad D_x = 0 \quad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \quad 400\text{ N} - 200\text{ N} - D_y = 0 \quad D_y = 200\text{ N} \quad \textit{Ans.}$$

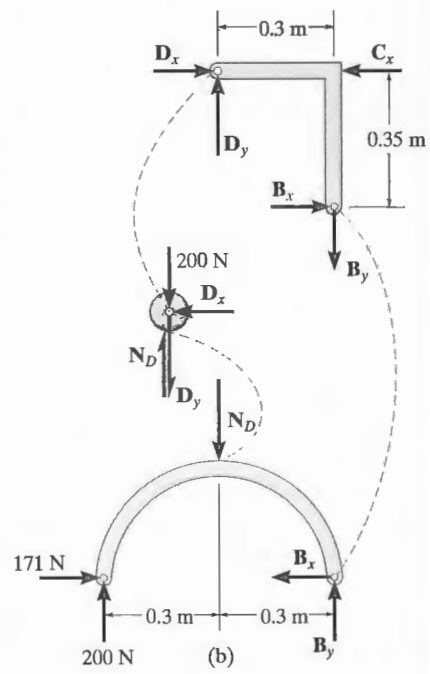


Fig. 6–29

EXAMPLE 6.18

Determine the tension in the cables and also the force **P** required to support the 600-N force using the frictionless pulley system shown in Fig. 6–30*a*.

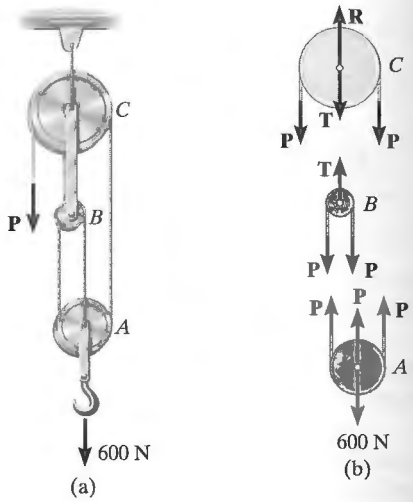


Fig. 6–30

6

SOLUTION

Free-Body Diagram. A free-body diagram of each pulley *including* its pin and a portion of the contacting cable is shown in Fig. 6–30*b*. Since the cable is *continuous*, it has a *constant tension P* acting throughout its length. The link connection between pulleys *B* and *C* is a two-force member, and therefore it has an unknown tension *T* acting on it. Notice that the *principle of action, equal but opposite reaction* must be carefully observed for forces **P** and **T** when the *separate* free-body diagrams are drawn.

Equations of Equilibrium. The three unknowns are obtained as follows:

Pulley A

$+\uparrow \Sigma F_y = 0; \qquad 3P - 600\text{ N} = 0 \qquad P = 200\text{ N}$ *Ans.*

Pulley B

$+\uparrow \Sigma F_y = 0; \qquad T - 2P = 0 \qquad T = 400\text{ N}$ *Ans.*

Pulley C

$+\uparrow \Sigma F_y = 0; \qquad R - 2P - T = 0 \qquad R = 800\text{ N}$ *Ans.*

EXAMPLE 6.19

The two planks in Fig. 6–31*a* are connected together by cable *BC* and a smooth spacer *DE*. Determine the reactions at the smooth supports *A* and *F*, and also find the force developed in the cable and spacer.

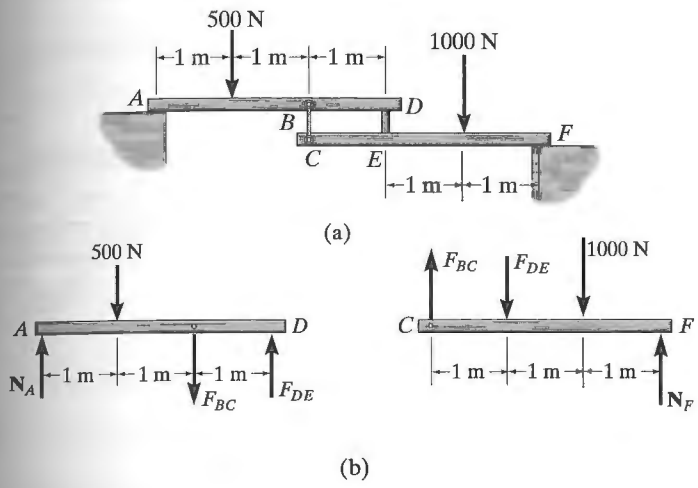


Fig. 6–31

SOLUTION

Free-Body Diagrams. The free-body diagram of each plank is shown in Fig. 6–31*b*. It is important to apply Newton's third law to the interaction forces as shown.

Equations of Equilibrium. For plank *AD*,

$$\circlearrowleft + \Sigma M_A = 0; \quad F_{DE}(3\text{ m}) - F_{BC}(2\text{ m}) - 500\text{ N}(1\text{ m}) = 0$$

For plank *CF*,

$$\circlearrowleft + \Sigma M_F = 0; \quad F_{DE}(2\text{ m}) - F_{BC}(3\text{ m}) + 1000\text{ N}(1\text{ m}) = 0$$

Solving simultaneously,

$$F_{DE} = 700\text{ N} \quad F_{BC} = 800\text{ N} \qquad \textit{Ans.}$$

Using these results, for plank *AD*,

$$+\uparrow \Sigma F_y = 0; \quad N_A + 700\text{ N} - 800\text{ N} - 500\text{ N} = 0$$

$$N_A = 600\text{ N} \qquad \textit{Ans.}$$

And for plank *CF*,

$$+\uparrow \Sigma F_y = 0; \quad N_F + 800\text{ N} - 700\text{ N} - 1000\text{ N} = 0$$

$$N_F = 900\text{ N} \qquad \textit{Ans.}$$

EXAMPLE 6.20

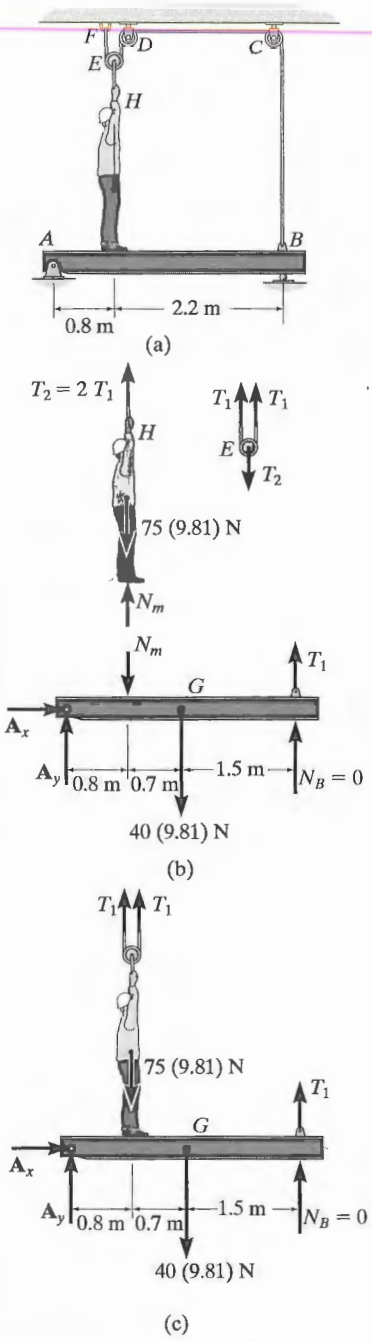


Fig. 6–32

The 75-kg man in Fig. 6–32*a* attempts to lift the 40-kg uniform beam off the roller support at B. Determine the tension developed in the cable attached to B and the normal reaction of the man on the beam when this is about to occur.

SOLUTION

Free-Body Diagrams. The tensile force in the cable will be denoted as T_1. The free-body diagrams of the pulley E, the man, and the beam are shown in Fig. 6–32*b*. The beam has no contact with roller B, so $N_B = 0$. When drawing each of these diagrams, it is very important to apply Newton's third law.

Equations of Equilibrium. Using the free-body diagram of pulley E,

$$+\uparrow \Sigma F_y = 0; \qquad 2T_1 - T_2 = 0 \qquad \text{or} \qquad T_2 = 2T_1 \tag{1}$$

Referring to the free-body diagram of the man using this result,

$$+\uparrow \Sigma F_y = 0; \qquad N_m + 2T_1 - 75(9.81)\text{ N} = 0 \tag{2}$$

Summing moments about point A on the beam,

$$\circlearrowleft + \Sigma M_A = 0; \quad T_1(3\text{ m}) - N_m(0.8\text{ m}) - [40(9.81)\text{ N}](1.5\text{ m}) = 0 \tag{3}$$

Solving Eqs. 2 and 3 simultaneously for T_1 and N_m, then using Eq. (1) for T_2, we obtain

$$T_1 = 256\text{ N} \qquad N_m = 224\text{ N} \qquad T_2 = 512\text{ N} \qquad \textit{Ans.}$$

SOLUTION II

A direct solution for T_1 can be obtained by considering the beam, the man, and pulley E as a *single system*. The free-body diagram is shown in Fig. 6–32*c*. Thus,

$$\circlearrowleft + \Sigma M_A = 0; \quad 2T_1(0.8\text{ m}) - [75(9.81)\text{ N}](0.8\text{ m})$$
$$-[40(9.81)\text{ N}](1.5\text{ m}) + T_1(3\text{ m}) = 0$$
$$T_1 = 256\text{ N} \qquad \textit{Ans.}$$

With this result, Eqs. 1 and 2 can then be used to find N_m and T_2.

EXAMPLE 6.21

The frame in Fig. 6–33*a* supports the 50-kg cylinder. Determine the horizontal and vertical components of reaction at A and the force at C.

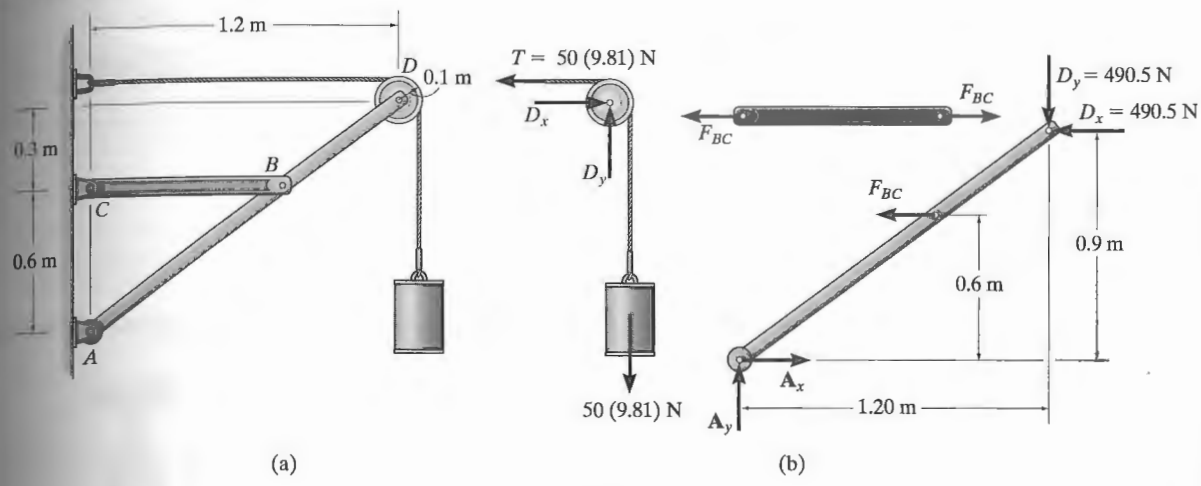


Fig. 6–33

6

SOLUTION

Free-Body Diagrams. The free-body diagram of pulley D, along with the cylinder and a portion of the cord (a system), is shown in Fig. 6–33*b*. Member BC is a two-force member as indicated by its free-body diagram. The free-body diagram of member ABD is also shown.

Equations of Equilibrium. We will begin by analyzing the equilibrium of the pulley. The moment equation of equilibrium is automatically satisfied with $T = 50(9.81)$ N, and so

$$\xrightarrow{+} \Sigma F_x = 0; \quad D_x - 50(9.81)\text{ N} = 0 \quad D_x = 490.5\text{ N}$$

$$+\uparrow \Sigma F_y = 0; \quad D_y - 50(9.81)\text{ N} = 0 \quad D_y = 490.5\text{ N} \qquad \textit{Ans.}$$

Using these results, F_{BC} can be determined by summing moments about point A on member ABD.

$$\circlearrowleft + \Sigma M_A = 0; \; F_{BC}(0.6\text{ m}) + 490.5\text{ N}(0.9\text{ m}) - 490.5\text{ N}(1.20\text{ m}) = 0$$

$$F_{BC} = 245.25\text{ N} \qquad \textit{Ans.}$$

Now A_x and A_y can be determined by summing forces.

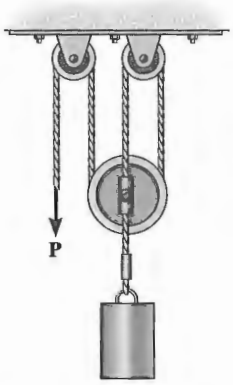
$$\xrightarrow{+} \Sigma F_x = 0; \quad A_x - 245.25\text{ N} - 490.5\text{ N} = 0 \quad A_x = 736\text{ N} \qquad \textit{Ans.}$$

$$+\uparrow \Sigma F_y = 0; \quad A_y - 490.5\text{ N} = 0 \quad A_y = 490.5\text{ N} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

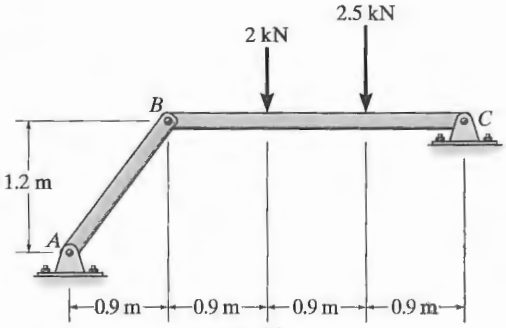
FUNDAMENTAL PROBLEMS

F6–13. Determine the force P needed to hold the 300-N weight in equilibrium.



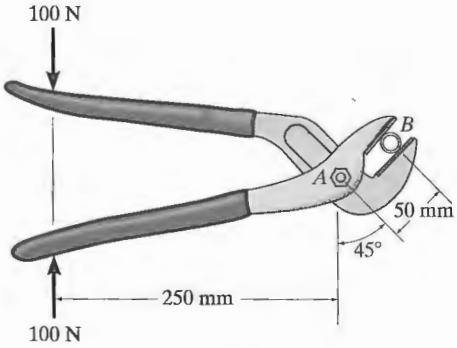
F6–13

F6–14. Determine the horizontal and vertical components of reaction at pin C.



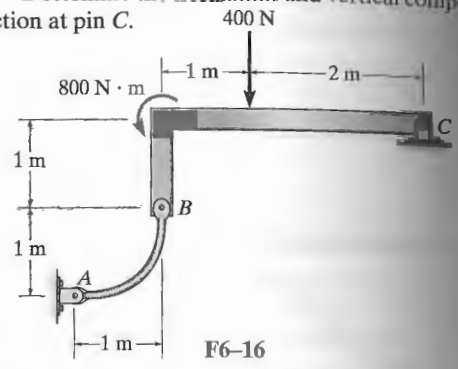
F6–14

F6–15. If a 100-N force is applied to the handles of the pliers, determine the clamping force exerted on the smooth pipe B and the magnitude of the resultant force at pin A.



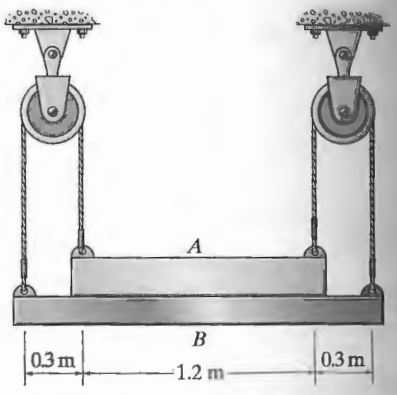
F6–15

F6–16. Determine the horizontal and vertical components of reaction at pin C.



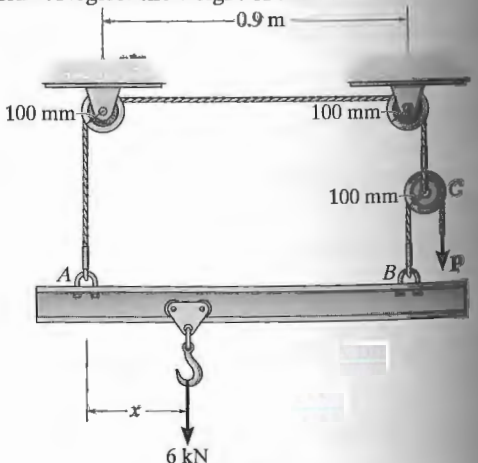
F6–16

F6–17. Determine the normal force that the 500-N plate A exerts on the 150-N plate B.



F6–17

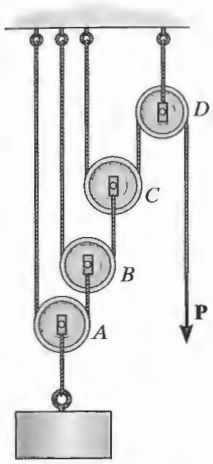
F6–18. Determine the force P needed to lift the load. Also, determine the proper placement x of the hook for equilibrium. Neglect the weight of the beam.



F6–18

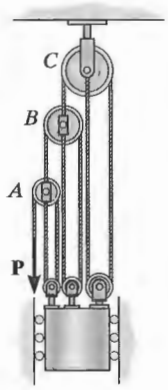
PROBLEMS

6–67. Determine the force **P** required to hold the 100-kg weight in equilibrium.



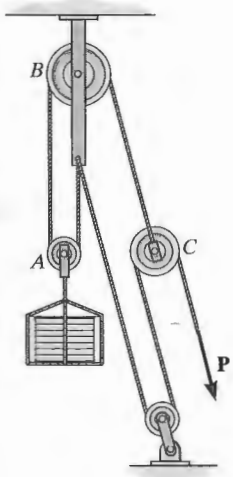
Prob. 6–67

•6–69. Determine the force **P** required to hold the 50-kg mass in equilibrium.



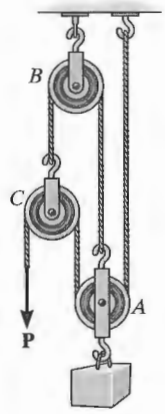
Prob. 6–69

***6–68.** Determine the force **P** required to hold the 150-kg crate in equilibrium.



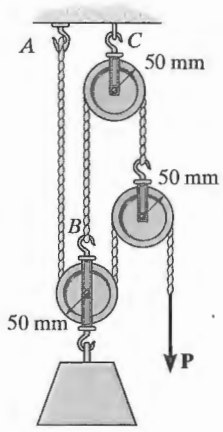
Prob. 6–68

6–70. Determine the force **P** needed to hold the 10-kg block in equilibrium.



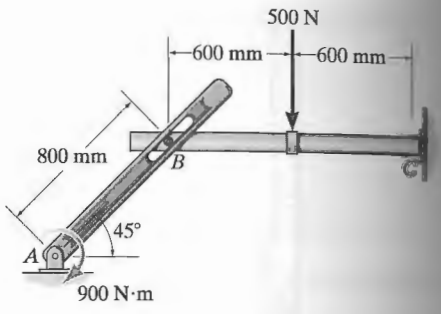
Prob. 6–70

6–71. Determine the force **P** needed to support the 50-kg weight. Each pulley has a weight of 50 N. Also, what are the cord reactions at A and B?



Prob. 6–71

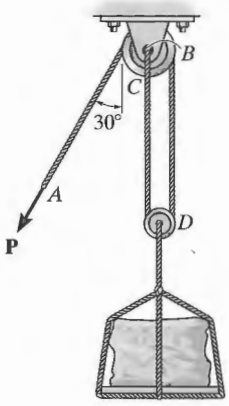
•6–73. If the peg at B is smooth, determine the components of reaction at pin A and fixed support C.



Prob. 6–73

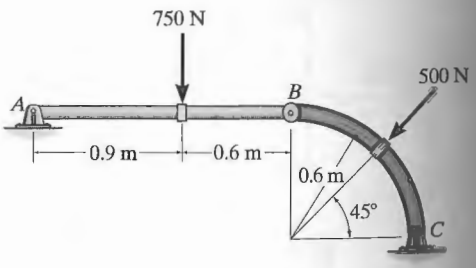
6

***6–72.** The cable and pulleys are used to lift the 300-kg stone. Determine the force that must be exerted on the cable at A and the corresponding magnitude of the resultant force the pulley at C exerts on pin B when the cables are in the position shown.



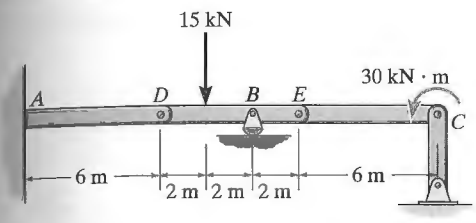
Prob. 6–72

6–74. Determine the horizontal and vertical components of reaction at pins A and C.



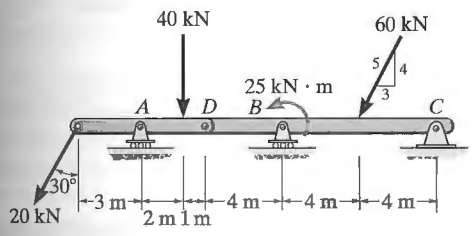
Prob. 6–74

6–75. The compound beam is fixed at A and supported by rockers at B and C. There are hinges (pins) at D and E. Determine the components of reaction at the supports.



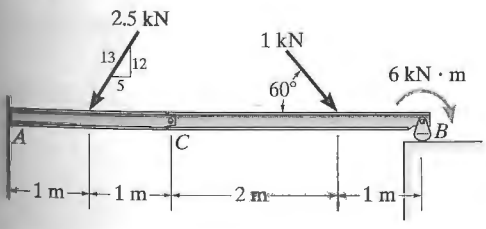
Prob. 6–75

***6–76.** The compound beam is pin-supported at C and supported by rollers at A and B. There is a hinge (pin) at D. Determine the components of reaction at the supports. Neglect the thickness of the beam.



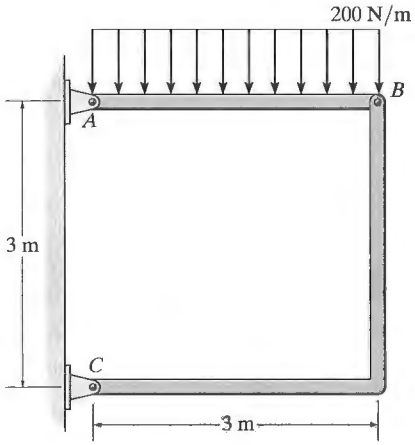
Prob. 6–76

6–77. The compound beam is supported by a rocker at B and is fixed to the wall at A. If it is hinged (pinned) together at C, determine the components of reaction at the supports. Neglect the thickness of the beam.



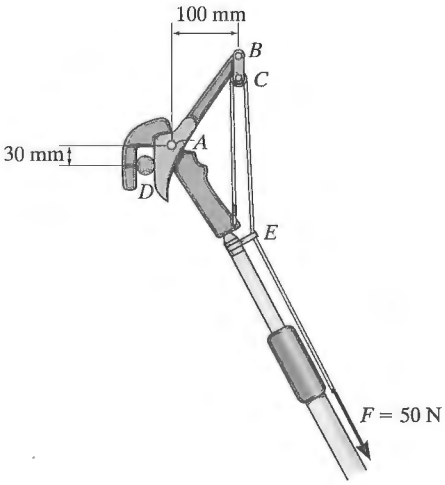
Prob. 6–77

6–78. Determine the horizontal and vertical components of reaction at pins A and C of the two-member frame.



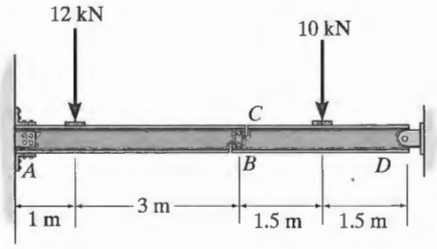
Prob. 6–78

6–79. If a force of $F = 50$ N acts on the rope, determine the cutting force on the smooth tree limb at D and the horizontal and vertical components of force acting on pin A. The rope passes through a small pulley at C and a smooth ring at E.



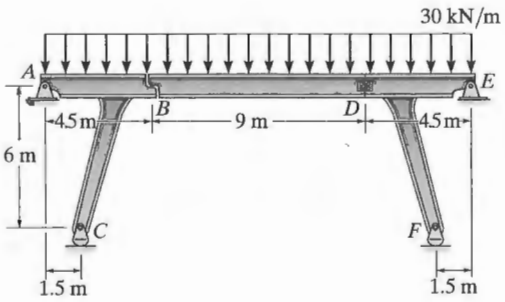
Prob. 6–79

***6–80.** Two beams are connected together by the short link BC. Determine the components of reaction at the fixed support A and at pin D.



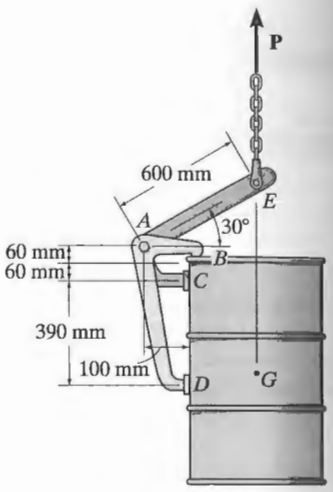
Prob. 6–80

•6–81. The bridge frame consists of three segments which can be considered pinned at A, D, and E, rocker supported at C and F, and roller supported at B. Determine the horizontal and vertical components of reaction at all these supports due to the loading shown.



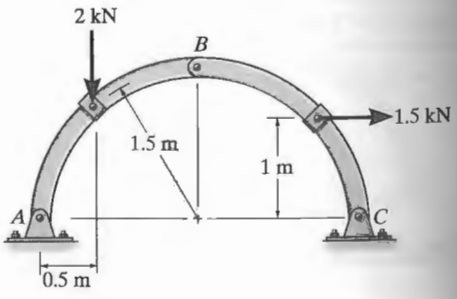
Prob. 6–81

6–82. If the 300-kg drum has a center of mass at point G, determine the horizontal and vertical component[redacted] of force acting at pin A and the reactions on the smooth pads C and D. The grip at B on member DAB resists both horizontal and vertical components of force at the rim of the drum.



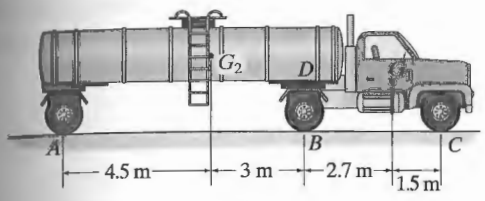
Prob. 6–82

6–83. Determine the horizontal and vertical compone[redacted] of reaction that pins A and C exert on the two-memb[redacted] arch.



Prob. 6–83

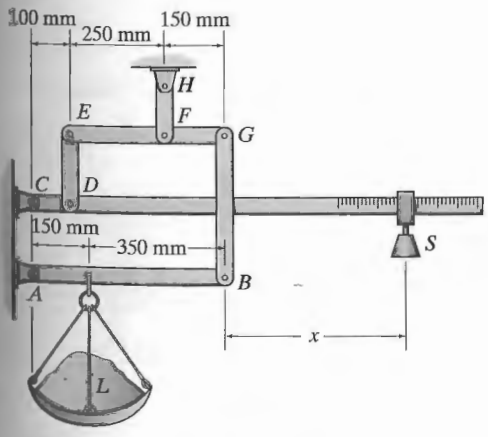
*6–84. The truck and the tanker have weights of 40 kN and 100 kN respectively. Their respective centers of gravity are located at points G_1 and G_2. If the truck is at rest, determine the reactions on both wheels at A, at B, and at C. The tanker is connected to the truck at the turntable D which acts as a pin.



Prob. 6–84

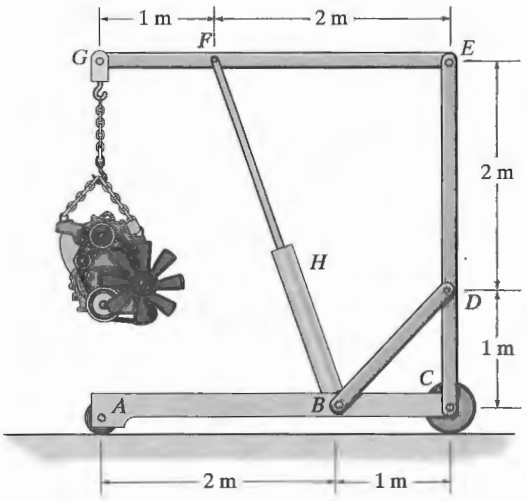
•6–85. The platform scale consists of a combination of third and first class levers so that the load on one lever becomes the effort that moves the next lever. Through this arrangement, a small weight can balance a massive object. If $x = 450$ mm, determine the required mass of the counterweight S required to balance a 90-kg load, L.

6–86. The platform scale consists of a combination of third and first class levers so that the load on one lever becomes the effort that moves the next lever. Through this arrangement, a small weight can balance a massive object. If $x = 450$ mm and, the mass of the counterweight S is 2 kg, determine the mass of the load L required to maintain the balance.



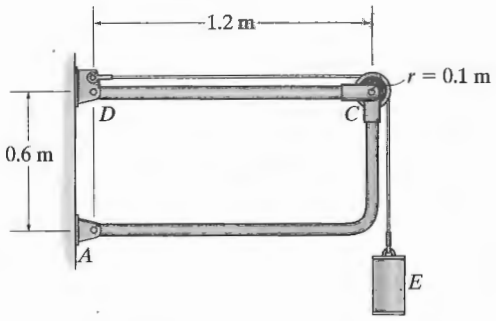
Probs. 6–85/86

6–87. The hoist supports the 125-kg engine. Determine the force the load creates in member DB and in member FB, which contains the hydraulic cylinder H.



Prob. 6–87

*6–88. The frame is used to support the 100-kg cylinder E. Determine the horizontal and vertical components of reaction at A and D.



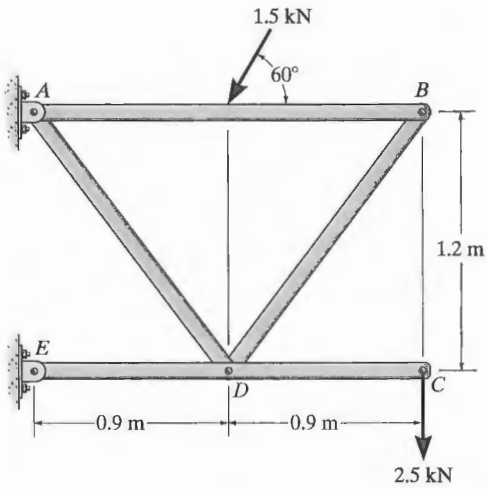
Prob. 6–88

ENGR10003 – ENGINEERING SYSTEMS DESIGN
A publication from Pearson Custom Publishing exclusively for The University of Adelaide

•6–89. Determine the horizontal and vertical components of reaction which the pins exert on member AB of the frame.

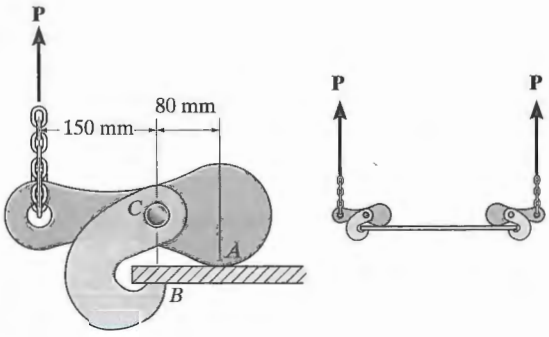
Video Solutions

6–90. Determine the horizontal and vertical components of reaction which the pins exert on member EDC of the frame.



Probs. 6–89/90

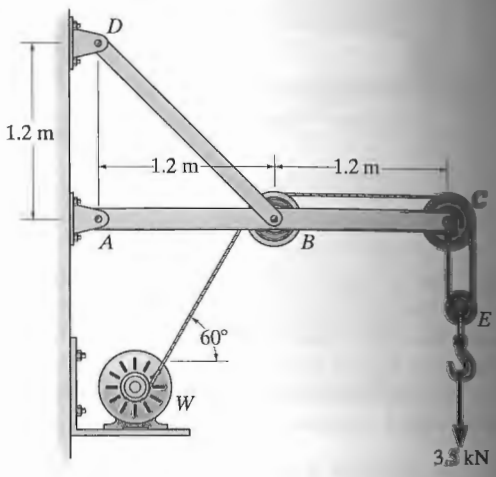
6–91. The clamping hooks are used to lift the uniform smooth 500-kg plate. Determine the resultant compressive force that the hook exerts on the plate at A and B, and the pin reaction at C.



Prob. 6–91

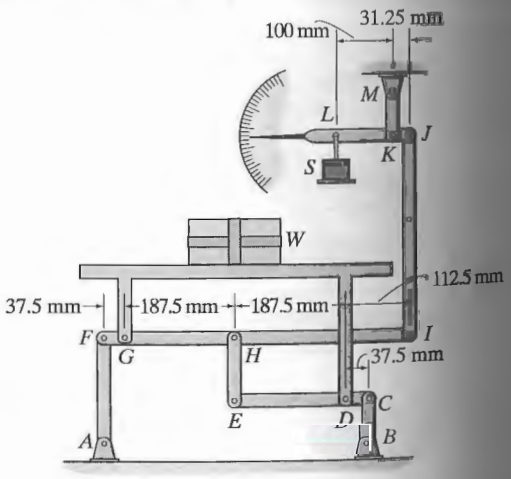
***6–92.** The wall crane supports a load of 3.5 kN. Determine the horizontal and vertical components of reaction at pins A and D. Also, what is the force in the cable at the winch W?

•6–93. The wall crane supports a load of 3.5 kN. Determine the horizontal and vertical components of reaction at pins A and D. Also, what is the force in the cable at the winch W? The jib ABC has a weight of 500 N and member BD has a weight of 200 N. Each member is uniform and has a center of gravity at its center.



Probs. 6–92/93

6–94. The lever-actuated scale consists of a series of compound levers. If a load of weight $W = 750$ N is placed on the platform, determine the required weight of the counterweight S to balance the load. Is it necessary to place the load symmetrically on the platform? Explain.

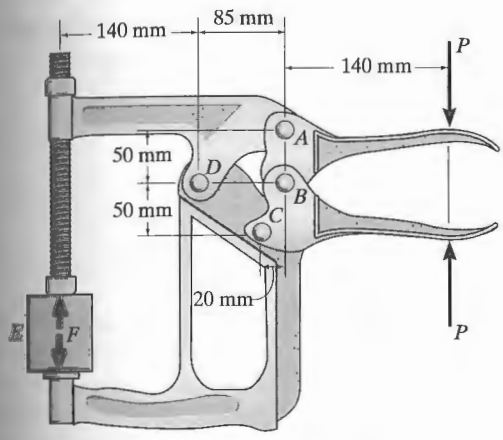


Prob. 6–94

6–95. If $P = 75$ N, determine the force F that the toggle clamp exerts on the wooden block.

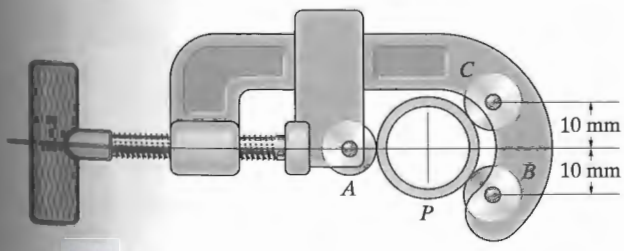
Video Solutions

6–96. If the wooden block exerts a force of $F = 600$ N on the toggle clamp, determine the force P applied to the handle.



Probs. 6–95/96

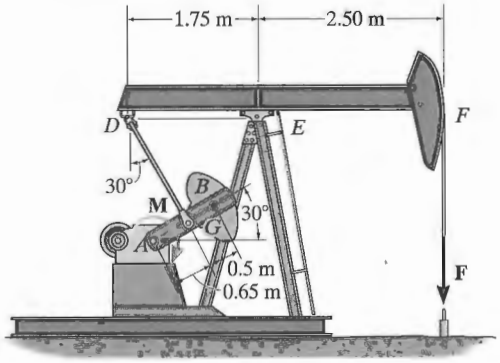
6–97. The pipe cutter is clamped around the pipe P. If the wheel at A exerts a normal force of $F_A = 80$ N on the pipe, determine the normal forces of wheels B and C on the pipe. The three wheels each have a radius of 7 mm and the pipe has an outer radius of 10 mm.



Prob. 6–97

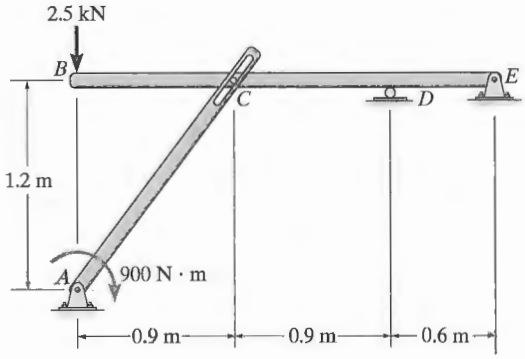
6–98. A 300-kg counterweight, with center of mass at G, is mounted on the pitman crank AB of the oil-pumping unit. If a force of $F = 5$ kN is to be developed in the fixed cable attached to the end of the walking beam DEF, determine the torque M that must be supplied by the motor.

6–99. A 300-kg counterweight, with center of mass at G, is mounted on the pitman crank AB of the oil-pumping unit. If the motor supplies a torque of $M = 2500$ N · m, determine the force **F** developed in the fixed cable attached to the end of the walking beam DEF.



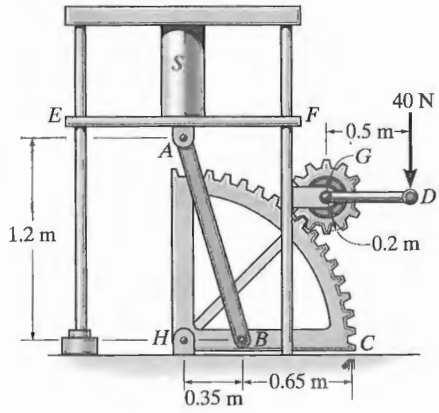
Probs. 6–98/99

***6–100.** The two-member structure is connected at C by a pin, which is fixed to BDE and passes through the smooth slot in member AC. Determine the horizontal and vertical components of reaction at the supports.



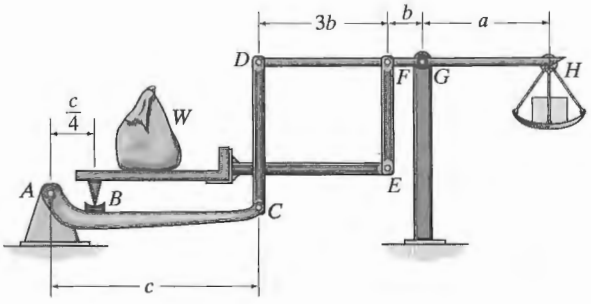
Prob. 6–100

***6–112.** The handle of the sector press is fixed to gear G, which in turn is in mesh with the sector gear C. Note that AB is pinned at its ends to gear C and the underside of the table EF, which is allowed to move vertically due to the smooth guides at E and F. If the gears only exert tangential forces between them, determine the compressive force developed on the cylinder S when a vertical force of 40 N is applied to the handle of the press.



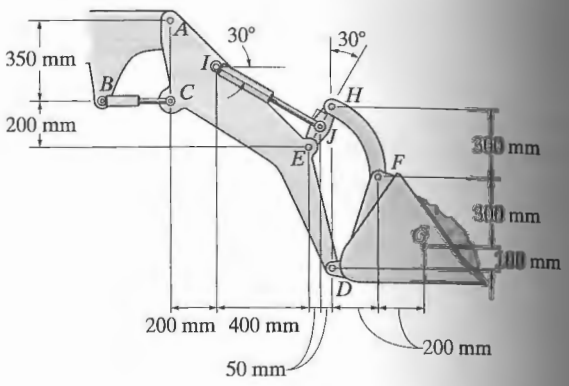
Prob. 6–112

•6–113. Show that the weight W_1 of the counterweight at H required for equilibrium is $W_1 = (b/a)W$, and so it is independent of the placement of the load W on the platform.



Prob. 6–113

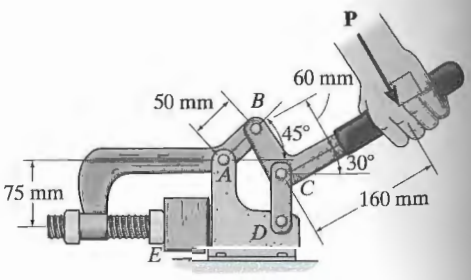
6–114. The tractor shovel carries a 500-kg load of soil, having a center of mass at G. Compute the forces developed in the hydraulic cylinders IJ and BC due to this loading.



Prob. 6–114

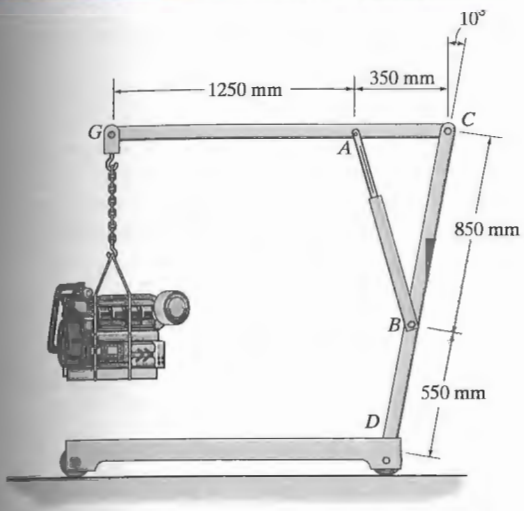
6–115. If a force of $P = 100$ N is applied to the handle of the toggle clamp, determine the horizontal clamping force N_E that the clamp exerts on the smooth wooden block at E.

***6–116.** If the horizontal clamping force that the toggle clamp exerts on the smooth wooden block at E is $N_E = 200$ N, determine the force **P** applied to the handle of the clamp.



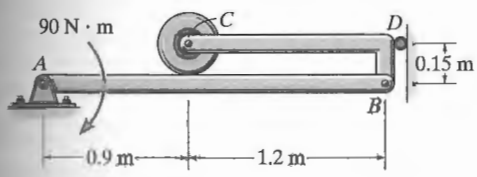
Probs. 6–115/116

6–117. The engine hoist is used to support the 200-kg engine. Determine the force acting in the hydraulic cylinder AB, the horizontal and vertical components of force at the pin C, and the reactions at the fixed support D.



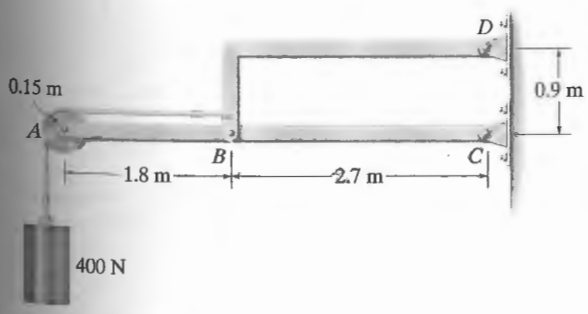
Prob. 6–117

6–118. Determine the force that the smooth roller C exerts on member AB. Also, what are the horizontal and vertical components of reaction at pin A? Neglect the weight of the frame and roller.



Prob. 6–118

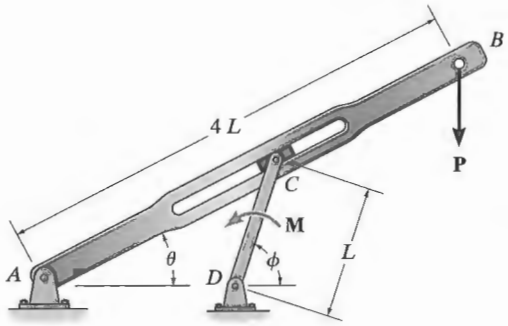
6–119. Determine the horizontal and vertical components of reaction which the pins exert on member ABC.



Prob. 6–119

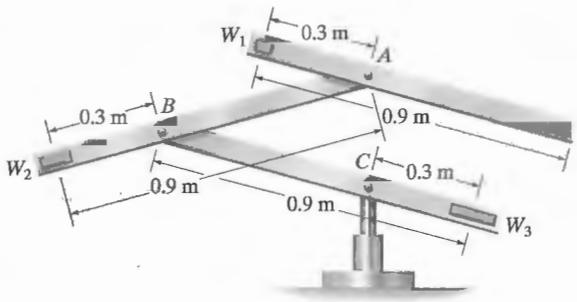
***6–120.** Determine the couple moment **M** that must be applied to member DC for equilibrium of the quick-return mechanism. Express the result in terms of the angles ϕ and θ, dimension L, and the applied *vertical force* **P**. The block at C is confined to slide within the slot of member AB.

•6–121. Determine the couple moment **M** that must be applied to member DC for equilibrium of the quick-return mechanism. Express the result in terms of the angles ϕ and θ, dimension L, and the applied force **P**, which should be changed in the figure and instead be directed horizontally to the right. The block at C is confined to slide within the slot of member AB.



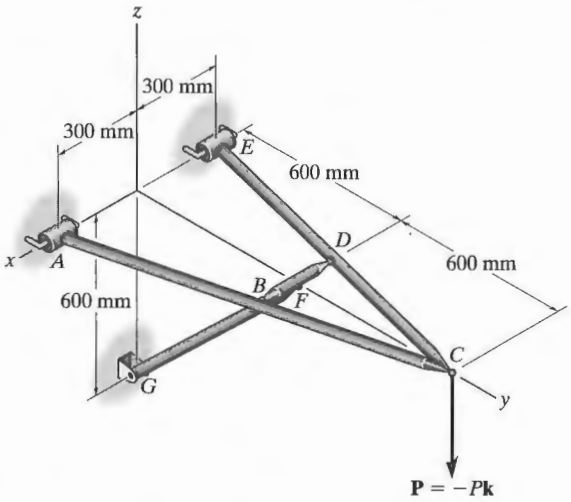
Probs. 6–120/121

6–122. The kinetic sculpture requires that each of the three pinned beams be in perfect balance at all times during its slow motion. If each member has a uniform weight of 50 N/m and length of 0.9 m, determine the necessary counterweights W_1, W_2, and W_3 which must be added to the ends of each member to keep the system in balance for any position. Neglect the size of the counterweights.



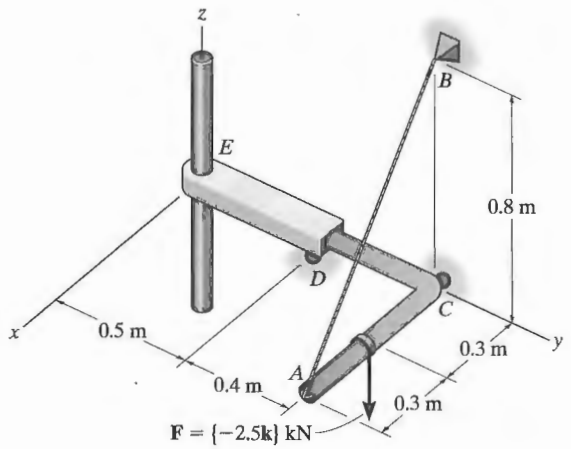
Prob. 6–122

6–123. The four-member "A" frame is supported at A and E by smooth collars and at G by a pin. All the other joints are ball-and-sockets. If the pin at G will fail when the resultant force there is 800 N, determine the largest vertical force P that can be supported by the frame. Also, what are the x, y, z force components which member BD exerts on members EDC and ABC? The collars at A and E and the pin at G only exert force components on the frame.



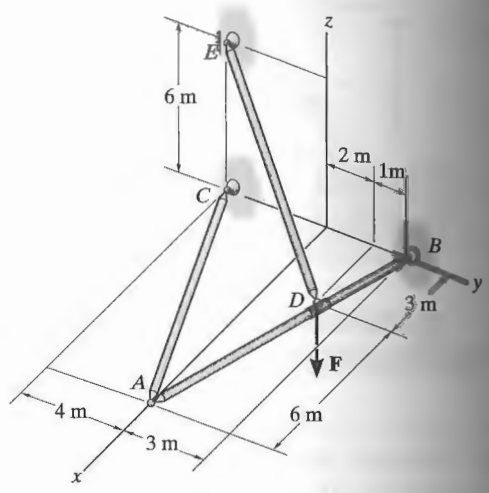
Prob. 6–123

***6–124.** The structure is subjected to the loading shown. Member AD is supported by a cable AB and roller at C and fits through a smooth circular hole at D. Member ED is supported by a roller at D and a pole that fits in a smooth snug circular hole at E. Determine the x, y, z components of reaction at E and the tension in cable AB.



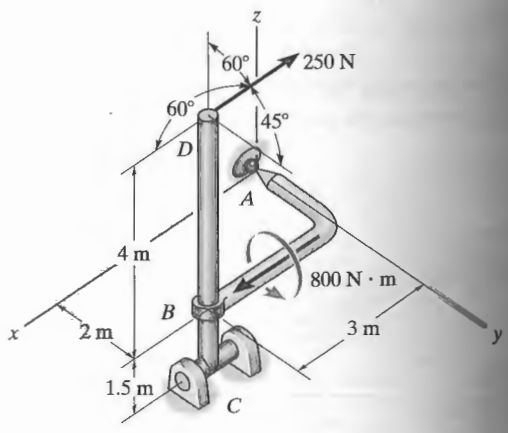
Prob. 6–124

•6–125. The three-member frame is connected at its ends using ball-and-socket joints. Determine the x, y, z components of reaction at B and the tension in member ED. The force acting at D is $\mathbf{F} = \{135\mathbf{i} + 200\mathbf{j} - 180\mathbf{k}\}$ kN.



Prob. 6–125

6–126. The structure is subjected to the loading shown. Member AB is supported by a ball-and-socket at A and smooth collar at B. Member CD is supported by a pin at C. Determine the x, y, z components of reaction at A and C.



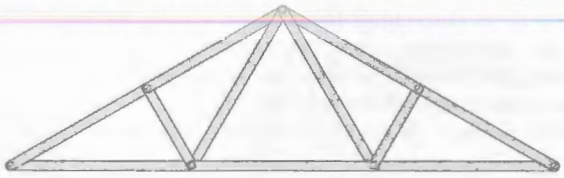
Prob. 6–126

ENGR10002 A publication from Pearson Custom Publishing exclusively for The University of Melbourne

CHAPTER REVIEW

Simple Truss

A simple truss consists of triangular elements connected together by pinned joints. The forces within its members can be determined by assuming the members are all two-force members, connected concurrently at each joint. The members are either in tension or compression, or carry no force.



Roof truss

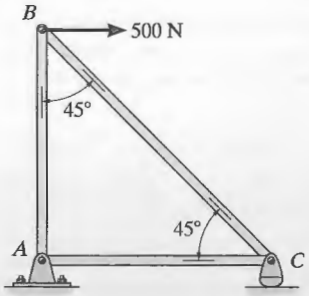
Method of Joints

The method of joints states that if a truss is in equilibrium, then each of its joints is also in equilibrium. For a plane truss, the concurrent force system at each joint must satisfy force equilibrium.

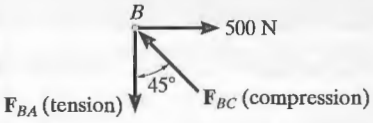
$$\Sigma F_x = 0$$

$$\Sigma F_y = 0$$

To obtain a numerical solution for the forces in the members, select a joint that has a free-body diagram with at most two unknown forces and one known force. (This may require first finding the reactions at the supports.)



Once a member force is determined, use its value and apply it to an adjacent joint.



Remember that forces that are found to *pull* on the joint are *tensile forces*, and those that *push* on the joint are *compressive forces*.

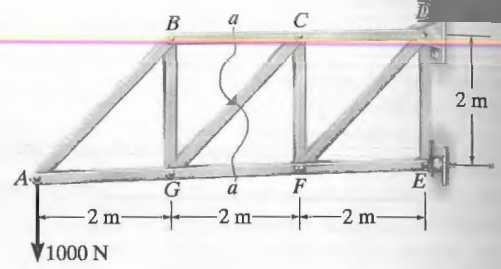
To avoid a simultaneous solution of two equations, set one of the coordinate axes along the line of action of one of the unknown forces and sum forces perpendicular to this axis. This will allow a direct solution for the other unknown.

The analysis can also be simplified by first identifying all the zero-force members.

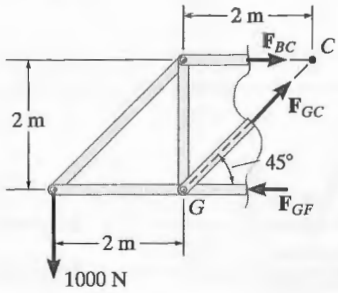
Hibbeler, RC. (2012). Structural Analysis. In *Engineering Mechanics Statics* (12th Ed. In SI Units, pp. 263-328). Singapore: Pearson Education South Asia Pte Ltd.

Method of Sections

The method of sections states that if a truss is in equilibrium, then each segment of the truss is also in equilibrium. Pass a section through the truss and the member whose force is to be determined. Then draw the free-body diagram of the sectioned part having the least number of forces on it.



Sectioned members subjected to *pulling* are in *tension*, and those that are subjected to *pushing* are in *compression*.



Three equations of equilibrium are available to determine the unknowns.

$$\Sigma F_x = 0$$
$$\Sigma F_y = 0$$
$$\Sigma M_O = 0$$

If possible, sum forces in a direction that is perpendicular to two of the three unknown forces. This will yield a direct solution for the third force.

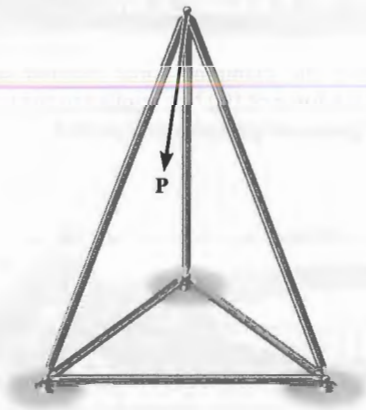
$$+\uparrow \Sigma F_y = 0$$
$$-1000\text{ N} + F_{GC}\sin 45° = 0$$
$$F_{GC} = 1.41\text{ kN (T)}$$

Sum moments about the point where the lines of action of two of the three unknown forces intersect, so that the third unknown force can be determined directly.

$$\circlearrowleft + \Sigma M_C = 0$$
$$1000\text{ N}(4\text{ m}) - F_{GF}(2\text{ m}) = 0$$
$$F_{GF} = 2\text{ kN (C)}$$

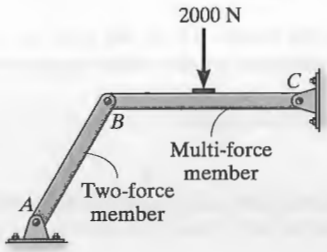
Space Truss

A space truss is a three-dimensional truss built from tetrahedral elements, and is analyzed using the same methods as for plane trusses. The joints are assumed to be ball-and-socket connections.

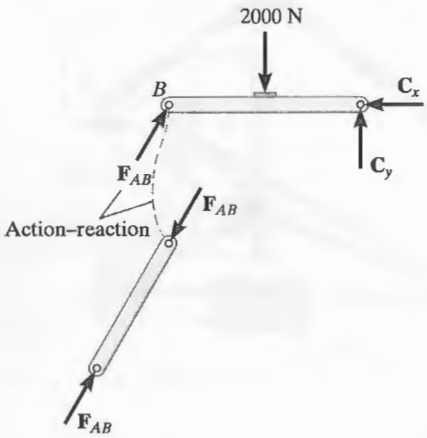


Frames and Machines

Frames and machines are structures that contain one or more multiforce members, that is, members with three or more forces or couples acting on them. Frames are designed to support loads, and machines transmit and alter the effect of forces.



The forces acting at the joints of a frame or machine can be determined by drawing the free-body diagrams of each of its members or parts. The principle of action–reaction should be carefully observed when indicating these forces on the free-body diagram of each adjacent member or pin. For a coplanar force system, there are three equilibrium equations available for each member.

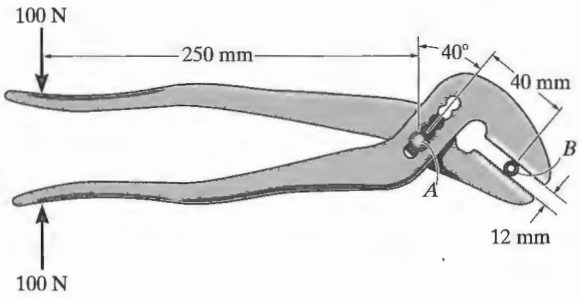


To simplify the analysis, be sure to recognize all two-force members. They have equal but opposite collinear forces at their ends.

MasteringENGINEERING
Customized coaching is available!

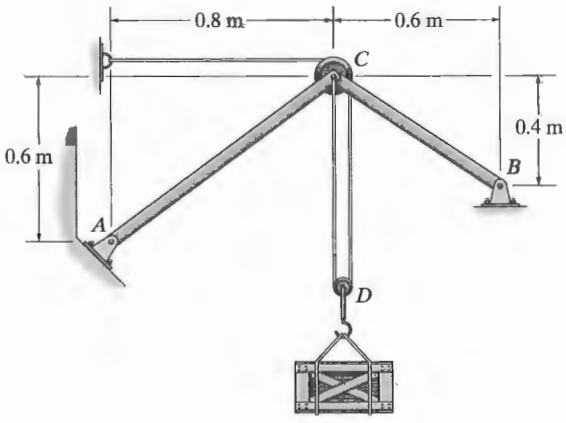
REVIEW PROBLEMS

6–127. Determine the clamping force exerted on the smooth pipe at B if a force of 100 N is applied to the handles of the pliers. The pliers are pinned together at A.



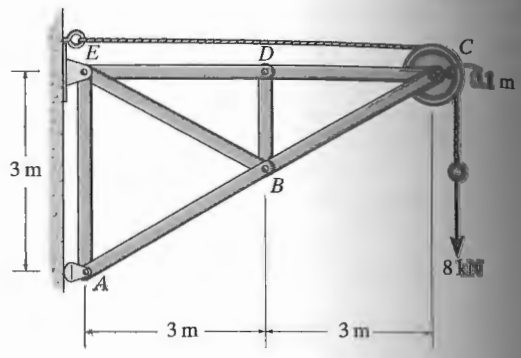
Prob. 6–127

***6–128.** Determine the forces which the pins at A and B exert on the two-member frame which supports the 100-kg crate.



Prob. 6–128

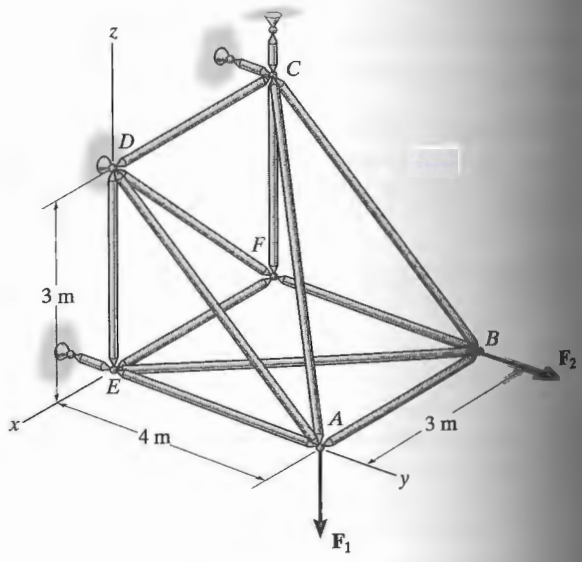
•6–129. Determine the force in each member of the truss and state if the members are in tension or compression.



Prob. 6–129

6–130. The space truss is supported by a ball-and-socket joint at D and short links at C and E. Determine the force in each member and state if the members are in tension or compression. Take $\mathbf{F}_1 = \{-500\mathbf{k}\}$ kN and $\mathbf{F}_2 = \{400\mathbf{j}\}$ kN.

6–131. The space truss is supported by a ball-and-socket joint at D and short links at C and E. Determine the force in each member and state if the members are in tension or compression. Take $\mathbf{F}_1 = \{200\mathbf{i} + 300\mathbf{j} - 500\mathbf{k}\}$ kN and $\mathbf{F}_2 = \{400\mathbf{j}\}$ kN.

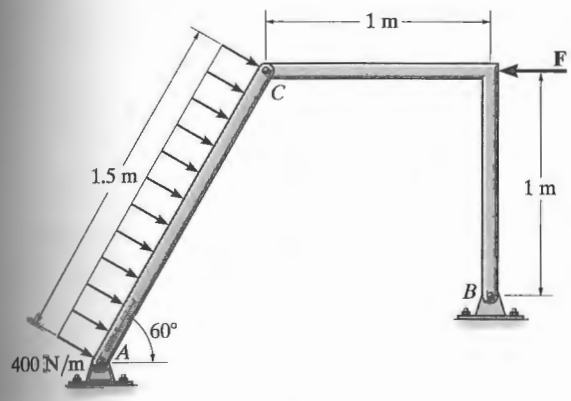


Probs. 6–130/131

son Custom Publishing exclusively for The University of Melbourne

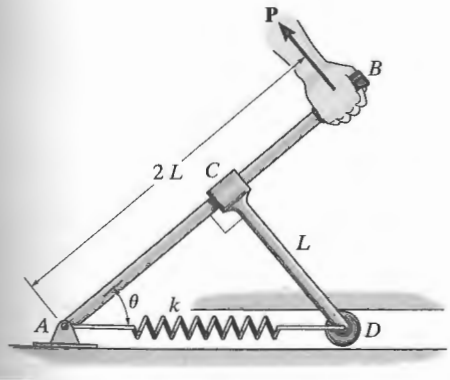
6–132. Determine the horizontal and vertical components of reaction that the pins A and B exert on the two-member frame. Set $F = 0$.

•6–133. Determine the horizontal and vertical components of reaction that pins A and B exert on the two-member frame. Set $F = 500$ N.



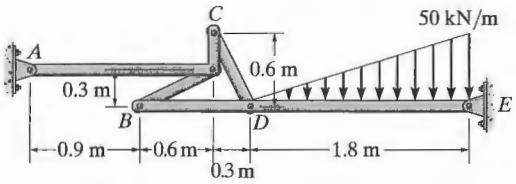
Probs. 6–132/133

6–134. The two-bar mechanism consists of a lever arm AB and smooth link CD, which has a fixed smooth collar at its end C and a roller at the other end D. Determine the force **P** needed to hold the lever in the position θ. The spring has a stiffness k and unstretched length $2L$. The roller contacts either the top or bottom portion of the horizontal guide.



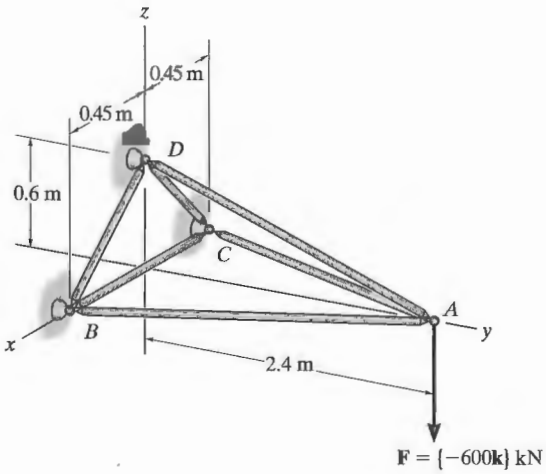
Prob. 6–134

6–135. Determine the horizontal and vertical components of reaction at the pin supports A and E of the compound beam assembly.



Prob. 6–135

***6–136.** Determine the force in members AB, AD, and AC of the space truss and state if the members are in tension or compression.



Prob. 6–136

IV. Mechanics: Dynamics

The content in this section is compiled from:

Engineering Mechanics: Dynamics
12th edition
Hibbeler

ENGR10003 – ENGI[redacted]EMS DESIGN 2
A publication from Pearson Custom Publishing exclusively for The Univ[redacted]y of Melbourne

Although each of these planes is rather large, from a distance their motion can be analysed as if each plane were a particle.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

Kinematics of a Particle

12

CHAPTER OBJECTIVES

- To introduce the concepts of position, displacement, velocity, and acceleration.
- To study particle motion along a straight line and represent this motion graphically.
- To investigate particle motion along a curved path using different coordinate systems.
- To present an analysis of dependent motion of two particles.
- To examine the principles of relative motion of two particles using translating axes.

12.1 Introduction

Mechanics is a branch of the physical sciences that is concerned with the state of rest or motion of bodies subjected to the action of forces. Engineering mechanics is divided into two areas of study, namely, statics and dynamics. *Statics* is concerned with the equilibrium of a body that is either at rest or moves with constant velocity. Here we will consider *dynamics*, which deals with the accelerated motion of a body. The subject of dynamics will be presented in two parts: *kinematics*, which treats only the geometric aspects of the motion, and *kinetics*, which is the analysis of the forces causing the motion. To develop these principles, the dynamics of a particle will be discussed first, followed by topics in rigid-body dynamics in two and then three dimensions.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

Historically, the principles of dynamics developed when it was possible to make an accurate measurement of time. Galileo Galilei (1564–1642) was one of the first major contributors to this field. His work consisted of experiments using pendulums and falling bodies. The most significant contributions in dynamics, however, were made by Isaac Newton (1642–1727), who is noted for his formulation of the three fundamental laws of motion and the law of universal gravitational attraction. Shortly after these laws were postulated, important techniques for their application were developed by Euler, D'Alembert, Lagrange, and others.

There are many problems in engineering whose solutions require application of the principles of dynamics. Typically the structural design of any vehicle, such as an automobile or airplane, requires consideration of the motion to which it is subjected. This is also true for many mechanical devices, such as motors, pumps, movable tools, industrial manipulators, and machinery. Furthermore, predictions of the motions of artificial satellites, projectiles, and spacecraft are based on the theory of dynamics. With further advances in technology, there will be an even greater need for knowing how to apply the principles of this subject.

Problem Solving. Dynamics is considered to be more involved than statics since both the forces applied to a body and its motion must be taken into account. Also, many applications require using calculus rather than just algebra and trigonometry. In any case, the most effective way of learning the principles of dynamics is *to solve problems*. To be successful at this, it is necessary to present the work in a logical and orderly manner as suggested by the following sequence of steps:

1. Read the problem carefully and try to correlate the actual physical situation with the theory you have studied.
2. Draw any necessary diagrams and tabulate the problem data.
3. Establish a coordinate system and apply the relevant principles generally in mathematical form.
4. Solve the necessary equations algebraically as far as practical; then, use a consistent set of units and complete the solution numerically. Report the answer with no more significant figures than the accuracy of the given data.
5. Study the answer using technical judgment and common sense to determine whether or not it seems reasonable.
6. Once the solution has been completed, review the problem. Try to think of other ways of obtaining the same solution.

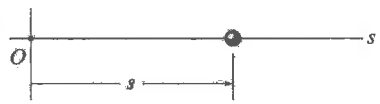
In applying this general procedure, do the work as neatly as possible. Being neat generally stimulates clear and orderly thinking, and vice versa.

12.2 Rectilinear Kinematics: Continuous Motion

We will begin our study of dynamics by discussing the kinematics of a particle that moves along a rectilinear or straight line path. Recall that a *particle* has a mass but negligible size and shape. Therefore we must limit application to those objects that have dimensions that are of no consequence in the analysis of the motion. In most problems, we will be interested in bodies of finite size, such as rockets, projectiles, or vehicles. Each of these objects can be considered as a particle, as long as the motion is characterized by the motion of its mass center and any rotation of the body is neglected.

Rectilinear Kinematics. The kinematics of a particle is characterized by specifying, at any given instant, the particle's position, velocity, and acceleration.

Position. The straight-line path of a particle will be defined using a single coordinate axis s, Fig. 12–1a. The origin O on the path is a fixed point, and from this point the *position coordinate* s is used to specify the location of the particle at any given instant. The magnitude of s is the distance from O to the particle, usually measured in meters (m), and the sense of direction is defined by the algebraic sign on s. Although the choice is arbitrary, in this case s is positive since the coordinate axis is positive to the right of the origin. Likewise, it is negative if the particle is located to the left of O. Realize that position is a vector quantity since it has both magnitude and direction. Here, however, it is being represented by the algebraic scalar s since the direction always remains along the coordinate axis.

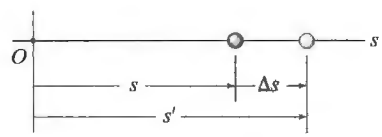


Position

(a)

Displacement. The *displacement* of the particle is defined as the *change* in its *position*. For example, if the particle moves from one point to another, Fig. 12–1b, the displacement is

$$\Delta s = s' - s$$



Displacement

(b)

Fig. 12–1

In this case Δs is *positive* since the particle's final position is to the *right* of its initial position, i.e., $s' > s$. Likewise, if the final position were to the *left* of its initial position, Δs would be *negative*.

The displacement of a particle is also a *vector quantity*, and it should be distinguished from the distance the particle travels. Specifically, the *distance traveled* is a *positive scalar* that represents the total length of path over which the particle travels.

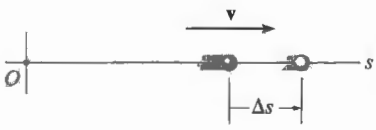
ENGR10003 – ENGI[redacted]
A publication from Pearson Custom Publishing exclusively for The Univ[redacted] of Melbo[redacted]

Velocity. If the particle moves through a displacement Δs during the time interval Δt, the *average velocity* of the particle during this time interval is

$$v_{avg} = \frac{\Delta s}{\Delta t}$$

If we take smaller and smaller values of Δt, the magnitude of Δs becomes smaller and smaller. Consequently, the *instantaneous velocity* is a vector defined as $v = \lim_{\Delta t \to 0} (\Delta s / \Delta t)$, or

$$(\overset{+}{\rightarrow}) \qquad \boxed{v = \frac{ds}{dt}} \qquad (12\text{–}1)$$



Velocity

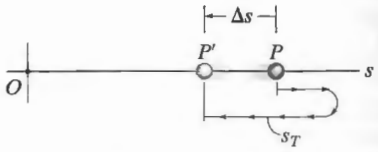
(c)

Since Δt or dt is always positive, the sign used to define the *sense* of the velocity is the same as that of Δs or ds. For example, if the particle is moving to the *right*, Fig. 12–1*c*, the velocity is *positive;* whereas if it is moving to the *left*, the velocity is *negative*. (This is emphasized here by the arrow written at the left of Eq. 12–1.) The *magnitude* of the velocity is known as the *speed*, and it is generally expressed in units of m/s.

Occasionally, the term "average speed" is used. The *average speed* is always a positive scalar and is defined as the total distance traveled by a particle, s_T, divided by the elapsed time Δt; i.e.,

$$(v_{sp})_{avg} = \frac{s_T}{\Delta t}$$

For example, the particle in Fig. 12–1*d* travels along the path of length s_T in time Δt, so its average speed is $(v_{sp})_{avg} = s_T / \Delta t$, but its average velocity is $v_{avg} = -\Delta s / \Delta t$.



Average velocity and
Average speed

(d)

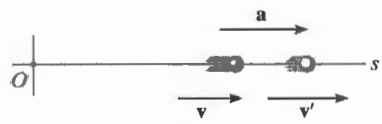
Fig. 12–1 (cont.)

eleration. Provided the velocity of the particle is known at two points, the *average acceleration* of the particle during the time interval Δt is defined as

$$a_{avg} = \frac{\Delta v}{\Delta t}$$

Here Δv represents the difference in the velocity during the time interval Δt, i.e., $\Delta v = v' - v$, Fig. 12–1*e*.

The *instantaneous acceleration* at time t is a vector that is found by taking smaller and smaller values of Δt and corresponding smaller and smaller values of Δv, so that $a = \lim_{\Delta t \to 0} (\Delta v / \Delta t)$, or



Acceleration

(e)

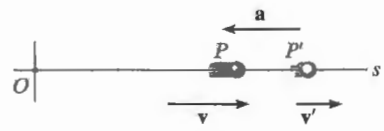
$$(\xrightarrow{+}) \qquad \boxed{a = \frac{dv}{dt}} \qquad (12\text{–}2)$$

bstituting Eq. 12–1 into this result, we can also write

$$(\xrightarrow{+}) \qquad a = \frac{d^2s}{dt^2}$$

Both the average and instantaneous acceleration can be either positive or negative. In particular, when the particle is *slowing down*, or its speed is decreasing, the particle is said to be *decelerating*. In this case, v' in Fig. 12–1*f* is *less* than v, and so $\Delta v = v' - v$ will be negative. Consequently, a will also be negative, and therefore it will act to the *left*, in the *opposite sense* to v. Also, note that when the *velocity* is *constant*, the *acceleration is zero* since $\Delta v = v - v = 0$. Units commonly used to express the magnitude of acceleration are m/s^2.

Finally, an important differential relation involving the displacement, velocity, and acceleration along the path may be obtained by eliminating the time differential dt between Eqs. 12–1 and 12–2, which gives



Deceleration

(f)

Fig. 12–1 (cont.)

$$(\xrightarrow{+}) \qquad \boxed{a\,ds = v\,dv} \qquad (12\text{–}3)$$

Although we have now produced three important kinematic equations, realize that the above equation is not independent of Eqs. 12–1 and 12–2.

Constant Acceleration, $a = a_c$. When the acceleration is constant, each of the three kinematic equations $a_c = dv/dt$, $v = ds/dt$, and $a_c\,ds = v\,dv$ can be integrated to obtain formulas that relate a_c, v, s, and t.

Velocity as a Function of Time. Integrate $a_c = dv/dt$, assuming that initially $v = v_0$ when $t = 0$.

$$\int_{v_0}^{v} dv = \int_0^t a_c\,dt$$

$$(\overset{+}{\rightarrow}) \qquad \boxed{\begin{array}{c} v = v_0 + a_c t \\ \text{Constant Acceleration} \end{array}} \qquad (12\text{–}4)$$

Position as a Function of Time. Integrate $v = ds/dt = v_0 + a_c t$, assuming that initially $s = s_0$ when $t = 0$.

$$\int_{s_0}^{s} ds = \int_0^t (v_0 + a_c t)\,dt$$

$$(\overset{+}{\rightarrow}) \qquad \boxed{\begin{array}{c} s = s_0 + v_0 t + \frac{1}{2}a_c t^2 \\ \text{Constant Acceleration} \end{array}} \qquad (12\text{–}5)$$

Velocity as a Function of Position. Either solve for t in Eq. 12–4 and substitute into Eq. 12–5, or integrate $v\,dv = a_c\,ds$, assuming that initially $v = v_0$ at $s = s_0$.

$$\int_{v_0}^{v} v\,dv = \int_{s_0}^{s} a_c\,ds$$

$$(\overset{+}{\rightarrow}) \qquad \boxed{\begin{array}{c} v^2 = v_0^2 + 2a_c(s - s_0) \\ \text{Constant Acceleration} \end{array}} \qquad (12\text{–}6)$$

The algebraic signs of s_0, v_0, and a_c, used in the above three equations, are determined from the positive direction of the s axis as indicated by the arrow written at the left of each equation. Remember that these equations are useful *only when the acceleration is constant and when* $t = 0$, $s = s_0$, $v = v_0$. A typical example of constant accelerated motion occurs when a body falls freely toward the earth. If air resistance is neglected and the distance of fall is short, then the *downward* acceleration of the body when it is close to the earth is constant and approximately 9.81 m/s^2. The proof of this is given in Example 13.2.

Important Points

- Dynamics is concerned with bodies that have accelerated motion.
- Kinematics is a study of the geometry of the motion.
- Kinetics is a study of the forces that cause the motion.
- Rectilinear kinematics refers to straight-line motion.
- Speed refers to the magnitude of velocity.
- Average speed is the total distance traveled divided by the total time. This is different from the average velocity, which is the displacement divided by the time.
- A particle that is slowing down is decelerating.
- A particle can have an acceleration and yet have zero velocity.
- The relationship $a\,ds = v\,dv$ is derived from $a = dv/dt$ and $v = ds/dt$, by eliminating dt.

During the time this rocket undergoes rectilinear motion, its altitude as a function of time can be measured and expressed as $s = s(t)$. Its velocity can then be found using $v = ds/dt$, and its acceleration can be determined from $a = dv/dt$.

Procedure for Analysis

Coordinate System.

- Establish a position coordinate s along the path and specify its *fixed origin* and positive direction.
- Since motion is along a straight line, the vector quantities position, velocity, and acceleration can be represented as algebraic scalars. For analytical work the sense of s, v, and a is then defined by their *algebraic signs*.
- The positive sense for each of these scalars can be indicated by an arrow shown alongside each kinematic equation as it is applied.

Kinematic Equations.

- If a relation is known between any *two* of the four variables a, v, s and t, then a third variable can be obtained by using one of the kinematic equations, $a = dv/dt$, $v = ds/dt$ or $a\,ds = v\,dv$, since each equation relates all three variables.*
- Whenever integration is performed, it is important that the position and velocity be known at a given instant in order to evaluate either the constant of integration if an indefinite integral is used, or the limits of integration if a definite integral is used.
- Remember that Eqs. 12–4 through 12–6 have only limited use. These equations apply *only* when the *acceleration is constant* and the initial conditions are $s = s_0$ and $v = v_0$ when $t = 0$.

*Some standard differentiation and integration formulas are given in Appendix A.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

12

EXAMPLE 12.1

The car in Fig. 12–2 moves in a straight line such that for a short time its velocity is defined by $v = (0.9t^2 + 0.6t)$ m/s, where t is in seconds. Determine its position and acceleration when $t = 3$ s. When $t = 0$, $s = 0$.

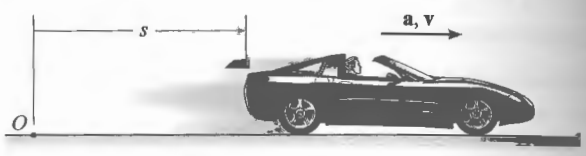


Fig. 12–2

SOLUTION

Coordinate System. The position coordinate extends from the fixed origin O to the car, positive to the right.

Position. Since $v = f(t)$, the car's position can be determined from $v = ds/dt$, since this equation relates v, s, and t. Noting that $s = 0$ when $t = 0$, we have*

$$(\overset{+}{\rightarrow}) \qquad v = \frac{ds}{dt} = (0.9t^2 + 0.6t)$$

$$\int_0^s ds = \int_0^t (0.9t^2 + 0.6t)dt$$

$$s\Big|_0^s = \left(0.3t^3 + 0.3t^2\right)\Big|_0^t$$

$$s = 0.3t^3 + 0.3t^2$$

When $t = 3$ s,

$$s = 0.3\,(3)^3 + 0.3\,(3)^2 = 10.8 \text{ m} \qquad \textit{Ans.}$$

Acceleration. Since $v = f(t)$, the acceleration is determined from $a = dv/dt$, since this equation relates a, v, and t.

$$(\overset{+}{\rightarrow}) \qquad a = \frac{dv}{dt} = \frac{d}{dt}(0.9t^2 + 0.6t)$$

$$= 1.8t + 0.6$$

When $t = 3$ s,

$$a = 1.8(3) + 0.6 = 6 \text{ m/s}^2 \rightarrow \qquad \textit{Ans.}$$

NOTE: The formulas for constant acceleration *cannot* be used to solve this problem, because the acceleration is a function of time.

*The *same result* can be obtained by evaluating a constant of integration C rather than using definite limits on the integral. For example, integrating $ds = (0.9t^2 + 0.6t)dt$ yields $s = 0.3t^3 + 0.3t^2 + C$. Using the condition that at $t = 0$, $s = 0$, then $C = 0$.

A publication from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 12.2

A small projectile is fired vertically *downward* into a fluid medium with an initial velocity of 60 m/s. Due to the drag resistance of the fluid the projectile experiences a deceleration of $a = (-0.4v^3)\ \text{m/s}^2$, where v is in m/s. Determine the projectile's velocity and position 4 s after it is fired.

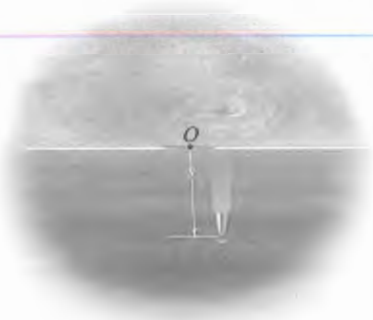


Fig. 12–3

SOLUTION

Coordinate System. Since the motion is downward, the position coordinate is positive downward, with origin located at O, Fig. 12–3.

Velocity. Here $a = f(v)$ and so we must determine the velocity as a function of time using $a = dv/dt$, since this equation relates v, a, and t. (Why not use $v = v_0 + a_c t$?) Separating the variables and integrating, with $v_0 = 60$ m/s when $t = 0$, yields

$$(+\downarrow) \qquad a = \frac{dv}{dt} = -0.4v^3$$

$$\int_{60\ \text{m/s}}^{v} \frac{dv}{-0.4v^3} = \int_0^t dt$$

$$\frac{1}{-0.4}\left(\frac{1}{-2}\right)\frac{1}{v^2}\bigg|_{60}^{v} = t - 0$$

$$\frac{1}{0.8}\left[\frac{1}{v^2} - \frac{1}{(60)^2}\right] = t$$

$$v = \left\{\left[\frac{1}{(60)^2} + 0.8t\right]^{-1/2}\right\}\ \text{m/s}$$

Here the positive root is taken, since the projectile will continue to move downward. When $t = 4$ s,

$$v = 0.559\ \text{m/s}\downarrow \qquad \textit{Ans.}$$

Position. Knowing $v = f(t)$, we can obtain the projectile's position from $v = ds/dt$, since this equation relates s, v, and t. Using the initial condition $s = 0$, when $t = 0$, we have

$$(+\downarrow) \qquad v = \frac{ds}{dt} = \left[\frac{1}{(60)^2} + 0.8t\right]^{-1/2}$$

$$\int_0^s ds = \int_0^t \left[\frac{1}{(60)^2} + 0.8t\right]^{-1/2} dt$$

$$s = \frac{2}{0.8}\left[\frac{1}{(60)^2} + 0.8t\right]^{1/2}\bigg|_0^t$$

$$s = \frac{1}{0.4}\left\{\left[\frac{1}{(60)^2} + 0.8t\right]^{1/2} - \frac{1}{60}\right\}\ \text{m}$$

When $t = 4$ s,

$$s = 4.43\ \text{m} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

12

EXAMPLE 12.3

During a test a rocket travels upward at 75 m/s, and when it is 40 m from the ground its engine fails. Determine the maximum height s_B reached by the rocket and its speed just before it hits the ground. While in motion the rocket is subjected to a constant downward acceleration of 9.81 m/s^2 due to gravity. Neglect the effect of air resistance.

SOLUTION

Coordinate System. The origin O for the position coordinate s is taken at ground level with positive upward, Fig. 12–4.

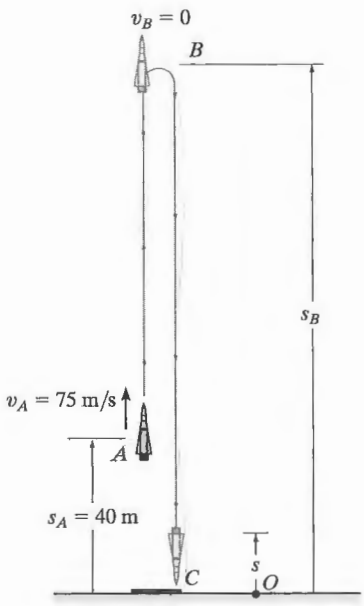


Fig. 12–4

Maximum Height. Since the rocket is traveling *upward*, $v_A = +75$m/s when $t = 0$. At the maximum height $s = s_B$ the velocity $v_B = 0$. For the entire motion, the acceleration is $a_c = -9.81$ m/s^2 (negative since it acts in the *opposite* sense to positive velocity or positive displacement). Since a_c is *constant* the rocket's position may be related to its velocity at the two points A and B on the path by using Eq. 12–6, namely,

$$(+\uparrow) \qquad v_B^2 = v_A^2 + 2a_c(s_B - s_A)$$

$$0 = (75 \text{ m/s})^2 + 2(-9.81 \text{ m/s}^2)(s_B - 40 \text{ m})$$

$$s_B = 327 \text{ m} \qquad \textit{Ans.}$$

Velocity. To obtain the velocity of the rocket just before it hits the ground, we can apply Eq. 12–6 between points B and C, Fig. 12–4.

$$(+\uparrow) \qquad v_C^2 = v_B^2 + 2a_c(s_C - s_B)$$

$$= 0 + 2(-9.81 \text{ m/s}^2)(0 - 327 \text{ m})$$

$$v_C = -80.1 \text{ m/s} = 80.1 \text{ m/s} \downarrow \qquad \textit{Ans.}$$

The negative root was chosen since the rocket is moving downward.

Similarly, Eq. 12–6 may also be applied between points A and C, i.e.,

$$(+\uparrow) \qquad v_C^2 = v_A^2 + 2a_c(s_C - s_A)$$

$$= (75 \text{ m/s})^2 + 2(-9.81 \text{ m/s}^2)(0 - 40 \text{ m})$$

$$v_C = -80.1 \text{ m/s} = 80.1 \text{ m/s} \downarrow \qquad \textit{Ans.}$$

NOTE: It should be realized that the rocket is subjected to a *deceleration* from A to B of 9.81 m/s^2, and then from B to C it is *accelerated* at this rate. Furthermore, even though the rocket momentarily comes to *rest* at B ($v_B = 0$) the acceleration at B is still 9.81 m/s^2 downward!

EXAMPLE 12.4

A metallic particle is subjected to the influence of a magnetic field as it travels downward through a fluid that extends from plate A to plate B, Fig. 12–5. If the particle is released from rest at the midpoint C, $s = 100$ mm, and the acceleration is $a = (4s)$ m/s^2, where s is in meters, determine the velocity of the particle when it reaches plate B, $s = 200$ mm, and the time it takes to travel from C to B.

SOLUTION

Coordinate System. As shown in Fig. 12–5, s is positive downward, measured from plate A.

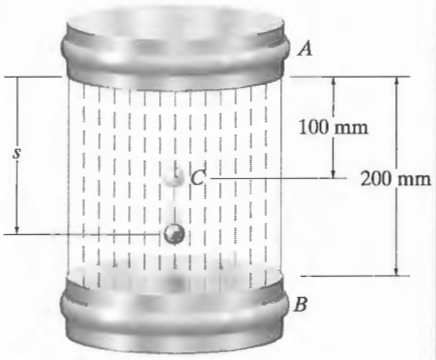


Fig. 12–5

Velocity. Since $a = f(s)$, the velocity as a function of position can be obtained by using $v\,dv = a\,ds$. Realizing that $v = 0$ at $s = 0.1$ m, we have

$$(+\downarrow) \qquad v\,dv = a\,ds$$

$$\int_0^v v\,dv = \int_{0.1\text{ m}}^s 4s\,ds$$

$$\frac{1}{2}v^2\Big|_0^v = \frac{4}{2}s^2\Big|_{0.1\text{ m}}^s$$

$$v = 2(s^2 - 0.01)^{1/2}\text{ m/s} \qquad (1)$$

At $s = 200$ mm $= 0.2$ m,

$$v_B = 0.346\text{ m/s} = 346\text{ mm/s}\downarrow \qquad \textit{Ans.}$$

The positive root is chosen since the particle is traveling downward, i.e., in the $+s$ direction.

Time. The time for the particle to travel from C to B can be obtained using $v = ds/dt$ and Eq. 1, where $s = 0.1$ m when $t = 0$. From Appendix A,

$$(+\downarrow) \qquad ds = v\,dt$$

$$= 2(s^2 - 0.01)^{1/2}dt$$

$$\int_{0.1}^s \frac{ds}{(s^2 - 0.01)^{1/2}} = \int_0^t 2\,dt$$

$$\ln\left(\sqrt{s^2 - 0.01} + s\right)\Big|_{0.1}^s = 2t\Big|_0^t$$

$$\ln\left(\sqrt{s^2 - 0.01} + s\right) + 2.303 = 2t$$

At $s = 0.2$ m,

$$t = \frac{\ln\left(\sqrt{(0.2)^2 - 0.01} + 0.2\right) + 2.303}{2} = 0.658\text{ s} \qquad \textit{Ans.}$$

Note: The formulas for constant acceleration cannot be used here because the acceleration changes with position, i.e., $a = 4s$.

12

EXAMPLE 12.5

A particle moves along a horizontal path with a velocity of $v = (3t^2 - 6t)$ m/s, where t is the time in seconds. If it is initially located at the origin O, determine the distance traveled in 3.5 s, and the particle's average velocity and average speed during the time interval.

SOLUTION

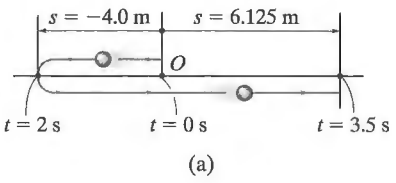


Fig. 12–6

Coordinate System. Here positive motion is to the right, measured from the origin O, Fig. 12–6a.

Distance Traveled. Since $v = f(t)$, the position as a function of time may be found by integrating $v = ds/dt$ with $t = 0$, $s = 0$.

$$(\overset{+}{\rightarrow}) \qquad ds = v\,dt$$

$$= (3t^2 - 6t)dt$$

$$\int_0^s ds = \int_0^t (3t^2 - 6t)\,dt$$

$$s = (t^3 - 3t^2)\text{m} \qquad (1)$$

In order to determine the distance traveled in 3.5 s, it is necessary to investigate the path of motion. If we consider a graph of the velocity function, Fig. 12–6b, then it reveals that for $0 < t < 2$ s the velocity is *negative*, which means the particle is traveling to the *left*, and for $t > 2$ s the velocity is *positive*, and hence the particle is traveling to the *right*. Also, note that $v = 0$ at $t = 2$ s. The particle's position when $t = 0$, $t = 2$ s, and $t = 3.5$ s can now be determined from Eq. 1. This yields

$$s|_{t=0} = 0 \qquad s|_{t=2\text{ s}} = -4.0\text{ m} \qquad s|_{t=3.5\text{ s}} = 6.125\text{ m}$$

The path is shown in Fig. 12–6a. Hence, the distance traveled in 3.5 s is

$$s_T = 4.0 + 4.0 + 6.125 = 14.125\text{ m} = 14.1\text{ m} \qquad \textit{Ans.}$$

Velocity. The *displacement* from $t = 0$ to $t = 3.5$ s is

$$\Delta s = s|_{t=3.5\text{ s}} - s|_{t=0} = 6.125\text{ m} - 0 = 6.125\text{ m}$$

and so the average velocity is

$$v_{avg} = \frac{\Delta s}{\Delta t} = \frac{6.125\text{ m}}{3.5\text{ s} - 0} = 1.75\text{ m/s} \rightarrow \qquad \textit{Ans.}$$

The average speed is defined in terms of the *distance traveled* s_T. This positive scalar is

$$(v_{sp})_{avg} = \frac{s_T}{\Delta t} = \frac{14.125\text{ m}}{3.5\text{ s} - 0} = 4.04\text{ m/s} \qquad \textit{Ans.}$$

Note: In this problem, the acceleration is $a = dv/dt = (6t - 6)\text{ m/s}^2$, which is not constant.

FUNDAMENTAL PROBLEMS

F12–1. Initially, the car travels along a straight road with a speed of 35 m/s. If the brakes are applied and the speed of the car is reduced to 10 m/s in 15 s, determine the constant deceleration of the car.

F12–1

F12–2. A ball is thrown vertically upward with a speed of 15 m/s. Determine the time of flight when it returns to its original position.

F12–2

F12–3. A particle travels along a straight line with a velocity of $v = (4t - 3t^2)$ m/s, where t is in seconds. Determine the position of the particle when $t = 4$ s. $s = 0$ when $t = 0$.

F12–3

F12–4. A particle travels along a straight line with a speed $v = (0.5t^3 - 8t)$ m/s, where t is in seconds. Determine the acceleration of the particle when $t = 2$ s.

F12–4

F12–5. The position of the particle is given by $s = (2t^2 - 8t + 6)$ m, where t is in seconds. Determine the time when the velocity of the particle is zero, and the total distance traveled by the particle when $t = 3$ s.

F12–5

F12–6. A particle travels along a straight line with an acceleration of $a = (10 - 0.2s)$ m/s^2, where s is measured in meters. Determine the velocity of the particle when $s = 10$ m if $v = 5$ m/s at $s = 0$.

F12–6

F12–7. A particle moves along a straight line such that its acceleration is $a = (4t^2 - 2)$ m/s^2, where t is in seconds. When $t = 0$, the particle is located 2 m to the left of the origin, and when $t = 2$ s, it is 20 m to the left of the origin. Determine the position of the particle when $t = 4$ s.

F12–7

F12–8. A particle travels along a straight line with a velocity of $v = (20 - 0.05s^2)$ m/s, where s is in meters. Determine the acceleration of the particle at $s = 15$ m.

F12–8

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

MasteringENGINEERING
Customized coaching is available!

12

PROBLEMS

•**12–1.** A car starts from rest and with constant acceleration achieves a velocity of 15 m/s when it travels a distance of 200 m. Determine the acceleration of the car and the time required.

12–2. A train starts from rest at a station and travels with a constant acceleration of $1\,\text{m/s}^2$. Determine the velocity of the train when $t = 30\,\text{s}$ and the distance traveled during this time.

12–3. An elevator descends from rest with an acceleration of $1.5\,\text{m/s}^2$ until it achieves a velocity of 4.5 m/s. Determine the time required and the distance traveled.

***12–4.** A car is traveling at 15 m/s, when the traffic light 50 m ahead turns yellow. Determine the required constant deceleration of the car and the time needed to stop the car at the light.

•**12–5.** A particle is moving along a straight line with the acceleration $a = (12t - 3t^{1/2})\,\text{m/s}^2$, where t is in seconds. Determine the velocity and the position of the particle as a function of time. When $t = 0$, $v = 0$ and $s = 15\,\text{m}$.

12–6. A ball is released from the bottom of an elevator which is traveling upward with a velocity of 2 m/s. If the ball strikes the bottom of the elevator shaft in 3 s, determine the height of the elevator from the bottom of the shaft at the instant the ball is released. Also, find the velocity of the ball when it strikes the bottom of the shaft.

12–7. A car has an initial speed of 25 m/s and a constant deceleration of $3\,\text{m/s}^2$. Determine the velocity of the car when $t = 4\,\text{s}$. What is the displacement of the car during the 4-s time interval? How much time is needed to stop the car?

***12–8.** If a particle has an initial velocity of $v_0 = 12\,\text{m/s}$ to the right, at $s_0 = 0$, determine its position when $t = 10\,\text{s}$, if $a = 2\,\text{m/s}^2$ to the left.

•**12–9.** The acceleration of a particle traveling along a straight line is $a = k/v$, where k is a constant. If $s = 0$, $v = v_0$ when $t = 0$, determine the velocity of the particle as a function of time t.

12–10. Car A starts from rest at $t = 0$ and travels along a straight road with a constant acceleration of $2\,\text{m/s}^2$ until it reaches a speed of 27 m/s. Afterwards it maintains this speed. Also, when $t = 0$, car B located 2000 m down the road is traveling towards A at a constant speed of 20 m/s. Determine the distance traveled by car A when they pass each other.

Prob. 12–10

12–11. A particle travels along a straight line with a velocity $v = (12 - 3t^2)\,\text{m/s}$, where t is in seconds. When $t = 1\,\text{s}$, the particle is located 10 m to the left of the origin. Determine the acceleration when $t = 4\,\text{s}$, the displacement from $t = 0$ to $t = 10\,\text{s}$, and the distance the particle travels during this time period.

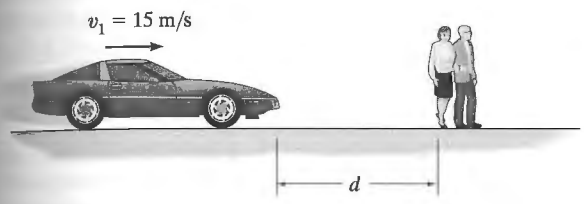
***12–12.** A sphere is fired downwards into a medium with an initial speed of 27 m/s. If it experiences a deceleration of $a = (-6t)\,\text{m/s}^2$, where t is in seconds, determine the distance traveled before it stops.

www Video Solutions

•**12–13.** A particle travels along a straight line such that in 2 s it moves from an initial position $s_A = +0.5\,\text{m}$ to a position $s_B = -1.5\,\text{m}$. Then in another 4 s it moves from s_B to $s_C = +2.5\,\text{m}$. Determine the particle's average velocity and average speed during the 6-s time interval.

12–14. A particle travels along a straight-line path such that in 4 s it moves from an initial position $s_A = -8\,\text{m}$ to a position $s_B = +3\,\text{m}$. Then in another 5 s it moves from s_B to $s_C = -6\,\text{m}$. Determine the particle's average velocity and average speed during the 9-s time interval.

12–15. Tests reveal that a normal driver takes about 0.75 s before he or she can *react* to a situation to avoid a collision. It takes about 3 s for a driver having 0.1% alcohol in his system to do the same. If such drivers are traveling on a straight road at 54 km/h (15 m/s) and their cars can decelerate at 0.6 m/s^2, determine the shortest stopping distance d for each from the moment they see the pedestrians. *Moral*: If you must drink, please don't drive!



Prob. 12–15

***12–16.** As a train accelerates uniformly it passes successive kilometer marks while traveling at velocities of 2 m/s and then 10 m/s. Determine the train's velocity when it passes the next kilometer mark and the time it takes to travel the 2-km distance.

•12–17. A ball is thrown with an upward velocity of 5 m/s from the top of a 10-m high building. One second later another ball is thrown vertically from the ground with a velocity of 10 m/s. Determine the height from the ground where the two balls pass each other.

12–18. A car starts from rest and moves with a constant acceleration of 1.5 m/s^2 until it achieves a velocity of 25 m/s. It then travels with constant velocity for 60 seconds. Determine the average speed and the total distance traveled.

12–19. A car is to be hoisted by elevator to the fourth floor of a parking garage, which is 14.4 m above the ground. If the elevator can accelerate at 0.18 m/s^2, decelerate at 0.09 m/s^2, and reach a maximum speed of 2.4 m/s, determine the shortest time to make the lift, starting from rest and ending at rest.

***12–20.** A particle is moving along a straight line such that its speed is defined as $v = (-4s^2)$ m/s, where s is in meters. If $s = 2$ m when $t = 0$, determine the velocity and acceleration as functions of time.

•12–21. Two particles A and B start from rest at the origin $s = 0$ and move along a straight line such that $a_A = (6t - 3)$ m/s^2 and $a_B = (12t^2 - 8)$ m/s^2, where t is in seconds. Determine the distance between them when $t = 4$ s and the total distance each has traveled in $t = 4$ s.

12–22. A particle moving along a straight line is subjected to a deceleration $a = (-2v^3)$ m/s^2, where v is in m/s. If it has a velocity $v = 8$ m/s and a position $s = 10$ m when $t = 0$, determine its velocity and position when $t = 4$ s.

12–23. A particle is moving along a straight line such that its acceleration is defined as $a = (-2v)$ m/s^2, where v is in meters per second. If $v = 20$ m/s when $s = 0$ and $t = 0$, determine the particle's position, velocity, and acceleration as functions of time.

***12–24.** A particle starts from rest and travels along a straight line with an acceleration $a = (30 - 0.2v)$ m/s^2, where v is in m/s. Determine the time when the velocity of the particle is $v = 30$ m/s.

•12–25. When a particle is projected vertically upwards with an initial velocity of v_0, it experiences an acceleration $a = -(g + kv^2)$, where g is the acceleration due to gravity, k is a constant and v is the velocity of the particle. Determine the maximum height reached by the particle.

12–26. The acceleration of a particle traveling along a straight line is $a = (0.02e^t)$ m/s^2, where t is in seconds. If $v = 0$, $s = 0$ when $t = 0$, determine the velocity and acceleration of the particle at $s = 4$ m.

Video Solutions

12–27. A particle moves along a straight line with an acceleration of $a = 5/(3s^{1/3} + s^{5/2})$ m/s^2, where s is in meters. Determine the particle's velocity when $s = 2$ m, if it starts from rest when $s = 1$ m. Use Simpson's rule to evaluate the integral.

***12–28.** If the effects of atmospheric resistance are accounted for, a falling body has an acceleration defined by the equation $a = 9.81[1 - v^2(10^{-4})]$ m/s^2, where v is in m/s and the positive direction is downward. If the body is released from rest at a very *high altitude*, determine (a) the velocity when $t = 5$ s, and (b) the body's terminal or maximum attainable velocity (as $t \rightarrow \infty$).

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

12

•12–29. The position of a particle along a straight line is given by $s = (1.5t^3 - 13.5t^2 + 22.5t)$ m, where t is in seconds. Determine the position of the particle when $t = 6$ s and the total distance it travels during the 6-s time interval. *Hint:* Plot the path to determine the total distance traveled.

12–30. The velocity of a particle traveling along a straight line is $v = v_0 - ks$, where k is constant. If $s = 0$ when $t = 0$, determine the position and acceleration of the particle as a function of time.

12–31. The acceleration of a particle as it moves along a straight line is given by $a = (2t - 1)$ m/s^2, where t is in seconds. If $s = 1$ m and $v = 2$ m/s when $t = 0$, determine the particle's velocity and position when $t = 6$ s. Also, determine the total distance the particle travels during this time period.

***12–32.** Ball A is thrown vertically upward from the top of a 30-m-high-building with an initial velocity of 5 m/s. At the same instant another ball B is thrown upward from the ground with an initial velocity of 20 m/s. Determine the height from the ground and the time at which they pass.

•12–33. A motorcycle starts from rest at $t = 0$ and travels along a straight road with a constant acceleration of 1.8 m/s^2 until it reaches a speed of 15 m/s. Afterwards it maintains this speed. Also, when $t = 0$, a car located 1800 m down the road is traveling toward the motorcycle at a constant speed of 9 m/s. Determine the time and the distance traveled by the motorcycle when they pass each other.

12–34. A particle moves along a straight line with a velocity $v = (200s)$ mm/s, where s is in millimeters. Determine the acceleration of the particle at $s = 2000$ mm. How long does the particle take to reach this position if $s = 500$ mm when $t = 0$?

■12–35. A particle has an initial speed of 27 m/s. If it experiences a deceleration of $a = (-6t)$ m/s^2, where t is in seconds, determine its velocity, after it has traveled 10 m. How much time does this take?

***12–36.** The acceleration of a particle traveling along a straight line is $a = (8 - 2s)$ m/s^2, where s is in meters. If $v = 0$ at $s = 0$, determine the velocity of the particle at $s = 2$ m, and the position of the particle when the velocity is maximum.

•12–37. Ball A is thrown vertically upwards with a velocity of v_0. Ball B is thrown upwards from the same point with the same velocity t seconds later. Determine the elapsed time $t < 2v_0/g$ from the instant ball A is thrown to when the balls pass each other, and find the velocity of each ball at this instant.

12–38. As a body is projected to a high altitude above the earth's *surface,* the variation of the acceleration of gravity with respect to altitude y must be taken into account. Neglecting air resistance, this acceleration is determined from the formula $a = -g_0[R^2/(R + y)^2]$, where g_0 is the constant gravitational acceleration at sea level, R is the radius of the earth, and the positive direction is measured upward. If $g_0 = 9.81$ m/s^2 and $R = 6356$ km, determine the minimum initial velocity (escape velocity) at which a projectile should be shot vertically from the earth's surface so that it does not fall back to the earth. *Hint:* This requires that $v = 0$ as $y \to \infty$.

12–39. Accounting for the variation of gravitational acceleration a with respect to altitude y (see Prob. 12–38), derive an equation that relates the velocity of a freely falling particle to its altitude. Assume that the particle is released from rest at an altitude y_0 from the earth's surface. With what velocity does the particle strike the earth if it is released from rest at an altitude $y_0 = 500$ km? Use the numerical data in Prob. 12–38.

***12–40.** When a particle falls through the air, its initial acceleration $a = g$ diminishes until it is zero, and thereafter it falls at a constant or terminal velocity v_f. If this variation of the acceleration can be expressed as $a = (g/v_f^2)(v_f^2 - v^2)$, determine the time needed for the velocity to become $v = v_f/2$. Initially the particle falls from rest.

•12–41. A particle is moving along a straight line such that its position from a fixed point is $s = (12 - 15t^2 + 5t^3)$ m, where t is in seconds. Determine the total distance traveled by the particle from $t = 1$ s to $t = 3$ s. Also, find the average speed of the particle during this time interval.

Pearson Custom Publishing exclusively for The University of Melbourne

12.3 Rectilinear Kinematics: Erratic Motion

When a particle has erratic or changing motion then its position, velocity, and acceleration *cannot* be described by a single continuous mathematical function along the entire path. Instead, a series of functions will be required to specify the motion at different intervals. For this reason, it is convenient to represent the motion as a graph. If a graph of the motion that relates any two of the variables s, v, a, t can be drawn, then this graph can be used to construct subsequent graphs relating two other variables since the variables are related by the differential relationships $v = ds/dt$, $a = dv/dt$, or $a\,ds = v\,dv$. Several situations occur frequently.

The *s–t*, *v–t*, and *a–t* Graphs. To construct the v–t graph given the s–t graph, Fig. 12–7*a*, the equation $v = ds/dt$ should be used, since it relates the variables s and t to v. This equation states that

$$\frac{ds}{dt} = v$$

slope of s–t graph = velocity

For example, by measuring the slope on the s–t graph when $t = t_1$, the velocity is v_1, which is plotted in Fig. 12–7*b*. The v–t graph can be constructed by plotting this and other values at each instant.

The a–t graph can be constructed from the v–t graph in a similar manner, Fig. 12–8, since

$$\frac{dv}{dt} = a$$

slope of v–t graph = acceleration

Examples of various measurements are shown in Fig. 12–8*a* and plotted in Fig. 12–8*b*.

If the s–t curve for each interval of motion can be expressed by a mathematical function $s = s(t)$, then the equation of the v–t graph for the same interval can be obtained by differentiating this function with respect to time since $v = ds/dt$. Likewise, the equation of the a–t graph for the same interval can be determined by differentiating $v = v(t)$ since $a = dv/dt$. Since differentiation reduces a polynomial of degree n to that of degree $n - 1$, then if the s–t graph is parabolic (a second-degree curve), the v–t graph will be a sloping line (a first-degree curve), and the a–t graph will be a constant or a horizontal line (a zero-degree curve).

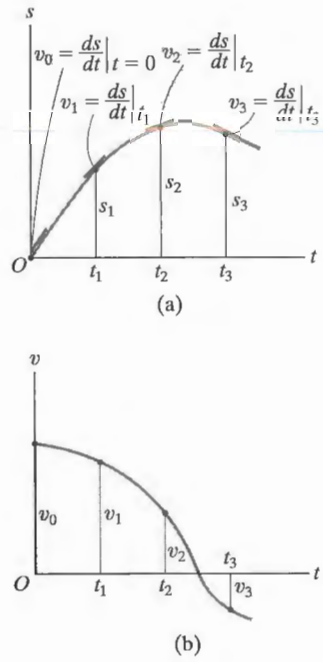


Fig. 12–7

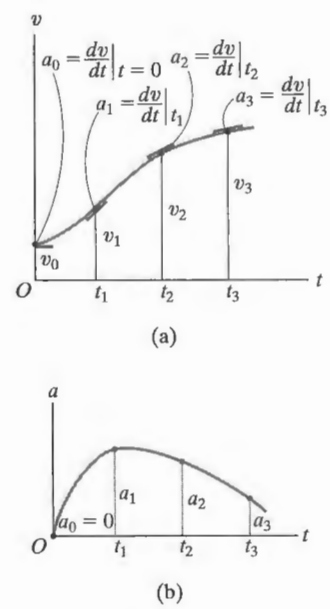


Fig. 12–8

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

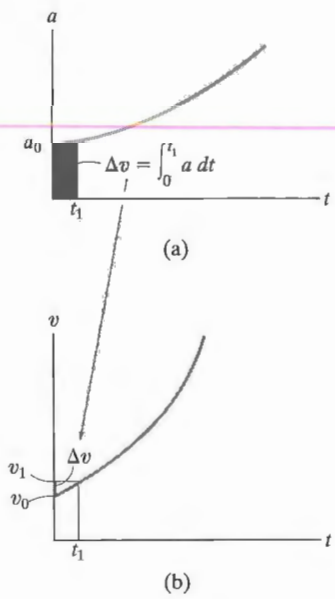


Fig. 12–9

If the a–t graph is given, Fig. 12–9a, the v–t graph may be constructed using $a = dv/dt$, written as

$$\Delta v = \int a\,dt$$

$$\text{change in velocity} = \text{area under } a\text{–}t \text{ graph}$$

Hence, to construct the v–t graph, we begin with the particle's initial velocity v_0 and then add to this small increments of area (Δv) determined from the a–t graph. In this manner successive points, $v_1 = v_0 + \Delta v$, etc., for the v–t graph are determined, Fig. 12–9b. Notice that an algebraic addition of the area increments of the a–t graph is necessary, since areas lying above the t axis correspond to an increase in v ("positive" area), whereas those lying below the axis indicate a decrease in v ("negative" area).

Similarly, if the v–t graph is given, Fig. 12–10a, it is possible to determine the s–t graph using $v = ds/dt$, written as

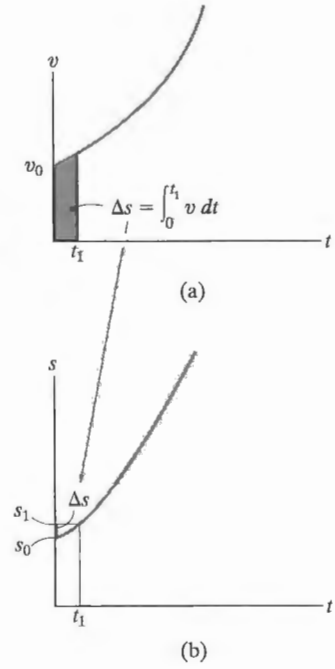


Fig. 12–10

$$\Delta s = \int v\,dt$$

$$\text{displacement} = \text{area under } v\text{–}t \text{ graph}$$

In the same manner as stated above, we begin with the particle's initial position s_0 and add (algebraically) to this small area increment Δs determined from the v–t graph, Fig. 12–10b.

If segments of the a–t graph can be described by a series of equations, then each of these equations can be *integrated* to yield equations describing the corresponding segments of the v–t graph. In a similar manner, the s–t graph can be obtained by integrating the equations which describe the segments of the v–t graph. As a result, if the a–t graph is linear (a first-degree curve), integration will yield a v–t graph that is parabolic (a second-degree curve) and an s–t graph that is cubic (third-degree curve).

The *v–s* and *a–s* Graphs. If the a–s graph can be constructed, then points on the v–s graph can be determined by using $v\,dv = a\,ds$. Integrating this equation between the limits $v = v_0$ at $s = s_0$ and $v = v_1$ at $s = s_1$, we have,

$$\boxed{\tfrac{1}{2}(v_1^2 - v_0^2) = \underbrace{\int_{s_0}^{s_1} a\,ds}_{\text{area under } a\text{–}s \text{ graph}}}$$

Therefore, if the red area in Fig. 12–11a is determined, and the initial velocity v_0 at $s_0 = 0$ is known, then $v_1 = \left(2\int_{s_0}^{s_1} a\,ds + v_0^2\right)^{1/2}$, Fig. 12–11$b$. Successive points on the v–s graph can be constructed in this manner.

If the v–s graph is known, the acceleration a at any position s can be determined using $a\,ds = v\,dv$, written as

$$\boxed{a = v\left(\frac{dv}{ds}\right) \qquad \text{acceleration} = \underbrace{\text{velocity times slope of } v\text{–}s \text{ graph}}}$$

Thus, at any point (s, v) in Fig. 12–12a, the slope dv/ds of the v–s graph is measured. Then with v and dv/ds known, the value of a can be calculated, Fig. 12–12b.

The v–s graph can also be constructed from the a–s graph, or vice versa, by approximating the known graph in various intervals with mathematical functions, $v = f(s)$ or $a = g(s)$, and then using $a\,ds = v\,dv$ to obtain the other graph.

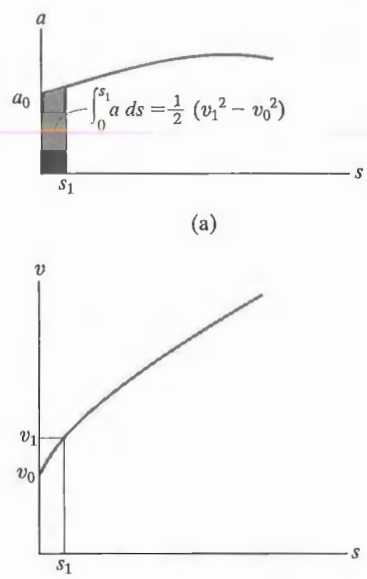


Fig. 12–11

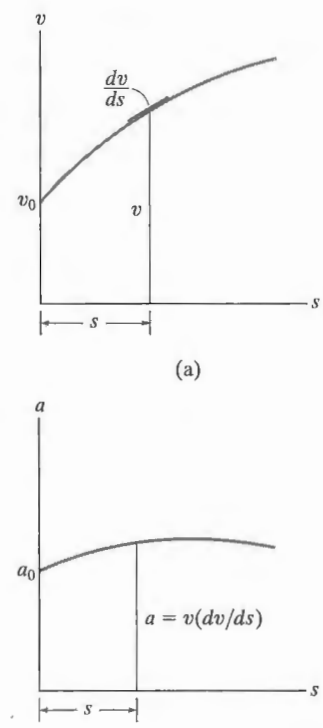
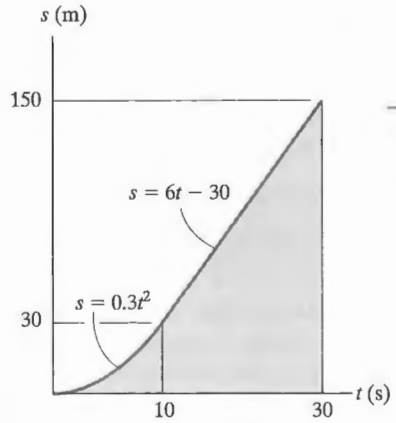


Fig. 12–12

12

EXAMPLE 12.6

A bicycle moves along a straight road such that its position is described by the graph shown in Fig. 12–13*a*. Construct the v–t and a–t graphs for $0 \le t \le 30$ s.



(a)

SOLUTION

v–t Graph. Since $v = ds/dt$, the v–t graph can be determined by differentiating the equations defining the s–t graph, Fig. 12–13*a*. We have

$$0 \le t < 10 \text{ s}; \qquad s = 0.3(t^2) \text{ m} \qquad v = \frac{ds}{dt} = (0.6t) \text{ m/s}$$

$$10 \text{ s} < t \le 30 \text{ s}; \qquad s = (6t - 30) \text{ m} \qquad v = \frac{ds}{dt} = 6 \text{ m/s}$$

The results are plotted in Fig. 12–13*b*. We can also obtain specific values of v by measuring the *slope* of the s–t graph at a given instant. For example, at $t = 20$ s, the slope of the s–t graph is determined from the straight line from 10 s to 30 s, i.e.,

$$t = 20 \text{ s}; \qquad v = \frac{\Delta s}{\Delta t} = \frac{150 \text{ m} - 30 \text{ m}}{30 \text{ s} - 10 \text{ s}} = 6 \text{ m/s}$$

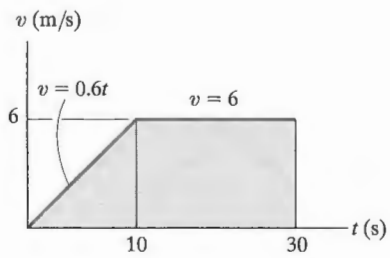
a–t Graph. Since $a = dv/dt$, the a–t graph can be determined by differentiating the equations defining the lines of the v–t graph. This yields

$$0 \le t < 10 \text{ s}; \qquad v = (0.6t) \text{ m/s} \qquad a = \frac{dv}{dt} = 0.6 \text{ m/s}^2$$

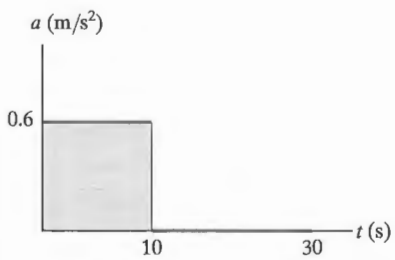
$$10 < t \le 30 \text{ s}; \qquad v = 6 \text{ m/s} \qquad a = \frac{dv}{dt} = 0$$

The results are plotted in Fig. 12–13*c*.

NOTE: Show that $a = 0.6 \text{ m/s}^2$ when $t = 5$ s by measuring the slope of the v–t graph.



(b)

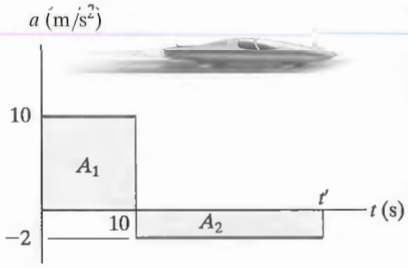


(c)

Fig. 12–13

EXAMPLE 12.7

The car in Fig. 12–14*a* starts from rest and travels along a straight track such that it accelerates at 10 m/s^2 for 10 s, and then decelerates at 2 m/s^2. Draw the *v–t* and *s–t* graphs and determine the time t' needed to stop the car. How far has the car traveled?



(a)

SOLUTION

v–t Graph. Since $dv = a\,dt$, the *v–t* graph is determined by integrating the straight-line segments of the *a–t* graph. Using the *initial condition* $v = 0$ when $t = 0$, we have

$$0 \le t < 10\text{ s}; \quad a = (10)\text{ m/s}^2; \quad \int_0^v dv = \int_0^t 10\,dt, \qquad v = 10t$$

When $t = 10$ s, $v = 10(10) = 100$ m/s. Using this as the *initial condition* for the next time period, we have

$$10\text{ s} < t \le t'; \; a = (-2)\text{ m/s}^2; \; \int_{100\text{ m/s}}^v dv = \int_{10\text{ s}}^t -2\,dt, \; v = (-2t + 120)\text{ m/s}$$

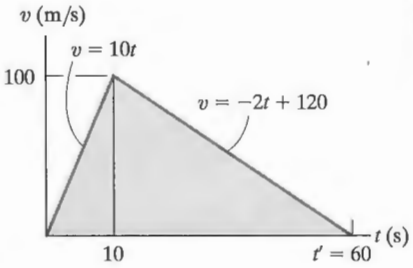
When $t = t'$ we require $v = 0$. This yields, Fig. 12–14*b*,

$$t' = 60\text{ s} \qquad \textit{Ans.}$$

A more direct solution for t' is possible by realizing that the area under the *a–t* graph is equal to the change in the car's velocity. We require $\Delta v = 0 = A_1 + A_2$, Fig. 12–14*a*. Thus

$$0 = 10\text{ m/s}^2(10\text{ s}) + (-2\text{ m/s}^2)(t' - 10\text{ s})$$

$$t' = 60\text{ s} \qquad \textit{Ans.}$$



(b)

s–t Graph. Since $ds = v\,dt$, integrating the equations of the *v–t* graph yields the corresponding equations of the *s–t* graph. Using the *initial condition* $s = 0$ when $t = 0$, we have

$$0 \le t \le 10\text{ s}; \quad v = (10t)\text{ m/s}; \quad \int_0^s ds = \int_0^t 10t\,dt, \qquad s = (5t^2)\text{ m}$$

When $t = 10$ s, $s = 5(10)^2 = 500$ m. Using this *initial condition*,

$$10\text{ s} \le t \le 60\text{ s}; \; v = (-2t + 120)\text{ m/s}; \; \int_{500\text{ m}}^s ds = \int_{10\text{ s}}^t (-2t + 120)\,dt$$

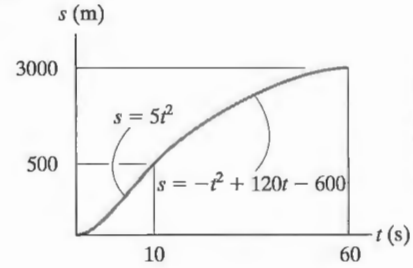
$$s - 500 = -t^2 + 120t - [-(10)^2 + 120(10)]$$

$$s = (-t^2 + 120t - 600)\text{ m}$$

When $t' = 60$ s, the position is

$$s = -(60)^2 + 120(60) - 600 = 3000\text{ m} \qquad \textit{Ans.}$$

The *s–t* graph is shown in Fig. 12–14*c*.



(c)

Fig. 12–14

NOTE: A direct solution for s is possible when $t' = 60$ s, since the *triangular area* under the *v–t* graph would yield the displacement $\Delta s = s - 0$ from $t = 0$ to $t' = 60$ s. Hence,

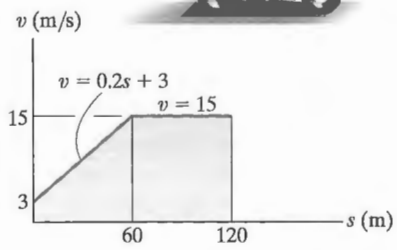
$$\Delta s = \tfrac{1}{2}(60\text{ s})(100\text{ m/s}) = 3000\text{ m} \qquad \textit{Ans.}$$

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

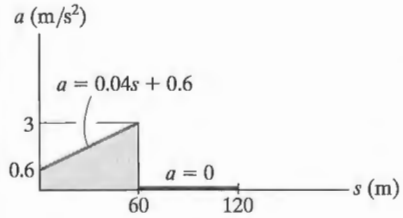
12

EXAMPLE 12.8

The v–s graph describing the motion of a motorcycle is shown in Fig. 12–15a. Construct the a–s graph of the motion and determine the time needed for the motorcycle to reach the position $s = 120$ m.



(a)



(b)

Fig. 12–15

SOLUTION

a–s Graph. Since the equations for segments of the v–s graph are given, the a–s graph can be determined using $a\,ds = v\,dv$.

$0 \le s < 60$ m; $\qquad v = (0.2s + 3)$ m/s

$$a = v\frac{dv}{ds} = (0.2s + 3)\frac{d}{ds}(0.2s + 3) = 0.04s + 0.6$$

$60 \text{ m} < s \le 120$ m; $\qquad v = 15$ m/s

$$a = v\frac{dv}{ds} = (15)\frac{d}{ds}(15) = 0$$

The results are plotted in Fig. 12–15b.

Time. The time can be obtained using the v–s graph and $v = ds/dt$, because this equation relates v, s, and t. For the first segment of motion, $s = 0$ when $t = 0$, so

$0 \le s < 60$ m; $\qquad v = (0.2s + 3)$ m/s; $\qquad dt = \dfrac{ds}{v} = \dfrac{ds}{0.2s + 3}$

$$\int_0^t dt = \int_0^s \frac{ds}{0.2s + 3}$$

$$t = (5\ln(0.2s + 3) - 5\ln 3)\text{ s}$$

At $s = 60$ m, $t = 5\ln[0.2(60) + 3] - 5\ln 3 = 8.05$ s. Therefore using these initial conditions for the second segment of motion,

$60 \text{ m} < s \le 120$ m; $\qquad v = 15$ m/s; $\qquad dt = \dfrac{ds}{v} = \dfrac{ds}{15}$

$$\int_{8.05\text{ s}}^t dt = \int_{60\text{ m}}^s \frac{ds}{15};$$

$$t - 8.05 = \frac{s}{15} - 4; \quad t = \left(\frac{s}{15} + 4.05\right)\text{ s}$$

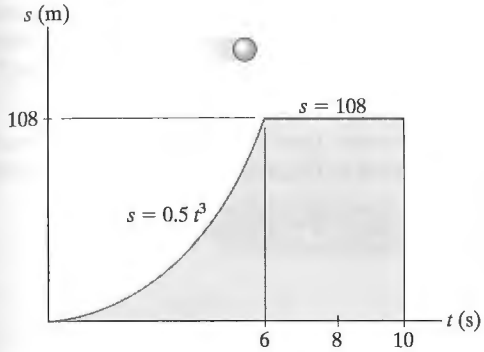
Therefore, at $s = 120$ m,

$$t = \frac{120}{15} + 4.05 = 12.05\text{ s} \qquad \textit{Ans.}$$

NOTE: The graphical results can be checked in part by calculating slopes. For example, at $s = 0$, $a = v(dv/ds) = 3(15 - 3)/60 = 0.6$ m/s². Also, the results can be checked in part by inspection. The v–s graph indicates the initial increase in velocity (acceleration) followed by constant velocity ($a = 0$).

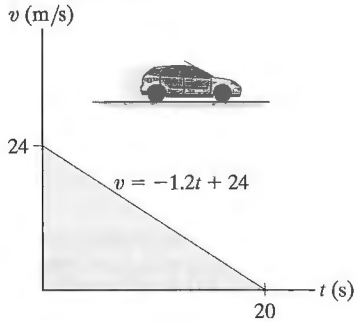
FUNDAMENTAL PROBLEMS

F12–9. The particle travels along a straight track such that its position is described by the $s-t$ graph. Construct the $v-t$ graph for the same time interval.



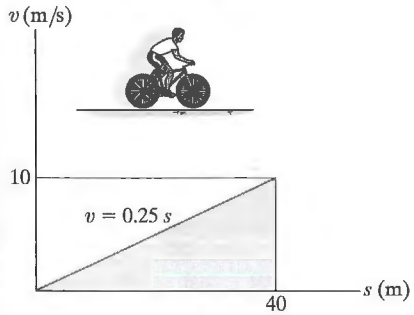
F12–9

F12–10. A van travels along a straight road with a velocity described by the graph. Construct the $s-t$ and $a-t$ graphs during the same period. Take $s = 0$ when $t = 0$.



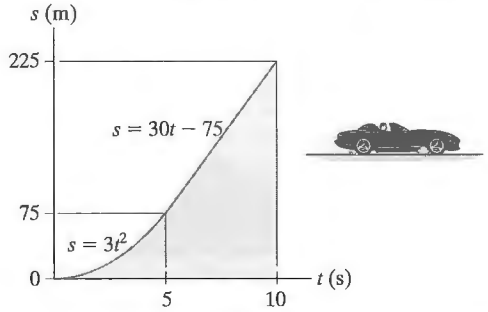
F12–10

F12–11. A bicycle travels along a straight road where its velocity is described by the $v-s$ graph. Construct the $a-s$ graph for the same time interval.



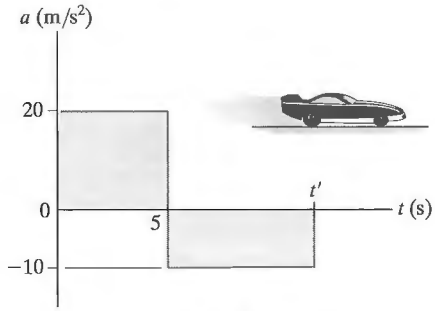
F12–11

F12–12. The sports car travels along a straight road such that its position is described by the graph. Construct the $v-t$ and $a-t$ graphs for the time interval $0 \le t \le 10$ s.



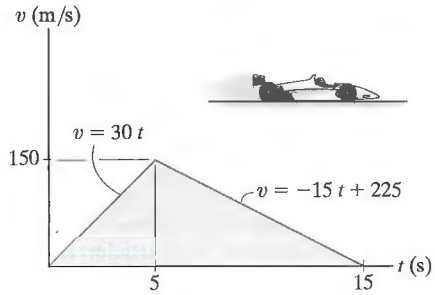
F12–12

F12–13. The dragster starts from rest and has an acceleration described by the graph. Construct the $v-t$ graph for the time interval $0 \le t \le t'$, where t' is the time for the car to come to rest.



F12–13

F12–14. The dragster starts from rest and has a velocity described by the graph. Construct the $s-t$ graph during the time interval $0 \le t \le 15$ s. Also, determine the total distance traveled during this time interval.



F12–14

MasteringENGINEERING
Customized coaching is available!

12

PROBLEMS

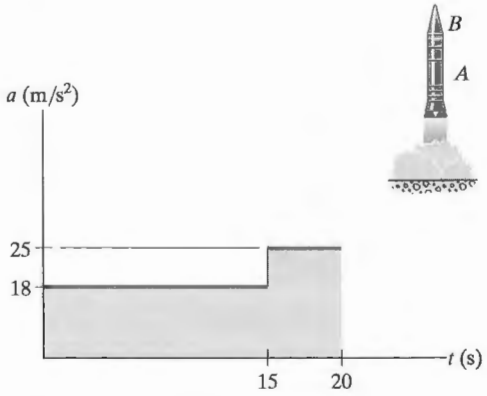
12–42. The speed of a train during the first minute has been recorded as follows.

Video Solutions

t (s)	0	20	40	60
v (m/s)	0	16	21	24

Plot the v–t graph, approximating the curve as straight-line segments between the given points. Determine the total distance traveled.

12–43. A two-stage missile is fired vertically from rest with the acceleration shown. In 15 s the first stage A burns out and the second stage B ignites. Plot the v–t and s–t graphs which describe the two-stage motion of the missile for $0 \le t \le 20$ s.



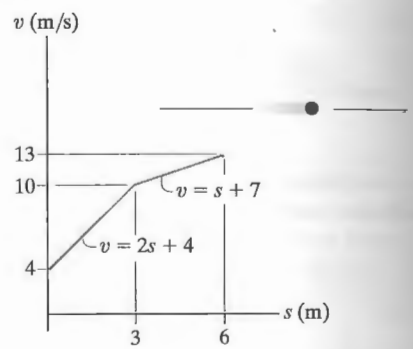
Prob. 12–43

***12–44.** A freight train starts from rest and travels with a constant acceleration of 0.15 m/s². After a time t' it maintains a constant speed so that when $t = 160$ s it has traveled 600 m. Determine the time t' and draw the v–t graph for the motion.

•12–45. If the position of a particle is defined by $s = [2 \sin(\pi/5)t + 4]$ m, where t is in seconds, construct the s–t, v–t, and a–t graphs for $0 \le t \le 10$ s.

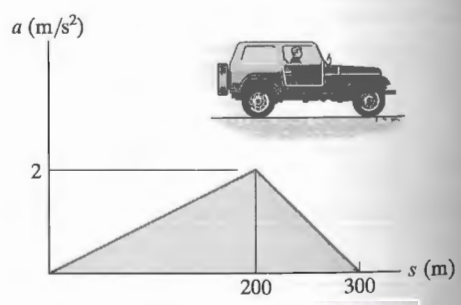
12–46. A train starts from station A and for the first kilometer, it travels with a uniform acceleration. Then, for the next two kilometers, it travels with a uniform speed. Finally, the train decelerates uniformly for another kilometer before coming to rest at station B. If the time for the whole journey is six minutes, draw the v–t graph and determine the maximum speed of the train.

12–47. The particle travels along a straight line with the velocity described by the graph. Construct the a–s graph.



Prob. 12–47

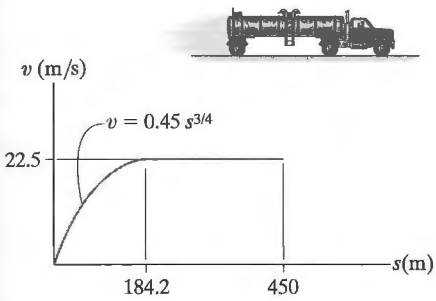
***12–48.** The a–s graph for a jeep traveling along a straight road is given for the first 300 m of its motion. Construct the v–s graph. At $s = 0$, $v = 0$.



Prob. 12–48

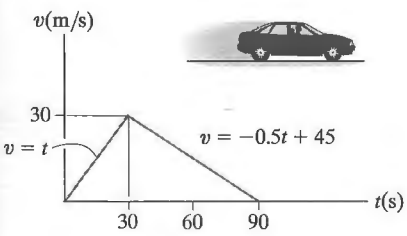
12–49. A particle travels along a curve defined by the equation $s = (t^3 - 3t^2 + 2t)$ m, where t is in seconds. Draw the $s - t$, $v - t$, and $a - t$ graphs for the particle for $0 \le t \le 3$ s.

12–50. A truck is traveling along the straight line with a velocity described by the graph. Construct the $a-s$ graph for $0 \le s \le 450$ m.



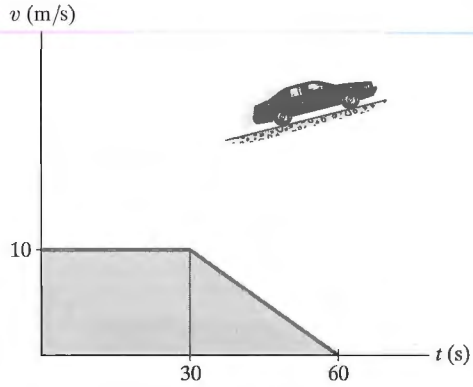
Prob. 12–50

12–51. A car starts from rest and travels along a straight road with a velocity described by the graph. Determine the total distance traveled until the car stops. Construct the s–t and a–t graphs.



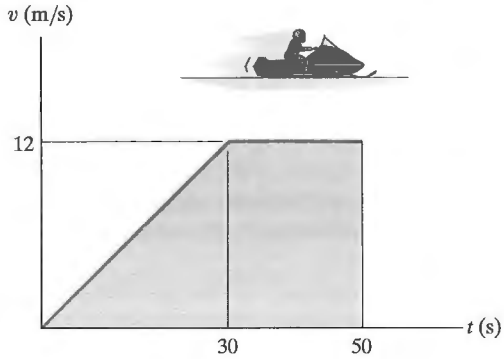
Prob. 12–51

***12–52.** A car travels up a hill with the speed shown. Determine the total distance the car travels until it stops ($t = 60$ s). Plot the $a-t$ graph.



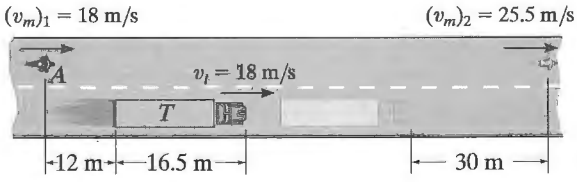
Prob. 12–52

•12–53. The snowmobile moves along a straight course according to the v–t graph. Construct the s–t and a–t graphs for the same 50-s time interval. When $t = 0$, $s = 0$.



Prob. 12–53

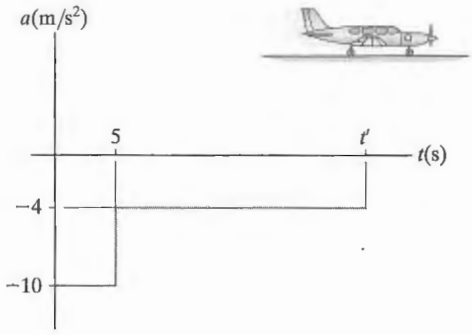
12–54. A motorcyclist at A is traveling at 18 m/s when he wishes to pass the truck T which is traveling at a constant speed of 18 m/s. To do so the motorcyclist accelerates at 1.8 m/s^2 until reaching a maximum speed of 25.5 m/s. If he then maintains this speed, determine the time needed for him to reach a point located 30 m in front of the truck. Draw the $v-t$ and $s-t$ graphs for the motorcycle during this time.



Prob. 12–54

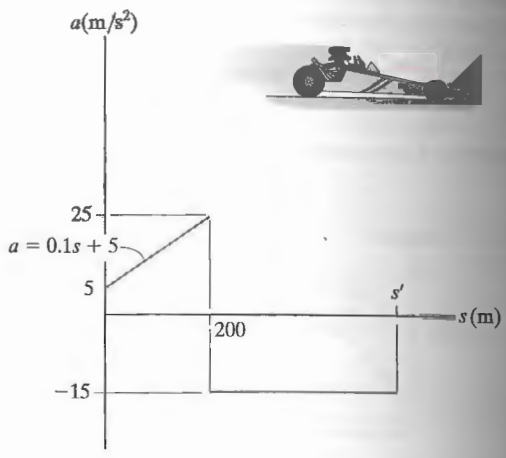
12

12–55. An airplane traveling at 70 m/s lands on a straight runway and has a deceleration described by the graph. Determine the time t' and the distance traveled for it to reach a speed of 5 m/s. Construct the v–t and s–t graphs for this time interval, $0 \le t \le t'$.



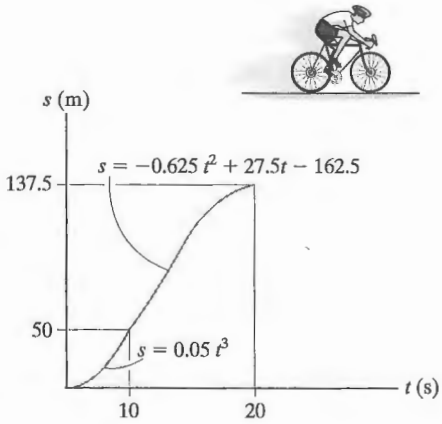
Prob. 12–55

•12–57. The dragster starts from rest and travels along a straight track with an acceleration-deceleration described by the graph. Construct the v–s graph for $0 \le s \le s'$, and determine the distance s' traveled before the dragster again comes to rest.



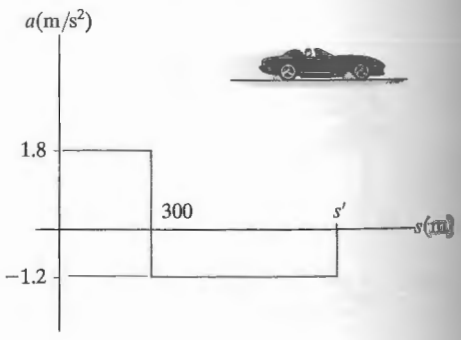
Prob. 12–57

***12–56.** The position of a cyclist traveling along a straight road is described by the graph. Construct the v–t and a–t graphs.



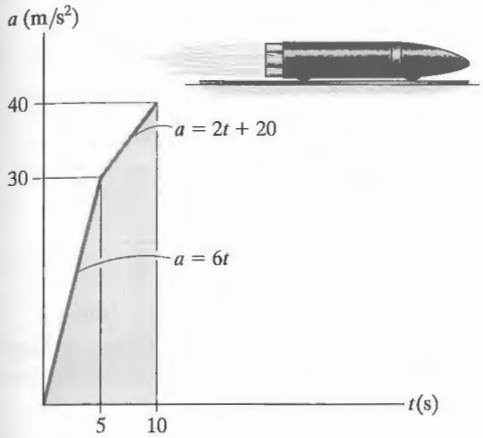
Prob. 12–56

12–58. A sports car travels along a straight road with an acceleration-deceleration described by the graph. If the car starts from rest, determine the distance s' the car travels until it stops. Construct the v–s graph for $0 \le s \le s'$.



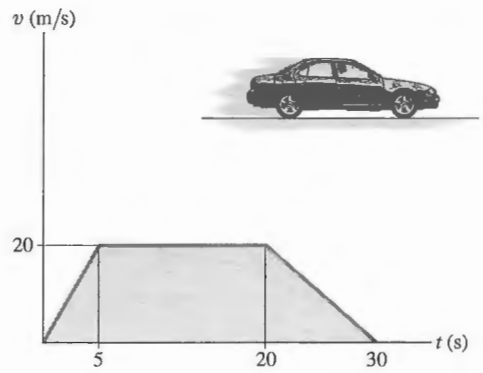
Prob. 12–58

12–59. A missile starting from rest travels along a straight track and for 10 s has an acceleration as shown. Draw the $v-t$ graph that describes the motion and find the distance traveled in 10 s.



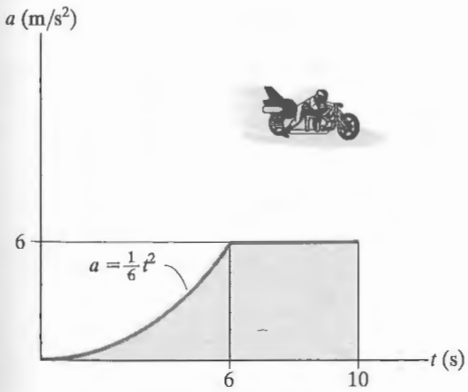
Prob. 12–59

•**12–61.** The $v-t$ graph of a car while traveling along a road is shown. Draw the $s-t$ and $a-t$ graphs for the motion.



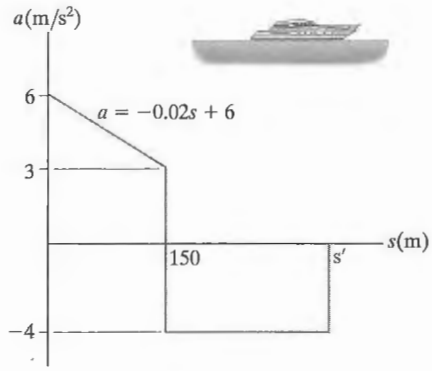
Prob. 12–61

***12–60.** A motorcyclist starting from rest travels along a straight road and for 10 s has an acceleration as shown. Draw the $v-t$ graph that describes the motion and find the distance traveled in 10 s.



Prob. 12–60

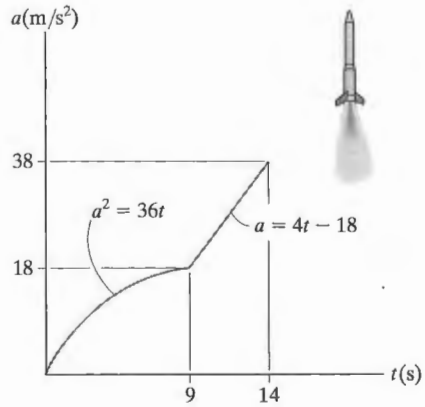
12–62. The boat travels in a straight line with the acceleration described by the $a-s$ graph. If it starts from rest, construct the $v-s$ graph and determine the boat's maximum speed. What distance s' does it travel before it stops?



Prob. 12–62

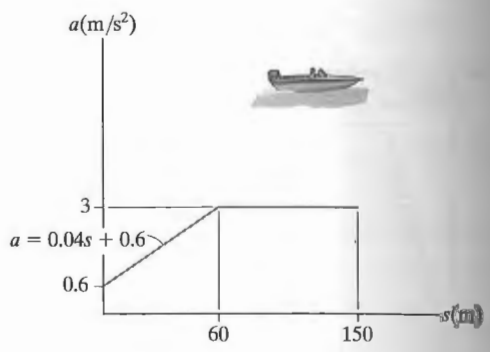
12

12–63. The rocket has an acceleration described by the graph. If it starts from rest, construct the $v-t$ and $s-t$ graphs for the motion for the time interval $0 \le t \le 14$ s.



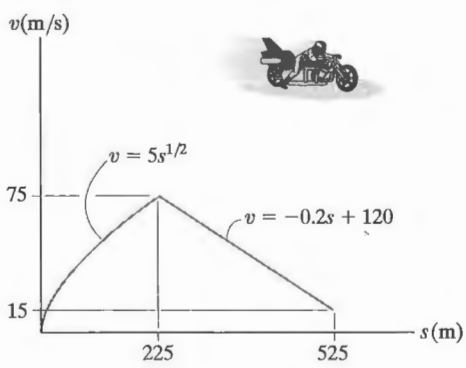
Prob. 12–63

•**12–65.** The acceleration of the speed boat starting from rest is described by the graph. Construct the $v-s$ graph.



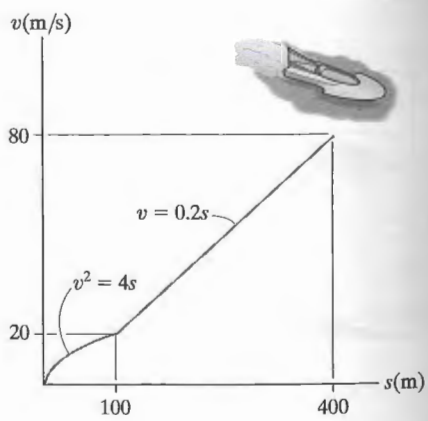
Prob. 12–65

***12–64.** The jet bike is moving along a straight road with the speed described by the $v-s$ graph. Construct the $a-s$ graph.



Prob. 12–64

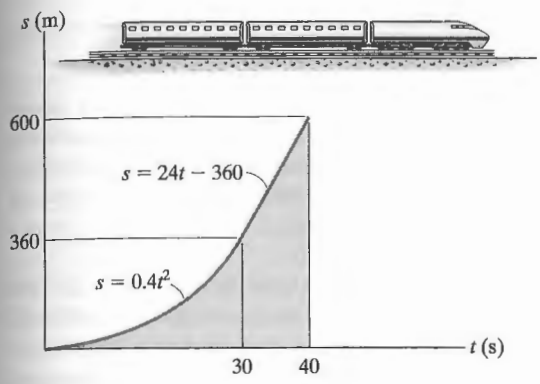
12–66. The boat travels along a straight line with the speed described by the graph. Construct the $s-t$ and $a-s$ graphs. Also, determine the time required for the boat to travel a distance $s = 400$ m if $s = 0$ when $t = 0$.



Prob. 12–66

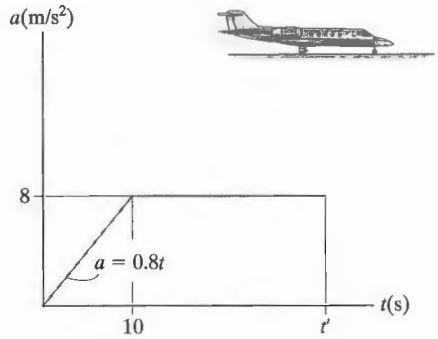
m Pearson Custom Publishing exclusively for The University of Melbourne

12–67. The s–t graph for a train has been determined experimentally. From the data, construct the v–t and a–t graphs for the motion.



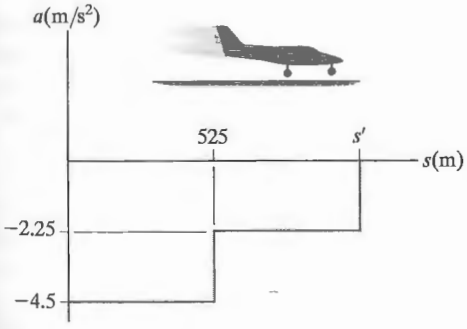
Prob. 12–67

•12–69. The airplane travels along a straight runway with an acceleration described by the graph. If it starts from rest and requires a velocity of 90 m/s to take off, determine the minimum length of runway required and the time t' for take off. Construct the v–t and s–t graphs.



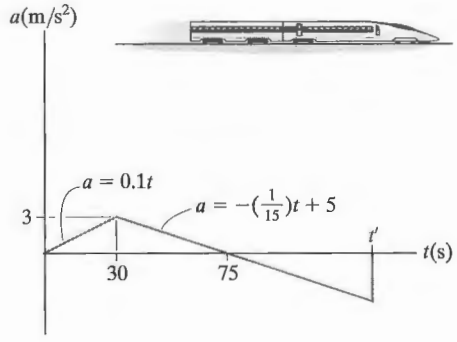
Prob. 12–69

***12–68.** The airplane lands at 75 m/s on a straight runway and has a deceleration described by the graph. Determine the distance s' traveled before its speed is decreased to 7.5 m/s. Draw the s–t graph.



Prob. 12–68

12–70. The a–t graph of the bullet train is shown. If the train starts from rest, determine the elapsed time t' before it again comes to rest. What is the total distance traveled during this time interval? Construct the v–t and s–t graphs.



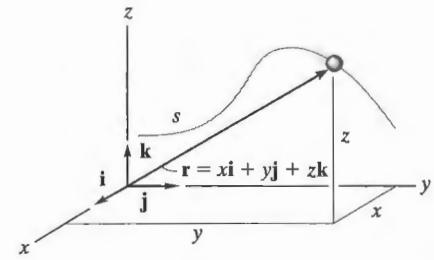
Prob. 12–70

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

12

12.5 Curvilinear Motion: Rectangular Components

Occasionally the motion of a particle can best be described along a path that can be expressed in terms of its x, y, z coordinates.



Position

(a)

Position. If the particle is at point (x, y, z) on the curved path s shown in Fig. 12–17a, then its location is defined by the *position vector*

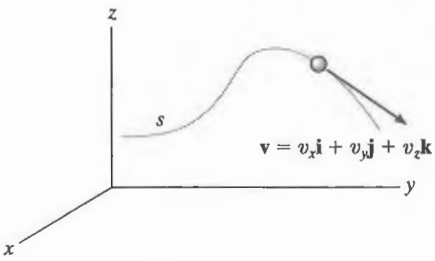
$$\boxed{\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}} \tag{12–10}$$

When the particle moves, the x, y, z components of $\mathbf{r}$ will be functions of time; i.e., $x = x(t)$, $y = y(t)$, $z = z(t)$, so that $\mathbf{r} = \mathbf{r}(t)$.

At any instant the *magnitude* of $\mathbf{r}$ is defined from Eq. C–3 in Appendix C as

$$r = \sqrt{x^2 + y^2 + z^2}$$

And the *direction* of $\mathbf{r}$ is specified by the unit vector $\mathbf{u}_r = \mathbf{r}/r$.



Velocity

(b)

Fig. 12–17

Velocity. The first time derivative of $\mathbf{r}$ yields the velocity of the particle. Hence,

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \frac{d}{dt}(x\mathbf{i}) + \frac{d}{dt}(y\mathbf{j}) + \frac{d}{dt}(z\mathbf{k})$$

When taking this derivative, it is necessary to account for changes in *both* the magnitude and direction of each of the vector's components. For example, the derivative of the $\mathbf{i}$ component of $\mathbf{r}$ is

$$\frac{d}{dt}(x\mathbf{i}) = \frac{dx}{dt}\mathbf{i} + x\frac{d\mathbf{i}}{dt}$$

The second term on the right side is zero, provided the x, y, z reference frame is *fixed*, and therefore the *direction* (and the *magnitude*) of $\mathbf{i}$ does not change with time. Differentiation of the $\mathbf{j}$ and $\mathbf{k}$ components may be carried out in a similar manner, which yields the final result,

$$\boxed{\mathbf{v} = \frac{d\mathbf{r}}{dt} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}} \tag{12–11}$$

where

$$\boxed{v_x = \dot{x} \quad v_y = \dot{y} \quad v_z = \dot{z}} \tag{12–12}$$

The "dot" notation $\dot{x}$, $\dot{y}$, $\dot{z}$ represents the first time derivatives of $x = x(t)$, $y = y(t)$, $z = z(t)$, respectively.

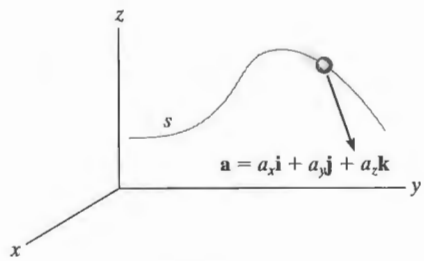
The velocity has a *magnitude* that is found from

$$v = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

and a *direction* that is specified by the unit vector $\mathbf{u}_v = \mathbf{v}/v$. As discussed in Sec. 12–4, this direction is *always tangent to the path*, as shown in Fig. 12–17*b*.

Acceleration. The acceleration of the particle is obtained by taking the first time derivative of Eq. 12–11 (or the second time derivative of Eq. 12–10). We have

$$\boxed{\mathbf{a} = \frac{d\mathbf{v}}{dt} = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}} \qquad (12\text{–}13)$$



Acceleration

(c)

Fig. 12–17 (cont.)

where

$$\boxed{\begin{aligned} a_x &= \dot{v}_x = \ddot{x} \\ a_y &= \dot{v}_y = \ddot{y} \\ a_z &= \dot{v}_z = \ddot{z} \end{aligned}} \qquad (12\text{–}14)$$

Here a_x, a_y, a_z represent, respectively, the first time derivatives of $v_x = v_x(t)$, $v_y = v_y(t)$, $v_z = v_z(t)$, or the second time derivatives of the functions $x = x(t)$, $y = y(t)$, $z = z(t)$.

The acceleration has a *magnitude*

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

and a *direction* specified by the unit vector $\mathbf{u}_a = \mathbf{a}/a$. Since $\mathbf{a}$ represents the time rate of *change* in both the magnitude and direction of the velocity, in general $\mathbf{a}$ will *not* be tangent to the path, Fig. 12–17*c*.

Important Points

- Curvilinear motion can cause changes in *both* the magnitude and direction of the position, velocity, and acceleration vectors.
- The velocity vector is always directed *tangent* to the path.
- In general, the acceleration vector is *not* tangent to the path, but rather, it is tangent to the hodograph.
- If the motion is described using rectangular coordinates, then the components along each of the axes do not change direction, only their magnitude and sense (algebraic sign) will change.
- By considering the component motions, the change in magnitude and direction of the particle's position and velocity are automatically taken into account.

Procedure for Analysis

Coordinate System.

- A rectangular coordinate system can be used to solve problems for which the motion can conveniently be expressed in terms of its x, y, z components.

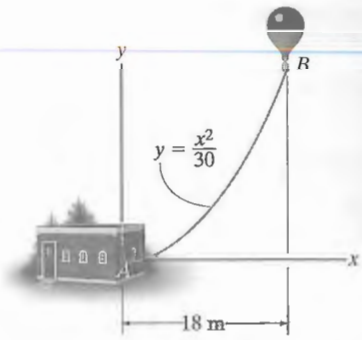
Kinematic Quantities.

- Since *rectilinear motion* occurs along *each coordinate axis*, the motion along each axis is found using $v = ds/dt$ and $a = dv/dt$; or in cases where the motion is not expressed as a function of time, the equation $a\,ds = v\,dv$ can be used.
- In two dimensions, the equation of the path $y = f(x)$ can be used to relate the x and y components of velocity and acceleration by applying the chain rule of calculus. A review of this concept is given in Appendix C.
- Once the x, y, z components of **v** and **a** have been determined, the magnitudes of these vectors are found from the Pythagorean theorem, Eq. B–3, and their coordinate direction angles from the components of their unit vectors, Eqs. B–4 and B–5.

A publication from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 12.9

At any instant the horizontal position of the weather balloon in Fig. 12–18*a* is defined by $x = (9t)$ m, where t is in seconds. If the equation of the path is $y = x^2/30$, determine the magnitude and direction of the velocity and the acceleration when $t = 2$ s.



(a)

SOLUTION

Velocity. The velocity component in the x direction is

$$v_x = \dot{x} = \frac{d}{dt}(9t) = 9 \text{ m/s} \rightarrow$$

To find the relationship between the velocity components we will use the chain rule of calculus. (See Appendix A for a full explanation.)

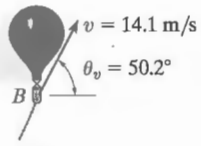
$$v_y = \dot{y} = \frac{d}{dt}(x^2/30) = 2x\dot{x}/30 = 2(18)(9)/30 = 10.8 \text{ m/s} \uparrow$$

When $t = 2$ s, the magnitude of velocity is therefore

$$v = \sqrt{(9 \text{ m/s})^2 + (10.8 \text{ m/s})^2} = 14.1 \text{ m/s} \qquad \textit{Ans.}$$

The direction is tangent to the path, Fig. 12–18*b*, where

$$\theta_v = \tan^{-1}\frac{v_y}{v_x} = \tan^{-1}\frac{10.8}{9} = 50.2° \qquad \textit{Ans.}$$



(b)

Acceleration. The relationship between the acceleration components is determined using the chain rule. (See Appendix C.) We have

$$a_x = \dot{v}_x = \frac{d}{dt}(9) = 0$$

$$a_y = \dot{v}_y = \frac{d}{dt}(2x\dot{x}/30) = 2(\dot{x})\dot{x}/30 + 2x(\ddot{x})/30$$

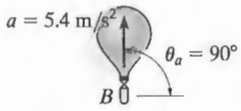
$$= 2(9)^2/30 + 2(18)(0)/30 = 5.4 \text{ m/s}^2 \uparrow$$

Thus,

$$a = \sqrt{(0)^2 + (5.4)^2} = 5.4 \text{ m/s}^2 \qquad \textit{Ans.}$$

The direction of **a**, as shown in Fig. 12–18*c*, is

$$\theta_a = \tan^{-1}\frac{5.4}{0} = 90° \qquad \textit{Ans.}$$



(c)

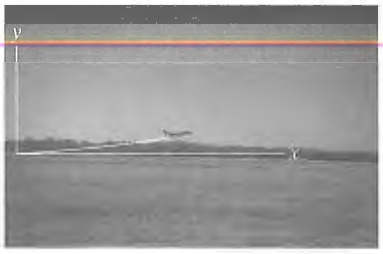
Fig. 12–18

NOTE: It is also possible to obtain v_y and a_y by first expressing $y = f(t) = (9t)^2/30 = 2.7t^2$ and then taking successive time derivatives.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

12

EXAMPLE 12.10



For a short time, the path of the plane in Fig. 12–19*a* is described by $y = (0.001x^2)$ m. If the plane is rising with a constant velocity of 10 m/s, determine the magnitudes of the velocity and acceleration of the plane when it is at $y = 100$ m.

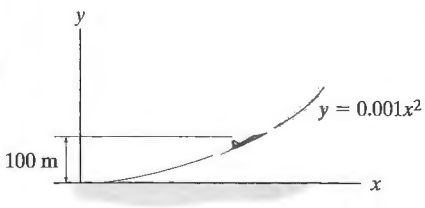
SOLUTION

When $y = 100$ m, then $100 = 0.001x^2$ or $x = 316.2$ m. Also, since $v_y = 10$ m/s, then

$$y = v_y t; \qquad 100 \text{ m} = (10 \text{ m/s})\, t \qquad t = 10 \text{ s}$$

Velocity. Using the chain rule (see Appendix C) to find the relationship between the velocity components, we have

$$v_y = \dot{y} = \frac{d}{dt}(0.001x^2) = (0.002x)\dot{x} = 0.002xv_x \qquad (1)$$



(a)

Thus

$$10 \text{ m/s} = 0.002(316.2 \text{ m})(v_x)$$
$$v_x = 15.81 \text{ m/s}$$

The magnitude of the velocity is therefore

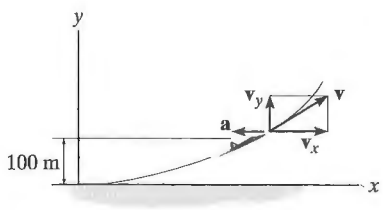
$$v = \sqrt{v_x^2 + v_y^2} = \sqrt{(15.81 \text{ m/s})^2 + (10 \text{ m/s})^2} = 18.7 \text{ m/s} \qquad \textit{Ans.}$$

Acceleration. Using the chain rule, the time derivative of Eq. (1) gives the relation between the acceleration components.

$$a_y = \dot{v}_y = 0.002\dot{x}v_x + 0.002x\dot{v}_x = 0.002(v_x^2 + xa_x)$$

When $x = 316.2$ m, $v_x = 15.81$ m/s, $\dot{v}_y = a_y = 0$,

$$0 = 0.002((15.81 \text{ m/s})^2 + 316.2 \text{ m}(a_x))$$
$$a_x = -0.791 \text{ m/s}^2$$



(b)

Fig. 12–19

The magnitude of the plane's acceleration is therefore

$$a = \sqrt{a_x^2 + a_y^2} = \sqrt{(-0.791 \text{ m/s}^2)^2 + (0 \text{ m/s}^2)^2}$$
$$= 0.791 \text{ m/s}^2 \qquad \textit{Ans.}$$

These results are shown in Fig. 12–19*b*.

12.6 Motion of a Projectile

The free-flight motion of a projectile is often studied in terms of its rectangular components. To illustrate the kinematic analysis, consider a projectile launched at point (x_0, y_0), with an initial velocity of $\mathbf{v}_0$, having components $(\mathbf{v}_0)_x$ and $(\mathbf{v}_0)_y$, Fig. 12–20. When air resistance is neglected, the only force acting on the projectile is its weight, which causes the projectile to have a *constant downward acceleration* of approximately $a_c = g = 9.81\ \text{m/s}^2$ or $g = 32.2\ \text{ft/s}^2$.*

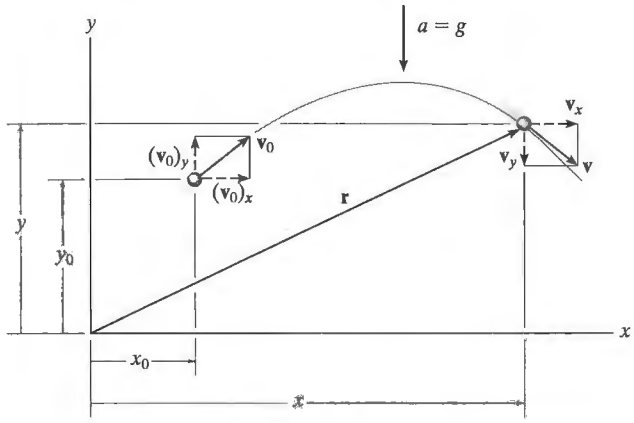


Fig. 12–20

Horizontal Motion. Since $a_x = 0$, application of the constant acceleration equations, 12–4 to 12–6, yields

$$(\overset{+}{\rightarrow}) \qquad v = v_0 + a_c t; \qquad v_x = (v_0)_x$$

$$(\overset{+}{\rightarrow}) \qquad x = x_0 + v_0 t + \tfrac{1}{2} a_c t^2; \qquad x = x_0 + (v_0)_x t$$

$$(\overset{+}{\rightarrow}) \qquad v^2 = v_0^2 + 2a_c(x - x_0); \qquad v_x = (v_0)_x$$

The first and last equations indicate that *the horizontal component of velocity always remains constant during the motion.*

Vertical Motion. Since the positive y axis is directed upward, then $a_y = -g$. Applying Eqs. 12–4 to 12–6, we get

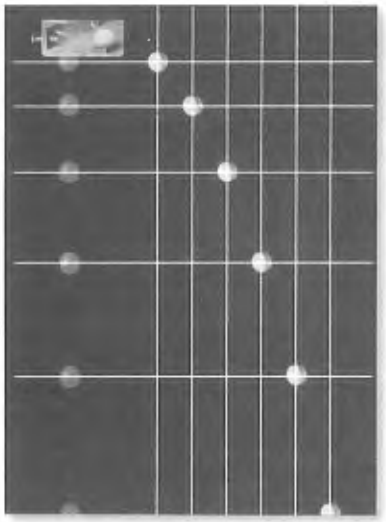
$$(+\uparrow) \qquad v = v_0 + a_c t; \qquad v_y = (v_0)_y - gt$$

$$(+\uparrow) \qquad y = y_0 + v_0 t + \tfrac{1}{2} a_c t^2; \qquad y = y_0 + (v_0)_y t - \tfrac{1}{2} g t^2$$

$$(+\uparrow) \qquad v^2 = v_0^2 + 2a_c(y - y_0); \qquad v_y^2 = (v_0)_y^2 - 2g(y - y_0)$$

Recall that the last equation can be formulated on the basis of eliminating the time t from the first two equations, and therefore *only two of the above three equations are independent of one another.*

* This assumes that the earth's gravitational field does not vary with altitude.



Each picture in this sequence is taken after the same time interval. The red ball falls from rest, whereas the yellow ball is given a horizontal velocity when released. Both balls accelerate downward at the same rate, and so they remain at the same elevation at any instant. This acceleration causes the difference in elevation between the balls to increase between successive photos. Also, note the horizontal distance between successive photos of the yellow ball is constant since the velocity in the horizontal direction remains constant.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

To summarize, problems involving the motion of a projectile can have at most three unknowns since only three independent equations can be written; that is, *one* equation in the *horizontal direction* and *two* in the *vertical direction*. Once $\mathbf{v}_x$ and $\mathbf{v}_y$ are obtained, the resultant velocity $\mathbf{v}$, which is *always tangent* to the path, can be determined by the *vector sum* as shown in Fig. 12–20.

Procedure for Analysis

Coordinate System.

- Establish the fixed x, y coordinate axes and sketch the trajectory of the particle. Between any *two points* on the path specify the given problem data and identify the *three unknowns*. In all cases the acceleration of gravity acts downward and equals 9.81 m/s^2. The particle's initial and final velocities should be represented in terms of their x and y components.
- Remember that positive and negative position, velocity, and acceleration components always act in accordance with their associated coordinate directions.

Kinematic Equations.

- Depending upon the known data and what is to be determined, a choice should be made as to which three of the following four equations should be applied between the two points on the path to obtain the most direct solution to the problem.

Horizontal Motion.

- The *velocity* in the horizontal or x direction is *constant*, i.e., $v_x = (v_0)_x$, and

$$x = x_0 + (v_0)_x t$$

Vertical Motion.

- In the vertical or y direction *only two* of the following three equations can be used for solution.

$$v_y = (v_0)_y + a_c t$$

$$y = y_0 + (v_0)_y t + \tfrac{1}{2} a_c t^2$$

$$v_y^2 = (v_0)_y^2 + 2a_c(y - y_0)$$

For example, if the particle's final velocity v_y is not needed, then the first and third of these equations will not be useful.

Gravel falling off the end of this conveyor belt follows a path that can be predicted using the equations of constant acceleration. In this way the location of the accumulated pile can be determined. Rectangular coordinates are used for the analysis since the acceleration is only in the vertical direction.

From Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 12.11

A sack slides off the ramp, shown in Fig. 12–21, with a horizontal velocity of 12 m/s. If the height of the ramp is 6 m from the floor, determine the time needed for the sack to strike the floor and the range R where sacks begin to pile up.

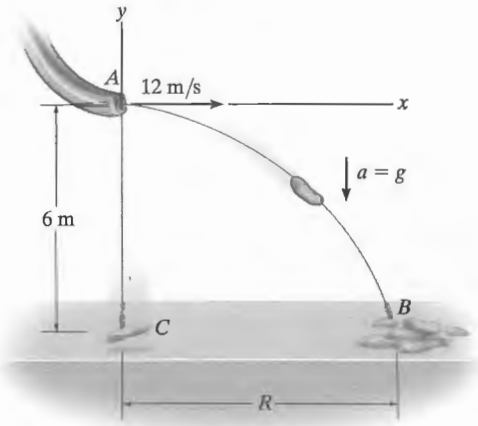


Fig. 12–21

SOLUTION

Coordinate System. The origin of coordinates is established at the beginning of the path, point A, Fig. 12–21. The initial velocity of a sack has components $(v_A)_x = 12$ m/s and $(v_A)_y = 0$. Also, between points A and B the acceleration is $a_y = -9.81$ m/s^2. Since $(v_B)_x = (v_A)_x = 12$ m/s, the three unknowns are $(v_B)_y$, R, and the time of flight t_{AB}. Here we do not need to determine $(v_B)_y$.

Vertical Motion. The vertical distance from A to B is known, and therefore we can obtain a direct solution for t_{AB} by using the equation

$$(+\uparrow) \qquad y_B = y_A + (v_A)_y t_{AB} + \tfrac{1}{2} a_c t_{AB}^2$$

$$-6\text{ m} = 0 + 0 + \tfrac{1}{2}(-9.81\text{ m/s}^2)t_{AB}^2$$

$$t_{AB} = 1.11\text{ s} \qquad \textit{Ans.}$$

Horizontal Motion. Since t_{AB} has been calculated, R is determined as follows:

$$(\overset{+}{\rightarrow}) \qquad x_B = x_A + (v_A)_x t_{AB}$$

$$R = 0 + 12\text{ m/s}\,(1.11\text{ s})$$

$$R = 13.3\text{ m} \qquad \textit{Ans.}$$

NOTE: The calculation for t_{AB} also indicates that if a sack were released *from rest* at A, it would take the same amount of time to strike the floor at C, Fig. 12–21.

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

ENGR10003 – ENGINEERING SYSTEMS DESIGN 2
A publication from Pearson Custom Publishing exclusively for Melbourne

12

EXAMPLE 12.12

The chipping machine is designed to eject wood chips at $v_O = 7.5$ m/s as shown in Fig. 12–22. If the tube is oriented at 30° from the horizontal, determine how high, h, the chips strike the pile if at this instant they land on the pile 6 m from the tube.

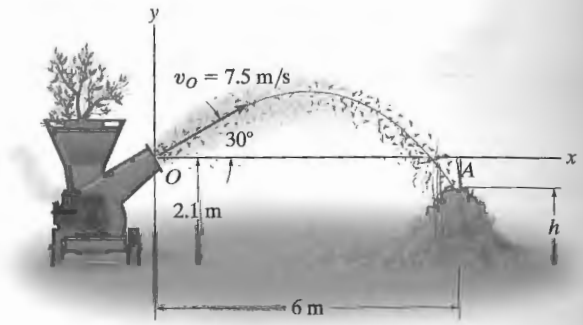


Fig. 12–22

SOLUTION

Coordinate System. When the motion is analyzed between points O and A, the three unknowns are the height h, time of flight t_{OA}, and vertical component of velocity $(v_A)_y$. [Note that $(v_A)_x = (v_O)_x$.] With the origin of coordinates at O, Fig. 12–22, the initial velocity of a chip has components of

$$(v_O)_x = (7.5 \cos 30°)\text{ m/s} = 6.5\text{ m/s} \rightarrow$$

$$(v_O)_y = (7.5 \sin 30°)\text{ m/s} = 3.75\text{ m/s} \uparrow$$

Also, $(v_A)_x = (v_O)_x = 6.5$ m/s and $a_y = -9.81$ m/s^2. Since we do not need to determine $(v_A)_y$, we have

Horizontal Motion.

$(\overset{+}{\rightarrow})$

$$x_A = x_O + (v_O)_x t_{OA}$$

$$6\text{ m} = 0 + (6.5\text{ m/s})t_{OA}$$

$$t_{OA} = 0.9231\text{ s}$$

Vertical Motion. Relating t_{OA} to the initial and final elevations of a chip, we have

$(+\uparrow)$ $\quad y_A = y_O + (v_O)_y t_{OA} + \frac{1}{2}a_y t_{OA}^2$

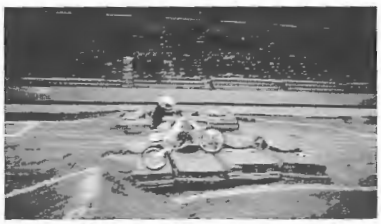
$$(h - 2.1\text{ m}) = 0 + (3.75\text{ m/s})(0.9231\text{ s}) + \tfrac{1}{2}(-9.81\text{ m/s}^2)(0.9231\text{ s})^2$$

$$h = 1.38\text{ m} \qquad \textit{Ans.}$$

NOTE: We can determine $(v_A)_y$ by using $(v_A)_y = (v_O)_y + a_y t_{OA}$.

EXAMPLE 12.13

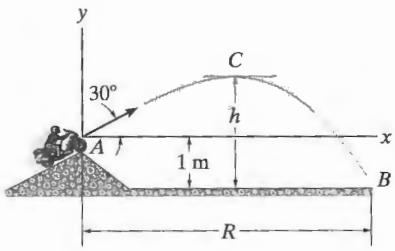
The track for this racing event was designed so that riders jump off the slope at 30°, from a height of 1 m. During a race it was observed that the rider shown in Fig. 12–23*a* remained in mid air for 1.5 s. Determine the speed at which he was traveling off the ramp, the horizontal distance he travels before striking the ground, and the maximum height he attains. Neglect the size of the bike and rider.



(a)

SOLUTION

Coordinate System. As shown in Fig. 12–23*b*, the origin of the coordinates is established at A. Between the end points of the path AB the three unknowns are the initial speed v_A, range R, and the vertical component of velocity $(v_B)_y$.



(b)

Fig. 12–23

Vertical Motion. Since the time of flight and the vertical distance between the ends of the path are known, we can determine v_A.

$$(+\uparrow) \qquad y_B = y_A + (v_A)_y t_{AB} + \tfrac{1}{2}a_c t_{AB}^2$$

$$-1\text{ m} = 0 + v_A \sin 30°(1.5\text{ s}) + \tfrac{1}{2}(-9.81\text{ m/s}^2)(1.5\text{ s})^2$$

$$v_A = 13.38\text{ m/s} = 13.4\text{ m/s} \qquad \textit{Ans.}$$

Horizontal Motion. The range R can now be determined.

$$(\overset{+}{\rightarrow}) \qquad x_B = x_A + (v_A)_x t_{AB}$$

$$R = 0 + 13.38\cos 30°\text{ m/s}(1.5\text{ s})$$

$$= 17.4\text{ m} \qquad \textit{Ans.}$$

In order to find the maximum height h we will consider the path AC, Fig. 12–23*b*. Here the three unknowns are the time of flight t_{AC}, the horizontal distance from A to C, and the height h. At the maximum height $(v_C)_y = 0$, and since v_A is known, we can determine h *directly* without considering t_{AC} using the following equation.

$$(v_C)_y^2 = (v_A)_y^2 + 2a_c[y_C - y_A]$$

$$0^2 = (13.38\sin 30°\text{ m/s})^2 + 2(-9.81\text{ m/s}^2)[(h - 1\text{ m}) - 0]$$

$$h = 3.28\text{ m} \qquad \textit{Ans.}$$

NOTE: Show that the bike will strike the ground at B with a velocity having components of

$$(v_B)_x = 11.6\text{ m/s}\rightarrow, \quad (v_B)_y = 8.02\text{ m/s}\downarrow$$

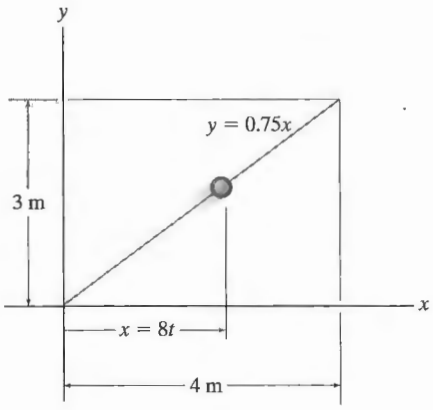
Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

12

FUNDAMENTAL PROBLEMS

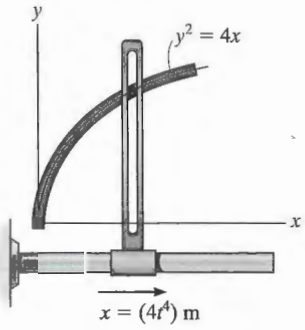
F12–15. If the x and y components of a particle's velocity are $v_x = (32t)$ m/s and $v_y = 8$ m/s, determine the equation of the path $y = f(x)$. $x = 0$ and $y = 0$ when $t = 0$.

F12–16. A particle is traveling along the straight path. If its position along the x axis is $x = (8t)$ m, where t is in seconds, determine its speed when $t = 2$ s.



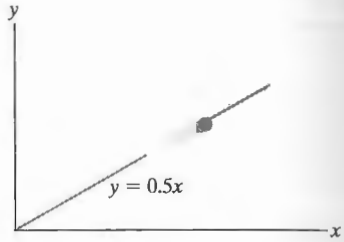
F12–16

F12–17. A particle is constrained to travel along the path. If $x = (4t^4)$ m, where t is in seconds, determine the magnitude of the particle's velocity and acceleration when $t = 0.5$ s.



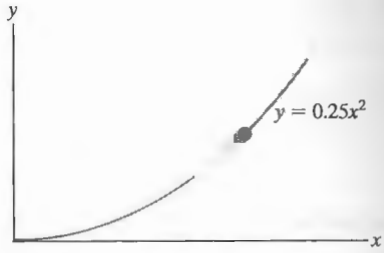
F12–17

F12–18. A particle travels along a straight-line path $y = 0.5x$. If the x component of the particle's velocity is $v_x = (2t^2)$ m/s, where t is in seconds, determine the magnitude of the particle's velocity and acceleration when $t = 4$ s.



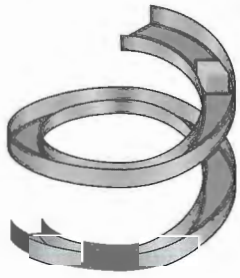
F12–18

F12–19. A particle is traveling along the parabolic path $y = 0.25x^2$. If $x = (2t^2)$ m, where t is in seconds, determine the magnitude of the particle's velocity and acceleration when $t = 2$ s.



F12–19

F12–20. The position of a box sliding down the spiral can be described by $\mathbf{r} = [2 \sin (2t)\mathbf{i} + 2 \cos t\mathbf{j} - 2t^2\mathbf{k}]$ m, where t is in seconds and the arguments for the sine and cosine are in radians. Determine the velocity and acceleration of the box when $t = 2$ s.

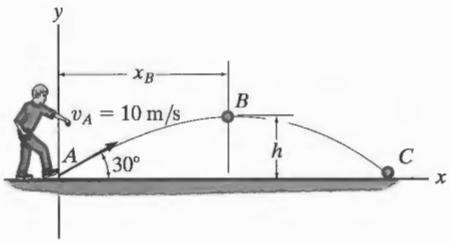


F12–20

from Pearson Custom Publishing exclusively for The University of Melbourne

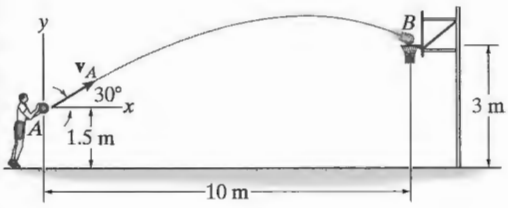
F12–21. The ball is kicked from point A with the initial velocity $v_A = 10$ m/s. Determine the maximum height h it reaches.

F12–22. The ball is kicked from point A with the initial velocity $v_A = 10$ m/s. Determine the range R, and the speed when the ball strikes the ground.



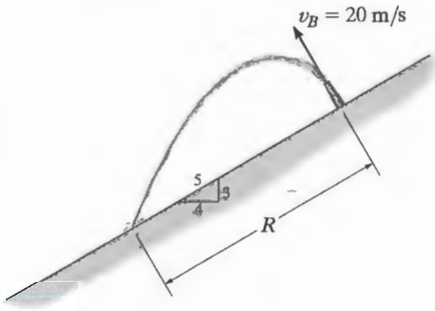
F12–21/22

F12–23. Determine the speed at which the basketball at A must be thrown at the angle of 30° so that it makes it to the basket at B.



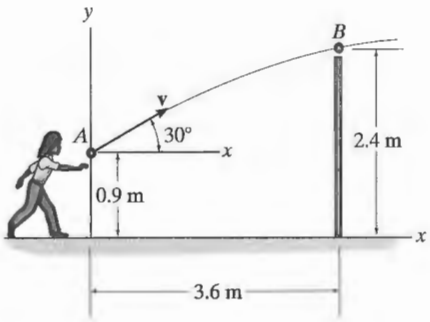
F12–23

F12–24. Water is sprayed at an angle of 90° from the slope at 20 m/s. Determine the range R.



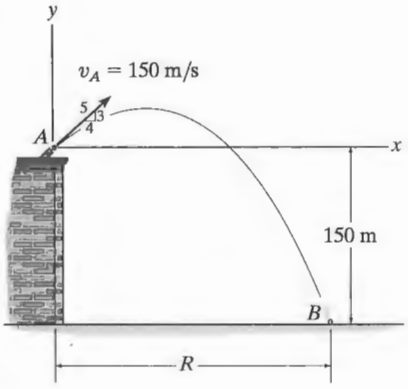
F12–24

F12–25. A ball is thrown from A. If it is required to clear the wall at B, determine the minimum magnitude of its initial velocity $\mathbf{v}_A$.



F12–25

F12–26. A projectile is fired with an initial velocity of $v_A = 150$ m/s off the roof of the building. Determine the range R where it strikes the ground at B.



F12–26

Hibbeler, RC. (2012). Force Systems Resultants. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 2-52). Singapore: Pearson Education South Asia Pte Ltd.

MasteringENGINEERING
Customized coaching is available

12 PROBLEMS

12–71. The position of a particle is $\mathbf{r} = \{(3t^3 - 2t)\mathbf{i} - (4t^{1/2} + t)\mathbf{j} + (3t^2 - 2)\mathbf{k}\}$ m, where t is in seconds. Determine the magnitude of the particle's velocity and acceleration when $t = 2$ s.

Video Solutions

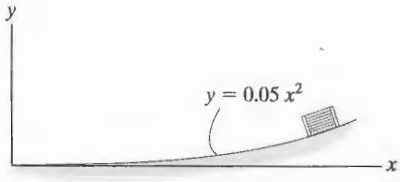
***12–72.** The velocity of a particle is $\mathbf{v} = \{3\mathbf{i} + (6 - 2t)\mathbf{j}\}$ m/s, where t is in seconds. If $\mathbf{r} = \mathbf{0}$ when $t = 0$, determine the displacement of the particle during the time interval $t = 1$ s to $t = 3$ s.

•12–73. A particle travels along the parabolic path $y = bx^2$. If its component of velocity along the y axis is $v_y = ct^2$, determine the x and y components of the particle's acceleration. Here b and c are constants.

12–74. The velocity of a particle is given by $\mathbf{v} = \{16t^2\mathbf{i} + 4t^3\mathbf{j} + (5t + 2)\mathbf{k}\}$ m/s, where t is in seconds. If the particle is at the origin when $t = 0$, determine the magnitude of the particle's acceleration when $t = 2$ s. Also, what is the x, y, z coordinate position of the particle at this instant?

12–75. A particle travels along the circular path $x^2 + y^2 = r^2$. If the y component of the particle's velocity is $v_y = 2r \cos 2t$, determine the x and y components of its acceleration at any instant.

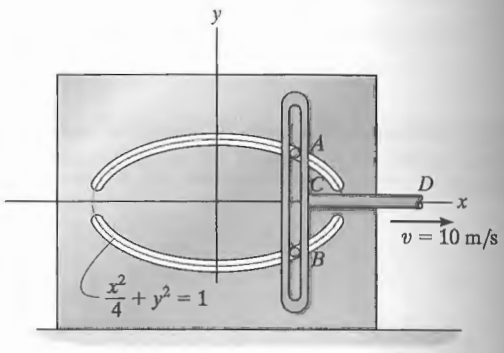
***12–76.** The box slides down the slope described by the equation $y = (0.05x^2)$ m, where x is in meters. If the box has x components of velocity and acceleration of $v_x = -3$ m/s and $a_x = -1.5$ m/s^2 at $x = 5$ m, determine the y components of the velocity and the acceleration of the box at this instant.



Prob. 12–76

•12–77. The position of a particle is defined by $\mathbf{r} = \{5 \cos 2t\,\mathbf{i} + 4 \sin 2t\,\mathbf{j}\}$ m, where t is in seconds and the arguments for the sine and cosine are given in radians. Determine the magnitudes of the velocity and acceleration of the particle when $t = 1$ s. Also, prove that the path of the particle is elliptical.

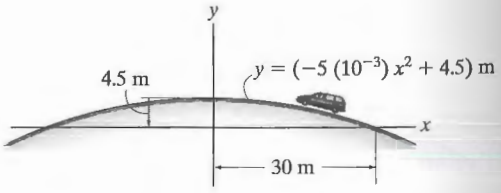
12–78. Pegs A and B are restricted to move in the elliptical slots due to the motion of the slotted link. If the link moves with a constant speed of 10 m/s, determine the magnitude of the velocity and acceleration of peg A when $x = 1$ m.



Prob. 12–78

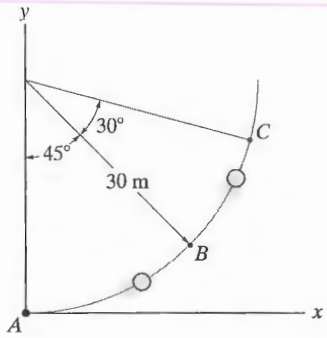
12–79. A particle travels along the path $y^2 = 4x$ with a constant speed of $v = 4$ m/s. Determine the x and y components of the particle's velocity and acceleration when the particle is at $x = 4$ m.

***12–80.** The van travels over the hill described by $y = (-5(10^{-3})\,x^2 + 4.5)$ m. If it has a constant speed of 22.5 m/s, determine the x and y components of the van's velocity and acceleration when $x = 15$ m.



Prob. 12–80

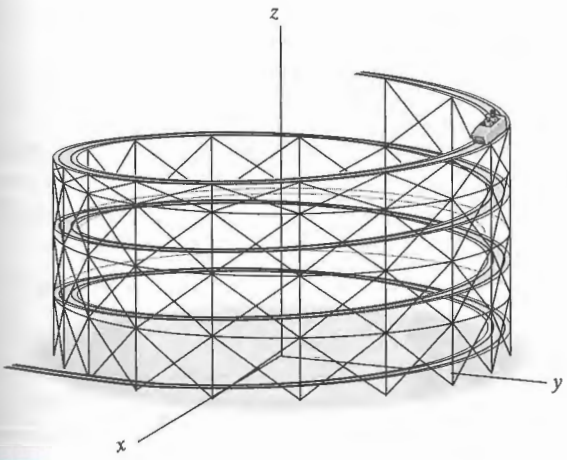
•**12–81.** A particle travels along the circular path from A to B in 1 s. If it takes 3 s for it to go from A to C, determine its *average velocity* when it goes from B to C.



Prob. 12–81

12–82. A car travels east 2 km for 5 minutes, then north 3 km for 8 minutes, and then west 4 km for 10 minutes. Determine the total distance traveled and the magnitude of displacement of the car. Also, what is the magnitude of the average velocity and the average speed?

12–83. The roller coaster car travels down the helical path at constant speed such that the parametric equations that define its position are $x = c \sin kt$, $y = c \cos kt$, $z = h - bt$, where c, h, and b are constants. Determine the magnitudes of its velocity and acceleration.



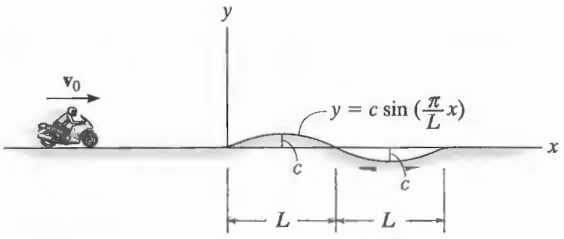
Prob. 12–83

**12–84.* The path of a particle is defined by $y^2 = 4kx$, and the component of velocity along the y axis is $v_y = ct$, where both k and c are constants. Determine the x and y components of acceleration when $y = y_0$.

Video Solutions

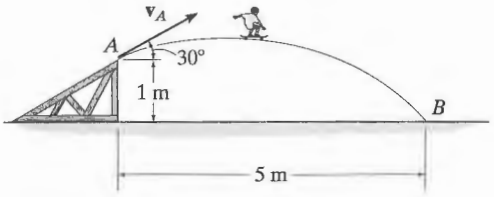
•**12–85.** A particle moves along the curve $y = x - (x^2/400)$, where x and y are in m. If the velocity component in the x direction is $v_x = 2$ m/s and remains *constant*, determine the magnitudes of the velocity and acceleration when $x = 20$ m.

12–86. The motorcycle travels with constant speed v_0 along the path that, for a short distance, takes the form of a sine curve. Determine the x and y components of its velocity at any instant on the curve.



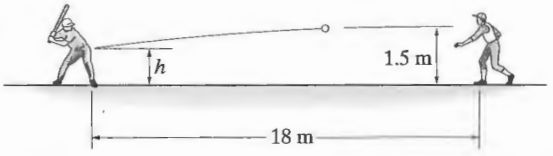
Prob. 12–86

12–87. The skateboard rider leaves the ramp at A with an initial velocity v_A at a 30° angle. If he strikes the ground at B, determine v_A and the time of flight.



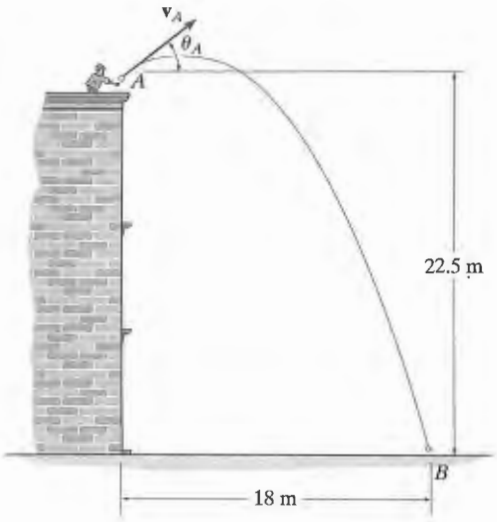
Prob. 12–87

**12–88.* The pitcher throws the baseball horizontally with a speed of 42 m/s from a height of 1.5 m. If the batter is 18 m away, determine the time for the ball to arrive at the batter and the height h at which it passes the batter.



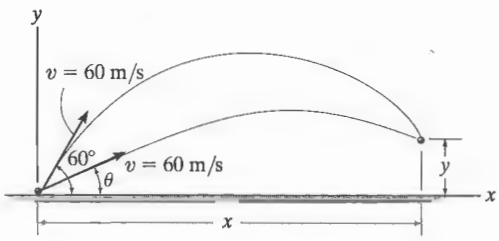
Prob. 12–88

•**12–89.** The ball is thrown off the top of the building. If it strikes the ground at B in 3 s, determine the initial velocity v_A and the inclination angle θ_A at which it was thrown. Also, find the magnitude of the ball's velocity when it strikes the ground.



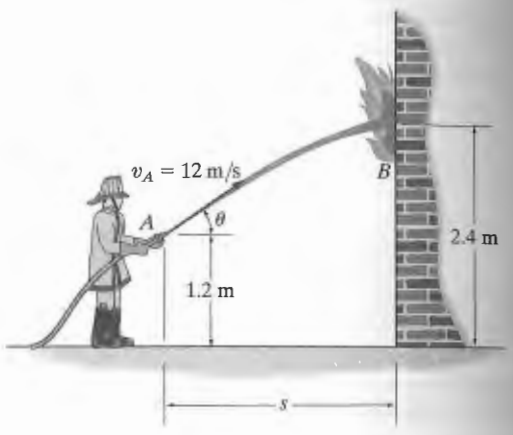
Prob. 12–89

12–90. A projectile is fired with a speed of $v = 60$ m/s at an angle of 60°. A second projectile is then fired with the same speed 0.5 s later. Determine the angle θ of the second projectile so that the two projectiles collide. At what position (x, y) will this happen?



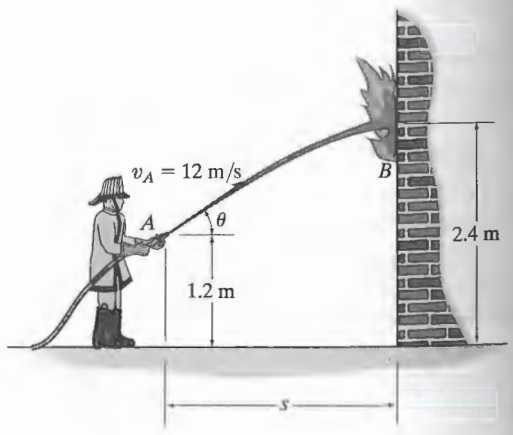
Prob. 12–90

12–91. The fireman holds the hose at an angle $\theta = 30°$ with horizontal, and the water is discharged from the hose at A with a speed of $v_A = 12$ m/s. If the water stream strikes the building at B, determine his two possible distances s from the building.



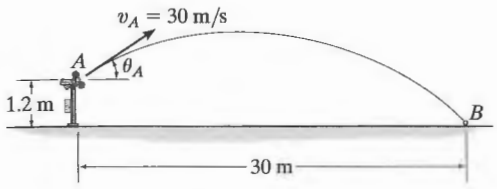
Prob. 12–91

***12–92.** Water is discharged from the hose with a speed of 12 m/s. Determine the two possible angles θ the fireman can hold the hose so that the water strikes the building at B. Take $s = 6$ m.



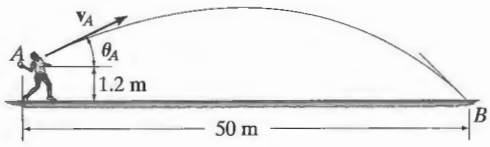
Prob. 12–92

•12–93. The pitching machine is adjusted so that the baseball is launched with a speed of $v_A = 30\,\text{m/s}$. If the ball strikes the ground at B, determine the two possible angles θ_A at which it was launched.



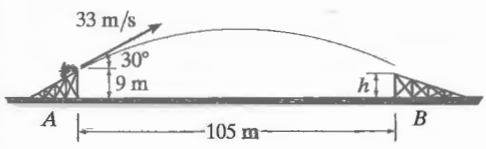
Prob. 12–93

12–94. It is observed that the time for the ball to strike the ground at B is 2.5 s. Determine the speed v_A and angle θ_A at which the ball was thrown.



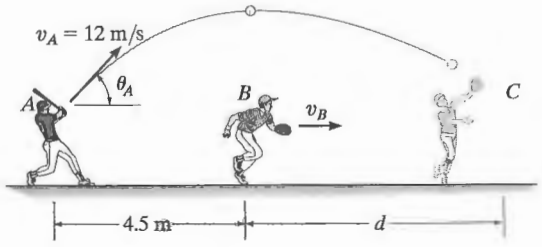
Prob. 12–94

12–95. If the motorcycle leaves the ramp traveling at 33 m/s, determine the height h ramp B must have so that the motorcycle lands safely.



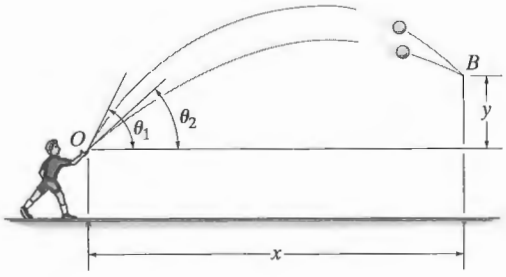
Prob. 12–95

***12–96.** The baseball player A hits the baseball with $v_A = 12\,\text{m/s}$ and $\theta_A = 60°$. When the ball is directly above of player B he begins to run under it. Determine the constant speed v_B and the distance d at which B must run in order to make the catch at the same elevation at which the ball was hit.



Prob. 12–96

•12–97. A boy throws a ball at O in the air with a speed v_0 at an angle θ_1. If he then throws another ball with the same speed v_0 at an angle $\theta_2 < \theta_1$, determine the time between the throws so that the balls collide in mid air at B.

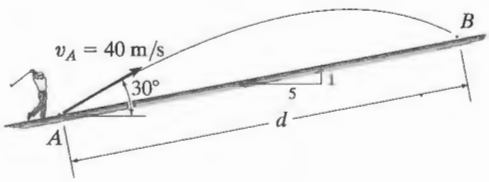


Prob. 12–97

12

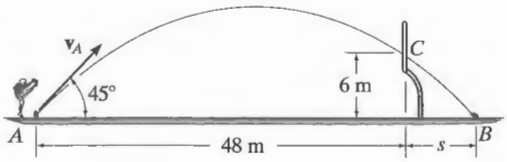
12–98. The golf ball is hit at A with a speed of $v_A = 40\,\text{m/s}$ and directed at an angle of 30° with the horizontal as shown. Determine the distance d where the ball strikes the slope at B.

Video Solutions



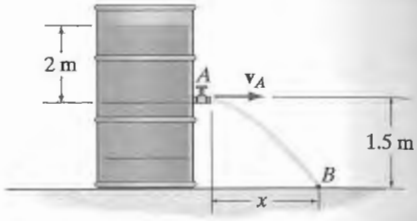
Prob. 12–98

12–99. If the football is kicked at the 45° angle, determine its minimum initial speed v_A so that it passes over the goal post at C. At what distance s from the goal post will the football strike the ground at B?



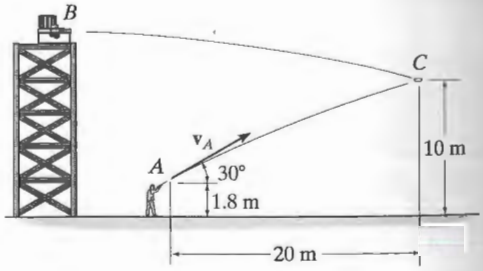
Prob. 12–99

***12–100.** The velocity of the water jet discharging from the orifice can be obtained from $v = \sqrt{2\,gh}$, where $h = 2\,\text{m}$ is the depth of the orifice from the free water surface. Determine the time for a particle of water leaving the orifice to reach point B and the horizontal distance x where it hits the surface.



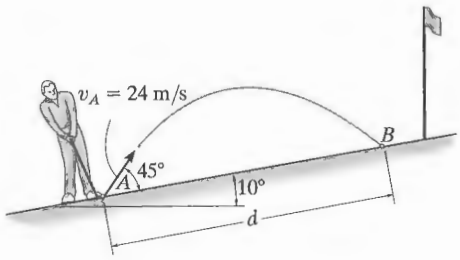
Prob. 12–100

•12–101. A projectile is fired from the platform at B. The shooter fires his gun from point A at an angle of 30°. Determine the muzzle speed of the bullet if it hits the projectile at C.



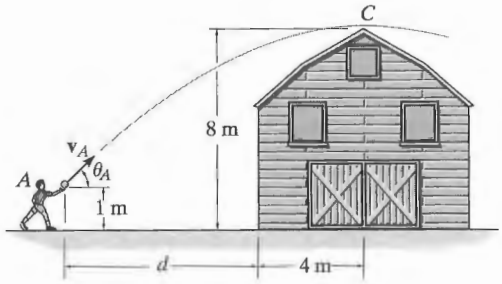
Prob. 12–101

12–102. A golf ball is struck with a velocity of 24 m/s as shown. Determine the distance d to where it will land.



Prob. 12–102

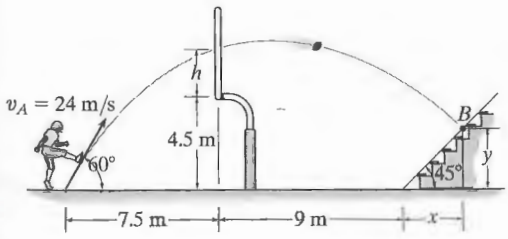
•12–105. The boy at A attempts to throw a ball over the roof of a barn with an initial speed of $v_A = 15$ m/s. Determine the angle θ_A at which the ball must be thrown so that it reaches its maximum height at C. Also, find the distance d where the boy should stand to make the throw.



Prob. 12–105

12–103. The football is to be kicked over the goalpost, which is 4.5 m high. If its initial speed is $v_A = 24$ m/s, determine if it makes it over the goalpost, and if so, by how much, h.

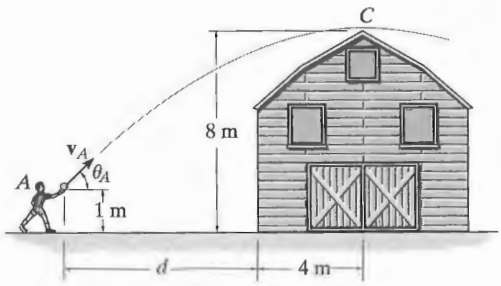
***12–104.** The football is kicked over the goalpost with an initial velocity of $v_A = 24$ m/s as shown. Determine the point B (x, y) where it strikes the bleachers.



Probs. 12–103/104

12–106. The boy at A attempts to throw a ball over the roof of a barn such that it is launched at an angle $\theta_A = 40°$. Determine the minimum speed v_A at which he must throw the ball so that it reaches its maximum height at C. Also, find the distance d where the boy must stand so that he can make the throw.

Video Solutions



Prob. 12–106

The design of conveyors for a bottling plant requires knowledge of the forces that act on them and the ability to predict the motion of the bottles they transport.

. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

Kinetics of a Particle: Force and Acceleration

13

CHAPTER OBJECTIVES

- To state Newton's Second Law of Motion and to define mass and weight.
- To analyze the accelerated motion of a particle using the equation of motion with different coordinate systems.
- To investigate central-force motion and apply it to problems in space mechanics.

13.1 Newton's Second Law of Motion

Kinetics is a branch of dynamics that deals with the relationship between the change in motion of a body and the forces that cause this change. The basis for kinetics is Newton's second law, which states that when an *unbalanced force* acts on a particle, the particle will *accelerate* in the direction of the force with a magnitude that is proportional to the force.

This law can be verified experimentally by applying a known unbalanced force **F** to a particle, and then measuring the acceleration **a**. Since the force and acceleration are directly proportional, the constant of proportionality, m, may be determined from the ratio $m = F/a$. This positive scalar m is called the *mass* of the particle. Being constant during any acceleration, m provides a quantitative measure of the resistance of the particle to a change in its velocity, that is its inertia.

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

If the mass of the particle is m, Newton's second law of motion may be written in mathematical form as

$$\mathbf{F} = m\mathbf{a}$$

The above equation, which is referred to as the *equation of motion* is one of the most important formulations in mechanics.* As previously stated, its validity is based solely on *experimental evidence*. In 1905, however, Albert Einstein developed the theory of relativity and placed limitations on the use of Newton's second law for describing general particle motion. Through experiments it was proven that *time* is not an absolute quantity as assumed by Newton; and as a result, the equation of motion fails to predict the exact behavior of a particle, especially when the particle's speed approaches the speed of light (0.3 Gm/s). Developments of the theory of quantum mechanics by Erwin Schrödinger and others indicate further that conclusions drawn from using this equation are also invalid when particles are the size of an atom and move close to one another. For the most part, however, these requirements regarding particle speed and size are not encountered in engineering problems, so their effects will not be considered in this book.

Newton's Law of Gravitational Attraction. Shortly after formulating his three laws of motion, Newton postulated a law governing the mutual attraction between any two particles. In mathematical form this law can be expressed as

$$\boxed{F = G\frac{m_1 m_2}{r^2}} \tag{13-1}$$

where

F = force of attraction between the two particles
G = universal constant of gravitation; according to experimental evidence $G = 66.73(10^{-12})\ \mathrm{m^3/(kg \cdot s^2)}$
m_1, m_2 = mass of each of the two particles
r = distance between the centers of the two particles

*Since m is constant, we can also write $\mathbf{F} = d(m\mathbf{v})/dt$, where $m\mathbf{v}$ is the particle's linear momentum. Here the unbalanced force acting on the particle is proportional to the time rate of change of the particle's linear momentum.

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

In the case of a particle located at or near the surface of the earth, the only gravitational force having any sizable magnitude is that between the earth and the particle. This force is termed the "weight" and, for our purpose, it will be the only gravitational force considered.

From Eq. 13–1, we can develop a general expression for finding the weight W of a particle having a mass $m_1 = m$. Let $m_2 = M_e$ be the mass of the earth and r the distance between the earth's center and the particle. Then, if $g = GM_e/r^2$, we have

$$W = mg$$

By comparison with $F = ma$, we term g the acceleration due to gravity. For most engineering calculations g is a point on the surface of the earth at sea level, and at a latitude of 45°, which is considered the "standard location." Here the values $g = 9.81\ \text{m/s}^2$ will be used for calculations.

In the SI system the mass of the body is specified in kilograms, and the weight must be calculated using the above equation, Fig. 13–1. Thus,

$$\boxed{W = mg\ (\text{N})} \qquad (13\text{–}2)$$

$$\boxed{(g = 9.81\ \text{m/s}^2)} \qquad (13\text{–}3)$$

As a result, a body of mass 1 kg has a weight of 9.81 N; a 2-kg body weighs 19.62 N; and so on.

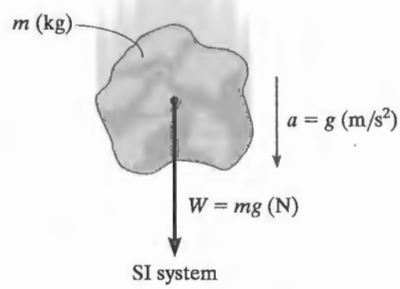


Fig. 13–1

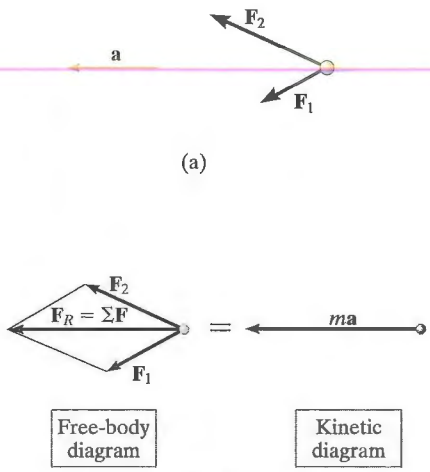


Fig. 13–2

13.2 The Equation of Motion

When more than one force acts on a particle, the resultant force is determined by a vector summation of all the forces; i.e., $\mathbf{F}_R = \Sigma\mathbf{F}$. For this more general case, the equation of motion may be written as

$$\boxed{\Sigma\mathbf{F} = m\mathbf{a}} \tag{13–4}$$

To illustrate application of this equation, consider the particle shown in Fig. 13–2*a*, which has a mass *m* and is subjected to the action of two forces, $\mathbf{F}_1$ and $\mathbf{F}_2$. We can graphically account for the magnitude and direction of each force acting on the particle by drawing the particle's *free-body diagram*, Fig. 13–2*b*. Since the *resultant* of these forces *produces* the vector $m\mathbf{a}$, its magnitude and direction can be represented graphically on the *kinetic diagram*, shown in Fig. 13–2*c*.* The equal sign written between the diagrams symbolizes the *graphical* equivalency between the free-body diagram and the kinetic diagram; i.e., $\Sigma\mathbf{F} = m\mathbf{a}$.† In particular, note that if $\mathbf{F}_R = \Sigma\mathbf{F} = \mathbf{0}$, then the acceleration is also zero, so that the particle will either remain at *rest* or move along a straight-line path with *constant velocity*. Such are the conditions of *static equilibrium*, Newton's first law of motion.

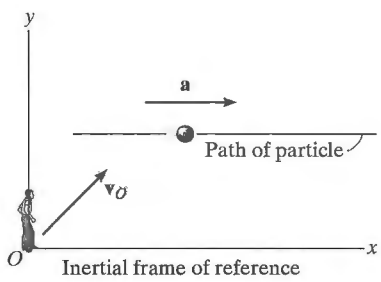


Fig. 13–3

Inertial Reference Frame. When applying the equation of motion, it is important that the acceleration of the particle be measured with respect to a reference frame that is *either fixed or translates with a constant velocity*. In this way, the observer will not accelerate and measurements of the particle's acceleration will be the *same* from *any reference* of this type. Such a frame of reference is commonly known as a *Newtonian* or *inertial reference frame*, Fig. 13–3.

When studying the motions of rockets and satellites, it is justifiable to consider the inertial reference frame as fixed to the stars, whereas dynamics problems concerned with motions on or near the surface of the earth may be solved by using an inertial frame which is assumed fixed to the earth. Even though the earth both rotates about its own axis and revolves about the sun, the accelerations created by these rotations are relatively small and so they can be neglected for most applications.

*Recall the free-body diagram considers the particle to be free of its surrounding supports and shows all the forces acting on the particle. The kinetic diagram pertains to the particle's motion as caused by the forces.

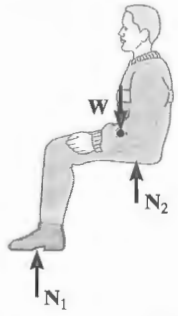
†The equation of motion can also be rewritten in the form $\Sigma\mathbf{F} - m\mathbf{a} = \mathbf{0}$. The vector $-m\mathbf{a}$ is referred to as the *inertia force vector*. If it is treated in the same way as a "force vector," then the state of "equilibrium" created is referred to as *dynamic equilibrium*. This method of application is often referred to as the *D'Alembert principle*, named after the French mathematician Jean le Rond d'Alembert.

RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

blication from Pearson Custom Publishing exclusively for The University of Melbourne

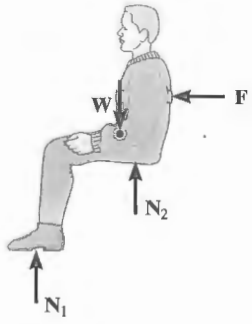
We are all familiar with the sensation one feels when sitting in a car that is subjected to a forward acceleration. Often people think this is caused by a "force" which acts on them and tends to push them back in their seats; however, this is not the case. Instead, this sensation occurs due to their inertia or the resistance of their mass to a change in velocity.

Consider the passenger who is strapped to the seat of a rocket sled. Provided the sled is at rest or is moving with constant velocity, then no force is exerted on his back as shown on his free-body diagram.



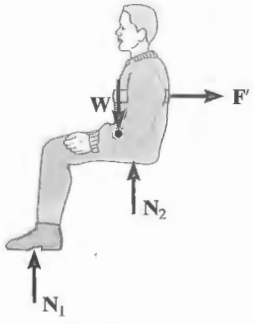
At rest or constant velocity

When the thrust of the rocket engine causes the sled to accelerate, then the seat upon which he is sitting exerts a force **F** on him which pushes him forward with the sled. In the photo, notice that the inertia of his head resists this change in motion (acceleration), and so his head moves back against the seat and his face, which is nonrigid, tends to distort backward.



Acceleration

Upon deceleration the force of the seatbelt **F′** tends to pull his body to a stop, but his head leaves contact with the back of the seat and his face distorts forward, again due to his inertia or tendency to continue to move forward. No force is pulling him forward, although this is the sensation he receives.



Deceleration

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

13.3 Equation of Motion for a System of Particles

The equation of motion will now be extended to include a system of particles isolated within an enclosed region in space, as shown in Fig. 13–4*a*. In particular, there is no restriction in the way the particles are connected so the following analysis applies equally well to the motion of a solid, liquid, or gas system.

At the instant considered, the arbitrary i-th particle, having a mass m_i, is subjected to a system of internal forces and a resultant external force. The *internal force*, represented symbolically as $\mathbf{f}_i$, is the resultant of all the forces the other particles exert on the ith particle. The *resultant external force* $\mathbf{F}_i$ represents, for example, the effect of gravitational electrical, magnetic, or contact forces between the ith particle and adjacent bodies or particles *not* included within the system.

The free-body and kinetic diagrams for the ith particle are shown in Fig. 13–4*b*. Applying the equation of motion,

$$\Sigma \mathbf{F} = m\mathbf{a}; \qquad \mathbf{F}_i + \mathbf{f}_i = m_i\mathbf{a}_i$$

When the equation of motion is applied to each of the other particles of the system, similar equations will result. And, if all these equations are added together *vectorially*, we obtain

$$\Sigma \mathbf{F}_i + \Sigma \mathbf{f}_i = \Sigma m_i \mathbf{a}_i$$

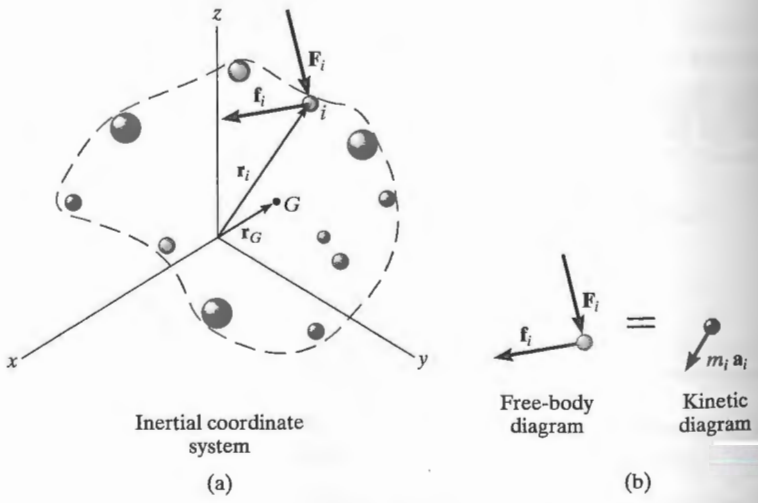


Fig. 13–4

The summation of the internal forces, if carried out, will equal zero, since internal forces between any two particles occur in equal but opposite collinear pairs. Consequently, only the sum of the external forces will remain, and therefore the equation of motion, written for the system of particles, becomes

$$\Sigma \mathbf{F}_i = \Sigma m_i \mathbf{a}_i \tag{13–5}$$

If $\mathbf{r}_G$ is a position vector which locates the *center of mass* G of the particles, Fig. 13–4*a*, then by definition of the center of mass, $m\mathbf{r}_G = \Sigma m_i \mathbf{r}_i$, where $m = \Sigma m_i$ is the total mass of all the particles. Differentiating this equation twice with respect to time, assuming that no mass is entering or leaving the system, yields

$$m\mathbf{a}_G = \Sigma m_i \mathbf{a}_i$$

Substituting this result into Eq. 13–5, we obtain

$$\boxed{\Sigma \mathbf{F} = m\mathbf{a}_G} \tag{13–6}$$

Hence, the sum of the external forces acting on the system of particles is equal to the total mass of the particles times the acceleration of its center of mass G. Since in reality all particles must have a finite size to possess mass, Eq. 13–6 justifies application of the equation of motion to a *body* that is represented as a single particle.

Important Points

- The equation of motion is based on experimental evidence and is valid only when applied within an inertial frame of reference.
- The equation of motion states that the *unbalanced force* on a particle causes it to *accelerate*.
- An inertial frame of reference does not rotate, rather its axes either translate with constant velocity or are at rest.
- Mass is a property of matter that provides a quantitative measure of its resistance to a change in velocity. It is an absolute quantity and so it does not change from one location to another.
- Weight is a force that is caused by the earth's gravitation. It is not absolute; rather it depends on the altitude of the mass from the earth's surface.

13.4 Equations of Motion: Rectangular Coordinates

When a particle moves relative to an inertial x, y, z frame of reference, the forces acting on the particle, as well as its acceleration, can be expressed in terms of their $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components, Fig. 13–5. Applying the equation of motion, we have

$$\Sigma \mathbf{F} = m\mathbf{a}; \quad \Sigma F_x\mathbf{i} + \Sigma F_y\mathbf{j} + \Sigma F_z\mathbf{k} = m(a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k})$$

For this equation to be satisfied, the respective $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components on the left side must equal the corresponding components on the right side. Consequently, we may write the following three scalar equations:

$$\boxed{\begin{aligned} \Sigma F_x &= ma_x \\ \Sigma F_y &= ma_y \\ \Sigma F_z &= ma_z \end{aligned}} \qquad (13\text{–}7)$$

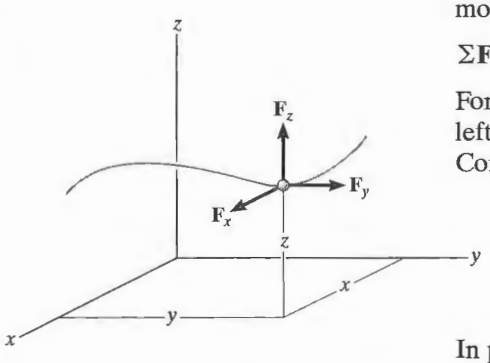


Fig. 13–5

In particular, if the particle is constrained to move only in the x–y plane, then the first two of these equations are used to specify the motion.

Procedure for Analysis

The equations of motion are used to solve problems which require a relationship between the forces acting on a particle and the accelerated motion they cause.

Free-Body Diagram.

- Select the inertial coordinate system. Most often, rectangular or x, y, z coordinates are chosen to analyze problems for which the particle has *rectilinear motion*.
- Once the coordinates are established, draw the particle's free-body diagram. Drawing this diagram is *very important* since it provides a graphical representation that accounts for *all the forces* ($\Sigma\mathbf{F}$) which act on the particle, and thereby makes it possible to resolve these forces into their x, y, z components.
- The direction and sense of the particle's acceleration $\mathbf{a}$ should also be established. If the sense is unknown, for mathematical convenience assume that the sense of each acceleration component acts in the *same direction* as its *positive* inertial coordinate axis.
- The acceleration may be represented as the $m\mathbf{a}$ vector on the kinetic diagram.*
- Identify the unknowns in the problem.

*It is a convention in this text always to use the kinetic diagram as a graphical aid when developing the proofs and theory. The particle's acceleration or its components will be shown as blue colored vectors near the free-body diagram in the examples.

eler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

A publication from Pearson Custom Publishing exclusively for The University of Melbourne

Equations of Motion.

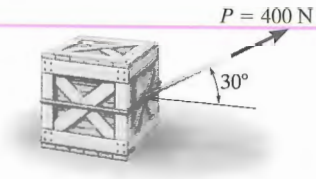
- If the forces can be resolved directly from the free-body diagram, apply the equations of motion in their scalar component form.
- If the geometry of the problem appears complicated, which often occurs in three dimensions, Cartesian vector analysis can be used for the solution.
- *Friction.* If a moving particle contacts a rough surface, it may be necessary to use the *frictional equation*, which relates the frictional and normal forces $\mathbf{F}_f$ and $\mathbf{N}$ acting at the surface of contact by using the coefficient of kinetic friction, i.e., $F_f = \mu_k N$. Remember that $\mathbf{F}_f$ always acts on the free-body diagram such that it opposes the motion of the particle relative to the surface it contacts. If the particle is *on the verge* of relative motion, then the coefficient of static friction should be used.
- *Spring.* If the particle is connected to an *elastic spring* having negligible mass, the spring force F_s can be related to the deformation of the spring by the equation $F_s = ks$. Here k is the spring's stiffness measured as a force per unit length, and s is the stretch or compression defined as the difference between the deformed length l and the undeformed length l_0, i.e., $s = l - l_0$.

Kinematics.

- If the velocity or position of the particle is to be found, it will be necessary to apply the necessary kinematic equations once the particle's acceleration is determined from $\Sigma\mathbf{F} = m\mathbf{a}$.
- If *acceleration* is a function of time, use $a = dv/dt$ and $v = ds/dt$ which, when integrated, yield the particle's velocity and position, respectively.
- If *acceleration* is a function of displacement, integrate $a\,ds = v\,dv$ to obtain the velocity as a function of position.
- If *acceleration is constant*, use $v = v_0 + a_c t$, $s = s_0 + v_0 t + \frac{1}{2}a_c t^2$, $v^2 = v_0^2 + 2a_c(s - s_0)$ to determine the velocity or position of the particle.
- If the problem involves the dependent motion of several particles, use the method outlined in Sec. 12.9 to relate their accelerations. In all cases, make sure the positive inertial coordinate directions used for writing the kinematic equations are the same as those used for writing the equations of motion; otherwise, simultaneous solution of the equations will result in errors.
- If the solution for an unknown vector component yields a negative scalar, it indicates that the component acts in the direction opposite to that which was assumed.

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 13.1



(a)

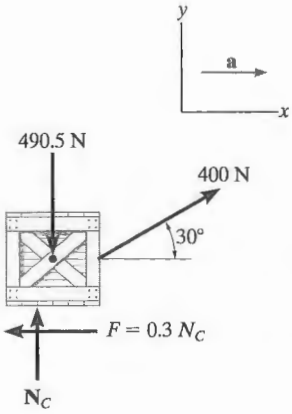
The 50-kg crate shown in Fig. 13–6*a* rests on a horizontal surface for which the coefficient of kinetic friction is $\mu_k = 0.3$. If the crate is subjected to a 400-N towing force as shown, determine the velocity of the crate in 3 s starting from rest.

SOLUTION

Using the equations of motion, we can relate the crate's acceleration to the force causing the motion. The crate's velocity can then be determined using kinematics.

Free-Body Diagram. The weight of the crate is $W = mg = 50\text{ kg}\,(9.81\text{ m/s}^2) = 490.5\text{ N}$. As shown in Fig. 13–6*b*, the frictional force has a magnitude $F = \mu_k N_C$ and acts to the left, since it opposes the motion of the crate. The acceleration **a** is assumed to act horizontally, in the positive x direction. There are two unknowns, namely N_C and a.

Equations of Motion. Using the data shown on the free-body diagram, we have



(b)

Fig. 13–6

$$\overset{+}{\rightarrow}\ \Sigma F_x = ma_x; \qquad 400\cos 30° - 0.3N_C = 50a \qquad (1)$$

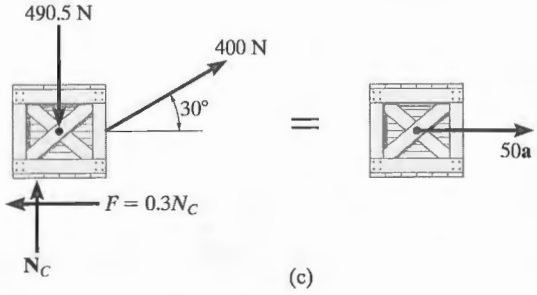
$$+\uparrow \Sigma F_y = ma_y; \qquad N_C - 490.5 + 400\sin 30° = 0 \qquad (2)$$

Solving Eq. 2 for N_C, substituting the result into Eq. 1, and solving for a yields

$$N_C = 290.5\text{ N}$$
$$a = 5.185\text{ m/s}^2$$

Kinematics. Notice that the acceleration is *constant*, since the applied force **P** is constant. Since the initial velocity is zero, the velocity of the crate in 3 s is

$$(\overset{+}{\rightarrow}) \qquad v = v_0 + a_c t = 0 + 5.185(3)$$
$$= 15.6\text{ m/s} \rightarrow \qquad \textit{Ans.}$$



(c)

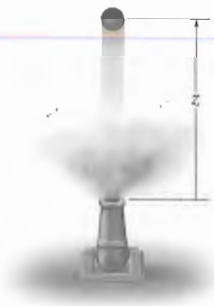
NOTE: We can also use the alternative procedure of drawing the crate's free-body *and* kinetic diagrams, Fig. 13–6*c*, prior to applying the equations of motion.

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

10003 – ENGINEERING SYSTEMS DESIGN 2
cation from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 13.2

A 10-kg projectile is fired vertically upward from the ground, with an initial velocity of 50 m/s, Fig. 13–7*a*. Determine the maximum height to which it will travel if (*a*) atmospheric resistance is neglected; and (*b*) atmospheric resistance is measured as $F_D = (0.01v^2)$ N, where v is the speed of the projectile at any instant, measured in m/s.

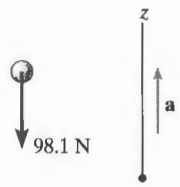


(a)

SOLUTION

In both cases the known force on the projectile can be related to its acceleration using the equation of motion. Kinematics can then be used to relate the projectile's acceleration to its position.

Part (a) Free-Body Diagram. As shown in Fig. 13–7*b*, the projectile's weight is $W = mg = 10(9.81) = 98.1$ N. We will assume the unknown acceleration **a** acts upward in the *positive* z direction.



(b)

Equation of Motion.

$$+\uparrow \Sigma F_z = ma_z; \qquad -98.1 = 10a, \qquad a = -9.81 \text{ m/s}^2$$

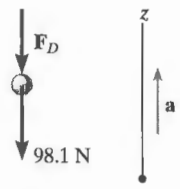
The result indicates that the projectile, like every object having free-flight motion near the earth's surface, is subjected to a *constant* downward acceleration of 9.81 m/s^2.

Kinematics. Initially, $z_0 = 0$ and $v_0 = 50$ m/s, and at the maximum height $z = h$, $v = 0$. Since the acceleration is *constant*, then

$$(+\uparrow) \qquad v^2 = v_0^2 + 2a_c(z - z_0)$$

$$0 = (50)^2 + 2(-9.81)(h - 0)$$

$$h = 127 \text{ m} \qquad \textit{Ans.}$$



(c)

Fig. 13–7

Part (b) Free-Body Diagram. Since the force $F_D = (0.01v^2)$ N tends to retard the upward motion of the projectile, it acts downward as shown on the free-body diagram, Fig. 13–7*c*.

Equation of Motion.

$$+\uparrow \Sigma F_z = ma_z; \qquad -0.01v^2 - 98.1 = 10a, \qquad a = -(0.001v^2 + 9.81)$$

Kinematics. Here the acceleration is *not constant* since F_D depends on the velocity. Since $a = f(v)$, we can relate a to position using

$$(+\uparrow)\ a\,dz = v\,dv; \qquad -(0.001v^2 + 9.81)\,dz = v\,dv$$

Separating the variables and integrating, realizing that initially $z_0 = 0$, $v_0 = 50$ m/s (positive upward), and at $z = h$, $v = 0$, we have

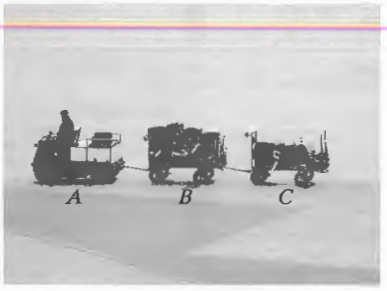
$$\int_0^h dz = -\int_{50}^0 \frac{v\,dv}{0.001v^2 + 9.81} = -500 \ln(v^2 + 9810)\Big|_{50\text{ m/s}}^{0}$$

$$h = 114 \text{ m} \qquad \textit{Ans.}$$

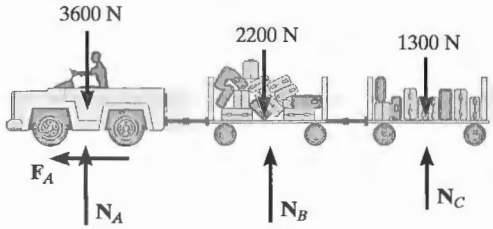
NOTE: The answer indicates a lower elevation than that obtained in part (*a*) due to atmospheric resistance or drag.

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 13.3



The baggage truck A shown in the photo has a weight of 3600-N ($\approx$ 360-kg) and tows a 2200-N ($\approx$ 220-kg) cart B and a 1300-N ($\approx$ 130-kg) cart C. For a short time the driving frictional force developed at the wheels of the truck is $F_A = (160t)$ N, where t is in seconds. If the truck starts from rest, determine its speed in 2 seconds. Also, what is the horizontal force acting on the coupling between the truck and cart B at this instant? Neglect the size of the truck and carts.



(a)

SOLUTION

Free-Body Diagram. As shown in Fig. 13–8a, it is the frictional driving force that gives both the truck and carts an acceleration. Here we have considered all three vehicles as a single system.

Equation of Motion. Only motion in the horizontal direction has to be considered.

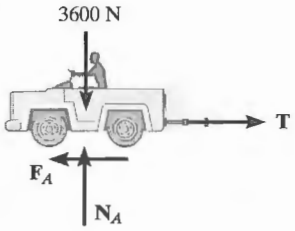
$$\overset{+}{\leftarrow} \Sigma F_x = ma_x; \qquad 160t = \left(\frac{3600 + 2200 + 1300}{9.81}\right)a$$

$$a = 0.221t$$

Kinematics. Since the acceleration is a function of time, the velocity of the truck is obtained using $a = dv/dt$ with the initial condition that $v_0 = 0$ at $t = 0$. We have

$$\int_0^v dv = \int_0^{2\,\text{s}} 0.221t\,dt; \quad v = 0.1105t^2\Big|_0^{2\,\text{s}} = 0.442 \text{ m/s} \qquad \textit{Ans.}$$

Free-Body Diagram. In order to determine the force between the truck and cart B, we will consider a free-body diagram of the truck so that we can "expose" the coupling force **T** as external to the free-body diagram, Fig. 13–8b.



(b)

Fig. 13–8

Equation of Motion. When $t = 2$ s, then

$$\overset{+}{\leftarrow} \Sigma F_x = ma_x; \qquad 160(2) - T = \left(\frac{3600}{9.81}\right)[0.221(2)]$$

$$T = 157.8 \text{ N} \qquad \textit{Ans.}$$

NOTE: Try and obtain this same result by considering a free-body diagram of carts B and C as a single system.

A publication from Pearson Custom Publishing exclusively for The University of Melbourne

EXAMPLE 13.4

A smooth 2-kg collar C, shown in Fig. 13–9a, is attached to a spring having a stiffness $k = 3$ N/m and an unstretched length of 0.75 m. If the collar is released from rest at A, determine its acceleration and the normal force of the rod on the collar at the instant $y = 1$ m.

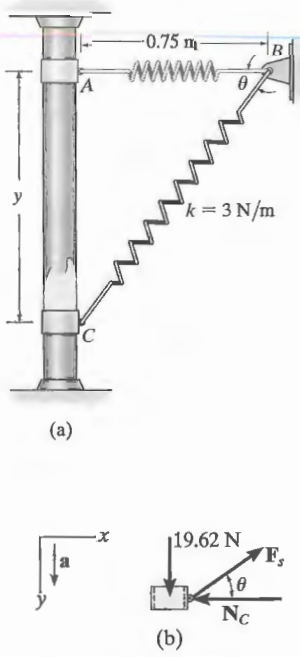


Fig. 13–9

SOLUTION

Free-Body Diagram. The free-body diagram of the collar when it is located at the arbitrary position y is shown in Fig. 13–9b. Furthermore, the collar is *assumed* to be accelerating so that "**a**" acts downward in the *positive* y direction. There are four unknowns, namely, N_C, F_s, a, and θ.

Equations of Motion.

$$\overset{+}{\rightarrow} \Sigma F_x = ma_x; \qquad -N_C + F_s \cos\theta = 0 \qquad (1)$$

$$+\downarrow \Sigma F_y = ma_y; \qquad 19.62 - F_s \sin\theta = 2a \qquad (2)$$

From Eq. 2 it is seen that the acceleration depends on the magnitude and direction of the spring force. Solution for N_C and a is possible once F_s and θ are known.

The magnitude of the spring force is a function of the stretch s of the spring; i.e., $F_s = ks$. Here the unstretched length is $AB = 0.75$ m, Fig. 13–9a; therefore, $s = CB - AB = \sqrt{y^2 + (0.75)^2} - 0.75$. Since $k = 3$ N/m, then

$$F_s = ks = 3\left(\sqrt{y^2 + (0.75)^2} - 0.75\right) \qquad (3)$$

From Fig. 13–9a, the angle θ is related to y by trigonometry.

$$\tan\theta = \frac{y}{0.75} \qquad (4)$$

Substituting $y = 1$ m into Eqs. 3 and 4 yields $F_s = 1.50$ N and $\theta = 53.1°$. Substituting these results into Eqs. 1 and 2, we obtain

$$N_C = 0.900 \text{ N} \qquad \textit{Ans.}$$

$$a = 9.21 \text{ m/s}^2 \downarrow \qquad \textit{Ans.}$$

NOTE: This is not a case of constant acceleration, since the spring force changes both its magnitude and direction as the collar moves downward.

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

EXAMPLE 13.5

The 100-kg block A shown in Fig. 13–10a is released from rest. If the masses of the pulleys and the cord are neglected, determine the speed of the 20-kg block B in 2 s.

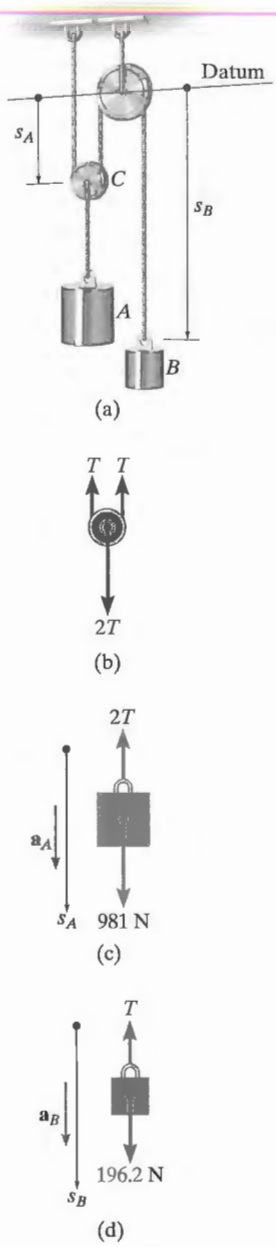


Fig. 13–10

SOLUTION

Free-Body Diagrams. Since the mass of the pulleys is *neglected*, then for pulley C, $ma = 0$ and we can apply $\Sigma F_y = 0$ as shown in Fig. 13–10b. The free-body diagrams for blocks A and B are shown in Fig. 13–10c and d, respectively. Notice that for A to remain stationary $T = 490.5$ N, whereas for B to remain static $T = 196.2$ N. Hence A will move down while B moves up. Although this is the case, we will assume both blocks accelerate downward, in the direction of $+s_A$ and $+s_B$. The three unknowns are T, a_A, and a_B.

Equations of Motion. Block A,

$$+\downarrow \Sigma F_y = ma_y; \qquad 981 - 2T = 100a_A \qquad (1)$$

Block B,

$$+\downarrow \Sigma F_y = ma_y; \qquad 196.2 - T = 20a_B \qquad (2)$$

Kinematics. The necessary third equation is obtained by relating a_A to a_B using a dependent motion analysis, discussed in Sec. 12.9. The coordinates s_A and s_B in Fig. 13–10a measure the positions of A and B from the fixed datum. It is seen that

$$2s_A + s_B = l$$

where l is constant and represents the total vertical length of cord. Differentiating this expression twice with respect to time yields

$$2a_A = -a_B \qquad (3)$$

Notice that when writing Eqs. 1 to 3, the *positive direction was always assumed downward*. It is very important to be *consistent* in this assumption since we are seeking a simultaneous solution of equations. The results are

$$T = 327.0 \text{ N}$$
$$a_A = 3.27 \text{ m/s}^2$$
$$a_B = -6.54 \text{ m/s}^2$$

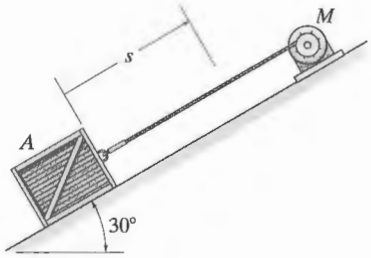
Hence when block A accelerates *downward*, block B accelerates *upward* as expected. Since a_B is constant, the velocity of block B in 2 s is thus

$$(+\downarrow) \qquad v = v_0 + a_B t$$
$$= 0 + (-6.54)(2)$$
$$= -13.1 \text{ m/s} \qquad \textit{Ans.}$$

The negative sign indicates that block B is moving upward.

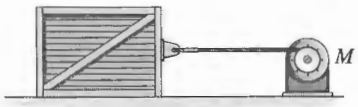
FUNDAMENTAL PROBLEMS

F13–1. The motor winds in the cable with a constant acceleration, such that the 20-kg crate moves a distance $s = 6$ m in 3 s, starting from rest. Determine the tension developed in the cable. The coefficient of kinetic friction between the crate and the plane is $\mu_k = 0.3$.



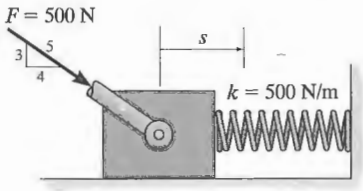
F13–1

F13–2. If motor M exerts a force of $F = (10t^2 + 100)$ N on the cable, where t is in seconds, determine the velocity of the 25-kg crate when $t = 4$ s. The coefficients of static and kinetic friction between the crate and the plane are $\mu_s = 0.3$ and $\mu_k = 0.25$, respectively. The crate is initially at rest.



F13–2

F13–3. A spring of stiffness $k = 500$ N/m is mounted against the 10-kg block. If the block is subjected to the force of $F = 500$ N, determine its velocity at $s = 0.5$ m. When $s = 0$, the block is at rest and the spring is uncompressed. The contact surface is smooth.

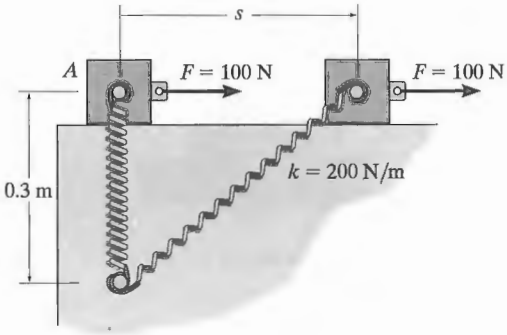


F13–3

F13–4. The 2-Mg car is being towed by a winch. If the winch exerts a force of $T = (100s)$ N on the cable, where s is the displacement of the car in meters, determine the speed of the car when $s = 10$ m, starting from rest. Neglect rolling resistance of the car.

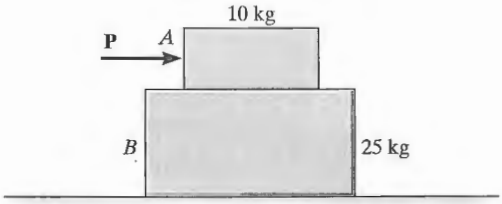
F13–4

F13–5. The spring has a stiffness $k = 200$ N/m and is unstretched when the 25-kg block is at A. Determine the acceleration of the block when $s = 0.4$ m. The contact surface between the block and the plane is smooth.



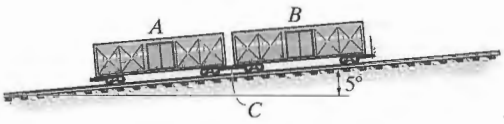
F13–5

F13–6. Block B rests upon a smooth surface. If the coefficients of static and kinetic friction between A and B are $\mu_s = 0.4$ and $\mu_k = 0.3$, respectively, determine the acceleration of each block if $P = 30$ N.



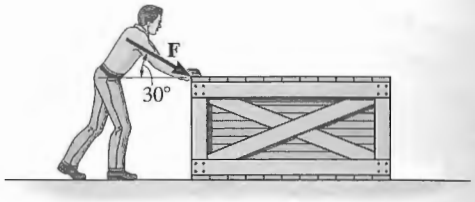
F13–6

•13–13. The two boxcars A and B have a mass of 10 Mg and 15 Mg, respectively. If they coast freely down the incline when the brakes are applied to all the wheels of car A causing it to skid, determine the force in the coupling C between the two cars. The coefficient of kinetic friction between the wheels of A and the tracks is $\mu_k = 0.5$. The wheels of car B are free to roll. Neglect their mass in the calculation. *Suggestion:* Solve the problem by representing single resultant normal forces acting on A and B, respectively.



Prob. 13–13

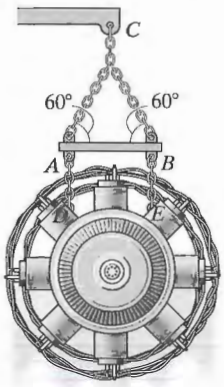
***13–16.** The man pushes on the 30-kg crate with a force **F**. The force is always directed down at 30° from the horizontal as shown, and its magnitude is increased until the crate begins to slide. Determine the crate's initial acceleration if the coefficient of static friction is $\mu_s = 0.6$ and the coefficient of kinetic friction is $\mu_k = 0.3$.



Prob. 13–16

13–14. The 3.5-Mg engine is suspended from a spreader beam AB having a negligible mass and is hoisted by a crane which gives it an acceleration of $4\ \text{m/s}^2$ when it has a velocity of 2 m/s. Determine the force in chains CA and CB during the lift.

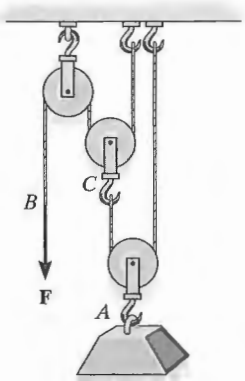
13–15. The 3.5-Mg engine is suspended from a spreader beam having a negligible mass and is hoisted by a crane which exerts a force of 40 kN on the hoisting cable. Determine the distance the engine is hoisted in 4 s, starting from rest.



Probs. 13–14/15

•13–17. A force of $F = 75$ N is applied to the cord. Determine how high the 15-kg block A rises in 2 s starting from rest. Neglect the weight of the pulleys and cord.

13–18. Determine the constant force **F** which must be applied to the cord in order to cause the 15-kg block A to have a speed of 3.6 m/s when it has been displaced 0.9 m upward starting from rest. Neglect the weight of the pulleys and cord.

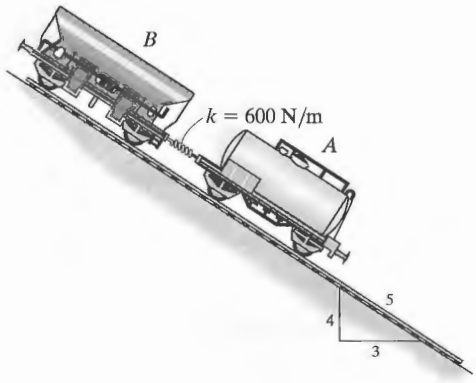


Probs. 13–17/18

(2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

publication from Pearson Custom Publishing exclusively for The University of Melbourne

13–19. The 800-kg car at B is connected to the 350-kg car at A by a spring coupling. Determine the stretch in the spring if (a) the wheels of both cars are free to roll and (b) the brakes are applied to all four wheels of car B, causing the wheels to skid. Take $(\mu_k)_B = 0.4$. Neglect the mass of the wheels.

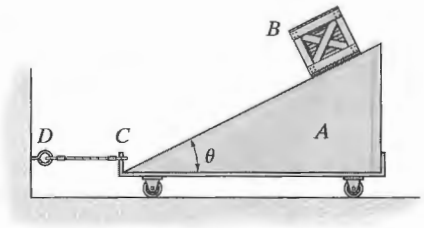


Prob. 13–19

•13–21. Block B has a mass m and is released from rest when it is on top of cart A, which has a mass of $3m$. Determine the tension in cord CD needed to hold the cart from moving while B slides down A. Neglect friction.

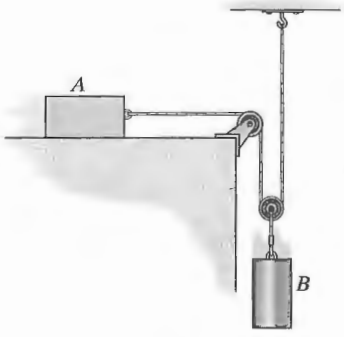
13–22. Block B has a mass m and is released from rest when it is on top of cart A, which has a mass of $3m$. Determine the tension in cord CD needed to hold the cart from moving while B slides down A. The coefficient of kinetic friction between A and B is μ_k.

Video Solutions



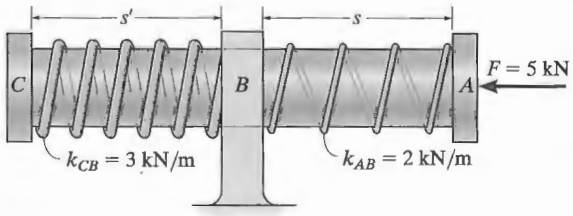
Probs. 13–21/22

***13–20.** The 5-kg block A travels to the right at $v_A = 0.6$ m/s at the instant shown. If the coefficient of kinetic friction is $\mu_k = 0.2$ between the surface and A, determine the velocity of A when it has moved 1.2 m. Block B has a weight of 100 N ($\approx$ 10 kg).



Prob. 13–20

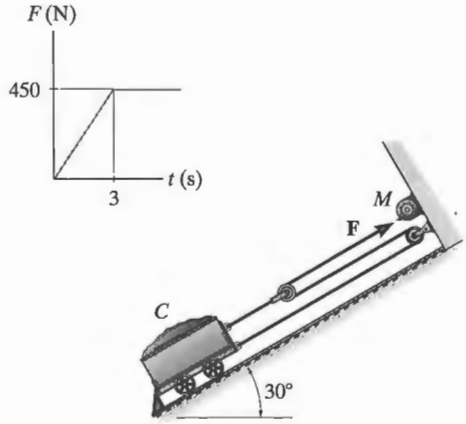
13–23. The 2-kg shaft CA passes through a smooth journal bearing at B. Initially, the springs, which are coiled loosely around the shaft, are unstretched when no force is applied to the shaft. In this position $s = s' = 250$ mm and the shaft is at rest. If a horizontal force of $F = 5$ kN is applied, determine the speed of the shaft at the instant $s = 50$ mm, $s' = 450$ mm. The ends of the springs are attached to the bearing at B and the caps at C and A.



Prob. 13–23

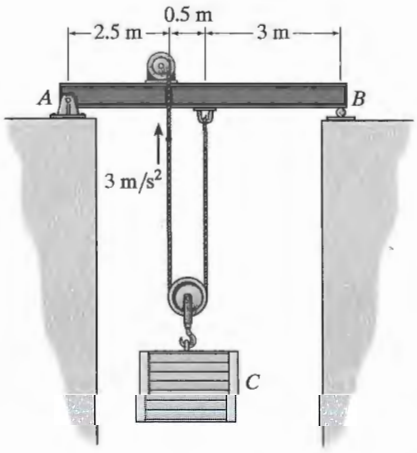
Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

***13–24.** If the force of the motor M on the cable is shown in the graph, determine the velocity of the cart when $t = 3$ s. The load and cart have a mass of 200 kg and the car starts from rest.



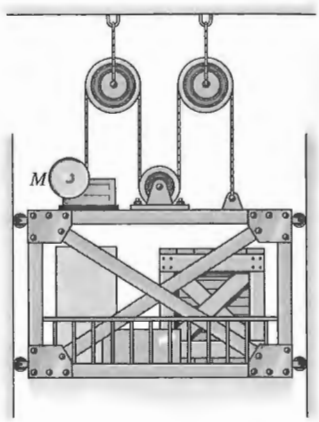
Prob. 13–24

•13–25. If the motor draws in the cable with an acceleration of $3\,\text{m/s}^2$, determine the reactions at the supports A and B. The beam has a uniform mass of 30 kg/m, and the crate has a mass of 200 kg. Neglect the mass of the motor and pulleys.



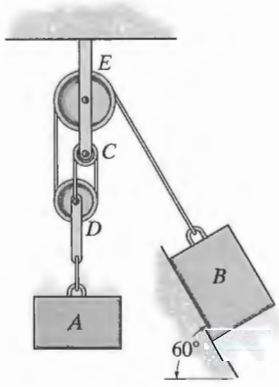
Prob. 13–25

13–26. A freight elevator, including its load, has a mass of 500 kg. It is prevented from rotating by the track and wheels mounted along its sides. When $t = 2$ s, the motor M draws in the cable with a speed of 6 m/s, *measured relative to the elevator*. If it starts from rest, determine the constant acceleration of the elevator and the tension in the cable. Neglect the mass of the pulleys, motor, and cables.



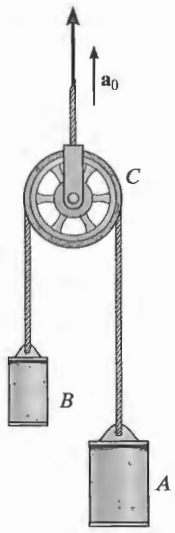
Prob. 13–26

13–27. Determine the required mass of block A so that when it is released from rest it moves the 5-kg block B a distance of 0.75 m up along the smooth inclined plane in $t = 2$ s. Neglect the mass of the pulleys and cords.



Prob. 13–27

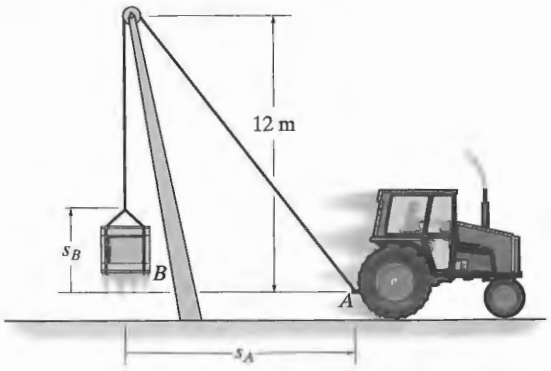
***13–28.** Blocks A and B have a mass of m_A and m_B, where $m_A > m_B$. If pulley C is given an acceleration of $\mathbf{a}_0$, determine the acceleration of the blocks. Neglect the mass of the pulley.



Prob. 13–28

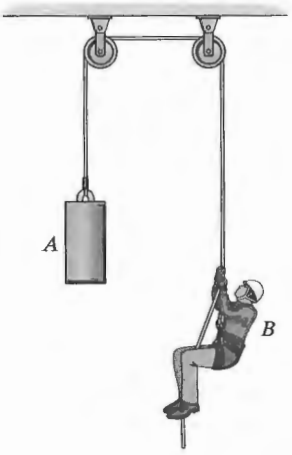
•13–29. The tractor is used to lift the 150-kg load B with the 24-m-long rope, boom, and pulley system. If the tractor travels to the right at a constant speed of 4 m/s, determine the tension in the rope when $s_A = 5$ m. When $s_A = 0$, $s_B = 0$.

13–30. The tractor is used to lift the 150-kg load B with the 24-m-long rope, boom, and pulley system. If the tractor travels to the right with an acceleration of 3 m/s^2 and has a velocity of 4 m/s at the instant $s_A = 5$ m, determine the tension in the rope at this instant. When $s_A = 0$, $s_B = 0$.



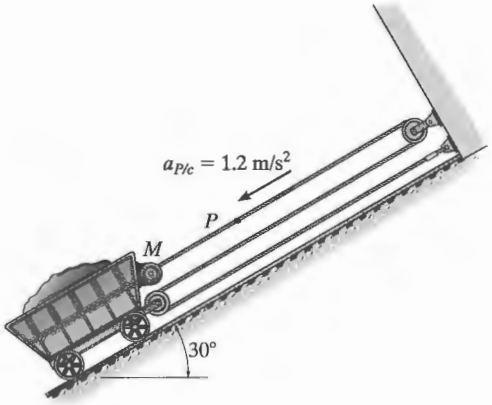
Probs. 13–29/30

13–31. The 75-kg man climbs up the rope with an acceleration of 0.25 m/s^2, measured relative to the rope. Determine the tension in the rope and the acceleration of the 80-kg block.



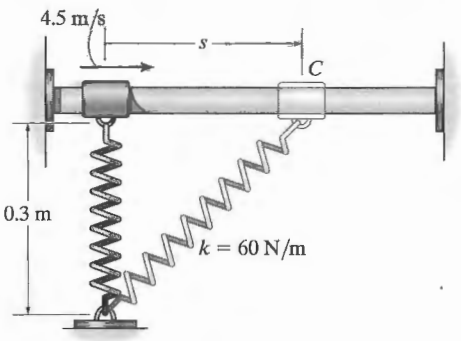
Prob. 13–31

***13–32.** Motor M draws in the cable with an acceleration of 1.2 m/s^2, measured relative to the 100-kg mine car. Determine the acceleration of the car and the tension in the cable. Neglect the mass of the pulleys.



Prob. 13–32

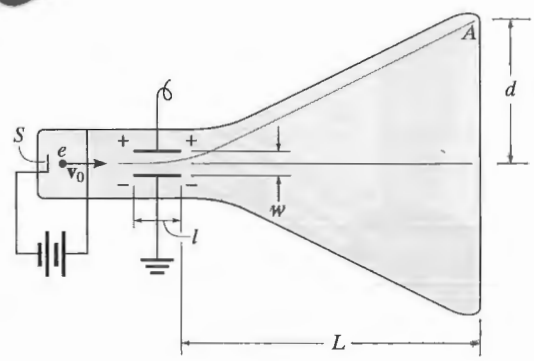
•**13–33.** The 1-kg collar C fits loosely on the smooth shaft. If the spring is unstretched when $s = 0$ and the collar is given a velocity of 4.5 m/s, determine the velocity of the collar when $s = 0.3$ m.



Prob. 13–33

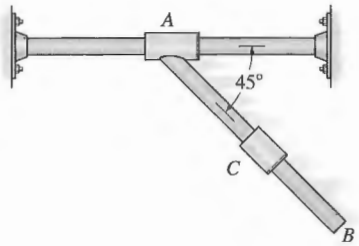
13–34. In the cathode-ray tube, electrons having a mass m are emitted from a source point S and begin to travel horizontally with an initial velocity $\mathbf{v}_0$. While passing between the grid plates a distance l, they are subjected to a vertical force having a magnitude eV/w, where e is the charge of an electron, V the applied voltage acting across the plates, and w the distance between the plates. After passing clear of the plates, the electrons then travel in straight lines and strike the screen at A. Determine the deflection d of the electrons in terms of the dimensions of the voltage plate and tube. Neglect gravity which causes a slight vertical deflection when the electron travels from S to the screen, and the slight deflection between the plates.

Video Solutions



Prob. 13–34

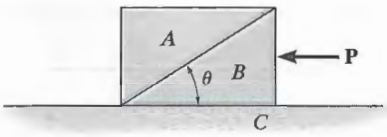
13–35. The 2-kg collar C is free to slide along the smooth shaft AB. Determine the acceleration of collar C if (a) the shaft is fixed from moving, (b) collar A, which is fixed to shaft AB, moves to the left at constant velocity along the horizontal guide, and (c) collar A is subjected to an acceleration of 2 m/s^2 to the left. In all cases, the motion occurs in the vertical plane.



Prob. 13–35

***13–36.** Blocks A and B each have a mass m. Determine the largest horizontal force $\mathbf{P}$ which can be applied to B so that A will not move relative to B. All surfaces are smooth.

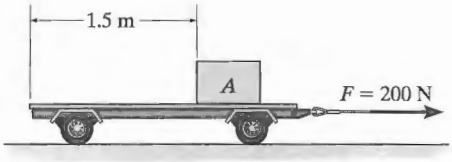
•**13–37.** Blocks A and B each have a mass m. Determine the largest horizontal force $\mathbf{P}$ which can be applied to B so that A will not slip on B. The coefficient of static friction between A and B is μ_s. Neglect any friction between B and C.



Probs. 13–36/37

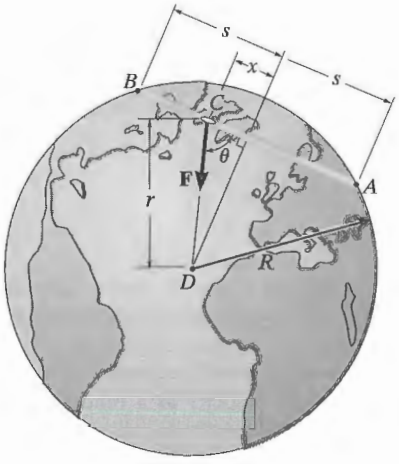
A publication from Pearson Custom Publishing exclusively for The University of Melbourne

13–38. If a force $F = 200$ N is applied to the 30-kg cart, show that the 20-kg block A will slide on the cart. Also determine the time for block A to move on the cart 1.5 m. The coefficients of static and kinetic friction between the block and the cart are $\mu_s = 0.3$ and $\mu_k = 0.25$. Both the cart and the block start from rest.



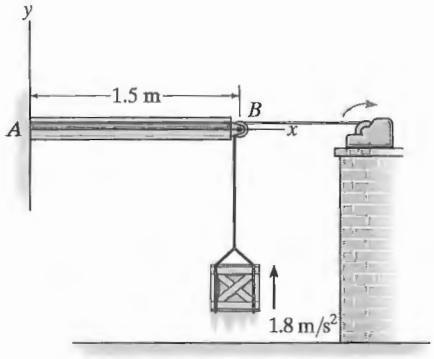
Prob. 13–38

13–39. Suppose it is possible to dig a smooth tunnel through the earth from a city at A to a city at B as shown. By the theory of gravitation, any vehicle C of mass m placed within the tunnel would be subjected to a gravitational force which is always directed toward the center of the earth D. This force **F** has a magnitude that is directly proportional to its distance r from the earth's center. Hence, if the vehicle has a weight of $W = mg$ when it is located on the earth's surface, then at an arbitrary location r the magnitude of force **F** is $F = (mg/R)r$, where $R = 6328$ km, the radius of the earth. If the vehicle is released from rest when it is at B, $x = s = 2$ Mm, determine the time needed for it to reach A, and the maximum velocity it attains. Neglect the effect of the earth's rotation in the calculation and assume the earth has a constant density. *Hint:* Write the equation of motion in the x direction, noting that $r \cos \theta = x$. Integrate, using the kinematic relation $v\,dv = a\,dx$, then integrate the result using $v = dx/dt$.



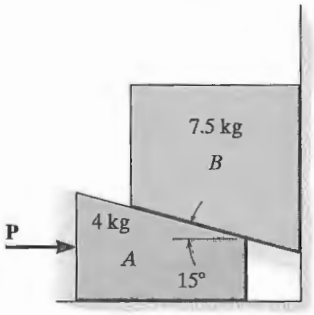
Prob. 13–39

***13–40.** The 15-kg crate is being hoisted upward with a constant acceleration of 1.8 m/s^2. If the uniform beam AB has a weight of 1000 N, determine the components of reaction at the fixed support A. Neglect the size and mass of the pulley at B. *Hint:* First find the tension in the cable, then analyze the forces in the beam using statics.



Prob. 13–40

•13–41. If a horizontal force of $P = 50$ N is applied to block A, determine the acceleration of block B. Neglect friction. *Hint:* Show that $a_B = a_A \tan 15°$.



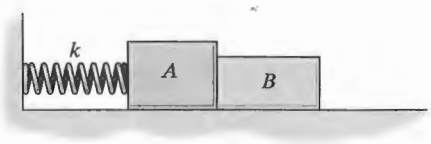
Prob. 13–41

Hibbeler, RC. (2012). Kinetics of a Particle: Force and Acceleration. In *Engineering Mechanics Dynamics*, (12th Ed. In SI Units, pp. 106-130). Singapore: Pearson Education South Asia Pte Ltd.

ENGINEERING
BRANCH LIBRARY

13–42. Block A has a mass m_A and is attached to a spring having a stiffness k and unstretched length l_0. If another block B, having a mass m_B, is pressed against A so that the spring deforms a distance d, determine the distance both blocks slide on the smooth surface before they begin to separate. What is their velocity at this instant?

13–43. Block A has a mass m_A and is attached to a spring having a stiffness k and unstretched length l_0. If another block B, having a mass m_B, is pressed against A so that the spring deforms a distance d, show that for separation to occur it is necessary that $d > 2\mu_k g(m_A + m_B)/k$, where μ_k is the coefficient of kinetic friction between the blocks and the ground. Also, what is the distance the blocks slide on the surface before they separate?



Probs. 13–42/43

***13–44.** The 600-kg dragster is traveling with a velocity of 125 m/s when the engine is shut off and the braking parachute is deployed. If air resistance imposed on the dragster due to the parachute is $F_D = (6000 + 0.9v^2)$ N, where v is in m/s, determine the time required for the dragster to come to rest.

Prob. 13–44

•13–45. The buoyancy force on the 500-kg balloon is $F = 6$ kN, and the air resistance is $F_D = (100v)$ N, where v is in m/s. Determine the terminal or maximum velocity of the balloon if it starts from rest.

Video Solutions

Prob. 13–45

13–46. The parachutist of mass m is falling with a velocity of v_0 at the instant he opens the parachute. If air resistance is $F_D = Cv^2$, determine his maximum velocity (terminal velocity) during the descent.

Prob. 13–46

13–47. The weight of a particle varies with altitude such that $W = m(gr_0^2)/r^2$, where r_0 is the radius of the earth and r is the distance from the particle to the earth's center. If the particle is fired vertically with a velocity v_0 from the earth's surface, determine its velocity as a function of position r. What is the smallest velocity v_0 required to escape the earth's gravitational field, what is r_{max}, and what is the time required to reach this altitude?