

**New Handbook of
Troubleshooting Techniques**
*for Microprocessors
and Microcomputers*

Robert G. Middleton

001233



**New Handbook of
Troubleshooting Techniques**
*for Microprocessors
and Microcomputers*

Robert G. Middleton

PRENTICE-HALL, INC.
Business & Professional Division

Englewood Cliffs,
New Jersey

Prentice-Hall International, Inc., London
Prentice-Hall of Australia, Pty., Ltd., Sydney
Prentice-Hall of Canada, Inc., Toronto
Prentice-Hall of India Private Ltd., New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Prentice-Hall of Southeast Asia, Pte., Ltd., Singapore
Whitehall Books, Ltd., Wellington, New Zealand
Editora Prentice-Hall do Brasil Ltda., Rio de Janeiro

© 1984, by

PRENTICE-HALL, INC.
Englewood Cliffs, N.J.

First Printing . . . October 1984

*All rights reserved. No part of this book
may be reproduced in any form or by any
means, without permission in writing
from the publisher.*

Editor: George E. Parker

Library of Congress Cataloging in Publication Data

Middleton, Robert Gordon

New handbook of troubleshooting techniques for micro-
processors and microcomputers.

Includes index.

1. Microcomputers--Maintenance and repair. 2. Micro-
processors--Maintenance and repair. I. Title.

TK7887.M54 1984 621.3819'58 84-9916

ISBN 0-13-613464-5

Printed in the United States of America

A Word from the Author

Microprocessors and microcomputers tend to seem rather arcane to those whose previous experience has been in linear (analog) and introductory digital technology. Evolution of large-scale and very-large-scale integration has compounded the outward mystery of microprocessors. This formidable aura rapidly disappears, however, when we observe that an elementary microcomputer is not difficult to understand, and that we can proceed in easy steps to comparatively sophisticated microprocessors and microcomputers.

This is the approach that is used in this book. It starts with a description of an "embryo" microprocessor, and shows how this simple device can add and subtract numbers. Since any microprocessor necessarily operates on binary numbers, the author introduces you to this topic at the outset. The text continues with a demonstration of multiplication and division by shift-left and shift-right action in an embryo microprocessor.

Next, we examine microprocessor building blocks—the basic gates. Fundamentals of gate troubleshooting are noted. From this point, you proceed step-by-step through the digital circuitry employed in microprocessor design. You become familiar with latches and flip-flops, along with reinforced learning provided by experimental projects and demonstrations. You then progress to more elaborate microprocessor subsections in the descriptions of counters, shift registers, and accumulators. Practical troubleshooting techniques are included along with explanations of circuit action.

With this basic foundation established, you are in good position to cope with the arithmetic subsection in a microprocessor. Thus, half adders and full adders are explained, and are supplemented by experimental projects. The fundamental 2's-complement adder/subtractor is detailed, and it is shown how this device processes both positive numbers and negative numbers with simple circuitry.

Building upon your knowledge of shift-register operation, a "painless" introduction to multiplication, division, and binary fractional processing is provided. Inasmuch as encoders and decoders are among the key subsections in microprocessor systems, their circuit action is described and supplemented by interesting experimental projects. ASCII, BCD, XS3, and other widely used codes are discussed, along with explanation of code-converter action. Standard troubleshooting techniques are illustrated, and case histories are presented to familiarize the reader with the "real world."

Multiplexers and demultiplexers are also key subsections in microprocessor systems, and examples of their circuitry are analyzed along with related data-selector and data-distributor applications. Clock skew troubleshooting notes are included, and oscilloscope tests are explained. The importance of static stimulus tests in preliminary troubleshooting procedures is illustrated.

Essential microprocessor comparator and parity generator/checker functions are explained, with notes on hardware operation in system environments. For example, a "firewalking" operating environment is noted wherein the display devices are greatly overdriven, but for such brief intervals that they are not damaged—although if the troubleshooter "stops the clock," the display devices would be subject to damage.

Experimental projects are included for reinforced learning. The reader proceeds with a practical overview of memory organization and operation. Widely used types of TTL, CMOS, and dynamic MOS memories are explained, and the various specialized troubleshooting requirements are noted. Terms that often confuse the beginner, such as "program memory," are clarified. Practical considerations in handling and replacement of MOS memory devices are included.

With this established perspective, you are in good position to cope with the elements of microprocessor architecture. This discussion starts with a "chopped down" version of a microcomputer that includes the minimum hardware requirements for stored-program addition of a pair of binary numbers. It is supplemented by an experimental microcomputer project which clearly illustrates "bare-bones" architecture. Notes are included on bugging and debugging the elementary system.

Next, you are introduced to buffer registers and to register-transfer action in bus-oriented microcomputers. It is shown how the program counter is the "heart" of microprocessing, and how it directs RAM or ROM address-decoding and unloading action. An elementary diagnostic

program for microcomputer troubleshooting is introduced, plus notes on the fundamentals of programming.

Instruction sets are described with detailed explanations of the fetch cycle, address phase, increment phase, and execution cycle. Basic LDA, ADD, SUB, and OUT routines are analyzed, with notes on instruction decoder and control matrix operation. At this point, you are briefly introduced to the logic analyzer, with examples of its importance in analysis of data-processing operations. Both hardware faults and program "bugs" are exemplified.

Progressively elaborated microprocessor architectures are described, with detailed examples of jump instructions and subroutine counter operations. Commercial arithmetic-logic units are illustrated. Index and output register operation is followed, along with explanation of compiler and assembler operation. At this point, you will have a good understanding of machine language and high-level language. Note in passing that a programmer does not need to know how a microcomputer works. On the other hand, the troubleshooter must know how programs are written, and how they are processed by a microcomputer.

"New Microcomputer Test Equipment You Can Build" is a helpful "plus" in this text—it shows the reader how to construct simple and useful microcomputer testers, and how to apply them. For example, a charge-storage logic-level beeper is described which is both economical and highly utilitarian. Troubleshooters with commercial microcomputer test equipment available will nevertheless be interested in some of the novel approaches that are explained, such as construction and use of a differential temperature probe.

Accordingly, this book will serve as an excellent training manual for anyone who wishes to understand how microprocessors and microcomputers work, and how to troubleshoot them. Because the text starts at the beginning and assumes no prior knowledge of the art, it is easily comprehensible by prep-school students and hobbyists. However, the reader should have a background of introductory physics; familiarity with the new math will also be helpful.

The most important new-math topic, from the troubleshooter's viewpoint, is base-2 arithmetic. Other important topics include the hexadecimal system (base-16 arithmetic), and the octal system (base-8 arithmetic). Discussion of data is also of importance.

Professional microcomputer troubleshooters know that knowledge is power, and that time is money. Their goal is an effective mastery of

microprocessor and microcomputer operation, with knowledge of troubleshooting techniques and recognition of malfunction symptoms in defective equipment. Your success in the field of computer technology is limited only by the horizons of your technical know-how. This readily understandable text, with its profuse illustrations, troubleshooting explanations, and case histories, provides key stepping-stones to your goal.

Robert G. Middleton

CONTENTS

A Word from the Author	5
Chapter 1: MICROPROCESSOR AND MICROCOMPUTER FUNDAMENTALS	13
Overview * Embryo Microprocessor * Up-Counter Action * Counter Troubleshooting * Down-Counter Action * Decimal Readout * Rules for Binary Addition * Rules for Binary Subtraction * Introduction to Binary Fractions * Dedicated Versus General-Purpose Microprocessing	
Chapter 2: GATES AND BASIC DIGITAL-LOGIC DEVICES ...	27
Microprocessor Building Blocks * AND, OR, NAND, NOR, XOR, and XNOR Gates * Gate Arrangements as Equivalent Buffers or Inverters * A AND-NOT B, and A OR-NOT B Gates * Gates with All Inputs Inverted * Equivalent Gates * AND-OR Gates * AND-OR-INVERT Gates * NAND Implementation * Experimental Project	
Chapter 3: BASIC LATCHES AND FLIP-FLOPS	49
Latches Configured from Cross-Connected Gates * Microprocessor Sections * Experimental Project * Gated Latches * Flip-Flops and Clocks * Master-Slave Flip-Flops * JK Flip-Flops * Preset and Clear Functions * Troubleshooting Considerations * Artificial Intelligence	
Chapter 4: COUNTERS AND ASSOCIATED CIRCUITRY	67
Basic Microprocessing Devices * Synchronous Versus Asynchronous Counters * Microprocessor Subsections * Experimental Project * Synchronous and Asynchronous Down-Counters * Even- and Odd-Mod Counters * Programmable Counter * BCD Counter * 2421 Counter * Troubleshooting Notes	

Chapter 5:	SHIFT-REGISTER CIRCUITRY AND OPERATION	84
	Basic Microprocessor Organization * Shift-Register Principles * Weighted Coding * Shift-Register Circuitry * Experimental Project * Feedback Shift Register * Universal Shift Register * Self-Correcting Ring Counter * Pseudo-Random Binary Sequence Generator * Typical Microprocessor System Faults.	
Chapter 6:	ADDERS AND SUBTRACTERS	102
	Microprocessing Arithmetic Devices * Half and Full Adders * Boolean Algebra * Serial Adder * Parallel Adder * BCD Adder * 2's Complement Adder/Subtractor * Algebraic Addition/Subtraction * Troubleshooting Notes	
Chapter 7:	MULTIPLICATION AND DIVISION OPERATIONS	124
	Microprocessor ALU Section * Shift-and-Add Multiplier Action * BCD Multiplication * Look-Up Tables * Shift-and-Subtract Divider Action * Continued Fractions * Divider Hardware * Troubleshooting Approach * The Inside Story * ASCII Code	
Chapter 8:	ENCODERS AND DECODERS	142
	Microprocessor Bus Activity * 2-to-4-Line Decoder * BCD-to-Decimal Decoder * Binary Number Decoder * 1-out-of-16 Decoder * 2'421-to-8421 Code Converter * Keyboard-to-BCD Encoder * Binary-to-Seven-Segment Encoder * Priority Encoder * Troubleshooting Considerations * Low-Impedance Gate Inputs	
Chapter 9:	MULTIPLEXERS AND DEMULTIPLEXERS	160
	Microprocessor Bus Lines * Eight-Channel Multiplexer Arrangement * Eight-Channel Demultiplexer Arrangement * Three-State Buffers * Virtual Multiplexer * Multiplexed Seven-Segment Display * Troubleshooting Notes * Novel Scanning-Action Quick Test * Disassembly Precautions * Measure V_{CC} First—and Also Check for Ripple	
Chapter 10:	COMPARATORS AND PARITY GENERATOR/CHECKERS	174
	Data Sources and Destinations * Comparator Circuitry and Operation * Wired-AND (Wired-OR) * Polarity Comparator * Phase Comparator * Parity Generator Checkers * Parity Trees * Troubleshooting Notes * Control Read-Only Memory * Control Random Access Memory	

Chapter 11: MEMORY ORGANIZATION AND OPERATION	190
Microcomputer Organization * Semiconductor Memory Principles * Instruction Register * Memory Address Register * Read-and-Write Memory Operation * Recirculating (Serial) Memory * "Embryo" Computer Experiment * Troubleshooting Considerations * IC and Card "Swapping" * Some Ground Rules	
Chapter 12: ELEMENTS OF MICROPROCESSOR ARCHITECTURE	207
Microprocessor Organizational Structure * Operation of a Simple Microprocessor System * Programming Topics * Programmed Addition of a Pair of Numbers * Programmed Instruction Words * Recap of Operating Sequence * Gating of Instruction Words * Hexadecimal Numbers	
Chapter 13: ELABORATED MICROPROCESSOR ARCHITECTURE	225
Starting of a Program Run * Instruction Word Flow * Operation of Instruction Decoder * Troubleshooting Notes * Programming Pointers * Ring Counter in Control Section * Operation of Control Matrix * Mnemonics and Assembly Language * High-Level Language * Short-hand Dialect * Simple Program Examples * Troubleshooting Note	
Chapter 14: STORED-PROGRAM PROCESSING	240
Microprocessor Characteristics * Instruction Set * Microprocessor System * Program Counter * Interrupts * Instruction Cycle * Accumulator Group Instruction Set * I/O and RAM Instruction Set * Programming Trick of the Trade * Language Selection * Historical Note	
Chapter 15: NEW MICROCOMPUTER TEST EQUIPMENT YOU CAN BUILD	262
Charge-Storage Logic-High Beeper * Differential Temperature Checker * Adjustable Single-Shot Pulser * Digital System Noise * Short-Circuit Localization * Troubleshooting Short-Circuited PC Conductors * Circuit Internal Resistance * RFI Checkout	
Glossary	273
Index	285



Microprocessor and Microcomputer Fundamentals

Overview * Embryo Microprocessor * Up-Counter
Action * Counter Troubleshooting * Down-Counter
Action * Decimal Readout * Rules for Binary Addition
* Rules for Binary Subtraction * Introduction to Binary
Fractions * Dedicated Versus General-Purpose Micro-
processing

Overview

Microprocessors and microcomputers are not the mysterious devices and machines that elude understanding and defy troubleshooting, as popular news writers might lead their readers to suppose. Quite to the contrary, as explained in the following chapters, microprocessors and microcomputers are as easy to understand as they are interesting, and they are no more difficult to troubleshoot than is a color-TV receiver, for example.

A microprocessor is exemplified in Fig. 1-1. It is a large-scale integrated circuit (LSI) which adds, subtracts, multiplies, divides, and performs logical operations such as: *If A is less than B, then transfer program control to line 29.* All arithmetical and logical operations are accomplished by means of microscopic electronic switching circuits—whence the “micro” terminology.

A microcomputer is illustrated in Fig. 1-2. Its “heart” is a microprocessor which is controlled by a stored program. This stored program is determined by the keyboard operator (programmer), and is stored in an electronic memory. The microcomputer has no intelligence (in the human sense of the term), and it can process data only as commanded by the programmer.

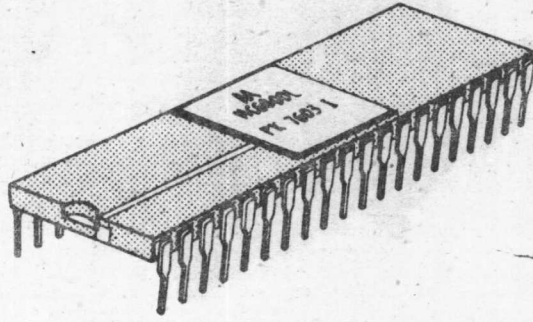
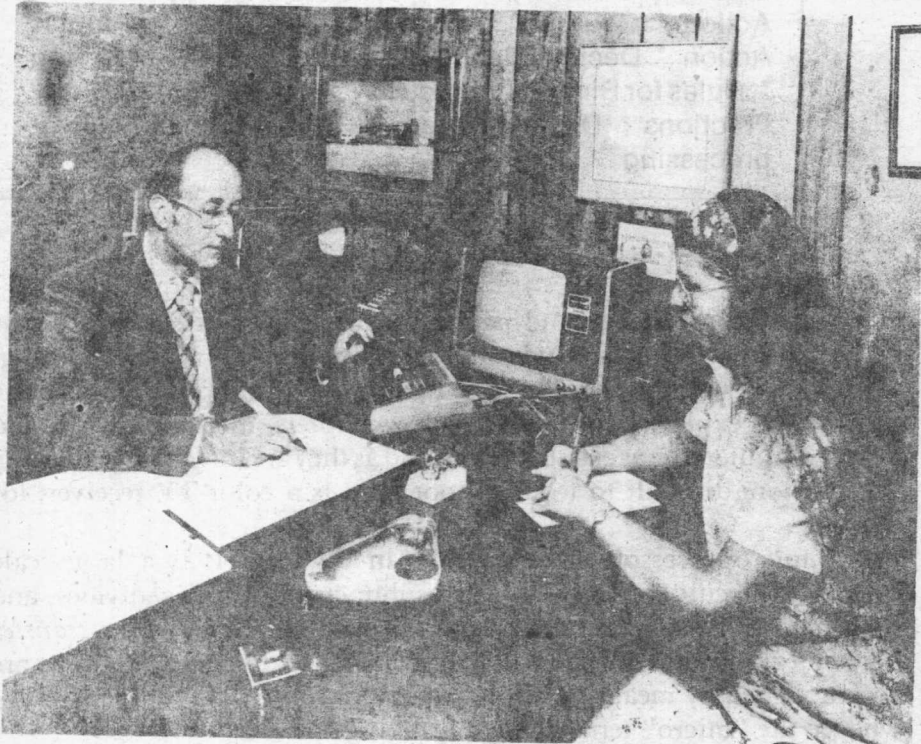


Fig. 1-1. A typical microprocessor IC package.



Radio Shack TRS-80 Microcomputer System

Fig. 1-2. The "heart" of a microcomputer is a microprocessor (Courtesy, Radio Shack, a Division of Tandy Corp).

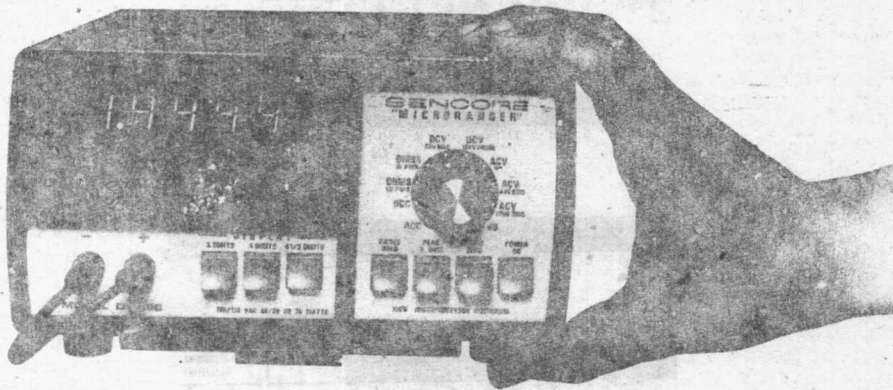


Fig. 1-3. A digital voltmeter that employs a microprocessor (Courtesy, Sencore).

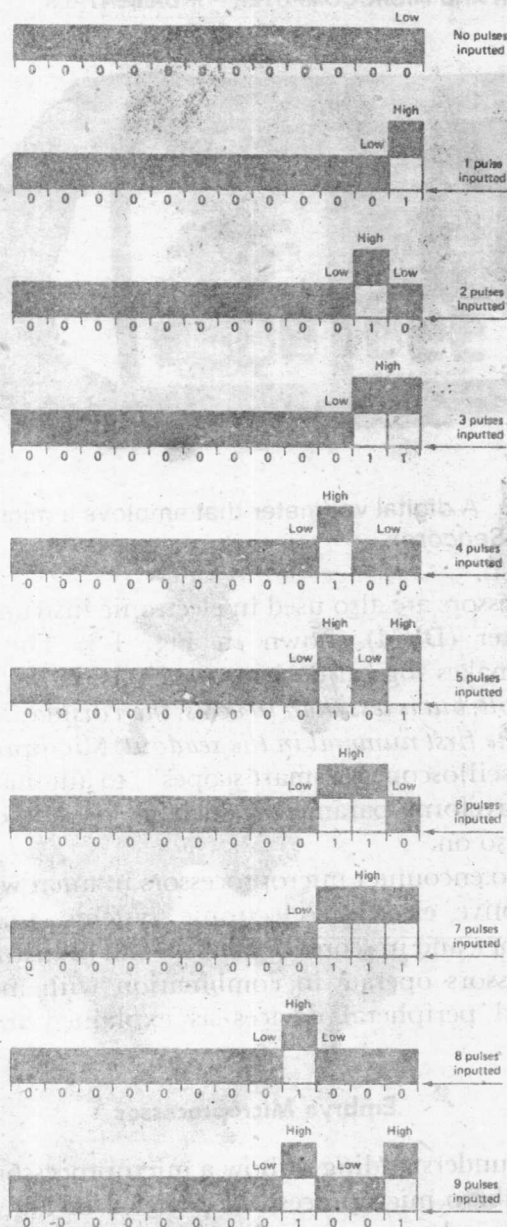
Microprocessors are also used in electronic instruments such as the digital voltmeter (DVM) shown in Fig. 1-3. The microprocessor automatically makes logical decisions such as: *If the input voltage is greater than 1 volt, but is less than 10 volts, then display the decimal point to the right of the first numeral in the readout.* Microprocessors are used in intelligent oscilloscopes ("smart scopes") to automatically calculate and display waveform parameters such as rise time, peak voltage, frequency, and so on.

We will also encounter microprocessors in microwave ovens, video games, automotive electrical/electronic systems, traffic controllers, digital calculators, and in scores of medical and industrial applications. All microprocessors operate in combination with memory, control, interfacing, and peripheral devices as explained in the following chapters.*

Embryo Microprocessor

For a basic understanding of how a microprocessor adds numbers, observe the "embryo microprocessor" depicted in Fig. 1-4. This is an example of a digital up-counter. It consists of a chain of electronic switches (12 switches, in this example). At the start, all of the switches are off, or their outputs are logic-low. When the first pulse is inputted, the first switch goes logic-high; the remainder of the switches remain logic-low.

*If you want to know more about IC's used in microprocessor systems, refer to pages 111-344 in *Encyclopedia of Integrated Circuits*, by Walter H. Buchsbaum, Sc.D.



Note: A binary counter also functions as a sequential adder. For example, if two pulses are inputted, the readout will be 10; then, if three more pulses are inputted, the readout will be 101 ($10 + 11 = 101$). This sum is retained as long as no further pulses are inputted. In other words, the counter also functions as a memory.

Fig. 1-4. Binary counter operation.