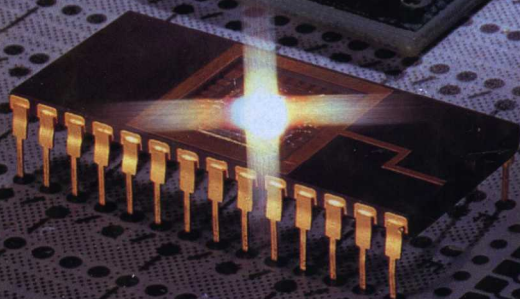


INTRODUCTION TO

MICROPROCESSORS

USING THE
MC6809
OR THE
MC68000



RALPH HORVATH

INTRODUCTION TO MICROPROCESSORS USING THE MC6809 OR THE MC68000

Ralph Horvath

*College of Engineering
Michigan Technological University*

McGraw-Hill, Inc.

New York St. Louis San Francisco Auckland Bogotá
Caracas Lisbon London Madrid Mexico Milan Montreal
New Delhi Paris San Juan Singapore Sydney Tokyo Toronto

This book was set in Times Roman by Publication Services, Inc.
The editors were Anne T. Brown and John M. Morriss.
The cover was designed by Carol Couch.
Project supervision was done by Publication Services, Inc.
R. R. Donnelley & Sons Company was printer and binder.

Cover photograph: © Srulik Haramaty/Phototake NYC.

INTRODUCTION TO MICROPROCESSORS USING THE MC6809 OR THE MC68000

Copyright © 1992 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

2 3 4 5 6 7 8 9 0 DOC DOC 9 0 9 8 7 6 5 4 3 2

ISBN 0-07-030507-2

Library of Congress Cataloging-in-Publication Data

Horvath, Ralph.

Introduction to microprocessors using the MC6809 or the MC68000/

Ralph Horvath.

p. cm. — (McGraw-Hill series in electrical and computer
engineering. Computer engineering)

Includes index.

ISBN 0-07-030507-2

1. Motorola 6809 (Microprocessor) 2. Motorola 68000
(Microprocessor) I. Title. II. Series.

QA76.5.H72 1992

004.165—dc20

91-42488

PREFACE

INTENDED AUDIENCE

An Introduction to Microprocessors Using the MC6809 or the MC68000 is intended for the typical sophomore engineering student who has had little formal electronic training and whose computer experience is probably limited to elementary programming in a higher level language such as FORTRAN or Pascal. In addition, it should be useful to engineers in the field as well as to managers who need an introduction to the area of microprocessors. It will provide any engineer with an adequate background so that he or she can intelligently use, program, modify, and maintain microprocessor systems or supervise people who do.

DEPTH OF COVERAGE

The approach emphasizes the point of view of the microprocessor system user. It introduces only those facts which will be of value to the engineer who must program, use, or maintain microprocessor systems or design hardware to couple with a microprocessor chip. It does not delve into the internal electronic aspects of the integrated circuits involved.

Many introductory books on microprocessors go into much more detail in logic design than is necessary for such a presentation, or they presume a strong background in logic design and relegate that subject to an appendix. These books are written primarily for electrical engineering (or computer engineering) students who are emphasizing the computer area and have a strong interest in understanding the internal workings of microprocessor chips. Not all electrical engineers and very few non-electrical engineers need this sort of preparation.

The coverage of logic circuits in this book is just sufficient to provide a satisfactory basis for an understanding of microprocessors. It is not intended to replace the traditional electrical engineering logic and switching course. In fact, it may serve to stimulate the reader to study this field in more depth.

The book introduces the reader to combinational circuits from a functional point of view with the emphasis on understanding the sorts of things they can accomplish. No design concepts are introduced other than to describe the minimization problem.

The discussion of sequential circuits is a descriptive one with enough detail to show the reader how simple registers can be constructed from flip-flops and gates. These topics have been included because I feel that at least a minimal coverage of both combinational and sequential logic circuits is a necessary prerequisite for a basic understanding of microprocessors.

TARGET PROCESSOR(S)

As is stated in the title, the book introduces two Motorola microprocessors as examples or targets, the MC6809 and the MC68000. The MC6809 is a very important processor from the pedagogical point of view. It is a simple enough machine to serve very well as an introductory vehicle to the world of microprocessors, yet it is complex enough to find extensive use in control applications in the real world. The MC68000 is the senior member of one of the major families of processors used in current personal computer systems and it too is used extensively in real time control applications.

For the first exposure to the material in the book, the reader should select one of the two processors and concentrate on it to the exclusion of the other. It would be a difficult and discouraging task to attempt to learn the details of both processors simultaneously. To support this approach, all processor specific sections, tables, figures, and problems are clearly labeled as such. When studying the details of one processor, those segments dealing with the other may be readily identified and omitted.

The choice of which processor to study initially will probably be determined by the availability of a laboratory system which may be used to prepare and run programs. It is very important that the student have such a system for the first (or only) processor to be studied.

After the initial exposure and after gaining an understanding of the details of the selected processor, the other may be studied for the purpose of contrast and comparison. However, such a comparison is not as important as the depth of coverage of the primary processor and the availability of a suitable laboratory platform.

ORGANIZATION OF MATERIAL

Following a brief presentation of the general aspects of computer systems, the book introduces the reader to the two primary digital areas of logic circuits and number systems. The basic aspects of microprocessors are then presented. As each new topic is introduced it is first discussed in a general way. The terms are defined, general examples are shown, and the fundamental concepts are established. Then, the specifics of the topic as related to the MC6809 are discussed followed by a similar presentation for the MC68000. The specifics are presented in such a way that either could stand alone; i.e., the coverage of each processor is independent of the other.

Following a brief historical overview, Chapter 1 introduces some very basic and important concepts and terminology. This chapter must be covered carefully and in detail, as it forms the basis for much of the material of the book.

Chapters 2 and 3 provide the introduction to logic circuits and number systems. These chapters include the minimum required exposure to these basic subjects which

are so important to understanding the subsequent material. Unless the subjects have been covered in a prior course, these chapters must also be covered in detail.

Chapter 4 describes the architectural and functional structures of microprocessors in general, including such concepts as the programming model, accumulator and general register processors, condition codes, the fetch/execute cycle, basic memory types, and bus cycles. It ends with a presentation of the programming models and simple example programs for each of the target processors.

Chapter 5 discusses instruction sets and addressing modes in general and includes a detailed description of the modes available with the target processors. The emphasis is on describing the sorts of things which the individual machine language instructions may accomplish in order to support the discussion of the details of the addressing modes. Understanding the addressing modes in detail is of particular importance. The lack of such an understanding is often the major stumbling block to the student's grasp of the subsequent material on assembly language programming.

Chapter 6 introduces assembly language programming and presents a more detailed discussion of the instruction sets of the target processors, with the emphasis on their mnemonic forms and addressing mode limitations.

Chapters 5 and 6 are by far the longest chapters in the book. The material in them should be thoroughly understood by the student before going on to the subsequent material. Some instructors may prefer to cover the latter part of Chapter 6 before introducing assembly language programming. However, I recommend the order followed in the book.

Chapter 7 presents several examples of program segments which illustrate such things as double precision and BCD arithmetic, loop structures, relocatability, bit level manipulations, and BCD/binary number conversions.

Chapter 8 introduces stacks and their uses and includes extensive subroutine techniques, including parameter passing and dynamic memory allocation.

Structured modular programming is introduced in Chapter 9. This programming discipline is intentionally introduced late in the book. I feel that the student must have a thorough understanding of the capabilities of the processor before the material in this chapter can be of benefit. However, some instructors may prefer to present this material earlier, perhaps immediately after Chapter 6. It has been written in such a way as to support this approach.

Chapters 10 and 11 describe the general aspects of I/O and interrupt structures and present the details of these structures as implemented in the target processors. In addition, Chapter 10 introduces the three primary Motorola interface chips: the PIA, the PTM, and the ACIA. Chapter 12 presents some additional material on memory types and structures including address decoding and interfacing. The material in these three chapters is less detailed than earlier material. Some of it may be deferred to a follow-on interfacing course. Alternatively, the material may be supplemented with detailed manufacturer's specifications on specific chips and expanded to include some interfacing techniques.

Chapter 13 includes the pin-by-pin hardware descriptions of the target processors, including instruction timing and bus cycle timing. Chapter 14 introduces some of the more common development tools. Some or all of this material need not be covered formally and may be left for a reading assignment. The material in Chapter

13 should be covered in detail if the choice is made to expand the material to include interfacing.

LABORATORY WORK

When first learning about microprocessors it is important that the student be able to perform supporting experiments as new material is presented. Such experiments may be done on real hardware such as logic kits and microprocessor educational kits. On the other hand, they may be simulated quite well on a general purpose computer. Either approach seems to work satisfactorily, although many instructors and students do prefer to see the actual components.

Two simulator programs are available for use with this book, ASSYM000 for the MC68000 and ASSYM09 for the MC6809. Each program can assemble a source program into machine code and then simulate the execution of the program on its processor. They run in the DOS operating system on any IBM compatible personal computer. These programs accompany the instructor's problem-solution manual which may be obtained from McGraw-Hill when the book is adopted for classroom use.

The assembler/simulator programs were written at Michigan Technological University and rights to the programs are retained by the university. They may be loaded on only one computer with copies allowed only for back-up purposes. Additional copies may be obtained by contacting Michigan Technological University at (906) 487-2487.

ACKNOWLEDGMENTS

McGraw-Hill and I would like to thank the following reviewers for their many helpful comments and suggestions: Ashok Krishnamurthy, Ohio State University; Stephen Prigozy, United States Merchant Marine Academy; Barna Szabados, McMaster University; and William Trotter, Jr., University of Maryland.

Ralph Horvath

CONTENTS

Preface	xvii
1 Introduction to Computers	1
1.1 Historical Background	1
1.2 The Digital Computer	4
1.2.1 The PSBE Machine	4
1.2.2 The Functional Structure of a Computer	5
1.2.3 Basic Operations	7
1.3 Hardware/Software	9
1.3.1 Computer Hardware	9
1.3.2 Computer Software	10
1.4 The Motorola MC6809 and MC68000 Processors	10
Summary	11
Review Problems	11
2 Logic Circuits	13
2.1 The Basic Gates	13
2.1.1 Binary Variables	14
2.1.2 The AND Gate	14
2.1.3 The OR Gate	16
2.1.4 The Inverter	17
2.1.5 Some Other Gates	18
2.2 Boolean Algebra	19
2.2.1 Boolean Functions and Equations	19
2.2.2 Perfect Induction	20
2.2.3 The Rules of Boolean Algebra	21
2.2.4 Circuit Representations	22
2.3 Combinational Logic Circuits	23
2.3.1 Simplification of Circuits	23
2.3.2 Decoders	25
2.3.3 The Full Adder	26
2.4 Basic Flip-Flops	27
2.4.1 The R-S Flip-Flop	27

2.4.2	The Clocked R-S Flip-Flop	29
2.4.3	Timing Diagrams	29
2.4.4	The J-K Flip-Flop	31
2.4.5	The D-Type Flip-Flop	32
2.5	Sequential Logic Circuits	33
2.5.1	Registers and Memory	33
2.5.2	Shift Registers	34
2.5.3	Counters	35
	Summary	37
	Review Problems	38
3	Bit Patterns	43
3.1	Number Systems	43
3.1.1	Positional Number Systems	43
3.1.2	Nondecimal Number Systems	44
3.1.3	The Binary Number System	46
3.1.4	The Octal and Hexadecimal Number Systems	47
3.2	Addition in Nondecimal Systems	49
3.2.1	Decimal Addition	49
3.2.2	Hexadecimal Addition	49
3.2.3	Binary Addition	50
3.3	Binary Subtraction	52
3.3.1	Unsigned Integers	52
3.3.2	Sign-Magnitude Notation	52
3.3.3	Two's Complement Signed Numbers	52
3.3.4	The Addition of Two's Complement Numbers	54
3.4	Floating Point Representation	58
3.4.1	Fixed Point Numbers	58
3.4.2	Floating Point Numbers	58
3.4.3	Binary Floating Point Numbers	59
3.5	Binary Coded Decimal	60
3.6	Character Codes	61
3.7	Error Detection	63
	Summary	64
	Review Problems	65
4	The Modus Operandi of Microcomputers	69
4.1	The Basic Architecture of a Computer	69
4.2	Memory and the Read/Write Cycle	71
4.2.1	Memory Organization	71
4.2.2	Basic Memory Types	73
4.2.3	Bus Cycles	74
4.2.4	Memory Contents	76
4.3	The Program Counter	77
4.4	The Fetch/Execute Cycle	77
4.5	The Accumulator	78
4.6	Data Registers	78
4.7	Address Registers	79
4.8	Condition Code Bits	79

4.9	Other Registers	80
4.10	The Programming Model	81
4.10.1	The Motorola MC6809 Programming Model	81
4.10.2	A First Example Program for the MC6809	83
4.10.3	The Motorola MC68000 Programming Model	85
4.10.4	A First Example Program for the MC68000	85
	Summary	89
	Review Problems	89
5	Instructions and Addressing Modes	91
5.1	Types of Instructions	91
5.1.1	Load/Store Instructions	92
5.1.2	Arithmetic/Logic Instructions	93
5.1.3	Test/Branch Instructions	95
5.1.4	Input/Output Instructions	95
5.2	Addressing Modes	95
5.2.1	Direct Addressing	96
5.2.2	Immediate Addressing	97
5.2.3	Indirect Addressing	98
5.2.4	Multi-Component Addressing Modes	100
5.2.5	Implied or Inherent Addressing	103
5.3	The MC6809 Addressing Modes	104
5.3.1	Extended Addressing in the MC6809	104
5.3.2	Immediate Addressing in the MC6809	106
5.3.3	Direct Addressing in the MC6809	107
5.3.4	Relative Addressing in the MC6809	109
5.3.5	Indexed Addressing in the MC6809	111
5.3.6	Auto Increment/Decrement Options in the MC6809	113
5.3.7	The Indirect Options in the MC6809	114
5.4	The MC68000 Addressing Modes	116
5.4.1	Absolute Addressing in the MC68000	118
5.4.2	Immediate Addressing in the MC68000	119
5.4.3	Register Direct Addressing in the MC68000	120
5.4.4	Address Register Indirect Addressing in the MC68000	120
5.4.5	Address Register Indirect with Auto Increment/Decrement in the MC68000	121
5.4.6	Address Register Indirect with Offset/Index in the MC68000	123
5.4.7	Program Counter Relative Addressing Modes in the MC68000	124
	Summary	127
	Review Problems	127
6	Assembly Language Programming	131
6.1	Computer Languages	132
6.1.1	Machine Language	132
6.1.2	Assembly Language	132
6.1.3	Assembly Language versus Higher-Level Languages	133
6.1.4	Characteristics of Assembly Languages	134
6.2	Motorola Assembly Languages	135
6.2.1	The Character Set	136

6.2.2	Program Statements	136
6.2.3	The Label Field	137
6.2.4	The Operation Mnemonic Field	138
6.2.5	Assembler Directives	138
6.2.6	The Operand Field	141
6.2.7	The Comment Field	143
6.3	The MC6809 Instruction Set	143
6.3.1	The MC6809 Data Movement Instructions	144
6.3.2	The MC6809 Arithmetic Instructions	145
6.3.3	The MC6809 Logic Instructions	148
6.3.4	The MC6809 Test/Branch Instructions	151
6.3.5	The MC6809 No-operation Instruction	152
6.4	The MC68000 Instruction Set	153
6.4.1	The MC68000 Data Movement Instructions	154
6.4.2	The MC68000 Arithmetic Instructions	156
6.4.3	The MC68000 Logic Instructions	160
6.4.4	The MC68000 Test/Branch Instructions	162
6.4.5	The MC68000 Input/Output Instructions	164
6.4.6	Miscellaneous MC68000 Instructions	165
	Summary	165
	Review Problems	166
7	Elementary Program Examples	173
7.1	Multiple Precision Arithmetic	173
7.1.1	Double-Precision Addition with the MC6809	174
7.1.2	Double-Precision Addition with the MC68000	175
7.2	Moving Tables in Memory, Simple Loops	177
7.2.1	The MC6809 Program to Move a Table	177
7.2.2	The MC68000 Program to Move a Table	179
7.3	Relocatable Programs, BCD Arithmetic	180
7.3.1	BCD Addition in the MC6809	181
7.3.2	BCD Addition in the MC68000	182
7.4	Bit Picking and Bit Packing	183
7.4.1	MC6809 Examples of Bit Manipulations	184
7.4.2	MC68000 Examples of Bit Manipulations	185
7.5	Conversion Between Decimal and Binary (Hex)	187
7.5.1	MC6809 BCD/Binary Conversion Routines	187
7.5.2	MC68000 BCD/Binary Conversion Routines	189
	Summary	190
	Review Problems	191
8	Subroutines and Stacks	193
8.1	Subroutines	193
8.2	The Last in First out Stack	194
8.3	Stack Applications	196
8.4	Stacks in the MC6809	197
8.5	Examples of Stack Operations in the MC6809	199
8.5.1	Stack Pointer Initialization in the MC6809	199

8.5.2	Parameter Passing in the MC6809	200
8.5.3	Allocating Memory Space Dynamically in the MC6809	203
8.5.4	Register Use in MC6809 Subroutines	203
8.6	Stacks in the MC68000	205
8.7	Examples of Stack Operations in the MC68000	207
8.7.1	Initialization of the MC68000 Stack Pointers	207
8.7.2	Passing Parameters in the MC68000	209
8.7.3	Allocating Memory Space Dynamically in the MC68000	210
	Summary	213
	Review Problems	213
9	Software Development	219
9.1	The Software Development Process	219
9.2	Programming Languages	220
9.2.1	Machine Languages	221
9.2.2	Assembly Languages	221
9.2.3	Higher-Level Languages	222
9.3	Programming Goals	223
9.4	Flowcharts	225
9.5	Structured Modular Programming	226
9.5.1	Program Structures	228
9.5.2	Modular Programming	231
9.5.3	Top-Down Design, Bottom-Up Implementation	231
9.5.4	SORTLIST, A Programming Example	232
9.6	Programming Tools	239
9.6.1	Text Editors	239
9.6.2	Assemblers and Compilers	240
	Summary	241
	Review Problems	241
10	Input/Output Structures	245
10.1	Accessing I/O	246
10.1.1	Special I/O	246
10.1.2	Memory-Mapped I/O	248
10.2	Initiation and Control of I/O	249
10.2.1	Programmed I/O	249
10.2.2	Interrupt I/O	250
10.2.3	Direct Memory Access	252
10.3	Formats for Data Transfer	253
10.3.1	Parallel Data Transfer	253
10.3.2	Serial Data Transfer	254
10.3.3	Synchronous Serial Data Transfer	254
10.3.4	Asynchronous Serial Data Transfer	255
10.3.5	Data Rate versus Bit Rate	256
10.4	Interfaces	256
10.4.1	Functions Required of Interfaces	257
10.4.2	Interface Chips	257
10.4.3	The MC6821 Peripheral Interface Adapter	259

10.4.4	The MC6840 Programmable Timer Module	261
10.4.5	The MC6850 Asynchronous Communications Interface Adapter	262
10.4.6	Other Programmable Interface Chips	263
10.5	Standard Buses	263
	Summary	264
	Review Problems	264
11	Interrupts and Exception Processing	267
11.1	The Interrupt Process	267
11.2	Vectored Interrupt	270
11.3	Prioritization and Polling	271
11.4	The Interrupt Service Routine	272
11.5	Other Exceptions	273
	11.5.1 Reset	273
	11.5.2 Traps	274
	11.5.3 Software Interrupts	275
	11.5.4 The Trace Exception	275
	11.5.5 System Failure Exceptions	275
11.6	The MC6809 Interrupt Structure	275
	11.6.1 General Structure of the MC6809 Interrupt System	275
	11.6.2 The MC6809 Reset	276
	11.6.3 The MC6809 Nonmaskable Interrupt	277
	11.6.4 The MC6809 Maskable Interrupts	277
	11.6.5 The MC6809 Software Interrupts	278
	11.6.6 Special Interrupt Instructions in the MC6809	279
11.7	The MC68000 Exception Structure	280
	11.7.1 General Structure of the MC68000 Exception System	281
	11.7.2 The MC68000 Reset	284
	11.7.3 The MC68000 Interrupts	284
	11.7.4 The MC68000 Instruction Exceptions	285
	11.7.5 The MC68000 Trace Exception	287
	11.7.6 The MC68000 Hardware Exceptions	287
	Summary	287
	Review Problems	288
12	Memory Structures	291
12.1	Integrated Circuit Memory	291
	12.1.1 The Functional Structure of a Memory Chip	292
	12.1.2 The Address Decoder	292
	12.1.3 The Memory Matrix	293
	12.1.4 Tri-State Buffers and Chip-Select Inputs	295
12.2	Read Only Memory Types	296
	12.2.1 Custom-Masked ROMs	297
	12.2.2 Fused-Link PROMs	297
	12.2.3 Ultraviolet Erasable PROMs	297
	12.2.4 Electrically Erasable PROMs	298
12.3	Read/Write Memory Types	299
	12.3.1 Dynamic RAMs	299
	12.3.2 Static RAMs	302

12.4	Other Types of Main Memory	302
12.4.1	Core Memory	302
12.4.2	Battery-Backed CMOS RAM	303
12.4.3	Shadow RAM	303
12.5	Chip Select Decoding	304
	Summary	307
	Review Problems	307
13	Microprocessor Chip Hardware	313
13.1	Instruction Timing	313
13.1.1	Timing in the MC6809	314
13.1.2	MC6809 Instruction Execution Time	316
13.1.3	Timing in the MC68000	318
13.2	Pin Designations and Uses	319
13.2.1	The MC6809 Chip	320
13.2.2	The MC68000 Chip	321
	Summary	323
	Review Problems	323
14	Tools for System Design	325
14.1	Software Design Tools	325
14.1.1	Compilers and Assemblers	326
14.1.2	Debuggers	326
14.1.3	Simulators	327
14.2	Hardware Design Tools	329
14.2.1	Evaluation Kits	329
14.2.2	Logic Analyzers	330
14.2.3	PROM Programmers	331
14.2.4	Emulators	331
14.3	Development Systems	332
	Summary	335
	Review Problems	335
	Appendix A: Selected Tables from the Programming Reference Card for the MC6809 Microprocessor	337
	Appendix B: MC6809 Instruction Set	343
	Appendix C: Selected Tables from the Programming Reference Card for the MC68000 Microprocessor	349
	Appendix D: MC68000 Instruction Set	359
	Bibliography	365
	Index	367

CHAPTER 1

INTRODUCTION TO COMPUTERS

Welcome to the exciting and dynamic world of computers! The computing power which in the 1950s filled several entire rooms is now available on an electronic component small enough to fit on the tip of your finger. This tiny device, known as a *microprocessor*, is ushering in a new era of technological progress through its widespread application in every realm of human endeavor. In addition to their uses in computers, microprocessors are now used to control household appliances, automobile engines, laboratory equipment, factory machinery, robots, consumer electronic devices, chemical processing plants, steel mills, and medical diagnostic and treatment equipment; the list goes on and on. They have made it possible to capture a bit of the essence of human intelligence and apply it to the control of inanimate devices. Speaking facetiously, it is now possible to purchase a spark of human genius for \$1.98! In this chapter we will look at some of the broader aspects of computers and microprocessors, including some historical developments, the evolution of the technology, and its division into hardware and software concerns.

1.1 HISTORICAL BACKGROUND

Most historians of technology agree that if any one person could be said to be the inventor or “father” of the computer it would be the English mathematician Charles Babbage. In the early part of the nineteenth century, Babbage first designed and attempted to build what he called an “Analytical Engine”. This machine was intended to automatically compute values for mathematical tables. In those days before the

invention of the calculator, mathematicians and engineers used tables to look up values of such things as logarithms, trigonometric functions, hyperbolic functions, factorials, and the like. The construction of these tables was a laborious and time-consuming task, and many people had attempted to devise machines to carry out the necessary calculations. Babbage's invention, however, was the first to attempt to automate the calculating process completely. It included the concept of a stored program and allowed the user to modify the program so as to alter the calculating behavior of the machine.

Unfortunately, with the limited technology of the nineteenth century, Babbage could not produce the mechanical components with sufficient accuracy to enable the machine to run properly. A machine built to his specifications in the early 1950s did perform as expected. By that time, however, the electronic computer had been developed and Babbage's invention was merely a historical curiosity.

During the early part of this century, much of the groundwork which led to the invention of the modern digital computer was accomplished by various telephone companies around the world, particularly by American Telephone and Telegraph's Bell Telephone Laboratories. Research in the design and development of the telephone switching network led to the invention of various components and design techniques that subsequently were implemented directly in the first successful computer. This computer was developed in the late 1930s to calculate ballistic information for use by the Allies during the Second World War. The computer used electromechanical devices known as *relays* to perform the various functions. Relays are electrically actuated mechanical switches that were used extensively in early telephone switching networks. (They are no longer used to perform logical functions of this type; now they are used simply to manipulate large electrical currents with small controlling currents.)

By the end of the Second World War several versions of the relay computer had been built, and their major shortcomings of speed and reliability were evident. In the early 1950s, however, a new generation of computers was conceived and built, using *vacuum tubes* rather than relays as its primary components. Vacuum tubes had been in use for many years in radio receivers and transmitters as well as in other electronic systems such as radar and sonar. In their new applications in computers they were used as electrically operated switches similar to relays in their behavior. In a relay, however, the switching action is enabled by the actual movement of a mechanical arm. In a vacuum tube switch, the switching action takes place through the movement of a stream of electrons flowing through a vacuum inside the device. As a consequence, the new computers were very much faster than their relay predecessors, performing hundreds of thousands of operations per second (as compared to tens of operations per second). They were also much more reliable because of their lack of mechanical motion. However, they eventually became so large that even with the best components available the mean time between failures was measured in hours!

Coincident with the development of the vacuum tube computer was the invention and subsequent development of the *transistor*. This device, which was destined to replace the vacuum tube in almost all of its applications, became the primary component in the next generation of computers, the transistorized solid-state computer.

When used as an electrically operated switch, the transistor has a significant advantage over the vacuum tube. In the transistor, the switching action takes place in streams of electric current flowing through the solid material of the device. This activity can take place at room temperature. In a vacuum tube, electrons must flow through a space between metal electrodes, and in order for an electrode to emit electrons into the space its temperature must be maintained at about 1000°C.

Because of its lower operating temperature and smaller size the transistor rapidly replaced the vacuum tube in the computers of the 1960s. The transistorized computer consisted of row upon row of printed circuit boards, each one containing as few as one or two or as many as 30 or 40 transistors together with associated components. A modest computer may have contained 5000 or so transistors, while a large one could contain as many as several hundred thousand transistors mounted on 20,000 or more printed circuit boards!

The next generation of computers had its roots in the space program of the 1960s, for which the United States expended enormous amounts of resources in order to put humans on the moon. Among the developments to come from this effort was the *integrated circuit*, which started as an attempt to assemble the equivalent of one printed circuit board's worth of components into a single component. The intent was to interconnect the components using the same process by which they were manufactured, with no manual intervention. This effort succeeded beyond imagination. The early integrated circuits contained the equivalent of one or two of the above-mentioned printed circuit boards, perhaps 10 transistors and about 100 other components. Modern integrated circuits can contain the equivalent of an entire computer, as many as several hundred thousand components.

The integrated circuit computer uses transistors that are similar to those of the earlier generation. The major difference is that the integrated circuit transistors are manufactured in integrated modules and interconnected with other components manufactured at the same time. Thus, a single component in this generation of computers may do the work of tens of thousands of components from the preceding generation. Yet this single integrated circuit component costs about the same and occupies about as much space as a dozen of the earlier components taken together. This is the basis of the computer revolution taking place today. The development cost of a modern computer may be comparable to that of one from an earlier generation, but the size and manufacturing cost are much lower. We are now manufacturing computers with the capabilities of the multi-million-dollar, multi-room-sized behemoths of the 1950s, but which cost only a few thousand dollars and fit on or under a (small) desk!

As a consequence of their reduced size and cost, modern integrated circuit computers are being used to do things that would never have been considered feasible in earlier generations. They are still used in the traditional applications of calculating mathematical results and performing various business support functions. But they are also used in the new area of computerized control, to monitor input signals from a system and to analyze these signals using traditional computation techniques. The results are then used by the computer to generate various signals which, in turn, control the system. The target system may be anything amenable to electrical monitoring and control.