Decimal Computation

HERMANN SCHMID

General Electric Company
Binghamton, New York

A Wiley-Interscience Publication
JOHN WILEY & SONS New York · London · Sydney · Toronto

Library of Congress Cataloging in Publication Data:

Schmid, Hermann, 1925–
Decimal computation.

"A Wiley-Interscience publication."
Includes bibliographical references.
1. Binary-coded decimal system. 2. Calculating-machines—Circuits. I. Title.

QA75.S34 519.4 74-10798
ISBN 0-471-76180-X

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Decimal Computation

Preface

I had intended to write a book on electronic calculators. Instead I ended up with a book on decimal—or, more specifically, BCD (binary-coded-decimal)—computation. Why?

While searching for material describing how calculator circuits perform the various arithmetic operations, I found—to my great dismay—that there was very little information on BCD computation. Some of the classical textbooks on computing techniques outline the simpler algebraic processes, such as addition, subtraction, multiplication, division, and, occasionally, square root. However, in comparison with the available literature on binary computation, decimal-computation literature is extremely scarce.

The situation is even worse in locating software or hardware information on implementing transcendental functions, such as sine, cosine, tangent, arc sine, arc cosine, arc tangent, exponential, and logarithms. Almost no literature exists that describes decimal algorithms or decimal mechanizations of these functions. In addition, I studied the operation of binary computing algorithms and circuits and converted them to BCD format.

Consequently, I collected every bit of available information on BCD computation, organized it according to function, and then described it in more detail. In this process, several alternatives of the "known" techniques became apparent and were also described.

This book is thus a collection, a catalog and a review of BCD computation techniques. The book describes how each of the most common arithmetic and transcendental operations can be implemented in a variety of ways. More specifically, this book covers the following subjects:

- A review of number systems, BCD codes, of early calculating instruments and electronic calculating machines
- A summary of Boolean algebra, ideal logic elements, and MOS logic circuits
- A discussion of the status and trend in large-scale integration (LSI), its effect on calculating circuit size and economy
- An outline of BCD computing circuit applications in the automotive, consumer, education, and entertainment fields, illustrated with some specific examples

- Mathematical developments of the algorithms and descriptions of how they operate (flow diagrams) and how they can be implemented in different ways, for each of these arithmetic operations:

  Add/subtract
  Multiply/divide
  Square root
  Log/exponential
  Trigonometric/hyperbolic functions
  Fixed/floating point operations

- Discussions and comparisons of circuit complexity and performance (accuracy, resolution, and speed of operation) for the different algorithms and implementations of each arithmetic function.

The most outstanding features of this book are as follows:

- The large number of arithmetic and transcendental functions described
- The variety of algorithms and hardware implementations discussed for each function
- The detailed software and hardware descriptions of each approach, starting with a mathematical development of the algorithms and numerical examples; then proceeding with block, logic, and timing diagrams; and concluding with a complexity–performance discussion
- The illustrative and step-by-step descriptions of the operation of the common multiplication and division circuits, with some practical examples
- The exclusive treatment of transcendental functions such as logarithms, exponentials, trigonometric and hyperbolic sine, cosine, tangent functions, and their inverse relationships
- The use of computer programs to simulate the algorithms and circuits used to implement the various transcendental functions
- The unique description and comparison of fixed-decimal-point versus floating-decimal-point techniques and circuits.

As a result calculator designers should find this book useful as a review and a catalog of existing techniques. Designers of industrial and consumer instrumentation, such as computing counters, frequency analyzers, and multiplying scales, will find assistance in this book in selecting the proper technique for a given application. However, the book should be of greatest value to all the numerous designers in the fields of automotive, consumer, educational, and entertainment electronics who are faced with the task of adding a moderate amount of computing capability to an instrument in which the inputs and/or outputs are in BCD format.

This book is intended as a reference for all those involved in the design, manufacture, and testing of BCD computing circuits. Its purpose is to convey the principles involved in BCD computation. Its scope is limited to descriptions of algorithms, of the operation and the performance of BCD computing techniques and BCD computing circuits.

Acknowledgment is due many at General Electric's Avionic Controls and Electric Systems Department who helped in the preparation of this manuscript, especially to Les Schowe for his review. I also would like to express my appreciation to Morris Grossman, editor of *Electronic Design*, who summarized the material in this book into an excellent series of articles and helped enormously to correct both text and diagrams.

HERMANN SCHMID

March, 1974

Decimal Computation

Contents

CHAPTER ONE

Introduction

NUMBER SYSTEMS

It started with pebbles, sticks, and beads. Early man used them as tools to count, account, and compare. Shepherds counted their sheep by transferring pebbles from one pouch to another. If the number of pebbles matched the number of sheep, their account was in order.

Then man drew signs in the sand or mud, such as "|" for one item, "||" for two items, and "|||" for three items. From these signs or groups of signs, the symbols of the various number systems developed. It took ages before symbols were developed that represented a quantity larger than 1. Early Roman numerals, for example, were written I, II, III, IIII, V, with the V replacing the group of five. In this system, however, the position of the digits had no significance. The rules for placing one symbol in front or in back of another symbol to represent the sum or difference of the two symbol values came much later. For example, a Roman IV now represents a 4 (5 − 1) and VI represents 6 (5 + 1). Although the Roman number system was advanced in its time, it is awkward and clumsy by today's standards. Just try to write 1873:

MDCCCLXXIII.

The real breakthrough in number systems came with the Arabic system, of which most people know only the decimal form. In this system there is only a limited number of symbols, the shapes of which are quite unimportant. What is important, however, is the positional significance of each of these symbols in the group. This means that the value of each symbol is determined by its position in the group with respect to the others. We usually refer to the symbol on the left as the most significant digit and to the symbol on the right as the least significant digit. The value of each digit thus increases from right to left by the *power* of the number system. The power associated with each digit is determined by the *base*, or the *radix*, which defines how many symbols are used in the system. In a base 8 system, for example, only the eight symbols 0, 1, 2, 3, 4, 5, 6, 7 are used.

In theory there is no limit as to how high the base can be or how many symbols are used. Practical systems, nevertheless, are limited to bases of less than 16.

The Decimal Number System

The decimal number system is the most popular form of the Arabic number system. Its origin is probably related to the fact that man has 10 fingers. It uses the 10 symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. The base or the radix is 10. A decimal number N can thus be expressed as

$$N = a_n 10^n + a_{n-1} 10^{n-1} + \cdots + a_1 10^1 + a_0 10^0,$$

where a_i is one of the 10 decimal digits 0 to 9, and 10^i is the power of 10 of each digit a_i that defines its positional significance, or simply the power of that digit. Take, for example, the number

$$845 = 8 \times 10^2 + 4 \times 10^1 + 5 \times 10^0.$$

This means that the digit on the far right, is multiplied by 1, the middle digit by 10, and the left digit by 100. More generally, each digit is multiplied by the power 10^i, where i defines the number of digits to the left of the rightmost, or least-significant, digit.

The Binary Number System

In the binary number system the base is 2 and there are only two symbols, 0 and 1. A binary number N_B can thus be expressed as

$$N_B = a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_1 2^1 + a_0 2^0,$$

where a_i is either 0 or 1 and where 2^i represents the power of 2 of the digit a_i. Hence, the digits in the binary system also have positional significance. However, each digit is multiplied by the proper power 2^i, where i again represents the number of digits from the rightmost digit. Take, for example, the number

$$11011 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 27.$$

Note that five digits, or five binary bits, are required to represent the decimal number 27. Thus, the lower the base the more digits are needed to represent a certain value.

HISTORY OF CALCULATING INSTRUMENTS

From the very beginning man had a need for counting. As a result, early man developed techniques for using pebbles, beads, knots on strings, notches on sticks, and signs in mud or sand.

Counting is the process of repetitively adding a fixed increment to the previous sum. Addition and subtraction can thus be regarded as extensions of the counting process. Seven can be added to 9 simply by counting first to 7 and

then continuing the process of incrementing 9 more times to a count of 16. Once the techniques for addition and subtraction were understood, the concepts for multiplication, division, and other arithmetic operations were within easy reach, since all of these operations are based on the repeated use of addition and subtraction.

The development of the various computing techniques was followed by the design of computing devices. The pebbles and sticks gave way to more sophisticated apparatus such as the Chinese abacus, which incorporates already the concepts of positional significance.

The next era in digital computation[1] involved the mechanical or rotary calculating machines, starting with Pascal's number wheels and ratchet (1642), Leibnitz's stepped cylinder (1671), and Babbage's analytical engine (1812), and resulted in the various mechanical and electromechanical adding and calculating machines that are still in use today. All these machines use a combination of gears and cylinders that were driven at first mechanically by a crank or a handle and later with an electrical motor. In 1857, Hill developed the first four-function rotary calculator with keyboard input.

The development of electromechanical computers was also a significant phase in digital computation. The combination of Hollerith's punch cards, the availability of electromechanical relays, and Turing's sequential machine theory led in 1944 to the design of Aiken's Mark I relay calculator.

The major thrust in digital computation came, however, after World War II with the development of the electronic computers. By that time, the operation and fabrication of vacuum tubes and of logic circuits such as Eccles and Jordan's flip-flop was well understood and Boolean algebra was no longer a curiosity. The electronic numerical integrator and automatic calculator (ENIAC) the first electronic digital computer, was born in 1947 under the direction of Mauchly and Eckert. However, in contrast to present machines, ENIAC used decimal arithmetic and was more like a programmable calculator.

A significant improvement in digital computers came with the development of the transistor by Bardeen, Brattain, and Shockley (1948). With it the size and power dissipation of the electronics could be reduced by a factor of 10 and reliability was up an order of magnitude. However, it took almost a decade before the first all-transistor computer appeared on the market.

The latest advance in digital computation is large-scale integration (LSI), which is now used to implement both calculators and computers. With LSI all the electronics required to (1) accept, translate, and store keyboard inputs, (2) execute the various arithmetic operations and store the results, and (3) drive an appropriate display can now be located on a small chip of silicon, approximately one-twenty-fifth the size of a postage stamp.

Such a small chip can contain the equivalent of several thousand vacuum tubes. To comprehend the magnitude of this tremendous reduction, compare

the parameters of today's LSI calculator electronics with those of a hypothetical vacuum-tube version:

	2000 vacuum tubes	*Single LSI chip*
Size	2000 in.^{3}	0.1 in.^{3}
Power dissipation	4000 W	1.0 W
Reliability	2000 F/MH	10 F/MH
Cost (in large quantities)	$5000	$5

From this table, it can be seen that if a calculator is built with tubes, it would probably sell in excess of $5000, have the size of a cabinet 24 × 24 × 72 in., require a small power station to drive it, and fail every 50 h. As is well known, single chip LSI calculators sell for less than $50, come in sizes as small as a cigarette pack, and require as little as 1 W of power; and failures in the LSI chip are extremely rare.

The long road from the Chinese abacus to the LSI computer thus went through several significant development phases, such as the mechanical (rotary) machines, the electromechanical (relay) machines, the vacuum-tube machines, and the transistor machines. In this transition, the principles of implementing the basic arithmetic operations, as used by the various computing devices, have changed relatively little. Multiplication was always X additions of Y; division and square root were always implemented with trial and error subtraction. Therefore, it is of interest to see how some of the older machines operate, what techniques they use, and what algorithm they implement.

The Chinese Abacus

This ingenious device provides efficient computation with an absolute minimum of hardware. It was used centuries before Christ by the Chinese and the Babylonians and is still in use today in many Far Eastern countries.

A typical Chinese abacus, or *souanpan*, is divided into two unequal sections, each being pierced by a number of rods (Figure 1.1). Each rod represents one digit. A nine-rod abacus is thus a nine-digit calculator using the biquinary number system. The mantissa of the number(s) is entered with five beads on each rod in the larger, lower section. The characteristic of the number is entered with two beads on each rod in the smaller, upper section. The two sections combined perform as a single multidigit register. The length (number of digit) of this register is made relatively long. The purpose of this is not only to hold large numbers but to enable one to enter two numbers side by side, as will be shown later in the multiplication example.

The processes of addition and subtraction on the abacus are rather obvious. To add for example, 4321 to 8765, enter the first of the two (Figure 1.1*a*)

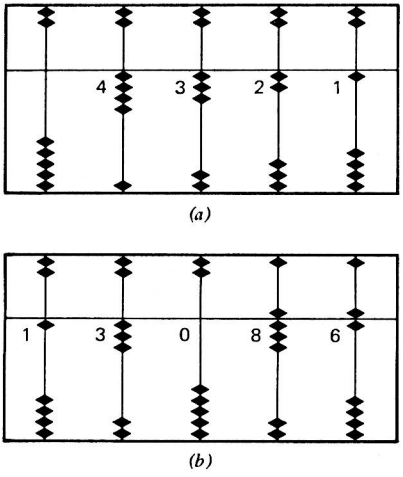

Figure 1.1 The abacus. Addition is carried out by mentally adding each digit of the addend to the augend, which is entered into the abacus first.

numbers, 4321, by placing 4, 3, 2, and 1 beads in the lower section on the four rightmost rods against the dividing bar. Now, mentally add the units digit, 5, of the addend to the units digit on the frame, changing this to 6. To do this place one bead from the lower half of the rod and one bead from the upper half against the dividing bar. The upper bead represents 5, the lower one represents 1. Then proceed as shown, line by line, carrying a 10 where necessary, as in the last two lines. The final state of the abacus is shown in Figure 1.1*b*, which gives the result: 13,086.

Multiplication is done with the help of a table, identical to the digit multiplication table (see Table 3.1). As an illustration, consider the process of multiplying 538 by 2457 (see Table 1.1). The two factors are placed on a 10-digit frame, the multiplicand into the 3 leftmost digits, and the multiplier into the middle 4 digits, leaving to its right as many free columns as there are digits in the multiplicand.

Start by considering the units digit of the multiplicand (i.e., 7), and multiply it in turn by each digit of the multiplier. First enter 8×7 ($= 56$) at the extreme right of the frame (line 2), and then add 3×7 ($= 21$) to the frame one place to the left (line 3). At line 4 we meet a slight difficulty. We have to add 5×7 ($= 35$) to the partial product 266 but find that the units digit, 7, of the multiplicand is already occupying the column we need. But since we have finished with this digit, we ignore it and enter the 3 of the partial product in its place. The 7 is not added to the 3 but replaced by it to form line 4. In the same way lines 5, 6, and 7 show the tens digit, 5, of the multiplicand multiplying

Table 1.1 *Multiplication of 538 by 2457 on a 10-digit abacus.*[a]

OPERATION PERFORMED	NUMBERS ON ABACUS AFTER EACH OPERATION										LINE NUMBER
	MCD.			MLTR.				PRODUCT			
	5	3	8	2	4	5	7	0	0	0	(1)
8 × 7 = 56									(5	6)	
	5	3	8	2	4	5	7	0	5	6	(2)
3 × 7 = 21								(2	1)		
	5	3	8	2	4	5	[7]	2	6	6	(3)
5 × 7 = 35							(3	5)			
	5	3	8	2	4	5	3	7	6	6	(4)
8 × 5 = 40								(4	0)		
	5	3	8	2	4	5	4	1	6	6	(5)
3 × 5 = 15							(1	5)			
	5	3	8	2	4	[5]	5	6	6	6	(6)
5 × 5 = 25						(2	5)				
	5	3	8	2	4	3	0	6	6	6	(7)
8 × 4 = 32							(3	2)			
	5	3	8	2	4	3	3	8	6	6	(8)
3 × 4 = 12						(1	2)				
	5	3	8	2	[4]	4	5	8	6	6	(9)
5 × 4 = 20					(2	0)					
	5	3	8	2	2	4	5	8	6	6	(10)
8 × 2 = 16						(1	6)				
	5	3	8	2	2	6	1	8	6	6	(11)
3 × 2 = 6						(6)					
	5	3	8	[2]	3	2	1	8	6	6	(12)
5 × 2 = 10				(1	0)						
	5	3	8	1	3	2	1	8	6	6	Result (13)

[a] (XX) identifies the partial product to be added; [X] signifies that this digit is dropped and not added to the partial product.