Preface

Socrates: *Arithmetic is a kind of knowledge in which the best natures should be trained, and which must not be given up.*
Glaucon: *I agree.*

Plato, The Republic

THE papers on computer arithmetic in this special issue are in most cases formalizations of presentations made at a workshop sponsored by the Logic Design Subcommittee of the Systems Technical Committee (IEEE Computer Group) which was held in June 1969.

Many recent and continuing advancements are expanding the scope of computer arithmetic. Microelectronics allows new dimensions in complexity because of the number of gates which can be placed on a single chip, with improved reliability and at reduced cost, and because an interesting cost tradeoff exists between the number of chips used versus the number of chip types. Solid-state embodiment of large random-access memories has made it possible to move the paging concept one level higher in the memory hierarchy to reduce the fundamental limitation to effective utilization of high-speed arithmetic units previously imposed by memory. Microprogram implementation of control has also contributed to arithmetic-unit effectiveness by allowing frequently used operations, otherwise executed as subroutines, to be included in the machine instruction set. During execution of these instructions, memory is used entirely for operand references and, in many cases, the number of references for intermediate results is also substantially reduced by effective use of internal registers.

Improved insight into the properties of number systems, new arithmetic-unit organizations, and an expanded repertoire of algorithmic procedures are an important complement to advances in devices. Matula's paper, "A Formalization of Floating-Point Numeric Base Conversion," is timely for both the user and designer. First, the mixed-mode arithmetic now allowed by higher level languages, notably PL/I, can lead to several base conversions, even in the process of executing a single assignment statement. Both the computer user and designer should therefore be aware of the round-off error which may result. The subject is of further interest to the designer because, when a fast binary arithmetic capability and an efficient base-conversion procedure are included in a design, the economy of implementing decimal arithmetic indirectly with these already extant tools is attractive.

In his paper "The Correspondence between Methods of Digital Division and Multiplier Recoding," Robertson ties together various approaches used to minimize the number of steps in multiplication and division, and relates these algorithms to the more general problem of recoding binary numbers into a base 2 representation where a third digit value, -1, is allowed in exchange for maximizing the probability of 0.

Combinational-logic multiplication has been found cost-effective in large computer systems. Flynn's "On Division by Functional Iteration" explores various analytic approaches (i.e., not digit-by-digit) to the execution of division which make use of the fast multiplication capability. Ling describes a novel organization of how such a multiplier might be organized in "High-Speed Computer Multiplication Using a Multiple-Bit Decoding Algorithm."

"The LX-1 Microprocessor and Its Application to Real-Time Signal Processing" by Hornbuckle and Ancona illustrates the enhancement of general-purpose arithmetic unit effectiveness possible with microprogramming. The repetitive procedures in signal processing are ideal candidates for microprogram memory, as illustrated by the example of real-time spectral analysis included in this paper.

"Design of the Arithmetic Units of ILLIAC III: Use of Redundancy and Higher Radix Methods" by Atkins describes the arithmetic procedures incorporated in ILLIAC III. The signed-binary-digit representation defined differs from the signed-digit system described elsewhere by Avižienis and Tung, since the latter requires radix ≥ 3.

The papers by Avižienis and Tung, Tung, Svoboda, and Ling are all indicative of the larger building blocks being proposed for arithmetic unit design. "A Universal Arithmetic Building Element, and Design Methods for Arithmetic Processors," by Avižienis and Tung, and "Signed-Digit Division Using Combinational Arithmetic Nets," by Tung are companion papers which describe the use of one

such element. Direct implementation of operations such as polynomial evaluation are considered in addition to the basic arithmetic processes. Tung applies a modified form of Svoboda's division algorithm, presented in an earlier paper, to realize division.

Svoboda's "An Adder with Distributed Control" describes an interesting method for summing several numbers in parallel.

"On the Addition of Binary Numbers" by Brent is a note on a paper in which Winograd determined the ultimate limit on the time required to perform addition as a function of circuit fan-in and circuit delay. (Redundant representations are excluded, of course.) An interesting byproduct of Winograd's result is that no nonredundant representation is more efficient for addition than residue representation. Rao and Trehan describe a residue-arithmetic representation which combines the use of moduli of the form 2^k and 2^k-1 together with the use of a magnitude index in "Binary Logic for Residue Arithmetic Using Magnitude Index." The objective is to reduce two of the more severe disadvantages of residue arithmetic, i.e., conversion and range determination.

RICHARD R. SHIVELY
Guest Editor

Richard R. Shively (S'55–M'58) was born in La Grange, Ill., on July 1, 1934. He received the B.S., M.S., and Ph.D. degrees from the University of Illinois, Urbana, in 1956, 1957, and 1963, respectively.

From 1957 to 1959 he worked with IBM on the design of large-scale computing systems. As a member of the Digital Computer Laboratory staff at the University of Illinois from 1959 to 1963, he participated in the design of ILLIAC II. Since joining Bell Telephone Laboratories, Whippany, N. J., in 1963, he has done research on the organization of several new computer systems, including planning and design of a fast Fourier transform processor. Most recently he has been concerned with real-time operating systems. He has taught a course in computer arithmetic in the Bell Telephone Laboratories after-hours program.

Dr. Shively is a member of Tau Beta Pi, Eta Kappa Nu, Pi Mu Epsilon, and Sigma Xi. He is the Chairman of the IEEE Computer Group Systems Subcommittee on Logic Design and is also Chairman of the Northern New Jersey Computer Group. He organized the 1969 IEEE Workshop on the Theory of Computer Arithmetic.