first part covers those complexity topics that one expects to see in any computational complexity textbook. Parts II and III are reasonably independent from one another, as seems only reasonable given the scope of the two parts. The problems at the end of each chapter are interesting, and serve to extend the material.

I enjoyed reading the book; each section flowed into the next, and there was an overarching organization that made it easier to follow the material. There were a few minor typos that I detected, but those were obvious from context. Program-size, or Kolmogorov, complexity is not discussed in the book, which is a pity since Kolmogorov complexity would have produced an interesting unified view of nonunifom models, but is understandable considering the scope of the topic.

Computer Arithmetic Algorithms
by Israel Koren
Published by A.K. Peters
296 pages, $49.00

Reviewer: George A. Constantinides (george.constantinides@ieee.org) [4]

1 Introduction

This book covers the broad topic of implementing arithmetic in digital logic and/or in software.

With the increasing acceptance of custom hardware designs for arithmetic processors, due in no small part to the commercial success of the Field-Programmable Gate Array, a sound knowledge of computer arithmetic is becoming important for an ever-increasing circle of designers.

This review is probably somewhat unusual for SIGACT members as the theoretical aspects of arithmetic design, such as fundamental limits on the speed of computation, are covered only fleetingly by this book. This in no way detracts from the value of the book as an introduction to the field.

The review is of a second edition (2002) of a book originally published in 1993. Apart from general corrections and minor additions, new sections have been introduced on: floating-point adders, floating-point exceptions, general carry-look-ahead adders, prefix adders, Ling adders, and fused multiply-add units.

2 Book Contents

I have listed below those elements of each chapter which are particularly noteworthy.

2.1 Conventional Number Systems

Basic number systems and radix conversions

2.2 Unconventional Fixed-Radix Number Systems

Negative-Radix fixed number systems; Signed-digit number systems; Binary signed-digit numbers

2.3 Sequential Algorithms for Multiplication and Division

Sequential multiplication; Sequential division; Nonrestoring division; Square root extraction

2.4 Binary Floating-Point Numbers

Choice of floating-point representations; The IEEE floating-point standard; Roundoff schemes; Guard digits; Floating-point adders; Exceptions; Round-off errors and their accumulation

2.5 Fast Addition

Ripple-carry adders; Carry-look-ahead adders; Conditional sum adders; Optimality of algorithms and their implementations; Carry-look-ahead addition revisited; Prefix adders; Ling adders; Carry-select adders; Carry-skip adders; Hybrid adders; Carry-save adders; Pipelining of arithmetic operations

2.6 High-Speed Multiplication

Reducing the number of partial products; Implementing large multipliers using smaller ones; Accumulating partial products; Alternative techniques for partial product accumulation; Fused multiply-add unit; Array multipliers; Optimality of multiplier implementations

2.7 Fast Division

SRT division; High-radix division; Array dividers; Fast square root extraction

2.8 Division through Multiplication

Division by convergence; Division by reciprocation

2.9 Evaluation of Elementary Functions

The exponential function; The logarithm function; The trigonometric functions; The inverse trigonometric functions; The hyperbolic functions; Bounds on the approximation error; Speed-up techniques

2.10 Logarithmic Number Systems

Sign-logarithm number systems; Arithmetic operations; Comparison to binary floating-point numbers; Conversions to/from conventional representations

2.11 The Residue Number System

Arithmetic operations; The associated mixed-radix system; Conversion of numbers from/to the residue system; Selecting the moduli; Error detection and correction

3 Opinion

The target audience of this book should be advanced undergraduate or graduate students, for whom it would serve as an excellent introduction. Although introductory in nature, each chapter provides an extensive list of primary reference material, which should provide any graduate student with a head-start in the subject.

Highlights of the book include a very clear description of non-restoring division and its importance, and the treatment of floating-point representations. Floating-point is described with an emphasis on the design trade-offs involved in the selection of a standard such as the IEEE floating-point standard, rather than just presenting IEEE as a *fait-accompli.* Division is given the detailed treatment it deserves, occupying two out of the eleven chapters of the book. The inclusion of logarithmic number systems is also of particular interest.

What is really unusual in a book on computer arithmetic, is to include a treatment of how to approximate the elementary functions. This chapter is only partially developed, providing one detailed approach for each function considered, and would benefit from a detailed discussion of CORDIC, modern table-based approaches and hybrids. Having said this, any treatment of elementary functions is appreciated, and references are given to some of the standard text-books on the subject.

Overall, I would definitely recommend this book to anyone with a background encompassing basic digital design.

4 Additional Information

Curious readers may wish to look at the book web-site,
`http://www.ecs.umass.edu/ece/koren/arith`,
and a nice Java-based simulator covering many of the topics in the book can be found at
`http://www.ece.umass.edu/ece/koren/arith/simulator`.