\[ \sqrt{x} \] comparison: new results discovered

Editor:

I believe that there is an error in the article by Michael Andrews, "Mathematical Microprocessor Software: A \( \sqrt{x} \) Comparison" (May 1982). In one section of the article, Newton's square-root approximation is discussed. Dr. Andrews determines the constants of the equation

\[ x_0 = c_1 + c_2x \]
\[ x_1 = (x_0 + x/x_0)/2 \]

by searching for the constants that give the least maximum error over a given range. In my analysis, I arrived at different constants for the intervals (1/8, 1/2] and (1/4, 1] in the eight-bit case.

For the interval (1/8, 1/2], my analysis showed that the best choices for \( c_1 \) and \( c_2 \) are

\[ c_1 = 0.25 = 0.01002 \]
\[ c_2 = 0.9375 = 0.11112 \]

and not \( c_i = 0.1875 \) and \( c_i = 1.125 \), as stated in the article. Similarly, for the interval (1/4, 1], the best choices are

\[ c_1 = 0.3750 = 0.01102 \]
\[ c_2 = 0.6250 = 0.10102 \]

rather than \( c_1 = 0.8750, c_2 = 0.5 \).

R. D. Teachey
Grumman Aerospace Corporation

Author's response:

Your results are probably accurate and should be published upon verification by other independent sources. As resources permit, we will attempt to verify your best \( c_1 \) and \( c_2 \) for the intervals stated. In the meantime, let me advise you that when we search the coefficient space, we can expect more than one optimal choice because the objective function is nonlinear and the surface response is multimodal. Our research group is delighted to see that other coefficients are possible.

We repeat, however, that coefficient choice depends to a great deal on the cost of implementation. If in an eight-bit machine multiplications are expensive, while

---

IEEE Micro

Editor-in-Chief: Richard C. Jaeger*

Associate Editor-in-Chief: Peter R. Rony

Editorial Board:

Tuvia Apelievicz, City College of New York
James H. Aylor, University of Virginia
Walter R. Beam, US Air Force
Patrick Buffet, Digital Equipment Corporation
John Burkin, University of New South Wales
J. Thomas Cain, University of Pittsburgh
Patrick P. Fasang, Siemens Corporation
John L. Hennessey, Stanford University
Kenji Kani, Nippon Electric Company
Fred Ligouri, US Navy
Victor P. Nelson, Auburn University
Jean-Daniel Nicoud, Swiss Federal Institute of Technology
Denee Ogden, Texas Instruments
Robert G. Stewart, Stewart Research Enterprises

Editor and Publisher: True Seaborn
Associate Publisher: Ginger Conrad

Managing Editor: Joe Schallan
Calendar Editor: Christina Taylor
Contributing Editor: Ware Myers
Art Director: Jay Simpson
Production: Larry Bauer, Donna Calcagno, Frank Dalton, Kas Winters
Advertising Director: Dawn Pack
Advertising Production: Barbara Cunningham

*Submit six copies of all articles and special issue proposals to Dr. Richard C. Jaeger, Dept. of Electrical Engineering, 207 Dunstan Hall, Auburn University, AL 36849, (205) 826-4330.

IEEE COMPUTER SOCIETY PUBLICATIONS BOARD

Chairman: K. S. Fu
Vice-Chairman: J. N. Snyder
Secretary: D. P. Agrawal

CS Technical Activities Board Rep.: Paul Losleben
CS Publications Rep. to CS TAB: Y. T. Chien
IEEE Publications Board Rep.: Merlin G. Smith


Computer Advisory Committee (CAC): Paul L. Hazan (chairman), S. S. Yau (ed.-in-chief, Computer), M. C. Mulder, D. B. Simmons, Stanley Winkler, T. Seaborn (editor & publisher)


New Publications Proposal Committee: C. V. Ramamoorthy (chairman), J. N. Snyder, R. G. Stewart, Susan Rosenbaum


CIRCULATION: IEEE Micro (ISSN 0272-1732) is published in February, May, August, and November by the IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380. Annual subscription: $8.00 in addition to IEEE Computer Society or any other IEEE society member dues; nonmembers $36.00. Single-copy prices: members $6.00; nonmembers $12.00. This journal is also available in microfiche form.

UNDELIVERED COPIES: Address all correspondence to IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854.

POSTMASTER: Send address changes to IEEE Micro, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Second class postage is paid at New York, NY, and at additional mailing offices.

COPYRIGHT AND REPRINT PERMISSIONS: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US Copyright law for private use of patrons; those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 21 Congress St., Salem, MA 01970. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint, or republication permission, write to Editor, IEEE Micro, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720. All rights reserved. Copyright © 1982 by The Institute of Electrical and Electronics Engineers, Inc., 345 E. 47th St., New York, NY 10017.

EDITORIAL: Unless otherwise stated, bylined articles, as well as products and services offered in New Products, the Product Summary, MicroNews, MicroLaw, Notes on Application and Design, MicroStandards, and Access, reflect the author’s or firm’s opinion; inclusion in the publication does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.
shifts and adds are not, coefficients which can be implemented by mere shifts and adds may be more attractive. I thank you for your interest in this matter and hope that I am able to substantiate your findings soon.

Michael Andrews
Colorado State University

More on instruction sets

Editor:

Let me see if I've got this straight: Dennis Fairclough, in his article on microprocessor instruction sets (May 1982) and in his subsequent reply to Thad Smith III, would replace the 6800 instruction RTI with "MOVE source 1, destination 1, source 2, destination 2, condition code." And this is supposed to be a simplification?

When I first saw Fairclough's article, I felt he was definitely on the right track ... perhaps because his data confirms a long-held suspicion of my own, that computer manufacturers tend to give us too cotton-pickin' many instructions. The familiar "embarrassment of riches" syndrome. The efforts to instrument actual code and use the results to synthesize a simpler instruction set are to be greatly applauded. Where Fairclough and I part company is where he begins to assign mnemonics to the set.

Perhaps we should ask ourselves what the goal is we seek. Is it to make the instruction set of a real microprocessor simpler, easier to use, more regular, and more efficient to mechanize? (All worthy goals, and definitely in the spirit of the May article.) Or is it simply to try to express an instruction set with a minimum number of mnemonics? I submit that the latter goal is a mere parlor game and has nothing to do with Fairclough's original direction. If that is the goal, though, we can probably think of many even smaller sets. I'm rather partial to the universal "TEST bit 1, set bit 2, clear bit 3, branch on condition," although traditionalists may prefer a Turing machine.

When Zilog introduced the Z-80 as an extension of the 8080, they also changed the instruction mnemonics. Their set was applauded by many because of its regularity and its rational extension to additional addressing modes. In using it, however, I and many others have found it not to be so superior. The reason is simple: Of the 158 Z-80 mnemonics, 39 of them begin with "I.D." The information content of these mnemonics is practically nil. In practice, one has to look at the form of the arguments to determine what instruction is really being called for. I have no doubt that Dennis Fairclough's MOVE instruction suffers from the same malady.

So please, Dennis, continue your work to define a simple and regular instruction set. Let the op codes for MOVE, CALL, RETURN, PUSH, etc., be similar if you like. But don't try to extend this regularity to the point of hiding the true function of the instruction from the programmer. There is no place in the real world for an instruction set that doesn't include CALL, RET, PUSH, and POP.

While we're on the subject, I'd like to suggest mnemonics that are even more meaningful to the programmer, even at risk of having a larger set of mnemonics. These include GOTO for an unconditional branch, IF for a conditional branch, LOAD for a memory-to-register move, STORE for register-to-memory, and MOVE (or COPY) for a register-to-register move. Form follows function. I'll bet I can map them onto the Fairclough set of operations.

Jack W. Crenshaw
Tampa, Florida

Errors detected in CAPS programmer's card

Editor:

The ASCII code table in the lower right corner of page 23 of the August 1982 issue contains several errors which should not have been passed on, even if the originator of the table had them wrong:

5E is a circumflex, not an up-arrow;
5F is an underline, not a left-arrow;
60 is a grave accent (reverse quote);
2E is a period, not a bullet; and
1F is US (unit separator) not VS (the legend, however, is correct).

It would be nice if those quoting ANSI standards would do so correctly.

Michael D. Shapiro
San Diego, California

The material in question was reprinted from a copyrighted programmer's reference card. IEEE Micro cannot alter such previously published material. However, we thank the writer for bringing this to our attention.

—Ed.