STANDARDS FOR COMPUTER ARITHMETIC

by

Alan B. Marcovitz*

Abstract

A set of standards for the design of the arithmetic unit of all general purpose digital computers has been proposed. This paper discusses that proposal and suggests that such standards are not now in the best interest of either the computer industry or the computer user.

* Dr. Marcovitz is Professor and Chairman of Electrical Engineering at Florida Atlantic University, Boca Raton, Florida, and is standards representative for the Computer Architecture Committees of IEEE and ACM.
STANDARDS FOR COMPUTER ARITHMETIC

Alan B. Marcovitz

The basic purpose of standards is to improve communications. It is useful for you to be able to understand exactly what I mean. Thus, there are standard definitions of computer terms.\(^1\) It is important that two computers be able to communicate. This has been the impetus for ASCII code and also for standardization efforts on tape formats. A third communication area is that of man-machine. It would be nice to be able to write the same program, no matter what machine it was to run on. Consequently, we have standard FORTRAN and COBOL. In each of these examples, the standards affect the externals, the way the computer appears, rather than the internals, the way the computer is built.

Along with these advantages, one must guard against the major disadvantage of standardization, namely the tendency to inhibit creativity. If everyone has agreed upon a standard format, then there is a negative force against searching for a better one. This is not too serious in the language area, in that one can always construct a "superset" of standard FORTRAN (i.e., a version that will accept all legal statements of the standard plus others as well). Eventually, efforts like this

\(^1\)We are now working on an update of the IEEE standards.
make the standard outdated since each manufacturer supplies
his own super-FORTRAN, none of which are compatible (at least
if you use any of the extra features). The other approach open
if you don't like standard FORTRAN is to design an entirely
different language. There is nothing in the standard specifi-
cation to inhibit this kind of development. Thus, languages
such as BASIC, PL/I and APL still flourish.

Before turning my attention to computer arithmetic stan-
dards, let me comment on enforcement. Clearly, the only place
standards are enforced is in the market. If enough customers
demand that their computers satisfy some standard, then even
the largest manufacturers are forced to meet the standard.
("Enough", in this context, may just mean the federal govern-
ment.) Consequently, if standards are to be at all meaningful,
we must convince the user that they are in his best interests.

Now, let us look at the specific case of computer arithme-
tic. In order to get an idea of the scope of the proposed
standard\(^2\), I summarize its most important points.

1. All word lengths must be a multiple of 8 in the
   range of 24 through 64.

2. All internal numeric representations must be in
   sign, magnitude.

3. All floating point numbers must have the follow-
ing format: one bit for the sign of the number,

\(^2\)Proposed standard submitted to American National Standards
Institute by Mr. Millard H. Peskein and Mr. Anthony Gargaro
the remainder of the first quarter of the word for the biased exponent\(^3\), and the last 3/4 of the word for the significand\(^3\).

4. All internal character representatives must be ASCII.

5. Negative zero must never result from arithmetic operations.

6. All machines must have jumps on at least >0, ≥0,

=0, ≠0, <0, ≤0.

Two questions must be answered about these? First, is there general agreement that these are technically sound? Second, does the consumer care enough about them to insist on them when he buys? I think that the answer is "no" to both questions in most of the areas. I will concentrate on the first two standards.

Let us look first at word length. Certainly, there is no general agreement in the industry on size. Major manufacturers use word sizes of 32, 36, 48 (or 52) and 60 bits for large computers and word sizes of (at least) 8, 12, 16, 18 and 24 bits for minicomputers. The only common factor is 2; all computer words contain an even number of bits. It certainly would be nice if all words were of the same size. On the other hand, I see little chance of getting people to agree on what that size should be. Furthermore, the technology is changing so rapidly that the "correct" choice for today may well be quite out of

---

\(^3\)I suspect that many of you (like me) have been using the "inappropriate" names, exponent for ex-rad and mantissa for significand.
date by the time it is accepted in the market place. (See, for example, the predictions by Caxton Foster in "The Next Three Generations," Computer, Vol. 5, No. 2, pp. 39-42, IEEE Computer Society, March-April, 1972). Finally, making all words a multiple of 8 in length does very little; a 32-bit word is as incompatible with a 40-bit word as it is with a 36-bit word. (It may be even worse if the 36-bit machine can effectively handle the same 4 characters as the 32-bit machine, whereas the 40-bit machine stores 5 characters.)

The second specification is that all internal numeric representations must be in sign, magnitude. Once again, this is not an industry standard. In fact, almost all machines now use either one's complement or two's complement, since these simplify the arithmetic element. Furthermore, if you ask the typical user, he doesn't even know what the internal representation is. Consequently, it seems rather unlikely that there would be enough user pressure to enforce a standard in this area. Even if the cost of computer circuits becomes so low as to be insignificant, little is to be gained by the standardization.

In summary, it is the belief of the author that standards for the internal structure of arithmetic elements provide very little in the way of advantages. Instead, they tie down the design at a time when we are not ready to have it set. Hardware is changing so rapidly that any specification we establish now is bound to become obsolete before it can be published.