Errors in digital filtering programs and algorithm

Editor:

There appear to be a number of programming errors in the articles on digital filtering by V. P. Nelson and H. T. Nagle, Jr., in IEEE Micro, February 1981, pp. 23-41. These errors are listed at the end of this letter.

There is, however, an algorithmic error that would prevent any of the programs listed from working in any case: All coefficients are scaled by 2^16 and then stored as 16-bit two's complement integers. That is, as given on page 31,

\[ \text{VALUE STORED} = \left(\lfloor S0 \times 2^{16} \rfloor + 0.5 \right) \]

If the 16 most significant bits of product are considered the "result," we have,

\[ R = \frac{\left(\lfloor S0 \times 2^{16} \rfloor + 0.5 \right) \times \text{SRC}}{2^{16}} = S0 \times \text{SRC} \times 2^{-2}. \]

Thus, \( R \) must be multiplied by 4 (a left shift by two bits) to approximate \( S0 \times \text{SRC} \), the desired product. The authors have erroneously left-shifted by only one bit in each case. For example,\[ S0 = 0.4383164 \text{ (p. 32)} \]
then \[ S0 = \lfloor S0 \times 2^{16} + 0.5 \rfloor = 7181. \]
Say \[ X = 1000 \],
then \[ S0 \times X = 7181000, \]
and \[ \lfloor S0 \times X/2^{16} \rfloor = 109. \]
Finally \[ 109 \times 4 = 436, \]
while \[ \lfloor S0 \times 1000 \rfloor = 438. \]

In the case of the 8086, the Z8000, and the TMS9900, the 32-bit result does appear in the two-bit registers.

For the MC68000, the 32-bit result resides in a single 32-bit register so that the required operations (following the multiply) are as follows: left-shift the register two bits (ASL #2,R), then swap upper and lower 16-bit words (SWAP), and consider the lower word the result:

\[ \begin{align*} & \left(\lfloor S0 \times X^2 \rfloor/2^{16}\right) \text{ \text{ (p. 32)} } \quad \text{so that } \frac{7181 \times 1000^2}{2^{16}} = 438 \\ & = \lfloor S0 \times 1000 \rfloor, \end{align*} \]

an exact result.

Further, there are some algorithmic improvements that will enhance the speed of program execution by elimination of extraneous instructions. For example, \( X, Y, T1, \text{ and } T2 \) can be left in their "divided by 4" format with only the final \( Y \) being left-shifted before output. This reduces the number of "left-shift by two" operations from 13 (1 in

\begin{verbatim}
;MC68000 4TH ORDER DIGITAL FILTER PROGRAM
MOVEQ #2.06, D0

FILTER: JSR INIT.W
FLOOP: JSR INPUT.W
MULS S0.W,D4
LEA A0.W,A0
LEA M0.W,A1
MOVEQ #4.07, D4

FLOOP1: MOVE W (A1)+, D0
MOVE W D0.D0, D4
MOVE W D0.D0, D5
MOVE W (A1)+, D0
MULS (A0)+, D0
MULS (A0)+, D2
ADD L D2.D0
ADD L D3.D1
ADD L D1.D4
ASL L D6.D4

SWAP D4
MOVE W D4,PORT2.W

MOVE W D5,(A1)+
MOVE W D4, D4-(A1)
MULS (A0)+, D4
ADD L D0.D4
DBRA D7,FLOOP1

ASL L D6.D4
SWAP D4
MOVE W D4,PORT2.W

BRA FLOOP

INIT.W: SAME AS NELSON AND NAGLE
INPUT W: SAME AS NELSON AND NAGLE
A0: A01,A02,B01,B02,A00
A11,A12,B11,B12,A10
A21,A22,B21,B22,A20
A31,A32,B31,B32,A30
M0: 0,0,0,0
.END

Figure 1. An improved version of the MC68000 digital filtering program.
\end{verbatim}
FILTER, 1 × 4 in OUTP._1D, and 2 × 4 in PRE._1D) to five. Secondly, if the criterion of minimization of time lag from input to output is discarded, T1 and T2 need never be stored and the operations needed to store and manipulate the M's are reduced.

Figure 1 lists an MC68000 program incorporating the improvements suggested; “on-paper” execution time is 256 μsec, versus 327 μsec for Nelson and Nagle's version. In Figure 2, I show an LSI-11/23 version incorporating all of these improvements. The actual execution time is 660 μsec. An earlier version paralleling the programs of Nelson and Nagle requires 900 μsec for execution. The bottleneck is, of course, the relatively slow 26-μsec multiply.

With reference to the Senior Editors' Introduction in the same issue of IEEE Micro, I do believe that the premise of Nagle and Nelson's articles—to show real implementations of important algorithms on current micros—is often of use. However, listings must be phototypesetted of actual, high-quality computer printouts, not typeset versions of supplied listings. Both Byte and IEEE Transactions on Acoustics, Speech, and Signal Processing follow this policy. Finally, when benchmarks are given, it should be indicated whether they are “on-paper” timings or actual runs. Although (as seen here) small errors often will not significantly impact “on-paper” timings, it is only by actual runs that the program can at least be seen to work.

The following is a list of the errors that occur in the assembly language listings on pages 35-40 of the Nelson and Nagle article:

**List 1.** (1) The second line of the program should be labeled “FLOOP”.
(2) Subroutine DELAY is not called.

**List 2.** (1) In subroutine OUTP._1D, DBRA D7, LOOP should be DBRA D7, OLP._1D. (2) LEA T1.W,A3 should be followed by LEA T2.W,A4 and MOV DO,A3@ + should be MOVE DO,A4@ + .

**List 4.** (1) MPY on the TMS 9900 is unsigned. Thus the multiplicand and multiplier must be tested, negated if necessary, and then the unsigned result negated if necessary; again the left shift must be by two bits, not one. The new TMS 9995 has a signed multiply.
(2) JNE PRLP._1D, not JNE @PRLP._1D in PRE._1D.

**List 5.** My local Fairchild agent tells me that the Microflame II chip has been out of production for 1.5 years. Why include it?

L. Robert Morris
Carleton University, Ottawa

Figure 2. An LSI-11/23 digital filtering program.