Digital Filtering Using the HPC

National Semiconductor
Application Note 485
Ashok Krishnamurthy
April 1987

INTRODUCTION

This report discusses the implementation of Infinite Impulse Response (IIR) digital filters using the National Semiconductor HPC microcontroller. A general program, that can be used to implement cascaded second order sections, up to a maximum of 8 sections, is also included. The program may have to be modified for specific A/D and D/A interfaces.

This report is not intended to be a tutorial on Digital Filter Design methods or their implementation details. Such information can be found in references 1 and 2 below. The general discussion included here closely follows that in reference 3.

DIGITAL FILTERING

The general IIR filter with input x (n) and output y (n) can be described by a transfer function of the form

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a(0) + a(1)\,z^{-1} + \ldots + a(m)\,z^{-m}}{1 + b(1)\,z^{-1} + \ldots + b(p)\,z^{-p}}$$

To minimize the effects of coefficient truncation, high order filters are usually implemented as a cascade of second order sections. (Another possible choice is parallel realization—see references below).

In cascade realizations, the numerator and denominator polynomials in the above are factored into second order terms, and the filter is realized as a cascade of such second order sections. This is shown in *Figure 1*. A typical second order section has a transfer function of the form

$$H(z) = \frac{A0 + A1 \times z^{-1} + A2 \times z^{-2}}{1 + B1 \times z^{-1} + B2 \times z^{-2}}$$

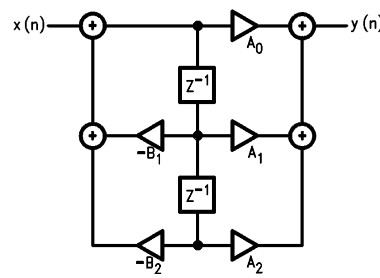
A second order section such as the above can be realized in a number of ways; the one of concern here is the so-called 1-D form (see Reference 3). The second order 1-D form is shown in *Figure 2*. Based on this figure, we can obtain the following equations:

$$m(k) = x(k) - B1 \times m(k-1) - B2 \times m(k-2)$$

$$y(k) = A0 \times m(k) + A1 \times m(k-1) + A2 \times m(k-2)$$

Define $T1 = -B1 \times m(k-1) - B2 \times m(k-2)$,

$$T2 = A1 \times m(k-1) + A2 \times m(k-2)$$



TL/DD/9247–2

FIGURE 2. One Second Order Section

Since T1 and T2 depend on signal values at time k − 1 and k − 2, we can precompute and store these quantities in the time interval from k − 1 to k. Then, when x(k) becomes available at time k, y(k) and m(k) can be quickly computed using

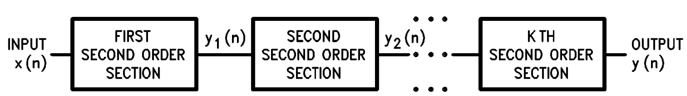
$$m(k) = x(k) + T1,$$

$$y(k) = A0 \times m(k) + T2$$

If there are a number of stages, then these computations should be repeated for each stage. Based on these discussions, the operation of a digital filter can be described using the flowchart in *Figure 3*.

USING THE FILTER PROGRAM

Appendix A contains the listing of the program FILTER that can be used to implement cascaded IIR filters as described above. The program as shown uses a codec interfaced to the HPC using MICROWIRE/PLUS™ to do the A/D and D/A conversion. The program can be used with other A/D and D/A converters by suitably modifying the following subroutines: INPUT, OUTPUT and INIT. Only the portions of INIT that deal with the codec interface need to be modified.



TL/DD/9247–1

FIGURE 1. Cascade Realization of a Digital Filter

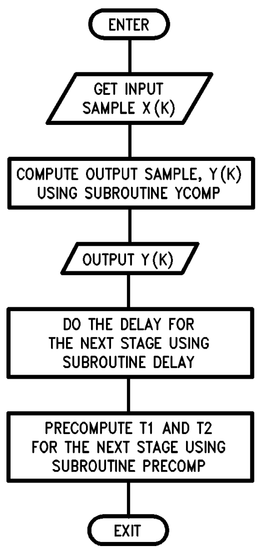
MICROWIRE/PLUS™ is a trademark of National Semiconductor Corporation.

RRD-B30M105/Printed in U. S. A.

The filter coefficients and the number of cascaded stages need to be supplied to the program. This is done as follows:

1. *Specification of filter order.* Define a word address called ROMNST and store the number of cascaded stages in that word. The program is presently set up for 4 cascaded stages.
2. *Specification of filter coefficients.* Each second order stage needs the specification of 5 coefficients, A0, A1, A2, B1 and B2. If the number of stages is m, let the coefficients be

   A0−1, A1−1, A2−1, B1−1, B2−1 for stage 1,
   A0−2, A1−2, A2−2, B1−2, B2−2 for stage 2,
   .
   .
   .
   A0−m, A1−m, A2−m, B1−m, B2−m for stage m.



TL/DD/9247−3

FIGURE 3. Flowchart for the Computations in a Second Order Module (Based on Reference 3)

Define 5 word addresses called ROMA0, ROMA1, ROMA2, ROMB1, ROMB2 and store these coefficients at these addresses as follows:

ROMA0: .WORD A0−1, A0−2, A0−3, … A0−m
ROMA1: .WORD A1−1, A1−2, A1−3, … A1−m
ROMA2: .WORD A2−1, A2−2, A2−3, … A2−m
ROMB1: .WORD B1−1, B1−2, B1−3, … B1−m
ROMB2: .WORD B2−1, B2−2, B2−3, … B2−m.

Note that the coefficients are signed and need to be in 2's complement representation. Also, the stored coefficients need to be half their actual value. This is because of the way that the program does 2's complement multiplication using the subroutine SMULT.

The FILTER program copies all the coefficients to on-chip RAM for faster execution. Also temporary storage for m (k), m (k − 1), m (k − 2), T1 and T2 is obtained from on-chip RAM. This, along with the storage of various addresses used by the program consumes the entire 192 bytes of user base page RAM.

Note that the filter program does not check for overflow during the various additions. This is because the HPC does not have a signed addition/subtraction overflow flag, and it was felt that the simulation of this feature in software would add excessive overhead. It is therefore the user's responsibility to ensure that the filter coefficients are properly scaled so that the overflow will not occur.

16 x 16 2's COMPLEMENT MULTIPLICATION

One of the basic operations in digital filtering is that of signed multiplication. Since the HPC supports unsigned multiplication only, a method to perform 2's complement multiply using the unsigned multiply is needed.

Let A and B be 2's complement 16 bit integers. Consider the following cases.

1. $A \geq 0$, $B \geq 0$. In this case the unsigned multiply result is $A \times B$, which is also the 2's complement multiply result. Thus no further processing is needed.
2. $A \geq 0$, $B < 0$. In this case the unsigned multiply result is $(2^{16}) \times A - A \times |B|$. However the desired result is $(2^{32}) - A \times |B|$. Thus we need to add $(2^{32}) - (2^{16}) \times A$ to the unsigned multiply result to obtain the correct value.
3. $A < 0$, $B \geq 0$. This case is similar to the previous one. $(2^{32}) - (2^{16}) \times B$ should be added to the unsigned multiply result to get the correct answer.
4. $A < 0$, $B < 0$. The unsigned multiply result in this case is $(2^{32}) - (2^{16}) \times (|A| + |B|) + |A| \times |B|$. The desired result in this case is $|A| \times |B|$. To get the correct answer, add $(2^{16}) \times (|A| + |B|)$ to the unsigned multiply result.

Based on the above discussion, an algorithm for 2's complement multiplication, where the result is a 32 bit 2's complement integer is shown in *Figure 4*.

1. Let A and B be the two 2's complement integers to be multiplied.
2. Compute $C = A \times B$, the unsigned product of A and B. Let the upper half of C be C-hi and its lower half C-lo.
3. If A is negative, then add $(2^{16}) - B$ to C-hi. This can be easily done using the SET C, SUBC instructions of the HPC. Let the result be C-hi1.
4. If B is negative, then add $(2^{16}) - A$ to C-hi1. Again it is easily done using the SET C, SUBC instructions. Let the result be C-hi2.
5. The 2's complement product of A and B is C-hi2. C-lo.

FIGURE 4. Algorithm for 2's Complement Multiplication.

MULTIPLICATION BY FILTER COEFFICIENTS

The coefficients that arise in most IIR filter designs are numbers that are usually in the range from $-2 < \text{coeff} < 2$. The coefficients, in most instances can be scaled to be in this range. The action of digital filtering involves successive multiplications. If we want no loss in accuracy due to multiplication, the word length needed to store successive partial products increases rapidly—clearly an impractical choice. Thus the results of the multiplication at the various stages need to be truncated to 16 bits before proceeding to the next stage. The program FILTER does this as follows: The filter state variables are regarded as integers, while the filter coefficients are regarded as fixed point fractions with the binary point to the immediate right of the sign bit. After the multiplication, the result is shifted so that the integer part of the product is in one word, and the fractional part in another. The integer part is then returned as the result of the multiplication, i.e. the product is truncated to 16 bits. This is performed by the subroutine SMULT. Since the filter coefficients are regarded as fixed point fractions, only coefficients in the range $-1 < \text{coeff} < 1$ can be represented. However, as discussed earlier, the coefficients are usually in the $-2 < \text{coeff} < 2$ range. This is handled by storing half the coefficient value, and SMULT performs a multiplication by 2 (Shift left) to compensate for it. This is why the coefficient values need to be half their value—a fact mentioned earlier.

REFERENCES

1. A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, New Jersey, 1975.
2. L.R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, New Jersey, 1975.
3. H.T. Nagle and V.P. Nelson, ''Digital Filter Implementation on 16-bit Microcomputers'', *IEEE Micro,* Feb. 1981, pp. 23–41.

The code listed in this App Note is available on Dial-A-Helper.

Dial-A-Helper is a service provided by the Microcontroller Applications Group. The Dial-A-Helper system provides access to an automated information storage and retrieval system that may be accessed over standard dial-up telephone lines 24 hours a day. The system capabilities include a MESSAGE SECTION (electronic mail) for communicating to and from the Microcontroller Applications Group and a FILE SECTION mode that can be used to search out and retrieve application data about NSC Microcontrollers. The minimum system requirement is a dumb terminal, 300 or 1200 baud modem, and a telephone.

With a communications package and a PC, the code detailed in this App Note can be downloaded from the FILE SECTION to disk for later use. The Dial-A-Helper telephone lines are:

Modem (408) 739-1162
Voice (408) 721-5582

For Additional Information, Please Contact Factory

APPENDIX A

Listing of Code for the Program FILTER

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  1
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

             ;
             ; THIS IS A DEMO PROGRAM TO ILLUSTRATE THE IMPLEMENTATION OF A DIGITAL
             ; FILTER ON THE HPC. THE PROGRAM CAN BE USED TO IMPLEMENT CASCADED
             ; SECOND ORDER STAGES. THE MAXIMUM NUMBER OF CASCADED STAGES POSSIBLE
             ; IS 8 (I.E. THE MAXIMUM FILTER ORDER IS 16).
             ;
             ; THE PROGRAM IS DESIGNED FOR THE ANALOG INTERFACE BEING THROUGH
             ; A CODEC. THE CODEC OUTPUT AND INPUT ARE INTERFACED TO THE HPC USING
             ; MICROWIRE/PLUS. THIS RESTRICTS THE SAMPLING RATE TO 8 KHZ. ALSO, AT
             ; THIS SAMPLING RATE, THE HPC CAN ONLY IMPLEMENT A SECOND ORDER FILTER.
             ; IF A DIFFERENT ANALOG INTERFACE THAT ALLOWS A LOWER SAMPLING RATE IS
             ; USED, HIGHER ORDER FILTERS CAN BE IMPLEMENTED. THIS WILL INVOLVE CHANGES
             ; TO THE FOLLOWING SUBROUTINES: INPUT, OUTPUT AND THE PORTIONS OF INIT
             ; CONCERNED WITH CODEC INITIALIZATION.
             ;
             ; THE PROGRAM IS BASED ON THE DESCRIPTION GIVE IN:
             ;
             ;       H.T. NAGLE AND V.P. NELSON, "DIGITAL FILTER IMPLEMENTATION
             ;       ON 16-BIT MICROCOMPUTERS," IEEE MICRO, FEB. 1981, 23-41.
             ;
             ;
                             .TITLE FILTER
             ;
             ; DEFINE FILTER VARIABLES AND STORAGE.
             ;
     0000                    YOUT = M(00)           ; OUTPUT SAMPLE STORAGE.
     0002                    YOFK = W(02)           ; TEMPORARY STORAGE.
     0004                    NSTG = W(04)           ; NUMBER OF FILTER STAGES.
     0006                    NCNT = W(06)           ; TEMPORARY STORAGE.
     0008                    PTEMP = W(08)          ; TEMPORARY STORAGE.
     000A                    MTEMP = W(0A)          ; TEMPORARY STORAGE.
     000C                    A0ADDR = W(0C)         ; ADDRESS OF START OF A0 AREA.
     000E                    A1ADDR = W(0E)         ; ADDR. OF START OF A1 AREA.
     0010                    A2ADDR = W(010)        ; ADDR. OF START OF A2 AREA.
     0012                    B1ADDR = W(012)        ; ADDR. OF START OF B1 AREA.
     0014                    B2ADDR = W(014)        ; ADDR. OF START OF B2 AREA.
     0016                    M0ADDR = W(016)        ; ADDR. OF START OF M0 AREA.
     0018                    M1ADDR = W(018)        ; ADDR. OF START OF M1 AREA.
     001A                    M2ADDR = W(01A)        ; ADDR. OF START OF M2 AREA.
     001C                    T1ADDR = W(01C)        ; ADDR. OF START OF T1 AREA.
     001E                    T2ADDR = W(01E)        ; ADDR. OF START OF T2 AREA.
             ; MAXIMUM NUMBER OF STAGES IS 8.
     0020                    A0 = W(020)            ; COEFF. A0.
     0030                    A1 = W(030)            ; COEFF. A1.
     0040                    A2 = W(040)            ; COEFF. A2.
     0050                    B1 = W(050)            ; COEFF. B1.
     0060                    B2 = W(060)            ; COEFF. B2.
     0070                    M0 = W(070)            ; M(K).
     0080                    M1 = W(080)            ; M(K-1).
     0090                    M2 = W(090)            ; M(K-2).
     00A0                    T1 = W(0A0)            ; T1.
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  2
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

        00B0                     T2 = W(0B0)             ; T2.
                    ;
                    ; DEFINITION OF HPC REGISTER NAMES.
                    ;
        00C0                     PSW = M(00C0)
        00D0                     ENIR = M(00D0)
        00D2                     IRPD = M(00D2)
        00D4                     IRCD = M(00D4)
        00D6                     SIO = M(00D6)
        00D8                     PORTI = M(00D8)
        00E2                     PORTBL = M(00E2)
        00E3                     PORTBH = M(00E3)
        00E2                     PORTB = W(00E2)
        00F2                     DIRBL = M(00F2)
        00F3                     DIRBH = M(00F3)
        00F2                     DIRB = W(00F2)
        00F4                     BFUNL = M(00F4)
        00F5                     BFUNH = M(00F5)
        00F4                     BFUN = W(00F4)
        0188                     T2TIM = W(0188)
        0186                     T2REG = W(0186)
        018E                     DIVBYL = M(018E)
        018F                     DIVBYH = M(018F)
        018E                     DIVBY = W(018E)
        0190                     TMMDL = M(0190)
        0191                     TMMDH = M(0191)
        0190                     TMMD = W(0190)
                    ;
                    ; INCLUDE THE MU-LAW TO LINEAR CODE CONVERSION TABLE.
                    ;
                                 .INCLD MUTBL.MAC
   F000                          MUTBL, 0F000
                    ;
    F200                         . = 0F200
                    FILTER:
   F200 B701F0C4                 LD SP, 01F0             ; INITIALIZE STACK POINTER.
   F204 305A                     JSR INIT                ; INITIALIZE THE CODEC
                                                         ; AND FILTER VARIABLES.
                    ; NEXT COMES THE BASIC FILTER LOOP.
                    FLOOP:
   F206 3101                     JSR INPUT               ; GET INPUT SAMPLE, OUTPUT
                                                         ; PREVIOUS FILTER OUTPUT.
   F208 311B                     JSR YCOMP               ; COMPUTE NEW OUTPUT.
   F20A 315B                     JSR OUTPUT              ; CONVERT OUTPUT VALUE TO
                                                         ; MU-255 LAW AND SAVE.
   F20C 31AA                     JSR PRECOM              ; PRECOMPUTE FOR NEXT SAMPLE.
   F20E 68                       JP FLOOP                ; GO DO NEXT SAMPLE.
                                 .LOCAL
                    ;
                    ;
                    ;
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  3
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

                 ; THIS SUBROUTINE COMPUTES 2*F*I, WHERE F IS A 2'S COMPLEMENT
                 ; BINARY FRACTION AND I IS A 2'S COMPLEMENT INTEGER. THE INTEGER
                 ; PART OF THE PRODUCT IS RETURNED IN A. ON INPUT, EITHER F OR I
                 ; SHOULD BE IN A AND THE ADDRESS OF THE OTHER IN B.
                 ;
                 ;
                 SMULT:
F20F B700000A                   LD MTEMP, 0             ; CLEAR TEMPORARY STORAGE.
F213 A9CC                       INC B                   ; B NOW POINTS TO UPPER BYTE
                                                        ; OF MULTIPLIER.
F215 17                         IF M(B).7               ; IS IT NEGATIVE?
F216 AB0A                       ST A, MTEMP             ; THEN SAVE MULTIPLICAND IN MTEMP.
F218 AACC                       DECSZ B                 ; B INTO WORD POINTER.
F21A 40                         NOP
F21B AE0A                       X A, MTEMP              ; SWAP A AND MTEMP.
F21D 960B17                     IF M(($MTEMP) + 1).7    ; IS MULTIPLICAND NEGATIVE?
F220 F8                         ADD A, W(B)             ; THEN ACCUMULATE MULTIPLIER
F221 AE0A                       X A, MTEMP
F223 FE                         MULT A, W(B)            ; UNSIGNED MULTIPLY.
F224 AECE                       X A, X                  ; UPPER HALF IN A.
F226 02                         SET C
F227 960AEB                     SUBC A, MTEMP
F22A E7                         SHL A
F22B 96CF17                     IF H(X).7
F22E 04                         INC A
F22F E7                         SHL A
F230 96CF16                     IF H(X).6
F233 04                         INC A
F234 3C                         RET
                 ;
                 ;
                 ; THIS SUBROUTINE PERFORMS THE INITIALIZATION FOR THE FILTER.
                 ; IT DOES THE FOLLOWING:
                 ;              1. SET UP THE FILTER VARIABLES.
                 ;              2. COPY THE FILTER COEFFS. FROM ROM TO ON CHIP RAM.
                 ;              3. INITIALIZE AND START THE CODEC.
                 ;
                 ;
                 ; DEFINE FILTER COEFFICIENTS.
                 ;
F235 40                         .EVEN
F236 0400        ROMNST:        .WORD 4
F238 C430        ROMA0:         .WORD 12484, 3217, 4574, 7636
F23A 910C
F23C DE11
F23E D41D
F240 C430        ROMA1:         .WORD 12484, 3217, 4574, 7636
F242 910C
F244 DE11
F246 D41D
F248 C430        ROMA2:         .WORD 12484, 3217, 4574, 7636
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  4
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

    F24A 910C
    F24C DE11
    F24E D41D
    F250 B939       ROMB1:          .WORD 14777, 9826, 19308, 11207
    F252 6226
    F254 6C4B
    F256 C72B
    F258 A1D5       ROMB2:          .WORD -10847, -15783, -6940, -14068
    F25A 59C2
    F25C E4E4
    F25E 0CC9
                    INIT:
    F260 B6F236A8                   LD A, W(ROMNST)
    F264 AB04                       ST A, NSTG              ; SET UP NO. OF STAGES.
    F266 9020                       LD A, $A0
    F268 AB0C                       ST A, A0ADDR            ; COPY ADDRESS OF A0 AREA.
    F26A 9030                       LD A, $A1
    F26C AB0E                       ST A, A1ADDR            ; COPY ADDRESS OF A1 AREA.
    F26E 9040                       LD A, $A2
    F270 AB10                       ST A, A2ADDR            ; COPY ADDRESS OF A2 AREA.
    F272 9050                       LD A, $B1
    F274 AB12                       ST A, B1ADDR            ; COPY ADDRESS OF B1 AREA.
    F276 9060                       LD A, $B2
    F278 AB14                       ST A, B2ADDR            ; COPY ADDRESS OF B2 AREA.
    F27A 9070                       LD A, $M0
    F27C AB16                       ST A, M0ADDR            ; COPY ADDRESS OF M0 AREA.
    F27E 9080                       LD A, $M1
    F280 AB18                       ST A, M1ADDR            ; COPY ADDRESS OF M1 AREA.
    F282 9090                       LD A, $M2
    F284 AB1A                       ST A, M2ADDR            ; COPY ADDRESS OF M2 AREA.
    F286 90A0                       LD A, $T1
    F288 AB1C                       ST A, T1ADDR            ; COPY ADDRESS OF T1 AREA.
    F28A 9080                       LD A, $T2
    F28C AB1E                       ST A, T2ADDR            ; COPY ADDRESS OF T2 AREA.
                    ;
                    ; COPY THE A0 COEFFS. TO ON-CHIP RAM.
                    ;
    F28E B3F238                     LD X, ROMA0
    F291 9220                       LD B, $A0
    F293 AC04CA                     LD K, NSTG
                    CA0LP:
    F296 F0                         LD A, W(X+)
    F297 E1                         XS A, W(B+)
    F298 40                         NOP
    F299 AACA                       DECSZ K
    F29B 65                         JP CA0LP
                    ;
                    ; COPY THE A1 COEFFS. TO ON-CHIP RAM.
    F29C B3F240                     LD X, ROMA1
    F29F 9230                       LD B, $A1
    F2A1 AC04CA                     LD K, NSTG
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  5
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

                   CA1LP:
    F2A4 F0                         LD A, W(X+)
    F2A5 E1                         XS A, W(B+)
    F2A6 40                         NOP
    F2A7 AACA                       DECSZ K
    F2A9 65                         JP CA1LP
                   ;
                   ; COPY THE A2 COEFFS. TO ON-CHIP RAM.
    F2AA B3F248                     LD X, ROMA2
    F2AD 9240                       LD B, $A2
    F2AF AC04CA                     LD K, NSTG
                   CA2LP:
    F2B2 F0                         LD A, W(X+)
    F2B3 E1                         XS A, W(B+)
    F2B4 40                         NOP
    F2B5 AACA                       DECSZ K
    F2B7 65                         JP CA2LP
                   ;
                   ; COPY THE B1 COEFFS. TO ON-CHIP RAM.
    F2BB B3F250                     LD X, ROMB1
    F2BB 9250                       LD B, $B1
    F2BD AC04CA                     LD K, NSTG
                   CB1LP:
    F2C0 F0                         LD A, W(X+)
    F2C1 E1                         XS A, W(B+)
    F2C2 40                         NOP
    F2C3 AACA                       DECSZ K
    F2C5 65                         JP CB1LP
                   ;
                   ; COPY THE B2 COEFFS. TO ON-CHIP RAM.
    F2C6 B3F258                     LD X, ROMB2
    F2C9 9260                       LD B, $B2
    F2CB AC04CA                     LD K, NSTG
                   CB2LP:
    F2CE F0                         LD A, W(X+)
    F2CF E1                         XS A, W(B+)
    F2D0 40                         NOP
    F2D1 AACA                       DECSZ K
    F2D3 65                         JP CB2LP
                   ;
                   ; ZERO OUT THE REST OF USER BASE PAGE RAM.
                   ;
    F2D4 8D70BE                     LD BK, $M0, OBE
                   ZEROLP:
    F2D7 00                         CLR A
    F2D8 E1                         XS A, W(B+)
    F2D9 62                         JP ZEROLP
                   ;
                   ;
                   ; NOW INITIALIZE AND START THE CODEC.
                   ;
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  6
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

                    ;
F2DA B7FFB7F2                   LD DIRB,0FFB7           ; SET B3 (T2I0) AND B6 (SK)
                                                        ; ON PORT B AS INPUTS. SET ALL
                                                        ; OTHER PINS ON B AS OUTPUT.
F2DE B70000E2                   LD PORTB, 0             ; OUTPUT 0 ON ALL PORT B PINS.
F2E2 96F40B                     SET BFUNL.3             ; ALT. FUN. ON B3-T2I0.
F2E5 96F40D                     SET BFUNL.5             ; ALT. FUN. ON B5-S0.
F2E8 96F508                     SET BFUNH.0             ; ALT. FUN. ON B8-TS0.
F2EB 9700D0                     LD ENIR, 0              ; DISABLE INTRPTS.
F2EE 9700D4                     LD IRCD, 0              ; SELECT SLAVE MODE FOR M-WIRE.
F2F1 83070188AB                 LD T2TIM, 07            ; LOAD 7-DEC INTO T2 TIMER.
F2F6 83070186AB                 LD T2REG, 07            ; LOAD 7-DEC INTO T2 REG.
F2FB 8300018F8B                 LD DIVBYH, 0            ; SELECT EXT. CLOCK FOR T2 TIMER.
                    ;
F300 8ED6                       X A, SIO
F302 8740400190AB               LD TMMD, 04040          ; START TIMER T2.
F308 3C                         RET
                    ;
                    ;
                    ; THIS SUBROUTINE OUTPUTS THE PREVIOUS Y(K) TO THE CODEC AND READS
                    ; THE NEW INPUT VALUE. THEN THE MU-255 VALUE IS CONVERTED TO LINEAR
                    ; BY TABLE LOOK UP. THE TABLE IS ASSUMED TO START AT F000.
                    ;
                    ;
                    INPUT:
F309 AB02                       LD A, YOFK              ; GET DATA TO BE OUTPUT.
                    NOTDN:
F30B 96D210                     IF IRPD.0               ; IS MICROWIRE DONE?
F30E 41                         JP MWDONE               ; YES, SO GET DATA.
F30F 64                         JP NOTDN                ; NO, SO TRY AGAIN.
                    MWDONE:
F310 8ED6                       X A, SIO                ; GET NEW SAMPLE, OUTPUT
                                                        ; COMPUTED DATA.
F312 01                         COMP A                  ; TAKE CARE OF CODEC INVERSION.
F313 99FF                       AND A, 0FF
F315 E7                         SHL A
F316 BAF000                     OR A, 0F000             ; FORM MU-LAW TO LINEAR
                                                        ; TABLE ADDRESS.
F319 AECE                       X A, X
F31B D0                         LD A, M(X+)             ; GET LINEAR VALUE
F31C AECA                       X A, K
F31E D4                         LD A, M(X)              ; A BYTE AT A TIME.
F31F 8CC8CB                     LD H(K), L(A)
F322 A8CA                       LD A, K
F324 3C                         RET
                    ;
                    ;
                    YCOMP:
                    ; THIS SUBROUTINE COMPUTES THE OUTPUT SAMPLE Y(K).
                    ; THE INPUT SAMPLE X(K) IS INPUT IN REG. A.
                    ; THE OUTPUT IS RETURNED IN REG. A.
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  7
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

                ;
F325 AC0406                     LD NCNT, NSTG           ; COPY THE NUMBER OF STAGES TO
                                                        ; NCNT.
                YLOOP:
F328 AD1CF8                     ADD A, W(T1ADDR)        ; A ≤ X(K) + T1.
F32B AD16AB                     ST A, W(M0ADDR)         ; M(K) ≤ X(K) + T1.
F32E AC0CCC                     LD B, A0ADDR            ; B ≤ ADDR(A0).
F331 3522                       JSR SMULT               ; A ≤ A0*M(K).
F333 AD1EF8                     ADD A, W(T2ADDR)        ; A ≤ A0*M(K) + T2.
                                                        ;
F336 AA06                       DECSZ NCNT              ; DONE ALL STAGES?
F338 941B       R               JMP YMORE               ; NO GO DO SOME MORE.
                                                        ;
                                                        ; GET HERE MEANS ALL STAGES DONE.
F33A AB02                       ST A, YOFK              ; SAVE TEMPORARILY.
F33C A804                       LD A, NSTG
F33E 05                         DEC A
F33F E7                         SHL A
F340 01                         COMP A
F341 04                         INC A                   ; A ≤ -2*(NSTG-1).
F342 A0C81CF8                   ADD T1ADDR, A           ; RESTORE T1ADDR.
F346 A0C816F8                   ADD M0ADDR, A           ; RESTORE M0ADDR.
F34A A0C80CF8                   ADD A0ADDR, A           ; RESTORE A0ADDR.
F34E A0C81EF8                   ADD T2ADDR, A           ; RESTORE T2ADDR.
F352 A802                       LD A, YOFK              ; A ≤ Y(K).
F354 3C                         RET
                ;
                ; PREPARE FOR NEXT STAGE ITERATION.
                ;
                YMORE:
F355 82021CF8                   ADD T1ADDR, 02
F359 820216F8                   ADD M0ADDR, 02
F35D 82020CF8                   ADD A0ADDR, 02
F361 82021EF8                   ADD T2ADDR, 02
F365 953D                       JMP YLOOP
                ;
                ; THIS SUBROUTINE CONVERTS THE 16 BIT OUTPUT VALUE TO
                ; 8 BIT MU-LAW.
                ;
                OUTPUT:
F367 96D41F                     RESET IRCD.7
F36A E7                         SHL A                   ; SIGN BIT TO C.
F36B 06                         IFN C                   ; IS IT POSITIVE?
F36C 45                         JP OPOS
F36D 96D40F                     SET IRCD.7
F370 01                         COMP A
F371 04                         INC A                   ; NEGATIVE, SO TAKE 2'S
                                                        ; COMPLEMENT.
                ; OPOS:
F372 B80108                     ADD A, 0108             ; ADD BIAS.
F375 9107                       LD K, 07                ; SET UP COUNTER.
```

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  8
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

                ALIGN:
F377 E7                     SHL A                ; LOOP AND LOCATE MS 1 BIT.
F378 07                     IF C
F379 44                     JP ODONE             ; FOUND MS 1 BIT.
F37A AACA                   DECSZ K
F37C 65                     JP ALIGN
F37D E7                     SHL A                ; HAS TO BE 1 IN C NOW.
                ODONE:
F37E AECA                   X R, K
F380 E7                     SHL A
F381 E7                     SHL A
F382 E7                     SHL A
F383 E7                     SHL A                ; COUNTER VALUE IN BITS 4-6.
F384 AECC                   X A, B
F386 00                     CLR A
F387 88CB                   LD A, H(K)
F389 3B                     SWAP A
F38A 990F                   AND A, 0F
F38C 96CCFA                 OR A, B
F38F 96D417                 IF IRCD.7
F392 96C80F                 SET A.7
F395 01                     COMP A
F396 8B00                   ST A, YOUT
F398 3C                     RET
                ;
                ; THIS SUBROUTINE UPDATES M(K-1) AND M(K-2) FOR THE NEXT SAMPLE.
                ;
F399 AC1ACC                 LD B, M2ADDR         ; B ≤ ADDR(M2),
F39C AC04CA                 LD K, NSTG           ; K ≤ NSTG.
F39F AC18CE                 LD X, M1ADDR         ; X ≤ ADDR(M1).
                DLYLP1:
F3A2 F0                     LD A, W(X+)          ; A ≤ M(K-1).
F3A3 E1                     XS A, W(B+)          ; M(K-2) ≤ M(K-1).
F3A4 40                     NOP
F3A5 AACA                   DECSZ K
F3A7 65                     JP DLYLP1
                ;
F3A8 AC18CC                 LD B, M1ADDR         ; B ≤ ADDR(M1),
F3AB AC04CA                 LD K, NSTG           ; K ≤ NSTG.
F3AE AC16CE                 LD X, M0ADDR         ; X ≤ ADDR(M0).
                DLYLP2:
F3B1 F0                     LD A, W(X+)          ; A ≤ M(K).
F3B2 E1                     XS A, W(B+)          ; M(K-1) ≤ M(K).
F3B3 40                     NOP
F3B4 AACA                   DECSZ K
F3B6 65                     JP DLYLP2
F3B7 3C                     RET
                ;
                ;
                PRECOMP:
                ; THIS SUBROUTINE PRECOMPUTES T1 AND T2 BEFORE THE NEXT INPUT
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  9
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

                  ; SAMPLE ARRIVES.
                  ;
F3B8 AC0406                   LD NCNT, NSTG            ; COPY NO. OF STAGES.
                  PRELP:
F3BB AD18A8                   LD A, W(M1ADDR)          ; A ≤ M(K-1).
F3BE AC12CC                   LD B, B1ADDR             ; B ≤ ADDR(-B1).
F3C1 35B2                     JSR SMULT                ; A ≤ -B1*M(K-1).
F3C3 AB08                     ST A, PTEMP
F3C5 AD1AA8                   LD A, W(M2ADDR)          ; A ≤ M(K-2).
F3C8 AC14CC                   LD B,B2ADDR              ; B ≤ ADDR(-B2).
F3CB 35BC                     JSR SMULT                ; A ≤ -B2*M(K-2).
F3CD 9608F8                   ADD A, PTEMP             ; A ≤ -B1*M(K-1) - B2*M(K-2).
F3D0 AD1CAB                   ST A,W(T1ADDR)
F3D3 AD18A8                   LD A, W(M1ADDR)          ; A ≤ M(K-1).
F3D6 AC0ECC                   LD B, A1ADDR             ; B ≤ ADDR(A1).
F3D9 35CA                     JSR SMULT                ; A ≤ A1*M(K-1).
F3DB AB08                     ST A, PTEMP
F3DD AD1AA8                   LD A, W(M2ADDR)          ; A ≤ M(K-2).
F3E0 AC10CC                   LD B, A2ADDR             ; B ≤ ADDR(A2).
F3E3 3504                     JSR SMULT                ; A ≤ A2*M(K-2).
F3E5 9608F8                   ADD A, PTEMP             ; A ≤ A1*M(K-1) + A2*M(K-2).
                                                       ;
F3E8 AA06                     DECSZ NCNT               ; DONE ALL STAGES?
F3EA 9427                     JMP PMORE                ; NO, GO DO SOME MORE.
                                                       ;
                                                       ; GET HERE MEANS DONE ALL STAGES.
F3EC A804                     LD A, NSTG
F3EE 05                       DEC A
F3EF E7                       SHL A
F3F0 01                       COMP A
F3F1 04                       INC A                    ; A ≤ -2*(NSTG - 1).
F3F2 A0C818F8                 ADD M1ADDR, A            ; RESTORE M1ADDR.
F3F6 A0C81AF8                 ADD M2ADDR, A            ; RESTORE M2ADDR.
F3FA A0C81CF8                 ADD T1ADDR, A            ; RESTORE T1ADDR.
F3FE A0C81EF8                 ADD T2ADDR, A            ; RESTORE T2ADDR.
F402 A0C812F8                 ADD B1ADDR, A            ; RESTORE B1ADDR.
F406 A0C814F8                 ADD B2ADDR, A            ; RESTORE B2ADDR.
F40A A0C80EF8                 ADD A1ADDR, A            ; RESTORE A1ADDR.
F40E A0C810F8                 ADD A2ADDR, A            ; RESTORE A2ADDR.
F412 3C                       RET
                  ;
                  ; PREPARE FOR NEXT STAGE ITERATION.
                  ;
                  PMORE:
F413 820218F8                 ADD M1ADDR, 02           ; UPDATE M1ADDR.
F417 82021AF8                 ADD M2ADDR, 02           ; UPDATE M2ADDR.
F41B 82021CF8                 ADD T1ADDR, 02           ; UPDATE T1ADDR.
F41F 82021EF8                 ADD T2ADDR, 02           ; UPDATE T2ADDR.
F423 820212F8                 ADD B1ADDR, 02           ; UPDATE B1ADDR.
F427 820214F8                 ADD B2ADDR, 02           ; UPDATE B2ADDR.
F42B 82020EF8                 ADD A1ADDR, 02           ; UPDATE A1ADDR.
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  10
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

445 F42F 820210F8                  ADD A2ADDR, 02          ; UPDATE A2ADDR.
446 F433 9578                      JMP PRELP
447                 ;
448                 ;
449 FFFE 00F2                      .END FILTER
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  11
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER
SYMBOL TABLE

A      00C8 W        A0     0020 W        A0ADDR 000C W        A1     0030 W
A1ADDR 000E W        A2     0040 W        A2ADDR 0010 W        ALIGN  F377
B      00CC W        B1     0050 W        B1ADDR 0012 W        B2     0060 W
B2ADDR 0014 W        BFUN   00F4 W*       BFUNH  00F5 M        BFUNL  00F4 M
CA0LP  F296          CA1LP  F2A4          CA2LP  F2B2          CB1LP  F2C0
CB2LP  F2CE          DIRB   00F2 W        DIRBH  00F3 M*       DIRBL  00F2 M*
DIVBY  018E W*       DIVBYH 018F M        DIVBYL 018E M*       DLYLP1 F3A2
DLYLP2 F3B1          ENIR   00D0 M        FILTER F200          FLOOP  F206
INCRM  0200          INIT   F260          INPUT  F309          IRCD   00D4 M
IRPD   00D2 M        K      00CA W        M0     0070 W        M0ADDR 0016 W
M1     0080 W        M1ADDR 0018 W        M2     0090 W        M2ADDR 001A W
MTEMP  000A W        MUAL   205F          MWDONE F310          NCNT   0006W
NOTDN  F30B          NSTG   0004 W        ODONE  F37E          OPOS   F372
OUTPUT F367          PC     00C6 W        PMORE  F413          PORTB  00E2 W
PORTBH 00E3 M*       PORTBL 00E2 M*       PORTI  0008 M*       PRECOM F3B8
PRELP  F3BB          PSW    00C0 M*       PTEMP  0008 W        ROMA0  F238
ROMA1  F240          ROMA2  F248          ROMB1  F250          ROMB2  F258
ROMNST F236          RVAL   E0A1          SIO    00D6 M        SMULT  F20F
SP     00C4 W        SVAL   2100          T1     00A0 W        T1ADDR 001C W
T2     00B0 W        T2ADDR 001E W        T2REG  0186 W        T2TIM  0188 W
TMMD   0190 W        TMMDH  0191 M*       TMMDL  0190 M*       X      00CE W
YCOMP  F325          YLOOP  F328          YMORE  F355          YOFK   0002 W
YOUT   0000 M        ZEROLP F2D7
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

```
NATIONAL SEMICONDUCTOR CORPORATION   PAGE:  12
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER
MACRO TABLE

MUTBL
   NO WARNING LINES

   NO ERROR LINES

 1079 ROM BYTES USED

SOURCE CHECKSUM = 4769
OBJECT CHECKSUM = 1378

INPUT   FILE C:FILTER.MAC
LISTING FILE C:FILTER.PRN
OBJECT  FILE C:FILTER.LM
```

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408