Interfacing Analog Audio Bandwidth Signals to the HPC

National Semiconductor
Application Note 484
Ashok Krishnamurthy
April 1987

INTRODUCTION

This report describes a method of interfacing analog audio bandwidth signals to the National Semiconductor HPC microcontroller. The analog signal is converted to a digital value using the National Semiconductor TP3054 codec/filter combo. The digital value is then transferred to the HPC using the MICROWIRE/PLUS™ synchronous serial interface. The digital output sample computed by the HPC is also transferred to the TP3054 using the MICROWIRE/PLUS interface. The TP3054 then converts this digital value to an analog signal.

ADVANTAGES OF USING A CODEC

There are a number of advantages in using a codec for A/D and D/A conversion of analog signals.

1. The codec/filter combos such as the TP3054 integrate a number of functions on a single chip. Thus the TP3054 includes the analog anti-aliasing filters, the Sample-and-Hold circuitry and the A/D and D/A converters for analog signal interfacing.
2. The μ-law coding effectively codes a 14-bit conversion accuracy in 8 bits. This allows the interface to the HPC to be greatly simplified.

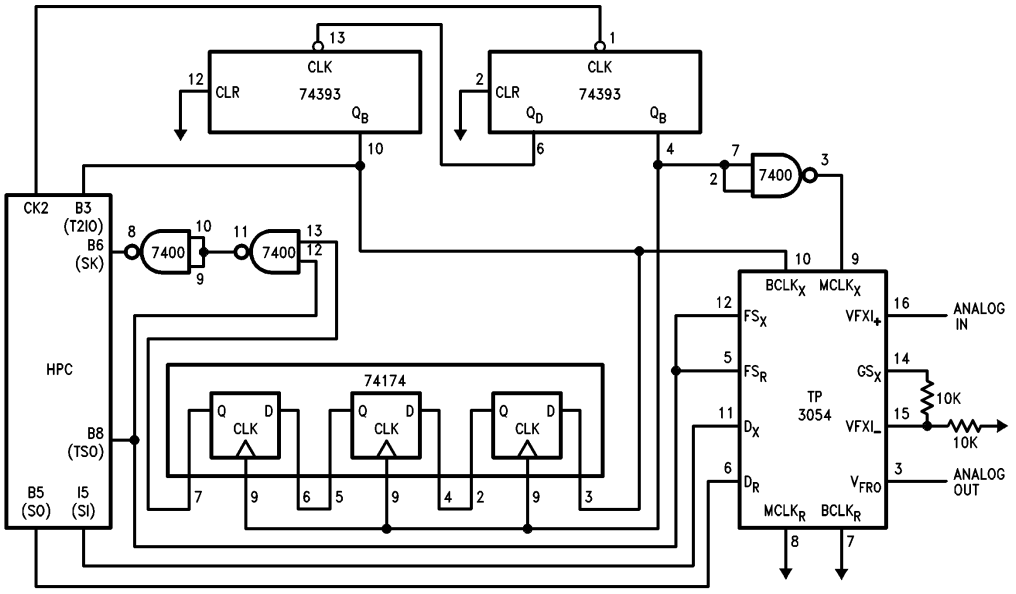
DISADVANTAGES IN USING A CODEC

While the use of a codec is appropriate for audio (in particular speech) processing applications, it has a number of disadvantages in other cases.

1. The sampling rate is fixed at 8 kHz. If lower or higher sampling rates are desired, the codec cannot be used. Note that the real-time signal processing that the HPC can perform at a 8 kHz sampling rate is limited.
2. The resolution is fixed, and is about 14 bits/sample.
3. Digital filtering algorithms require that the samples used in the processing be linear coded PCM. Thus the 8-bit μ-law PCM values output by the codec need to be converted to linear coded PCM. Correspondingly, the output of the digital filter, which is in linear coded PCM needs to be converted to 8-bit μ-law PCM before outputting to the codec. This requires additional processing per sample.

DESCRIPTION OF THE INTERFACE

The circuit schematic of the interface is shown in *Figure 1*. Note that the schematic does not show complete details of the HPC. Only the HPC pins that are relevant to this interface are shown. A wire-wrapped version of the circuit has been constructed on a NSC HPC 16040 Chip Carrier Board.



TL/DD/9246–1

FIGURE 1. Circuit Schematic

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

Note that this report does not go into the details about the use of the TP3054 codec chip or programming the HPC. It also does not discuss the μ-law to linear and linear to μ-law code conversion in detail. For more information on these issues, please consult the references listed at the end.

1. **Codec Signalling Considerations.** The TP3054 can operate in either synchronous or asynchronous modes. Further, in each of these modes, it uses short or long frame sync operation. The circuit shown in *Figure 1* runs the codec in synchronous mode with long-frame-sync operation.

The codec requires 4 clock sources for proper operation in the synchronous mode. These are MCLK-x, BCLK-x, FS-x and FS-r. MCLK-x is a master clock and is used to clock the switched-capacitor filters. BCLK-x is the bit shift clock, and FS-x and FS-r are the frame sync clocks. These clocks need to be synchronous.

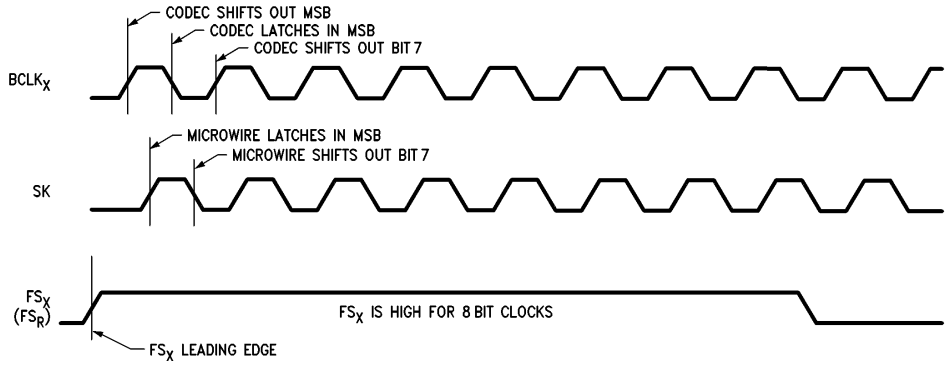
These clocks are obtained in the circuit as follows. MCLK-x is obtained by dividing the HPC CK2 clock output by 4. If the HPC is using a 16 MHz crystal, this results in MCKL-x being 2 MHz.

BCLK-x is obtained by dividing CK2 by 64. This gives an effective value for BCLK-x of 125 kHz. Note that MCLK-x is inverted before being fed to the codec. This is done to synchronize MCLK-x and BCLK-x on their leading edges.

FS-x and FS-r are the same clocks in the circuit. They are obtained by dividing BCLK-x by 16 using the timer T2 on the HPC. BCLK-x is used as the external clock input on pin T2IO of the HPC and FS-x (FS-r) is obtained from the timer synchronous output TS0. Note that the delay inherent in the HPC between the underflow of a timer and the toggling of the corresponding output allows FS-x and BCLK-x to be leading edge synchronized (more accurately, the delay is within the codec's acceptable limits.) Note that in order to accomplish these functions, the HPC pins need to be properly configured. This is not described here. Please refer to the appropriate HPC documentation and consult the sample program included with this report.

2. **MICROWIRE/PLUS Interface Considerations.** MICROWIRE/PLUS is a National Semiconductor defined 8-bit serial synchronous communication interface. It is designed to allow easy interfacing of NSC microcontrollers and peripheral chips. The HPC microcontroller has a MICROWIRE/PLUS interface; however the TP3054 codec does not. Thus some external ''glue logic'' is necessary to allow the HPC and the TP3054 to be interfaced.

The HPC MICROWIRE/PLUS interface is operated in Slave mode for this application. This means that the shift clock needed to latch-in/shift-out data from the Micro-wire SIO register is provided externally on the SK pin. Micro-wire latches in data on the leading edge of the SK clock and shifts out data on the trailing edge of SK. Also SK needs to be a burst clock for proper operation.



TL/DD/9246–2

FIGURE 2. Timing Waveforms

The codec shifts out data on the D-x pin on the first 8 leading edges of BCLK-x after a FS-x leading edge. Also, it latches in data on the D-r pin on the first 8 trailing edges of BCLK-x after a FS-r leading edge. Note that FS-x and FS-r are the same in this application. Refer to the timing diagram in *Figure 2*.

Thus, it is seen that there is a timing difference in the way the codec and the Micro-wire interfaces work. However, as seen in *Figure 2*, if the shift clock, SK, to the Microwire interface is delayed with respect to BCLK-x, the two interfaces should work compatibly. This delay is accomplished by clocking BCLK-x through a shift register using MCLK-x as the clock source. This can be seen in the circuit schematic in *Figure 1*. (The author thanks Mr. Richard Lazovick for this suggestion.)

μ-LAW TO LINEAR/LINEAR TO μ-LAW CONVERSION

It was explained earlier that the codec outputs digital values that are companded using the MU-255 PCM standard. However, for linear digital filtering applications, the input needs to be in linear PCM format. Similarly, it is necessary to provide the conversion from linear PCM to MU-255 PCM before output to the codec. The HPC accomplishes this in software.

1. μ-law to linear conversion. The codec output is actually the complement of the μ-law value. Thus, this first needs to be complemented to obtain the true μ-law value. The simplest way to obtain the corresponding linear value is through table look-up. The output of the table is the 16-bit 2's complement linear value. The sample program included with this report utilizes this technique. A macro that constructs this table is also provided.
2. Linear to μ-law conversion. An algorithm to convert a 13-bit positive linear number to 7-bit μ-law is described in *Figure 3*. The algorithm is based on the description in the book by Bellamy listed in the reference. The most significant 8th bit for the μ-law code is obtained from the sign of the input linear code.

1. Get 13-bit positive input value.
2. Add to it the bias value of 31-decimal.
3. The compressed μ-law word is then obtained as follows:

Biased Linear Value Bits												
12	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	1	Q3	Q2	Q1	Q0	a
0	0	0	0	0	0	1	Q3	Q2	Q1	Q0	a	b
0	0	0	0	0	1	Q3	Q2	Q1	Q0	a	b	c
0	0	0	0	1	Q3	Q2	Q1	Q0	a	b	c	d
0	0	0	1	Q3	Q2	Q1	Q0	a	b	c	d	e
0	0	1	Q3	Q2	Q1	Q0	a	b	c	d	e	f
0	1	Q3	Q2	Q1	Q0	a	b	c	d	e	f	g
1	Q3	Q2	Q1	Q0	a	b	c	d	e	f	g	h

μ-Law Value Bits						
6	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	Q3	Q2	Q1	Q0
0	0	1	Q3	Q2	Q1	Q0
0	1	0	Q3	Q2	Q1	Q0
0	1	1	Q3	Q2	Q1	Q0
1	0	0	Q3	Q2	Q1	Q0
1	0	1	Q3	Q2	Q1	Q0
1	1	0	Q3	Q2	Q1	Q0
1	1	1	Q3	Q2	Q1	Q0

FIGURE 3. 13-Bit Linear to 8-Bit μ-Law Conversion

POSSIBLE APPLICATIONS

The codec/HPC interface described above can be used in a number of speech processing applications. One application, ADPCM coding of speech, is presently under development. Other applications include: a voiced/unvoiced/silence classifier, a voice pitch tracker, speech detection circuitry etc. Note that the main limitation here (at least for real-time applications) is the amount of effective computation that can be done by the HPC between samples.

REFERENCES

1. National Semiconductor Corporation, *Telecommunications Databook,* Santa Clara, California, 1984.
2. National Semiconductor Corporation, *HPC Programmers Reference Manual,* Santa Clara, California, 1986.
3. National Semiconductor Corporation, *HPC Hardware Reference Manual,* Santa Clara, California, 1986.
4. J. C. Bellamy, *Digital Telephony,* John Wiley & Sons, New York, 1982.

APPENDIX A

PROGRAM TO TEST CODEC INTERFACE

```
NATIONAL SEMICONDUCTOR CORPORATION   Page:  1
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
TSTCDC

  1             ;
  2             ;
  3                             .TITLE TSTCDC
  4             ;
  5 01C0                        Y0FK = M(01C0)        ; OUTPUT SAMPLE STORAGE.
  6 00C0                        PSW = M(00C0)
  7 00D0                        ENIR = M(00D0)
  8 00D2                        IRPD = M(00D2)
  9 00D4                        IRCD = M(00D4)
 10 00D6                        SI0 = M(00D6)
 11 00D8                        P0RTI = M(00D8)
 12 00E2                        P0RTBL = M(00E2)
 13 00E3                        P0RTBH = M(00E3)
 14 00E2                        P0RTB = W(00E2)
 15 00F2                        DIRBL = M(00F2)
 16 00F3                        DIRBH = M(00F3)
 17 00F2                        DIRB = W(00F2)
 18 00F4                        BFUNL = M(00F4)
 19 00F5                        BFUNH = M(00F5)
 20 00F4                        BFUN = W(00F4)
 21 0188                        T2TIM = W(0188)
 22 0186                        T2REG = W(0186)
 23 018E                        DIVBYL = M(018E)
 24 018F                        DIVBYH = M(018F)
 25 018E                        DIVBY = W(018E)
 26 0190                        TMMDL = M(0190)
 27 0191                        TMMDH = M(0191)
 28 0190                        TMMD = W(0190)
 29             ;
 30             ;
 31             ;
 32                             .MACRO MUTBL,STADR
 33             ;
 34             ; MACRO TO CREATE LOOKUP TABLE FOR MU-255 LAW PCM TO LINEAR CONVERSION.
 35             ; STADR IS THE STARTING ADDRESS FOR THE TABLE, AND MUST BE AN EVEN ADDRESS.
 36             ; THE TABLE OCCUPIES 512 BYTES.
 37             ;
 38                             . = STADR
 39                             .SET SVAL,021
 40                             .SET INCRM,02
 41                             .DO 08
 42                                   .SET MVAL,SVAL-021
 43                                   .DO 010
 44                                     .WORD MVAL
 45                                     .SET MVAL,MVAL+INCRM
 46                                   .ENDDO
 47                                   .SET SVAL,SVAL*02
 48                                   .SET INCRM,INCRM*02
 49                             .ENDDO
 50             ;
 51                             .SET SVAL, 021
```

```
 52                             .SET INCRM, 02
 53                             .DO 08
 54                                   .SET MVAL,SVAL-021
 55                                   .DO 010
 56                                     .SET RVAL,-1*MVAL
 57                                     .WORD RVAL
 58                                     .SET MVAL,MVAL+INCRM
 59                                   .ENDDO
 60                                   .SET SVAL,SVAL*02
 61                                   .SET INCRM,INCRM*02
 62                             .ENDDO
 63             ;
 64                             .ENDM
 65             ;
 66             ;
 67             ;
 68                             .LOCAL
 69 F000                       MUTBL, 0F000
 70             ;
 71 F200                       .= 0F200
 72             CODEC:
 73 F200 B701F0C4              LD SP, 01F0           ; INITIALIZE STACK POINTER.
 74             ;
 75 F204 3059                  JSR INITCD            ; INITIALIZE THE CODEC
 76             FLOOP:
 77 F206 3005                  JSR INPUT             ; GET INPUT SAMPLE, OUTPUT
 78                                                  ; PREVIOUS SAMPLE.
 79 F208 E7                    SHL A
 80 F209 E7                    SHL A
 81 F20A 301F                  JSR OUTPUT            ; CONVERT OUTPUT VALUE TO
 82                                                  ; MU-255 LAW AND SAVE.
 83 F20C 66                    JP FLOOP              ; GO DO NEXT SAMPLE.
 84             ;
 85             ;
 86             INPUT:
 87 F20D B601C088              LD A, YOFK            ; GET DATA TO BE OUTPUT.
 88             NOTDN:
 89 F211 96D210                IF IRPD,0             ; IS MICROWIRE DONE?
 90 F214 41                    JP MWDONE             ; YES, SO GET DATA.
 91 F215 64                    JP NOTDN              ; NO, SO TRY AGAIN.
 92             MWDONE:
 93 F216 BED6                  X A, SIO              ; GET NEW SAMPLE, OUTPUT
 94                                                  ; COMPUTED DATA.
 95 F218 01                    COMP A                ; TAKE CARE OF CODEC INVERSION.
 96 F219 99FF                  AND A, 0FF
 97 F21B E7                    SHL A
 98 F21C BAF000                OR A,0F000            ; FORM MU-LAW TO LINEAR
 99                                                  ; TABLE ADDRESS.
100 F21F AECE                  X A, X
101 F221 D0                    LD A, M(X+)           ; GET LINEAR VALUE
102 F222 AECA                  X A, K
```

```
103 F224 04                 LD A, M(X)          ; A BYTE AT A TIME.
104 F225 BCC8CB             LD H(K), L(A)
105 F228 ABCA               LD A, K
106 F22A 3C                 RET
107             ;
108             ;
109             OUTPUT:
110 F22B 96D41F             RESET IRCD.7
111 F22E E7                 SHL A               ; SIGN BIT TO C.
112 F22F 06                 IFN C               ; IS IT POSITIVE?
113 F230 45                 JP OPOS
114 F231 96D40F             SET IRCD.7
115 F234 01                 COMP A
116 F235 04                 INC A               ; NEGATIVE, SO TAKE 2'S
117                                             ; COMPLEMENT.
118             OPOS:
119 F236 B80108             ADD A, 0108         ; ADD BIAS.
120 F239 9107               LD K, 07            ; SET UP COUNTER.
121             ALIGN:                          ; LOOP AND LOCATE MS 1 BIT.
122 F238 E7                 SHL A
123 F23C 07                 IF C
124 F23D 44                 JP ODONE            ; FOUND MS 1 BIT.
125 F23E AACA               DECSZ K
126 F240 65                 JP ALIGN
127 F241 E7                 SHL A               ; HAS TO BE 1 IN C NOW.
128             ODONE:
129 F242 AECA               X A, K
130 F244 E7                 SHL A
131 F245 E7                 SHL A
132 F246 E7                 SHL A
133 F247 E7                 SHL A               ; COUNTER VALUE IN BITS 4-6.
134 F248 AECC               X A, B
135 F24A 00                 CLR A
136 F24B 88CB               LD A, H(K)
137 F24D 3B                 SWAP A
138 F24E 990F               AND A , 0F
139 F250 96CCFA             OR A, B
140 F253 96D417             IF IRCD.7
141 F256 96C80F             SET A.7
142 F259 01                 COMP A
143 F25A B601C08B           ST A, Y0FK
144 F25E 3C                 RET
145             ;
146             INITCD:
147 F25F B7FFB7F2           LD DIRB, 0FFB7      ; SET B3 (T2IO) AND B6 (SK)
148                                             ; ON PORT B AS INPUTS. SET ALL
149                                             ; OTHER PINS ON B AS OUTPUT.
150 F263 B70000E2           LD PORTB, 0         ; OUTPUT 0 ON ALL PORT B PINS.
151 F267 96F40B             SET BFUNL.3         ; ALT. FUN. ON B3-T2IO.
152 F26A 96F40D             SET BFUNL.5         ; ALT. FUN. ON B5-SO.
153 F26D 96F508             SET BFUNH.0         ; ALT. FUN. ON B8-TSO.
```

```
154 F270 9700D0                 LD ENIR, 0            ; DISABLE INTRPTS.
155 F273 9700D4                 LD IRCD,0             ; SELECT SLAVE MODE FOR M-WIRE.
156 F276 83070188AB             LD T2TIM, 07          ; LOAD 7-DEC INTO T2 TIMER.
157 F27B 83070186AB             LD T2REG, 07          ; LOAD 7-DEC INTO T2 REG.
158 F280 8300018F8B             LD DIVBYH, 0          ; SELECT EXT, CLOCK FOR T2 TIMER.
159             ;
160 F285 8ED6                   X A, SIO
161 F287 8740400190AB           LD TMMD,04040         ; START TIMER T2.
162 F28D 3C                     RET
163             ;
164             ;
165 FFFE 00F2                   .END CODEC
```

SYMBOL TABLE

A	00C8 W	ALIGN	F23B	B	00CC W	BFUN	00F4 W*
BFUNH	00F5 M	BFUNL	00F4 M	CODEC	F200	DIRB	00F2 W
DIRBH	00F3 M*	DIRBL	00F2 M*	DIVBY	018E W*	DIVBYH	018F M
DIVBYL	018E M*	ENIR	00D0 M	FLOOP	F206	INCRM	0200
INITCD	F25F	INPUT	F20D	IRCD	00D4 M	IRPD	00D2 M
K	00CA W	MVAL	205F	MWDONE	F216	NOTDN	F211
ODONE	F242	OPOS	F236	OUTPUT	F22B	PC	00C6 W
PORTB	00E2 W	PORTBH	00E3 M*	PORTBL	00E2 M*	PORTI	00D8 M*
PSW	00C0 M*	RVAL	EDA1	SIO	00D6 M	SP	00C4 W
SVAL	2100	T2REG	D186 W	T2TIM	0188 W	TMMD	0190 W
TMMDH	0191 M*	TMMDL	0190 M*	X	00CE W	YOFK	01C0 M

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

MUTBL

  NO WARNING LINES

  NO ERROR LINES

  656 ROM BYTES USED

SOURCE CHECKSUM = 81D3
OBJECT CHECKSUM = 0C3C

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

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

A       00C8 W        ALIGN   F23B          B       00CC W        BFUN    00F4 W*
BFUNH   00F4 M        BFUNL   00F4 M        CODEC   F200          DIRB    00F2 W
DIRBH   00F3 M*       DIRBL   00F2 M*       DIVBY   018E W*       DIVBYH  018F M
DIVBYL  01B3 M*       ENIR    00D0 M        FLOOP   F206          INCRM   0200
INITCD  F25F          INPUT   F20D          IRCD    00D4 M        IRPD    00D2 M
K       00CA W        MVAL    205F          MWDONE  F216          NOTDN   F211
ODONE   F242          OPOS    F236          OUTPUT  F22B          PC      00C6 W
PORTB   00E2 W        PORTBH  00E3 M*       PORTBL  00E2 M*       PORTI   00D8 M*
PSW     00C0 M*       RVAL    EDA1          SIO     00D6 M        SP      00C4 W
SVAL    2100          T2REG   D186 W        T2TIM   0188 W        TMMD    0190 W
TMMDH   0191 M*       TMMDL   0190 M*       X       00CE W        YOFK    01C0 M
```

Lit. # 100484

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

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