THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 7, September 1981
Printed in U.S.A.

Digital Signal Processor:

Tone Detection for CCITT No. 5 Transceiver

By R. N. GADENZ

(Manuscript received July 17, 1980)

This paper describes the application of a recently developed large-scale-integration digital signal processor, the DSP, *to tone detection in a proposed digital* CCITT *No. 5 signaling unit. The design of the digital filters required in the receiver is discussed and an algorithm presented for tone detection. Two channels per* DSP *can be accommodated by using a sampling frequency of 8 kHz and a* DSP *clock of 5 MHz.*

I. INTRODUCTION

The application of a recently developed LSI digital signal processor, the DSP,[1] to tone detection in a proposed digital CCITT No. 5 signaling unit is described. This unit is to be part of an echo canceler terminal for No. 4 ESS international switching centers. At present, an analog configuration, which includes an analog transceiver, terminates trunks with CCITT No. 5 signaling.

Line signaling information in the CCITT No. 5 signaling system is transmitted via 2400- and 2600-Hz tones used either separately or in combination. The block diagram of Fig. 1 shows the section of the proposed digital receiver to be implemented using the DSP. The band-pass filters (BPFs) detect energy at one of the two signaling frequencies. The band elimination filter (BEF) serves both as a guard filter, to detect energy other than at the signaling frequencies, and as an attenuator of any signaling energy present in the input signal. The detector compares the outputs of the three filters and determines if either, or both, of the tones are present. This information is then used by the time validation circuit that follows. The circuit, called control and output logic in Fig. 1, determines if the tone(s) is(are) present for

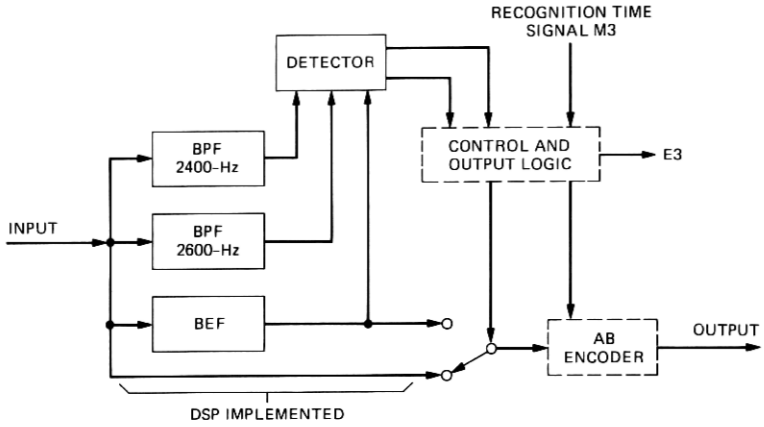

Fig. 1—Receiver.

the prescribed time, as determined by the M3 signal, and also controls the switch that selects either the input or the BEF output as the signal to be sent to the AB encoder and the output. The DSP implements the filtering and detection functions, excluding time validation.

II. FILTER DESIGN

2.1 Bandpass filters

The design of the digital BPFs was based on the performance of the corresponding filters presently used in the analog CCITT No. 5 signaling transceiver. For each filter, the transfer function in the z-domain, $T(z)$, was obtained from the transfer function in the s-domain, $T(s)$, via the bilinear transformation. A sampling frequency of 8 kHz was assumed. The transformation was accomplished using one of the interactive computer programs available, e.g., FILSYN.[2]

Optimization in the z-domain can then be used to correct for the warping effect of the bilinear transformation. A program exists for this purpose,[3] and it usually requires very few iterations to achieve excellent matching between the actual and the desired responses. Figure 2 illustrates the loss response, after optimization, for the 2400-Hz BPF, and Fig. 3, for the 2600-Hz BPF. Figures 2 and 3 also show the specifications, as derived from the analog filters mentioned above. The response of each digital filter is slightly better than that of the corresponding analog design.

Before implementing each transfer function, another step is required—the pairing of the poles and the zeros, and the ordering and scaling of the sections to be connected in cascade. This is done to avoid overflow and to reduce the quantization noise (error) due to rounding

or truncation at the output of the DSP arithmetic unit (AU). In-house programs developed by K. Mina are available to assist in these operations.

The rounding (or truncation) of the transfer function coefficients also introduces a deviation in the frequency response which could be corrected by optimization. However, this was not necessary here,

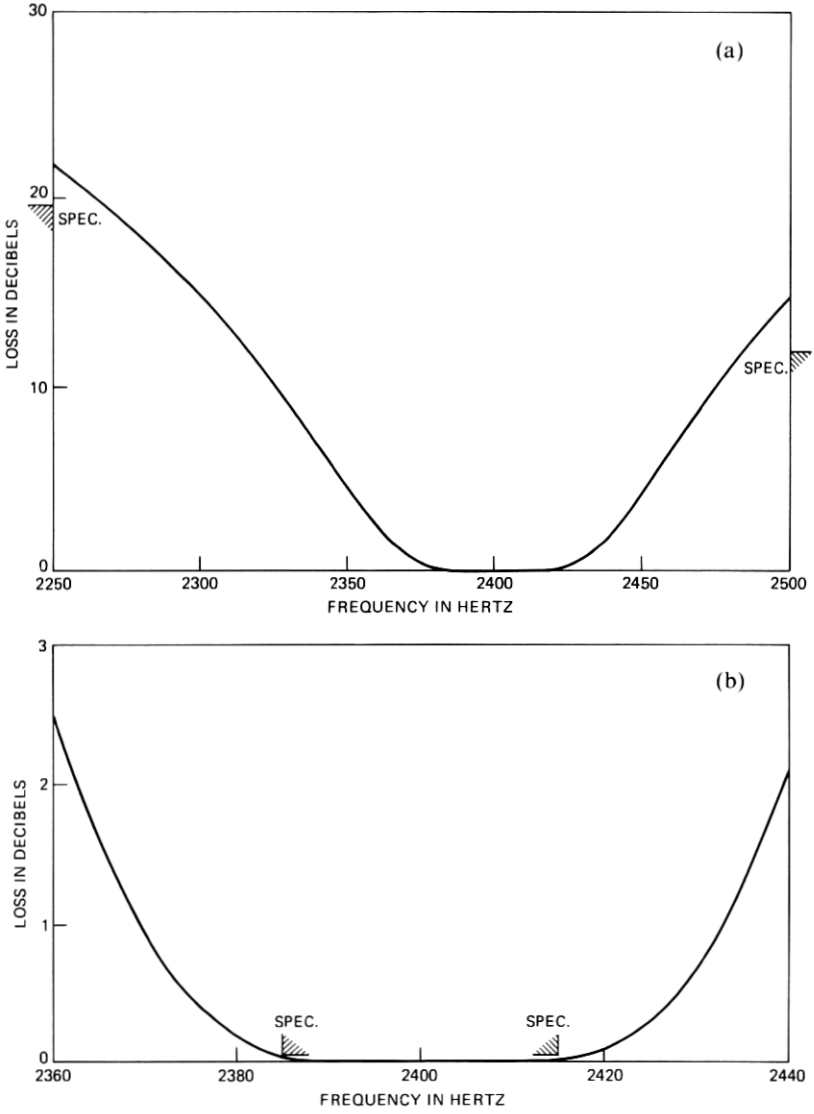

Fig. 2—Bandpass filter—2400 Hz. (a) Loss response. (b) Passband response—expanded scale.

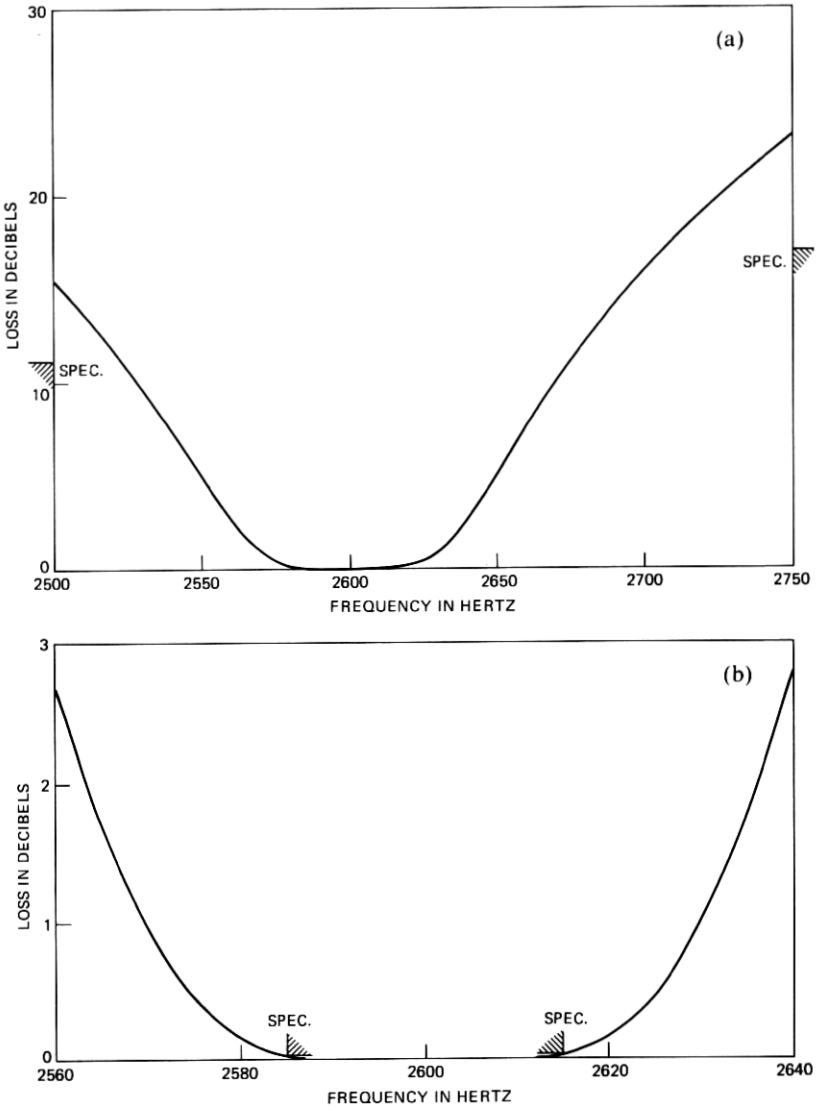

Fig. 3—Bandpass filter—2600 Hz. (a) Loss response. (b) Passband response—expanded scale.

because the DSP allows 16-bit coefficients and, thus, the distortion introduced in the loss response is negligible.

2.2 Band elimination filter

The analog signaling transceiver uses a two-section guard filter to detect energy other than at the signaling frequencies, and a separate

three section BEF to remove tones, if present, from the input signal. In the digital signaling transceiver, only one BEF is to be used to perform both functions, as indicated in Fig. 1. Thus, the design of the digital BEF was not based on either of the two analog BEFs. Rather, a new analog BEF was designed to give the proper loss with only two second-order sections. For the same passband ripple, as in the 3-section BEF mentioned above, the passbands are now slightly reduced, but this is acceptable. The saving of one second-order section is crucial for the implementation of more than one channel per DSP, as discussed later.

The transfer function of this new analog BEF was then transformed into the z-domain and processed in a similar fashion as described before for the BPFs. Note that, in this case, instead of correcting the warping effect by optimization, a prewarped analog design could be used. The two approaches were in fact compared with very similar results. The optimization program in Ref. 3 is very efficient and, thus, attractive. The loss response for the digital BEF is shown in Fig. 4.

III. DETECTOR DESIGN

To perform the detection function, the outputs of the three filters are first rectified and then smoothed with a low-pass filter (LPF) (see Fig. 5). The signals P1, P2, and G then go to a threshold detector where the presence or absence of signaling tones is established.

One first-order section is used for each LPF. The corresponding loss and step responses are illustrated in Figs. 6 and 7, respectively.

The presence of tones is determined by the following criteria:

(*i*) $f_1 = 2400$ Hz is present if

$$t_l \le P_1 \le t_h \quad \text{and} \quad \frac{P_1}{G} > t_G$$

(*ii*) $f_2 = 2600$ Hz is present if

$$t_l \le P_2 \le t_h \quad \text{and} \quad \frac{P_2}{G} > t_G,$$

where the limits t_l, t_h, and t_G are determined from the CCITT recommendations for the receiver performance. The upper limit t_h should be 5 dB below the maximum amplitude of the tone (3 dBm0), and the lower limit t_l should be 19 dB below the maximum amplitude. Then, considering that P_1, P_2, and G are approximately the average values of the respective signals, and that for a tone, the average value is $2/\pi$ times the amplitude, we get:

$$t_h = 0.3579976064 \cdot A$$

$$t_l = 0.0714299132 \cdot A,$$

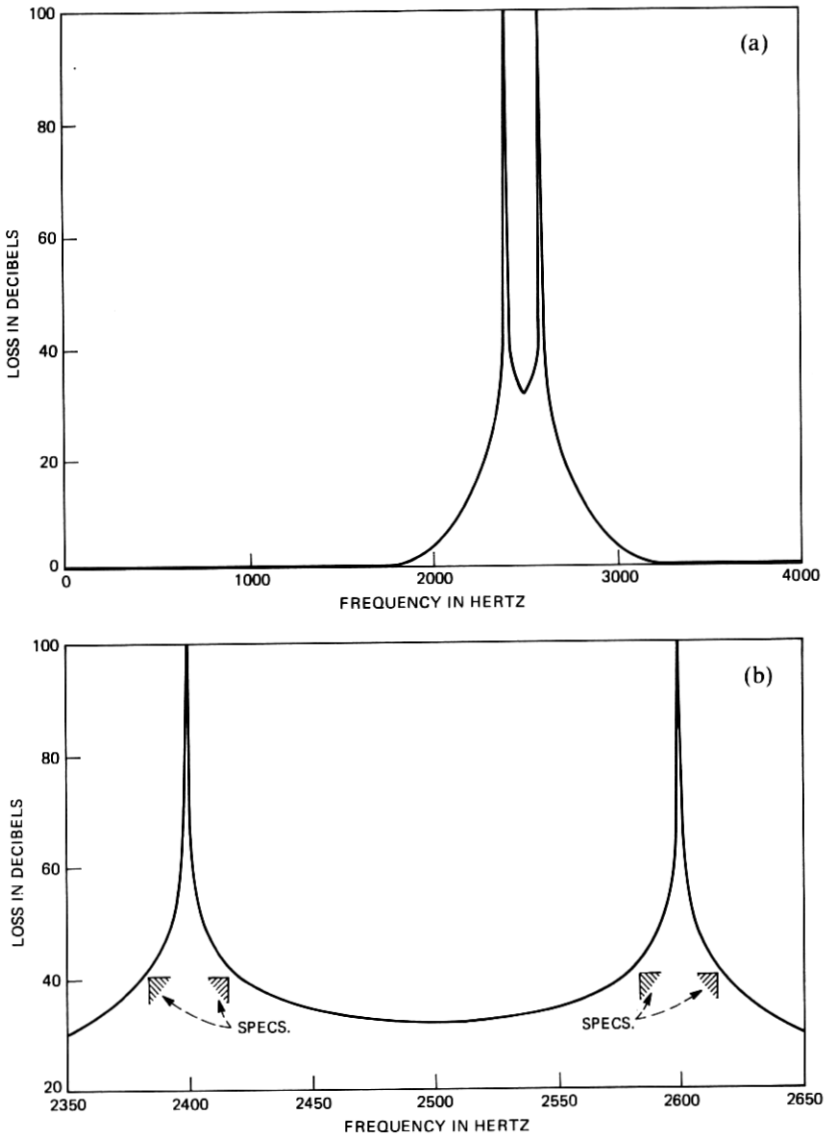

Fig. 4—Band elimination filter. (a) Loss response. (b) Stopband response—expanded scale.

where A is the maximum amplitude of either tone. The value of T_G is 5.

IV. THE DSP ALGORITHM

The filtering and rectification functions are easily implemented

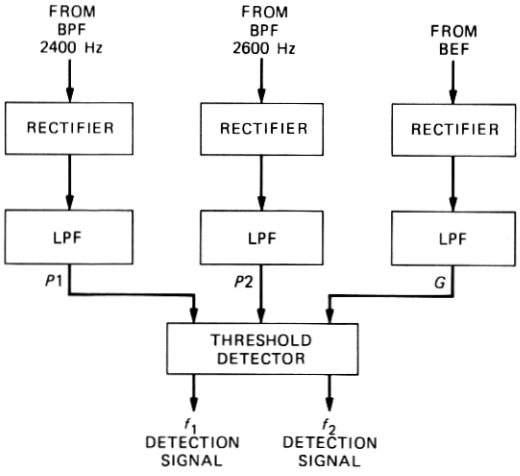

Fig. 5—Details of detector function.

using the DSP.[1] The order in which these operations are executed, as well as the way in which the threshold detector is implemented, are described below.

The S0 and S1 pins of the DSP are used to output the information on the presence of tones, thus reserving the normal (serial) DSP output for

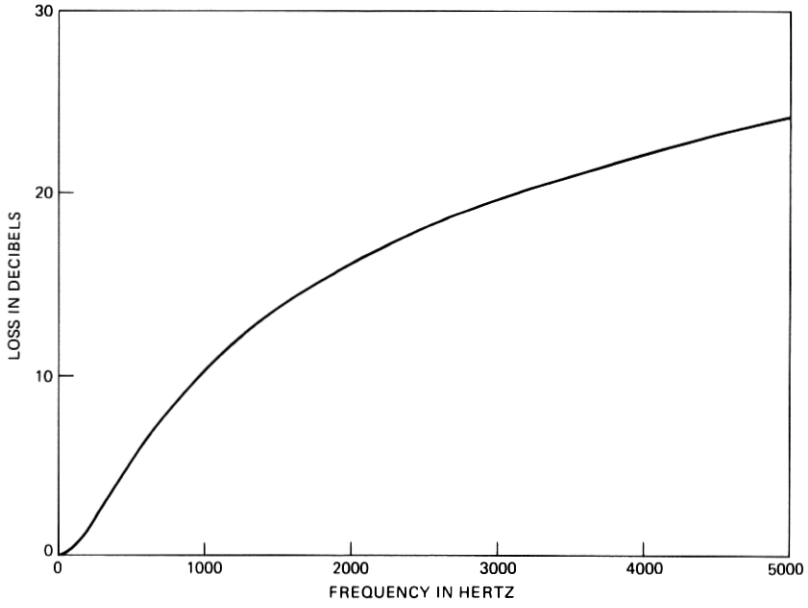

Fig. 6—Low-pass filter—loss response.

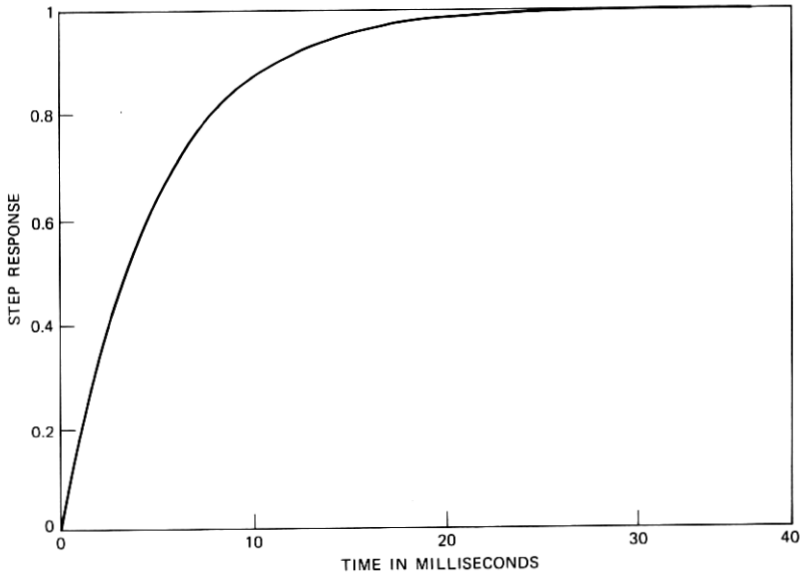

Fig. 7—Low-pass filter—step response.

the output signal of the BEF. The setting of S pins is determined as follows: If any condition required for tone f_1 or f_2 to be present is violated, a negative number is produced whose sign causes the corresponding pin (s0 or s1, respectively) to be set to zero. A logical one on either pin indicates that the corresponding tone is present, i.e., has been detected.

An alternative approach could be based on the following: If any condition required for tones f_1 or f_2 to be present is violated, a negative number is stored in memory; otherwise, a positive number is stored. The signs of these stored numbers could then be used to construct a 2-bit output word in which each bit indicates whether the corresponding tone is present or not. (Actually an 8-bit word would have to be output, but the remaining 6-bits would be irrelevant.) This approach would use the normal DSP output for both the tone information and the output of the BEF, which was not desirable in this application. However, a few instructions would be saved, thus, freeing up some processing time.

In more detail, the algorithm used here is as follows (see Fig. 8):

(a) Set the AU and I/O control registers, clear the RAM, and initialize the memory pointers.

(b) Read a μ-law PCM sample from the input buffer, convert it to linear format, and save the result s in RAM location 127.

(c) Process the linear sample s through the BEF and save the result r in the output register of the DSP AU.

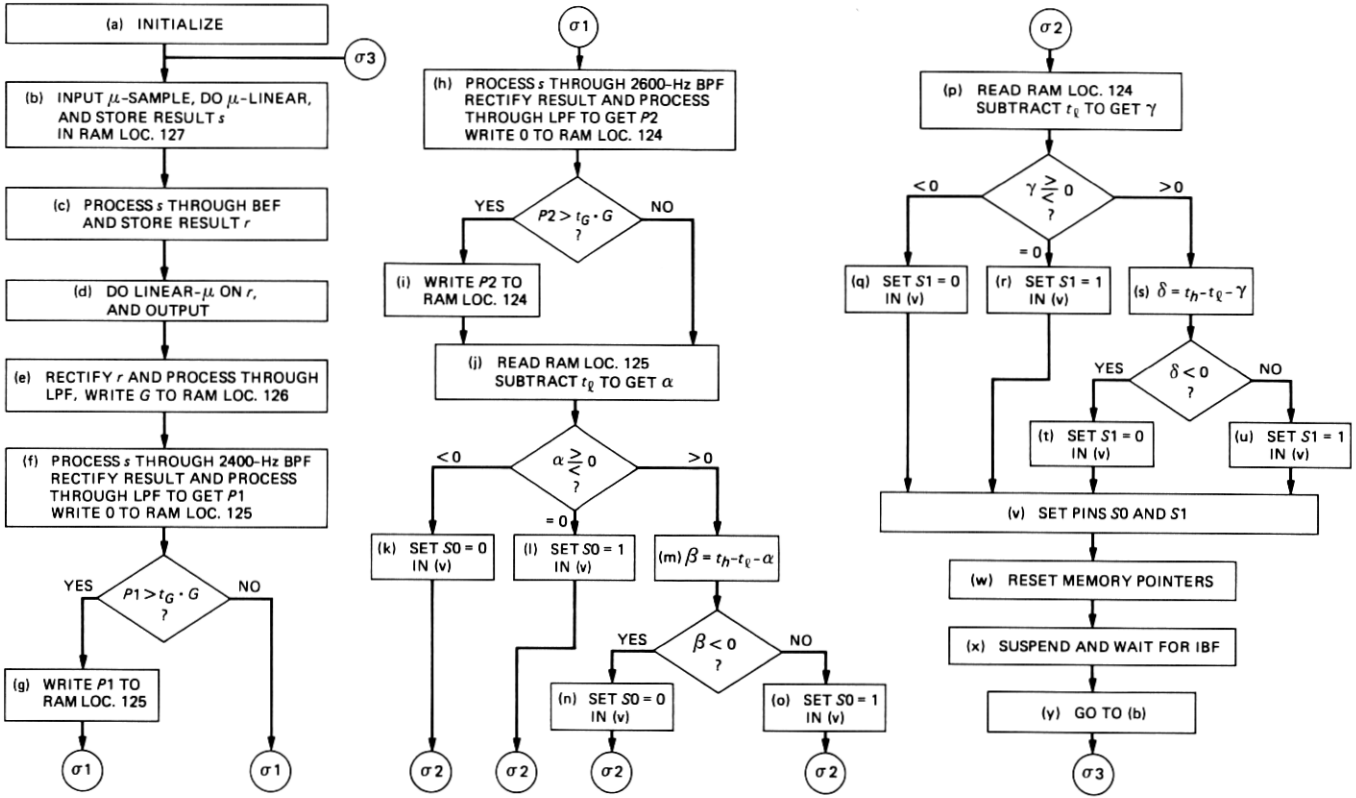

Fig. 8—Flow chart for the DSP algorithm.

(d) Convert the output r of the BEF to μ-law format and output it.

(e) Rectify the output r of the BEF and process it through the LPF. Save the result G in RAM location 126.

(f) Process the linear sample s through the 2400-Hz BPF, rectify the result and process it through the LPF. The output is P_1. Write zero to RAM location 125.

(g) Compare P_1 with $t_G \cdot G$. If $P_1 > t_G \cdot G$, write P_1 into RAM location 125. Otherwise, a zero remains in that RAM location, which was written during the execution of step (f).

(h) Process the linear sample s through the 2600-Hz BPF, rectify the result and process it through the LPF. The output is P_2. Write zero to RAM location 124.

(i) Compare P_2 with $t_G \cdot G$. If $P_2 > t_G \cdot G$, write P_2 into RAM location 124. Otherwise, a zero remains in that RAM location, which was written during the execution of (h).

(j) Read RAM location 125. If the write operation in (g) did not occur because $P_1 \leq t_G \cdot G$, zero is obtained; otherwise, P_1 is read. Subtract t_l to get α.

(k) If the result α in (j) is negative, no f_1 tone is present and s0 will be set to zero in (v) at the end of the program; continue with (p).

(l) If the result α in (j) is zero, $P_1 = t_l$; for reasons of compactness in the DSP code, $t_h - t_l$ is added and subtracted; the result is still zero and its sign is also zero, as for a positive number; this causes s0 to be set to one in (v) at the end of the program, indicating that the f_1 tone is present; continue with (p).

(m) If the result α in (j) is positive, subtract it from $t_h - t_l$ to get β.

(n) If the result β obtained in (m) is negative, $P_1 > t_h$ and no f_1 tone is present; s0 will be set to zero in (v) at the end of the program; continue with (p).

(o) If the result β obtained in (m) is non-negative, $P_1 \leq t_h$ and the f_1 tone is present; s0 will be set to one in (v) at the end of the program; continue with (p).

(p) Read RAM location 124. If the write operation in (i) did not occur because $P_2 \leq t_G \cdot G$, zero is obtained; otherwise, $P2$ is read. Subtract t_l to get γ.

(q) If the result γ in (p) is negative, no f_2 tone is present and s1 will set to zero in (v); continue with step (v).

(r) If the result γ in (p) is zero, $P_2 = t_l$; for reasons of compactness in the DSP code, $t_h - t_l$ is added and subtracted; the result is still zero and its sign is also zero, as for a positive number; this causes s1 to be set to one in (v), indicating that the f_2 tone is present; continue with (v).

(s) If the result γ in (p) is positive, subtract it from $t_h - t_l$ to obtain δ.

(t) If the result δ obtained in (s) is negative, $P_2 > t_h$ and no f_2 tone is present; s1 will be set to zero in (v); continue with step (v).
(u) If the result δ obtained in (s) is non-negative, $P_2 \leq t_h$ and the f_2 tone is present; s1 will be set to one in (v).
(v) Set the s0 and s1 pins to the proper values.
(w) Reset the memory pointers.
(x) Suspend the DSP operation and wait for the input buffer to be filled with another sample. Execution resumes when the IBF flag goes high.
(y) Loop back to step (b).

The RAM is organized as follows:

Location		Contents
0	y_1	State variables for BEF
1	x_1	
2	y_2	
3	x_2	
4	x	State variable for LPF following the BEF
5	y_1	State variables for 2400-Hz BPF
6	x_1	
7	y_2	
8	x_2	
9	x	State variable for LPF following the 2400-Hz BPF
10	y_1	State variables for 2600-Hz BPF
11	x_1	
12	y_2	
13	x_2	
14	x	State variable for LPF following the 2600-Hz BPF
...	...	
124	P_2	See Fig. 5
125	P_1	
126	G	
127	s	Input sample in linear format.

Note that, for one channel, only the first 15 and the last four RAM locations are used.

This algorithm can be easily translated into the corresponding DSP code, which is then assembled[4] and stored in the DSP ROM. The filter coefficients are also stored in ROM, in line with the code. In the program, the quantities $-(\frac{1}{8})t_G$, $-t_l/A$, and $(t_h - t_l)/A$ are used instead of the quantities t_G, t_l, and t_h, because they are more convenient.

The loop in the program has 77 instructions, independent of the path followed. Then, the code for two channels will have a loop of 154 instructions. With a 5-MHz clock, the instruction cycle is 800 ns. This implies that for a sampling frequency of 8 kHz, 156 instructions can be

accommodated in a period (125 μs) between samples. Therefore, with the algorithm given here, two channels can be implemented per DSP, and 12 DSPs are needed per digroup of 24 channels.

The program has been tested both in software (using the DSP simulator[5]) and in hardware (using the DSP device with external ROM[6]), and was found to perform as expected.

V. CONCLUSION

Use of the DSP in implementing filtering and tone detection functions in the receiver of a proposed digital CCITT No. 5 signaling unit has been shown. Characteristics of the required digital filters have been described, along with a procedure for designing the filters. A way to realize the detector has been illustrated and the criteria used to determine the presence of tones have been presented. Finally, an algorithm has been given which, for a sampling frequency of 8 kHz and a DSP clock of 5 MHz, allows the implementation of two channels per DSP.

VI. ACKNOWLEDGMENTS

The author is grateful to D. M. Brady, A. M. Gupta, K. Mina, D. C. Stanzione, H. C. Kirsch, and J. R. Boddie for their support and suggestions.

REFERENCES

1. J. R. Boddie et al., "Digital Signal Processor: Architecture and Performance," B.S.T.J., this issue.
2. G. Szentirmai, "FILSYN—A General Purpose Filter Synthesis Program," Proc. of the IEEE, *65*, No. 10 (October 1977), pp. 1443–58.
3. M. T. Dolan and J. F. Kaiser, "An Optimization Program for the Design of Digital Filter Transfer Functions," in *Programs for Digital Signal Processing*, IEEE Press Book, 1979.
4. C. L. Semmelman, "Digital Signal Processor: Design of the Assembler," B.S.T.J., this issue.
5. J. Aagesen, "Digital Signal Processor: "Software Simulator," B.S.T.J., this issue.
6. J. R. Boddie, "Digital Signal Processor: Overview: The Device, Support Facilities, and Applications," B.S.T.J., this issue.