

A Low-Complexity ITU-Compliant Dual Tone Multiple Frequency Detector

Amey A. Deosthali, *Member, IEEE*,
Shawn R. McCaslin, *Member, IEEE*, and
Brian L. Evans, *Senior Member, IEEE*

A. A. Deosthali (deosthali@sugar-land.spc.slb.com) performed this research while at The University of Texas in Austin, TX, and is now with Schlumberger Technology Corporation, 110 Schlumberger Dr., MD4, Sugar Land, TX 77478. S. R. McCaslin (srm@cicada-semi.com) performed this research while at Crystal Semiconductor in Austin, TX, and is now with Cicada Semiconductor, Barton Oaks Plaza One, 901 South Mopac Expressway, Austin, TX 78746. Prof. B. L. Evans (bevans@ece.utexas.edu) is at ENS Bldg., Dept. of Electrical and Computer Eng., The University of Texas, Austin, TX 78712-1084, <http://signal.ece.utexas.edu/>.

This work was supported by the US National Science Foundation CAREER Award under Grant MIP-9702707. Touchtone is a registered trademark of AT&T.

Abstract

We present the first dual tone multiple frequency (DTMF) signal detector that meets the International Telecommunications Union (ITU) Q.24 DTMF standard when implemented on an 8-bit microcontroller. Key innovations include the use of adaptive notch filters and sophisticated decision logic. The DTMF detector is well suited for a multichannel digital signal processor implementation.

Index terms– touchtone decoding, DTMF detection, signal processing on microcontrollers

I. INTRODUCTION

Dual Tone Multiple Frequency (DTMF) signaling is used in telephone dialing, digital answer machines, and interactive banking systems. DTMF signaling represents each symbol on a telephone touchtone keypad (0-9,*,#) using two sinusoidal tones, as shown in Fig. 1. When a key is pressed, a DTMF signal consisting of a row frequency tone plus a column frequency tone is transmitted. Keys A-D are not on commercial telephone sets, but are used in military and radio signaling applications. The maximum dialing rate is 10 symbols/s in the Bellcore standard [1], [2] and 12.5 symbols/s in the International Telecommunication Union (ITU) Q.24 standard [3].

The ITU DTMF specifications, which are given in Table I, are more strict than the Bellcore specifications. So, ITU-compliance implies Bellcore-compliance, but not vice-versa. ITU specifications require that valid DTMF signals have their high and low frequency tones within a tolerance of $\pm 1.5\%$ of an ideal DTMF frequency. If the tolerance of either tone is outside $\pm 3.5\%$, then the signal should be rejected as invalid. ITU specifications place requirements on the duration of and pauses between valid DTMF signals. ITU specifications require 100% detection of valid DTMF signals at 15 dB SNR. Bellcore provides test tapes to measure the performance of a DTMF detector against *talk-off*, which is false detection of speech signals as DTMF signals.

The ITU-compliant detector in [4] and Bellcore-compliant detectors in [1], [5] are based on the Discrete Fourier Transform (DFT). They use 16-bit data, require 15–17 16-bit multiplications per input sample, and do not buffer input data. For n channels, the ITU-compliant detector in [4] requires n MIPS, $120 + 30n$ words of data memory, and 1000 words of program memory.

We present the first DTMF detector that meets the ITU specifications and passes the Bellcore talk-off tests when implemented on an 8-bit microcontroller. The detector uses two constrained adaptive notch filters and two frequency estimators for each channel, and does not require buffering of input data. The detector provides an optional dial tone suppressor for central office appli-

cations. For each input sample, the detector requires 3.5 8-bit multiplications. For single channel detection, we implemented the detector on a 5-MIP 8-bit Microchip PIC16C711 microcontroller using 64 bytes of data memory and 800 bytes of program memory. For simultaneously detecting n channels, we estimate that the implementation on a digital signal processor (DSP) would require $0.7n$ MIPS, $19 + 45n$ words of data memory, and 1000 words of program memory. So, a 30-MIP 16-bit Texas Instruments TMS320C54 DSP with 16 kwords of RAM and 4k words of ROM could decode 42 time-division multiplexed channels, which is plenty for an E1 line (32 channels) or T1 line (24 channels).

II. PREVIOUS DTMF DECODERS

Decoding DTMF signals amounts to detecting two sinusoids in noise. DTMF detectors typically consist of a signal analysis front end followed by a decision logic back end. The back end enforces compliance with a standard. Integrated circuit detectors may mix a front end consisting of eight bandpass resonators implemented in switched capacitor technology followed by a digital post-processor to measure tone duration and provide correctly coded digital outputs [6].

Popular DFT-based digital methods for DTMF detection also use eight resonating bandpass IIR filters [1], [5]. For DFT of length M , the M th output of each digital filter is the DFT coefficient computed using the Goertzel algorithm [7]. The Goertzel algorithm uses 1 word of ROM and 2 words of RAM, and requires $M + 2$ multiplications and $2M + 2$ additions for every M samples [1]. The absolute value of the filter output is squared to measure the signal energy present at the DTMF frequency. A DTMF tone pair is considered valid if enough energy exists in the DTMF frequencies and if that energy exceeds the other DTMF tone pair energies [1].

Increasing the DFT length increases frequency resolution but decreases timing resolution. No single DFT length, however, can simultaneously meet the timing and frequency specifications in the ITU DTMF standard [4], [5]. For the Bellcore DTMF standard, many single DFT lengths work. A DFT length that simplifies the detection logic is the minimum DTMF signal duration divided by 3 [1]. A DTMF symbol is valid if it is valid across two or more consecutive windows.

The non-uniform DFT (NDFT) may be used to calculate signal energy at exact DTMF frequencies. A direct application of the NDFT does not meet the ITU frequency tolerance and timing requirements [8]. By using two NDFTs of different lengths, another approach meets all ITU recommendations at an implementation cost of 15 multiplications/sample [4].

Three detectors based on adaptive linear filtering [5], [6], [9] are not ITU-compliant. Two

approaches split the signal into low and high frequency paths. In the first, two adaptive IIR filters estimate the frequency and amplitude parameters of the two tones [6]. The second adaptively estimates the roots of a second-order transfer function to estimate the frequency content in each band [9]. The third approach uses normalized direct adaptive frequency estimation [5].

Subspace decomposition techniques such as Multiple Signal Classification (MUSIC) have been applied to DTMF detection [5]. Although subspace techniques could meet ITU specifications, they are computationally and memory intensive, generally require floating-point arithmetic, and are difficult to implement for real-time performance on embedded processors.

III. PROPOSED METHOD

Fig. 2 shows a block diagram of our DTMF detector. We assume that the input is sampled at 8 kHz, which we decimate by two using a two-tap moving average anti-aliasing filter. Decimation by two reduces the noise variance by $\frac{1}{\sqrt{2}}$ which increases the effective dynamic range. After decimation, the signal processing front end estimates the dominant DTMF frequencies in the input signal. The decision logic back end imposes ITU specifications.

A. Signal Processing Front End

The ITU specifications require detection at an SNR of 15 dB and a signal power of -26 dBm (-1 dBm is 1 dB of attenuation for 1 mW of power). Since 8 bits provides roughly 48 dB SNR, only a small margin of 7 dB SNR remains. We increase the effective dynamic range by employing automatic gain control (AGC). The AGC stage boosts the input signal level by a factor of 4 (12 dB) if it falls below -24 dB and by a factor of 2 (6 dB) if it falls below -18 dB.

A notch filter has a perfect null at a desired frequency. For central office applications, the input signal passes through a 300 Hz notch filter to suppress dial tone interference from the 350 Hz and 440 Hz dial tone frequencies. A dial tone suppressor is common in DTMF decoder ICs [6] but not required by DTMF standards. After the optional 300 Hz dial tone suppressor, the signal passes through the low frequency, high frequency, and power estimator data paths. In the low and high frequency data paths, the adaptive notch filters remove the high and low frequency tones, respectively. Once these filters converge, the fully adaptive filter in the low (high) frequency data path provides a low (high) frequency signal.

For the fully adaptive notch filters, we use second-order finite impulse response (FIR) adaptive notch filters. A notch filter has a complex conjugate zero pair on the unit circle at $z = e^{\pm j\omega_0}$

where ω_0 is the notch frequency. With b_0 being a scale factor that prevents overflow,

$$H(z) = b_0 (1 - 2 \cos \omega_0 z^{-1} + z^{-2}) \quad (1)$$

is the transfer function [7]. We choose $b_0 = \frac{1}{2}$. An FIR notch filter has a relatively large stopband width centered at ω_0 , so it attenuates other frequency components near the null. We use this property to suppress the high (low) frequency interference in the low (high) frequency data path. The adaptive notch filters put a notch at the frequency estimated by the frequency estimator in the opposite data path. In order to increase their convergence rate, we restrict the adaptive notch filters to the eight $\pm 3.5\%$ tolerance bands around the eight DTMF frequencies.

The frequency estimators use a highly accurate algorithm [10] based on zero crossings. The period of the incoming sinusoid can be estimated from the number of samples between two zero crossings. The spectrum of the estimated period contains a DC component whose amplitude is equal to the actual period of the signal. Our detector uses the number of samples between two zero crossings directly in all of the computations. To obtain the DC component, we lowpass filter the zero crossings estimate. We decrease the order of the lowpass filter by implementing it in two stages. The first stage is an *averaging by two* filter, implemented implicitly in the frequency estimation algorithm. The second stage uses a first-order IIR lowpass filter

$$\hat{F}(n) = \alpha \hat{F}(n-1) + (1-\alpha) \hat{F}(n) \quad (2)$$

where $\hat{F}(n)$ is the zero crossings estimate at index n . We choose $\alpha = \frac{7}{8} = 0.875$.

The fully adaptive notch filters introduce DC bias which interferes with the tracking of zero crossings by the frequency estimators. We use DC notch filters to remove the bias. The DC notch filter in the low frequency data path has high gain in the low frequency DTMF spectrum (below 1000 Hz) and extremely low gain in the high frequency portion (above 1000 Hz). The converse is true for the DC notch filter in the high frequency data path. The unequal gain emphasizes the dominant tone in each data path. The DC notch filter coefficients are $\{0.25, 0.25, 0, -0.25, -0.25\}$ in the low frequency data path and $\{0.25, -0.25, 0, 0.25, -0.25\}$ in the high frequency data path.

Fig. 3 shows the frequency response of the low frequency data path after a '1' DTMF symbol (697 Hz and 1209 Hz tones) was input and the frequency estimates had stabilized. Notches in the low frequency path occur at 0 Hz, 300 Hz, 1209 Hz, and 1336 Hz.

In the power estimator and signal detector stage, we use a novel technique for detecting the presence of a signal component. The power estimator tracks the power of the signal component.

Since the DTMF signal occurs in bursts, the signal power can be tracked as a short-term average

$$P(n) = \beta P(n-1) + (1-\beta) |s(n)| \quad (3)$$

where $P(n)$ is the current power and $s(n)$ is the current signal sample. The closer β is to 1, the faster the power estimator will track the signal power. We choose $\beta = 0.9$.

The signal power is compared with the noise floor to detect the presence of a signal component in noise, as shown in Fig. 4. Whenever the signal power is less than the noise floor, the noise floor is decreased exponentially with respect to time. However, the noise floor is increased exponentially whenever the signal power is greater than the noise floor. The *detect* signal is true if the signal power is greater than the noise floor, and false otherwise. Since we adaptively set the noise floor, the power estimator robustly detects the presence of a signal component in noise.

B. Decision Logic Back End

Fig. 5 shows the decision logic, which runs at 2 kHz, that enforces ITU specifications. To check timing specifications, we use three counters. The timer count TC is used to determine if a tone interruption lasts for more than 10 ms. It is initialized to 20 samples (10 ms at 2 kHz). The signal timer ST is a cyclic upcounter from 0 to 10 samples (5 ms at 2 kHz). It is incremented by one whenever the last two detected DTMF symbols are the same and reset otherwise. The signal duration timer CT is the total number of samples for which the DTMF symbol has been present. Timing estimates are accurate to 0.5 ms at a sampling rate of 2 kHz.

The decision logic accepts the low frequency estimate (LF), the high frequency estimate (HF), and the detect signal (DT) as input, as shown in Fig. 2. If LF and HF are within the DTMF frequency tolerance requirements, then the new symbol NS is encoded by an integer between 1 and 16 inclusive; otherwise, NS is set to an invalid index (e.g. 34). If $NS \in [1, 16]$ and DT is true, then the timer count (TC) is reinitialized, and the previous symbol (PS) is compared with NS . If PS does not equal NS , then the signal duration count CT is incremented by ST and ST is reset to 0. If PS equals NS and ST is less than 10 samples, then ST is incremented by one. If PS equals NS and ST equals 10 samples, then CT is updated and ST is reset to 0. The first time that ST equals 10 samples means that the frequency estimates for NS are considered stabilized.

When a pause after a DTMF burst exceeds 20 samples (10 ms at 2 kHz), the decision logic checks (CT) against 46 samples (23 ms at 2 kHz). A valid detect is signaled if the DTMF symbol meets the ITU requirements (see Table I). Before exiting, the decision logic assigns NS to PS .

IV. DESIGN AND IMPLEMENTATION

The zero crossings estimate $\hat{F}(n)$ in (2) is stored in two words, as an integer part and fractional part, respectively. The DTMF detector directly uses $\hat{F}(n)$ in computations so as to avoid having to convert it to a period or frequency value. All other quantities are stored in one word.

Each of the two frequency estimators requires a division, in which the denominator is always greater than or equal to the numerator. The numerator value is on $[0, 0.5]$ and the denominator value is on $[0, 1]$, or $[0, 127]$ and $[0, 255]$ in unsigned 8-bit format, respectively. We implement division as a series of shifts and subtractions. After every shift, if the numerator is greater than the denominator, then a bit in the quotient is set, starting at the most significant bit. Division stops if the numerator equals the denominator or if the numerator has been shifted left 8 times.

We restrict the notch frequency in the adaptive notch filters to the $\pm 3.5\%$ tolerance bands around the 8 nominal DTMF frequencies in Fig. 1 to improve the convergence rate. Each adaptive notch filter requires a cosine computation, as shown in (1). Since the cosine function in the 8 tolerance bands is monotonically decreasing, we approximate the cosine using linear interpolation based on the value of $\hat{F}(n)$. For each band, we precalculate the limits that $\hat{F}(n)$ can take, e.g. $[5.445, 5.947]$ for the tolerance band around 697 Hz where 5.947 corresponds to the lower end and 5.445 corresponds to the higher end of the band. Then, we compare the current zero crossings estimate to the eight acceptable ranges. If the estimate falls inside one of the eight tolerance bands, then we subtract the fractional part of the lower end of band (e.g. 0.947 for the 697 Hz band) from the fractional part of the current estimate. We multiply this difference by the slope of the cosine function for the particular tolerance band, and add it to the intercept which is the cosine value corresponding to the lower end of tolerance band (e.g. 5.947 for the 697 Hz band). For all 8 frequencies except 1336 Hz, the integer part of the acceptable range remains the same.

Each of the two divisions requires 8 shifts and 8 additions in the worst case. Each of the two adaptive notch filters requires 1 multiplication, 3 additions, and 4 shifts, plus 1 multiplication and 1 addition for the linear interpolation to compute the cosine value. Each of the two lowpass filters in (2) requires 18 shifts and 6 additions because $\hat{F}(n)$ is represented as two words. The power estimator in (3) uses 2 multiplications, 8 shifts and 15 additions. Each of the two DC notch filters requires 3 additions and 3 shifts. The decimator requires 1 addition and 2 shifts. The 300-Hz notch filter requires 1 multiplication, 2 additions, and 4 shifts.

The detector requires 3.5 multiplications, 30 additions, and 40 shifts per input sample. Single-

channel DTMF detection requires 500 instructions/sample (4 MIPS), 64 bytes of data memory, and 800 words of program memory. The Microchip Technology PIC16C711 microcontroller has 68 bytes of data memory and 1000 words of program memory. At an 8-kHz sampling rate, a 5-MIP (20-MHz) PIC16C711 microcontroller could execute 625 instructions/sample. On the PIC16C711, a carry bit must be emulated in software. Hence, the two-byte addition in (2) requires 6 cycles. Digikey sells the PIC16C711 for \$2.38 each in volume of 100 units.

For a DSP processor implementation, we assume 16-bit fixed-point data words, single cycle multiply-accumulate (MAC), and simultaneous instruction execution and data access. The detector requires 3.5 MACs and 30 additions per sample per channel. For n channels, the detector requires $0.7n$ MIPS for a sampling rate of 8 kHz, $34 + 30n$ words of data memory, and 1000 words of program memory. With a 30-MIP (60-MHz) TMS320C54 DSP, we could decode a maximum of 42 channels, which is plenty for decoding either an E1 line (32 voice channels) or a T1 line (24 voice channels). Texas Instruments sells the C5402-30 for \$5.00 each in volume of 100 units.

V. EXPERIMENTAL RESULTS

We validated that our detector meets the ITU recommendations in Table I using 8-bit arithmetic on 8-bit data. All tests were performed in the presence of additive white Gaussian noise. Unless otherwise stated, the SNR was 13 dB. We did not bandlimit the noise.

An ITU-compliant detector detects all symbols having a frequency tolerance of $\pm 1.5\%$ or less and rejects symbols with a frequency tolerance of $\pm 3.5\%$ or greater. To test the frequency tolerance for each DTMF symbol, we held the high frequency tolerance at -4% , varied the low frequency tolerance from -4% to $+4\%$ in steps of 0.5% , and tested the detector for each combination. Next, the high frequency tolerance was increased by 0.5% to -3.5% , and the process was repeated. We continued until the high frequency tolerance was $+4\%$.

Guard time is the minimum tone length that the detector accepts as valid, e.g. 40 ms or less for an ITU-compliant detector. Our guard time is 36 ms. Table II shows the results of the other four timing tests using the DTMF symbol '1'. For each DTMF symbol, the Bellcore decode test determines the minimum length for which the detector has 100% detection. We passed 10 pulses of each DTMF symbol to the detector, starting with 50 ms ON time and 50 ms OFF time. We decreased the ON time by 1 ms until 100% detection was no longer observed. Table III shows the minimum tone length for each symbol, which varies between 32 ms and 36 ms, inclusive.

Twist is the difference in dB between the amplitudes of the two DTMF tones. Since the

telephone channel is lowpass, the high frequency tone is usually received at a lower amplitude than the low frequency tone, which is called *normal twist*. *Reverse twist* occurs when the low frequency tone is weaker than the high frequency tone. All DTMF symbols were tested by varying the twist over the ITU recommended range of -4 dB to 8 dB in steps of 0.1 dB (-4 dB means 4 dB of reverse twist). The detector shows 100% detection for all DTMF symbols based on 10 pulses per test. For $100,000$ '1' symbols, the detector's miss rate was less than 10^{-4} .

The Bellcore power level test measures the dynamic range and sensitivity of the DTMF detector. The test starts with the DTMF tone level at 0 dBm (dBm is defined as $10 \log_{10} \frac{\text{signal power}}{10^{-3} \text{W}}$). We decrease the DTMF power level from 0 dBm to -36 dBm in steps of 1 dBm. Our detector's sensitivity, which is the lowest power level for 100% detection, is -26 dBm which meets the ITU specifications. Our detector also shows 100% detection.

Talk-off is the false detection of DTMF symbols when speech is input into the detector. Bellcore provides audio test tapes [2] containing 1 million calls to a central office, including $50,000$ speech segments to test the DTMF detector performance against talk-off. Our talk-off performance meets the maximum allowable false detections specified by Bellcore, as shown in Table IV.

The results in this section have been for 8 -bit wrap-around arithmetic on 8 -bit data. For an implementation of the detector under 16 -bit saturation arithmetic as found on a digital signal processor, we expect improvement for talk-off and dynamic range. For example, automatic gain control would no longer be needed.

VI. CONCLUSION

We present a new, low-complexity DTMF detector that meets the ITU Q.24 specifications under 8 -bit wrap-around arithmetic on 8 -bit sampled data. To meet ITU requirements, we use two frequency estimators running concurrently in combination with adaptive notch filters and DC notch filters in the signal analysis front end. In the decision logic back end, we add high-resolution timing and sophisticated decision logic. The detector is the first reported ITU-compliant detector that can be implemented on an 8 -bit microcontroller. It is also amenable to multi-channel DTMF detection on a digital signal processor and on single-instruction multiple-data processing found on to modern general-purpose processors, such as Intel's Pentium MMX extensions. We have released a Matlab version of the detector at <ftp://pepperoni.ece.utexas.edu/pub/dtmf/mat8bit.zip>. For future research, reliable detection of shorter tones than ITU requires may be possible by increasing the number of estimators— to four, or even eight.

REFERENCES

- [1] P. Mock, "Add DTMF generation and decoding to DSP- μ p designs," *EDN*, vol. 30, pp. 205–220, Mar. 1985.
- [2] "Digit simulation test tape," Tech. Rep. TR-TSY-000763, Bell Communications Research, July 1987.
- [3] ITU Blue Book, *Recommendation Q.24: Multi-Frequency Push-Button Signal Reception*. 1989.
- [4] M. D. Felder, J. C. Mason, and B. L. Evans, "Efficient dual-tone multi-frequency detection using the non-uniform discrete Fourier transform," *IEEE Signal Processing Letters*, vol. 5, pp. 160–163, July 1998.
- [5] G. Arslan, B. L. Evans, F. A. Sakarya, and J. L. Pino, "Performance evaluation and real-time implementation of subspace, adaptive, and DFT algorithms for multi-tone detection," in *Proc. IEEE Int. Conf. Telecommunications*, (Istanbul, Turkey), pp. 884–887, Apr. 1996.
- [6] S. Park and D. M. Funderburk, "DTMF detection having sample rate decimation and adaptive tone detection." United States Patent, Feb. 1995. Patent Number: 5,392,348.
- [7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- [8] S. Bagchi and S. K. Mitra, "An efficient algorithm for DTMF decoding using the subband NDFT," in *Proc. IEEE Int. Sym. Circ. Sys.*, pp. 1936–1939, May 1995.
- [9] S. L. Gay, J. Hartung, and G. L. Smith, "Algorithms for multi-channel DTMF detection for the WEDSP32 family," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1134–1137, May 1989.
- [10] V. Friedman, "A zero crossing algorithm for the estimation of the frequency of a single sinusoid in white noise," *IEEE Trans. Signal Processing*, vol. 42, pp. 1565 – 1569, June 1994.

LIST OF TABLES

I	ITU specifications for DTMF detection.	17
II	We pass 10 pulses of the DTMF symbol '1' to the detector to demonstrate that the proposed DTMF detector meets the ITU specifications on signal duration, tone interruption, and pause duration for the DTMF symbol '1'.	18
III	Experimental tests indicate that the proposed detector achieves 100% detection when passing ten instances of each DTMF symbol using the tone duration given. All detections are less than the ITU specification of 40 ms and greater than the ITU specification of 23 ms.	19
IV	Talk-off test results using the Bellcore test tapes for our DTMF detector under 8-bit wrap-around arithmetic on 8-bit input data.	20

LIST OF FIGURES

1	Dual-Tone Multiple Frequency (DTMF) scheme for touchtone dialing. When a key is pressed, two sinusoids at the row and column frequencies are added together. . .	12
2	Block diagram of the proposed DTMF detector. We assume that the input signal is sampled at 8 kHz. The decimator reduces the sampling rate to 4 kHz.	13
3	Frequency response of the low frequency data path in the DTMF detector when responding to '1' DTMF symbol consisting of tones at 697 Hz and 1209 Hz. Notches occur at 0 Hz, 300 Hz, 1209 Hz, and 1336 Hz. Over the range of DTMF frequencies, the response peaks at 697 Hz.	14
4	The power estimator and signal detector in the DTMF detector. Power estimator accurately tracks the power of the signal component. The noise floor increases or decreases exponentially with respect to time. The <i>detect</i> signal is true whenever the signal power is greater than the noise floor.	15
5	Decision logic for the proposed DTMF detector. The detector runs at the decimated rate of 2 kHz. Inputs are the low frequency estimate (LF), the high frequency estimate (HF), and the detect signal (DT). The output signals are valid detect (VD) and new symbol (NS). Each time that the detector receives new inputs, VD is set to false (0).	16

		Column →			
		1209 Hz	1336 Hz	1477 Hz	1633 Hz
Row ↓	697 Hz	1	2	3	A
	770 Hz	4	5	6	B
	852 Hz	7	8	9	C
	941 Hz	*	0	#	D

Fig. 1. Dual-Tone Multiple Frequency (DTMF) scheme for touchtone dialing. When a key is pressed, two sinusoids at the row and column frequencies are added together.

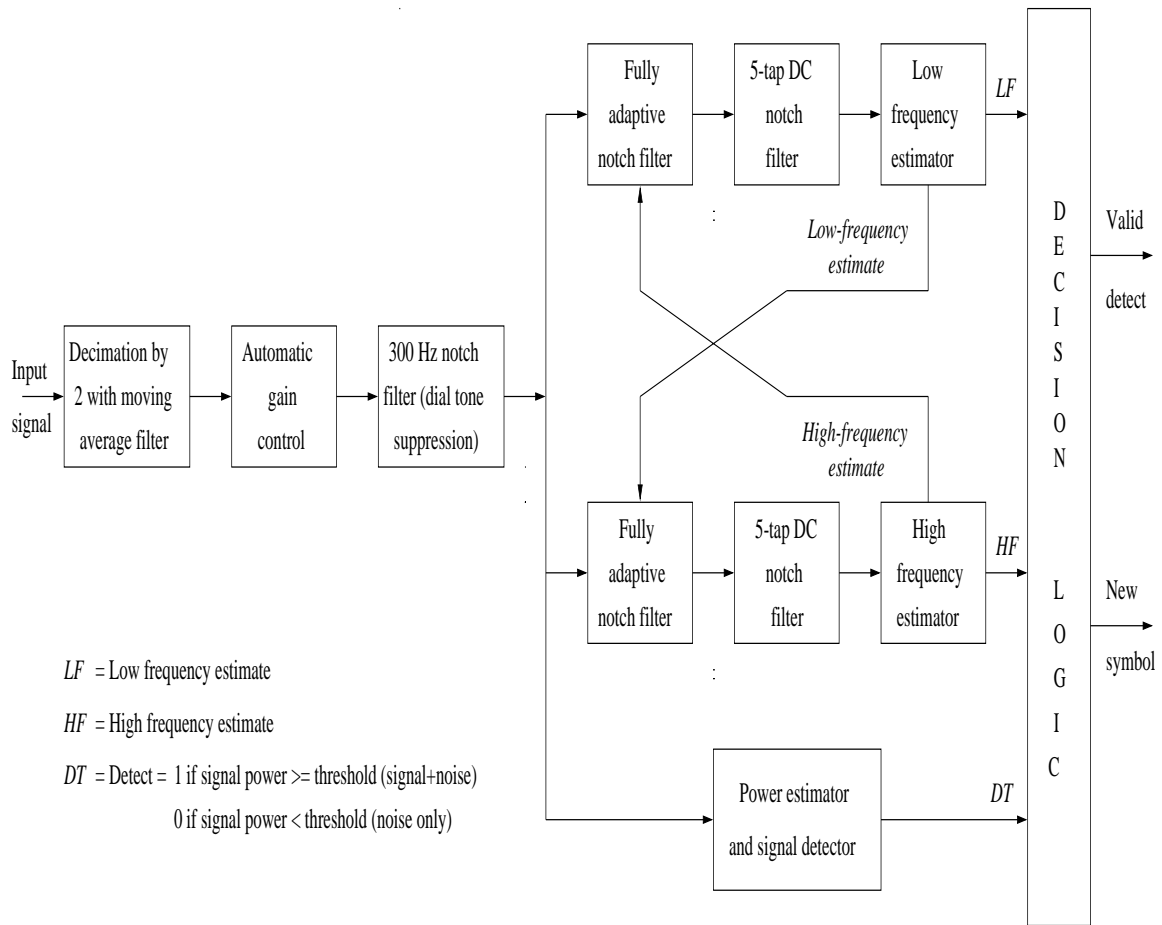


Fig. 2. Block diagram of the proposed DTMF detector. We assume that the input signal is sampled at 8 kHz. The decimator reduces the sampling rate to 4 kHz.

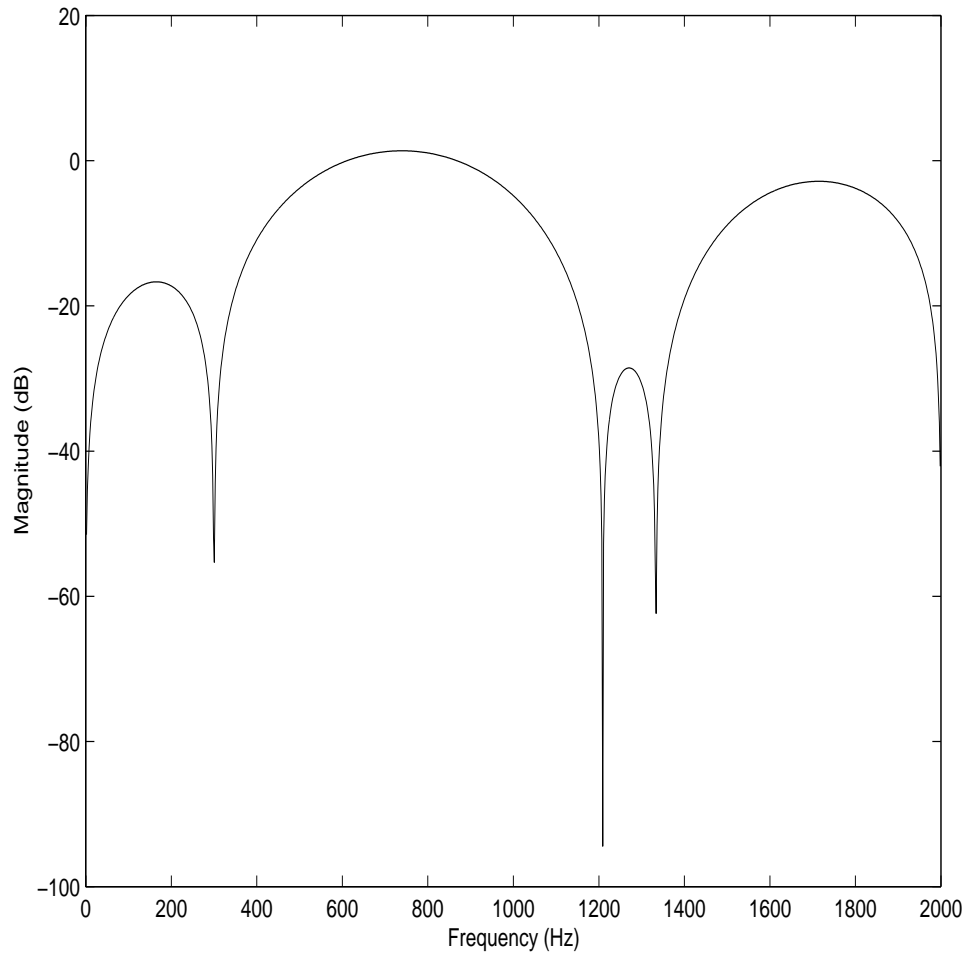


Fig. 3. Frequency response of the low frequency data path in the DTMF detector when responding to '1' DTMF symbol consisting of tones at 697 Hz and 1209 Hz. Notches occur at 0 Hz, 300 Hz, 1209 Hz, and 1336 Hz. Over the range of DTMF frequencies, the response peaks at 697 Hz.

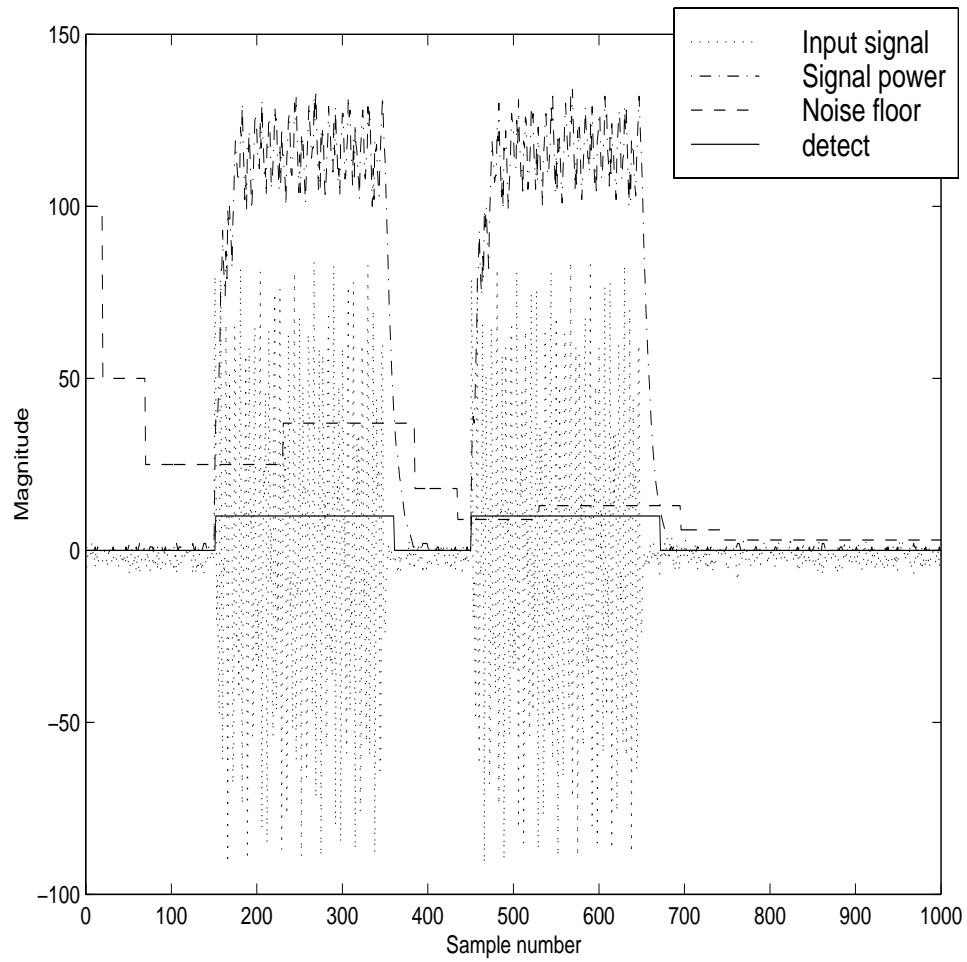


Fig. 4. The power estimator and signal detector in the DTMF detector. Power estimator accurately tracks the power of the signal component. The noise floor increases or decreases exponentially with respect to time. The *detect* signal is true whenever the signal power is greater than the noise floor.

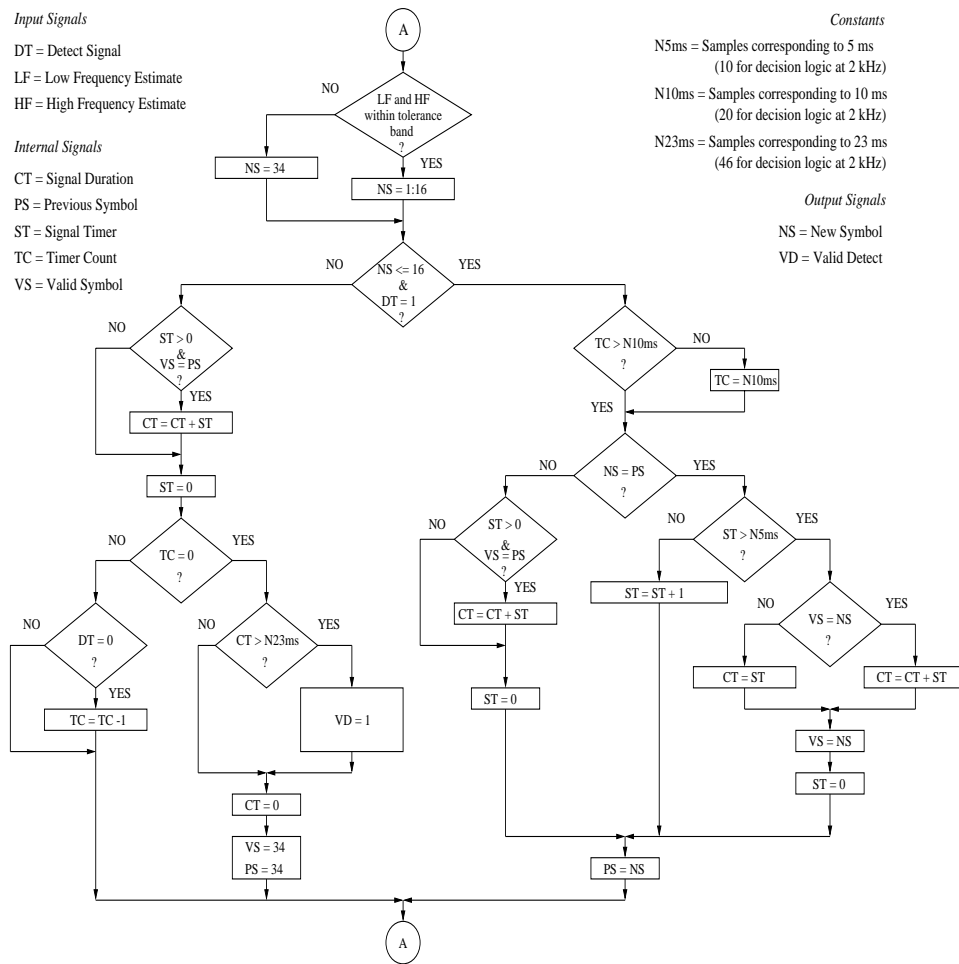


Fig. 5. Decision logic for the proposed DTMF detector. The detector runs at the decimated rate of 2 kHz. Inputs are the low frequency estimate (LF), the high frequency estimate (HF), and the detect signal (DT). The output signals are valid detect (VD) and new symbol (NS). Each time that the detector receives new inputs, VD is set to false (0).

<i>Frequency tolerance</i>	A valid DTMF tone should have a frequency tolerance within $\pm 1.5\%$. A tone offset by $\pm 3.5\%$ should not be detected.
<i>Signal duration</i>	A valid DTMF tone with a duration of 40 ms should be considered valid. Tones of duration of less than 23 ms should be rejected.
<i>Signal interruption</i>	A valid DTMF signal interrupted for 10 ms or less should not be detected as two distinct tones.
<i>Signal pause</i>	A valid DTMF signal separated by a pause time of 40 ms must be detected as two distinct symbols.
<i>Signal strength</i>	The detector should work in a worst-case signal-to-noise ratio (SNR) of 15 dB and signal power of -26 dBm (attenuation of 26 dB for a 1 mW of transmitted power).
<i>Twist</i>	The detector must operate with a maximum of 8 dB normal twist and 4 dB of reverse twist. Twist is defined as the difference in decibels in the amplitudes of the two fundamental tones of the DTMF signal.
<i>Talk-off</i>	The detector should operate in the presence of speech without incorrectly identifying the speech signal as a valid DTMF tone.

TABLE I
ITU SPECIFICATIONS FOR DTMF DETECTION.

	Signal Power		SNR		Specification	
	0 dBm	-20 dBm	15 dB	40 dB	Min.	Max.
Minimum accepted tone length	37 ms	37 ms	37 ms	37 ms	23 ms	40 ms
Maximum rejected tone length	33 ms	33 ms	33 ms	33 ms	23 ms	40 ms
Minimum tone interruption causing two detections	14 ms	14 ms	11 ms	15 ms	10 ms	40 ms
Maximum pause time not causing two detections	14 ms	14 ms	15 ms	14 ms	10 ms	40 ms

TABLE II

WE PASS 10 PULSES OF THE DTMF SYMBOL '1' TO THE DETECTOR TO DEMONSTRATE THAT THE PROPOSED DTMF DETECTOR MEETS THE ITU SPECIFICATIONS ON SIGNAL DURATION, TONE INTERRUPTION, AND PAUSE DURATION FOR THE DTMF SYMBOL '1'.

DTMF Symbol	Minimum Tone Length for 100% detection
1	35 ms
2	35 ms
3	36 ms
4	35 ms
5	35 ms
6	35 ms
7	33 ms
8	34 ms
9	32 ms
0	33 ms
*	34 ms
#	31 ms
A	34 ms
B	33 ms
C	32 ms
D	32 ms

TABLE III

EXPERIMENTAL TESTS INDICATE THAT THE PROPOSED DETECTOR ACHIEVES 100% DETECTION WHEN PASSING TEN INSTANCES OF EACH DTMF SYMBOL USING THE TONE DURATION GIVEN. ALL DETECTIONS ARE LESS THAN THE ITU SPECIFICATION OF 40 MS AND GREATER THAN THE ITU SPECIFICATION OF 23 MS.

DTMF Symbols	Allowed False Detects	Actual False Detects
0-9	333	297
0-9,*,#	500	345
0-9,*,#,A-D	600	502

TABLE IV

TALK-OFF TEST RESULTS USING THE BELLCORE TEST TAPES FOR OUR DTMF DETECTOR UNDER
8-BIT WRAP-AROUND ARITHMETIC ON 8-BIT INPUT DATA.