

Efficient Multitone Detection

This article presents the digital signal processing (DSP) tricks employed to build a computationally efficient multitone detection system, implemented without multiplications and with minimal data memory requirements. More specifically, the article describes the detection of incoming dial tones, validity checking to differentiate valid tones from noise signals, and the efficient implementation of the detection system. While our discussion focuses on dual-tone multifrequency (DTMF) telephone dial tone detection, the processing tricks presented may be employed in other multitone detection systems.

MULTITONE DETECTION

Multitone detection is the process of detecting the presence of spectral tones, each of which has the frequencies $\omega_1, \omega_2, \dots, \omega_k$, where $k = 8$ in our application. A given combination of tones is used to represent a symbol of information, so the detection system's function is to determine what tones are present in

the $x(n)$ input signal. A traditional method for multitone detection is illustrated in Figure 1.

As shown in this figure, the incoming $x(n)$ multitone signal is multiplied by the frequency references $\exp(j\omega_k t)$ for all possible multitone frequencies ω_k , down-converting any incoming tones to be centered at 0 Hz. Next, the complex $u_k(n)$ products are low-pass filtered. Finally, the magnitudes of the complex low-pass-filtered sequences are logically compared to determine which of the ω_k tones are present. For the multitone detection system in Figure 1, the DSP tricks employed are as follows:

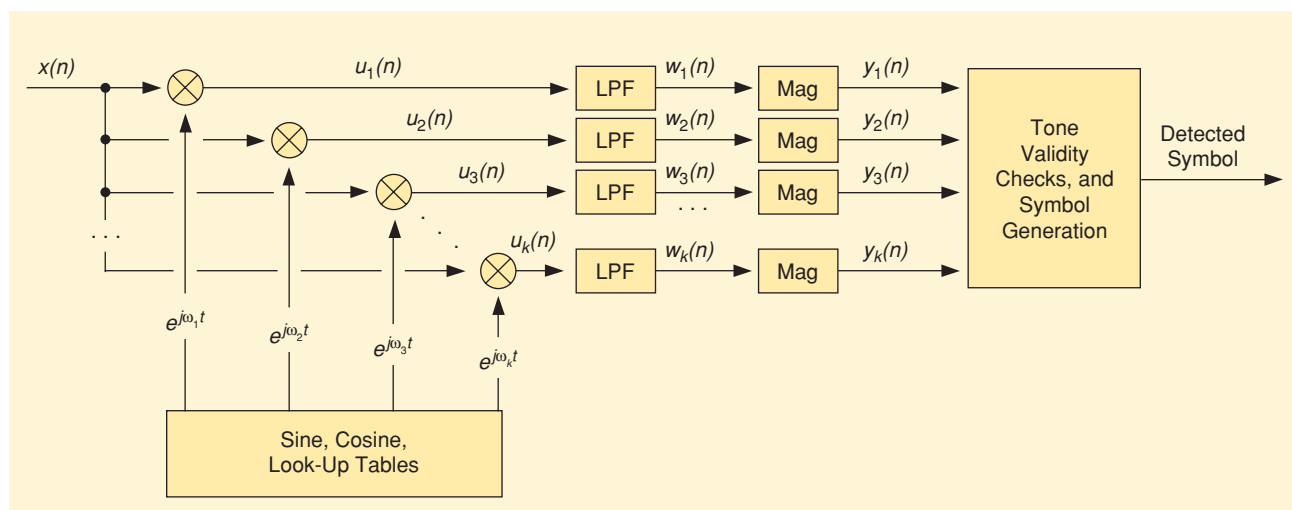
- Use a 1-b signal representation so multiplications are replaced with exclusive-OR operations.
- Group signal samples into 8-b words so eight signal samples are manipulated simultaneously in one clock cycle.
- Perform multiplier-free complex down-conversion using a look-up table (LUT).

- Perform multiplier-free low-pass filtering.
- Perform computation-free elimination of quantization dc bias error.
- Perform multiplier-free complex sample magnitude estimation.

COMPLEX DOWN-CONVERSION

As a first trick to simplify our implementation, the $x(t)$ input signal is converted from an analog signal to a 1-b binary signal, and the $\exp(j\omega_k t)$ reference frequencies are quantized to a 1-b representation. Thanks to this step, the down-conversion multiplication of 1-b data by a 1-b reference frequency sample

"DSP Tips and Tricks" introduces practical design and implementation signal processing algorithms that you may wish to incorporate into your designs. We encourage readers to submit their contributions to Associate Editors Rick Lyons (r.lyons@ieee.org) or Britt Rorabaugh (dspboss@aol.com).



[FIG1] Traditional multitone detection process.

can be performed using a simple exclusive-OR (XOR) operation. To improve computational efficiency, the incoming 1-b data samples from the comparator, arriving at an 8-kHz sample rate, are collected into one 8-bit byte. The 8-b XOR operations—one XOR for the sine part and one XOR for the cosine part of the complex reference frequencies—are performed to process eight $x(n)$ samples simultaneously. The logical zeros in the result of XOR operations correspond to the situation where the input is in phase with an $\exp(j\omega_k t)$ reference, and the logical ones of the XOR result indicate that the input is in quadrature phase with an $\exp(j\omega_k t)$ reference.

The number of zeros and ones in the XOR results, which comprise the real and imaginary parts of the $u_k(n)$ sequences, can be found by counting the nonzero bits directly. However, to enhance execution speed, we use an LUT indexed by the XOR result in lieu of the direct count of ones and zeroes.

The XOR LUT entries, optimized to provide the best numeric accuracy in conjunction with the follow-on low-pass filters, take advantage of the full numeric range ($-128 \dots +127$) of a signed 8-b word. If N is the number of ones in an 8-b XOR result, the LUT entries are computed as

$$\text{Table}[N] = \text{Round}[29.75(N - 4) + 8], \quad (1)$$

where Round $[\cdot]$ denotes rounding to the nearest integer. Using (1) for our application, the XOR LUT contains the entries listed in Table 1. Because XOR results are 8-b words, the XOR LUT has 256 entries, where each entry is one of the nine values in Table 1, depending on the number of logic ones in an XOR result. The multiplication factor 29.75 in (1) was chosen to partition the LUT entry range, $-128 \dots +127$, into nine intervals while accommodating the $+8$ term. [The constant term in (1) equal to $+8$ will be explained shortly.]

The reference frequency samples (square waves, actually) are also stored in LUTs. Ideally, the size of the sine and cosine LUTs should be equal to the least

common period for all DTMF frequencies. However, the least common period of the DTMF frequencies is 1 s. For the sample rate of 8 kHz, the reference frequency LUT sizes are equal to $16k$ (2^{14}). Because tables of this size cannot be realized in low-end microcontrollers, the values of the frequencies were modified to fit the least common period into a smaller table. We found that the common period of 32 ms is a good compromise between the frequency accuracy and the size of the LUT. In this case, the size of the LUT is equal to 512 B, and the difference between the LUT frequencies and the DTMF standard frequencies is 10 Hz or less. This mismatch does not affect the operation of the multitone detector.

The LUT of 512 B for the reference frequencies may be too large for some applications. Numerically controlled oscillator (NCO) sine and cosine generation can be used as an alternative [3], [4]. In that scenario, typical 16-b NCOs require an extra 16 B of random-access memory (RAM) and create an additional computing workload on the order of the three million operations per second (MIPS); however, the NCO method frees the 512 B of ROM.

LOW-PASS FILTERING

The $w_k(n)$ product sequences must be low-pass filtered. As our next trick, we apply the $w_k(n)$ sequences to a bank of computationally efficient low-pass filters defined by

$$w_k(n) = w_k(n-1) + [u_k(n) - w_k(n-1)]/16. \quad (2)$$

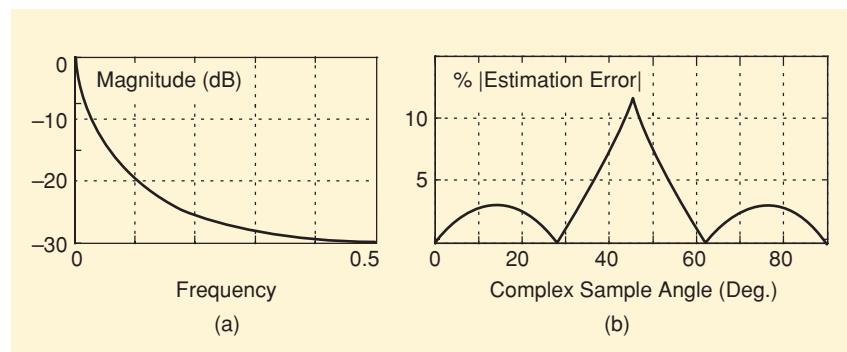
[TABLE 1] XOR LUT.

N	LUT ENTRY, $u_k(n)$
0	-111
1	-81
2	-52
3	-22
4	+8
5	+38
6	+68
7	+97
8	+127

This economical filter requires only a single data memory storage location and no filter coefficient storage. To enhance execution speed, we implement the divide by 16 with an arithmetic right-shift by 4 b. The filters are computed with the precision of the 8 b.

The astute reader may recognize (2) as the computationally efficient exponential averaging low-pass filter. That filter, whose frequency magnitude response is shown in Figure 2(a), exhibits nonlinear phase, but that is of no consequence in our application. The filter's weighting factor of 1/16 is determined by the filter's necessary 3-dB bandwidth of 10 Hz, as mandated by the accuracy of the entries stored in the reference frequencies LUT.

When the arithmetic right-shift is executed in the low-pass filters, a round to the nearest integer operation is performed to improve precision by adding 8 to the number before making the right-shift by 4 b. Rather than repeatedly adding 8 to each filter output sample, the trick we use next to eliminate all these addition operations is to merely add a constant value of 8 to the XOR LUT. This scheme accounts for the constant 8 term in (1).



[FIG2] Detector performance: (a) low-pass filter response and (b) approximation error of the $w_k(n)$ magnitude.

MAGNITUDE APPROXIMATION

Every 16 ms, the magnitudes of all the complex $w_k(n)$ sequences are computed by obtaining the real and imaginary parts of $w_k(n)$ and using the approximation

$$|w_k(n)| \approx y_k(n) = \max\{|real[w_k(n)]|, |imag[w_k(n)]|\} + \min\{|real[w_k(n)]|, |imag[w_k(n)]|\}/4 \quad (3)$$

to estimate the magnitude of the complex $w_k(n)$ sequences. This magnitude approximation trick requires no multiplications because the multiplication by 1/4 is implemented using an arithmetic right-shift by 2 b. The maximum magnitude estimation error of (3), shown in Figure 2(b), is roughly equal to 11%, which is acceptable for the operation of the DTMF decoder. (Magnitude approximation algorithms that are similar to (3) and have improved accuracy at the expense of additional shift and add/subtract operations are described in [1]).

VALIDITY CHECKING

The computed magnitudes are processed to decide whether valid DTMF tones are present; if so, the DTMF symbol is decoded. First, the three maximum magnitudes are found and arranged in descending order: $M_1 \geq M_2 \geq M_3$. Then the validity conditions are checked using the following rules:

- 1) The two frequencies corresponding to M_1 and M_2 must fall into the two different touch-tone telephone frequency groups (the < 1 kHz group and the > 1 kHz group).

2) Magnitude M_2 must be greater than the absolute threshold value T_1 .

3) Ratio M_2/M_3 must be greater than the relative threshold value T_2 .

The threshold values T_1 and T_2 depend on the real-world system's characteristics such as the signal sample rate, the analog signal's spectrum, the analog filtering prior to digital processing, the probability density of the signal and the noise, the desired probability of false alarm, and the probability of detection. As such, the threshold values are found empirically while working with the hardware implementation of the multitone detector. Note that if we normalize the ideal single tone maximum amplitude at the output to unity, the value of the threshold T_1 will be on the order of the absolute value range of the XOR LUT divided by 4. As such, in this application T_1 is approximately $120/4 = 30$. The relative threshold T_2 does not depend on tone amplitudes. A typical value for T_2 is approximately 2.

Also note that the third conditional check above does not require a division operation. That condition is verified if $M_2 \geq T_2 M_3$. To enhance speed, the multiplication by the constant threshold T_2 is implemented by arithmetic shift and addition operations. Finally, if all three validity conditions are satisfied, the DTMF tones are considered to be valid. This validity checking allows us to distinguish true DTMF tones from speech, the silence in speech pauses, noise, or other signals.

IMPLEMENTATION ISSUES

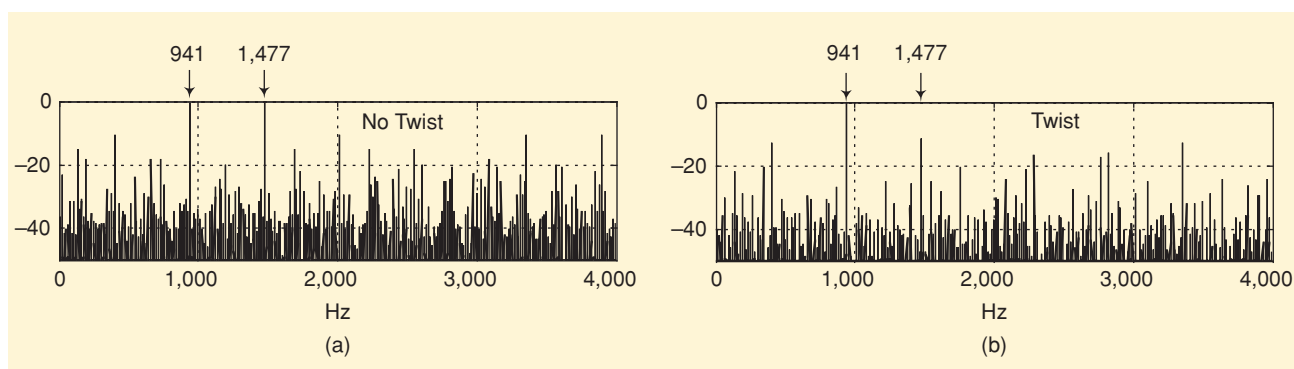
The multitone detection algorithm requires roughly 1,000 B of read-only memory (ROM) and 64 B of read/write

memory (RAM). The algorithm requires no analog-to-digital (A/D) converter and no multiply operations, and its processing workload is quite low, equal to only 0.5 MIPS. (For comparison, the DTMF detection system described in [2] requires 24 MIPS of processing power on a fixed-point DSP chip.) As such, fortunately, it is possible to implement the DTMF multitone detector/decoder using a low-cost 8-b microcontroller such as Atmel's AVR, Microchip Technology's PIC, Freescale's HC08, or the x51 family of microcontrollers.

OTHER CONSIDERATIONS

The performance of the above algorithm is satisfactory for most practical DTMF detection applications. However, in some cases, more robust processing is needed. For example, if the DTMF signal has become distorted by the processing of a low-bitrate speech coder/decoder, the above algorithm may not operate properly. The weakness of the algorithm is its sensitivity to *frequency twist* and other noise interference. Frequency twist occurs when the power levels of the two detected tones differ by 4–8 dB. The sensitivity in this case is due to the input signal being digitized to 1 b by the comparator (a highly nonlinear procedure.)

To illustrate this behavior, Figure 3(a) shows the output spectra of the 1-b comparator for the input tones of 944 and 1,477 Hz when no frequency twist occurs. Here we can see the mix of the input tones and the miscellaneous nonlinear spectral products. If the amplitudes of both input tones are equal, the nonlinear spectral artifacts are at least



[FIG3] 1-b comparator output spectra (a) with no frequency twist and (b) with frequency twist.

10 dB lower than the desired signals. Figure 3(b) also shows the output spectra of the comparator with frequency twist. If the amplitudes of the two input tones differ by 6 dB or more (twist), then the strongest nonlinear spectral products have approximately equal amplitude with that of the weakest of the two input tones. Some of the products are falling into the bandwidth of the low-pass filters. Note also that a difference in the levels of input tones equal to 6 dB causes that difference to be 12 dB after the comparator, which limits the performance of the multitone detector.

As such, we caution the reader that 1-b quantization is not always appropriate for signal detection systems. Sadly, there are no hard-and-fast rules to identify the cases where 1-b quantization can be used.

For the detection of spectrally simple, high signal-to-noise ratio, stable-amplitude, oversampled signals, 1-b quantization may be applicable. However, because the statistical analysis of 1-b quantization errors is complicated, careful modeling must be conducted to evaluate the nonlinear performance and spectral aliasing effects of 1-b quantization.

If improved performance beyond that of 1-b quantization is required, a similar algorithm can be applied for processing $x(n)$ input signals from an A/D converter. The multibit $x(n)$ signal's multiplication by the 1-b reference frequencies can, again, be performed using simple 16-b addition and subtraction operations. The A/D-converter-based algorithm is able to decode the DTMF under any conditions; however, the tradeoff is that the processing must be done for every sample, not for eight samples simultaneously. This increases the computing burden to several MIPS.

Finally, note that the DTMF detection algorithm presented here is not strictly compliant with the ITU/Bellcore standards [5]. However, this is not detrimental in most practical cases.

ACKNOWLEDGMENT

The author thanks Rick Lyons for his great help in the preparation of this article.

AUTHOR

Vladimir L. Vassilevsky (vlv@abvolt.com) is a consultant in the areas of the embedded systems and digital signal processing and founder of Abvolt Ltd. He received M.S. and Ph.D. degrees from Moscow Institute for Physics and Technology (State University), Moscow, Russia, in 1994 and 1997, respectively. His interests are in nontraditional design approaches in the areas of telecommunication, audio engineering, control, and automation systems.

life SCIENCES continued from page 139

accurate processing functions. Understanding these basic structures may help us understand the dynamics and functions of large genetic transcription networks. Because of the huge number of nodes and links in these networks, current and future work includes designing faster and more accurate methods to detect motifs, as well as accurate mathematical models to discover their dynamic properties and functions, especially in integrated cellular network motifs.

ACKNOWLEDGMENTS

The authors thank Mr. Difei Li for his help in preparing materials for this article. This work was supported by the Fok Ying Tung Education Foundation under grant 101064, the National Natural Science Foundation of China under grant 60502009, the Program for New Century Excellent Talents in University, the Distinguished Youth Foundation of Sichuan Province, the MEXT of Japan grant-in-aid 17022012 for Scientific Research on Priority Areas.

AUTHORS

Chunguang Li (cgli@uestc.edu.cn) is with the School of Electronic Engineering, University of Electronic Science and Technology of China, China.

Luonan Chen (chen@eic.osaka-sandai.ac.jp) is with the Institute of Systems Biology, Shanghai University, China; the Faculty of Engineering, Osaka

REFERENCES

- [1] M. Allie and R. Lyons, "A root of less evil," *IEEE Signal Processing Mag.*, vol. 22, no. 2, pp. 93–96, Mar. 2005.
- [2] M. Felder, J. Mason, and B. Evans, "Efficient dual-tone multifrequency detection using the nonuniform discrete Fourier," *IEEE Signal Processing Lett.*, vol. 5, no. 7, pp. 160–163, July 1998.
- [3] Analog Devices Inc., "A technical tutorial on digital signal synthesis" [Online]. Available: http://www.analog.com/UploadedFiles/Tutorials/450968421DDS_Tutorial_rev12-2-99.pdf
- [4] L. Cordesses, "Direct digital synthesis: A tool for periodic wave generation," *IEEE Signal Processing Mag.*, vol. 21, no. 2, pp. 50–54, July 2005.
- [5] R. Freeman, *Reference Manual for Telecommunications Engineering*. New York: Wiley-Interscience, 2002.

SP

Sanyo University; the Institute of Industrial Science, University of Tokyo; and the Japan Science and Technology Agency, Japan.

Kazuyuki Aihara (aihara@sat.t.u-tokyo.ac.jp) is with the Institute of Industrial Science, University of Tokyo, and the Japan Science and Technology Agency, Japan.

REFERENCES

- [1] D.J. Watts and S.H. Strogatz, "Collective dynamics of 'small world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [2] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [3] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.
- [4] U. Alon, *An Introduction to System Biology: Design Principles of Biological Circuits*. Boca Raton, FL: Chapman & Hall, 2006.
- [5] M. Ptashne, *A Genetic Switch: Gene Control and Phage λ*. Oxford, England: Blackwell, 1992.
- [6] E.M. Ozbudak, M. Thattai, H.N. Lim, B.I. Shraiman, and A. van Oudenaarden, "Multistability in the lactose utilization network of *Escherichia coli*," *Nature*, vol. 427, no. 6976, pp. 737–740, 2004.
- [7] B. Ghosh, R. Karmakar, and I. Bose, "Noise characteristics of feed forward loops," *Phys. Biol.*, vol. 2, no. 1, pp. 36–45, 2005.
- [8] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 21, pp. 11,980–11,985, 2003.
- [9] E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Pinter, U. Alon, and H. Margalit, "Network motifs in integrated cellular networks of transcription–regulation and protein–protein interaction," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 16, pp. 5934–5939, 2004.
- [10] P. Francois and V. Hakim, "Core genetic module: The mixed feedback loop," *Phys. Rev. E*, vol. 72, no. 3, 031908, 2005.

SP