DTMF Tone Generation and Detection: An Implementation Using the TMS320C54x

Gunter Schmer, MTSA SC Group Technical Marketing

ABSTRACT

This application note describes the implementation of a dual tone multiple frequency (DTMF) tone generator and detector for the TMS320C54x™. This application note provides some theoretical background on the algorithms used for tone generation and detection. It documents the actual implementation in detail. Finally the code is benchmarked in terms of its speed and memory requirements.

Contents

List of Figures

List of Tables

1 DTMF Touch Tone Dialing: Background

A DTMF (dual tone multiple frequency) codec incorporates an encoder that translates key strokes or digit information into dual tone signals, as well as a decoder detecting the presence and the information content of incoming DTMF tone signals. Each key on the keypad is uniquely identified by its row and its column frequency as shown in Figure 1. The DTMF generating and decoding scheme is not too computationally extensive and can easily be handled by a DSP concurrently with other tasks. This article describes an implementation of the DTMF codec on the TMS320C54x, TI’s fixed-point DSP designed especially for telecommunication applications.

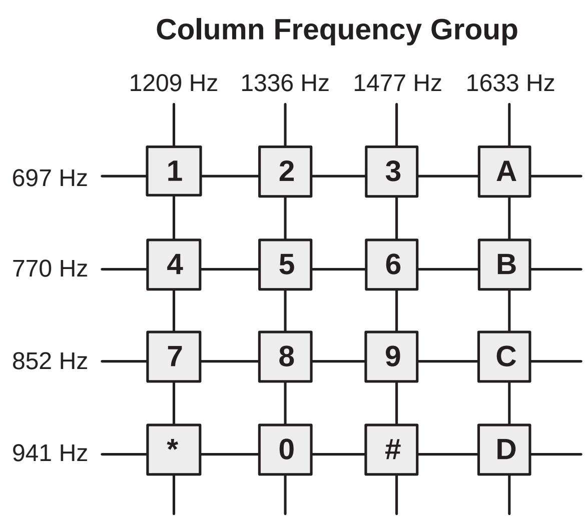

Figure 1. Touch-Tone Telephone Keypad: A Row and a Column Tone is Associated With Each Digit

2 DTMF Tone Generator

The encoder portion and tone generation part of a DTMF codec is based on two programmable, second order digital sinusoidal oscillators, one for the row the other one for the column tone. Two oscillators instead of eight facilitate the code and reduce the code size. Of course for each digit that is to be encoded, each of the two oscillators needs to be loaded with the appropriate coefficient and initial conditions before oscillation can be initiated. As typical DTMF frequencies range from approx. 700 Hz to 1700 Hz, a sampling rate of 8 kHz for this implementation puts us in a safe area of the Nyquist criteria. Figure 2 displays the block diagram of the digital oscillator pair. Table 1 specifies the coefficients and initial conditions necessary to generate the DTMF tones.

If you are interested in some more detail, appendix A gives some refreshing theoretical background and a guideline for determining coefficients and initial conditions for digital sinusoidal oscillators.

Tone duration specifications by AT&T state the following: 10 digits/sec are the maximum data rate for touch tone signals. For a 100 msec time slot the duration for the actual tone is at least 45 msec and not longer than 55 msec. The tone generator must be quiet during the remainder of the 100 msec time slot.

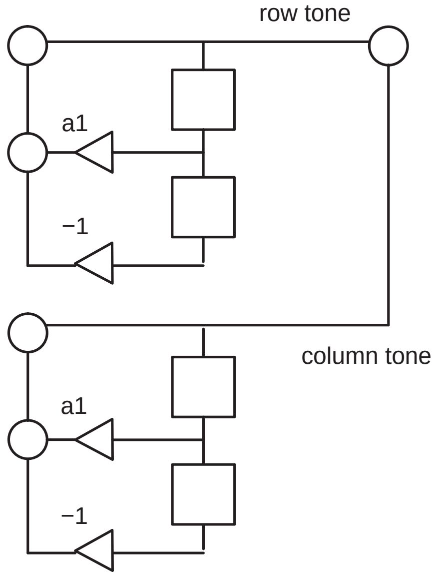

Figure 2. Two Second-Order Digital Sinusoidal Oscillators

Table 1. Coefficients and Initial Conditions

f/Hz	a1	y(−1)	y(−2)/A
697	0.85382	0	−0.52047
770	0.82263	0	−0.56857
852	0.78433	0	−0.62033
941	0.73911	0	−0.67358
1209	0.58206	0	−0.81314
1336	0.49820	0	−0.86706
1477	0.39932	0	−0.91680
1633	0.28424	0	−0.95874

2.1 Program Flow Description of the DTMF Tone Generator

For the following description of the program flow it is helpful to simultaneously consult the flowchart given in Figure 3. Essentially the series of keypad entries (digits) will be translated into a series of dual-tones of certain duration which are interrupted by pauses of certain duration. Later the dual-tones will enable the decoder to identify the associated digits. The pauses are also necessary to discriminate between two or more identical digits entered successively.

The DTMF tone generator follows a buffered approach, which means that the results of its execution will be frames of data forming a continuous data stream. Each frame – 15 ms or 120 samples long - contains either DTMF tone samples or pause samples. The program flow of the DTMF tone generator is controlled by a set of variables. The variable *encStatus* reflects the current status of the encoder. The encoder is either in idle mode (*encStatus* = 0) and is currently not used to encode digits, or it is active (*encStatus* = 1) and generates DTMF tones and pauses of certain duration. Tone duration and pause duration will be monitored with the variables *toneTime* and *pauseTime*. At the beginning of each encoding process of a given phone number *toneTime* and *pauseTime* are initialized with the desired values and the encoder is activated (*encStatus* = 1). The encoder retrieves the first digit from the digit buffer and unpacks it. Unpacking means that the digit is mapped to the row/column tone properties (oscillator coefficients, initial conditions) and pointers are loaded, pointing to the appropriate locations in the oscillator property table. The encoder then generates DTMF tone frames and decrements *toneTime* accordingly. When the desired tone duration is reached (*toneTime* = 0) the encoder starts outputting pause frames. As the encoder is decrementing *pauseTime* with each pause frame it reaches the desired pause duration when *pauseTime* = 0. The encoder just completed encoding the first digit in the digit buffer and now continues with the next digit. It has to reinitialize *toneTime* and before going to the next tone/pause cycle. The encoder recognizes the completion of the encoding process of the entire phone number with a digit equal to –1 in the digit buffer and switches itself to idle state (*encStatus* = 0).

2.2 Multichannel DTMF Tone Generation

The software is written as C-callable reentrant functions. This enables the user to set up multichannel tone detectors in C without adding significant additional code. In order to facilitate and structure multichannel applications, the code uses a structure to hold all the global variables and pointers to various arrays for a single channel. All the user has to do, is to define a structure for each channel and initialize it properly. A call of the function *dtmfEncode(DTMFENCOBJ *)* to which a pointer to the defined structure is passed will invoke the encoding process.

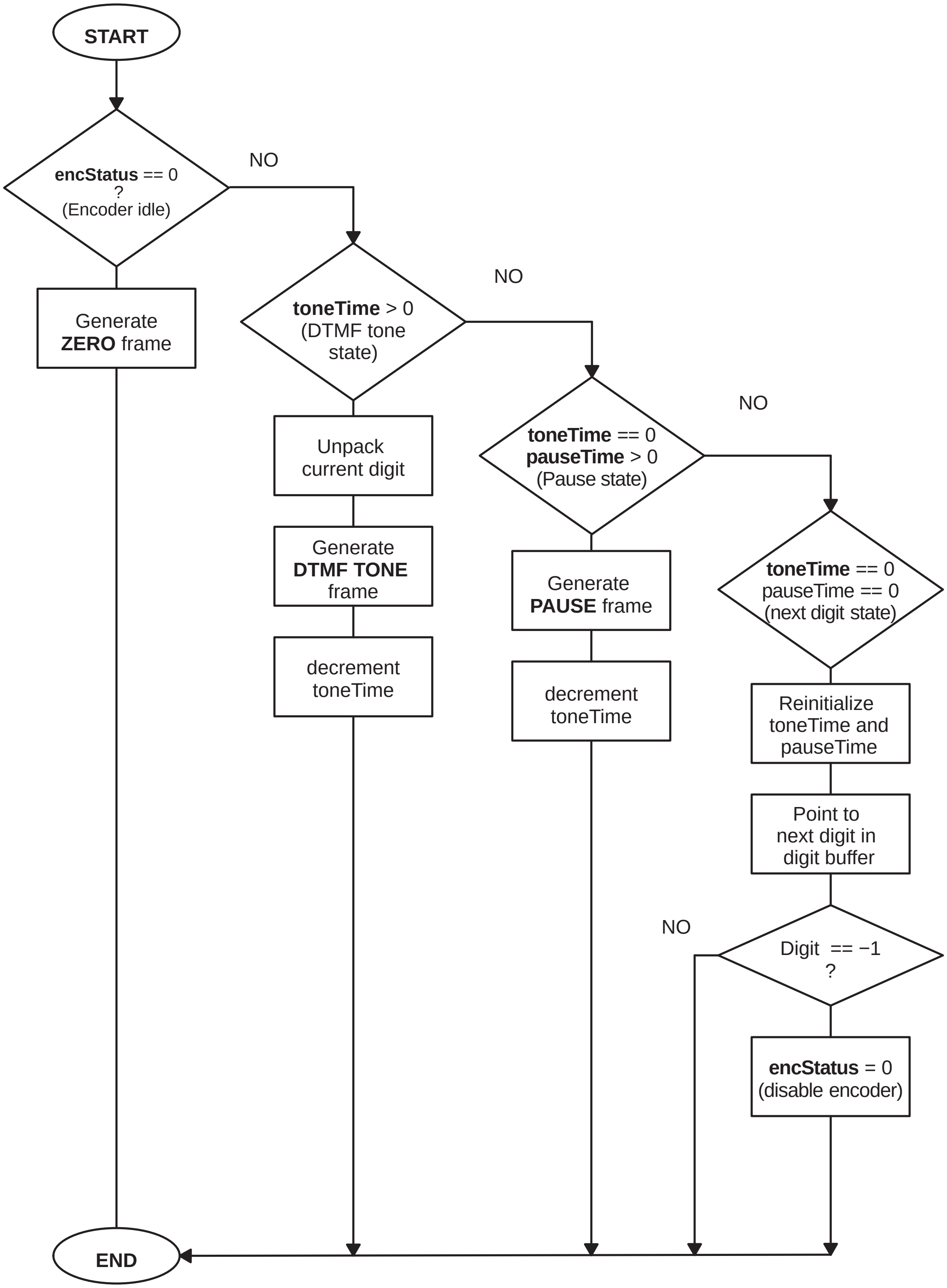

Figure 3. Flowchart of the DTMF Encoder Implementation

3 DTMF Tone Detector

The task to detect DTMF tones in a incoming signal and convert them into actual digits is certainly more complex then the encoding process. The decoding process is by its nature a continuous process, meaning it needs to search an ongoing incoming data stream for the presence of DTMF tones continually.

3.1 Collecting Spectral Information

The Goertzel algorithm is the basis of the DTMF detector. This method is a very effective and fast way to extract spectral information from an input signal. This algorithm essentially utilizes two-pole IIR type filters to effectively compute DFT values. It thereby is a recursive structure always operating on one incoming sample at a time, as compared to the DFT (or FFT) which needs a block of data before being able to start processing. The IIR structure for the Goertzel filter incorporates two complex-conjugate poles and facilitates the computation of the difference equation by having only one real coefficient. For the actual tone detection the magnitude (here squared magnitude) information of the DFT is sufficient. After a certain number of samples N (equivalent to a DFT block size) the Goertzel filter output converges towards a pseudo DFT value *vk(n),* which can then be used to determine the squared magnitude. See Figure 4 for a short mathematical description of the algorithm. More detail is provided in Appendix B.

Goertzel Algorithm in short:

1. Recursively compute for n = 0 .. N

$$v_k(n) = 2\cos\left(\frac{2\pi}{N}k\right) \cdot v_k(n-1) - v_k(n-2) + x(n)$$

$$where \quad v_k(-1) = 0 \quad v_k(-2) = 0$$

$$x(n) = input$$

2. Compute once every N

$$|X(k)|^2 = y_k(N) y^*_k(n)$$

$$= v^2{}_k(n) + v^2{}_k(N-1) - 2\cos(2\pi f_k/f_s) v^2{}_k(N)\, v^2{}_k(N-1)$$

Figure 4. Goertzel Algorithm

The Goertzel algorithm is much faster than a true FFT, as only a few of the set of spectral line values are needed and only for those values are filters provided.

Squared magnitudes are needed for 8 row/column frequencies and for their 8 corresponding 2nd harmonics. The 2nd harmonics information will later enable us to discriminate DTMF tones from speech or music. Table 2 contains a list of frequencies and filter coefficients. Each filter is tuned to most accurately coincide with the actual DTMF frequencies. This is also true for corresponding 2nd harmonics. The exception is the fundamental column frequencies. Each column frequency has two frequency bins attached, which deviate +/–9Hz from center (see Table 2). This modification was necessitated by the strict recognition bandwidth requirements of the Bellcore specification, and allows to widen the mainlobe for column frequencies, while the mainlobe for row frequencies remains tight. Since Bellcore specifies frequency deviation relative (in percent of center frequency) across all frequencies, the absolute deviation for row frequencies is lower than for column frequencies.

The parameter *N* defines the number of recursive iterations and also provides a means to tune for frequency resolution. The following relationship maps *N* to the width of a frequency bin mainlobe and thereby frequency resolution

mainlobe = fs / N

Again the Bellcore recognition bandwidth requirements necessitate the setting for *N*=136, which corresponds to a mainlobe width of ~58Hz. The mainlobe width is sufficiently narrow to allow the rejection of tones at +/−3.5% off the center frequency, when the signal strength threshold is set appropriately. Even at the lowest row frequency the detector will reject tones >3.5% off center frequency. Guaranteed recognition of tones deviating +/−1.5% from center frequency is more demanding for higher column frequencies where the mainlobe is too narrow (e.g. 58Hz/1633Hz = 3.5% or +/−1.7%). To allow more margin to the specification dual frequency bins are used as mentioned above. The +/−9Hz deviation buys at least another 1% (18Hz/1633Hz) of mainlobe width for column frequencies.

Table 2. Filter Coefficients for Row, Column and 2nd Harmonic Frequencies

1st Harmonics fs = 8 kHz			2nd Harmonics fs = 8 kHz		
DTMF frequency f/Hz	**Detection freq bins at fk/Hz**	**Coefficient cos(2pi *fk/fs*)**	**2nd harm frequency f/Hz**	**Detection freq bin at *fk*/Hz**	**Coefficient cos(2pi *fk/fs*)**
rows					
697	697	27980	1394	1394	15014
770	770	26956	1540	1540	11583
852	852	25701	1704	1704	7549
941	941	24219	1882	1882	3032
columns					
1209	1200 1218	19261 18884	2418	2418	−10565
1336	1327 1345	16525 16123	2672	2672	−16503
1477	1468 1486	13297 12872	2954	2954	−22318
1633	1624 1642	9537 9093	3266	3266	−27472

3.2 Validity Checks

Once the spectral information in form of squared magnitude at each of the row and column frequencies is collected, a series of tests needs to be executed to determine the validity of tone and digit results. Note, that the spectral information of the 2nd harmonic frequencies have not yet been computed. To improve the execution speed they will be computed conditionally within the 2nd harmonics check.

A first check makes sure the signal strength of the possible DTMF tone pair is sufficient. The sum of the squared magnitudes of the peak spectral row component and the peak spectral column component need to be above a certain threshold (THR_SIG). Since already small twists (row and column tone strength are not equal) result in significant row and column peak differences, the sum of row and column peak provides a better parameter for signal strength than separate row and column checks. Tone twists are investigated in a separate check to make sure the twist ratio specifications are met.

The spectral information can reflect two types of twists. The more likely one, called “reverse twist”, assumes the row peak to be larger than the column peak. Row frequencies (lower frequency band) are typically less attenuated as than column frequencies (higher frequency band), assuming a low-pass filter type telephone line. The decoder computes therefore a reverse twist ratio and sets a threshold (THR_TWIREV) of 8dB acceptable reverse twist. The other twist, called “standard twist”, occurs when the row peak is smaller than the column peak. Similarly, a “standard twist ratio” is computed and its threshold (THR_TWISTD) is set to 4dB acceptable standard twist.

The program makes a comparison of spectral components within the row group as well as within the column group. The strongest component must stand out (in terms of squared amplitude) from its proximity tones within its group by more than a certain threshold ratio (THR_ROWREL, THR_COLREL).

The program checks on the strength of the second harmonics in order to be able to discriminate DTMF tones from possible speech or music. It is assumed that the DTMF generator generates tones only on the fundamental frequency, however speech will always have significant even-order harmonics added to its fundamental frequency component. This second harmonics check therefore makes sure that the ratio of second harmonics component and fundamental frequency component is below a certain threshold (THR_ROW2nd, THRCOL2nd). If the DTMF signal pair passes all these checks, we say, a valid DTMF tone pair, which corresponds to a digit, is present. Essentially only the two second harmonic energies that correspond to the detected two fundamental frequencies are computed using the Goertzel algorithm.

Finally we need to determine if the valid DTMF tone exists for a time-duration specified by Bellcore. We need to guarantee recognition of tones longer than 45ms and also need to guarantee rejection of tones shorter than 23ms. The duration check then requires the tone information to be present for at least *two buffer durations*. In order to additionally tune the effective detection duration, the detection buffers are overlapped (overlap-and-save scheme). After careful testing the required overlap to meet tone Bellcore tone recognition/rejection duration was determined: 136-long buffers are overlapped by 16 samples. The algorithm effectively processes 136 samples every 15ms (120 samples). Accordingly every new 136-sample input buffer consists of the last 16 samples of the previous input buffer and 120 new samples from the A/D converter. If valid DTMF tone information exists for at least two successive 136-long buffers (which are overlapped) the tone is valid and mapped to the corresponding digit.

3.3 Program Flow Description of the DTMF Detector

Once the input buffer has been filled with new input data the frame process can start. The content of the input data buffer is copied into an intermediate buffer for processing. All the detection functions will then operate on the intermediate buffer. In order for the DTMF detector to operate in the presence of a strong dial tone pair (350Hz/440Hz), a special front-end notch filter was designed to notch the dial tone signal component while conserving the DTMF signal. The gain control function attenuates strong signal inputs and protects the succeeding functions from overflow of the accumulators. Then the Goertzel filters are executed for the 8 fundamental DTMF frequencies, where the column frequencies are represented with two frequency bins each. Since the preceding gain control ensures that overflow cannot occur, overflow checking was removed and optimized loops allow fast execution. From the resulting filter delay states energy values for the 8 fundamental DTMF tones are computed and logged in an energy template to complete the actual Goertzel algorithm. For the next round of execution the filter delay states are initialized to zero.

The digit validation checks are invoked with the collected energy information. The energy template is first searched for row and column energy peaks. From then on the detector essentially operates in two modes: The tone/digit detection mode or the pause detection mode. In the tone/digit detection mode the detector searches for DTMF tone presence and executes all the digit validation tests. In the pause detection mode DTMF tone detection is disabled and the decoder first has to await a pause signal. Tone/digit or pause modes are controlled by the *detectstat* variable. Digit validation checks include signal strength, reverse and standard twist, relative peaks, second harmonics and digit stability. The computation of the 2nd harmonics energy information has been intentionally made part of the digit validation tests to compute 2nd harmonics only when needed and only the two that are needed. With the successful completion of these tests the valid digit is stored into the digit output buffer.

3.4 Multichannel DTMF Tone Detection

The software is written as C-callable reentrant functions. This enables the user to set up multichannel tone detectors in C without adding significant additional code. In order to facilitate and structure multichannel applications, the code uses a structure to hold all the global variables and pointers to various arrays for a single channel. All the user has to do, is to define a structure for each channel and initialize it properly. When the user calls the function *dtmfDecode(DTMFDECOBJ *)* and passes a pointer to the defined structure, DTMF tone detection is executed.

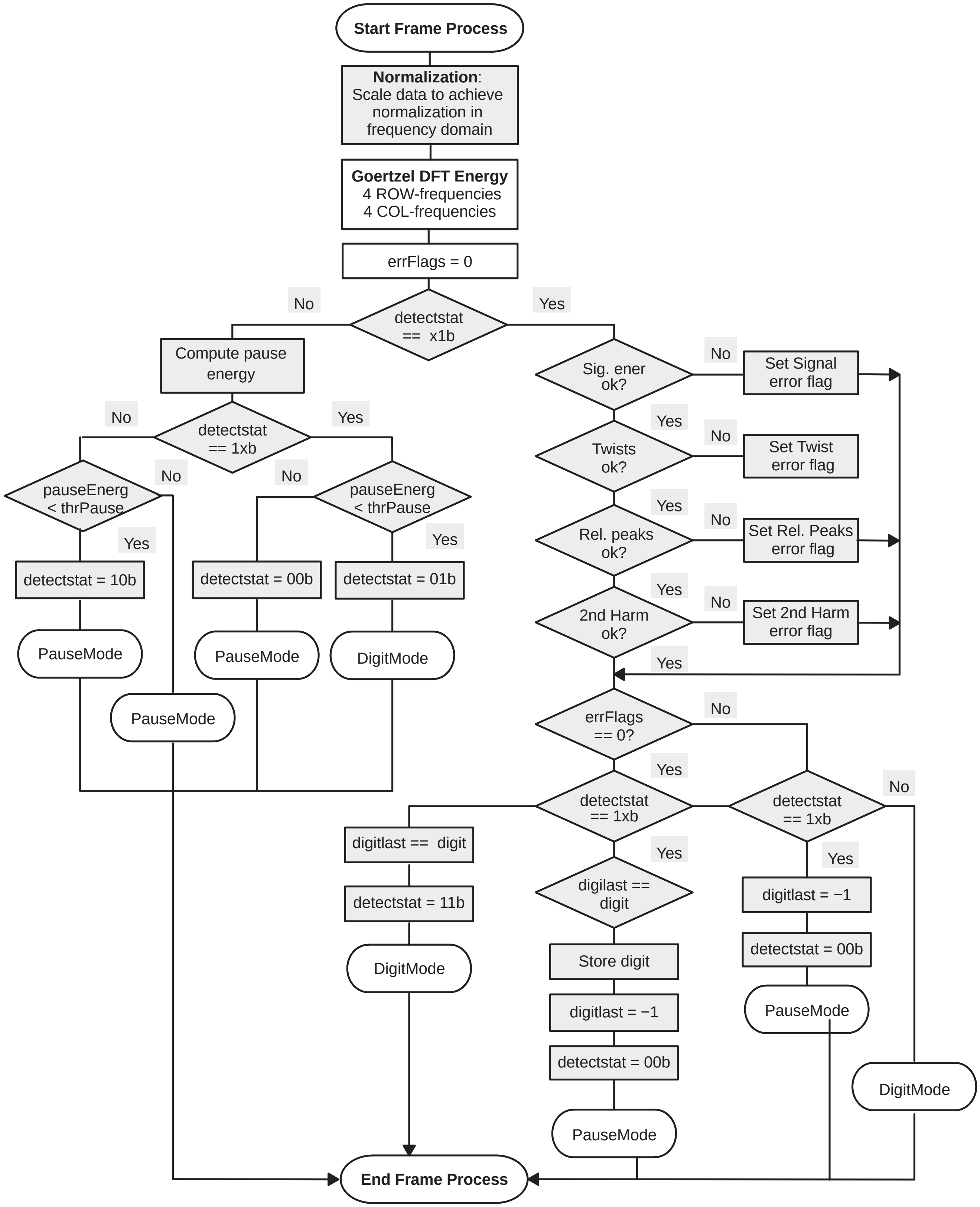

Figure 5. Flowchart of the DTMF Decoder Implementation

4 Speed and Memory Requirements

Table 3 and Table 4 summarize the speed and memory requirements of the DTMF encoder/decoder implementation. The encoder as well as the decoder implementation has been designed for multichannel applications and a single DSP can process a large amount of channels. The MIPS count for the DTMF encoder is approximately 0.15 MIPS. The isolated DTMF decoder uses approximately 0.8 MIPS. These speed specifications include all processing necessary after the completion of the receive interrupt service routine and do not include interrupt service processing. Also, a sampling rate of 8 kHz is assumed. The MIPS performance of the DTMF encoder and decoder is achieved using a buffered concept instead of a sample by sample concept. This reduces calling overhead. For the DTMF decoder, the speed critical Goertzel-DFT function has the 8 filters coded inline for fast execution. Due to an improved gain control the typical overflow check within the goertzel routine could be avoided, which improves the speed of the algorithm. Additionally the computation of 2nd harmonics energy information was reduced to only the two necessary frequency bins achieving additional speed improvement. However, in order to meet the Bellcore specifications some modifications to the straightforward Goertzel algorithm had to be made. This includes the overlap-and-save scheme combined with additional frequency bins for column tones. These improvements required additional MIPS. Table 3 and Table 4 summarize the benchmarks for the DTMF encoder as well as decoder.

Table 3. Speed and Memory Requirements for the DTMF Encoder

Module Name	Tasks Included	Program	Data (n channels)	Max. Cycles per 120 samp	MIPS
DTMF encoder	tone generation	198	161*n+24†	2143	0.143
HW/SW initialization I/O	ISRs Inits C-environment	351	–	–	–
TOTAL		**549**	161*n+24†	**2143**	**0.143**

† When the encoder is used together with decoder, data will be 25 · n + 24

Table 4. Speed and Memory Requirements for the DTMF Decoder

Module Name	C-Functions	Program	Data (n channels)	Max. Cycles per 120 samp	MIPS
DTMF decoder	Notch dial-tone			2205	0.147
	gain control			684	0.046
	Goertzel-DFT			8911	0.594
	DTMF checks			1878	0.125
	Copy overlap	669	318*n + 56	37	0.002
HW/SW initialization I/O	ISRs inits C-environment	351	–	–	–
TOTAL		**1011**	**318*n + 56**	**11510**	**0.914**

5 Performance

Two well-known tests to evaluate the performance of DTMF decoders are available through MITEL and Bell Communications Research (Bellcore), which both supply the associated test tapes and test procedures.

5.1 MITEL Tests

The DTMF tone decoder has been tested using the MITEL test procedures. For a preliminary test a set of files with the digitized signal data for the various MITEL tests was acquired through TI internal sources. The files essentially contain a subset of the signal data of the real MITEL test tapes. Since none of the files contained more than 32k words of data, the files were reformatted as ".inc"-files and then included in the source code. At link-time the data of a given test file was mapped into external memory of the TMS320C54xEVM occupying at maximum 32k words of data space. The code was then tested in real-time using the contents of the 32k-words test buffer as its input source. During this preliminary test the various decoder thresholds were set to proper levels.

The complete MITEL test was then executed according to the test procedures specified in the MITEL test document. A digital audio tape (DAT) recording of the test tapes was used as input signal source. The decoder was executed on the TMS320C54x EVM utilizing a TMS320AC01 analog interface to convert the incoming signal into the digital domain. The MITEL test essentially has two sections. The first section measures the DTMF tone decoder in terms of Recognition Bandwidths (RBW), Recognition Center Frequency Offset (RCFO), Standard Twist, Reverse Twist, Dynamic Range (DR), Guard time and Signal-to-Noise Ratio (SNR). The second section, called "talk-off test", consists of recordings of conversations on telephone trunks made over a long period of time and condensed into a 30 minute period. In this section the decoder's capabilities to reject other sources such as speech and music is measured. MITEL specifies a maximum of 30 responses of the DTMF decoder as acceptable speech-rejection.

Table 5 summarizes the MITEL test results. With the exception of the recognition bandwidth results for the low-band frequencies, the DTMF decoder passes all the tests and exceeds the specifications. Recognition bandwidths are within the Bellcore specifications mainly achieved by the increased frequency resolution of the Goertzel DFT. The threshold settings for acceptable twists helped pass the twist test and exceed the specifications. The dynamic range of the decoder of 27dB is better than the specification. The decoder was able to correctly detect all the 1000 tone bursts for each of the given noise environments of −24dBV, −18dBV and −12dBV AWGN. When the decoder was exposed to the 30 minutes of speech and music samples, it did not respond a single time and exceeded the MITEL talk-off specification of 30 permissible responses by a significant margin.

As required by the Mitel and Bellcore specification the detector is capable of detecting DTMF tones in the presence of a strong dial-tone. A front-end cascaded bi-quad IIR with notches at exactly 350Hz and 440Hz was designed to eliminate dial-tone signal components before detection of DTMF tones.

Table 5. MITEL Test Results

BW Tests	Frequency	RBW%	RCFO%
Specification		1.5% < RBW < 3.5%	
Low band	697 Hz 770 Hz 852 Hz 941 Hz	2.7% 2.5% 2.4% 2.1%	0.05% 0.00% 0.05% 0.05%
High band	1209 Hz 1336 Hz 1477 Hz 1633 Hz	2.6% 2.3% 2.1% 1.9%	0.05% 0.05% 0.00% 0.05%
TWIST Tests	**Std Twist**	**Rev Twist**	
Specification	> 4 dB	> 8 dB	
DIGIT 1 DIGIT 5 DIGIT 9 DIGIT 16	6 dB 6 dB 7 dB 7 dB	9 dB 9 dB 9 dB 9 dB	
DR Tests	**Dyn Range**		
Specification	> 25 dB		
DIGIT 1 DIGIT 5 DIGIT 9 DIGIT 16	27 dB 27 dB 27 dB 27 dB		
Guard Time	**Pause Time**	**Tone Time**	
Specification		45 ms	recognition
DIGIT 1	>30 ms	>40 ms	recognition
Specification		<23 ms	rejection
DIGIT 1		<24 ms	rejection
SNR Tests	**Noise**	**Result**	
Specification	− 24 dBV		
DIGIT 1 DIGIT 1 DIGIT 1	− 24 dBV − 12 dBV − 18 dBV	passed passed passed	

Talk-Off Test	Decoder Responses	Result
specification	< 30	
result	2	very robust

5.2 Bellcore Talk-Off Test

Through TI internal sources the Bellcore series-1 Digit Simulation Test Tapes for DTMF receivers were available for testing. These tapes consist of six half-hour sequences of speech samples, designated parts 1 through 6, which are known to contain energy at or near valid DTMF frequency pairs. This test exhaustively measures the speech-rejection capabilities of DTMF receivers in telecommunication systems. There are over 50,000 speech samples including some music. It is estimated that the six parts of the series-1 tapes are equivalent to the exposure of one million customer dialing attempts in a local central office. In other words, exposing a DTMF receiver to all the speech samples in series-1, will produce the same number of digit simulations that the receiver would experience if it were exposed to customer speech and room noise present during network control signaling on one million calls. The Bellcore talk-off test is far more exhaustive than the MITEL talk-off test.

The test setup was identical to the one used for the MITEL talk-off. Table 6 summarizes the test results for the Bellcore talk-off. In the three hours of testing the decoder responded only in six cases to digit simulations. This is far less than the specifications require to pass the talk-off. The decoder proved to be very robust in terms of its speech-rejection capabilities.

Table 6. Bellcore Talk-Off Test Results

Test	Digits	Specification	Results
Part 1 through 6 (3 hours)	0 − 9	Per 1,000,000 calls < 333 responses	6 responses
Part 1 through 6 (3 hours)	0 − 9, *, #	Per 1,000,000 calls < 500 responses	6 responses
Part 1 through 6 (3 hours)	0 − 9, *, #, A, B, C, D	Per 1,000,000 calls < 666 responses	6 responses

6 Summary

DTMF tone encoding and decoding concepts and algorithms were described here in some detail. Further theoretical background is provided in the appendix. The DTMF encoder and decoder implementations were explained and the associated speed and memory requirements were presented. The DTMF tone decoder has been tested according to the MITEL and BELLCORE test specifications and the results are documented. It is important to note that the encoder and decoder was implemented as reentrant, C-callable functions, which facilitate setting up a multichannel DTMF decoder system. The code is modular and easy to integrate into any given telephony application. The decoder algorithm was greatly optimized to meet the test specifications as well as offer a very attractive MIPS count of around 1.1 MIPS per channel of generation and detection.

7 References

1. Proakis, J.G., Manolakis, D.G., *Digital Signal Processing*, Macmillan Publishing Company, New York, 1992, Pages 737-739.
2. Mock, P., *Add DTMF Generation and Decoding to DSP-µP Designs*, DSP Applications with the TMS320 Family, Vol. 1, Texas Instruments, 1989.
3. Ziemer, R.E., Tranter, W.H., *Principles of Communications*, Houghton Mifflin Company, Boston, 1995.
4. MITEL Technical Data, *Tone Receiver Test Cassette CM7291*, 1980.
5. Bell Communications Research, *Digit Simulation Test Tape*, Technical Reference TR-TSY-000763, Issue 1, July 1987.

Appendix A Background: Digital Oscillators

A.1 Digital Sinusoidal Oscillators

A digital sinusoidal oscillator may in general be viewed as a form of a two pole resonator for which the complex-conjugate poles lie on the unit circle. It can be shown that the poles of a second order system with system function

$$H(z) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{A.1}$$

with parameters

$b_0 = A \sin \omega_0$

$a_1 = -2 \cos \omega_0$

$a_2 = 1$

are exactly located at the unit circle. That is

$$p_{1,2} = e^{\pm j\omega_0} \tag{A.2}$$

The discrete-time impulse response

$$h(n) = A \sin((n + 1)\omega_0) \cdot u(n) \tag{A.3}$$

corresponding to the above second order system clearly indicates a clean sinusoidal output due to a given impulse input. We can therefore term our system a digital sinusoidal oscillator or digital sinusoidal generator.

For the actual implementation of a digital sinusoidal oscillator the corresponding difference equation is the essential system descriptor, given by

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 \delta(n) \tag{A.4}$$

where initial conditions y(−1) and y(−2) are zero. We note that the impulse applied at the system input serves the purpose of beginning the sinusoidal oscillation. Thereafter the oscillation is self sustaining as the system has no damping and is exactly marginally stable. Instead of applying a delta impulse at the input, we let the initial condition y(−2) be the systems oscillation initiator and remove the input. With this in mind our final difference equation is given by

$$y(n) = 2 \cos \omega_0 \cdot y(n-1) - y(n-2) \tag{A.5}$$

where

$y(-1) = 0$

$y(-2) = -A \sin \omega_0$

$\omega_0 = 2\pi f_0 / f_s$

with *fs* being the sampling frequency, *fo* being the frequency and A the amplitude of the sinusoid to be generated. We note that the initial condition *y*(−2) solely determines the actual amplitude of the sinewave.

Appendix B Background: Goertzel Algorithm

B.1 Goertzel Algorithm

Body Text for Definition:

As the first stage in the tone detection process the Goertzel Algorithm is one of the standard schemes used to extract the necessary spectral information from an input signal. Essentially the Goertzel algorithm is a very fast way to compute DFT values under certain conditions. It takes advantage of two facts:

(1) The periodicity of phase factors $\{w_N^k\}$ allows to express the computation of the DFT as a linear filter operation utilizing recursive difference equations

(2) Only few of the set of spectral values of an actual DFT are needed (in this application we have 8 row/column tones plus an additional 8 tones for corresponding 2nd harmonics)

Having in mind that a DFT of size N is defined as

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j\frac{2\pi}{N}km} \qquad \text{B.1}$$

we can indeed find the sequence of a one-pole resonator

$$y_k(n) = \sum_{m=0}^{N-1} x(m) e^{j\frac{2\pi}{N}k(N-m)} \qquad \text{B.1}$$

which has a sample value at n = N coinciding exactly with the actual DFT value. In other words each DFT value *X(k)* can be expressed in terms of the sample value at *n* = *N* resulting from a linear filter process (one-pole filter).

We can verify, that

$$\begin{aligned} X(k) = y_k(n) &= \sum_{m=0}^{N-1} x(m) e^{j\frac{2\pi}{N}k(N-m)} \\ &= \sum_{m=0}^{N-1} x(m) e^{j\frac{2\pi}{N}kN} e^{-j\frac{2\pi}{N}km} \\ &= \sum_{m=0}^{N-1} x(m) e^{j\frac{2\pi}{N}kN} e^{-j\frac{2\pi}{N}km} \qquad \text{B.3} \end{aligned}$$

The difference equation corresponding to the above one-pole resonator sequence (B.2), which is essential for the actual implementation, is given by

$$y_k(n) = e^{j\frac{2\pi}{N}k} y_k(n-1) + x(n) \qquad \text{B.4}$$

with y(−1) = 0 and pole location $p = e^{-j\frac{2\pi}{N}K}$. Being a one-pole filter, this recursive filter description yet contains complex multiplication, not very convenient for a DSP implementation. Instead we utilize a two-pole filter with complex conjugate poles

$p_{1,2} = e^{\pm j2\frac{\pi}{N}k}$ and only real multiplication in its difference equation

$$v_k(n) = 2\cos\frac{2\pi}{N}k) \cdot v_k(n-1) - v_k(n-2) + x(n) \qquad \text{B.5}$$

where $vk(-1)$ and $vk(-2)$ are zero.

Only in the Nth iteration a complex multiplication is needed to compute the DFT value, which is

$$X(k) = y_k(N) = v_k(N) - e^{-j\frac{2\pi}{N}k} v_k(N-1) \qquad \text{B.6}$$

However the DTMF tone detection process does not need the phase information of the DFT and squared magnitudes of the computed DFT values in general suffices. After some mathematical manipulation we find

$$|X(k)|^2 = y_k(N)\, y_k{}^*(N) \qquad \text{B.6}$$

$$= v_k{}^2(N) + v_k{}^2(N-1) - 2\cos(\frac{2\pi}{N}k) v_k{}^2(N) v_k{}^2(N-1) \qquad \text{B.7}$$

In short: For the actual DSP implementation equations (B.5) and (B.7) will be used retrieve the spectral information from the input signal *x(n)* for further evaluation. Note that equation (B.5) is the actual recursive linear filter expression, which is looped through for *n* = 0 .. *N*. Equation (B.7) is only computed once every *N* samples to determine the squared magnitudes.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265