An accurate Dual Tone Multiple Frequency Detector based on the low-complexity Goertzel algorithm

Fotis E. Andritsopoulos, Newton Bomeisel Cardoso, Gregory A. Doumenis, Yannis M. Mitsos, Lambros E. Sarakis

E-mail: `fandrit@telecom.ntua.gr`, `newton@cpqd.com.br`, `{Gregory.Doumenis, Yannis.Mitsos}@gdt.gr`, `ls6@telecom.ntua.gr`

Abstract— **This article presents a low complexity dual tone multiple frequency (DTMF) signal detector that meets the International Telecommunication Union (ITU) Q.24 DTMF standard while implemented on a general purpose Digital Signal Processor (DSP). The proposed DTMF detector is based on the Goertzel algorithms and is well suited for a multichannel implementation. Our solution offers increased accuracy and noise margins, while preserving the inherited advantages of the Goertzel algorithm (less memory requirements and computational complexity).**

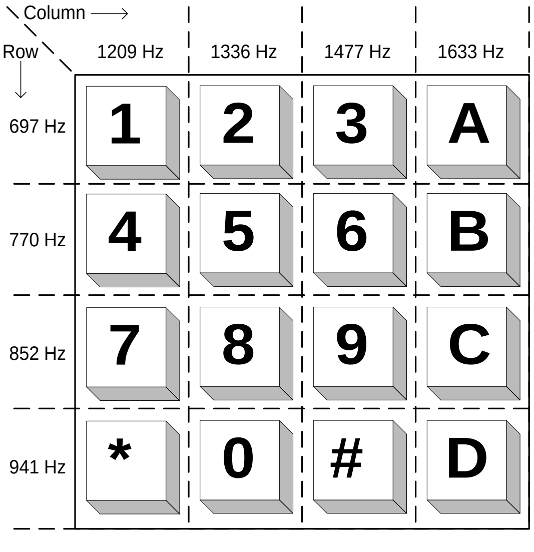
Keywords— **DTMF, detector, Goertzel, signal processing.**

I. Introduction

Dual Tone Multiple Frequency (DTMF) signaling [1] is used in telephone dialing, digital answer machines, interactive banking systems and in-band end to end PSTN signaling in general. DTMF signaling represents each symbol on a telephone touch-tone keypad (0-9,*,#) as a conjunction of two sinusoidal tones, as shown in Figure 1. When a key is pressed, a DTMF signal consisting of a row frequency tone plus a column frequency tone is transmited. Keys A-D are not on commercial telephone sets, but are used in military and radio signaling applications. The purpose of DTMF decoding is to detect DTMF encoded sinusoidal signals in the presence of noise. There is plethora of cost effective integrated circuits on the market that does this quite well. In many cases, the DTMF decoder IC[2] interfaces with a microcontroller implementing control and management functions. In such implementations the signal processing associated with the decoding is usually beyond the scope of the microcontroller's capabilities. So the designer must use the dedicated IC, or alternatively must upgrade the microcontroller to a digital signal processor (DSP), capable of performing the DTMF decoding in S/W. The latter implementation is cost effective because it reduces the overall component count (by eliminating the DTMF decoder IC). The detector presented in this paper implements a methodology based on the well-known Goertzel algorithm[3], providing an efficient way to implement a DTMF detector and decoder. Other implementations are also possible (e.g. DFT [4], NDFT[5]), but an implementation of the Goertzel algorithm with suitably modified filter coefficients has been tested and turned out to perform well in such applications, still requiring much less computational and memory resources. Existing implementation, however, suffers from reduced center frequency accuracy, often resulting into marginal conformance to the standards [1], [9]. Our approach improves both the detection accuracy and noise margin, thus producing a totally conformant solution.

The work presented on this paper is joint international applied research between the National Technical University of Athens (Greece), GDT (Greece) and CPqD Foundation (Brasil).

The paper provides an introduction to the algorithm, a description of the proposed S/W structure, an estimation of actual complexity, program and data memory usage and finally flexible test scenarios. The proposed DTMF detector has been implemented on a TMS320VC5402[1] DSP from Texas Instruments. For each input sample, the detector requires $O(M)$ multiplications and $O(M)$ additions for every M samples[6]. The proposed detector uses less than 1.5K words of program memory, $O(n)$ of DATA MEMORY and



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

Fig. 1. DTMF scheme for touch-tone dialing. When a key is pressed, two sinusoids at the row and column are added together.

[1] TMS320VC5402 is a low-power 16-bit fixed point DSP tailored to such kind of telecommunication applications. It operates at 100MHz offering maximum 100MIPS.

less than $O(n)$ MIPS. The factor n stands for number of simultaneous channels.

II. Proposed method

The DTMF tone detector has been implemented on the TMS320VC5402-100MHz DSP. The sampled data, as they are acquired from the A/D converter, are forwarded to the DSP via its serial port. The DSP executes the main algorithm in order to determine if the sampled data represent a valid tone.

The Goertzel algorithm is more efficient than the Fast Fourier Transform in computing an N-point DFT, if less than $2log_2N$ DFT coefficients are required[7]. In DTMF detection, we only need 8 instead of, for example, 128 DFT coefficients to detect the 8 possible tones, and then apply decision logic to choose the strongest touch-tone. Since DTMF signals do not encompass second harmonics, we must compute another 8 DFT coefficients for the second harmonics to detect the presence of speech or music (instead of valid DTMF signalling).

The Goertzel algorithm acts as an IIR filter that uses the feedback path to generate a very high Q bandpass filter where the coefficients are easily generated from the required center frequency. The most common configuration for using this technique is to measure the signal energy before and after the filter and to compare the two. If the measurements are similar, then the input signal is centered in the pass-band; if the output energy is significantly lower than the input energy then the signal is outside the pass band. The Goertzel algorithm is most commonly implemented as a second order recursive IIR filter, as shown in Figure 2. The Goertzel algorithm computes the $k-th$ DFT coefficient of the input signal $x[n]$ using the second-order filter

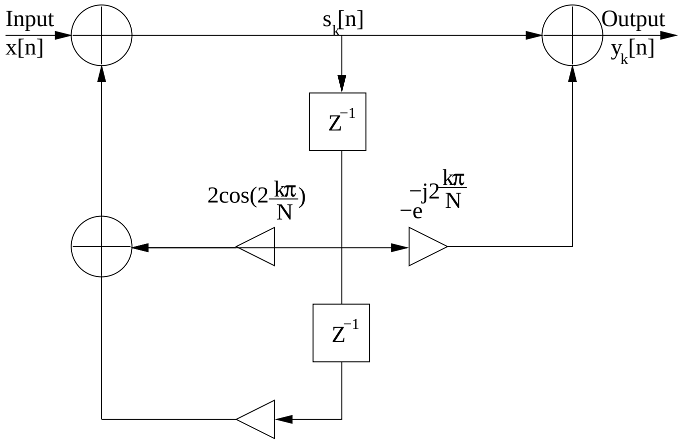


Fig. 2. Direct-Form Realization of the Goertzel Algorithm

$$s_k[n] = x[n] + 2cos(\frac{2k\pi}{N})s_k[n-1] - s_k[n-2] \quad (1)$$

$$y_k[n] = s_k[n] - W_N^k s_k[n-1] \quad (2)$$

$$k = N * \frac{f_{dtmf}}{f_{sample}} \quad (3)$$

DTMF	k
697	9
770	10
852	11
941	12
1209	16
1336	17
1477	19
1633	21

TABLE I
Frequencies and Filter Coefficients for $N = 102$ assuming k integer

DTMF	k	DTMF	k	DTMF	k
697	8.88	697+T	9.05	697-T	8.73
770	9.82	770+T	9.99	770-T	9.64
852	10.86	852+T	11.05	852-T	10.67
941	12.00	941+T	12.2	941-T	11.79
1209	15.42	1209+T	15.67	1209-T	15.16
1336	17.03	1336+T	17.32	1336-T	16.75
1477	18.83	1477+T	19.14	1477-T	18.52
1633	20.82	1633+T	21.16	1633-T	20.48

TABLE II
Frequencies and Filter Coefficients for $N = 102$ proposed by this implementation

where $x[n]$ is the input signal, $y[n]$ is the output signal, f_{sample} is the sampling frequency, f_{dtmf} is the DTMF frequency,

$W_N^k = -e^{-j2\frac{k\pi}{N}}$, $s_k[-2] = s_k[-1] = 0$. The k-th DFT coefficient is produced after the filter has processed N samples: $X[k] = y_k[n] \mid_{n=N}$

The key in an implementation is to run $s_k[n]$ for N samples and then evaluate $y_k[n]$. The computation for $s_k[n]$ takes one add ($x[n]-s_k[n-2]$) and one multiply-accumulate per sample. In DTMF detection, we are only concerned with the power of the k-th coefficient, $y_k[N]y_k^*[N]$.

Table I illustrates the coefficients needed for the computation of the DFT, as they are used by other implementations[8]. To be able to meet the acceptable bandwidth specifications[9] a modification of the algorithm departs from the true DFT and tunes frequencies exactly with the DTMF tone frequencies. Furthermore, we have calculated the values of the k for the acceptable tolerance of $T = 1.5\% + 2Hz$ (presented in Table II).

Table II also provides the values of ks for detecting frequencies which depart from the center frequency by an acceptable amount of $\pm(1.5\% + 2Hz)$[9] (Figure 3). The outputs of the Goertzel filters for these frequencies will be used to achieve a specified recognition bandwidth. According to the specifications the *reverse bandwidth* (RBW) for positive deviation from the center frequency should be between $1.5\% + 2Hz$ and 3.5%, and for negative deviation between $1.5\% - 2Hz$ and 3.5%.

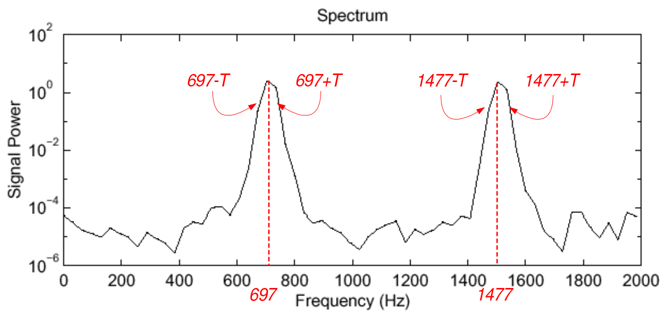


Fig. 3. Two additional measurement point for increased accuracy

III. Implementation structure

The proposed DTMF detector has been modeled and verified on *Matlab*© and has been implemented and verified on the TMS320VC5402 Texas Instruments DSP. In this section we present the implementation.

The incoming digital data fill a buffer of N octets. The actual size of the buffer is determined by the strictness of the test specifications; a value of $N = 102$ is a good balance between computational complexity and accuracy [10]. Determining the coefficients' value for a given tone frequency involves a trade-off between accuracy and detection time. These parameters are dependent on the value choosen for N. If N is very large, resolution in the frequency domain is very good but the length of time between output samples increases because the feedback phase of the Goertzel algorithm is executed N times before the feedforward phase is executed once. Once the buffer is full, the DSP task scheduler initiates the DTMF detection process.

An abstracted flowchart of the DTMF implementation is depicted in Figure 4. First, the content of the input data buffer is copied into an intermediate buffer for processing (double buffering). In this way there are no lost samples. The detection functions will then operate on the intermediate buffer; the input buffer is still capturing data (interrupt based). The gain control function attenuates strong input signals and protects the next functions from numeric overflows.

Next, the Goertzel filters are executed. Since the preceding gain control ensures that overflow cannot occur, overflow checking is removed and optimized loops allow fast execution. The outputs of the Goertzel function are the delay states of the 16 filters, which are collected in an array. On completion of the Goertzel function, the DTMF digit validation checks are invoked. The spectral information is computed from the filter delay states and collected in an energy template. For the next round of execution, the filter delay states are initialized to zero. The energy template is then 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 a second time for DTMF tone presence and executes all the digit validation tests. If the detected digit is the same that the previous one the digit information is stable. In the pause detection mode, DTMF tone detection is disabled and the decoder first has to await for a pause signal. With the successful completion of the digit validation tests, the valid digit (if any) is stored into the digit output buffer.

The implementation steps are presented in figures 5, 6.

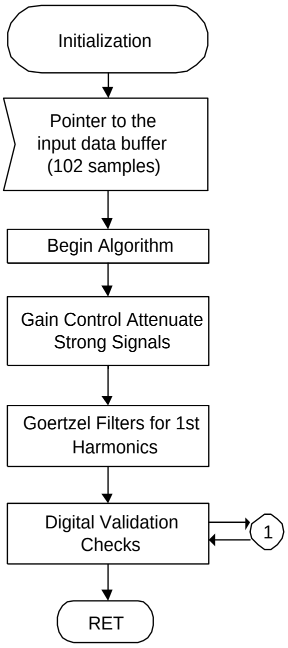

Initialization
Pointer to the input data buffer (102 samples)
Begin Algorithm
Gain Control Attenuate Strong Signals
Goertzel Filters for 1st Harmonics
Digital Validation Checks
1
RET

Fig. 4. Flowchart of the DTMF Decoder Implementation (1 of 5)

The gain control function attenuates strong signal inputs and protects the next functions from accumulator overflows. The attenuation factor takes into account the energy of the signal as derived from the sum of the discrete square values. Thus, the energy of the real sequence $x(n)$ is equal to:

$$E = \sum_{k=0}^{k=n} |x[k]|^2$$

In the next step, the 8 center DTMF frequencies must be investigated. For this purpose we execute 8 Goertzel algorithms with the parameters defined in Table II. Each Goertzel algorithm returns the value of the estimated energy for a particular frequency. It thus combines the recursive algorithm for implementing the Goertzel filter and the energy computation. Next, the outputs of the Goertzel algorithm functions are scanned and the 2 maximum outputs (one from the lower frequency band and one from the upper band) are collected. If it is necessary (depends on the test data), 2 more algorithms can be added to test the presence of second harmonics (i.e. differentiate speech from DTMF).

For the frequencies that produce these maximum outputs, 2 additional Goertzel algorithms (for each one) are executed. The Goertzel algorithms use the coefficients defined in table I in the 4th and 6th columns. Each check searches for signals in the bounds of the acceptable frequency range ($+1.5\% + 2Hz$ and $-1.5\% - 2Hz$). The outputs at these frequencies provide information regarding the placement of the signal's center frequency with respect to the specified RBW. If the ratio of these two outputs for each frequency is within a certain range we conclude that the DTMF frequencies obey the specified recognition bandwidth and we continue with further checks.

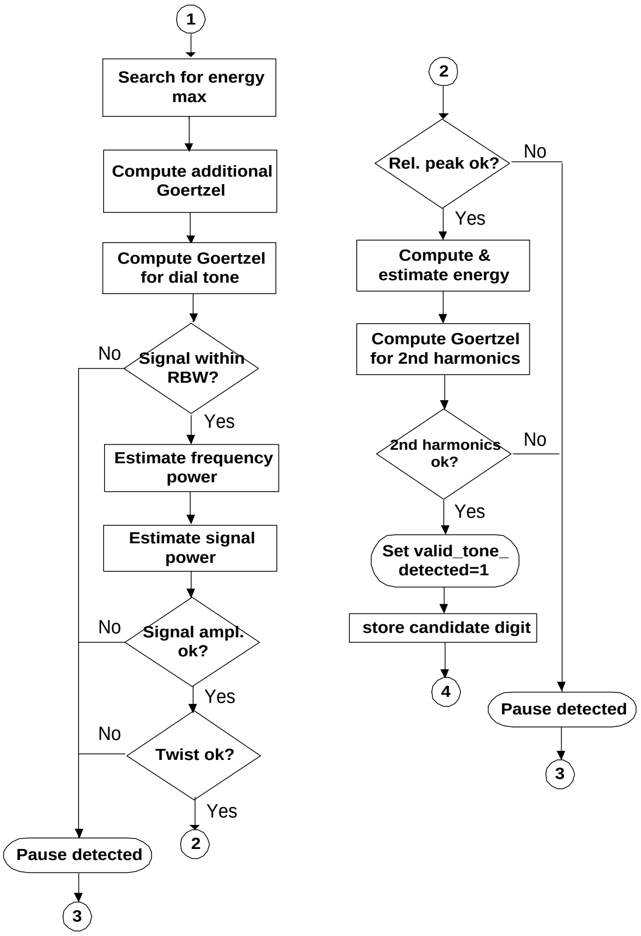


Fig. 5. Flowchart of the DTMF Decoder Implementation (2 & 3 of 5)

It is a challenging task to estimate the power of each DTMF frequency. If the input signal contains an exact pair of DTMF frequencies, then the output of the Goertzel algorithm represents its power. Besides that, the DTMF decoder must also tolerate the presence of a dial tone of maximum power of -3dBm. 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[2] but it is not required by DTMF standards. Such a strong signal adds a significant amount of power especially to the low-band DTMF frequencies 697, 770, 852 and 941Hz. The strength of the dial tone is investigated through an appropriately tuned Goertzel filter. The amount of interference is subtracted from the power of the low-band DTMF frequencies. If one frequency departs from the exact DTMF frequency, estimation is given by its power. If the signal contained only two frequencies, its power would be the sum of the frequencies' power. In this way we come up with an estimation of the ratio of the DTMF tone energy and the signal energy.

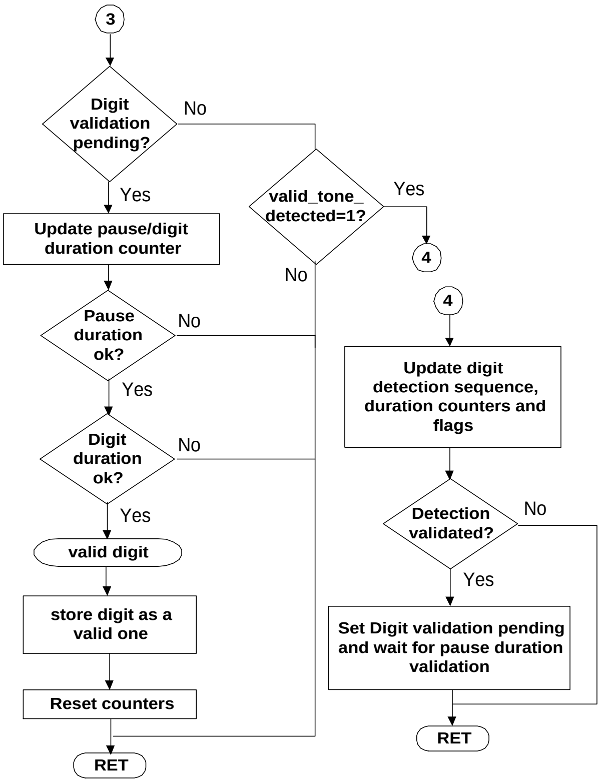

3
Digit validation pending?
No
Yes
Update pause/digit duration counter
Pause duration ok?
No
Yes
Digit duration ok?
No
Yes
valid digit
store digit as a valid one
Reset counters
RET
valid_tone_ detected=1?
Yes
No
4
4
Update digit detection sequence, duration counters and flags
Detection validated?
No
Yes
Set Digit validation pending and wait for pause duration validation
RET

Fig. 6. Flowchart of the DTMF Decoder Implementation (4 & 5 of 5)

The amplitude, twist and relative peak checks are performed to guarantee conformance with the specifications[9]. Then, the estimated energy is subtracted from the energy of the signal. A large value of this difference denotes that considerably large power lies somewhere in the signal spectrum, outside the DTMF signals. If the signal passes all checks the corresponding DTMF digit is stored in a buffer and is described as a candidate for a valid digit.

To ensure a minimum tone duration of 50ms[9], the same algorithm must run on 4 consecutive input buffers ($4 * 102/8000 = 4 * 12.75ms = 51ms$) before we decide on a valid tone. If the same tone (corresponding to a valid pair of DTMF frequencies) is detected 4 times in a raw, we conclude that the corresponding digit has valid tone duration and does not have to be detected again inside the duration of the digit. However, in order to compensate for a signal interruption (max. 10ms) we allow for the possibility of one failure in these four tests (Figure 7).

Figure 7 illustrates several different cases that can occur in the detection sequence, prior to digit validation. It is a common case that the tone is not present for the whole duration of an input buffer, especially at the beginning (or/and the end) of the DTMF tone active period. In this case, the energy of the DTMF tone that will be detected by the Goertzel algorithm will be less than the energy that will be detected for the same tone in the next buffer, assuming

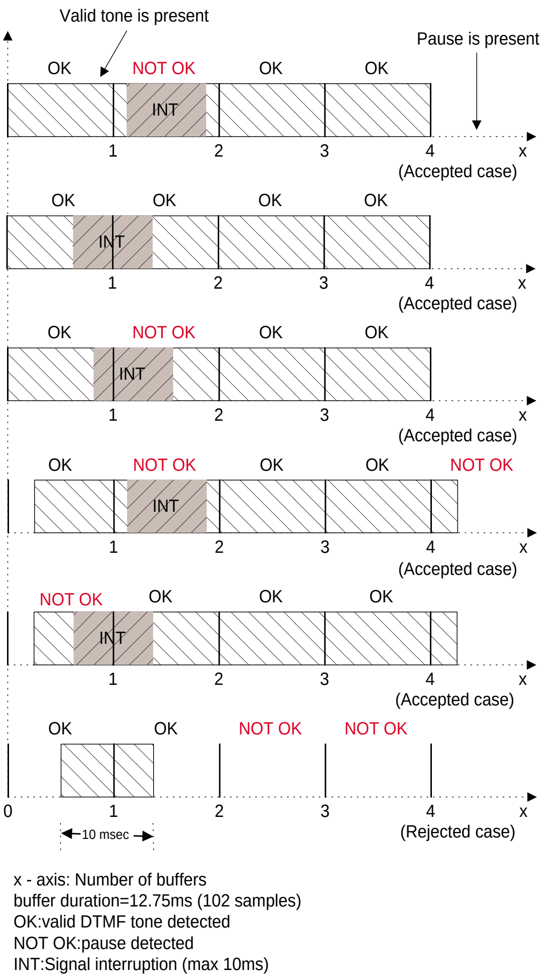

Valid tone is present
Pause is present
OK
NOT OK
INT
1
2
3
4
x
0
(Accepted case)
(Rejected case)
10 msec
x - axis: Number of buffers
buffer duration=12.75ms (102 samples)
OK:valid DTMF tone detected
NOT OK:pause detected
INT:Signal interruption (max 10ms)

Fig. 7. Several detection sequences

MODULE NAME	C FUNCTION	PROGRAM MEMORY	DATA MEMORY n channels
DTMF Decoder (buffered)	C-overhead Gain Control Goertzel-DFT DTMF checks	$< 1.5k$	Approx. $260n + 100$
HW/SW Initialiazation I/O	ISRs inits C-environment	< 500	-
TOTAL		$< 2.0k$	$260n + 100$

TABLE III
Memory requirements

no signal interruption. If the tone has a power level of -A dBm and is present for the half duration of a buffer, it will be detected as a tone having a power level of -(A+3) dBm. This is not a problem even if A=25 dBm, since the dynamic range of the detected tone must be >25dB. The last case in Figure 7 shows why two consecutive OKs can not guarantee the minimum tone duration.

When no DTMF tone is detected, it is assumed that either the digit goes into the pause period (digit validation pending) or that the signal has been interrupted. To ensure the minimum pause duration of 50ms, the algorithm should not detect a valid DTMF tone inside the next three input buffers. Then, if the down-counter for the digit minimum duration (120ms) is not zero (when zero, the pause duration criterion has been achieved), the algorithm should not detect a valid tone until the counter is zeroed. After that, the digit is finally validated. This counter gets an initial value of 9 or 10 (10.120ms/12.75ms) after the first detection of a valid DTMF tone (not digit) following a pause period. It is decreased by one after the completion of the detection algorithm and before the processing of a new buffer begins.

The algorithm illustrated in Figures 4, 5 and 6 passes through all necessary steps for a digit acceptance, including duration validation checks according to the specifications. These tests are mainly implemented in steps 3 and 4. If there is no need for duration validation checks, the algorithm as implemented in Figures 4 and 5 is sufficient. Then, step 3 can be replaced by a function that places a pause indication (e.g. the value -1) in a circular buffer containing the history of the detection sequence. Similarly, step 4 can be replaced by a function that places a digit indication (e.g. the value 6 for the digit 6 and the value 10 for the digit A) in the same circular buffer. Assuming a buffer of 100 words we can retain a history of 100 detected signals representing valid tones and pauses (e.g. PP555555PPPP777777P...). In this way the algorithm only detects tones. The external digit validation routine will use the circular buffer and the pointer representing the last input, and decides on the presence of a digit.

IV. Performance evaluation

The memory and processing requirements are summarized in Table III and Table IV. The values are slightly larger than those presented in [8]. This is mainly due to the fact that two additional Goertzel algorithms are executed for every one of the two outstanding frequencies and one Goertzel algorithm is executed for the dial tone. This fact increases the processing required for the Goertzel-DFT by 31%. It also increases the MIPS for the DTMF checks. The program memory is also increased due to the code for the additional $2 * 8 + 1 = 17$ Goertzel algorithms. It is assumed that all eight Goertzel algorithms are executed for the second harmonics. If eventually the tone validation routine needs only two of them, the figure for the MIPS of the Goertzel-DFT routine can be decreased. Thus, the designer is presented with the option to integrate a low complexity function (where processing resources are scarce), or to integrate a highly capable functions (where accuracy and conformance to the standards are imperative).

The following table summarizes the performance of the algorithm as it is implemented in Matlab. Deviations are expected due to the finite precision of the computations in the embedded host (in contrast with the almost "infinite" precision of the computations in Matlab). Attention has been paid to derive a recognition bandwidth that is close to the specifications. This RBW can be made looser or tighter simply by changing the appropriate parameters in

MODULE NAME	**C FUNCTION n channels**	**MAX CYCLES**	**MIPS per 102 samples**
DTMF Decoder (buffered)	C-overhead Gain Control Goertzel-DFT DTMF checks	$< n200$ $< n1000$ $< n12000$ $n * 1000$	$< n0.016$ $< n0.08$ $< n0.94$ $n * 0.08$
HW/SW Initialiazation I/O	ISRs inits C-environment	-	-
TOTAL		$< n14200$	$< n1.12$

TABLE IV

Processing requirements

Bandwidth test	**Frequency**	**RBW%**
Specification		$T < \lvert RBW \rvert < 3.5\%$
	697	2.6%
	770	2.4%
	852	2.8%
	941	2.7%
	1209	2.8%
	1336	2.5%
	1477	2.5%
	1633	2.1%
Twist test	**STD Twist**	**REV Twist**
Specification	$< 4dB$	$< 6dB$
All digits	$< 5dB$	$< 7dB$
DR test	**DYN Range**	
Specification	$> 25dB$	
All digits	$> 25dB$	

TABLE V

Test results

where, $T = 1.5\% + 2Hz$.

the RBW check routine. The noise threshold has been set to -24dBV and neither dial tone nor any other interference (e.g. speech) is present. The following 3-D figure illustrates the results of the bandwidth test for the digit 3 (assuming the presence of white noise) with power of the low frequency equal to -24dBm, power of the high frequency equal to -28dBm and a standard twist equal to 3dB.

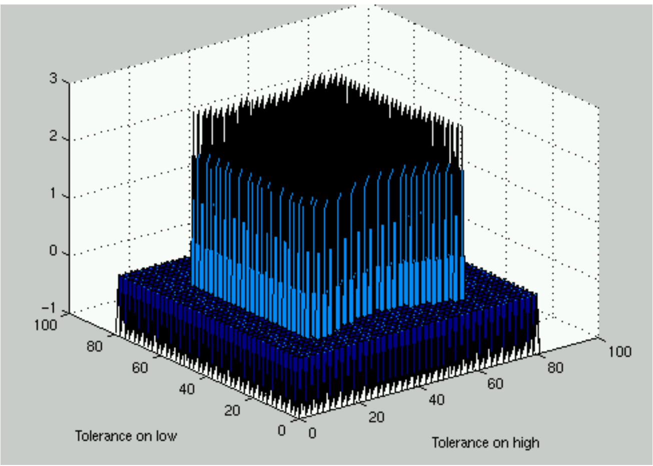


Fig. 8. Test results for the RBW of digit 3

The z-axis represents the output of the detection algorithm, which returns the value -1 for no valid digit and the value 3 for the detected digit 3. The x and y axis represent deviations from the DTMF frequencies 697Hz and 1477Hz respectively. The value 0 on the x-axis corresponds to a signal having a low frequency component at 697 - 4%. The value 80 on the x-axis corresponds to a signal having a low frequency component at 697 + 4%. The value 40 corresponds to the exact DTMF frequency at 697Hz. The same scaling holds also for the y-axis.

V. Conclusions

In this paper a robust implementation of a DTMF decoder was presented. There is plethora of algorithms for DTMF decoding. Still, most of them requires significant computacional power as well as memory, to be executed. The Goertzel algorithm gives an alternative to the latter problems by implementing an IIR filter for the centered DTMF frequencies to be detected instead of examining the whole spectrum. The advantage of our implementation is the significant accuracy provided. Whenever a frequency is detected, two additional Goertzel algorithms are executed in the frequencies $-(1.5\% + 2Hz)$ and $+(1.5\% + 2Hz)$. The overhead, caused by the extra computations, was calculated to be ≈ 0.2 MIPS per channel. Thus, for 32 channels an accumulative overhead of 6 MIPS is introduced. Furthermore in our approach, a function for the Dial-Tone frequency tracing is proposed to provide even better detection results. The latter is not needed in case where notch filters are used for the suppression of the Dial-Tone frequency.

The results of this implementation have been calculated using *Matlab*© and verified on a test board. The decoder is fully-compliant with the specifications of the ITU-T.

References

[1] ITU Blue Book, Recommendation Q.23: *Technical features of push-button telephone sets,* 1988

[2] 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.

[3] G. Goertzel, *"An Algorithm for the evaluation of finite trigonometric Series",* Amer. Math. Monthly, Vol.65, Jan. 1958, pp. 34-35

[4] M. D. Fedler, 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] 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.

[6] P. Mock, *"Add DTMF generation and decoding to DSP- µp designs,"* EDN, vol 30, pp. 205-220, Mar. 1985.

[7] A. V. Oppenheim and R. W. Schafer, *"Discrete-Time Signal Processing"*, Prentice-Hall, 1990.

[8] G. Schmer, *"DTMF Tone Generation and Detection, An implementation using the TMS320C54x,"* Texas Instruments 1997.

[9] ITU Blue Book, Recommendation Q.24: *General recommendations on telephone switching and signaling,* 1989.

[10] G. L. Smith, *"Dual-Tone Multifrequency Receiver Using the WEDSP16 Digital Signal Processor",* AT&T Application Note.

Fotis Andritsopoulos was born in Piraeus, Greece, in 1976. He received the Degree in Electrical and Computer Engineering from the Democritus University of Thrace in 1999. His Diploma thesis "Wireless Network on ISM band" received a prize of distinction from the Technical Chamber of Greece and it also awarded from Ericsson with the 2nd "Award of excellence in Telecommunications". Since 1999 he is a PhD candidate in the Telecommunications Laboratory of the National Technical University of Athens (NTUA). He has worked in the design and in implementation of various applications on FPGAs and software development for digital signaling processing on DSPs. His expertise is in h/W design and embedded system design. His interests are in the area of Communication Protocols, development of Communication Systems, IP Networks, Local and Wide Area Networks.

Newton Bomeisel Cardoso was born in Jundiai, Brazil, in 1962. He graduated in EE at University of Campinas (UNICAMP), Brazil, in 1985. Since 1985 he works with development in telecommunications, mainly with H/W and S/W designs of voice and data user equipments. He has got specialization in Digital Signal Processing at University of Campinas (UNICAMP), Brazil.

Gregory Doumenis was born in Athens in 1966. He received his Diploma in EE (1990) and his PhD in Broadband communication systems (1994) from the Electrical Engineering Department of NTUA. He has worked in a number of research and commercial projects as project leader. He participated in several R&D projects (RACE, ACTS and ESPRIT) with leading companies within Europe in the area of telecom networks and hardware and software systems development. Currently, he is chief technical officer of GDT SA. Gregory Doumenis has gained expertise in studying and researching telecommunication systems and has published nine papers in the area of broadband telecommunications and high performance computing. His research interests include: Digital systems architectures, formal design techniques, high performance computing and networking, broadband local and wide area networks, communication protocols, real time algorithms and systems, multimedia applications, video coding and transmission.

Yannis Mitsos was born in Athens, Greece, in 1976. He received the Degree in Electrical and Computer Engineering from the Democritus University of Thrace in 1999. His Diploma thesis was focused on the implementation of control algorithms using digital signal processor interworking with FPGAs. He is now design engineer in GDT SA and a PhD candidate in the Telecommunications Laboratory of the National Technical University of Athens (NTUA). His expertise is in H/W design and embedded system design. His interests are in the area of Communication Protocols, development of Communication Systems, IP and ATM networks.

Lambros Sarakis was born in Athens, Greece, in 1973. He holds a Diploma in Electrical & Computer Engineering from the National Technical University of Athens (NTUA) and an MSc in Communications and Digital Signal processing from Imperial College, University of London. He is currently a Ph.D. candidate at the Telecommunications Laboratory of the NTUA. His research interests include S/W design for DSP applications, H/W and S/W design for embedded systems, network design and evaluation, and QoS in packet networks.