Tone Event Detection in Packet Telephony Using the StarCore™ SC140 Core

By Lúcio F. C. Pessoa, Wen W. Su, Ahsan U. Aziz, and Kim-chyan Gan

This application note is a continuation of the application note AN2384/D [1], which presents the use of Teager-Kaiser (TK) energy operators for detecting multi-frequency tones with high accuracy and low cost. Key concepts presented in [1] are reused in this discussion, but important new processing blocks are added in order to handle a larger set of signaling tones, which we call tone events. This document describes a low-complexity tone event detection architecture that is both robust and suitable for packet telephony systems with high channel density. The proposed architecture is composed of:

- Multiple dependent tone detectors.
- A tone indication unit for selectively enabling the multiple dependent tone detectors.
- A single tone detection control unit with error correction capability.
- Common tone detection decision logic, regardless of the type of modulation used.
- Multi-stage tone detection architecture to minimize complexity.
- A multi-component tone detector that is independent of magnitude modulations [1].
- An efficient phase detection method that is independent of the carrier frequency offset and can function as a robust frequency and magnitude detector.

This document presents an efficient algorithm for faster and more reliable detection of tone events modulated by various modulation schemes.

CONTENTS

freescale™ semiconductor

1 Tone Event Detection Basics

Tone event detection is an important application in digital telephony systems that use shared data and voice communications over the telephone network. The International Telecommunication Union (ITU) has developed standards for data transmission (V-series modem signals) over packet networks [2–4] that define inter-operation between Public Switched Telephone Network (PSTN) and Internet Protocol (IP) networks. Prior to or during data transmission, a series of tone events are exchanged, so that when a tone event is detected, specific actions must be taken to initiate and maintain proper data communication over the IP network.

A tone event can include a single tone or a combination of multiple tone segments with different modulation schemes. For example, a tone event may include a single frequency tone with an on/off amplitude modulation (AM). Another tone event may include a dual frequency component tone followed by a single frequency tone. Yet another tone event may include a single frequency tone with periodic phase changes. Furthermore, these tone events may include more than one frequency component per segment of the signal with the same tone characteristics. The challenge is to detect tone events reliably according to different modulation schemes, which may include magnitude, frequency, phase, or any combination thereof.

The large number of signaling tones transmitted over digital telephony systems demands different types of tone detectors to identify predetermined tone events. However, multiple independent tone detectors running in the system can result in high complexity and possible event detection errors. One traditional approach to tone event detection is based on Goertzel Filters [5], which efficiently estimates the Fourier transform of the signal at specific frequencies. However, depending on the range of frequency values, the number of components, or timing, the required processing and memory cost can spiral upwards as detection rates slow down. Furthermore, the use of Goertzel Filters to detect tones with low-frequency components (as compared to the sampling rate) may require a large number of samples, thus increasing processing time.

1.1 Tone Event Detector Architecture

The top-level structure of the tone event detector presented in this document is illustrated in **Figure 1**.

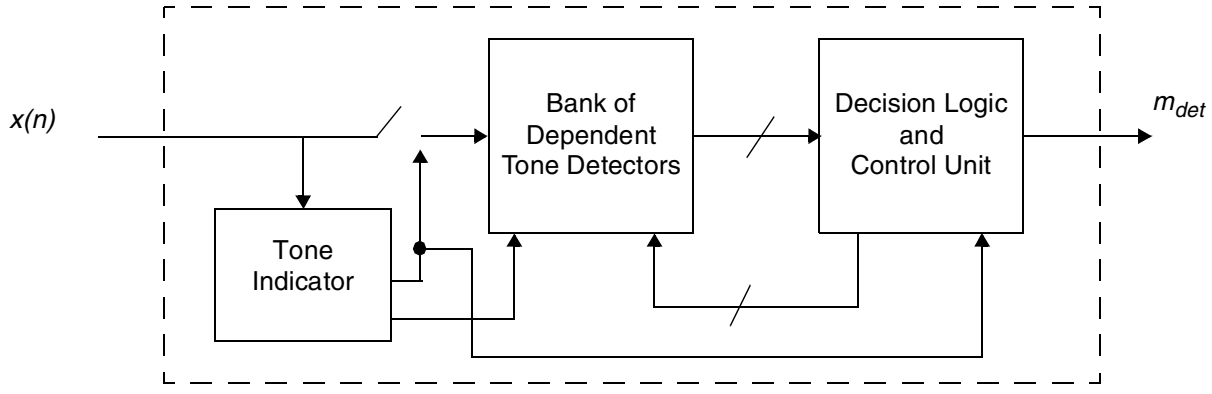


Figure 1. Top-Level Structure of the Proposed Tone Event Detector

A tone indicator selectively enables a bank of tone detectors only when the presence of a tone is indicated. If the presence of a tone is not indicated, no tone detection is performed. This approach significantly reduces the complexity of tone event detection because it uses a simplified method to indicate the tone presence and enable the more complex tone detection algorithms only when needed. Furthermore, the decision logic and control unit can use multiple tone event features to perform error correction efficiently when it detects tones with conflicting specifications.

Unneeded processing blocks are selectively disabled depending upon the characteristics of the input signal. The underlying state machine controlling tone event detection defines the multiple-stage processing flow shown in **Figure 2**. This flow includes (1) scanning the signal with a tone indicator, (2) identifying tone characteristics in one or more stages, and (3) detecting a tone event with the identified tone characteristics. The *fi*, *fd,* and *ID* control signals represent tone indication flag, tone detection flag, and tone identification tag, respectively.

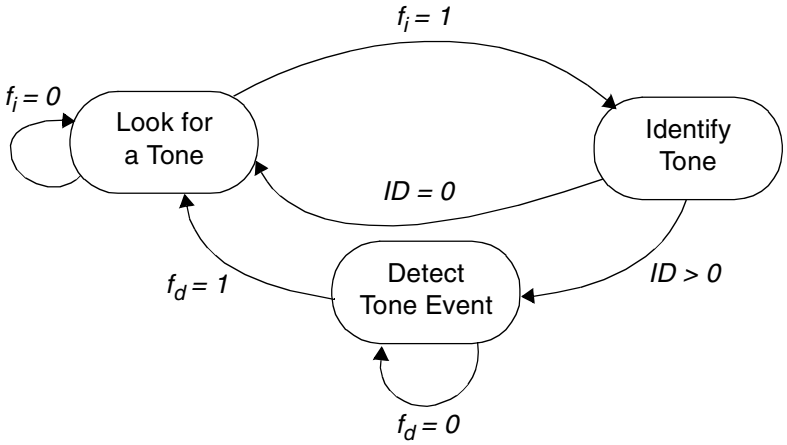


Figure 2. State Diagram of Multi-stage Detection Flow

A power estimator (see **Figure 3**) searches for narrow-band input signals, $x(n)$, that produce an output signal, $P(n)$, with small variance so that a tone is indicated when a small variance power signal is detected. This power estimator generates $P(n) = aP(n-1) + (1-a) \mid x^2(n-k) - x(n)x(n-2k) \mid$.

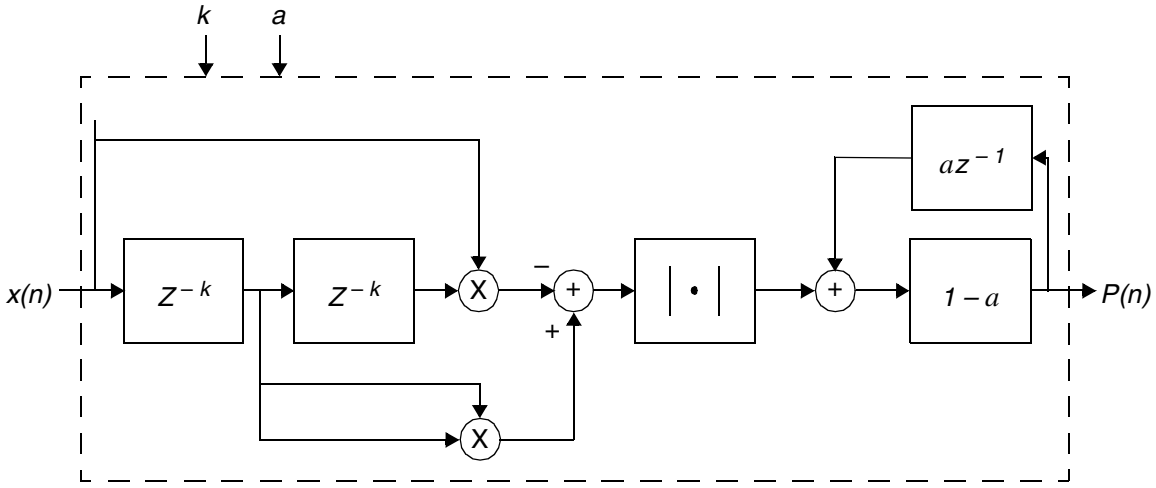


Figure 3. Power Estimator for Tone Indication

To identify a tone, a set of dependent tone detectors is enabled according to the frequency characteristics of the input signal. This set of detectors includes either a single component frequency detector, a single component phase detector, or a multi-component frequency detector. The methods for frequency detection are similar to the approach described in [1]. The subtle difference is that a first stage of tone indication is employed so that frequency/phase detection occurs only if a tone is successfully indicated in the input signal.

1.2 Phase Detection With Frequency Offset Compensation

To detect modulated-in-phase tone events, an efficient method of single-component phase detection is proposed. Since an input modulated-carrier signal may contain a frequency or phase offset, receivers of modulated-carrier data signals typically contain frequency tracking loops. These loops are useful because frequency must be adjusted accurately for differential coherent or incoherent reception. In addition, a Costas or other phase-locked loop for coherent reception may require frequency aiding for acquisition purposes. **Figure 4** depicts a frequency-tracking loop in its simplest form. This loop consists of a frequency-difference detector (FDD) [6], a loop filter, and a voltage-controlled oscillator (VCO).

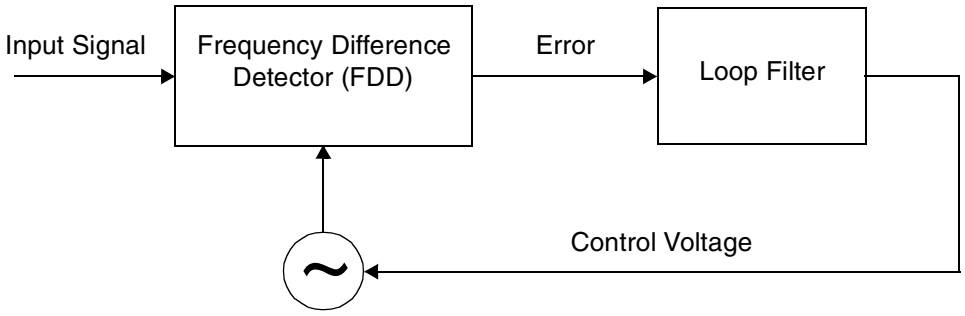


Figure 4. Classical Example of a Frequency Tracking Loop

There are various frequency compensation techniques used with different types of signaling schemes. Although the choice of frequency/phase offset detection and compensation scheme depends on the modulation scheme, the techniques can be generalized to data aided (DA), non-data aided (NDA), decision directed (DD), timing aided, and timing and data aided.

Most non-data and non timing-aided techniques have a fairly large frequency acquisition range but cannot achieve a low error variance for the frequency estimate. A technique proposed in [7] can compensate for frequency offset on the order of the symbol rate, but this feed-forward scheme requires a long acquisition time (200 bits) to achieve a low variance. Other schemes achieve a low variance but the estimation range is small. Most closed loop techniques use frequency difference detectors (FDD) [6]. An FDD measures the difference between the incoming carrier frequency and the local reference. Although compensation methods depend largely on the modulation scheme and the specific application, frequency difference detectors are the basic building-blocks for some frequency tracking loops in digital links. The Costas loop, a widely used method for carrier tracking, relies on feedback concepts related to a phase-lock loop (PLL) and offers an inherent ability to self-correct the phase (and frequency) of the recovered carrier. Its main disadvantage is loop settling time. Many variants of Costas loops address specific performance/complexity issues.

The proposed single-component phase detector depicted in **Figure 5** can also be used for frequency and magnitude detection. The input signal $x(n)$ is processed as follows:

1. Generate a sinusoidal signal $w(n) = 2c_0(n-1)w(n-1) - w(n-2)$ to track the input carrier frequency.
2. Select a sample delay:

$$n_0 = \left\lceil \pi / 2\Omega_0 \right\rceil$$

   where Ω_0 is the normalized nominal frequency of the signal detected.
3. Define $X(n) = x(n) + jx(n-n_0)$, $W(n) = w(n) + jw(n-n_0)$ and compute $Z(n) = X(n)W(n)^*$.
4. Compute the received signal $R(n) = bR(n-1) + (1-b)Z(n)$, $0 < b < 1$.

5. Estimate the phase offset $E(n) = R(n)R(n-1)^*$.
6. Adjust $c_o(n) = c_o(n-1) + \mu\, Imag\{E(n)\}$.

Notice that $c(n)$ and $s(n)$ can be used for phase detection, for example, arctan$[s(n) / c(n)]$). You can use $c_o(n)$ for frequency detection, for example, $cos(\Omega)$. This structure is related to [9] and [10], but it presents some unique advantages. Using ideas from [1], the signal magnitude can also be estimated as a function of the power estimator $P(n)$ and the coefficient $c_o(n)$ by simply estimating the ratio $P(n) / [1 - c_o(n)^2]$.

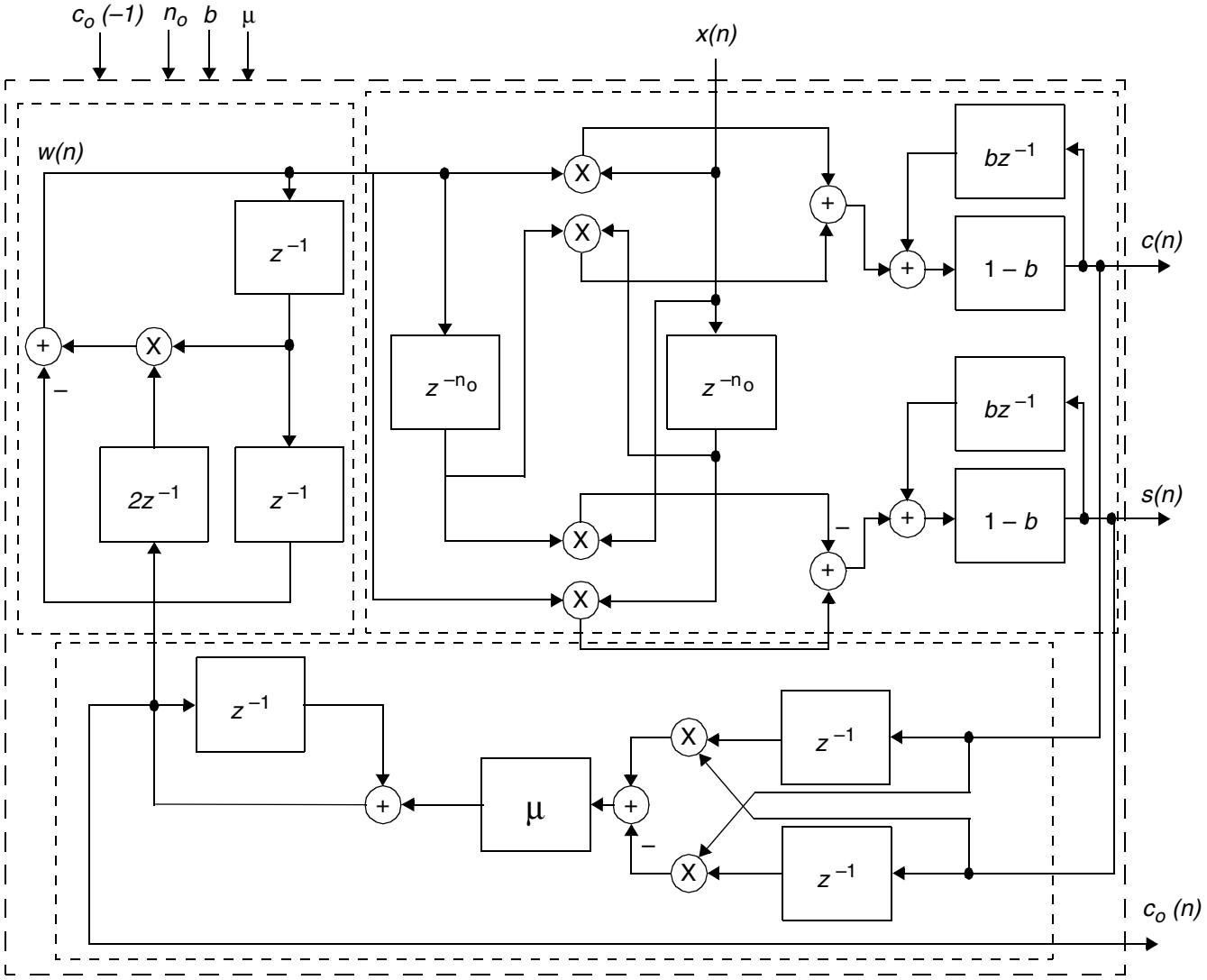


Figure 5. New Architecture for Phase/Frequency/Magnitude Detection of Single-Component Tones

1.3 Summary of Theoretical Results

The main theoretical results for tone event detection are summarized as follows:

1. Any single frequency tone is mapped to a constant value via the modified TK energy operator:

$$\Psi_k[x(n)] = x^2(n-k) - x(n)x(n-2k) = A^2 sin^2(k\Omega)$$

where $\Omega = 2\pi f / f_s$, f is the tone frequency and f_s is the sampling frequency, $k = 1, 2, \ldots$ ($k = 1$ in the original definition of the TK operator).

If the tone frequency is known, the signal power can be estimated as $P[x(n)] = [0.5 / \sin^2(k\Omega)]\ \Psi_k[x(n)]$, where the frequency dependent correction factor is pre-computed.

2. For $k = (l_1 - l_2) / 2$, where l_1 and l_2 are two delay lags, the tone frequency and amplitude values are indirectly estimated via the following ratios of TK operators:

$$\rho_\Omega = \frac{\Psi_k\left[\frac{1}{2}(x(n-l_1) + x(n-l_2))\right]}{\Psi_k[x(n)]} = cos^2(k\Omega)$$

$$\rho_A = \frac{\Psi_k[x(n)]}{1-\rho_\Omega} = A^2$$

3. For $x(n) = Acos(\Omega n + \phi(n))$, $w(n) = 2c_0(n-1)w(n-1) - w(n-2)$, $w(-1) = c_0(-1) = cos(\Omega)$, $w(-2) = 2[c_0(-1)]^2 - 1$ and $n_0 = \left\lceil \pi / 2\Omega \right\rceil$:

$$c(n) = ac(n-1) + (1-a)[x(n)w(n) + x(n-n_0)w(n-n_0)] \approx A\cos(\phi(n))$$

$$s(n) = as(n-1) + (1-a)[x(n-n_0)w(n) - x(n)w(n-n_0)] \approx A\sin(\phi(n))$$

$$c_0(n) = c_0(n-1) + \mu[c(n-1)s(n) - c(n)s(n-1)],\ 0 < \mu < 1$$

$$\rho_\phi = \frac{s(n)}{c(n)} \approx \tan(\phi(n)),\ 0 < \alpha < 1$$

For phase demodulation, the signal phase is conveniently estimated by:

$$\hat{\phi}(n) = \varphi(n)\ \frac{\pi}{2}\text{, where } \varphi(n) = 1 + 2u(s(n)) + \{1 - 2[u(c(n)) + u(s(n))]\}u(|c(n)| - |s(n)|)$$

and $u(\cdot)$ is the standard step function.

4. Multi-component tones

$$x(n) = \sum_{l=1}^{N} A_l^{(m)} \cos(\Omega_l^{(m)} n + \phi_l),\ m = 1, \ldots, M$$

are decomposed via adaptive notch filters of the form:

$$H_l^{(m)}(z) = \Gamma_l^{(m)} \prod_{i \neq l} \frac{1 - b_i^{(m)} z^{-1} + z^{-2}}{1 - r b_i^{(m)} z^{-1} + r^2 z^{-2}},\ l = 1, \cdots N$$

5. Inverse functions are estimated using third-order polynomial approximation. More specifically,

$$f(x) = \frac{1}{2x} \approx p(x) = 2^3 \{a_3 + x\,[a2 + x(a_1 + x a_0)]\}, \frac{1}{2} \leq x \leq 1$$

a_0	– 0.224822998046875
a_1	0.669525146484375
a_2	– 0.73565673828125
a_3	0.35321044921875

Therefore, a ratio $q = N / D$ is estimated by first normalizing the denominator to the range between 1/2 and 1, so that $q \approx N\,p(D')2^{b+4}$, where D' = D$2^b$ and b is the corresponding number of leading bits of the normalization.

6. To improve immunity to noise, a low-pass filter (LPF) is used at various processing nodes of the system. The preferred filter is a single-pole LPF of the form $A(z) = (1 - a) / (1 - az^{-1})$, $0 < a < 1$.
7. Frequency estimation is performed by computing $\rho_\Omega(n)$ and finding the closest frequency according to predefined reference values $\rho^{(l)}$, $l = 1, \ldots, L$. When the closest reference is found, the measurements are validated by imposing a frequency tolerance constraint for the tone event being processed. After a tone is validated, its identification (tone ID) is set for further processing.
8. Depending on the tone event, optional prefiltering may be needed to remove interference by noise or guard tones.
9. To improve dynamic range of the detector, the input signal level can be adjusted via an automatic level control device (ALC).
10. Depending on the frequency range of interest, sub-rate processing can be employed to reduce computational complexity, for example, the DTMF detector presented in [1].

Independent of the modulation scheme in use, the actual detection relies on common level-based decision logic, and controlling parameters depend on the tone ID. The pseudo-code in **Example 1** illustrates this process. It also shows how to detect a known tone event segment.

Example 1. Integer Measurements, μ(n) (AM/FSK/DPSK) on a Per-Sample Basis and Detect a Known Tone Event Segment

```
IF (μ(n)≠ μest)
        nn++;
        IF (nn > Nn)
                np = nn;
                nn = 0;
                μest = μ(n);
                Fd = 0;
        END
ELSE
        np++;
        IF (np > Np)
                IF (Fd = 0)
                        Fd = 1;
                        Δμ = (μest - μold) mod μmax;
                        μold = μest;
                END
                IF (np > Nd)
                        np = 0;
                        nn = 0;
                        Report μest and Δμ;
                        Fd = 0;
                END
        END
END
```

The calculation of Δμ is critical for DPSK demodulation but optional for AM/FSK demodulation.

2 Tone Event Detector on StarCore DSPs

The tone event detector is designed to identify a set of tones, including TTY, ANS, V.21, CNG, USB1, and so on. Its expected functionality verification is to detect the correct tone event and not to signal any tone in its absence. Since verification of functionality is not based on bit-exactness, we can more aggressively optimize code by rearranging computation order and even changing computation precision. This section illustrates key optimization strategies to boost code performance significantly. All techniques are tested with a highly efficient Metrowerks® C compiler that takes maximum advantage of StarCore architectural features. The Metrowerks C compiler provides intrinsic function access through `prototype.h`, which supports the basic ITU fixed-point operation, `basop.h`, and to the fixed-point operation extensions.

2.1 Automatic Level Control (ALC)

An ALC can boost or decrease the signal level so that the processing signal is well contained within a predetermined range. A simple ALC is a normalization process followed by a scaling process. It first searches for the maximum absolute magnitude of the signal for the processing frame. The scaling factor is based on the maximum absolute magnitude. The MAXM special instruction is used to compute the maximum absolute value. In C programming, this instruction can be accessed by the `L_maxm` intrinsic function. In combination with loop unrolling of four, a factor of eight speed-up can be achieved.

2.2 Tone Indication

The modified TK energy operator with low-pass (LP) filtering is as follows:

$$P(n) = aP(n-1) + (1-a)\,|\,x^2(n-k) - x(n)x(n-2k)|$$

All variables ($P(n),a,x(n)$) have 16 bits. Using rounding along with the MAC instruction, the TK energy operator is computed in three steps and the LP filter in two steps. The 32-bit resolution of intermediate values of $P(n)$ is rounded into 16-bit data in final computation. Rounding removes the bias (flooring by truncating) in the original program, thus changing the internal state/memory of the program. Such optimization yields slightly different results than alternative implementations, but all alternatives should pass the verification at the system level. **Table 1** shows details of the underlying DSP operations.

Table 1. Computation Steps of R(n)

Computation	Output Resolution	Instruction
$A = x^2(n-k)$	32 bits	MPY
B = A $- x(n)x(n-2k)$	16 bits	MACR
C = |B|	16 bits	ABS
D = $(1-a)$ C	32 bits	MPY
E = $aP(n-1)$ + D	16 bits	MACR

2.3 Tone Indicator Counter

If a valid tone is present, a constant level of TK energy value is obtained. To filter out the false tone indications, a minimum duration of a constant level must be met. The duration is measured by a tone indicator counter. When the tone indicator counter is larger than a minimum threshold, different tone event detectors should be invoked. The following implementation invokes many checking steps, which usually involve branching.

```
if (TK energy is a constant level)
      if (tone_ind_count < MIN_TONE_COUNT_THRESHOLD)
          tone_ind_count++;
else
      tone_ind_count = 0;
if (tone_ind_count == MIN_TONE_COUNT_THRESHOLD)
       Invoke tone event detector algorithm
```

Changing the logic and using fixed-point instructions with saturation on the SC140 core, one checking step can be eliminated, as follows.

```
if (TK energy is a constant level)
      tone_ind_count = add(tone_ind_count,1);
else
      tone_ind_count = 0;
if (tone_ind_count >= MIN_TONE_COUNT_THRESHOLD)
       Invoke tone event detector algorithm
```

The INC.F instruction is used to increase the counter with saturation, thus avoiding a test operation.

2.4 Finding the Closest Reference Frequency Tone

The ratio of energy operators, ρ_Ω, contains the frequency information of a signal. The minimum absolute distance between ρ_Ω and a set of reference values yields the best match. The index of the best match reference values corresponds to the frequency information. The conventional compare and select approach does not make efficient use of the SC140 ALU units, since there is only one compare bit in the SC140 architecture. The optimized approach uses the MIN and ADD instructions to find the minimum absolute value and its corresponding index simultaneously. The absolute value of the distance is placed into the upper 16 bits of the register, and the index is stored in the lower 16 bits. The conventional approach uses the CMPGT and TFRF instructions. **Table 2** shows the details of both approaches.

Table 2. Details of the Frequency Search Implementation

Conventional Approach	Optimized Approach	Comments
SUB	SUB	$\rho_\Omega - \rho_k$
ABS, INC	ABS, INC	$\lvert \rho_\Omega - \rho_k \rvert$, k++
CMPGT, MIN	ADD	$\arg\min_k \lvert \rho_\Omega - \rho_k \rvert$
TFRF	MIN	

Both approaches have the same number of critical paths. The optimized search uses one less instruction and further decouples the data dependency, allowing the compiler to perform more aggressive software pipelining.

2.5 FIR Filtering Implementation

During the detection of some tone events, such as the USB1 signal, an FIR filter may be used to eliminate unwanted signals. The tone event detector processes the signals on a sample-by-sample basis, so multi-sampling optimization is not viable. Moreover, filtering is not performed on a frame-by-frame basis, so it is difficult for the compiler to generate circular buffering automatically. However, a pseudo circular buffer can be achieved by storing the fixed filtering coefficient twice, similar to a periodic extension. The pseudo circular buffer increases code efficiency with only a modest increase in memory use. In **Figure 6**, there are four fixed filtering coefficients. The multiply accumulation starts from the first coefficient, indicated by the left arrow, to the last coefficient, indicated by the right arrow. Repeating the FIR filter coefficients twice achieves a wrap-around mechanism without additional checking.

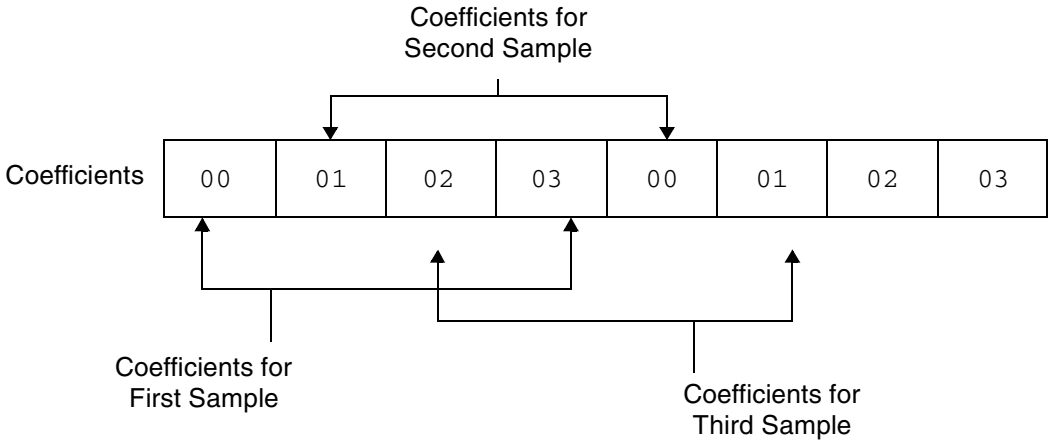


Figure 6. Pseudo Circular FIR Processing.

3 Tone Events in Packet Telephony

Table 3 summarizes typical tone events to be detected in packet telephony systems, along with the number of the reference document listed in **Section 6**, *References,* on page 21.

Table 3. Tone Events to Detect in Packet Telephony Systems

Event	Reference
T.30 Calling Tone (CNG)	[11]
V.25 Calling Tone (CT)	[12]
V.25 Answering Tone (ANS)	[12]
T.30 CED	[11]
V.25 Answering Tone with Phase Reversal (ANS_PR)	[12]
V.8 Answering Tone with Amplitude Modulation (ANSam)	[13]
V.8 Answering Tone with Amplitude Modulation and Phase Reversal (ANSam_PR)	[13]
V.22 Unscrambled Binary Ones (USB1)	[14]
V.21 Channel 2 HDLC flag	[15]
TIA/EIA-825 TTY tones	[16–17]
V.8bis Initiate/Respond	[18]
Q.24 Dual Tone Multiple Frequency (DTMF)	[19]

Transporting these tone events over IP networks is the subject of ongoing development effort [2–4]. A well-known example of a transport protocol for tone events is the IETF RFC2833 standard [20–21]. Because it is a relatively simple specification, it is described in detail in the next section.

4 RFC2833 Overview

Procedures for using RFC2833-compliant packets are described in this section. The RFC2833 transport protocol is a good example of a way to transmit tone events over IP networks. It defines two payload formats:

- One for carrying dual-tone multi-frequency (DTMF) digits and other line and trunk signals.
- One for general multi-frequency tones in real-time protocol (RTP) packets.

A gateway is a platform for handling communications between IP packet-based networks and circuit-switched networks. It offers two options for handling DTMF digits and events. First, it can simply measure the frequency components of the voice band signals and transmit this information to the RTP receiver [22]. In this mode, RTP packets with tone representations are sent from an RTP sender to an RTP receiver. Second, a gateway can recognize the tones and translate them into a name, such as a DTMF tone. The receiver then produces a tone signal. In this mode, RTP packets with named tone events are sent from an RTP sender to an RTP receiver.

Gateways that send signaling events via RTP can transmit both named telephone events and the tone representation as a single RTP session. Before the session starts, the RTP sender and the corresponding RTP receivers must agree on the payload type to assign to each kind of payload so that an RTP receiver can interpret the payload data and decode it to its corresponding audio sampled signal.

If gateways send only one kind of RTP payload in a single RTP packet (either a named telephone event or tone representation), the packet type (PT) field of the RTP header itself contains the payload type. If the gateway sends an RTP packet containing both payload types, the PT field of the RTP header contains a dynamic payload type that is assigned according to the RFC2198-style redundant payload type [23], which is also a point of agreement between the RTP sender and RTP receiver(s) before the session starts. This assignment uses out-of-band means not specified in the RFC2833. For each payload block within an RTP payload, the payload type is specified in the *block PT* field of that payload block.

If a gateway cannot represent a tone, it should send the audio tones as regular RTP audio packets (for example, as payload format PCMU) in addition to the named telephone event. If this feature is turned on, the *block PT* field in the payload must be assigned the payload type assigned to the vocoder that encodes and decodes the audio signals. The RTP receiver must be able to determine the kind of payload data it receives so that it can render (decode) the data correctly.

The RTP transport is designed as a non-guaranteed transport system. The consequence of packet loss is greater for the named telephone event packet than for the regular audio signal packet. Therefore, mechanisms on top of the RTP transport mechanism help to ensure that important tone events such as DTMF signaling events are transferred reliably from the RTP sender to the RTP receiver when a small fraction of packets are lost, duplicated, or delivered too late. The error resilience issue is one of the most critical for RFC2833.

Events lasting longer than one event period are updated periodically so that the receiver can reconstruct the event and duration if it receives the update packets with minimal delay, which is most helpful for long events. The last event packet is transmitted a number of times (three times by default) if there is no subsequent event, which improves reliability of delivery of the last event packet.

4.1 Payload Format for Named Telephone Events

DTMF digits and named telephone events are part of the audio stream. Therefore, to simplify the generation of audio waveforms at a gateway, they must use the same sequence number and time-stamp base as the regular audio channel. The default clock frequency is 8000 Hz, but the clock frequency can be redefined when the dynamic payload type is assigned. Clock frequency is typically changed before the RTP session starts and how to change it is outside the scope of RFC2833. The RTP timestamp reflects the measurement point for the current event duration and extends forwards from that time. The marker bit indicates the beginning of a new event, such as the start of a DTMF tone. The RTP receiver must be designed so that it remains functional when the packet with its mark bit set is lost. The RFC2833 payload format is shown in **Figure 7** and described in **Table 4**.

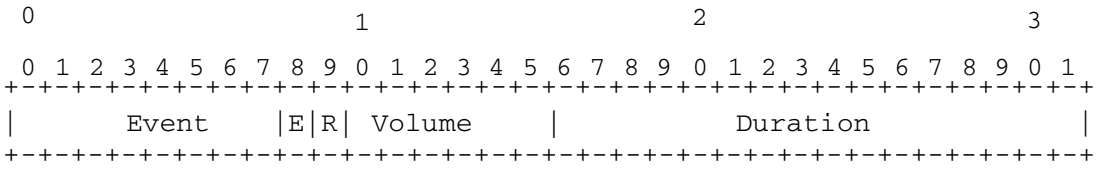


Figure 7. Format of the RFC2833 Payload for Named Telephone Events

Table 4. Components of the RFC2833 Payload Format

RFC2833 Payload Component	Description
Event	Events are decoded as shown in **Table 5** through **Table 8**.

Table 4. Components of the RFC2833 Payload Format (Continued)

RFC2833 Payload Component	Description
E	If set to a value of one, the *end* bit indicates that this packet contains the end of the event. Thus, the duration parameter measures the complete duration of the event.
R	This field is reserved for future use. The sender must set it to zero, and the receiver must ignore it.
Volume	For DTMF digits and other events representable as tones, this field describes the power level of the tone, expressed in dBm0 after the sign is dropped. Power levels range from 0 to –63 dBm0. The range of valid DTMF is from 0 to –36 dBm0 (must accept); lower than –55 dBm0 must be rejected [19]. Thus, larger values denote lower volume.
Duration	Duration of this digit, in timestamp units. Thus, the event begins at the instant identified by the RTP timestamp and lasts as long as indicated by this parameter. The event may or may not have ended. For a sampling rate of 8000 Hz, this field is sufficient to express event durations of up to approximately 8 seconds.

Table 5. Data and Fax Named Events

Event	Encoding
DTMF 0–9	0–9
DTMF *	10
DTMF #	11
DTMF A–D	12–15
Flash	16
Answer tone (ANS)	32
Answer tone with phase reversal (/ANS)	33
Answer tone with AM modulation (ANSam)	34
Answer tone with AM modulation and phase reversal (/ANSam)	35
Calling tone (CNG)	36
V.21 channel 1,”0” bit	37
V.21 channel 1, “1” bit	38
V.21 channel 2, “0” bit	39
V.21 channel 2, “1” bit	40
CRdi	41
CRdr	42
CRe	43
ESi	44
ESr	45
MRdi	46
MRdr	47
MRe	48
CT	49

Table 6. E.182 [24] Line Events

Event	Encoding
Off Hook	64
On Hook	65
Dial tone	66
PABX internal dial tone	67
Special dial tone	68
Second dial tone	69
Ringing tone	70
Special ringing tone	71
Busy tone	72
Congestion tone	73
Special information tone	74
Comfort tone	75
Hold tone	76
Record tone	77
Caller waiting tone	78
Call waiting tone	79
Pay tone	80
Positive indication tone	81
Negative indication tone	82
Warning tone	83
Intrusion tone	84
Calling card service tone	85
Payphone recognition tone	86
CPE alerting signal (CAS)	87
Off-hook warning tone	88
Ring	89

Table 7. Country-Specific Line Events

Event	Encoding
Acceptance tone	96
Confirmation tone	97
Dial tone recall	98
End of three party service tone	99
Facilities tone	100
Line lockout tone	101
Number unobtainable tone	102
Offering tone	103

Table 7. Country-Specific Line Events (Continued)

Event	Encoding
Permanent signal tone	104
Preemption tone	105
Queue tone	106
Refusal tone	107
Route tone	108
Valid tone	109
Waiting tone	110
Warning tone (end of period)	111
Warning Tone (PIP tone)	112

Table 8. Trunk Events

Event	Encoding
MF 0–9	128–137
MF K0 or KP (start-of-pulsing)	138
MF K1	139
MF K2	140
MF S0 to ST (end-of-pulsing)	141
MF S1–S3	142–143
ABCD signaling (see below)	144–159
Wink	160
Wink off	161
Incoming seizure	162
Seizure	163
Unseize circuit	164
Continuity test	165
Default continuity tone	166
Continuity tone (single tone)	167
Continuity test send	168
Continuity verified	170
Loopback	171
Old milliwatt tone (1000 Hz)	172
New milliwatt tone (1004 Hz)	173

4.2 Sending and Receiving Named Telephone Event Packets

An audio source starts transmitting event packets as soon as it recognizes an event. It sends a packet at each interval of 50 ms (or any other user-specified interval) or the packet interval for the audio codec used in this session, if known. If an event continues for more than one period, the source generating the events sends a new event packet with the RTP timestamp value corresponding to the beginning of the event and the duration of the

event increased correspondingly. The RTP sequence number is incremented by one for each packet. If there is no new event in the last interval, the event is transmitted three times or until the next event is recognized to ensure that the duration of the event is recognized correctly even if fewer than three of the last packets for an event are lost.

DTMF digits and events are sent incrementally so that the receiver does not have to wait for the completion of the event. For example, tones that are two seconds long would otherwise incur a substantial delay. The transmitter cannot determine whether event length is important and thus needs to transmit immediately and incrementally. The incremental transmission mechanism avoids delay. This is important for some applications, such as gateways into the PSTN, which are affected by both delays and event duration.

4.3 Payload Format for Telephony Tones

Instead of describing tones and events by name, it is sometimes preferable to describe them by their waveform properties, speeding up recognition since waveform properties do not depend on recognizing durations or pauses. In this method, the RTP timestamp reflects the measurement point for the current packet. The event duration extends forwards from that time. The default timestamp rate is 8000 Hz, but other rates can be defined. The timestamp rate does not affect the interpretation of the frequency. This format does not have a static payload type number. It uses a RTP payload type number established dynamically and out-of-band. This payload format is shown in **Figure 8** and described in **Table 9**.

The RTP timestamp is updated for each packet generated, in contrast to the timestamp for packets carrying named telephone events. The marker bit of the first RTP packet for a tone is set to 1. In subsequent packets for the same tone, the marker bit is set to 0, and the RTP timestamp in each subsequent packet equals the sum of the timestamp and the duration in the preceding packet.

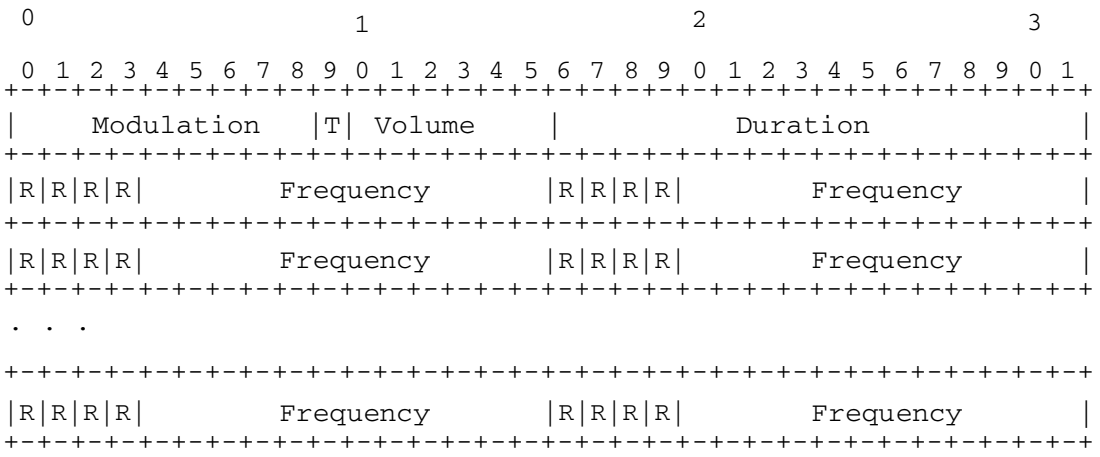


Figure 8. Format of the RFC2833 Payload for Tones

Table 9. Components of the RFC2833 Payload for Tones Format

RFC2833 Payload Component	Description
Modulation	The modulation frequency, in Hz. The field is a 9-bit unsigned integer, allowing modulation frequencies up to 511 Hz. If there is no modulation, this field has a value of zero.
T	If the T bit is set (one), the modulation frequency is divided by three. Otherwise, the modulation frequency is taken as is. This bit allows frequencies accurate to 1/3 Hz, since modulation frequencies such as 16 2/3 Hz are in practical use.

Table 9. Components of the RFC2833 Payload for Tones Format (Continued)

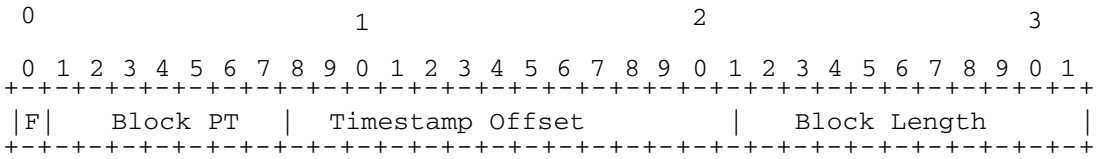
RFC2833 Payload Component	Description
Volume	The power level of the tone, expressed in dBm0 after the sign is dropped, with a range from 0 to –63 dBm0. A preferred level range for digital tone generators is –8 dBm0 to –3 dBm0.
Duration	The duration of the tone, measured in timestamp units. The tone begins at the instant identified by the RTP timestamp and lasts for the duration value. The definition of duration corresponds to that for sample-based codecs, where the timestamp represents the sampling point for the first sample.
Frequency	The frequencies of the tones to be added measured in Hz and represented as a 12-bit unsigned integer. The field size is sufficient to represent frequencies up to 4095 Hz, which exceeds the range of telephone systems. A value of zero indicates silence. A single tone can contain any number of frequencies. Silence is represented by a zero frequency.
R	This field is reserved for future use. The sender must set it to zero, the receiver must ignore it.

4.4 Multiple Tones and Named Events in One Packet

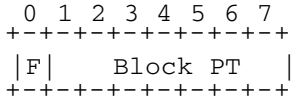
The two RFC2833 payload formats can be combined into a single payload using the method specified in RFC2198 [23]. That is, the payload portion of an RTP packet can contain multiple named telephone events or tone representations or multiples of both. The RFC2198 method decomposes the payload portion of the RTP packet into more than one (*N*) sub-level blocks of data. Each block specifies one named telephone event or one entity of telephone tone representation. More specifically, each block consists of a block header and a block payload. For *N* blocks, there is the block_header[i], block_payload[i] where i = 1–*N*. The order of these block headers and block payloads is as follows:

rtp_header
block_header[1]
....
block_header[N]
block_payload[1]
....
block_payload[N]

With the exception the last block_header, all other block headers are in the following 4-byte format:



The last block header, block_header[N], is just one byte long.



The F field of all block headers except the last must be set to 1; the F field of the last block header must be set to 0. The block PT (payload type) indicates the payload type of the data stored in the corresponding block payload. There are three choices for the block PT:

- Named telephone event payload type.
- Telephone tone representation payload type.
- Vocoder payload type (for example, PCMU).

The first two payload types are dynamic types determined by an out-of-band mechanism before the RTP session starts. The vocoder payload is typically a static payload type defined by RFC3551 [25].

The value in the PT field of the RTP header for the RFC2198-style blocking method can be considered as a redundant payload type. It indicates a set of block PTs for use by any blocks in an RTP packet. This redundant payload type is established by out-of-band means before the RTP session starts. The block length is the size of the corresponding block payload, in bytes. When the timestamp for the starting tone or audio data in block_payload[i], i=1..N is computed, there is no alignment requirement to specify where each block payload starts.

The last block header, block_header[*N*], has neither the timestamp offset field nor the block length field because the 32-bit RTP timestamp field specifies when the tone or audio starts, that is, when the receiver starts rendering the tone or audio signal (the latter is the vocoder case). The start time for rendering the data in block_payload[i], i=1–*N*–1, is the timestamp for the very first audio sample. This timestamp appears in the RTP header, minus the always positive value in the timestamp offset field of block_header[i], where i=1–*N*–1.

The last block header does not contain the block length field because the length field in the header of the lower transport layer—for example, the length field of the UDP header—implies where the payload ends. If the P (padding) bit in the RTP header is set to 1, up to three bytes can be padded to the RTP payload at the end. The number of padded bytes is indicated by the value of the last byte of the payload. If the P bit in the RTP header is cleared to 0, there are no padding bytes. No more than one RTP packet can be included in the payload field of the RFC2198 packet in the lower transport layer. Instead, multiple blocks should be used in a single packet.

Two examples from [21] illustrate how multiple blocks of named tone event blocks or telephone tone representation blocks are packed into one RTP packet. The first example is a typical RTP packet, where the user is just dialing the last digit of the DTMF sequence “911.” The first digit is 200 ms long (1600 timestamp units) and starts at time 0, the second digit is 250 ms long (2000 timestamp units) and starts at time 800 ms (6400 timestamp units), the third digit is pressed at time 1.4 s (11,200 timestamp units) and the packet is sent at 1.45 s (11,600 timestamp units). The frame duration is 50 ms. To make the parts recognizable, **Figure 9** ignores byte alignment. Timestamp and sequence number are assumed to be zero at the beginning of the first digit. The dynamic payload types 96 and 97 are assigned for the redundancy mechanism and the telephone event payload, respectively.

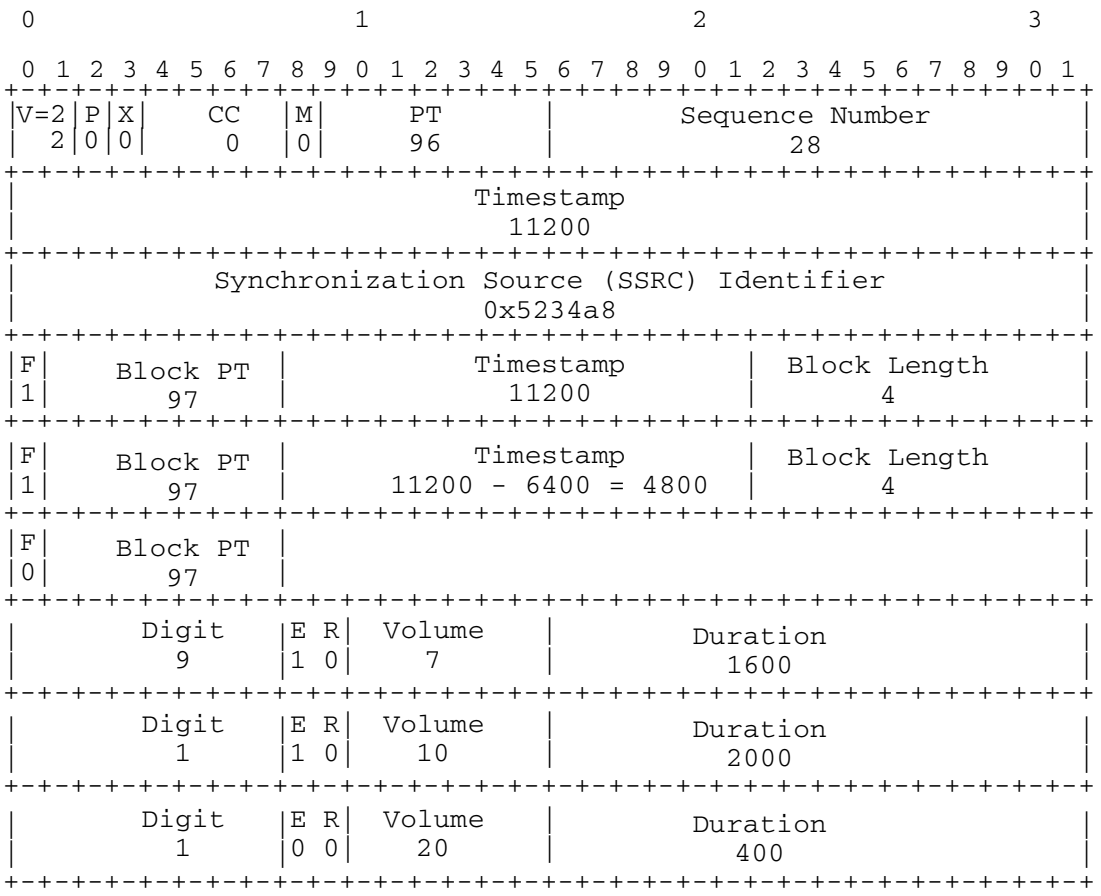


Figure 9. Example RTP Packet After Dialing 911

If the gateway sends an RTP packet containing both payload types in the same packet, the PT field contains a dynamic payload type that is assigned to the RFC2198-style redundant payload type. This is a point of agreement between the RTP sender and the RTP receiver(s) before the session starts. This agreement is established via out-of-band means outside the scope of the RFC2833. The dynamic payload types 96 and 97 are established by this means.

In the second example, the RTP packet combines two tone payloads and one telephone event payload (see **Figure 10**). The payload types are chosen arbitrarily as 97 and 98, respectively, with a sample rate of 8 KHz. The redundancy format has the dynamic payload type 96. The packet represents a snapshot of a U.S. ringing tone, 1.5 seconds (12,000 timestamp units) into the second on part of the 2.0/4.0 second cadence—that is, a total of 7.5 seconds (60,000 timestamp units) into the ring cycle. The 440 + 480 Hz tone of this second cadence starts at RTP timestamp 48,000. Four seconds of silence precede it, but since RFC2198 has only a fourteen-bit offset, only 2.05 seconds (16383 timestamp units) can be represented. Although the tone sequence is not complete, the sender can determine that this is indeed ringback and thus include the corresponding named event.

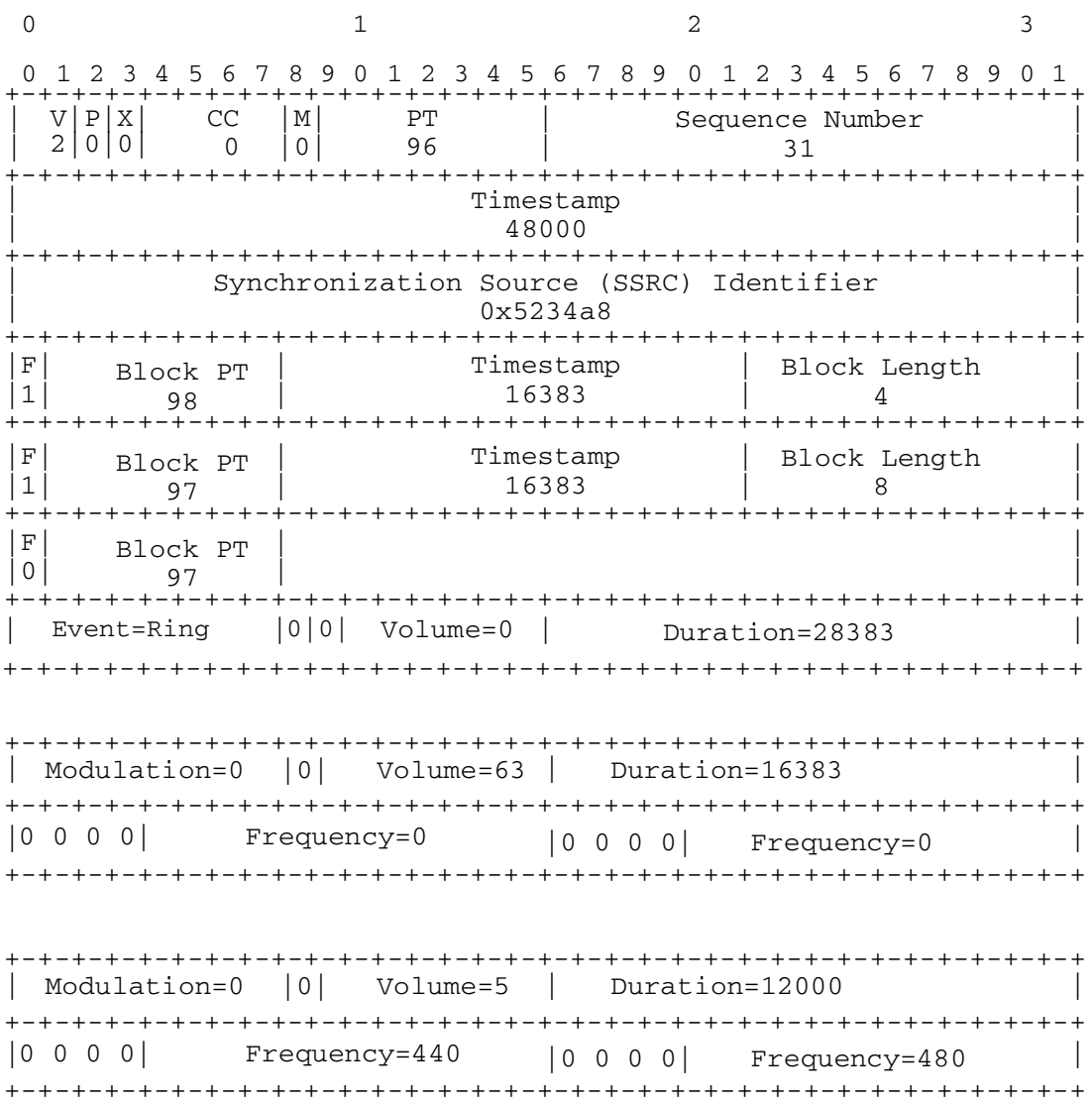


Figure 10. Combining Tones and Events in a Single RTP Packet

5 Conclusion

Tone event detection is an effective way to enable successful communication of signaling tones over IP networks. This application note considers tone event detection in packet telephony systems, in terms of both theory and implementation on efficient tone event detection architecture. The RFC2833 transport protocol of tone events over IP networks illustrates an important practical use of the proposed tone event detection method. Finally, selected code optimization strategies are explored along with key parallel architectural features of the StarCore DSP core. The robust tone event detector developed on StarCore can handle all the tone events described in **Section 3**, *Tone Events in Packet Telephony,* on page 11. During testing with a large representative set of test vectors, this tone event detector demonstrated the expected functionality in both real-time and file I/O tests, requiring less than 2 MCPS on average.

6 References

[1] V. Emiya, L. F. C. Pessoa, D. Vallot and D. Melles, *Generic Tone Detection using Teager-Kaiser Energy Operators on the StarCore SC140 core*, Freescale application note AN2384/D, August 2003.

[2] ITU-T V.150.1, *Procedures for the End-to-End Connection of V-series DCEs Over an IP Network*, 2002.

[3] ITU-T V.151, *Procedures for the End-to-End Connection of Analogue Text Telephones Over an IP Network*, initial draft, 2004.

[4] ITU-T V.152, *Procedures for Supporting Voice-Band Data over IP Networks*, initial draft, 2004.

[5] R. Beck, A. G. Dempster, and I. Kale, "Finite-precision Goertzel filters used for signal tone detection," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 48, no. 6, pp.691-700, 2001.

[6] F. M. Gardner, "Properties of Frequency Difference Detectors," *IEEE Transactions on Communications*, vol. COM-33, pp. 131-138, February 1985.

[7] A. D'Andrea, A. Ginesi, and U. Mengali, "Digital Carrier Frequency Estimation for Multilevel CPM Signals," *Proceedings ICC'95*, Seattle, WA, pp. 1041-1045, June 1995.

[8] L. F. C. Pessoa, R. A. Dyba and P. He, "Joint Audio Power Estimation and Tone Indication in Network Echo Cancellers," in *Global Signal Processing Expo and Conference (GSPx)*, 2003.

[9] M. J. Werter, "A Digital Phase-Locked Loop for Frequency Detection," in *Proc. of 38th Midwest Symposium on Circuits and Systems*, pp. 13-16, 1995.

[10] C. Dick, F. Harris, and M. Rice, "Synchronization in Software Radios - Carrier and Timing Recovery Using FPGAs," In *IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 195-204, 2000.

[11] ITU-T T.30, *Procedures for Document Facsimile Transmission in the General Switched Telephone Network*, 1996.

[12] ITU-T V.25, *Automatic Answering Equipment and General Procedures for Automatic Calling Equipment on the General Switched Telephone Network Including Procedures for Disabling of Echo Control Devices for both Manually and Automatically Established Calls*, 1996.

[13] ITU-T V.8, *Procedures for Starting Sessions of Data Transmission Over the Public Switched Telephone Network*, 2000.

[14] ITU-T V.22, *1200 Bits per Second Duplex Modem Standardized for Use in the General Switched Telephone Network and on Point-to-Point 2-Wire Leased Telephone Type Circuits*, 1988.

[15] ITU V.21, *300 Bits per Second Duplex Modem Standardized for Use in the General Switched Telephone Network and on Point-to-Point 2-Wire Leased Telephone Type Circuits*, 1988.

[16] TIA/EIA/IS-840-A, *Minimum Performance Standards for Text Telephone Signal Detector and Text Telephone Signal Generator*, 2001.

[17] TIA-825-A, *A Frequency Shift Keyed Modem for Use on the Public Switched Telephone Network*, 2003.

[18] ITU-T V.8bis, *Procedures for the Identification and Selection of Common Modes of Operation between DCEs and DTEs Over the Public Switched Telephone Network and Leased Point-to-Point Telephone Type Circuits*, 2000.

[19] ITU-T Q.24, *Multifrequency Push-Button Signal Reception*, 1988.

[20] IETF RFC2833, *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*, 2000.

[21] IETF RFC2833bis, *RTP Payload for DTMF Digits, Telephony Tones and Telephony SIgnals*, 2004.

[22] IETF RFC3550, RTP: *A Transport Protocol for Real-Time Applications*, 2003.

[23] IETF RFC2198, *RTP Payload for Redundant Audio Data*, 1997.

[24] ITU-T E.182, *Application of Tones and Recorded Announcements in Telephone Services*, 1998.

[25] IETF RFC3551, *RTP Profile for Audio and Video Conferences with Minimal Control*, 2003.

NOTES:

How to Reach Us:

USA/Europe/Locations not listed:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu. Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
852-26668334

Learn More:
For more information about Freescale Semiconductor products, please visit
http://www.freescale.com

AN2775
Rev. 2
6/2004

freescale™ semiconductor