(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 December 2010 (02.12.2010)

PCT

(10) International Publication Number
WO 2010/138014 A1

(51) International Patent Classification:
H04Q 1/453 (2006.01)

(21) International Application Number:
PCT/RU2009/000268

(22) International Filing Date:
28 May 2009 (28.05.2009)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **LSI CORPORATION** [US/US]; 1621 Barber Lane, Milpitas, California 95035 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LETUNOVSKIY, Aleksey Alexandrovich** [RU/RU]; Svobody St., 199-1, Tokarevka, Tambov region, 393550 (RU). **LYALIN, Ilya Viktorovich** [RU/RU]; Novoyasenevskii pr., 21-3 -120, Moscow, 117593 (RU). **MARKOVIC, Alexander** [RS/US]; 31 Soldier Song Ln, Media, Pennsylvania 19063 (US). **MAZURENKO, Ivan Leonidovich** [RU/RU]; Molodezhnaya, 34-52, Khimki, 141407 (RU). **NIKITIN, Andrey Anatolevich** [RU/RU]; Leninsky pr., 82-193, Moscow, 119261 (RU).

(74) Agents: **LAW FIRM "GORODISSKY & PARTNERS" LTD.** et al.; MITS Alexander Vladimirovich, B.Spasskaya str., 25, bldg. 3, Moscow, 129090 (RU).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

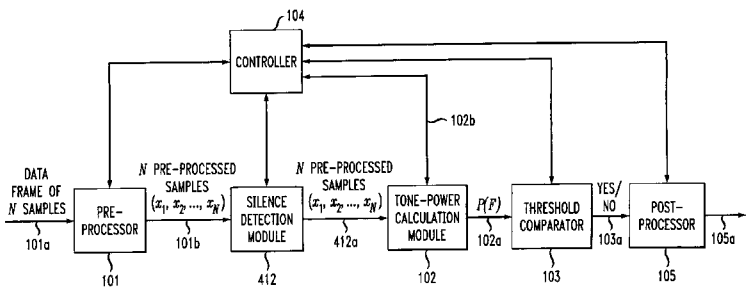
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: HIGH-PERFORMANCE TONE DETECTION USING A DIGITAL SIGNAL PROCESSOR (DSP) HAVING MULTIPLE ARITHMETIC LOGIC UNITS (ALUs)

FIG. 4



(57) **Abstract**: In one embodiment, a DSP having four arithmetic logic units (ALUs) and able to have two read/write operations per clock cycle performs silence detection and tone detection for data frames containing samples of an audio signal. The ALUs are used together in parallel to process the samples in the data frames received by the DSP. A received data frame is filtered by the silence detection so that substantially silent frames are dropped and non-silent frames are further processed. In the tone detection, a filtered data frame is processed, four samples at a time, to determine the power of the signal at a given frequency, where the power determination is used to determine whether a given tone (i.e., a signal at a given frequency) is present in the data frame.

WO 2010/138014 A1

HIGH-PERFORMANCE TONE DETECTION USING A DIGITAL SIGNAL PROCESSOR (DSP) HAVING MULTIPLE ARITHMETIC LOGIC UNITS (ALUs)

BACKGROUND OF THE INVENTION

Field of the Invention

The current invention relates to digital signal processors (DSPs), and in particular, DSPs having multiple arithmetic logic units (ALUs).

Description of the Related Art

Conventional telephone keypads generate Dual Tone Multi Frequency (DTMF) signals when pressed. Pressing any particular key on a telephone keypad produces a unique combination of two tones, where a low tone represents the row of the key on the keypad and a high tone represents the column of the key on the keypad. The frequencies of the tones range from about 697 Hz to about 1633 Hz. Note that column frequency 1633 represents keys A, B, C, and D, where these keys, although part of DTMF signaling, are absent from many conventional telephone keypads. It should be noted that communication systems may produce other tone combinations comprising the same or a different number of tones at different frequencies. Corresponding communication equipment, such as equipment at a telephone company's central office that connects with the telephone, may need to be able to detect the presence of particular tones in a sample of audio content from the telephone.

Audio content in conventional modern telephone systems is usually digitized at a sampling rate of 8 kHz for processing and transmission by the telephone service provider(s). It should be noted that other sampling rates are possible. The digitized audio content is typically processed as data frames where each frame represents a window of time. Typical data frame lengths are 5ms and 10ms, which, at an 8kHz sampling rate, are equivalent to 40 and 80 samples, respectively. A tone-detection decision is typically made once per frame. Several methods are known in the prior art for determining whether a frame contains audio content at a particular frequency.

The power $P(F)$ of the input signal at a given frequency F in an N-sample data frame can be determined using the formula of Equation (1) below:

$$P(F) = \left| \sum_{k=1}^{N} x_k e^{-j\frac{2\pi F}{F_s}(N-k)} \right|^2, \quad (1)$$

where $x_1, x_2, ..., x_N$ are the samples of the frame, j is the square root of -1, and F_s is the sampling frequency for the frame. The samples $x_1, x_2, ..., x_N$ represent voltage values of an electrical signal that represents corresponding sound pressure levels of an audio signal. Once the value of power $P(F)$ is determined for the particular frame, that value is compared to a

threshold and the result of the comparison is used in determining whether a tone at the given frequency has been detected for that frame.

Power $P(F)$ for an N-sample frame can be calculated iteratively or recursively using the algorithm of Equation (2) below:

$$\mathrm{Re}(0) = \mathrm{Im}(0) = 0 \tag{2.1}$$

$$\mathrm{Re}(n) = x_n + \cos\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Re}(n-1) - \sin\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Im}(n-1), \text{ and}$$

$$\mathrm{Im}(n) = \sin\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Re}(n-1) + \cos\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Im}(n-1) \tag{2.2}$$

for $n = 1, 2, ..., N$

$$P(F) = \mathrm{Re}\,(N)^2 + \mathrm{Im}\,(N)^2 \tag{2.3}$$

Equation (2) calls for calculating values for Re(n) and Im(n) for each sample of the frame. The calculations for each sample are based on the values of x_n, Re(n-1), and Im(n-1) (where Re(0) and Im(0) are 0). Power $P(F)$ is then calculated for the N-sample frame based on Re(N) and Im(N). It should be noted that, generally, a recursive calculation would involve implementing the calculation using a procedure that calls itself repeatedly (*e.g.*, N times) until some condition is met, while an iterative calculation would involve implementing the calculation using a procedure that includes an explicit instruction loop that is executed a certain number (*e.g.*, N) of times. Thus, to illustrate a recursive function, a recursive pseudo-code implementation for a factorial function could be factorial(n) where if $n = 0$ then return 1 else return n·factorial(n-1). Similarly, to illustrate an iterative function, an iterative pseudo-code implementation for a factorial function could be factorial(n) where temp = 1; for $i = 2$ to n, temp = temp·i; return temp. Since, generally, recursive algorithms can be transformed into corresponding iterative algorithms and vice-versa, the terms, as used herein, unless otherwise indicated, are interchangeable.

Another iterative or recursive way to calculate power $P(F)$ for an N-sample frame involves using the Goertzel algorithm, as shown in Equation (3) below:

$$\mathrm{S}(0) = \mathrm{S}(-1) = 0 \tag{3.1}$$

$$S(n) = x_n + 2\cos\left(\frac{2\pi F}{F_s}\right) \cdot S(n-1) - S(n-2) \tag{3.2}$$

for $n = 1, 2, ..., N$

$$P(F) = S(N)^2 - 2\cos\left(\frac{2\pi F}{F_s}\right) \cdot S(N) \cdot S(N-1) + S(N-1)^2 \quad (3.3)$$

The Goertzel algorithm involves calculating an N-item series of values from $S(1)$ to $S(N)$ for the samples $x_1, x_2, \ldots, x_N$ of the frame, where each $S(n)$ value is based on x_n, $S(n\text{-}1)$, and $S(n\text{-}2)$, and where $S(0)$ and $S(-1)$ are 0. Power $P(F)$ for the N-sample frame is then calculated based on the last two values of the series, *i.e.*, $S(N)$ and $S(N\text{-}1)$.

When any of the above calculations are performed by a processor, such as an Application-Specific Integrated Circuit (ASIC) or a Digital Signal Processor (DSP), slight modifications may be made to the formulas to account for the limitations of the fixed-point arithmetic that may be used by those processors. For example, a saturation function may be used to implement saturation arithmetic where results of arithmetic operations, which may otherwise overflow, are clamped between a maximum value and a minimum value. Saturation may also be used in rounding off numbers, such as, for example, in converting a 32-bit fixed-point number into a 16-bit fixed-point number. 32-bit fixed-point numbers are also known as Q31- or Q1.31-format numbers, where the 31 represents the number of bits after the binary point (*i.e.*, the binary equivalent of a decimal point) and the 1, when present, represents the number of bits before the binary point. It should be noted that, in general, if no number is present before the binary point (*e.g.*, Q31), it is assumed that "1" is intended there (*i.e.*, Q1.31). Similarly, 16-bit fixed-point numbers are known as Q15-format or Q1.15-format numbers. As used herein, unless otherwise noted, references to Q*c*.15 and Q*c*.31 formats indicate generic format references that include formats with zero or more bits before the binary point. Thus, for example, the term Q*c*.31 format includes Q0.31, Q1.31, Q2.31, etc. formats.

In some implementations of a saturation function, the saturation function merely discards the least significant bits of the saturated number. This can cause round-off errors which may be corrected using an additive correction. Thus, if, for example, a saturation function SAT[a] discards the 16 least significant bits when saturating 32-bit number a to 16-bit number a', then an additive correction of 2^{-16} may be used so that SAT[$a + 2^{-16}$] functions like a rounding-off function *round*[a] for rounding off a to a 16-bit number. An illustrative decimal example may be helpful to understand how this works. Suppose that the decimal-number function SAT[a] discards the digits after the decimal point of a. Thus, SAT[5.5] would result in 5, while SAT[4.99] would result in 4. Using an additive correction of 0.5, SAT[a + 0.5] can be used as a rounding-off function where, for example, (a) SAT[5.5+0.5] = SAT[6.0] = 6 = *round*(5.5) and (b) SAT[4.99+.5] = SAT[5.49] = 5 = *round*[4.99].

Equation (2) can be modified to accommodate the above-described saturation and additive correction, and also incorporate a normalization factor, as shown in Equation (4) below:

$$\mathrm{Re}(0) = \mathrm{Im}(0) = 0 \tag{4.1}$$

$$\mathrm{Re}(n) = SAT\left[2^{-16} + M \cdot x_n + \cos\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Re}(n-1) - \sin\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Im}(n-1)\right], \text{ and}$$

$$\mathrm{Im}(n) = SAT\left[2^{-16} + \sin\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Re}(n-1) + \cos\left(\frac{2\pi F}{F_s}\right) \cdot \mathrm{Im}(n-1)\right] \tag{4.2}$$

for $n = 1, 2, ..., N$

$$P(F) = SAT[2^{-16} + \mathrm{Re}(N)^2 + \mathrm{Im}(N)^2] \tag{4.3}$$

where M is a pre-calculated normalization factor, $SAT[\ ]$ represents a saturation function for truncating Q $c.31$ numbers to Q $c.15$ format, and 2^{-16} is an additive correction factor to make saturation function $SAT[\ ]$ operate like a rounding-off function. Normalization is a process of adjusting data points in order to have them fit some particular rule, and is commonly used in signal processing. Note that, in a recursive implementation, the calculation would start with trying to determine Re(N) and Im(N), which would involve determining Re(N-1) and Im(N-1), which would in turn involve determining Re(N-2) and Im(N-2), and so forth down to Re(0) and Im(0). In contrast, an iterative implementation would involve first calculating Re(1) and Im(1) and then using the results to calculate Re(2) and Im(2), and so forth up to Re(N) and Im(N).

A conventional DSP would require $3N$+O(1) clock cycles to calculate power $P(F)$ using Equation (4). It should be noted that O(1) is in "big O" notation and represents a function bound by a constant and not dependent on N. Thus, for a 40-sample data frame (*i.e.*, $N = 40$) assuming, for example, O(1) = 10, a conventional DSP would require 130 clock cycles to calculate power $P(F)$ for the data frame using Equation (4). The DSP requires 2 clock cycles to perform multiplication operations, including multiply-and-accumulate (MAC) operations and multiply-and-subtract (MSU) operations. A MAC instruction operates such that MAC (a, b, c) adds the product of a and b to c, *i.e.*, $c = c + a \cdot b$. An MSU instruction operates such that MSU (a, b, c) subtracts the product of a and b from c, *i.e.*, $c = c - a \cdot b$. Since $\cos(2\pi F/F_s)$ and $\sin(2\pi F/F_s)$ are constants, they can be pre-calculated once and stored for use by each iteration. Each iteration of Equation (4.2) requires several MAC and MSU operations that take 2 clock cycles per iteration. Saturation requires another clock cycle per iteration. Equation (4.2) could be modified to remove the saturation function. This will make the procedure unstable and therefore liable to overflow and provide erroneous results, but would also reduce the number of cycles required for processing an N-sample frame to $2N$ + O(1) clock cycles.

Equation (3) can also be modified to accommodate the above-described saturation, additive correction, and normalization, as shown in Equation (5) below:

$$S(0) = S(-1) = 0 \tag{5.1}$$

$$S(n) = SAT\left[2^{-16} + M \cdot x_n + 2\cos\left(\frac{2\pi F}{F_s}\right) \cdot S(n-1) - S(n-2)\right] \tag{5.2}$$

for $n = 1, 2, ..., N$

$$P(F) = SAT\left[2^{-16} + S(N)^2 - 2\cos\left(\frac{2\pi F}{F_s}\right) \cdot S(N) \cdot S(N-1) + S(N-1)^2\right] \tag{5.3}$$

where M is a pre-calculated normalization factor, $SAT[\ ]$ represents a saturation function for truncating Q $c.31$ numbers to Q $c.15$ format, and 2^{-16} is an additive correction.

A conventional DSP would require at least 4N+O(1) clock cycles to calculate power $P(F)$ in accordance with Equation (5). It should be noted that, in some implementations, the DSP can store $2\cos(2\pi F/F_s)$ as a constant only if $\cos(2\pi F/F_s)$ is substantially between -0.5 and 0.5; otherwise, that DSP stores $\cos(2\pi F/F_s)$ as a constant and multiplies it by 2 in every iteration. In other implementations, the DSP can store $2\cos(2\pi F/F_s)$ as a constant regardless of the value of $\cos(2\pi F/F_s)$. Assuming $2\cos(2\pi F/F_s)$ is stored as a constant, each iteration of Equation (5.2) requires MAC, multiplication, subtraction, and saturation operations that take 3 clock cycles per sample. Saturation requires another clock cycle per iteration. Equation (5.2) could be modified to remove the saturation function. However, that will make the procedure unstable and therefore liable to overflow and provide erroneous results, but would also reduce the number of cycles required for processing an N-sample frame to $3N + O(1)$ clock cycles.

Some DSPs have multiple arithmetic logic units (ALUs) and multiple input/output (I/O) units that can process multiple instructions in a single clock cycle utilizing the multiple ALUs, multiple I/O units, and a pipeline architecture. A pipeline architecture allows the preparation of variables for a next iteration during a current iteration. Thus, after the first few clock cycles of a data set during which the pipeline is loaded and before the last few clock cycles of the data set during which the pipeline is unloaded, the DSP operates with a loaded pipeline, where extra clock cycles are not needed to load data for use by the ALUs since the pipeline is continually loaded as the ALUs perform their arithmetic operations. Conventional tone-power calculation systems might not make optimal use of these features of such DSPs.

SUMMARY OF THE INVENTION

One embodiment of the invention can be a digital signal processor (DSP) comprising a plurality D of arithmetic logic units (ALUs). The DSP is adapted to (a) receive a data frame comprising digital samples corresponding to an audio signal and (b) perform at least

one of tone detection and silence detection for the data frame using the plurality of ALUs in parallel. The tone detection comprises (i) determining a power $P(F)$ of a frequency F for the data frame and (ii) thresholding the determined power $P(F)$ to determine whether a tone corresponding to the frequency F is present in the audio signal. The silence detection comprises filtering a given frequency range of the signal.

BRIEF DESCRIPTION OF THE DRAWINGS

Other aspects, features, and advantages of the present invention will become more fully apparent from the following detailed description, the appended claims, and the accompanying drawings in which like reference numerals identify similar or identical elements.

FIG. 1 shows a simplified block diagram of a tone-detection system in accordance with one embodiment of the present invention.

FIG. 2 shows a simplified block diagram of an implementation of the tone-power calculation module of **FIG. 1**.

FIG. 3 shows a flowchart for an exemplary implementation of the processing of a 40-sample frame.

FIG. 4 shows a simplified block diagram of a tone-detection system in accordance with another embodiment of the invention.

DETAILED DESCRIPTION

FIG. 1 shows a simplified block diagram of tone-detection system **100** in accordance with one embodiment of the invention. Tone-detection system **100** comprises pre-processor **101**, tone-power calculation module **102**, threshold comparator **103**, controller **104**, and post-processor **105**. Pre-processor **101** receives a data frame of N samples via data bus **101a**. Pre-processor **101** performs appropriate pre-processing, such as filtering and automatic gain control, for the data frame and provides to tone-power calculation module **102**, via data bus **101b**, the pre-processed frame of length N in the form of samples $x_1, x_2, ..., x_N$. Tone-power calculation module **102** determines power $P(F)$ for the frame, and provides that power $P(F)$ to threshold comparator **103** via data bus **102a**. Threshold comparator **103** determines whether or not power $P(F)$ meets a pre-determined threshold. Threshold comparator **103** then outputs a corresponding yes or no via path **103a** to post-processor **105**, which uses that determination to generate output **105a**, which is the output of tone-detection system **100**.

Controller **104** is connected to and controls the various components of tone-detection system **100**.

FIG. 2 shows a simplified block diagram of an implementation of tone-power calculation module **102** of **FIG. 1**. A frame of N samples $x_1, x_2, ..., x_N$ is provided to data registers module **210** via data bus **101b**. The samples are provided as needed. The samples are accessed by ALUs **206**, **207**, **208**, and **209**, which use the sample values stored in data registers module **210** to calculate power $P(F)$ for the frame. Only a few sample values need to be in data registers module **210** during any clock cycle. The samples are pipelined through data registers module **210** as the calculation progresses. Data registers module **210** comprises registers used to store intermediate values used in calculating power $P(F)$. The flow of data in data registers module **210** and the operations performed by ALUs **206**, **207**, **208**, and **209** are determined by DSP controller **104** of **FIG. 1**, which provides instructions to tone-power calculation module **102** via path **102b**. Tone-power calculation module **102** communicates with electronic memory **211** for storing and retrieving information via path ACc.

The tone-power calculation formula of Equation (4) can be transformed into the formulation of Equation (6) below for an accelerated calculation of power $P(F)$ that requires only $0.75N + O(1)$ clock cycles:

$$\mathrm{Re}_1(0) = 2^{-16} \cdot \sum_{k=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}k\right) - \sin\left(\frac{2\pi F}{F_s}k\right)\right]$$

$$\mathrm{Im}_1(0) = 2^{-16} \cdot \sum_{k=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}k\right) + \sin\left(\frac{2\pi F}{F_s}k\right)\right]$$

$$\mathrm{Re}_2(0) = \mathrm{Im}_2(0) = 0$$

$$\text{for } n = 1,2,\ldots,\frac{N}{4} \begin{cases} V_a(n) = M \cdot x_{4n-4+a} \quad \text{for } a = 1,2,3,4 & (6.1)-(6.4) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n-1) + V_1(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n+3)\right) & (6.5) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n-1) + V_1(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n+3)\right) & (6.6) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n-1) + V_2(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n+2)\right) & (6.7) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n-1) + V_2(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n+2)\right) & (6.8) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) + V_3(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n+1)\right) & (6.9) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n) + V_3(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n+1)\right) & (6.10) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) + V_4(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n)\right) & (6.11) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n) + V_4(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n)\right) & (6.12) \end{cases}$$

$$P(F) = SAT\left[2^{-16} + (SAT[\mathrm{Re}_1(N/4) + \mathrm{Re}_2(N/4)])^2 + (SAT[\mathrm{Im}_1(N/4) + \mathrm{Im}_2(N/4)])^2\right] \quad (6.13)$$

It should be noted that the additive correction here (2^{-16}) is multiplied in the Equation (6) algorithm as an operational shortcut; it still functions as an additive correction in the algorithm. Equation (6) runs only one fourth of the iterations of Equation (4), with each iteration still requiring 3 clock cycles, thereby reducing the clock cycles for a frame of length N from $3N + O(1)$ to $0.75N + O(1)$. Multiple operations can be performed in each clock cycle by using parallel processing, *i.e.*, simultaneously using ALUs **206**, **207**, **208**, and **209** of **FIG. 2**. $\mathrm{Re}_1(n)$, $\mathrm{Re}_2(n)$, $\mathrm{Im}_1(n)$, and $\mathrm{Im}_2(n)$ refer to various intermediate values calculated on the way to calculating the values $\mathrm{Re}_1(N/4)$, $\mathrm{Re}_2(N/4)$, $\mathrm{Im}_1(N/4)$, and $\mathrm{Im}_2(N/4)$, which are determined in the final four steps, *i.e.*, steps (6.9) – (6.12), of iteration number N/4, and which are used to calculate power $P(F)$ for the N-sample frame.

For example, for a 40-sample frame, the equations for the first iteration, *i.e.*, where $n = 1$, are shown below:

$$V_1(1) = M \cdot x_1 \quad (6.1)$$

$$V_2(1) = M \cdot x_2 \quad (6.2)$$

$$V_3(1) = M \cdot x_3 \quad (6.3)$$

$$V_4(1) = M \cdot x_4 \quad (6.4)$$

$$\mathrm{Re}_1(1) = \mathrm{Re}_1(0) + V_1(1) \cdot \cos\left(\frac{2\pi F}{F_s}(39)\right) \quad (6.5)$$

$$\mathrm{Im}_1(1) = \mathrm{Im}_1(0) + V_1(1) \cdot \sin\left(\frac{2\pi F}{F_s}(39)\right) \quad (6.6)$$

$$\mathrm{Re}_2(1) = \mathrm{Re}_2(0) + V_2(1) \cdot \cos\left(\frac{2\pi F}{F_s}(38)\right) \quad (6.7)$$

$$\mathrm{Im}_2(1) = \mathrm{Im}_2(0) + V_2(1) \cdot \sin\left(\frac{2\pi F}{F_s}(38)\right) \quad (6.8)$$

$$\mathrm{Re}_1(1) = \mathrm{Re}_1(1) + V_3(1) \cdot \cos\left(\frac{2\pi F}{F_s}(37)\right) \quad (6.9)$$

$$\mathrm{Im}_1(1) = \mathrm{Im}_1(1) + V_3(1) \cdot \sin\left(\frac{2\pi F}{F_s}(37)\right) \quad (6.10)$$

$$\mathrm{Re}_2(1) = \mathrm{Re}_2(1) + V_4(1) \cdot \cos\left(\frac{2\pi F}{F_s}(36)\right) \quad (6.11)$$

$$\mathrm{Im}_2(1) = \mathrm{Im}_2(1) + V_4(1) \cdot \sin\left(\frac{2\pi F}{F_s}(36)\right) \quad (6.12)$$

Note that each iteration comprises 12 steps that can be divided into three sets of four steps. In the first four steps of each iteration, *i.e.*, steps (6.1) – (6.4), values are calculated for $V_1(n)$, $V_2(n)$, $V_3(n)$, and $V_4(n)$ based on the samples x_{4n-3}, x_{4n-2}, x_{4n-1}, and x_{4n}, respectively. In the middle four steps of each iteration, *i.e.*, steps (6.5) – (6.8), temporary values of $\mathrm{Re}_1(n)$, $\mathrm{Re}_2(n)$, $\mathrm{Im}_1(n)$, and $\mathrm{Im}_2(n)$ are calculated. In the final four steps of each iteration, *i.e.*, steps (6.9) – (6.12), the temporary values of $\mathrm{Re}_1(n)$, $\mathrm{Re}_2(n)$, $\mathrm{Im}_1(n)$, and $\mathrm{Im}_2(n)$ are overwritten with newly calculated intermediate values. These intermediate values may be stored in a data array. It is more efficient, however, to use memory-storage (MS) modules to hold the temporary and intermediate values of $\mathrm{Re}_1(n)$, $\mathrm{Re}_2(n)$ $\mathrm{Im}_1(n)$, and $\mathrm{Im}_2(n)$, overwriting the old values as new corresponding values are calculated based on the old values. The values and their corresponding variable names and MS modules are represented as Re_1, Re_2, Im_1, and Im_2. Depending on the particular implementation, MS modules could be implemented using data registers, cache, random access memory (RAM), EEPROM, and/or any other suitable memory. In **FIG. 2**, one or both of data registers module **210** and memory **211** comprise MS modules. In general, an MS module refers to an addressable segment of memory, where the

segment is addressable using an address, an offset, and/or any other suitable addressing technology.

In one implementation of tone-power calculation module **102**, MS modules Re_1, Re_2, Im_1, and Im_2 are updated with each iteration, based on the old values of the MS modules, as reflected by the formulation of Equation (7) below, which is based on Equations (6.1) – (6.12):

$$initialize\begin{cases} \mathrm{Re}_1 = 2^{-16}\cdot\sum_{k=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}k\right)-\sin\left(\frac{2\pi F}{F_s}k\right)\right] \\ \mathrm{Im}_1 = 2^{-16}\cdot\sum_{k=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}k\right)+\sin\left(\frac{2\pi F}{F_s}k\right)\right] \\ \mathrm{Re}_2 = \mathrm{Im}_2 = 0 \end{cases}$$

$$for\quad n=1,2,\ldots,N/4\begin{cases} V_a = M\cdot x_{4n-4+a} \quad a=1,2,3,4 \qquad (7.1)-(7.4) \\ \mathrm{Re}_1 = \mathrm{Re}_1 + V_1\cdot\cos\left(\frac{2\pi F}{F_s}(N-4n+3)\right) \qquad (7.5) \\ \mathrm{Im}_1 = \mathrm{Im}_1 + V_1\cdot\sin\left(\frac{2\pi F}{F_s}(N-4n+3)\right) \qquad (7.6) \\ \mathrm{Re}_2 = \mathrm{Re}_2 + V_2\cdot\cos\left(\frac{2\pi F}{F_s}(N-4n+2)\right) \qquad (7.7) \\ \mathrm{Im}_2 = \mathrm{Im}_2 + V_2\cdot\sin\left(\frac{2\pi F}{F_s}(N-4n+2)\right) \qquad (7.8) \\ \mathrm{Re}_1 = \mathrm{Re}_1 + V_3\cdot\cos\left(\frac{2\pi F}{F_s}(N-4n+1)\right) \qquad (7.9) \\ \mathrm{Im}_1 = \mathrm{Im}_1 + V_3\cdot\sin\left(\frac{2\pi F}{F_s}(N-4n+1)\right) \qquad (7.10) \\ \mathrm{Re}_2 = \mathrm{Re}_2 + V_4\cdot\cos\left(\frac{2\pi F}{F_s}(N-4n)\right) \qquad (7.11) \\ \mathrm{Im}_2 = \mathrm{Im}_2 + V_4\cdot\sin\left(\frac{2\pi F}{F_s}(N-4n)\right) \qquad (7.12) \end{cases}$$

$$P(F) = SAT\left[2^{-16} + (SAT[\mathrm{Re}_1+\mathrm{Re}_2])^2 + (SAT[\mathrm{Im}_1+\mathrm{Im}_2])^2\right] \qquad (7.13)$$

After variables Re_1, Re_2, Im_1, and Im_2 are initialized, each iteration of N/4 iterations requires 12 arithmetic operations, which are carried out by tone-power calculation module **102**. Operations (7.1) – (7.4) can be carried out together in parallel and in one cycle by ALUs **206**, **207**, **208**, and **209**. Operations (7.5) – (7.8) can also be carried out together in parallel and in

one cycle by ALUs **206**, **207**, **208**, and **209**. And operations (7.9) – (7.12) can also be carried out together in parallel and in one cycle by ALUs **206**, **207**, **208**, and **209**. An exemplary description of the operation of tone-power calculation module **102** processing 40-sample data frames is provided below.

The values of $\cos(2\pi F \cdot k/F_s)$ and $\sin(2\pi F \cdot k/F_s)$ for $k = 1$ to 40, which are constant (assuming F, F_s, and N are not changed between frames), used in every frame, and do not depend on particular data-sample (*i.e.*, x_n) values, may be calculated before any frame is fully processed, and the calculated values may then be stored in memory for use in subsequent frames. The value of normalization constant M may similarly be stored in memory **211**. These constants may then be read into data registers **210** via path ACc as needed by tone-power calculation module **102**. Storing frequently used constants in memory and using them as necessary, rather than recalculating their value every time they are needed, reduces the number of calculations needed per frame and, consequently, frame-processing time.

FIG. 3 shows flowchart **300** for an exemplary implementation for the processing of a 40-sample frame using the algorithm of Equation (7). Frame processing commences with getting the frame for processing (step **301**). Next, Re_1, Re_2, Im_1, and Im_2 are initialized (step **302**). The initialization values in every frame for variables Re_1, Re_2, Im_1, and Im_2 are constant. The initial values of Re_2 and Im_2 are 0, while Re_1 and Im_1 are initialized using the calculated values of $\cos(2\pi F \cdot k/F_s)$ and $\sin(2\pi F \cdot k/F_s)$. Since the initial values of Re_1 and Im_1 are the same for every frame, they may also be calculated before any frames are received and stored in memory for retrieval and use as needed. The initial values of Re_2 and Im_2 may also be stored in memory, but since those initial values are 0, they may simply be set to 0 at the start of processing of every frame (step **302**). After variables Re_1, Re_2, Im_1, and Im_2 are initialized for the frame, the process enters the first of 10 iterations for the 40-sample frame (steps **303-307**). Each iteration processes four consecutive samples of the forty samples of the frame and proceeds as described below.

First, the four samples are retrieved (step **303**). Next, V_1, V_2, V_3, and V_4 are calculated for the four samples, in parallel, using ALUs **206**, **207**, **208**, and **209**, in accordance with Equations (7.1) – (7.4) (step **304**). The calculated values are then stored in the appropriate MS modules. Next, ALUs **206**, **207**, **208**, and **209** calculate, in parallel, new values for variables Re_1, Re_2, Im_1, and Im_2 based on V_1 and V_2, in accordance with Equations (7.5) – (7.8) (step **305**). Then, ALUs **206**, **207**, **208**, and **209** calculate, in parallel, newer values for variables Re_1, Re_2, Im_1, and Im_2 based on V_3 and V_4,in accordance with Equations (7.9) –

(7.12) (step **306**). If there are more samples in the frame, then the next iteration is commenced (step **307**). After all the iterations are completed, power $P(F)$ is calculated based on the final values of variables Re_1, Re_2, Im_1, and Im_2 for the frame, in accordance with Equation (7.13) (step **308**). The value of power $P(F)$ is then provided to threshold comparator **103** of **FIG. 1** via path **102a**, and the process terminates for the frame (step **309**).

As would be appreciated by one of ordinary skill in the art, the above-described embodiment may be implemented in a variety of ways without departing from the scope of the invention. One typical implementation is defined using Assembler code. The Assembler code may define N/4 loops or may combine iterations to have fewer loops. Particular programming choices may vary depending on implementation-specific factors such as the number of registers available in data registers module **210**, the number format for numbers stored in those registers, the number format for numbers stored in memory **211**, and other factors specific to tone-power calculation module **102** and related modules.

In one alternative embodiment, $N/8$ iterations are used to process an N-sample frame of samples $x_1, x_2, \ldots, x_N$. The corresponding tone-power calculation, based on Equation (4), is represented in Equation (8) below, where $\alpha = 2\pi F/F_s$, M is a normalization factor, 2^{-16} is an additive correction factor, and $SAT[\,]$ is a saturation function.

$$\mathrm{Re}(0) = \mathrm{Im}(0) = 0 \tag{8.1}$$

$$\text{for } n = 1,2,\ldots,N/8 \quad \begin{cases} \mathrm{Re}_1(n) = 2^{-16}\cdot\sum_{k=0}^{3}[\cos(k\alpha)-\sin(k\alpha)]+\sum_{k=1}^{3}[\cos(k\alpha)\cdot M\cdot x_{8n-4-k}]+ \\ \qquad \cos(4\alpha)\cdot\mathrm{Re}(n-1)-\sin(4\alpha)\cdot\mathrm{Im}(n-1)+M\cdot x_{8n-4} \\ \mathrm{Im}_1(n) = 2^{-16}\cdot\sum_{k=0}^{3}[\cos(k\alpha)+\sin(k\alpha)]+\sum_{k=1}^{3}[\sin(k\alpha)\cdot M\cdot x_{8n-4-k}]+ \\ \qquad \sin(4\alpha)\cdot\mathrm{Re}(n-1)+\cos(4\alpha)\cdot\mathrm{Im}(n-1) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}_1(n)],\quad \mathrm{Im}(n) = SAT[\mathrm{Im}_1(n)] \\ \\ \mathrm{Re}_2(n) = 2^{-16}\cdot\sum_{k=0}^{3}[\cos(k\alpha)-\sin(k\alpha)]+\sum_{k=1}^{3}[\cos(k\alpha)\cdot M\cdot x_{8n-k}]+ \\ \qquad \cos(4\alpha)\cdot\mathrm{Re}(n)-\sin(4\alpha)\cdot\mathrm{Im}(n)+M\cdot x_{8n} \\ \mathrm{Im}_2(n) = 2^{-16}\cdot\sum_{k=0}^{3}[\cos(k\alpha)+\sin(k\alpha)]+\sum_{k=1}^{3}[\sin(k\alpha)\cdot M\cdot x_{8n-k}]+ \\ \qquad \sin(4\alpha)\cdot\mathrm{Re}(n)+\cos(4\alpha)\cdot\mathrm{Im}(n) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}_2(n)],\quad \mathrm{Im}(n) = SAT[\mathrm{Im}_2(n)] \end{cases} \tag{8.2}$$

$$P(F) = \mathrm{Re}(N/8)^2 + \mathrm{Im}(N/8)^2 \tag{8.3}$$

A more-detailed formulation appears below as Equation (9).

$$\mathrm{Re}(0) = \mathrm{Im}(0) = 0$$

$$n = 1, 2, \ldots, N/8 \begin{cases} V_a(n) = M \cdot x_{8n-8+a} \quad a = 1,2,\ldots 8 & (9.1) - (9.8) \\ \mathrm{Re}_1(n) = 2^{-16} \cdot \sum_{k=0}^{3} [\cos(k\alpha) - \sin(k\alpha)] & (9.9) \\ \mathrm{Im}_1(n) = 2^{-16} \cdot \sum_{k=0}^{3} [\cos(k\alpha) + \sin(k\alpha)] & (9.10) \\ \mathrm{Re}_2(n) = 2^{-16} \cdot \sum_{k=0}^{3} [\cos(k\alpha) - \sin(k\alpha)] & (9.11) \\ \mathrm{Im}_2(n) = 2^{-16} \cdot \sum_{k=0}^{3} [\cos(k\alpha) + \sin(k\alpha)] & (9.12) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) + V_4(n) & (9.13) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) + \cos(2\alpha) \cdot V_2(n) & (9.14) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n) + \sin(2\alpha) \cdot V_2(n) & (9.15) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) + \cos(3\alpha) \cdot V_1(n) & (9.16) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n) + \sin(3\alpha) \cdot V_1(n) & (9.17) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) + \cos(\alpha) \cdot V_3(n) & (9.18) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n) + \sin(\alpha) \cdot V_3(n) & (9.19) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) + \cos(4\alpha) \cdot \mathrm{Re}(n-1) & (9.20) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n) + \cos(4\alpha) \cdot \mathrm{Im}(n-1) & (9.21) \\ \mathrm{Re}_1(n) = \mathrm{Re}_1(n) - \sin(4\alpha) \cdot \mathrm{Im}(n-1) & (9.22) \\ \mathrm{Im}_1(n) = \mathrm{Im}_1(n) + \sin(4\alpha) \cdot \mathrm{Re}(n-1) & (9.23) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}_1(n)] & (9.24) \\ \mathrm{Im}(n) = SAT[\mathrm{Im}1(n)] & (9.25) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) + V_8(n) & (9.26) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) + \cos(2\alpha) \cdot V_6(n) & (9.27) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n) + \sin(2\alpha) \cdot V_6(n) & (9.28) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) + \cos(3\alpha) \cdot V_5(n) & (9.29) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n) + \sin(3\alpha) \cdot V_5(n) & (9.30) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) + \cos(\alpha) \cdot V_7(n) & (9.31) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n) + \sin(\alpha) \cdot V_7(n) & (9.32) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) + \cos(4\alpha) \cdot \mathrm{Re}(n) & (9.33) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n) + \cos(4\alpha) \cdot \mathrm{Im}(n) & (9.34) \\ \mathrm{Re}_2(n) = \mathrm{Re}_2(n) - \sin(4\alpha) \cdot \mathrm{Im}(n) & (9.35) \\ \mathrm{Im}_2(n) = \mathrm{Im}_2(n) + \sin(4\alpha) \cdot \mathrm{Re}(n) & (9.36) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}_2(n)] & (9.37) \\ \mathrm{Im}(n) = SAT[\mathrm{Im}_2(n)] & (9.38) \end{cases}$$

$$P(F) = \mathrm{Re}(N/8)^2 + \mathrm{Im}(N/8)^2 \quad (9.39)$$

Assuming that constants are calculated, stored, and retrieved for later use, rather than re-calculated for each use, a DSP having four ALUs and two read/write operations per clock

cycle using the formulation represented by Equations (9.1)-(9.39) would process an N-sample frame in $N + O(1)$ clock cycles.

The formulations presented above, which refer to DSPs having 4 ALUs and 2 read/write operations per cycle, can be generalized for different multi-core (*i.e.*, having multiple processors) DSPs. For a DSP with D ALUs and at least 2 read/write operations per cycle, the formulation of Equation (10) below may be used, where, unless otherwise indicated, variables are used in substantially the same way as described above.

$$\mathrm{Re}_1(0) = 2^{-16} \cdot \sum_{m=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}m\right) - \sin\left(\frac{2\pi F}{F_s}m\right)\right]$$

$$\mathrm{Im}_1(0) = 2^{-16} \cdot \sum_{m=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}m\right) + \sin\left(\frac{2\pi F}{F_s}m\right)\right]$$

$$\mathrm{Re}_2(0) = \mathrm{Im}_2(0) = 0$$

...

$$\mathrm{Re}_D(0) = \mathrm{Im}_D(0) = 0$$

$$for \quad n = 0,1,...,\left[\frac{N}{D} - 1\right] \quad \begin{cases} V_a(n) = M \cdot x_{nD+a} \quad a = 1,2,...,D & (10.1) \\ \mathrm{Re}_1(n+1) = \mathrm{Re}_1(n) + V_1(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N - Dn - 1)\right) & (10.2) \\ \mathrm{Im}_1(n+1) = \mathrm{Im}_1(n) + V_1(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N - Dn - 1)\right) & (10.2') \\ \mathrm{Re}_2(n+1) = \mathrm{Re}_2(n) + V_2(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N - Dn - 2)\right) & (10.3) \\ \mathrm{Im}_2(n+1) = \mathrm{Im}_2(n) + V_2(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N - Dn - 2)\right) & (10.3') \\ ... \\ ... \\ \mathrm{Re}_D(n+1) = \mathrm{Re}_D(n) + V_D(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N - Dn - D)\right) & (10.D) \\ \mathrm{Im}_D(n+1) = \mathrm{Im}_D(n) + V_D(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N - Dn - D)\right) & (10.D') \end{cases}$$

$$P(F) = SAT\left[\begin{array}{l} 2^{-16} + (SAT[\mathrm{Re}_1(N/D) + \mathrm{Re}_2(N/D) + ... + \mathrm{Re}_D(N/D)])^2 \\ + (SAT[\mathrm{Im}_1(N/D) + \mathrm{Im}_2(N/D) + ... + \mathrm{Im}_D(N/D)])^2 \end{array}\right]$$

Using the above formulation, D Re(n) and D Im(n) variables are initialized. Then, N/D iterations are run to process the samples in the frame, where each iteration takes D samples from the N-sample frame and performs $2D$ arithmetic operations. After all the samples are processed, power $P(F)$ is calculated. Calculating power $P(F)$ using the above formulation on a DSP with D ALUs would require $3N/D + O(1)$ clock cycles. That is so

because, in each iteration, (a) the calculations of Equation (10.1) are performed in parallel in one clock cycle, then (b) the calculations of Equations (10.2)-(10.D) are performed in parallel in one clock cycle, and then (c) the calculations of Equations (10.2')-(10.D') are performed in parallel in one clock cycle.

A generic alternative implementation of the above formulation is represented by the formulation of Equation (11) below, where, unless otherwise indicated, the variables and functions retain the same meaning as above.

$$\mathrm{Re}(0) = \mathrm{Im}(0) = 0 \tag{11.1}$$

$$\text{for } n = 1,2,\ldots\frac{N}{2D} \quad \begin{cases} \mathrm{Re}_1(n) = 2^{-16} \cdot \sum_{s=0}^{D}[\cos(s\alpha) - \sin(s\alpha)] + \sum_{s=1}^{D}[\cos(s\alpha) \cdot M \cdot x_{2Dn-D-s}] + \\ \qquad \cos(D\alpha) \cdot \mathrm{Re}(n-1) - \sin(D\alpha) \cdot \mathrm{Im}(n-1) + M \cdot x_{2Dn-D} \\ \mathrm{Im}_1(n) = 2^{-16} \cdot \sum_{s=0}^{D}[\cos(s\alpha) + \sin(s\alpha)] + \sum_{s=1}^{D}[\sin(s\alpha) \cdot M \cdot x_{2Dn-D-s}] + \\ \qquad \sin(D\alpha) \cdot \mathrm{Re}(n-1) + \cos(D\alpha) \cdot \mathrm{Im}(n-1) \\ \mathrm{Re}(n) = SAT(\mathrm{Re}_1(n)), \quad \mathrm{Im}(n) = SAT(\mathrm{Im}_1(n)) \\ \\ \mathrm{Re}_2(n) = 2^{-16} \cdot \sum_{s=0}^{D}[\cos(s\alpha) - \sin(s\alpha)] + \sum_{s=1}^{D}[\cos(s\alpha) \cdot M \cdot x_{2Dn-s}] + \\ \qquad \cos(D\alpha) \cdot \mathrm{Re}(n) - \sin(D\alpha) \cdot \mathrm{Im}(n) + M \cdot x_{2Dn} \\ \mathrm{Im}_2(n) = 2^{-16} \cdot \sum_{s=0}^{D}[\cos(s\alpha) + \sin(s\alpha)] + \sum_{s=1}^{D}[\sin(s\alpha) \cdot M \cdot x_{2Dn-s}] + \\ \qquad \sin(D\alpha) \cdot \mathrm{Re}(n) + \cos(D\alpha) \cdot \mathrm{Im}(n) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}_2(n)], \quad \mathrm{Im}(n) = SAT[\mathrm{Im}_2(n)] \end{cases} \tag{11.2}$$

$$P(F) = \mathrm{Re}(N/2D)^2 + \mathrm{Im}(N/2D)^2 \tag{11.3}$$

A more-detailed formulation appears in Equation (12) below, where, unless otherwise noted, variables retain the same meaning as above.

$$\mathrm{Re}(0) = \mathrm{Im}(0) = 0$$

$$\text{for } n = 1, 2, \ldots, N/2D \quad \begin{cases} V_a(n) = M \cdot x_{2Dn-2D+s} \quad a = 1,2,\ldots 2D & (12.1.1 - 12.1.2D) \\ \mathrm{Re}1(n) = 2^{-16} \cdot \sum_{s=0}^{D-1} [\cos(s\alpha) - \sin(s\alpha)] & (12.2.1) \\ \mathrm{Im}1(n) = 2^{-16} \cdot \sum_{s=0}^{D-1} [\cos(s\alpha) + \sin(s\alpha)] & (12.2.2) \\ \mathrm{Re}2(n) = 2^{-16} \cdot \sum_{s=0}^{D-1} [\cos(s\alpha) - \sin(s\alpha)] & (12.2.3) \\ \mathrm{Im}2(n) = 2^{-16} \cdot \sum_{s=0}^{D-1} [\cos(s\alpha) + \sin(s\alpha)] & (12.2.4) \\ \mathrm{Re}1(n) = \mathrm{Re}1(n) + V_D(n) & (12.3.1) \\ \mathrm{Re}1(n) = \mathrm{Re}1(n) + \cos(\alpha) \cdot V_{D-1}(n) & (12.3.2) \\ \mathrm{Im}1(n) = \mathrm{Im}1(n) + \sin(\alpha) \cdot V_{D-1}(n) & (12.3.3) \\ \ldots \\ \ldots \\ \mathrm{Re}1(n) = \mathrm{Re}1(n) + \cos((D-1)\alpha) \cdot V_1(n) & (12.3.2D-2) \\ \mathrm{Im}1(n) = \mathrm{Im}1(n) + \sin((D-1)\alpha) \cdot V_1(n) & (12.3.2D-1) \\ \mathrm{Re}1(n) = \mathrm{Re}1(n) + \cos(D\alpha) \cdot \mathrm{Re}(n-1) & (12.4.1) \\ \mathrm{Im}1(n) = \mathrm{Im}1(n) + \cos(D\alpha) \cdot \mathrm{Im}(n-1) & (12.4.2) \\ \mathrm{Re}1(n) = \mathrm{Re}1(n) - \sin(D\alpha) \cdot \mathrm{Im}(n-1) & (12.4.3) \\ \mathrm{Im}1(n) = \mathrm{Im}1(n) + \sin(D\alpha) \cdot \mathrm{Re}(n-1) & (12.4.4) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}1(n)] & (12.5.1) \\ \mathrm{Im}(n) = SAT[\mathrm{Im}1(n)] & (12.5.2) \\ \mathrm{Re}2(n) = \mathrm{Re}2(n) + V_{2D}(n) & (12.6.1) \\ \mathrm{Re}2(n) = \mathrm{Re}2(n) + \cos(\alpha) \cdot V_{2D-1}(n) & (12.6.2) \\ \mathrm{Im}2(n) = \mathrm{Im}2(n) + \sin(\alpha) \cdot V_{2K-1}(n) & (12.6.3) \\ \ldots \\ \ldots \\ \mathrm{Re}2(n) = \mathrm{Re}2(n) + \cos((D-1)\alpha) \cdot V_{D+1}(n) & (12.6.2D-2) \\ \mathrm{Im}2(n) = \mathrm{Im}2(n) + \sin((D-1)\alpha) \cdot V_{D+1}(n) & (12.6.2D-1) \\ \mathrm{Re}2(n) = \mathrm{Re}2(n) + \cos(D\alpha) \cdot \mathrm{Re}(n) & (12.7.1) \\ \mathrm{Im}2(n) = \mathrm{Im}2(n) + \cos(D\alpha) \cdot \mathrm{Im}(n) & (12.7.2) \\ \mathrm{Re}2(n) = \mathrm{Re}2(n) - \sin(D\alpha) \cdot \mathrm{Im}(n) & (12.7.3) \\ \mathrm{Im}2(n) = \mathrm{Im}2(n) + \sin(D\alpha) \cdot \mathrm{Re}(n) & (12.7.4) \\ \mathrm{Re}(n) = SAT[\mathrm{Re}2(n)] & (12.8.1) \\ \mathrm{Im}(n) = SAT[\mathrm{Im}2(n)] & (12.8.2) \end{cases}$$

$$P(F) = \mathrm{Re}(N/2D)^2 + \mathrm{Im}(N/2D)^2$$

FIG. 4 shows a simplified block diagram of tone-detection system **400** in accordance with another embodiment of the invention. Tone-detection system **400** contains many of the same modules as tone-detection system **100** of **FIG. 1**, but with the addition of silence-

detection module **412**, which receives frames of N pre-processed samples from pre-processor **101** via path **101b** and conditionally provides those frames to tone-power calculation module **102** via path **412a**. Silence-detection module **412** determines whether a received frame substantially represents silence. If a frame is determined to be substantially silence, then that frame is not provided to tone-power calculation module **102**. Otherwise, the frame is provided to tone-power calculation module **102** to determine the presence of particular tones, as described above. It should be noted that silence-detection module **412** and tone-power calculation module **102** may be implemented using different DSPs or the same DSP, wherein the DSP functions as both silence-detection module **412** and tone-power calculation module **102**. For example, a DSP, functioning as a silence-detection module, can receive a frame and first determine whether it is substantially silence and, if it determines that the frame is not silence, then the DSP, functioning as a tone-power calculation module, can calculate the power of a particular frequency in the frame.

In accordance with one implementation of silence-detection module **412**, silence-detection module **412** reliably detects whether the energy in the 200Hz-3400Hz band is below -34dBm. A second-order bypass filter to achieve that can be represented by Equation (13) below:

$$z_n = 0.825 \cdot x_n - 0.825 \cdot x_{n-2} - 0.16 \cdot z_{n-1} - 0.6499 \cdot z_{n-2} \tag{13}$$

If we denote $B_0 = 0.825$, $A_1 = -0.16$, and $A_2 = -0.6499$ and use saturation and additive correction (*e.g.*, where x_k and z_k are in Q*c*.15 format, and arithmetic operations provide Q*c*.31 results), then we can represent Equation (13) for an N-sample frame according to Equation (14) as follows:

$$z_n = 2^{-16} + B_0 \cdot x_n - B_0 \cdot x_{n-2} + A_1 \cdot z_{n-1} + A_2 \cdot z_{n-2} \rceil \tag{14}$$

for $n = 1, 2, \ldots, N$

A prior-art DSP would require at least $2 \cdot N$ clock cycles to process an N-sample frame. One implementation of silence-detection module **412** can process an N-sample frame, comprising samples $x_1, x_2, \ldots, x_N$, in $1.5 \cdot N + O(1)$ clock cycles using a filter represented by the algorithm of Equation (15) below, where the terms have the values described above.

$$z_n = 2^{-16} + B_0 \cdot x_n - B_0 \cdot x_{n-2} + A_1 \cdot z_{n-1} + A_2 \cdot z_{n-2} \tag{15.1}$$

$$z_{n-1} = 2^{-16} + B_0 \cdot x_{n-1} - B_0 \cdot x_{n-3} + A_1 \cdot z_{n-2} + A_2 \cdot z_{n-3} \tag{15.2}$$

for $n = 2, 4, 6, \ldots, N$

Or, alternatively, the implementation can be represented by the algorithm of Equation (16) below, where the term $A_{1\cdot2}$ represents $A_1 \cdot A_2$, the term $A_{1\cdot1+2}$ represents $A_1 \cdot A_1 + A_2$, and 2^{-16} is an additive correction.

$$z_n = 2^{-16} + B_0 \cdot x_n - B_0 \cdot x_{n-2} + A_1 \cdot (2^{-16} + B_0 \cdot x_{n-1} - B_0 \cdot x_{n-3}) + A_{1\cdot1+2} \cdot z_{n-2} + A_{1\cdot2} \cdot z_{n-3} \quad (16.1)$$

$$z_{n-1} = 2^{-16} + B_0 \cdot x_{n-1} - B_0 \cdot x_{n-3} + A_1 \cdot z_{n-2} + A_2 \cdot z_{n-3} \quad (16.2)$$

for $n = 2, 4, 6, \ldots, N$

The algorithm represented by Equation (16) takes samples x_n, x_{n-1}, x_{n-2}, x_{n-3}, and results z_{n-2}, z_{n-3} and iteratively calculates z_n and z_{n-1}. Each iteration of the algorithm takes up three clock cycles. As a result, the total number of clock cycles required to process an N-sample frame is $1.5 \cdot N + O(1)$. Silence-detection module **412** can implement Equation (16) by processing 3 iterations of the algorithm together in 9 clock cycles. In other words, 6 inputs (4 samples and 2 results) are handled every 9 clock cycles. A more-detailed representation of Equation (16) for silence-detection module **412** is provided as Equation (17) below.

$$q_n = 2^{-16} \quad (17.1)$$

$$q_{n-1} = 2^{-16} \quad (17.2)$$

$$q_n = q_n - B_0 \cdot x_{n-2} \quad (17.3)$$

$$q_{n-1} = q_{n-1} - B_0 \cdot x_{n-3} \quad (17.4)$$

read x_{n-1} and x_n (17.5)

$$q_n = q_n + B_0 \cdot x_n \quad (17.6)$$

$$q_{n-1} = q_{n-1} + B_0 \cdot x_{n-1} \quad (17.7)$$

$$z_n = q_n \quad (17.8)$$

$$z_{n-1} = q_{n-1} \quad (17.9)$$

$$z_n = z_n + A_1 \cdot z_{n-1} \quad (17.10)$$

$$z_n = z_n + A_{1\cdot2} \cdot z_{n-3} \quad (17.11)$$

$$z_{n-1} = z_{n-1} + A_2 \cdot z_{n-3} \quad (17.12)$$

$$z_n = z_n + A_{1\cdot1+2} \cdot z_{n-2} \quad (17.13)$$

$$z_{n-1} = z_{n-1} + A_1 \cdot z_{n-2} \quad (17.14)$$

write z_{n-1} and z_n (17.15)

for $n = 2, 4, 6, \ldots, N$

Silence-detection module **412** performs operations (17.1)-(17.15) as shown as Equation (18) below, where (a) the square brackets denote a set of operations that are executed in parallel in one clock cycle and (b) the subscript indexes (*e.g.*, "*s*," "*s*-1," and "*s*-2") mean indexes of iterations. For example, operations (17.1) and (17.2) of the iteration $n = s$ are performed together with (*i.e.*, in the same clock cycle as) operation (17.8) and (17.9) of the iteration $n = s\text{-}2$ and operations (17.11) and (17.12) of the iteration $n = s\text{-}4$.

$$\begin{array}{llll} [& (17.1)_s, (17.2)_s, & (17.8)_{s-2}, (17.9)_{s-2}, & (17.11)_{s-4}, (17.12)_{s-4} \ ] \\ [& (17.3)_s, (17.4)_s, (17.5)_s, & & (17.13)_{s-4}, (17.14)_{s-4} \ ] \\ [& (17.6)_s, (17.7)_s, & (17.10)_{s-2}, & (17.15)_{s-4} \ ] \end{array} \quad (18)$$

It should be noted that the above-described implementation for silence-detection module **412** is designed for N-sample frames where $N \bmod 6 = 4$. For example, for $N = 40$, $40 \bmod 6 = 4$. As would be appreciated by one of ordinary skill in the art, the described algorithm may be modified for frames of other lengths.

In an alternative embodiment of silence-detection module **412**, silence-detection module **412** uses a multi-processor DSP having m ALUs and multiple read-write operations per clock cycle that is able to perform multiple operations in parallel. A filter for such a multi-ALU DSP can be represented by the formulation of Equation (19) below:

$$\begin{aligned} z_n = 2^{-16}(1 + B_0 + B_1 + \ldots + B_{m-3}) + B_0(x_n - x_{n-2}) + B_1(x_{n-1} - x_{n-3}) \\ + B_2(x_{n-2} - x_{n-4}) + \ldots + B_{m-3}(x_{n-m-3} - x_{n-m-5}) + B_{m-2} \cdot z_{n-m+2} + \\ B_{m-1} \cdot z_{n-m+1} \end{aligned} \quad (19)$$

Since B_i does not depend on n, the values of B_i may be pre-calculated and stored in a memory for use as needed. Values for B_i may be calculated recursively using the algorithm of Equation (20) below.

$$B_0 = B_0, \quad (20.1)$$

$$B_1 = A_1 \cdot B_0 \quad (20.2)$$

For $i = 2$ to m-3,

$$B_i = A_1 \cdot B_{i-1} + A_2 \cdot B_{i-2} \quad (20.3)$$

Thus, for example, $B_2 = A_2 \cdot B_0 + (A_1)^2 \cdot B_0$ and $B_3 = B_0 \cdot (2 \cdot A_1 \cdot A_2 + (A_1)^3)$.

Equation (19) can be transformed for implementation using silence-detection module **412** according to Equation (21) as follows, where q_n is the output:

$$q_n = 2^{-16}(1+ B_0+ B_1+\ldots+ B_{m-3}) \quad (21.1)$$

$$q_n = q_n + B_{m-3}\cdot(x_{n-m-3} - x_{n-m-5}) \quad (21.2)$$

...

$$q_n = q_n + B_1\cdot (x_{n-1} - x_{n-3}) \quad (21.m\text{-}2)$$

$$\text{read } x_n, x_{n-2} \quad (21.m\text{-}1)$$

$$d_n = x_n - x_{n-2} \quad (21.m)$$

$$q_n = q_n + B_0\cdot d_n \quad (21.m\text{+}1)$$

$$q_n = q_n + B_{m+1}\cdot q_{n-m} \quad (21.m\text{+}2)$$

$$\text{write } q_n \quad (21.m\text{+}3)$$

$$\text{for } n = m + 1, m + 2, m + 3, ..., N$$

Silence-detection module **412** performs operations (21.1)-(21.*m*+3) as shown in Equation (22) below, where (a) the square brackets denote a set of operations that are executed in one clock cycle and (b) the subscript indexes (*e.g.*, "*s*," "*s*-1," and "*s*-2") mean indexes of iterations.

$$[(21.1)_s, (21.2)_{s-1}, (21.3)_{s-2,\ldots} \quad (21.m\text{+}2)_{s-m-1}, (21.m\text{+}3)_{s-m-2},] \quad (22)$$

It should be noted that operations (21.1), (21.*m-1*), and (21.*m*+3) are read/write operations, while the others are arithmetic operations. Note that operation (21.1) here is a read operation because the value of $2^{-16}(1+ B_0+ B_1+\ldots+ B_{m-3})$ is pre-calculated and operation (21.1) consists of loading that pre-calculated value into q_n. As can be seen, one clock cycle is needed to implement one step of the iteration. However, approximately m clock cycles are needed to get to the point where all the operations can be performed simultaneously since the operational pipeline needs to be filled. For example, if silence-detection module **412** uses 4 ALUs, then, on the second iteration (*e.g.*, $s = 2$), only operations from the first two iterations can be performed. Once $s >= m + 3$, all the algorithms of Equation (21) can be implemented simultaneously. Thus, a total of $N+m+3$ clock cycles are needed to process one N-sample frame of data. It should be further noted that silence-detection module **412** needs to be able to perform at least three read/write operations per clock cycle in this implementation.

Embodiments of the invention have been described in the context of DTMF telephony signals. The invention is not limited to DTMF tones or telephony applications. Some alternative embodiments are used for tone-power calculation and/or silence detection in contexts outside of DTMF signals. Some alternative embodiments are used for tone-power calculation and/or silence detection in contexts outside of telephony.

An embodiment of the invention has been described where the silence-detection module **412** of **FIG. 4** is used in conjunction with tone-power calculation module **102**. In an alternative implementation, silence-detection module **412** is used independently of a tone-power calculation module.

Embodiments of the invention have been described wherein a pre-processor, a threshold comparator, and a post-processor are used. These devices are not necessary for the invention. Alternative embodiments of the invention are implemented without one or more of a pre-processor, threshold comparator, and post-processor.

Embodiments of the invention have been described wherein the frames comprise 40 samples. As already noted, data frames of different sizes are possible. Alternative embodiments of the invention process data frames having a number of samples other than 40.

An embodiment of the invention has been described where the silence-detection module was shown as separate from the pre-processor. In an alternative embodiment, the silence-detection module is part of the pre-processor. In general, as also indicated elsewhere, the various modules described may be physically implemented in a wide variety of ways, as would be appreciated by one of ordinary skill in the art.

References herein to the verb "to set" and its variations in reference to values of fields do not necessarily require an active step and may include leaving a field value unchanged if its previous value is the desired value. Setting a value may nevertheless include performing an active step even if the previous or default value is the desired value.

As used herein in reference to data transfers between entities in the same device, and unless otherwise specified, the terms "receive" and its variants can refer to receipt of the actual data, or the receipt of one or more pointers to the actual data, wherein the receiving entity can access the actual data using the one or more pointers.

Exemplary embodiments have been described wherein particular entities (a.k.a. modules) perform particular functions. However, the particular functions may be performed by any suitable entity and are not restricted to being performed by the particular entities named in the exemplary embodiments.

References herein to the verb "to generate" and its variants in reference to information or data do not necessarily require the creation and/or storage of new instances of that information. The generation of information could be accomplished by identifying an accessible location of that information. The generation of information could also be accomplished by having an algorithm for obtaining that information from accessible other information.

As used herein in reference to an element and a standard, the term "compatible" means that the element communicates with other elements in a manner wholly or partially specified by the standard, and would be recognized by other elements as sufficiently capable of communicating with the other elements in the manner specified by the standard. The compatible element does not need to operate internally in a manner specified by the standard.

The present invention may be implemented as circuit-based processes, including possible implementation as a single integrated circuit (such as an ASIC or an FPGA), a multi-chip module, a single card, or a multi-card circuit pack. As would be apparent to one skilled in the art, various functions of circuit elements may also be implemented as processing steps in a software program. Such software may be employed in, for example, a digital signal processor, micro-controller, or general-purpose computer.

It will be further understood that various changes in the details, materials, and arrangements of the parts which have been described and illustrated in order to explain the nature of this invention may be made by those skilled in the art without departing from the scope of the invention as expressed in the following claims.

Reference herein to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment can be included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments necessarily mutually exclusive of other embodiments. The same applies to the term "implementation."

Unless explicitly stated otherwise, each numerical value and range should be interpreted as being approximate as if the word "about" or "approximately" preceded the value of the value or range. As used in this application, unless otherwise explicitly indicated, the term "connected" is intended to cover both direct and indirect connections between elements.

For purposes of this description, the terms "couple," "coupling," "coupled," "connect," "connecting," or "connected" refer to any manner known in the art or later developed in which energy is allowed to be transferred between two or more elements, and the interposition of one or more additional elements is contemplated, although not required. The terms "directly coupled," "directly connected," etc., imply that the connected elements are either contiguous or connected via a conductor for the transferred energy.

The use of figure numbers and/or figure reference labels in the claims is intended to identify one or more possible embodiments of the claimed subject matter in order to facilitate the interpretation of the claims. Such use is not to be construed as limiting the scope of those claims to the embodiments shown in the corresponding figures.

Although the steps in the following method claims are recited in a particular sequence with corresponding labeling, unless the claim recitations otherwise imply a particular sequence for implementing some or all of those steps, those steps are not necessarily intended to be limited to being implemented in that particular sequence.

CLAIMS

We claim:

1. A digital signal processor (DSP) comprising a plurality D of arithmetic logic units (ALUs), wherein the DSP is adapted to:

receive a data frame comprising digital samples corresponding to an audio signal; and

perform at least one of tone detection and silence detection for the data frame using the plurality of ALUs in parallel, wherein:

the tone detection comprises (i) determining a power $P(F)$ of a frequency F for the data frame and (ii) thresholding the determined power $P(F)$ to determine whether a tone corresponding to the frequency F is present in the audio signal; and

the silence detection comprises filtering a given frequency range of the signal.

2. The DSP of claim 1, wherein the DSP performs the tone detection for an N-sample data frame.

3. The DSP of claim 2, wherein the DSP is adapted to determine the power $P(F)$ by performing the following operations on the N-sample data frame:

(a) $V_a(n) = M \cdot x_{nD+a}$, for $a = 1, ..., D$ and $n = 0, ..., [N/D - 1]$;

(b) $\mathrm{Re}_a(n+1) = \mathrm{Re}_a(n) + V_a(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N - Dn - a)\right)$, for $a = 1, ..., D$ and $n = 0, ...,$ $[N/D - 1]$; and

(c) $\mathrm{Im}_a(n+1) = \mathrm{Im}_a(n) + V_a(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N - Dn - a)\right)$, for $a = 1, ..., D$ and $n = 0, ...,$ $[N/D - 1]$,

wherein $V_a(k)$, $\mathrm{Re}_a(k)$, and $\mathrm{Im}_a(k)$ are storage variables, F_s is a sampling frequency for the data frame, x_i is the ith sample of the data frame, a , k, and n are counting variables, and M is a normalization factor.

4. The DSP of claim 3, wherein the DSP is adapted to determine the power $P(F)$ in accordance with the equation $P(F) =$

$$SAT\left[A+\left(SAT\left[\mathrm{Re}_1\left(\frac{N}{D}\right)+\mathrm{Re}_2\left(\frac{N}{D}\right)+\ldots+\mathrm{Re}_D\left(\frac{N}{D}\right)\right]\right)^2+\left(SAT\left[\mathrm{Im}_1\left(\frac{N}{D}\right)+\mathrm{Im}_2\left(\frac{N}{D}\right)+\ldots+\mathrm{Im}_D\left(\frac{N}{D}\right)\right]\right)^2\right],$$

where SAT[ ] is a saturation function and A is an additive correction.

5. The DSP of claim 3, wherein the DSP is adapted to:

(a) initialize $\mathrm{Re}_1(0)$ to $A\cdot\sum_{m=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}m\right)-\sin\left(\frac{2\pi F}{F_s}m\right)\right]$;

(b) initialize $\mathrm{Im}_1(0)$ to $A\cdot\sum_{m=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}m\right)+\sin\left(\frac{2\pi F}{F_s}m\right)\right]$; and

(c) initialize $\mathrm{Re}_j(0)$ and $\mathrm{Im}_j(0)$ to 0 for $j = 2, \ldots, D$, wherein A is an additive correction and m and j are counting variables.

6. The DSP of claim 5, wherein the DSP is adapted to determine the power $P(F)$ in accordance with the equation $P(F) =$

$$SAT\left[A+\left(SAT\left[\mathrm{Re}_1\left(\frac{N}{D}\right)+\mathrm{Re}_2\left(\frac{N}{D}\right)+\ldots+\mathrm{Re}_D\left(\frac{N}{D}\right)\right]\right)^2+\left(SAT\left[\mathrm{Im}_1\left(\frac{N}{D}\right)+\mathrm{Im}_2\left(\frac{N}{D}\right)+\ldots+\mathrm{Im}_D\left(\frac{N}{D}\right)\right]\right)^2\right],$$

where SAT[ ] is a saturation function and A is an additive correction.

7. The DSP of claim 3, wherein the DSP is adapted to:

perform multiple read/write operations per clock cycle;

perform step (a) in one clock cycle for any one value of n by using all D ALUs in parallel;

perform step (b) in one clock cycle for any one value of n by using all D ALUs in parallel; and

perform step (c) in one clock cycle for any one value of n by using all D ALUs in parallel.

8. The DSP of claim 2, wherein:

$D = 4$;

the DSP is adapted to perform multiple read/write operations per clock cycle;

the DSP is adapted to determine the power $P(F)$ by performing the following operations for $n = 1, \ldots, N/4$, wherein the DSP reads a sequential 4-sample group from the N-sample data frame for each value of n:

(a) $V_a(n) = M \cdot x_{4n-4+a}$, for $a = 1, ..., 4$;

(b) $\mathrm{Re}_1(n) = \mathrm{Re}_1(n-1) + V_1(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n+3)\right)$;

(c) $\mathrm{Im}_1(n) = \mathrm{Im}_1(n-1) + V_1(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n+3)\right)$;

(d) $\mathrm{Re}_2(n) = \mathrm{Re}_2(n-1) + V_2(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n+2)\right)$;

(e) $\mathrm{Im}_2(n) = \mathrm{Im}_2(n-1) + V_2(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n+2)\right)$;

(f) $\mathrm{Re}_1(n) = \mathrm{Re}_1(n) + V_3(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n+1)\right)$;

(g) $\mathrm{Im}_1(n) = \mathrm{Im}_1(n) + V_3(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n+1)\right)$;

(h) $\mathrm{Re}_2(n) = \mathrm{Re}_2(n) + V_4(n) \cdot \cos\left(\frac{2\pi F}{F_s}(N-4n)\right)$; and

(i) $\mathrm{Im}_2(n) = \mathrm{Im}_2(n) + V_4(n) \cdot \sin\left(\frac{2\pi F}{F_s}(N-4n)\right)$, wherein $V_a(n)$, $\mathrm{Re}_1(n)$, $\mathrm{Re}_2(n)$, $\mathrm{Im}_1(n)$, and $\mathrm{Im}_2(n)$ are storage variables, F_s is the sampling frequency for the data frame, x_i is the ith sample of the data frame, a is a counting variable, and M is a normalization factor.

9. The DSP of claim 8, wherein the DSP is adapted to determine the power $P(F)$ in accordance with the equation $P(F) =$

$$SAT\left[A+\left(SAT\left[\mathrm{Re}_1\left(\frac{N}{4}\right)+\mathrm{Re}_2\left(\frac{N}{4}\right)\right]\right)^2+\left(SAT\left[\mathrm{Im}_1\left(\frac{N}{4}\right)+\mathrm{Im}_2\left(\frac{N}{4}\right)\right]\right)^2\right],$$

wherein SAT[ ] is a saturation function and A is an additive correction.

10. The DSP of claim 8, wherein the DSP is adapted to:

(a) initialize $\mathrm{Re}_1(0)$ to $A \cdot \sum_{m=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}m\right)-\sin\left(\frac{2\pi F}{F_s}m\right)\right]$;

(b) initialize $\mathrm{Im}_1(0)$ to $A \cdot \sum_{m=0}^{N-1}\left[\cos\left(\frac{2\pi F}{F_s}m\right)+\sin\left(\frac{2\pi F}{F_s}m\right)\right]$; and

(c) initialize $Re_2(0)$ and $Im_2(0)$ to 0, wherein A is an additive correction and m is a counting variable.

11. The DSP of claim 8, wherein the DSP is adapted to:

perform multiple read/write operations per clock cycle;

perform step (a) in one clock cycle for any one value of n by using all 4 ALUs in parallel;

perform steps (b) – (e) in one clock cycle for any one value of n by using all 4 ALUs in parallel; and

perform steps (f) – (i) in one clock cycle for any one value of n by using all 4 ALUs in parallel.

12. The DSP of claim 2, wherein the DSP is adapted to determine the power $P(F)$ by performing the following operations on the N-sample data frame:

(a) initialize Re(0) and Im(0) to 0;

(b) for $n = 1, ..., [N/2D]$ calculate:

(1) $$Re_1(n) = A \cdot \sum_{s=0}^{D}[\cos(s\alpha) - \sin(s\alpha)] + \sum_{s=1}^{D}[\cos(s\alpha) \cdot M \cdot x_{2Dn-D-s}] + \cos(D\alpha) \cdot Re(n-1) - \sin(D\alpha) \cdot Im(n-1) + M \cdot x_{2Dn-D};$$

(2) $$Im_1(n) = A \cdot \sum_{s=0}^{D}[\cos(s\alpha) + \sin(s\alpha)] + \sum_{s=1}^{D}[\sin(s\alpha) \cdot M \cdot x_{2Dn-D-s}] + \sin(D\alpha) \cdot Re(n-1) + \cos(D\alpha) \cdot Im(n-1);$$

(3) $Re(n) = SAT[Re_1(n)]$;

(4) $Im(n) = SAT[Im_1(n)]$;

(5) $$Re_2(n) = A \cdot \sum_{s=0}^{D}[\cos(s\alpha) - \sin(s\alpha)] + \sum_{s=1}^{D}[\cos(s\alpha) \cdot M \cdot x_{2Dn-s}] + \cos(D\alpha) \cdot Re(n) - \sin(D\alpha) \cdot Im(n) + M \cdot x_{2Dn};$$ and

(6) $$Im_2(n) = A \cdot \sum_{s=0}^{D}[\cos(s\alpha) + \sin(s\alpha)] + \sum_{s=1}^{D}[\sin(s\alpha) \cdot M \cdot x_{2Dn-s}] + \sin(D\alpha) \cdot Re(n) + \cos(D\alpha) \cdot Im(n);$$

(7) $Re(n) = SAT[Re_2(n)]$;

(8) $Im(n) = SAT[Im_2(n)]$; and

(c) $P(F) = Re(N/2D)^2 + Im(N/2D)^2$, where $Re(n)$, $Re_1(n)$, $Re_2(n)$, $Im(n)$, $Im_1(n)$, and $Im_2(n)$ are storage variables, s is a counting variable, A is an additive correction, M is a

normalization factor, SAT[ ] is a saturation function, and α is $2\pi F/F_s$, where F_s is a sampling frequency for the data frame.

13. The DSP of claim 1, wherein the DSP performs the silence detection for an N-sample data frame.

14. The DSP of claim 13, wherein the filtering uses the filter equation below:

$z_n = A\cdot(1+ B_0 + B_1 +\ldots+ B_{D-3}) + B_0(x_n - x_{n-2}) + B_1(x_{n-1} - x_{n-3}) + B_2(x_{n-2} - x_{n-4}) + \ldots + B_{D-3}(x_{n-D-3} - x_{n-D-5}) + B_{D-2}\cdot z_{n-D+2} + B_{D-1}\cdot z_{n-D+1}$, where A is an additive correction, B_i is the ith multiplicative factor, x_i is the ith sample of the data frame, and z_n is the filter output.

15. The DSP of claim 14, wherein:

(a) $B_0 = B_0$;

(b) $B_1 = A_1\cdot B_0$; and

(c) for $i = 2$ to D-3

$B_i = A_1\cdot B_{i-1} + A_2\cdot B_{i-2}$, where A_i and B_i are multiplicative factors.

16. The DSP of claim 15, wherein B_0 is about 0.825, A_1 is about -0.16, and A_2 is about -0.6499.

17. The DSP of claim 13, wherein the DSP comprises at least four ALUs and silence detection comprises performing the filtering operations below for $n = 2, 4, 6, \ldots, N$:

(1) $q_n = A$;

(2) $q_{n-1} = A$;

(3) $q_n = q_n - B_0\cdot x_{n-2}$;

(4) $q_{n-1} = q_{n-1} - B_0\cdot x_{n-3}$;

(5) read x_{n-1} and x_n;

(6) $q_n = q_n + B_0\cdot x_n$;

(7) $q_{n-1} = q_{n-1} + B_0\cdot x_{n-1}$;

(8) $z_n = q_n$;

(9) $z_{n-1} = q_{n-1}$;

(10) $z_n = z_n + A_1\cdot z_{n-1}$;

(11) $z_n = z_n + A_{1\cdot 2}\cdot z_{n-3}$;

(12) $z_{n-1} = z_{n-1} + A_2 \cdot z_{n-3}$;

(13) $z_n = z_n + A_{1 \cdot 1+2} \cdot z_{n-2}$;

(14) $z_{n-1} = z_{n-1} + A_1 \cdot z_{n-2}$; and

(15) write z_{n-1} and z_n, where z_k is an output of the silence detection, q_k is storage variable, A is an additive correction, k and n are counting variables, B_0, A_1, and A_2 are multiplicative factors, $A_{1 \cdot 2}$ represents $A_1 \cdot A_2$, and $A_{1 \cdot 1+2}$ represents $A_1 \cdot A_1 + A_2$.

18. The DSP of claim 17, wherein the DSP is adapted to perform:

(a) operations $(1)_s$, $(2)_s$, $(8)_{s-2}$, $(9)_{s-2}$, $(11)_{s-4}$, $(12)_{s-4}$ in parallel and in one clock cycle;

(b) operations $(3)_s$, $(4)_s$, $(5)_s$, $(13)_{s-4}$, $(14)_{s-4}$ in parallel and in one clock cycle; and

(c) operations $(6)_{s-2}$, $(7)_{s-2}$, $(10)_{s-2}$, $(15)_{s-4}$ in parallel and in one clock cycle, where the subscript indices s, s-2, and s-4 correspond to values of n for which the corresponding operation is performed.

19. The DSP of claim 13, wherein the DSP performs the tone detection for the data frame.

FIG. 1

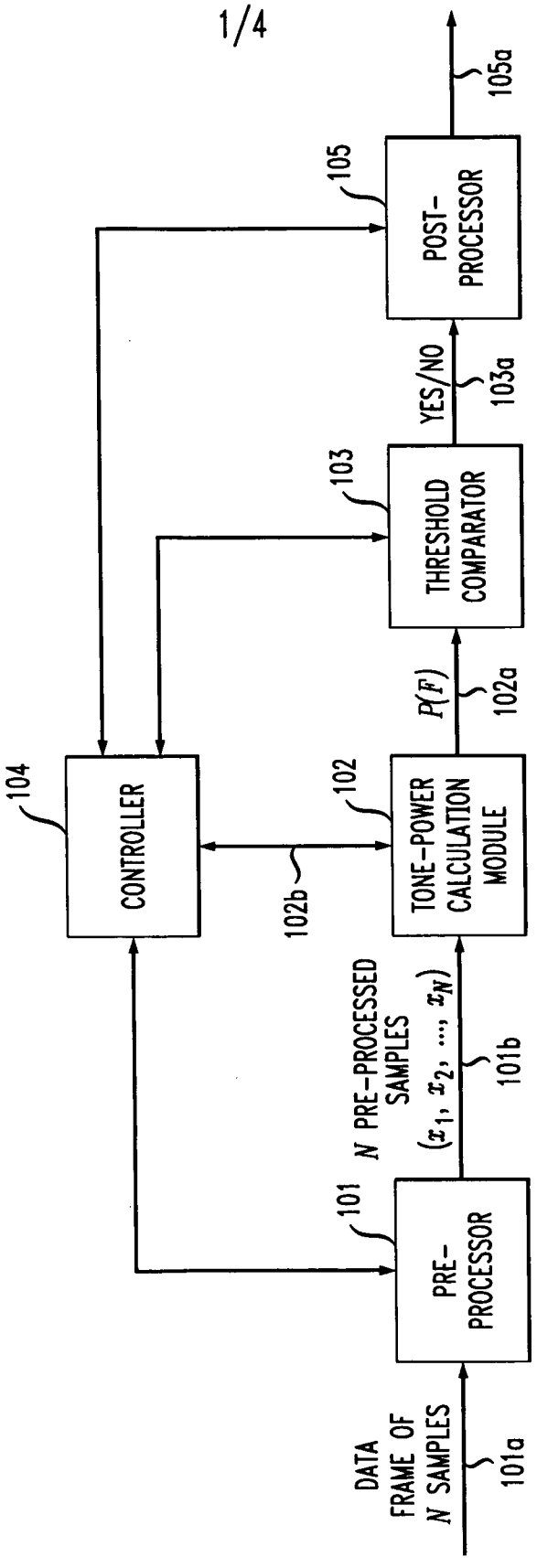

100
DATA FRAME OF N SAMPLES
101a
PRE-PROCESSOR
101
N PRE-PROCESSED SAMPLES
$(x_1, x_2, ..., x_N)$
101b
TONE-POWER CALCULATION MODULE
102
P(F)
102a
102b
CONTROLLER
104
THRESHOLD COMPARATOR
103
YES/NO
103a
POST-PROCESSOR
105
105a

FIG. 2

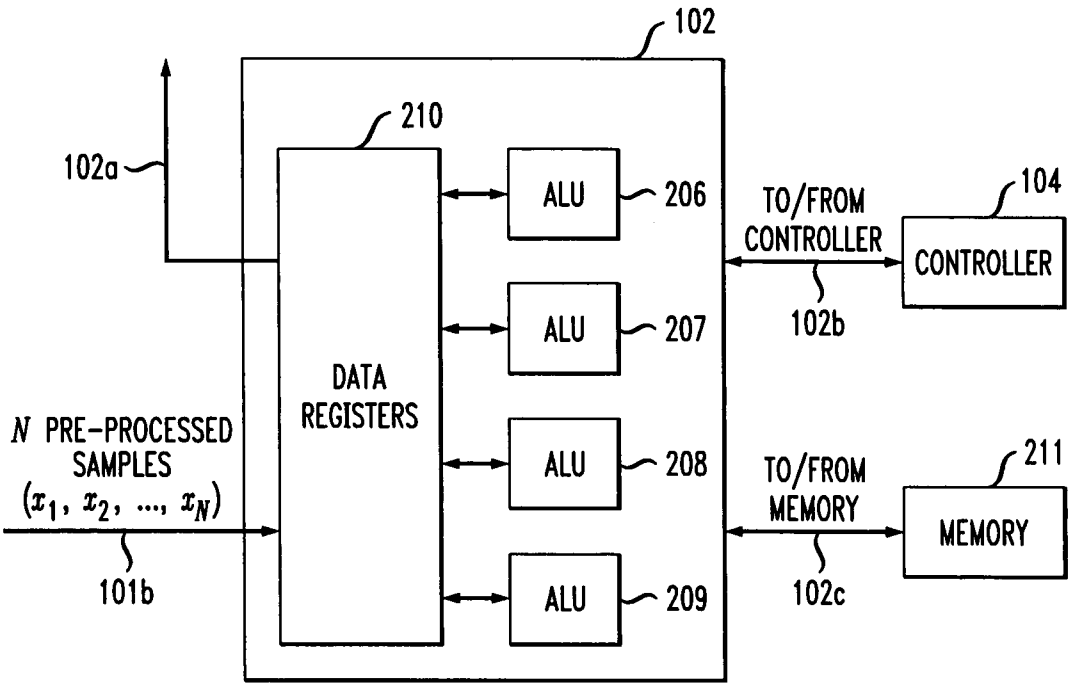

102
210
102a
ALU
206
ALU
207
ALU
208
ALU
209
DATA REGISTERS
TO/FROM CONTROLLER
104
CONTROLLER
102b
N PRE-PROCESSED SAMPLES
$(x_1, x_2, ..., x_N)$
101b
TO/FROM MEMORY
211
MEMORY
102c

FIG. 3

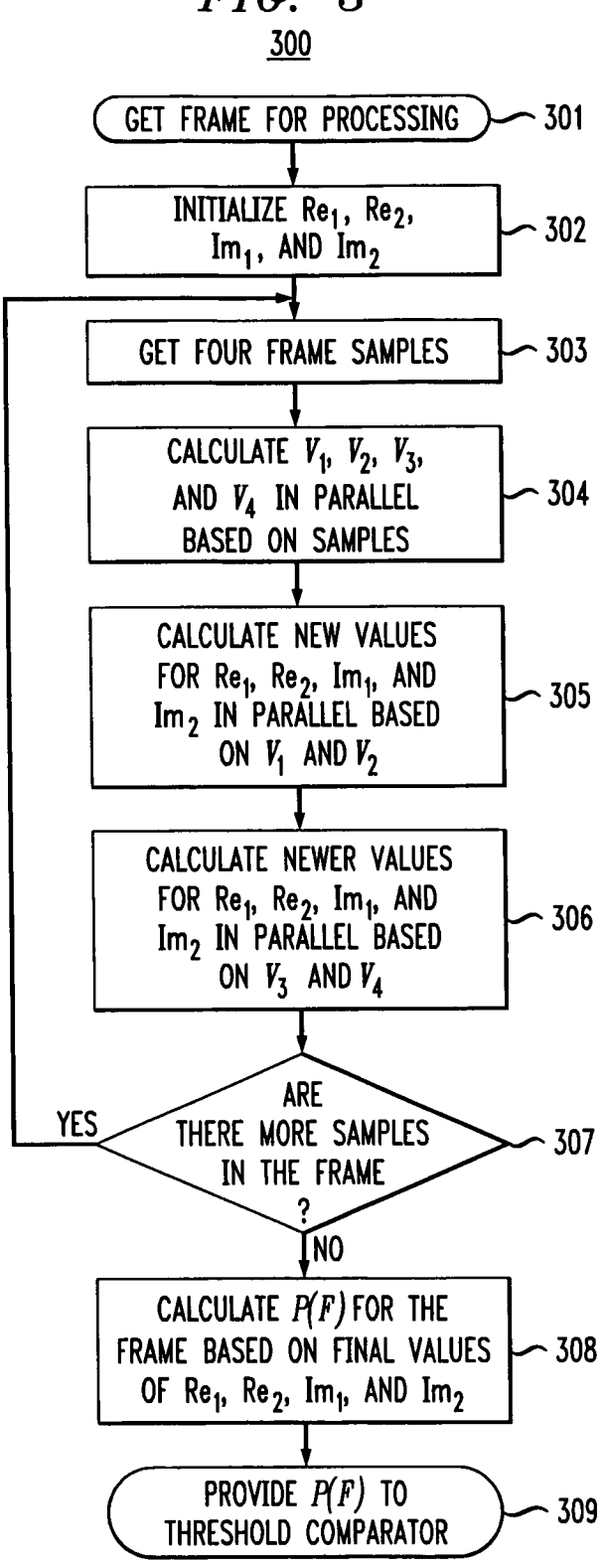
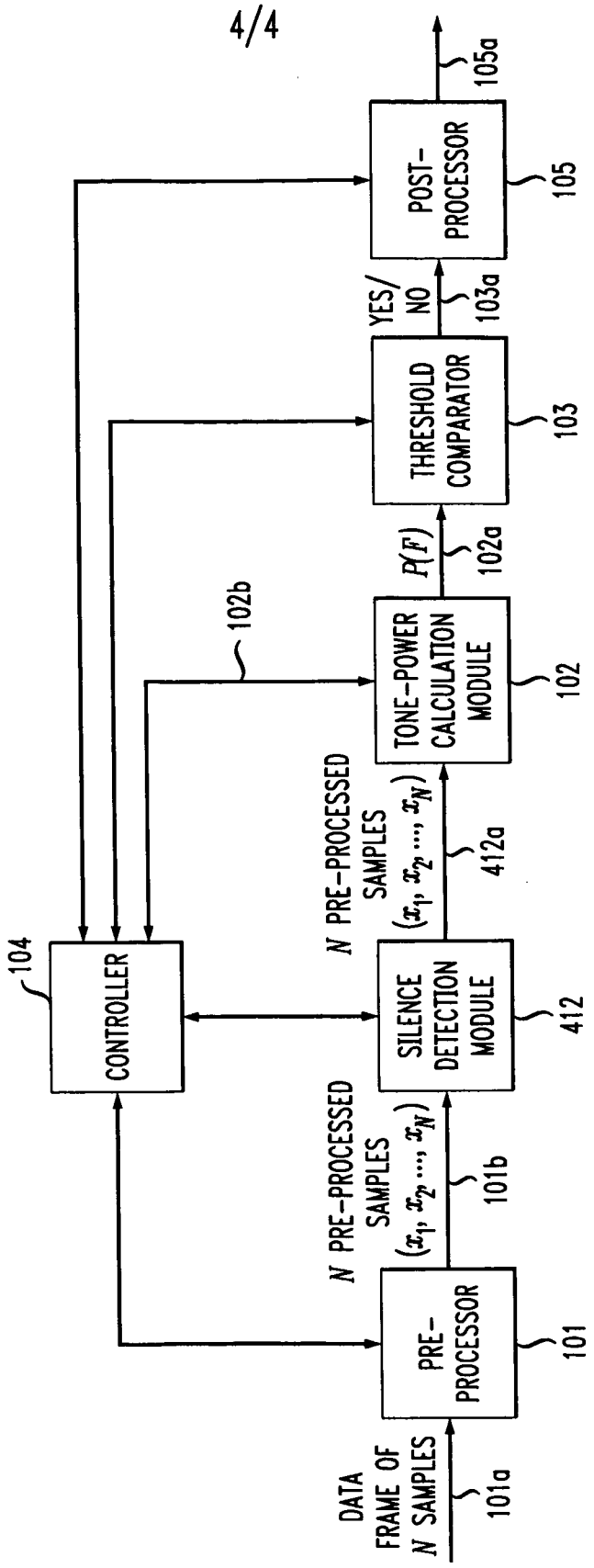

300
GET FRAME FOR PROCESSING
301
INITIALIZE Re_1, Re_2, Im_1, AND Im_2
302
GET FOUR FRAME SAMPLES
303
CALCULATE V_1, V_2, V_3, AND V_4 IN PARALLEL BASED ON SAMPLES
304
CALCULATE NEW VALUES FOR Re_1, Re_2, Im_1, AND Im_2 IN PARALLEL BASED ON V_1 AND V_2
305
CALCULATE NEWER VALUES FOR Re_1, Re_2, Im_1, AND Im_2 IN PARALLEL BASED ON V_3 AND V_4
306
ARE THERE MORE SAMPLES IN THE FRAME ?
307
YES
NO
CALCULATE $P(F)$ FOR THE FRAME BASED ON FINAL VALUES OF Re_1, Re_2, Im_1, AND Im_2
308
PROVIDE $P(F)$ TO THRESHOLD COMPARATOR
309

FIG. 4


400
DATA FRAME OF N SAMPLES
101a
PRE-PROCESSOR
101
N PRE-PROCESSED SAMPLES $(x_1, x_2, ..., x_N)$
101b
SILENCE DETECTION MODULE
412
N PRE-PROCESSED SAMPLES $(x_1, x_2, ..., x_N)$
412a
CONTROLLER
104
TONE-POWER CALCULATION MODULE
102
102b
P(F)
102a
THRESHOLD COMPARATOR
103
YES/NO
103a
POST-PROCESSOR
105
105a

INTERNATIONAL SEARCH REPORT

International application No
PCT/RU2009/000268

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04Q1/453
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G10L H04Q H04M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Valentin Emiya, Lúcio F.C. Pessoa, Delphine Vallot, and David Melles: "Generic Tone Detection Using Teager-Kaiser Energy Operators on the StarCore SC140 Core" Freescale Semiconductor Application Note AN2384 31 December 2004 (2004-12-31), XP002597663 Retrieved from the Internet: URL:http://www.freescale.com/files/dsp/doc/app_note/AN2384.pdf [retrieved on 2010-08-20]	1,2,13, 19
A	* abstract pages 2-4, 7 figures 2,3	3-12, 14-18
	----- -/--	

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 August 2010	03/09/2010

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Schmid, Andreas

INTERNATIONAL SEARCH REPORT

International application No
PCT/RU2009/000268

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2003/012358 A1 (KURTZ SCOTT DAVID [US] ET AL KURTZ SCOTT DAVID [US] ET AL) 16 January 2003 (2003-01-16)	1-12
A	paragraphs [0005], [0014] - [0019] figure 1 paragraphs [0022] - [0024]	13-19
Y	Anonymous: "MSC8156 Product Brief - Broadband Wireless Access DSP" Freescale Semiconductor Product Brief MSC8156PB 31 December 2008 (2008-12-31), XP002597664 Retrieved from the Internet: URL:http://www.ebv.com/fileadmin/products/Products/Freescale/MSC8156/MSC8156_PB.pdf [retrieved on 2010-08-20] pages 1,4 page 2, lines 8-13 page 8, lines 20-26 page 14, lines 15-17	1-19
Y	US 3 679 830 A (UFFELMAN MALCOLM R ET AL) 25 July 1972 (1972-07-25)	1,13-19
A	figures 2,3 column 2, lines 25-31 column 3, lines 1-12	2-12
A	US 2007/110042 A1 (LI HENRY [CA] ET AL) 17 May 2007 (2007-05-17) paragraphs [0135] - [0145] paragraphs [0243] - [0259]	1-19
A	WO 02/101724 A1 (GLOBESPAN VIRATA INC [US]) 19 December 2002 (2002-12-19) page 16, lines 14-18	1,13-19
A	US 6 959 316 B2 (PARVIAINEN JARI A [FI]) 25 October 2005 (2005-10-25) paragraphs [0019] - [0022] figures 1,6,9	1-19
A	US 4 782 523 A (GALAND CLAUDE [FR] ET AL) 1 November 1988 (1988-11-01) column 4, lines 9-42	1-12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/RU2009/000268

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☒ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-12

   Tone detector implemented on a DSP processor comprising a plurality of algorithmic logic units (ALUs).

   ---

2. claims: 13-19(completely); 1(partially)

   Silence detector implemented on a DSP processor comprising a plurality of algorithmic logic units (ALUs).

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/RU2009/000268

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003012358 A1	16-01-2003	NONE	
US 3679830 A	25-07-1972	NONE	
US 2007110042 A1	17-05-2007	NONE	
WO 02101724 A1	19-12-2002	CN 1539137 A	20-10-2004
		CN 1539138 A	20-10-2004
		JP 2004536334 T	02-12-2004
		JP 2004534263 T	11-11-2004
		WO 02101727 A1	19-12-2002
		WO 02101722 A1	19-12-2002
		WO 02101723 A1	19-12-2002
US 6959316 B2	25-10-2005	EP 1211595 A1	05-06-2002
US 4782523 A	01-11-1988	CA 1284679 C	04-06-1991
		DE 3678717 D1	16-05-1991
		EP 0243561 A1	04-11-1987
		JP 1896687 C	23-01-1995
		JP 6024398 B	30-03-1994
		JP 62261255 A	13-11-1987