

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
29 April 2010 (29.04.2010)

PCT

(10) International Publication Number
WO 2010/048498 A1

- (51) **International Patent Classification:**
G11B 20/10 (2006.01) *H04H 20/31* (2008.01)
H04H 60/58 (2008.01) *H04H 60/37* (2008.01)
- (21) **International Application Number:**
PCT/US2009/061827
- (22) **International Filing Date:**
23 October 2009 (23.10.2009)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/108,380 24 October 2008 (24.10.2008) US
- (71) **Applicant:** THE NIELSEN COMPANY (US), LLC [US/US]; A Delaware Limited Liability Company, 150 N. Martingale Road, Schaumburg, IL 60173 (US).
- (72) **Inventors:** SRINIVASAN, Venugopal; 2845 Jarvis Circle, Palm Harbor, FL 34683 (US). TOPCHY, Alexander Pavlovich; 7106 Waxwing Drive, New Port Richey, FL 34653 (US).
- (74) **Agent:** ZIMMERMAN, Michael, W.; HANLEY FLIGHT & ZIMMERMAN, LLC., 150 S. Wacker Drive, Suite 2100, Chicago, IL 60606 (US).
- (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, CL, 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, PE, 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, 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, SM,

[Continued on next page]

(54) **Title:** METHODS AND APPARATUS TO EXTRACT DATA ENCODED IN MEDIA CONTENT

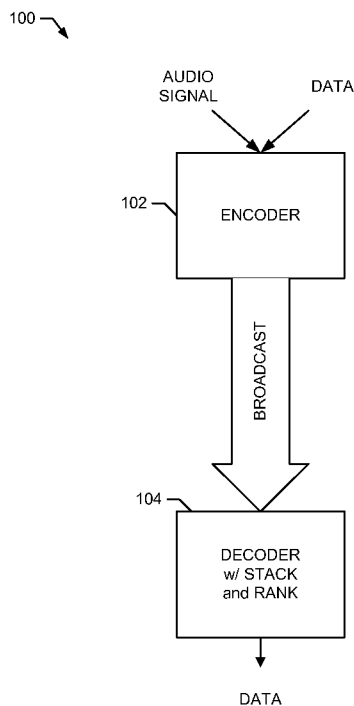


FIG. 1

(57) **Abstract:** Methods and apparatus to extract data encoded in media content are disclosed. An example method includes receiving a media content signal, sampling the media content signal to generate digital samples, determining a frequency domain representation of the digital samples, determining a first rank of a first frequency in the frequency domain representation, determining a second rank of a second frequency in the frequency domain representation, combining the first rank and the second rank with a set of ranks to create a combined set of ranks, comparing the combined set of ranks to a set of reference sequences, determining a data represented by the combined set of ranks based on the comparison, and storing the data in a tangible memory.

WO 2010/048498 A1

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

Published:

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

METHODS AND APPARATUS TO EXTRACT DATA ENCODED IN MEDIA CONTENT

RELATED APPLICATIONS

[0001] This patent claims priority to US Provisional Patent Application Serial No. 61/108,380, "STACKING METHOD FOR ENHANCED WATERMARK DETECTION," filed on October 24, 2008, the disclosure of which is incorporated by reference in its entirety.

FIELD OF THE DISCLOSURE

[0002] The present disclosure pertains to monitoring media content and, more particularly, to methods and apparatus to extract data encoded in media content.

BACKGROUND

[0003] Identifying media information and, more specifically, audio streams (e.g., audio information) is useful for assessing audience exposure to television, radio, or any other media. For example, in television audience metering applications, a code may be inserted into the audio or video of media, wherein the code is later detected at monitoring sites when the media is presented (e.g., played at monitored households). The information payload of the code/watermark embedded into original signal can consist of unique source identification, time of broadcast information, transactional information or additional content metadata.

[0004] Monitoring sites typically include locations such as, for example, households where the media consumption of audience members or audience member exposure to the media is monitored. For example, at a monitoring site, codes from the audio and/or video are captured and may be associated with audio or video streams of media associated with a selected channel, radio station, media source, etc. The collected codes may then be sent to a central data collection facility for analysis. However, the collection of data pertinent to media exposure or consumption need not be limited to in-home exposure or consumption.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 is a block diagram of an example system for encoding data in a media content signal to transmit the data to a location where the media content signal is decoded to extract the data.

[0006] FIG. 2 is a graph of an example frequency spectrum and code indexing.

[0007] FIG. 3 illustrates an example sequence that may be encoded in an audio signal by the example encoder of FIG. 1.

[0008] FIG. 4 illustrates an example message thread.

[0009] FIG. 5 is a block diagram of an example apparatus to implement the decoder of FIG. 1 that includes stack and rank functionality.

[0010] FIG. 6 is a flowchart of an example process to decode a message in audio.

[0011] FIG. 7 is a flowchart of an example process to decode a message in audio using stacking.

[0012] FIG. 8 is a schematic illustration of an example processor platform that may be used and/or programmed to perform any or all of the example machine accessible instructions of FIGS. 6-7 to implement any or all of the example systems, example apparatus and/or example methods described herein.

DETAILED DESCRIPTION

[0013] FIG. 1 is a block diagram of an example system 100 for encoding data in a media content signal to transmit the data to a location where the media content signal is decoded to extract the data. The example system 100 includes an encoder 102 and a decoder 104 with stack and rank functionality. According to the illustrated example, the encoder 102 encodes a received audio signal with a received data by amplifying or attenuating frequencies of interest as described in detail herein. The encoded audio signal is transported to another location where it is received by the decoder 104. The decoder 104 includes a stack functionality to stack consecutively received portions of the audio signal. In addition, the decoder 104 includes rank functionality to assign ranks to frequencies that may have been amplified or attenuated by the encoder 102. For example, where frequencies are grouped in neighborhoods of five frequencies, a rank of 0 to 4 may be assigned to each frequency. The decoder 104 then extracts the data from the stacked audio signal as described in detail herein. Stacking the encoded audio signal will, for example, improve the detection reliability of the decoder 104 when stacked portions include redundant or semi-redundant encoded data. While not shown in the illustrated example, the audio signal may also be output by the decoder 104 to be presented on a media presentation device (e.g., a radio, a television, etc.). Alternatively, the encoded audio signal may be transmitted to a media presentation device in parallel with the example decoder 104.

[0014] According to the example of FIG. 1, the encoder 102 receives as input an audio signal and data. The encoder 102 further divides the audio signal into frames, which are blocks of digital audio samples. As described in detail below, the encoder 102 encodes (embeds) the data into the framed audio signal and the encoded frame of audio is tested by

the encoder 102 to determine if the modifications to the framed audio signal are significant enough to cause the encoding to be audibly perceptible by a human when the framed audio signal is presented to a viewer (e.g., using psychoacoustic masking). If the modifications to the framed audio signal are too significant and would result in an audible change in the audio, the framed audio is transmitted (e.g., broadcast, delivered to a broadcaster, etc.) without being encoded. Conversely, if the encoded audio frame has audio characteristics that are imperceptibly different from the un-encoded audio frame, the encoded audio frame is transmitted.

[0015] The encoder 102 inserts a unique or semi-unique 15-bit pseudorandom number (PN) synchronization sequence at the start of each message packet. To signal to the decoder 104 that a synchronization sequence is to be transmitted, the first code block of the synchronization sequence uses a triple tone. The triple tone is an amplification of three frequencies causing those frequencies to be maxima in their spectral neighborhoods. Thus, by looking for the triple tone, the decoder 104 can detect that a synchronization sequence is about to be sent without the need for decoding the entire synchronization sequence. An example implementation of a triple tone is described in United States Patent 6,272,176 ('176 patent'), which is hereby incorporated by reference in its entirety. The example synchronization sequence is one approach for enabling the decoder 104 to detect the start of a new message packet. However, any other indication, signal, flag, or approach may be used.

[0016] The example encoder 102 transmits as many as ten 15-bit PN sequences of message data following the synchronization. Thus, each message in the illustrated example comprises 11 groups: one 15-bit synchronization sequence followed by ten 15-bit message data sequences. However, any number of message data sequences may be transmitted between synchronization sequences. The example message data is transmitted in 15-bit PN sequences having ten error correction bits and five message data bits. In other words, message data is divided into groups of five bits each (e.g., ten 5-bit groups for a 50-bit message). Alternatively, any combination of message data bits and error correction bits may be included in a message data sequence. Each bit of the 15-bit PN sequence is encoded into a 512-sample block of audio. In the example system 100, one bit is transmitted at a time. Each 5 bits of payload data that is encoded as a 15-bit sequence uses 15 blocks of 512 samples (i.e., 7680 samples total). The example encoder 102 includes a 16th block called the null block after the 15 blocks representing the 15-bit sequence. Thus, each message in the illustrated example uses 176 audio blocks: 16 blocks per sequence and 11 sequences per

message. In the illustrated example, each message is followed by 11 unencoded blocks to adjust the total message duration to be approximately two seconds in the example encoding. While example encoding and block sizes are described, any desired encoding and block sizes may be used.

[0017] To insert a data bit (e.g., one bit of a 15-bit sequence) into an audio frame, the example encoder 102 makes a first selected frequency of the audio frame a local maximum and makes a second selected frequency of the audio frame a local minimum. For example, as shown in FIG. 2, the encoder 102 uses two audio frequency bands or neighborhoods 202 and 204, each including five frequencies or residents. One of the neighborhoods 202 and 204 is encoded to include a resident that is a local maximum and the other neighborhood 202 and 204 is encoded to include a resident that is a local minimum. The residents that are selected to be local maximum and local minimum are based on the coding block on which the example encoder 102 is operating and the value of the data bit to be transmitted. For example, to encode a logical "1" in the fifth encoding block, a resident having index number 50 in the neighborhood 202 is made a local maximum and a resident having index number 60 in the neighborhood 204 is made a local minimum. Conversely, to encode a logical "0" for the same encoding block, the resident having index number 50 in the neighborhood 202 would be made a local minimum and the resident having index number 60 in the neighborhood 204 would be made a local maximum. In other words, the frequencies that are selected do not represent the bit to be sent, the amplitudes at the selected frequencies represent the value of the bit because the same frequencies may be used whether the bit is a logical "1" or a logical "0". After encoding, the audio signal may be broadcast to a consumer location, may be transmitted to a broadcaster for broadcasting, may be stored to a storage media, etc.

[0018] The example system 100 may be configured to perform stacking and ranking in a system that is implemented with the Nielsen Audio Encoding System (NAES) described in the '176 patent. While this disclosure makes reference to encoding and decoding techniques of the NAES system described in the '176 patent by way of example, the methods and apparatus described herein are not limited to operation in conjunction with the techniques of the '176 patent. To the contrary, the example methods and apparatus may be implemented in conjunction with any type of encoding or decoding system. For example, the data rates, data grouping, message lengths, parameter lengths, parameter order in messages, number of parameters, etc. may vary based on the implemented encoding system.

[0019] FIG. 3 illustrates an example sequence 300 that may be encoded in an audio signal by the example encoder 102 of FIG. 1. The example sequence 300 includes 15 bits that are encoded in 15 blocks of audio data (e.g., 512 sample blocks). The message bits 302 convey five bits of message data. The message bits 302 are the payload data to be conveyed by the encoding. The error correction bits 304 convey ten bits of error correction data that may be used by the decoder 104 to verify and correct a received message. Each bit of the sequence 300 is encoded in a block of audio data. As described in conjunction with FIG. 1, for each block of audio data, a first selected frequency is made a local maximum and a second selected frequency is made a local minimum.

[0020] FIG. 4 illustrates an example message thread 400. The message thread 400 of the illustrated example includes a synch sequence 402, a first sequence 406, a second sequence 410, a third sequence 414, and no mark blocks 404, 408, and 412. The example synch sequence 402 is a 15 bit sequence that indicates the start of a new message thread. The first sequence 406, the second sequence 410, and the third sequence 414 of the illustrated example are 15 bit sequences that each convey five message payload bits and ten error correction bits as described in conjunction with FIG. 3. The no mark blocks 404, 408, and 412 are single blocks that include no encoding (e.g., 512 samples of audio data in which no frequencies are amplified or attenuated by the encoder 102). While the example message thread 400 is formatted as described, any other formatting may be used. For example, more or fewer sequences may be included in a message thread 400, sequences 406, 410, and 414 may contain more or fewer data bits and/or error correction bits, the no mark blocks 404, 408, and 412 may include multiple blocks, more or fewer no mark blocks 404, 408, and 412 may be included, etc.

[0021] FIG. 5 is a block diagram of an example apparatus to implement the decoder 104 of FIG. 1 that includes stack and rank functionality. The example decoder 104 includes a sampler 502, a time domain to frequency converter 504, a ranker 506, a rank buffer 508, a stacker 510, a stacker control 512, a comparator 514, and a reference sequence datastore 516. The example decoder 104 receives an input audio (e.g., an audio portion of a television program) and processes the audio to extract and output data encoded in the audio.

[0022] The sampler 502 of the illustrated examples samples the incoming audio. The sampler 502 may be implemented using an analog to digital converter (A/D) or any other suitable technology, to which encoded audio is provided in analog format. The sampler 502 samples the encoded audio at, for example, a sampling frequency of 48 KHz. Of course,

other sampling frequencies may be selected in order to increase resolution or reduce the computational load at the time of decoding. Alternatively, the sampler 502 may be eliminated if audio is provided in digitized format.

[0023] The time domain to frequency domain converter 504 of the illustrated example may be implemented using a discrete Fourier transformation (DFT), or any other suitable technique to convert time-based information into frequency-based information. In one example, the time domain to frequency domain converter 504 may be implemented using a sliding DFT in which a spectrum of the code frequencies of interest (e.g., frequencies indexed 1 to N in FIG. 5) is calculated each time four new samples are provided to the example time domain to frequency domain converter 504. In other words, four new samples are shifted into the analysis windows, four old samples are shifted out of the analysis window, and the DFT of the analysis window is computed. Because the boundaries of blocks are not known when decoding, a sliding DFT may operate by sliding 4 samples at a time to give 128 distinct message threads to analyze per 512 samples of audio that are received. Thus, at the end of 128 slides (of four samples each), all 512 samples (i.e., one block worth of samples) will have been processed and analyzed. The resolution of the spectrum produced by the time domain to frequency domain converter 504 increases as the number of samples (e.g., 512 or more) used to generate the spectrum increases. Thus, the number of samples processed by the time domain to frequency domain converter 504 should match the resolution used to select the residents shown in FIG. 2. The finer the frequency spacing between the residents, the more samples that will be used to generate the spectrum for detection of the residents.

[0024] The spectrum produced by the time domain to frequency domain converter 504 passes to the ranker 506. The ranker 506 of the illustrated example ranks the amplitude of each frequency of interest (e.g., RANK 1 to RANK N for the 1 to N frequency indices of interest in FIG. 5) in neighborhoods in the received spectrum relative to the amplitude of the other frequencies in the neighborhood. For example, when there are five frequencies in each neighborhood, the amplitude of each frequency may be ranked on a scale of 0 to 4, where 0 is the lowest amplitude and 4 is the greatest amplitude. While the forgoing example describes ranking each spectrum frequency, any subset of frequencies may alternatively be ranked such as, for example, only frequencies of interest that may have been amplified or attenuated to embed information in the audio data. The ranker 506 outputs a set of rank values to the rank buffer 508.

[0025] The rank buffer 508 stores the set of rank values in a circular buffer such that once the buffer has been filled, each new set of ranks will replace the oldest set of ranks in the buffer. The rank buffer 508 of the illustrated example stores the 128 sets of ranks (e.g., 128 sets of ranks 1 to N) corresponding to each slide of the time domain to frequency domain converter 504. In addition, the rank buffer 508 may store multiple messages worth of ranks. For example, as described in detail below, the rank buffer 508 may store five messages worth of ranks so that the blocks of messages may be averaged. While the rank buffer 508 is described as a circular buffer and type of data structure and storage may be used. For example, the rank buffer 508 may comprise one or more registers, one or more files, one or more databases, one or more buffers of any type, etc.

[0026] An example set of ranks may be:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RANKS	2,4	1,4	4,1	4,0	0,4	3,1	3,0	4,1	4,1	4,2	2,3	4,3	4,1	0,4	4,0	0,0

[0027] The stacker 510 takes advantage of message-to-message redundancy to improve the detection of data encoded in audio signals. In particular, when enabled by the stacker control 512, the stacker 510 retrieves the ranks of consecutive messages from the rank buffer 508 and adds the ranks of corresponding blocks of the consecutive messages. The stacker 510 then divides the sums by the number of messages added together. Accordingly, the stacker 510 determines an average of the ranks for consecutive blocks. When messages include redundancy, the ranks will average in order to eliminate errors introduced by noise or host audio. For example, an encoded message may be 50 bits including a broadcaster identifier (e.g., a 16-bit station identifier) followed by a timestamp (e.g., a 32-bit timestamp that denotes time elapsed in seconds since, for example, January 1, 1995), followed by a level specification that allows multiple levels of messages to be included (e.g., a 2-bit level specification). In the example 50 bit message, all but the least significant bits of the message will be repeated for several messages in a row. In the example encoding where messages are divided into ten groups and include one synch group (e.g., 11 total groups), it is expected that the first ten groups will repeat from message to message and the last group (e.g., that contains the three least significant bits of the timestamp and two level specification bits) will change from message to message. Because the three least significant bits can represent eight seconds and messages in the example encoding are encoded into approximately two seconds of audio

each, the fourth least significant bit of the message will change after four messages. Accordingly, the synchronization group and the first nine data groups are expected to repeat for four messages (approximately eight seconds).

[0028] The stacking process may be performed according to the following formulas:

$$r_{1km_n} = \frac{\sum_{p=n-S}^{p=n-1} r_{1km_p}}{S} \quad \text{and} \quad r_{2km_n} = \frac{\sum_{p=n-S}^{p=n-1} r_{2km_p}}{S}$$

where p is a message index (e.g., $0 \leq p \leq 5$) when five consecutive messages are to be averaged), k is a block index (e.g., $0 \leq k \leq 16$ when there are 16 blocks per sequence), S is the number of consecutive messages to be averaged (e.g., 5 when five consecutive messages are to be averaged), r_{1km_n} is the average rank of the first frequency of interest in the k^{th} block of a message m_n , and r_{2km_n} is the average rank of the second frequency of interest in the k^{th} block of message m_n . For example, a message may be a station identifier and a timestamp that are encoded every 2 seconds. While the least significant bits of the time stamp (e.g., seconds) may change from message to message, the other bits (e.g., more significant bits of a timestamp) will not change between every message. Accordingly, when the ranks of the current message are added to the ranks of the previous four messages, the average ranking can improve detection by reducing the effect of any noise that may have been present for less than all of the messages. When the stacker 510 is enabled, the stacker 510 outputs the stacked set of ranks (e.g., RANK_S 1 to stacked RANK_S in FIG. 5) to the comparator 514. When the stacker 512 is not enabled, the stacker 510 outputs the set of ranks (e.g., RANK_S 1 to RANK_S N) retrieved from the rank buffer 508 to the comparator 514.

[0029] In an example, the following ranks may be determined for corresponding packets that are repetitions of the same message:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RANK MSG0	2,4	1,4	4,1	4,0	1,4	3,1	3,0	4,1	4,1	4,2	2,3	4,3	4,1	0,3	4,0	0,0
RANK MSG1	0,4	1,4	4,1	4,1	0,2	4,1	3,0	3,1	4,1	4,2	2,3	4,3	4,2	0,4	4,1	0,0
RANK MSG2	0,4	1,4	3,1	4,2	0,4	3,1	3,0	4,1	4,2	4,2	2,3	4,2	4,1	0,4	4,2	0,0
RANK MSG3	1,4	1,4	4,2	4,0	2,4	3,2	3,0	4,1	4,1	4,1	2,4	4,3	4,1	0,4	4,0	0,0
RANK MSG4	4,2	1,4	4,1	4,2	0,3	3,1	3,0	4,1	4,1	4,2	2,3	4,3	4,1	0,4	4,0	0,0

The sum of the ranks is:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RANK	7,	5,	19,	20,	3,	16,	15,	19,	20,	20,	10,	20,	20,	0,	20,	0,
SUM	18	20	6	5	17	6	0	5	6	9	16	14	6	19	3	0

The average of the ranks is:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RANK	1.4,	1,	3.8,	4,	0.6,	3.2,	3,	3.8,	4,	4,	2,	4,	4,	0,	4,	0,
AVG	3.6	4	1.2	1	3.4	1.2	0	1	1.2	1.8	3.2	2.8	1.2	3.8	0.6	0

As shown in the example, even when Block 0 of Message 4 has been ranked in a manner that suggests the opposite data bit as the previous four messages (i.e., 4,2 would suggest a bit value of 1, while the other values suggest a bit value of 0), averaging of the ranking results in an average that suggests a bit value of 0. Accordingly, even when error due to noise is

introduced, averaging of the ranks can result in ranking that more closely matches the encoded data.

[0030] The stacker control 512 controls when the stacker 510 is enabled or disabled. For example, when the stacker 510 is disabled, messages may be processed one at time without any averaging of the ranks. When the stacker 510 is enabled by the stacker control 512, stacking of messages is performed as described herein or using any other process. The stacker control 512 may enable stacking based on any criteria. For example, the stacker control 512 may enable provide selective stacking by automatically enabling stacking when noise is detected, when a poor quality audio connection is present (e.g., when a microphone is used rather than a physical connection), when the decoder 104 is at a distance from an audio source (e.g., a mobile device across the room from an audio source), etc. Additionally or alternatively, the stacker control 512 may be manually controlled to enable stacking when requested by a user and/or may be remotely controlled by a message from a central location, the encoder 102, etc.

[0031] The comparator 514 of the illustrated example receives the set of ranks or stacked ranks (“set of ranks”) for a sequence from the stacker 510 and determines if a synch sequence has been recognized. If a synch sequence has not been detected, the comparator 514 compares the received set of ranks to a reference synch sequence and sets a synch detected flag if the set of ranks is determined to correspond to a synch sequence. If a synch sequence has previously been detected, the comparator 514 compares the set of ranks to a reference set of sequences stored in the reference sequence data store 516. The reference set of sequence comprise a listing of possible ranks and associated high or low indications for the frequencies of interest for each block. For example, when each sequence includes 5 data bits, 10 error correction bits, and one blank block, there would be 2⁵ possible Bose and Ray-Chaudhuri (BCH) codewords of 15 bits, each bit having an indication of whether each of two frequencies of interest were attenuated or amplified (i.e., 30 indications). To determine the sequence corresponding to the set of ranks, the set of ranks is compared to each of the reference sequences. The reference sequence with the smallest different from the set of ranks is identified as the received sequence.

[0032] For example, when the received set of ranks provided by the stacker 510 is:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

RANKS	2,4	1,4	4,1	4,0	0,4	3,1	3,0	4,1	4,1	4,2	2,3	4,3	4,1	0,4	4,0	0,0
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

The closest reference sequence may be the following set for data bits 0,0,1,1,0:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RANKS	0,4	0,4	4,0	4,0	0,4	4,0	4,0	4,0	4,0	4,0	4,0	4,0	4,0	0,4	4,0	0,0
Bit Val.	0	0	1	1	0	1	1	1	1	1	1	1	1	0	1	-

When compared by determining the distance or absolute value of the difference of the reference ranks and the received set of ranks, the difference is:

	DATA BITS					ERROR CORRECTION BITS										
BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIFF.	2,0	1,0	0,1	0,0	0,0	1,1	1,0	0,1	0,1	0,2	2,3	0,3	0,1	0,0	0,0	0,0

The numerical difference (e.g., hamming distance) is the sum of the difference row, which equals 20. This difference would be compared to the difference for all other possible sequences. If this difference was less than all other distances, then the reference sequence is determined to be the closest match.

[0033] In addition to determining the closest sequence from the reference set of sequences, the comparator 514 may also determine if the difference for the closest sequence exceeds a threshold. For example, the comparator 514 may discard the result if the difference is greater than a threshold, meaning that the closest reference sequence was significantly different than the received set of ranks. In other words, the comparator 514 may ensure that the received set of ranks are close enough to the determined reference sequence before outputting the sequence.

[0034] The example comparator 514 is further configured to reconstruct the least significant bits (LSB) of a detected sequence. The LSB may need to be reconstructed when the stacker is enabled and several messages are averaged. Such averaging will cause the LSB (or other rapidly changing data) that varies among the averaged messages to be recreated. Any method for reconstructed the data may be used. For example, if the data to be reconstructed is the LSB of a timestamp, one message may be detected without the use of stacking and a timer may be used to determine the difference in time between the known LSB

and the current message so that the LSB of the timestamp can be recreated and the determined message modified to include the correct LSB.

[0035] The reference sequence 516 of the illustrated example may be implemented by any type of data storage. For example, the reference sequence datastore 516 may be a file, a database, a table, a list, an array, or any other type of datastore. While the example reference sequence 516 stores the 32 possible BCH sequences, any number of sequences may be stored. For example, a partial set of sequences may be stored.

[0036] Flowcharts representative of example processes that may be executed to implement some or all of the elements of the system 100 and the decoder 104 are shown in FIGS. 6-7.

[0037] In these examples, the process represented by each flowchart may be implemented by one or more programs comprising machine readable instructions for execution by: (a) a processor, such as the microprocessor 812 shown in the example computer 800 discussed below in connection with FIG. 8, (b) a controller, and/or (c) any other suitable device. The one or more programs may be embodied in software stored on a tangible medium such as, for example, a flash memory, a CD-ROM, a floppy disk, a hard drive, a DVD, or a memory associated with the processor 812, but the entire program or programs and/or portions thereof could alternatively be executed by a device other than the microprocessor 812 and/or embodied in firmware or dedicated hardware (e.g., implemented by an application specific integrated circuit (ASIC), a programmable logic device (PLD), a field programmable logic device (FPLD), discrete logic, etc.). For example, any one, some or all of the example mobile communications system components could be implemented by any combination of software, hardware, and/or firmware. Also, some or all of the processes represented by the flowcharts of FIGS. 6-7 may be implemented manually.

[0038] Further, although the example processes are described with reference to the flowcharts illustrated in FIGS. 6-7, many other techniques for implementing the example methods and apparatus described herein may alternatively be used. For example, with reference to the flowcharts illustrated in FIGS. 6-7, the order of execution of the blocks may be changed, and/or some of the blocks described may be changed, eliminated, combined, and/or subdivided into multiple blocks. While the processes of FIGS. 6-7 are described in conjunction with the decoder 104, any apparatus or system may implement the processes of FIGS. 6-7.

[0039] FIG. 6 is a flowchart of an example process to decode a message in audio. The process of FIG. 6 begins when the sampler 502 updates a current audio block by sampling 4 samples and discarding 4 samples from an analysis window (block 602). The example time domain to frequency converter 504 performs a sliding FFT to convert the sampled audio from the time domain to the frequency domain (block 604). The ranker 506 ranks the code frequencies in the converted audio (block 606). For example, as described above, frequencies of interest may be ranked on a scale of 0 to 4 when there are five frequencies in each neighborhood. The determined ranks are stored in the rank buffer 508 (block 608). When the rank buffer 508 is a circular buffer, the addition of the determined ranks will eliminate a previously stored rank. In addition, when the rank buffer 508 is a circular buffer, an index indicating the point at which the next set of ranks should be inserted to the rank buffer 508 is incremented (block 610).

[0040] The comparator 512 then generates a rank distribution array across the number of blocks in a sequence (e.g., 15 blocks) (block 612). Next, the comparator 514 determines if a synch sequence has previously been detected (block 614). The synch sequence indicates the start of a message. Therefore, when the synch has previously been detected, a message thread has started. When a synch sequence has not previously been detected, control proceeds to block 624, which is described below.

[0041] When a synch sequence has previously been detected (block 614), the comparator 514 generates match scores against all potential sequences (e.g., 32 possible BCH sequences) (block 616). For example, the comparator 514 may determine a distance between the rank distribution and each of the potential sequences. The comparator 514 then selects the potential sequence with the greatest score (e.g., smallest distance) (block 618). The comparator 514 determines if the selected score exceeds a threshold (block 620). For example, if the score is a distance, the comparator 514 determines if the distance is less than a threshold distance. When the score does not exceed the threshold, control proceeds to block 602 to continue processing.

[0042] When the score exceeds the threshold (block 620), the comparator 514 assigns the value to the sequence (block 622). Control then proceeds to block 602 to continue processing.

[0043] Returning to block 624, when a match has not been previously detected (block 614), the comparator 514 generates a match score for the synch sequence (block 624). For example, as described above the comparator 514 may determine a distance between the rank

distribution and the reference synch sequence. The comparator 514 determines if the score exceeds a threshold (block 626). When the score does not exceed the threshold, control proceeds to block 602 to continue processing. When the score exceeds the threshold, a flag is set indicating that a synch has been detected (block 628). Control then proceeds to block 602 to continue processing. While a flag is described above, any indication that a synch has been detected may be used. For example, a variable may be stored, the synch sequence may be stored in a table, etc. In addition, while the example process includes a separate branch for detecting a synch sequence, synch sequences may be detected in the same branch as other sequences and processing may later be performed to identify a synch sequence that indicates that start of a message thread. Further, while the process of FIG. 6 is illustrated as a continuous loop, any flow may be utilized.

[0044] FIG. 7 is a flowchart of an example process to decode a message in audio. The process of FIG. 7 utilizes stacking to improve decoding accuracy. The process of FIG. 7 begins when the sampler 502 updates a current audio block by sampling 4 samples and discarding 4 samples from an analysis window (block 702). The example time domain to frequency converter 504 performs a sliding FFT to convert the sampled audio from the time domain to the frequency domain (block 704). The ranker 506 ranks the code frequencies in the converted audio (block 706). For example, as described above, frequencies of interest may be ranked on a scale of 0 to 4 when there are five frequencies in each neighborhood. The stacker 510 then adds the determined ranks to the ranks of corresponding blocks of previous messages and divided by the number of messages to determine an average rank (block 707). For example, the determined ranks may be added to the corresponding ranks of the previous 4 messages.

[0045] The average ranks are stored in the rank buffer 508 (block 708). When the rank buffer 508 is a circular buffer, the addition of the average ranks will eliminate a previously stored rank. In addition, when the rank buffer 508 is a circular buffer, an index indicating the point at which the next set of ranks should be inserted to the rank buffer 508 is incremented (block 710). Alternatively, the ranks may be stored in the rank buffer 508 after block 706 and may retrieved from the rank buffer 508 as part of block 707.

[0046] The comparator 514 then generates a rank distribution array across the number of blocks in a sequence (e.g., 15 blocks) (block 712). Next, the comparator 514 determines if a synch sequence has previously been detected (block 714). The synch sequence indicates the start of a message. Therefore, when the synch has previously been detected, a message

thread has started. When a synch sequence has not previously been detected, control proceeds to block 724, which is described below.

[0047] When a synch sequence has previously been detected (block 714), the comparator 514 generates match scores against all potential sequences (e.g., 32 possible BCH sequences) (block 716). For example, the comparator 514 may determine a distance between the rank distribution and each of the potential sequences. The comparator 514 then selects the potential sequence with the greatest score (e.g., smallest distance) (block 718). The comparator 514 determines if the selected score exceeds a threshold (block 720). For example, if the score is a distance, the comparator 514 determines if the distance is less than a threshold distance. When the score does not exceed the threshold, control proceeds to block 702 to continue processing.

[0048] When the score exceeds the threshold (block 720), the comparator 514 assigns the value to the sequence (block 722). The comparator 512 then reconstructs any data that may have been corrupted by the stacking process. For example, that comparator 512 may determine a corrupted portion of a timestamp (e.g., a second indication) by decoding one message and tracking the amount of time that passes between the decoded message and a currently detected message. Control then proceeds to block 702 to continue processing.

[0049] Returning to block 724, when a match has not been previously detected (block 714), the comparator 514 generates a match score for the synch sequence (block 724). For example, as described above the comparator 514 may determine a distance between the rank distribution and the reference synch sequence. The comparator 514 determines if the score exceeds a threshold (block 726). When the score does not exceed the threshold, control proceeds to block 702 to continue processing. When the score exceeds the threshold, a flag is set indicating that a synch has been detected (block 728). Control then proceeds to block 702 to continue processing. While a flag is described above, any indication that a synch has been detected may be used. For example, a variable may be stored, the synch sequence may be stored in a table, etc. In addition, while the example process includes a separate branch for detecting a synch sequence, synch sequences may be detected in the same branch as other sequences and processing may later be performed to identify a synch sequence that indicates that start of a message thread. Further, while the process of FIG. 7 is illustrated as a continuous loop, any flow may be utilized.

[0050] FIG. 8 is a schematic diagram of an example processor platform 800 that may be used and/or programmed to implement any or all of the example system 100 and the

decoder 104, and/or any other component described herein. For example, the processor platform 800 can be implemented by one or more general purpose processors, processor cores, microcontrollers, etc. Additionally, the processor platform 800 may be implemented as a part of a device having other functionality. For example, the processor platform 800 may be implemented using processing power provided in a mobile telephone, or any other handheld device.

[0051] The processor platform 800 of the example of FIG. 8 includes at least one general purpose programmable processor 805. The processor 805 executes coded instructions 810 and/or 812 present in main memory of the processor 805 (e.g., within a RAM 815 and/or a ROM 820). The processor 805 may be any type of processing unit, such as a processor core, a processor and/or a microcontroller. The processor 805 may execute, among other things, example machine accessible instructions implementing the processes described herein. The processor 805 is in communication with the main memory (including a ROM 820 and/or the RAM 815) via a bus 825. The RAM 815 may be implemented by DRAM, SDRAM, and/or any other type of RAM device, and ROM may be implemented by flash memory and/or any other desired type of memory device. Access to the memory 815 and 820 may be controlled by a memory controller (not shown).

[0052] The processor platform 800 also includes an interface circuit 830. The interface circuit 830 may be implemented by any type of interface standard, such as a USB interface, a Bluetooth interface, an external memory interface, serial port, general purpose input/output, etc. One or more input devices 835 and one or more output devices 840 are connected to the interface circuit 830.

[0053] Although certain example apparatus, methods, and articles of manufacture are described herein, other implementations are possible. The scope of coverage of this patent is not limited to the specific examples described herein. On the contrary, this patent covers all apparatus, methods, and articles of manufacture falling within the scope of the invention.

What is claimed is:

1. A method to extract information from media content, the method comprising:
 - receiving a media content signal;
 - sampling the media content signal to generate digital samples;
 - determining a frequency domain representation of the digital samples;
 - determining a first rank of a first frequency in the frequency domain representation;
 - determining a second rank of a second frequency in the frequency domain representation;
 - combining the first rank and the second rank with a set of ranks to create a combined set of ranks;
 - comparing the combined set of ranks to a set of reference sequences;
 - determining information represented by the combined set of ranks based on the comparison; and
 - storing the data in a tangible memory.
2. A method as defined in claim 1, wherein comparing the combined set of ranks to the reference set of ranks includes determining a set of distances between the combined set of ranks and one or more of the sequences in the reference set of sequences.
3. A method as defined in claim 2, wherein determining the data represented by the combined set of ranks includes selecting a sequence in the reference set of sequences that has the smallest distance.
4. A method as defined in claim 1, wherein the first rank indicates an amplitude of the first frequency of interest relative to other frequencies in a neighborhood.
5. A method as defined in claim 4, wherein a number of frequencies in the neighborhood is equal to a number of possible rank values.
6. A method as defined in claim 1,
 - wherein the first rank and the second rank are associated with a first message at a first time and
 - wherein combining the first rank and the second rank with the set of ranks to create the combined set of ranks includes determining an average of the first rank and at least another rank for the first frequency of interest of at least another message at a time earlier than the first time to determine a first average rank, determining an average of the second rank and at least another rank for the second frequency of interest of at least another message at a time earlier than the first time to determine a second average rank, combining the first

average rank and the second average rank with the set of ranks to create the combined set of ranks.

7. A method as defined in claim 1, further comprising determining a first average rank for a first frequency in the frequency domain representation and determining a second average rank for a second frequency in the frequency domain representation.

8. A method as defined in claim 7, wherein the information is encoded in the media content signal after T seconds and wherein determining a first average rank for the first frequency comprises determining a third rank for a first frequency approximately T seconds before determining the first rank and adding the first rank and the third rank.

9. An apparatus to extract information from media content, the apparatus comprising:
a sampler to receive a media content signal and to sample the media content signal to generate digital samples;

a time domain to frequency domain converter to determine a frequency domain representation of the digital samples;

a ranker to determine a first rank of a first frequency in the frequency domain representation and to determine a second rank of a second frequency in the frequency domain representation; and

a comparator to combine the first rank and the second rank with a set of ranks to create a combined set of ranks, to compare the combined set of ranks to a set of reference sequences, to determine information represented by the combined set of ranks based on the comparison, and to store the data in a tangible memory.

10. An apparatus as defined in claim 9, wherein comparing the combined set of ranks to the reference set of ranks includes determining a set of distances between the combined set of ranks and one or more of the sequences in the reference set of sequences.

11. An apparatus as defined in claim 10, wherein comparator is to determine the data represented by the combined set of ranks by selecting a sequence in the reference set of sequences that has the smallest distance.

12. An apparatus as defined in claim 9, wherein the first rank indicates an amplitude of the first frequency of interest relative to other frequencies in a neighborhood.

13. An apparatus as defined in claim 12, wherein a number of frequencies in the neighborhood is equal to a number of possible rank values.

14. An apparatus as defined in claim 9,
wherein the first rank and the second rank are associated with a first message at a first time and

wherein the comparator is to combine the first rank and the second rank with the set of ranks to create the combined set of ranks by determining an average of the first rank and at least another rank for the first frequency of interest of at least another message at a time earlier than the first time to determine a first average rank, determining an average of the second rank and at least another rank for the second frequency of interest of at least another message at a time earlier than the first time to determine a second average rank, and combining the first average rank and the second average rank with the set of ranks to create the combined set of ranks.

15. An apparatus as defined in claim 9, further comprising a stacker to determine a first average rank for a first frequency in the frequency domain representation and to determine a second average rank for a second frequency in the frequency domain representation.

16. An apparatus as defined in claim 15, wherein the information is encoded in the media content signal after T seconds and wherein determining a first average rank for the first frequency comprises determining a third rank for a first frequency approximately T seconds before determining the first rank and adding the first rank and the third rank.

17. A computer readable medium storing instructions that, when executed, cause a machine to extract information from media content by:

receiving a media content signal;

sampling the media content signal to generate digital samples;

determining a frequency domain representation of the digital samples;

determining a first rank of a first frequency in the frequency domain representation;

determining a second rank of a second frequency in the frequency domain

representation;

combining the first rank and the second rank with a set of ranks to create a combined set of ranks;

comparing the combined set of ranks to a set of reference sequences;

determining information represented by the combined set of ranks based on the comparison; and

storing the data in a tangible memory.

18. A machine readable medium as defined in claim 17, wherein comparing the combined set of ranks to the reference set of ranks includes determining a set of distances between the combined set of ranks and one or more of the sequences in the reference set of sequences.
19. A machine readable medium as defined in claim 18, wherein determining the data represented by the combined set of ranks includes selecting a sequence in the reference set of sequences that has the smallest distance.
20. A machine readable medium as defined in claim 17, wherein the first rank indicates an amplitude of the first frequency of interest relative to other frequencies in a neighborhood.
21. A machine readable medium as defined in claim 20, wherein a number of frequencies in the neighborhood is equal to a number of possible rank values.
22. A machine readable medium as defined in claim 17,
wherein the first rank and the second rank are associated with a first message at a first time and
wherein combining the first rank and the second rank with the set of ranks to create the combined set of ranks includes determining an average of the first rank and at least another rank for the first frequency of interest of at least another message at a time earlier than the first time to determine a first average rank, determining an average of the second rank and at least another rank for the second frequency of interest of at least another message at a time earlier than the first time to determine a second average rank, combining the first average rank and the second average rank with the set of ranks to create the combined set of ranks.
23. A machine readable medium as defined in claim 17, further comprising determining a first average rank for a first frequency in the frequency domain representation and determining a second average rank for a second frequency in the frequency domain representation.
24. A machine readable medium as defined in claim 23, wherein the information is encoded in the media content signal after T seconds and wherein determining a first average rank for the first frequency comprises determining a third rank for a first frequency approximately T seconds before determining the first rank and adding the first rank and the third rank.

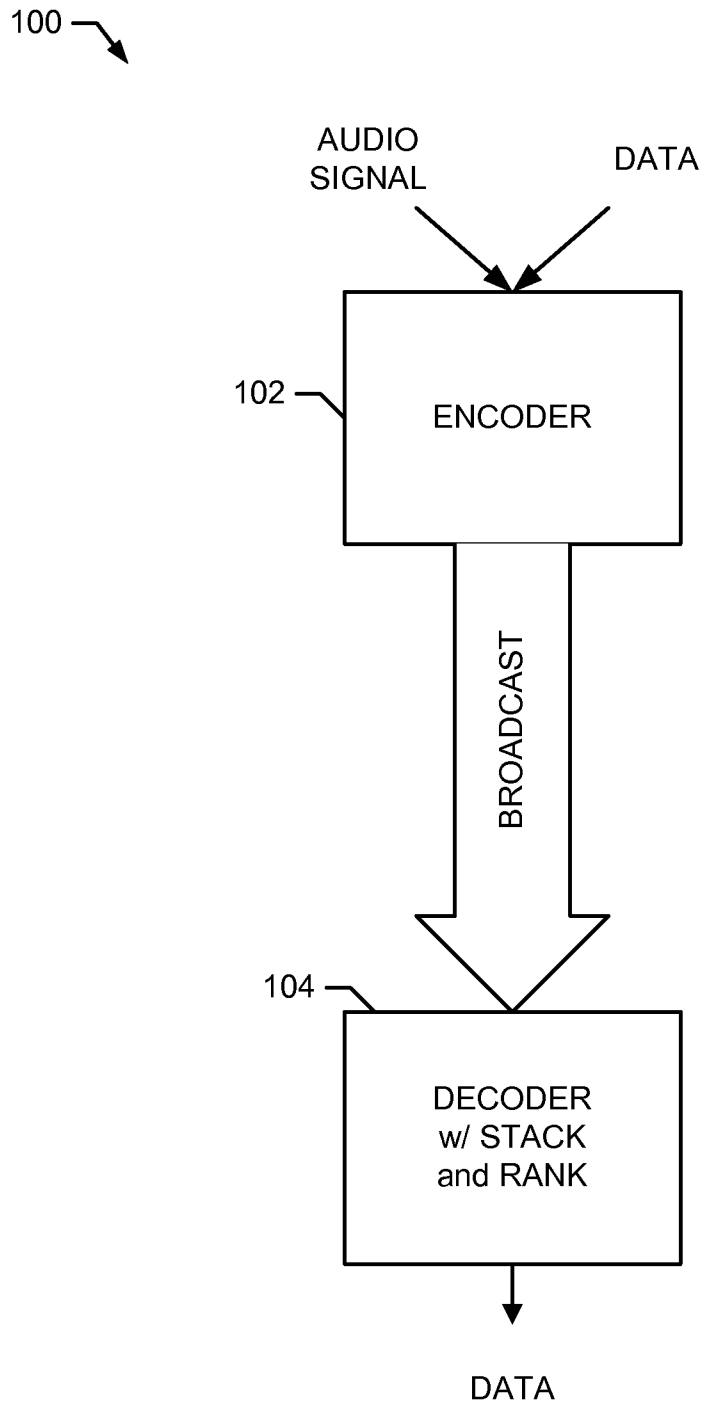


FIG. 1

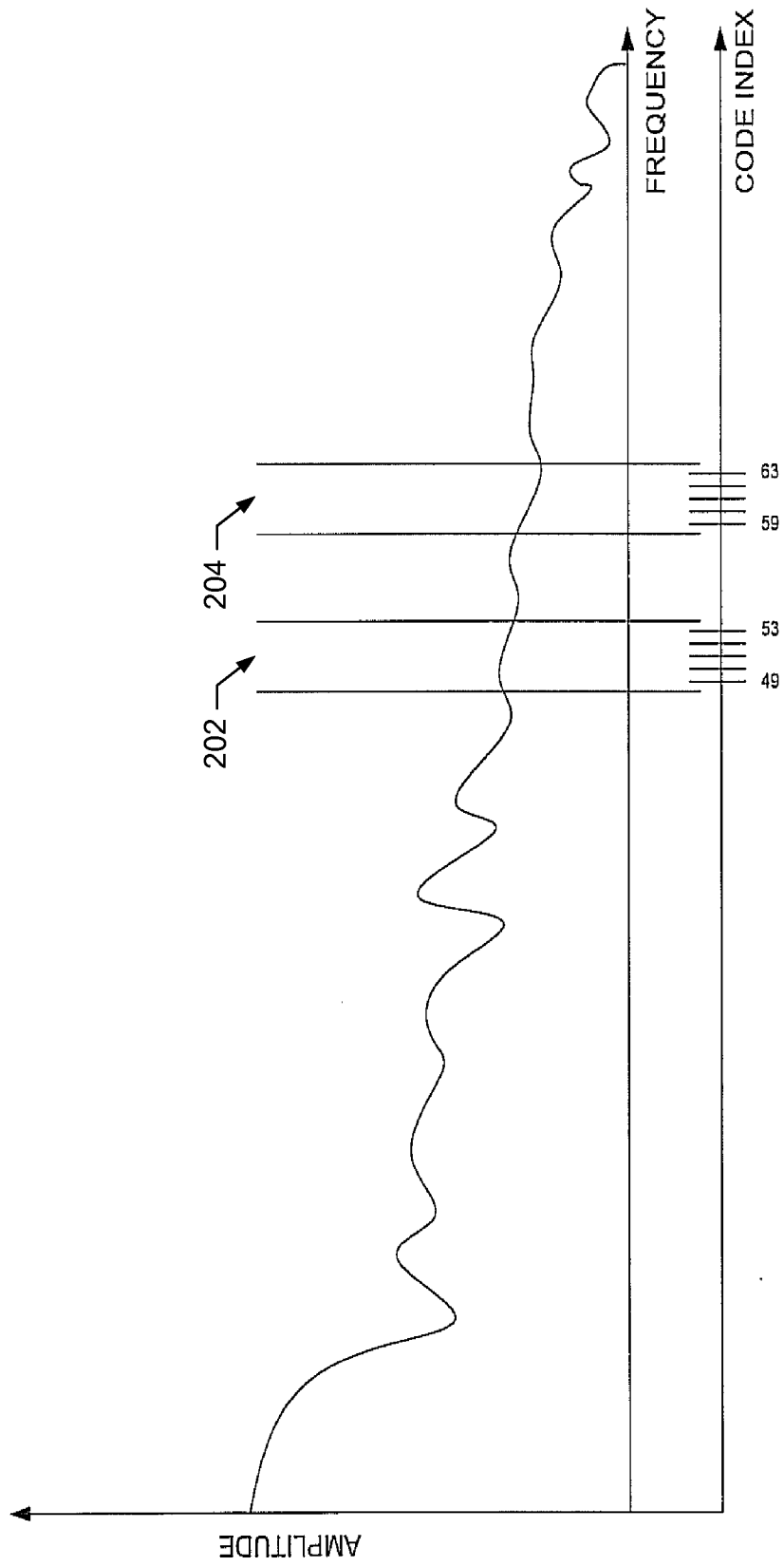


FIG. 2

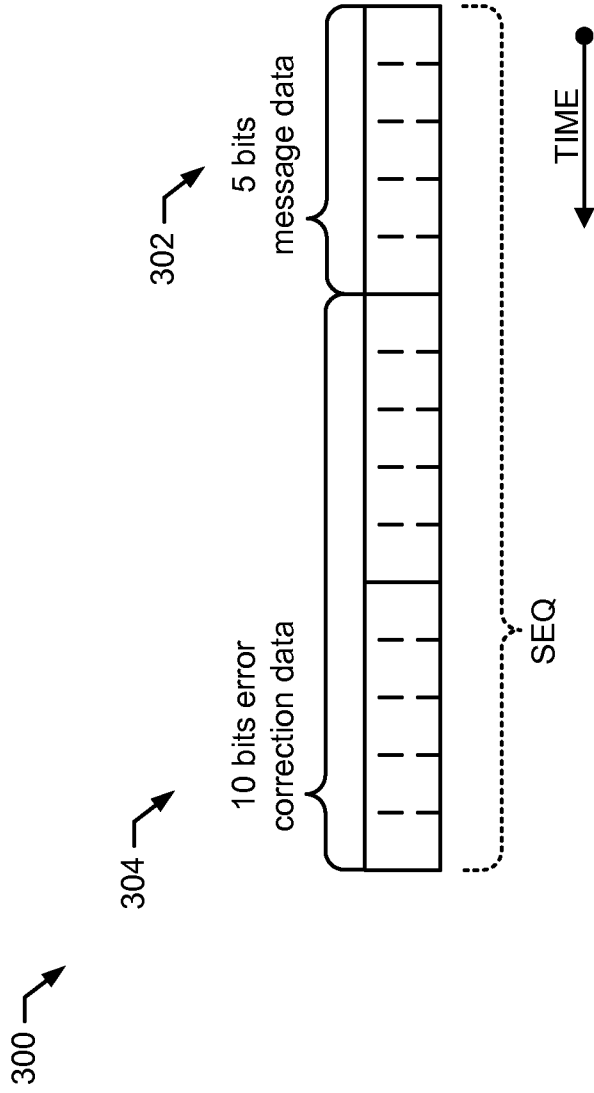


FIG. 3

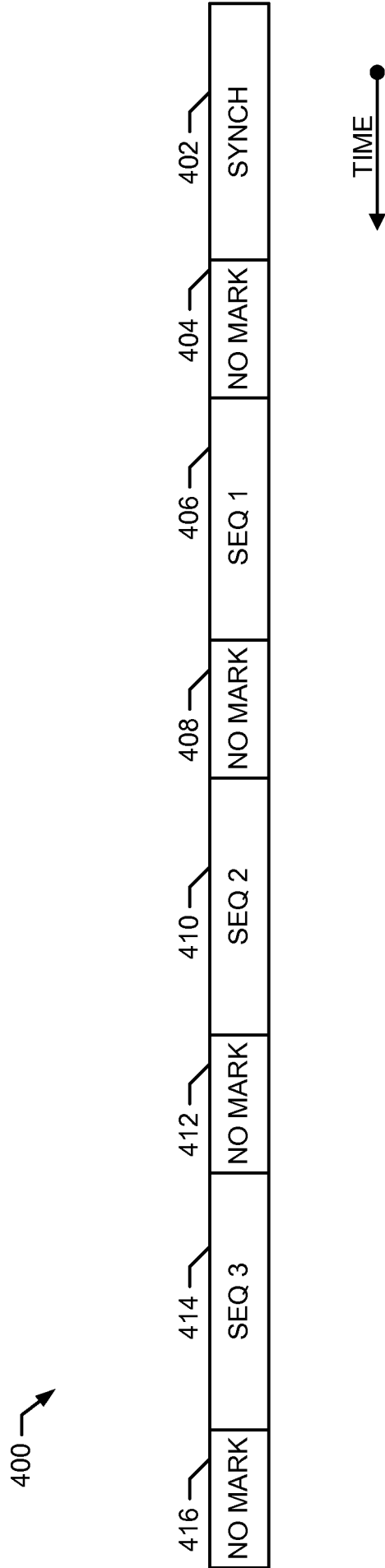


FIG. 4

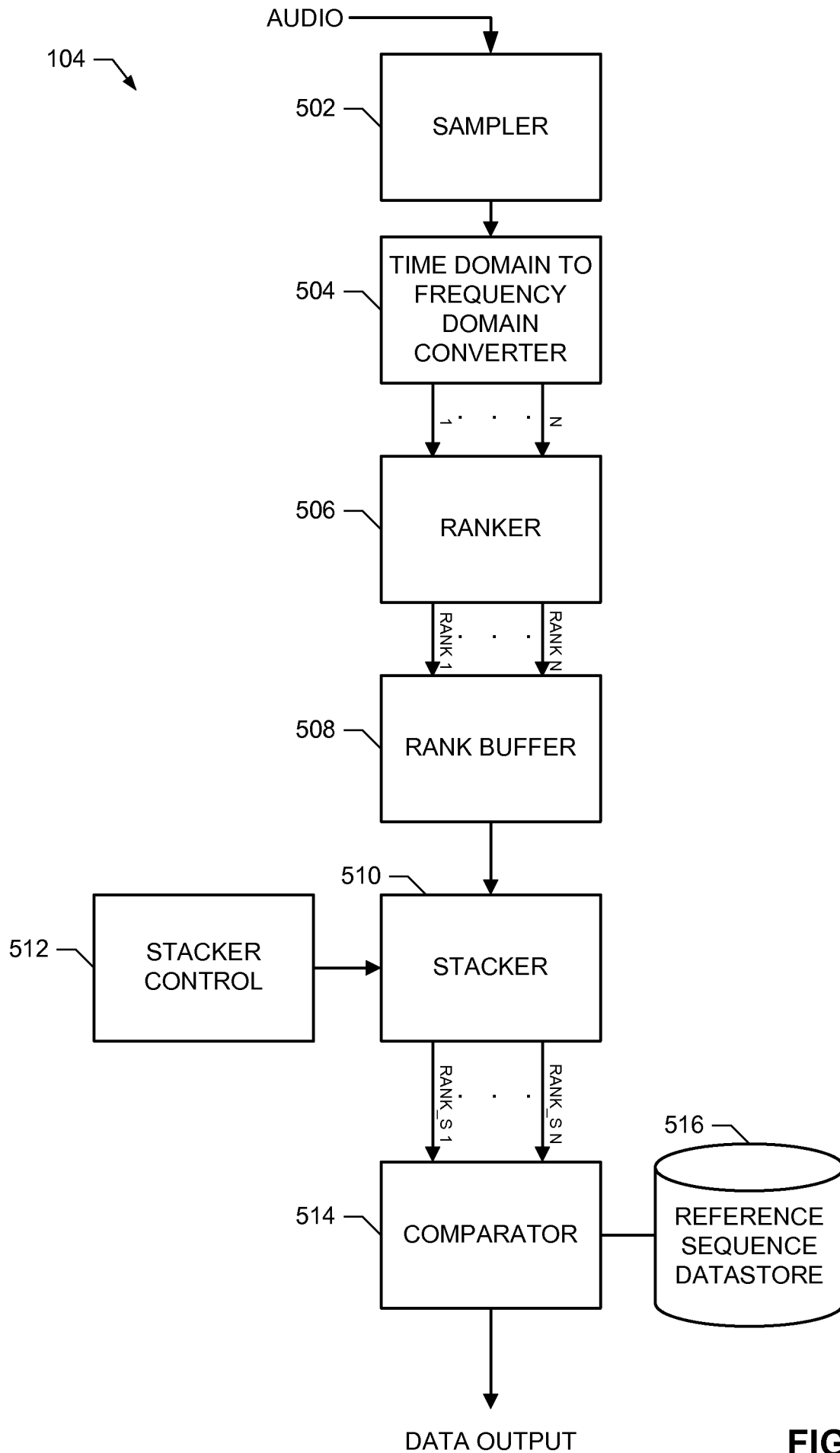


FIG. 5

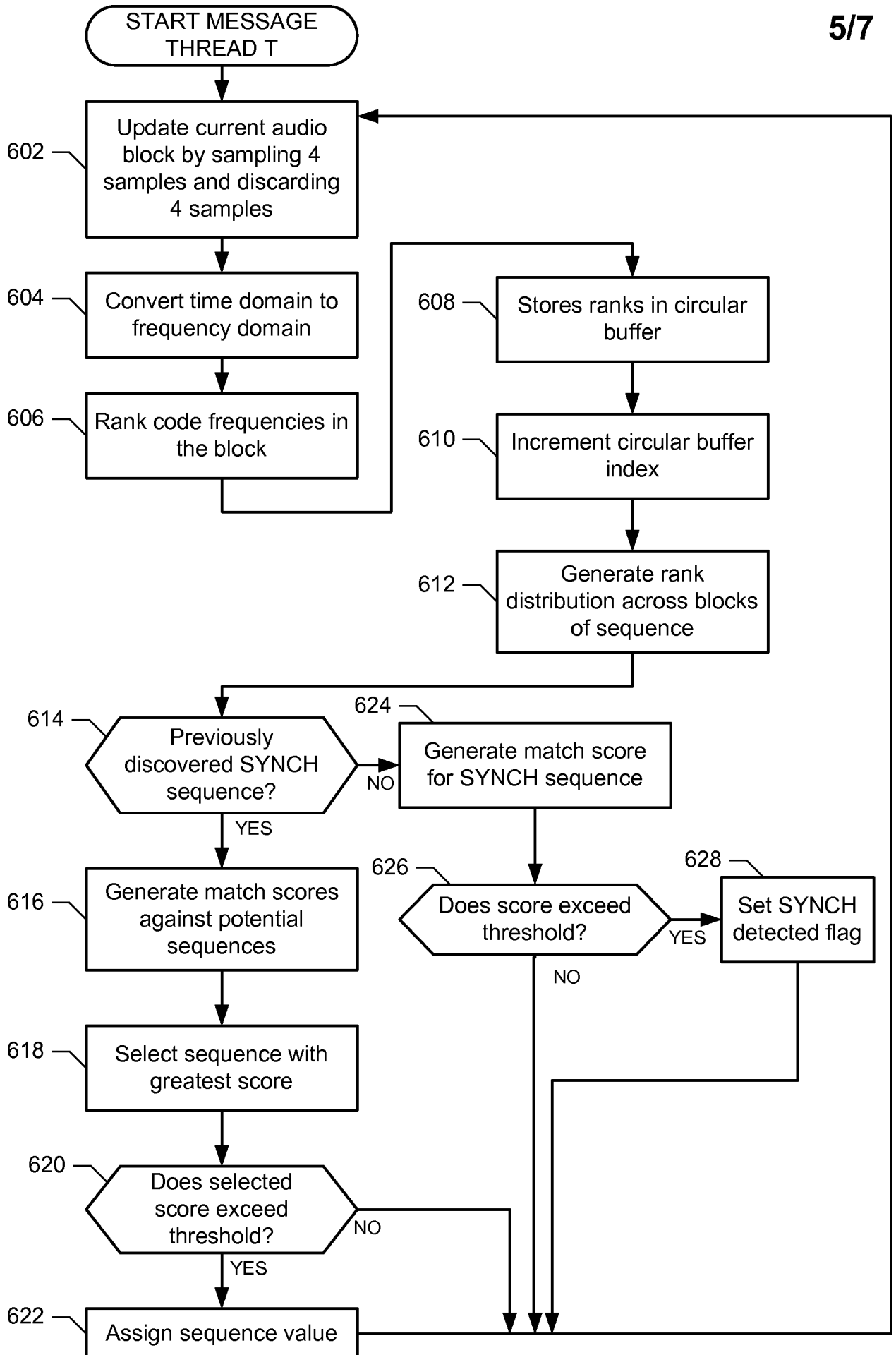


FIG. 6

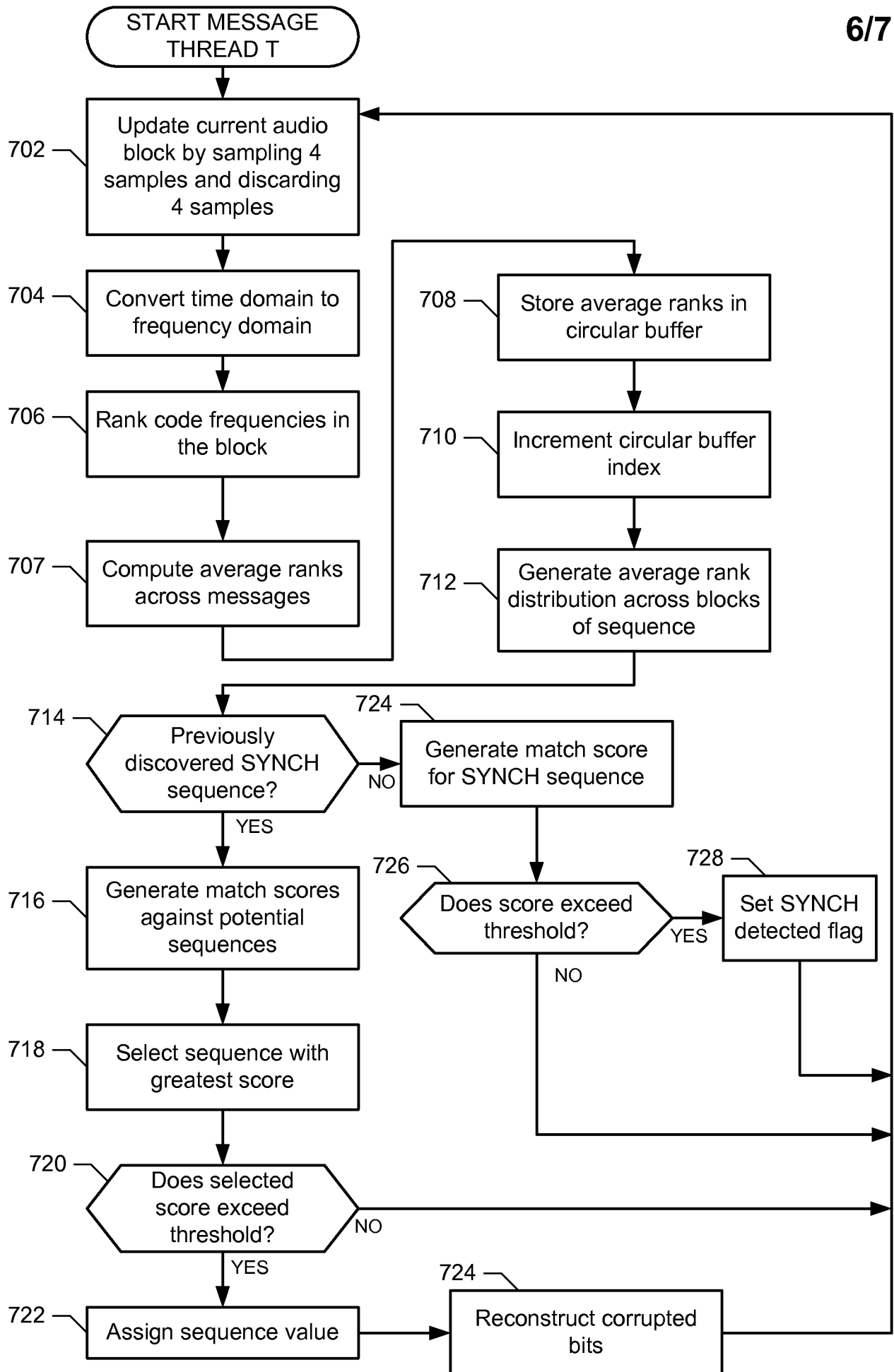


FIG. 7

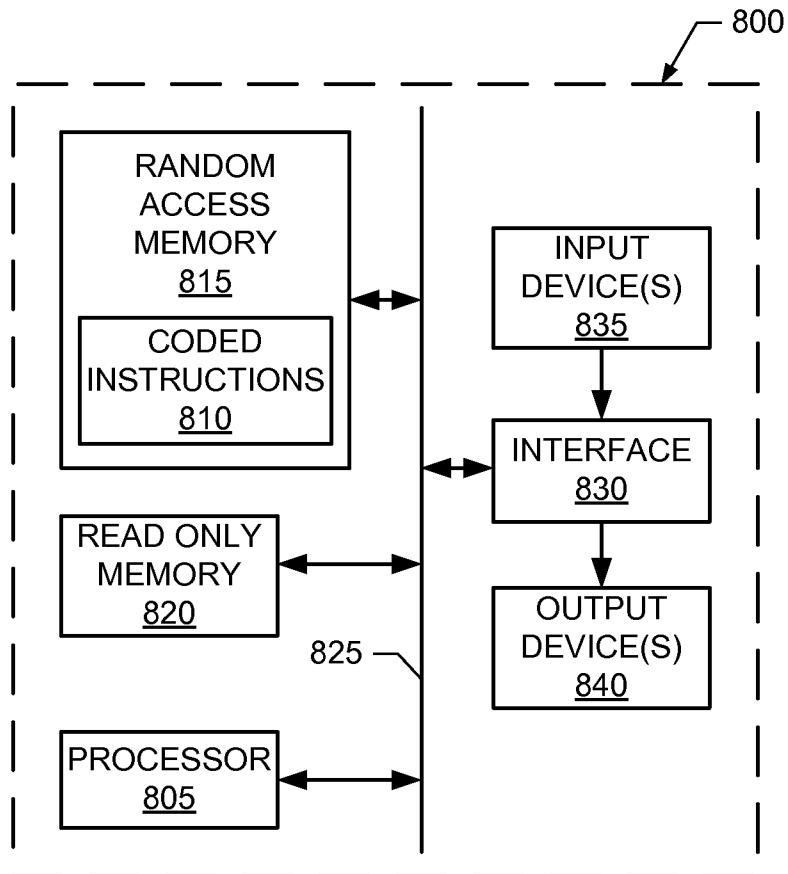


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2009/061827

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G11B20/10 H04H60/58 H04H20/31 H04H60/37

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)
 G11B H04H G10L

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, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	AU 2006 203 639 A1 (ARBITRON INC) 14 September 2006 (2006-09-14) page 3, line 11 - page 4, line 13 page 16, line 4 - page 23, line 2; figures 1,7,8,10	1,9,17 2-8, 10-16, 18-24
X A	US 6 421 445 B1 (JENSEN JAMES M [US] ET AL) 16 July 2002 (2002-07-16) paragraphs [0146] - [0161]	1,9,17 2-8, 10-16, 18-24
A	WO 00/04662 A (NIELSEN MEDIA RES INC [US]) 27 January 2000 (2000-01-27) the whole document	1-24
	-/--	

Further documents are listed in the continuation of Box C. 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	"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
"E" earlier document but published on or after the international filing date	"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
"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)	"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.
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 5 February 2010	Date of mailing of the international search report 15/03/2010
--	--

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2009/061827

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008/133223 A1 (SON CHANG-YONG [KR] ET AL) 5 June 2008 (2008-06-05) paragraphs [0014] - [0024], [0062] - [0107]; figures -----	1-24

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2009/061827

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
AU 2006203639	A1	14-09-2006	NONE
US 6421445	B1	16-07-2002	AT 410047 B 27-01-2003 AU 709873 B2 09-09-1999 AU 2196995 A 23-10-1995 BR 9507230 A 16-09-1997 CA 2185790 A1 12-10-1995 CH 694652 A5 13-05-2005 CN 1149366 A 07-05-1997 CZ 9602840 A3 16-04-1997 DE 19581594 T0 27-03-1997 DK 105996 A 14-11-1996 EP 0753226 A1 15-01-1997 FI 963827 A 29-11-1996 GB 2302000 A 18-12-1996 HU 219628 B 28-05-2001 HU 219627 B 28-05-2001 HU 219668 B 28-06-2001 HU 219667 B 28-06-2001 HU 76453 A2 29-09-1997 IL 113190 A 29-06-2000 JP 10500263 T 06-01-1998 JP 2006154851 A 15-06-2006 LU 88820 A1 18-03-1997 NO 964062 A 19-11-1996 NZ 283612 A 24-09-1998 NZ 331166 A 28-07-2000 PL 316631 A1 03-02-1997 PL 187110 B1 31-05-2004 PL 183307 B1 28-06-2002 PL 183573 B1 28-06-2002 PL 180441 B1 28-02-2001 SE 519882 C2 22-04-2003 SE 9603570 A 29-11-1996 WO 9527349 A1 12-10-1995
WO 0004662	A	27-01-2000	AU 771289 B2 18-03-2004 AU 2003204499 A1 17-07-2003 AU 2004201423 A1 29-04-2004 AU 2007200368 A1 01-03-2007 CA 2332977 A1 27-01-2000 CN 1303547 A 11-07-2001 DE 69838401 T2 19-06-2008 EP 1095477 A1 02-05-2001 ES 2293693 T3 16-03-2008 HK 1040334 A1 11-01-2008 HK 1066351 A1 11-07-2008 JP 4030036 B2 09-01-2008 JP 2002521702 T 16-07-2002 US 6272176 B1 07-08-2001 US 2003194004 A1 16-10-2003 US 2002034224 A1 21-03-2002 US 2001053190 A1 20-12-2001
US 2008133223	A1	05-06-2008	WO 2008069468 A1 12-06-2008 KR 20080050900 A 10-06-2008