## ABSTRACT

A method for adding a binary code bit to a block of a signal varying within a predetermined signal bandwidth comprising the following steps: a) selecting a reference frequency within the predetermined signal bandwidth, and associating therewith both a first code frequency having a first predetermined offset from the reference frequency .and a second code frequency having a second predetermined offset from the reference frequency; b) measuring the spectral power of the signal in a first neighborhood of frequencies extending about the first code frequency and in a second neighborhood of frequencies extending about the second code frequency; c) increasing the spectral power at the first code frequency so as to render the spectral power at the first code frequency a maximum in the first neighborhood of frequencies; and d) decreasing the spectral power at the second code frequency so as to render the spectral power at the second code frequency a minimum in the second neighborhood of frequencies.

(74)

# FIGURE 1
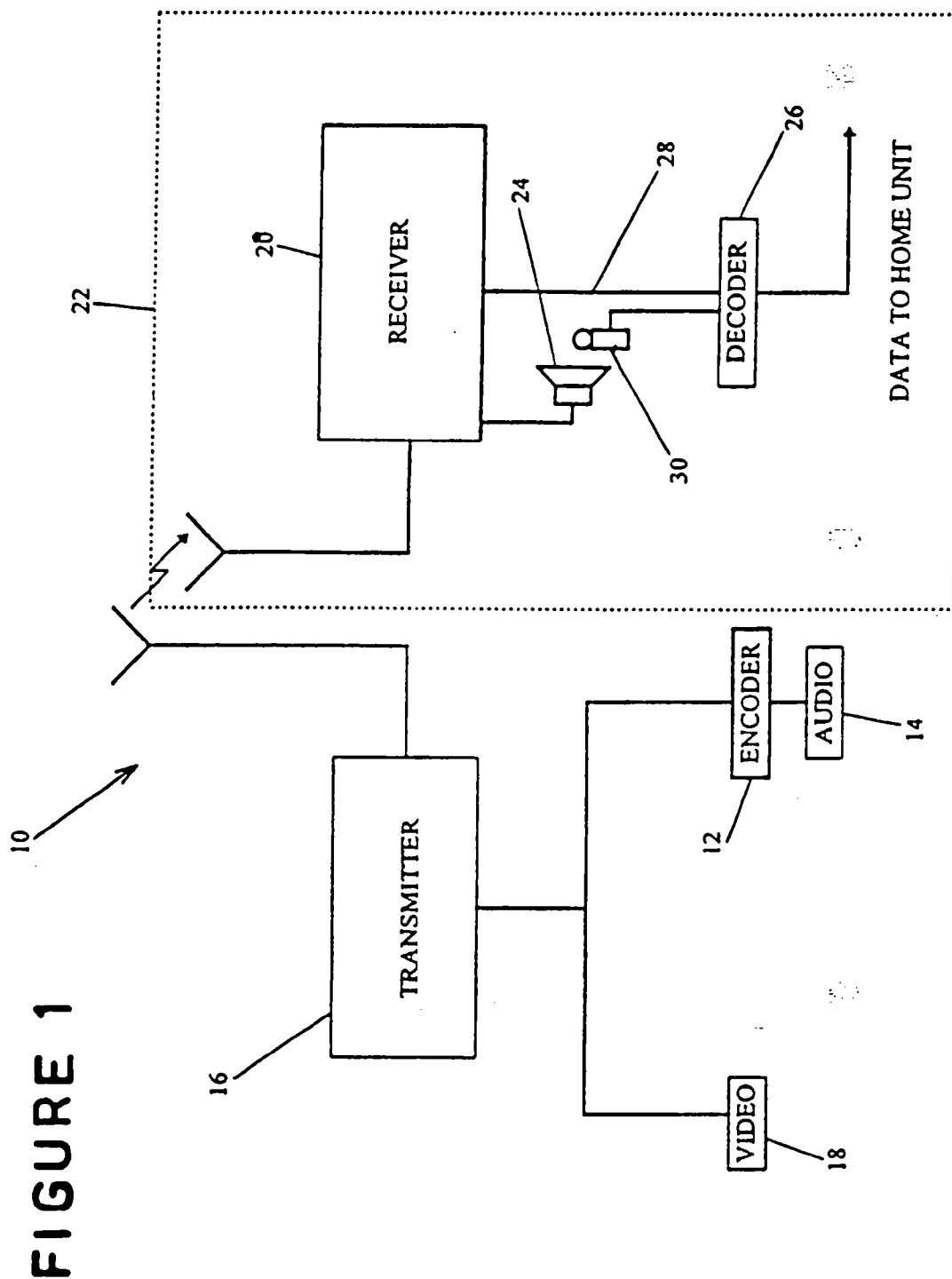
# AUSTRALIA

## *Patents Act 1990*

# COMPLETE SPECIFICATION

# STANDARD PATENT

## (ORIGINAL)

Name of Applicant(s):  **Nielsen Media Research, Inc.**, of 150 N. Martingale Road, Schaumburg, Illinois 60173, United States of America

Actual Inventor(s):  SRINIVASAN, Venugopal

Address for Service:

DAVIES COLLISON CAVE, Patent & Trademark Attorneys, of 1 Nicholson Street, Melbourne, 3000, Victoria, Australia
Ph: 03 9254 2777    Fax: 03 9254 2770    Attorney Code: DM

Invention Title:  **"System and method for encoding an audio signal, by adding an inaudible code to the audio signal, for use in broadcast programme identification systems"**

The following statement is a full description of this invention, including the best method of performing it known to us:-

- 1 -

# SYSTEM AND METHOD FOR ENCODING AN AUDIO SIGNAL, BY ADDING AN INAUDIBLE CODE TO THE AUDIO SIGNAL, FOR USE IN BROADCAST PROGRAMME IDENTIFICATION SYSTEMS

## Field of the Invention

The invention relates to a method of encoding a signal. For example, the present invention relates to a system and method for adding an inaudible code to an audio signal and subsequently retrieving that code. Such a code may be used, for example, in an audience measurement application in order to identify a broadcast program.

## Background of the Invention

There are many arrangements for adding an ancillary code to a signal in such a way that the added code is not noticed. It is well known in television broadcasting, for example, to hide such ancillary codes in non-viewable portions of video by inserting them into either the video's vertical blanking interval or horizontal retrace interval. An exemplary system which hides codes in non-viewable portions of video is referred to as "AMOL" and is taught in U.S. Patent No. 4,025,851. This system is used by the assignee of this application for monitoring broadcasts of television programming as well as the times of such broadcasts.

Other known video encoding systems have sought to bury the ancillary code in a portion of a television signal's transmission bandwidth that otherwise carries little signal

- 1a -

energy. An example of such a system is disclosed by Dougherty in U.S. Patent No. 5,629,739, which is assigned to the assignee of the present application.

Other methods and systems add ancillary codes to audio signals for the purpose of identifying the signals and, perhaps, for tracing their courses through signal distribution systems. Such arrangements have the obvious advantage of being applicable not only to television, but also to radio broadcasts and to pre-recorded music. Moreover, ancillary codes which are added to audio signals may be reproduced in the audio signal output by a speaker. Accordingly, these arrangements offer the possibility of non-intrusively intercepting and decoding the codes with equipment that has microphones as inputs. In particular, these arrangements provide an approach to measuring broadcast audiences by the use of portable metering equipment carried by panelists.

In the field of encoding audio signals for broadcast audience measurement purposes, Crosby, in U.S. Patent No. 3,845,391, teaches an audio encoding approach in which the code is inserted in a narrow frequency "notch" from which the original audio signal is deleted. The notch is made at a fixed predetermined frequency (e.g., 40 Hz). This approach led to codes that were audible when the original audio signal containing the code was of low intensity.

A series of improvements followed the Crosby patent. Thus, Howard, in U.S. Patent No. 4,703,476, teaches the use of two separate notch frequencies for the mark and the space portions of a code signal. Kramer, in U.S. Patent No.

-2-

4,931,871 and in U.S. Patent No. 4,945,412 teaches, *inter alia*, using a code signal having an amplitude that tracks the amplitude of the audio signal to which the code is added.

Broadcast audience measurement systems in which panelists are expected to carry microphone-equipped audio monitoring devices that can pick up and store inaudible codes broadcast in an audio signal are also known. For example, Aijalla et al., in WO 94/11989 and in U.S. Patent No. 5,579,124, describe an arrangement in which spread spectrum techniques are used to add a code to an audio signal so that the code is either not perceptible, or can be heard only as low level "static" noise. Also, Jensen et al., in U.S. Patent No. 5,450,490, teach an arrangement for adding a code at a fixed set of frequencies and using one of two masking signals, where the choice of masking signal is made on the basis of a frequency analysis of the audio signal to which the code is to be added. Jensen et al. do not teach a coding arrangement in which the code frequencies vary from block to block. The intensity of the code inserted by Jensen et al. is a predetermined fraction of a measured value (e.g., 30 dB down from peak intensity) rather than comprising relative maxima or minima.

Moreover, Preuss et al., in U.S. Patent No. 5,319,735, teach a multi-band audio encoding arrangement in which a spread spectrum code is inserted in recorded music at

-3-

a fixed ratio to the input signal intensity (code-to-music
ratio) that is preferably 19 dB.  Lee et al., in U.S. Patent
No. 5,687,191, teach an audio coding arrangement suitable for
use with digitized audio signals in which the code intensity
is made to match the input signal by calculating a signal-to-
mask ratio in each of several frequency bands and by then
inserting the code at an intensity that is a predetermined
ratio of the audio input in that band.  As reported in this
patent, Lee et al. have also described a method of embedding
digital information in a digital waveform in pending U.S.
application Serial No. 08/524,132.

It will be recognized that, because ancillary codes
are preferably inserted at low intensities in order to prevent
the code from distracting a listener of program audio, such
codes may be vulnerable to various signal processing
operations.  For example, although Lee et al. discuss
digitized audio signals, it may be noted that many of the
earlier known approaches to encoding a broadcast audio signal
are not compatible with current and proposed digital audio
standards, particularly those employing signal compression
methods that may reduce the signal's dynamic range (and
thereby delete a low level code) or that otherwise may damage
an ancillary code.  In this regard, it is particularly
important for an ancillary code to survive compression and
subsequent de-compression by the AC-3 algorithm or by one of

-4-

the algorithms recommended in the ISO/IEC 11172 MPEG standard, which is expected to be widely used in future digital television broadcasting systems.

5  It is generally desirable to overcome or ameliorate one or more of the above described difficulties, or to at least provide a useful alternative.

## Summary of the Invention

10

In accordance with one aspect of the present invention, there is provided a method of encoding a signal, comprising:

a)    measuring a characteristic of the signal at a plurality of frequencies associated with the signal;

b)     modulating the signal at one or more of the plurality
       of frequencies if the characteristic of the signal at
       the one or more of the plurality of frequencies is not
       one of a local minimum or a local maximum; and

5     c)     foregoing modulating the signal at the one or more of
       the plurality of frequencies if the characteristic of
       the signal at the one or more of the plurality of
       frequencies is one of the local minimum or the local
       maximum.

10

## Brief Description of the Drawings

Preferred embodiments of the present invention are hereafter
described, by way of non-limiting example only, with

15     reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of an audience
measurement system employing the signal coding and decoding
arrangements of a preferred embodiment of the present

20     invention;

Figure 2 is flow chart depicting steps performed by an
encoder of the system shown in Figure 1;

25     Figure 3 is a spectral plot of an audio block, wherein the
thin line of the plot is the spectrum of the original audio
signal and the thick line of the plot is the spectrum of the
signal modulated in accordance with a preferred embodiment
of the present invention;

30

Figure 4 depicts a window function which may be used to
prevent transient effects that might otherwise occur at the

boundaries between adjacent encoded blocks;

Figure 5 is a schematic block diagram of an arrangement for generating a seven-bit pseudo-noise synchronization sequence;

Figure 6 is a spectral plot of a "triple tone" audio block which forms the first block of a preferred synchronization sequence, where the thin line of the plot is the spectrum of the original audio signal and the thick line of the plot is the spectrum of the modulated signal;

Figure 7a schematically depicts an arrangement of synchronization and information blocks usable to form a complete code message;

Figure 7b schematically depicts further details of the synchronization block shown in Fig. 7a;

Figure 8 is a flow chart depicting steps performed by a decoder of the system shown in Figure 1; and,

Figure 9 illustrates an encoding arrangement in which audio encoding delays are compensated in the video data stream.

## Detailed Description of Preferred Embodiments of the Invention

Audio signals are usually digitized at sampling rates that range between thirty-two kHz and forty-eight kHz.

31 Mar 2009

2007200368

PAGES 8 TO 14 HAVE BEEN LEFT INTENTIONALLY BLANK

For example, a sampling rate of 44.1 kHz is commonly used during the digital recording of music. However, digital television ("DTV") is likely to use a forty eight kHz sampling rate. Besides the sampling rate, another parameter of interest in digitizing an audio signal is the number of binary bits used to represent the audio signal at each of the instants when it is sampled. This number of binary bits can vary, for example, between sixteen and twenty four bits per sample. The amplitude dynamic range resulting from using sixteen bits per sample of the audio signal is ninety-six dB. This decibel measure is the ratio between the square of the highest audio amplitude ($2^{16} = 65536$) and the lowest audio amplitude ($1^2 = 1$). The dynamic range resulting from using twenty-four bits per sample is 144 dB. Raw audio, which is sampled at the 44.1 kHz rate and which is converted to a sixteen-bit per sample representation, results in a data rate of 705.6 kbits/s.

Compression of audio signals is performed in order to reduce this data rate to a level which makes it possible to transmit a stereo pair of such data on a channel with a throughput as low as 192 kbits/s. This compression typically is accomplished by transform coding. A block consisting of $N_d$ = 1024 samples, for example, may be decomposed, by application of a Fast Fourier Transform or other similar frequency analysis process, into a spectral representation. In order to

prevent errors that may occur at the boundary between one block and the previous or subsequent block, overlapped blocks are commonly used. In one such arrangement where 1024 samples per overlapped block are used, a block includes 512 samples of "old" samples (i.e., samples from a previous block ) and 512 samples of "new" or current samples. The spectral representation of such a block is divided into critical bands where each band comprises a group of several neighboring frequencies. The power in each of these bands can be calculated by summing the squares of the amplitudes of the frequency components within the band.

Audio compression is based on the principle of masking that, in the presence of high spectral energy at one frequency (i.e., the masking frequency), the human ear is unable to perceive a lower energy signal if the lower energy signal has a frequency (i.e., the masked frequency) near that of the higher energy signal. The lower energy signal at the masked frequency is called a masked signal. A masking threshold, which represents either (i) the acoustic energy required at the masked frequency in order to make it audible or (ii) an energy change in the existing spectral value that would be perceptible, can be dynamically computed for each band. The frequency components in a masked band can be represented in a coarse fashion by using fewer bits based on this masking threshold. That is, the masking thresholds and

the amplitudes of the frequency components in each band are coded with a smaller number of bits which constitute the compressed audio. Decompression reconstructs the original signal based on this data.

Figure 1 illustrates an audience measurement system 10 in which an encoder 12 adds an ancillary code to an audio signal portion 14 of a broadcast signal. Alternatively, the encoder 12 may be provided, as is known in the art, at some other location in the broadcast signal distribution chain. A transmitter 16 transmits the encoded audio signal portion with a video signal portion 18 of the broadcast signal. When the encoded signal is received by a receiver 20 located at a statistically selected metering site 22, the ancillary code is recovered by processing the audio signal portion of the received broadcast signal even though the presence of that ancillary code is imperceptible to a listener when the encoded audio signal portion is supplied to speakers 24 of the receiver 20. To this end, a decoder 26 is connected either directly to an audio output 28 available at the receiver 20 or to a microphone 30 placed in the vicinity of the speakers 24 through which the audio is reproduced. The received audio signal can be either in a monaural or stereo format.

ENCODING BY SPECTRAL MODULATION

In order for the encoder 12 to embed digital code data in an audio data stream in a manner compatible with compression technology, the encoder 12 should preferably use frequencies and critical bands that match those used in compression.  The block length $N_c$ of the audio signal that is used for coding may be chosen such that, for example, $jN_c = N_d = 1024$, where $j$ is an integer.  A suitable value for $N_c$ may be, for example, 512.  As depicted by a step 40 of the flow chart shown in Figure 2, which is executed by the encoder 12, a first block $v(t)$ of $jN_c$ samples is derived from the audio signal portion 14 by the encoder 12 such as by use of an analog to digital converter, where $v(t)$ is the time-domain representation of the audio signal within the block.  An optional window may be applied to $v(t)$ at a block 42 as discussed below in additional detail.  Assuming for the moment that no such window is used, a Fourier Transform $\mathfrak{F}\{v(t)\}$ of the block $v(t)$ to be coded is computed at a step 44.  (The Fourier Transform implemented at the step 44 may be a Fast Fourier Transform.)

The frequencies resulting from the Fourier Transform are indexed in the range -256 to +255, where an index of 255 corresponds to exactly half the sampling frequency $f_s$.  Therefore, for a forty-eight kHz sampling frequency, the highest index would correspond to a frequency of twenty-four kHz.  Accordingly, for purposes of this indexing, the index

- 18 -

closest to a particular frequency component $f_j$ resulting from
the Fourier Transform $\Im\{v(t)\}$ is given by the following
equation:

$$I_j = (\frac{255}{24}) \cdot f_j \qquad \text{(1)}$$

where equation (1) is used in the following discussion to
relate a frequency $f_j$ and its corresponding index $I_j$.

The code frequencies $f_i$ used for coding a block may
be chosen from the Fourier Transform $\Im\{v(t)\}$ at a step 46 in
the 4.8 kHz to 6 kHz range in order to exploit the higher
auditory threshold in this band.  Also, each successive bit of
the code may use a different pair of code frequencies $f_1$ and $f_0$
denoted by corresponding code frequency indexes $I_1$ and $I_0$.
There are two preferred ways of selecting the code frequencies
$f_1$ and $f_0$ at the step 46 so as to create an inaudible wide-band
noise like code.


(a)   Direct Sequence

One way of selecting the code frequencies $f_1$ and $f_0$
at the step 46 is to compute the code frequencies by use of a
frequency hopping algorithm employing a hop sequence $H_s$ and a
shift index $I_{shift}$.  For example, if $N_s$ bits are grouped together

to form a pseudo-noise sequence, $H_s$ is an ordered sequence of $N_s$ numbers representing the frequency deviation relative to a predetermined reference index $I_{5k}$. For the case where $N_s = 7$, a hop sequence $H_s = \{2,5,1,4,3,2,5\}$ and a shift index $I_{shift} = 5$ could be used. In general, the indices for the $N_s$ bits resulting from a hop sequence may be given by the following equations:

$$I_1 = I_{5k} + H_s - I_{shift} \tag{2}$$

and

$$I_0 = I_{5k} + H_s + I_{shift}. \tag{3}$$

One possible choice for the reference frequency $f_{5k}$ is five kHz, corresponding to a predetermined reference index $I_{5k} = 53$. This value of $f_{5k}$ is chosen because it is above the average maximum sensitivity frequency of the human ear. When encoding a first block of the audio signal, $I_1$ and $I_0$ for the first block are determined from equations (2) and (3) using a first of the hop sequence numbers; when encoding a second block of the audio signal, $I_1$ and $I_0$ for the second block are determined from equations (2) and (3) using a second of the hop sequence numbers; and so on. For the fifth bit in the sequence $\{2,5,1,4,3,2,5\}$, for example, the hop sequence value is three

and, using equations (2) and (3), produces an index $I_1 = 51$ and an index $I_0 = 61$ in the case where $I_{shift} = 5$. In this example, the mid-frequency index is given by the following equation:

$$I_{mid} = I_{Sk} + 3 = 56 \tag{4}$$

where $I_{mid}$ represents an index mid-way between the code frequency indices $I_1$ and $I_0$. Accordingly, each of the code frequency indices is offset from the mid-frequency index by the same magnitude, $I_{shift}$, but the two offsets have opposite signs.

(b)    Hopping based on low frequency maximum

Another way of selecting the code frequencies at the step 46 is to determine a frequency index $I_{max}$ at which the spectral power of the audio signal, as determined as the step 44, is a maximum in the low frequency band extending from zero Hz to two kHz. In other words, $I_{max}$ is the index corresponding to the frequency having maximum power in the range of 0 - 2 kHz. It is useful to perform this calculation starting at index 1, because index 0 represents the "local" DC component and may be modified by high pass filters used in compression. The code frequency indices $I_1$ and $I_0$ are chosen relative to the frequency index $I_{max}$ so that they lie in a higher frequency band at which the human ear is relatively less sensitive.

-21-

Again, one possible choice for the reference frequency $f_{5k}$ is

five kHz corresponding to a reference index $I_{5k} = 53$ such that

$I_1$ and $I_0$ are given by the following equations:

$$I_1 = I_{5k} + I_{max} - I_{shift} \qquad (5)$$

and

$$I_0 = I_{5k} + I_{max} + I_{shift} \qquad (6)$$

where $I_{shift}$ is a shift index, and where $I_{max}$ varies according to

the spectral power of the audio signal. An important

observation here is that a different set of code frequency

indices $I_1$ and $I_0$ from input block to input block is selected

for spectral modulation depending on the frequency index $I_{max}$

of the corresponding input block. In this case, a code bit is

coded as a single bit: however, the frequencies that are used

to encode each bit hop from block to block.

Unlike many traditional coding methods, such as

Frequency Shift Keying (FSK) or Phase Shift Keying (PSK), the

present invention does not rely on a single fixed frequency.

Accordingly, a "frequency-hopping" effect is created similar

to that seen in spread spectrum modulation systems. However,

unlike spread spectrum, the object of varying the coding

- 22 -

frequencies of the present invention is to avoid the use of a
constant code frequency which may render it audible.

For either of the two code frequencies selection
approaches (a) and (b) described above, there are at least
four methods for encoding a binary bit of data in an audio
block, i.e., amplitude modulation and phase modulation.  These
two methods of modulation are separately described below.

(i)  Amplitude Modulation

In order to code a binary '1' using amplitude
modulation, the spectral power at $I_1$ is increased to a level
such that it constitutes a maximum in its corresponding
neighborhood of frequencies.  The neighborhood of indices
corresponding to this neighborhood of frequencies is analyzed
at a step 48 in order to determine how much the code
frequencies $f_1$ and $f_0$ must be boosted and attenuated so that
they are detectable by the decoder 26.  For index $I_1$, the
neighborhood may preferably extend from $I_1 - 2$ to $I_1 + 2$, and
is constrained to cover a narrow enough range of frequencies
that the neighborhood of $I_1$ does not overlap the neighborhood
of $I_0$.  Simultaneously, the spectral power at $I_0$ is modified in
order to make it a minimum in its neighborhood of indices
ranging from $I_0 - 2$ to $I_0 + 2$.  Conversely, in order to code a
binary '0' using amplitude modulation, the power at $I_0$ is

- 23 -

boosted and the power at $I_1$ is attenuated in their corresponding neighborhoods.

As an example, Figure 3 shows a typical spectrum 50 of an $jN_c$ sample audio block plotted over a range of frequency index from forty five to seventy seven. A spectrum 52 shows the audio block after coding of a '1' bit, and a spectrum 54 shows the audio block before coding. In this particular instance of encoding a '1' bit according to code frequency selection approach (a), the hop sequence value is five which yields a mid-frequency index of fifty eight. The values for $I_1$ and $I_0$ are fifty three and sixty three, respectively. The spectral amplitude at fifty three is then modified at a step 56 of Figure 2 in order to make it a maximum within its neighborhood of indices. The amplitude at sixty three already constitutes a minimum and, therefore, only a small additional attenuation is applied at the step 56.

The spectral power modification process requires the computation of four values each in the neighborhood of $I_1$ and $I_0$. For the neighborhood of $I_1$ these four values are as follows: (1) $I_{max1}$ which is the index of the frequency in the neighborhood of $I_1$ having maximum power; (2) $P_{max1}$ which is the spectral power at $I_{max1}$; (3) $I_{min1}$ which is the index of the frequency in the neighborhood of $I_1$ having minimum power; and (4) $P_{min1}$ which is the spectral power at $I_{min1}$. Corresponding values for the $I_0$ neighborhood are $I_{max0}$, $P_{max0}$, $I_{min0}$, and $P_{min}$.

If $I_{max1} = I_1$, and if the binary value to be coded is a '1,' only a token increase in $P_{max1}$ (i.e., the power at $I_1$) is required at the step 56. Similarly, if $I_{min0} = I_0$, then only a token decrease in $P_{max0}$ (i.e., the power at $I_0$) is required at the step 56. When $P_{max1}$ is boosted, it is multiplied by a factor $1 + A$ at the step 56, where $A$ is in the range of about 1.5 to about 2.0. The choice of $A$ is based on experimental audibility tests combined with compression survivability tests. The condition for imperceptibility requires a low value for $A$, whereas the condition for compression survivability requires a large value for A. A fixed value of A may not lend itself to only a token increase or decrease of power. Therefore, a more logical choice for $A$ would be a value based on the local masking threshold. In this case, A is variable, and coding can be achieved with a minimal incremental power level change and yet survive compression.

In either case, the spectral power at $I_1$ is given by the following equation:

$$P_{I1} = (1 + A) \cdot P_{max1} \tag{7}$$

with suitable modification of the real and imaginary parts of the frequency component at $I_1$. The real and imaginary parts are multiplied by the same factor in order to keep the phase

- 25 -

angle constant.  The power at $I_0$ is reduced to a value corresponding to $(1 + A)^{-1} P_{min0}$ in a similar fashion.

The Fourier Transform of the block to be coded as determined at the step 44 also contains negative frequency components with indices ranging in index values from -256 to -1.  Spectral amplitudes at frequency indices $-I_1$ and $-I_0$ must be set to values representing the complex conjugate of amplitudes at $I_1$ and $I_0$, respectively, according to the following equations:

$$Re[f(-I_1)] = Re[f(I_1)] \tag{8}$$

$$Im[f(-I_1)] = -Im[f(I_1)] \tag{9}$$

$$Re[f(-I_0)] = Re[f(I_0)] \tag{10}$$

$$Im[f(-I_0)] = -Im[f(I_0)] \tag{11}$$

where f(I) is the complex spectral amplitude at index I.  The modified frequency spectrum which now contains the binary code (either '0' or '1') is subjected to an inverse transform operation at a step 62 in order to obtain the encoded time domain signal, as will be discussed below.

Compression algorithms based on the effect of masking modify the amplitude of individual spectral components

by means of a bit allocation algorithm. Frequency bands subjected to a high level of masking by the presence of high spectral energies in neighboring bands are assigned fewer bits, with the result that their amplitudes are coarsely quantized. However, the decompressed audio under most conditions tends to maintain relative amplitude levels at frequencies within a neighborhood. The selected frequencies in the encoded audio stream which have been amplified or attenuated at the step 56 will, therefore, maintain their relative positions even after a compression/decompression process.

It may happen that the Fourier Transform $\mathfrak{F}\{v(t)\}$ of a block may not result in a frequency component of sufficient amplitude at the frequencies $f_i$ and $f_0$ to permit encoding of a bit by boosting the power at the appropriate frequency. In this event, it is preferable not to encode this block and to instead encode a subsequent block where the power of the signal at the frequencies $f_1$ and $f_0$ is appropriate for encoding.

(ii) Modulation by Frequency Swapping

In this approach, which is a variation of the amplitude modulation approach described above in section (i), the spectral amplitudes at $I_1$ and $I_{max1}$ are swapped when encoding a one bit while retaining the original phase angles

at $I_1$ and $I_{max1}$. A similar swap between the spectral amplitudes at $I_0$ and $I_{max0}$ is also performed. When encoding a zero bit, the roles of $I_1$ and $I_0$ are reversed as in the case of amplitude modulation. As in the previous case, swapping is also applied to the corresponding negative frequency indices. This encoding approach results in a lower audibility level because the encoded signal undergoes only a minor frequency distortion. Both the unencoded and encoded signals have identical energy values.

(iii)   Phase Modulation

The phase angle associated with a spectral component $I_0$ is given by the following equation:

$$\phi_0 = \tan^{-1} \frac{Im[f(I_o)]}{Re[f(I_0)]} \qquad (12)$$

where $0 \leq \phi_0 \leq 2\pi$. The phase angle associated with $I_1$ can be computed in a similar fashion. In order to encode a binary number, the phase angle of one of these components, usually the component with the lower spectral amplitude, can be modified to be either in phase (i.e., $0°$) or out of phase (i.e., $180°$) with respect to the other component, which becomes the reference. In this manner, a binary 0 may be encoded as

- 28 -

an in-phase modification and a binary 1 encoded as an out-of-phase modification. Alternatively, a binary 1 may be encoded as an in-phase modification and a binary 0 encoded as an out-of-phase modification. The phase angle of the component that is modified is designated $\phi_M$, and the phase angle of the other component is designated $\phi_R$. Choosing the lower amplitude component to be the modifiable spectral component minimizes the change in the original audio signal.

In order to accomplish this form of modulation, one of the spectral components may have to undergo a maximum phase change of $180^0$, which could make the code audible. In practice, however, it is not essential to perform phase modulation to this extent, as it is only necessary to ensure that the two components are either "close" to one another in phase or "far" apart. Therefore, at the step 48, a phase neighborhood extending over a range of $\pm\pi/4$ around $\phi_R$, the reference component, and another neighborhood extending over a range of $\pm\pi/4$ around $\phi_R + \pi$ may be chosen. The modifiable spectral component has its phase angle $\phi_M$ modified at the step 56 so as to fall into one of these phase neighborhoods depending upon whether a binary '0' or a binary '1' is being encoded. If a modifiable spectral component is already in the appropriate phase neighborhood, no phase modification may be necessary. In typical audio streams, approximately 30 % of the segments are "self-coded" in this manner and no modulation

- 29 -

is required.  The inverse Fourier Transform is determined at the step 62.

### (iv)  Odd/Even Index Modulation

In this odd/even index modulation approach, a single code frequency index, $I_1$, selected as in the case of the other modulation schemes, is used.  A neighborhood defined by indexes $I_1$, $I_1 + 1$, $I_1 + 2$, and $I_1 + 3$, is analyzed to determine whether the index $I_m$ corresponding to the spectral component having the maximum power in this neighborhood is odd or even. If the bit to be encoded is a '1' and the index $I_m$ is odd, then the block being coded is assumed to be "auto-coded." Otherwise, an odd-indexed frequency in the neighborhood is selected for amplification in order to make it a maximum.  A bit '0' is coded in a similar manner using an even index.  In the neighborhood consisting of four indexes, the probability that the parity of the index of the frequency with maximum spectral power will match that required for coding the appropriate bit value is 0.25.  Therefore, 25% of the blocks, on an average, would be auto-coded.  This type of coding will significantly decrease code audibility.

A practical problem associated with block coding by either amplitude or phase modulation of the type described above is that large discontinuities in the audio signal can

- 30 -

arise at a boundary between successive blocks.  These sharp
transitions can render the code audible.  In order to
eliminate these sharp transitions, the time-domain signal v(t)
can be multiplied by a smooth envelope or window function w(t)
at the step 42 prior to performing the Fourier Transform at
the step 44.  No window function is required for the
modulation by frequency swapping approach described herein.
The frequency distortion is usually small enough to produce
only minor edge discontinuities in the time domain between
adjacent blocks.

The window function w(t) is depicted in Figure 4.
Therefore, the analysis performed at the step 54 is limited to
the central section of the block resulting from $\mathfrak{S}_m\{v(t)w(t)\}$.
The required spectral modulation is implemented at the step 56
on the transform $\mathfrak{S}\{v(t)w(t)\}$.

Following the step 62, the coded time domain signal
is determined at a step 64 according to the following
equation:

$$v_0(t) = v(t) + (\mathfrak{S}_m^{-1}(v(t)w(t)) - v(t)w(t)) \tag{13}$$

where the first part of the right hand side of equation (13)
is the original audio signal v(t), where the second part of

the right hand side of equation (13) is the encoding, and where the left hand side of equation (13) is the resulting encoded audio signal $v_0(t)$.

While individual bits can be coded by the method described thus far, practical decoding of digital data also requires (i) synchronization, so as to locate the start of data, and (ii) built-in error correction, so as to provide for reliable data reception. The raw bit error rate resulting from coding by spectral modulation is high and can typically reach a value of 20%. In the presence of such error rates, both synchronization and error-correction may be achieved by using pseudo-noise (PN) sequences of ones and zeroes. A PN sequence can be generated, for example, by using an m-stage shift register 58 (where m is three in the case of Figure 5) and an exclusive-OR gate 60 as shown in Figure 5. For convenience, an n-bit PN sequence is referred to herein as a PNn sequence. For an $N_{PN}$ bit PN sequence, an m-stage shift register is required operating according to the following equation:

$$N_{PN} = 2^m - 1 \qquad\qquad (14)$$

where m is an integer. With m = 3, for example, the 7-bit PN sequence (PN7) is 1110100. The particular sequence depends upon an initial setting of the shift register 58. In one

- 32 -

robust version of the encoder 12, each individual bit of data

is represented by this PN sequence - i.e.,  1110100 is used

for a bit '1,' and the complement 0001011 is used for a bit

'0.'  The use of seven bits to code each bit of code results

in extremely high coding overheads.

An alternative method uses a plurality of PN15

sequences, each of which includes five bits of code data and

10 appended error correction bits.  This representation

provides a Hamming distance of 7 between any two 5-bit code

data words.  Up to three errors in a fifteen bit sequence can

be detected and corrected.  This PN15 sequence is ideally

suited for a channel with a raw bit error rate of 20%.

In terms of synchronization, a unique

synchronization sequence 66 (Figure 7a) is required for

synchronization in order to distinguish PN15 code bit

sequences 74 from other bit sequences in the coded data

stream.  In a preferred embodiment shown in Figure 7b, the

first code block of the synchronization sequence 66 uses a

"triple tone" 70 of the synchronization sequence in which

three frequencies with indices $I_0$, $I_1$, and $I_{mid}$ are all

amplified sufficiently that each becomes a maximum in its

respective neighborhood, as depicted by way of example in

Figure 6.  It will be noted that, although it is preferred to

generate the triple tone 70 by amplifying the signals at the

three selected frequencies to be relative maxima in their

respective frequency neighborhoods, those signals could instead be locally attenuated so that the three associated local extreme values comprise three local minima. It should be noted that any combination of local maxima and local minima could be used for the triple tone 70. However, because broadcast audio signals include substantial periods of silence, the preferred approach involves local amplification rather than local attenuation. Being the first bit in a sequence, the hop sequence value for the block from which the triple tone 70 is derived is two and the mid-frequency index is fifty-five. In order to make the triple tone block truly unique, a shift index of seven may be chosen instead of the usual five. The three indices $I_0$, $I_1$, and $I_{mid}$ whose amplitudes are all amplified are forty-eight, sixty-two and fifty-five as shown in Figure 6. (In this example, $I_{mid} = H_s + 53 = 2 + 53 = 55$.) The triple tone 70 is the first block of the fifteen block sequence 66 and essentially represents one bit of synchronization data. The remaining fourteen blocks of the synchronization sequence 66 are made up of two PN7 sequences: 1110100, 0001011. This makes the fifteen synchronization blocks distinct from all the PN sequences representing code data.

As stated earlier, the code data to be transmitted is converted into five bit groups, each of which is represented by a PN15 sequence. As shown in Figure 7a, an

unencoded block 72 is inserted between each successive pair of PN sequences 74. During decoding, this unencoded block 72 (or gap) between neighboring PN sequences 74 allows precise synchronizing by permitting a search for a correlation maximum across a range of audio samples.

In the case of stereo signals, the left and right channels are encoded with identical digital data. In the case of mono signals, the left and right channels are combined to produce a single audio signal stream. Because the frequencies selected for modulation are identical in both channels, the resulting monophonic sound is also expected to have the desired spectral characteristics so that, when decoded, the same digital code is recovered.

### DECODING THE SPECTRALLY MODULATED SIGNAL

In most instances, the embedded digital code can be recovered from the audio signal available at the audio output 28 of the receiver 20. Alternatively, or where the receiver 20 does not have an audio output 28, an analog signal can be reproduced by means of the microphone 30 placed in the vicinity of the speakers 24. In the case where the microphone 30 is used, or in the case where the signal on the audio output 28 is analog, the decoder 20 converts the analog audio to a sampled digital output stream at a preferred sampling rate matching the sampling rate of the encoder 12. In

decoding systems where there are limitations in terms of

memory and computing power, a half-rate sampling could be

used.  In the case of half-rate sampling, each code block

would consist of $N_c/2$ = 256 samples, and the resolution in the

frequency domain (i.e., the frequency difference between

successive spectral components) would remain the same as in

the full sampling rate case.  In the case where the receiver

20 provides digital outputs, the digital outputs are processed

directly by the decoder 26 without sampling but at a data rate

suitable for the decoder 26.

The task of decoding is primarily one of matching

the decoded data bits with those of a PN15 sequence which

could be either a synchronization sequence or a code data

sequence representing one or more code data bits.  The case of

amplitude modulated audio blocks is considered here.  However,

decoding of phase modulated blocks is virtually identical,

except for the spectral analysis, which would compare phase

angles rather than amplitude distributions, and decoding of

index modulated blocks would similarly analyze the parity of

the frequency index with maximum power in the specified

neighborhood.  Audio blocks encoded by frequency swapping can

also be decoded by the same process.

In a practical implementation of audio decoding,

such as may be used in a home audience metering system, the

ability to decode an audio stream in real-time is highly

desirable.  It is also highly desirable to transmit the
decoded data to a central office.  The decoder 26 may be
arranged to run the decoding algorithm described below on
Digital Signal Processing (DSP) based hardware typically used
in such applications.  As disclosed above, the incoming
encoded audio signal may be made available to the decoder 26
from either the audio output 28 or from the microphone 30
placed in the vicinity of the speakers 24.  In order to
increase processing speed and reduce memory requirements, the
decoder 26 may sample the incoming encoded audio signal at
half (24 kHz) of the normal 48 kHz sampling rate.

Before recovering the actual data bits representing
code information, it is necessary to locate the
synchronization sequence.  In order to search for the
synchronization sequence within an incoming audio stream,
blocks of 256 samples, each consisting of the most recently
received sample and the 255 prior samples, could be analyzed.
For real-time operation, this analysis, which includes
computing the Fast Fourier Transform of the 256 sample block,
has to be completed before the arrival of the next sample.
Performing a 256-point Fast Fourier Transform on a 40 MHZ DSP
processor takes about 600 microseconds.  However, the time
between samples is only 40 microseconds, making real time
processing of the incoming coded audio signal as described
above impractical with current hardware.

Therefore, instead of computing a normal Fast Fourier Transform on each 256 sample block, the decoder 26 may be arranged to achieve real-time decoding by implementing an incremental or sliding Fast Fourier Transform routine 100 (Figure 8) coupled with the use of a status information array SIS that is continuously updated as processing progresses. This array comprises $p$ elements SIS[0] to SIS[$p$-1]. If $p$ = 64, for example, the elements in the status information array SIS are SIS[0] to SIS[63].

Moreover, unlike a conventional transform which computes the complete spectrum consisting of 256 frequency "bins," the decoder 26 computes the spectral amplitude only at frequency indexes that belong to the neighborhoods of interest, i.e., the neighborhoods used by the encoder 12. In a typical example, frequency indexes ranging from 45 to 70 are adequate so that the corresponding frequency spectrum contains only twenty-six frequency bins. Any code that is recovered appears in one or more elements of the status information array SIS as soon as the end of a message block is encountered.

Additionally, it is noted that the frequency spectrum as analyzed by a Fast Fourier Transform typically changes very little over a small number of samples of an audio stream. Therefore, instead of processing each block of 256 samples consisting of one "new" sample and 255 "old" samples,

- 38 -

256 sample blocks may be processed such that, in each block of 256 samples to be processed, the last $k$ samples are "new" and the remaining 256-k samples are from a previous analysis. In the case where $k = 4$, processing speed may be increased by skipping through the audio stream in four sample increments, where a skip factor $k$ is defined as $k = 4$ to account for this operation.

Each element SIS[p] of the status information array SIS consists of five members: a previous condition status PCS, a next jump index JI, a group counter GC, a raw data array DA, and an output data array OP. The raw data array DA has the capacity to hold fifteen integers. The output data array OP stores ten integers, with each integer of the output data array OP corresponding to a five bit number extracted from a recovered PN15 sequence. This PN15 sequence, accordingly, has five actual data bits and ten other bits. These other bits may be used, for example, for error correction. It is assumed here that the useful data in a message block consists of 50 bits divided into 10 groups with each group containing 5 bits, although a message block of any size may be used.

The operation of the status information array SIS is best explained in connection with Figure 8. An initial block of 256 samples of received audio is read into a buffer at a processing stage 102. The initial block of 256 samples is

analyzed at a processing stage 104 by a conventional Fast Fourier Transform to obtain its spectral power distribution. All subsequent transforms implemented by the routine 100 use the high-speed incremental approach referred to above and described below.

In order to first locate the synchronization sequence, the Fast Fourier Transform corresponding to the initial 256 sample block read at the processing stage 102 is tested at a processing stage 106 for a triple tone, which represents the first bit in the synchronization sequence. The presence of a triple tone may be determined by examining the initial 256 sample block for the indices $I_0$, $I_1$, and $I_{mid}$ used by the encoder 12 in generating the triple tone, as described above. The SIS[p] element of the SIS array that is associated with this initial block of 256 samples is SIS[0], where the status array index p is equal to 0. If a triple tone is found at the processing stage 106, the values of certain members of the SIS[0] element of the status information array SIS are changed at a processing stage 108 as follows: the previous condition status PCS, which is initially set to 0, is changed to a 1 indicating that a triple tone was found in the sample block corresponding to SIS[0]; the value of the next jump index JI is incremented to 1; and, the first integer of the raw data member DA[0] in the raw data array DA is set to the value (0 or 1) of the triple tone. In this case, the first

- 40 -

integer of the raw data member DA[0] in the raw data array DA

is set to 1 because it is assumed in this analysis that the

triple tone is the equivalent of a 1 bit.  Also, the status

array index $p$ is incremented by one for the next sample block.

If there is no triple tone, none of these changes in the

SIS[0] element are made at the processing stage 108, but the

status array index $p$ is still incremented by one for the next

sample block.  Whether or not a triple tone is detected in

this 256 sample block, the routine 100 enters an incremental

FFT mode at a processing stage 110.

Accordingly, a new 256 sample block increment is

read into the buffer at a processing stage 112 by adding four

new samples to, and discarding the four oldest samples from,

the initial 256 sample block processed at the processing

stages 102 - 106.  This new 256 sample block increment is

analyzed at a processing stage 114 according to the following

steps:

STEP 1:  the skip factor $k$ of the Fourier Transform is applied

according to the following equation in order to modify each

frequency component $F_{old}(u_0)$ of the spectrum corresponding to

the initial sample block in order to derive a corresponding

intermediate frequency component $F_1(u_0)$:

$$F_1(u_0) = F_{old}(u_0)\exp-(\frac{2\pi u_0 k}{256})$$ (15)

where $u_0$ is the frequency index of interest. In accordance with the typical example described above, the frequency index $u_0$ varies from 45 to 70. It should be noted that this first step involves multiplication of two complex numbers.

STEP 2: the effect of the first four samples of the old 256 sample block is then eliminated from each $F_1(u_0)$ of the spectrum corresponding to the initial sample block and the effect of the four new samples is included in each $F_1(u_0)$ of the spectrum corresponding to the current sample block increment in order to obtain the new spectral amplitude $F_{new}(u_0)$ for each frequency index $u_0$ according to the following equation:

$$F_{new}(u_0) = F_1(u_0) + \sum_{m=1}^{m=4}(f_{new}(m) - f_{old}(m))\exp-(\frac{2\pi u_0(k-m+1)}{256})$$ (16)

where $f_{old}$ and $f_{new}$ are the time-domain sample values. It should be noted that this second step involves the addition of a complex number to the summation of a product of a real number

- 42 -

and a complex number. This computation is repeated across the frequency index range of interest (for example, 45 to 70).

STEP 3: the effect of the multiplication of the 256 sample block by the window function in the encoder 12 is then taken into account. That is, the results of step 2 above are not confined by the window function that is used in the encoder 12. Therefore, the results of step 2 preferably should be multiplied by this window function. Because multiplication in the time domain is equivalent to a convolution of the spectrum by the Fourier Transform of the window function, the results from the second step may be convolved with the window function. In this case, the preferred window function for this operation is the following well known "raised cosine" function which has a narrow 3-index spectrum with amplitudes (-0.50, 1, +0.50):

$$w(t) = \frac{1}{2}[1 - \cos(\frac{2\pi t}{T_W})] \tag{17}$$

where $T_W$ is the width of the window in the time domain. This "raised cosine" function requires only three multiplication and addition operations involving the real and imaginary parts of the spectral amplitude. This operation significantly

improves computational speed. This step is not required for the case of modulation by frequency swapping.

STEP 4: the spectrum resulting from step 3 is then examined for the presence of a triple tone. If a triple tone is found, the values of certain members of the SIS[1] element of the status information array SIS are set at a processing stage 116 as follows: the previous condition status PCS, which is initially set to 0, is changed to a 1; the value of the next jump index JI is incremented to 1; and, the first integer of the raw data member DA[1] in the raw data array DA is set to 1. Also, the status array index $p$ is incremented by one. If there is no triple tone, none of these changes are made to the members of the structure of the SIS[1] element at the processing stage 116, but the status array index $p$ is still incremented by one.

Because $p$ is not yet equal to 64 as determined at a processing stage 118 and the group counter GC has not accumulated a count of 10 as determined at a processing stage 120, this analysis corresponding to the processing stages 112 - 120 proceeds in the manner described above in four sample increments where p is incremented for each sample increment. When SIS[63] is reached where p = 64, p is reset to 0 at the processing stage 118 and the 256 sample block increment now in

the buffer is exactly 256 samples away from the location in the audio stream at which the SIS[0] element was last updated. Each time p reaches 64, the SIS array represented by the SIS[0] - SIS[63] elements is examined to determine whether the previous condition status PCS of any of these elements is one indicating a triple tone. If the previous condition status PCS of any of these elements corresponding to the current 64 sample block increments is not one, the processing stages 112 - 120 are repeated for the next 64 block increments. (Each block increment comprises 256 samples.)

Once the previous condition status PCS is equal to 1 for any of the SIS[0] - SIS[63] elements corresponding to any set of 64 sample block increments, and the corresponding raw data member DA[p] is set to the value of the triple tone bit, the next 64 block increments are analyzed at the processing stages 112 - 120 for the next bit in the synchronization sequence.

Each of the new block increments beginning where p was reset to 0 is analyzed for the next bit in the synchronization sequence. This analysis uses the second member of the hop sequence $H_s$ because the next jump index JI is equal to 1. From this hop sequence number and the shift index used in encoding, the $I_1$ and $I_0$ indexes can be determined, for example from equations (2) and (3). Then, the neighborhoods of the $I_1$ and $I_0$ indexes are analyzed to locate maximums and

minimums in the case of amplitude modulation. If, for example, a power maximum at $I_1$ and a power minimum at $I_0$ are detected, the next bit in the synchronization sequence is taken to be 1. In order to allow for some variations in the signal that may arise due to compression or other forms of distortion, the index for either the maximum power or minimum power in a neighborhood is allowed to deviate by 1 from its expected value. For example, if a power maximum is found in the index $I_1$, and if the power minimum in the index $I_0$ neighborhood is found at $I_0 - 1$, instead of $I_0$, the next bit in the synchronization sequence is still taken to be 1. On the other hand, if a power minimum at $I_1$ and a power maximum at $I_0$ are detected using the same allowable variations discussed above, the next bit in the synchronization sequence is taken to be 0. However, if none of these conditions are satisfied, the output code is set to -1, indicating a sample block that cannot be decoded. Assuming that a 0 bit or a 1 bit is found, the second integer of the raw data member DA[1] in the raw data array DA is set to the appropriate value, and the next jump index JI of SIS[0] is incremented to 2, which corresponds to the third member of the hop sequence $H_S$. From this hop sequence number and the shift index used in encoding, the $I_1$ and $I_0$ indexes can be determined. Then, the neighborhoods of the $I_1$ and $I_0$ indexes are analyzed to locate maximums and minimums in the case of amplitude modulation so that the value

- 46 -

of the next bit can be decoded from the third set of 64 block

increments, and so on for fifteen such bits of the

synchronization sequence.    The fifteen bits stored in the raw

data array DA may then be compared with a reference

synchronization sequence to determine synchronization.    If the

number of errors between the fifteen bits stored in the raw

data array DA and the reference synchronization sequence

exceeds a previously set threshold, the extracted sequence is

not acceptable as a synchronization, and the search for the

synchronization sequence begins anew with a search for a

triple tone.

If a valid synchronization sequence is thus

detected, there is a valid synchronization, and the PN15 data

sequences may then be extracted using the same analysis as is

used for the synchronization sequence, except that detection

of each PN15 data sequence is not conditioned upon detection

of the triple tone which is reserved for the synchronization

sequence.    As each bit of a PN15 data sequence is found, it is

inserted as a corresponding integer of the raw data array DA.

When all integers of the raw data array DA are filled, (i)

these integers are compared to each of the thirty-two possible

PN15 sequences, (ii) the best matching sequence indicates

which 5-bit number to select for writing into the appropriate

array location of the output data array OP, and (iii) the

group counter GC member is incremented to indicate that the

first PN15 data sequence has been successfully extracted. If
the group counter GC has not yet been incremented to 10 as
determined at the processing stage 120, program flow returns
to the processing stage 112 in order to decode the next PN15
data sequence.

When the group counter GC has incremented to 10 as
determined at the processing stage 120, the output data array
OP, which contains a full 50-bit message, is read at a
processing stage 122. The total number of samples in a
message block is 45,056 at a half-rate sampling frequency of
24 kHz. It is possible that several adjacent elements of the
status information array SIS, each representing a message
block separated by four samples from its neighbor, may lead to
the recovery of the same message because synchronization may
occur at several locations in the audio stream which are close
to one another. If all these messages are identical, there is
a high probability that an error-free code has been received.

Once a message has been recovered and the message
has been read at the processing stage 122, the previous
condition status PCS of the corresponding SIS element is set
to 0 at a processing stage 124 so that searching is resumed at
a processing stage 126 for the triple tone of the
synchronization sequence of the next message block.

MULTI-LEVEL CODING

Often there is a need to insert more than one
message into the same audio stream.  For example in a
television broadcast environment, the network originator of
the program may insert its identification code and time stamp,
and a network affiliated station carrying this program may
also insert its own identification code.  In addition, an
advertiser or sponsor may wish to have its code added.  In
order to accommodate such multi-level coding, 48 bits in a 50-
bit system can be used for the code and the remaining 2 bits
can be used for level specification.  Usually the first
program material generator, say the network, will insert codes
in the audio stream.  Its first message block would have the
level bits set to 00, and only a synchronization sequence and
the 2 level bits are set for the second and third message
blocks in the case of a three level system.  For example, the
level bits for the second and third messages may be both set
to 11 indicating that the actual data areas have been left
unused.

The network affiliated station can now enter its
code with a decoder/encoder combination that would locate the
synchronization of the second message block with the 11 level
setting.  This station inserts its code in the data area of
this block and sets the level bits to 01.  The next level
encoder inserts its code in the third message block's data

area and sets the level bits to 10. During decoding, the level bits distinguish each message level category.

CODE ERASURE AND OVERWRITE

It may also be necessary to provide a means of erasing a code or to erase and overwrite a code. Erasure may be accomplished by detecting the triple tone/synchronization sequence using a decoder and by then modifying at least one of the triple tone frequencies such that the code is no longer recoverable. Overwriting involves extracting the synchronization sequence in the audio, testing the data bits in the data area and inserting a new bit only in those blocks that do not have the desired bit value. The new bit is inserted by amplifying and attenuating appropriate frequencies in the data area.

DELAY COMPENSATION

In a practical implementation of the encoder 12, $N_c$ samples of audio, where $N_c$ is typically 512, are processed at any given time. In order to achieve operation with a minimum amount of throughput delay, the following four buffers are used: input buffers IN0 and IN1, and output buffers OUT0 and OUT1. Each of these buffers can hold $N_c$ samples. While samples in the input buffer IN0 are being processed, the input buffer IN1 receives new incoming samples. The processed

output samples from the input buffer IN0 are written into the output buffer OUT0, and samples previously encoded are written to the output from the output buffer OUT1. When the operation associated with each of these buffers is completed, processing begins on the samples stored in the input buffer IN1 while the input buffer IN0 starts receiving new data. Data from the output buffer OUT0 are now written to the output. This cycle of switching between the pair of buffers in the input and output sections of the encoder continues as long as new audio samples arrive for encoding. It is clear that a sample arriving at the input suffers a delay equivalent to the time duration required to fill two buffers at the sampling rate of 48 kHz before its encoded version appears at the output. This delay is approximately 22 ms. When the encoder 12 is used in a television broadcast environment, it is necessary to compensate for this delay in order to maintain synchronization between video and audio.

Such a compensation arrangement is shown in Figure 9. As shown in Figure 9, an encoding arrangement 200, which may be used for the elements 12, 14, and 18 in Figure 1, is arranged to receive either analog video and audio inputs or digital video and audio inputs. Analog video and audio inputs are supplied to corresponding video and audio analog to digital converters 202 and 204. The audio samples from the audio analog to digital converter 204 are provided to an audio

encoder 206 which may be of known design or which may be arranged as disclosed above. The digital audio input is supplied directly to the audio encoder 206. Alternatively, if the input digital bitstream is a combination of digital video and audio bitstream portions, the input digital bitstream is provided to a demultiplexer 208 which separates the digital video and audio portions of the input digital bitstream and supplies the separated digital audio portion to the audio encoder 206.

Because the audio encoder 206 imposes a delay on the digital audio bitstream as discussed above relative to the digital video bitstream, a delay 210 is introduced in the digital video bitstream. The delay imposed on the digital video bitstream by the delay 210 is equal to the delay imposed on the digital audio bitstream by the audio encoder 206. Accordingly, the digital video and audio bitstreams downstream of the encoding arrangement 200 will be synchronized.

In the case where analog video and audio inputs are provided to the encoding arrangement 200, the output of the delay 210 is provided to a video digital to analog converter 212 and the output of the audio encoder 206 is provided to an audio digital to analog converter 214. In the case where separate digital video and audio bitstreams are provided to the encoding arrangement 200, the output of the delay 210 is provided directly as a digital video output of the encoding

arrangement 200 and the output of the audio encoder 206 is provided directly as a digital audio output of the encoding arrangement 200. However, in the case where a combined digital video and audio bitstream is provided to the encoding arrangement 200, the outputs of the delay 210 and of the audio encoder 206 are provided to a multiplexer 216 which recombines the digital video and audio bitstreams as an output of the encoding arrangement 200.

Certain modifications of the present invention have been discussed above. Other modifications will occur to those practicing in the art of the present invention. For example, according to the description above, the encoding arrangement 200 includes a delay 210 which imposes a delay on the video bitstream in order to compensate for the delay imposed on the audio bitstream by the audio encoder 206. However, some embodiments of the encoding arrangement 200 may include a video encoder 218, which may be of known design, in order to encode the video output of the video analog to digital converter 202, or the input digital video bitstream, or the output of the demultiplexer 208, as the case may be. When the video encoder 218 is used, the audio encoder 206 and/or the video encoder 218 may be adjusted so that the relative delay imposed on the audio and video bitstreams is zero and so that the audio and video bitstreams are thereby synchronized. In

this case, the delay 210 is not necessary. Alternatively, the delay 210 may be used to provide a suitable delay and may be inserted in either the video or audio processing so that the relative delay imposed on the audio and video

5    bitstreams is zero and so that the audio and video bitstreams are thereby synchronized.

In still other embodiments of the encoding arrangement 200, the video encoder 218 and not the audio encoder 206 may be

10   used. In this case, the delay 210 may be required in order to impose a delay on the audio bitstream so that the relative delay between the audio and video bitstreams is zero and so that the audio and video bitstreams are thereby synchronized.

15

Accordingly, the description of the present invention is to be construed as illustrative only and is for the purpose of teaching those skilled in the art the best mode of carrying out the invention. The details may be varied substantially

20   without departing from the spirit of the invention, and the exclusive use of all modifications which are within the scope of the appended claims is reserved.

Throughout this specification and the claims which follow,

25   unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

30

The reference in this specification to any prior publication (or information derived from it), or to any matter which is

known, is not, and should not be taken as an acknowledgment
or admission or any form of suggestion that that prior
publication (or information derived from it) or known matter
forms part of the common general knowledge in the field of
5 endeavour to which this specification relates.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A method of encoding a signal, comprising:

   a) measuring a characteristic of the signal at a plurality of frequencies associated with the signal;

   b) modulating the signal at one or more of the plurality of frequencies if the characteristic of the signal at the one or more of the plurality of frequencies is not one of a local minimum or a local maximum; and

   c) foregoing modulating the signal at the one or more of the plurality of frequencies if the characteristic of the signal at the one or more of the plurality of frequencies is one of the local minimum or the local maximum.

2. A method as defined in claim 1, wherein the characteristic of the signal is a spectral characteristic.

3. A method as defined in claim 1, wherein the characteristic of the signal is an amplitude or a power.

4. A method as defined in claim 1, wherein each of the plurality of frequencies is selected based on a frequency hop sequence.

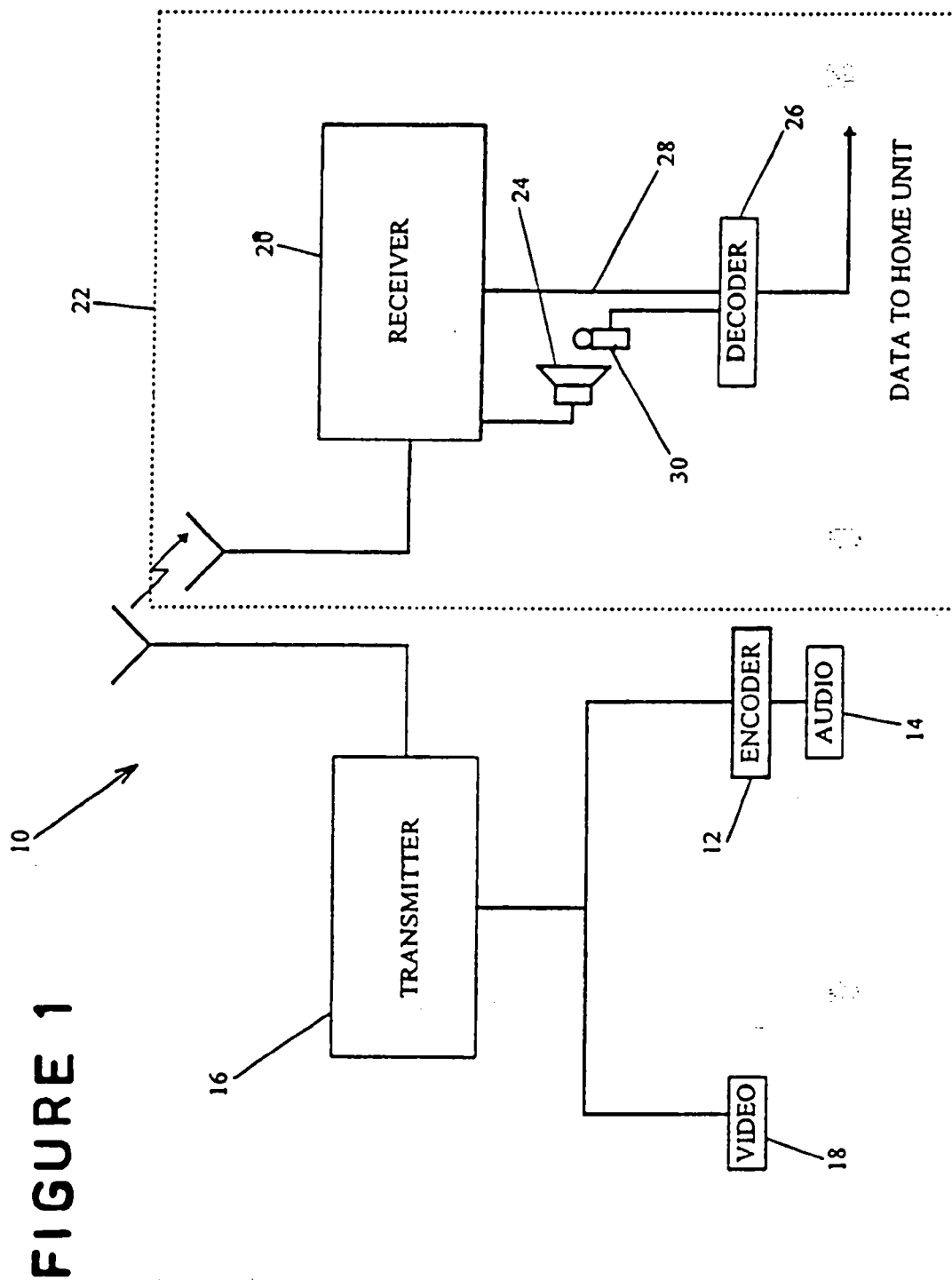5. A method as defined in claim 1, wherein the signal contains audio information.

6.    A   method   of   encoding   a   signal   substantially   as
hereinbefore   described,   with   reference   to   the
accompanying drawings.

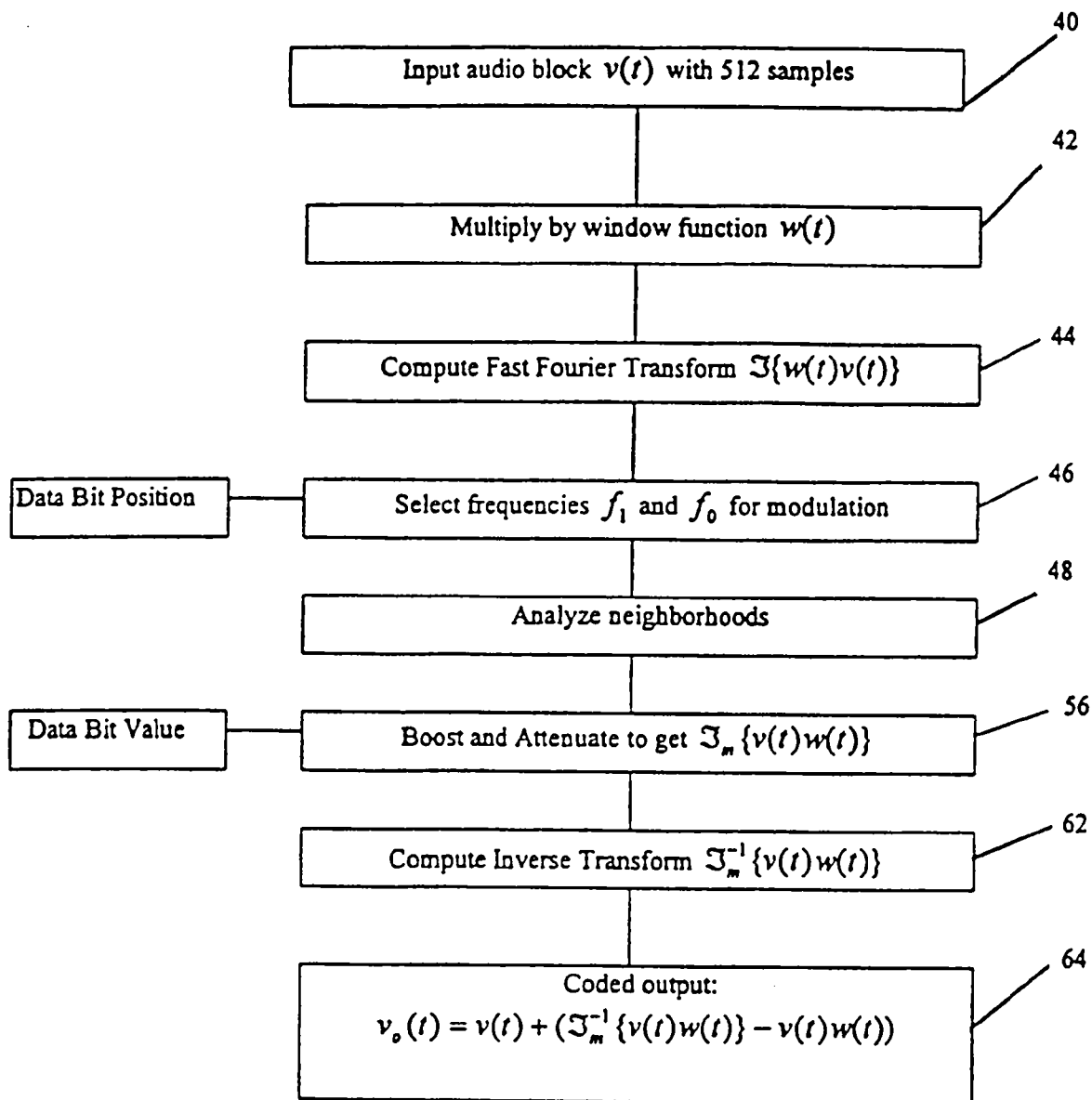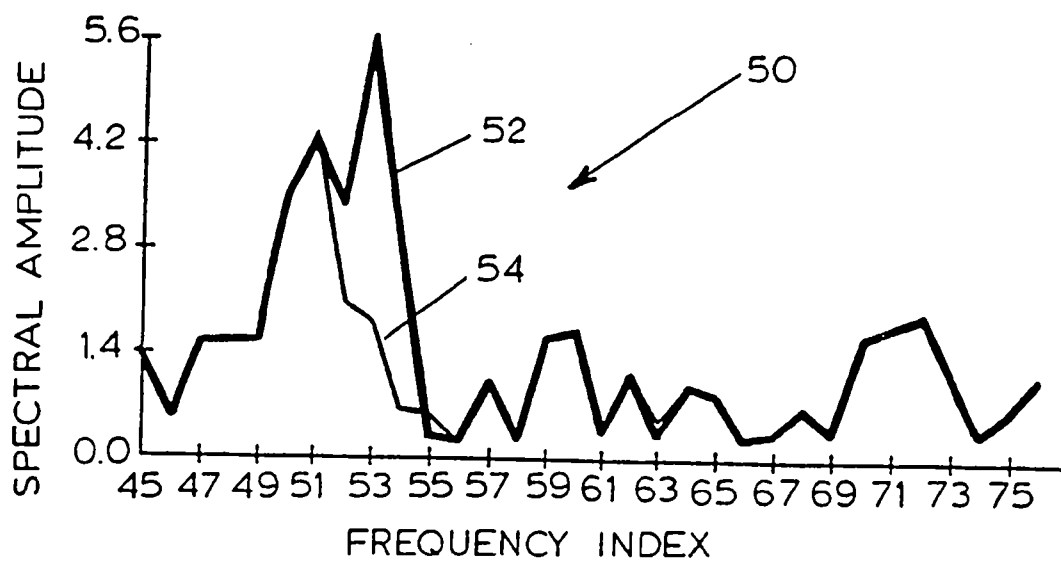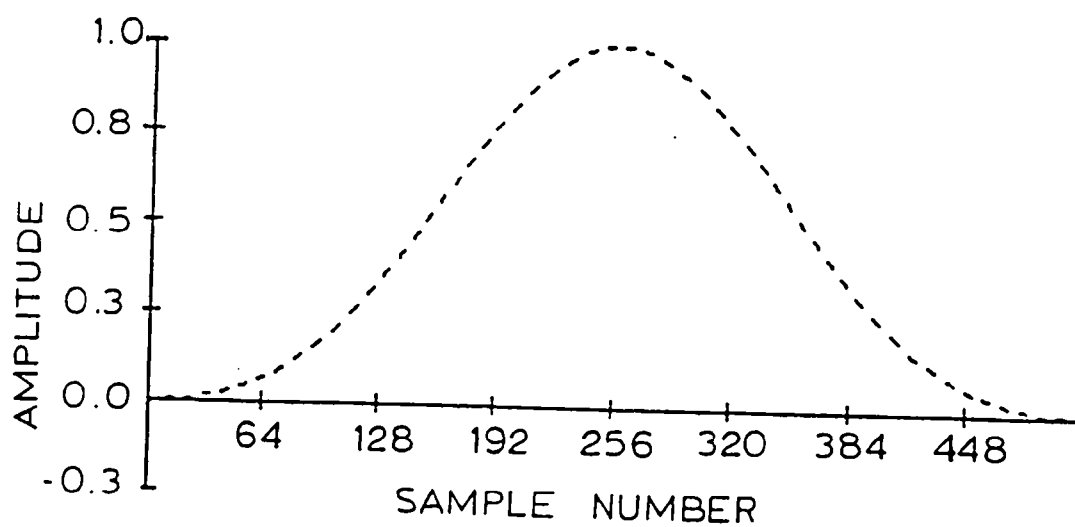# FIGURE 1

FIGURE 2

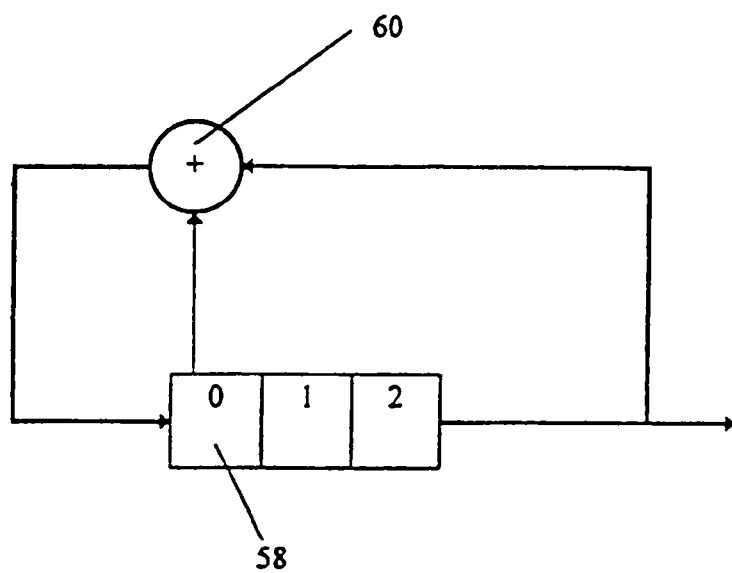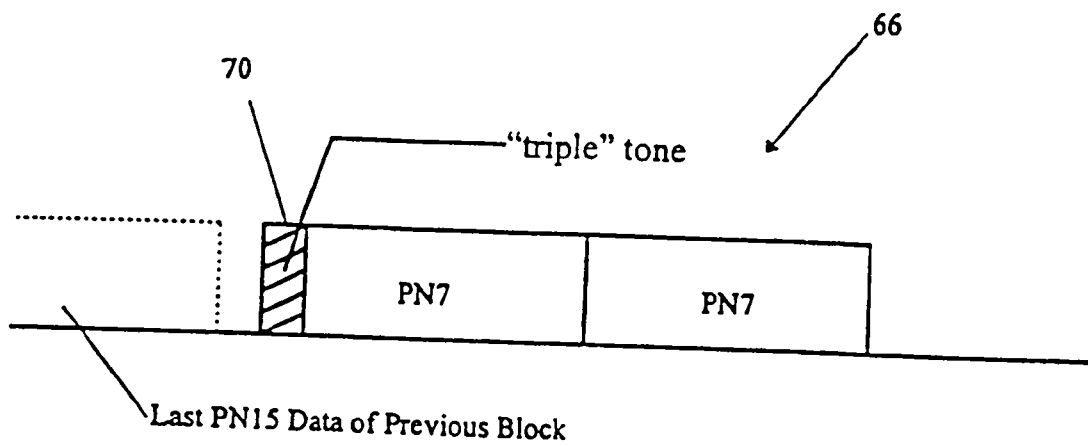**FIGURE 3**

FIGURE 4

FIGURE 5

FIGURE 6

Unencoded 512 sample block

66

72

74

PN15

PN15

DATA

DATA

synchronization

# FIGURE 7A

66

70

"triple" tone

PN7

PN7

Last PN15 Data of Previous Block

# FIGURE 7B

100

| Read first 256 samples |—102

| Perform Fast Fourier Transform |—104

| Test for triple tone |—106

| Update SIS[0], status array index p = 1 |—108

| Enter Incremental FFT mode |—110

| Read next 4 samples |—112

| Update FFT |—114

| Update SIS[p] , p = p+1 |—116

| If p = 64, reset p = 0 |—118

| Is SIS[p] .GC = 10 ? |—120

NO          YES

122
| Read Data in SIS[p].OP |

124
| Reset SIS[p] |

126
| Begin search for next message |

# FIGURE 8

Analog video output

Analog audio output

Digital Video Bitstream output

Digital Audio/Video Bitstream output

Audio output

Video D/A

Audio D/A

212

214

Mux

216

FIFO Delay D

210

Current Digital Video Encoder

218

Current Digital Audio Encoder with delay D

200

Video A/D

Audio A/D

202

204

De-Mux

208

206

Analog video input

Analog audio input

Digital Video Bitstream input

Digital Audio/Video Bitstream input

Audio input

**FIGURE 9**