

UP-SAMPLING TELEVISION AUDIO SIGNALS FOR ENCODING

RELATED APPLICATION AND TECHNICAL FIELD

[0001] This application is related to the following U.S. application, of common assignee, from which priority is claimed, and the contents of which are incorporated herein in their entirety by reference: "Digital Interpolating BTSC Stereo Encoder with SAP," U.S. Provisional Patent Application Serial No. 60/598,303, filed August 3, 2004.

[0002] This disclosure relates to processing television audio signals and, more particularly, to encoding television audio signals for transmission.

BACKGROUND

[0003] In 1984, the United States, under the auspices of the Federal Communications Commission, adopted a standard for the transmission and reception of stereo audio for television. This standard is codified in the FCC's Bulletin OET-60, and is often called the BTSC (after the committee which proposed it) system, or the MTS (Multi-channel Television Sound) system.

[0004] Prior to the BTSC system, broadcast television audio was monophonic, consisting of a single "channel" or signal of audio. Stereo audio requires the transmission of two independent audio channels, and receivers capable of detecting and accurately recovering both of these channels. In order to meet the FCC's requirement that the transmission standard be 'compatible' with existing monophonic television sets (meaning, in other words, that mono receivers be capable of reproducing an appropriate monophonic audio signal from the new type of stereo broadcast), the BTSC committee adopted an approach which had worked for the FM radio world: the stereo Left and Right audio signals are combined to form two new signals, the Sum signal and the Difference signal, which are then modulated for broadcast.

[0005] Monophonic television receivers detect and demodulate only the Sum signal, consisting of the addition of the Left and Right stereo signals. Stereo-capable receivers detect and demodulate both the Sum and the Difference signals, recombining them to extract the original stereo Left and Right signals.

[0006] For transmission, the Sum signal directly modulates the aural FM carrier just as would a monophonic audio signal. The Difference channel, however, is first modulated onto an AM subcarrier located 31.768 kHz above the aural carrier's center frequency. The nature of FM modulation is such that background noise increases by 3dB per octave, and as a result, because the new subcarrier is located further from the aural carrier's center frequency than the Sum or mono signal, additional noise is introduced into the Difference channel, and hence into the recovered stereo signal. In many circumstances, in fact, this rising noise characteristic renders the stereo signal too noisy to meet the requirements imposed by the FCC, and so the BTSC system mandates a noise reduction system in the Difference channel signal path.

[0007] This system, sometimes referred to as dbx-TV noise reduction (named after the company that developed the system) is of the companding type, comprising an encoder and decoder. The encoder adaptively filters the Difference signal prior to transmission so that its amplitude and frequency content will, upon decoding, appropriately hide ("mask") noise picked up during the transmission process. The decoder completes the process by restoring the Difference signal to its original form and in so doing ensures that noise is audibly masked by the signal content.

[0008] The BTSC system also defines a Secondary Audio Programming (SAP) signal, an additional monophonic information channel often used to carry programming in an alternative language, reading services for the blind, or other services. The SAP channel is also susceptible to added noise during broadcast, and so the dbx-TV noise reduction system is used to encode and decode the SAP channel, as well as the aforementioned stereo

signals.

[0009] The BTSC system is designed to provide audio signals of reasonably high fidelity, and as such its performance can be quantified with those parameters traditionally used to measure the quality of audio delivery systems. In particular, stereo separation is of prime importance, given that the delivery of stereo audio is the main reason for the development of the BTSC system.

[0010] As noted earlier, in order to maintain backwards compatibility with existing monophonic TV signals the BTSC system actually broadcasts a Sum signal and a Difference signal, each derived from the original Left and Right audio signals. At the receiver, the Sum and Difference signals are recombined to recover and reproduce the Left and Right signals. Accurate recovery by the receiver, and particularly recovery with good stereo separation, occurs if the various filters in both the broadcast and receiving equipment – especially those that comprise the dbx-TV encoder and decoder – comply closely with the ideal transfer functions defined in OET-60, the BTSC standards document. Inaccuracies in these filters result not just in poor stereo separation, but in degradation of other important audio parameters including frequency response, distortion, and dynamic amplitude accuracy.

[0011] The quality of the recovered SAP signal, too, is dependent on the accuracy of the various filters defined by the BTSC standard, again especially those within the dbx-TV encoder and decoder. Since the SAP signal is monophonic, stereo separation is not an issue, of course. However, other audio characteristics will degrade as a result of filter inaccuracies, just as they will with respect to the stereo Left and Right signals.

[0012] The BTSC standard defines all of the critical filters in terms of their analog filter transfer functions. While it is generally possible to design a digital filter so that either the magnitude or phase response of the digital filter matches that of an analog filter, as is

well known matching both the amplitude and phase responses simultaneously can require prohibitively complex (and expensive) filter topologies.

[0013] With no compensating filters, the audio performance of the recovered signal may suffer dramatically. Thus, heretofore, the alternative has been to use simpler digital filters and either accept the reduced audio performance or add additional compensatory networks, usually in the form of all-pass filters, to attempt to correct at least some of the inevitable phase inaccuracies. In the latter case, the compensatory networks add complexity and cost. Further, because certain critical filters used in the dbx-TV encoder and decoder are dynamic in nature with transfer functions that change from moment to moment depending on signal content, fixed compensation networks are necessarily a compromise, and do not adequately track the filter errors as the various signals change over time.

SUMMARY OF THE DISCLOSURE

[0014] In accordance with an aspect of the disclosure, a television audio signal encoder includes an up-sampler that inserts additional samples into a sum signal to increase the sample rate of the sum signal. The sum signal represents the sum of a left channel digital audio signal and a right channel digital audio signal.

[0015] In one embodiment, the television audio signal encoder further includes another up-sampler that inserts additional samples into a difference signal to increase the sample rate of the difference signal. The difference signal represents the subtraction of one of the left from the right digital audio signal or the right from the left digital audio signal from the other. The encoder may also include a matrix that sums the left channel digital audio signal and the right channel digital audio signal to produce the sum signal. The matrix may also subtract the left and right audio signals from each other to produce the difference signal. A digital filter may also be included in the encoder for receiving the up-sampled sum signal.

The increased sample rate may be selected based on characteristics of the digital filter. The encoder may also include a compressor that receives the up-sampled difference signal in which the increased sample rate is selected based on characteristics of the compressor. The television audio signals may comply with various television standards such as the Broadcast Television System Committee (BTSC) standard, the A2/Zweiton standard, the EIA – J standard, or other similar standard.

[0016] In accordance with another aspect of the disclosure, a television audio signal encoder includes an up-sampler that inserts additional samples into a difference signal to increase the sample rate of the difference signal. The difference signal represents the subtraction of one of the left channel digital audio signal from a right channel digital audio signal from the other.

[0017] In one embodiment, the television audio signal encoder may also include another up-sampler that inserts additional samples into a sum signal to increase the sample rate of the sum signal. The sum signal may represent the sum of the left channel digital audio signal and the right channel digital audio signal. The television audio signal encoder may also include a compressor that receives the up-sampled difference signal in which the increased sample rate may be selected based on characteristics of the compressor.

[0018] In accordance with another aspect of the disclosure, a television audio signal encoder includes an up-sampler that inserts additional samples into a secondary audio program signal to increase the sample rate of the secondary audio program signal.

[0019] In one embodiment, the television audio signal encoder may also include another up-sampler that may insert additional samples in a compressed version of the secondary audio program signal to increase the sample rate of the compressed secondary audio program signal.

[0020] In accordance with another aspect of the disclosure, a television audio signal

encoder includes a summer that sums a signal associated with the sum of a left channel digital audio signal and a right channel digital audio signal and a signal associated with the difference between the left and right digital audio signals. The encoder also includes an up-sampler that inserts additional samples into an output signal of the summer to increase the sample rate of the output signal.

[0021] In one embodiment of the television audio signal encoder, the signal associated with the difference between the left and right digital audio signals may be amplitude modulated.

[0022] In accordance with another aspect of the disclosure, a television audio signal encoder includes a summer that sums: a signal associated with the sum of a left channel digital audio signal and a right channel digital audio signal and associated with the difference between the left and right digital audio signals, and a signal associated with a secondary audio program channel. The encoder also includes an up-sampler that inserts additional samples into an output signal of the summer to increase the sample rate associated with the signals input into the summer.

[0023] In one embodiment of the television audio signal encoder, the signal associated with the secondary audio program channel may be frequency modulated. Also, the up-sampled output signal of the summer may be coupled to a transmitter. The television audio signal encoder may also include another up-sampler that may insert additional samples into an output signal of the first up-sampler to increase the sample rate associated with the output signal of the summer.

[0024] In accordance with another aspect of the disclosure, a computer program product residing on a computer readable medium includes instructions that when executed by a processor, cause that processor to insert additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio

program signal.

[0025] In one embodiment, the computer program product may include further instructions to insert additional samples into a sum signal to increase the sample rate of the sum signal. The sum signal may represent the sum of a television left channel digital audio signal and a television right channel digital audio signal. Still further instructions may be included to insert additional samples into a difference signal to increase the sample rate of the difference signal. The difference signal may represent the subtraction of one of a television left channel digital audio signal and a television right channel digital audio signal from the other. Additional instructions may be included to insert additional samples into an output signal of a summer that sums a signal associated with the sum a television left channel digital audio signal and a television right channel digital audio signal and a signal associated with the difference between the left and right digital audio signals. Even more instructions may be included to insert additional samples into an output signal of a summer that sums: a signal associated with the sum of a television left channel digital audio signal and a television right channel digital audio signal and associated with the difference between the left and right digital audio signals, and a signal associated with the secondary audio program channel.

[0026] In accordance with another aspect of the disclosure, a computer program product residing on a computer readable medium includes instructions that when executed by a processor, cause that processor insert additional samples into a sum signal to increase the sample rate of the sum signal. The sum signal represents the sum of a television left channel digital audio signal and a television right channel digital audio signal.

[0027] In one embodiment, the computer program product may include further instructions to insert additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal. The computer product may also include instructions to insert additional samples into a

difference signal to increase the sample rate of the difference signal. The difference signal represents the subtraction of one of the television left channel digital audio signal and the television right channel digital audio signal from the other.

[0028] In accordance with another aspect of the disclosure, a method of encoding television audio signals includes inserting additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal.

[0029] In one embodiment, the method may further include inserting additional samples into a sum signal to increase the sample rate of the sum signal. The sum signal may represent the sum of a left channel digital audio signal and a right channel digital audio signal. The method may also include inserting additional samples into a difference signal to increase the sample rate of the difference signal. The difference signal may represent the subtraction of one of a left channel digital audio signal and a right channel digital audio signal from the other. The method may also include inserting additional samples into an output signal of a summer that sums a signal associated with the sum a left channel digital audio signal and a right channel digital audio signal and a signal associated with the difference between the left and right digital audio signals. The method may also include inserting additional samples into an output signal of a summer that sums: a signal associated with the sum a left channel digital audio signal and a right channel digital audio signal and associated with the difference between the left and right digital audio signals, and a signal associated with the secondary audio program channel.

[0030] In accordance with another aspect of the disclosure, a method of encoding television audio signals includes inserting additional samples into a sum signal to increase the sample rate of the sum signal. The sum signal represents the sum of a left channel digital audio signal and a right channel digital audio signal.

[0031] In one embodiment, the method may further include inserting additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal. The method may also include inserting additional samples into a difference signal to increase the sample rate of the difference signal. The difference signal represents the subtraction of one of the left channel digital audio signal and the right channel digital audio signal from the other.

[0032] Additional advantages and aspects of the present disclosure will become readily apparent to those skilled in the art from the following detailed description, wherein embodiments of the present invention are shown and described, simply by way of illustration of the best mode contemplated for practicing the present invention. As will be described, the present disclosure is capable of other and different embodiments, and its several details are susceptible of modification in various obvious respects, all without departing from the spirit of the present disclosure. Accordingly, the drawings and description are to be regarded as illustrative in nature, and not as limitative.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram representing a television signal transmission system that is configured to comply with the BTSC television audio signal standard.

FIG. 2 is a block diagram representing a television receiver system that is configured to receive and decode BTSC television audio signals sent by the television signal transmission system shown in FIG. 1.

FIG. 3 is a block diagram representing a portion of the television signal transmission system shown in FIG. 1 that up-samples audio signals for encoding and modulating prior to transmission according to the teaching herein.

FIG. 4 is a block diagram representing a portion of the television signal transmission system shown in FIG. 1 that up-samples audio signals for digital modulating the signals prior to transmission according to the teaching herein.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0033] Referring to FIG. 1, a functional block diagram of a BTSC compatible television signal transmitter 10 is presented that processes audio and video content of a television signal for transmission to one or more reception sites. In this exemplary design, left and right audio channels are provided on respective lines (e.g., conductive wires, cables, buses, etc.) 12 and 14. A secondary audio program (SAP) signal that includes additional channel information content (e.g., alternative languages, either as audio signals or subtitles for video viewing, etc.) is provided by a line 16. A fourth line 18 provides a professional channel that is typically used by broadcast television and cable television companies. In this exemplary design, the audio signals are over these lines are digitized. Similarly, the signal contents from the professional channel may be digital, or alternatively analog. To transmit video content, video channel signals are provided by a line 20 to a transmitter 22.

[0034] Digital signals from the left, right, and SAP channels are provided to a BTSC encoder 24, which in turn prepares the audio signals for transmission. As mentioned above, sum and difference signals are produced from the left and right channel signals by BTSC encoder 24. Additionally, signals from the SAP channel are processed by BTSC encoder 24. An audio modulator stage 26 receives the processed sum signal, difference signal, and SAP signal and prepares the signals for transmission. Additionally, signals from the professional channel are prepared for transmission by audio modulator stage 26. In this exemplary design, audio modulator stage 26 combines and provides the four signals to transmitter 22 that receives the video signals provided by line 20. Along with the video

signals, the four audio signals are preferably conditioned for transmission and provided, for example, to an antenna (or an antenna system) 28, when the latter is needed for transmitting the signal. Various signal transmitting techniques known to one skilled in the art of television systems and telecommunications may be implemented by transmitter 22 and when needed antenna 28. For example, transmitter 22 may be incorporated into a cable television system (where an antenna is not needed), a broadcast television system, or other similar television audio and video content generation system.

[0035] Referring to FIG. 2, a block diagram is shown that represents a television receiver system 30 that includes an antenna 32 (or a system of antennas), if necessary, for receiving BTSC compatible broadcast signals from television transmission systems such as system 10 (shown in FIG. 1). The received signals are preferably provided to a receiver 34, the latter being capable of detecting and isolating the television transmission signals. However, in some arrangements receiver 34 may receive the BTSC compatible signals from another television signal transmission technique known to one skilled in the art of television signal broadcasting. For example, the television signals may be provided to receiver 34 over a cable television system or a satellite television network.

[0036] Upon receiving the television signals, receiver 34 conditions (e.g., amplifies, filters, frequency scales, etc.) the signals and separates the video signals and the audio signals from the transmission signals. The video content is preferably provided to a video processing system 36 that prepares the video content contained in the video signals for presentation on a screen (e.g., a cathode ray tube, etc.) associated with the television receiver system 30. Signals containing the separate audio content are preferably provided to a demodulator stage 38 that e.g., removes the modulation applied to the audio signals by television transmission system 10. The demodulated audio signals (e.g., the SAP channel signal, the professional channel signal, the sum signal, the difference signal) are preferably provided to a BTSC decoder 40 that appropriately decodes each signal. The SAP channel

signal is preferably provided a SAP channel decoder 42 and the professional channel signal is preferably provided to a professional channel decoder 44. Along with separating the SAP channel and the professional channel, a demodulated sum signal (i.e., $L + R$ signal) is preferably provided to a de-emphasis unit 46 that processes the sum signal in a substantially complementary fashion to the processing by BTSC encoder 24 (shown in FIG. 1). Upon de-emphasizing the spectral content of the sum signal, the signal is preferably provided to a matrix 48 for separating the left and right channel audio signals.

[0037] The difference signal (i.e., $L - R$) is also demodulated by demodulation stage 38 and is preferably provided to a BTSC expander 50 that is included in BTSC decoder 40. BTSC expander 50 complies with the BTSC standard to condition the difference signal. Matrix 48 receives the difference signal from BTSC expander 50 and with the sum signal, separates the right and left audio channels into independent signals (identified in the figure as "L" and "R"). By separating the signals, the individual right and left channel audio signals may be conditioned and provided to separate speakers. In this example, both the left and right audio channels are provided to an amplifier stage 52 that applies the same (or different) gains to each channel prior to providing the appropriate signals to a speaker 54 for broadcasting the left channel audio content and another speaker 56 for broadcasting the right channel audio content.

[0038] As mentioned above, the audio signals (e.g., left channel, right channel, SAP channel) are preferably digitized prior to being encoded for transmission. In such a digital implementation, digital filters such as finite impulse response (FIR) filters or infinite impulse response (IIR) filters are extensively used to encode and decode the digital audio signals prior to transmission and after reception. Accurate recovery of the L and R signals by the receiver system, particularly recovery with good stereo separation, may occur if the various filters in both the broadcast and receiving equipment – especially those that comprise the BTSC encoder and decoder – comply with the ideal transfer functions defined

in the BTSC standards document (OET-60). Inaccuracies in these filters may not result in poor stereo separation, but may cause degradation of other important audio parameters including frequency response, distortion, and dynamic amplitude accuracy.

[0039] The quality of the recovered SAP signal, too, is dependent on the accuracy of the various filters defined in the BTSC standard, again especially those within the BTSC encoder and decoder. Since the SAP signal is monophonic, stereo separation is not an issue. However, other audio characteristics may degrade as a result of filter inaccuracies, similar to the stereo L and R signals degradation.

[0040] The BTSC standard defines many (if not all) of the critical filters in terms of analog filter transfer functions. As is well known in the art, while it is generally possible to design a digital filter so that either the magnitude or phase response of the digital filter matches that of an analog filter, matching both the amplitude and phase responses simultaneously may call for prohibitively complex (and expensive) filter topologies, adding to the size, cost, and power consumption of the resulting circuitry.

[0041] To avoid the need of such complex filters, the digital audio signals may be pre-processed such that relatively simple filter implementations perform similar to the ideal transfer functions as described in the BTSC standards document (OET-60). For example, by inserting additional samples into one or more of the digital audio signals and increasing the sample rate correspondingly, the magnitude and phase of the filters over the frequency band of interest may provide a better match to the ideal transfer functions. Furthermore, by inserting additional samples into a digital audio signal and thereby increasing the sample rate correspondingly (known in the art as up-sampling), the resulting signal may provide a better match for other types of components. For example, by increasing the sample rate, a digital audio signal may better match the input characteristics of a digital-to-analog converter (DAC), a digital modulator, or other component or stage included in a television transmission or reception system.

[0042] Referring to FIG. 3, portions of an exemplary BTSC encoder 58 and an audio modulator stage 60 process digital audio signals to provide signals with appropriate sample rates to potentially achieve the ideal transfer functions described in the BTSC standards document (OET-60). Digitized audio signals from a left, right, and SAP channel are preferably provided over respective lines 62, 64, and 66 to BTSC encoder 58. Specifically, signals from the left and right audio channels are preferably provided to a matrix 68 that calculates a sum signal (e.g., $L + R$) and a difference signal (e.g., $L - R$) from the digital audio signals. Typically operations of matrix 68 are performed by utilizing a digital signal processor (DSP) or similar hardware techniques known to one skilled in the art of television audio and video signal processing. Alternatively, the operations of matrix 68 may be partially or completely implemented with software techniques known to one skilled in the art. Once produced, sum and difference signals (i.e., $L + R$ and $L - R$) are encoded for transmission.

[0043] Since filtering within BTSC encoder 58 and audio modulator stage 60 provided by digital filters, transfer functions that represent analog filter implementations are converted from the s-plane (Laplace domain) to the z-plane (discrete domain). This transformation, known as a bilinear transformation is described in U.S. Patent No. 6,037,993 that is incorporated by reference herein. To substantially match the performance of the analog implementations, the magnitude and phase responses of the digital implementations preferably approximately match the ideal responses provided by the OET-60 standard document. To provide this match, the digital audio signals are up-sampled to increase the sample rate of each digital audio signal. As mentioned above, up-sampling is performed by inserting additional samples into a digital signal. Various types of up-sampling techniques known to one skilled in the art of digital signal processing may be employed. For example, zero-value samples may be interwoven between existing sample pairs of the digital audio signals and the resulting signal passed through a low-pass filter. Interpolation techniques may also be implemented. For example, one or more

additional samples may be determined by interpolating a data point (or data points) between each existing pair of samples e.g., using polynomial or other similar estimation techniques. Once determined, the additional sample (or samples) may be inserted between the appropriate sample pair in the digital audio signal.

[0044] In this exemplary design, the sum, difference, and SAP digital audio signals are up-sampled by respective up-samplers 70, 72, and 74 to insert additional samples into each digital signal. In one scenario, each of the digital audio signals may be initially sampled at a rate of 48 KHz. To improve filter magnitude and phase matching, up-samplers 70-74 may increase the sample rate by a factor of four to 192 KHz. This increase provides improved digital filter matching to the ideal continuous time characteristics described in "Multichannel Television Sound" BTSC System Recommended Practices from the Electronic Industries Association (EIA Television Systems Bulletin No. 5, July 1985) that is incorporated by reference herein.

[0045] The up-sampled sum signal (i.e., $L + R$) is preferably provided to a pre-emphasis unit 76 that alters the magnitude of select frequency components of the sum signal with respect to other frequency components. The alteration may be in a negative sense in which the magnitudes of the select frequency components are suppressed, or the alteration may be in a positive sense in which the magnitudes of the select frequency components are enhanced.

[0046] The up-sampled difference signal (i.e., $L - R$) is preferably provided to a BTSC compressor 78 that adaptively filters the signal prior to transmission such that when decoded, the signal amplitude and frequency content suppress noise imposed during transmission. BTSC compressor 78 essentially generates the encoded difference signal by dynamically compressing, or reducing the dynamic range of the difference signal so that the encoded signal may be transmitted through a limited dynamic range transmission path, and so that a decoder receiving the encoded signal may recover substantially all the

dynamic range in the original difference signal by expanding the compressed difference signal in a complementary fashion. In some arrangements, BTSC compressor 78 is a particular form of the adaptive signal weighing system described in U.S. Patent No. 4,539,526, incorporated by reference herein, and which is known to be advantageous for transmitting a signal having a relatively large dynamic range through a transmission path or medium having a relatively narrow, frequency dependent, dynamic range. Similar to the up-sampled difference signal, the up-sampled SAP signal is preferably provided to a BTSC compressor 80. The BTSC standard rigorously defines the desired operation of pre-emphasis unit 76 and BTSC compressors 78 and 80. Specifically, the BTSC standard provides transfer functions and/or guidelines for the operation of each component included e.g., in BTSC compressor 78 and the transfer functions are described in terms of mathematical representations of idealized analog filters. These characteristics of BTSC compressors 78 and 80 (and associated components) may be used to define the up-sampling of their input signals. After processing by pre-emphasis unit 76 and BTSC compressors 78 and 80, the up-sampled signals are preferably provided to audio modulator stage 60 for combining prior to transmission.

[0047] Audio modulator stage 60 receives the processed up-sampled sum signal, difference signal, and SAP signal. Additionally, in some arrangements (not shown here), signals from the professional channel are also preferably provided to audio modulator stage 60. In this example, the up-sampled sum signal is received by audio modulator stage 60 and is summed with the up-sampled difference signal. Prior to summing, the difference signal is preferably provided to an amplitude modulator 82 that amplitude modulates the difference signal. For example, the difference signal may be modulated upon a carrier waveform with a frequency that is twice the video horizontal synchronization frequency, F_H , (e.g., $2 * 15.734 \text{ KHz}$) that is used by transmission system 10. The amplitude modulated difference signal is preferably provided to a summer 84 that sums this signal with the sum signal from BTSC encoder 58. Summer 84 also sums these two signals with

a signal that has a frequency that is equivalent the video horizontal synchronization frequency (F_H) used by the system. To identify the output signal of summer 84, the output signal is referred to as a composite signal.

[0048] Audio modulator stage 60 also receives the up-sampled SAP signal from BTSC compressor 80. Audio modulator stage 60 then up-samples (inserts additional samples) the signal to a higher sample rate to prepare the signal for frequency modulating. In some arrangements, up-sampler 86 may up-sample the signal by a factor of two to provide a sample rate of 384 KHz. This up-sampled SAP signal has the appropriate sample rate for a frequency modulator 90 that frequency modulates a carrier signal with the up-sampled SAP signal. For example, a carrier signal with a center frequency of five times F_H (5×15.734 KHz) may be frequency modulated by the up-sampled SAP signal. In some arrangements, this frequency modulation may be limited to a maximum deviation of 10 KHz. Once frequency modulated, the output signal of frequency modulator 90 is combined with a version of the composite signal produced by summer 84.

[0049] Prior to combining the composite signal (i.e., the combined up-sampled sum and difference signal) output from summer 84, the composite signal is up-sampled by an up-sampler 92. In this example, up-sampler 92 increases the sample rate of the composite signal by a factor of two. Continuing with this example, the sample rate of the composite signal is increased from 192 KHz to a sample rate of 384 KHz. By increasing the sample rate of the composite signal, the signal may be directly summed with the frequency modulated SAP signal that is output from frequency modulator 90. In this exemplary design, a summer 94 sums the up-sampled composite signal from up-sampler 92 and the frequency modulated SAP signal output by frequency modulator 90.

[0050] Although absent in this arrangement, another portion of audio modulator stage 60 also may appropriately modulate signals that are provided from the professional channel. In some arrangements these professional channel signals are combined with the

other signals (e.g., the sum signals (L+R), the difference signals (L – R), and the SAP signals) prior to transmission. Alternatively, signals from the professional channel may be transmitted separate from the other signals by using one or more methodologies known to one skilled in the art of signal transmission and reception.

[0051] Prior to sending the output signal from summer 94 (i.e., the composite signal of the sum signal (L + R), the difference signal (L – R), and the SAP signal) to transmitter 22, the signal may be further up-sampled to provide an appropriate composite signal to one or more components in the transmitter. For example, transmitter 22 may include a digital-to-analog converter (DAC) that operates with relatively few bits of resolution and at relatively high sample rates. As is well known in the art, particular signal-to-noise ratios may be achieved and maintained with fewer bits of resolution if the DAC operates with an increased sample rate. In order to improve the signal-to-noise ratio, the frequency bandwidth of the signal needs to remain substantially equivalent independent of this increase in sample rate. In this example, to provide a increased sample rate to a DAC included in transmitter 22, up-sampler 96 increases the sample rate by up-sampling the output signal of summer 94. Up-sampler 96 may increase the sample rate to e.g., 3 MHz to match the sample rate needed by the DAC. Along with up-sampling the output signal of summer 94, noise shaping techniques (e.g., sigma-delta, etc.) may be implemented in combination with (or separate from) up-sampling to improve signal-to-noise ratios.

[0052] By inserting additional samples into the output signal of summer 94, up-sampler 96 preferably provides a composite digital signal (that includes the digital audio signals) to a low resolution DAC in transmitter 22. For example, inexpensive DACs that have ten to twelve bits of resolution and operate with relatively high sample rates (e.g., larger than 192 KHz) may be used by the transmitter. Also, considerable amounts of low-resolution, high sample rate DACs (for video applications) are available on the application specific integrated circuit (ASIC) market. Additionally, by utilizing a high

sample rate DACs, post conversion filtering is improved since additional spectral content (multiple frequency octaves) is preserved. Along with up-sampling the output of summer 94 to prepare the digital audio signals for one or more DACs included in transmitter 22, additional processing may be performed for other types of components and portions of a transmitter.

[0053] Referring to FIG. 4, a portion of an exemplary audio modulator stage 98 is shown and is preferably provided the same signals from BTSC encoder 58 as audio modulator stage 60. Similar to stage 60, audio modulator stage 98 preferably includes up-samplers and summers to increase the sample rate of sum signal ($L + R$), the difference signal ($L - R$), the SAP signal, and the corresponding composite signal produced by the summers. By increasing the sample rates, digital signals that contain the appropriate magnitude and phase content are preferably provided for digital filtering and other processing functions such as digital-to-analog conversion. Up-sampling may also prepare a composite digital audio signal (e.g., that includes the sum, difference, and SAP channel signals) for a digital modulator that processes the composite signal for transmission. For example, transmitter 22 may include a digital modulator that efficiently operates if provided a frequency modulated aural signal (i.e., the portion of a television channel signal that includes the audio information) with a center frequency of 4.5 MHz.

[0054] To provide an appropriate composite signal to the digital modulator, audio modulate stage 98 preferably processes the signals from BTSC encoder 58 in a similar manner. Appropriate signals are up-sampled and modulated (similar to audio modulator stage 60). The appropriate signals are then combined into a composite signal by a summer 100 (similar to summer 94 shown in FIG. 3). The composite output signal from summer 100 is preferably provided to an up-sampler 102 that increases the sample rate of the signal by inserting additional samples into the digital signal. As mentioned above, these inserted samples may be determined by interpolating the existing signal samples or by inserting

other sample values (e.g., zero value samples), or other similar methodology known in the art. In this exemplary design, up-sampler 102 increases the sample rate to 3 MHz (similar to up-sampler 96 shown in FIG. 3). To prepare the composite digital audio signal for the digital modulator, the up-sampled output of up-sampler 102 is preferably provided to another up-sampler 104 to further increase the signal sample rate. In this particular example, up-sampler 104 increases the sample rate (e.g., to 10 MHz) as dictated by the characteristics (e.g., resolution, bandwidth, etc.) of the digital modulator included in the transmitter. After increasing the sample rate to an appropriate amount, the composite signal is preferably provided to a frequency modulator 106 that frequency modulates an aural carrier of 4.5 MHz with the composite signal.

[0055] As is known to one skilled in the art of signal processing, various interpolation schemes may be implemented in the BTSC encoders and audio modulator stages described above. For example, the audio signals may be initially sampled at 192 KHz, thereby eliminating the need for up-samplers 70, 72, and 74 (shown in FIG. 3). Alternatively, an initial sampling rate of 384 KHz further reduces the need for some of the other up-samplers. Further, sample rate adjustments may be based on many factors such as: processing bandwidth of implementation medium, available input sample rate, power consumption, performance requirements, cost and DAC availability. Additionally, while the above described up-samplers and summer are distributed in BTSC encoders and audio modulator stages, these devices or other similar devices that perform these operations may be distributed in a different manner or located entirely in either an encoder or an audio modulator stage.

[0056] In this example, the BTSC encoders and audio modulator stages are implemented with hardware components (and may be manufactured, for example, as a single integrated circuit, and chip set, a hybrid circuit, or a circuit with discrete components), however, in some arrangements one or more operational portions of the

BTSC encoders and/or audio modulator stages may be implemented in software or a combination of both. One exemplary listing of code that performs some of the operations of the BTSC encoders and audio modulator stages is presented in appendix A. The exemplary code is preferably provided in Verilog, which, in general, is a hardware description language that is used by electronic designers to describe and design chips and systems prior to fabrication. This code may be stored on and retrieved from a storage device (e.g., RAM, ROM, hard-drive, CD-ROM, etc.) and executed on one or more general purpose processors and/or specialized processors such as a dedicated DSP.

[0057] While the previous examples described using encoders and decoders compatible with the BTSC standards, up-sampling digital audio signals may be implemented into encoders and decoders that comply with other television audio standards. For example, encoders and/or decoders designed for standards associated with A2/Zweiton (currently used in parts of Europe and Asia), may incorporate up-sampling to increase sample rates of digital audio signals. Similarly, encoders and decoders implementing standards associated with the Electronics Industry Association of Japan (EIA – J) may implement up-sampling methodologies described above.

[0058] A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made. Accordingly, other implementations are within the scope of the following claims.

APPENDIX A

/*
BTSC compatible stereo encoder.

Bandwidth : DC - 15 Khz
Dynamic Range : Approx. 80dB

Note: For this version of code, 30hz head switching phase error
on hsync must be cleaned up prior to reaching this module,
or it will modulate the pilot and difference carrier.

This module btsc is the top level of the BTSC Stereo Encoder. This
design takes left and right channel audio inputs and encodes them
into the format used for multichannel television sound. The design
performs in accordance with OST Bulletin No. 60.

The design is synchronous and runs on a 49.152Mhz clock. The data rate
of the left and right inputs is 48kHz. The data rate of the composite
output is 192kHz.

The 'Q' notation used throughout this document signifies the location
of the binary point of the number. For instance, a value of Q23 means
that the binary point is located 23 places from the right. Equivalently,
it means that the integer number represents the actual number multiplied
by 2^{23} .

*****/

```
module btsc_encoder(i1Clk,
                    i1nReset,
                    i1RightLeftEn,
                    i16Right,
                    i16Left,
                    i16SAP,
                    i1HSync,
                    w18xOutEn,
                    oInterp8xOut) ;

input      i1Clk;                //49.152Mhz.
input      i1nReset;            //Asynchronous system reset
input      i1RightLeftEn;       //48Khz clock enables for left and right.
input [15:0] i16Right;          //Q15 Right input samples @48Khz
input [15:0] i16Left;           //Q15 Left input samples @48Khz
input [15:0] i16SAP;            //Q15 SAP input samples @48Khz
input      i1HSync;             //Horizontal sync pulses.
output     w18xOutEn;           //Clock enable for output samples
output [17:0] oInterp8xOut;      //Q17 3.072 MHz composite output samples

//Sum Channel Signals
wire [16:0] w17SumIn;           //Q15 Left+Right
reg [16:0] r17SumUpsampled;     //Q15 Upsampled Sum Signal
wire [20:0] w21SumFixedPreemph; //Q15 Sum Fixed Pre-emphasis signal
reg [16:0] r17SumFixedPreemphLim; //Q15 Limited Sum Channel Fixed Pre-Emphasis signal
wire [18:0] w19SumBandlimited;   //Q15 Sum Bandlimited signal
reg [15:0] r16SumBandlimitedDelay1;
reg [15:0] r16SumBandlimitedDelay2;

//Difference Channel Main Signals
wire [16:0] w17DiffIn;         //Q15 Left-Right
reg [16:0] r17DiffUpsampled;    //Q15
wire [24:0] w25DiffFixedPreemph; //Q15
wire [22:0] w23DiffVariableGain; //Q14
reg [20:0] w21DiffVariableGainLim; //Q14
wire [28:0] w29DiffSpecComp;    //Q15
reg [15:0] w16DiffSpecCompLim;  //Q15
wire [17:0] w18DiffBandlimited;  //Q15

//Difference Channel Modulation
wire [23:0] w24DiffPhase;       //Phase of difference carrier
wire [23:0] w24DiffPhaseLSBs;
wire [23:0] w24DiffError;       //LUT error of diff carrier
wire [23:0] w24DiffCarrierIntrp;
wire [23:0] w24DiffCarrier;     //Q23
wire [23:0] w24DiffModulated;   //Q22

//Diff Gain Control Signals
wire [18:0] w19DiffGainCtrlBandpass; //Q17 then Q13 - Includes a shift of 4 as part of 1/0.06249
wire [23:0] w24DiffGainCtrlSquare;   //Q13
```

```

reg      [21:0]    r22DiffGainCtrlClip;           //Q13
wire     [21:0]    w22DiffGainCtrlIntegrator;     //Q13
wire     [47:0]    w48DiffGainCtrlIntegratorPos;  //Q40
wire     [23:0]    w24DiffGainCtrlSqrt;          //Q20
reg      [20:0]    r21DiffA;                     //Q16 Gain Control Feedback signal 'a'

//Diff Spectral Control Signals
wire     [19:0]    w20DiffSpecCtrlBandpass;      //Q17
wire     [23:0]    w24DiffSpecCtrlSquare;        //Q19
reg      [19:0]    r20DiffSpecCtrlClip;          //Q19
wire     [22:0]    w23DiffSpecCtrlIntegrator;    //Q22
wire     [47:0]    w48DiffSpecCtrlIntegratorPos; //Q48
wire     [23:0]    w24DiffSpecCtrlSqrt;          //Q24 - also represents 'b' in Q19
wire     [23:0]    w24DiffB;                    //Q17 Spectral Control Feedback signal 'b'
wire     [25:0]    w26DiffBPlusC1Q19;           //Q19 Used in Spectral Compression Coefficient Calculation
wire     [20:0]    w21DiffBPlusC1;              //Q15 Used in Spectral Compression Coefficient Calculation(no
bit,always positive)
wire     [23:0]    w24DiffBTimesC1;              //Q15
wire     [23:0]    w24DiffBTimesC1Plus1;         //Q15 Used in Spectral Compression Coefficient Calculation
wire     [23:0]    w24DiffBTimesC2;              //Q15
wire     [23:0]    w24DiffBTimesC2PlusC3;        //Q15 Used in Spectral Compression Coefficient Calculation
wire     [23:0]    w24DiffBTimesNegC3;           //Q15
wire     [23:0]    w24DiffBTimesNegC3PlusNegC2;  //Q15 Used in Spectral Compression Coefficient Calculation
wire     [22:0]    w23DiffB0;                   //Q16
wire     [22:0]    w23DiffB1;                   //Q16
wire     [22:0]    w23DiffA1;                   //Q16

//SAP Channel Main Signals
reg      [15:0]    r16SAPUpsampled;              //Q15
wire     [22:0]    w23SAPFixedPreemph;          //Q15
wire     [22:0]    w23SAPVariableGain;          //Q14
reg      [20:0]    w21SAPVariableGainLim;        //Q14
wire     [28:0]    w29SAPSpecComp;              //Q15
reg      [15:0]    w16SAPSpecCompLim;           //Q15
wire     [17:0]    w18SAPBandlimited;            //Q15

//SAP Gain Control Signals
wire     [18:0]    w19SAPGainCtrlBandpass;      //Q17 then Q13 - Includes a shift of 4 as part of 1/0.06249 g
wire     [23:0]    w24SAPGainCtrlSquare;        //Q13
reg      [21:0]    r22SAPGainCtrlClip;          //Q13
wire     [21:0]    w22SAPGainCtrlIntegrator;    //Q13
wire     [47:0]    w48SAPGainCtrlIntegratorPos; //Q40
wire     [23:0]    w24SAPGainCtrlSqrt;          //Q20
reg      [20:0]    r21SAPA;                     //Q16 Gain Control Feedback signal 'a'

//SAP Spectral Control Signals
wire     [19:0]    w20SAPSpecCtrlBandpass;      //Q17
wire     [23:0]    w24SAPSpecCtrlSquare;        //Q19
reg      [19:0]    r20SAPSpecCtrlClip;          //Q19
wire     [22:0]    w23SAPSpecCtrlIntegrator;    //Q22
wire     [47:0]    w48SAPSpecCtrlIntegratorPos; //Q48
wire     [23:0]    w24SAPSpecCtrlSqrt;          //Q24 - also represents 'b' in Q19
wire     [23:0]    w24SAPB;                    //Q17 Spectral Control Feedback signal 'b'
wire     [25:0]    w26SAPBPlusC1Q19;           //Q19 Used in Spectral Compression Coefficient Calculation
wire     [20:0]    w21SAPBPlusC1;              //Q15 Used in Spectral Compression Coefficient Calculation(no
bit,always positive)
wire     [23:0]    w24SAPBTimesC1;              //Q15
wire     [23:0]    w24SAPBTimesC1Plus1;         //Q15 Used in Spectral Compression Coefficient Calculation
wire     [23:0]    w24SAPBTimesC2;              //Q15
wire     [23:0]    w24SAPBTimesC2PlusC3;        //Q15 Used in Spectral Compression Coefficient Calculation
wire     [23:0]    w24SAPBTimesNegC3;           //Q15
wire     [23:0]    w24SAPBTimesNegC3PlusNegC2;  //Q15 Used in Spectral Compression Coefficient Calculation
wire     [22:0]    w23SAPB0;                   //Q16
wire     [22:0]    w23SAPB1;                   //Q16
wire     [22:0]    w23SAPA1;                   //Q16

reg      [15:0]    composite;                    //Q14 Composite output signal(sum + pilot + modulated diff)
reg      outen;                                   //Output signifying composite data is ready
reg      r1coslut2In;

wire     w12xOutEn;                               //Enables 2x interpolator output (384 kHz)
wire     w18xOutEn;                               //Enables 8x interpolator output (3.072 MHz)
wire     [27:0]    w28FMPPhase, w28FMPPhaseDelta;
wire     [17:0]    w18Interp8xOut; // Encoder Output at Fsamp = 3.072 MHz
wire     [23:0]    p1tintpr; //Q15 Gain Control Square signal,Pilot Interpolation signal
wire     [23:0]    pilot; // Pilot signal, Diff Interpolation signal
wire     [23:0]    w24FMPPhaseLSBs;
wire     [23:0]    w24FMPPhaseErr;
wire     [23:0]    w24FMIntrp;
wire     [23:0]    w24FMCcarrier; // Unfiltered SAP

```

```

wire [17:0] w18FMCcarrierFilt; // Filtered SAP
wire [27:0] pltpphase; //Phase of pilot carrier
wire [23:0] pltpphaselsbs; //LSBs of pilot and diff phases
wire [23:0] plterr; //LUT error of pilot and diff carriers
wire [23:0] pltcarrrier; //Q23, Q23 Pilot and diff carriers
wire [3:0] multsel; //Selects the multiplier inputs
wire [1:0] divsel; //Selects the divider inputs
wire [17:0] w18Dout1, w18Dout2; // outputs of two channel 2x interpolator
wire [17:0] w18Interp8Out; // Composite output of 8x interpolator
wire [15:0] w16LeftHPF, w16RightHPF; // Input HPF outputs
wire w1HPFOutEn; // Output enable for Input HPF

reg [23:0] r24pltcarrrier; //Q23, Q23 Pilot and diff carriers
reg [23:0] r24plterr; //LUT error of pilot and diff carriers
reg [23:0] r24pltpphaselsbs; //LSBs of pilot and diff phases

reg [15:0] r16LeftHPFOut, r16RightHPFOut; // TESTESTTEST locations

//reg [17:0] r18SAPBandlimited; //Clipped SAP audio input to interpolate2

parameter pltmag = 24'd419430; //Q22 Magnitude of pilot signal(0.1)

//The following constants are used to calculate the spectral compression coefficients
parameter c1 = 24'd54294; //Q21 Constant 1
parameter clq19 = 21'd13574; //Q19 Constant 1
//parameter c2 = -24'd27420; //Q21 Constant 2
//parameter c3 = -24'd2070277; //Q21 Constant 3
//parameter c3q15 = -24'd32348; //Q15 Constant 3
parameter c2 = 24'hFF94E4; //Q21 Constant 2
parameter c3 = 24'hE068FB; //Q21 Constant 3
parameter c3q15 = 24'hFF81A4; //Q15 Constant 3
parameter nc2 = 24'd27420; //Q21 Negative of Constant 2
parameter nc2q15 = 24'd428; //Q15 Negative of Constant 2
parameter nc3 = 24'd2070277; //Q21 Negative of Constant 3

//assign w17DiffIn = {i16Left[15], i16Left} - {i16Right[15], i16Right};
//assign w17SumIn = {i16Left[15], i16Left} + {i16Right[15], i16Right};
assign w17DiffIn = {r16LeftHPFOut[15], r16LeftHPFOut} - {r16RightHPFOut[15], r16RightHPFOut};
assign w17SumIn = {r16LeftHPFOut[15], r16LeftHPFOut} + {r16RightHPFOut[15], r16RightHPFOut};
assign w24DiffPhase = {pltpphase[26:25] + 1'h1, pltpphase[24:3]};

assign w48DiffSpecCtrlIntegratorPos = w23DiffSpecCtrlIntegrator[22] ? 0 : {w23DiffSpecCtrlIntegrator[21:0],
assign w48DiffGainCtrlIntegratorPos = w22DiffGainCtrlIntegrator[21] ? 0 : {w22DiffGainCtrlIntegrator[20:0],
assign w48SAPSpecCtrlIntegratorPos = w23SAPSpecCtrlIntegrator[22] ? 0 : {w23SAPSpecCtrlIntegrator[21:0], 26'h
assign w48SAPGainCtrlIntegratorPos = w22SAPGainCtrlIntegrator[21] ? 0 : {w22SAPGainCtrlIntegrator[20:0], 27'h

assign w24DiffB = {2'd0, w24DiffSpecCtrlSqrt[23:2]}; //b(Q17), w24DiffSpecCtrlSqrt(Q19)
positive
assign w26DiffBPlusC1Q19 = {w24DiffB, 2'd0} + c1q19; //All three in Q19
assign w21DiffBPlusC1 = w26DiffBPlusC1Q19[24:4]; //Change to Q15 and remove sign bit.
assign w24DiffBTimesC1Plus1 = w24DiffBTimesC1 + 32768; //All three in Q15
assign w24DiffBTimesC2PlusC3 = w24DiffBTimesC2 + c3q15; //All three in Q15
assign w24DiffBTimesNegC3PlusNegC2 = w24DiffBTimesNegC3 + nc2q15; //All three in Q15
assign w24SAPB = {2'd0, w24SAPSpecCtrlSqrt[23:2]}; //b(Q17), w24SAPSpecCtrlSqrt(Q19) bo
positive
assign w26SAPBPlusC1Q19 = {w24SAPB, 2'd0} + c1q19; //All three in Q19
assign w21SAPBPlusC1 = w26SAPBPlusC1Q19[24:4]; //Change to Q15 and remove sign bit.
assign w24SAPBTimesC1Plus1 = w24SAPBTimesC1 + 32768; //All three in Q15
assign w24SAPBTimesC2PlusC3 = w24SAPBTimesC2 + c3q15; //All three in Q15
assign w24SAPBTimesNegC3PlusNegC2 = w24SAPBTimesNegC3 + nc2q15; //All three in Q15
assign oInterp8xOut = w18Interp8Out;

timer timer (.clk (i1Clk), //System Clock (49.152MHz)
.resetN (i1nReset), //Asynchronous Global Reset
.dregclear (w1DRegClear), //Clears D registers upon global reset
.khz48 (i1RightLeftEn), //Externally Created 48kHz pulses
.khz192 (w1FiltEn), //Internally Created 192kHz pulses
.multen (multen), //Multiplier Enable
.multsel (multsel), //Multiplier Input Select
.sqrten1 (sqrten1), //Square Root Module Enable 1
.sqrten2 (sqrten2), //Square Root Module Enable 2
.diven (diven), //Divider Enable
.divsel (divsel), //Divider Input Select
.encomposite (encomposite), //Composite Register Enable
.clearlr (clearlr), //Clear Left and Right Registers
.coslutsel (coslutsel), //Cosine

Input Select
.r1Interp2Out (w12xOutEn),

```

```

        .rlInterp8Out (w18xOutEn)) ;

SOS1VCom
SOS1      (.i1Clk          (i1Clk),
          .i1Start        (w1FiltEn),
          .i1DRegClear    (w1DRegClear),
          .i17DataIn1     (r17SumUpsampled),
          .or21DataOut1   (w21SumFixedPreemph),
          .i17DataIn2     (r17SumFixedPreemphLim),
          .or19DataOut2   (w19SumBandlimited),
          .i17DataIn3     (r17DiffUpsampled),
          .or25DataOut3   (w25DiffFixedPreemph),
          .i20DataIn4     ({w18DiffBandlimited, 2'b0}),
          //              (i21Feedback[19:0]),
          .or19DataOut4   (w19DiffGainCtrlBandpass),
          .i22DataIn5     (r22DiffGainCtrlClip),
          .or22DataOut5   (w22DiffGainCtrlIntegrator),
          //              ({w18DiffBandlimited, 2'b0}),
          .i20DataIn6     (i21Feedback[19:0]),
          .or20DataOut6   (w20DiffSpecCtrlBandpass),
          .i23DataIn7     ({r20DiffSpecCtrlClip, 3'b0}),
          .or23DataOut7   (w23DiffSpecCtrlIntegrator),
          .i22DataIn8     ({w21DiffVariableGainLim, 1'b0}),
          .or29DataOut8   (w29DiffSpecComp),
          .i18B0_08       (w23DiffB0[22:5]),
          .i18B1_08       (w23DiffB1[22:5]),
          .i18A1_08       (w23DiffA1[18:1]),
          .i16DataIn9     (w16DiffSpecCompLim),
          .or18DataOut9   (w18DiffBandlimited),
          .i16DataIn10    (r16SAPUpsampled),
          .or23DataOut10  (w23SAPFixedPreemph),
          .i20DataIn11    ({w18SAPBandlimited, 2'b0}),
          .or19DataOut11  (w19SAPGainCtrlBandpass),
          .i22DataIn12    (r22SAPGainCtrlClip),
          .or22DataOut12  (w22SAPGainCtrlIntegrator),
          .i20DataIn13    ({w18SAPBandlimited, 2'b0}),
          .or20DataOut13  (w20SAPSpecCtrlBandpass),
          .i23DataIn14    ({r20SAPSpecCtrlClip, 3'b0}),
          .or23DataOut14  (w23SAPSpecCtrlIntegrator),
          .i22DataIn15    ({w21SAPVariableGainLim, 1'b0}),
          .or29DataOut15  (w29SAPSpecComp),
          .i18B0_15       (w23SAPB0[22:5]),
          .i18B1_15       (w23SAPB1[22:5]),
          .i18A1_15       (w23SAPA1[18:1]),
          .i16DataIn16    (w16SAPSpecCompLim),
          .or18DataOut16  (w18SAPBandlimited));

mult
mult1     (.clk          (i1Clk),
          .i1Reset       (w1DRegClear),
          .en            (multen),
          .inputsel      (multsel),
          .in1a          (w24DiffCarrier),
          .in1b          ({w18DiffBandlimited[16:0], 7'h0}),
          .out1           (w24DiffModulated), //Diff Channel Modulation
          .in2a          ({w20DiffSpecCtrlBandpass, 4'b0}),
          .in2b          ({w20DiffSpecCtrlBandpass, 4'b0}),
          .out2           (w24DiffSpecCtrlSquare), //Spectral Control Squaring
          .in3a          ({w19DiffGainCtrlBandpass, 5'b0}),
          .in3b          ({w19DiffGainCtrlBandpass, 5'b0}),
          .out3           (w24DiffGainCtrlSquare), //Gain Control Squaring
          .in4a          (r24pltphaselsbs),
          .in4b          (r24plterr),
          .out4           (pltintpr), //Pilot Carrier Interpolation
          .in5a          (r24pltcarrier),
          .in5b          (pltmag),
          .out5           (pilot), //Pilot Modulation
          .in6a          (w24DiffB),
          .in6b          (c1),
          .out6           (w24DiffBTimesC1), //B0 Coefficient Multiplication
          .in7a          (w24DiffB),
          .in7b          (c2),
          .out7           (w24DiffBTimesC2), //B1 Coefficient Multiplication
          .in8a          (w24DiffB),
          .in8b          (nc3),
          .out8           (w24DiffBTimesNegC3), //A1 Coefficient Multiplication
          .in9a          (w24DiffPhaseLSBs),
          .in9b          (w24DiffError),
          .out9           (w24DiffCarrierIntpr)); //Diff Carrier Interpolation

```

```

mult
  mult2      (.clk      (i1Clk),
               .ilReset  (w1DRegClear),
               .en        (multen),
               .inputsel  (multsel),
               .in1a      (w24DiffCarrier),
               .in1b      ({w18DiffBandlimited[16:0], 7'h0}),
               .out1      (w24DiffModulated), //Diff Channel Modulation
               .in2a      ({w20SAPSpecCtrlBandpass, 4'b0}),
               .in2b      ({w20SAPSpecCtrlBandpass, 4'b0}),
               .out2      (w24SAPSpecCtrlSquare), //Spectral Control Squaring
               .in3a      ({w19SAPGainCtrlBandpass, 5'b0}),
               .in3b      ({w19SAPGainCtrlBandpass, 5'b0}),
               .out3      (w24SAPGainCtrlSquare), //Gain Control Squaring
               .in4a      (pltphaselsbs),
               .in4b      (pltterr),
               .out4      (pltintrp), //Pilot Carrier Interpolation
               .in5a      (pltcarrier),
               .in5b      (pltmag),
               .out5      (pilot), //Pilot Modulation
               .in6a      (w24SAPB),
               .in6b      (c1),
               .out6      (w24SAPBTimesC1), //B0 Coefficient Multiplication
               .in7a      (w24SAPB),
               .in7b      (c2),
               .out7      (w24SAPBTimesC2), //B1 Coefficient Multiplication
               .in8a      (w24SAPB),
               .in8b      (nc3),
               .out8      (w24SAPBTimesNegC3)); //A1 Coefficient Multiplication
               .in9a      (w24DiffPhaseLSBs),
               .in9b      (w24DiffError),
               .out9      (w24DiffCarrierIntrp)); //Diff Carrier Interpolation

sqrt
  sqrt1      (.clk      (i1Clk),
               .ilReset  (w1DRegClear),
               .en1      (sqrten1),
               .din1      (w48DiffSpecCtrlIntegratorPos),
               .dout1      (w24DiffSpecCtrlSqrt), //Spectral Control Square Root
               .en2      (sqrten2),
               .din2      (w48DiffGainCtrlIntegratorPos),
               .dout2      (w24DiffGainCtrlSqrt)); //Gain Control Square Root

sqrt
  sqrt2      (.clk      (i1Clk),
               .ilReset  (w1DRegClear),
               .en1      (sqrten1),
               .din1      (w48SAPSpecCtrlIntegratorPos),
               .dout1      (w24SAPSpecCtrlSqrt), //Spectral Control Square Root
               .en2      (sqrten2),
               .din2      (w48SAPGainCtrlIntegratorPos),
               .dout2      (w24SAPGainCtrlSqrt)); //Gain Control Square Root

div
  div1       (.clk      (i1Clk),
               .ilReset  (w1DRegClear),
               .en        (diven),
               .inputsel  (divsel),
               .num1      (w25DiffFixedPreemph[24:1]),
               .denom1     (r21DiffA),
               .dout1      (w23DiffVariableGain), //Gain Control Divide
               .num2      (w24DiffBTimesC1Plus1),
               .denom2     (w21DiffBPlusC1),
               .dout2      (w23DiffB0), //B0 Coefficient Divide
               .num3      (w24DiffBTimesC2PlusC3),
               .denom3     (w21DiffBPlusC1),
               .dout3      (w23DiffB1), //B1 Coefficient Divide
               .num4      (w24DiffBTimesNegC3PlusNegC2),
               .denom4     (w21DiffBPlusC1),
               .dout4      (w23DiffA1)); //A1 Coefficient Divide

div
  div2       (.clk      (i1Clk),
               .ilReset  (w1DRegClear),
               .en        (diven),
               .inputsel  (divsel),
               .num1      ({ {w23SAPFixedPreemph[22]}}, w23SAPFixedPreemph[22:1]), //Wasting two bit:
               .denom1     (r21SAPA),
               .dout1      (w23SAPVariableGain), //Gain Control Divide
               .num2      (w24SAPBTimesC1Plus1),
               .denom2     (w21SAPBPlusC1),

```

```

        .dout2      (w23SAPB0),          //B0 Coefficient Divide
        .num3       (w24SAPBTimesC2PlusC3),
        .denom3     (w21SAPBPlusC1),
        .dout3      (w23SAPB1),          //B1 Coefficient Divide
        .num4       (w24SAPBTimesNegC3PlusNegC2),
        .denom4     (w21SAPBPlusC1),
        .dout4      (w23SAPA1));        //A1 Coefficient Divide

coslut
coslut1      (.inputsel      (coslutsel),          //Cosine Lookup Table
              .phasein1     (pltphase[27:4]),
              .angle1lsbs   (pltphaselsbs),
              .err1         (plterr),
              .intrp1       (pltintrp),
              .cosout1      (pltcarrier),
              .phasein2     (w24DiffPhase),
              .angle2lsbs   (w24DiffPhaseLSBs),
              .err2         (w24DiffError),
              .intrp2       (w24DiffCarrierInterp),
              .cosout2      (w24DiffCarrier));

coslut
coslut2      (.inputsel      (rlcoslut2In),        //Cosine Lookup Table
              .phasein1     (w28FMPhase[27:4]),
              .angle1lsbs   (w24FMPhaseLSBs),
              .err1         (w24FMPhaseErr),
              .intrp1       (w24FMInterp),
              .cosout1      (w24FMCarrier),
              .phasein2     (),
              .angle2lsbs   (),
              .err2         (),
              .intrp2       (),
              .cosout2      ());

SAPFMMMod
SAPFMMMod    (.iclk          (i1Clk),
              .iresetN       (i1nReset),
              .ikhz384       (w12xOutEn),
              .iSAPAudio     (w18Dout1),
              .iphasein      (w28FMPhaseDelta),
              .iLSBs         (w24FMPhaseLSBs),
              .iErr          (w24FMPhaseErr),
              .oInterp       (w24FMInterp),
              .ophaseout     (w28FMPhase));

pilot
pilot1      (.clk          (i1Clk),
              .resetN       (i1nReset),
              .khz48        (i1RightLeftEn),
              .khz192       (w1FiltEn),
              .hsync        (i1HSync),
              .phaseout     (pltphase),
              .FMPhaseDelta (w28FMPhaseDelta));

Interpolate2
Interpolate2(.iclk          (i1Clk),
              .iReset       (i1nReset),
              .iNewData     (outen),
              .iFIFORead    (w12xOutEn),
              .oReadyForData(),
              .iDIN1        ({composite, 2'b00}), // {composite, 2'b00}
              .iDIN2        (w18SAPBandlimited), // {w18SAPBandlimited[15:0], 2'b00}
              .oDOUT1       (w18Dout1),
              .oDOUT2       (w18Dout2),
              .oSEL_I       (w1SEL_I));

Interpolate8
Interpolate81(.iCLK         (i1Clk),
              .iRESET       (i1nReset),
              .iND          (w12xOutEn),
              .iDIN         (w18Dout2 + w18FMCarrierFilt), // w18Dout2 + w18FMCarrierFilt
              .oDOUT        (w18Interp8Out),
              .iFIFOREAD    (w18xOutEn));

SAPFilt
SAPFilt      (.iCLK         (i1Clk),
              .iRESET       (i1nReset),
              .iND          (w12xOutEn),

```

```

        .iDIN            (w24FMCCarrier[23:6]),
        .oDOUT           (w18FMCCarrierFilt),
        .w1RFD           ();

highpass
    highpass    (.iCLK      (i1Clk),
                 .iKHZ48    (i1RightLeftEn),
                 .iRESET    (i1nReset),
                 .iLeft     (i16Left),
                 .iRight    (i16Right),
                 .oLeftHPF  (w16LeftHPF),
                 .oRightHPF (w16RightHPF),
                 .oHPFOutEn (w1HPFOutEn));

/*****
The following limiters help to protect the composite output from overmodulating
the aural carrier. The specification for the sum channel is a maximum of 25kHz
deviation of the aural carrier. Its magnitude is limited here to 1.05 of that max
(26.25kHz). There is extra margin to prevent distortion in signals that may just
exceed the max. This value can be adjusted to match the needs of the application.
If it is not acceptable to exceed the 25kHz max at all then a value somewhat less
than 1 must be used because the bandlimiting filter will potentially output a
signal whose peaks exceed those of the clipped signal that it is filtering. But,
the problem there is that you will be clipping some valid large signals that would
not overmodulate the aural carrier. Separation and harmonic distortion measurements
would suffer.

The value of 1 for the difference channel corresponds to its maximum allowable
deviation of 50kHz. This value could potentially be adjusted with the same
tradeoffs as listed for the sum channel.
*****/

//This limits the diff signal to 1(0 dB)    (0x7FFF)
always @(w29DiffSpecComp)
    if (w29DiffSpecComp[28] & (w29DiffSpecComp[27:15] != 13'h1FFF))
        w16DiffSpecCompLim <= 16'h8000;
    else if (~w29DiffSpecComp[28] & (w29DiffSpecComp[27:15] != 13'h0000))
        w16DiffSpecCompLim <= 16'h7FFF;
    else w16DiffSpecCompLim <= w29DiffSpecComp[15:0];

//This limits the SAP signal to 1(0 dB)    (0x7FFF)
always @(w29SAPSpecComp)
    if (w29SAPSpecComp[28] & (w29SAPSpecComp[27:15] != 13'h1FFF))
        w16SAPSpecCompLim <= 16'h8000;
    else if (~w29SAPSpecComp[28] & (w29SAPSpecComp[27:15] != 13'h0000))
        w16SAPSpecCompLim <= 16'h7FFF;
    else w16SAPSpecCompLim <= w29SAPSpecComp[15:0];

//Limit to 1.0625 going from Q15 to Q15    (< 0x087FF)
always @(w21SumFixedPreemph)
    if (w21SumFixedPreemph[20] && (w21SumFixedPreemph[19:11] < 9'b1111_0111_1))
        r17SumFixedPreemphLim <= 17'h1_7800;
    else if (~w21SumFixedPreemph[20] && (w21SumFixedPreemph[19:11] > 9'b0000_1000_0))
        r17SumFixedPreemphLim <= 17'h0_87FF;
    else r17SumFixedPreemphLim <= w21SumFixedPreemph[16:0];

/*****
The following minimum doesn't allow the spcsqr result to pass a 0 or -1 to
the integrator. That would cause ridiculously low (and wrong) values for b. This
forces the minimum value to be a single Q23 bit, which is a very small magnitude.
Unless the precision is increased here, the b0, b1, and a1 coefficients will never
exceed a magnitude of 32, although their dynamic range potentially allows them to.
Either reduce their dynamic range or increase the precision here to optimize this.
*****/
always @(w24DiffSpecCtrlSquare)
    if ((w24DiffSpecCtrlSquare == 24'h00_0000) || (w24DiffSpecCtrlSquare[23]))
        r20DiffSpecCtrlClip <= 20'h0_0001;
    else if (w24DiffSpecCtrlSquare > 24'h07_1EB8) //Clip to 0.89
        r20DiffSpecCtrlClip <= 20'h7_1EB8;
    else r20DiffSpecCtrlClip <= w24DiffSpecCtrlSquare[19:0];

always @(w24SAPSpecCtrlSquare)
    if ((w24SAPSpecCtrlSquare == 24'h00_0000) || (w24SAPSpecCtrlSquare[23]))
        r20SAPSpecCtrlClip <= 20'h0_0001;
    else if (w24SAPSpecCtrlSquare > 24'h07_1EB8) //Clip to 0.89
        r20SAPSpecCtrlClip <= 20'h7_1EB8;
    else r20SAPSpecCtrlClip <= w24SAPSpecCtrlSquare[19:0];

always @(w24DiffGainCtrlSquare)

```

```

if ((w24DiffGainCtrlSquare == 24'h00_0000) || (w24DiffGainCtrlSquare[23]))
    r22DiffGainCtrlClip <= 22'h00_0001;
else if (w24DiffGainCtrlSquare > 24'd1843200) // Clip to 225
    r22DiffGainCtrlClip <= 22'd1843200;
else r22DiffGainCtrlClip <= w24DiffGainCtrlSquare[21:0];

always @(w24SAPGainCtrlSquare)
    if ((w24SAPGainCtrlSquare == 24'h00_0000) || (w24SAPGainCtrlSquare[23]))
        r22SAPGainCtrlClip <= 22'h00_0001;
    else if (w24SAPGainCtrlSquare > 24'd1843200) // Clip to 225
        r22SAPGainCtrlClip <= 22'd1843200;
    else r22SAPGainCtrlClip <= w24SAPGainCtrlSquare[21:0];

always @(w23DiffVariableGain)
    if (w23DiffVariableGain[22] & (w23DiffVariableGain[21:20] != 2'b11))
        w21DiffVariableGainLim <= 21'h100000;
    else if (~w23DiffVariableGain[22] & (w23DiffVariableGain[21:20] != 2'b00))
        w21DiffVariableGainLim <= 21'h0FFFFFF;
    else w21DiffVariableGainLim <= w23DiffVariableGain[20:0];

always @(w23SAPVariableGain)
    if (w23SAPVariableGain[22] & (w23SAPVariableGain[21:20] != 2'b11))
        w21SAPVariableGainLim <= 21'h100000;
    else if (~w23SAPVariableGain[22] & (w23SAPVariableGain[21:20] != 2'b00))
        w21SAPVariableGainLim <= 21'h0FFFFFF;
    else w21SAPVariableGainLim <= w23SAPVariableGain[20:0];

always @(w24DiffGainCtrlSqrt[23:4])
    if (w24DiffGainCtrlSqrt[23:15] == 0)
        r21DiffA <= 21'h800;
    else r21DiffA <= {1'b0, w24DiffGainCtrlSqrt[23:4]};

always @(w24SAPGainCtrlSqrt[23:4])
    if (w24SAPGainCtrlSqrt[23:15] == 0)
        r21SAPA <= 21'h800;
    else r21SAPA <= {1'b0, w24SAPGainCtrlSqrt[23:4]};
/*****
The composite signal is made up of three signals: the 15.734kHz pilot signal,
the main channel(or sum channel), and the modulated difference channel. Those
three are summed in the following 'always' block. The 16-bit composite register
is in Q14 format. Therefore, its dynamic range is +/-2(in Q14.) The magnitudes of
the three signals that make up the composite signal are set such that a value of
+/-1(in Q14) corresponds to an aural carrier deviation of 50kHz. The modulated
difference channel(w24DiffModulated) is set for a maximum deviation of 50kHz or 1(in Q14.)
The sum signal (sbl) is scaled for a maximum value of 25kHz deviation or 0.5(in Q14.)
The pilot signal is scaled for a maximum deviation of 5kHz or 0.1(in Q14.)
*****/

Signal          Maximum      Composite      Eval Brd      Magnitude relative
                Deviation    Value(Q14)    Output        to L+R,L-R inputs
-----
w24DiffModulated 50kHz        +/-1          +/-3V          0dB
sbl              25kHz        +/-0.5        +/-1.5V        0dB
pilot            5kHz         +/-0.1        +/-0.3V        n/a

**For calibration of the SRD-1, a 100%EIM input signal will output a sum channel
with magnitude of +/- 1.5V on the evaluation board.
*****/

always @(posedge ilClk)
begin
    r17DiffUpsampled <= clearlr ? 0 : (ilRightLeftEn ? w17DiffIn : r17DiffUpsampled); //Upsampling
    r17SumUpsampled <= clearlr ? 0 : (ilRightLeftEn ? w17SumIn : r17SumUpsampled); //Upsampling
    r16SAPUpsampled <= clearlr ? 0 : (ilRightLeftEn ? il6SAP : r16SAPUpsampled); //Upsampling

    r16SumBandlimitedDelay1 <= encomposite ? w19SumBandlimited[17:2] : r16SumBandlimitedDelay1;
    r16SumBandlimitedDelay2 <= encomposite ? r16SumBandlimitedDelay1 : r16SumBandlimitedDelay2;

    r1coslut2In <= 1'b0;
    r24pltcarrrier <= pltcarrrier;
    r24plterr <= plterr;
    r24pltphaselsbs <= pltphaselsbs;

    composite <= encomposite ? (pilot[23:8] + w24DiffModulated[23:8] + r16SumBandlimitedDelay2) : composite;
    //composite <= encomposite ? (pilot[23:8]) : composite;
    //Added SAP at baseband temporarily just to make sure nothing gets optimized out in synthesis
    // composite <= encomposite ? (pilot[23:8] + w24DiffModulated[23:8] + r16SumBandlimitedDelay2 +
    w18SAPBandlimited[17:2]) : composite;
    outen <= encomposite;

```

```
if (w1HPFOutEn == 1) begin
    r16LeftHPFOut <= w16LeftHPF;
    r16RightHPFOut <= w16RightHPF;
end

end

endmodule
```

WHAT IS CLAIMED IS:

1. A television audio signal encoder, comprising:
a first up-sampler configured to insert additional samples into a sum signal to increase the sample rate of the sum signal, wherein the sum signal represents the sum of a left channel digital audio signal and a right channel digital audio signal.
2. The television audio signal encoder of claim 1, further comprising:
a second up-sampler configured to insert additional samples into a difference signal to increase the sample rate of the difference signal, wherein the difference signal represents the subtraction of one of the left and right digital audio signals from the other of the left and right digital audio signals.
3. The television audio signal encoder of claim 2, further comprising:
a matrix configured to sum the left channel digital audio signal and the right channel digital audio signal to produce the sum signal, and to subtract one of the left and right audio signals from the other of the left and right signals to produce the difference signal.
4. The television audio signal encoder of claim 1, further comprising:
a digital filter configured to receive the up-sampled sum signal, wherein the increased sample rate is selected based on characteristics of the digital filter.
5. The television audio signal encoder of claim 2, further comprising:
a compressor configured to receive the up-sampled difference signal, wherein the increased sample rate is selected based on characteristics of the compressor.

6. The television audio signal encoder of claim 1, wherein the television audio signals comply to the Broadcast Television System Committee (BTSC) standard.
7. The television audio signal encoder of claim 1, wherein the television audio signals comply to the A2/Zweiton standard.
8. The television audio signal encoder of claim 1, wherein the television audio signals comply to the EIA – J standard.
9. A television audio signal encoder, comprising:
 - a first up-sampler configured to insert additional samples into a difference signal to increase the sample rate of the difference signal, wherein the difference signal represents the subtraction of one of a left channel digital audio signal and a right channel digital audio signal.
10. The television audio signal encoder of claim 9, further comprising:
 - a second up-sampler configured to insert additional samples into a sum signal to increase the sample rate of the sum signal, wherein the sum signal represents the sum of the left channel digital audio signal and the right channel digital audio signal.
11. The television audio signal encoder of claim 9, further comprising:
 - a compressor configured to receive the up-sampled difference signal, wherein the increased sample rate is selected based on characteristics of the compressor.

12. A television audio signal encoder, comprising:

a first up-sampler configured to insert additional samples into a secondary audio program signal to increase the sample rate of the secondary audio program signal.

13. The television audio signal encoder of claim 12, further comprising:

a second up-sampler configured to insert additional samples in a compressed version of the secondary audio program signal to increase the sample rate of the compressed secondary audio program signal.

14. A television audio signal encoder, comprising:

a summer configured to sum a signal associated with the sum of a left channel digital audio signal and a right channel digital audio signal and a signal associated with the difference between the left and right digital audio signals; and

a first up-sampler configured to insert additional samples into an output signal of the summer to increase the sample rate of the output signal.

15. The television audio signal encoder of claim 14, wherein the signal associated with the difference between the left and right digital audio signals is amplitude modulated.

16. A television audio signal encoder, comprising:

a summer configured to sum,

a signal associated with the sum a left channel digital audio signal and a right channel digital audio signal and associated with the difference between the left and right digital audio signals, and

a signal associated with a secondary audio program channel; and

a first up-sampler configured to insert additional samples into an output

signal of the summer to increase the sample rate associated with the output signal.

17. The television audio signal encoder of claim 16, wherein the signal associated with the secondary audio program channel is frequency modulated.

18. The television audio signal encoder of claim 16, further comprising:

a frequency modulator configured to modulate a signal associated with the output of the first up-sampler.

19. The television audio signal encoder of claim 16, wherein the up-sampled output signal of the summer is provided to a transmitter.

20. The television audio signal encoder of claim 16, further comprising:

a second up-sampler configured to insert additional samples into an output signal of the first up-sampler to increase the sample rate associated with the output signal of the summer.

21. The television audio signal encoder of claim 20, further comprising:

a frequency modulator configured to modulate the output signal of the second up-sampler.

22. A computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by the processor, cause that processor to:

insert additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal.

23. The computer program product of claim 22, further comprising instructions to:

insert additional samples into a sum signal to increase the sample rate of the sum signal, wherein the sum signal represents the sum of a television left channel digital audio signal and a television right channel digital audio signal.

24. The computer program product of claim 22, further comprising instructions to:

insert additional samples into a difference signal to increase the sample rate of the difference signal, wherein the difference signal represents the subtraction of one of a television left channel digital audio signal and a television right channel digital audio signal from the other of the left and right digital audio signals.

25. The computer program product of claim 22, further comprising instructions to:

insert additional samples into an output signal of a summer that is configured to sum a signal associated with the sum a television left channel digital audio signal and a television right channel digital audio signal and a signal associated with the difference between the left and right digital audio signals.

26. The computer program product of claim 22, further comprising instructions to:

insert additional samples into an output signal of a summer, wherein the summer configured to sum,

a signal associated with the sum a television left channel digital audio signal and a television right channel digital audio signal and associated with the difference between the left and right digital audio signals, and

a signal associated with the secondary audio program channel.

27. A computer program product residing on a computer readable medium having a

plurality of instructions stored thereon which, when executed by the processor, cause that processor to:

insert additional samples into a sum signal to increase the sample rate of the sum signal, wherein the sum signal represents the sum of a television left channel digital audio signal and a television right channel digital audio signal.

28. The computer program product of claim 27, further comprising instructions to:

insert additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal.

29. The computer program product of claim 27, further comprising instructions to:

insert additional samples into a difference signal to increase the sample rate of the difference signal, wherein the difference signal represents the subtraction of one of the television left channel digital audio signal and the television right channel digital audio signal from the other of the left and right digital audio signals.

30. A method of encoding television audio signals, comprising:

inserting additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal.

31. A method of claim 30, further comprising:

inserting additional samples into a sum signal to increase the sample rate of the sum signal, wherein the sum signal represents the sum of a left channel digital audio signal and a right channel digital audio signal.

32. The method of claim 30, further comprising:

inserting additional samples into a difference signal to increase the sample

rate of the difference signal, wherein the difference signal represents the subtraction one of a left channel digital audio signal and a right channel digital audio signal from the other of the left and right digital audio signals.

33. The method of claim 30, further comprising:

inserting additional samples into an output signal of a summer configured to sum a signal associated with the sum a left channel digital audio signal and a right channel digital audio signal and a signal associated with the difference between the left and right digital audio signals.

34. The method of claim 30, further comprising:

inserting additional samples into an output signal of a summer, wherein the summer configured to sum,

a signal associated with the sum a left channel digital audio signal and a right channel digital audio signal and associated with the difference between the left and right digital audio signals, and

a signal associated with the secondary audio program channel.

35. A method of encoding television audio signals, comprising:

inserting additional samples into a sum signal to increase the sample rate of the sum signal, wherein the sum signal represents the sum of a left channel digital audio signal and a right channel digital audio signal.

36. A method of claim 35, further comprising:

inserting additional samples into a signal associated with a secondary audio program channel to increase the sample rate of the secondary audio program signal.

37. The method of claim 35, further comprising:

inserting additional samples into a difference signal to increase the sample rate of the difference signal, wherein the difference signal represents the subtraction one of the left channel digital audio signal and the right channel digital audio signal from the other of the left and right digital audio signals.