

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 Architecture for a BTSC Encoder/Decoder with SAP," U.S. Provisional Patent Application Serial No. 60/602,169, filed August 17, 2004.

[0002] This disclosure relates to processing television audio signals and, more particularly, to a configurable architecture for use with encoding and decoding television audio signals.

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 system after the Broadcast Television Systems Committee that proposed it, 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 content. Stereo audio typically requires the transmission of two independent audio channels, and receivers capable of detecting and recovering both channels. In order to meet the FCC's requirement that the new transmission standard be 'compatible' with existing monophonic television sets (i.e., that mono receivers be capable of reproducing an appropriate audio signal from the new type of stereo broadcast), the Broadcast Television Systems Committee adopted an approach similar to FM radio systems: stereo Left and Right audio signals are combined to form two new signals, a Sum signal and a Difference signal.

[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 receive both the Sum and the Difference signals, recombining the signals 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 3 decibel (dB) 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 noise reduction (after the company that developed the technique) is of the companding type, comprising an encoder and decoder. The encoder adaptively filters the Difference signal prior to transmission such that amplitude and frequency content, upon decoding, hide ("mask") noise picked up during the transmission process. The decoder completes the process by restoring the Difference signal to original form and thereby ensuring that noise is audibly masked by the signal content.

[0008] The dbx noise reduction system is also used to encode and decode Secondary Audio Programming (SAP) signals, which is defined in the BTSC standard as an additional information channel and is often used to e.g., carry programming in an alternative language, reading services for the blind, or other services.

[0009] Cost is, of course, of prime concern to television manufacturers. As a result of intense competition and consumer expectations, profit margins on consumer electronics products, especially television products, can be vanishingly small. Because the dbx

decoder is located in the television receiver, manufacturers are sensitive to the cost of the decoder, and reducing the cost of the decoder is a necessary and worthwhile goal. While the encoder is not located in a television receiver and is not as sensitive from a profit standpoint, any development which will decrease manufacturing costs of the encoder also provides a benefit.

SUMMARY OF THE DISCLOSURE

[0010] In accordance with an aspect of the disclosure, a television audio signal encoder includes a device that sums a left channel audio signal and a right channel audio signal to produce a sum signal. The matrix also subtracts one of the left and right audio signals from the other to produce a difference signal. The encoder also includes a configurable infinite impulse response digital filter that selectively uses one or more sets of filter coefficients to filter the difference signal. The set of filter coefficients is applied to the difference signal by a single multiplier in a recursive manner to prepare the difference signal for transmission.

[0011] In one embodiment, the configurable infinite impulse response digital filter may include a feedback path to apply the set of filter coefficients to the difference signal in a recursive manner. This feedback path may include a shift register to delay digital signals associated with the difference signal. The configurable infinite impulse response digital filter may multiply a signal associated with the difference signal and provide an output of this multiplication. The configurable infinite impulse response digital filter may include a selector that selects a digital input signal or selects one of the filter coefficients. In some arrangements the selector may include a multiplexer. The infinite impulse response digital filter may be configured to provide various filtering functions such as a low pass filter. The configurable infinite impulse response digital filter may also include a single adder for applying the filter coefficients to the difference signal in a recursive manner. The television audio signal may comply to the Broadcast Television System Committee (BTSC) standard, the Near Instantaneously Companded Audio Multiplex (NICAM) standard, the A2/Zweiton standard, the EIA – J standard, or other similar audio standard. The

configurable infinite impulse response digital filter may be implemented in an integrated circuit.

[0012] In accordance with another aspect of the disclosure, a television audio signal decoder includes a configurable infinite impulse response digital filter that selectively uses one or more sets of filter coefficients to filter a difference signal. The difference signal is produced by subtracting one of a left channel and a right channel audio signal from the other audio signal. The set of filter coefficients is applied to the difference signal by a single multiplier in a recursive manner to prepare the difference signal for separating the left channel and right channel audio signals. The decoder also includes a device that separates the left channel and right channel audio signals from the difference signal and a sum signal. The sum signal includes the sum the left channel audio signal and the right channel audio signal.

[0013] In one embodiment, the configurable infinite impulse response digital filter may include a feedback path to apply the set of filter coefficients to the difference signal in a recursive manner. This feedback path may include a shift register to delay digital signals associated with the difference signal. The configurable infinite impulse response digital filter may multiply a signal associated with the difference signal and provide an output of this multiplication. The configurable infinite impulse response digital filter may include a selector that selects a digital input signal or selects one of the filter coefficients. In some arrangements the selector may include a multiplexer. The infinite impulse response digital filter may be configured to provide various filtering functions such as a low pass filter. The configurable infinite impulse response digital filter may also include a single adder for applying the filter coefficients to the difference signal in a recursive manner. The television audio signal may comply to the Broadcast Television System Committee (BTSC) standard, the Near Instantaneously Companded Audio Multiplex (NICAM) standard, the A2/Zweiton standard, the EIA – J standard, or other similar audio standard. The configurable infinite impulse response digital filter may be implemented in an integrated circuit.

[0014] 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 portion of a BTSC encoder included in the television signal transmission system shown in FIG. 1.

FIG. 3 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. 4 is a block diagram representing a portion of a BTSC decoder included in the television receiver system shown in FIG. 3.

FIG. 5 is a diagrammatic view of a configurable infinite impulse response filter for performing operations of the encoder and decoder shown in FIG. 2 and FIG. 4.

FIG. 6 is a graphical representation of a transfer function of a second-order infinite impulse response filter that may be implemented by the infinite impulse response filter shown in FIG. 5.

FIG. 7 is a block diagram of a portion of a BTSC encoder that highlights operations that may be performed by the configurable infinite impulse response filter shown in FIG. 5.

FIG. 8 is a block diagram of a portion of a BTSC decoder that highlights operations that may be performed by the configurable infinite impulse response filter shown in FIG. 5.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0015] Referring to FIG. 1, a functional block diagram of a BTSC compatible television signal transmitter 10 includes five lines (e.g., conductive wires, cables, etc.) that provide signals for transmission. In particular, left and right audio channels are provided on respective lines 12 and 14. An SAP signal is provided by line 16 in which the signal has content to provide additional channel information (e.g., alternative languages, etc.). A fourth line 18 provides a professional channel that is typically used by broadcast television and cable television companies. Video signals are provided by a line 20 to a transmitter 22. The left, right, and SAP channels are provided to a BTSC encoder 24 that prepares the audio signals for transmission. Specifically, the left and right audio channels are provided to a matrix 26 that calculates a sum signal (e.g., $L + R$) and a difference signal (e.g., $L - R$) from the audio signals. Typically operations of matrix 26 are performed by utilizing a digital signal processor (DSP) or similar hardware or software – based techniques known to one skilled in the art of television audio and video signal processing. Once produced, sum and difference signals (i.e., $L + R$ and $L - R$) are encoded for transmission. In particular, the sum signal (i.e., $L + R$) is provided to a pre-emphasis unit 28 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 magnitude of the select frequency components are suppressed, or the alteration may be in a positive sense in which the magnitude of the select frequency components are enhanced.

[0016] The difference signal (i.e., $L - R$) is provided to a BTSC compressor 30 that adaptively filters the signal prior to transmission such that when decoded, the signal amplitude and frequency content suppress noise imposed during transmission. Similar to the difference signal, the SAP signal is provided to a BTSC compressor 32. An audio

modulator stage 34 receives the processed sum signal, difference signal, and SAP signal. Additionally, signals from the professional channel are provided to audio modulator stage 34. The four signals are modulated by audio modulator stage 34 and provided to transmitter 22. Along with the video signals provided by the video channel, the four audio signals are conditioned for transmission and provided to an antenna 36 (or an antenna system). Various signal transmitting techniques known to one skilled in the art of television systems and telecommunications may be implemented by transmitter 22 and antenna 36. For example, transmitter 22 may be incorporated into a cable television system, a broadcast television system, or other similar television system.

[0017] Referring to FIG. 2, a block diagram representing operations performed by a portion of BTSC compressor 30 is shown. In general, the difference channel (i.e., $L - R$) processing performed by BTSC compressor 30 is considerably more complex than the sum channel (i.e., $L + R$) processing by pre-emphasis unit 28. The additional processing provided by the difference channel processing BTSC compressor 30, in combination with complementary processing provided by a decoder (not shown) receiving a BTSC signal, maintains the signal-to-noise ratio of the difference channel at acceptable levels even in the presence of the higher noise floor associated with the transmission and reception of the difference channel. BTSC compressor 30 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 30 implements a particular form of an 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 having a relatively narrow, frequency dependent, dynamic range.

[0018] The BTSC standard rigorously defines the desired operation of BTSC encoder 24 and BTSC compressors 30 and 32. Specifically, the BTSC standard provides transfer functions and/or guidelines for the operation of each component included e.g., in BTSC compressor 30 and the transfer functions are described in terms of mathematical representations of idealized analog filters. Upon receiving the difference signal (i.e., $L - R$) from matrix 26, the signal may be provided to an interpolation and fixed pre-emphasis stage 38. In some digital BTSC encoders, the interpolation is set for twice the sample rate and the interpolation may be accomplished by linear interpolation, parabolic interpolation, or a filter (e.g., a finite impulse response (FIR) filter, an infinite impulse response (IIR) filter, etc.) of n -th order. The interpolation and fixed pre-emphasis stage 38 also provides pre-emphasis. After interpolation and pre-emphasis, the difference signal is provided to a divider 40 that divides the difference signal by a quantity determined from the difference signal and described in detail below.

[0019] The output of divider 40 is provided to a spectral compression unit 42 that performs emphasis filtering of the difference signal. In general, spectral compression unit 42 "compresses", or reduces the dynamic range, of the difference signal by amplifying signals having relatively low amplitudes and attenuating signals having relatively large amplitudes. In some arrangements spectral compression unit 42 produces an internal control signal from the difference signal that controls the pre-emphasis/de-emphasis that is applied. Typically, spectral compression unit 42 dynamically compresses high frequency portions of the difference signal by an amount determined by the energy level in the high frequency portions of the encoded difference signal. Spectral compression unit 42 thus provides additional signal compression toward the higher frequency portions of the difference signal. This is done because the difference signal tends to be noisier in the higher frequency portion of the spectrum. When the encoded difference signal is decoded with a spectral expander in a decoder, respectively in a complementary manner to the spectral compression unit of the encoder, the signal - to - noise ratio of the $L - R$ signal is substantially preserved.

[0020] Once processed by spectral compression unit 42, the difference signal is provided to an over-modulation protection unit 44 and band-limiting unit 46. Similar to the other components, the BTSC standard provides suggested guidelines for the operation of over-modulation protection unit 44 and band-limiting unit 46. Generally, band-limiting unit 46 and a portion of over-modulation protection unit 44 may be implemented as low pass filters. Over-modulation protection unit 44 also performs as a threshold device that limits the amplitude of the encoded difference signal to full modulation, where full modulation is the maximum permissible deviation level for modulating an audio subcarrier in a television signal.

[0021] Two feedback paths 48 and 50 are included in BTSC compressor 30. Feedback path 50 includes a spectral control bandpass filter 52 that typically has a relatively narrow pass band that is weighted towards higher audio frequencies to provide a control signal for spectral compression unit 42. To condition the control signal produced by spectral control bandpass filter 52, feedback path 50 also includes a multiplier 54 (configured to square the signal provided by spectral control bandpass filter 52), an integrator 56, and a square root device that provides the control signal to spectral compression unit 42. Feedback path 48 also includes a bandpass filter (i.e., gain control bandpass filter 60) that filters the output signal from band-limiting unit 46 to set the gain applied to the output signal of interpolation and fixed pre-emphasis stage 38 via divider 40. Similar to feedback path 50, feedback path 48 also includes a multiplier 62, an integrator 64, and a square root device 66 to condition the signal that is provided to divider 40.

[0022] Referring to FIG. 3, a block diagram is shown that represents a television receiver system 68 that includes an antenna 70 (or a system of antennas) to receive BTSC compatible broadcast signals from television transmission system 10 (shown in FIG. 1). The signals received by antenna 70 are provided to a receiver 72 that is capable of detecting and isolating the television transmission signals. However, in some arrangements receiver 72 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 72 over a cable television system or a satellite television network.

[0023] Upon receiving the television signals, receiver 72 conditions (e.g., amplifies, filters, frequency scales, etc.) the signals and separates the video signals and the audio signals out of the transmission signals. The video content is provided to a video processing system 74 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 68. Signals containing the separate audio content are provided to a demodulator stage 76 that e.g., removes the modulation applied to the audio signals at television transmission system 10. The demodulated audio signals (e.g., the SAP channel, the professional channel, the sum signal, the difference signal) are provided to a BTSC decoder 78 that appropriately decodes each signal. The SAP channel is provided a SAP channel decoder 80 and the professional channel is provided to a professional channel decoder 82. After separating the SAP channel and the professional channel, a demodulated sum signal (i.e., $L + R$ signal) is provided to a de-emphasis unit 84 that processes the sum signal in a substantially complementary fashion in comparison to pre-emphasis unit 28 (shown in FIG. 1). Upon de-emphasizing the spectral content of the sum signal, the signal is provided to a matrix 88 for separating the left and right channel audio signals.

[0024] The difference signal (i.e., $L - R$) is also demodulated by demodulation stage 76 and is provided to a BTSC expander 86 included in BTSC decoder 78. BTSC expander 86 complies with the BTSC standard, and as described in detail below, conditions the difference signal. Matrix 88 receives the difference signal from BTSC expander 86 and with the sum signal, separates the right and left audio channels into independent signals (identified in FIG. 3 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 90 that applies the same (or different) gains to each channel prior to providing the respective

signals to a speaker 92 for broadcasting the left channel audio content and another speaker 94 for broadcasting the right channel audio content.

[0025] Referring to FIG. 4, a block diagram identifies some of the operations performed by BTSC expander 86 to condition the difference signal. In general, BTSC expander 86 performs operations that are complementary to the operations performed by BTSC compressor 32 (shown in FIG. 2). In particular, the compressed difference signal is provided to a signal path 96 for un-compressing the signal, and to two paths 98 and 100 that produce a respective control and gain signal to assist the processing of the difference signal. To initiate the processing, the compressed difference signal is provided to a band-limiting unit 102 that filters the compressed difference signal. The band-limiting unit 102 provides a signal to path 98 to produce a control signal and to path 100 to produce a gain signal. Path 100 includes a gain control bandpass filter 104, a multiplier 106 (that squares the output of the gain control bandpass filter), an integrator 108, and a square root device 110. Signal path 98 also receives the signal from band-limiting unit 102 and processes the signal with a spectral control bandpass filter 112, a squaring device 114, an integrator 116, and a square root device 118. Path 98 then provides a control signal to a spectral expansion unit 120 that performs an operation that is complementary to the operation performed by spectral compression unit 42 shown in FIG. 2. The gain signal produced by path 100 is provided to a multiplier 122 that receives an output signal from spectral expansion unit 120. Multiplier 122 provides the spectrally expanded difference signal to a fixed de-emphasis unit 124 that filters the signal in a complementary manner in comparison to filtering performed by BTSC compressor 30. In general, the term "de-emphasis" means the alteration of the select frequency components of the decoded signal in either a negative or positive sense in a complementary manner in which the original signal is encoded.

[0026] Both BTSC encoder 24 and BTSC decoder 78 include multiple filters that adjust the amplitude of audio signals as a function of frequency. In some prior art television transmission systems and reception systems, each of the filters are implemented

with discrete analog components. However, with advancements in digital signal processing, some BTSC encoders and BTSC decoders may be implemented in the digital domain with one or more integrated circuits (ICs). Furthermore, multiple digital BTSC encoders and/or decoders may be implemented on a single IC. For example, encoders and decoders may be incorporated into a single IC as a portion of a very large scale integration (VLSI) system.

[0027] A significant portion of the cost of an IC is directly proportional to the physical size of the chip, particularly the size of its 'die', or the active, non-packaging part of the chip. In some arrangements filtering operations performed in digital BTSC encoders and decoders may be executed using general purpose digital signal processors that are designed to execute a range of DSP functions and operations. These DSP engines tend to have relatively large die areas, and are thereby costly to use for implementing BTSC encoders and decoders. Additionally the DSP may be dedicated to executing other functions and operations. By sharing this resource, the processing performed by the DSP may overload and interfere with the processing of the BTSC encoder and decoder functions and operations.

[0028] In some arrangements, BTSC encoders and decoders may incorporate groups of basic components to reduce cost. For example, groups of multipliers, adders, and multiplexers may be incorporated to produce the BTSC encoder and decoder functions. However, while the groups of nearly identical components may be easily fabricated, the components represent significant die area and add to the total cost of the IC. Thus, a need exists to reduce the number of duplicated circuit components used to implement a digital BTSC encoder and/or decoder.

[0029] Referring to FIG. 5, a block diagram of a configurable infinite impulse response (IIR) filter 126 is shown that is capable of performing multiple types of operations for a digital BTSC encoder and/or decoder. In particular, configurable IIR filter 126 includes a digital architecture that is capable of performing various filtering, multiplication, and delay operations. Regarding filtering operations, by providing

selectable filtering coefficients, configurable IIR filter 126 may be configured for various types of filters and different filtering operations. For example, filtering coefficients may be selected to provide a low pass filter, a high pass filter, a band pass filter, or other type of filters known to one skilled in the art of filter design. Thus, one or a relatively small number of implementations of configurable IIR filter 126 may be used to provide most or all of the filtering needs of a BTSC encoder or a BTSC decoder. By reducing the number of decoder and encoder filters, the implementation area of an IC chip is reduced along with the production cost of the BTSC encoders and decoders. Other embodiments of configurable IIR filter 126 are described in "Configurable Filter for Processing Television Audio Signals," U.S. Patent Application Serial No. 11/089,385, filed March 24, 2005, which is incorporated by reference herein.

[0030] Along with using components for selecting filter coefficients, by using a recursive digital architecture, the number of components may be further reduced. In this exemplary design, configurable IIR filter 126 includes a feedback path 128 that passes digital signals from the output portion of the architecture to components for further processing. By passing processed digital signals through feedback path 128, various types of recursive processing may be provided by configurable IIR filter 126. For example, higher order filters (e.g., second-order or higher) may be realized by passing signals through feedback path 128.

[0031] In this implementation, various digital input signals are provided on inputs of a multiplexer 130 that functions as a selector. For example, signals may be input from various portions of a compressor such as BTSC compressor 30 (shown in FIG. 2). Interpolation and fixed pre-emphasis stage 38, gain control bandpass filter 60, and spectral control bandpass filter 52 may provide digital signals to multiplexer 130. Dependent upon appropriate scheduling, multiplexer 130 selects one input for processing an appropriate input signal. The selected signal is provided to an input register 132 and then to a multiplexer 134 at an appropriate time. Multiplexer 134 provides a single adder 136 with data from either input register 132 (e.g., new input data) or previously computed product

data from a single multiplier 138 (via a product register 140). Adder 136 also receives input data from a multiplexer 142 that is either previously accumulated data from a sum register 144 (that is preferably connected the output of adder 136) or product data from multiplier 138 (preferably provided through product register 140 and a register 146).

[0032] To provide the digital input signals for processing and recursive processing for previously processed signals, feedback path 128 provides the output of adder 136 to multiplier 138. In particular, the output of adder 136 is provided a multiplexer 148 that provides an output signal to a shift register 150. Either the output signal of adder 136 or a delayed version of a signal (output from shift register 150) is provided to the input of shift register 150. By including shift register 150 in feedback path 128, a time delay may be applied to a digital signal prior to processing by multiplier 138. For filtering applications, time delays introduced by shift register 150 may be used for implementing higher order filters (e.g., a second-order filter).

[0033] The output of shift register 150 is provided (as mentioned above) to the input of multiplexer 148. Feedback path 128 provides data to multiplier 138 through a multiplexer 152. In particular, digital signals may be feedback directly over conductor 154 from the output of adder 136. Signals may also be feedback as provided by the output of shift register 150 or a delayed version of the output of shift register 150 (via a register 156). External multiplicands may also be provided to the inputs of multiplexer 158. As shown in the figure, external data may be provided to one or more input lines 158 of multiplexer 152. A register 160 is provided an output signal from multiplexer 152 in preparation for multiplication by multiplier 138.

[0034] Data such as filter coefficients (with fixed or variable values) may be provided to configurable IIR filter 126 by a multiplexer 162. In particular, data representing filter coefficients may be provided to multiplexer 162 from input lines 164. External multiplicands may also be provided by input lines 164. Along with being supplied externally, coefficient or multiplicands may be provided to multiplexer 162 by a register 166. Similar to multiplexer 152, multiplexer 162 provides data to a register 168 in

preparation for providing the data to multiplier 138.

[0035] Since feedback path 128 is included in configurable IIR filter 126, a single multiplier (i.e., multiplier 138) may be incorporated to provide the multiplication function within for implementing the filter. By implementing this single multiplier scheme, integrated circuit real estate may be conserved and used to provide other functionality. For example, a series of output registers may be implemented to directly provide the output of product register 140 to external devices and components. Additionally, due to feedback path 128, a single adder (i.e., adder 136) provides the addition functionality to implement various types of IIR filters. Again, by using a single component, in this case adder 136, additional chip real estate is conserved for other components. For example, a series of output registers 172 may be implemented for directing the output of adder 136 (via sum register 144) to external components or modules that are located on the same integrated circuit or on an external device.

[0036] In addition to providing a multiplication function (with outputs provided by output registers 170) and filtering functions (with outputs provided by output registers 172), configurable IIR filter 126 may also provide a time delay function. For example, the output of shift register 150 and/or the output of register 156 may be used to provide time-delayed version of one or more digital signals provided to the registers.

[0037] To allow configurable IIR filter 126 to perform multiple types of filtering operations, the multiplexer 130 controls which input provides an input signal. Referring briefly to FIG 2, some of the inputs to multiplexer 130 may be connected to provide input signals for each of the filtering operations performed within BTSC compressor 30. For example, the input to gain control bandpass filter 60 may be connected an input of multiplexer 130. Similarly, the input to spectral control bandpass filter 52 may be connected to another input of multiplexer 130. Then, multiplexer 130 may control which particular filtering operation is performed by configurable IIR filter 126. For example, during one time period, the appropriate input may be selected and configurable IIR filter 126 may be configured to provide the filtering function of gain control bandpass filter 60.

Then, at another time period, multiplexer 130 may be used to select another input to perform a different filtering operation. Along with selecting the other input, configurable IIR filter 126 may be correspondingly configured to provide a different type of filtering function, such as the filtering provided by spectral control bandpass filter 52.

[0038] In order to perform multiple filtering operations e.g., for a BTSC compressor or a BTSC expander, configurable IIR filter 126 operates at a clock speed substantially faster than the other portions of the digital compressor or expander. By operating at a faster clock speed, configurable IIR filter 126 may perform one type of filtering without causing other operations of the digital compressor or expander to be delayed. For example, by operating configurable IIR filter 126 at a substantially fast clock speed, the architecture may first be configured to perform filtering for gain control bandpass filter 60 without substantially delaying the execution of the next filter configuration (e.g., filter operations for spectral control bandpass filter 52).

[0039] In one arrangement, configurable IIR filter 126 may be implemented as a second-order IIR filter. Referring to FIG. 6, a z-domain signal flow diagram 174 is presented for a typical second-order IIR filter. An input node 176 receives an input signal identified as $X(z)$. The input signal is provided to an adder 178 that adds the signal to a processed signal that is described below. The output of adder 178 is provided to a gain stage 180 that applies a filter coefficient a_0 to the input signal. In some applications the filter coefficient a_0 has a unity value. Similarly, a filter coefficient b_0 is applied to the input signal at gain stage 182. At a delay stage 184, a time delay (i.e., represented in the z-domain as z^{-1}) is applied as the input signal enters the first-order portion of the filter and filter coefficients a_1 and b_1 are applied at respective gain stages 186 and 188. A second delay (i.e., z^{-1}) is applied at delay stage 190 for producing the second-order portion of filter 174 and filter coefficients a_2 and b_2 are applied at respective gain stages 192 and 194. Respective adders 196, 198, and 200 add signals from the gain stages and the filtered signal is provided to an output node 202 such that output signal $Y(z)$ may be determined from the transfer function $H(z)$ of the second-order filter 174, as described in the following

Equation (1) :

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

[0040] Each of the coefficients (i.e., b_0 , a_0 , b_1 , a_1 , b_2 , and a_2) included in the transfer function may be assigned particular values to produce a desired type of filter. For example, particular values may be assigned to the coefficients to produce a low-pass filter, a high-pass filter, or a band-pass filter, etc. Thus, by providing the appropriate values for each coefficient, the type and characteristics (e.g., pass band, roll-off, etc) of the second-order filter may be configured and re-configured into another type of filter (dependent upon the application) with a different set of coefficients. While this example describes a second-order filter, in other arrangements an n^{th} -order filter may be implemented. For example, higher order (e.g. third-order, fourth-order, etc.) filters or lower order (e.g., first-order filters) may be implemented. Furthermore, for some applications, the recursive digital architecture of configurable IIR filter 126 may be cascaded to produce n^{th} -order filters.

[0041] Referring back to FIG. 5, along with using multiplexer 130 to select a particular input for configurable IIR filter 126, the coefficients used by the filter are selected to implement different types of filters and to provide particular filter characteristics. For example, coefficients may be selected to implement a low-pass filter, a high-pass filter, a band-pass filter, or other similar type of filter used to encode or decode BTSC audio signals. Due to the recursive processing provided by feedback path 128, different coefficients or sets of coefficients may be selected by multiplexer 152 and/or multiplexer 162. By selecting different coefficients for different recursive iterations, various filters may be implemented. For example, multiplexer 162 may be controlled to select a filter coefficient (e.g., a_0 , b_0 , a_1 , b_1 , etc.) associated with a second-order filter. Then, for the next iteration, multiplexer 162 may select another filter coefficient. By providing these selectable coefficients values, configurable IIR filter 126 may be configured to provide filters for both encoding and decoding operations. Upon completing the filtering for one

application (e.g., gain control bandpass filter 60) for in a recursive manner, multiplexer 130 may then be placed in a position to provide input signals for another application (e.g., spectral control bandpass filter 52). By selecting this input, new filter coefficients may be selected by multiplexer 162 and/or multiplexer 152 to provide the particular filter type and filter characteristics needed to perform the filtering for this next application.

[0042] In this example illustrated in FIG. 6, configurable IIR filter 126 is configured for a second-order filter, however, some encoding and/or decoding filtering applications may call for a higher order filter. To provide higher order filters, additional recursive iterations may be performed through feedback path 128. By using the feedback path, signals may pass through the IIR filter multiple times using the same (or different) filter coefficients. Thus, filtering operations may be performed with a single multiplier (i.e., multiplier 138) and a single adder (i.e., adder 136) for various types of filters and various order filter implementations. To illustrate the iterations that are performed by configurable IIR filter 126, numerical indicators (i.e., 1, 2, 3, 4, 5) are shown to represent the individual clock cycles in which each function is executed. In this illustration, these functions execute in a sequence of: 1, 2, 3, 4, 5. Thus, five clock cycles are needed to compute an output for the second order filter. Additionally, this sequence of executed functions may be repeated in a periodic manner (e.g., 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, etc.).

[0043] Various techniques and components known to one skilled in the art of electronics and filter design may be used to implement the multiplexers (e.g., multiplexer 130, 152, 162, etc.). For example, multiplexer 130 may be implemented by one or more multiplexers to select among the inputs. Besides multiplexers, or other types digital selection devices may be implemented to select appropriate filter coefficients. Various coefficient values may be used to configure IIR filter such as IIR filter 174. For example, coefficients described in U.S. Patent 5,796,842 to Hanna, which is herein incorporated by reference, may be used by configurable IIR filter 126. In some arrangements, the filter coefficients are stored in a memory (not shown) associated with the BTSC encoder or decoder and are retrieved by the appropriate multiplexers at appropriate times. For

example, the coefficients may be stored in a memory chip (e.g., random access memory (RAM), read – only memory (ROM), etc.) or another type of storage device (e.g., a hard-drive, CD-ROM, etc.) associated with the BTSC encoder or decoder. The coefficients may also be stored in various software structures such as a look-up table, or other similar structure.

[0044] Configurable IIR filter 126 also includes a single adder 136 along with the single multiplier 138. Various techniques and/or components known to one skilled in the art of electronic circuit design and digital design may be used to implement adder 136 and the multiplier 138 included in configurable IIR filter 126. For example, logic gates such as one or more “AND” gates may be implemented as each of the multipliers. To introduce time delays, various techniques and/or components known to one skilled in the art of electronic circuit design and digital design may be implemented to produce shift register 150 (shown in FIG. 5) and provide delays by storing and holding the digitized input signal values for an appropriate number of clock cycles.

[0045] In this example, configurable IIR filter 126 is implemented with hardware components, however, in some arrangements one or more operational portions of the architecture may be implemented in software. One exemplary listing of code that performs the operations of configurable IIR filter 126 is presented in appendix A. The exemplary code is 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.

[0046] Referring to FIG. 7, a block diagram of BTSC compressor 30 is provided in which portions of the diagram are highlighted to illustrate functions that may be performed by a single (or multiple) implementations of configurable IIR filter 126. In particular, filtering performed by interpolation and fixed pre-emphasis stage 38 may be performed by configurable IIR filter 126. For example, an input of multiplexer 130 may be connected to

the appropriate filter input within interpolation and fixed pre-emphasis stage 38. Correspondingly, when this input of multiplexer 130 is selected, filter coefficients may be retrieved from memory and used to produce to an appropriate filter type and filter characteristics. Similarly, gain control bandpass filter 60 may be assigned to another input of multiplexer 130 in digital configurable IIR filter 126 and spectral control bandpass filter 52 may be assigned to still another input of multiplexer 130. Band-limiting unit 46 may be assigned to another input of multiplexer 130. For each of these selectable inputs, corresponding filter coefficients are stored (e.g., in memory) and may be retrieved by multiplexer 152 and/or multiplexer 162 of configurable IIR filter 126. In this example, filtering associated with four portions of BTSC compressor 30 is selectively performed by configurable IIR filter 126, however, in other arrangements, more or less filtering operations of the compressor may be performed by the configurable IIR filter. Additionally, configurable IIR filter 126 also provides a multiplication function via multiplier 138 and output registers 170 (shown in FIG. 5). Thereby, the operations of multipliers 54 and 62 may each be provided configurable IIR filter 126.

[0047] Referring to FIG. 8, portions of BTSC expander 86 are highlighted to identify filtering operations that may be performed by one or more configurable IIR filters that may be implemented with configurable IIR filter 126. For example, filtering associated with band-limiting unit 102 may be performed by configurable IIR filter 126. In particular, an input of multiplexer 130 may be assigned to band-limiting unit 102 such that when the input is selected, appropriate filtering coefficients are retrieved and used by configurable IIR filter 126. Similarly, filtering associated with gain control bandpass filter 104 (assigned to another input of multiplexer 130), spectral control bandpass filter 112 (assigned to another input of multiplexer 130), and fixed de-emphasis unit 124 (assigned to a still another input of selector 130) is consolidated into configurable IIR filter 126. Additionally, due to its multiplication function, configurable IIR filter 126 may provide the multiplication function for one or more of multipliers 106, 114, and 122.

[0048] While the previous example described using configurable IIR filter 126 with

BTSC encoders and BTSC decoders, encoders and decoders that comply with television audio standards may implement the configurable IIR filter. For example, encoders and/or decoders associated with the Near Instantaneously Companded Audio Multiplex (NICAM), which is used in Europe, may incorporate one or more configurable IIR filters such as IIR filter 126. Similarly, encoders and decoders implementing the A2/Zweiton television audio standard (currently used in parts of Europe and Asia) or the Electronics Industry Association of Japan (EIA - J) standard may incorporate one or more configurable IIR filters.

[0049] While the previous example described using configurable IIR filter 126 to encode and decode a difference signal produced from right and left audio channel, the configurable IIR filter may be used to encode and decode other audio signals. For example, configurable IIR filter 126 may be used to encode and/or decode an SAP channel, a professional channel, a sum channel, or one or more other individual or combined types of television audio channels.

[0050] 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

Appendix A

```

/*****
This module is the cascaded direct-form II implementation of
one or more discrete-time filters. It is actually a single
second-order section that can be 'recycled'.
*****/

// Generated by ParallelSOSFilterGenerator.m on 03-Jun-2004 08:59:29
//
// Delay Register Width = 35
// Filter Coefficients are in Q15
// **** Filter 1 Sum Interpolation & Preemphasis ****
// This filter is in Q15 format. (Q15 is required.)
// The 'b' coefficients in the last stage are in Q18 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 127.936 -----> 99.976% headroom
// Max Output Value = 23.781 -----> 99.964% headroom
//
//
//      b0      b1      b2      a0      a1      a2
//-----
// -6.6033712e-012 3.2117006e-002 1.3600689e+000 1.0000000e+000 -1.5758348e+000 6.2789060e-001
// 2.0305836e-002 9.3963698e-002 6.5186177e-002 1.0000000e+000 -1.5671917e+000 6.7399133e-001
// 1.1605752e-001 2.4121539e-002 3.7646657e-004 1.0000000e+000 -1.5759558e+000 7.6826398e-001
// 4.8703374e-001 -4.5435961e-001 0.0000000e+000 1.0000000e+000 -1.6429813e+000 9.1158385e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      9.9273109932813      13.4026919779221
// 2      127.9360639360639      22.7268252592786
// 3      127.9360639360638      17.9496934617571
// 4      127.9360639360638      23.7933114009230
//
// **** Filter 2 Sum Lowpass ****
// This filter is in Q15 format. (Q15 is required.)
// The 'b' coefficients in the last stage are in Q17 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 268.666 -----> 99.949% headroom
// Max Output Value = 4.086 -----> 99.997% headroom
//
//

```

```
//      b0      b1      b2      a0      a1      a2
//-----
// 8.4345402e-001 3.3105654e-001 8.4345402e-001 1.0000000e+000 -1.5131160e+000 5.8400921e-001
// 1.7907148e-001 -2.3574553e-001 1.7907148e-001 1.0000000e+000 -1.5854796e+000 7.0937033e-001
// 3.0401514e-001 -4.9151080e-001 3.0401514e-001 1.0000000e+000 -1.6621327e+000 8.4198754e-001
// 3.5735700e-001 -6.0794041e-001 3.5735700e-001 1.0000000e+000 -1.7107429e+000 9.2558392e-001
// 2.6136244e-001 -4.5235123e-001 2.6136243e-001 1.0000000e+000 -1.7366273e+000 9.6885152e-001
// 3.3644153e-001 -5.8564210e-001 3.3644153e-001 1.0000000e+000 -1.7519173e+000 9.9157326e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      14.8950457230439      30.0525971031239
// 2      268.6657342657337      32.1765299950301
// 3      268.6657342657345      26.2168014264289
// 4      268.6657342657350      17.3374523088056
// 5      268.6657342657346      7.0533046891134
// 6      268.6657342657326      4.0876878442666
//
// ***** Filter 3 Diff Interpolation & Preemphasis *****
// This filter is in Q15 format (Q15 is required.)
// The 'b' coefficients in the last stage are in Q12 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 255.872 -----> 99.951% headroom
// Max Output Value      = 259.440 -----> 99.994% headroom
//
//
//      b0      b1      b2      a0      a1      a2
//-----
// -1.2723363e-010 4.8662511e-002 3.5810704e+000 1.0000000e+000 -9.7203440e-001 1.0899437e-001
// 7.2682533e-003 4.9327184e-002 5.0728513e-002 1.0000000e+000 -1.5758348e+000 6.2789060e-001
// 1.3165964e-001 5.6960961e-002 3.6425943e-003 1.0000000e+000 -1.5671917e+000 6.7399133e-001
// 2.4084812e-001 1.3833431e-003 0.0000000e+000 1.0000000e+000 -1.5759558e+000 7.6826398e-001
// 9.4429177e+000 -1.8107524e+001 8.6732681e+000 1.0000000e+000 -1.6429813e+000 9.1158385e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      4.4056120909651      12.2436103717271
// 2      255.8721278721281      27.3647542209154
// 3      255.8721278721284      49.1968802729879
// 4      255.8721278721282      61.9802894946887
// 5      255.8721278721280      259.5699347572192
```

```
//
// ***** Filter 4 Diff Gain Ctrl Bandpass *****
// This filter is in Q17 format. (Q17 is required.)
// The 'b' coefficients in the last stage are in Q21 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 13051.645 -----> 90.042% headroom
// Max Output Value      = 1.915 -----> 99.907% headroom
//
//
//      b0      b1      b2      a0      a1      a2
//-----
// 3.3052890e-002 0.0000000e+000 -3.3052890e-002 1.0000000e+000 -1.9327087e+000 9.3278529e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      13051.6450743394910      1.9154790026336
//
// ***** Filter 5 Diff Gain Ctrl Integrator *****
// This filter is in Q13 format. (Q13 is required.)
// The 'b' coefficients in the last stage are in Q28 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 1499902.076 -----> 28.479% headroom
// Max Output Value      = 225.000 -----> 12.109% headroom
//
//
//      b0      b1      b2      a0      a1      a2
//-----
// 7.5042399e-005 7.5042399e-005 0.0000000e+000 1.0000000e+000 -9.9984992e-001 0.0000000e+000
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      1499902.0762455594000      225.1124999993376
//
// ***** Filter 6 Diff Spec Ctrl Bandpass *****
// This filter is in Q17 format. (Q17 is required.)
// The 'b' coefficients in the last stage are in Q18 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 39.333 -----> 99.970% headroom
// Max Output Value      = 3.683 -----> 99.978% headroom
```

```
//
//
//  b0      b1      b2      a0      a1      a2
//-----
// 8.6691012e-001 -1.7351060e+000 8.6819671e-001 1.0000000e+000 -1.7132232e+000 7.6815543e-001
// 3.3300661e-001 1.8508507e-004 -3.3343983e-001 1.0000000e+000 -1.2652200e+000 3.3953192e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
//-----
// 1      39.3329526574003      3.7578420694186
// 2      14.8039499335385      3.6850497626675
//
// ***** Filter 7 Diff Spec Ctrl Integrator *****
// This filter is in Q22 format. (Q22 is required )
// The 'b' coefficients in the last stage are in Q27 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 1949.451 -----> 52.406% headroom
// Max Output Value      = 0.684 -----> 31.633% headroom
//
//
//  b0      b1      b2      a0      a1      a2
//-----
// 1.7543809e-004 1.7543809e-004 0.0000000e+000 1.0000000e+000 -9.9954323e-001 0.0000000e+000
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
//-----
// 1      1949.4512384999191      0.6840160013750
//
// ***** Filter 8 Diff Spectral Compression *****
// This filter is in Q15 format. (Q15 is required.)
// The 'b' coefficients in the last stage are in Q11 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 4819.452 -----> 99.081% headroom
// Max Output Value      = 4880.066 -----> 99.942% headroom
//
//
//  b0      b1      b2      a0      a1      a2
//-----
// 3.8625515e+001 -3.8130535e+001 0.0000000e+000 1.0000000e+000 -9.8671384e-001 0.0000000e+000
//
```

// Magnitudes are relative to the system input, not necessarily the filter input.

// Section Del. max Outmax

// -----

// 1 4819.4523948899350 4882 5058955489840

//

// ***** Filter 9 Diff Lowpass *****

// This filter is in Q15 format. (Q15 is required.)

// The 'b' coefficients in the last stage are in Q17 format.

//

// ----- (192 kHz Sample Rate) -----

// Max Delay Register Value = 255 872 -----> 99.951% headroom

// Max Output Value = 3.891 -----> 99.997% headroom

//

//

// b0 b1 b2 a0 a1 a2

// -----

// 8.4345402e-001 3.3105654e-001 8.4345402e-001 1.0000000e+000 -1.5131160e+000 5.8400921e-001

// 1.7907148e-001 -2.3574553e-001 1.7907148e-001 1.0000000e+000 -1.5854796e+000 7.0937033e-001

// 3.0401514e-001 -4.9151080e-001 3.0401514e-001 1.0000000e+000 -1.6621327e+000 8.4198754e-001

// 3.5735700e-001 -6.0794041e-001 3.5735700e-001 1.0000000e+000 -1.7107429e+000 9.2558392e-001

// 2.6136244e-001 -4.5235123e-001 2.6136243e-001 1.0000000e+000 -1.7366273e+000 9.6885152e-001

// 3.3644153e-001 -5.8564210e-001 3.3644153e-001 1.0000000e+000 -1.7519173e+000 9.9157326e-001

//

// Magnitudes are relative to the system input, not necessarily the filter input.

// Section Del. max Outmax

// -----

// 1 14.1857578314703 28.6215210505942

// 2 255.8721278721279 30.6443142809810

// 3 255.8721278721285 24.9683823108847

// 4 255.8721278721286 16.5118593417196

// 5 255.8721278721287 6.7174330372509

// 6 255.8721278721255 3.8930360421587

//

// ***** Filter 10 SAP Interpolation & Preemphasis *****

// This filter is in Q15 format. (Q15 is required.)

// The 'b' coefficients in the last stage are in Q12 format.

//

// ----- (192 kHz Sample Rate) -----

// Max Delay Register Value = 127.936 -----> 99.976% headroom

// Max Output Value = 79.936 -----> 99.998% headroom

//

//

// b0 b1 b2 a0 a1 a2

// -----

```
// -1.8231226e-009 2.1884032e-002 1.7219441e+000 1.0000000e+000 -9.7203440e-001 1.0899437e-001
// 3.1584799e-003 2.2857938e-002 2.4903862e-002 1.0000000e+000 -1.7083043e+000 7.3325661e-001
// '6 0720051e-002 2.7399739e-002 1.8184569e-003 1.0000000e+000 -1.7179059e+000 7.6872240e-001
// 9.5595731e-002 5.6899673e-004 0.0000000e+000 1.0000000e+000 -1.7487378e+000 8.3882941e-001
// 1.0900737e+001 -2.0903005e+001 1.0012267e+001 1.0000000e+000 -1.8173075e+000 9.4015130e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----      -
// 1      2.2028060429820      2.9414855734591
// 2      127.9360639360640      6.5017997596439
// 3      127.9360639360640      11 5065707750231
// 4      127.9360639360641      12.3029392557771
// 5      127.9360639360646      79.9758599046644
//
// ***** Filter 11 SAP Gain Ctrl Bandpass *****
// This filter is in Q17 format. (Q17 is required.)
// The 'b' coefficients in the last stage are in Q21 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 13051.591 -----> 90.042% headroom
// Max Output Value      = 1.984 -----> 99.903% headroom
//
//
//      b0      b1      b2      a0      a1      a2
// -----
// 3.3052890e-002 0.0000000e+000 -3.3052890e-002 1.0000000e+000 -1.9327087e+000 9.3278529e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----      -
// 1      13051.5913591675250      1.9854692831948
//
// ***** Filter 12 SAP Gain Ctrl Integrator *****
// This filter is in Q13 format. (Q13 is required.)
// The 'b' coefficients in the last stage are in Q28 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 1499902.076 -----> 28 479% headroom
// Max Output Value      = 225.000 -----> 12.109% headroom
//
//
//      b0      b1      b2      a0      a1      a2
// -----
```

```
// 7.5042399e-005 7.5042399e-005 0.0000000e+000 1.0000000e+000 -9.9984992e-001 0.0000000e+000
//
// Magnitudes are relative to the system input, not necessarily the filter input
// Section      Del. max      Outmax
// -----
// 1      1499902.0762455594000      225.1124999993376
//
// ***** Filter 13 SAP Spec Ctrl Bandpass *****
// This filter is in Q17 format. (Q17 is required.)
// The 'b' coefficients in the last stage are in Q18 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 60.285 -----> 99.954% headroom
// Max Output Value      = 3 340 -----> 99.980% headroom
//
//
//      b0      b1      b2      a0      a1      a2
// -----
// 8.6691012e-001 -1.7351060e+000 8.6819671e-001 1.0000000e+000 -1.7132232e+000 7.6815543e-001
// 3.3300661e-001 1.8508507e-004 -3.3343983e-001 1.0000000e+000 -1.2652200e+000 3.3953192e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      60.2854167958472      3.3663545412242
// 2      18.1337922344007      3.3413388460805
//
// ***** Filter 14 SAP Spec Ctrl Integrator *****
// This filter is in Q22 format. (Q22 is required.)
// The 'b' coefficients in the last stage are in Q27 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 1949.451 -----> 52.406% headroom
// Max Output Value      = 0.684 -----> 31.633% headroom
//
//
//      b0      b1      b2      a0      a1      a2
// -----
// 1.7543809e-004 1.7543809e-004 0.0000000e+000 1.0000000e+000 -9.9954323e-001 0.0000000e+000
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      1949.4512384999191      0.6840160013750
```

```
//
// ***** Filter 15 SAP Spectral Compression *****
// This filter is in Q15 format. (Q15 is required.)
// The 'b' coefficients in the last stage are in Q11 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 4819.452 -----> 99.081% headroom
// Max Output Value      = 4880.066 -----> 99.942% headroom
//
//
//      b0      b1      b2      a0      a1      a2
// -----
// 3.8625515e+001 -3.8130535e+001 0.0000000e+000 1.0000000e+000 -9.8671384e-001 0.0000000e+000
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      4819.4523948899350      4882.5058955489840
//
// ***** Filter 16 SAP Lowpass *****
// This filter is in Q15 format. (Q15 is required.)
// The 'b' coefficients in the last stage are in Q15 format.
//
// ----- (192 kHz Sample Rate) -----
// Max Delay Register Value = 127.936 -----> 99.976% headroom
// Max Output Value      = 3.873 -----> 99.999% headroom
//
//
//      b0      b1      b2      a0      a1      a2
// -----
// 1.4236972e-001 -6.1114088e-002 1.4236972e-001 1.0000000e+000 -1.6681586e+000 7.0166935e-001
// 1.7291208e-001 -2.8898617e-001 1.7291208e-001 1.0000000e+000 -1.7369812e+000 7.9467313e-001
// 3.0167936e-001 -5.5024915e-001 3.0167936e-001 1.0000000e+000 -1.8076638e+000 8.9011358e-001
// 3.5737308e-001 -6.6631149e-001 3.5737309e-001 1.0000000e+000 -1.8511403e+000 9.4867213e-001
// 2.6223393e-001 -4.9257885e-001 2.6223391e-001 1.0000000e+000 -1.8738553e+000 9.7876206e-001
// 1.4418675e+000 -2.7151244e+000 1.4418675e+000 1.0000000e+000 -1.8862768e+000 9.9419075e-001
//
// Magnitudes are relative to the system input, not necessarily the filter input.
// Section      Del. max      Outmax
// -----
// 1      30 0017134252655      6.7084273462983
// 2      127.9360639360641      7.1340558588011
// 3      127.9360639360641      5.7714816868819
// 4      127.9360639360642      3.8011918024620
```

```
// 5      127.9360639360640      1.5533507892725
// 6      127.9360639360671      3.8749910462902
//
module SOS1VCom (i1Cik,
    i1Start,
    i1DRegClear,
    i17DataIn1,
    or21DataOut1,
    i17DataIn2,
    or19DataOut2,
    i17DataIn3,
    or25DataOut3,
    i20DataIn4,
    or19DataOut4,
    i22DataIn5,
    or22DataOut5,
    i20DataIn6,
    or20DataOut6,
    i23DataIn7,
    or23DataOut7,
    i22DataIn8,
    or29DataOut8,
    i18B0_08,
    i18B1_08,
    i18A1_08,
    i16DataIn9,
    or18DataOut9,
    i16DataIn10,
    or23DataOut10,
    i20DataIn11,
    or19DataOut11,
    i22DataIn12,
    or22DataOut12,
    i20DataIn13,
    or20DataOut13,
    i23DataIn14,
    or23DataOut14,
    i22DataIn15,
    or29DataOut15,
    i18B0_15,
    i18B1_15,
    i18A1_15,
    i16DataIn16,
    or18DataOut16);
```

```
input    i1Clk;
input    i1Start;
input    i1DRegClear;
input [16:0] i17DataIn1;
output [20:0] or21DataOut1;
input [16:0] i17DataIn2;
output [18:0] or19DataOut2;
input [16:0] i17DataIn3;
output [24:0] or25DataOut3;
input [19:0] i20DataIn4;
output [18:0] or19DataOut4;
input [21:0] i22DataIn5;
output [21:0] or22DataOut5;
input [19:0] i20DataIn6;
output [19:0] or20DataOut6;
input [22:0] i23DataIn7;
output [22:0] or23DataOut7;
input [21:0] i22DataIn8;
output [28:0] or29DataOut8;
input [17:0] i18B0_08;
input [17:0] i18B1_08;
input [17:0] i18A1_08;
input [15:0] i16DataIn9;
output [17:0] or18DataOut9;
input [15:0] i16DataIn10;
output [22:0] or23DataOut10;
input [19:0] i20DataIn11;
output [18:0] or19DataOut11;
input [21:0] i22DataIn12;
output [21:0] or22DataOut12;
input [19:0] i20DataIn13;
output [19:0] or20DataOut13;
input [22:0] i23DataIn14;
output [22:0] or23DataOut14;
input [21:0] i22DataIn15;
output [28:0] or29DataOut15;
input [17:0] i18B0_15;
input [17:0] i18B1_15;
input [17:0] i18A1_15;
input [15:0] i16DataIn16;
output [17:0] or18DataOut16;
reg signed [20:0] or21DataOut1;
reg signed [18:0] or19DataOut2;
reg signed [24:0] or25DataOut3;
```

```

reg signed [18:0] or19DataOut4;
reg signed [21:0] or22DataOut5;
reg signed [19:0] or20DataOut6;
reg signed [22:0] or23DataOut7;
reg signed [28:0] or29DataOut8;
reg signed [17:0] or18DataOut9;
reg signed [22:0] or23DataOut10;
reg signed [18:0] or19DataOut11;
reg signed [21:0] or22DataOut12;
reg signed [19:0] or20DataOut13;
reg signed [22:0] or23DataOut14;
reg signed [28:0] or29DataOut15;
reg signed [17:0] or18DataOut16;
reg signed [37:0] r38InputReg;
reg signed [49:0] r50Addend1;
reg signed [49:0] r50Addend2;
wire signed [49:0] w50Sum;
reg signed [49:0] r50Sum;
reg signed [34:0] r35MultInput1;
reg signed [17:0] r18MultInput2;
wire signed [49:0] w50Product;
reg signed [49:0] r50Product;
reg signed [49:0] r50ProductReg;
reg          r1ShiftRegEnable;
reg [34:0] r35ShiftRegInput;
wire signed [34:0] w35D1;
reg signed [34:0] r35D2;
reg [86:0] r87DelayShiftReg34;
reg [86:0] r87DelayShiftReg33;
reg [86:0] r87DelayShiftReg32;
reg [86:0] r87DelayShiftReg31;
reg [86:0] r87DelayShiftReg30;
reg [86:0] r87DelayShiftReg29;
reg [86:0] r87DelayShiftReg28;
reg [86:0] r87DelayShiftReg27;
reg [86:0] r87DelayShiftReg26;
reg [86:0] r87DelayShiftReg25;
reg [86:0] r87DelayShiftReg24;
reg [86:0] r87DelayShiftReg23;
reg [86:0] r87DelayShiftReg22;
reg [86:0] r87DelayShiftReg21;
reg [86:0] r87DelayShiftReg20;
reg [86:0] r87DelayShiftReg19;
reg [86:0] r87DelayShiftReg18;

```

```

reg    [86:0] r87DelayShiftReg17;
reg    [86:0] r87DelayShiftReg16;
reg    [86:0] r87DelayShiftReg15;
reg    [86:0] r87DelayShiftReg14;
reg    [86:0] r87DelayShiftReg13;
reg    [86:0] r87DelayShiftReg12;
reg    [86:0] r87DelayShiftReg11;
reg    [86:0] r87DelayShiftReg10;
reg    [86:0] r87DelayShiftReg09;
reg    [86:0] r87DelayShiftReg08;
reg    [86:0] r87DelayShiftReg07;
reg    [86:0] r87DelayShiftReg06;
reg    [86:0] r87DelayShiftReg05;
reg    [86:0] r87DelayShiftReg04;
reg    [86:0] r87DelayShiftReg03;
reg    [86:0] r87DelayShiftReg02;
reg    [86:0] r87DelayShiftReg01;
reg    [86:0] r87DelayShiftReg00;
wire    w1CoefficientClkEnable;
reg    r1CoefficientClkEnable;
wire signed [17:0] w18Coefficient;
reg    [3:0] r4SectionState;
reg    [5:0] r6SectionNumber;
parameter
    FILTER01_LAST_SECTION = 6'd3,
    FILTER02_LAST_SECTION = 6'd9,
    FILTER03_LAST_SECTION = 6'd14,
    FILTER04_LAST_SECTION = 6'd15,
    FILTER05_LAST_SECTION = 6'd16,
    FILTER06_LAST_SECTION = 6'd18,
    FILTER07_LAST_SECTION = 6'd19,
    FILTER08_LAST_SECTION = 6'd20,
    FILTER09_LAST_SECTION = 6'd26,
    FILTER10_LAST_SECTION = 6'd31,
    FILTER11_LAST_SECTION = 6'd32,
    FILTER12_LAST_SECTION = 6'd33,
    FILTER13_LAST_SECTION = 6'd35,
    FILTER14_LAST_SECTION = 6'd36,
    FILTER15_LAST_SECTION = 6'd37,
    FILTER16_LAST_SECTION = 6'd43,
    LAST_SECTION          = FILTER16_LAST_SECTION;
//State Machine States
parameter
    IDLE          = 4'd0,

```

```

WAIT1  = 4'd6,
STATE1A = 4'd7,
STATE1  = 4'd1,
STATE2  = 4'd2,
STATE3  = 4'd3,
STATE4  = 4'd4,
STATE5  = 4'd5,
LAST    = 4'd8;

//synthesis translate_off
initial
begin
    r4SectionState = IDLE;
end
//synthesis translate_on
assign w1CoefficientClkEnable = r1CoefficientClkEnable | i1Start;
CoefSelectSOS1VCom
    CoefSelect1  (.i1Clk      (i1Clk),
                  .i1ClkEnable (w1CoefficientClkEnable),
                  .i18B0_08    (i18B0_08),
                  .i18B1_08    (i18B1_08),
                  .i18A1_08    (i18A1_08),
                  .i18B0_15    (i18B0_15),
                  .i18B1_15    (i18B1_15),
                  .i18A1_15    (i18A1_15),
                  .or18Data    (w18Coefficient));
MultSOS1VCom
    Multiplier1 (.i35MultInput1 (r35MultInput1),
                 .i18MultInput2 (r18MultInput2),
                 .ow50Product    (w50Product));
AdderSOS1VCom
    Adder1 (.i50Addend1 (r50Addend1),
            .i50Addend2 (r50Addend2),
            .ow50Sum     (w50Sum));
assign w35D1 = {r87DelayShiftReg34[0],
                r87DelayShiftReg33[0],
                r87DelayShiftReg32[0],
                r87DelayShiftReg31[0],
                r87DelayShiftReg30[0],
                r87DelayShiftReg29[0],
                r87DelayShiftReg28[0],
                r87DelayShiftReg27[0],
                r87DelayShiftReg26[0],
                r87DelayShiftReg25[0],
                r87DelayShiftReg24[0],

```

```

r87DelayShiftReg23[0],
r87DelayShiftReg22[0],
r87DelayShiftReg21[0],
r87DelayShiftReg20[0],
r87DelayShiftReg19[0],
r87DelayShiftReg18[0],
r87DelayShiftReg17[0],
r87DelayShiftReg16[0],
r87DelayShiftReg15[0],
r87DelayShiftReg14[0],
r87DelayShiftReg13[0],
r87DelayShiftReg12[0],
r87DelayShiftReg11[0],
r87DelayShiftReg10[0],
r87DelayShiftReg09[0],
r87DelayShiftReg08[0],
r87DelayShiftReg07[0],
r87DelayShiftReg06[0],
r87DelayShiftReg05[0],
r87DelayShiftReg04[0],
r87DelayShiftReg03[0],
r87DelayShiftReg02[0],
r87DelayShiftReg01[0],
r87DelayShiftReg00[0]);
always @(r4SectionState or i1DRegClear or w35D1 or w50Sum)
begin
  if (i1DRegClear) begin
    r1ShiftRegEnable <= 1'b1;
    r35ShiftRegInput <= 35'b0;
  end else begin
    case (r4SectionState)
      IDLE:
        begin
          r1ShiftRegEnable <= 1'b0;
          r35ShiftRegInput <= 35'bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
        end
      STATE1A, STATE1:
        begin
          r1ShiftRegEnable <= 1'b1;
          r35ShiftRegInput <= w50Sum[49:15];
        end
      STATE5:
        begin
          r1ShiftRegEnable <= 1'b1;

```

```

        r35ShiftRegInput <= w35D1;
    end
default:
    begin
        r1ShiftRegEnable <= 1'b0;
        r35ShiftRegInput <= 35'bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
    end
endcase
end
end
always @(r4SectionState or r50Sum or r38InputReg or r50Product or r50ProductReg)
case (r4SectionState)
    STATE1A, STATE1: begin
        r50Addend1 <= r50Sum;
        r50Addend2 <= { 12 {r38InputReg[37]} , r38InputReg };
    end
    STATE2: begin
        r50Addend1 <= r50ProductReg;
        r50Addend2 <= r50Product;
    end
    STATE3: begin
        r50Addend1 <= r50Sum;
        r50Addend2 <= r50Product;
    end
    STATE5: begin
        r50Addend1 <= r50ProductReg;
        r50Addend2 <= r50Product;
    end
    default: begin
        r50Addend1 <= 50'bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
        r50Addend2 <= 50'bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
    end
endcase
always @(posedge i1Clk)
begin
    r50Product <= w50Product;
    r18MultInput2 <= w18Coefficient;
    if (r1ShiftRegEnable) begin
        r87DelayShiftReg34 <= {r35ShiftRegInput[34], r87DelayShiftReg34[86:1]};
        r87DelayShiftReg33 <= {r35ShiftRegInput[33], r87DelayShiftReg33[86:1]};
        r87DelayShiftReg32 <= {r35ShiftRegInput[32], r87DelayShiftReg32[86:1]};
        r87DelayShiftReg31 <= {r35ShiftRegInput[31], r87DelayShiftReg31[86:1]};
        r87DelayShiftReg30 <= {r35ShiftRegInput[30], r87DelayShiftReg30[86:1]};
        r87DelayShiftReg29 <= {r35ShiftRegInput[29], r87DelayShiftReg29[86:1]};
    end
end

```

```

r87DelayShiftReg28 <= {r35ShiftRegInput[28], r87DelayShiftReg28[86:1]};
r87DelayShiftReg27 <= {r35ShiftRegInput[27], r87DelayShiftReg27[86:1]};
r87DelayShiftReg26 <= {r35ShiftRegInput[26], r87DelayShiftReg26[86:1]};
r87DelayShiftReg25 <= {r35ShiftRegInput[25], r87DelayShiftReg25[86:1]};
r87DelayShiftReg24 <= {r35ShiftRegInput[24], r87DelayShiftReg24[86:1]};
r87DelayShiftReg23 <= {r35ShiftRegInput[23], r87DelayShiftReg23[86:1]};
r87DelayShiftReg22 <= {r35ShiftRegInput[22], r87DelayShiftReg22[86:1]};
r87DelayShiftReg21 <= {r35ShiftRegInput[21], r87DelayShiftReg21[86:1]};
r87DelayShiftReg20 <= {r35ShiftRegInput[20], r87DelayShiftReg20[86:1]};
r87DelayShiftReg19 <= {r35ShiftRegInput[19], r87DelayShiftReg19[86:1]};
r87DelayShiftReg18 <= {r35ShiftRegInput[18], r87DelayShiftReg18[86:1]};
r87DelayShiftReg17 <= {r35ShiftRegInput[17], r87DelayShiftReg17[86:1]};
r87DelayShiftReg16 <= {r35ShiftRegInput[16], r87DelayShiftReg16[86:1]};
r87DelayShiftReg15 <= {r35ShiftRegInput[15], r87DelayShiftReg15[86:1]};
r87DelayShiftReg14 <= {r35ShiftRegInput[14], r87DelayShiftReg14[86:1]};
r87DelayShiftReg13 <= {r35ShiftRegInput[13], r87DelayShiftReg13[86:1]};
r87DelayShiftReg12 <= {r35ShiftRegInput[12], r87DelayShiftReg12[86:1]};
r87DelayShiftReg11 <= {r35ShiftRegInput[11], r87DelayShiftReg11[86:1]};
r87DelayShiftReg10 <= {r35ShiftRegInput[10], r87DelayShiftReg10[86:1]};
r87DelayShiftReg09 <= {r35ShiftRegInput[9], r87DelayShiftReg09[86:1]};
r87DelayShiftReg08 <= {r35ShiftRegInput[8], r87DelayShiftReg08[86:1]};
r87DelayShiftReg07 <= {r35ShiftRegInput[7], r87DelayShiftReg07[86:1]};
r87DelayShiftReg06 <= {r35ShiftRegInput[6], r87DelayShiftReg06[86:1]};
r87DelayShiftReg05 <= {r35ShiftRegInput[5], r87DelayShiftReg05[86:1]};
r87DelayShiftReg04 <= {r35ShiftRegInput[4], r87DelayShiftReg04[86:1]};
r87DelayShiftReg03 <= {r35ShiftRegInput[3], r87DelayShiftReg03[86:1]};
r87DelayShiftReg02 <= {r35ShiftRegInput[2], r87DelayShiftReg02[86:1]};
r87DelayShiftReg01 <= {r35ShiftRegInput[1], r87DelayShiftReg01[86:1]};
r87DelayShiftReg00 <= {r35ShiftRegInput[0], r87DelayShiftReg00[86:1]};

```

end

if (!DRegClear) begin

```

    r4SectionState    <= IDLE;
    r1CoefficientClkEnable <= 1'b0;
    r35D2             <= 35'd0;
    r50ProductReg     <= 50'd0;
    r50Sum            <= 50'd0;
    or21DataOut1      <= 21'd0;
    or19DataOut2      <= 19'd0;
    or25DataOut3      <= 25'd0;
    or19DataOut4      <= 19'd0;
    or22DataOut5      <= 22'd0;
    or20DataOut6      <= 20'd0;
    or23DataOut7      <= 23'd0;
    or29DataOut8      <= 29'd0;

```

```

or18DataOut9  <= 18'd0;
or23DataOut10 <= 23'd0;
or19DataOut11 <= 19'd0;
or22DataOut12 <= 22'd0;
or20DataOut13 <= 20'd0;
or23DataOut14 <= 23'd0;
or29DataOut15 <= 29'd0;
or18DataOut16 <= 18'd0;
end else begin
  case (r4SectionState)
    IDLE:
      begin
        if (i1Start) begin
          r4SectionState  <= WAIT1;
          r1CoefficientClkEnable <= 1'b1;
          r6SectionNumber  <= 6'd0;
          r38InputReg      <= { {6{i17DataIn1[16]}}, i17DataIn1, 15'b0 };
        end
      end
    WAIT1:
      begin
        r4SectionState <= STATE1A;
        r35MultInput1  <= r35D2;
      end
    STATE1A:
      begin
        r4SectionState <= STATE2;
        r35D2          <= w35D1;
        r35MultInput1  <= w50Sum[49:15];
        r50Sum         <= w50Sum;
      end
    STATE1:
      begin
        r4SectionState <= STATE2;
        r35D2          <= w35D1;
        r35MultInput1  <= w50Sum[49:15];
        r50Sum         <= w50Sum;
        r50ProductReg  <= r50Product;
      end
    STATE2:
      begin
        r4SectionState <= STATE3;
        r35MultInput1  <= w35D1;
        r50Sum         <= w50Sum;

```

```

end
STATE3:
begin
    r4SectionState <= STATE4;
    r35MultInput1 <= r35D2;
    r50Sum <= w50Sum;
end
STATE4:
begin
    r4SectionState <= STATE5;
    case (r6SectionNumber)
        FILTER01_LAST_SECTION: or21DataOut1 <= r50Sum[38:18];
        FILTER02_LAST_SECTION: or19DataOut2 <= r50Sum[35:17];
        FILTER03_LAST_SECTION: or25DataOut3 <= r50Sum[36:12];
        FILTER04_LAST_SECTION: or19DataOut4 <= r50Sum[39:21];
        FILTER05_LAST_SECTION: or22DataOut5 <= r50Sum[49:28];
        FILTER06_LAST_SECTION: or20DataOut6 <= r50Sum[37:18];
        FILTER07_LAST_SECTION: or23DataOut7 <= r50Sum[49:27];
        FILTER08_LAST_SECTION: or29DataOut8 <= r50Sum[39:11];
        FILTER09_LAST_SECTION: or18DataOut9 <= r50Sum[34:17];
        FILTER10_LAST_SECTION: or23DataOut10 <= r50Sum[34:12];
        FILTER11_LAST_SECTION: or19DataOut11 <= r50Sum[39:21];
        FILTER12_LAST_SECTION: or22DataOut12 <= r50Sum[49:28];
        FILTER13_LAST_SECTION: or20DataOut13 <= r50Sum[37:18];
        FILTER14_LAST_SECTION: or23DataOut14 <= r50Sum[49:27];
        FILTER15_LAST_SECTION: or29DataOut15 <= r50Sum[39:11];
        FILTER16_LAST_SECTION: or18DataOut16 <= r50Sum[32:15];
    endcase
    r38InputReg <= r50Sum[37:0];
    r50ProductReg <= r50Product;
    r35MultInput1 <= w35D1;
end
STATE5:
begin
    r6SectionNumber <= r6SectionNumber + 1;
    if (r6SectionNumber == LAST_SECTION)
        begin
            r4SectionState <= LAST;
            r1CoefficientClkEnable <= 1'b0;
        end
    else begin
        r4SectionState <= STATE1;
        r35MultInput1 <= r35D2;
    end
end

```

```

case (r6SectionNumber)
  FILTER01_LAST_SECTION: r38InputReg <= { {6{i17DataIn2[16]}}, i17DataIn2, 15'b0 };
  FILTER02_LAST_SECTION: r38InputReg <= { {6{i17DataIn3[16]}}, i17DataIn3, 15'b0 },
  FILTER03_LAST_SECTION: r38InputReg <= { {3{i20DataIn4[19]}}, i20DataIn4, 15'b0 };
  FILTER04_LAST_SECTION: r38InputReg <= { i22DataIn5[21], i22DataIn5, 15'b0 };
  FILTER05_LAST_SECTION: r38InputReg <= { {3{i20DataIn6[19]}}, i20DataIn6, 15'b0 };
  FILTER06_LAST_SECTION: r38InputReg <= { i23DataIn7, 15'b0 };
  FILTER07_LAST_SECTION: r38InputReg <= { i22DataIn8[21], i22DataIn8, 15'b0 };
  FILTER08_LAST_SECTION: r38InputReg <= { {7{i16DataIn9[15]}}, i16DataIn9, 15'b0 };
  FILTER09_LAST_SECTION: r38InputReg <= { {7{i16DataIn10[15]}}, i16DataIn10, 15'b0 };
  FILTER10_LAST_SECTION: r38InputReg <= { {3{i20DataIn11[19]}}, i20DataIn11, 15'b0 };
  FILTER11_LAST_SECTION: r38InputReg <= { i22DataIn12[21], i22DataIn12, 15'b0 };
  FILTER12_LAST_SECTION: r38InputReg <= { {3{i20DataIn13[19]}}, i20DataIn13, 15'b0 };
  FILTER13_LAST_SECTION: r38InputReg <= { i23DataIn14, 15'b0 };
  FILTER14_LAST_SECTION: r38InputReg <= { i22DataIn15[21], i22DataIn15, 15'b0 };
  FILTER15_LAST_SECTION: r38InputReg <= { {7{i16DataIn16[15]}}, i16DataIn16, 15'b0 };
endcase
r50Sum      <= w50Sum;
end
LAST:
begin
  r4SectionState <= IDLE;
  r50ProductReg <= r50Product;
end
default:
begin
  r4SectionState <= IDLE;
end
endcase
end
end
endmodule

```

WHAT IS CLAIMED IS:**1. A television audio signal encoder, comprising:**

a device configured to sum a left channel audio signal and a right channel audio signal to produce a sum signal, and to subtract one of the left and right audio signals from the other of the left and right signals to produce a difference signal; and

a configurable infinite impulse response digital filter configured to selectively use at least one set of filter coefficients to filter the difference signal, wherein the set of filter coefficients is applied to the difference signal by a single multiplier in a recursive manner to prepare the difference signal for transmission.

2. The television audio signal encoder of claim 1, wherein the configurable infinite impulse response digital filter includes a feedback path to apply the set of filter coefficients to the difference signal in a recursive manner.

3. The television audio signal encoder of claim 2, wherein the feedback path includes a shift register to delay digital signals associated with the difference signal.

4. The television audio signal encoder of claim 1, wherein the configurable infinite impulse response digital filter is configured to multiply a signal associated with the difference signal and provide an output of this multiplication.

5. The television audio signal encoder of claim 1, where the configurable infinite impulse response digital filter includes a selector configured to select a digital input signal.

6. The television audio signal encoder of claim 1, wherein the configurable infinite impulse response digital filter includes a selector configured to select one of the filter coefficients.

7. The television audio signal encoder of claim 5, wherein the selector includes a multiplexer.
8. The television audio signal encoder of claim 1, wherein the configurable infinite impulse response digital filter is configured to provide a low pass filter.
9. The television audio signal encoder of claim 1, wherein the configurable infinite impulse response digital filter includes a single adder for applying the filter coefficients to the difference signal in a recursive manner.
10. The television audio signal encoder of claim 1, wherein the television audio signal complies to the Broadcast Television System Committee (BTSC) standard.
11. The television audio signal encoder of claim 1, wherein the television audio signal complies to the Near Instantaneously Companded Audio Multiplex (NICAM) standard.
12. The television audio signal encoder of claim 1, wherein the television audio signal complies to the A2/Zweiton standard.
13. The television audio signal encoder of claim 1, wherein the television audio signal complies to the EIA – J standard.
14. The television audio signal encoder of claim 1, wherein the configurable infinite impulse response digital filter is implemented in an integrated circuit.
15. A television audio signal decoder, comprising:
 - a configurable infinite impulse response digital filter configured to selectively use at least one set of filter coefficients to filter a difference signal, wherein the difference signal is produced by subtracting one of a left channel and a

right channel audio signal from the other of the left channel and right channel audio signal, wherein the set of filter coefficients is applied to the difference signal by a single multiplier in a recursive manner to prepare the difference signal for separating the left channel and right channel audio signals; and

a device configured to separate the left channel and right channel audio signals from the difference signal and a sum signal, wherein the sum signal includes the sum the left channel audio signal and the right channel audio signal.

16. A television audio signal decoder of claim 15, wherein the configurable infinite impulse response digital filter includes a feedback path to apply the set of filter coefficients to the difference signal in a recursive manner.

17. The television audio signal decoder of claim 16, wherein the feedback path includes a shift register to delay digital signals associated with the difference signal.

18. The television audio signal decoder of claim 15, wherein the configurable infinite impulse response digital filter is configured to multiple a signal associated with the difference signal and provide an output of this multiplication.

19. The television audio signal decoder of claim 15, where the configurable infinite impulse response digital filter includes a selector configured to select a digital input signal.

20. The television audio signal decoder of claim 15, wherein the configurable infinite impulse response digital filter includes a selector configured to select one of the filter coefficients.

21. The television audio signal decoder of claim 19, wherein the selector includes a multiplexer.

22. The television audio signal decoder of claim 15, wherein the configurable infinite

impulse response digital filter is configured to provide a low pass filter.

23. The television audio signal decoder of claim 15, wherein the configurable infinite impulse response digital filter includes a single adder for applying the filter coefficients to the difference signal in a recursive manner.

24. The television audio signal decoder of claim 15, wherein the television audio signal complies to the Broadcast Television System Committee (BTSC) standard.

25. The television audio signal decoder of claim 15, wherein the television audio signal complies to the Near Instantaneously Companded Audio Multiplex (NICAM) standard.

26. The television audio signal decoder of claim 15, wherein the television audio signal complies to the A2/Zweiton standard.

27. The television audio signal decoder of claim 15, wherein the television audio signal complies to the EIA – J standard.

28. The television audio signal decoder of claim 15, wherein the configurable infinite impulse response digital filter is implemented in an integrated circuit.