FILED

10 OCT 20 AM 10: 33

CLERK U.S. DISTRICT COURT
CENTRAL DIST. OF CALIF.
LOS ANGELES

BY:

Wayne W. Call, Bar No. 56676
wcall@calljensen.com
Aaron L. Renfro, Bar No. 255086
arenfro@calljensen.com
CALL & JENSEN
A Professional Corporation
610 Newport Center Drive, Suite 700
Newport Beach, CA 92660
Tel: (949) 717-3000; Fax: (949) 717-3100

Bruce S. Sostek, Texas Bar No. 18855700
(Pro Hac Vice Pending)
bruce.sostek@tklaw.com
Jane Politz Brandt, Texas Bar No. 02882090
(Pro Hac Vice Pending)
jane.brandt@tklaw.com
Richard L. Wynne, Jr., Texas Bar No. 24003214
(Pro Hac Vice Pending)
richard.wynne@tklaw.com
THOMPSON & KNIGHT LLP
1722 Routh Street, Suite 1500
Dallas, TX 75201
(214) 969-1700; (214) 969-1751

Attorneys for Plaintiff
LSI Corporation, dba LSI Logic Corporation

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA, WESTERN DIVISION

LSI CORPORATION, dba LSI Logic Corporation,

Plaintiff,

vs.

VIZIO, INC.,

Defendant.

SACV10-01602 AG AJWx

Case No.

COMPLAINT

DEMAND FOR JURY TRIAL

Complaint Filed:
Trial Date: None Set

CALL & JENSEN
A PROFESSIONAL CORPORATION

Plaintiff, LSI Corporation, dba LSI Logic Corporation, by and through its attorneys, complains against VIZIO, Inc. ("Defendant") and alleges as follows:

PARTIES

1. Plaintiff LSI Corporation, dba LSI Logic Corporation ("LSI"), is a corporation organized and existing under the laws of the State of Delaware with a principal place of business in Milpitas, California.

2. On information and belief, Defendant VIZIO, Inc. is a corporation organized and existing under the laws of the State of California with a principal place of business in Irvine, California.

JURISDICTION AND VENUE

3. This is an action arising under the patent laws of the United States, 35 U.S.C. §§ 101, *et seq.* This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a).

4. Venue is proper in this judicial district under 28 U.S.C. §§ 1391(b), (c), and 1400(b).

THE PATENTS

5. United States Patent No. 5,452,006, invented by David R. Auld, entitled "Two-Part Synchronization Scheme For Digital Video Decoders" ("the '006 patent"), was duly and legally issued by the United States Patent and Trademark Office on September 19, 1995. A copy of the '006 patent is attached hereto as Exhibit A.

6. United States Patent No. 5,686,965, invented by David R. Auld, entitled "Two-Part Synchronization Scheme For Digital Video Decoders" ("the '965 patent"), was duly and legally issued by the United States Patent and Trademark Office on November 11, 1997. A copy of the '965 patent is attached hereto as Exhibit B.

7. United States Patent No. 5,398,072, invented by David R. Auld, entitled "Management of Channel Buffer in Video Decoders" ("the '072 patent"), was duly and legally issued by the United States Patent and Trademark Office on March 14, 1995. A copy of the '072 patent is attached hereto as Exhibit C.

8. United States Patent No. 5,379,356, invented by Stephen C. Purcell, David E. Galbi, Frank H. Liao, and Yvonne C. Tse, entitled "Decompression Processor For Video Applications" ("the '356 patent"), was duly and legally issued by the United States Patent and Trademark Office on January 3, 1995. A copy of the '356 patent is attached hereto as Exhibit D.

9. United States Patent No. 5,870,087, invented by Kwok Kit Chau, entitled "MPEG Decoder System and Method Having A Unified Memory For Transport Decode and System Controller Functions" ("the '087 patent"), was duly and legally issued by the United States Patent and Trademark Office on February 9, 1999. A copy of the '087 patent is attached hereto as Exhibit E.

10. United States Patent No. 5,982,830, invented by Greg Maturi and Gregg Dierke, entitled "Hysteretic Synchronization System For MPEG Audio Frame Decoder" ("the '830 patent"), was duly and legally issued by the United States Patent and Trademark Office on November 9, 1999. A copy of the '830 patent is attached hereto as Exhibit F.

11. United States Patent No. 5,918,205, invented by Greg Dierke, entitled "Audio Decoder Employing Error Concealment Technique" ("the '205 patent"), was duly and legally issued by the United States Patent and Trademark Office on June 29, 1999. A copy of the '205 patent is attached hereto as Exhibit G.

12. United States Patent No. 5,864,817, invented by David E. Galbi, entitled "Method For Decoding MPEG Audio Data" ("the '817 patent"), was duly and legally issued by the United States Patent and Trademark Office on January 26, 1999. A copy of the '817 patent is attached hereto as Exhibit H.

BACKGROUND

13. The LSI patents cover inventions relating to various aspects of audio and video decoding.

14. Defendants have made, used, offered for sale, or sold in the United States, or imported into the United States, products covered by the LSI patents and/or products made using the methods claimed in the LSI patents, and/or have induced and/or contributed to such activities by others.

15. LSI has put Vizio on notice of infringement of the LSI patents in accordance with 35 U.S.C. § 287. At a minimum, the filing of this complaint constitutes notice in accordance with that section.

COUNT 1

(The '006 Patent)

16. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

17. Defendant has infringed, contributorily infringed, and/or induced infringement of the '006 patent by its making, using, selling, offering to sell, and/or importation of the Vizio Model No. VF552XVT high-definition television and other products that implement the invention of the '006 patent. On information and belief, Defendant sells, offers to sell, uses, or imports the products covered by the '006 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or imports them into the United States.

18. Defendant's infringement of the '006 patent has damaged and will continue to damage LSI. Defendant's infringement of the '006 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

///

COUNT 2

(The '965 Patent)

19. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

20. Defendant has infringed, contributorily infringed, and/or induced infringement of the '965 patent by its making, using, selling, offering to sell, and/or importation of sale of products including but not limited to the Vizio Model No. VF552XVT high-definition television and other products that implement the invention of the '965 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '965 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

21. Defendant's infringement of the '965 patent has damaged and will continue to damage LSI. Defendant's infringement of the '965 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

COUNT 3

(The '072 Patent)

22. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

23. Defendant has infringed, contributorily infringed, and/or induced infringement of the '072 patent by its making, using, selling, offering to sell, and/or importation of sale of products including but not limited to the Vizio Model No. VO370M high-definition television, the Vizio Model No. VBR120 Blu-ray Player, and other products that implement the invention of the '072 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '072 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

24. Defendant's infringement of the '072 patent has damaged and will continue to damage LSI. Defendant's infringement of the '072 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

COUNT 4

(The '356 Patent)

25. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

26. Defendant has infringed, contributorily infringed, and/or induced infringement of the '356 patent by its making, using, selling, offering to sell, and/or importation of sale of products including but not limited to the Vizio Model No. VO370M high-definition television, the Vizio Model No. VBR120 Blu-ray Player, and other products that implement the invention of the '356 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '356 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

27. Defendant's infringement of the '356 patent has damaged and will continue to damage LSI. On information and belief, Defendant's infringement of the '356 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

COUNT 5

(The '087 Patent)

28. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

29. Defendant has infringed, contributorily infringed, and/or induced infringement of the '087 patent by its making, using, selling, offering to sell, and/or

importation of sale of products including but not limited to the Vizio Model No. VO370M high-definition television, the Vizio Model No. VBR120 Blu-ray Player, and other products that implement the invention of the '087 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '087 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

30. Defendant's infringement of the '087 patent has damaged and will continue to damage LSI. Defendant's infringement of the '087 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

COUNT 6

(The '830 Patent)

31. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

32. Defendant has infringed, contributorily infringed, and/or induced infringement of the '830 patent by its making, using, selling, offering to sell, and/or importation of sale of products including but not limited to the Vizio Model No. VW32L high-definition television and other products that implement the invention of the '830 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '830 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

33. Defendant's infringement of the '830 patent has damaged and will continue to damage LSI. Defendant's infringement of the '830 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

///

COUNT 7

(The '205 Patent)

34. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

35. Defendant has infringed, contributorily infringed, and/or induced infringement of the '205 patent by its making, using, selling, offering to sell, and/or importation of sale of products including but not limited to the Vizio Model No. VBR100 Blu-ray Player and other products that implement the invention of the '205 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '205 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

36. Defendant's infringement of the '205 patent has damaged and will continue to damage LSI. Defendant's infringement of the '205 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

COUNT 8

(The '817 Patent)

37. LSI repeats and realleges the allegations in paragraphs 1-15 of this Complaint.

38. Defendant has infringed, contributorily infringed, and/or induced infringement of the '817 patent by its making, using, selling, offering to sell, and/or importation of sale of products including but not limited to the Vizio Model No. VL470M high-definition television, the Vizio Model No. VBR100 Blu-ray Player, and other products that implement the invention of the '817 patent. Defendant sells, offers to sell, uses, or imports the products covered by the '817 patent in the United States, or sells them to third parties who use them, offer them for sale, or sell them in the United States, or import them into the United States.

39. Defendant's infringement of the '817 patent has damaged and will continue to damage LSI. Defendant's infringement of the '817 patent is willful and has caused and will continue to cause LSI irreparable harm unless enjoined by the Court. LSI has no adequate remedy at law.

PRAYER FOR RELIEF

For these reasons, LSI Corporation respectfully requests that this Court enter judgment in its favor and grant the following relief:

a. Adjudge that VIZIO, Inc. has infringed the LSI patents;

b. Enter an order preliminary and permanently enjoining VIZIO from any further acts of infringement of the LSI patents;

c. Award LSI damages in an amount adequate to compensate LSI for VIZIO's infringement of the LSI patents, but in no event less than a reasonable royalty under 35 U.S.C. § 284, together with pre-judgment and post-judgment interest;

d. Adjudge that VIZIO willfully infringed the LSI patents and that treble damages for such willful infringement should be awarded under 35 U.S.C. § 284;

e. Adjudge this case to be deemed an exceptional case and award LSI the costs of this action and its reasonable attorneys' fees pursuant to 35 U.S.C. § 285; and

///

///

///

///

///

///

///

///

///

f. Such other and further relief as this Court may deem just and appropriate.

Dated: October 19, 2010

CALL & JENSEN
A Professional Corporation
Wayne W. Call, Bar No. 56676
Aaron L. Renfro, Bar No. 255086

By: _____
Aaron L. Renfro

THOMPSON & KNIGHT LLP
Bruce S. Sostek (*Pro Hac Vice Pending*)
Jane Politz Brandt (*Pro Hac Vice Pending*)
Richard L. Wynne, Jr. (*Pro Hac Vice Pending*)

ATTORNEYS FOR DEFENDANT LSI CORPORATION, DBA LSI LOGIC CORPORATION

DEMAND FOR JURY

Pursuant to Rule 38(b) of the Federal Rules of Civil Procedure, LSI Corporation, dba LSI Logic Corporation, demands a trial by jury of all issues triable to a jury.

Dated: October 19, 2010

CALL & JENSEN
A Professional Corporation
Wayne W. Call, Bar No. 56676
Aaron L. Renfro, Bar No. 255086

By: _____
Aaron L. Renfro

THOMPSON & KNIGHT LLP
Bruce S. Sostek (*Pro Hac Vice Pending*)
Jane Politz Brandt (*Pro Hac Vice Pending*)
Richard L. Wynne, Jr. (*Pro Hac Vice Pending*)

ATTORNEYS FOR DEFENDANT LSI CORPORATION, DBA LSI LOGIC CORPORATION

EXHIBIT A

US005452006A

United States Patent [19]

Auld

[11] **Patent Number: 5,452,006**

[45] **Date of Patent: Sep. 19, 1995**

[54] **TWO-PART SYNCHRONIZATION SCHEME FOR DIGITAL VIDEO DECODERS**

[75] Inventor: **David R. Auld**, San Jose, Calif.

[73] Assignee: **LSI Logic Corporation**, Milpitas, Calif.

[21] Appl. No.: **143,316**

[22] Filed: **Oct. 25, 1993**

[51] **Int. Cl.**[6] **H04N 7/12**

[52] **U.S. Cl.** **348/390**; 348/423

[58] **Field of Search** 348/423, 512, 515, 518, 348/390, 384, 385, 387, 501; H04N 7/12, 11/02, 9/475; 370/105.3, 106, 91; 375/365, 366, 368

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,660,079	4/1987	Devimeux et al.	348/423
5,202,761	4/1993	Cooper	348/515
5,351,090	9/1994	Nakamura	348/515
5,351,092	9/1994	Poimboeuf et al.	348/515

OTHER PUBLICATIONS

Moving Pictures Experts Group, Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to about 1.5 Mbit/s (Nov. 25, 1991, 2-11172 Ed. Rev 1) pp. 2-E1.

Primary Examiner—James J. Groody

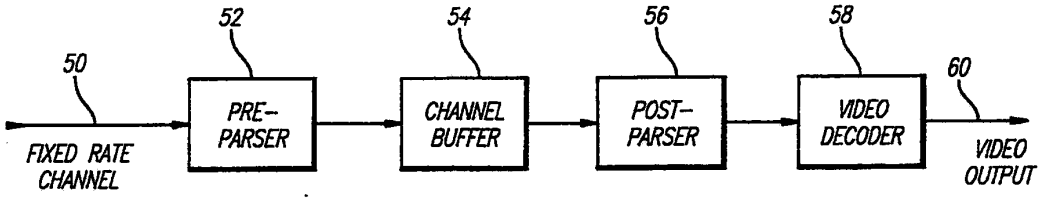
Assistant Examiner—Michael H. Lee

Attorney, Agent, or Firm—Poms, Smith, Lande & Rose

[57] **ABSTRACT**

A novel synchronization scheme for use in connection with digital signal video decoder comprises a pre-parser, a channel buffer, and a post-parser. The pre-parser synchronizes to a multiplexed system bitstream received from a fixed rate channel. The video bitstream component of a multiplexed system bitstream is then extracted and synchronized prior to being transferred bit-serially from the pre-parser to a channel buffer. The post-parser is coupled to the channel buffer and to a video decoder in a series configuration. The post-parser separates the various layers of video data from the video bitstream component. The post-parser performs a translation operation on the video bitstream component and converts the bitstream data into symbol data. The symbol data is subsequently processed by the video decoder so as to reconstruct an originally encoded picture or frame. Preferably, the multiplexed system bitstream data structure conforms to some format agreed upon among video digital businesses involved in transmission and reception. In accordance with one aspect of the present invention, the pre-parser and the post-parser operate independent of each other, and operate at different processing rates.

30 Claims, 6 Drawing Sheets



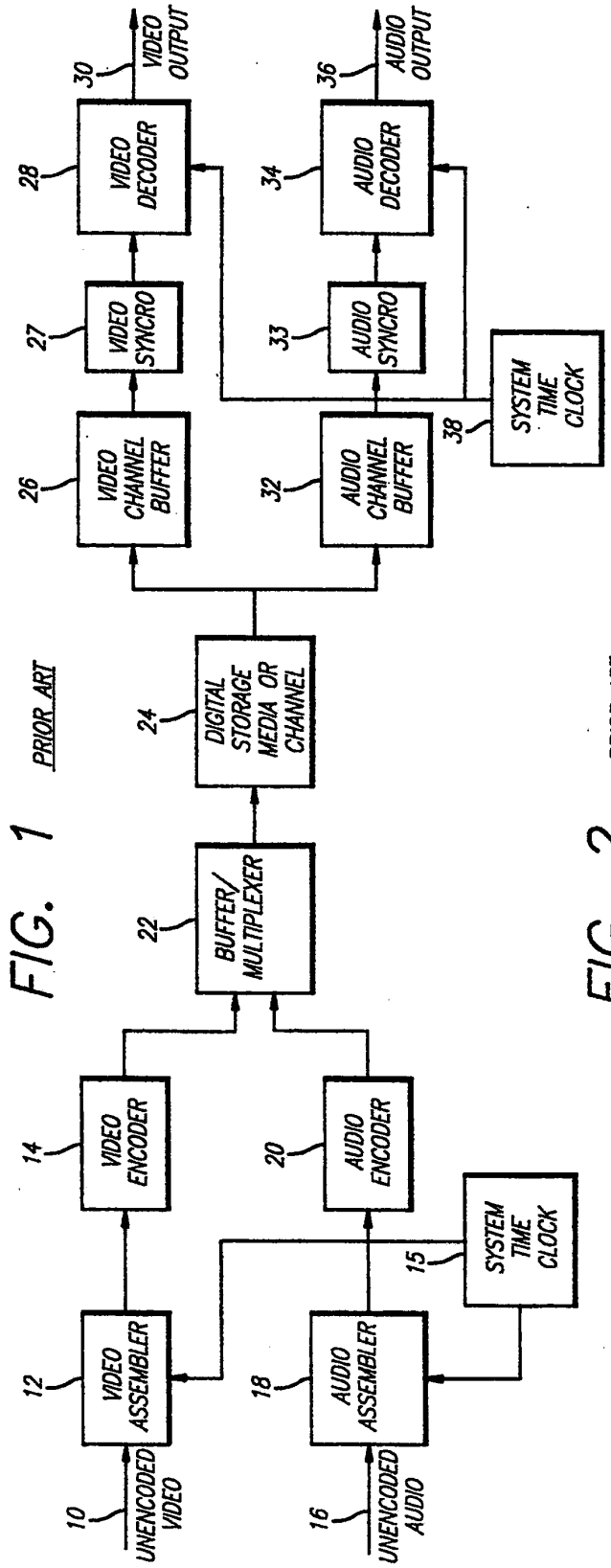


FIG. 1 PRIOR ART

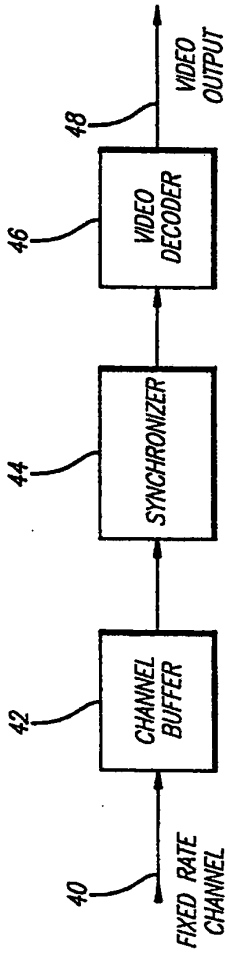


FIG. 2 PRIOR ART

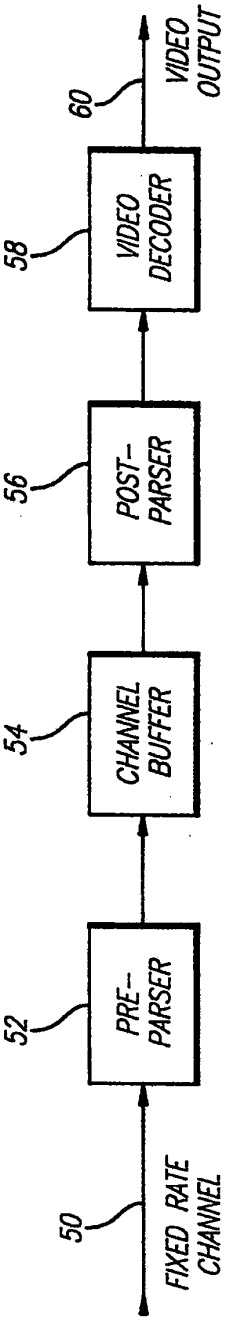

FIXED RATE CHANNEL
50
PRE-PARSER
52
CHANNEL BUFFER
54
POST-PARSER
56
VIDEO DECODER
58
VIDEO OUTPUT
60

FIG. 3

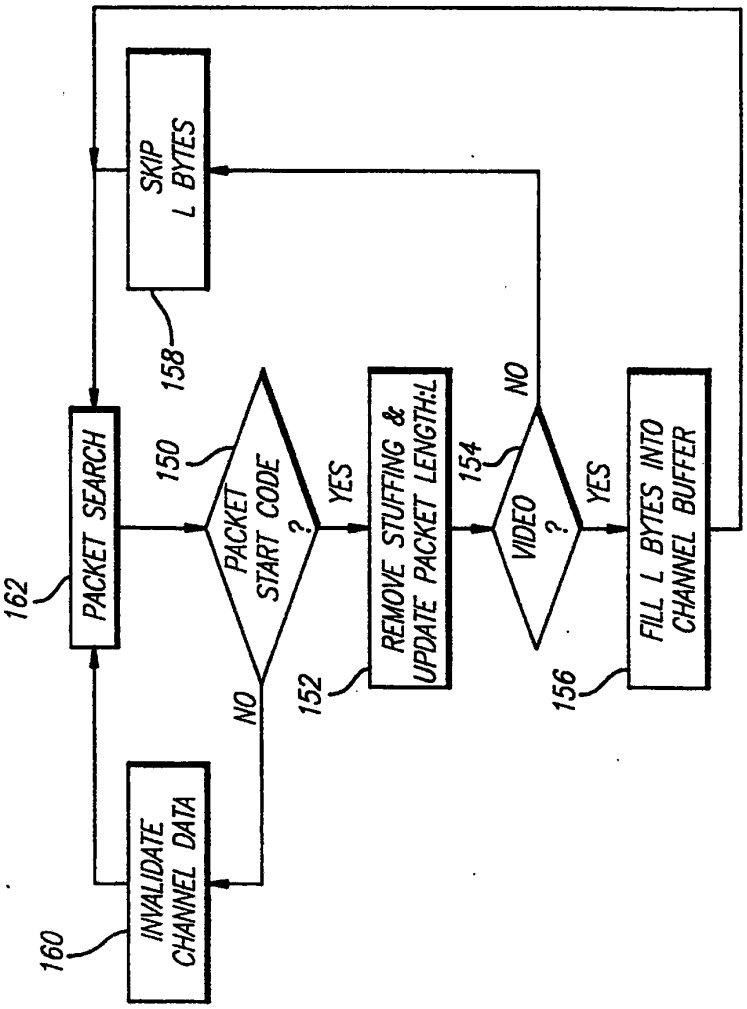

PACKET SEARCH
162
PACKET START CODE ?
150
YES
NO
REMOVE STUFFING & UPDATE PACKET LENGTH:L
152
VIDEO ?
154
YES
NO
FILL L BYTES INTO CHANNEL BUFFER
156
SKIP L BYTES
158
INVALIDATE CHANNEL DATA
160

FIG. 7

FIG. 4

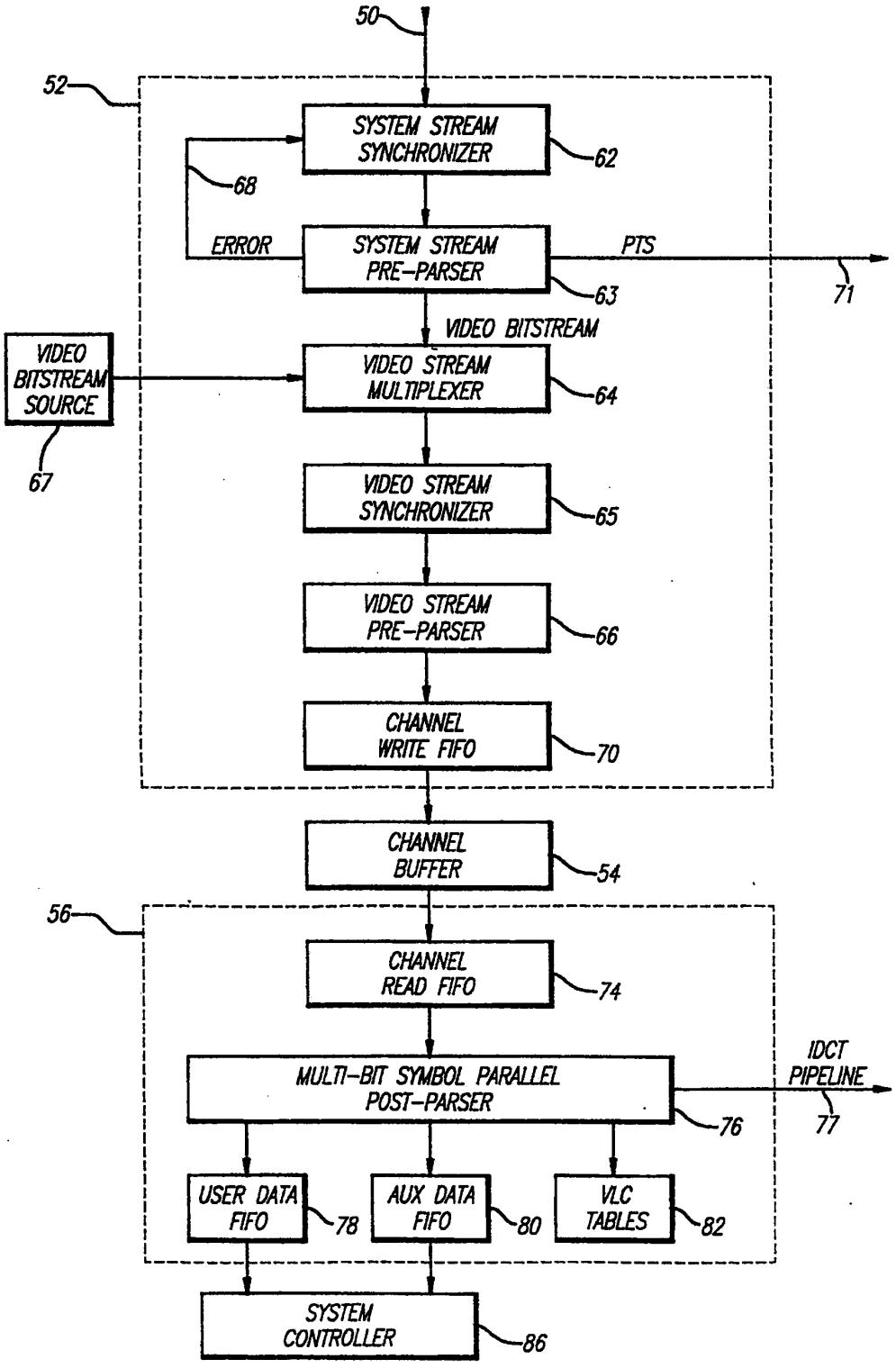

50
52
SYSTEM STREAM SYNCHRONIZER
62
68
ERROR
SYSTEM STREAM PRE-PARSER
PTS
63
71
VIDEO BITSTREAM
VIDEO BITSTREAM SOURCE
VIDEO STREAM MULTIPLEXER
64
67
VIDEO STREAM SYNCHRONIZER
65
VIDEO STREAM PRE-PARSER
66
CHANNEL WRITE FIFO
70
CHANNEL BUFFER
54
56
CHANNEL READ FIFO
74
MULTI-BIT SYMBOL PARALLEL POST-PARSER
IDCT PIPELINE
76
77
USER DATA FIFO
78
AUX DATA FIFO
80
VLC TABLES
82
SYSTEM CONTROLLER
86

FIG. 5

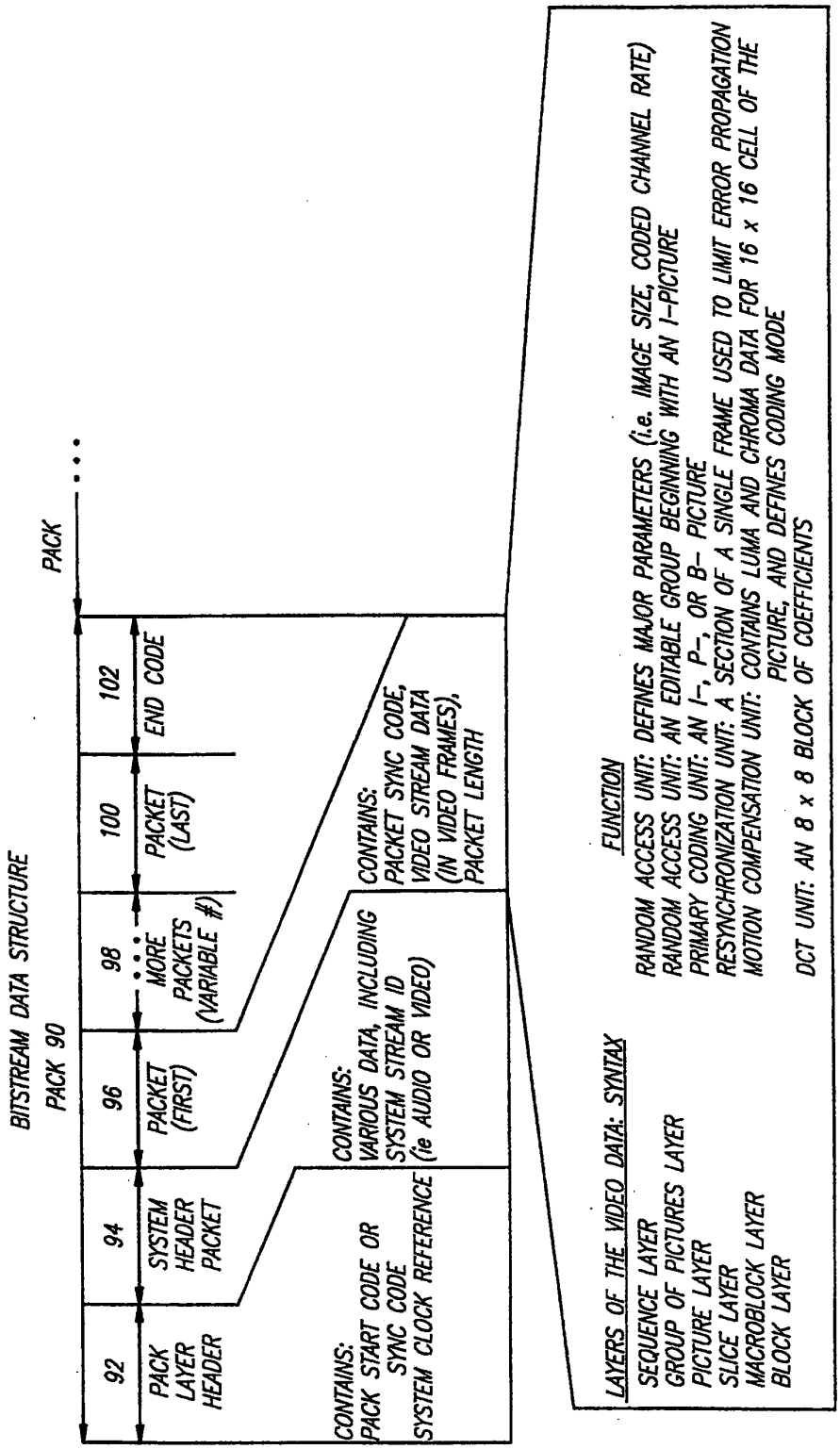
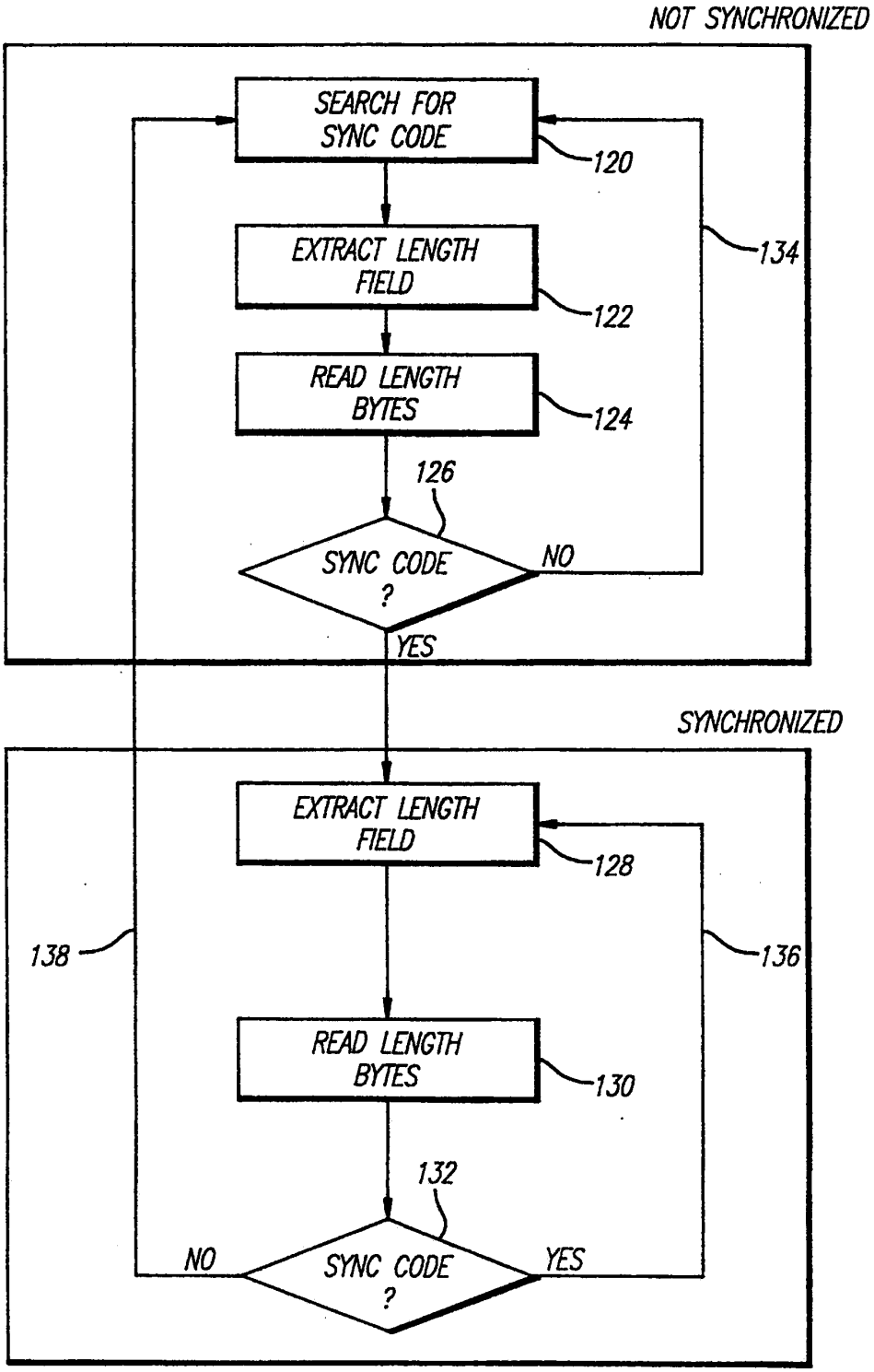

BITSTREAM DATA STRUCTURE
PACK 90
PACK
92
PACK LAYER HEADER
CONTAINS: PACK START CODE OR SYNC CODE SYSTEM CLOCK REFERENCE
94
SYSTEM HEADER PACKET
CONTAINS: VARIOUS DATA, INCLUDING SYSTEM STREAM ID (ie AUDIO OR VIDEO)
96
PACKET (FIRST)
CONTAINS: PACKET SYNC CODE, VIDEO STREAM DATA (IN VIDEO FRAMES), PACKET LENGTH
98
MORE PACKETS (VARIABLE #)
100
PACKET (LAST)
102
END CODE
LAYERS OF THE VIDEO DATA: SYNTAX
SEQUENCE LAYER
GROUP OF PICTURES LAYER
PICTURE LAYER
SLICE LAYER
MACROBLOCK LAYER
BLOCK LAYER
FUNCTION
RANDOM ACCESS UNIT: DEFINES MAJOR PARAMETERS (i.e. IMAGE SIZE, CODED CHANNEL RATE)
RANDOM ACCESS UNIT: AN EDITABLE GROUP BEGINNING WITH AN I-PICTURE
PRIMARY CODING UNIT: AN I-, P-, OR B- PICTURE
RESYNCHRONIZATION UNIT: A SECTION OF A SINGLE FRAME USED TO LIMIT ERROR PROPAGATION
MOTION COMPENSATION UNIT: CONTAINS LUMA AND CHROMA DATA FOR 16 x 16 CELL OF THE PICTURE, AND DEFINES CODING MODE
DCT UNIT: AN 8 x 8 BLOCK OF COEFFICIENTS

FIG. 6


NOT SYNCHRONIZED
SEARCH FOR SYNC CODE
120
EXTRACT LENGTH FIELD
122
READ LENGTH BYTES
124
126
SYNC CODE ?
NO
YES
134
SYNCHRONIZED
EXTRACT LENGTH FIELD
128
READ LENGTH BYTES
130
132
SYNC CODE ?
NO
YES
136
138

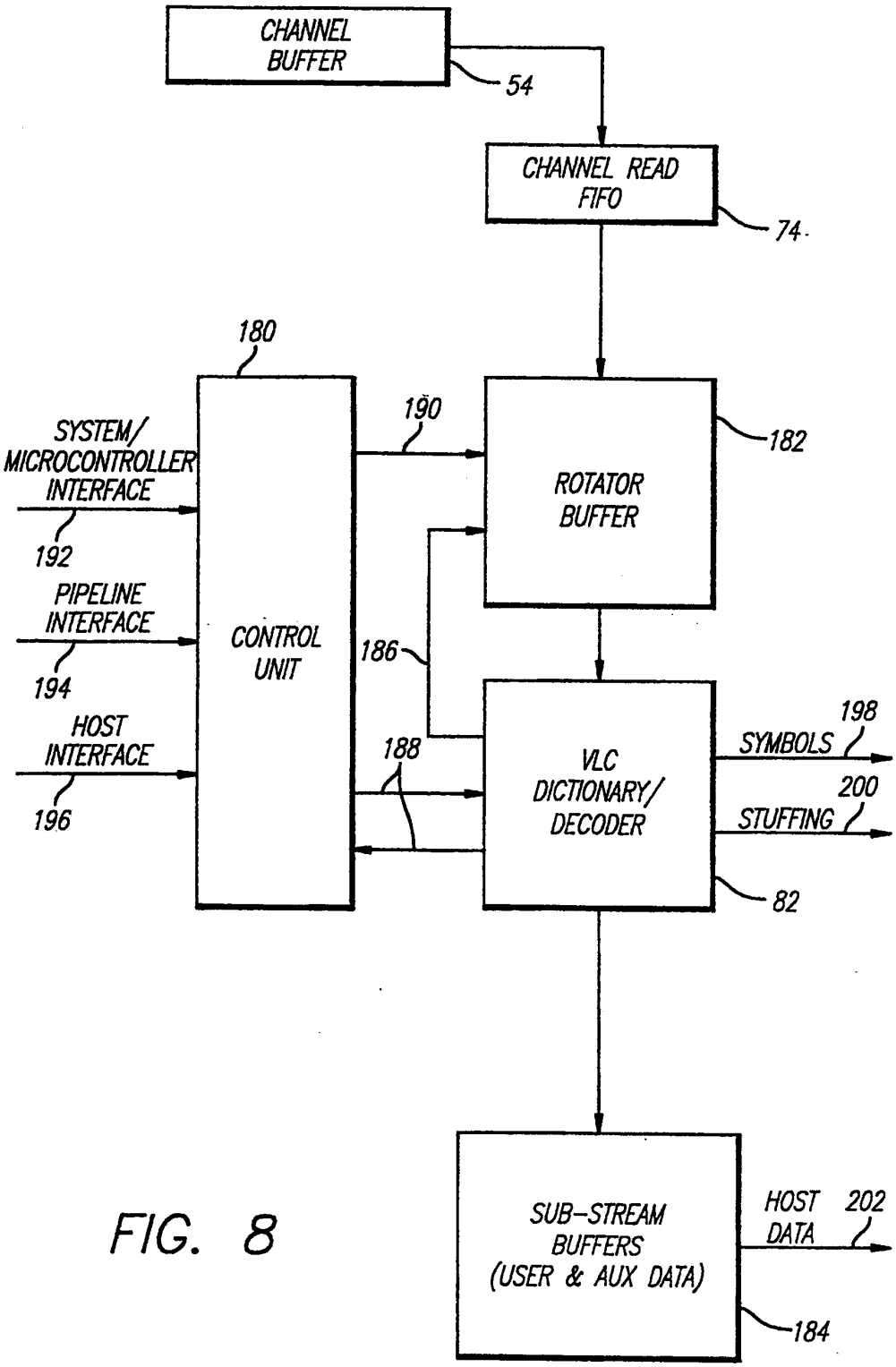

CHANNEL BUFFER
54
CHANNEL READ FIFO
74
180
190
182
SYSTEM/ MICROCONTROLLER INTERFACE
192
PIPELINE INTERFACE
194
HOST INTERFACE
196
CONTROL UNIT
186
ROTATOR BUFFER
188
VLC DICTIONARY/ DECODER
198
SYMBOLS
200
STUFFING
82
SUB-STREAM BUFFERS (USER & AUX DATA)
HOST DATA
202
184

FIG. 8

TWO-PART SYNCHRONIZATION SCHEME FOR DIGITAL VIDEO DECODERS

FIELD OF THE INVENTION

The present invention relates generally to digital video data, and, more particularly to an apparatus and method for synchronizing a video decoder to a compressed digitized video signal.

BACKGROUND OF THE INVENTION

In the communications industry, much attention has been focused on making more effective use of the limited number of transmission channels available for delivering video information and programming to an end user. Various methodologies have been developed to achieve an increase in the number of available transmission channels within the frequency bandwidth that was previously allocated to a single video transmission channel. An increase on the number of available transmission channels would allow the communications industry to reduce costs and to increase broadcast capacity. It has been estimated that a typical cable operator could have the capability to deliver as many as 500 channels to a home viewer.

A dramatic increase in the number of separate program channels that could be broadcast within the currently available transmission bandwidth may be realized by employing a process of compressing and decompressing video signals. Typically, the video and audio signals comprising a video program are converted into a digital format, compressed and encoded in accordance with an established compression algorithm or methodology. The compressed system signal or bitstream, which is understood to include a video portion, an audio portion, and other informational portions, may then be transmitted over existing television channels, cable television channels, satellite communication channels, and the like. A decoder is then typically employed to decompress and decode the received system signal in accordance with the same compression algorithm or methodology previously mentioned. The decoded video information may then be output to a display device, such as a television monitor.

VIDEO ENCODING

Video compression and encoding is typically performed by a video encoder. The video encoder normally implements a selected data compression algorithm that conforms to a recognized standard or specification agreed to among the senders and receivers of digital video signals. One such emerging standard developed by the Moving Pictures Experts Group, is generally referred to as the MPEG International Standard DIS 11172. The MPEG standard defines a format for compressed digital video which supports data rates of about 1 to 1.8 Mbps (Megabits per second), resolutions of about 352 pixels (picture elements) horizontally to about 288 lines vertically, picture rates of about 24 to 30 pictures per second, and several VCR-like viewing options such as Normal Forward, Play, Slow Forward, Fast Forward, Fast Reverse, and freeze.

In order for a video signal to be compressed, it is typically sampled, digitized, and represented by luminance and color difference signals. The MPEG standard specifies that a luminance component (Y) of the video signal be sampled with respect to color difference signals (Cr, Cb) by a ratio of two-to-one (2:1). That is,

for every two samples of the Y component, there is to be one sub-sample each of the Cr and Cb components. It has been determined that the 2:1 sampling ratio is appropriate because the human eye is much more sensitive to luminance (brightness) components than to color components. Video sampling takes place in both the vertical and horizontal directions. Once the video signal is sampled, it is reformatted, for example, into a non-interlaced signal. An interlaced signal is one that contains only part of the picture content (i.e. every other horizontal line) for each complete display scan. A non-interlaced signal, in contrast, is one that contains all of the picture content. After the video signal is sampled and reformatted, the encoder may process it further by converting it to a different resolution in accordance with the image area to be displayed. In doing so, the encoder must determine which type of picture is to be encoded, A picture may be considered as corresponding to a single fame of motion video, or to a frame of movie film. However, different types of picture types may be employed for digital video transmission. The most prevalent picture types are: I- Pictures (Intra-Coded Pictures) which are coded without reference to any other pictures and are often referred to as anchor frames; P-Pictures (Predictive-Coded Pictures) which are coded using motion-compensated prediction from the past I- or P-reference picture, and may also be considered anchor frames; and B-Pictures (Bi-directionally Predictive-Coded Pictures) which are coded using motion compensation from a previous and a future I- or P-Picture.

A typical coding scheme may employ a mixture of I-, P-, and B-Pictures. Typically, an I-Picture may occur every half a second, with two B-Pictures inserted between each pair of I- or P-pictures. I-Pictures provide random access points within the coded sequence of pictures where decoding can begin, out are coded with only a moderate degree of compression. P-Pictures are coded more efficiently using motion compensated prediction from a past I- or P-Picture and are generally used as a reference for further prediction. B-Pictures provide the highest degree of compression but require both past and future reference pictures for motion compensation. B-Pictures are generally not used as references for prediction. The organization of the three picture types in a particular video sequence is very flexible. A fourth picture type is defined by the MPEG standard as a D-Picture, or DC-Picture, which is provided to allow a simple, but limited quality, Fast-Forward mode.

Once the picture types have been defined, the encoder may estimate motion vectors for each 16×16 macroblock in a picture. A macroblock consists of a 16-pixel by 16-line section of the luminance component (Y) and two spatially corresponding 8-pixel by 8-line sections, one for each chrominance component Cr and Cb. Motion vectors provide displacement information between a current picture and a previously stored picture. P-Pictures use motion compensation to exploit temporal redundancy, or lack of motion, between picture frames in the video. Apparent motion between sequential pictures is caused by pixels in a previous picture occupying different positions with respect to the pixels in a current macroblock. This displacement between pixels in a previous and a current macroblock is represented by motion vectors encoded in the MPEG bitstream. Typically, the encoder chooses which picture type is to be used for each given frame. Having

defined the picture type, the encoder then estimates motion vectors for each 16×16 macroblock in the picture. Typically in P-Pictures, one vector is employed for each macroblock, and in B-Pictures, one or two vectors are used. When the encoder processes B-Pictures, it usually reorders the picture sequence so that a video decoder receiving the digital video signal operates properly. Since B-Pictures are usually coded using motion compensation based on previously sent I- or P-Pictures, the B-Pictures can only be decoded after the subsequent reference pictures (an I- or P-Picture) has been decoded. Thus, the sequence of the series of pictures may be re-ordered by the encoder so that the pictures arrive at the decoder in a proper sequence for decoding of the video signal. The decoder may then re-order the pictures in proper sequence for viewing.

As mentioned previously, a macroblock is a 16×16 region of video data, corresponding to 16 pixels in the horizontal direction and 16 display lines in the vertical direction. When sampling of a video frame is performed by the video encoder, every luminance component (Y) of every pixel in the horizontal direction of the frame is captured, and every luminance component of every line in the vertical direction of the frame is captured. However, only every other Cb and Cr chrominance component of the frame is similarly captured. The result is a 16×16 block of luminance components and two 8×8 blocks each of Cr and Cb chrominance component. Each macroblock of video data thus consists of a total of six 8×8 blocks (four 8×8 luminance blocks, one 8×8 Cr block, and one 8×8 Cb block). The spatial picture area covered by four 8×8 blocks of luminance occupies an area equivalent to the region covered by each of the 8×8 chrominance blocks. Since there are half as many chrominance samples needed to cover the same area, they fit into an 8×8 block instead of a 16×16 block.

For a given macroblock of video data, the encoder is programmed to select a coding mode depending on the picture type, the effectiveness of motion compensation in the particular region of the picture, and the nature of the signal within the block. After the coding method is selected, the encoder performs a motion-compensated prediction of the block contents based on past and/or future reference pictures. The encoder then produces an error signal by subtracting the prediction from the actual data in the current macroblock. The error signal is similarly separated into 8×8 blocks (four luminance blocks and two chrominance blocks). A Discrete Cosine Transform (DCT) may then be performed on each block to achieve further compression. The DCT operation converts an 8×8 block of pixel values to an 8×8 matrix of horizontal and vertical coefficients of spatial frequency. An 8×8 block of pixel values can subsequently be reconstructed by a video decoder performing an Inverse Discrete Cosine Transform (IDCT) on the spatial frequency coefficients.

Additional compression is provided through predictive coding since the difference in the average value of neighboring 8×8 blocks tends to be relatively small. Predictive coding is a technique employed to improve compression based on the blocks of pixel information previously operated on by an encoder. A prediction of the pixels values for a block yet to be encoded may be performed by the encoder. The difference between the predicted and actual pixel values may then be computed and encoded. This difference value represents a prediction error which may later be used by a video decoder to correct the information of a predicted block of pixel values.

In addition to the signal compression that is achieved by the encoding process itself, a substantial degree of intentional signal compression is achieved by a process of selecting a quantization step size, where the quantization intervals or steps are identified by an index. The quantization level of the coefficients corresponding to the higher spatial frequencies favors the creation of coefficient values of zero by choosing an appropriate quantization step size such that the human visual perception system is unlikely to notice the loss of a particular spatial frequency, unless the coefficient value for that spatial frequency rises above the particular quantization level chosen. The statistical encoding of the expected runs of consecutive zeroed-valued coefficients of higher-order coefficients accounts for considerable compression gain.

In order to cluster non-zero coefficients early in the series and to encode as many zero coefficients as possible following the last non-zero co-efficient in the ordering, the coefficient sequence is organized in a specified orientation termed zigzag ordering. Zigzag ordering concentrates the highest spatial frequencies at the end of the series. Once the zigzag ordering has been performed, the encoder typically performs "run-length coding" on the special frequency coefficients. This process reduces each 8×8 block of DCT coefficients to a number of events represented by a non-zero coefficient and the number of preceding zero coefficients. Because the high-frequency coefficients are more likely to be zero, run-length coding results in additional video compression.

The encoder may then perform Variable-Length Coding (VLC) on the resulting data. VLC is a reversible procedure for coding that assigns shorter code words to frequent events and longer code words to less frequent events, thereby achieving additional video compression. Huffman encoding is a particularly well-known VLC technique that reduces the number of bits necessary to represent a data set without losing any information. The final compressed video data is then ready to be transmitted to a storage device or over a transmission medium for reception and decompression by a remotely located decoder. The MPEG standard specifies a particular syntax for a compressed bitstream. The MPEG video syntax comprises six layers, each of which supports either a signal processing function or a system function. The MPEG syntax layers correspond to a hierarchical structure. A "sequence" is the top layer of the video coding hierarchy and consists of a header and some number of "Groups-of-Pictures" (GOPs). The sequence header generally initializes the state of the decoder, which allows the decoder to decode any sequence without being affected by past decoding history. A GOP is a random access point, that is, it is the smallest coding unit that can be independently decoded within a sequence. A GOP typically consists of a header and some number of "pictures." The GOP header contains time and editing information. As discussed previously, there are four types of pictures: I-Pictures, P-Pictures, B-Pictures, and D-Pictures. Because of the picture dependencies, the order in which the pictures are transmitted, stored, or retrieved, is not the display order, but rather an order required by the decoder to properly decode the pictures in the bitstream. For example, a typical sequence of pictures, in display order, might be shown as follows:

I	B	B	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

By contrast, the bitstream order corresponding to the given display order would be as follows:

PRIOR ART DECODING SCHEMES

I	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P	B	B
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Because the B-Pictures depend on a subsequent I- or P-Picture in display order, the I- or P-Picture must be transmitted and decoded before the dependent B-Pictures.

Each of the "picture" portions of a GOP in turn consist of a header and one or more "slices." The picture header contains time stamp, picture type, and coding information. A slice consist of an integral number of macroblocks comprising a picture and can be used by a video decoder to recover from decoding errors. Should the bitstream become unreadable within a picture, the decoder will normally be able to recover by waiting for the next slice, without having to drop the entire picture. A slice also includes a header that contains position and quantizer scale information. "Blocks" are the basic coding unit, and the DCT is applied at this block level. Each block typically contains 64 component pixels arranged in an 8×8 order. The pixel values are not individually coded, but are components of the coded block. A macroblock is the basic unit for motion compensation and quantizer scale changes. As discussed previously, each macroblock consists of a header and six component 8×8 blocks: four blocks of luminance, one block of Cb chrominance, and one block of Cr chrominance. The macroblock header contains quantizer scale and motion compensation information.

VIDEO DECODING

Video decoding is the inverse of video encoding and is employed to reconstruct a motion picture sequence from a compressed, encoded bitstream. The data in the bitstream is decoded according to the syntax defined in the data compression standard. The decoder must first identify the beginning of a coded picture, identify the type of picture, then decode each individual macroblock within a particular picture. If there are motion vectors and macroblock types (each of the picture types I, P, and B have their own macroblock types) present in the bitstream, they can be used to construct a prediction of the current macroblock based on past and future reference pictures that the decoder has already stored. Coefficient data is then inverse quantized and operated on by an inverse DCT (IDCT) process that transforms the macroblock data from the frequency domain to the time and space domain.

Once all the macroblocks have been processed by the decoder, the picture reconstruction is complete. If the picture just reconstructed is a reference picture (I-Picture), it replaces the oldest stored reference picture and is used as the new reference for subsequent pictures. As noted above the pictures may also need to be re-ordered before they are displayed in accordance with their display order instead of their coding order. After the pictures are re-ordered, they may then be displayed on an appropriate output device.

A typical and conventional video encoding and decoding scheme is shown in FIG. **1**. An initial video signal **10** is input to a video assembler **12** wherein pictures or frames are assembled. The assembled pictures are then input into a video encoder **14** which compresses the video data in accordance with a selected compression algorithm. Similarly, an initial audio signal **16** is input into an audio assembler **18** wherein audio frames are assembled. The audio frames are then input to an audio encoder **20** for audio compression. A system clock **15** provides time stamp information which is incorporated into the encoded audio and video signals for synchronization purposes. The encoded video signal output from video encoder **14** and the encoded audio signal output from audio encoder **20** are then input to a buffer/multiplexer **22** to produce a multiplexed system bitstream. This multiplexed system bitstream may then be stored in a digital storage media **24** or transmitted along an appropriate transmission channel **24** for later reception, decoding, and display.

At a receiving end, a multiplexed system bitstream is typically written from a fixed rate channel **24** to a video channel buffer **26** and an audio channel buffer **32**. The video portion of the multiplexed system bitstream is extracted by a video synchronizer **27** and then decoded by a video decoder **28**. The decoded video information **30** may then be further processed by a display controller for eventual output to a video monitor. Similarly, an audio synchronizer **33** extracts from an audio buffer **32** the audio portion of the multiplexed system bitstream which may be decoded by an audio decoder **34** and made available for further processing at an audio output **36**. A system time clock **38** provides a time base by which the video decoder **28** and audio decoder **34** remain in synchronization.

In FIG. **2**, there is shown a portion of a conventional video compression/decompression system for processing the video component of an encoded multiplexed system bitstream. An encoded system bitstream, containing both video, audio, and other information, is typically written directly to a channel buffer **42** from a fixed rate channel **40**. A synchronizer **44** receives the multiplexed system bitstream from the channel buffer **42** and pre-processes the system bitstream prior to its being input to a video decoder **46**. Synchronization generally involves finding a unique pattern of bits, often termed sync codes or start codes, in the multiplexed system bitstream, and aligning the bitstream data following the sync code. The various groupings of bits making up the bitstream are often referred to as variable length symbols. These variable length symbols typically represent specific signal information in accordance with the syntax of the encoding and decoding algorithm employed, such as the MPEG standard. In the configuration illustrated in FIG. **2**, both synchronization and decoding

must proceed at a high rate of speed. Since the alignment of symbols in the channel buffer **42** is not known, rapid synchronization is difficult.

A conventional synchronizer **44** must normally perform two typically complex and time consuming operations at a relatively high processing rate. First, the synchronizer **44** must establish the alignment of the symbols from the channel buffer **42** before the decoder **46** can translate the symbols. Translation typically involves matching the symbols against a dictionary or set of look-up tables. The total throughput of the conventional video decoding scheme shown in FIG. **2** is normally limited by the speed of the synchronizer **44**. This can easily lead to overflow or underflow of bitstream data in the channel buffer **42** and starvation of data to the decoder **46**, making the decoding system unreliable. Second, the synchronizer **44** must evaluate and parse the multiplexed system bitstream to extract the relevant coded video data from the non-video data. In such a scheme, the video decoder **46** must analyze a voluminous amount of extraneous data, including the audio portion of the multiplexed system bitstream. The video decoder **46** typically must perform numerous unsuccessful look-ups to the symbol dictionary as it attempts to interpret the non-video data components of the system bitstream. Further, the likelihood of synchronization and decoding errors is significantly increased since the video decoder **46** must analyze and perform error checking on a substantial amount of non-video data. Recovery from decoding and channel errors is therefore generally more difficult and time-consuming. Also, the channel buffer **42** must be configured with sufficient memory capacity to store the non-video data components of the multiplexed system bitstream, even though such non-video components are subsequently discarded by video decoder **46**.

In view of the difficulties discussed above, digital video signal transmission is still highly complex and expensive. Thus, there still exists in the communications industry a keenly felt need to increase the efficiency of video decoders while minimizing both the complexity and cost of effective implementations. The present invention fulfills this need.

SUMMARY OF THE INVENTION

Broadly, and in general terms, one aspect of the present invention provides an increased efficiency in decoding of variable bit-rate coded signals by employment of a novel signal synchronization scheme. A first synchronizer or pre-parser is first employed to synchronize to an initial variable bit-rate coded signal that is received from a transmission channel, and to parse a selected portion of the coded signal from this input. The synchronized portion of the variable bit-rate coded signal is then preferably transferred to a buffer for temporary storage and subsequently decoded by a decoder. Another aspect of the present invention concerns a novel two-part synchronization scheme for synchronizing in a bit-serial fashion to a variable bit-rate coded signal prior to transferring the coded signal to a channel buffer. Preferably, the synchronized portion of the variable bit-rate coded signal stored in the channel buffer is further parsed and translated by a second synchronizer when it is extracted from the channel buffer. The novel two-part synchronization scheme of the present invention ensures that the parsed portion of the variable bit-rate coded signal that is transferred to the channel buffer is properly aligned and synchronized. This pre-synchronization of the parsed portion of the variable bit-rate coded signal significantly reduces the complexity of the operations and circuitry of a decoding system that is used to subsequently decode the variable bit-rate coded signal.

In accordance with one aspect of the present invention, a multiplexed system bitstream, comprising at least an audio bitstream portion and a video bitstream portion, is input to a pre-parser through a fixed rate channel. A system bitstream synchronizer then initially aligns and synchronizes to the received multiplexed system bitstream. The system stream pre-parser also extracts the video bitstream portion from the multiplexed system bitstream. A video stream synchronizer then analyzes the video bitstream for the presence of a video sync code. The various layers comprising the video bitstream are subsequently synchronized and aligned by a video stream pre-parser. Preferably, the parsed video bitstream is then bit-serially written to a channel write FIFO (First-in-First-Out) buffer prior to being transferred to a channel buffer.

Another aspect of the present invention concerns the source of the video bitstream operated on by the video stream pre-parser. Preferably, a video stream multiplexer is provided to switchably select between a number of bitstream sources. In one embodiment of the present invention, a video stream multiplexer can be employed to provide access to a multiplexed system bitstream from a fixed rate channel or other transmission channel, or, alternatively, to a secondary video bitstream source. One such secondary video bitstream source, for example, may be a CD-ROM or other digital storage media.

Still another aspect of the novel two-part synchronization scheme involves a post-parser, which receives the synchronized video bitstream from the FIFO channel buffer. The multi-bit symbol parallel post-parser further parses the various layers comprising the video bitstream and translates the encoded video data to corresponding video symbols. In one embodiment, each layer comprising the video bitstream has a corresponding look-up table or dictionary stored in the parallel post-parser. The resultant video symbols may then be organized and decoded to reconstruct the encoded picture or frame. Other data contained in the video bitstream not directly utilized to perform picture reconstruction are preferably written to a user data FIFO and an auxiliary data FIFO. The data stored in these FIFOs are preferably accessible by a system controller which may extract the data for further processing.

Another aspect of the present invention concerns the independent operation of the pre-parser with respect to operations by the post-parser. The pre-parser operates on a multiplexed system bitstream or video bitstream to produce a parsed and synchronized bitstream which is subsequently bit-serially written to a channel buffer. The post-parser is preferably coupled to the decoder in a parallel fashion. In this regard, the post-parser operates on the synchronized video bitstream received from the channel buffer in a parallel manner and at a significantly higher rate of speed compared to the operational speed of the pre-parser. This de-coupling of the pre-parser and the post-parser results in a significant simplification of the design of the pre-parser, channel buffer, post-parser, and the video decoder. A concomitant reduction in design and production costs with respect to the aforementioned components is also realized.

The novel features of the present invention will be better understood from the following detailed description, considered in connection with the accompanying drawings. It should be expressly understood, however, that the drawings are for purposes of illustration and description only, and are not intended as a definition of the limits of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a conventional video encoding and decoding scheme;

FIG. **2** is a block diagram representative of a conventional synchronization scheme for a digital video decoder;

FIG. **3** is a block diagram of one embodiment of a novel two-part video synchronization scheme in accordance with one aspect of the present invention;

FIG. **4** is a block diagram representative of the synchronization processes of a novel two-part synchronization scheme of the present invention;

FIG. **5** is a graphic representation of the data structure comprising an MPEG compliant multiplexed bitstream;

FIG. **6** is a state diagram illustrating a synchronization process of the novel pre-parser aspect of the present invention;

FIG. **7** is a flow diagram depicting the video parsing operation of the pre-parser in accordance with one embodiment of the present invention; and

FIG. **8** is a block diagram representative of the component blocks comprising the post-parser in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to the figures, and more particularly to FIG. **3**, there is shown a block diagram illustrating a novel video decoding architecture in accordance with one aspect of the present invention. In a preferred embodiment, synchronization to a variable bit-rate coded signal is accomplished in a two-part process performed by a pre-parser **52** and a post-parser **56**. In accordance with this aspect of the present invention, an initial multiplexed system bitstream, preferably transmitted along a fixed rate channel **50**, is coupled to the pre-parser **52**. The pre-parser **52** performs an initial set of operations on the bitstream which is then written to the channel buffer **54**. The post-parser **56** operates on the bitstream data after reading it from the channel buffer **54**.

The pre-parser **52** performs an initial channel synchronization to the multiplexed system bitstream received from the fixed rate channel **50**. Preferably, data is only written to the channel buffer **54** if the pre-parser **52** is synchronized to the bitstream. Additionally, the pre-parser **52** effectively filters the bitstream and removes unwanted digital information in the bitstream corresponding to signal components irrelevant to video image data decoding such as audio and other non-video related information. The filtered video bitstream is then temporarily stored in the channel buffer **54** to await further processing. The operations performed by the post-parser **56** and the video decoder **58** are significantly simplified since the video bitstream stored in the channel buffer **54** is already synchronized and properly aligned.

In FIG. **4**, there is shown a block diagram representative of the processes performed by the novel two-part synchronization of the present invention. Alignment, synchronization, and parsing of a multiplexed system bitstream is performed in two phases; the first by a pre-parser **52**, and second by a post-parser **56**. Disposed between the pre-parser **52** and the post-parser **56** is a channel buffer **54**. In operation, an initial multiplexed system bitstream is input to the pre-parser **52** along a fixed rate channel **50**. Within the pre-parser **52**, a system stream synchronizer **62**, coupled to input channel **50**, synchronizes to the multiplexed system bitstream. A system stream pre-parser **63** then separates the video signal component, commonly referred to as a video bitstream, from the multiplexed system bitstream and extracts the Presentation Time Stamp. The Presentation Time Stamp is a parameter used for subsequent resynchronizing of the audio and video components.

The pre-parser **52** preferably has the capability to process a bitstream from several sources, including a multiplexed system bitstream from a fixed rate channel **50**, and a video bitstream from an external source **67**, such as a CD-ROM or other digital storage media. In one embodiment of the present invention, a video stream multiplexer **64** can be instructed to accept a video bitstream from either the system stream pre-parser **63** or from the external video bitstream source **67**. The video bitstream may then be synchronized by a video stream synchronizer **65**, then aligned and parsed by a video stream pre-parser **66**. The parsed video bitstream output from the video stream pre-parser **62** is then preferably written to a channel write FIFO register or buffer **70** which temporarily stores the video bitstream data prior to writing the data to a channel buffer **54**.

As further illustrated in FIG. **4**, the post-parser **56** receives the parsed video bitstream data from the channel buffer **54**. This parsed video bitstream may first be input to a channel read FIFO **74** within the post,parser **56**. A multi-bit symbol parallel post-parser **76** then accesses the video bitstream data from channel read FIFO **74** for further processing of the data. In accordance with some data encoding approaches, the video bitstream may include various data not directly used in the reconstruction of a video picture or frame. This data may be temporarily stored in a user data FIFO **78** or an auxiliary data FIFO **80**, depending on the nature of the data, for later use in the decoding process or for access by an external system controller **86**. The multi-bit symbol parallel post-parser **76** preferably interacts with a variable length code (VLC) dictionary **82** or set of look-up tables. The variable length codes comprising the video bitstream are parsed by the parallel post-parser **76** by extracting out the various layers comprising the video bitstream for subsequent translation. A number of error checking and recovery procedures are also executed by the parallel post-parser **76**. The resultant parsed video information may then be output to the IDCT (Inverse Discrete Cosine Transform) pipeline **77** for later reconstruction of the picture.

In one embodiment of the present invention, the multiplexed system bitstream is of a type conforming to the MPEG standard. It should be understood, however, that variable bit-rate coding schemes other than those conforming to the MPEG standard or those currently employed in the compression and decompression of video signals may be employed without departing from the scope and spirit of the present invention. The logical structure of a typical MPEG system bitstream is illustrated in FIG. **5**, which represents the highest level MPEG data structure, and is often termed an ISO (International Standards Organization) system stream. A

MPEG ISO system bitstream typically includes a sequence of packs **90**. A typical pack **90**, in turn, includes a sequence pack layer headers **92**, a system header packet **94**, a sequence of packets **96, 98**, and **100**, and an end code **102**. The pack layer header **92** contains a pack start code, or sync code, used for synchronization purposes, and a system clock value. The system header packet **94** contains a variety of information commonly referred to as system housekeeping data, and a system stream ID which is used to differentiate among multiple system streams. A sequence of one or more packets typically contains either encoded audio or encoded video stream data. The final element of a pack **90** is an end code **102**. It should be noted that any one packet **96, 98, 100** may carry either audio or video data, but not both simultaneously. With respect to a packet containing video data, typically a video packet header and one or more video frames is contained within the packet. Further illustrated in FIG. **5** are the hierarchical layers comprising a video packet **96**. The various layers comprising the video data stream include a sequence layer, a group of pictures layer, a picture layer, a slice layer, a macroblock layer, and a block layer. The functions associated with each of these layers is also provided in FIG. **5**. It should be noted that each packet or layer may contain one or more synchronization codes, referred to in the art as sync code, which are necessary to properly synchronize and align the encoded bitstream.

A state diagram showing the system synchronization process is shown in FIG. **6**. Assuming, for example, that the system synchronizer **62** is not synchronized to the bitstream, the system synchronizer **62** searches for a sync code, or start code, comprising the packet layer **96**, as at **120**. The length field of the packet layer **96** is then extracted, and the length of the packet data field is read, as at **122** and **124** respectively. The packet data field of the packet layer **96** may contain emulations of the sync codes of other bitstream data, such as those comprising audio packets. The system synchronizer **62**, however, analyzes the bitstream to determine whether a legitimate video packet sync code has been located, as at **126**. If either the video sync code has not been located or a sync code emulation is located, the system synchronizer **62** will continue to predict the occurrence of a legitimate video sync code in the bitstream. This prediction is based on the length field of the previously occurring header. The system synchronizer **62** then proceeds to repeat the sync code searching process, as at **134**. If a sync code is located within the bitstream, as at **126** and **132**, the length of the field in bytes is extracted and read, as at **128** and **130** respectively. When the sync code is located, as at **126** and **132**, the bitstream is considered synchronized. The system synchronizer **62** will then attempt to locate the sync code of the next packet, as at **136**. If no sync code can be located for the next packet **98**, the system synchronizer **62** begins to search for a sync code and the bitstream transitions from a synchronized state to a non-synchronized state, as at **138**.

One important advantage of the present invention concerns the alignment and synchronization of a multiplexed system bitstream received from a fixed rate channel **50** by the pre-parser **52**. As shown in FIG. **4**, the system stream synchronizer **62** aligns the multiplexed system bitstream with respect to sync words embedded in the system bitstream data. Synchronization involves the alignment of data within a packet relative to the position of a sync word within the bitstream. A typical sync word may be represented by a 24-bit or 32-bit word made up of consecutive zeros and ending with a "1" bit. Alignment involves the recognition of word or byte boundaries within a system bitstream in accordance with the syntax of the selected compression algorithm. The location of sync words distributed within the system bitstream delineates the boundaries between informational word groupings within the packet. These word groupings conform to the particular grammar and syntax rules dictated by the compression algorithm employed for data encoding. Once aligned, the organization of the data structure comprising a particular informational word grouping following a sync code is thus of an established format. In a preferred embodiment, the system stream synchronizer **62** performs synchronization and alignment in a serial fashion, one bit at a time. The synchronized system bitstream is then processed by the system stream pre-parser **63**.

Turning now to FIG. **7**, there is shown a flow diagram illustrating the process of extracting the video bitstream component of a multiplexed system bitstream that is performed by the system stream pre-parser **63**. The system stream pre-parser **63** is intended to pass only the video bitstream component of a multiplexed system bitstream. Any non-video information components, such as the audio portion of the system bitstream, are discarded by the system stream pre-parser **63**. Essentially, the packet sync code, stream ID, and packet length information at the packet layer are used to identify the type of packet being analyzed, and to skip any non-video packets. When a packet sync code has been located, as at **150**, so-called stuffing data is removed, and the length of the packet data field is updated to reflect the decrease in length of the packet word resulting from the removal of the stuffing data, as at **152**. Stuffing data, also referred to as stuffing bits or stuffing bytes, are typically code words that may be inserted into a compressed bitstream to increase the bit rate of the stream to that required by the digital storage or transmission medium. The stuffing data may later be discarded by the decoder during the decoding process.

After removal of the stuffing data and updating the length of the packet data field, the stream ID of the packet is read to determine whether the bitstream is a video bitstream, as at **154**. If the system ID indicates that the bitstream does indeed contain video data, the data representative of the video information is transferred to the channel buffer **54**, as shown at **156**. If the system ID indicates that the packet does not contain video data, the number of bytes comprising the non-video data are skipped, as at **158**, and subsequent packets are searched, as at **162**. If a packet sync code cannot be located, the data in the fixed rate channel **50** is considered invalid and is purged, as at **160**, and subsequent packet searching continues, as at **162**.

While the system synchronizer **62** may initially synchronize incorrectly to a video bitstream, the probability of incorrect synchronization occurring is very low. Incorrect synchronization may occur as a result of data emulating video start codes, or sync codes, in the packet data field of a packet **96**. In order to mis-synchronize, the system synchronizer **62** must find a second sync code after discarding the "length" bits of the packet data field, as at **158** of FIG. **7**. Preferably, the "length" field is represented by a 16-bit symbol, and the sync code is represented by a 32-bit symbol. Assuming that the bits in the data field of the packet layer **96** arrive in random order, that is, "0" and "1" bits are equally prob-

able, the mis-synchronization sequence just described has a probability of $0.5^{48}=3.55\times10^{-15}$. With a bitstream rate of 15 megabytes per second, a mis-synchronization failure would occur every 18.7×10^{6} seconds, which is approximately 217 days. Note that any mis-synchronization failures can occur only if the system synchronizer **62** is not already synchronized to the bitstream. If such a failure is encountered, the system synchronizer **62** can subsequently reacquire synchronization within a period roughly equivalent to the time required to analyze two packets of data.

Another aspect of the present invention concerns a video bitstream synchronizer **65** as shown in FIG. **4**. As discussed previously, the system stream pre-parser **63** passes only the video packet data field of the packet layer **96** to the video stream synchronizer **65**. The video sequence layer, which is the first layer in the video data syntax hierarchy, also contains a header which the video stream synchronizer **65** attempts to locate within the video bitstream. The parsed video data layers may then be written to a channel write FIFO **70** one bit at a time, and subsequently transferred to the channel buffer **54** in a similar manner.

Error propagation due to the emulation of non-video sync codes is generally not a concern with respect to the video bitstream input to the video stream synchronizer **65** and video stream pre-parser **66**. Once the system stream pre-parser **63** parses the video bitstream from the multiplexed system bitstream, non-video sync codes generally are not present in the video bitstream. The emulation of non-video sync codes associated with conventional decoding schemes results from mis-interpreting audio and other non-video bitstream sync codes as legitimate video bitstream sync codes. Since these extraneous non-video bitstreams have, in accordance with the present invention, been stripped from the system bitstream by the system stream pre-parser **63**, it is highly unlikely that such errant sync codes could remain within the video bitstream. Moreover, in the event that the video stream synchronizer **65** locates an unexpected legitimate sync code within the video bitstream, the video stream synchronizer **65** will synchronize to this unexpected sync code. Thus, synchronization of the video bitstream would simply be established earlier than expected, and would not result in the propagation of a synchronization error into the post-parser **56**.

Another important advantage of the two-part synchronization scheme of the present invention concerns the substantial simplification of the post-parser **56** circuitry design, and that of the video decoder **58**. As discussed previously, the pre-parser **52** establishes alignment and synchronization to the video bitstream prior to buffering the video bitstream in the channel buffer **54**. The pre-parser **52** is preferably a bit-serial parser which can accept a multiplexed system bitstream from a serial fixed rate channel **50**. The pre-parser **52** examines the system bitstream one bit at a time, and at the average channel rate of the fixed rate channel **50**. The average channel rate of the fixed rate channel **50** is normally significantly slower than the processing rate of the post-parser **56** and decoder **58**, both of which must operate at a rate sufficient to ensure proper decoding and reconstruction of each encoded frame or picture. By examining the multiplexed system bitstream from a fixed rate channel in a bit-serial fashion, the system stream synchronizer **62** can easily locate the sync codes and ensure proper alignment and synchronization to the multiplexed system bitstream. Although operating in a serial mode, the pre-parser **52** can thus perform all required alignment, synchronization, and parsing functions without adversely affecting the efficiency of the decoding process.

In one embodiment of the present invention, the pre-parser **52** utilizes a channel clock, operating at the rate of the fixed rate channel **50**, and the decoder clock operating at a significantly higher rate. Re-synchronization between the pre-parser **52** and channel buffer **54** preferably occurs at the input of the channel write FIFO **70**. An additional transition clock operating at an intermediate rate may be employed to facilitate re-synchronization as the data is transferred from the relatively low speed pre-parser **52** to the relatively high-speed channel buffer **54**. The channel buffer **54** is also preferably constructed to manage 64-bit words. Rather than performing re-synchronization bit by bit, the re-synchronization occurs with respect to every sixty-four bits of data. In this fashion, re-synchronization occurs at a rate sixty-four times slower than the rate re-synchronization would occur on a word length of one bit.

Another important aspect of the present invention concerns the addition and operation of a multi-bit symbol parallel post-parser **76**. The parallel post-parser **76** parses the layers comprising the video bitstream, beginning with the video sequence layer, by synchronizing on a sync code embedded in the sequence layer. If the parallel post-parser **76** is synchronized to the video sequence layer, and if no errors have been detected, parsing of the group of pictures (GOP) layer then ensues. The parallel post-parser **76** enters the picture layer from the GOP layer if no errors have been detected in the GOP header. Errors found in the bitstream syntax, or other errors occurring from processing of the GOP layer header, are not typically maskable. Such errors are typically flagged as decode errors. Error recovery however, requires waiting for a new GOP sync code or sequence layer sync code. Errors in the header of the picture layer normally have different propagation effects depending on the picture coding type. However, because an error in the picture header would usually result in the transmutation of the picture type to another picture type, all errors in the picture layer header are typically flagged as decode errors. Error recovery requires waiting for a new picture layer, GOP layer, or video sequence layer. Errors in the slice layer and lower syntax layers are also usually concealed by using the last anchor frame, with any motion compensation corrections made by referencing the last correct motion vector.

The multi-bit symbol parallel post-parser **76** is essentially a one-event per cycle parser that performs look-ups and translations with respect to the video bitstream and the VLC tables **82**. A block diagram depicting a multi-bit symbol parallel post-parser **76** is provided in FIG. **8**. The parallel post-parser **76** is preferably comprised of three groups of buffers; a rotator buffer **182** for bit extraction, a VLC dictionary/decoder **82** used for symbol translations, and a control unit **180**. The VLC dictionary/decoder **82** examines a field of contiguous bits from the channel read FIFO **74**. The particular operation mode of the VLC dictionary/decoder **82** and the bit field size are preferably determined by a four-bit operation code input from a microcontroller **192** which is coupled to the control unit **180**. The control unit **180** controls the VLC dictionary/decoder **82** through control lines **188**. The parallel post-parser **76** parses the five layers of video information comprising the video bit-