US 20140056572A1

(19) **United States**

(12) **Patent Application Publication**

Barton et al.

(10) **Pub. No.: US 2014/0056572 A1**

(43) **Pub. Date: Feb. 27, 2014**

(54) **DIGITAL VIDEO RECORDING SYSTEM**

(71) Applicant: **TiVo Inc.**, Alviso, CA (US)

(72) Inventors: **James M. Barton**, Alviso, CA (US); **Alan Moskowitz**, Oakland, CA (US); **Andrew Martin Goodman**, Menlo Park, CA (US); **Tapani Otala**, San Jose, CA (US); **Eric Vannier**, Fremont, CA (US)

(73) Assignee: **TiVo Inc.**, Alviso, CA (US)

(21) Appl. No.: **14/072,678**

(22) Filed: **Nov. 5, 2013**

Related U.S. Application Data

(63) Continuation of application No. 10/418,646, filed on Apr. 18, 2003, now Pat. No. 8,577,205, which is a continuation-in-part of application No. 09/827,029, filed on Apr. 5, 2001, now abandoned, which is a continuation of application No. 09/126,071, filed on Jul. 30, 1998, now Pat. No. 6,233,389.

(60) Provisional application No. 60/374,101, filed on Apr. 19, 2002.

Publication Classification

(51) **Int. Cl.**

H04N 9/808	(2006.01)
G11B 20/10	(2006.01)

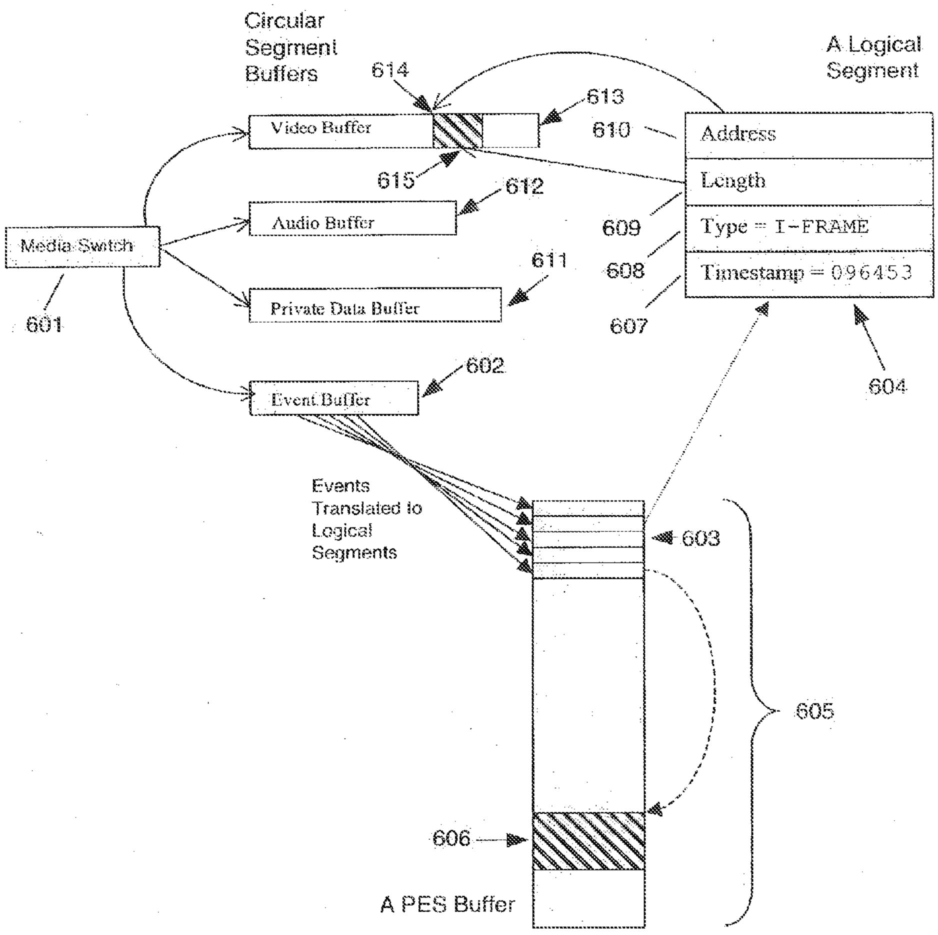
(52) **U.S. Cl.**

CPC ***H04N 9/8081*** (2013.01); ***G11B 20/10527*** (2013.01)

USPC **386/235**

(57) **ABSTRACT**

A digital video recorder (DVR) system processes digital streams and associates stream information with segments of video and audio components. Segments of video and audio components from memory are sent to one or more decoders essentially simultaneously with the storing of the segments of video and audio components from the memory on one or more storage devices. The decoder converts the video and audio components into output signals.



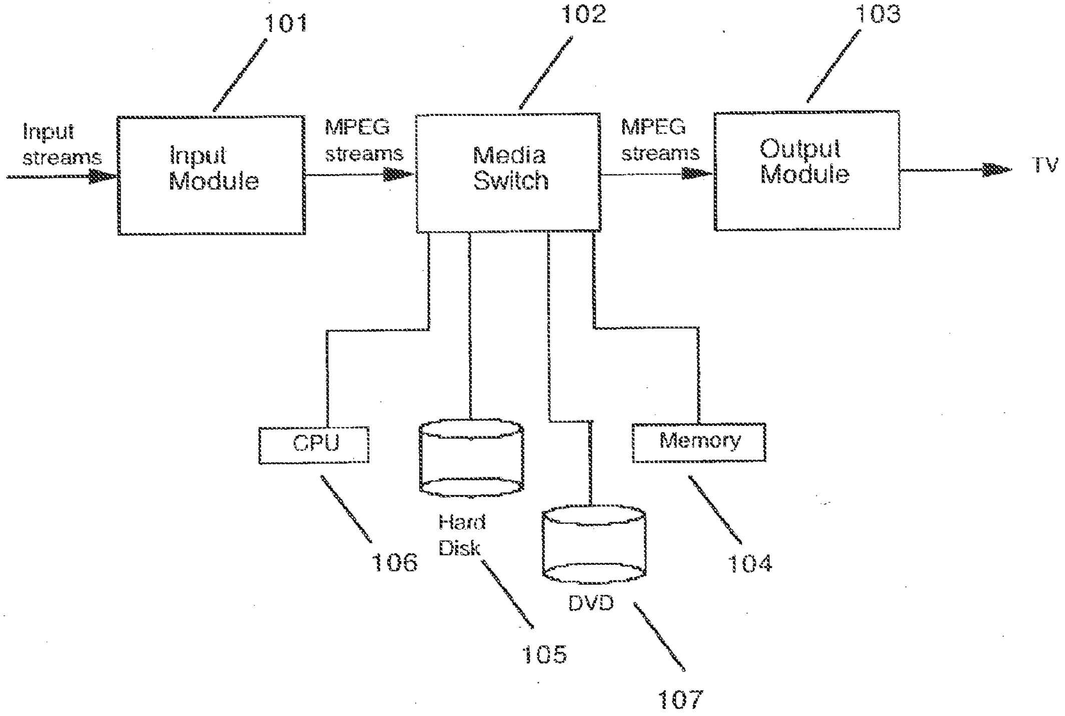

101
102
103
Input streams
Input Module
MPEG streams
Media Switch
MPEG streams
Output Module
TV
CPU
Hard Disk
DVD
Memory
106
105
107
104

Fig. 1

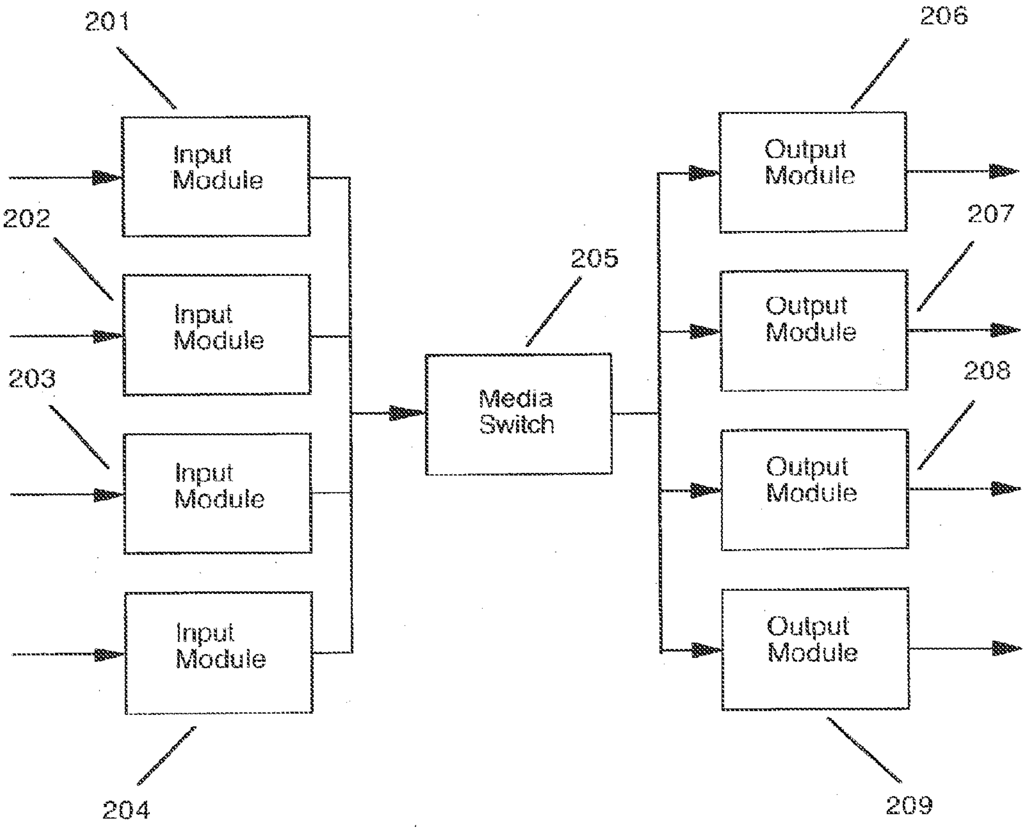

201
202
203
204
Input Module
Input Module
Input Module
Input Module
205
Media Switch
206
207
208
209
Output Module
Output Module
Output Module
Output Module

Fig. 2

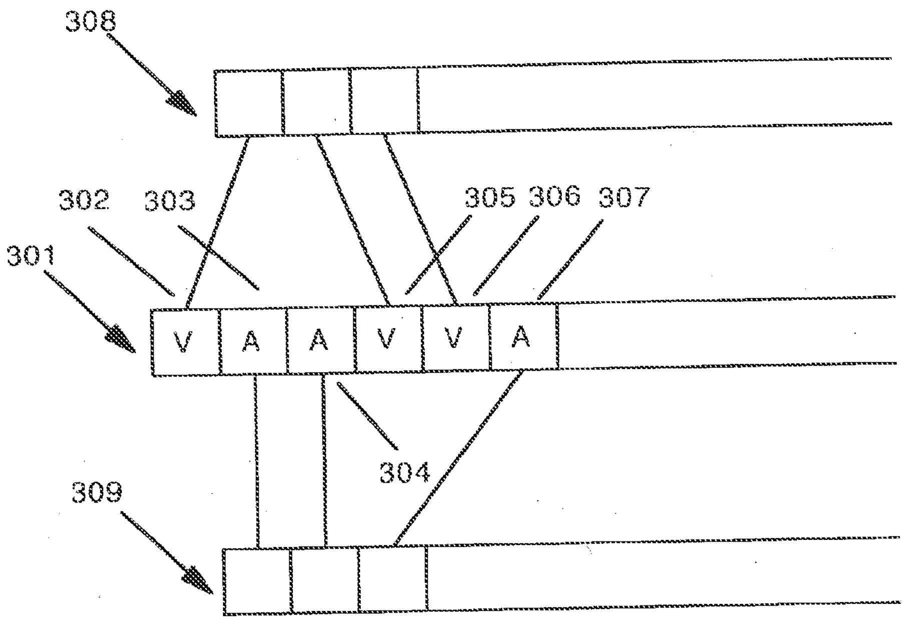


Fig. 3

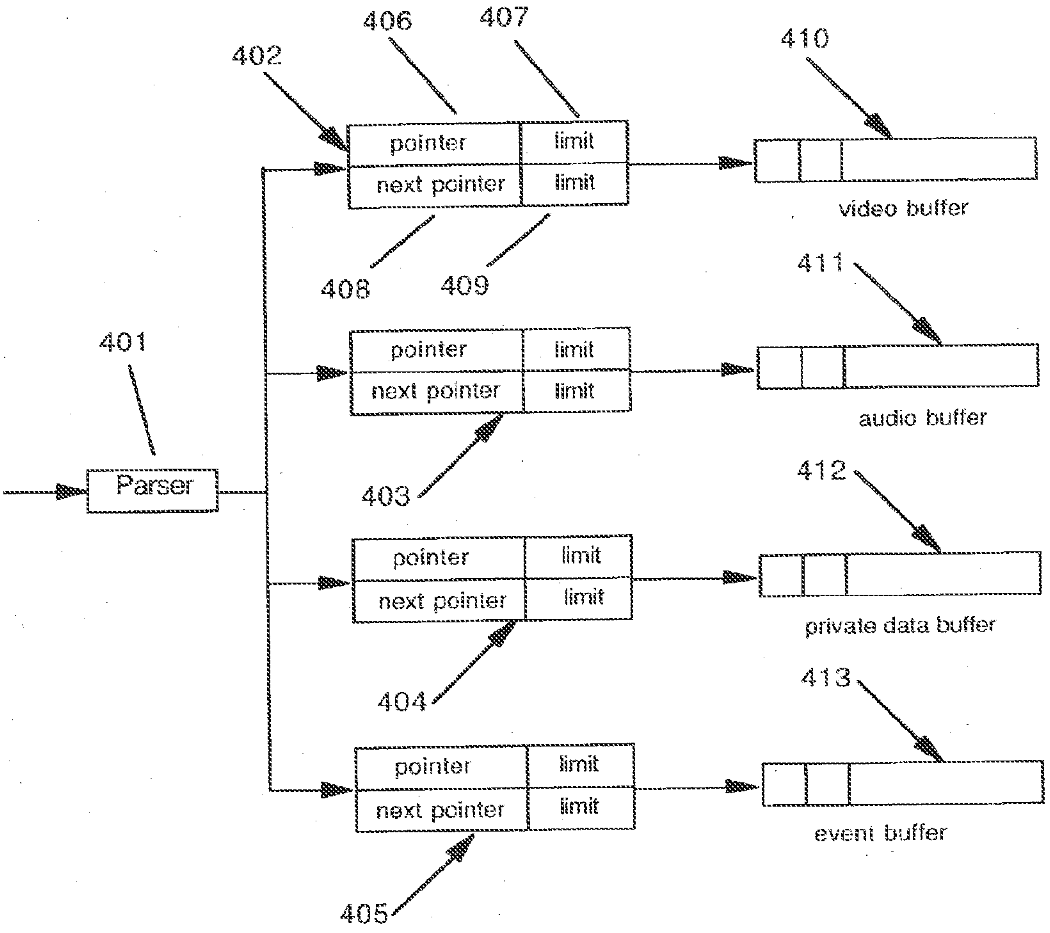

406
407
402
410
pointer
limit
next pointer
limit
video buffer
408
409
411
401
pointer
limit
next pointer
limit
audio buffer
Parser
403
412
pointer
limit
next pointer
limit
private data buffer
404
413
pointer
limit
next pointer
limit
event buffer
405

Fig. 4

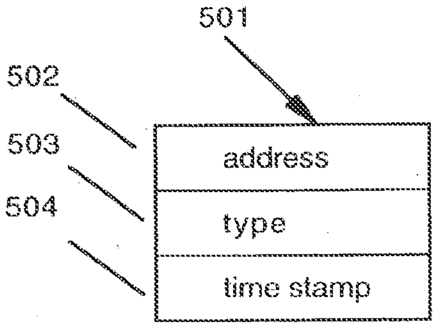


Fig. 5

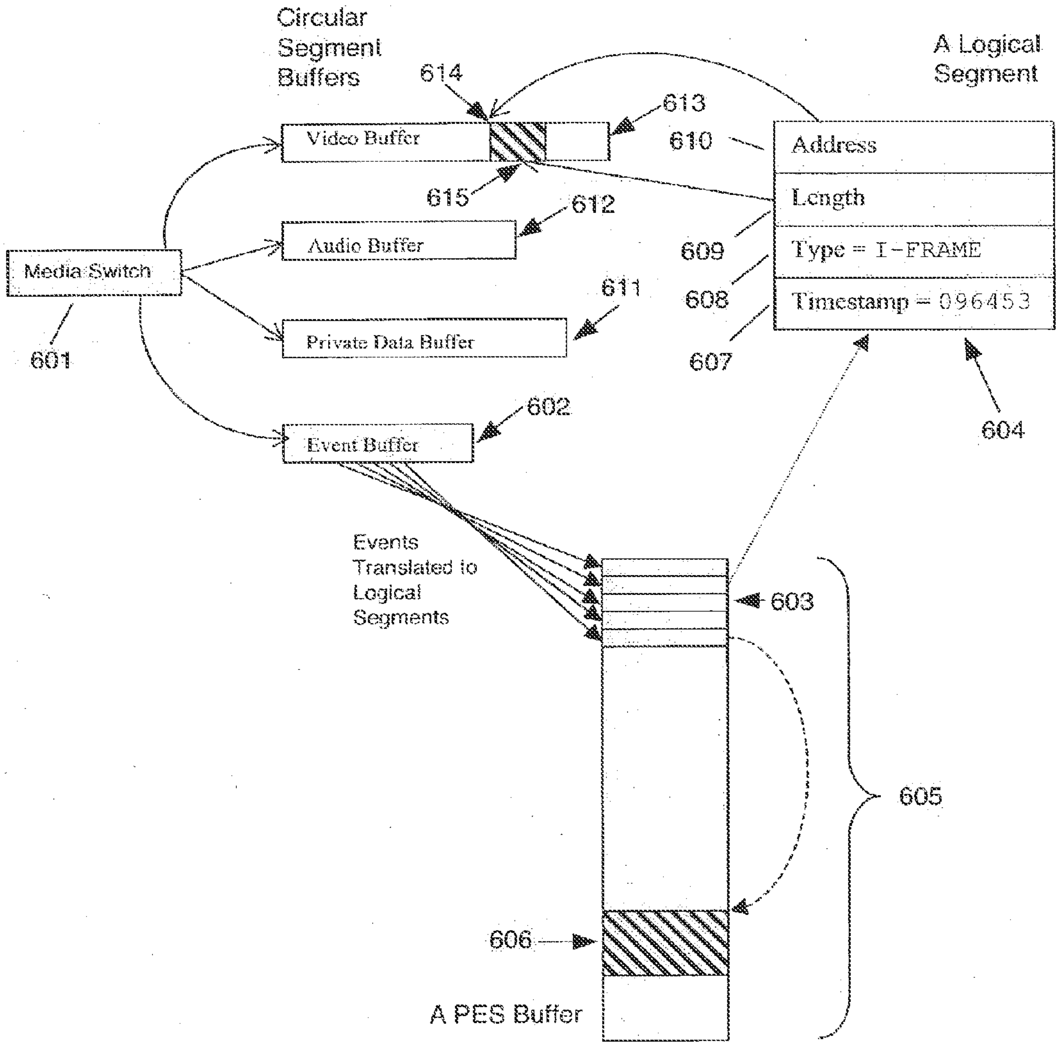

Circular
Segment
Buffers
614
613
Video Buffer
610
Address
A Logical
Segment
615
612
Length
Audio Buffer
Media Switch
609
Type = I-FRAME
611
608
Timestamp = 096453
Private Data Buffer
601
607
602
604
Event Buffer
Events
Translated to
Logical
Segments
603
605
606
A PES Buffer

Fig. 6

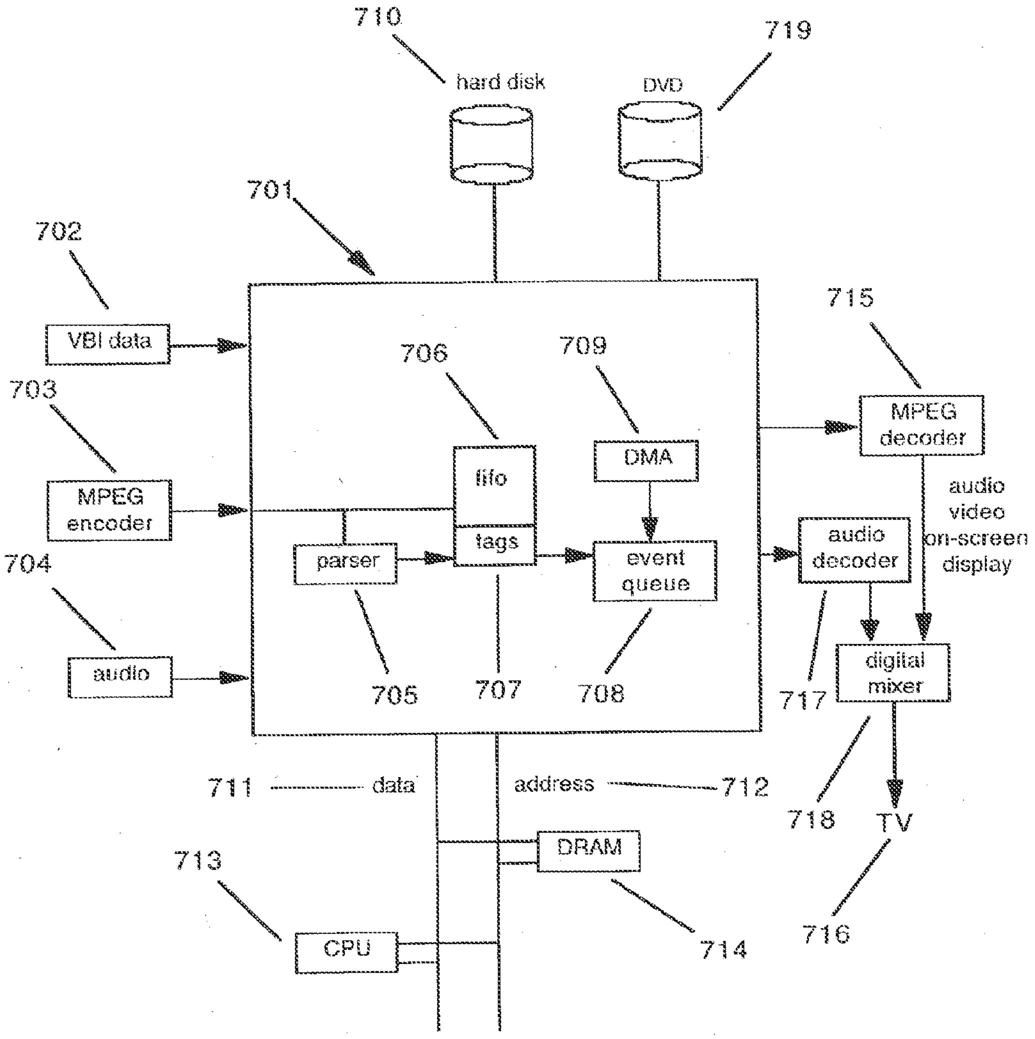

710
hard disk
DVD
719
701
702
VBI data
703
MPEG encoder
704
audio
706
fifo
tags
709
DMA
parser
event queue
705
707
708
715
MPEG decoder
audio
video
on-screen
display
audio decoder
digital mixer
717
718
TV
716
711
data
address
712
DRAM
713
CPU
714

Fig. 7

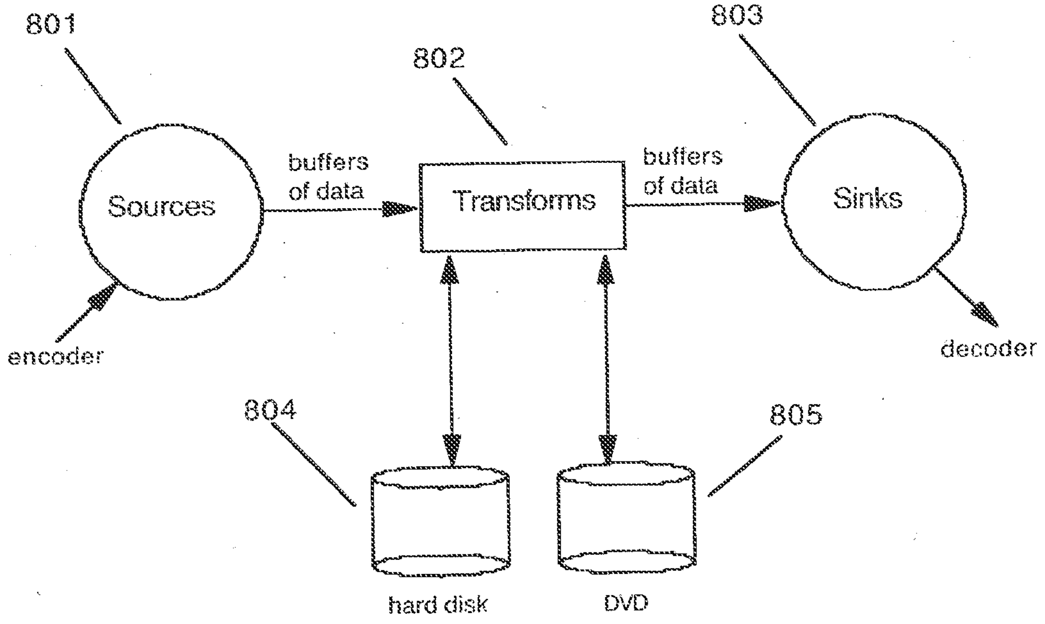

801
802
803
Sources
buffers of data
Transforms
buffers of data
Sinks
encoder
decoder
804
805
hard disk
DVD

Fig. 8

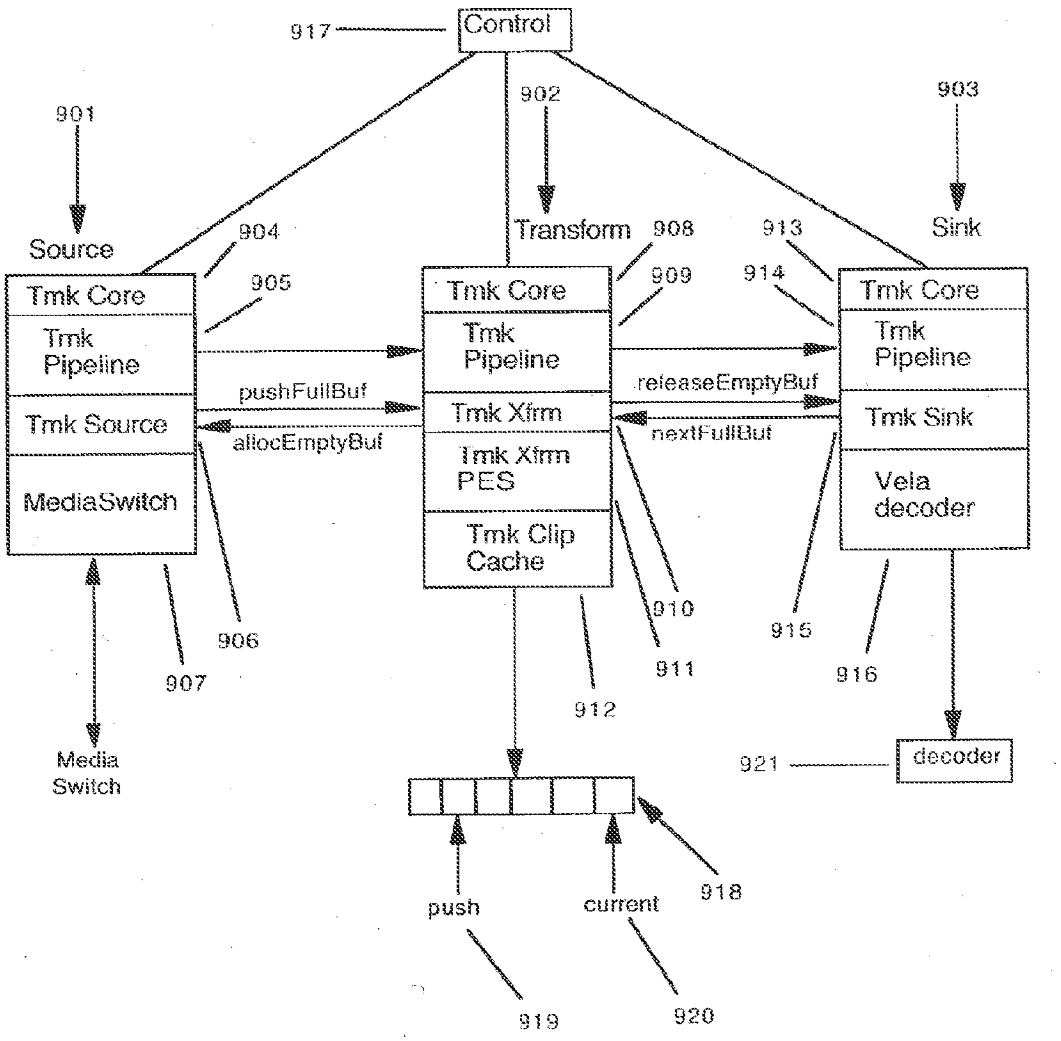

917
Control
901
902
903
Source
Transform
Sink
904
905
908
909
913
914
Tmk Core
Tmk Pipeline
Tmk Source
MediaSwitch
Tmk Core
Tmk Pipeline
Tmk Xfrm
Tmk Xfrm PES
Tmk Clip Cache
Tmk Core
Tmk Pipeline
Tmk Sink
Vela decoder
pushFullBuf
allocEmptyBuf
releaseEmptyBuf
nextFullBuf
906
907
910
911
912
915
916
Media Switch
921
decoder
push
current
918
919
920

Fig. 9

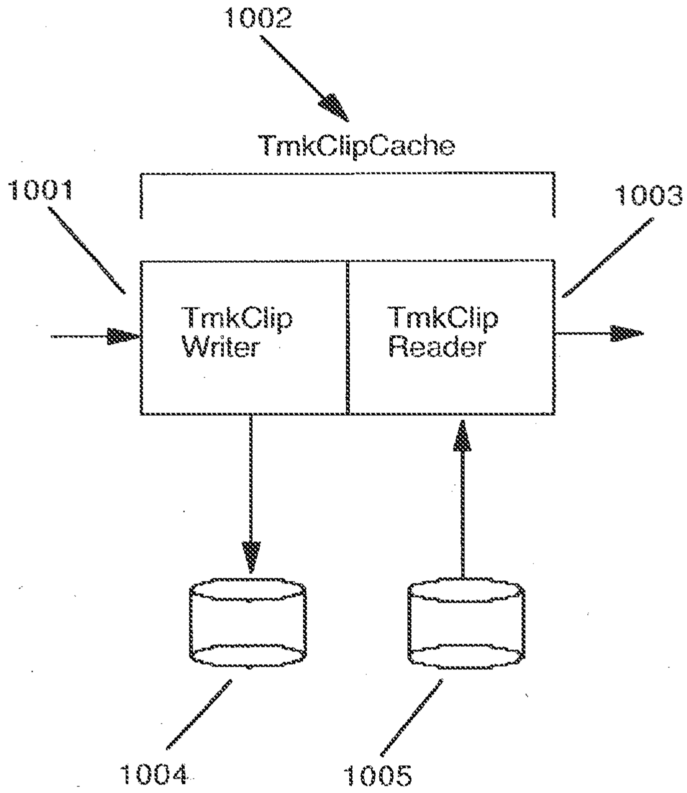

1002
TmkClipCache
1001
1003
TmkClip Writer
TmkClip Reader
1004
1005

Fig. 10

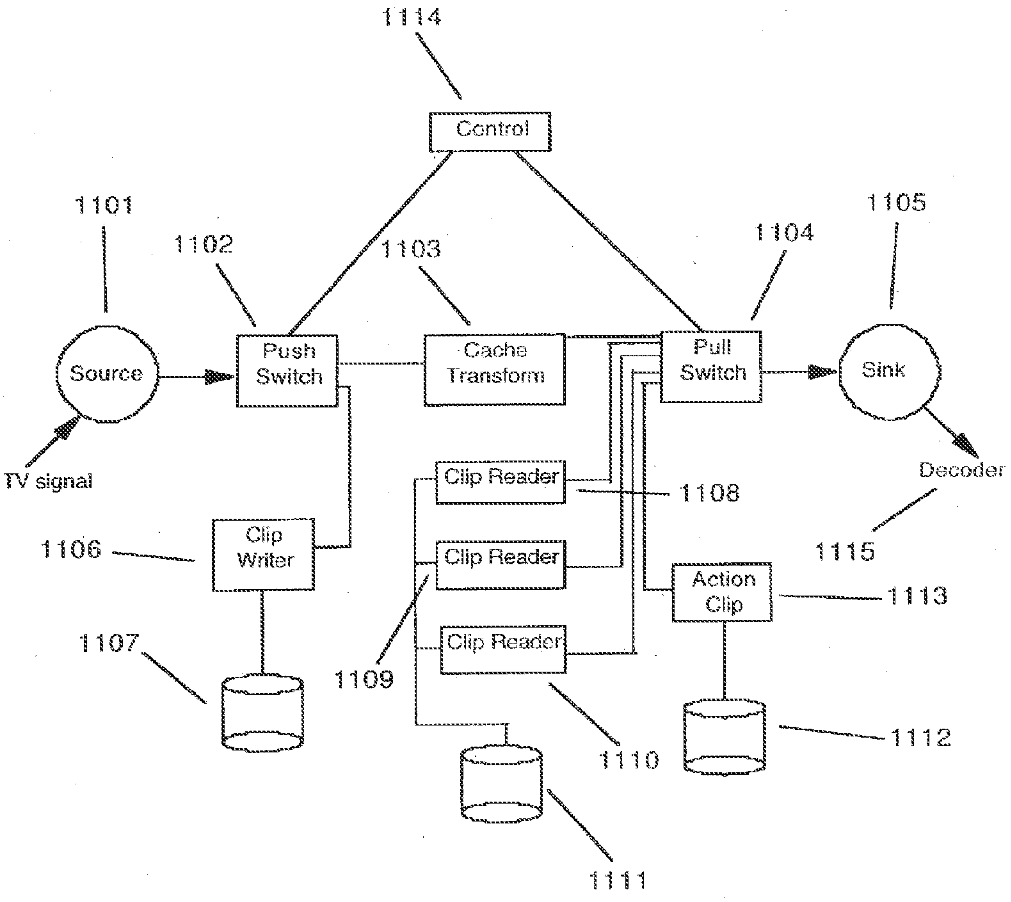

1114
Control
1101
1102
1103
1104
1105
Source
Push Switch
Cache Transform
Pull Switch
Sink
TV signal
Decoder
Clip Reader
1108
1106
Clip Writer
Clip Reader
Action Clip
1113
1115
1107
1109
Clip Reader
1110
1112
1111

Fig. 11

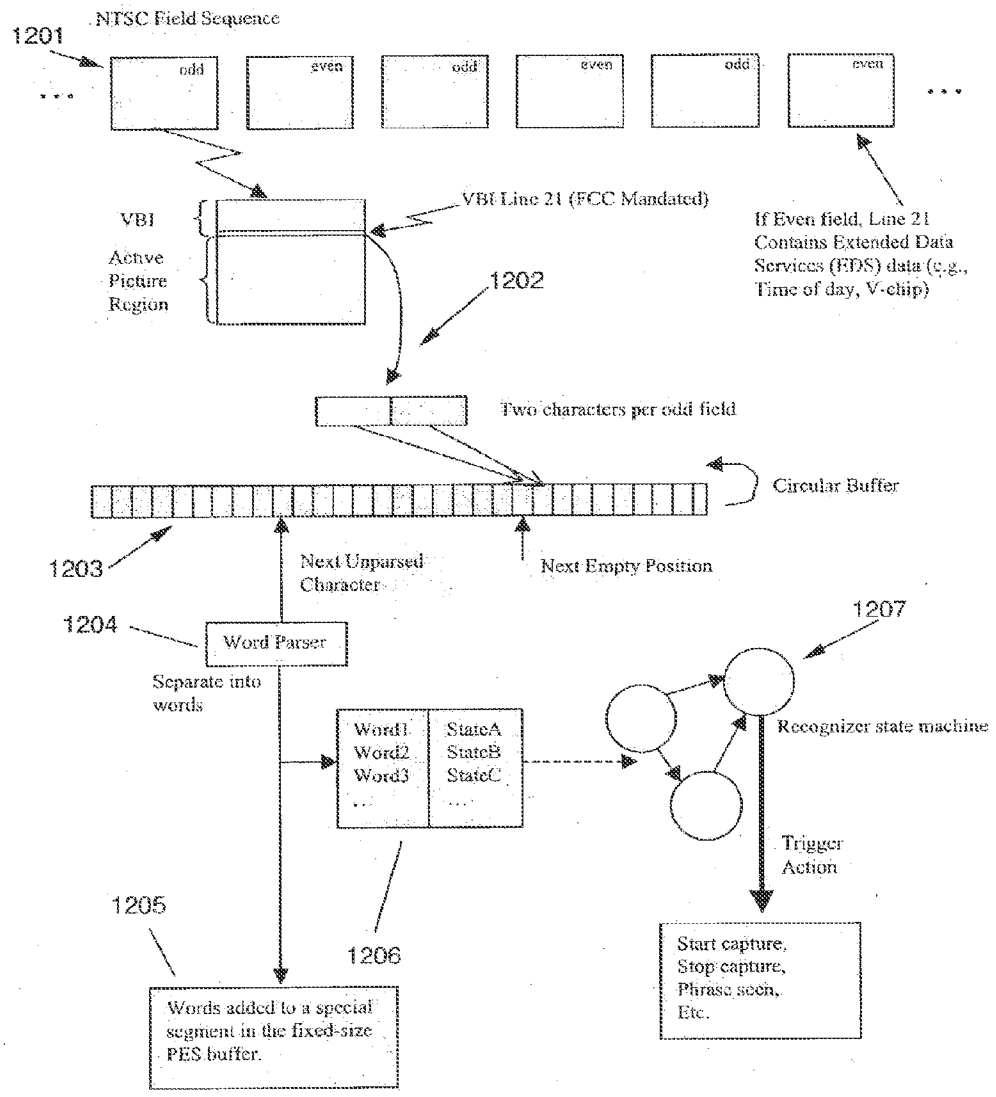

NTSC Field Sequence
1201
odd
even
odd
even
odd
even
VBI
Active Picture Region
VBI Line 21 (FCC Mandated)
If Even field, Line 21 Contains Extended Data Services (EDS) data (e.g., Time of day, V-chip)
1202
Two characters per odd field
Circular Buffer
1203
Next Unparsed Character
Next Empty Position
1204
Word Parser
Separate into words
Word1
Word2
Word3
StateA
StateB
StateC
1207
Recognizer state machine
Trigger Action
1205
1206
Words added to a special segment in the fixed-size PES buffer.
Start capture,
Stop capture,
Phrase seen,
Etc.

Fig. 12

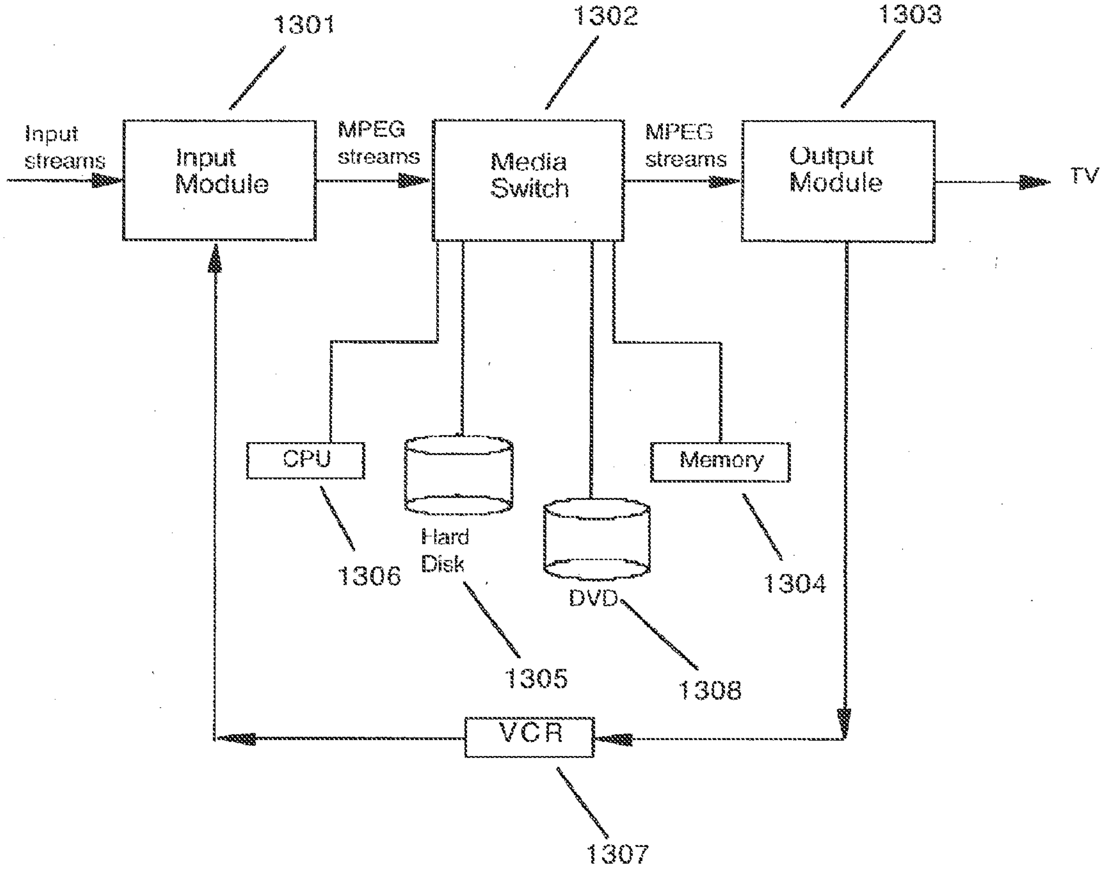

1301
1302
1303
Input streams
Input Module
MPEG streams
Media Switch
MPEG streams
Output Module
TV
CPU
Hard Disk
DVD
Memory
1306
1305
1308
1304
VCR
1307

Fig. 13

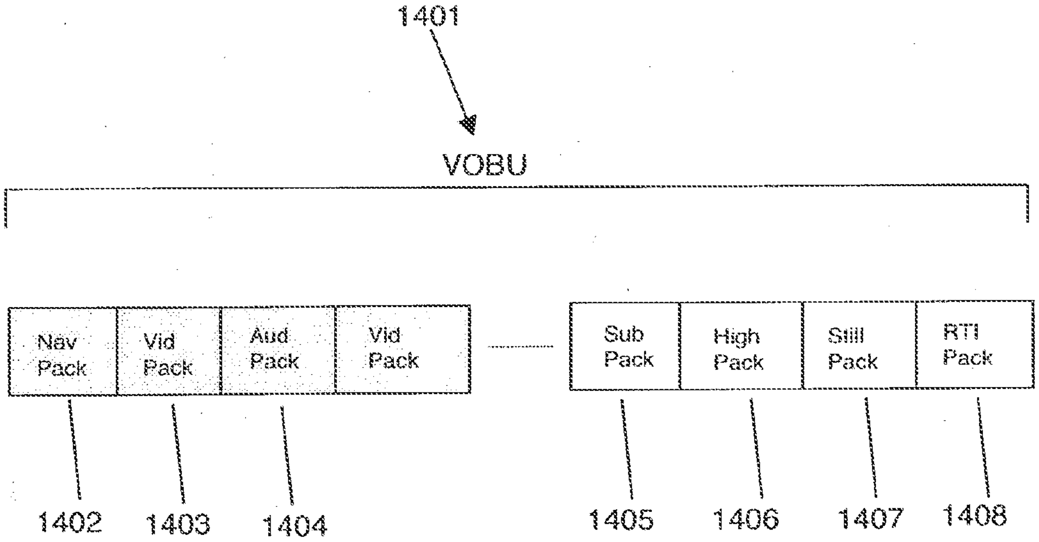


Fig. 14

Prior Art

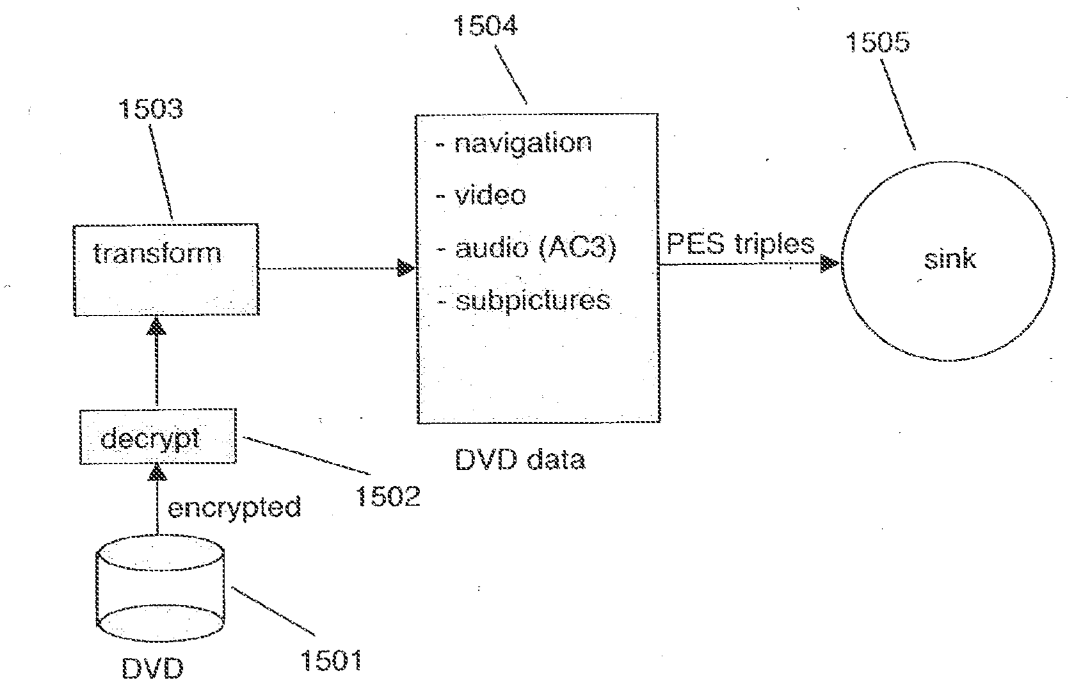
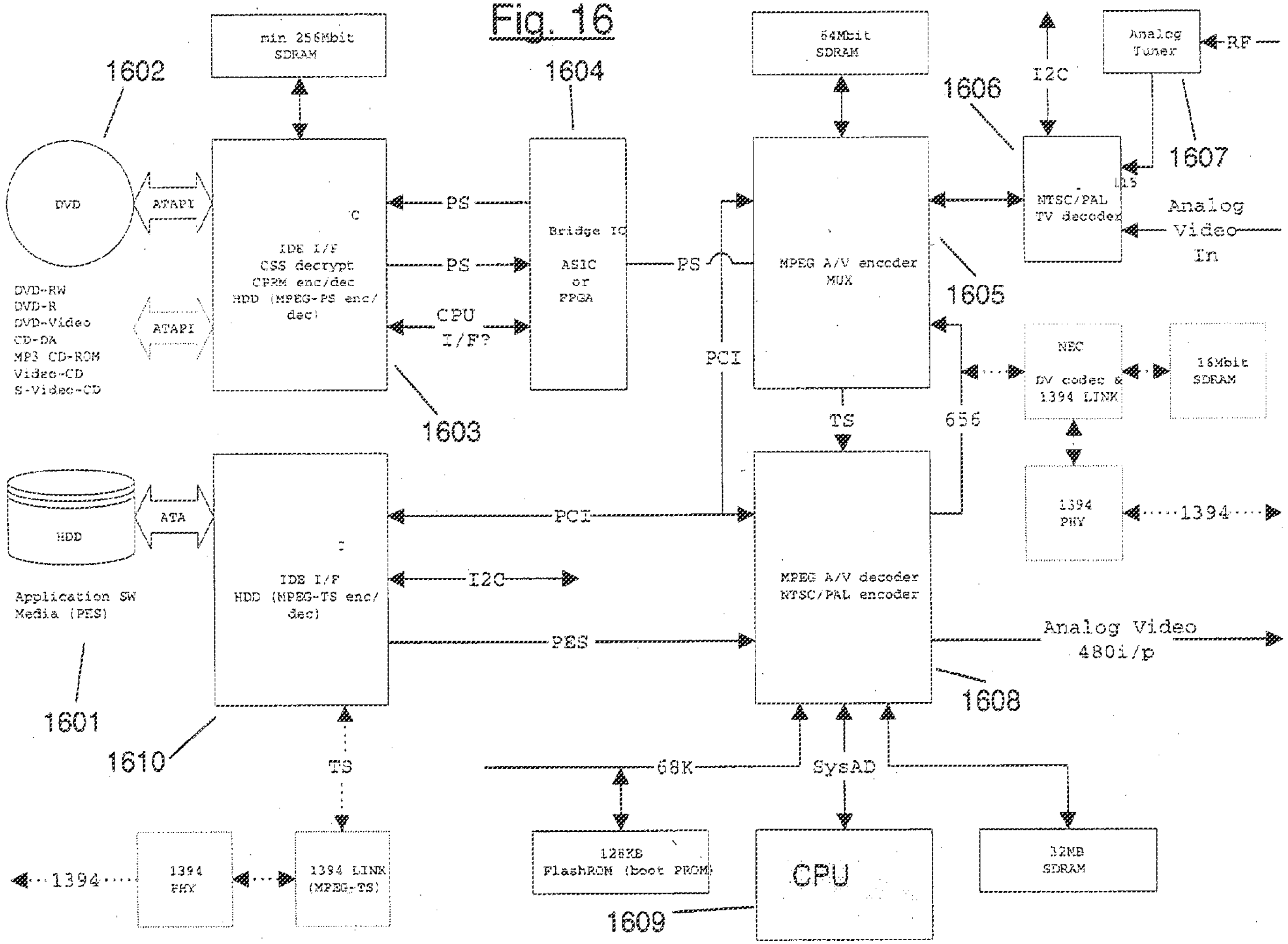

1504
1505
1503
- navigation
- video
- audio (AC3)
- subpictures
transform
PES triples
sink
decrypt
DVD data
encrypted
1502
DVD
1501

Fig. 15

Fig. 16


min 256Mbit SDRAM
1602
1604
64Mbit SDRAM
1606
I2C
Analog Tuner
RF
1607
DVD
ATAPI
IDE I/F
CSS decrypt
CPRM enc/dec
HDD (MPEG-PS enc/ dec)
PS
PS
CPU I/F?
Bridge IC
ASIC or FPGA
PS
MPEG A/V encoder MUX
NTSC/PAL TV decoder
Analog Video In
1605
DVD-RW
DVD-R
DVD-Video
CD-DA
MP3 CD-ROM
Video-CD
S-Video-CD
ATAPI
PCI
NEC
DV codec & 1394 LINK
16Mbit SDRAM
1603
TS
656
1394 PHY
1394
HDD
ATA
IDE I/F
HDD (MPEG-TS enc/ dec)
PCI
I2C
PES
MPEG A/V decoder
NTSC/PAL encoder
Analog Video 480i/p
Application SW
Media (PES)
1601
1610
1608
TS
68K
SysAD
1394
1394 PHY
1394 LINK (MPEG-TS)
128KB FlashROM (boot PROM)
CPU
32MB SDRAM
1609

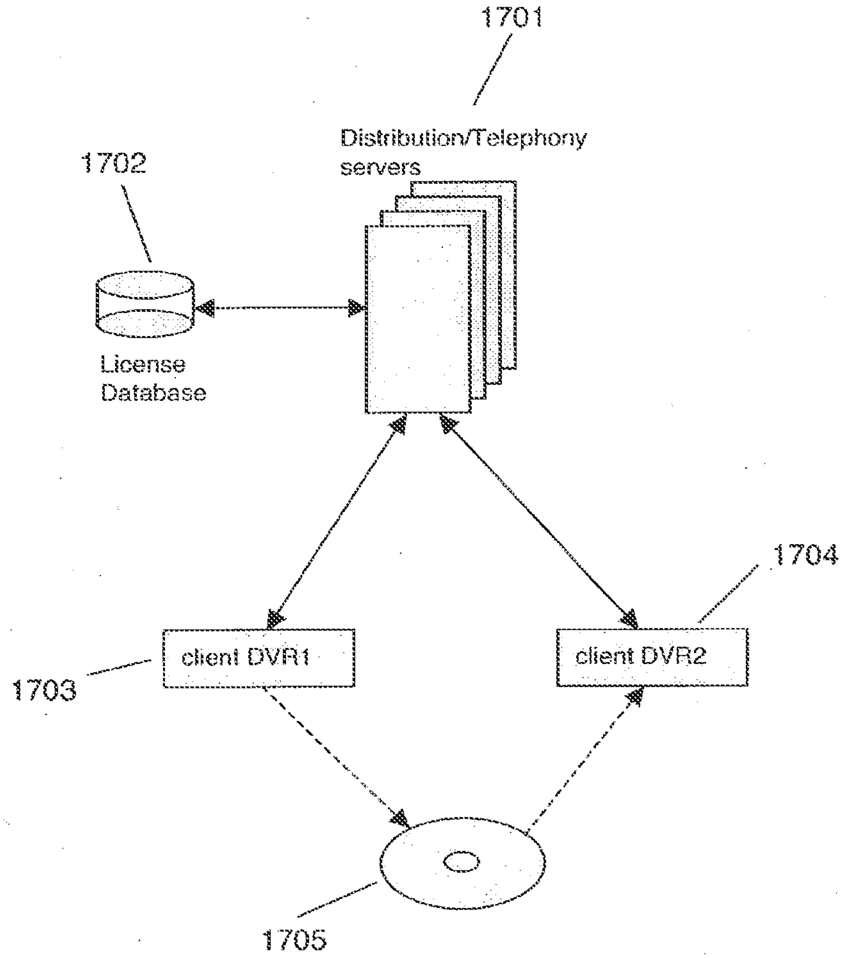

1701
Distribution/Telephony servers
1702
License Database
1704
client DVR1
client DVR2
1703
1705

Fig. 17

DIGITAL VIDEO RECORDING SYSTEM

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims benefit as a Continuation of U.S. application Ser. No. 10/418,646, filed Apr. 18, 2003, now U.S. Pat. No. 8,577,205, issued Nov. 5, 2013, which claims benefit as a Continuation-in-part of U.S. patent application Ser. No. 09/827,029, filed Apr. 5, 2001, which claims benefit as a Continuation of U.S. application Ser. No. 09/126, 071, filed Jul. 30, 1998, now U.S. Pat. No. 6,233,389. U.S. patent application Ser. No. 09/827,029 further claims benefit of U.S. Provisional Patent Application Ser. No. 60/374,101, filed Apr. 19, 2002, the entire contents of the aforementioned are hereby incorporated by reference as if fully set forth herein, under 35 U.S.C. §120. The applicant(s) hereby rescind any disclaimer of claim scope in the parent application(s) or the prosecution history thereof and advise the USPTO that the claims in this application may be broader than any claim in the parent application(s).

BACKGROUND OF THE INVENTION

[0002] 1. Technical Field

[0003] The invention relates to the time shifting of television broadcast signals. More particularly, the invention relates to the real time capture, storage, and display of television broadcast signals and the transfer and playback of said television broadcast signals from a DVD player/recorder.

[0004] 2. Description of the Prior Art

[0005] The Video Cassette Recorder (VCR) has changed the lives of television (TV) viewers throughout the world. The VCR has offered viewers the flexibility to time shift TV programs to match their lifestyles.

[0006] The viewer stores TV programs onto magnetic tape using the VCR. The VCR gives the viewer the ability to play, rewind, fast forward and pause the stored program material. These functions enable the viewer to pause the program playback whenever he desires, fast forward through unwanted program material or commercials, and to replay favorite scenes. However, a VCR cannot both capture and play back information at the same time.

[0007] One approach to solving this problem is to use several VCRs. For example, if two video tape recorders are available, it might be possible to Ping-Pong between the two. In this case, the first recorder is started at the beginning of the program of interest. If the viewer wishes to rewind the broadcast, the second recorder begins recording, while the first recorder is halted, rewound to the appropriate place, and playback initiated. However, at least a third video tape recorder is required if the viewer wishes to fast forward to some point in time after the initial rewind was requested. In this case, the third recorder starts recording the broadcast stream while the second is halted and rewound to the appropriate position. Continuing this exercise, one can quickly see that the equipment becomes unwieldy, unreliable, expensive, and hard to operate, while never supporting all desired functions. In addition, tapes are of finite length, and may potentially end at inconvenient times, drastically lowering the value of the solution.

[0008] The use of digital computer systems to solve part of this problem has been suggested. U.S. Pat. No. 5,371,551 issued to Logan et al., on 6 Dec. 1994, teaches a method for concurrent video recording and playback using a fixed length circular buffer. It presents a microprocessor controlled broadcast and playback device. Said device compresses and temporarily stores video data onto a hard disk. However, this approach is difficult to implement because the processor requirements for keeping up with the high video rates makes the device expensive and problematic. The microprocessor must be extremely fast to keep up with the incoming and outgoing video data. Further, the circular buffer is meant to function as a delay circuit and does not take into account storing an entire program for longer than the length of the circular buffer.

[0009] More recently, digital videos recorders (DVR) have emerged in the marketplace that are based on structures beyond what was previously conceived. One example is U.S. Pat. No. 6,233,389 owned by the Applicant. There are a number of technology trends in force today that are continuing to expand the opportunities for DVR functionality. One such trend is the ability to record data onto a recordable DVD as well as a recordable CD. Currently, DVDs offer a platform that can store up to 15.9 GBs of video and audio data on a single disc. However, a typical recordable DVD can only store approximately five GBs of data.

[0010] Although video rates can be highly variable, the average five GB single-sided DVD translates to approximately two hours of video. As the capacity of recordable DVDs increases, the capability to use the media to store broadcast audio and video program material becomes more cost-effective. Even with the lower storage capacities of current recordable DVDs, recordable and pre-recorded DVDs still offer an opportunity to extend the storage and playback capabilities of a DVR.

[0011] Some VCR manufacturers have combined dual VCRs into one set-top box in order to facilitate easy transfer of content from one VCR to another. Other manufacturers have integrated DVD players into their VCR set-top boxes in an attempt to offer to two playback mediums. However, these approaches do not take advantage of the recordable DVD and further do not take advantage of storing large amounts of information on a hard disk.

[0012] It would be advantageous to provide a digital video recorder system with an integrated DVD recording device that gives a user the ability to record and play back TV broadcast programs and digital video in a set-top box. It would further be advantageous to provide a digital video recorder system with an integrated DVD recording device that integrates a recordable DVD for backing up and playing recorded program material.

SUMMARY OF THE INVENTION

[0013] The invention provides a digital video recorder (DVR) system with an integrated DVD recording device. The invention gives a user the ability to record and play back TV broadcast programs and digital video in a set-top box. In addition, the invention integrates a recordable DVD into the set-top box for backing up and playing recorded program material.

[0014] A preferred embodiment of the invention accepts television (TV) input streams in a multitude of forms, for example, analog forms such as National Television Standards Committee (NTSC) or PAL broadcast, and digital forms such as Digital Satellite System (DSS), Digital Broadcast Services (DBS), or Advanced Television Standards Committee (ATSC). Analog TV streams are converted to an Moving Pictures Experts Group (MPEG) formatted stream for internal transfer and manipulation, while pre-formatted MPEG streams are extracted from the digital TV signal and presented in a similar format to encoded analog streams.

[0015] The invention parses the resulting MPEG stream and stores the stream in temporary buffers. Events are recorded that indicate the type of component that has been found, where it is located, and when it occurred. The program logic is notified that an event has occurred and the data is extracted from the buffers. Indexes within the MPEG stream are determined and saved at predefined intervals. The indexes are stored along with the MPEG stream program material and used to create navigation packets when writing to a DVD inserted into an integrated DVD player/recorder.

[0016] The parser and event buffer decouple the CPU from having to parse the MPEG stream and from the real time nature of the data streams. This decoupling allows for slower CPU and bus speeds which translate to lower system costs.

[0017] The program material are stored on a hard disk. When a program is requested for display from the hard disk or the integrated DVD player/recorder, the program material are extracted from the hard disk or the integrated DVD player/recorder and reassembled into an MPEG stream. The MPEG stream is sent to a decoder. The decoder converts the MPEG stream into TV output signals and delivers the TV output signals to a TV receiver.

[0018] User control commands are accepted and sent through the system. These commands affect the flow of the MPEG stream and allow the user to view stored programs with at least the following special functions: reverse, fast forward, play, pause, index, fast/slow reverse play, and fast/slow play.

[0019] The invention can cache the program material being played from the integrated DVD player/recorder onto the hard disk to allow for better control of the special functions.

[0020] The user can select program material stored on the hard disk to be written to a DVD inserted in the integrated DVD player/recorder. The invention writes the selected program material to the DVD using the associated stored indexes to create navigation packets. The user can also select program material stored on a DVD to be transferred to the hard disk.

[0021] The invention also provides a copyright protection scheme that provides a registration server. The user registers each DVR device that he owns with the registration server. The list of user owned DVRs is distributed to each device on the list. When a DVD is created on a DVR, the DVR's unique serial number or encrypted key is written to the DVD. When the DVD is placed in a DVR, the DVR reads the unique serial number or encrypted key and verifies that the unique serial number or encrypted key is on the user owned DVR list. If the unique serial number or encrypted key is on the list, then the DVR allows reading from the DVD. Otherwise, the DVD is locked out.

[0022] Other aspects and advantages of the invention will become apparent from the following detailed description in combination with the accompanying drawings, illustrating, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] FIG. **1** is a block schematic diagram of a high level view of a preferred embodiment of the invention according to the invention;

[0024] FIG. **2** is a block schematic diagram of a preferred embodiment of the invention using multiple input and output modules according to the invention;

[0025] FIG. **3** is a schematic diagram of an Moving Pictures Experts Group (MPEG) data stream and its video and audio components according to the invention;

[0026] FIG. **4** is a block schematic diagram of a parser and four direct memory access (DMA) input engines contained in the Media Switch according to the invention;

[0027] FIG. **5** is a schematic diagram of the components of a packetized elementary stream (PES) buffer according to the invention;

[0028] FIG. **6** is a schematic diagram of the construction of a PES buffer from the parsed components in the Media Switch output circular buffers;

[0029] FIG. **7** is a block schematic diagram of the Media Switch and the various components that it communicates with according to the invention;

[0030] FIG. **8** is a block schematic diagram of a high level view of the program logic according to the invention;

[0031] FIG. **9** is a block schematic diagram of a class hierarchy of the program logic according to the invention;

[0032] FIG. **10** is a block schematic diagram of a preferred embodiment of the clip cache component of the invention according to the invention;

[0033] FIG. **11** is a block schematic diagram of a preferred embodiment of the invention that emulates a broadcast studio video mixer according to the invention;

[0034] FIG. **12** is a block schematic diagram of a closed caption parser according to the invention;

[0035] FIG. **13** is a block schematic diagram of a high level view of a preferred embodiment of the invention utilizing a VCR as an integral component of the invention according to the invention;

[0036] FIG. **14** is a block schematic diagram of an MPEG standard Video Object Unit (VOBU) layout;

[0037] FIG. **15** is a block schematic diagram of a DVD interface with the invention's transform and sink architecture according to the invention;

[0038] FIG. **16** is a block schematic diagram of an exemplary hardware implementation of a hard disk based digital video recorder with an integrated DVD player/recorder according to the invention; and

[0039] FIG. **17** is a block schematic diagram of a digital video recorder DVD copyright protection scheme according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0040] The invention is embodied in a digital video recorder system with an integrated DVD recording device. A system according to the invention gives a user the ability to record and play back TV broadcast programs and digital video in a set-top box. The invention additionally integrates a recordable DVD into the set-top box for backing up, transferring, and playing recorded program material.

[0041] Referring to FIG. **1**, a preferred embodiment of the invention has an Input Section **101**, Media Switch **102**, and an Output Section **103**. The Input Section **101** takes television (TV) input streams in a multitude of forms, for example, National Television Standards Committee (NTSC) or PAL broadcast, and digital forms such as Digital Satellite System (DSS), Digital Broadcast Services (DBS), or Advanced Television Standards Committee (ATSC). DBS, DSS and ATSC are based on standards called Moving Pictures Experts Group 2 (MPEG2) and MPEG2 Transport. MPEG2 Transport is a standard for formatting the digital data stream from the TV source transmitter so that a TV receiver can disassemble the input stream to find programs in the multiplexed signal. The Input Section **101** produces MPEG streams. An MPEG2 transport multiplex supports multiple programs in the same broadcast channel, with multiple video and audio feeds and private data. The Input Section **101** tunes the channel to a particular program, extracts a specific MPEG program out of it, and feeds it to the rest of the system. Analog TV signals are encoded into a similar MPEG format using separate video and audio encoders, such that the remainder of the system is unaware of how the signal was obtained. Information may be modulated into the Vertical Blanking Interval (VBI) of the analog TV signal in a number of standard ways; for example, the North American Broadcast Teletext Standard (NABTS) may be used to modulate information onto lines **10** through **20** of an NTSC signal, while the FCC mandates the use of line **21** for Closed Caption (CC) and Extended Data Services (EDS). Such signals are decoded by the input section and passed to the other sections as if they were delivered via an MPEG2 private data channel.

[0042] The Media Switch **102** mediates between a microprocessor CPU **106**, hard disk or storage device **105**, DVD player/recorder **107**, and memory **104**. Input streams are converted to an MPEG stream and sent to the Media Switch **102**. The Media Switch **102** buffers the MPEG stream into memory. It then performs two operations if the user is watching real time TV: the stream is sent to the Output Section **103** and it is written simultaneously to the hard disk or storage device **105**.

[0043] The Output Section **103** takes MPEG streams as input and produces an analog TV signal according to the NTSC, PAL, or other required TV standards. The Output Section **103** can also take MPEG streams as input and produce a digital signal that is compatible with digital monitors. The Output Section **103** contains an MPEG decoder, On-Screen Display (OSD) generator, analog TV encoder and audio logic. The OSD generator allows the program logic to supply images which will be overlaid on top of the resulting analog TV signal. Additionally, the Output Section can modulate information supplied by the program logic onto the VBI of the output signal in a number of standard formats, including NABTS, CC and EDS.

[0044] The Media Switch **102** allows a user to playback program material stored on the hard disk **105** and also stored on a DVD inserted into the DVD player/recorder **107**. The user can transfer program material between the hard disk **105** and the DVD player/recorder **107**. The DVD player/recorder **107** records program material stored on the hard disk **105** onto a recordable DVD. The Media Switch **102** also transfers program material from the DVD player/recorder **107** onto the hard disk **105**.

[0045] With respect to FIG. **2**, the invention easily expands to accommodate multiple Input Sections (tuners) **201**, **202**, **203**, **204**, each can be tuned to different types of input. Multiple Output Modules (decoders) **206**, **207**, **208**, **209** are added as well. Special effects such as picture in a picture can be implemented with multiple decoders. The Media Switch **205** records one program while the user is watching another. This means that a stream can be extracted off the disk while another stream is being stored onto the disk.

[0046] Referring to FIG. **3**, the incoming MPEG stream **301** has interleaved video **302**, **305**, **306** and audio **303**, **304**, **307** segments. These elements must be separated and recombined to create separate video **308** and audio **309** streams or buffers. This is necessary because separate decoders are used to convert MPEG elements back into audio or video analog components. Such separate delivery requires that time sequence information be generated so that the decoders may be properly synchronized for accurate playback of the signal.

[0047] The Media Switch enables the program logic to associate proper time sequence information with each segment, possibly embedding it directly into the stream. The time sequence information for each segment is called a time stamp. These time stamps are monotonically increasing and start at zero each time the system boots up. This allows the invention to find any particular spot in any particular video segment. For example, if the system needs to read five seconds into an incoming contiguous video stream that is being cached, the system simply has to start reading forward into the stream and look for the appropriate time stamp.

[0048] A binary search can be performed on a stored file to index into a stream. Each stream is stored as a sequence of fixed-size segments enabling fast binary searches because of the uniform timestamping. If the user wants to start in the middle of the program, the system performs a binary search of the stored segments until it finds the appropriate spot, obtaining the desired results with a minimal amount of information. If the signal were instead stored as an MPEG stream, it would be necessary to linearly parse the stream from the beginning to find the desired location.

[0049] With respect to FIG. **4**, the Media Switch contains four input Direct Memory Access (DMA) engines **402**, **403**, **404**, **405** each DMA engine has an associated buffer **410**, **411**, **412**, **413**. Conceptually, each DMA engine has a pointer **406**, a limit for that pointer **407**, a next pointer **408**, and a limit for the next pointer **409**. Each DMA engine is dedicated to a particular type of information, for example, video **402**, audio **403**, and parsed events **405**. The buffers **410**, **411**, **412**, **413** are circular and collect the specific information. The DMA engine increments the pointer **406** into the associated buffer until it reaches the limit **407** and then loads the next pointer **408** and limit **409**. Setting the pointer **406** and next pointer **408** to the same value, along with the corresponding limit value creates a circular buffer. The next pointer **408** can be set to a different address to provide vector DMA.

[0050] The input stream flows through a parser **401**. The parser **401** parses the stream looking for MPEG distinguished events indicating the start of video, audio or private data segments. For example, when the parser **401** finds a video event, it directs the stream to the video DMA engine **402**. The parser **401** buffers up data and DMAs it into the video buffer **410** through the video DMA engine **402**. At the same time, the parser **401** directs an event to the event DMA engine **405** which generates an event into the event buffer **413**. When the parser **401** sees an audio event, it redirects the byte stream to the audio DMA engine **403** and generates an event into the event buffer **413**. Similarly, when the parser **401** sees a private data event, it directs the byte stream to the private data DMA engine **404** and directs an event to the event buffer **413**. The Media Switch notifies the program logic via an interrupt mechanism when events are placed in the event buffer.

[0051] Referring to FIGS. **4** and **5**, the event buffer **413** is filled by the parser **401** with events. Each event **501** in the event buffer has an offset **502**, event type **503**, and time stamp field **504**. The parser **401** provides the type and offset of each event as it is placed into the buffer. For example, when an audio event occurs, the event type field is set to an audio event and the offset indicates the location in the audio buffer **411**. The program logic knows where the audio buffer **411** starts and adds the offset to find the event in the stream. The address offset **502** tells the program logic where the next event occurred, but not where it ended. The previous event is cached so the end of the current event can be found as well as the length of the segment.

[0052] With respect to FIGS. **5** and **6**, the program logic reads accumulated events in the event buffer **602** when it is interrupted by the Media Switch **601**. From these events the program logic generates a sequence of logical segments **603** which correspond to the parsed MPEG segments **615**. The program logic converts the offset **502** into the actual address **610** of each segment, and records the event length **609** using the last cached event. If the stream was produced by encoding an analog signal, it will not contain Program Time Stamp (PTS) values, which are used by the decoders to properly present the resulting output. Thus, the program logic uses the generated time stamp **504** to calculate a simulated PTS for each segment and places that into the logical segment timestamp **607**. In the case of a digital TV stream, PTS values are already encoded in the stream. The program logic extracts this information and places it in the logical segment timestamp **607**.

[0053] The program logic continues collecting logical segments **603** until it reaches the fixed buffer size. When this occurs, the program logic generates a new buffer, called a Packetized Elementary Stream (PES) **605** buffer containing these logical segments **603** in order, plus ancillary control information. Each logical segment points **604** directly to the circular buffer, e.g., the video buffer **613**, filled by the Media Switch **601**. This new buffer is then passed to other logic components, which may further process the stream in the buffer in some way, such as presenting it for decoding or writing it to the storage media. Thus, the MPEG data is not copied from one location in memory to another by the processor. This results in a more cost effective design since lower memory bandwidth and processor bandwidth is required.

[0054] A unique feature of the MPEG stream transformation into PES buffers is that the data associated with logical segments need not be present in the buffer itself, as presented above. When a PES buffer is written to storage, these logical segments are written to the storage medium in the logical order in which they appear. This has the effect of gathering components of the stream, whether they be in the video, audio or private data circular buffers, into a single linear buffer of stream data on the storage medium. The buffer is read back from the storage medium with a single transfer from the storage media, and the logical segment information is updated to correspond with the actual locations in the buffer **606**. Higher level program logic is unaware of this transformation, since it handles only the logical segments, thus stream data is easily managed without requiring that the data ever be copied between locations in DRAM by the CPU.

[0055] A unique aspect of the Media Switch is the ability to handle high data rates effectively and inexpensively. It performs the functions of taking video and audio data in, sending video and audio data out, sending video and audio data to disk or DVD and extracting video and audio data from the disk or DVD on a low cost platform. Generally, the Media Switch runs asynchronously and autonomously with the microprocessor CPU, using its DMA capabilities to move large quantities of information with minimal intervention by the CPU.

[0056] Referring to FIG. **7**, the input side of the Media Switch **701** is connected to an MPEG encoder **703**. There are also circuits specific to MPEG audio **704** and vertical blanking interval (VBI) data **702** feeding into the Media Switch **701**. If a digital TV signal is being processed instead, the MPEG encoder **703** is replaced with an MPEG2 Transport Demultiplexor, and the MPEG audio encoder **704** and VBI decoder **702** are deleted. The demultiplexor multiplexes the extracted audio, video and private data channel streams through the video input Media Switch port.

[0057] The parser **705** parses the input data stream from the MPEG encoder **703**, audio encoder **704** and VBI decoder **702**, or from the transport demultiplexor in the case of a digital TV stream. The parser **705** detects the beginning of all of the important events in a video or audio stream, the start of all of the frames, the start of sequence headers—all of the pieces of information that the program logic needs to know about in order to both properly play back and perform special effects on the stream, e.g. fast forward, reverse, play, pause, fast/slow play, indexing, and fast/slow reverse play.

[0058] The parser **705** places tags **707** into the FIFO **706** when it identifies video or audio segments, or is given private data. The DMA **709** controls when these tags are taken out. The tags **707** and the DMA addresses of the segments are placed into the event queue **708**. The frame type information, whether it is a start of a video I-frame, video B-frame, video P-frame, video PES, audio PES, a sequence header, an audio frame, or private data packet, is placed into the event queue **708** along with the offset in the related circular buffer where the piece of information was placed. The program logic operating in the CPU **713** examines events in the circular buffer after it is transferred to the DRAM **714**.

[0059] The Media Switch **701** has a data bus **711** that connects to the CPU **713** and DRAM **714**. An address bus **712** is also shared between the Media Switch **701**, CPU **713**, and DRAM **714**. A hard disk or storage device **710** is connected to one of the ports of the Media Switch **701**. A DVD player/recorder **719** is also connected to one of the ports of the Media Switch **701**. The Media Switch **701** outputs streams to an MPEG video decoder **715** and a separate audio decoder **717**. The audio decoder **717** signals contain audio cues generated by the system in response to the user's commands on a remote control or other internal events. The decoded audio output from the MPEG decoder is digitally mixed **718** with the separate audio signal. The resulting signals contain video, audio, and on-screen displays and are sent to the TV **716**.

[0060] The Media Switch **701** takes in 8-bit data and sends it to the disk, while at the same time extracts another stream of data off of the disk and sends it to the MPEG decoder **715**. All of the DMA engines described above can be working at the same time. The Media Switch **701** can be implemented in hardware using a Field Programmable Gate Array (FPGA), ASIC, or discrete logic.

[0061] Rather than having to parse through an immense data stream looking for the start of where each frame would be, the program logic only has to look at the circular event buffer in DRAM **714** and it can tell where the start of each frame is and the frame type. This approach saves a large amount of CPU power, keeping the real time requirements of the CPU **713** small. The CPU **713** does not have to be very fast at any point in time. The Media Switch **701** gives the CPU **713** as much time as possible to complete tasks. The parsing mechanism **705** and event queue **708** decouple the CPU **713** from parsing the audio, video, and buffers and the real time nature of the streams, which allows for lower costs. It also allows the use of a bus structure in a CPU environment that operates at a much lower clock rate with much cheaper memory than would be required otherwise.

[0062] The CPU **713** has the ability to queue up one DMA transfer and can set up the next DMA transfer at its leisure. This gives the CPU **713** large time intervals within which it can service the DMA controller **709**. The CPU **713** may respond to a DMA interrupt within a larger time window because of the large latency allowed. MPEG streams, whether extracted from an MPEG2 Transport or encoded from an analog TV signal, are typically encoded using a technique called Variable Bit Rate encoding (VBR). This technique varies the amount of data required to represent a sequence of images by the amount of movement between those images. This technique can greatly reduce the required bandwidth for a signal, however sequences with rapid movement (such as a basketball game) may be encoded with much greater bandwidth requirements. For example, the Hughes DirecTV satellite system encodes signals with anywhere from 1 to 10 Mb/s of required bandwidth, varying from frame to frame. It would be difficult for any computer system to keep up with such rapidly varying data rates without this structure.

[0063] The parser **705** parses the output from the DVD player **719**. The parser **705** can parse the DVD output to the hard disk **710**, which allows the DVD output to be buffered. The buffering allows DVD events to be recognized (described below) which allows the system to perform special effects on the DVD output stream. The DVD output can also be sent through the same route as live input streams without the having to cache DVD output onto the hard disk **710**, thus bypassing any copyright concerns.

[0064] The DVD output can also be from a DVD prerecorded by the invention containing pre-parsed information that allows the invention to perform special effects without caching content onto the hard disk **710** and display program information (e.g., program title, actor's names, genre, program description, etc.).

[0065] Referring to FIG. **14**, DVD-video and DVD-audio information is laid out on the DVD in Video Object Units (VOBU). A VOBU for a DVD-video format can have a navigation packet **1402**, one or more video packets **1403**, one or more audio packets **1404**, and one or more subpicture packets **1405**. DVD-audio formats use the highlight packets **1406**, still packets **1407**, and RT text packets **1408**. A VOBU does not always contain video.

[0066] Navigation packets **1402** contain information that determines how the physical data is accessed. The video **1403** and audio **1404** packets carry the video and audio content in recording order. Subpicture information **1405** overlays the video for subtitles, captions, menus, etc.

[0067] The video VOBUs are compliant with the MPEG program stream standard. VOBUs that contain video are organized as an MPEG group of pictures (GOP). This means that the parser **705** receives the same information format from the DVD player **719** as it does from the MPEG encoder **703**.

[0068] With respect to FIG. **8**, the program logic within the CPU has three conceptual components: sources **801**, transforms **802**, and sinks **803**. The sources **801** produce buffers of data. Transforms **802** process buffers of data and sinks **803** consume buffers of data. A transform is responsible for allocating and queuing the buffers of data on which it will operate. Buffers are allocated as if "empty" to sources of data, which give them back "full". The buffers are then queued and given to sinks as "full", and the sink will return the buffer "empty".

[0069] A source **801** accepts data from encoders, e.g., a digital satellite receiver. It acquires buffers for this data from the downstream transform, packages the data into a buffer, then pushes the buffer down the pipeline as described above. The source object **801** does not know anything about the rest of the system. The sink **803** consumes buffers, taking a buffer from the upstream transform, sending the data to the decoder, and then releasing the buffer for reuse.

[0070] There are two types of transforms **802** used: spatial and temporal. Spatial transforms are transforms that perform, for example, an image convolution or compression/decompression on the buffered data that is passing through. Temporal transforms are used when there is no time relation that is expressible between buffers going in and buffers coming out of a system. Such a transform writes the buffer to a file **804** on the storage medium. The buffer is pulled out at a later time, sent down the pipeline, and properly sequenced within the stream.

[0071] The transform **802** is used for writing buffers to a recordable DVD in the DVD player/recorder **805**. Sequenced buffers are written to the recordable DVD in several formats: a standard DVD format containing periodic navigation packets that allow the DVD to be played by any DVD player; and/or a custom format that the invention understands which contains information needed for program information display and special effects.

[0072] The system must create navigation packs when writing to a DVD. When recording a show from any source the system can create index marks which are needed for the navigation pack. The index marks are pointers to frames indexed at predefined time intervals in the program material and are stored as overhead with the program material. Later on, when the user selects program material to be written from the hard drive to a DVD, the navigation data already exists to set up the navigation pack data for recording. The indexes can also be created on-the-fly when the program material is being read from the hard drive and written to the DVD. Further, navigation packets can be arbitrarily placed within a program material stream and stored in a VOBU on the DVD.

[0073] Transform **802** also pulls buffers from the DVD player/recorder **805** when playing a DVD. The buffers are identified and sent down the pipeline in the same manner as when buffers are pulled from files on the storage medium **804**.

[0074] With respect to FIG. **15**, DVD **1501** acts as a source to DVD transform **1503**. Encrypted data stream from the DVD **1501** is decrypted **1502** before being routed to the DVD transform **1503**. Buffers of data are sent to the DVD transform **1503**.

[0075] The DVD transform **1503** receives data that contains navigation, video, audio, and subpicture information **1504** and transforms the data into PES triples. The PES triples are sent to the sink **1505**. The sink **1505** operates as described throughout.

[0076] DVD data can be prefetched from the DVD **1501** before processing through the system to make up for DVD latencies. For special effects, the system looks at GOP frames. In fast forward (3×), the system jumps to the next frame using the GOP info. For faster speeds, the system needs to scan ahead for the GOP frames. This method is needed when the program material is not allowed to touch the hard disk.

[0077] Otherwise, data can be buffered on the hard disk. The data gets decrypted and sent through the transform **1503** and stored on the hard disk. If there is a problem (such as copyright issues) with the program material being on the hard disk in raw form, then the system can store the data on the hard disk after it runs the data through the encrypter, thereby preserving the copy protection of the data. Data are decrypted before sending to the sink **1505**.

[0078] Referring to FIG. **9**, a C++ class hierarchy derivation of the program logic is shown. The TiVo Media Kernel (Tmk) **904**, **908**, **913** mediates with the operating system kernel. The kernel provides operations such as: memory allocation, synchronization, and threading. The TmkCore **904**, **908**, **913** structures memory taken from the media kernel as an object. It provides operators, new and delete, for constructing and deconstructing the object. Each object (source **901**, transform **902**, and sink **903**) is multi-threaded by definition and can run in parallel.

[0079] The TmkPipeline class **905**, **909**, **914** is responsible for flow control through the system. The pipelines point to the next pipeline in the flow from source **901** to sink **903**. To pause the pipeline, for example, an event called "pause" is sent to the first object in the pipeline. The event is relayed on to the next object and so on down the pipeline. This all happens asynchronously to the data going through the pipeline. Thus, similar to applications such as telephony, control of the flow of MPEG streams is asynchronous and separate from the streams themselves. This allows for a simple logic design that is at the same time powerful enough to support the features described previously, including pause, rewind, fast forward and others. In addition, this structure allows fast and efficient switching between stream sources, since buffered data can be simply discarded and decoders reset using a single event, after which data from the new stream will pass down the pipeline. Such a capability is needed, for example, when switching the channel being captured by the input section, or when switching between a live signal from the input section and a stored stream.

[0080] The source object **901** is a TmkSource **906** and the transform object **902** is a TmkXfrm **910**. These are intermediate classes that define standard behaviors for the classes in the pipeline. Conceptually, they handshake buffers down the pipeline. The source object **901** takes data out of a physical data source, such as the Media Switch, and places it into a PES buffer. To obtain the buffer, the source object **901** asks the down stream object in his pipeline for a buffer (allocEmptyBuf). The source object **901** is blocked until there is sufficient memory. This means that the pipeline is self-regulating; it has automatic flow control. When the source object **901** has filled up the buffer, it hands it back to the transform **902** through the pushFullBuf function.

[0081] The sink **903** is flow controlled as well. It calls nextFullBuf which tells the transform **902** that it is ready for the next filled buffer. This operation can block the sink **903** until a buffer is ready. When the sink **903** is finished with a buffer (i.e., it has consumed the data in the buffer) it calls releaseEmptyBuf. ReleaseEmptyBuf gives the buffer back to the transform **902**. The transform **902** can then hand that buffer, for example, back to the source object **901** to fill up again. In addition to the automatic flow-control benefit of this method, it also provides for limiting the amount of memory dedicated to buffers by allowing enforcement of a fixed allocation of buffers by a transform. This is an important feature in achieving a cost-effective limited DRAM environment.

[0082] The MediaSwitch class **909** calls the allocEmptyBuf method of the TmkClipCache **912** object and receives a PES buffer from it. It then goes out to the circular buffers in the Media Switch hardware and generates PES buffers. The MediaSwitch class **909** fills the buffer up and pushes it back to the TmkClipCache **912** object.

[0083] The TmkClipCache **912** maintains a cache file **918** on a storage medium. It also maintains two pointers into this cache: a push pointer **919** that shows where the next buffer coming from the source **901** is inserted; and a current pointer **920** which points to the current buffer used.

[0084] The buffer that is pointed to by the current pointer is handed to the Vela decoder class **916**. The Vela decoder class **916** talks to the decoder **921** in the hardware. The decoder **921** produces a decoded TV signal that is subsequently encoded into an analog TV signal in NTSC, PAL or other analog format. When the Vela decoder class **916** is fmished with the buffer it calls releaseEmptyBuf.

[0085] The structure of the classes makes the system easy to test and debug. Each level can be tested separately to make sure it performs in the appropriate manner, and the classes may be gradually aggregated to achieve the desired functionality while retaining the ability to effectively test each object.

[0086] The control object **917** accepts commands from the user and sends events into the pipeline to control what the pipeline is doing. For example, if the user has a remote control and is watching TV, the user presses pause and the control object **917** sends an event to the sink **903**, that tells it pause. The sink **903** stops asking for new buffers. The current pointer **920** stays where it is at. The sink **903** starts taking buffers out again when it receives another event that tells it to play. The system is in perfect synchronization; it starts from the frame that it stopped at.

[0087] The remote control may also have a fast forward key. When the fast forward key is pressed, the control object **917** sends an event to the transform **902**, that tells it to move forward two seconds. The transform **902** finds that the two second time span requires it to move forward three buffers. It then issues a reset event to the downstream pipeline, so that any queued data or state that may be present in the hardware decoders is flushed. This is a critical step, since the structure of MPEG streams requires maintenance of state across multiple frames of data, and that state will be rendered invalid by repositioning the pointer. It then moves the current pointer **920** forward three buffers. The next time the sink **903** calls nextFullBuf it gets the new current buffer. The same method works for fast reverse in that the transform **902** moves the current pointer **920** backwards.

[0088] A system clock reference resides in the decoder. The system clock reference is sped up for fast play or slowed down for slow play. The sink simply asks for full buffers faster or slower, depending on the clock speed.

[0089] With respect to FIG. **10**, two other objects derived from the TmkXfrm class are placed in the pipeline for disk access. One is called TmkClipReader **1003** and the other is called TmkClipWriter **1001**. Buffers come into the TmkClipWriter **1001** and are pushed to a file on a storage medium **1004**. TmkClipReader **1003** asks for buffers which are taken off of a file on a storage medium **1005**. A TmkClipReader **1003** provides only the allocEmptyBuf and pushFullBuf methods, while a TmkClipWriter **1001** provides only the nextFullBuf and releaseEmptyBuf methods. A TmkClipReader **1003** therefore performs the same function as the input, or "push" side of a TmkClipCache **1002**, while a TmkClipWriter **1001** therefore performs the same function as the output, or "pull" side of a TmkClipCache **1002**.

[0090] Referring to FIG. **11**, a preferred embodiment that accomplishes multiple functions is shown. A source **1101** has a TV signal input. The source sends data to a PushSwitch **1102** which is a transform derived from TmkXfrm. The PushSwitch **1102** has multiple outputs that can be switched by the control object **1114**. This means that one part of the pipeline can be stopped and another can be started at the users whim. The user can switch to different storage devices. The PushSwitch **1102** could output to a TmkClipWriter **1106**, which goes onto a storage device **1107** or write to the cache transform **1103**.

[0091] An important feature of this apparatus is the ease with which it can selectively capture portions of an incoming signal under the control of program logic. Based on information such as the current time, or perhaps a specific time span, or perhaps via a remote control button press by the viewer, a TmkClipWriter **1106** may be switched on to record a portion of the signal, and switched off at some later time. This switching is typically caused by sending a "switch" event to the PushSwitch **1102** object.

[0092] An additional method for triggering selective capture is through information modulated into the VBI or placed into an MPEG private data channel Data decoded from the VBI or private data channel is passed to the program logic. The program logic examines this data to determine if the data indicates that capture of the TV signal into which it was modulated should begin. Similarly, this information may also indicate when recording should end, or another data item may be modulated into the signal indicating when the capture should end. The starting and ending indicators may be explicitly modulated into the signal or other information that is placed into the signal in a standard fashion may be used to encode this information.

[0093] With respect to FIG. **12**, an example is shown which demonstrates how the program logic scans the words contained within the closed caption (CC) fields to determine starting and ending times, using particular words or phrases to trigger the capture. A stream of NTSC or PAL fields **1201** is presented. CC bytes are extracted from each odd field **1202**, and entered in a circular buffer **1203** for processing by the Word Parser **1204**. The Word Parser **1204** collects characters until it encounters a word boundary, usually a space, period or other delineating character. Recall from above, that the MPEG audio and video segments are collected into a series of fixed-size PES buffers. A special segment is added to each PES buffer to hold the words extracted from the CC field **1205**. Thus, the CC information is preserved in time synchronization with the audio and video, and can be correctly presented to the viewer when the stream is displayed. This also allows the stored stream to be processed for CC information at the leisure of the program logic, which spreads out load, reducing cost and improving efficiency. In such a case, the words stored in the special segment are simply passed to the state table logic **1206**.

[0094] During stream capture, each word is looked up in a table **1206** which indicates the action to take on recognizing that word. This action may simply change the state of the recognizer state machine **1207**, or may cause the state machine **1207** to issue an action request, such as "start capture", "stop capture", "phrase seen", or other similar requests. Indeed, a recognized word or phrase may cause the pipeline to be switched; for example, to overlay a different audio track if undesirable language is used in the program.

[0095] Note that the parsing state table **1206** and recognizer state machine **1207** may be modified or changed at any time. For example, a different table and state machine may be provided for each input channel Alternatively, these elements may be switched depending on the time of day, or because of other events.

[0096] Referring to FIG. **11**, a PullSwitch is added **1104** which outputs to the sink **1105**. The sink **1105** calls nextFullBuf and releaseEmptyBuf to get or return buffers from the PullSwitch **1104**. The PullSwitch **1104** can have any number of inputs. One input could be an ActionClip **1113**. The remote control can switch between input sources. The control object **1114** sends an event to the PullSwitch **1104**, telling it to switch. It will switch from the current input source to whatever input source the control object selects.

[0097] An ActionClip class provides for sequencing a number of different stored signals in a predictable and controllable manner, possibly with the added control of viewer selection via a remote control. Thus, it appears as a derivative of a TmkXfrm object that accepts a "switch" event for switching to the next stored signal.

[0098] This allows the program logic or user to create custom sequences of video output. Any number of video segments can be lined up and combined as if the program logic or user were using a broadcast studio video mixer. TmkClipReaders **1108**, **1109**, **1110** are allocated and each is hooked into the PullSwitch **1104**. The PullSwitch **1104** switches between the TmkClipReaders **1108**, **1109**, **1110** to combine video and audio clips. Flow control is automatic because of the way the pipeline is constructed. The Push and Pull Switches are the same as video switches in a broadcast studio.

[0099] The derived class and resulting objects described here may be combined in an arbitrary way to create a number of different useful configurations for storing, retrieving, switching and viewing of TV streams. For example, if multiple input and output sections are available, one input is viewed while another is stored, and a picture-in-picture window generated by the second output is used to preview previously stored streams. Such configurations represent a unique and novel application of software transformations to achieve the functionality expected of expensive, sophisticated hardware solutions within a single cost-effective device.

[0100] With respect to FIG. **13**, a high-level system view is shown which implements a VCR backup. The Output Module **1303** sends TV signals to the VCR **1307**. This allows the user to record TV programs directly on to video tape. The invention allows the user to queue up programs from disk to be recorded on to video tape and to schedule the time that the programs are sent to the VCR **1307**. Title pages (EPG data) can be sent to the VCR **1307** before a program is sent. Longer programs can be scaled to fit onto smaller video tapes by speeding up the play speed or dropping frames.

[0101] The VCR **1307** output can also be routed back into the Input Module **1301**. In this configuration the VCR acts as a backup system for the Media Switch **1302**. Any overflow storage or lower priority programming is sent to the VCR **1307** for later retrieval.

[0102] The Input Module **1301** can decode and pass to the remainder of the system information encoded on the Vertical Blanking Interval (VBI). The Output Module **1303** can encode into the output VBI data provided by the remainder of the system. The program logic may arrange to encode identifying information of various kinds into the output signal, which will be recorded onto tape using the VCR **1307**. Playing this tape back into the input allows the program logic to read back this identifying information, such that the TV signal recorded on the tape is properly handled. For example, a particular program may be recorded to tape along with information about when it was recorded, the source network, etc. When this program is played back into the Input Module, this information can be used to control storage of the signal, presentation to the viewer, etc.

[0103] One skilled in the art will readily appreciate that such a mechanism may be used to introduce various data items to the program logic which are not properly conceived of as television signals. For instance, software updates or other data may be passed to the system. The program logic receiving this data from the television stream may impose controls on how the data is handled, such as requiring certain authentication sequences and/or decrypting the embedded information according to some previously acquired key. Such a method works for normal broadcast signals as well, leading to an efficient means of providing non-TV control information and data to the program logic.

[0104] Additionally, one skilled in the art will readily appreciate that although a VCR is specifically mentioned above, any multimedia recording device (e.g., a Digital Video Disk-Random Access Memory (DVD-RAM) recorder) is easily substituted in its place.

[0105] Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. For example, the invention can be used in the detection of gambling casino crime. The input section of the invention is connected to the casino's video surveillance system. Recorded video is cached and simultaneously output to external VCRs. The user can switch to any video feed and examine (i.e., rewind, play, slow play, fast forward, etc.) a specific segment of the recorded video while the external VCRs are being loaded with the real-time input video.

[0106] Referring to FIG. **16**, an exemplary block diagram of a hardware interface between a hard disk, DVD and encoding/decoding section of the invention is shown. The system is bi-directionally connected to a hard disk **1601** via an IDE interface **1610**. A DVD player/recorder **1602** is also bi-directionally connected via an IDE interface **1603**.

[0107] The CPU **1609** selects the source or the destination for data flow. The hard disk **1601** stores application software and recorded program material. When the hard disk **1601** is selected for output, program material flows from the hard disk **1601** through the IDE interface **1610** to the MPEG decoder **1608**. The program material can be output to the viewer from the MPEG decoder **1608**.

[0108] Program information can also flow between (to and from) the hard disk **1601** and the DVD player/recorder **1602**. Data can be transferred from the DVD player/recorder **1602** via the IDE interface **1603** through the bridge **1604** to the MPEG encoder **1605**. The data are passed through the MPEG encoder **1605** to the hard disk **1601** via the IDE interface **1610**. The pass through occurs for MPEG data in both directions (DVD player/recorder **1602** to hard disk **1601** and hard disk **1601** to DVD player/recorder **1602**).

[0109] Broadcast input enters through the tuner **1607** and is decoded by the NTSC/PAL decoder **1606**. The decoded input is encoded into MPEG by the MPEG encoder **1605**. The MPEG output is directed to the hard disk **1601** for storage and can be passed from the hard disk **1601** to the MPEG decoder **1608** for live TV feed to the viewer.

[0110] When the DVD player/recorder **1602** is used for playback of program material, the program material path flows through the MPEG decoder **1608** to the viewer or is stored temporarily on the hard disk **1601** before being sent through the MPEG decoder **1608** to the viewer.

[0111] The application software stored on the hard disk **1601** allows the CPU **1609** to display the contents of program material stored on the hard disk **1601** and a DVD inserted into the DVD player/recorder **1602** to the user. The user has full control of program material transferred between the hard disk **1601** and the DVD player/recorder **1602** through the application software.

[0112] A system as described in U.S. patent application Ser. No. 10/339,698 entitled *Electronic Content Distribution and Exchange System*, also owned by the Applicant, can be used to enhance the invention's DVD interface. DVDs that are recorded by the invention can be created in a way that prevents the DVD from being played on other DVRs or DVD players unless the DVR or DVD player is registered to the same user.

[0113] With respect to FIG. **17**, a user owns DVR1 **1703** and DVR2 **1704**. The user registers the DVR1 **1703** and DVR2 **1704** serial numbers with a central registry or license database server **1702**. The central registry server **1702** creates a list of the DVRs owned by the user.

[0114] DVR1 **1703** and DVR2 **1704** are periodically updated with program guide information from distribution servers **1701**. The distribution servers **1701** notify the user's DVRs of the DVRs that the user owns by sending the list created by the central registry server **1702**. In this example, DVR1 **1703** is aware that the user also owns DVR2 **1704** and vice versa.

[0115] When a DVD **1705** is created on DVR1 **1703**, for example, the DVD **1705** has DVR1's serial number or encoded key written to the DVD **1705**. When the DVD **1705** is inserted into DVR2 **1704** for playback, DVR2 **1704** reads the serial number or encoded key and verifies that the DVD **1705** was created by a DVR that the user owns by comparing the serial number or encoded key to the registry list. If the serial number or encoded key was not from a DVR that the user owns, then access to the DVD is denied. Here, DVR2 **1704** finds that the DVD **1705** was created by DVR1 **1703** which is on DVR2's list of valid DVRs. DVR2 **1704** then plays or reads the DVD **1705**.

[0116] This approach protects the copyright holder's material by enforcing a fair use of the material that the user has recorded. The user is not able to pass recorded DVDs to other users. The user can only use his recorded DVDs for his own backing up of program material and viewing or restoring of the backed up material.

[0117] Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Accordingly, the invention should only be limited by the Claims included below.

1. A method for the simultaneous storage and play back of multimedia data, comprising:

receiving a digital stream that includes video and audio components;

buffering by a media switch segments of video and audio components in a memory;

associating stream information with each segment of a plurality of segments of video and audio components;

storing by the media switch the buffered video and audio components from the memory on one or more storage devices which contain previously stored segments of video and audio components that have associated stream information, the memory separate from the one or more storage devices;

sending by the media switch a particular segment of video and audio components from the memory to one or more decoders essentially simultaneously with the storing of the particular segment of video and audio components from the memory on the one or more storage devices;

decoding the particular segment of video and audio components into display output signals with the one or more decoders;

receiving a control command;

processing with at least a CPU the received control command and stream information associated with the previously stored video and audio components to send the previously stored video and audio components to the one or more decoders; and

mediating, by the media switch, between the CPU, the one or more storage devices, and the memory, the media switch operates asynchronously from the CPU.

2. The method of claim **1**, further comprising separating the video and audio components for storage and play back.

3. The method of claim **1**, wherein the associated stream information includes any of:

location of I-frames, time stamps, random access information, or index information.

4. The method of claim **1**, further comprising converting television signals from analog signals to digital streams and formatting the digital streams in accordance with an MPEG standard.

5. The method of claim **1**, further comprising parsing at least the video components to generate associated stream information.

6. The method of claim **5**, wherein the parsing is performed by the media switch.

7. The method of claim **1**, wherein the receiving step receives television signals via a tuner.

8. The method of claim **1**, wherein the receiving step receives digital streams via an input device.

9. A method for simultaneous storage and play back of multimedia data, comprising:

accepting multimedia data including video and audio data from an input device;

requesting, by a source object, a buffer and filling, by the source object, the buffer with the video and audio data;

transferring, by a transform object, the video and audio data from the buffer to a storage device, the storage device is separate from the buffer;

storing the video and audio data from the buffer on the storage device;

controlling automatically, by the transform object, flow of the video and audio data from the at least one tuner to the storage device;

placing, by the transform object, a particular segment of video and audio data from the buffer into data stream buffers essentially simultaneously with the storing of the particular segment of video and audio data from the buffer on the storage device;

requesting, by a sink object to the transform object, for the data stream buffers that contain the particular segment of video and audio data;

supplying, by the transform object to the sink object, the data stream buffers that contain the particular segment of video and audio data;

outputting the particular segment of video and audio data from the data stream buffers to a decoder by the sink object, the sink object is automatically flow controlled by the transform object;

converting with the decoder the particular segment of video and audio data into display signals;

sending the display signals for display;

receiving control commands with a control object; and

sending by the control object, in response to the control commands, flow command events to control flow of video and audio data through the source object, the transform object, and the sink object.

10. The method of claim **9**, further comprising analyzing the video and audio data to generate stream information.

11. The method of claim **10**, wherein the stream information includes any of: location of I-frames, time stamps, random access information, or index information.

12. The method of claim **9**, wherein the source object obtains the buffer from the transform object.

13. An apparatus for the simultaneous storage and play back of multimedia data, comprising:

a stream receiver that receives a digital stream that includes video and audio components;

a stream information associate that associates stream information with each segment of a plurality of segments of video and audio components;

a memory for buffering by a media switch segments of video and audio components;

one or more storage devices where the media switch stores the buffered video and audio components from the memory and which contain previously stored segments of video and audio components that have associated stream information, the memory separate from the one or more storage devices;

one or more decoders that receive a particular segment of video and audio components from the memory from the media switch essentially simultaneously with the media switch storing the particular segment of video and audio components from the memory on the one or more storage devices;

wherein the one or more decoders decode the particular segment of video and audio components into display output signals;

a control command receiver that receives a control command;

a CPU that processes the received control command and stream information associated with the previously stored video and audio components to send the previously stored video and audio components to the one or more decoders; and

wherein the media switch mediates between the CPU, the one or more storage devices, and the memory and wherein the media switch operates asynchronously from the CPU.

14. The apparatus of claim **13**, further comprising a video and audio component separator that separates the video and audio components for storage and play back.

15. The apparatus of claim **13**, wherein the associated stream information includes any of: location of I-frames, time stamps, random access information, or index information.

16. The apparatus of claim **13**, further comprising a stream converter that converts television signals from analog signals to digital streams and formats the digital streams in accordance with an MPEG standard.

17. The apparatus of claim **13**, further comprising a parser that parses at least the video components to generate associated stream information.

18. The apparatus of claim **17**, wherein the parser is a component of the media switch.

19. The apparatus of claim **13**, wherein the stream receiver comprises a tuner that receives television signals.

20. The apparatus of claim **13**, wherein the stream receiver comprises an input device that accepts the digital streams.

21. An apparatus for simultaneous storage and play back of multimedia data, comprising:

at least one input device that accepts multimedia data including video and audio data;

a buffer granted upon a request by a source object and filled with the video and audio data by the source object;

a storage device that is separate from the buffer and receives from a transform object the video and audio data from the buffer;

wherein the transform object automatically controls flow of the video and audio data from the at least one tuner to the storage device;

wherein the storage device stores the video and audio data from the buffer;

wherein the transform object places a particular segment of video and audio data from the buffer into data stream buffers essentially simultaneously with the storage device storing the particular segment of video and audio data from the buffer;

a sink object that requests and receives from the transform object, the data stream buffers that contain the particular segment of video and audio data;

a decoder that receives the particular segment of video and audio data from the data stream buffers from the sink object, the sink object is automatically flow controlled by the transform object;

wherein the decoder converts the particular segment of video and audio data into display signals;

wherein the display signals are sent for display;

a control object that receives control commands; and

wherein the control object sends, in response to the control commands, flow command events to control flow of video and audio data through the source object, the transform object, and the sink object.

22. The apparatus of claim **21**, wherein the transform object analyzes the video and audio data to generate stream information.

23. The apparatus of claim **22**, wherein the stream information includes any of:

location of I-frames, time stamps, random access information, or index information.

24. The apparatus of claim **21**, wherein the source object obtains the buffer from the transform object.

25. A non-transitory computer-readable medium storing one or more sequences of instructions for simultaneous storage and play back of multimedia data, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

accepting multimedia data including video and audio data from an at least one input device;

requesting, by a source object, a buffer and filling, by the source object, the buffer with the video and audio data;

transferring, by a transform object, the video and audio data from the buffer to a storage device, the storage device is separate from the buffer;

controlling automatically, by the transform object, flow of the video and audio data from the at least one tuner to the storage device;

storing the video and audio data from the buffer in the storage device;

placing, by the transform object, a particular segment of video and audio data from the buffer into data stream buffers essentially simultaneously with the storing of the particular segment of video and audio data from the buffer on the storage device;

supplying, by the transform object to a sink object, the data stream buffers that contain the particular segment of video and audio data;

requesting, by a sink object to the transform object, for the data stream buffers that contain the particular segment of video and audio data;

outputting the particular segment of video and audio data from the data stream buffers to a decoder by the sink object, the sink object is automatically flow controlled by the transform object;

converting with the decoder the particular segment of video and audio data into display signals;

sending the display signals for display;

receiving control commands with a control object; and

sending by the control object, in response to the control commands, flow command events to control flow of video and audio data through the source object, the transform object, and the sink object.

26. The non-transitory computer-readable medium of claim **25**, further comprising analyzing the video and audio data to generate stream information.

27. The non-transitory computer-readable medium of claim **26**, wherein the stream information includes any of: location of I-frames, time stamps, random access information, or index information.

28. The non-transitory computer-readable medium of claim **25**, wherein the source object obtains the buffer from the transform object.

29. A non-transitory computer-readable medium storing one or more sequences of instructions for simultaneous storage and play back of multimedia data, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving a digital stream that includes video and audio components;

buffering by a media switch segments of video and audio components in a memory;

associating stream information with each segment of a plurality of segments of video and audio components

storing by the media switch the buffered video and audio components from the memory on one or more storage devices which contain previously stored segments of video and audio components that have associated stream information, the memory separate from the one or more storage devices;

sending by the media switch a particular segment of video and audio components from the memory to one or more decoders essentially simultaneously with the storing of the particular segment of video and audio components from the memory on the one or more storage devices;

decoding the particular segment of video and audio components into display output signals with the one or more decoders;

receiving a control command;

processing with at least a CPU the received control command and stream information associated with the previously stored video and audio components to send the previously stored video and audio components to the one or more decoders; and

mediating, by the media switch, between the CPU, the one or more storage devices, and the memory, the media switch operates asynchronously from the CPU.

30. The non-transitory computer-readable medium of claim **29**, further comprising separating the video and audio components for storage and play back.

31. The non-transitory computer-readable medium of claim **29**, wherein the associated stream information includes any of: location of I-frames, time stamps, random access information, or index information.

32. The non-transitory computer-readable medium of claim **29**, further comprising converting television signals from analog signals to digital streams and formatting the digital streams in accordance with an MPEG standard.

33. The non-transitory computer-readable medium of claim **29**, further comprising parsing at least the video components to generate associated stream information.

34. The non-transitory computer-readable medium of claim **33**, wherein the parsing is performed by the media switch.

35. The non-transitory computer-readable medium of claim **29**, wherein the receiving step receives television signals via a tuner.

36. The non-transitory computer-readable medium of claim **29**, wherein the receiving step receives digital streams via an input device.

* * * * *