• Bracketed numbers in the Figures (e.g. (1)) designate explanatory notes which contain textual descriptions pertaining to various signals. The notes are given in the text relating to the relevant Figure. As it is believed that the Figures are self-explanatory, only brief comments will be made regarding the figures.

Joining of New Members

An "empty" synchronization group is one in which no members 18 are active. When the first member 18 joins, it is designated as the distinguished member by default. Figure 6 depicts a member 18 joining an empty group. The sequence of events is as depicted in Figure 6, to which attention is directed. The abbreviations used in the Figures are as follows: APPL means an application task; SNYC AGT means the synchronization agent task; MEM AGT means the member agent task; SYNC MST menas the synchronization master task 20; GRP MST means the Group Master Tast 25; MEM AGT (DM) means the distinguished member agent task; APPL (DM) means the application task which corresponds to the distinguished member agent.

The following notes refer to the bracketed numbers in Figure 6.

Notes:

(1) The Sync Agent will create a member agent task only if it had not existed previously, otherwise it will REINIT a previously created task instance. The START-AGT signal which follows initialization is used to pass initial data to the member agent 11.

(2) The AGT-MST-MSG signal to the sync master 13 contains the complete information about all member agents 11 on this processing element 12, including the newly-created member 18. (This ensures state convergence even in the presence of design faults.) The MST-REPLY message is used for positive acknowledgement so that the Sync Agent can send the next message to the Sync Master if it has one. (Only one outstanding message is allowed between the Sync Master and a Sync Agent.)

(3) If this group has not previously existed, a new Group Master is created. In that case a STARTUP signal follows to pass initial data to the Group Master Task, and an ACTIVATE signal is used to force it into an operational mode. If the group had existed previously (but had lost all its members) the existing Group Master Task is used.

(4) Once the Group Master has been activated, a GRP-EVENT signal is sent by the Sync Master informing it of the joining of the first member.

(5) Upon receipt of the GRP-EVENT signal, the Group Master selects the newly-created member agent 11 as the distinguished member and sends it a GRP-STATE signal. This signal establishes a connection between the Group Master and the Distinguished Member. All subsequent GRP-STATE signals are sequenced to ensure proper event ordering as well as to guard against communication failures. The GRP-REPLY signal is used to acknowledge one or more GRP-STATE signals and provides reliable communication.

The GRP-STATE signal contains the complete new state of the group rather than just information about the changes. This ensures that the system will converge to the true state even in the presence of design faults.

(6) The CHECK-FAIL signal is used to poll the application task to detect unexpected failures of the application. The application task never receives this signal; however, should the task fail, the member agent task will be notified by the underlying operating system kernel.

(7) The SYNC-REPLY signal contains a reply code of SYNC-DONE.

Figure 7 depicts the sequence for joining an existing synchronization group; the Group Master Task already exists, and the distinguished member is used to notify (broadcast) the other members 18 of the presence of a new member 18.

Notes:

(8) If the application has so requested, a GROUP-CHANGE signal is sent by each member agent 11 to the application whenever it detects a change in status of the group. In the case of the distinguished member the status change is gleaned from the GRP-STATE signal.

(9) When the distinguished member receives a GRP-STATE signal which indicates a group change (not all do), it broadcasts the new state to all other group members 18 using a MEM-MSG signal. Each member 18 acknowledges such messages with an ACK signal to provide reliable communication.

(10) When the newly-joining member 18 receives its MEM-MSG from the distinguished member it will send a SYNC-REPLY signal to the application (instead of a GROUP-CHANGE signal).

The control flow for the departure of a member 18 is shown in Figure 8. Note that the case of a processing element 12 failure is not shown here but is instead treated separately.

Notes:

(11) The sequence shown here corresponds to a voluntary departure initiated by the application program issuing an UNSYNC directive. This results in the member agent task terminating which, in turn, sends a COMPLETE signal to the parent task, the Sync Agent. The sequence is similar in situ-

LINKPLAY EXHIBIT 1004
Page 1564 of 4822

ations where the departure is not voluntary:
-When the application task fails, the member agent 11 is notified (through the failure of the CHECK-FAIL message) which results in the termination of the member agent task (and consequently, raising of the COMPLETE signal).
-If the member agent task itself fails, the Sync Agent is notified by the operating system kernel with a COMPLETE signal.

(12) Indicates a <GROUP-CHANGE> signal to an application task not shown in the Figure.

Figures 9a and 9b together depict the recovery of the synchronization master control 13 (i.e. Sync Master). The most general case is considered, i.e. the case of a running synchronization service 10 with active synchronization groups. This includes, as a subset, the case of a "cold" start.

Notes:

(13) Upon activation, Sync Master sends an ACTIVATE signal to each configured Sync Agent. The Sync Agent, whether they are already active or not, will respond with an AGT-REPLY message which includes a list of all member agents supported on that processing element. (The MST-REPLY signal is used for acknowledgement: refer to Figure 6.)

(14) Following activation of the Sync Agent, the Polling Control subcomponent 22 of Sync Master Control sends a POLL-AGT message to which the Sync Agent responds with an AGT-REPLY message. This exchange is repeated periodically to detect outages of the processing element.

(15) A Group Master is initiated only the first time a group is encountered. Refer to Figure 6 for further details on Group Master initiation.

(16) Before activating a Group Master, it is provided with data regarding the status of its members through GRP-DATA signals. Each signal contains the information for one member agent of one group. (This info is obtained from the AGT-REPLY messages.) In this way, the Group Master reconstructs the status of its group.

(17) After all Sync Agents have responded, the reconstruction is complete and an ACTIVATE signal is sent to all Group Masters. The Group Masters respond by sending a GRP-STATE signal to all distinguished members. Since this signal contains the complete group state, any group changes that might have occurred while the Sync Master was down are detected.

Figure 10 depicts the procedure for handling the failure of a processing element 12 which contains a synchronization agent 14 (Sync Agent).

Notes:

(18) The Sync Master detects a failure of a processing element when a TIME_OUT event is received. This means that a Sync Agent has not responded to a poll.

(19) For each group affected by the processing element failure, the Sync Master will send a GRP-EVENT signal to the respective Group Master.

Figure 11 depicts the procedure for the recovery of a processing element 12 which is part of synchronization service 10. Note that the recovery of processing element 12 does not extend to recovering member agent 11 tasks. It is assumed that these will be recovered when the application tasks (i.e. programs) which use them are restarted. Thus, the only action to recover a process element 12 is to integrate the sync agent 14 with the rest of synchronization service 10.

Notes:

(20) When a previously failed Sync Agent finally responds to a POLL_AGT signal, the Sync Master initiates the recovery procedure.

(21) The Sync Master registers the new processing element 12 and sends an ACTIVATE signal to the Sync Agent on that processing element 12 (Refer to Figure 9a for a more detailed description of the activation sequence).

Figure 12 depicts the procedure for member 18 to member 18 messages. This procedure (protocol) is used both for group broadcasts and point-to-point messages between members 18.

Notes:

(22) In case of a broadcast, a copy of the message is sent to each member 18. If the member 18 is not active, the message is not sent.

(23) Upon receiving a MEM-MSG signal which indicates an application-level message the message is relayed to the application task responsible for receiving asynchronous messages.

(24) A SYNC-REPLY (codes: GRP-ACK or MSG_ACK) signal is sent back to the originator.

Figure 1 depicts the procedure employed in the processing of all directives which require distinguished member intervention (rights handling directives).

Notes:

(25) If the request is made on the distinguished member site, then no message is sent.

(26) A reply signal (MEM-MSG followed by a SYNC-REPLY to the application).

In one implementation made by the inventors, the code for synchronization service 10 was contained in ten files which were distributed into six units, the useage dependencies (and compilation order) of which are shown in Figure 14.

Notes:

SYNCCTRL contains the stub of the Sync Master unit and directives to include three files (SYNCMST, SYNCGMST, and SYNCPOLL) which implement the Sync Master function. It also contains the definitions required for the master database 16. SYNCMST is an "include" file which

LINKPLAY EXHIBIT 1004
Page 1565 of 4822

contains the code for the Sync Master Task.
SYNCGMST is an "include" file which contains the code for the Group Master Task.
SYNCPOLL contains the Polling Control 22.
SYNCLOCL contains the stub of the Sync Agent unit as well as a definition of data and procedure objects shared by the Sync Agent Task and the Member Agent Tasks. It also contains directives to include two files (SYNCAGT, SYNCMAGT).
SYNCRESX contains the definition of the SYNCHRONIZE primitive. This unit must be loaded with the application code which uses the synchronization service.
SYNCDEFI contains a set of internal compile-object definitions for the synchronization service. This includes the definition of all entries used for communication between synchronization service components which are hidden from user programs. Since this file contains only compile objects it is not loaded.
SYNCDEFX contains a definition of all compile-objects which are exported by the synchronization servie to its users. Since this file contains only compile objects it is not loaded.
Application tasks which use the synchronization service need to include SYNCRESX and SYNCDEFX in their useage lists.

Note that SYNCCTRI implements the function of synchronization master control 13 (Fig. 1), and that SYNCLOCL and SYNCRESX together implement the functions of member agent 11 and synchronization agent 14 (Fig. 1). The files SYNCDEFI and SYNCDEFX are not resident in the service; they can be thought of as tools used in the construction of the synchronization service but they are not themselves a part of it.

Simplified pseudocode listings for the main constituents of the invention follow as appendix I. They are believed to be self-explanatory. Any elaboration of the material is accomplished through the use of appended notes, to which attention is directed.

As a further aid to the understanding and to the use of the present invention the following (a copy of a "User's Reference" to the synchronization service of the present invention, as prepared by one of the inventors) is included as Appendix II. It will expand on the use of the present invention.

Claims

1. A synchronization service [10] for use with a computer having a distributed operating system, to allow the construction of a customized synchronization scheme, for synchronizing the constituent portions of a distributed program, said service comprising:

a general set of application-independent synchronization primitives, whereby the construction of said customized synchronization scheme is achieved by the selective implementation of said application-independent synchronization primitives.

2. The synchronization service of claim 1 wherein said application-independent primitives comprise the following functions: synchronize; synchronize done; and unsynchronize.

3. The synchronization service of claim 2 wherein said primitives further comprise the following functions: request right; right granted; right refused; release right; group broadcast; and group acknowledge.

4. The synchronization service of claim 3 wherein said primitives further comprise the following functions: unsynchronize done; send to member; and message acknowledge.

5. A synchronization service [10] for use with a computer having an operating system [15] distributed over a plurality of processing elements [12], to allow the construction of a customized synchronization scheme, for synchronizing the constituent portions [18] of a distributed program, said service comprising:
a common synchronization master control means [13];
a synchronization agent means [14] for each processing element;
a plurality of application program components [18], each component located on a different processing element, each said component having associated therewith a member agent [11], said member agent being a program for interfacing with said synchronization agent means, and said synchronization agent means interfacing between said master control means and said member agent, whereby a customized synchronization scheme can be constructed based upon a general set of application-independent synchronization primitives contained in both said synchronization agent means [14] and said member agent [11] and accessed via said synchronization agent means.

6. The synchronization service of claim 5 wherein said application-independent primitives comprise the following functions: synchronize; synchronize done; and unsynchronize.

7. The synchronization service of claim 6 wherein said application-independent primitives further comprise the following functions: request right; right granted; right refused; release right; group broadcast; and group acknowledge.

8. A synchronization service [10] for use with a computer having an operating system [15] distributed over a plurality of processing elements, to allow the construction of a customized synchroniza-

LINKPLAY EXHIBIT 1004
Page 1566 of 4822

tion scheme, for synchronizing the constituent component of a distributed program, said service [10] comprising the steps of:

a) joining a program component [18] on a first processing element [12] to a group of existing program components [18] on at least a second processing element [12] so that each of the existing components is aware of the presence and location of the joining components;

b) informing each member of a group of physically distributed program components when one or more components which are members of said group, depart from it;

c) selecting, as a distinguished member, one program component from a group of distributed program components such that, within said group, there is never more than one said distinguished member; and

d) providing mutually exclusive rights to said group of distributed program components such that no more than one said component can appropriate a given right at any time.

9. The synchronization service of claim 8 further including the step of providing reliable point-to-point communication between said distributed program components on the basis of their internal group identifiers.

10. The synchronization service of claim 9 further including the step of providing a broadcast mechanism from any one program component to all other program components which are currently declared as being in the same group as the broadcasting component.

11. The synchronization service of claim 10 wherein said program components are components of an application program.

12. The synchronization service of claim 10 wherein said program components are components of an operating system program.

13. The synchronization service of claim 8 wherein said physical processing elements are logically distributed entities at one physical location.

14. A synchronization service [10], for use with a computer having an operating system [15] distributed over a plurality of processing elements [12], to allow the construction of customized synchronization schemes for synchronizing the constituent components [18] of a distributed program, said service comprising, as required, the steps of:

a) establishing a synchronization group for said distributed program, said group comprising at least one distributed program component [18];

b) joining a program component [18] to said group of existing program components so that each of the components is aware of the presence and the location of all the other components in said group;

c) informing each member of said group of distributed program components when one or more components which are members of said group, depart from it;

d) selecting, as a distinguished member for said group, one program component from said group of distributed program components such that, within said group, there is never more than one said distinguished member; and

e) providing mutually exclusive rights to said group of distributed program components such that no more than one said component can appropriate a given right at any time.

15. The synchronization service of claim 14 further including the step of providing full connectivity between all said distributed program components of said group.

16. The synchronization service of claim 15 wherein said distributed program is an application program.

17. The synchronization service of claim 15 wherein said distributed program is an operating system program.

18. The synchronization service of claim 15 wherein each said program component [18] is on a different processing element [12].

19. A synchronization service [10], for use with a computer having an operating system [15] distributed over a plurality of processing elements [12], to allow the construction of customized synchronization schemes for synchronizing the constituent components [18] of a distributed program, said service including a synchronization master control [13] comprising:

master control means [21] for activating said synchronization service;

polling means [22] for polling the processing elements [12] associated with said components of said distributed program so as to monitor the status of said processing element;

control means [24] for joining new members [18] to said group, and for handling departures of members [18] from said group; and

a database means [16] containing information representative of the current state of said synchronization service at a given point in time.

20. The synchronization service of claim 19 further including, at each said processing element, a synchronization agent [14] comprising:

means for accepting synchronization directives and for creating corresponding member agents; and

means for monitoring the status of all active member agents on said processing element and reporting same to said synchronization master control [13].

21. The synchronization service of claim 20 further including at each said processing element [12], a member agent [11] each synchronization

LINKPLAY EXHIBIT 1004
Page 1567 of 4822

group, comprising:
communications means [33] for providing a reliable communications service between program components [18];
storage means [32] for maintaining a local version of the current state of all other program components [18];
handler means [31] for providing the interface between user tasks and said member agent [11]; and
distinguished member means [30] for implementing the distinguished member function on only one program component [18] at any given time.

LINKPLAY EXHIBIT 1004
Page 1568 of 4822

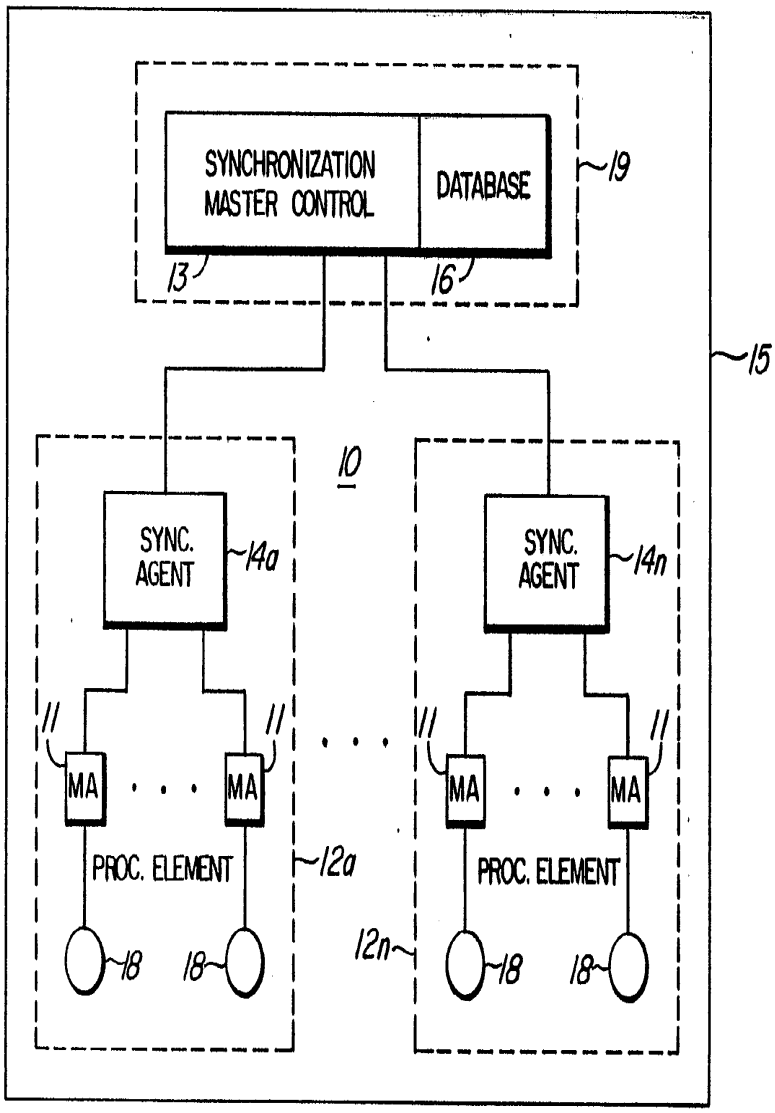


FIG. 1

0 251 584

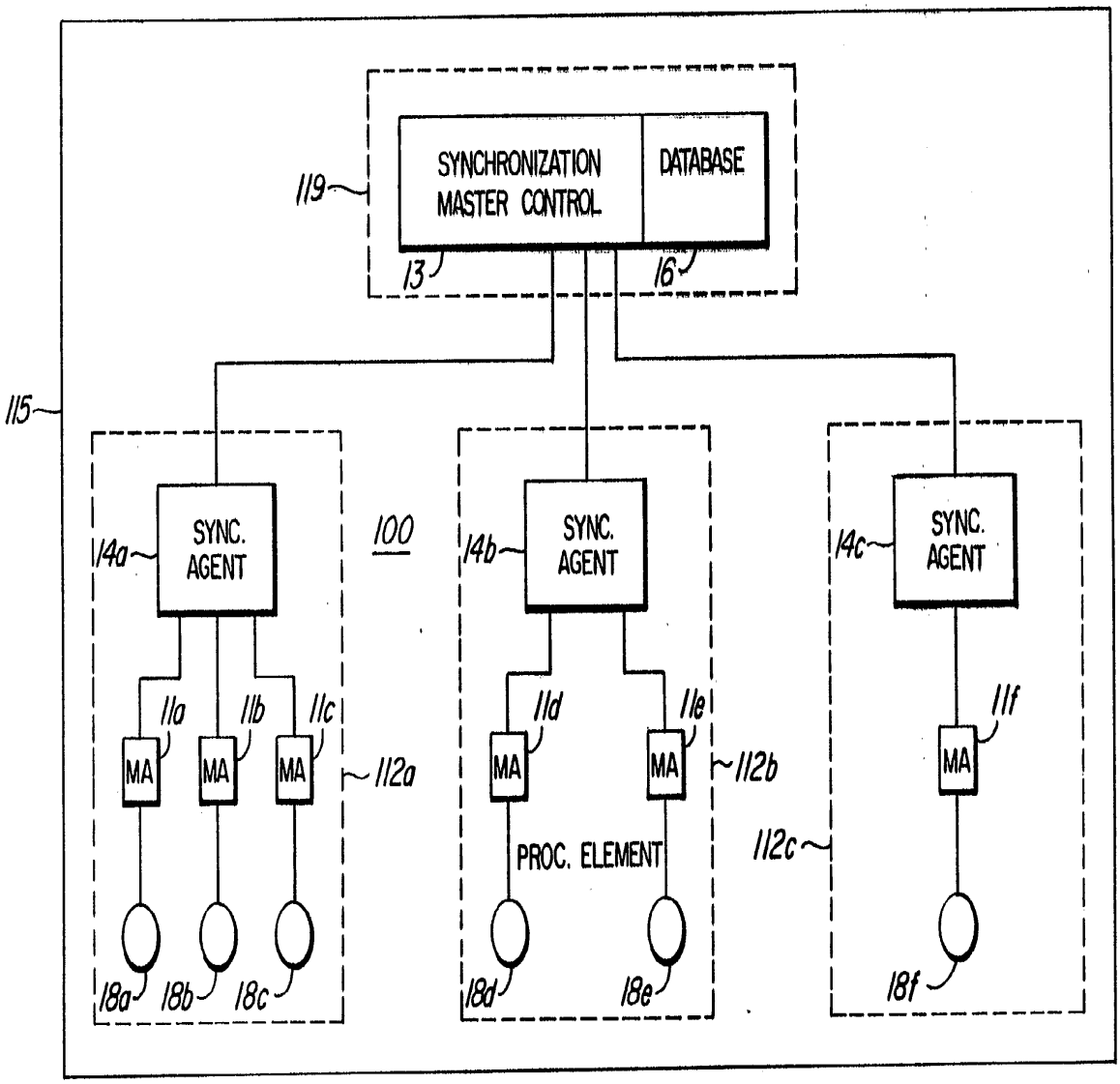


FIG. 2a

0 251 584

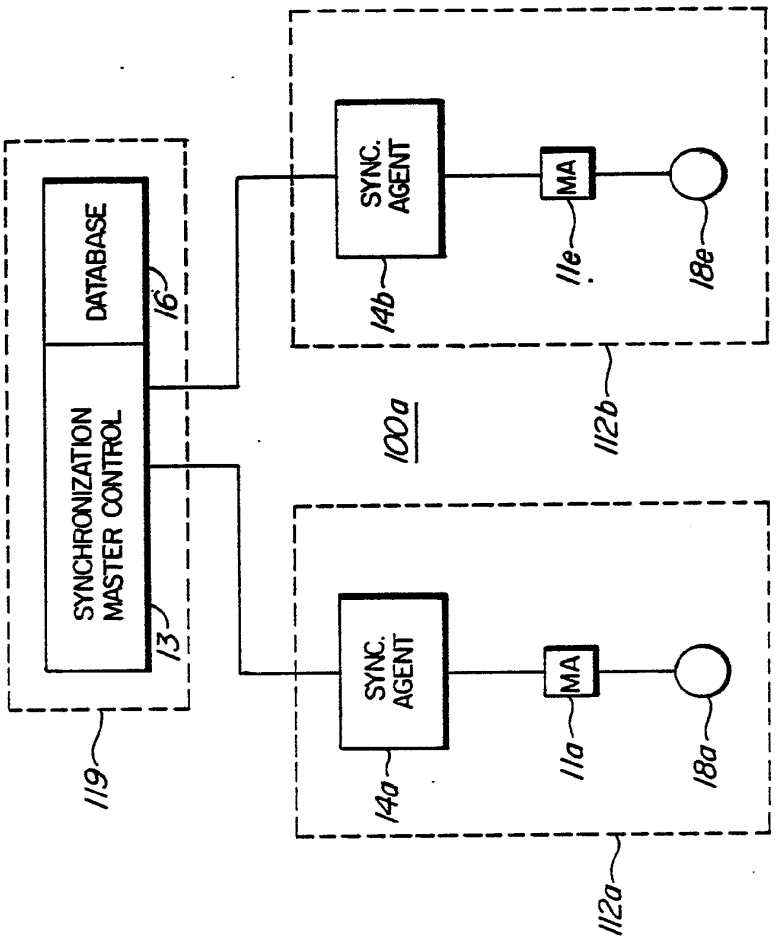

SYNCHRONIZATION MASTER CONTROL
13
DATABASE
16
119
100a
SYNC. AGENT
14a
MA
11a
18a
112a
SYNC. AGENT
14b
MA
11e
18e
112b

FIG. 2b

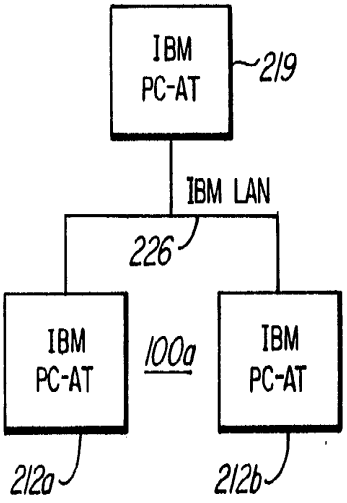


FIG. 2c

SYNCHRONOUS PRIMITIVES

REQUEST	REPLY
SYNCHRONIZE (group-Id)	SYNC_DONE (group-status)
UNSYNC	UNSYNC_DONE
REQ_RIGHT (right-id, Immed)	R_GRANTED
	R_REFUSED
REQ_RIGHT (right-id, Queued)	R:GRANTED
REL_RIGHT (right-id)	R_RELEASED
QRY_RIGHTS	R_STATUS (rights-status)
GRP_BRDCST (message)	GRP_ACK
SND_TO_MEM (message)	MSG_ACK

ASYNCHRONOUS NOTIFICATIONS

GROUP-CHANGE (group-status)
GROUP-MSG (message)

FIG. 3a

0 251 584

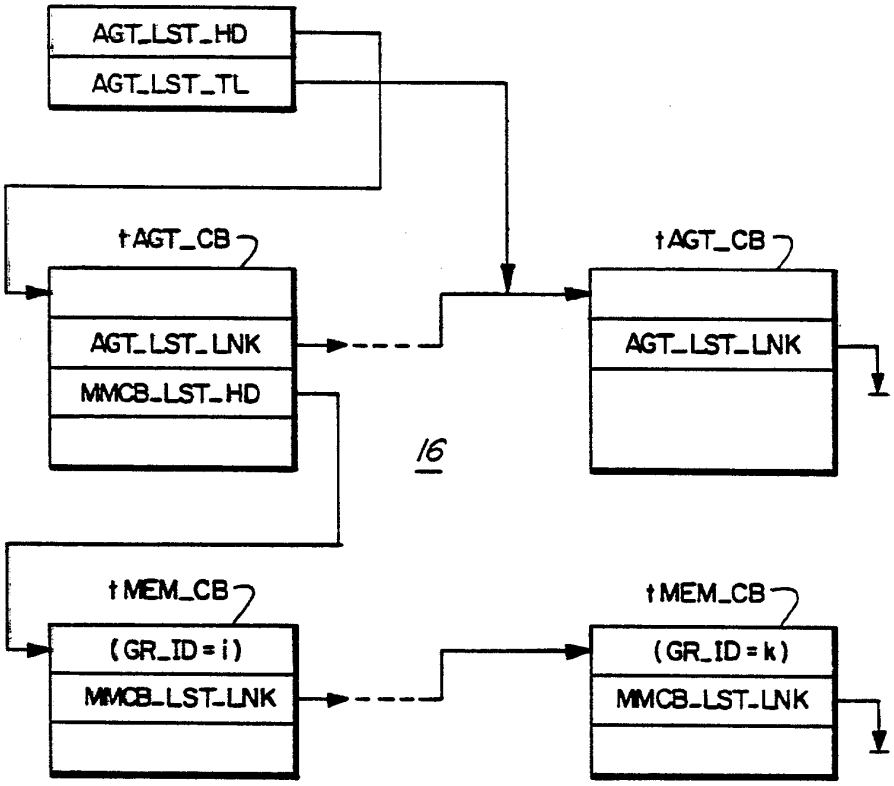

AGT_LST_HD
AGT_LST_TL
tAGT_CB
AGT_LST_LNK
MMCB_LST_HD
tAGT_CB
AGT_LST_LNK
16
tMEM_CB
(GR_ID = i)
MMCB_LST_LNK
tMEM_CB
(GR_ID = k)
MMCB_LST_LNK

FIG. 4

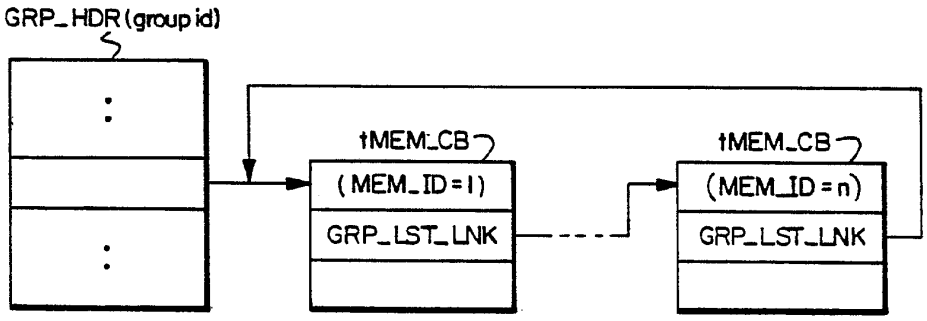

GRP_HDR (group id)
tMEM_CB
(MEM_ID = 1)
GRP_LST_LNK
tMEM_CB
(MEM_ID = n)
GRP_LST_LNK

FIG. 5

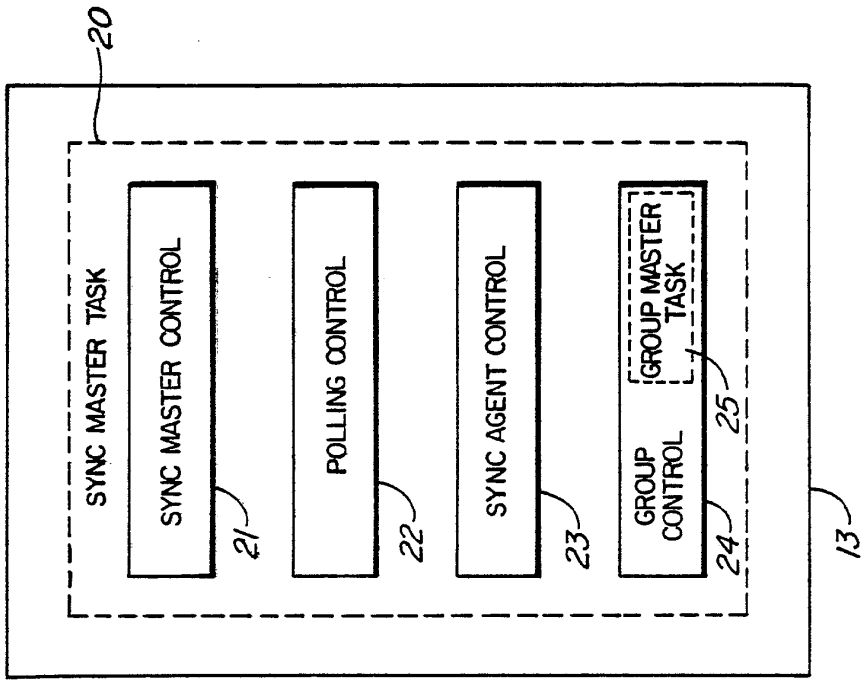

20
SYNC MASTER TASK
SYNC MASTER CONTROL
21
POLLING CONTROL
22
SYNC AGENT CONTROL
23
GROUP CONTROL
24
GROUP MASTER TASK
25
13

FIG. 3b

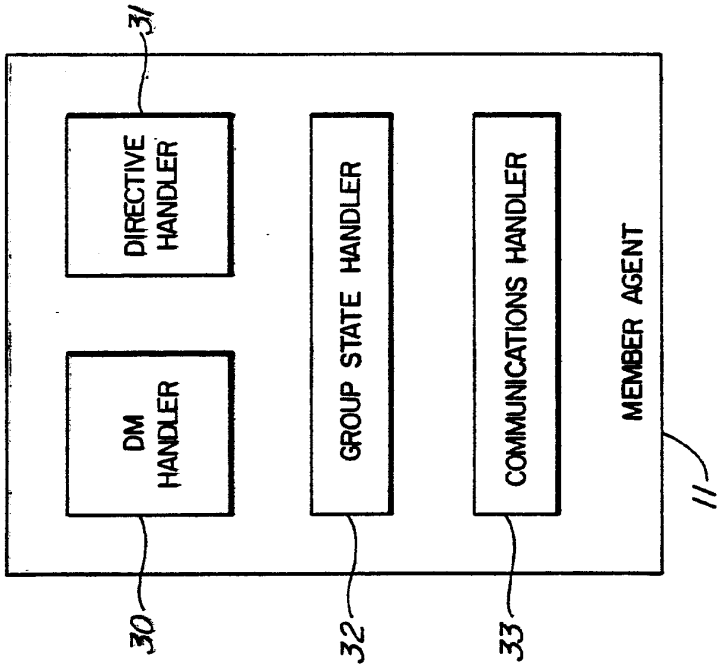
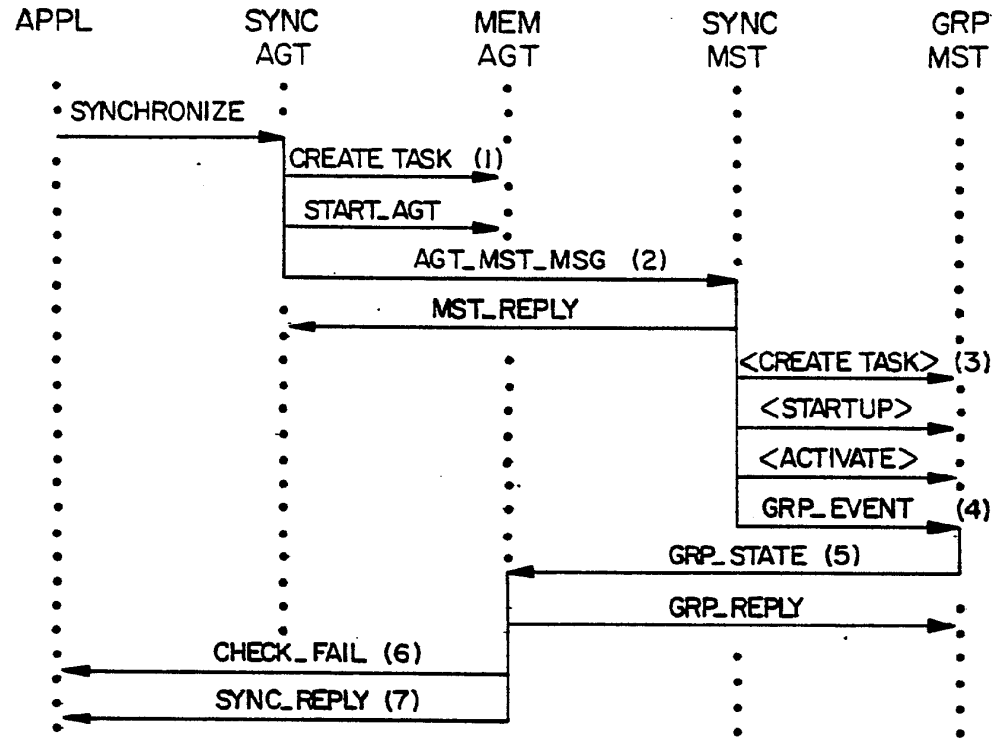

31
DIRECTIVE HANDLER
DM HANDLER
GROUP STATE HANDLER
COMMUNICATIONS HANDLER
MEMBER AGENT
11
30
32
33

FIG. 3c



LEGEND
APPL = MEMBER 18
SYNC AGT = SYNCHRONIZATION AGENT 14
MEM AGT = MEMBER AGENT 11
SYNC MST = SYNCHRONIZATION MASTER 13
GRP MST = GROUP MASTER 25

FIG. 6

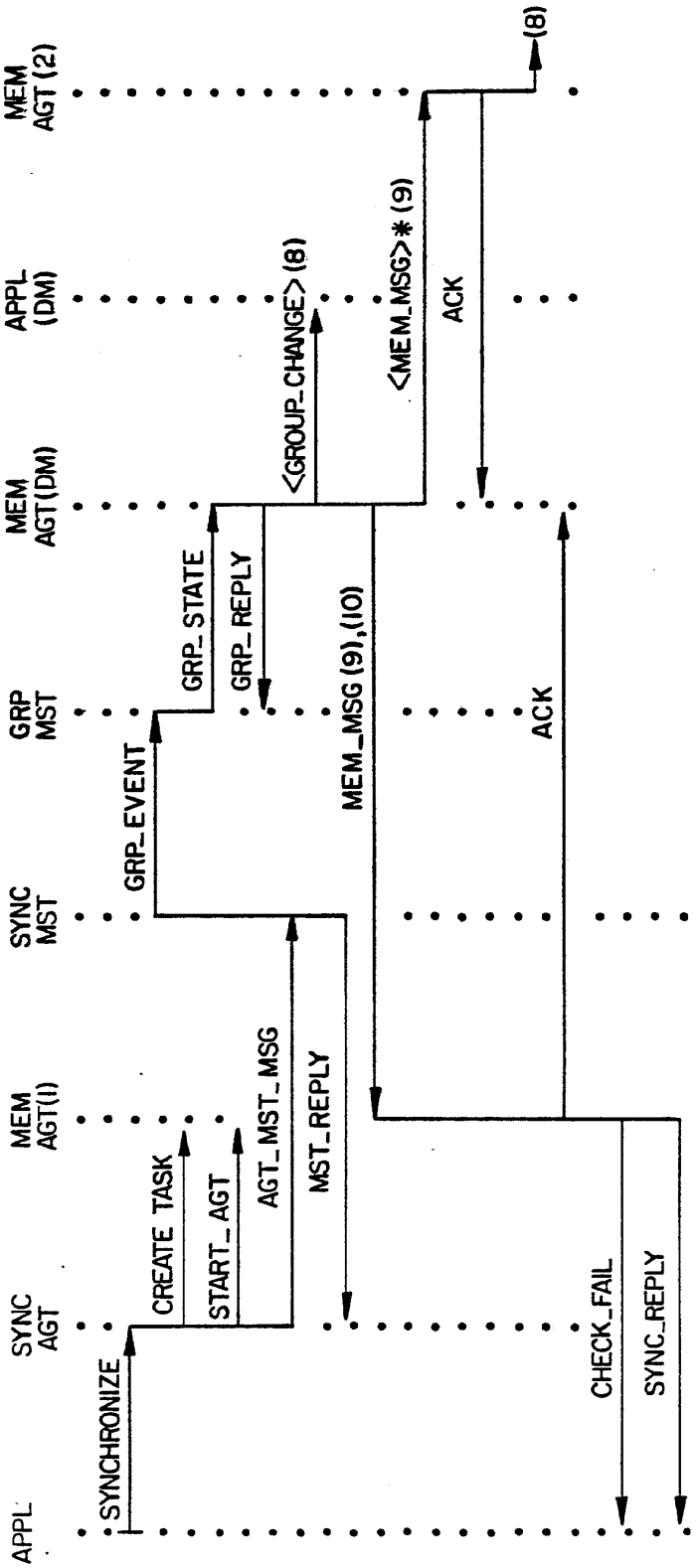

APPL
SYNC AGT
MEM AGT(1)
SYNC MST
GRP MST
MEM AGT(DM)
APPL (DM)
MEM AGT(2)
SYNCHRONIZE
CREATE TASK
START_AGT
AGT_MST_MSG
MST_REPLY
GRP_EVENT
GRP_STATE
GRP_REPLY
<GROUP_CHANGE> (8)
MEM_MSG (9),(10)
<MEM_MSG>* (9)
ACK
(8)
ACK
CHECK_FAIL
SYNC_REPLY

FIG. 7

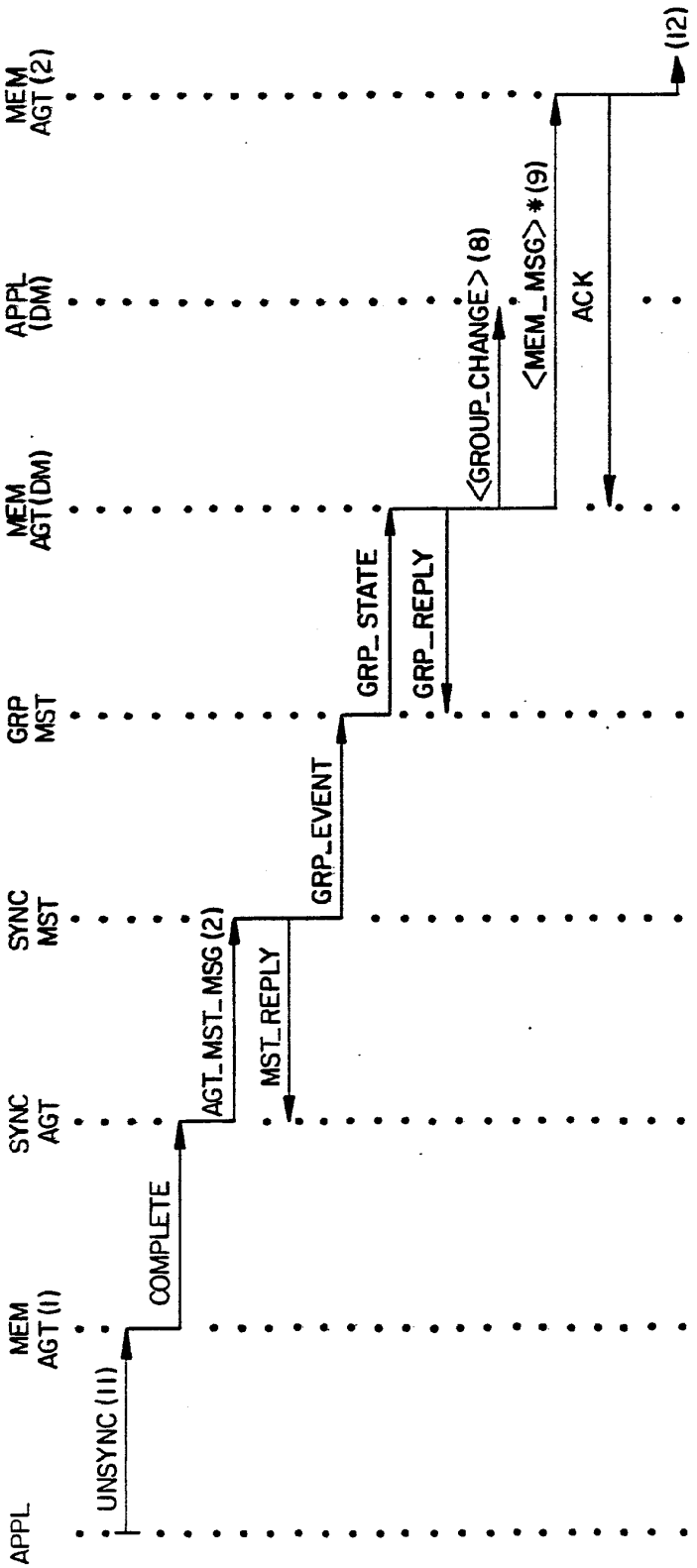

APPL
MEM AGT (1)
SYNC AGT
SYNC MST
GRP MST
MEM AGT(DM)
APPL (DM)
MEM AGT (2)
UNSYNC (11)
COMPLETE
AGT_MST_MSG (2)
MST_REPLY
GRP_EVENT
GRP_STATE
GRP_REPLY
<GROUP_CHANGE> (8)
<MEM_MSG> *(9)
ACK
(12)

FIG. 8

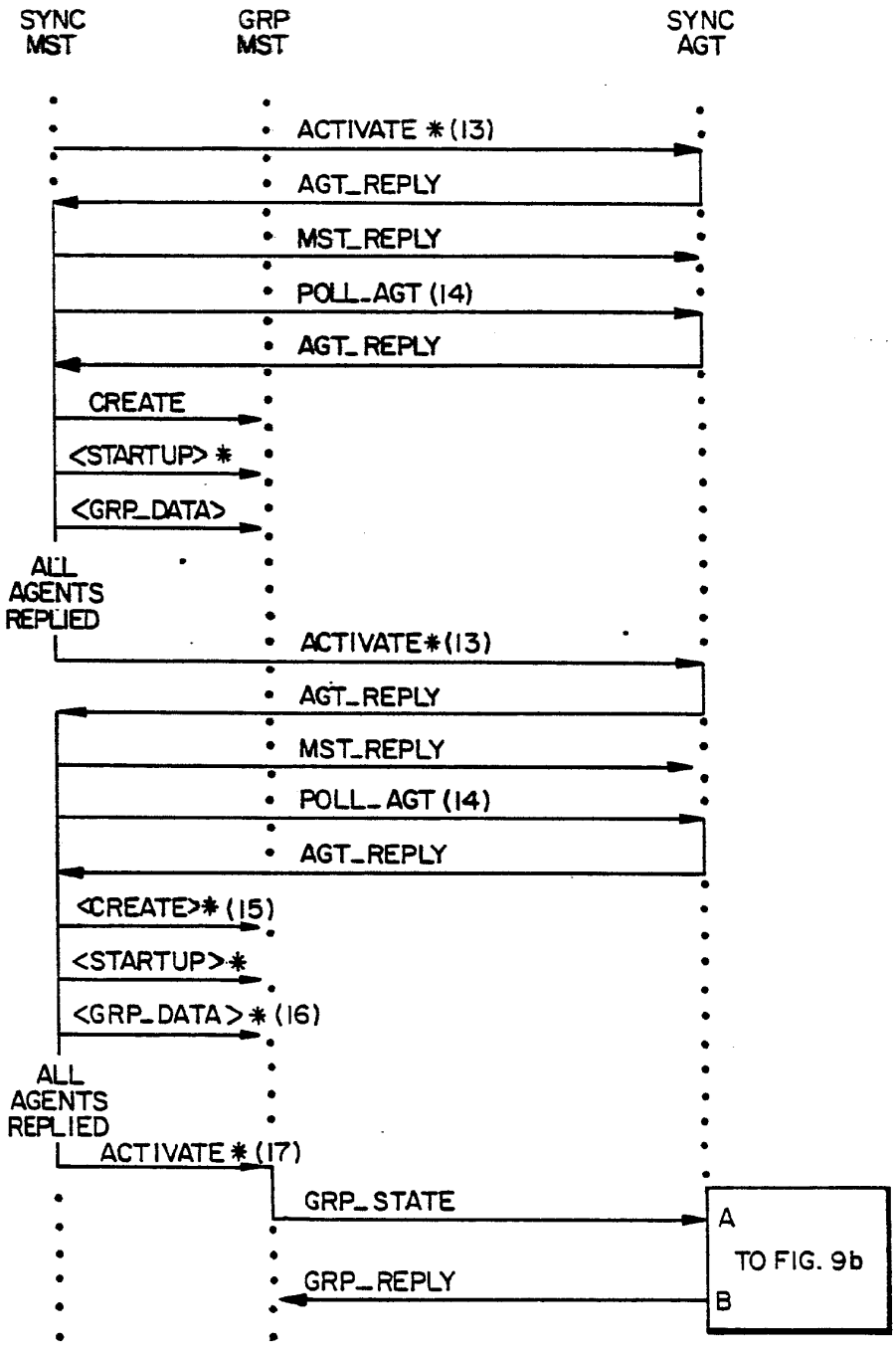

SYNC MST
GRP MST
SYNC AGT
ACTIVATE * (13)
AGT_REPLY
MST_REPLY
POLL_AGT (14)
AGT_REPLY
CREATE
<STARTUP> *
<GRP_DATA>
ALL AGENTS REPLIED
ACTIVATE*(13)
AGT_REPLY
MST_REPLY
POLL_AGT (14)
AGT_REPLY
<CREATE>* (15)
<STARTUP>*
<GRP_DATA>* (16)
ALL AGENTS REPLIED
ACTIVATE * (17)
GRP_STATE
GRP_REPLY
A
TO FIG. 9b
B

FIG. 9a

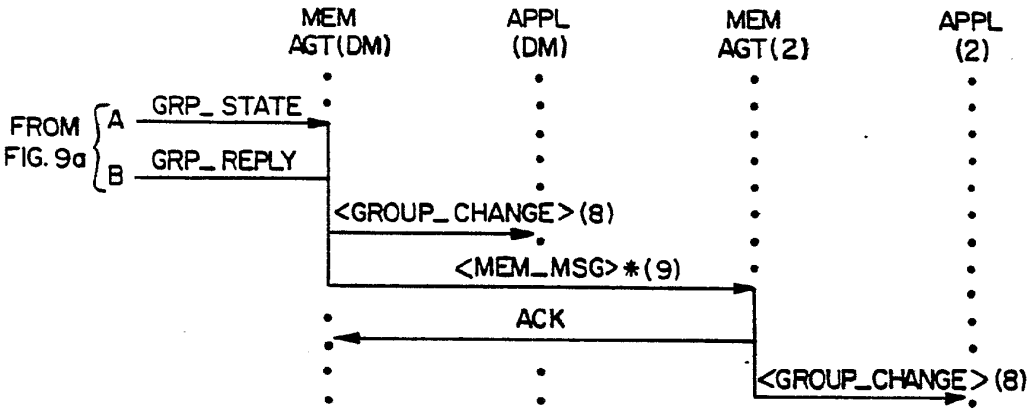

MEM AGT(DM)
APPL (DM)
MEM AGT(2)
APPL (2)
FROM FIG. 9a
A
B
GRP_ STATE
GRP_ REPLY
<GROUP_ CHANGE> (8)
<MEM_MSG>*(9)
ACK
<GROUP_CHANGE> (8)

FIG. 9b

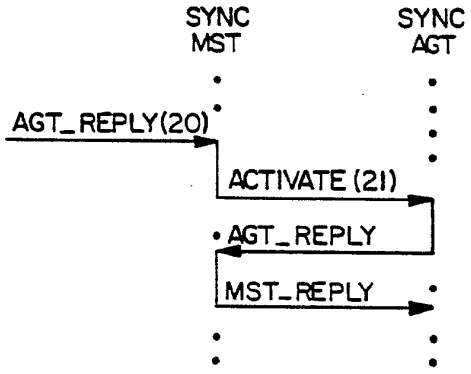

SYNC MST
SYNC AGT
AGT_REPLY(20)
ACTIVATE (21)
AGT_REPLY
MST_REPLY

FIG. 11

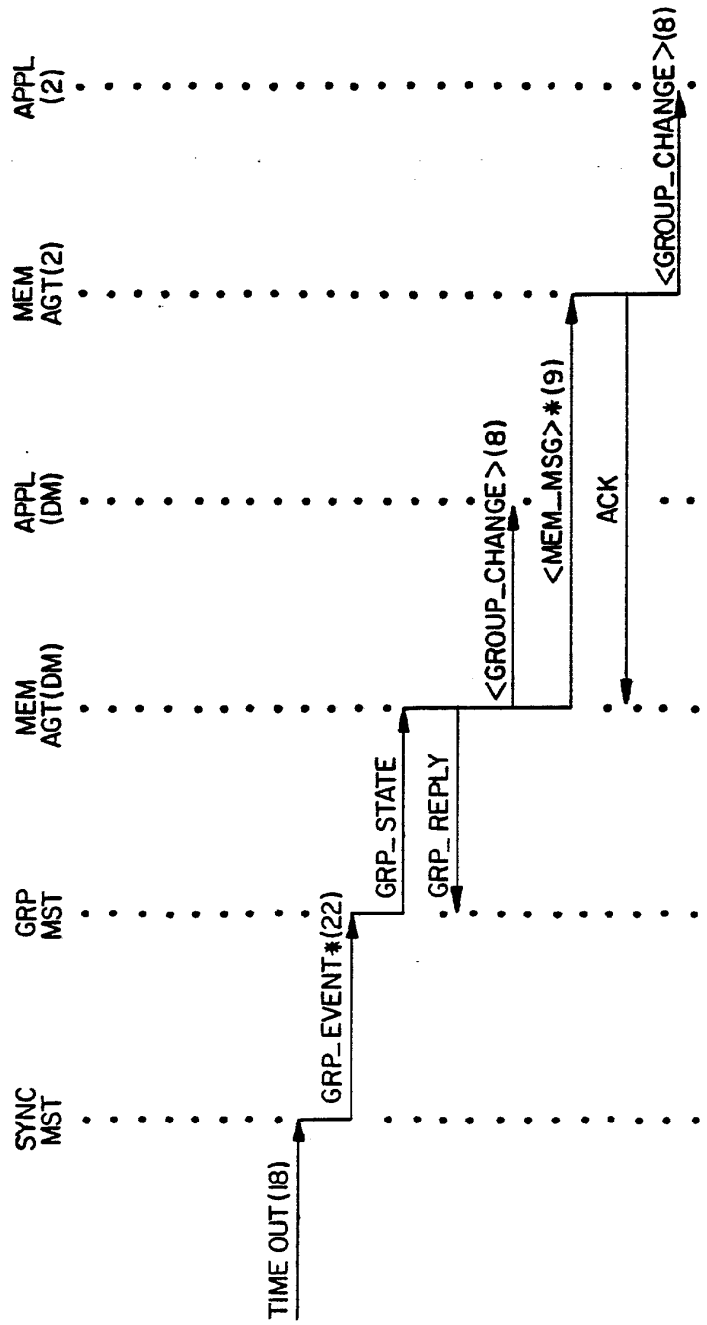

SYNC MST
GRP MST
MEM AGT(DM)
APPL (DM)
MEM AGT(2)
APPL (2)
TIME OUT (18)
GRP_EVENT*(22)
GRP_STATE
GRP_REPLY
<GROUP_CHANGE>(8)
<MEM_MSG>*(9)
ACK
<GROUP_CHANGE>(8)

FIG. 10

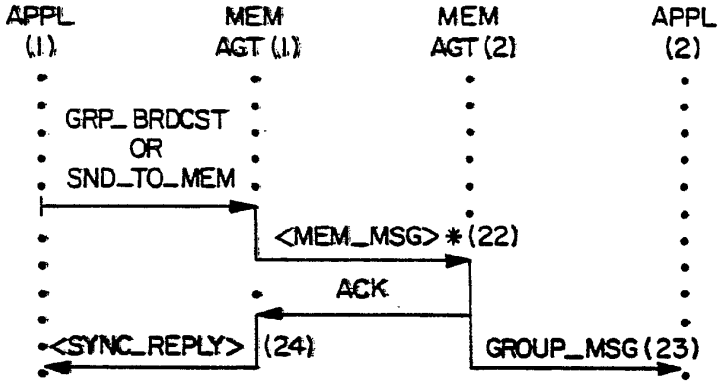

APPL (1)
MEM AGT (1)
MEM AGT (2)
APPL (2)
GRP_BRDCST OR SND_TO_MEM
<MEM_MSG> * (22)
ACK
<SYNC_REPLY> (24)
GROUP_MSG (23)

FIG. 12

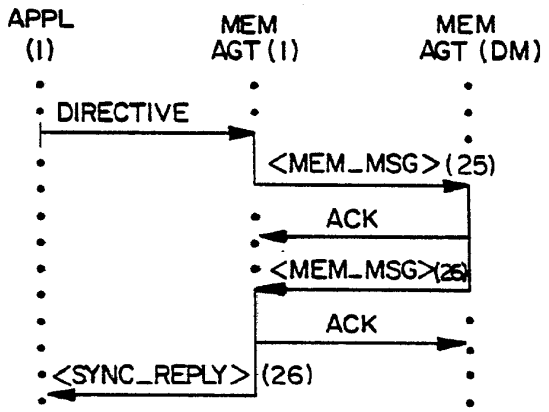

APPL (1)
MEM AGT (1)
MEM AGT (DM)
DIRECTIVE
<MEM_MSG> (25)
ACK
<MEM_MSG> (26)
ACK
<SYNC_REPLY> (26)

FIG. 13

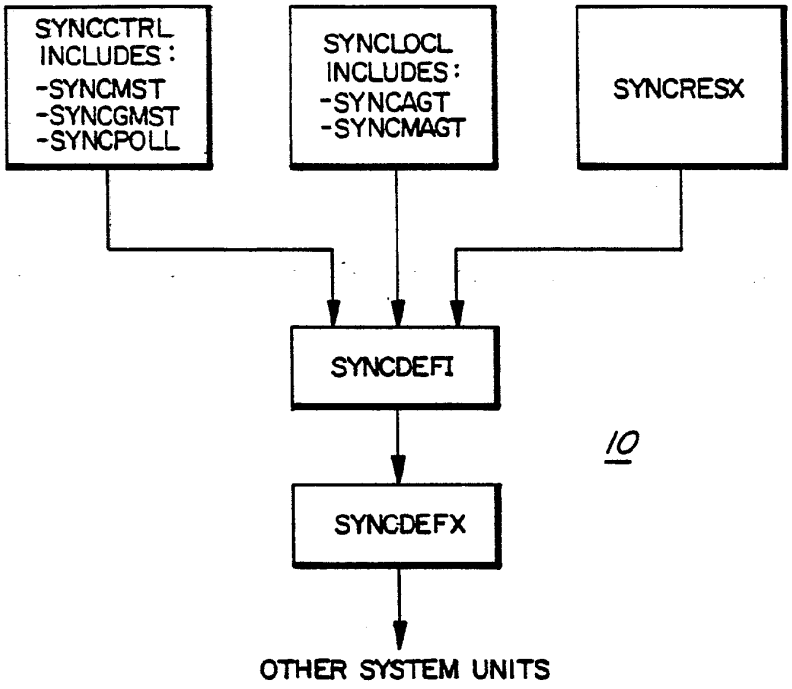

SYNCCTRL
INCLUDES:
-SYNCMST
-SYNCGMST
-SYNCPOLL
SYNCLOCL
INCLUDES:
-SYNCAGT
-SYNCMAGT
SYNCRESX
SYNCDEFI
10
SYNCDEFX
OTHER SYSTEM UNITS

FIG. 14

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 672 985 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **95300753.1**

(51) Int. Cl.[6] : **G06F 11/14**, G06F 11/20

(22) Date of filing : **07.02.95**

(30) Priority : **22.02.94 US 199444**

(43) Date of publication of application :
20.09.95 Bulletin 95/38

(84) Designated Contracting States :
DE FR GB

(71) Applicant : **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

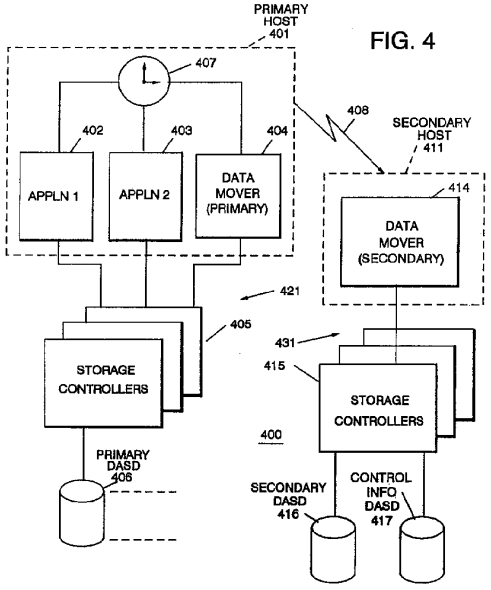
(72) Inventor : **Kern, Robert Frederic**
8338 East Calexico Street
Tucson, AZ 85730 (US)
Inventor : **Kern, Ronald Maynard**
761 North Colette Place
Tucson, AZ 85748 (US)
Inventor : **McBride, Gregory Edward**
8622 East Fairmount Place
Tucson, AZ 85715 (US)
Inventor : **Shackelford, David Michael**
1348 West Thurber Place Drive
Tucson, AZ 85705 (US)

(74) Representative : **Litherland, David Peter**
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) Asynchronous remote data duplexing.

(57) A remote data shadowing system provides storage based, real time disaster recovery capability. Record updates at a primary site cause write I/O operations in a storage subsystem therein. The write I/O operations are time stamped and the time, sequence, and physical locations of the record updates are collected in a primary data mover. The primary data mover groups sets of the record updates and associated control information based upon a predetermined time interval, the primary data mover appending a prefix header to the record updates thereby forming self describing record sets. The self describing record sets are transmitted to a remote secondary site wherein consistency groups are formed such that the record updates are ordered so that the record updates can be shadowed in an order consistent with the order the record updates cause write I/O operations at the primary site.

FIG. 4



EP 0 672 985 A1

LINKPLAY EXHIBIT 1004
Page 1583 of 4822

The present invention relates generally to disaster recovery techniques in data processing systems, and more particularly, to a system for real-time remote copying of data.

Data processing systems, in conjunction with processing data, typically are required to store large amounts of data (or records), which data can be efficiently accessed, modified, and re-stored. Data storage is typically separated into several different levels, or hierarchically, in order to provide efficient and cost effective data storage. A first, or highest level of data storage involves electronic memory, usually dynamic or static random access memory (DRAM or SRAM). Electronic memories take the form of semiconductor integrated circuits wherein millions of bytes of data can be stored on each circuit, with access to such bytes of data measured in nano-seconds. The electronic memory provides the fastest access to data since access is entirely electronic.

A second level of data storage usually involves direct access storage devices (DASD). DASD storage, for example, can comprise magnetic and/or optical disks, which store bits of data as micrometer sized magnetic or optical altered spots on a disk surface for representing the "ones" and "zeros" that make up those bits of the data. Magnetic DASD, includes one or more disks that are coated with remnant magnetic material. The disks are rotatably mounted within a protected environment. Each disk is divided into many concentric tracks, or closely spaced circles. The data is stored serially, bit by bit, along each track. An access mechanism, known as a head disk assembly (HDA), typically includes one or more read/write heads, and is provided in each DASD for moving across the tracks to transfer the data to and from the surface of the disks as the disks are rotated past the read/write heads. DASDs can store giga-bytes of data with the access to such data typically measured in milli-seconds (orders of magnitudes slower than electronic memory). Access to data stored on DASD is slower due to the need to physically position the disk and HDA to the desired data storage location.

A third or lower level of data storage includes tape and/or tape and DASD libraries. At this storage level, access to data is much slower in a library since a robot is necessary to select and load the needed data storage medium. The advantage is reduced cost for very large data storage capabilities, for example, terabytes of data storage. Tape storage is often used for back-up purposes, that is, data stored at the second level of the hierarchy is reproduced for safe keeping on magnetic tape. Access to data stored on tape and/or in a library is presently of the order of seconds.

Having a back-up data copy is mandatory for many businesses as data loss could be catastrophic to the business. The time required to recover data lost at the primary storage level is also an important recovery consideration. An improvement in speed over tape or library back-up, includes dual copy. An example of dual copy involves providing additional DASD's so that data is written to the additional DASDs (sometimes referred to as mirroring). Then if the primary DASDs fail, the secondary DASDs can be depended upon for data. A drawback to this approach is that the number of required DASDs is doubled.

Another data back-up alternative that overcomes the need to provide double the storage devices involves writing data to a redundant array of inexpensive devices (RAID) configuration. In this instance, the data is written such that the data is apportioned amongst many DASDs. If a single DASD fails, then the lost data can be recovered by using the remaining data and error correction procedures. Currently there are several different RAID configurations available.

The aforementioned back-up solutions are generally sufficient to recover data in the event that a storage device or medium fails. These back-up methods are useful only for device failures since the secondary data is a mirror of the primary data, that is, the secondary data has the same volume serial numbers (VOLSERs) and DASD addresses as the primary data. System failure recovery, on the other hand, is not available using mirrored secondary data. Hence still further protection is required for recovering data if a disaster occurs destroying the entire system or even the site, for example, earthquakes, fires, explosions, hurricanes, etc. Disaster recovery requires that the secondary copy of data be stored at a location remote from the primary data. A known method of providing disaster protection is to back-up data to tape, on a daily or weekly basis, etc. The tape is then picked up by a vehicle and taken to a secure storage area usually some kilometers away from the primary data location. A problem is presented in this back-up plan in that it could take days to retrieve the back-up data, and meanwhile several hours or even days of data could be lost, or worst, the storage location could be destroyed by the same disaster. A somewhat improved back-up method would be to transmit data to a back-up location each night. This allows the data to be stored at a more remote location. Again, some data may be lost between back-ups since back-up does not occur continuously, as in the dual copy solution. Hence, a substantial data amount could be lost which may be unacceptable to some users.

More recently introduced data disaster recovery solutions include remote dual copy wherein data is backed-up not only remotely, but also continuously. In order to communicate duplexed data from one host processor to another host processor, or from one storage controller to another storage controller, or some combination thereof, a substantial amount of control data is required for realizing the process. A high overhead, however, can interfere with a secondary site's ability to keep up with a primary site's processing, thus threatening the ability of the secondary site to be

LINKPLAY EXHIBIT 1004
Page 1584 of 4822

able to recover the primary in the event a disaster occurs.

Accordingly it is desired to provide a method and apparatus for providing a real time update of data consistent with the data at a primary processing location using minimal control data, wherein the method and apparatus operates independently of a particular application data being recovered, that is, generic storage media based rather than specific application data based.

An aim of the present invention is to provide an improved system and method for shadowing DASD data to a secondary site for disaster recovery.

According to a first aspect of the present invention, a method for forming consistency groups provides for disaster recovery capability from a remote site. Data updates generated by one or more applications running in a primary processor are received by a primary storage subsystem, wherein the primary storage subsystem causes I/O write operations to write each data update therein. The primary storage subsystem is synchronized by a common timer, and a secondary system, remote from the primary processor, shadows the data updates in sequence consistent order such that the secondary site is available for disaster recovery purposes. Thee method comprising steps of: (a) time stamping each write I/O operation occurring in the primary storage subsystem; (b) capturing write I/O operation record set information from the primary storage subsystem for each data update; (c) generating self describing record sets from the data updates and the respective record set information, such that the self describing record sets are sufficient to re-create a sequence of the write I/O operations; (d) grouping the self describing record sets into interval groups based upon a predetermined interval threshold; and (e) selecting a first consistency group as that interval group of self describing record sets having an earliest operational time stamp, the individual data updates being ordered within the first consistency group based upon time sequences of the I/O write operations in the primary storage subsystem.

In another aspect of the present invention, a primary system has a primary processor running one or more applications, wherein the applications generating record updates, and the primary processor generating self describing record sets therefrom. Each self describing record set is sent to a secondary system remote from the primary system, wherein the secondary system shadows the record updates in sequence consistent order based upon the self describing record sets for real time disaster recovery purposes. The primary processor is coupled to a primary storage subsystem wherein the primary storage subsystem receives the record updates and causes I/O write operations for storing each record update therein. The primary processor comprises a sysplex timer for providing a common time source to the applications and to the primary storage subsystem for synchronization purposes, and a primary data mover, synchronized by the sysplex timer, prompts the primary storage subsystem for providing record set information to the primary data mover for each record update. The primary data mover groups a plurality of record updates and each corresponding record set information into time interval groups, and inserts a prefix header thereto. Each time interval group forms the self describing record sets.

A preferred embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 is a block diagram of a disaster recovery system having synchronous remote data shadowing capabilities.

FIG. 2 is a flow diagram of a method for providing synchronous remote copy according to the disaster recovery system of FIG. 1.

FIG. 3 is a flow diagram of a method of an I/O error recovery program (I/O ERP) operation.

FIG. 4 is a block diagram of a disaster recovery system having asynchronous remote data shadowing capabilities.

FIG. 5 is a data format diagram showing a prefix header that prefixes read record sets from the primary site of FIG. 4.

FIG. 6 is a data format diagram describing fields making up a read record set.

FIG. 7 is a state table identifying volume configuration information.

FIG. 8 is a master journal as used by the secondary site of FIG. 4.

FIG. 9 is an example sequence for forming a consistency group.

FIG. 10 is a flow diagram showing a method of collecting information and read record sets for forming consistency groups.

FIG. 11 is a flow diagram showing a method of forming consistency groups.

FIG. 12 is a table indicating full consistency group recovery rules application for an ECKD architecture device for a given sequence of I/O operations to a DASD track.

FIG. 13 is a description of the rules to be used in the table of FIG. 12.

FIG. 14 is a flow diagram of a method of writing read record set copies to a secondary site with full consistency group recovery capability.

A typical data processing system may take the form of a host processor, such as an IBM System/360 or IBM System/370 processor for computing and manipulating data, and running, for example, data facility storage management subsystem/multiple virtual systems (DFSMS/MVS) software, having at least one IBM 3990 storage controller attached thereto, the storage controller comprising a memory controller

LINKPLAY EXHIBIT 1004
Page 1585 of 4822

and one or more cache memory types incorporated therein. The storage controller is further connected to a group of direct access storage devices (DASDs) such as IBM 3380 or 3390 DASDs. While the host processor provides substantial computing power, the storage controller provides the necessary functions to efficiently transfer, stage/destage, convert and generally access large databases.

Disaster recovery protection for the typical data processing system requires that primary data stored on primary DASDs be backed-up at a secondary or remote location. The distance separating the primary and secondary locations depends upon the level of risk acceptable to the user, and can vary from several kilometers to thousands of kilometers. The secondary or remote location, in addition to providing a back-up data copy, must also have enough system information to take over processing for the primary system should the primary system become disabled. This is due in part because a single storage controller does not write data to both primary and secondary DASD strings at the primary and secondary sites. Instead, the primary data is stored on a primary DASD string attached to a primary storage controller while the secondary data is stored on a secondary DASD string attached to a secondary storage controller.

The secondary site must not only be sufficiently remote from the primary site, but must also be able to back-up primary data in real time. The secondary site needs to back-up primary data as the primary data is updated with some minimal delay. Additionally, the secondary site has to back-up the primary data regardless of the application program (e.g., IMS, DB2) running at the primary site and generating the data and/or updates. A difficult task required of the secondary site is that the secondary data must be order consistent, that is, secondary data is copied in the same sequential order as the primary data (sequential consistency) which requires substantial systems considerations. Sequential consistency is complicated by the existence of multiple storage controllers each controlling multiple DASDs in a data processing system. Without sequential consistency, secondary data inconsistent with primary data would result, thus corrupting disaster recovery.

Remote data duplexing falls into two general categories, synchronous and asynchronous. Synchronous remote copy involves sending primary data to the secondary location and confirming the reception of such data before ending a primary DASD input/output (I/O) operation (providing a channel end (CE) and device end (DE) to the primary host). Synchronous copy, therefore, slows the primary DASD I/O response time while waiting for secondary confirmation. Primary I/O response delay is increased proportionately with the distance between the primary and secondary systems - a factor that limits the remote distance to tens of kilometers. Synchronous copy, however, provides sequentially consistent data at the secondary site with relatively little system overhead.

Asynchronous remote copy provides better primary application system performance because the primary DASD I/O operation is completed (providing a channel end (CE) and device end (DE) to the primary host) before data is confirmed at the secondary site. Therefore, the primary DASD I/O response time is not dependent upon the distance to the secondary site and the secondary site could be thousands of kilometers remote from the primary site. A greater amount of system overhead is required, however, for ensuring data sequence consistency since data received at the secondary site will often not be in order of the primary updates. A failure at the primary site could result in some data being lost that was in transit between the primary and secondary locations.

Synchronous real time remote copy for disaster recovery requires that copied DASD volumes form a set. Forming such a set further requires that a sufficient amount of system information be provided to the secondary site for identifying those volumes (VOLSERs) comprising each set and the primary site equivalents. Importantly, the secondary site forms a "duplex pair" with the primary site and the secondary site must recognize when one or more volumes are out of sync with the set, that is, "failed duplex" has occurred. Connect failures are more visible in synchronous remote copy than in asynchronous remote copy because the primary DASD I/O is delayed while alternate paths are retried. The primary site can abort or suspend copy to allow the primary site to continue while updates for the secondary site are queued, the primary site marking such updates to show the secondary site is out of sync. Recognizing exception conditions that may cause the secondary site to fall out of sync with the primary site is needed in order that the secondary site be available at any time for disaster recovery. Error conditions and recovery actions must not make the secondary site inconsistent with the primary site.

Maintaining a connection between the secondary site and the primary site with secondary DASD present and accessible, however, does not ensure content synchronism. The secondary site may lose synchronism with the primary site for a number of reasons. The secondary site is initially out of sync when the duplex pair is being formed and reaches sync when an initial data copy is completed. The primary site may break the duplex pair if the primary site is unable to write updated data to the secondary site in which case the primary site writes updates to the primary DASD under suspended duplex pair conditions so that the updating application can continue. The primary site is thus running exposed, that is, without current disaster protection copy until the duplex pair is restored. Upon restoring the duplex pair, the secondary site is not immediately in sync. After applying now pending up-

LINKPLAY EXHIBIT 1004
Page 1586 of 4822

dates, the secondary site returns to sync. The primary site can also cause the secondary site to lose sync by issuing a suspend command for that volume to the primary DASD. The secondary site re-syncs with the primary site after the suspend command is ended, duplex pair is re-established, and pending updates are copied. On-line maintenance can also cause synchronization to be lost.

When a secondary volume is out of sync with a primary volume, the secondary volume is not useable for secondary system recovery and resumption of primary applications. An out-of-sync volume at the secondary site must be identified as such and secondary site recovery-takeover procedures need to identify the out-of-sync volumes for denying application access (forcing the volumes off-line or changing their VOLSERs). The secondary site may be called upon to recover the primary site at any instant wherein the primary site host is inaccessible - thus the secondary site requires all pertinent information about a sync state of all volumes. The secondary storage subsystem, that is the secondary storage controllers and DASD, is unable to determine all conditions causing the primary site to break synchronism due to primary site-encountered exceptions. For example, the primary site may break a duplex pair if the primary site is unable to access the secondary peer due to a primary I/O path or link failure that the secondary site is unaware of. In this case the secondary site shows in-sync state while the primary site indicates the duplex pair is broken.

External communication may notify the secondary site that an out-of-sync duplex pair volume exists. This is realizable by employing a user systems management function. Primary I/O operations end with channel end/device end/unit check (CE/DE/UC) status and sense data indicates the nature of the error. With this form of I/O configuration an error recovery program (ERP) processes the error and send an appropriate message to the secondary processor before posting the primary application that I/O is complete. The user is then responsible to recognize the ERP suspend duplex pair message and secure that information at the secondary location. When the secondary site is depended upon to become operational in place of the primary site, a start-up procedure brings the secondary DASD on-line to the secondary host wherein sync status stored in the secondary DASD subsystem is retrieved for ensuring that out-of-sync volumes are not brought on-line for application allocation. This sync status merged with all ERP suspend duplex pair messages gives a complete picture of the secondary out-of-sync volumes.

Referring now to FIG. 1, a disaster recovery system 10 is shown having a primary site 14 and a secondary site 15, wherein the secondary site 15 is located, for example, 20 kilo-meters remote from the primary site 14. The primary site 14 includes a host processor or primary processor 1 having an application and system I/O and Error Recovery Program 2 running therein (hereinafter referred to as I/O ERP 2). The primary processor 1 could be, for example, an IBM Enterprise Systems/9000 (ES/9000) processor running DFSMS/MVS operating software and further may have several application programs running thereon. A primary storage controller 3, for example, an IBM 3990 Model 6 storage controller, is connected to the primary processor 1 via a channel 12. As is known in the art, several such primary storage controllers 3 can be connected to the primary processor 1, or alternately, several primary processors 1 can be attached to the primary storage controllers 3. A primary DASD 4, for example, an IBM 3390 DASD, is connected to the primary storage controller 3. Several primary DASDs 4 can be connected to the primary storage controller 3. The primary storage controller 3 and attached primary DASD 4 form a primary substorage system. Further, the primary storage controller 3 and the primary DASD 4 could be single integral units.

The secondary site 15 includes a secondary processor 5, for example, an IBM ES/9000, connected to a secondary storage controller 6, for example an IBM 3990 Model 3, via a channel 13. A DASD 7 is further connected to the secondary storage controller 6. The primary processor 1 is connected to the secondary processor 5 by at least one host-to-host communication link 11, for example, channel links or telephone T1/T3 line links, etc. The primary processor 1 may also have direct connectivity with the secondary storage controller 6 by, for example, multiple Enterprise Systems Connection (ESCON) links 9. As a result, the I/O ERP 2 can communicate, if required, with the secondary storage controller 6. The primary storage controller 3 communicates with the secondary storage controller 6 via multiple peer-to-peer links 8, for example, multiple ESCON links.

When a write I/O operation is executed by an application program running in the primary processor 1, a hardware status channel end/device end (CE/DE) is provided indicating the I/O operation completed successfully. Primary processor 1 operating system software marks the application write I/O successful upon successful completion of the I/O operation, thus permitting the application program to continue to a next write I/O operation which may be dependent upon the first or previous write I/O operation having successfully completed. On the other hand, if the write I/O operation was unsuccessful, the I/O status of channel end/device end/unit check (hereinafter referred to as CE/DE/UC) is presented to the primary processor 1 operating system software. Having presented unit check, the I/O ERP 2 takes control obtaining specific sense information from the primary storage controller 3 regarding the nature of the failed write I/O operation. If a unique error to a volume oc-

LINKPLAY EXHIBIT 1004
Page 1587 of 4822

curs then a unique status related to that error is provided to the I/O ERP 2. The I/O ERP 2 can thereafter perform new peer-to-peer synchronization error recovery for maintaining data integrity between the primary storage controller 3 and the secondary storage controller 6, or in the worst case, between the primary processor 1 and the secondary processor 5.

Referring to FIGs 2 and 3, the error recovery procedure is set forth. In FIG. 2, a step 201 includes an application program running in the primary processor 1 sending a data update to the primary storage controller 3. At step 203 the data update is written to the primary DASD 4, and the data update is shadowed to the secondary storage controller 6. At step 205 the duplex pair status is checked to determine whether the primary and secondary sites are synchronized. If the duplex pair status is in a synchronized state, then the data update is written to the secondary DASD 7 at step 207 while processing then continues at the primary processor 1 via application programs running thereat.

In the case that the duplex pair is in a "failed" state, then at step 209 the primary storage controller 3 notifies the primary processor 1 that duplex pair has suspended or failed. The duplex pair can become "failed" due to communication failure between the primary storage controller 3 and the secondary storage controller 6 via communication links 8. Alternatively, duplex pair can become "failed" due to errors in either the primary or secondary subsystem. If the failure is in the communication links 8, then the primary storage controller 3 is unable to communicate the failure directly to the secondary storage controller 6. At step 211 the primary storage controller 3 returns I/O status CE/DE/UC to the primary processor 1. The I/O ERP 2 quiesces the application programs hence taking control of the primary processor 1 at step 213 for error recovery and data integrity before returning control to the application requesting the write I/O operation.

FIG. 3 represents steps performed by the I/O ERP 2. The I/O ERP 2 issues a sense I/O to the primary storage controller 3 at step 221. The sense I/O operation returns information describing the cause of the I/O error, that is, the data description information is unique to the storage controllers or duplex pair operation regarding specific errors. In the event that the data description information indicates that the peer-to-peer communication links 8 have failed between the primary storage controller 1 and the secondary storage controller 6, then at step 223 the I/O ERP 2 issues a storage controller level I/O operation against the primary storage controller 3 and the secondary storage controller 6 indicating that the affected volume is to be placed in failed synchronous remote copy state. This secondary storage controller 6 is able to receive the state of the affected volume from the I/O ERP 2 via the - multiple ESCON links 9 or the host-to-host communication link 11. Consequently, the current status of the duplex pair operation is maintained at both the primary processor 1 and the secondary processor 5 in conjunction with applications running in the primary processor 1. Consoles 18 and 19 are provided for communicating information from the primary processor 1 and secondary processor 4, respectively, wherein the I/O ERP posts status information to both consoles 18 and 19.

Data integrity has been maintained at step 225 upon successful completion of the failed synchronous remote copy I/O operation to the primary storage controller 3 and the secondary storage controller 6. Therefore, if a recovery is attempted at the secondary site 15 the secondary storage controller 6 identifies the volume marked "failed synchronous remote copy" as not being useable until data on that volume are synchronized with other data in that synchronization group by data recovery means (conventional data base logs and/or journals for determining the state of that data on the volume).

Step 227 tests to determine whether the I/O ERP 2 received successful completion of the I/O operations at the primary storage controller 3 and the secondary storage controller 6 on the failed synchronous remote copy status update. Upon successful completion, the I/O ERP 2 returns control to the Primary processor 1 at step 229. Otherwise step 231 performs a next level recovery notification which involves notifying an operator, via the console 18, of the failed volume and that a status of that volume at either the primary storage controller 3 or the secondary storage controller 6 may not be correct. The notification is shadowed to the secondary site 15, via the console 19 or a shared DASD data set, for indicating the specific volume status there.

An error log recording data set is updated at step 233. This update is written to either the primary DASD 4 or some other storage location and is shadowed to the secondary site 15. Having completed the error recovery actions, the I/O ERP 2, at step 235, posts to the primary application write I/O operation a "permanent error" for causing the primary application an error normal "permanent error" recovery for the failed write I/O operation. Once the error is corrected, the volume states can be recovered, first to pending (recopy changed data) and then back to full duplex. The data may later be re-applied to the secondary DASD 7 once duplex pair is re-established.

When establishing a duplex pair a volume can be identified as CRITICAL according to a customer's needs. For a CRITICAL volume, when an operation results in failing a duplex pair, a permanent error failure of the primary volume is reported irrespective of the actual error's location. With CRIT=Y, all subsequent attempts to write to the primary DASD 405 of the failed pair will receive a permanent error, ensuring that no data is written to that primary volume that can-

LINKPLAY EXHIBIT 1004
Page 1588 of 4822

not also be shadowed to the paired secondary volume. This permits complete synchronization with the primary application actions and the I/O data operations when required.

Consequently, the disaster recovery system 10 described herein, introduces outboard synchronous remote copy such that a primary host process error recovery procedure having an I/O order (channel command word (CCW)) may change a status of a primary and secondary synchronous remote copy volume from duplex pair to failed duplex thereby maintaining data integrity for several types of primary and secondary subsystem errors. Storage based back-up, rather than application based back-up, wherein data updates are duplicated in real time has been provided. The disaster recovery system 10 also attempts several levels of primary/secondary status updates, including: (1) primary and secondary storage controller volume status updates; (2) primary and secondary host processor notification on specific volume update status via operator messages or error log recording common data sets; and (3) CRITICAL volume indication, future updates to the primary volume can be prevented if the volume pair goes failed duplex. Hence, real time, full error disaster recovery is accomplished.

Asynchronous remote data shadowing is used when it is necessary to further increase a distance between primary and secondary sites for reducing the probability that a single disaster will corrupt both primary and secondary sites, or when primary application performance impact needs to be minimized. While the distance between primary and secondary sites can now stretch across the earth or beyond, the synchronization of write updates across multiple DASD volumes behind multiple primary subsystems to multiple secondary subsystems is substantially more complicated. Record write updates can be shipped from a primary storage controller via a primary data mover to a secondary data mover for shadowing on a secondary storage subsystem, but the amount of control data passed therebetween must be minimized while still being able to re-construct an exact order of the record write updates on the secondary system across several storage controllers as occurred on the primary system across multiple DASD volumes behind several storage controllers.

FIG. 4 depicts an asynchronous disaster recovery system 400 including a primary site 421 and a remote or secondary site 431. The primary site 421 includes a primary processor 401, for example, an IBM ES/9000 running DFSMS/MVS host software. The primary processor 401 further includes application programs 402 and 403, for example, IMS and DB2 applications, and a primary data mover (PDM) 404. A common sysplex clock 407 is included in the primary processor 401 for providing a common reference to all applications (402, 403) running therein, wherein all system clocks or time sources (not shown) synchronize to the sysplex clock 407 ensuring all time dependent processes are properly timed relative to one another. The primary storage controllers 406, for example, synchronize to a resolution appropriate to ensure differentiation between record write update times, such that no two consecutive write I/O operations to a single primary storage controller 404 can exhibit the same time stamp value. The resolution, and not the accuracy, of the sysplex timer 407 is critical. The PDM 404, though shown connected to the sysplex timer 407, is not required to synchronize to the sysplex timer 407 since write I/O operations are not generated therein. A sysplex timer 407 is not required if the primary processor 401 has a single time reference (for example, a single multi-processor ES/9000 system).

A plurality of primary storage controllers 405, for example, IBM 3990 Model 6 storage controllers, are connected to the primary processor 401 via a plurality of channels, for example, fiber optic channels. Connected to each primary storage controller 405 is at least one string of primary DASDs 406, for example, IBM 3390 DASDs. The primary storage controllers 405 and the primary DASDs 406 form a primary storage subsystem. Each storage controller 405 and primary DASD 406 need not be separate units, but may be combined into a single drawer.

The secondary site 431, located for example, some thousands of kilometers remote from the primary site 421, similar to the primary site 421, includes a secondary processor 411 having a secondary data mover (SDM) 414 operating therein. Alternatively, the primary and secondary sites can be the same location, and further, the primary and secondary data movers can reside on a single host processor (secondary DASDs may be just over a fire-wall). A plurality of secondary storage controllers 415 are connected to the secondary processor 411 via channels, for example, fiber optic channels, as is known in the art. Connected to the storage controllers 415 are a plurality of secondary DASDs 416 and a control information DASD(s) 417. The storage controllers 415 and DASDs 416 and 417 comprise a secondary storage subsystem.

The primary site 421 communicates with the secondary site 431 via a communication link 408. More specifically, the primary processor 401 transfers data and control information to the secondary processor 411 by a communications protocol, for example, a virtual telecommunications access method (VTAM) communication link 408. The communication link 408 can be realized by several suitable communication methods, including telephone (T1, T3 lines), radio, radio/telephone, microwave, satellite, etc.

The asynchronous data shadowing system 400 encompasses collecting control data from the primary storage controllers 405 so that an order of all

LINKPLAY EXHIBIT 1004
Page 1589 of 4822

data writes to the primary DASDs 406 is preserved and applied to the secondary DASDs 416 (preserving the data write order across all primary storage subsystems). The data and control information transmitted to the secondary site 431, must be sufficient such that the presence of the primary site 421 is no longer required to preserve data integrity.

The applications 402, 403 generate data or record updates, which record updates are collected by the primary storage controllers 405 and read by the PDM 404. The primary storage controllers 405 each grouped its respective record updates for an asynchronous remote data shadowing session and provides those record updates to the PDM 404 via non-specific primary DASD 406 READ requests. Transferring record updates from the primary storage controllers 405 to the PDM 404 is controlled and optimized by the PDM 404 for minimizing a number of START I/O operations and time delay between each read, yet maximizing an amount of data transferred between each primary storage controller 405 and the primary processor 401. The PDM 404 can vary a time interval between non-specific READs to control this primary storage controller-host optimization as well as a currency of the record updates for the secondary DASDs 416.

Collecting record updates by the PDM 404, and transmitting those record updates to the SDM 414, while maintaining data integrity, requires the record updates to be transmitted for specific time intervals and in appropriate multiple time intervals with enough control data to reconstruct the primary DASDs 406 record WRITE sequence across all primary storage subsystems to the secondary DASDs 416. Re-constructing the primary DASDs 406 record WRITE sequences is accomplished by passing self describing records from the PDM 404 to the SDM 414. The SDM 414 inspects the self describing records for determining whether any records for a given time interval have been lost or are incomplete.

FIGs 5 and 6 show a journal record format created by the PDM 404 for each self describing record, including a prefix header 500 (FIG. 5), and a record set information 600 (FIG. 6) as generated by the primary storage controller 405. Each self describing record is further journaled by the SDM 414 for each time interval so that each self describing record can be applied in time sequence for each time interval to the secondary DASDs 416.

Referring now to FIG. 5, the prefix header 500, which is inserted at the front of each record set, includes a total data length 501 for describing the total length of the prefix header 500 and actual primary record set information 600 that is transmitted to the SDM 414 for each record set. An operational time stamp 502 is a time stamp indicating a start time for the operational set that the PDM 404 is currently processing. The operational time stamp 502 is generated by the PDM 404 (according to the sysplex timer 407) when performing a READ RECORD SET function to a set of the primary storage controllers 405. An I/O time 610 (FIG. 6) of the primary DASDs 406 write is unique for each primary storage controller 405 READ RECORD SET. The operational time stamp 502 is common across all storage controllers.

A READ RECORD SET command is issued by the PDM 404 and can be predicated upon one of the following conditions:

(1) Primary storage controller 405 attention interrupt based upon that primary storage controller predetermined threshold;

(2) Primary processor 401 timer interrupt based upon a predetermined time interval; or

(3) Record set information indicates additional information on outstanding record sets available but not yet read.

Condition (2) uses a timer interval to control how far behind the secondary system 431 executes during periods of low activity. Condition (3) occurs when the PDM 404 fails to drain all record sets during a processing interval which drives further activity for ensuring that the PDM 404 keeps up with primary storage controller 405 activity.

A time interval group number 503 is supplied by the PDM 404 to identify a time interval (bounded by operational time stamp 502 and a records read time 507) for which the current record sets belong (sets of records across all primary storage controllers 405 for a given time interval group form consistency groups). A sequence number within group 504 is derived based upon a hardware provided identification (to the PDM 404) of a write sequence order of application WRITE I/Os for primary storage controller 405 for each record set within a given time interval group 503. A primary SSID (substorage identification) 505 uniquely identifies the specific primary storage controller of the primary storage controllers 405 for each record set. A secondary target volume 506 is assigned by either the PDM 404 or the SDM 414 depending upon performance considerations. A records read time 507 supplies an operational time stamp that is common to all primary storage controllers 405 indicating an end time for the PDM 404 read record set process current interval.

The operational time stamp 502 and the records read time 507 are used by the PDM 404 to group sets of read record sets from each of the primary storage controllers 405. Time synchronization for grouping sets of read record sets is key only to the PDM 404 and as such, the PDM 404 could be synchronized to a central processing unit (CPU) clock running only the PDM 404 not attached to the syspelx timer 407. The PDM 404 does not write record updates, but the record updates, as stated previously, must be synchronized to a common time source.

Referring now to FIG. 6, the record set informa-

LINKPLAY EXHIBIT 1004
Page 1590 of 4822

tion 600 is generated by the primary storage controllers 405 and collected by the PDM 404. Update Specific Information 601-610, includes a primary device unit address 601 of each record indicating the actual primary DASD 406 that the record update occurred on. A cylinder number/head number (CCHH) 602 indicates a location on primary DASD 406 for each record update. Primary SSID 603, the primary storage controller session identifier is the same as primary SSID 505. Status flags 604 provide status information regarding whether specific data records 620 follow. Sequence numbers 605 and 630 assign a number to each record for indicating whether the entire record set has been read (all data transferred to the PDM 404). Primary DASD write I/O type 606 is an operation indicator identifying the type of write operation performed on each record, the operation indicators including: update write; format write; partial track records follow; full track data follows; erase command performed; or write any performed. Search argument 607 indicates initial positioning information for the first read record set data record 620. A sector number 608 identifies that sector that the record was updated at. Count field 609 describes a number of specific record data fields 620 that follow. A host application time when the primary DASD 406 write update occurred is recorded in time of updates 610. Specific record data 620 provides a count/key/data (CKD) field of each record update. Lastly, the sequence number 630 is compared to the sequence number 605 for indicating whether the entire read record set was transferred to the PDM 404.

The update records are handled in software groups called consistency groups so that the SDM 414 can copy the record updates in the same order they were written at the primary DASDs 406. The information used for creating the consistency groups (across all record sets collected from all storage controllers 405) includes the: operational time stamp 502; time interval group number 503; sequence number within group 504; primary controller SSID 505; records read time 507; primary device address 601; the primary SSID 603; and the status flags 604. The information used for determining whether all records for a time interval group have been received for each storage controller 405 at the SDM 414 includes the: time interval group number 503; sequence number within group 504; physical controller ID 505; the primary SSID 603; and a total number of read record sets returned from each primary storage controller 405 for each operational time interval. The information necessary to place record updates on the secondary DASDs 416 equivalently to the primary DASDs 406 record updates with full recover possible includes the: secondary target volume 506; CCHH 602; primary DASD write I/O type 606; search argument 607; sector number 608; count 609; time of updates 610; and the specific record data 620.

FIGs 7 and 8 show a state table 700 and a master journal 800, respectively, for describing a current journal contents, which simplifies recovery time and journal transfer time. The state table 700 provides configuration information, collected by and common to the PDM 404 and SDM 414, and includes primary storage controller session identifiers (SSID numbers) and the volumes therein, and the corresponding secondary storage controller session identifiers and the corresponding volumes. Thus the configuration information tracks which primary volumes 710 or primary DASD extents map to secondary volumes 711 or secondary DASD extents. With a simple extension to the state table 700 indicating partial volume extents 712 (CCHH to CCHH), partial volume remote copy can be accomplished using the same asynchronous remote copy methods described herein, but for a finer granularity (track or extent) than full volume.

The master journal 800 includes: consistency group number; location on journal volumes; and operational time stamp. The master journal 800 further maintains specific record updates as grouped in consistency groups. The state table 700 and master journal 800 support disaster recovery, and hence must be able to operate in a stand-alone environment wherein the primary system 401 no longer exists.

A time stamp control is placed at the front and back of each master journal 800 to ensure that the entire control entry was successfully written. The time stamp control is further written to the secondary DASDs 417. The control elements include dual entries (1) and (2), wherein one entry is always a current entry, for example:

(1) Timestamp control | Control Info | Timestamp Control

(2) Timestamp Control | Control Info | Timestamp Control.

At any point in time either entry (1) or (2) is the current or valid entry, wherein a valid entry is that entry with equal timestamp controls at the front and back. Disaster recovery uses the valid entry with the latest timestamp to obtain control information. This control information, along with state information (environmental information regarding storage controllers, devices, and applied consistency groups), is used for determining what record updates have been applied to the secondary storage controllers 415.

After all read record sets across all primary storage controllers 405 for a predetermined time interval are received at the secondary site 431, the SDM 414 interprets the received control information and applies the received read record sets to the secondary DASDs 416 in groups of record updates such that the record updates are applied in the same sequence that those record updates were originally written on the primary DASDs 406. Thus, all primary application order (data integrity) consistency is maintained at the secondary site 431. This process is hereinafter refer-

LINKPLAY EXHIBIT 1004
Page 1591 of 4822

red to as forming consistency groups. Forming consistency groups is based on the following assumptions: (A) application writes that are independent can be performed in any order if they do not violate controller sequence order; (B) application writes that are dependent must be performed in timestamp order, hence an application cannot perform a dependent write number two before receiving control unit end, device end from write number one; and (C) a second write will always be either (1) in a same record set consistency group as a first write with a later timestamp or (2) in a subsequent record set consistency group.

Referring to FIG. 9, an example of forming a consistency group (the consistency group could be formed at either the primary site 421 or secondary site 431), for example, for storage controllers SSID 1, SSID 2, and SSID 3 is shown (any number of storage controllers can be included but three are used in this example for clarity). Time intervals T1, T2 and T3 are assumed to occur in ascending order. An operational time stamp 502 of time interval T1 is established for storage controllers SSID 1, SSID 2 and SSID 3. The PDM 404 obtains record set data from storage controller SSIDs 1, 2, and 3 for time interval T1-T3. The record sets for SSIDs 1, 2, and 3 for time interval T1 are assigned to time interval group 1, G1 (time interval group number 503). The sequence number within group 504 is shown for each SSID 1, 2, and 3, wherein SSID has three updates as 11:59. 12:00, and 12:01, SSID 2 has two updates at 12:00 and 12:02, and SSID 3 has three updates at 11:58, 11:59, and 12:02. Record sets of time intervals T2 and T3 are listed but example times of updates are not given for simplicity.

Consistency group N can now be formed based upon the control information and record updates received at the secondary site 431. In order to ensure that no record update in time interval group number one is later than any record update of time interval group number two, a min-time is established which is equal to a the earliest read record set time of the last record updates for each storage controller SSID 1, 2, and 3. In this example then, min-time is equal to 12:01. Any record updates having a read record set time greater than or equal to min-time is included in the consistency group N + 1. If two record update times to a same volume were equal, though unlikely given sufficient resolution of the sysplex timer 407, the record update having the earlier sequence number within the time interval group N is kept with that group for consistency group N. The record updates are now ordered based upon read record set times. Record updates having equal times will cause the record update having the lower sequence number to be place before the later sequence numbered record update. Alternatively, record updates having equal time stamps, but to differing volumes, may be ordered arbitrarily as long as they are kept in the same consistency group.

If a primary storage controller 405 fails to complete a response to a read record set during a specified time interval, then a consistency group cannot be formed until that primary storage controller 405 completes. In the event that the primary storage controller 405 fails to complete its operation, then a missing interrupt results causing a system missing interrupt handler to receive control and the operation will be terminated. On the other hand, if the primary storage controller 405 timely completes the operation then the I/O will be driven to completion and normal operation will continue. Consistency group formation expects that write operations against the primary storage controllers 405 will have time stamps. Some programs, however, will cause writes to be generated without time stamps, in which case the primary storage controller 405 will return zeros for the time stamp. Consistency group formation can bound those records without time stamps based upon the timestamp that the data was read. If too many record updates without time stamps occur over a time interval such that the record updates are not easily bounded by consistency group times, then an error that the duplex volumes are out of synchronization may result.

FIGs 10 and 11 are flow diagrams presenting the method of forming consistency groups. Referring to FIG. 10, the process starts at step 1000 with the primary site 421 establishing remote data shadowing to occur. At step 1010 all application I/O operations are time stamped using the sysplex timer 407 as a synchronization clock (FIG. 4). The PDM 404 starts a remote data shadowing session with each primary storage controller 405 at step 1020 which includes identifying those primary volumes that will have data or records shadowed. Record set information 600 is trapped by the primary storage controllers 405 for each application WRITE I/O operation (see FIG. 6) by step 1030.

Step 1040 involves the PDM 404 reading the captured record set information 600 from each primary storage controller 405 according to a prompt including an attention message, a predetermined timing interval, or a notification of more records to read as described earlier. When the PDM 404 begins reading record sets, at step 1050, the PDM 404 prefixes each record set with a prefix header 500 (see FIG. 5) for creating specific journal records (a journal record includes the prefix header 500 and the record set information 600). The journal records contain the control information (and records) necessary for forming consistency groups at the secondary site 431 (or at the primary site 421).

At step 1060 the PDM 404 transmits the generated journal records to the SDM 414 via the communications link 408 (or within the same data mover system if the consistency groups are formed therein). The SDM 414 uses the state table 700 at step 1070

LINKPLAY EXHIBIT 1004
Page 1592 of 4822

to gather the received record updates by group and sequence numbers for each time interval group and primary storage controller 405 established for the data shadowing session. The SDM 414 inspects the journal records at step 1080 to determine whether all record information has been received for each time interval group. If the journal records are incomplete, then step 1085 causes the SDM 414 to notify the PDM 404 to resend the required record sets. If the PDM 404 is unable to correctly resend, then the duplex volume pair is failed. If the journal records are complete, then step 1090 is performed which encompasses the SDM 414 forming the consistency groups.

Referring to FIG. 11, steps 1100-1160 representing step 1090 (FIG. 10) for forming consistency groups is shown. Consistency group formation starts at step 1100 wherein each software consistency group is written to an SDM 414 journal log ("hardened") on the secondary DASD 417 (FIG. 4). Step 1110 performs a test for determining whether the time interval groups are complete, that is, each primary storage controller 405 must have either presented at least one read record set buffer or have confirmation from the PDM 404 that no such record updates were placed in the record set buffer, and all read record set buffers with data (or null) must have been received by the SDM 414. If a time interval group is incomplete, then step 1110 retries reading the record sets from the primary storage controller 405 until the required data is received. If errors occur, a specific duplex volume pair or pairs may be failed. Having received complete time interval groups, step 1120 determines a first consistency group journal record. The first (or current) consistency group journal record is that record which contains the earliest operational time stamp 502 and the earliest time of update 610 of all records having equal operational time stamps 502.

Step 1130 inspects the records contained in the current consistency group journal record to determine which record will be the last record to be included therein (some records will be dropped and included in the next consistency group journal record). The last record in the current consistency group journal record is determined as a minimum update time (min-time) of the maximum update times for each primary storage controller 405 (that is, the last update of each primary storage controller 405 is compared and only the earliest of these remains in the current consistency group journal record).

Those remaining record updates in the current consistency group journal record are ordered according to time of update 610 and sequence number within group 504 by step 1140. A primary storage controller 405 that had no record updates does not participate in the consistency group. At step 1150, the remaining record updates of the current consistency group (having update times later than min-time) are passed to the next consistency group. Each sequence number within a group 504 should end with a null buffer indicating that all read record sets have been read for that operational time interval. If the null buffer is absent, then the step 1120 of defining the last record in the current software consistency group, coupled with the records read time 507 and time of update 610 can be used to determine the proper order of the application WRITE I/O operations across the primary storage controllers 405.

Step 1160 represents a back-end of the remote data shadowing process wherein specific write updates are applied to secondary DASDs 416 under full disaster recovery constraint. If when writing the updates to the secondary DASDs 416 an I/O error occurs, or the entire secondary site 431 goes down and is re-initialized, then the entire consistency group that was in the process of being written can be re-applied from the start. This permits the remote shadowing to occur without having to track which secondary DASDs 416 I/Os have occurred, which I/Os have not occurred, and which I/Os were in process, etc.

A key component of step 1160 is that the PDM 414 causes the records to be written efficiently to the secondary DASDs 416 so that the secondary site 431 can keep pace with the primary site 421. The requisite efficiency is accomplished, in part, by concurrently executing multiple I/O operations to different secondary DASDs 416. Serially writing one secondary DASD 416 at a time would cause the secondary site 431 to fall too far behind the primary site 421. Yet more efficiency is gained at the secondary site 431 by writing the records for each consistency group destined for a single secondary device via a single channel command word (CCW) chain. Within each single CCW chain, the I/O operations therein can be further optimized as long as those I/O operations to each secondary DASD 416 data track are maintained in the order of occurrence on the primary volumes.

Optimizing secondary I/O operations for specific consistency groups and within single CCW chains is based in part upon the pattern of primary write I/O operations, and in part upon the physical characteristics of the secondary DASDs 416. Optimization may vary somewhat depending upon whether secondary DASD 415 is count/key/data (CKD), extended count/key/data (ECKD), fixed block architecture (FBA), etc. Consequently, a number of WRITE I/Os (m) to a primary DASD 406 volume during a given time interval can be reduced to a single START I/O operation to a secondary DASD 416 volume. This optimization of the number of START I/Os to the secondary storage controllers 415 of m:1 can allow the secondary DASDs 416 to catch up with and thereby closer shadow the record updates at the primary site 421.

A key to successful remote data shadowing, and hence secondary I/O optimization, is minimizing unrecoverable errors in any of the concurrent multiple I/O operations to secondary DASDs 416 so that con-

LINKPLAY EXHIBIT 1004
Page 1593 of 4822

sistent copies are available for recovery. A failure in a given secondary write could permit a later dependent write to be recorded without the conditioning write (e.g., a log entry indicating that a data base record has been updated when in reality the actual update write for the data base had failed violates the sequence integrity of the secondary DASD 416 copy).

A failed secondary 416 copy is unusable for application recovery until that failure to update has been recovered. The failed update could be corrected by having the SDM 414 request a current copy from the PDM 404. In the mean time the secondary data copy is inconsistent and hence unusable until the PDM 404 responds with the current update and all other previous updates are processed by the PDM 414. The time required to recover the failed update typically presents an unacceptably long window of non-recovery for adequate disaster recovery protection.

Effective secondary site 431 I/O optimization is realized by inspecting the data record sets to be written for a given consistency group and building chains based upon rules of the particular secondary DASD 416 architecture, for example, an ECKD architecture. The optimization technique disclosed herein simplifies recovery from I/O errors such that when applying a consistency group, if an I/O error occurs, the CCW chain can be re-executed, or in the event of a secondary initial program load (IPL) recovery, the entire consistency group can be re-applied without data loss.

FIG. 12 summarizes full consistency group recovery (FCGR) rules for building CCW chains for all WRITE I/O combinations for an ECKD architecture, wherein CCHHR record format is used (cylinder number, head number, record number). FIG. 12 is created by inspecting each possible combination of WRITE I/O operations to a DASD track within a consistency group's scope. The FCGR rules of FIG. 12, described in FIG.s 13A and 13B, are then followed to govern data placement (secondary DASD 416 I/O write CCW chains) for yielding full recovery for an error in applying a consistency group. The FCGR rules depicted in FIG. 12 would be extended appropriately as new WRITE I/O operations are added. These rules can exist in hardware or software at the secondary site 431. The FCGR rules advantageously reduces READ record set to a same DASD track analysis to an inspection of the primary DASD 406 WRITE I/O type, search argument, and count and key fields.

If a DASD track is written without inspecting the consistency group write operations as shown in FIG. 12, then previously written data records potentially cannot be re-written. For example, assume that a chain includes:

WRITE UPDATE to record five; and

FORMAT WRITE to record one,

wherein record one and record five occur on the same DASD track with record one preceding record five. Record five is updated by an UPDATE WRITE CCW and a FORMAT WRITE I/O CCW updates record one erasing a remainder of the track thus deleting record five. If this chain had to be re-executed, a LOCATE RECORD CCW that will position to the beginning of record five will no longer have a positioning point (record five no longer exists), and the chain is not fully recoverable from the beginning. Since the write operations have already been successful at the primary site 421, always being able to apply an entire consistency group on the secondary DASD 416 is required to maintain data consistency and integrity.

FIG. 14, steps 1410 through 1470, provides more detail as to the process represented by step 1160 of FIG. 11, while using the FCGR rules defined in FIG. 12. At step 1410 the SDM 414 divides the records of the current consistency group into two categories. A first category includes I/O orders directed to a same secondary DASD volume, and a second category includes I/O orders of those records in the first category that are directed to a same CCHH (i.e., records being updated to a same DASD track).

Having categorized the records of the current consistency group, step 1420 conforms application WRITE I/Os and SDM 414 WRITE I/Os to the architecture of the secondary DASDs 416, for example, to ECKD architecture FCGR rules (see FIG. 12) for identifying data placement on a track and track/record addressing. The SDM 414 groups secondary DASD WRITE I/O operations to the same volumes into single I/O CCW chains at step 1430. Step 1440 involves moving the head disk assembly (HDA) of each secondary DASD 416 according to search arguments and specific record data (CKD fields) for the actual secondary DASD 416 writes.

Step 1450 compares READ SET BUFFERS one and two for those records making up the second categories (there typically will be a plurality of second categories, one for each track receiving records), using the FCGR rules of FIG. 12 for determining whether a subsequent write operation invalidates a previous write operation or DASD search argument (positioning at a record that is now erased, etc.). The READ SET BUFFERS one and two contain adjacent read record sets. Following the FCGR rules ensures that the SDM 414 can re-write an entire consistency group, in the event of an error, without re-receiving record updates from the primary site 421. After the SDM 414 applies the current consistency group to the secondary DASD 416, step 1460 updates the state table (FIG. 7) and the master journal (FIG. 8).

The remote copy process continues in real time as step 1470 gets a next consistency group (which becomes the current consistency group) and returns processing to step 1410. The remote copy process will stop if the primary site 421 to secondary site 431 communication terminates. The communication may terminate if volume pairs are deleted from the process by the PDM 404, the primary site is destroyed

LINKPLAY EXHIBIT 1004
Page 1594 of 4822

(disaster occurs), an orderly shutdown is performed, or a specific takeover action occurs at the secondary site 431. Consistency groups journaled on the secondary site 431 can be applied to the secondary DASD 416 during a takeover operation. The only data lost is that data captured by the primary site 421 that has not been completely received by the SDM 414.

In summary, synchronous and asynchronous remote data duplexing systems have been described. The asynchronous remote data duplexing system provides storage based, real time data shadowing. A primary site runs applications generating record updates, and a secondary site, remote from the primary site, shadows the record updates and provides disaster recovery for the primary site. The asynchronous remote data duplexing system comprises a sysplex timer for synchronizing time dependent processes in the primary site, and a primary processor at the primary site for running the applications, the primary processor having a primary data mover therein. A plurality of primary storage controllers are coupled to the primary processor for issuing write I/O operations for each record update, each primary storage controller DASD write I/O operation being synchronized to the sysplex timer. A plurality of primary storage devices receive the write I/O operations and store the record updates therein accordingly. The primary data mover collects record set information from the plurality of primary storage controllers for each record update and appends a prefix header to a predetermined group of record set informations. The prefix header and predetermined group of record set informations form the self describing record sets. Each record set information includes a primary device address, a cylinder number and head number (CCHH), a record update sequence number, a write I/O type, a search argument, a sector number, and a record update time. The prefix header includes a total data length, an operational time stamp, a time interval group number, and a records read time. A secondary processor at the secondary site has a secondary data mover, the secondary data mover receiving the self describing record sets from the primary site. A plurality of secondary storage controllers are coupled to the secondary processor, and a plurality of secondary storage devices are coupled to the secondary storage controllers for storing the record updates copies. The secondary data mover determines whether the transmitted self describing record sets are complete and forms consistency groups from the self describing record sets and provides the record updates from each consistency group to the plurality of secondary storage controllers for writing to the plurality of secondary storage devices in an order consistent with a sequence that the record updates were written to the plurality of primary storage devices.

While the invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention as defined in the appended claims. For example, the consistency groups have been described as being formed by the secondary data mover based upon received self describing record sets, however, the consistency groups could be formed at the primary site based upon write record sets or elsewhere in the secondary site. The formats the storage devices at the primary and secondary sites need not be identical. For example, CKD records could be converted to fixed block architecture (FBA) type records, etc. Nor are the storage devices meant to be limited to DASD devices.

Claims

1. A method of providing asynchronous data duplexing, wherein data updates generated by one or more applications running in a primary processor are received by a primary storage subsystem, the primary storage subsystem causing I/O write operations to write each data update therein, each write I/O operation being time stamped, the time stamps synchronized by a common timer, and wherein a secondary system, whether local to or remote from the primary processor, shadows the data updates in sequence consistent order such that the secondary site is available for disaster recovery purposes, the method comprising the steps of:

   (a) time stamping each write I/O operation occurring in the primary storage subsystem;

   (b) capturing write I/O operation record set information from the primary storage subsystem for each data update;

   (c) generating self describing record sets from the data updates and respective record set informations, the self describing record sets containing sufficient control information to enable recreation of a sequence of the write I/O operations solely by the secondary system;

   (d) grouping the self describing record sets into interval groups, each interval group being measured from an operational time stamp start time and continuing for a predetermined interval threshold; and

   (e) selecting a current consistency group as that interval group of self describing record sets having an earliest operational time stamp, the individual data updates being ordered within the current consistency group based upon time sequences of the I/O write operations in the primary storage subsystem.

LINKPLAY EXHIBIT 1004
Page 1595 of 4822

2. The method as claimed in claim 1 wherein the step (b) further includes initiating sessions with the primary storage subsystem based upon the operational time stamps for identifying a starting time for each interval group, each interval group being bounded by consecutive operational time stamps.

3. The method as claimed in claim 1 or claim 2 wherein the step (d) includes adding a prefix header describing each interval group.

4. The method as claimed in any preceding claim further comprising a step (f) transmitting the interval groups of self describing record sets to the secondary site .

5. The method as claimed in claim 4 further comprising a step (g) determining at the secondary site whether each received self describing record set is complete.

6. The method as claimed in claim 5 wherein the step (g) further includes the secondary site requesting the primary site to re-transmit any missing data updates if the secondary site determined a self describing record set is incomplete.

7. The method as claimed in claim 6 further comprising a step (h) determining at the secondary site whether each time interval group is complete.

8. The method as claimed in claim 7 wherein the step (h) further includes the secondary site requesting the primary site to re-send a missing record set if the secondary site determined that an interval group was incomplete.

9. The method as claimed in claim 8 further comprising a step (i) writing the received data updates at the secondary site to the secondary storage subsystem according to the sequence of the corresponding write I/O operations at the primary site as ordered in the consistency groups.

10. A method for providing remote data shadowing for disaster recovery purposes in a data processing system including a primary site having a primary processor running a primary data mover and applications generating record updates, the primary processor coupled to a primary storage subsystem having storage devices for storing the record updates according to write I/O operations issued by the primary processor to the primary storage subsystem, the primary site further including a common system timer for synchronizing time dependent operations in the primary site, the system further including a secondary site having a secondary processor communicating with the primary processor, and a secondary storage subsystem for storing copies of the record updates in sequence consistent order, the method comprising the steps of:
    (a) time stamping each write I/O operation in the primary storage subsystem;
    (b) establishing a session with each storage device in the primary storage subsystem;
    (c) capturing record set information from each storage device in the primary storage subsystem;
    (d) reading record sets and respective record set information in the primary data mover;
    (e) prefixing each record set with a header and creating self describing record sets therefrom;
    (f) transmitting the self describing record sets to the secondary processor in time interval groups according to predetermined time intervals;
    (g) forming consistency groups from the self describing record sets; and
    (h) shadowing the record updates of each consistency group to the secondary storage subsystem in a sequence consistent order.

11. The method as claimed in claim 10 wherein the record sets are transmitted to the secondary processor asynchronously.

12. The method as claimed in claim 10 or claim 11 wherein the step (g) is performed at the secondary site.

13. The method as claimed in any of claims 10 to 12 wherein the step (f) further includes determining at the secondary site whether each received self describing record set is complete.

14. The method as claimed in claim 13 wherein the step (f) further includes requesting the primary site to re-transmit any missing record updates if the primary site determined a received self describing record set is incomplete.

15. The method as claimed in any of claims 10 to 14 further comprising a step (i) determining at the secondary site whether each time interval group is complete.

16. The method as claimed in claim 15 wherein the step (i) further includes requesting the primary site to re-send a missing record set if the secondary site determined that a interval group was incomplete.

LINKPLAY EXHIBIT 1004
Page 1596 of 4822

17. The method as claimed in any of claims 10 to 16 wherein the step (c) identifies, in the record set information, a physical location on the primary storage devices where each record update is stored.

18. The method as claimed in claim 17 wherein the step (c) identifies, in the record set information, a sequence and time of update of each record update stored to the primary storage devices within the session.

19. The method as claimed in any of claims 10 to 18 wherein the step (e) identifies, in the prefix header, an interval group number for the session and sequence within group for each record update referred to therein.

20. A primary data processing system having a primary processor running one or more applications, the one or more applications generating record updates, the primary processor generating self describing record sets therefrom, the self describing record sets being sent to a secondary system, the secondary system shadowing the record updates in sequence consistent order based upon the self describing record sets for real time disaster recovery purposes, the primary processor being coupled to a primary storage subsystem wherein the primary storage subsystem receives the record updates and executes write I/O operations for storing each record update therein, the primary processor comprising:

a timer for providing a common time source to the applications and primary storage subsystem for synchronization purposes; and

primary data mover means prompting the primary storage subsystem for providing record set information to the primary data mover means for each record update, the primary data mover means grouping a plurality of record updates and each corresponding record set information into time interval groups, and inserting a prefix header thereto, each time interval group the self describing record sets. interval groups, and inserting a prefix header thereto, each time interval group the self describing record sets.

21. The primary system as claimed in claim 20 wherein the primary storage subsystem comprises:

a plurality of primary storage controllers, the plurality of primary storage controllers issuing the write I/O operations; and

a plurality of primary storage devices coupled to the plurality of primary storage controllers.

22. The primary system as claimed in claim 20 or claim 21 wherein the primary data mover means collects record set information for each write I/O operation for each primary storage controller of the plurality of primary storage controllers participating with each time interval group.

23. The primary system as claimed in any of claims 20 to 22 wherein each write I/O operation is time-stamped in the primary processor relative to the sysplex timer, each write I/O operation being issued to a primary storage controller of the plurality of primary storage controllers, each primary storage controller preserving the time-stamp and returning that time-stamp in the corresponding read record set to the primary data mover means.

24. The primary system as claimed in claim 23 wherein each record set information identifies a corresponding record update's physical location on a primary storage device of the plurality of primary storage devices.

25. The primary system as claimed in claim 23 or claim 24 wherein each record set information identifies a corresponding record update's primary subsystem identification, primary device address, cylinder number, and head number.

26. The primary system as claimed in claim 23 wherein the primary data mover means identifies a relative sequence of each write I/O update across all primary storage controllers participating in a time interval group.

27. The primary system as claimed in claim 26 wherein the primary data mover means creates a state table for journaling record updates and cross referencing a storage location of each record update on the primary system and the secondary system.

28. The primary system as claimed in claim 26 wherein the primary data mover means communicates the state table to the secondary system.

29. A remote data shadowing system including a primary site and a secondary site, the secondary site asynchronously shadowing record updates of the primary site in real time for disaster recovery purposes, the record updates generated by applications running at the primary site, the primary site comprising:

a sysplex timer;

a primary processor running the applications generating the record updates and issuing a corresponding write I/O operation for each record update, the primary processor having a pri-

LINKPLAY EXHIBIT 1004
Page 1597 of 4822

mary data mover therein;

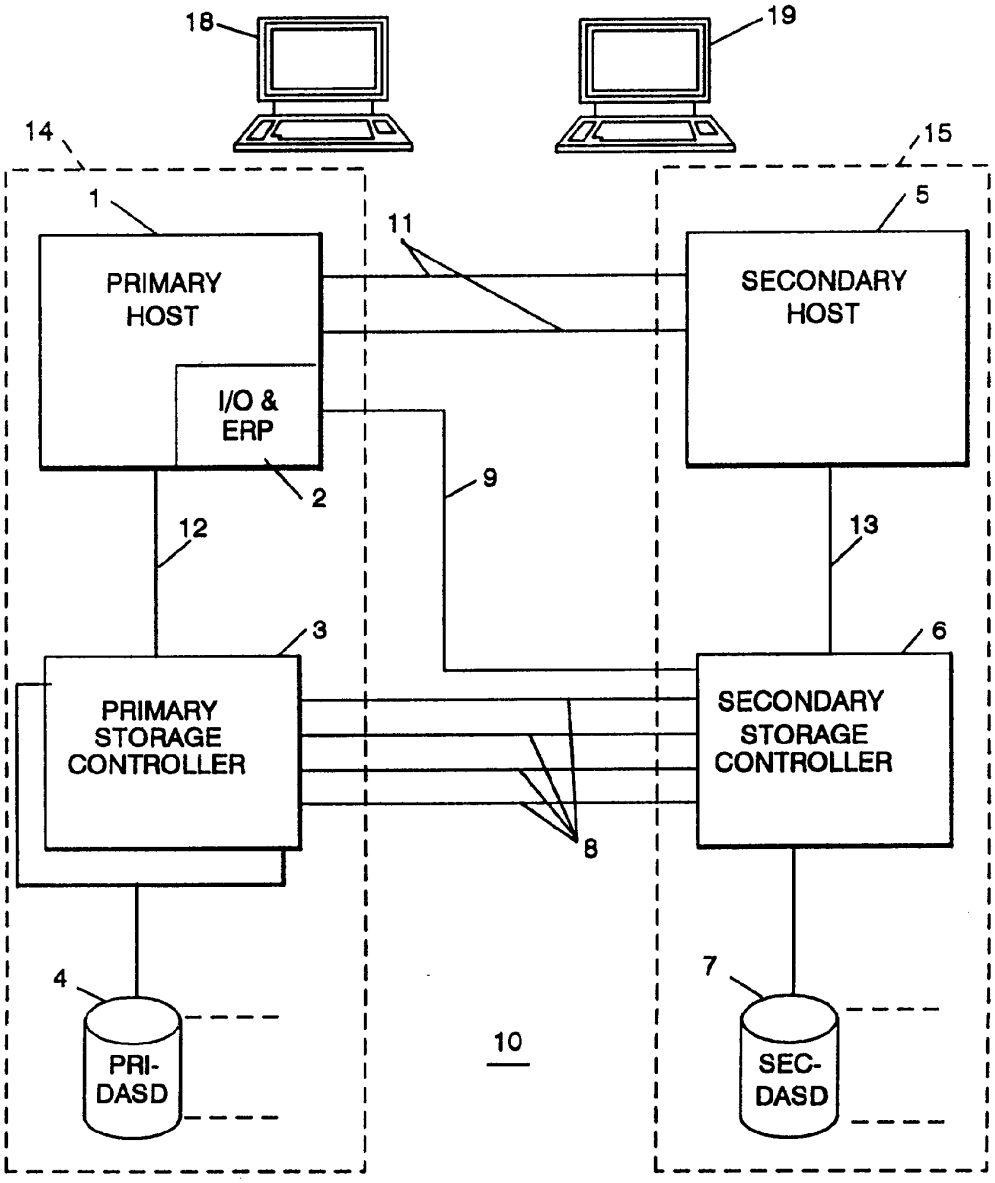
a plurality of primary storage controllers directed to store the record updates, the plurality of primary storage controllers executing the issued write I/O operation for each record update; and

a plurality of primary storage devices receiving and storing the record updates therein according to the corresponding write I/O operations,

wherein the primary processor and each write I/O are time-stamped by the primary processor, as synchronized by the sysplex timer, such that write I/O operations are accurately sequence ordered relative to each other, the primary data mover collecting sets of record updates and combining each record set information as provided by each one of the plurality of primary storage controllers with the corresponding record update, each record set information including a relative sequence and time of each corresponding write I/O operation, the primary data mover collecting record updates into time interval groups and inserting a prefix header to each time interval group, wherein the prefix header includes information identifying the record updates included in each time interval group, each record set information and prefix header combined for creating self describing record sets, the self describing record sets being transmitted to the secondary site, wherein the self describing record sets provide information adequate for the secondary site to shadow the record updates therein in sequence consistent order without further communications from the primary site.

LINKPLAY EXHIBIT 1004
Page 1598 of 4822

FIG. 1



LINKPLAY EXHIBIT 1004
Page 1599 of 4822

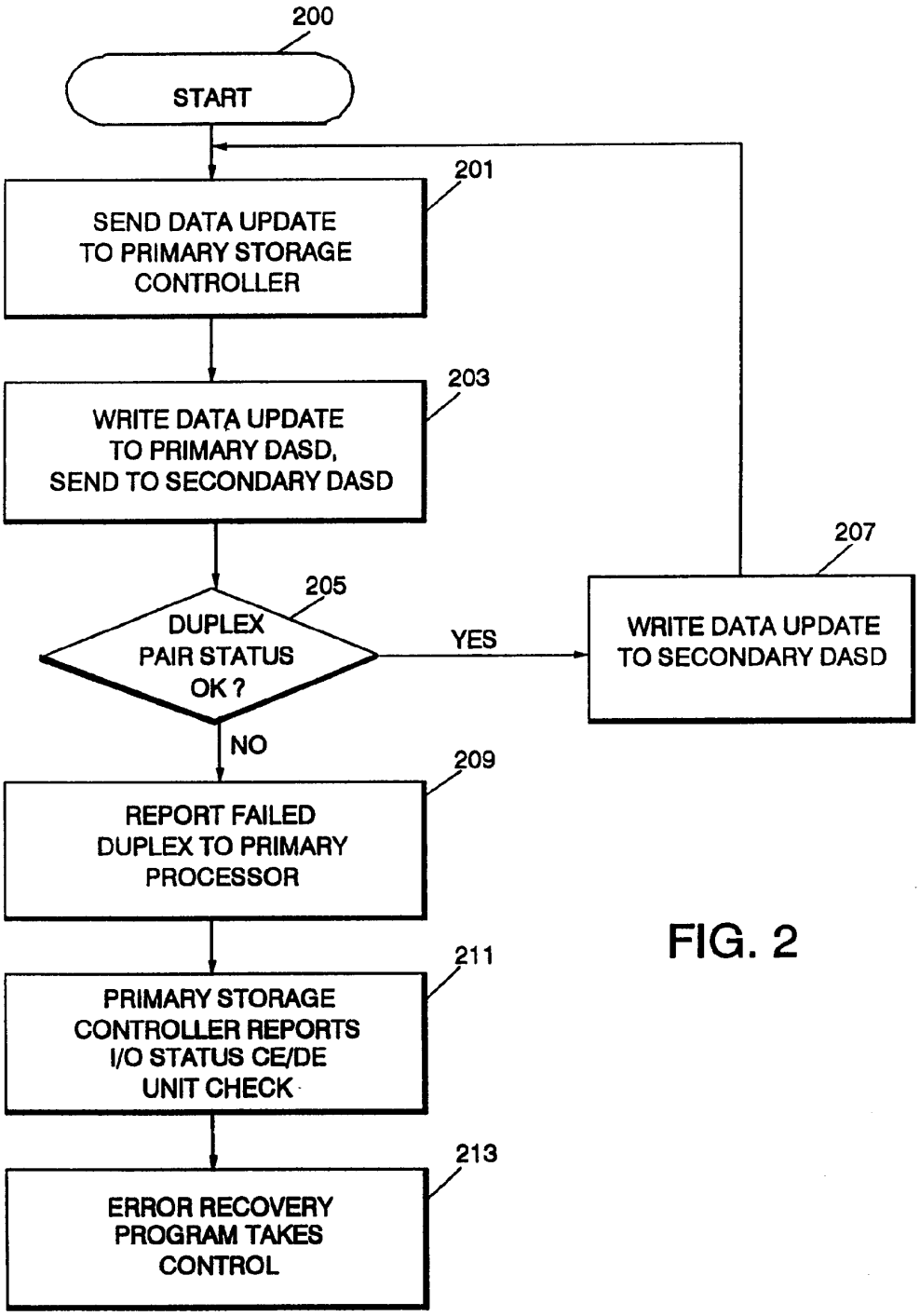


FIG. 2

LINKPLAY EXHIBIT 1004
Page 1600 of 4822

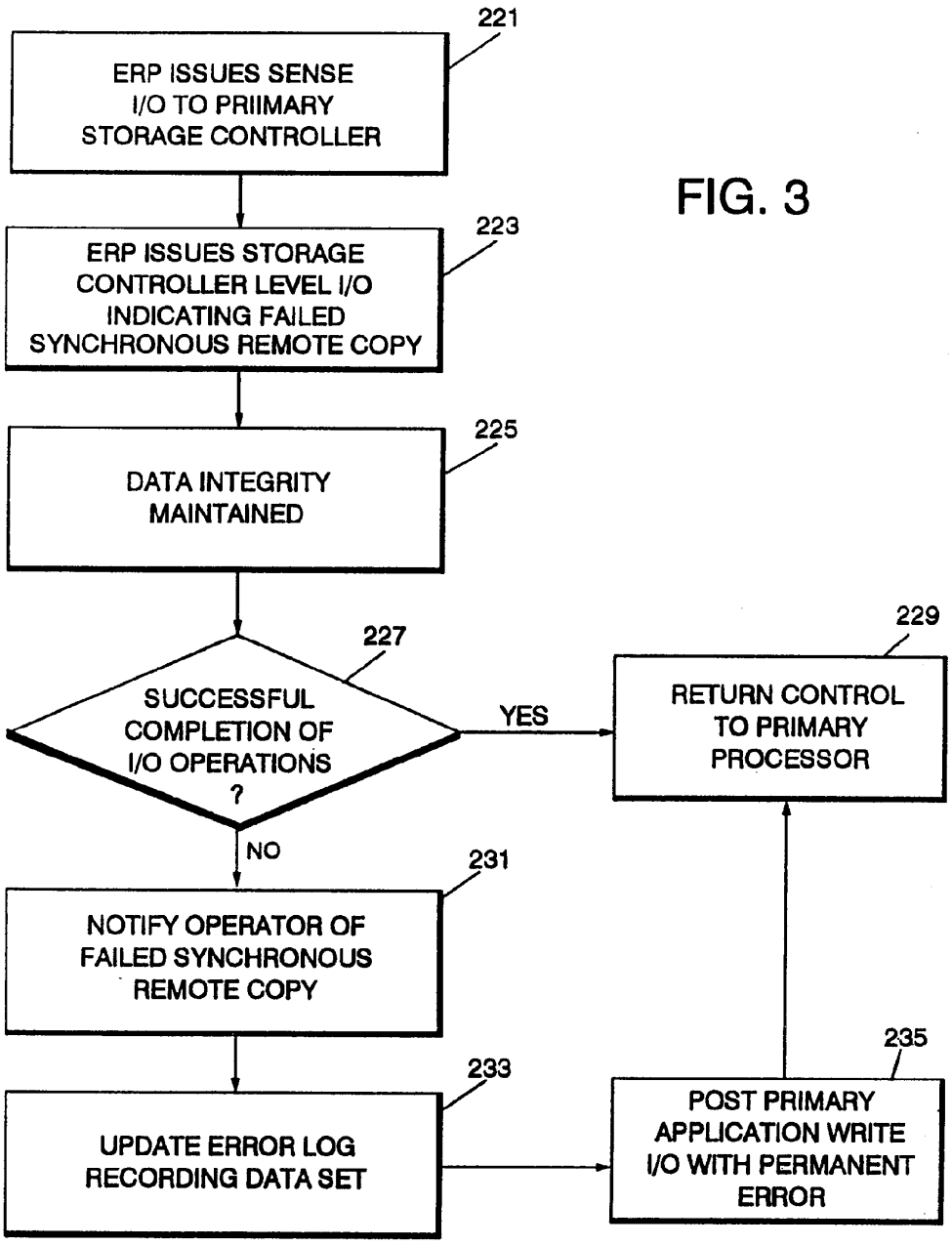


FIG. 3

LINKPLAY EXHIBIT 1004
Page 1601 of 4822

FIG. 4

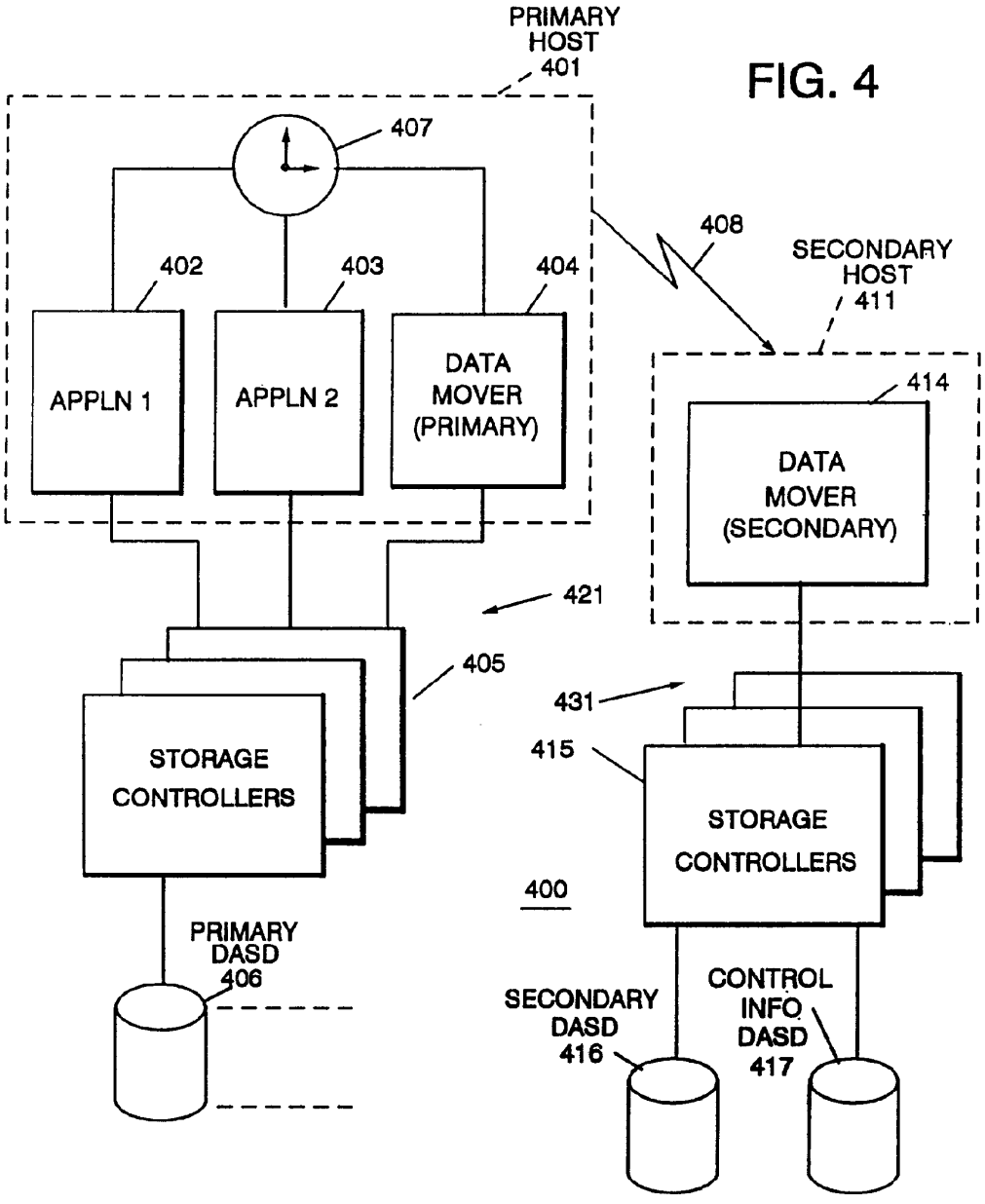
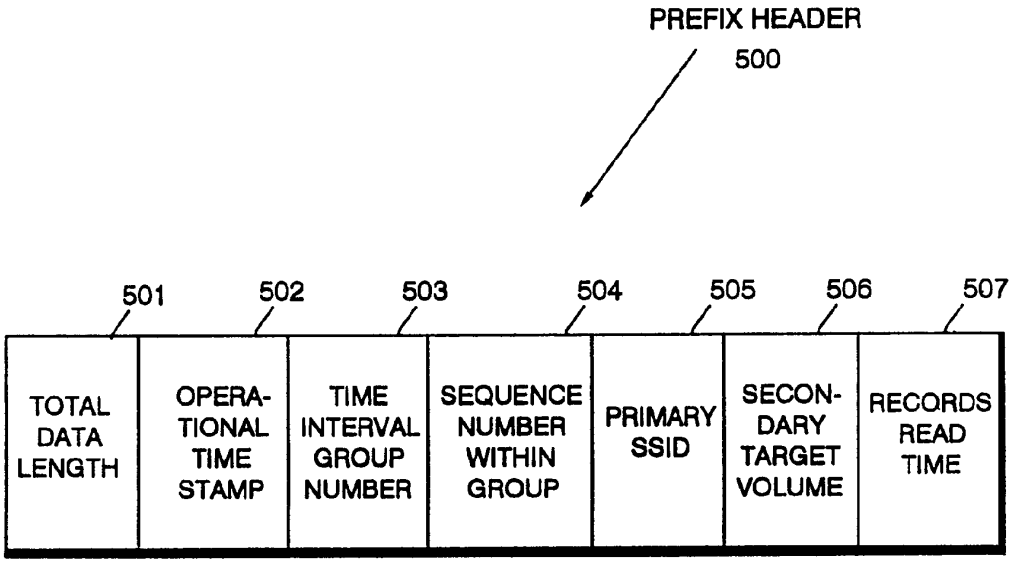

PRIMARY HOST
401
407
408
SECONDARY HOST
411
402
403
404
APPLN 1
APPLN 2
DATA MOVER (PRIMARY)
414
DATA MOVER (SECONDARY)
421
405
431
415
STORAGE CONTROLLERS
STORAGE CONTROLLERS
400
PRIMARY DASD
406
SECONDARY DASD
416
CONTROL INFO DASD
417

FIG. 5



LINKPLAY EXHIBIT 1004
Page 1603 of 4822

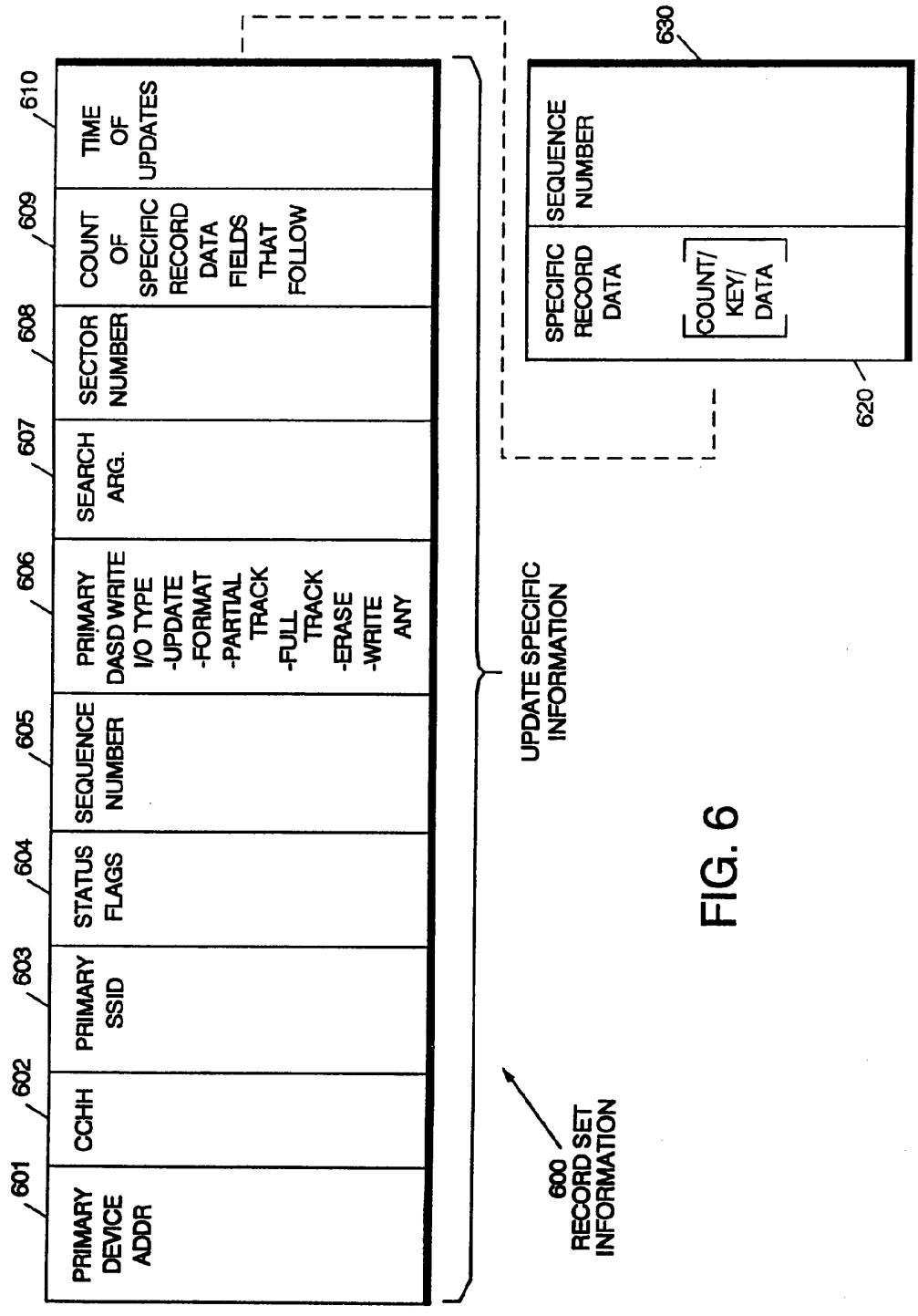
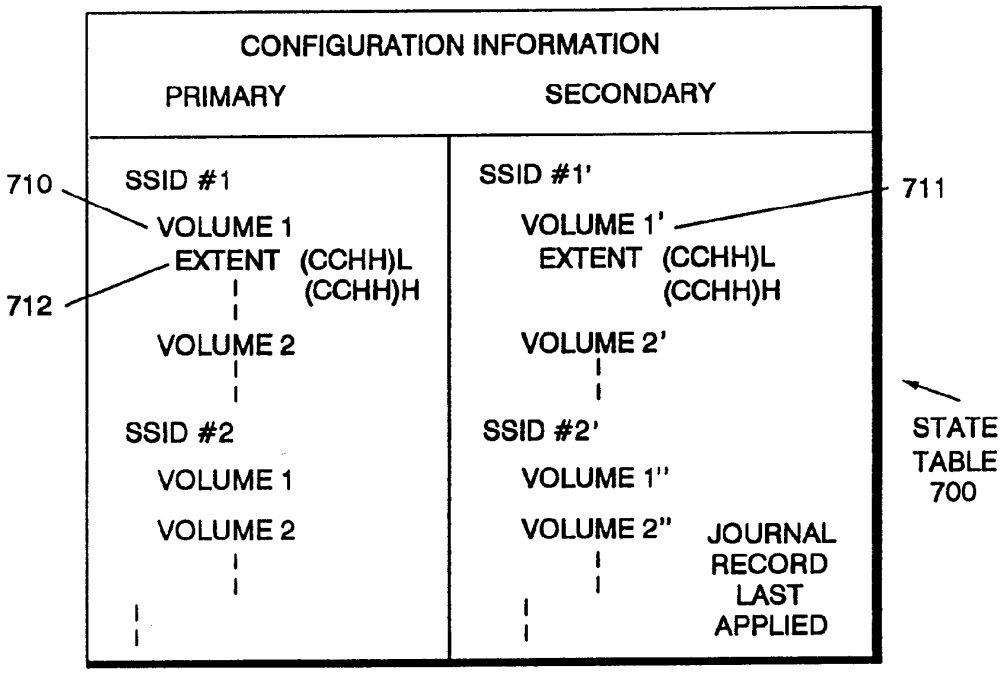


FIG. 6

LINKPLAY EXHIBIT 1004
Page 1604 of 4822

FIG. 7



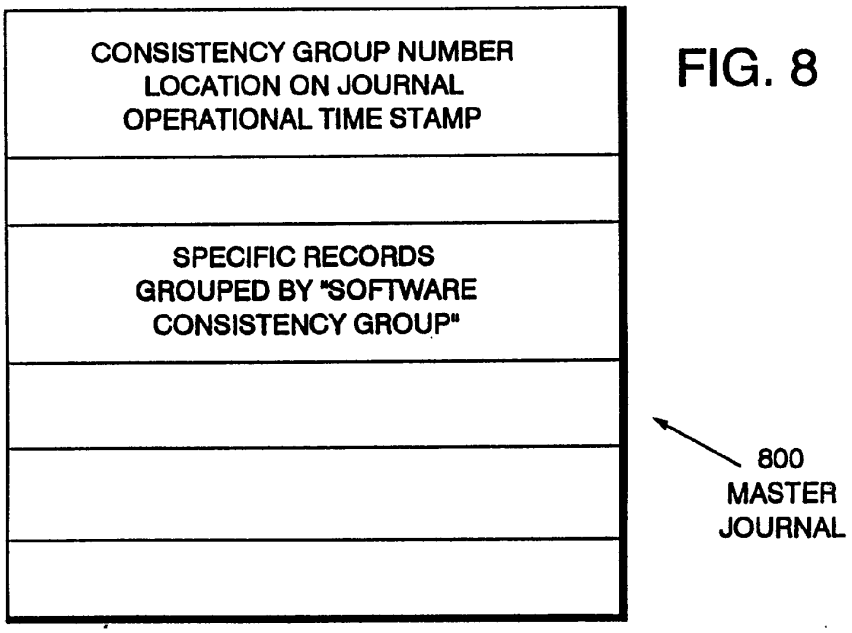


FIG. 8

LINKPLAY EXHIBIT 1004
Page 1605 of 4822

FIG. 9

PHYSICAL CONTROL-LER ID	OPER-ATIONAL TIME STAMP	TIME INT GROUP #	READ RECORD SET TIME OF UPDATE / CONTROLLER		
			SEQ. 1 OF 3	SEQ. 2 OF 3	SEQ. 3 OF 3
SSID1	T_1	G_1	11:59 (2)	12:00 (5)	12:01 (6)
SSID2	T_1	G_1	12:00 (4)	12:02 (7)*	———
SSID3	T_1	G_1	11:58 (1)	11:59 (3)	12:02 (8)*
SSID1	T_2	G_2			
SSID2	T_2	G_2			
SSID3	T_2	G_2			
	T_3	G_3			

CONSISTENCY GROUP #1

(1) 11:58
(2) 11:59
(3) 11:59
(4) 12:00
(5) 12:00
(6) 12:01

EARLIEST OPERATIONAL TIME T_1, EARLIEST TIME OF UPDATE ACROSS SSID, ORDERED IN READ SEQUENCE

MIN TIME OF THE MAX TIMES OF UPDATE ACROSS ALL SSIDS

CONSISTENCY GROUP #2

(7) - - - - -
(8) - - - - -

LINKPLAY EXHIBIT 1004
Page 1606 of 4822

FIG. 10

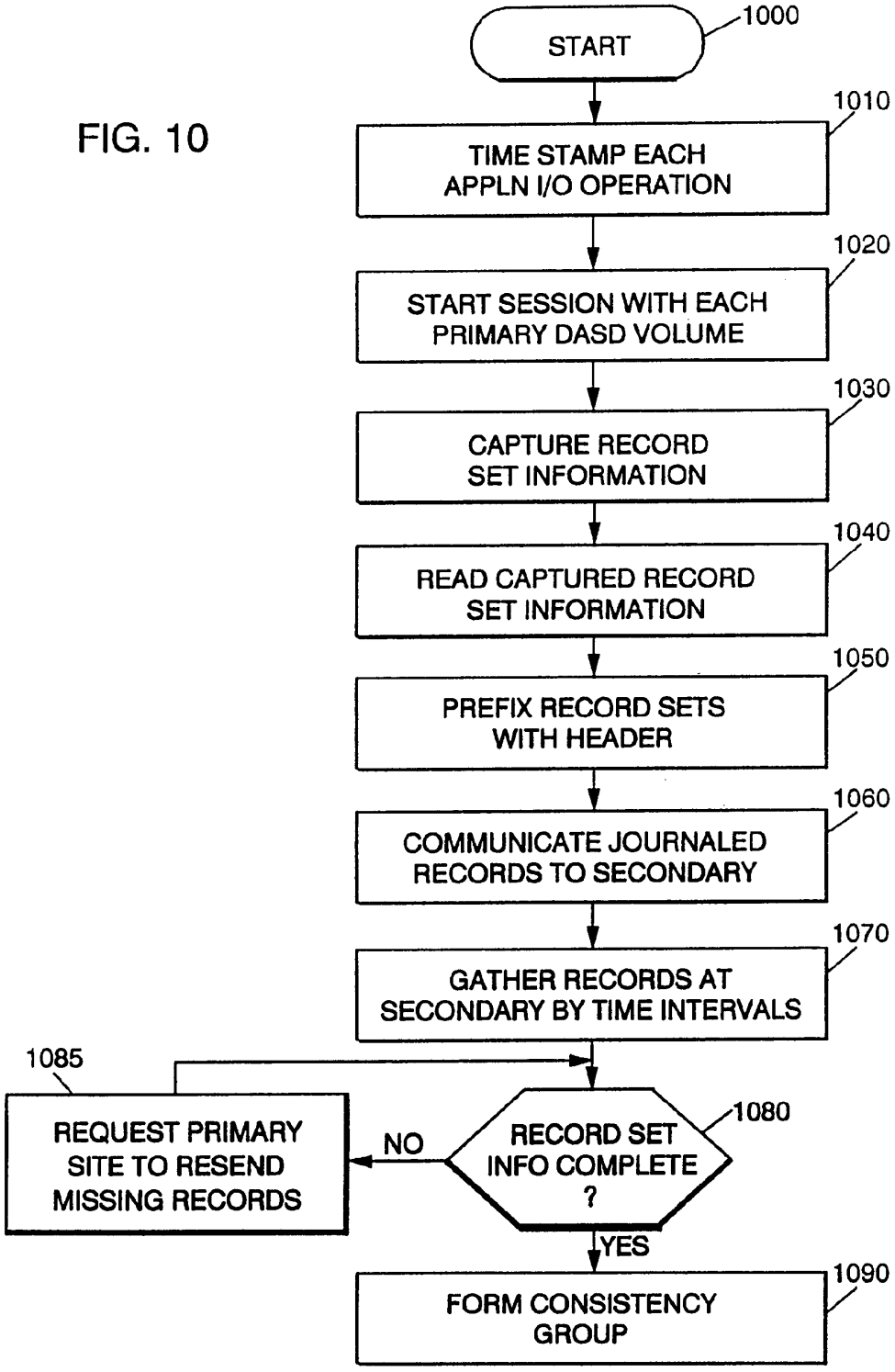
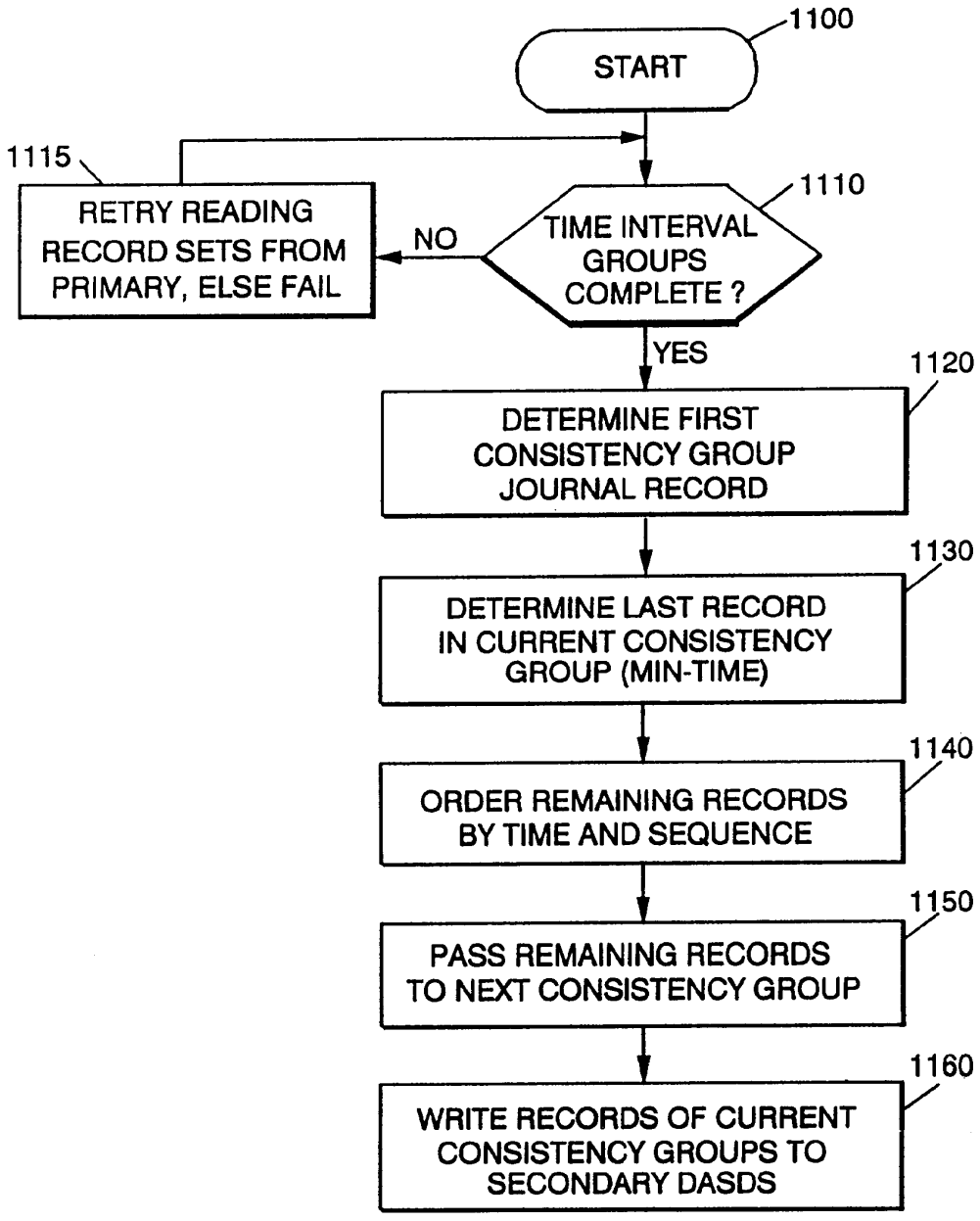

START
1000
TIME STAMP EACH APPLN I/O OPERATION
1010
START SESSION WITH EACH PRIMARY DASD VOLUME
1020
CAPTURE RECORD SET INFORMATION
1030
READ CAPTURED RECORD SET INFORMATION
1040
PREFIX RECORD SETS WITH HEADER
1050
COMMUNICATE JOURNALED RECORDS TO SECONDARY
1060
GATHER RECORDS AT SECONDARY BY TIME INTERVALS
1070
1085
REQUEST PRIMARY SITE TO RESEND MISSING RECORDS
NO
RECORD SET INFO COMPLETE ?
1080
YES
FORM CONSISTENCY GROUP
1090

FIG. 11



LINKPLAY EXHIBIT 1004
Page 1608 of 4822

FIG. 12

FULL CONSISTENCY GROUP RECOVERY RULES

READ RECORD SET BUFFER #2	READ RECORD SET BUFFER #1							
TYPE I/O WRITE OPERATION								
	UPDATE WRITE KL = 0	UPDATE WRITE KL ≠ 0	FORMAT WRITE FULL	FORMAT WRITE PARTIAL	ERASE FULL	ERASE PARTIAL	WRITE ANY KL = 0	WRITE ANY KL ≠ 0
UPDATE WRITE KL = 0	W*	E*	N	J	D	K	W	E*
UPDATE WRITE KL ≠ 0	E*	W*	N	J	D	K	E*	W
FORMAT WRITE FULL	T	T	R	T	R	T	R	R
FORMAT WRITE PARTIAL	C	C	N	H	F	L	W	W
ERASE FULL	T	T	R	R	R	T	T	T
ERASE PARTIAL	B	B	N	M	E	G	W	W
WRITE ANY KL = 0	W	E*	W	W	E	W	W	E*
WRITE ANY KL ≠ 0	E*	W	W	W	E	W	E*	W

EP 0 672 985 A1

LINKPLAY EXHIBIT 1004
Page 1609 of 4822

FIG. 13

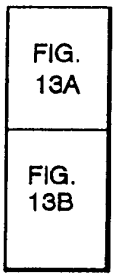


FIG. 13A

B - IF #1'S SEARCH ARG IS HIGHER THAN THE SEARCH ARG FOR #2, THEN THROW #1, ELSE DO BOTH.

C - IF #1'S RECORD IS EQUAL TO OR HIGHER THAN THE FIRST RECORD IN #2, THEN THROW #1, ELSE DO BOTH.

D - IF #2 IS UPDATING R0, THEN DO BOTH, ELSE ERROR.

E - ERROR (SHOULD NEVER HAPPEN).

E*- ERROR IF #1 AND #2 ARE THE SAME RECORD. (SHOULD NEVER HAPPEN WITHOUT A FORMAT WRITE IN BETWEEN).

F - IF FIRST RECORD IN #2 IS R1, THEN WRITE BOTH, ELSE ERROR.

G - IF #1'S SEARCH ARG IS EQUAL TO OR HIGHER THAN THE SEARCH ARG FOR #2, THEN THROW #1, ELSE ERROR.

H - IF #1'S SEARCH ARG IS HIGHER THAN THE LAST RECORD FOR #2. THEN THROW #1. ELSE IF #2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD IN #1, THEN ERROR, ELSE WRITE BOTH.

TO OPTIMIZE FURTHER, CAN DO THE FOLLOWING INSTEAD:

```
IF (1'S SEARCH IS EQUAL TO OR HIGHER THAN THE LAST RECORD
      IN #2) OR (THE LAST RECORD IN #1 IS EQUAL TO OR
      HIGHER THAN THE LAST RECORD IN #2) AND (2'S SEARCH
      IS LESS THAN OR EQUAL TO THE LAST RECORD IN #1)
   THEN THROW 1
   ELSE IF (#2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD
         IN #1)
          THEN ERROR
          ELSE WRITE BOTH
```



J - IF #2'S RECORD (OR SEARCH) IS HIGHER THAN THE LAST RECORD IN #1, THEN ERROR, ELSE WRITE BOTH.

K -IF #2'S RECORD (OR SEARCH) IS HIGHER THAN #1'S SEARCH, THEN ERROR, ELSE WRITE BOTH.

L -IF #1'S SEARCH ARG IS EQUAL TO OR HIGHER THAN #2'S SEARCH ARG THEN EITHER WRITE BOTH OR OK TO THROW #1, ELSE ERROR.

M -IF (#1'S SEARCH ARG IS EQUAL TO OR HIGHER THAN #2'S SEARCH ARG) THEN THROW 1
ELSE IF (#2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD IN #1)
THEN ERROR
ELSE WRITE BOTH.

N -IF #2'S SEARCH ARG IS HIGHER THAN THE LAST RECORD IN #1, THEN ERROR, ELSE WRITE BOTH.

R -OK TO THROW #1.

T -MUST THROW 1.

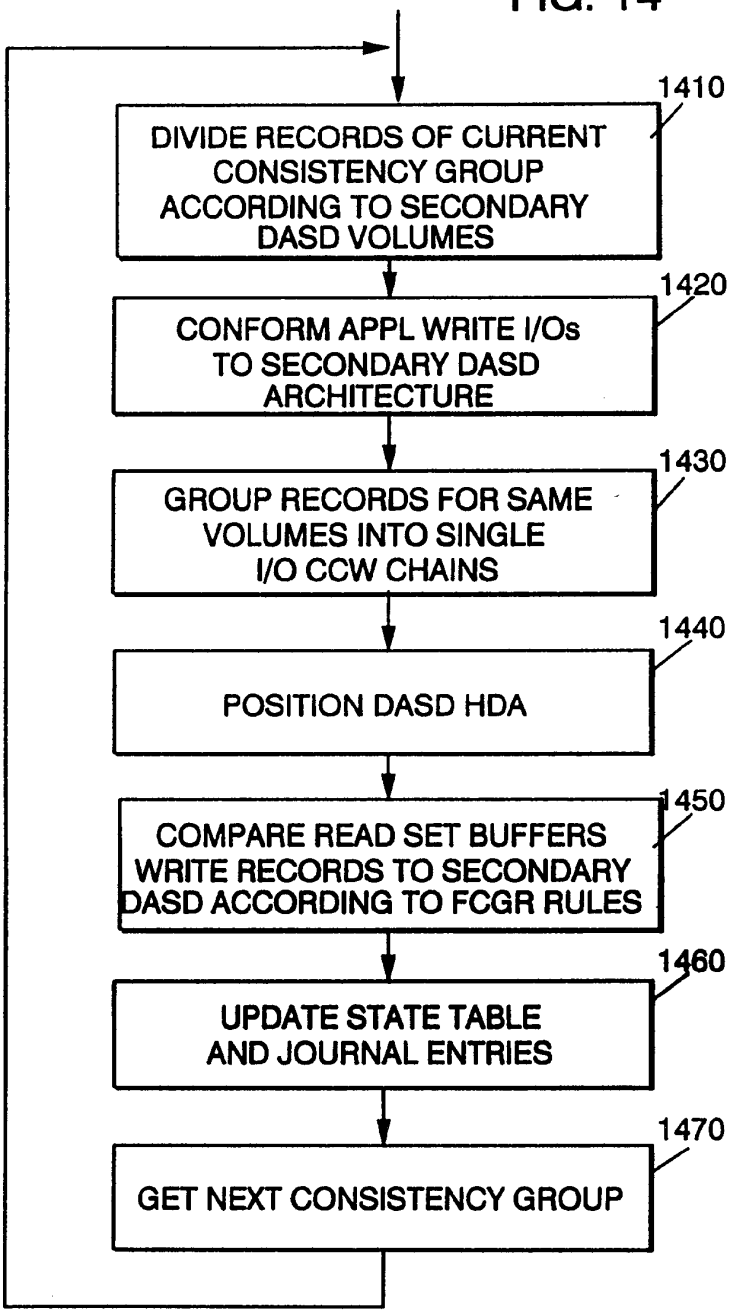
W -WRITE BOTH.

W*-IF #1 AND #2 HAVE THE SAME RECORDS, THEN THROW #1, ELSE DO BOTH OR MERGE RECORDS AND DO ONE WRITE.

FIG. 13B

LINKPLAY EXHIBIT 1004
Page 1611 of 4822

FIG. 14


1410
DIVIDE RECORDS OF CURRENT CONSISTENCY GROUP ACCORDING TO SECONDARY DASD VOLUMES
1420
CONFORM APPL WRITE I/Os TO SECONDARY DASD ARCHITECTURE
1430
GROUP RECORDS FOR SAME VOLUMES INTO SINGLE I/O CCW CHAINS
1440
POSITION DASD HDA
1450
COMPARE READ SET BUFFERS WRITE RECORDS TO SECONDARY DASD ACCORDING TO FCGR RULES
1460
UPDATE STATE TABLE AND JOURNAL ENTRIES
1470
GET NEXT CONSISTENCY GROUP

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 0753

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO-A-91 20034 (STORAGE TECHNOLOGY CORP) 26 December 1991 * abstract * * page 13, line 6 - page 14, line 20 * * page 18, line 34 - page 19, line 6 * * page 20, line 1 - line 35 * * figure 3 * ---	1,10,20, 21,29	G06F11/14 G06F11/20
A	PROCEEDINGS OF THE ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMEN OF DATA, SAN DIEGO, JUNE 2 - 5, 1992, vol. 21, STONEBRAKER M, pages 236-245, XP 000393592 ANUPAM BHIDE ET AL 'AN EFFICIENT SCHEME FOR PROVIDING HIGH AVAILABILITY' * abstract * * page 237, left column, line 36 - right column, line 52 * -----	1,11,20, 29	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
BERLIN	27 June 1995	Masche, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

LINKPLAY EXHIBIT 1004
Page 1613 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 742 674 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
09.04.2014 Bulletin 2014/15

(51) Int Cl.:
H04N 19/60 *(2014.01)*

(21) Application number: **96400966.6**

(22) Date of filing: **06.05.1996**

(54) Video encoding method and system using a rate-quantizer model

Methode und System zur Videokodierung mit einem Ratenquantifizierermodell

Méthode et système de codage vidéo employant un modèle débit-quantificateur

(84) Designated Contracting States:
DE FR GB NL

(30) Priority: **08.05.1995 US 438014**
06.06.1995 US 473783
31.08.1995 US 522495

(43) Date of publication of application:
13.11.1996 Bulletin 1996/46

(73) Proprietor: **Kabushiki Kaisha Toshiba**
Kawasaki-shi,
Kanagawa-ken 210-8572 (JP)

(72) Inventors:
- **Azadegan, Faramarz**
  Waltham,
  Massachusetts 02154 (US)
- **Unno, Hiroaki**
  Ichikawa-shi,
  Chiba 272 (JP)
- **Mimura, Hideki**
  Yokohama 236 (JP)
- **Kitamura, Tetsuya**
  Komea-shi,
  Tokyo 201 (JP)

(74) Representative: **Ahner, Philippe et al**
BREVALEX
95 rue d'Amsterdam
75378 Paris Cedex 8 (FR)

(56) References cited:

EP-A- 0 336 040	**EP-A- 0 475 251**
EP-A- 0 493 130	**EP-A- 0 574 724**
EP-A- 0 597 647	**EP-A- 0 606 629**
EP-A- 0 629 090	**EP-A- 0 645 930**
EP-A- 0 677 842	**EP-A- 0 677 969**
EP-A- 0 703 711	**WO-A-95/12179**
WO-A-96/35998	**WO-A-96/35999**
WO-A-96/36170	**WO-A-96/36173**
WO-A-97/03443	**US-A- 4 942 476**
US-A- 5 146 324	**US-A- 5 231 484**
US-A- 5 245 427	**US-A- 5 283 656**
US-A- 5 381 275	**US-A- 5 410 352**

- **PATENT ABSTRACTS OF JAPAN vol. 015, no. 135 (E-1052), 4 April 1991 & JP 03 016489 A (HITACHI LTD;OTHERS: 01), 24 January 1991,**
- **RIPLEY G D: "DVI - A DIGITAL MULTIMEDIA TECHNOLOGY" COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY, vol. 32, no. 7, July 1989, pages 811-822, XP000045770**
- **LIPPMAN A ET AL: "CODING IMAGE SEQUENCES FOR INTERACTIVE RETRIEVAL" COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY, vol. 32, no. 7, July 1989, pages 852-860, XP000045774**
- **TURNER R R: "1,001 QUESTIONS TO ASK BEFORE DECIDING ON A NONLINEAR VIDEO EDITING SYSTEM" SMPTE JOURNAL, vol. 103, no. 3, 1 March 1994, pages 160-173, XP000445583**
- **H.G.J. THEUNIS, G.J. KEESMAN, J.M.G. KÜPPERS: "Variable Bit-Rate Coding of Image Series for Optical Recording in a MPEG-2 Context" PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON HDTV 94, 26 - 28 October 1994, TURIN, ITALY, pages 191-199, XP002071956**
- **YOKOUCHI ET AL.: "Development of Variable Bit Rate Disc System" OPTICAL DATA STORAGE 95, 5 - 7 July 1995, SAN DIEGO, CALIFORNIA, pages 46-54, XP002071957**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

(Cont. next page)

EP 0 742 674 B1

- "MPEG DIGITAL VIDEO-CODING STANDARDS - DELIVERING PICTURE-PERFECT COMPRESSION FOR STORAGE, TRANSMISSION, AND MULTIMEDIA APPLICATIONS", IEEE SIGNAL PROCESSING MAGAZINE, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 14, no. 5, 1 September 1997 (1997-09-01), pages 82-100, XP011089789, ISSN: 1053-5888, DOI: 10.1109/79.618010

LINKPLAY EXHIBIT 1004
Page 1615 of 4822

Description

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application is related to U.S. Patent Application S/N 08/438,014, entitled "METHOD AND SYSTEM TO REPLACE SECTIONS OF AN ENCODED VIDEO BITSTREAM", filed May 8, 1995 (WO 97 03 443 A). This application is also related to and being concurrently filed with four other patent applications: These applications are U.S. Patent Application S/N 08/XXX,XXX, entitled "AUDIOVISUAL ENCODING SYSTEM WITH A REDUCED NUMBER OF AUDIO ENCODERS" (WO 96 36 173 A), U.S. Patent Application S/N 08/XXX,XXX, entitled "METHOD AND SYSTEM FOR A USER TO MANUALLY ALTER THE QUALITY OF PREVIOUSLY ENCODED VIDEO FRAMES" (WO 96 35 999 A), U.S. Patent Application S/N 08/XXX,XXX, entitled "METHOD AND SYSTEM FOR A USER TO MANUALLY ALTER THE QUALITY OF A PREVIOUSLY ENCODED VIDEO SEQUENCE" (WO 96 36 170 A), and U.S. Patent Application S/N 08/XXX,XXX, entitled "A RATE-CONTROLLED DIGITAL VIDEO EDITING METHOD AND SYSTEM WHICH CONTROLS BIT ALLOCATION OF A VIDEO ENCODER BY VARYING QUANTIZATION LEVELS" WO 96 35 998 A, each filed on June 6, 1995.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention relates to video encoding systems, such as MPEG and MPEG-2 video encoders, and more particularly, to video pre-encoders which analyze the bit-rate of the pre-encoded digital video on a frame by frame basis to determine an average bit-rate for each quantization level used in the pre-encoding process. The present invention is used to encode digital video to fit within a fixed capacity storage medium based on bit-rates of future frames as determined in the pre-encoding process.

Discussion of the Background

[0003] Digital video encoding systems encode raw digital video signals representing the intensity of portions of a frame into compressed, encoded digital video for use in one of two different playback modes; these modes are a constant bit-rate (CBR) mode and a variable bit-rate (VBR) mode. The constant bit-rate mode is suitable for use in transmission technologies such as cable television and direct satellite broadcast which will often use a fixed bandwidth to deliver a signal from a transmitter to a remote receiver. The variable bit-rate mode is suitable for playing back digital video from digital storage media such as compact discs.

[0004] In CBR mode, the transmitter must send, typically, 30 frames per second to the receiver while avoiding video buffer underflow and overflow. Underflow occurs in constant bit-rate mode when at the end of a 1/30th of a second time interval, there is insufficient data in the video decoding buffer to construct a picture. Overflow in CBR mode occurs when there is too much video data placed in the buffer for the decoder to decode within the 1/30th of a second interval.

[0005] To control the bit-rate in the CBR mode, video encoders use a technique called adaptive quantization to vary the instantaneous bit-rate to meet the target bit-rate. Adaptive quantization involves estimating a number of bits to be generated by a series of macroblocks based on the quantization levels of the macroblocks, encoding the series of macroblocks, determining the number of bits of the encoded macroblocks, comparing the number of bits resulting from the encoding with the estimated number of bits and changing the quantization levels of a succeeding set of macroblocks based on the accuracy of the previous estimate.

[0006] In order to accurately estimate the number of bits that will be generated by a series of macroblocks, a quantizer level versus bit-rate model indicates what the average number of bits to be generated by a particular quantization level will be. In a paper entitled "Joint Multi-Channel Bit-Rate Control for Multiplexed MPEG Video Using Rate-Distortion Curves", by Reininger et al. and presented at Princeton Conference 1994, a constant bit-rate encoder is disclosed which splits the encoding process into two phases. In the first phase, a first frame is pre-encoded with parts of frames being pre-encoded using different quantization levels. This gives a quantization level versus bit-rate curve that describes how the first frame responds to the first phase. The first frame then enters a second phase where the quantization level versus bit-rate curve generated in the first phase is used to regulate the encoding of the first frame into part of the final encoded digital video stream. By using the curve generated in the first phase, the second phase may more accurately decide on the quantization levels to be assigned to each macroblock such that the frame is encoded to avoid underflow and overflow. In parallel with the second phase encoding of the first frame, the Reininger method begins the first phase on the second frame and gathers statistics. The process is repeated for all the frames in a movie. This technique, however, provides only a single frame look ahead which is limited in its ability to control rate.

[0007] In the VBR mode, overflow can be avoided by only reading data from the digital storage medium when the

LINKPLAY EXHIBIT 1004
Page 1616 of 4822

buffer can accept new data. In the VBR mode, underflow is still possible and results from too much rather than too little, data being used to represent video frames. When too much data represents a frame, the video decoder cannot perform all the necessary decoding operations in 1/30th of a second. An error results because a frame needs to be displayed before it has been totally decoded.

[0008] In order to provide encoded digital video representing a full-length movie on a single compact disc, various parts of the raw video may be encoded using different quantization levels to produce various image qualities. To determine the amount of space that a particular section of raw video will consume when encoded, the raw video can be pre-encoded at a particular quantization level and the resulting bit lengths of each frame recorded to aid in a later encoding process. To determine the resultant bit-rates for different quantization levels, this process can be repeated for the whole raw video multiple times, and the results recorded after encoding each frame. However, even though this technique provides the greatest accuracy with which the quantization model can be generated, it is very time consuming. As a typical movie will last between an hour and a half and two hours, even pre-encoding a movie twice using a different quantization level each time would take at least three hours.

[0009] EP 0 597 647 A discloses a method for processing video signals included in video frames to determine characteristics of the video frames, comprising the steps of:

a) obtaining video signals included in plural video frames;

b') assigning a quantization level to one of said plural video frames;

c') pre-encoding said one of said plural video frames using the quantization level to obtain a bit-rate;

d') determining the bit-rate for the quantization level resulting from said pre-encoding step;

e) repeating steps b', c' and d' sequentially for plural succeeding video frames; and

f) encoding a plurality of said plural video frames after determining the bit-rates for each of said plural video frames;

g) determining a target number of bits for said plurality of said plural video frames;

h) assigning a first set of quantization levels for each frame in said plurality of said plural video frames using said target number of bits and the plural determined bit-rates; and

i) encoding one video frame of said plurality of said plural video frames into an encoded frame with a resultant number of bits.

SUMMARY OF THE INVENTION

[0010] Accordingly, one object of the present invention is to overcome the problems of video pre-encoding as outlined above.

[0011] A further object of this invention is to pre-encode raw video with at least two quantization levels such that a bit-rate versus quantization level model can be generated for plural frames of the raw video.

[0012] It is a further object of this invention that two quantization levels be uniformly assigned to the plural frames during the pre-encoding process so as to generate two quantization level versus bit-rate points in a line.

[0013] It is a further object of this invention that the two quantization levels be assigned to different regions of the plural frames in a checkerboard style fashion.

[0014] It is a further object of this invention that four quantization levels be assigned to different regions of the plural frames during the pre-encoding process.

[0015] It is a further object of this invention that four quantization levels be assigned uniformly to the plural frames such that four points on a quantization level versus bit-rate curve can be generated.

[0016] These and other objects are accomplished by a method and system according to the independent claims. The method and system generate and record bit-rates for each quantization level for the plural frames during a pre-encoding process, such that these quantization level versus bit-rate statistics can be used in a later encoding of the raw video. Bit-rates for quantization levels not used in the pre-encoding process can be extrapolated using known quantization levels from the pre-encoding phase. Furthermore, because statistics have been gathered in advance of the encoding process, the encoder can be directed to use greater or fewer bits in a particular section of raw video based on future needs as indicated by quantization level versus bit-rate curves for later frames.

LINKPLAY EXHIBIT 1004
Page 1617 of 4822

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] A more complete appreciation of the invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

Figure 1A illustrates a block diagram of the encoder system architecture;
Figure 1B illustrates devices for obtaining the audio and video to be encoded, and devices for writing and transmitting the encoded audiovisual data generated by the encoding system;
Figure 2 is a flowchart illustrating the general operation of the system of Figure 1;
Figure 3 is a flowchart illustrating a process performed in the set-up state;
Figure 4 illustrates a process performed in the compression state;
Figure 5 illustrates the process performed in the edit state;
Figure 6 illustrates a process performed for the start, completion, and exit states;
Figure 7 is a flowchart showing how to determine the number of audio encoders needed to encode a predetermined number of audio tracks while using a multipass video encoder, and the process of simultaneously performing audio and multipass video encoding;
Figures 8A - 8D illustrate the Sequence Layer log file format;
Figures 9A - 9C illustrate the Picture Layer log file format;
Figures 10A and 10B illustrate the full and short forms of the Macroblock Layer log file format;
Figure 11 illustrates a process for utilizing the information within the log files;
Figure 12 illustrates the general process performed during the manual adjustment of picture quality for different time periods;
Figure 13 illustrates a user interface which is used to input the parameters for changing the picture quality over various time periods;
Figure 14 illustrates the general procedure for calculating the number of bits associated with the desired picture quality;
Figure 15 illustrates the process used for processing sections having maximum or minimum user selected priorities;
Figures 16A and 16B illustrate a flowchart for calculating the number of bits corresponding to desired picture qualities which are not maximum or minimum priorities;
Figure 17 illustrates a flowchart of the process used to check if underflow results from the changing of the picture quality;
Figure 18 illustrates a flowchart for determining if buffer underflow will occur in frames after the edit segment due to changes made within the edit segment;
Figures 19A and 19B illustrate two different examples of how the bit rates of the encoded video are changed by the user selected quality;
Figure 20 illustrates the general process performed during the manual adjustment of picture quality for different regions within a frame;
Figure 21 illustrates an example of different regions having user selected priorities in a frame of video;
Figure 22 illustrates a graph of the user selected priorities versus the fraction of the original quantizer values used for the selected priority;
Figure 23 illustrates a process for determining if the difference between the number of bits resulting from the user selected priorities and the number of bits resulting from the original encoding of a frame is acceptable or whether a correction procedure must be performed;
Figures 24A-24C illustrate a process for correcting a frame when it is determined to have too many bits and therefore, the quantizer levels of the macroblocks are incremented in order to reduce the number of bits;
Figures 25A-25C illustrate a process for correcting for too few bits in a frame by reducing the quantizer levels of macroblocks within a frame in order to increase the number of bits;
Figure 26 is a flowchart illustrating the general process of loading from memory the expected number of bits to be used in each of the sets of n macroblocks, and then for each set of n macroblocks re-encoding, counting the resulting number of bits, calculating the accuracy of the estimation based on the expected number and generated number of bits and adjusting the pre-assigned quantization levels of the next n macroblocks to remain close to the target size of the series of frames;
Figure 27A is a picture of frame illustrating the pre-assigned quantization levels;
Figure 27B is a picture of the frame of Figure 27A when each set of macroblocks contains one slice and after the first set of macroblocks has been re-encoded, a first correction factor has been calculated and the first correction factor has been added to the second set of macroblocks;
Figure 27C is a picture of the frame of Figure 27B after the second set of macroblocks has been re-encoded, a

LINKPLAY EXHIBIT 1004
Page 1618 of 4822

second correction factor has been calculated and the second correction factor has been added to the third set of macroblocks;
Figure 27D is a picture of the frame of Figure 27A when each set of macroblocks contains two slices and after the first set of macroblocks has been re-encoded, a first correction factor has been calculated and the first correction factor has been added to the second set of macroblocks;
Figure 28A is a frame containing two different regions whose quantization levels are changed;
Figure 28B is a picture of two possible encodings of the frame of Figure 28A illustrating the change in the number of bits in a video stream used to represent the two different regions whose quantization levels were changed;
Figure 29 is a video stream illustrating the change in the number of bits used to represent a series of images where the total length of the re-encoded series equals the series original length;
Figure 30 is a flowchart illustrating how a correction factor index is determined from a ratio lookup table using an estimation ratio;
Figure 31 is a flowchart illustrating how a correction factor is chosen from a correction factor table using a correction factor index;
Figure 32 is a flowchart illustrating how a correction factor is calculated using both an estimation ratio and the number of remaining unre-encoded sets of macroblocks;
Figure 33A illustrates a section of original video in an encoded format;
Figure 33B illustrates a section of new video in an unencoded format which is to be substituted into the original encoded video;
Figure 33C illustrates the unencoded new section of video with unencoded sections of the original video at the ends thereof;
Figure 33D illustrates the video in Figure 33C in an encoded format which is to be substituted into the original encoded video;
Figure 33E illustrates the original encoded video having the substituted encoded video included therein;
Figure 34 illustrates a process for substituting a section of video into the encoded video data stream which results in video with reduced decoding artifacts;
Figure 35A illustrates an unencoded new section of video with an unencoded section of the original video at the trailing end;
Figure 35B illustrates a section of encoded video to be substituted into the originally encoded video illustrated in Figure 33A;
Figure 35C illustrates the originally encoded video of Figure 33A containing the substitute encoded video of Figure 35B;
Figure 36 illustrates a process for reducing decoding artifacts when editing an encoded video bitstream by encoding the substitute video using a closed group of pictures mode;
Figure 37A is a frame of raw video that has two quantization levels assigned to it to generate quantization level versus bit-rate models in the pre-encoding phase;
Figure 37B is a frame with four quantization levels assigned in a rotating pattern;
Figure 37C is a frame with four quantization levels distributed in block formation;
Figure 38 is a flowchart which illustrates the process of obtaining video signals and pre-encoding them to determine a bit-rate for each quantization level specified in the pre-encoding phase;
Figure 39 is two recorded frames and their associated quantization levels versus bit-rates;
Figure 40 illustrates an outline of the final format of the encoded data which is ultimately included on an optical disc;
Figure 41 illustrates the disc information file 908 of Figure 40;
Figure 42 illustrates the data file management information contents of the data files illustrated in Figure 40;
Figure 43 illustrates the details of the data stored in the data files of Figure 40;
Figure 44A illustrates a data pack in which the data thereof does not require a stuffing packet and Figure 44B illustrates a data pack which requires a packet for padding;
Figure 45 illustrates a playback information pack;
Figure 46 illustrates the information contained within the data search information 996 of Figure 45;
Figure 47 illustrates the general information of Figure 46;
Figure 48 illustrates the synchronous reproduction information illustrated in Figure 46;
Figure 49 illustrates a video pack;
Figure 50 illustrates the relation between a Group of Pictures (GOP) and a series of video packs;
Figure 51A illustrates an audio pack encoded according to the MPEG audio encoding specification;
Figure 51B illustrates an audio pack constructed according to AC-3 or linear PCM audio encoding;
Figure 52 illustrates the relationship between the encoded audio stream and the audio packs;
Figure 53 illustrates the structure of a sub-picture unit;
Figure 54 illustrates the structure of a sub-picture pack;

LINKPLAY EXHIBIT 1004
Page 1619 of 4822

Figure 55 illustrates the structure of a sub-picture unit in relation to sub-picture packs; and
Figure 56 illustrates a transition between displaying sub-picture unit n and sub-picture unit n+1.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0018] The present invention is a digital video encoding system which includes a plurality of components and functional aspects. In order to more easily identify the various components of the system, this writing is organized into the following sections:

I. INTRODUCTION
II. SYSTEM COMPONENTS
III. GENERAL SYSTEM OPERATION
IV. SYSTEM ALTERNATIVES
V. REDUCING THE NUMBER OF AUDIO ENCODERS WHILE MAINTAINING OPTIMUM ENCODING EFFICIENCY
VI. LOG FILES FOR STORING INFORMATION OF THE ENCODED VIDEO, AND LOG INTERPRETER UTILITIES TO ACCESS THE LOG FILES AND THE ENCODED VIDEO
VII. CHANGING THE QUALITY OF THE ENCODED VIDEO

A. Changing the Quality Over Periods of Time
B. Changing the Quality Within Regions of Frames

VIII. RE-ENCODING THE VIDEO AFTER CHANGING THE QUALITY

A. Bit Allocation Control When the Quality Has Been Changed Within Single Frames
B. Re-Encoding While Avoiding Decoding Artifacts At Edit Points

1. Restoring the Encoder to the State Used When Originally Encoding
2. Re-Encoding in a Closed Group of Pictures Mode

IX. RATE-QUANTIZER MODELING USED FOR ENCODING
X. COMBINING AUDIO, VIDEO, AND SUB-PICTURE DIGITAL SIGNALS USING DATA STRUCTURES

I. INTRODUCTION

[0019] The disclosed system relates to a digital audiovisual encoding system, also known as a digital compression system. The ideas disclosed herein are usable with any type of encoding scheme and are particularly applicable to MPEG-2 (Moving Picture Experts Group) formally known as ISO/IEC 13818-2, ISO/IEC 13818-1 which specifies MPEG-2 system information, and the MPEG-2 predecessor, MPEG-1 (ISO/IEC 11172-video). MPEG-1 and MPEG-2 will collectively be referred to herein as "MPEG". The MPEG standard specifies only the syntax of the compressed or encoded video stream but not how to perform compression. It is to be noted that as an MPEG encoded data stream is a compressed data stream, the encoding process may be referred to as a compression process and vice versa. The system also relates to the encoding of audio which may be performed using Dolby AC-3, or the MPEG-2 standard as defined in ISO/IEC 13818-3.

[0020] MPEG video compression is a temporal causal predictive compression. As the MPEG encoder receives a new video picture, it compares the new video picture to a predicted picture that the encoder has generated based on previously encoded pictures. The term "causal" predictive coding is used because the prediction is made based on previously transmitted information, and the term "temporal" causal predictive coding is used because the prediction is performed in terms of temporal picture units. The prediction procedure involves estimating the motion of areas within a video picture. Therefore, the temporal prediction is sometimes called motion-compensated prediction.

[0021] The result of the comparison between old and new pictures represents new information contained in the later video picture. The new information, also called "residual information," is then subjected to a series of information compaction processes. First, a linear mathematical transformation known as the discrete cosine transformation (DCT) is performed. The DCT operation transforms the pixel residual information into a series of coefficients. In MPEG compression, the DCT is performed over blocks of 8x8 pixels. Similarly, the DCT coefficients are also in 8x8 arrays of numbers. These coefficients are then individually quantized (or rounded) to a set degrees of precision, based upon a quantization step size or q-level. The quantization process typically produces many coefficients having the value of zero. It is this quantizing process which is mostly responsible for introducing coding artifacts, otherwise referred to as quantization noise.

[0022] The quantized coefficients are then coded using a combination of run-length coding on the zero-valued coef-

LINKPLAY EXHIBIT 1004
Page 1620 of 4822

ficients and variable-length coding of the resultant pairs of two numbers, each representing the run-length value and non-zero coefficient value, respectively. The codes for these run-length non-zero value pairs, together with other overhead information, form the compressed digital data stream corresponding to the video picture. Due to physical limitations such as output data rate or storage capacity, it is necessary to vary the degree of quantization so that an appropriate amount of compression is accomplished. This is referred to as buffer feedback.

[0023] An MPEG-compliant decoder reads the encoded digital data stream and performs operations opposite to the encoding process.

[0024] Typically, the total number of bits in the digital data stream is substantially less than the total number of bits needed to represent the video pictures, and hence compression is achieved. It is important, however, to recognize that such compression as exemplified by MPEG video coding is by no means free of distortion or artifacts. As mentioned, the principal source of compression artifacts is the quantization process.

[0025] The disclosed system is aimed at dynamically adjusting the degree of quantization during the compression or encoding process. In particular, adjustment is made in accordance with instructions provided by a human operator. The adjustments may apply to parts of a larger video sequence, or possibly to the entire video sequence.

II. SYSTEM COMPONENTS

[0026] Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout the several views and more particularly to Figure 1A thereof, there is illustrated a block diagram of the encoder system architecture. It is to be noted that the video encoding process is a complex process which requires a number of different devices which operate synchronously in order to achieve the encoding and editing without interruption or errors. However, the audio and video encoding can be separately performed as long as the appropriate time codes are used. While four workstations have been illustrated in Figure 1A, ordinarily, the user or operator of the encoding system will interact with only one keyboard and workstation. The workstations contain conventional general purpose computer components such as a processor, RAM, ROM, a network interface, a hard disk, a keyboard, monitor, and other conventional components.

[0027] A workstation 30 functions as the system control station which, in the preferred embodiment, is a Sun SPARC20 Unix workstation. Workstation 30 is equipped with a graphical console display monitor 31 as well as conventional workstation input devices such as a keyboard 32 and a pointing device 33 which is a mouse or any other pointing device. The user visually interacts with the workstation 30 primarily through a graphical user interface (GUI) which is displayed on the console display 31.

[0028] The workstation 30 is connected to and controls the other workstations 40, 10 and 20 which are described below. Workstation 30 also communications with the audio interface unit 72, and the digital video capture and storage device 60 which are also discussed below. The workstation 30 is connected to the workstation 40 through a serial interface as defined by the Electronic Industries Association (EIA) standard RS-232. Similarly, the connection between workstation 30 and the audio interface unit 72 is also by means of the serial RS-232 standard. The connection between the workstation 30 and the digital video capture and storage device 60 is through an industry-standard S-bus interface.

[0029] A frame jogger 41 is connected to the workstation 40 to allow the operator to position the video of either of the VTRs 51 or 52 to where editing is to be performed. The system is capable of using time codes input by an operator to control the position of the video. However the jog controller enhances the user-friendliness of the system by allowing the use of a jog knob to position the video. Workstation 30 controls whether the displayed video on the video display monitor 61 is from the digital video capture and storage device 60 and/or from the record VTR 52.

[0030] Video encoding device 50 is a digital video encoder which performs compression according to the MPEG-1, MPEG-2, or other video compression standard. A commercially available MPEG-1 encoder is the Sony RTE-3000. An MPEG-2 encoder can be constructed according to the teachings set forth in Japanese Patent Application 6-326435, filed December 27, 1994 (JP 8 186 821 A). Workstation 10 directly controls the video encoding device 50 through the industry standard S-bus interface and the workstation 10 receives commands from workstation 30 through standard networking equipment such as an Ethernet network. During the encoding process, workstation 10 supports real-time execution of the video encoding control program of the video encoding device 50. In the preferred embodiment, the workstation 10 is a Sun SPARC20 Unix workstation.

[0031] The workstation 10 responds to commands from the central workstation 30 to begin the compression process. During the compression process, workstation 10 dynamically controls the degree of quantization applied to particular areas of the video signal. The particular areas for which the quantization is varied extend over the spatial and/or temporal domains and the exact manner in which the quantization is controlled is explained below.

[0032] The input video to the video encoding device 50 is from a tape within the playback VTR 51. The playback VTR 51 is a profession digital video standard, ITU-R 601 (formerly known as CCIR 601 video standard) video tape player. This standard applies to both NTSC as well as PAL systems. It is noted that the tape within the playback VTR 51 contains electromagnetic information representing various images and scenes which have been captured through a video or film

LINKPLAY EXHIBIT 1004
Page 1621 of 4822

camera, or generated by a computer. The playback VTR 51 operates under the control of the workstation 40. The encoded output from the video encoding device 50 is transferred to a workstation 20 for storage on one or more hard disks 21.

[0033] In order to view video data which has been encoded and output by the video encoding device 50, a digital video decoder such as an MPEG-2 decoder is necessary. However, at the time of this writing, the implemented system does not incorporate a separate MPEG-2 decoder as such a decoder was not readily available during the early development stages of the system. Therefore, the video encoding device 50 not only outputs an MPEG-2 video stream to workstation 20, but also a decoded video image equivalent of the MPEG-2 encoded data is output from the video encoding device 50 to the record VTR 52. The record VTR 52 is connected to the video encoding device 50 by the industry-standard digital video signal connector known as the D1 interface and controlled by the workstation 40. Both the playback VTR 51 and the record VTR 52 are suitably implemented using Ampex DCT professional video tape recorders.

[0034] The audio is encoded and compressed, for example, into the Dolby AC-3 format, or alternatively an MPEG-2 compliant format such as described in the ISO/IEC 13818-3 standard. The audio sources for the encoding system are four digital audio tape (DAT) players 71a, 71b, 71c, and 71d. Studio quality DAT players are commercially available from Sony. The DAT player 71a-71d are connected and controlled via the industry-standard protocol RS-422 by the workstation 40. The audio signals output by the DAT players 71a-71d are input into audio encoders 70a, 70b, 70c, and 70d respectively. These audio encoders are implemented using the commercially available DOLBY AC-3 model DP 525 encoders and are controlled by the audio interface unit 72. Typically, the audio encoding will begin and end at specified time codes. The audio interface unit 72 receives instructions such as encoding start and ending time codes from the workstation 30 over an RS-232 connection therebetween. Additionally, the audio interface unit 72 multiplexes the various digitally compressed audio data into a single digital stream transmitted to workstation 20. Workstation 20 demultiplexes the input digital audio stream into multiple files corresponding to the compressed digital data output by the audio encoding devices 70. The video encoding device 50 provides a synchronization signal based on its video synchronization signal to the audio interface unit 72. The sync signal allows the audio bitstream to be synchronized to the video bitstream output by the video encoding device 50.

[0035] Workstation 30, has the capability of instructing the audio interface unit 72 to provide a duplicate of the audio data from any one of the various audio encoding devices 70a-70d to the audio decoder 73. The audio decoder 73 decodes the compressed audio so that a user may listen to the audio via a speaker 75. The choice of which audio encoding device to monitor is made by the human operator through interaction with the graphical user interface on the console display 31 via either the keyboard 32 or pointing device 33. The construction of the audio interface unit 72 is hardware which is capable of performing the above-described functions. This hardware is one or more programmed microprocessors and/or conventional circuit elements arranged to achieve the disclosed functions.

[0036] Workstation 40 is the studio equipment control station and issues appropriate commands to control both the playback video tape recorder 51 and the digital audio tape players 71a-71d along with the recording video tape recorder 52. The workstation 40 also instructs the digital video capture and storage device 60 to begin capturing video at the appropriate time. The connections between workstation 40 and the various video and audio tape decks is by means of the industry standard RS-422 protocol. This protocol allows each tape deck to inform the workstation 40 of its current tape location in standard time code format. The workstation 40 uses such information to properly synchronize the various tape decks so that the video and audio information is played out and recorded correctly. In the present embodiment, workstation 40 is a standard IBM compatible personal computer running DOS and Ensemble Pro, a commercially available software program from the Editing Technologies Corp., Moorpark, California. It is noted that a minor modification was made to the Ensemble Pro program so that workstation 30 can communicate with Ensemble Pro running on the workstation 40 through an RS-232 communication port. This modification is either commercially available through Editing Technologies Corp., or may be performed by one of ordinary skill in the art without undue experimentation.

[0037] A digital video capture and storage device 60 serves two functions within the encoding system; first, this device uses its capture and storage capability to provide a convenient "before-and-after" comparison of manual editing that has been applied to the video signal, and second, provides a graphical overlay feature to the video display. This graphic overlay feature is used, for example, during the spatial manual editing phase, to allow the human operator to define the regions of interest in which the quantization and consequently the number of bits allocated is varied.

[0038] In the present invention, the digital video capture and storage device is implemented as a commercially available DRAM-based video product, model VS6000, from Viewgraphics, Mountain View, California. The digital video input to this device is by means of the industry standard D1 having ITU-R-601 (formerly CCIR 601) image resolution. The output of the digital video capture and storage device 60 is connected to the video display monitor 61 through three separate analog signals representing the red, green and blue components of the video signal. The graphics overlay information is processed by the digital video and capture storage device 60 prior to creation of the final analog signal output by the digital video capture and storage device 60 and therefore will be visible to the human operator along with the other video data on the video display monitor 61.

[0039] The workstation 30 running under the Unix environment maps graphical windows thereof onto the video display

LINKPLAY EXHIBIT 1004
Page 1622 of 4822

monitor 61 using the digital video capture and storage device 60. This allows the human operator to perform graphical operations (e.g., drawing rectangles) on the graphical overlay plane of the display shown in monitor 61. The device 60 provides an overlay function. The overlay is performed in video memory prior to the conversion of the information in the memory to an analog video signal which drives the monitor. In the VS6000 used by the system, there is an 8-bit memory plane on top of the video memory. On this 8-bit plane, the computer can draw any color graphics and text. One of the "colors" on this 8 bit plane is the transparent color. Any pixel on this overlay plane that has the transparent color will take on the value of the video memory underneath it. In general, most of the overlay plane will be filled with the transparent color, with graphics (e.g. lines, text, etc) using non-transparent colors. Thus, a display memory is formed which consists of both the video memory and the overlay graphics memory. It is the display memory that is ultimately converted to an analog signal for display on the monitor 61. The human operator typically uses the pointing device 33 associated with the keyboard 32 to perform such graphical operations and manipulations of graphical objects thus created.

[0040] A workstation 20 receives digital data streams from the video encoding device 50 as well as the audio interface unit 72. The workstation 20 is connected to the workstation 30 via an Ethernet connection, and the workstation 20 is also connected to the video encoding device 50 and the audio interface unit 72 through the industry-standard S-bus interface. The received digital data streams are stored in one or more hard disks 21 as separate data files. Separate Direct-Memory-Access (DMA) cards are used in transferring the captured digital stream data to the hard disks 21. For long video sequences that result in large quantities of digital data, the hard disks 21 may be implemented using commercially available hard disks such as the SPARCstorage Array X655A/G5 from Sun Microsystems. Workstation 20, in this embodiment, is a commercially available SPARCserver 1000 from Sun Microsystems.

[0041] Workstation 20 also formats the data files corresponding to the video and audio compressed data into a single formatted file suitable for transport or transmission to decoding devices. The final medium is preferably disc-based, in which case the formatting will be in accordance to the pertinent disc format. Typically, the formatted file is transported by some intermediate digital storage means, such as a digital tape, to a facility where the transfer to the final transmission medium or media takes place. The digital storage devices 22, shown in the figures, serve such a purpose. In the present embodiment, digital storage devices 22 include a commercially available 8mm Exabyte tape drive. The interface of the digital storage device 22 is the industry-standard Small Computer System Interface (SCSI).

[0042] It is known that multiple SCSI devices can be attached to the same SCSI bus. Therefore, the digital storage devices 22 can include other SCSI devices such as the commercially available digital linear tape (DLT) drive, a magneto-optic (MO) disc drive, or a floppy disk drive. These configurations are easily accommodated, and would serve to provide a flexible input mechanism for any auxiliary data that may be needed prior to or during the formatting stage. For example, in movie production, subtitle information is often available in a medium that requires one of the SCSI devices mentioned above. In such cases, the auxiliary data is read by software programs executing in workstation 20 and manipulated so that the auxiliary data is included into the formatted files. The auxiliary data may include subtitle data which is in the U.S. Close-Caption format. The auxiliary data can include any desired data such as program codes or postscript data.

[0043] Figure 1B illustrates an audio/visual encoding system 98, which corresponds to and functions in the same manner as the system of Figure 1A. In Figure 1B, video input to the encoding system is supplied from a camera 80 and audio is supplied from a microphone 82. Movie film containing visual and possibly audio information is transferred to the encoding system 98 through the transfer device 88 which contains a light to illuminate the film and a light detector to capture images from the film. The information from the input devices 80, 82, and 88 is stored in an electro-magnetic format before encoded by the encoding system 98, such as by an audio or video tape recorder. The audio and video tapes are encoded by the system of Figure 1A.

[0044] The final digital bitstream produced by the encoding system is sent to optical disc mastering equipment 90 which presses optical discs, an optical disc writer 84 which writes optical discs, or may be transmitted to a television receiver or set-top box decoder via the television transmitter 86. The pressing and creation of optical discs is performed in a known and commercially available manner using a master disc, for example. The encoded information is also usable with a video on demand system. The encoded bitstream is ultimately decoded by decoding processes which correspond to the encoding processes and the audiovisual information is displayed to a user on a television or monitor. The decoding devices include digital video disk players which decode the encoded disks for display on a television or monitor.

[0045] It is evident from Figure 1B that the encoding system inputs electrical signals representing audio and images, transforms the electrical signals to a new format, and ultimately, the signals are decoded to recreate the original information which was encoded.

III. SYSTEM OPERATION

[0046] Figure 2 is a flowchart illustrating the general operation of the system illustrated in Figure 1A. As seen in Figure 2, the system goes through several "states" during its cycle of operation. The dark solid lines show the normal operational path, the solid lines illustrate an interrupt exit path, and the dashed lines illustrate a by-pass path.

[0047] The first operation by the system occurs in the set-up state 100. This step allows input parameters to be specified

LINKPLAY EXHIBIT 1004
Page 1623 of 4822

before the compression state 102 is entered. In the compression state 102, a video image stored on a tape in the playback VTR 51 is encoded into a digitally compressed format by the video encoding device 50.

[0048] After the compression state 102, the user is permitted to manually edit the compression process based on the picture quality of the previously compressed data in the edit state 104. This allows the user to increase or decrease the picture quality within regions of a frame of video or over a time period. It is to be noted that the term "edit" does not mean that the scenes themselves are added or removed from the picture stream. In the present application, the term "edit" means to change the quantization and subsequently the quality of the video. After the image is edited in the edit state 104, it is then necessary to perform the compression operation again in the compression state 102 in order to encode the data according to the new user-specified parameters.

[0049] After the user is satisfied with the edited form of the video, all information including the audio, video, and other information such as subtitles are combined together into a desired format in the format state 106. The completion state 108 erases any temporary files and the process terminates in the exit state 110. More detailed information with respect to the individual states illustrated in Figure 2 will now be explained with respect to Figures 3-6.

[0050] Figure 3 illustrates the steps performed in the set-up state 100. Step 120 begins the set-up state and step 122 initialized the system. This step will begin the execution of system control software contained within workstation 30 and start up other programs such as a graphical user interface (GUI) program which is responsible for the user interface displayed on the console display 31 and having parameters input through the keyboard 32 and pointing device 33. Also during step 122, the system controller software in workstation 30 queries other devices to determine the system condition including the available devices of the system.

[0051] Step 124 displays information to the user through the GUI. A welcome message, information pertaining to the system, parameters used for the encoding process, and the existence of any previously encoded files are displayed. The displayed system information includes the devices connected to the system, including audio encoders, VTRs, audio tape decks, and their configuration. In the event that the system contains previously encoded files, information is displayed to the user describing the audio and video encoding parameters used to generate the files.

[0052] Step 126 then inputs information from the user such as a command to begin encoding using standard or user specified system parameters, any changes to the system parameters, or a confirmation that the selected parameters are acceptable. The user also enters the beginning and ending time codes for the encoding operation.

[0053] Step 128 then processes the input data by transferring the input data from the graphical user interface to the system controller software in order to prepare for the encoding and compression operation. The set-up state ends in step 130.

[0054] Figure 4 illustrates the process performed in the compression state 102. According to the normal operation flow, the user has already specified a video and any accompanying audio sequence to be compressed in the set-up state. The user must then instruct the system to perform an automatic or auto-compression procedure. The video compression during the auto-compression procedure does not permit the user to control the dynamic quantization process over a time period or within a frame or video. It is intended as an initial compression run in which the coding decisions are made based on objective measures as computed by the video encoding device 50. If the auto-compression has already been performed, the user may manually adjust the quantization in an edit-compress mode either spatially (over a frame or video) or temporally (over a video sequence).

[0055] In the compression state, after starting in step 140, step 142 prepares for the compression operation. In this operation, parameters for the compression process such as the time codes of the video and any accompanying audio to be compressed along with the compression parameters are downloaded to the video encoding device 50 from the workstation 30 through the workstation 10.

[0056] Step 144 then determines the type of compression which is desired. The first time information is compressed, the compression process is performed automatically and without user intervention. Step 146 prepares for the auto-compression process. During this preparation, the workstation 30 determines which of the digital audio tape players 71a-71d and the associated audio encoding devices 70a-70d are to be used. Also, the workstation 30 commands the workstation 20 to prepare for bitstream capture from the output of the video encoding device 50 and the audio encoders 70a-70d through the audio interface unit 72. Further, the playback VTR 51, the record VTR 52 and the DAT players 71a-71d are rolled to the appropriate starting position of the tapes. Workstation 40 then signals the playback video tape recorder 51, the record tape recorder 52, and the DAT players 71a-71d to begin rolling in step 148. Step 150 then encodes the video data according to the desired format such as an MPEG-1 or MPEG-2 format using the video encoding device 50. Also, the audio data is encoded according to the desired format such the Dolby AC-3 format.

[0057] It is necessary to create log files to assist later in the manual editing process. The log files indicate parameters of the encoded video data which are necessary in order to perform later editing of the video data. The encoding process is stopped in step 160 when the appropriate time code is reached. This time code may be the end of the information to be encoded or may be at a user specified point which is not at the end of the video sequence.

[0058] If step 144 determines that the auto-compression process has already been performed and a user desires to manually change the picture quality of one or more frames of video using the parameters specified during the edit state

LINKPLAY EXHIBIT 1004
Page 1624 of 4822

as described with respect to Figure 5 below, step 144 will determine that the edit-compression operation is to be performed. Step 152 prepares for the edit compression operation which includes cueing of the playback VTR 51 to the desired starting point which is the point at which the manual editing begins. Also, the audio equipment is disabled as there is no need to change the audio information once it is encoded in the auto-compression operation. Step 154 then rolls the playback VTR 51 and the record VTR 52.

[0059] Step 156 subsequently begins encoding the video data according to the parameters which have been specified by the user in the edit state described in Figure 5. The edit compression is terminated in step 160 when the ending time code is reached. Step 162 displays a message to the user that the compression and encoding is complete and the process ends in step 164.

[0060] Figure 5 illustrates the process performed in the edit state 104. As previously explained, the editing referred to is not the conventional editing of video where scenes are removed, shortened, or moved within a video sequence; the edit state in the present writing refers to changing the quality of a portion of or time sequence of the video by manually setting the quantization of the video.

[0061] After starting the edit state in step 170, step 172 asks a user if he wishes to manually edit the video by changing the bit allocation of the encoded video. If the user does not desire to change the bit allocation, the process ends at step 188. If the user desires to change the bit allocation, the user defines the video segment to be edited in step 174. This is performed by selecting a time period of the video over which the picture quality is to be changed. Step 176 then determines if the user desires spatial or temporal editing. Spatial editing is where the quantization or allocation of bits is changed within a frame of video whereas temporal editing is where the quantization or allocation of bits is changed over a time period. If the user selects spatial editing, flow proceeds to step 178 which inputs regions within a frame to be edited. Also inputted is the relative degree of change to be applied. For the present invention, an integer scale of -5 through +5 including zero is used to indicate the relative amount of change. A user will mark a region of video with the keyboard 32 and/or pointing device 33 and assign one of the integers between and including -5 through +5. It is also possible for the user to indicate that the previously set quantization level for a region is not to be changed. For example, if the user desires an increased bit allocation for a certain region, it is necessary that the bit allocation for other regions must be decreased. If the user sets a region to be non-changeable, also referred to as "protected", the needed bits for the improved quality area will not be taken from the protected area.

[0062] If step 176 determines that the user desires temporal editing, flow proceeds to step 180 which inputs how the time periods are to be edited. Similar to the spatial editing, for the temporal editing, the user assigns integer values between -5 and +5 indicating the relative degree of change to be applied to a particular sequence of video. The change will be effective for the entire time period which has been selected.

[0063] After either the spatial or temporal editing is performed in steps 178 and 180, step 182 re-encodes and recompresses the video scene to an MPEG-2 format, for example, according to the parameters which were manually entered. The previously encoded data is displayed next to the newly edited data on the video display monitor 61 in order for the user to closely compare how the quality has been changed.

[0064] When the operator indicates the segment to be edited, the video from the VTR 52 is transferred to the storage device 60, taking up at most half of the total available memory in the storage device 60. That segment corresponds to the "before" segment. During the edit-compression process, the bitstream is captured by workstation 20, and stored as a separate disk file from the master video bitstream file. The master video bitstream file is the compressed bitstream for the entire movie. The edit-compressed video is recorded in VTR 52. During the encoding process, the operator will be viewing the picture at the same time it is being recorded on VTR 52. Should the operator wish to view the picture after the encoding, the VTR can be used as a playback machine. Should the operator wish to do a "before-and-after" comparison, then the corresponding segment from the VTR 52 is transferred to the storage device 60, consuming up to the remaining half of the total memory of the device 60.

[0065] After this encoding, the operator makes a decision as to whether the changed video is to be kept. Should the operator decide to disregard the "after" segment, the master bitstream is left intact, and the edit-compressed bitstream is deleted. To allow the VTR 52 to reflect the master bitstream, an additional auto-compression on the edited segment is performed solely for the purpose of recording the "before" segment. The alternative of using the device 60 to transfer the "before" segment back to the VTR 52 is also possible. Should the operator decide to keep the "after" segment, then the master bitstream file must be updated to include the edit-compressed bitstream file, as discussed below.

[0066] If the user determines that the video is acceptable in step 184, the previously encoded data is replaced with the newly encoded data. This step is explained in further detail in a separate section below. After the newly encoded data replaces the previously encoded data, step 192 determines if the user desires more editing. If more editing is desired, flow proceeds back to step 174. If the user has finished editing the video, flow proceeds to step 188 which terminates the edit state.

[0067] If the user determines that the video is not acceptable in step 184, the user may exit the editing operation in step 186 so that the editing changes are not recorded or the user may re-edit the video. The video is re-edited by the users selecting new parameters in the editing. It is also possible, if the user desires, to define a new video segment to

LINKPLAY EXHIBIT 1004
Page 1625 of 4822

be edited.

[0068] When the user is satisfied with the compression (either auto-compression alone, or auto-compression in conjunction with edit-compression), the individual data files including the compressed video, compressed audio, and any other files are ready for formatting into the final format and the format state 106 is entered. The auxiliary data may include information such as subtitles to be included into the final formatted data file. In the flowchart illustrated in Figure 6, after starting the format state in step 200, the workstation 20 reads the auxiliary data, should the user decide to include the auxiliary data, from the digital storage devices 22 which include, for example, a floppy disk within a floppy disk drive. Step 204 then combines the auxiliary data with the encoded audio and video to generate the final formatted file. The final digital bitstream generated by the system may be constructed as a 'Program Stream' as defined in ISO/IEC 13818-1. Alternatively, the final bitstream may be any type of format such as the direct satellite DSS format, a VBR (variable bit rate) format suitable for use on an optical disc, or any other type of format. As the encoded video and audio may be stored separately, the system allows the same encoded video and audio to be encoded into different final formats. This is accomplished either using one formatter having the capability of generating different formats or by separate formatters. Step 206 stores the formatted file onto the disk 21.

[0069] The completion state is entered in step 208 which writes the formatted file to a tape within the digital storage devices 22 in step 210. Subsequently, the exit state is entered in step 212 which performs various "house cleaning" functions which delete temporary files which are no longer needed. In the event of an abnormal exit request such as before the compression process is finished, an abort procedure will be carried out within the workstation 30 which stops the audio and video tape players and also cleans up potentially corrupted data files. The process then ends in step 216.

IV. SYSTEM ALTERNATIVES

[0070] The preferred system components and their operation have been described above and of course, alternative hardware may be substituted for the hardware disclosed in Figures 1A and 1B. The video encoding device 50 is preferably an MPEG-2 video encoder, if available. However, a commercially available ISO/IEC11172-Video standard encoder otherwise known as an MPEG-1 video encoder may be used. Necessary modifications to the MPEG-1 encoder will need to be made to allow the MPEG-1 buffer management module to be controlled from an external source such as the workstation 10. Additionally, the input video support must also be augmented to include ITU-R 601 (formally CCIR 601) resolution video. Such modifications can be implemented by those skilled in the art of digital video encoding without undue experimentation.

[0071] While four audio tape players and four audio encoders have been illustrated in Figure 1A, other configurations can be easily adopted for the audio system. For example, a commercially available 8mm digital audio player may be used instead of a multiplicity of DAT players. Using a single TASCAM digital audio player, only a single RS-422 control signal is needed from the workstation 40 while up to eight separate audio channels can be simultaneously supported. The audio encoders must of course be able to accept eight audio inputs simultaneously and the audio interface unit 72 must also be modified to accommodate the increase in audio data streams from four to eight.

[0072] As an alternative to simultaneous audio and video encoding, the audio and video encoding can be done at separate times and/or locations and combined into the final format at a later time. This requires the use of time codes, which are known, such as the SMPTE (Society of Motion Picture and Television Engineers) time code format. The time codes are generated by a video tape player, a workstation, or a separate stand-alone time code generator.

[0073] The record VTR 52 may be eliminated if a video decoder is available and connected to the workstation 20. In that case, the reconstructed video would be generated from the disk files in disks 21 instead of being recorded from the video encoding device 50 during the compression phase. Eliminating the record VTR 52 will have a significant savings in cost of the system, both in terms of equipment as well as tape cost.

[0074] Furthermore, the ability to perform graphical operations onto the graphic overlay plane of the display video can be supported on the console display using the commercially available X-terminal option. For example, the console display 31 may be replaced with an X-terminal that is capable of displaying video with an overlay graphic window. A commercially available multimedia display terminal such as the HDSV View Station from Human Designed Systems of King of Prussia, Pennsylvania, provides the necessary features. The video from the video decoder can then be connected to the X-terminal for display and for manual editing region definition operations. The professional quality video from the video decoder should, however, be displayed on a professional monitor such as monitor 61 shown in Figure 1A to allow the human operators to assess the quality of the reconstructed video signal.

V. REDUCING THE NUMBER OF AUDIO ENCODERS WHILE MAINTAINING OPTIMUM ENCODING EFFICIENCY

[0075] As explained above, Figure 1A illustrates an encoding system having a multipass video encoding device 50 and four one-pass audio encoders 70a, 70b, 70c, and 70d. The playback VTR 51 supplies the video signal to be encoded to the video encoding device 50 and the digital audio tape (DAT) players 71a-71d supply the audio signals to be encoded

LINKPLAY EXHIBIT 1004
Page 1626 of 4822

to the one-pass audio encoders 70a-70d, respectively.

[0076] The preferred embodiment of the invention encodes eight audio tracks in the final formatted audiovisual data stream. Each audio track includes one or more audio channels. As an example, an audio track may include left and right channels, although the audio tracks can include audio in other arrangements. The manner of determining the number of audio encoders in order to perform the most efficient audiovisual encoding and the manner of performing this audiovisual encoding is explained in Figure 7. In Figure 7, after starting, step 240 determines P, the number of passes required for video encoding. In order to encode the video into an MPEG format, typically two or more passes are necessary to encode the video. For the present invention, the preferred number of passes is two but three or more passes are also possible. Step 242 then determines T, the number of audio tracks to be encoded. In the present invention, the number of audio tracks is preferably eight although different numbers of audio tracks are also possible. Next, step 244 calculates AE, the number of one-pass audio encoders which are needed to perform the most efficient audiovisual encoding. The number of audio encoders necessary is equal to the number of audio tracks to be encoded divided by the number of passes required for the video encoding. In the preferred embodiment, eight audio tracks divided by two passes for the encoder is equal to four audio encoders necessary. If the calculated number of audio encoders is not an integer, it is necessary to round up the calculated number of audio encoders to be an integer. For example, if only seven audio tracks are to be encoded for a two-pass video encoder, 7/2 = 3.5, and 3.5 rounded up to the next highest integer is 4. During the encoding process performed by this system, one video pass uses only three of the audio encoders while the other video encoding pass uses all four audio encoders. It is to be noted that steps 240-244 may only need to be performed upon the initial designing of the system, and it is not necessary to calculate the number of one-pass audio encoders needed every time an audiovisual work is to be encoded. Further, steps 240, 242 and 244 are either computer/machine implemented by a computer such as workstation 30, or may be determined by a user or designer of the audiovisual encoding system.

[0077] After the number of audio encoders has been determined and the audiovisual encoding system is set up, the audiovisual encoding process may begin. Step 246 sets COUNTER equal to one. COUNTER is a variable used to count each pass of the video encoding system. Step 248 then performs the first video encoding pass while performing one-pass audio encoding on some, but not all, of the audio tracks. For example, if there are eight audio tracks and a two-pass video encoder, four audio encoders may be used during the first pass. Step 250 then increments the COUNTER by one. Step 252 performs the next video encoding pass while performing the one-pass audio encoding of unencoded audio tracks using the same audio encoders which were used in step 248. For example, during the second video encoding pass, the four audio encoders used in step 248 can encode audio tracks five through eight. Step 254 then determines if COUNTER is equal to P, the number of required video encoding passes. In the preferred embodiment, there are only two video encoding passes and therefore, the process would end. Otherwise, flow returns back to perform the loop of steps 250, 252, and 254.

[0078] Of course, other embodiments are possible besides using a two-pass video encoder for an audiovisual work which has eight audio tracks. For example, a three-pass video encoder may be used with six audio tracks. In this situation, the most efficient number of audio encoders is three (6/2 = 3). If a two-pass video encoder were used and a total of six audio tracks were to be encoded, the most efficient number of audio encoders would be three. As an alternative, a three-pass video encoder may by used in a system in which there are five audio tracks to encode. This alternative system requires two one-pass audio encoders with any one of the video encoding passes operating while only one of the audio encoders is used and the other two video encoding passes occurring while both of the audio encoders are being used. Of course, other combinations of video encoders and numbers of audio tracks are possible as will become apparent to one of ordinary skill, in light of the teachings contained herein.

[0079] An important aspect of the present invention in being as efficient as possible is to always perform audio encoding when a pass of the video encoding is occurring and to always perform a pass of the video encoding when the audio encoding is occurring. In this arrangement, at least some of the audio encoders will always be used during each pass of the video encoding operation. To be as efficient as possible, when the number of audio tracks is not evenly divisible by the number of audio encoders (i.e., there is a remainder when the number of audio tracks to be encoded is divided by the number of video passes), during the entire encoding process, there will be a number of audio encoders idle which is equal to the number of video passes minus the remainder. For example, for a three pass video encoder and four audio tracks, two audio encoders are needed and there is a remainder of 1 (e.g., 4/3 = 1 with a remainder of 1). Therefore, to be as efficient as possible and complete the audio encoding at the same time the three pass video encoding is completed, there can be no more than two (e.g., 3 video passes minus a remainder of 1 is equal to 2) times that the audio encoders are idle. The two times that the encoders are idle may occur in the same pass (there are two different audio encoders idle in the same video encoding pass) which would means that the other two video encoding passes in the above example each utilize all of the audio encoders, or there may be two video encoding passes in which there is one idle audio encoder. In these two passes, the same or different audio encoders may be idle.

LINKPLAY EXHIBIT 1004
Page 1627 of 4822

VI. LOG FILES FOR STORING INFORMATION OF THE ENCODED VIDEO, AND LOG INTERPRETER UTILITIES TO ACCESS THE LOG FILES AND THE ENCODED VIDEO

[0080] The disclosed encoding system initially encodes video into a compressed digital format such as MPEG-2, MPEG-1, or other digital video encoding format. This encoding is described above in section II pertaining to the system components and section III pertaining to the general system operation. The encoded video data is then stored in a digital format on a hard disk, for example.

[0081] During the digital video encoding process, statistics and other information of the video are generated and stored in log files. One of ordinary skill in the art who is familiar with MPEG video encoding is familiar with the type of statistics and information which describe the video and how to generate these statistics and information. In a conventional encoding system, information and data of this type is generated but ordinarily discarded after the video encoding. However, for the present invention, this information which has been generated such as the total bits used to code a picture, the mean squared error encoding a picture, the mean quantizer scale for each picture, etc., is dumped from system RAM to log files after the encoding process. Examples of log files used by the present invention are illustrated in Figures 8A-10B. These log files are particularly useful in a VBR system and are generated during an initial pass (passes) of the video encoder. The log files are then used during the final encoding pass and during later editing and re-encoding of the video.

[0082] There are three different types of log files which may be used by the present invention; a sequence layer log file, a picture layer log file, and a macroblock layer log file. Each of these log files describes statistics and information of its respective layer. There are two different types of macroblock layer log file formats; the full format illustrated in Figure 10A and a short format illustrated in Figure 10B. The full format is used when detailed information of the macroblock layer is desired and a short format is used when it is not necessary to keep track of all of the details of the macroblocks. Due to the large number of macroblocks in a two hour movie, the macroblock log files will consume a large amount of storage space. Therefore, unless a large storage space is available, it may not be possible to store the full or short format macroblock files. However, if it is necessary to obtain the macroblock information, it can be reconstructed by decoding the encoding video or by other means.

[0083] It is to be noted that many of the details of the log file formats illustrated in Figures 8A-10B are unimportant and these file formats have been provided simply as illustrations of the data which may be generated during the encoding process. However, some of the information in the log files is important as it is used during the process of changing the quality of the video. As will be explained below, in order to determine an edit point of the encoded video, it is necessary to sum the number of bits consumed by earlier pictures. The information which is summed up is illustrated in Figures 9B and 9C and pertains to the number of bits for each picture (generated_bit_picture), the number of bits for the sequence header (bits_sequence_header), the number of bits for the group of pictures (GOP) header (bits_GOP_header), and the number of bits used as stuffing or bits used to fill space (stuffing_size).

[0084] Figure 11 illustrates a flowchart of different functions which are performed using the log files. After the initial encoding of a video segment, step 270 reads in the picture and sequence layer log files generated from the automatic run of the video encoder which are illustrated in Figures 8A-9C. Step 272 then creates an index table indicating the byte offsets of the log file records for every picture recorded in the encoded video. Two tables are created; one indexed by frame number and the other by time code. The time code index table contains the byte offset into the log file of each picture and is sorted in the increasing order of time codes. Similarly, the picture frame index table contains the byte offsets into the log file which are sorted in the increasing order of input picture numbers (as seen by the encoder).

[0085] The generated index tables help to quickly pick out the log information of any desired picture or macroblock. Although fixed size records are used to store the sequence, picture or macroblock data within the log file, the encoder re-orders the incoming pictures prior to coding them. In addition, when dealing with a movie (24 frames per second) source that has been 3:2 pulled down to 30 frames per second, certain time codes are skipped. The index tables provide a means to quickly locate the appropriate data despite the reordering of pictures and the skipping of time codes.

[0086] It is to be noted that the present encoding system does not maintain a directory or index of the number of encoded bits used in coding each picture as the encoded video will frequently change as various frames and time periods of the video have their quality manually altered. However, the log files describing the information of the encoded video are fixed size records and therefore, it is easy to maintain a directory or index of the information within the log files, even when the quality of the pictures or time periods is altered.

[0087] After the creation of the index tables, step 274 allows a user to quickly obtain various information from the log files using various log file utilities illustrated in blocks 276, 278, and 280. When records in the log files are needed, the created index is used to determine the desired record in the log file and the complete record is pulled out of the log file. This record is then parsed for retrieving the individual parameters which are desired. The process of parsing individual records with the help of index tables is the basis for all of the other log interpreter utilities. Parsing processes are known and one of ordinary skill in the art knows how to retrieve desired information from an organized file.

[0088] Step 276 is the log file cut and paste utility. After manually changing the quantization (and picture quality) within a frame or over time periods, the original set of log files needs to be updated to correspond to the changes which occurred

LINKPLAY EXHIBIT 1004
Page 1628 of 4822

during editing. The log file cut and paste utility locates offsets in the log files corresponding to the edited segment time codes and replaces the information in the log files with new information corresponding to the re-encoded video. The process of re-encoding is explained below in section VIII.

[0089] Step 278 illustrates a utility to obtain parameters to enable editing of the digitally encoded bitstream. After the video is initially encoded and a user desires to alter the quality of frames or a time period of the video, it is necessary to determine the appropriate parts in the encoded video which are to be removed and replaced with the newly encoded video having the desired quality.

[0090] The byte offsets for the edit points in the bitstream of the encoded video are determined by calculating a cumulative tally of the bytes used to encode each picture up to the pictures corresponding to the edit point. In order to determine the total number of bytes or bits from the beginning of the file at which a specific picture is located, the number of bits used to encode each of the previous pictures is summed. The information which is obtained from the picture log file which is summed are the number of bits B_i used to encode the ith picture, the number of bits S_i comprising the sequence header, the number of bits G_i used to encode the Group Of Pictures (GOP) header, and the number of bits Ti used as stuffing. The bit offset for the Nth picture is calculated as

$$\sum_{i=1}^{N-1} (B_i + S_i + G_i + T_i) \qquad (1)$$

The byte offset is simply calculated by dividing the bit offset by eight. As an alternative to the use of the log files to determine the address offset in the encoded video bitstream, a bitstream decoder could be used to create a directory of picture offsets. However, this approach is cumbersome as the decoder has to run through the entire bitstream after each edit in order to update the directory.

[0091] Finally, step 280 illustrates a utility for rounding time codes to the start or end of the nearest Group Of Pictures (GOP). All cutting of the encoded bitstream or edits are performed at the boundaries of a Group Of Pictures (GOP). However, the GOP boundaries are unknown to the operator performing edits on the video. Once the operator specifies the edit segments start and end points, the system computes the GOP boundaries. The GOP boundary is established by looking at the picture layer parameter called GOP number stored within each picture record. By tracing through successive records of the picture layer log file, when the GOP number (GOP_number) in Figure 9B changes, there is determined to be a GOP boundary.

[0092] As the log files and log file utilities are closely related to video encoder, the log files are stored in a hard disk in the workstation 10 illustrated in Figure 1A and the log file utilities are executed by the workstation 10. Alternatively, workstation 30 or any other processing system may contain the log files and execute the log file utilities.

VII. CHANGING THE QUALITY OF THE ENCODED VIDEO

A. Changing the Quality Over Periods of Time

[0093] The encoded video produced by the encoding system is intended to be written to a final digital storage medium such as an optical disc. The encoded video on the disc is encoded at a Variable Bit Rate (VBR) which affords the possibility of large changes in bit rates among the different sections of video, as compared to Constant Bit Rate (CBR) operation. As the VBR coding allows a significantly different number of bits to be assigned to different pictures, the picture quality can be varied over time. Further, VBR affords the possibility of reassigning data rates to some user-defined sections of video in a scale not possible with CBR operation. Extremely large buffers would be required at the decoder to accomplish the same with CBR.

[0094] The encoding system allows what shall be referred to as manual editing but the manual editing is not the shortening or lengthening of scenes or substituting different scenes in a movie but is the changing of the picture quality. The picture quality may be changed over time and is referred to as temporal editing or temporal manual editing as will be discussed in this section. Another section will discuss the changing of picture quality within frames of data by shifting bits from one section of a frame to another section of a frame and will be referred to as spatial editing or spatial manual editing.

[0095] Three major constraints need to be satisfied in temporal editing or the re-allocation of bits among the various sections of an edit segment. These are:

1. The total capacity for the movie/video that is edited does not exceed the allocated capacity;
2. There is no buffer underflow within the edit segment; and

LINKPLAY EXHIBIT 1004
Page 1629 of 4822

3. There is no buffer underflow outside of the edit segment caused by the temporal editing.

[0096] Constraint 1 is handled by re-encoding the edit segment such that resulting edited segment consumes the same number of bits as the originally encoded segment. It is to be noted that an edit segment is the entire time period under consideration. Different periods within the edit segment are assigned qualities by a user and these periods of the edit segment are referred to simply as sections or time periods.

[0097] In order to determine if an underflow condition will exist, it is necessary to check the target bits using a model of the decoder. In VBR operation of the encoder, a modified version of the Video Buffer Verifier (VBV) model is used. This model is referred to as the simple modified model for VBR and may be used for a Digital Storage Medium (DSM). The VBV for VBR mode is known and defined in the MPEG-2 standard and may be specifically found at ISO/IEC 13818-2, Annex C, section C.3.2.

[0098] In the VBV model for the VBR, the buffer of the decoder may remain full but is restricted from underflowing. In addition, the buffer fills up at a constant rate, R_{max}, until the buffer is full and then no bits are transmitted to the buffer until the buffer begins to empty. At every frame interval, the bits required to decode one frame are removed from the buffer. With this in mind, it can be seen that in a VBR mode, buffer underflow is caused by allocating too many bits to a series of pictures. As the bits of the pictures are removed from the buffer, the buffer becomes empty and underflows if large numbers of bits of high quality pictures are removed at a rate faster than the buffer can be filled.

[0099] To prevent underflow, the number of bits used to encode a picture is reduced. Reducing the number of bits for a picture reduces the rate at which bits are removed from the buffer while keeping the fill rate the same. Buffer overflow cannot occur for VBR because if the VBV occupancy is full, then the bitstream transmission to the VBV buffer is immediately stopped. This means that there is no buffer overflow but simply a buffer full situation. The simple modified VBV model for VBR, as explained in ISO/IEC 13818-2, Annex C, Section C.3.2 is as follows:

1. bit_rate field at sequence_header represents the max_bit_rate.
2. The VBV is initially empty. After filling the VBV buffer full (i.e., VBV buffer size) at the rate of max_bit_rate, the decoding process begins. The picture data is immediately removed and decoded like CBR's VBV model defined in the MPEG-2 International Standard.
3. The bitstream is placed into the VBV buffer at the rate of max_bit_rate until the VBV buffer is full. If the VBV buffer occupancy is full, then the bitstream transmission to the VBV is immediately stopped. After the data for the next picture is removed, the bitstream transmission immediately starts.
4. The VBV buffer shall not underflow.
5. Every vbv_delay field is set to FFFF.

[0100] Changing the quality of the video over periods of time requires adding or removing bits from frames of video. An explanation into the number of bits that may be added or taken away from a section of video will now be provided. The number of bits which may be added or taken away from a section of video depends on the effective bit rate of the section of video under consideration. The effective bit rate, R, is calculated as:

$$R = (Total\ Allocated\ Bits/Number\ of\ Frames) * Frame\_Rate \quad (2)$$

Because the total number of allocated bits is directly proportional to the bit rate, referring to the number of allocated bits conveys the same meaning as the bit rate, except with different units. Assuming that a guaranteed minimum rate is applied to any section of video, say R_{min}, then the maximum number of bits that may be taken away from section 'i' is:

$$T_i = (R_i - R_{min}) * N_i / Frame\_Rate \quad (3)$$

where Ni is the number of frames in section 'i', and R_i is the effective rate of the same section. The guaranteed minimum is required in order to assure some minimum quality level for the section. Similarly, the maximum number of bits that may be added to a given section 'j' is calculated as:

$$T_j = (R_{max} - R_j) * N_j / Frame\_Rate \quad (4)$$

LINKPLAY EXHIBIT 1004
Page 1630 of 4822

where R_{max} is the maximum rate, or the rate at which the buffer fills up when not full and N_j is the number of frames of section j.

[0101] A description of the changing of the picture quality over time periods (temporal editing) will now be given with respect to Figure 12 and the Figures referenced therein. After starting, step 281 performs an automatic video encoding without any user intervention, as described above in sections II and III. It is to be noted that for the automatic encoding, there is a maximum and minimum rate of the generated data stream. The maximum bit rate is determined by the capabilities of the video decoder and R_{max} is, for example, 8 Mbits/s. The minimum bit rate may be specified by the user, depending on the minimum picture quality desired. A typical minimum bit rate R_{min} is 3 Mbits/s. It is desirable to set R_{min} to be different from and somewhat below the average encoding rate in order to allow portions of the video to have a lower bit rate and portions of the video to have a higher bit rate. If R_{min} is set slightly below the average encoding rate of the video, the resulting encoded video will have a fairly constant quality. Typically, if R_{max} = 8 Mbits/s and R_{min} is approximately 3 Mbits/s, the average value of the video should be around 3½ to 5 Mbits/s.

[0102] After the automatic video encoding is performed in step 281, it is necessary to determine a rate-quantizer function. The process of the invention determines how many bits are necessary to obtain a desired quality but when re-encoded, a quantizer value must be set in order to result in the desired bit rate. The rate-quantizer function may be calculated as in a conventional manner such as described in the articles "Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Disclosures" by Ramchandran et al from IEEE Transactions on Image Processing, Vol. 3, No. 5, September, 1994, pp. 533-545, "Bit Allocation for Dependent Quantization with Applications to MPEG Video Coders" by Ramchandran et al, IEEE ICASSP, 1993, pp. V-381-V-384, and "Joint Multi-channel Bit-Rate Control for Multiplexed MPEG Video using Rate-Distortion Curves," by Reininger, presented at a Princeton video conference, 1994.

[0103] Next, step 284 inputs the user quality settings for selected time periods. The user inputs priorities for different sections which are ultimately used to control the quantization of the pictures. The quantization controls the picture quality, also referred to as resolution, for lossy MPEG encodings. Figure 13 illustrates a user interface which is used in order to gather the information to change the quality of the video over periods of time. Initially, a user will enter the time period for the entire edit segment over which the quality is to be changed. The input of this information is not illustrated in the user interface of Figure 13 but is inputted before the user interface of Figure 13 is displayed. For the example illustrated in Figure 13, the time period for the edit segment which is to be changed may be, for example, time 0 minutes to time 5 minutes. After the user has described the total time period of the edit segment, then smaller time periods within the edit segment are assigned different priorities in the range of -5 through +5 including a priority P, indicating that the quality of this time period is protected and should not be altered. Priority "none" is synonymous with a priority of P. Of course, the encoding system is not limited to the above input scheme and other ranges or symbols may be used to input the user's desired priorities.

[0104] In addition to assigning priorities, the user will also enter the minimum bit rate. This bit rate may be different than the minimum bit rate used to encode the video during the automatic video encoding of step 281. Another feature of the user interface is that it allows the user to set a "background priority" which means that time periods within the selected time period which are not assigned a priority by the user have this background priority. For example, in Figure 13, if the background priority were assigned "1" the unspecified time period 00:03:20:01 through 00:03:59:99 would automatically be assigned a priority of "1" without the user having to manually assign the priority for this particular section.

[0105] On the bottom of the interface illustrated in Figure 13, the "insert" block is used to create a new time period within the edit segment, the "recall" block is used to alter a previously created time period, "update" is used to save the changes made to a time period, and "delete" is used to delete a time period.

[0106] It is important to remember that the ultimate digital storage medium for the encoded video such as an optical disc has a limited storage capacity which ideally will be filled to capacity if the automatically encoded data were written to the disc. Therefore, whenever the picture quality is increased and the bit rate for a particular time period is increased, there must be a section of the video which has its quality reduced in order to supply the necessary extra storage capacity to the time period which has its quality increased. Accordingly, it is not possible in the preferred embodiment of the invention to assign only one period a high picture quality without assigning another time period with a different picture quality because the bits necessary to increase the quality must be taken from another section of the video. The segment of video which has been edited therefore should have at least one section which has a priority which is different from the background section and not marked as a "protected" section in order for the temporal manual editing to properly operate. Additionally, the user selected priorities do not result in the same quality and bit rate every time they are selected but the quality of a particular section also depends on the assigned priorities for other sections and the lengths of the other sections, as will become apparent upon understanding the process to determine the modified number of bits for the pictures within the edit segment.

[0107] A feature utilized by the present encoding system is that during the automatic encoding process or at a later time, a viewer may review the encoded video. As the encoded video is being displayed to the viewer, the viewer may perform a function such as pressing the return key or "clicking" a graphical user interface on a "button" displayed on the

LINKPLAY EXHIBIT 1004
Page 1631 of 4822

screen in order to store points or regions of interest of the video without the need to stop the video or write down the time code of interest. These points of interest can be later called-up by the user and studied closely without the need to review the entire movie. This provides a convenient and efficient manner of determining points within the encoded video which a user may desire to change. This function is accomplished by storing a time code corresponding to the position within the encoded video of when the point or region of interest occurred.

[0108] After the user inputs the desired quality settings in step 284, step 286 computes the modified bits for each frame based on the user's input. This process is described in detail with respect to Figures 14-16B. The general procedure for changing the bit rates of the time periods within the edit segment of video is illustrated in the flowchart of Figure 14. In order to compute the bit rate based on the user's input, step 302 maps the user's input which was in the range between -5 and +5 to the range between 10 and 0. The initial input of -5 corresponds to the mapped setting of 10 and is the minimum quality and the input of +5 corresponds to the mapped setting of 0 and is the maximum quality. Similarly, an input quality selection of 0 corresponds to a mapped setting of 5, etc. The purpose of the mapping performed in step 302 is that the number of bits allocated is proportional to e (2.71828...) raised to a power of the quantity - 1 multiplied by the mapped setting. In this case, e^{-10} is less than e^{0} so that the number of bits for the smaller priorities settings is less than for the larger priorities. The process of determining target bits corresponding to priorities of -4 through +4 is explained in detail in Figures 16A and 16B.

[0109] Step 304 determines if there are any "protected" sections. A protected section is a section in which the quality of the picture should not be changed from the previous automatic encoding. If there are protected sections, flow proceeds to step 306 in which the number of bits which is allocated for the re-encoding of the protected sections is set to be equal to the previous number of bits resulting from the initial automatic encoding. In this manner, the picture quality of the protected sections will not change.

[0110] Step 308 determines if there are any sections which are set to the maximum or minimum qualities. The maximum quality corresponds to a user input of +5 and the minimum quality corresponds to the user input of -5. For the maximum and minimum picture qualities, it is desired to obtain encoded video which has the highest and lowest quality video, respectively, while still satisfying the video buffer constraints and not exceeding the storage capacity of the medium containing the encoded video. If there are sections which are set to the maximum or minimum qualities, flow proceeds to step 310 which computes the modified number of bits for the maximum or minimum cases. This process is shown in detail in Figure 15.

[0111] In Figure 15, the process of determining the number of bits allocated for the maximum and minimum cases begins in step 320 by allocating the minimum and maximum rate to sections of the video assigned to mapped priorities of 10 and 0, respectively. Step 302 of Figure 14 performed this mapping and the remainder of the description of the temporal editing, unless specifically stated otherwise, will refer to the mapped priorities. The maximum data rate, as previously described, is limited by the ability of the decoder and may be 8 Mbits/s and the minimum data rate is input as illustrated by the user interface in Figure 13 and may go below or higher than the minimum bit rate used during the autoencoding mode which typically may be 3 Mbits/s.

[0112] Step 322 then determines if there are any sections having a mapped priority of 1 through 9. This means that in addition to there being a maximum or minimum quality assigned, there are other intermediate qualities assigned by the user. If there are other mapped priorities, flow proceeds to step 324 which computes the average bit rate for all sections with mapped priorities between 1 and 9. If this average is not in a permissible range (between the maximum and minimum bit rates), flow proceeds to step 328. In step 328, the maximum rate, the rate providing the highest picture quality, is reduced when the average is less than the minimum rate. By reducing the maximum rate, more bits will be available for the non-maximum and non-minimum situations (mapped priorities of 1-9), thereby placing the average within or closer to the desired range. Alternatively, in step 328, if the average is greater than the maximum rate, the minimum data rate is increased, thus reducing the number of bits for the non-maximum and non-minimum situations (e.g. priorities of 1-9). After step 328, flow returns to 324 where the loop of steps 324, 326 and 328 is repeated until the average is in range. Additionally, the maximum and minimum bit rates are adjusted to ensure that the "Protected" sections receive the same bits as before the adjustment of quality.

[0113] Once the average is determined to be in range in step 326, flow proceeds to step 330 which assigns the calculated minimum and maximum rates to the sections having the mapped priorities of 10 and 0, respectively. The remaining bits are then distributed in step 332 to sections which are not protected nor maximum or minimum situations. The sections of video having the mapped ranges of 1-9 and the manner of changing the data rates are described later with respect to Figures 16A and 16B.

[0114] If step 322 determines that there are not any sections having mapped priorities of 1-9, meaning there are only maximum, minimum or protected sections, step 340 adjusts the minimum and maximum rates, as necessary, until the final allocation of bits is equal to the initial allocation of bits from the automatic encoding. There are two alternative manners of performing step 340. First, the maximum bit rate R_{max} is fixed, for example, at 8 Mbits/s. The total bits for the overall edit segment is known and therefore, R_{min}, the minimum data rate is simply adjusted in order to achieve the same total number of bits as were obtained during the automatic encoding. If it is not possible to adjust R_{min} to achieve

LINKPLAY EXHIBIT 1004
Page 1632 of 4822

the previous number of total bits, then the maximum data rate, R_{max} is adjusted in order to achieve the total number of bits as were obtained during the automatic encoding.

[0115] As an alternative, step 340 assigns the maximum quality section with a priority of 1 and the minimum quality section with a priority 9 and perform the bit allocation algorithm illustrated in Figures 16A and 16B to determine the new bit rates for the time periods of the edit segment. Step 342 then assigns the data rates calculated according to Figures 16A and 16B, which are explained below, to the sections having the lowest and highest assigned qualities and the process of Figure 15 ends.

[0116] After step 308 and step 310, if necessary, are performed, step 312 determines if any sections have a mapped priority of 1-9. If this is the case, flow proceeds to step 314 which computes the modified number of bits for the mapped priorities of 1-9 as described in detail in Figures 16A and 16B.

[0117] The first step in Figure 16A is to determine the size of a common bit pool to be used. Initially, all sections with a mapped priority of 1-9 which are not protected have a certain number of bits removed. A factor k_1, called the redistribution depth, is used for this purpose and for example, is set to 0.4 which means that 40% of the previously allocated bits (i.e. from the automatic encoding run) are to be removed from every section which is not protected nor marked with the maximum or minimum quality.

[0118] Next, step 362 determines how much the user selected quality will affect the quality. A variable k_3, called the dynamic range factor, is set and determines the quality change which will result from the priorities Depending on how k_3 is set, a priority of 2, for example, may impose different quality changes such as a relative small increase in quality or a relatively large increase in quality. Typically, the dynamic range factor k_3 is set to 1.0.

[0119] To use the user selected priority on the redistribution of bits, an exponential relationship of the form exp(-priority/k_3) is used. The dynamic range factor, k_3, in the exponent determines the shape of the exponential curve. The curve gets steeper for larger values of the dynamic range factor, effectively increasing the disparity in bits reallocated among the sections with varying priorities. This negative exponential means that as the absolute value of the mapped priority increases (lower priority sections), the bits allocated to that section decrease, i.e., mapped priority of one has the highest priority and increasing numbers up to nine have a lower priority.

[0120] Step 364 normalizes the user selected priorities. This means that the user selected priorities are used to determine the relative priority of each section. First, E_i is calculated.

$$E_i = \exp(-p_i/k_3) \qquad (5)$$

where p_i is the mapped user selected priority and i is the section time period number within the edit segment, excluding the time periods which are protected or have a maximum or minimum quality. Next, the priority normalization factor k_2 is calculated such that

$$k_2 = \sum_{i=1}^{N} E_i \qquad (6)$$

where N is the number of section. The priority normalization factor is needed to ensure that the sum of all bits reassigned according to the priorities always equals the available common pool of bits.

[0121] The relative priority of each section is then calculated as

$$R_i = E_i/k_2 \qquad (7)$$

As R_i is the relative priority fraction for the ith section, the summation of all R_i will be unity.

[0122] Step 366 then normalizes the sizes of the sections. This is necessary as the time periods with the assigned priorities have different sizes and the bits should be distributed in proportion to the size of the different sections. First, the fraction T_i of the number of frames in each time period over the total number of edited frames (frames with a nonprotected non-max or non-min priority) is determined according to

LINKPLAY EXHIBIT 1004
Page 1633 of 4822

$$T_i = N_i/N \quad (8)$$

where N_i is the number of frames in the ith time period, and N is the total number of frames in the edit segment; that is all frames which are being altered and assigned a priority which is not protected and not a maximum nor minimum priority. Then, the size normalization factor k_4 is calculated according to the equation

$$k_4 = \sum_{i=1}^{N} (T_i * R_i) \quad (9)$$

where N is the number of sections. From step 366 in Figure 16A, flow proceeds to step 368 in Figure 16B.

[0123] In Figure 16B, step 368 determines the common pool of bits according to the equation

$$C = \sum_{i=1}^{N} (k_1 * B_i) \quad (10)$$

where B_i is the total number of bits which were used during the automatic encoding procedure of the ith section plus any bits distributed in step 332 of Figure 15. The summation is over 'N' sections.

[0124] Step 370 then determines the target number of bits allocated for each section according to the equation

$$F_i = C * T_i * R_i / k_4 + (1 - k_1) * B_i \quad (11)$$

The target number of bits for each section F_i is equal to a percentage of the common pool of bits plus the original number of bits (B_i) reduced by the bits given to the common pool of bits. The percentage of bits from the common pool of bits is based on the fraction of the total number of frames of each section (T_i), the relative priority of the section (R_i) and the size normalization factor (k_4) which is dependent on the priority of a section multiplied by its corresponding fraction of frames summed for each of the sections.

[0125] It is possible that the target number of bits allocated for each section as computed in step 370 is not between the maximum and minimum data rates. If this is the case, step 372 reduces the redistribution depth k_1 by 0.01 and flow returns to recalculate the common pool of bits in step 368 and to recalculate the target number of bits in step 370. The loop of steps 368, 370, 372 and 374 is performed until the targeted number of bits calculated in step 370 is between the maximum and minimum data rates.

[0126] Step 376 then computes the bit scale factor Si and the scale factors for intra and non-intra frames SI_i and SN_i, respectively. Intra and non-intra frames require,different number of bits to obtain the desired quality. First, the bit scale factor Si is calculated according to the equation

$$S_i = F_i/B_i \quad (12)$$

Next, the scale factors for the intra and the non-intra pictures are calculated. The intra scale factor SI_i is calculated according to the equation

$$SI_i = Si - (I_f * S_i') \quad (13)$$

Large intra frames can potentially cause buffer underflow and the intra factor If provides a safety factor by limiting the variation in the bit assignment of the intra pictures to 20% when If = 0.8 (i.e., SI_i is limited to the range $0.8 < SI_i < 1.2$). Note that $S_i' = S_i - 1.0$. The non-intra bit scale factor SN_i is computed as

LINKPLAY EXHIBIT 1004
Page 1634 of 4822

$$SN_i = S_i - (-I_f * S_i' * BI_i / BN_i) \qquad (14)$$

where BI_i and BN_i are the total intra and non-intra bits summed over the ith section by picture type.

[0127] Finally, the number of bits to be used for each frame is determined using the original number of bits multiplied by the appropriate scale factor (either the intra or non-intra scale factor). The process of Figure 16B (and also Figure 14) then ends.

[0128] At this time, the modified number of bits for each frame has been calculated and step 286 of Figure 12 is complete. However, it is possible that the changed number of bits for the frames may cause buffer underflow or the occupancy of the number of bits at the trailing edge of the edit segment has changed so that subsequent frames outside of the edit segment will underflow. These situations are examined and corrected, if necessary, in step 288 of Figure 12 which is detailed in Figure 17.

[0129] Figure 17 illustrates the process to check and correct for buffer underflow and to check and correct for possible problems due to bit reallocations at the trailing edge of the edit segment. In Figure 17, step 390 first scans the bit allocation for each of the frames of the edit segment. Step 392 then checks for underflow for the new bit distribution using the VBV model described above. A list of points is created where underflow problems exist (u_flow list) and the number of bits necessary to eliminate each underflow is determined and stored in u_val. It is important to keep in mind that underflow occurs due to a picture being represented by too many bits. When a high quality picture represented by a large number of bits is read out of the buffer by the decoder, a large number of bits is removed from the buffer when the picture is decoded and these bits cannot be replenished fast enough. Therefore, to eliminate underflow, bits are removed from pictures. Additionally, step 392 creates a list of points where the buffer is full and stores these points in o_flow list. Note that as explained above, there is no problem with a full buffer as when the buffer becomes full, data is no longer transmitted to the buffer.

[0130] Step 394 examines the buffer occupancy at the trailing edge of the edit segment. In the VBR mode of operation, it is possible for the buffer occupancy at the trailing edge of the edited video segment to be different from that of the run that was modified, even if the edited segment is replaced by exactly the same number of bits. This arises from the idiosyncrasy of the VBV model for VBR. Hence it is important to examine buffer occupancy constraints. It is fairly straightforward to show that no problem exists if the buffer occupancy at the trailing edge of an edit segment is exactly the same or higher as it was before the redistribution of bits. If the buffer occupancy is exactly the same as before, then that of subsequent frames also remains exactly the same as the previous run, which was already checked for VBV violations. If the occupancy is higher than before, there is the possibility of the buffer becoming full during subsequent frames, and that is a perfectly allowable condition as when the buffer is full, data will no longer be fed to the buffer and underflow does not occur. However, a third case arises when the buffer occupancy is lower at the trailing edge, compared to the previous run. This has the potential for creating buffer underflow in subsequent frames and needs to be examined closely.

[0131] There are two manners in which the last frame of the edit segment can be treated for preventing buffer underflow in frames outside of the buffer segment. The first approach is the easier of the two. If the number of bits of the last frame of the edit segment results in a buffer occupancy which is greater than or equal to the original encoding (stored in the log files), no underflow problem exists and therefore, no action needs to be taken. If the number of bits in the buffer for the last frame of the edit segment is less than the number of bits in the buffer for the originally encoded segment, this reduced number of bits may or may not cause underflow for frames outside of the edit segment, depending on the content of the frames after the edit segment. Instead of determining the number of bits needed to prevent underflow from occurring, the occupancy of the buffer for the last frame of the edit segment is simply adjusted to be the same as for the original encoding. For safety purposes, the buffer occupancy or fullness for the end of the edit segment may be raised to 10% above the occupancy of the original encoding to assure that there is no chance of underflow. If the buffer occupancy of the last frame of the edit segment needs to be raised to be at the level of the original encoding, the number of bits of the ending frames need to be reduced as performed in step 398. The last frame is treated as underflow point, is placed in u_flow list and the number of bits which need to be taken from the last frame to raise the buffer level up to the level of the original encoding are added to u_val.

[0132] A second approach for handling possible underflow caused by the re-encoding of the edit segment is an iterative process which estimates the minimum possible level of the buffer for the last frame of the edit segment which will not cause a buffer underflow condition outside of the edit frame. This is done by a process which estimates the greatest possible difference between the buffer occupancy of the original encoding and the buffer occupancy of the re-encoding for the last frame of the edit segment.

[0133] Turning to the flowchart illustrate in Figure 18, the buffer occupancy of the original encoding is assigned B_0 and has been stored in the log files of the original encoding. The buffer occupancy at frame 'i' beyond the boundary of the edit segment is B_i, where i = 1...n, where 'n' corresponds to the last frame in the movie to be encoded. How much

LINKPLAY EXHIBIT 1004
Page 1635 of 4822

lower than the original encoding the buffer occupancy for re-encoding may go without creating underflow conditions for any of the successive frames must be determined as opposed to the first approach which simply sets the buffer level of the re-encoding to at least the point it was for the original encoding. Let the new lower buffer occupancy at the boundary that satisfies this condition be B_0'. This is the minimum allowable buffer occupancy at the edit segment boundary. The possibility of having a buffer occupancy lower than that of the prior run arises as a result of the fact that the difference may be absorbed in later frame intervals after the edit segment and therefore, underflow will not occur.

[0134] Beginning in step 420 of Figure 18, assume that the buffer occupancy, B_0', of the re-encoded video of the last frame of the edit segment is zero without creating underflow in the future. This is the most liberal estimate for B_0' as it is possible that the buffer occupancy must be higher to prevent buffer underflow but the buffer occupancy cannot be lower than zero. However, the process of Figure 18 will check to see if the occupancy of zero for the last frame of the re-encoded edit segment creates an underflow and if it does, will increase the buffer occupancy at the end of the edit segment until a value of B_0' is found which will not create underflow.

[0135] The process of Figure 18 evaluates the estimated buffer occupancy difference, X', between the original encoding and the re-encoding.

$$X' = B_0 - B_0' \qquad (15)$$

The difference X' is called an estimate because the difference cannot be finally determined until it is assured that X' has been checked and adjusted as necessary until all chances of underflow have been eliminated. The procedure of Figure 18 steps through one frame at a time, starting with the frame immediately following the boundary frame, i.e. i = 1. For the first frame, with B_0' equal to zero, the estimated difference is simply the value of the buffer occupancy for the first frame of the original encoding, as calculated in step 422.

$$X' = B_0 \qquad (16)$$

[0136] Next, step 424 determines the number of bits absorbed by the frames to the right of the boundary frame. The number of bits absorbed is a phrase used to refer to bits which could have been placed into the buffer by frames after the edit segment of the original encoding but in fact were not placed into the buffer during the original encoding because the buffer was full and therefore could not accept any more bits. The quantity Δt_i represents the time period of the ith frame of the original encoding after the edit segment during which the buffer was full, and due to the reduced value of B_0', can now accept additional bits. The time period Δt_i must be less than the time period of a frame as the buffer cannot be full for an entire frame and is greater than or equal to zero. The time period Δt_i is zero if the buffer was never full during the original encoding. Therefore,

$$1/P > \Delta t_i \geq 0 \qquad (17)$$

where 'P' is the picture rate measured in frames per second and is typically 30 frames per second. The total number of bits absorbed to the right of the edit segment as a result of the buffer being full is calculated to be

$$a_i = \sum_{k=1}^{i} \Delta t_k R_{max} \qquad (18)$$

where R_{max} is the fill rate of the buffer.

[0137] Step 426 determines if the procedure for determining the buffer occupancy level of the last frame of the edit segment can end. If the sum of the bits absorbed by the ith frame exceeds X', i.e. if $a_i > X'$, then the procedure is stopped because the buffer is able to be filled by later frames so buffer underflow will not occur and step 428 is performed. Alternatively, if the last frame n is reached (i=n), the estimated value of X' will not cause an underflow problem up to the end of the video.

[0138] If the answer to step 426 is "No", step 430 calculates the number of bits, b_i, remaining to be absorbed

LINKPLAY EXHIBIT 1004
Page 1636 of 4822

$$b_i = X' - a_i \qquad (19)$$

Step 432 then determines if $b_i > B_i$, where B_i is the buffer occupancy for the ith frame after the edit segment. If $b_i > B_i$, the estimated value of the difference, X', is too big and will cause underflow because more bits remain to be-absorbed than the current buffer occupancy 'B_i' allows. Therefore, among the bits that remain to be absorbed, only 'B_i' bits may be absorbed without causing underflow. Next, the estimate X' of the difference needs to be revised is step 434. In step 434, the revised estimate of X' is calculated as the bits remaining to be absorbed plus the bits absorbed

$$X' = B_i + a_i \qquad (20)$$

If step 432 determines that $b_i \leq B_i$, there is no underflow problem because the buffer occupancy at the ith frame is larger than the bits that remain to be absorbed, there is no need to change the estimate of X', and flow proceeds to step 436. In step 436, the next frame is examined (i is incremented by 1) and the loop of steps 424, 426, 430, 432, 434, and 436 is repeated until the determination in step 426 has an affirmative answer. If step 426 has an affirmative answer, the estimate X' is acceptable and therefore in step 428, the minimum allowed buffer occupancy B_0' at the boundary of the edit segment is

$$B_0' = B_0 - X' \qquad (21)$$

If the actual buffer occupancy of the re-encoded last frame of the edit segment is lower than B_0', an underflow situation will occur due to the re-allocation of bits. In this case, the last frame is marked as an underflow point in u_flow list, and the number of bits necessary to be taken from the last frame of the edit segment to prevent the underflow condition (the difference between the number of bits for the changed video and the number of bits B_0' which result in the minimum allowable buffer occupancy) are placed in u_val in a similar manner as described for step 392.

[0139] It is to be noted that for the encoding system of the present invention, it is desirable to know the buffer occupancy for all points in the video and this information is stored in the log files. However, if the buffer occupancy at the last frame of the edit segment is reduced, the buffer occupancy for frames after the edit segment could be changed, making the buffer occupancy information in the log files inaccurate and therefore, corrective action would need to be taken which re-calculates the buffer occupancy for frames after the edit segment.

[0140] Referring back to the process in Figure 17, step 396 then determines if there are any entries in u_flow list. If there are none, there is no underflow problems and the process of Figure 17 ends. If there are entries in u_flow list, flow proceeds to step 398 which marks sections to the left of each position in u_flow list up to the nearest o_flow point as being protected and bits equal to the corresponding number of bits in u_val are removed from these sections. As buffer underflow is caused by too many bits in one or more frames, reducing the number of bits in the frames will solve the underflow problem. In order not to have a sharp reduction in picture quality due to the bits being removed to prevent underflow, the bits are taken not from just the frame at the underflow point but equally from all frames back to the last point that the buffer was full. This provides a manner for obtaining the highest quality video while still preventing underflow.

[0141] These bits which are removed in step 398 are then equally distributed to the sections which are not marked protected in step 400 and the process of Figure 17 begins again. The distributing of bits in step 400 does not guarantee that underflow will no longer be a problem and for this reason, it is necessary to again check for the underflow conditions using the entire process of Figure 17 in an iterative manner.

[0142] After the process of Figure 17 is complete, step 288 of Figure 12 is complete and step 290 in Figure 12 is executed. Step 290 determines the average quantizer value for each frame using the rate-quantizer function determined in step 282. The number of desired bits has previously been computed in step 286 and checked to see if underflow problems exist in step 288 but in order to re-encode the data so that the encoded data has the desired number of bits, the quantizer value must be determined. This is simply determined using the rate-quantizer function and determining the quantizer value by looking up the rate.

[0143] To determine the quantization level on a picture-by-picture basis, a rate-quantizer function (RQF) was determined in step 282. To estimate this function, two data points for each picture are required. The RQF is:

LINKPLAY EXHIBIT 1004
Page 1637 of 4822

$$R = X*Q^{(-g)} + H \qquad (22)$$

where R is the number of bits used to code one picture, Q is the quantizer scale, X, g, and H are the picture by picture basis coding characteristic parameters. H represents the bits which are independent of the quantizer scale so H is the bits consumed by the header, the motion vectors, the DC coefficients, etc. X and g are estimated from a pair of data points corresponding to the AC coefficient bits used to code the picture for a given quantizer scale.

[0144] The first run of the encoder may be used to generate the two data points for each coded picture in the video sequence. This may be done by a pair of alternating quantizer scale values used for each macroblock in the picture. These are processed to compute the rate-quantizer model parameters 'X' and 'g'. These two parameters, along with the value of H may be stored in a file, one per frame of coded video.

[0145] In order to determine the quantizer scale which will produce the target bits assigned to a particular coded frame of video, the above equation is used with 'R' representing the target bits, and the values of (X, g, H) are read from the stored data file which may be the log file described above. The value of 'Q' that best fits this equation is the desired quantizer scale.

[0146] Then, step 292 re-encodes the changed sections of video using the quantizer values determined in step 290. It is to be noted that re-encoding segments of previously encoded video may create encoding artifact problems unless special procedures are followed during the re-encoding. Details of the re-encoding process to avoid encoding and decoding artifacts are described in a separate section below. After the video is re-encoded, it is substituted into the bitstream for the previously encoded video, and the process ends.

[0147] As described with respect to Figure 5, the user may view the video after it is re-encoded to determine if it is acceptable. If it is not acceptable, the user does not have to accept the newly encoded video and may continue using the previously encoded video. If the user does find the newly encoded video desirable, the newly encoded video will replace the previously encoded video. The user may then edit a new segment of the video, either spatially or temporally, as illustrated in Figure 5.

[0148] Figures 19A and 19B illustrate two different examples of bit rates from an original video encoding and after the user has entered a manual quality priorities. In Figures 19A and 19B, the bit rate (at a scale of 10^6 which illustrate megabits) is plotted against a Group Of Pictures Number. In the illustrated examples, one GOP contains 15 pictures and each GOP is displayed per one half second. Also, the solid lines illustrate the original encoding and the dashed lines illustrate the encoding after the user has manually adjusted the quality of the picture.

[0149] Figure 19A illustrates an encoding having five different priority regions with priorities of 2, -3, 3, 0, and protected. The time period with the priority of 3 uses more bits after the re-encoding as compared to the original encoding. As the priorities are merely relative to each other, the section marked with a priority of 2 does not have a significantly increased bit rate whereas the sections of -3 and 0 have a reduced bit rate. The protected section should have a bit rate which is similar after the re-encoding as existed before the re-encoding.

[0150] Figure 19B is a second example having regions with user selected priorities of 0, 2, 0, 4, and 0 respectively. Again, as the priorities are relative to each other a priority of 0 does not mean that the bit rate is unchanged but is simply a priority relative to the other sections having priorities of 2 and 4. In order to give the section with the priority 4 a high bit rate, the three sections with the user selected priority of 0 are reduced and the section with a priority of 2 has approximately the same bit rate before and after the encoding.

[0151] The processing for the temporal manual editing occurs in workstation 30 of Figure 1A in the preferred embodiment of the invention. However, as the temporal manual editing does not occur during the video encoding process, the workstation 10 may perform the temporal manual editing calculations without degrading the speed of the video encoding process.

[0152] It is to be noted that the present invention is described in detail as changing the picture quality of various time segments within an edit segment. Of course, it is clearly possible to extend this concept to allow the changing of the quality of sections of pictures which are not in the same time segment. For example, it is possible to apply the teachings of the present invention to increase the quality of the last 5 minutes of a movie at the expense of the first 5 minutes. If it is desired to change the quality of time periods outside of a single edit segment, as described below, it would be necessary for the user to view and check out the segments of interest sequentially, reassign the bits, and ensure that the constraints relating to the total bits of the movie, as well as the buffer occupancy at the trailing edge of each section is within the prescribed limits. However, this may be accomplished by one of ordinary skill in the art without undue experimentation based on the teachings contained herein.

[0153] The temporal manual editing has been described as keeping the number of bits for the original encoding of the edit segment to be the same as the number of bits of the re-encoding of the edit segment. However, if it is desired, the re-encoded edit segment may consume less bits or if sufficient storage capacity exists, the number of bits for the re-encoded edit segment may exceed the number of bits of the original encoding. Also, it is possible to perform the original

LINKPLAY EXHIBIT 1004
Page 1638 of 4822

encoding of the video so that it will not consume all available storage space on the final digital storage medium. Therefore, the total number of bits after re-encoding may consume any amount more or fewer than the original, for example between 0% and 20% more or less which includes 5% and 1% more or less bits.

[0154] The explanation of the temporal manual editing has been made with respect to changing the video after it has been encoded. However, the teachings contained herein may be applied to systems in which there was not an initial encoding process. Additionally, the teachings of the temporal manual editing pertaining to the use of a common bit pool may be applied to the concept of spatial manual editing for video frames, described below. Also, the manner in which the spatial manual editing operates on individual frames, as described below, may be applied to a series of frames in order to achieve temporal editing of quality.

B. Changing the Quality Within Regions of Frames

[0155] Figure 20 illustrates the general procedure for altering the quality of regions within one or more frames. After starting, step 450 performs automatic video encoding which encodes the input video into a compressed digital format such as an MPEG format or other format, as described above in the sections pertaining to the system components and general system operation. Step 452 displays the encoded video to a user and step 454 allows the user to enter commands indicating quality changes for regions within one or more frames.

[0156] Figure 21 illustrates a frame of video which is displayed on a monitor such as the video display monitor 61 illustrated in Figure 1. The frame of video in Figure 21 is illustrated as having a tree 465, a person 467, a bird 468 and two clouds 471.

[0157] In Figure 21, regions within the frame have been defined including region 466 surrounding the tree 465, region 469 surrounding the person, and region 470, overlapping region 469 and including the bird 468 and the head of the person 467. These regions were drawn using the graphical pointing device 33 illustrated in Figure 1A. Quality priorities set by the user in Figure 21 may include -5 for region 466 containing the tree, +2 for region 469 containing the person, and +4 for region 470 containing the bird and the person's head. The other regions of the frame of video in Figure 21 do not contain any user defined priorities and are therefore assigned a "Don't Care" priority. As will be explained later, the "Don't Care" regions are the regions first modified after changing the quantizer levels for the user defined regions in order to adjust the total number of bits for the encoding of the frame containing the user defined priorities to equal the number of bits resulting from the original encoding of the frame. If the user did not want the quality of regions of the frame to change, these regions would be marked by the user as having a priority of zero.

[0158] In the preferred embodiment of the invention, when defining priority regions within a frame of video, the user initially creates rectangular regions on a displayed frame of video using a graphical pointing device. The later defined regions are placed on top of and may overlap earlier defined regions. Even though the present invention is discussed with respect to defining rectangular regions, of course the teachings of the invention can be applied to any shaped region including regions having curves such as circles or ellipses, polygonal regions such as an octagon, or hexagon, or any other user defined shape including curves and/or straight lines. After the user defines each region, the user defines the priority for the region. Alternatively, the user can define the shape of all regions and thereafter, assign the priorities for these regions.

[0159] When the user initially defines priorities within a region, the regions correspond to pixel locations. However, the process of assigning priorities described below operates on a macroblock basis, a macroblock being a unit used in digital video encoding. Accordingly, if a macroblock is within or on the outer peripheral of a user defined region, that macroblock is assigned the priority of the region. One of ordinary skill in the art can simply determine the macroblocks corresponding to the user defined region using simple mathematics. For the outer peripheral of a user defined region, as an alternative to simply including a macroblock in a user defined region if the outer periphery of the region intersects the macroblock, a more sophisticated process of determining whether the macroblock should be included within the region may be performed by including the macroblock within the user defined region if more than 50% of the macroblock lies within the user defined area and excluding the macroblock from the user defined region is less than 50% of the macroblock is within the user defined region.

[0160] After the regions and their priorities are defined, the user may change the overlapping of the regions, if desired. For example, the user can "click" on region 469 of Figure 21. while holding a predetermined function key and change region 469 to be on top of the region 470 which would make the entire person have the priority of region 469 and only the bird 468 would have the priority assigned to the region 470. The overlapping regions are given a stacking priority when entered by the user. This stacking priority is adjusted as necessary when user defined regions are added, deleted, or modified.

[0161] After the regions are defined, the user also may reshape the regions as desired or move the regions using the graphical pointing device 33. Also, the regions which have been defined by the user may be deleted using the graphical pointing device 33. It is to be noted that even though the regions are being deleted, only the user defined priorities are being deleted but the original video data within the region being deleted is of course not deleted.

LINKPLAY EXHIBIT 1004
Page 1639 of 4822

[0162] As explained in the above section on changing the quality over periods of time, during or after the automatic encoding, the user may review the encoded video and whenever a frame or time period of interest occurs, the user can simply press a key on a keyboard or enter input from another device indicating that the particular frame or time period is of interest. Later, the user can go back to that frame or time period and spend more time studying the time period and altering features within that time period, as desired.

[0163] The process of a user defining regions within a frame of video has been explained above for a simple case in which there is only one frame of data being modified by a user. However, it will be quite tedious to work on frames of data in a video sequence one frame at a time and therefore, the invention allows a user to define a region at a first time period and a corresponding region in a later time period. Subsequently, an interpolation process is performed so that all frames of video intermediate to the first frame and last frame automatically have regions created which correspond to the regions defined by the user for the first and last frames, and are assigned the same priorities as the first region and/or last region or a priority based on the priorities of the first and last frames. Additionally, the stacking priorities of overlapping regions are maintained during the interpolation process. After the intermediate frames have their regions automatically created, the user may then alter the automatically created regions by changing the priorities, the shapes of the regions, add new regions, or delete the regions, as desired.

[0164] Below is an example of how the interpolation process operates. Let the user select priority regions for two frames at time instances t_1 and t_n. After the user assigns a region in the frame t_1 and a corresponding region in t_n, the coordinates of these two regions are interpolated over intermediate frames to obtain region specifications for all the frames between t_1 and t_n. Not only can the interpolation be performed for the actual region being defined but the interpolation can be performed for the quality priority as well. For the interpolation process, a simple linear function is used. Suppose a user has defined a region A of frame t_1 having a point a_1 and there is a corresponding point a_n in the corresponding region A in the frame t_n, the corresponding points in the intermediate frames a_2, a_3,..., a_(n-1) have a horizontal ordinate defined as

$$a\_i(h)=a\_1(h)+(a\_n(h)-a\_1(h)) \times (i-1)/(n-1) \qquad (23)$$

and a vertical ordinate,

$$a\_i(v)=a\_1(v)+(a\_n(v)-a\_1(v)) \times (i-1)/(n-1) \qquad (24)$$

where h and v correspond to horizontal and vertical ordinates of the point in question. In this process, only the vertices of a polygon that define the region are used and the vertices are interpolated. The regions are then defined by the vertices. Other interpolation techniques besides linear interpolation may be performed.

[0165] The above interpolation process can be performed as many times as required until all regions for all frames in the segment to be edited are properly defined. Furthermore, this process can be recursive with a fine-tuning process occurring such that a user may move or alter the size of a region determined through interpolation for an intermediate frame and the interpolation process may be executed again for the first frame and intermediate frame which has been altered, and then for the intermediate frame and the last frame. The interpolation process is primarily used for defining regions that move across frames. However, the invention also includes defining regions once which are stationary and used in a series of frames.

[0166] After the regions are defined and the user defined priorities are entered, the new quantizer values for each macroblock are estimated or determined in step 456 of Figure 20 based on the priority input by the user and the old quantizer value of the macroblock. This is determined according to the relationship illustrated in Figure 22 which illustrates the fraction of the original quantizer value used with each of the user selected priorities. For example, if the user inputs a priority of zero, the resulting fraction of the original quantizer value which is used is 1, meaning that there is no change to the original quantizer value. If the user selects a priority of -3, the fraction of the original quantizer value is 1.58, meaning that the original quantizer value for the macroblock having the priority of 3 is multiplied by 1.58 in order to determine the resulting quantizer value. As the priority -3 means reduced quality, the quantizer level must be increased to achieve a reduced quality meaning less bits are to be used for the macroblock. In an opposite manner, if a positive priority such as 3 is input by the user, the quantizer value of a macroblock from the original automatic encoding is multiplied by 0.42 which reduces the quantizer level. A reduced quantizer level means that more bits are required to represent the image within the macroblock and accordingly, the picture quality is increased. The points on the line illustrated in Figure 22 are-(-5, 100), (-4, 1.75), (-3, 1.58), (-2, 1.42), (-1, 1.25), (0, 1), (1, 0.72), (2, 0.58), (3, 0.42), (4, 0.25), and (5, 0.01). The mapping illustrated in Figure 22 is based on a piece-wise linear relationship between the q-

LINKPLAY EXHIBIT 1004
Page 1640 of 4822

levels and the priority. The points determined in Figure 22 have been determined through experimentation and different fractions may be used with the user selected priorities, as desired.

[0167] The priorities of -5 and 5 are intended to obtain the lowest and highest possible qualities, respectively. The lowest quality results from a maximum permissible quantizer level of 112 when the non-linear quantizer scale is used and 62 when the linear quantizer scale is used for MPEG-2 encoding. Also, the lowest quantizer level used to achieve the highest quality is 1. Therefore, the relationship illustrated in Figure 22 may result in values which are off the actual allowed q scale but these calculated q values are simply then placed at their saturation level. As an example, assume that initially, a macroblock has a quantizer scale of 20 and the user selects a priority for the macroblock to be -5. The quantizer value 20 multiplied by 100 yields a new quantizer level of 2,000 which is off of the quantizer scale and therefore, the maximum or saturation quantizer value is used.

[0168] In addition to using saturation levels as described above, the present invention takes a precaution for the I, P, and B frames of MPEG encoding. The present invention limits the highest quantizer level for the I-type frame to be 96 for the non-linear quantizer cases and 58 when the linear quantizer scale is used. Similarly, the highest quantizer level for the B-type and P-type frames is 112 for the non-linear quantizer scale and 62 for the linear quantizer scale cases. The linear and non-linear quantizer scales are defined in the MPEG standard. The lowest quantizer level used for each of the I, P, and B frame types is 1. Of course, the user defined inputs and the fraction of the original quantizer values illustrated in Figure 22 can be modified in order to satisfy the user's requirements.

[0169] After step 456 computes the new quantizer values for each macroblock, step 458 determines the number of bits for each frame resulting from the quantizer values determined in step 456. This step is important because in the preferred embodiment of the invention, the same number of bits for each frame must be obtained after the quality of some regions is increased while the quality of other regions are reduced. In order to determine the number of bits resulting from the new quantizer values, it is necessary to have a function which properly estimates the quantizer level versus the resulting number of bits for each macroblock. To determine the relationship between the resulting number of bits versus the quantizer level, a study of the empirical results of the video encoder, such as an MPEG-2 encoder, was performed and the determined relationship is based on actual empirical results. Simulation results for various quantizer levels and frame type using a number of video sequences were conducted. Based on these results, the following relationship between the quantizer level (q) for a macroblock and the number of bits (b) used in encoding that macroblock, depending on the frame type was determined to be

$$\log(b)=m\ \log(q)+n \qquad (25)$$

where m = -0.75 for I-type frames and m = -1.0 for B and P-type frames. Additionally, n = 15, 14.5 and 13.8 for I, P, and B frames, respectively. These values are, as noted before, averaged over a number of video sequences. The above equation serves as a good approximation measure to predict the behavior of the encoder, and the relationship between the number of bits and the quantizer level is linear in the log domain. Of course, other relationships, if accurate, between the quantizer level and number of resulting bits may be used to determine the number of bits resulting from a given quantization level, including the other rate-quantizer functions described herein.

[0170] The above process describes how to determine the number of bits resulting from a given quantization level. However, if more than one encoding is used, it is possible to further refine the data rate versus quantizer function using the information from the new encoding. During the initial automatic encoding process, there is a first video encoding pass used to gather statistics of the video to be encoded. Then during the second pass which is the actual encoding, another actual point of the data rate versus quantizer function will be obtained such as in a b versus q plot corresponding to the above equation describing the number of bits b generated from a given q-level. After the data is re-encoded according to the user specified selections, as described below, the resulting re-encoded video is either accepted or rejected by the user depending on if it satisfies the desired quality improvements. Each re-encoding generates actual data for the q-level versus bits relationship which was estimated. Given the new q-level assignment, q', a determined data rate will be more accurate for this point based on the additional data obtained by the re-encoding. If the new quantizer level, q', falls between any two points which have been previously obtained, it is possible to do a linear interpolation in the log domain to obtain an estimate for the number of bits to be outputted by the encoder with the new q-level. If the new quantizer level, q', does not fall in between the two determined q-levels, the model can be used to extrapolate from the closest q-level to the new quantizer level, q', to estimate the number of bits to be generated at the encoding time. Note that the more times the data is re-encoded, the higher the probability of obtaining a better estimate will be for the output bits from the encoder.

[0171] After the number of bits for each frame resulting from the estimated quantizer values is determined in step 458 of Figure 20, step 460 determines if the estimated number of bits is close enough to the original number of bits for the frame and corrects the number of bits, if necessary. This is performed according to the process illustrated in Figures

LINKPLAY EXHIBIT 1004
Page 1641 of 4822

23-25C.

[0172] When step 460 is performed, step 472 of Figure 23 first determines the difference between the estimated number of bits and the number of bits of the original encoding. This is calculated according to the equation

$$D=B'-B \qquad (26)$$

where B' is the estimated number of bits of the modified frame, B is the number of bits of the frame resulting from the original encoding, and D is the difference. It is of course possible to use a value for B which is other than the number of bits resulting from the original encoding. For example, if a user wants to increase the total number of bits for a frame, he may set B to a number which is a predetermined amount higher than a number of bits resulting from the original encoding. Similarly, the user may set B to be lower than the original number of bits of the frame. Therefore, the re-encoded frame may consume more or less bits than the original encoding, depending on storage constraints, minimum quality constraints, and the possibility of video buffer underflow. Also, if regions in a frame have not had their quantizer values altered, it is possible to analyze only the numbers of bits resulting from regions having their quantizer values altered and ignoring the regions which have not been changed. Step 474 then analyzes this difference and if there are determined to be too many bits, flow proceeds to B for the process illustrated in Figures 24A-24C in order to increase the quantization level of the macroblocks to reduce the number of bits. If step 474 determines that there are too few bits in the modified frame, the quantization level of the macroblocks of the new frame is to be decreased in order to generate more bits according to process F illustrated in Figures 25A-25C. Otherwise, if the difference is within an acceptable threshold, there is no need to modify the quantization levels of the macroblocks and the process returns to perform step 462 in Figure 20. When keeping the number of bits in the frame the same for the re-encoded video, the bits can be considered to be shifted from one region to another.

[0173] In step 474, the difference D is analyzed by comparing the difference to a threshold, T. The threshold, is, set to 0.001 x B which means that if the number of bits resulting from the change in quality of the region is within 0.001 of the total number of bits, then the difference is acceptable and the process may stop. Of course, the stopping threshold which indicates that the difference is acceptable may be determined in alternative manners, may be increased or reduced, as desired, or may be based on both the original number of bits for the frame and the resulting number of bits resulting from the user selected priorities.

[0174] The processes illustrated in Figures 24A-24C, and Figures 25A-25C are quite similar except that Figures 24A-24B are used to reduce the number of bits resulting from the user selected qualities and Figures 25A-25C are used to increase the number of bits resulting from the user selected qualities. A general overview of the processes performed in Figures 24A-25C is now described before explaining the details of these processes. Initially, the macroblocks in a frame of video are labelled as one of four types; "Don't Care", "Negative Priority", "Positive Priority", and "Don't Change". When a region has not been assigned a priority by a user, or if the system allows the user to assign a "Don't Care" status, that region is considered to be of the "Don't Care" type. If the user assigns a priority between and including -5 and -1, the macroblocks within these regions are considered to be "Negative Priority". If a region is assigned a priority between and including 1 and 5, those regions are considered to have a "Positive Priority". Last, all regions assigned with a priority of 0, or if the system allows the user in some other manner to indicate that a region is not to have its quality changed, the macroblocks in those regions are considered to be of the "Don't Change" type.

[0175] Each of the macroblocks within each of the four macroblock types is labelled with an index, i, such that each of the four types has its own set of indexes. The index i of zero is assigned to the first macroblock and the index of N-1 is assigned to the last macroblock where N is the total number of macroblocks within a region type. For example, for all of the "Don't Care" regions, each of the macroblocks within these regions are indexed from 0 to N-1. Note that if there are more than two different "Don't Care" regions, there is not two different indices i=0 for the "Don't Change" regions. For all "Don't Care" regions, there is only one macroblock with an index 0, one macroblock with an index 1, etc.

[0176] If there are too many bits in the frame after the user has determined the priority of the regions, quantizer levels for the frame must be increased in order to reduce the number of bits for the frame. The original quantizer value for the macroblock is taken from the macroblock layer log file having a format as illustrated in either of Figures 10A or 10B. Alternatively, the encoder may be re-run to determine the quantizer values if they were not stored after the original run of the encoder. Yet another alternative is to decode the encoded bitstream to determine the quantizer values of the original encoding. The way the procedure works for decreasing the number of bits is to increase the quantizer level for each macroblock of the "Don't Care" type by one quantizer level, one macroblock at a time. If this does not solve the problem, each macroblock of the "Don't Care" has its q level again increased by one, one macroblock at a time. A predetermined maximum quantizer level is set and the quantizer levels of no macroblocks can be increased beyond this maximum level. If all macroblocks of the "Don't Care" type reach this maximum level, then the "Negative Priority" and the "Positive Priority" regions are increased by one in a similar manner. First, all negative priority macroblocks are

LINKPLAY EXHIBIT 1004
Page 1642 of 4822

increased by one quantizer level and if this does not solve the bit differential problem, then the macroblocks of the "Positive Priority" macroblocks have their quantizer level increased by one. If this does not solve the problem, then the quantizer levels of the "Negative Priority" macroblocks are again increased by one and if this does not solve the problem, then the macroblocks of the "Positive Priority" type have their quantizer level increased by one. This process continues until all macroblocks in the "Negative Priority" and the "Positive Priority" types have their quantizer levels increased to the maximum level described above. If this does not solve the bit differential problem, then the macroblocks of the "Don't Change" type have their quantizer levels increased by one quantization level at a time until the bit differential problem is solved. In all likelihood, the bit differential problem will have been corrected before the quantization levels of the "Don't Change" type begin to be changed. A similar process occurs when the frame having the user selected priorities has too few bits and the quantization level of the macroblocks needs to be decreased in order to increase the number of bits according to the process of Figures 25A-25C.

[0177] Turning now to the flowchart illustrated in Figure 24A which is executed by Figure 23 when too many bits result, the process begins in Figure 24A by step 480 setting the block type to the "Don't Care" type. Step 482 then sets the macroblocks index to the starting point which means that the index, i, is set equal to zero. Step 486 is then performed which determines if the quantizer level of the macroblock in question (the macroblock within index=i) having the macroblock type which has been set (the first time through the flowchart, the first macroblock of the "Don't Care" type) has the maximum predefined quantizer level. The maximum quantizer level for this correction process should not be set too high so that there is a great decrease in quality for the macroblocks which are corrected. It is desirable to attempt to decrease the quality in order to decrease the number of bits in a uniform manner so that there is not a great differential in quality of the macroblocks, if possible. Therefore, this maximum is set at a quantization level of 62 when the linear quantization scale is used and 112 when the non-linear quantization scale is used for MPEG-2 encoding, for example. However, other values may be used for the maximum quantizer value. If the quantizer level of the macroblock in question is determined not to be the maximum in step 486, step 488 increments the quantizer level of the macroblock in question by one quantization level. Step 490 then recalculates the bit estimate for the frame using the bit rate-quantizer function described above. Step 490 then increments the index of the macroblocks. If the index of the macroblock indicates that the last macroblock of the macroblock type has not been processed, flow loops back to step 484. Otherwise, flow proceeds from step 492 to step 498 which resets the block index to zero, indicating the first macroblock.

[0178] When flow returns to step 484, the bit estimate recalculated in step 490 is used to recalculate the difference D, described above which is compared to a threshold, as described above. If the bit estimate is no longer too high, the number of bits for the frame containing the user selected priorities has been corrected and flow returns to the calling process. Otherwise, flow proceeds to step 486 which again determines if the quantizer value of the macroblock in question (having index i) is at a maximum.

[0179] If step 486 determines that the quantizer value of the macroblock in question is at a maximum, the quantizer level of that block is not to be increased and step 494 then increments the block index i by one. If step 496 determines that the index of the macroblock type is not past the last block, flow proceeds back to step 486 to determine if the block with the incremented index has a maximum quantizer value. Otherwise, if step 496 determines that the macroblock is the last macroblock of the macroblock type in question, step 498 is executed which resets the block index to the first macroblock (i=0). Flow then proceeds to process C illustrated in Figure 24B.

[0180] In Figure 24B, step 500 examines if the block type is of the "Don't Care" type. If it is the "Don't Care" type, step 502 determines if all "Don't Care" macroblocks have the maximum q level. If all of the "Don't Care" macroblocks do have the maximum q level, no further adjustment can be performed on the "Don't Care" type macroblocks and therefore the macroblock type is changed to "Negative Priority" in step 504 and flow proceeds back to process D illustrated in Figure 24A and step 484 is executed. Otherwise, if step 502 determines that all macroblocks of the "Don't Care" type do not have the maximum q level, flow proceeds back to process D without changing the type of macroblock being modified. This procedure continues until the resulting number of bits for the frame is within the threshold or all macroblocks of the "Don't Care" type are set to the maximum quantization level, which results in the block type being changed to "Negative Priority".

[0181] If step 500 determines that the block type is not "Don't Care", step 506 determines if the block type is "Negative Priority". If the block type is "Negative Priority", step 508 determines if all blocks of the "Positive Priority" have a maximum q level. If the result of the determination in step 508 is negative, step 510 is performed to set the block type to "Positive Priority" and flow proceeds to process D illustrated in Figure 24A.

[0182] If step 508 determines that all macroblocks of the "Positive Priority" have the maximum q level, step 512 examines if all blocks of the "Negative Priority" type have a maximum q level. If they do not, the block type is not changed and flow proceeds back to process D in Figure 24A. If all macroblocks of the "Negative Priority" type are determined to have the maximum q level in step 512, all "Don't Care", "Negative Priority", and "Positive Priority" macroblocks have the maximum quantization level and the block type is then set to "Don't Change" in step 514. Flow then proceeds to process D in Figure 24A.

[0183] Process E and step 520 of Figure 24C is performed after step 506 of Figure 24B has a negative determination.

LINKPLAY EXHIBIT 1004
Page 1643 of 4822

Step 520 determines if the block type is "Positive Priority". If it does, step 522 determines if all blocks of the "Negative Priority" type have a maximum Q level. If they do not, the block type is set to "Negative Priority" and flow returns to Process D illustrated in Figure 24A. If step 522 determines that all blocks of the "Negative Priority" type do have the maximum q level, step 526 examines if all blocks of the "Positive Priority" type have the maximum q level. If they do not, flow returns to process D illustrated in Figure 24A. Otherwise, all "Negative Priority" and "Positive Priority" along with all "Don't Care" blocks have the maximum quantization level and step 528 sets the block type to "Don't Change" and flow returns to process D illustrated in Figure 24A.

[0184] If step 520 determines that the block type is not "Positive Priority", the block type must therefore be "Don't Change" and step 530 determines if all blocks of the "Don't Change" type have the maximum q level. If they do, there is an error as all block types have been set to the maximum quantization level without resolving the problem of too many bits being in the frame having the user selected priority. Setting all blocks to the maximum predefined q level should result in a number of bits which does not exceed the original encoding of the frame. If the determination in step 530 is negative, flow proceeds back to process D illustrated in Figure 24A.

[0185] If the process of Figure 23 determines that there are too few bits in the frame having the user selected priority regions, the process in Figures 25A-25C is performed. Figures 25A-25C are based on the process illustrated in Figures 24A-24C except that the quantization levels are decreased instead of increased in order to increase the number of bits for the frame and the quantization levels of the macroblocks cannot exceed a predetermined minimum level, such as a quantizer level of 1, when decreased. As the process of Figures 25A-25C is identical to the process of Figures 24A-24C, except for the above noted differences, a further description of Figures 25A-25C will not be made for the sake of brevity.

[0186] The correction process described in Figures 23-25C is one manner in which the invention may operate. However, alternative procedures of adjusting the quantization levels of the different types of macroblocks is possible. For example, instead of changing the quantization levels of the "Negative Priority" and the "Positive Priority" macroblocks at the same time as described above, the macroblocks of the "Negative Priority", "Positive Priority", and "Don't Change" types can be uniformly increased. Alternatively, if there are too many bits, the quantizer levels of the "Negative Priority" types may be increased until the maximum quantization level is reached and then the quantization levels for the "Positive Priority" type macroblocks are then increased. In an opposite manner, if the number of bits of the frame having the user selected priorities is too low, the "Positive Priority" type macroblocks may have their quantization levels decreased in order to increase the number of bits before the quantization levels of the "Negative Priority" type macroblocks are decreased. In the latter two cases, the altering of the "Don't Care" type macroblocks may occur before, during, or after the "Negative Priority" or "Positive Priority" type macroblocks are altered.

[0187] After the process of Figures 23A-25C is performed, flow returns to step 462 in Figure 20 which re-encodes the video using the new quantizer values. This re-encoding is performed while fine-tuning the quantization values using a rate-control function in order to have the resulting number of bits for the re-encoded frames be as close to the number of bits for the original encoding of the frames as close as possible. The rate-controller process is described in detail in a separate section below. Also, during the re-encoding process, encoding artifacts may result and therefore, special procedures may be performed in order to reduce these encoding artifacts, as described in a separate section below.

[0188] The process for the spatial manual editing occurs in workstation 30 of Figure 1A of the preferred embodiment of the invention. However, as the spatial manual editing does not occur at the same time as the video encoding process, the workstation 10 may perform the temporal manual editing calculations without degrading the speed of the video encoding process. It is to be noted that the present invention is described in detail as changing the picture quality of various regions within a frame. Of course, it is possible to extend the concept in the present application to shifting bits within a region of one frame to a region in a different frame. Additionally, the spatial manual editing may be applied in conjunction with the temporal manual editing described above.

[0189] The spatial manual editing has been described as keeping the number of bits of the frame having a user supplied regional priorities the same as the original encoding of the frame. However, if it is desired, the re-encoded edit segment may consume a predetermined number of bits which is more or less than the original number of bits for the frame, if sufficient storage capacity exists. In these cases, it will be desirable to know the number of bits by which the altered frame can be changed and therefore, the processes illustrated in Figures 24A-25C will operate until the set difference is achieved. Accordingly, the total number of bits for the re-encoding of any frame may consume any amount more or less than the original encoding, for example between 0% and 20% more or less which includes both 5% and 1% more or less bits.

[0190] The explanation of the spatial manual editing has been made with respect to changing the video after it has been encoded. However, the teachings contained herein may be applied to systems in which there was not an initial encoding process.

[0191] The manner in which the spatial manual editing works is to assign a quantizer value based on the user selected priority and then adjust the quantization levels of the macroblocks of a frame, as necessary, in order to achieve the same number of bits for the frame as previously obtained. In contrast, the temporal manual editing described in the section

LINKPLAY EXHIBIT 1004
Page 1644 of 4822

above removes a common pool of bits from sections of the video and redistributes those bits from the common pool in accordance with the user selected priorities. All teachings relating to the process in which the spatial manual editing is performed with respect to one frame may be applied to the concept of changing the quality of frames of video over time, as described above in the temporal manual editing section, and vice versa. For example, the spatial editing has been described as modifying quantizer levels to achieve the changes in quality whereas bit numbers are adjusted in the temporal editing section. As the number of bits for a frame or macroblock directly depends on quantization levels, the modification of bit numbers and quantization levels achieves the same results and therefore the temporal and spatial editing may modify the number of bits and/or the quantization levels.

VIII. RE-ENCODING THE VIDEO AFTER CHANGING THE QUALITY

A. Bit Allocation Control When the Quality Has Been Changed Within Single Frames

[0192] To control the bit-rate in the video encoding device 50 illustrated in Figure 1A, a controller, such as workstation 10 or any specialized hardware that can perform the same controlling function, loads quantization levels, or q-levels, and an estimated or expected number of bits per slice into a rate-controlled encoder. The encoder then begins encoding, and quantization levels are adjusted after measuring the accuracy of the generated number of bits compared to the previously determined estimated number of bits. The estimated number of bits are determined using a rate-quantizer function, as described above in the sections pertaining to the spatial and temporal manual editing. This process is performed sequentially on a set of n macroblocks at a time, where each set is designated MB(i), and where:

$$1 \leq i \leq (\text{total \# of macroblocks} / n). \qquad (27)$$

When re-encoding the last set of macroblocks, the rate-controller assures that the total number of generated bits in the re-encoding process is less than the originally encoded length since any minor overestimation in the number of bits that will be generated can be absorbed using bit-stuffing.

[0193] Figure 26 is a flowchart illustrating a method for re-encoding a set of digital video frames with pre-assigned quantization levels. Adjustments are made to the pre-assigned quantization levels based on how accurately a quantizer model estimated the number of bits that would be generated by re-encoding versus the actual number of generated bits. In Fig. 26, the rate-controlled coding process starts at step 700 and then designates, by setting i=1 in step 702, the first set of n macroblocks as the current set of macroblocks. Step 704 loads the complete set of estimated number of bits, EB(i), that will be used in re-encoding each set of macroblocks, MB(i). MB(i) is then re-encoded in step 706, and the number of actually generated bits is stored as GB(i). The percentage difference between the generated bits, GB(i), and estimated bits, EB(i), is calculated in step 708 as an estimation ratio, PB(i) = GB(i)/EB(i). Additionally, the system can calculate the cumulative estimated number of bits, EB'(i), and the cumulative generated number of bits, GB'(i), for all re-encoded sets of macroblocks, where EB'(1) = EB(1), GB'(1) = GB(1), EB'(i) = EB'(i-1) + EB(i) and GB'(i) = GB'(i-1) + GB(i), for i≥2. Therefore, as an alternative to the illustrated step 708, PB(i) can be calculated using the cumulative estimated number of bits, EB'(i), and the cumulative generated number of bits, GB'(i), by using PB(i) = GB'(i)/EB'(i). This provides a more accurate indication of how the re-encoding process as a whole is meeting the target number of bits, instead of an indication of how accurately a particular set of macroblocks was encoded. In step 710, the estimation ratio is used to determine a correction factor for the q-levels in the next set of macroblocks, MB(i+1).

[0194] This correction factor is determined using one of two methods. The first correction factor determination method steps through a single sorted list of ratios in a table of ratio values, called a ratio lookup table, and determines between which two ratios the estimation ratio falls. The index, j, of the larger of the two numbers in the table is used as a correction factor index. The correction factor index is used to look up a correction factor, $\Delta(i)$, in a correction factor table by choosing the jth element in the table as the correction factor, and the next set of macroblocks, MB(i+1), are adjusted by $\Delta(i)$. The ratio lookup table, RT, contains, for example, the sorted values:

$$RT = \{ 0.85, 0.90, 0.95, 1.0, 1.05, 1.10, 1.15, 1.2, 1.5\}$$

and the correction factor table, CT, contains the values:

LINKPLAY EXHIBIT 1004
Page 1645 of 4822

$$CT = \{-3, -2, -1, 0, 1, 2, 3, 4, 9\}.$$

[0195] Figure 30 is a flowchart which illustrates a process for determining a correction factor index from a ratio lookup table using an estimation ratio. Assume, for purposes of illustration only, that PB(i) = 0.87. The index, j, is initially set to 1 in step 722. The process of comparing PB(i) to RT(j) is begun, and in step 724, PB(i)=0.87 is compared to RT(1) = 0.85, as defined in the above lookup table. Since 0.87 is greater than 0.85, the process continues to step 726. Since in step 726 it is determined that there are remaining indices, control is passed to step 728 and j is incremented by one. Returning to step 724, PB(i) = 0.87 is now less than RT(2) = 0.90, so control is passed to step 730 which ends the search for the correction factor index.

[0196] Figure 31 is a flowchart which illustrates how to look up a value in a lookup table given an index into the table. In Figure 31, to determine the correction factor according to the first method, the index j, which is 2 is used to look up the corresponding entry in CT. Reading from the second position in CT, the correction factor is determined to be -2 in step 736. Therefore, the q-levels of MB(i+1) are changed by -2. Likewise, if PB(i) = 1.12, then index 7 corresponds to the first entry in RT that is larger than PB(i), so the correction factor in CT at the seventh position is 3. Thus the q-levels of MB(i+1) are changed by 3, e.g., by adding 3 to the q-level of MB(i+1).

[0197] The second correction factor determination method uses a similar process to the first method, except the percentage of sets of macroblocks that have been re-encoded is used as a second parameter in determining the correction factor. Additionally, instead of reading a correction factor from a table, the correction factor is read from a matrix. By using the percentage of sets of macroblocks that have been re-encoded, or the percentage of sets of macroblocks unre-encoded, a more gradual correction can be made at the beginning of a series of frames and more significant changes can be made at the end, if necessary. This allows the system to correct more accurately for variations between the quantizer model and the actual number of generated bits. By using an inverse relation between the number of sets of macroblocks which remain to be re-encoded and the significance of correction, and assuming that an overestimation in one set of macroblocks will be offset by an underestimation in another set of macroblocks, unnecessary corrections are avoided.

[0198] As an example of a correction factor matrix, CTM, of the second correction factor method, assume CTM is defined as below:

$$CTM = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 & 1 & 2 & 4 \\ -2 & -1 & 0 & 0 & 0 & 1 & 1 & 2 & 5 \\ & & & & \vdots & & & & \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 & 8 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 & 9 \end{bmatrix}$$

If at the beginning of a series of frames, the estimation ratio indicates an overestimation of the number of bits necessary, there is no need to immediately reallocate all the unused bits to the next slice. If the unused bits were immediately consumed by MB(i+1), then an MB(k), where $k \geq i+1$, which consumed more bits than estimated, would force MB(k+1) to a higher quantization level unnecessarily. The unused bits could have been carried forward in the re-encoding process from MB(i) until needed and then used by MB(k).

[0199] Figure 32 is a flowchart which illustrates how a correction factor is calculated using both an estimation ratio and the number of remaining unre-encoded sets of macroblocks. For a case of overestimation of the number of bits required, i.e., having previously determined that the correction factor, j, equals 2 for PB(i)=0.87, step 744 of Figure 32 determines that the overestimation occurs in the first percent of the re-encoding process. The element at (2,1) of CTM is -1 and is therefore chosen as the correction factor. This dampens the correction effect early in the re-encoding process, which in the first method would have been -2. However, if ninety-nine percent of the re-encoding process has been performed and the estimation ratio is PB(i) = 1.12, element (99, 7) which is 3 would be chosen, thus fully correcting at the end of the re-encoding process.

[0200] Referring back to Figures 27A - 27C, a frame with pre-assigned q-levels is illustrated as a further example where the number of macroblocks in a set is equal to the number of macroblocks in a single slice. EB(i), the estimated number of bits, is calculated as described above and loaded into a memory used in the rate control process for MB(i), macroblock i, of Figure 27A and then MB(i) is actually re-encoded. The number of generated bits, GB(i), is determined,

LINKPLAY EXHIBIT 1004
Page 1646 of 4822

the estimation ratio is calculated, and the correction factor Δ(i) is looked up. As is shown in Figure 27B, the macroblocks of MB(i+1) are then corrected by Δ(i) and the process is repeated for MB(i+1). Figure 27C shows the change in MB(i+2) after the length of MB(i+1) has been estimated and calculated and its estimation ratio determined. This process is continued until all macroblocks and partial macroblocks have been re-encoded and corrected.

[0201] A further example is given in reference to Figures 27A and 27D. In this example, the number of macroblocks per set is the number of macroblocks in two slices. After calculating EB(i), GB(i) and PB(i), the macroblocks of MB(i+1), which comprise two slices, are corrected by Δ(i), as illustrated in Figure 27D. This re-encoding and correction process is continued as in the previous example until all macroblocks have been re-encoded.

[0202] The changes caused by the re-encoding process are illustrated in Figures 28A and 28B. Two equal size regions, Region A and Region B, of Frame N are initially encoded using a nearly equal number of bits. The user designates that the quality of Region A is to be decreased and the quality of Region B is to be increased. However, it is desired to have the number of bits used for the original encoding be as close to the number of bits of the re-encoding as possible. After re-encoding Frame N, Frame N contains the same number of bytes as were generated from the original encoding, but Region A is coded with fewer bits and Region B uses the additional bits. This decreases the quality of Region A and increases the quality of Region B as requested by the user. The rate control process of the present invention makes the re-encoded number of bits of the frame be as close to the original number of bits as possible by correcting for minor errors in the estimated number of bits determined using a rate-quantizer function.

[0203] Figure 29 illustrates changes to a digital video stream by reallocating bits from a plurality of frames and giving them to a second plurality of frames. Although individual frame sizes change, the total size of the set of frames should remain the same. The rate controller allows the number of bits for the re-encoding of the plurality of frames to be finely tuned to be the same as the number of bits from the original encoding, even if the rate-quantizer function used to determine the estimated number of bits is slightly in error.

B. Re-Encoding While Avoiding Decoding Artifacts At Edit Points

1. Restoring the Encoder to the State Used When Originally Encoding

[0204] The above-described temporal and spatial quality editing operations allow the quality of video to be changed by re-encoding the video with new quantizer values. However, simply substituting the re-encoded video into the original video may result in unacceptable decoding artifacts which appear as a visible glitch at the edit point of the video. These visible glitches will occur even if the new segment being substituted has the proper syntax such as MPEG-2 syntax and the buffer constraints across the edit boundary are satisfied. The problem results from the non-availability of the last reference frame (the last P-frame) preceding the edit segment, and a different reference frame (the last P-frame) at the end of the edit segment. This causes a disparity between the encoding and decoding of the first set of B-frames at the start of the edit segment, and in the first set of B-frames immediately following the edit segment.

[0205] In order to properly perform the edit of the digitally encoded and compressed video, it is first necessary to determine the proper positions in the encoded bitstream which correspond to the desired edit points. This process has been explained above in section VI which describes utilities to access the encoded video. As explained above, this is performed by summing up the number of bits used for each previously encoded frame in order to determine the bit offset for the picture in question. Alternatively, a directory could be used to maintain information indicating the exact location of a specific frame or time period in the encoded bitstream.

[0206] Turning now to Figures 33A-33E, there are illustrated time periods of video which are used as examples for explaining the operation of the invention. In Figure 33A, an original section of encoded video, O, is illustrated. It is desired to replace the encoded time period of 5m 0s to 10m 0s (five minutes, zero seconds to ten minutes, zero seconds) with a different section of video. Figure 33B illustrates a new section of unencoded video, N, which is to be substituted into the encoded video of Figure 33A. The dashed lines in Figures 33B and 33C are used to denote unencoded video and the solid lines in Figures 33A, 33D and 33E are used to denote the encoded video.

[0207] If only the section of new video illustrated in Figure 33B was encoded using a normal MPEG-2 mode in which a GOP referenced a previous GOP, and substituted into the video illustrated in Figure 33A, decoding artifacts would exist at the edit points 5m 0s and 10m 0s due to the lack of an accurate reference frame necessary for the bidirectional decoding used in MPEG-2 video compression. This problem exists at both the five minute and the ten minute points which are handled differently by the invention, due to the fact that the beginning of a GOP refers back to a previous GOP but the end of the GOP does not refer forward to the next GOP.

[0208] The manner in which decoding artifacts are prevented at the beginning of a GOP is to re-create the last P-frame of the preceding GOP which is used as a reference frame for the first two B-frames of the GOP. This is accomplished by placing, for example, one GOP of the original video at the beginning of the video to be inserted and this added video is illustrated from 4m 58.5s to 4m 59s in Figure 1. In the examples, one GOP is equal to one half of a second, although other time periods for the GOPs are possible. The encoder is re-run for this section with the same quantizer values used

LINKPLAY EXHIBIT 1004
Page 1647 of 4822

for the original encoding to create the last P-frame of the GOP immediately before 4m 59s. All information resulting from the re-encoding of this short section of video, except for the last P-frame may be discarded.

[0209] In order to prevent decoding artifacts at the end of the edit segment, a one second period (e.g. from 10m 0s to 10m ls in Figure 33C) of the original video is added to the end of the new video to be substituted and this one second segment is encoded using the same quantizer values as were used during the original encoding. If this one second period were not added to the end of the video, the two B-pictures immediately beyond the 10m 0s mark, assuming that 10m 0s is the dividing point between two GOPs, would refer back to a reference picture which is different after the edit than before the edit. While the examples have been described as adding one half second or one second periods of video to new the section, the concept can be generalized to adding GOPs.

[0210] Even though the decoding artifacts will no longer appear at the 10 minute mark, as the edit point at the end is now the 10m is point, one must consider if decoding artifacts will result at 10m 1s. Visible decoding artifacts will not result because the two B-frames immediately after 10m 1s mark will refer back to essentially the same reference frame to which they previously referred. Even though the two B-frames immediately after the 10m 0s mark will not refer back to the same reference frame to which they previously referred, the two B-frames were re-encoded and therefore properly refer back to a new reference frame. Accordingly, no visible decoding artifacts will be present at the 10m 0s or the 10m is points.

[0211] The two GOPs (one second) added to the end in Figure 33 should be sufficient time for the last P-frame to be nearly equivalent to the original encoding. If it could be guaranteed that the exact same quantizer values are used when re-encoding the original video after the 10 minute point, one GOP (one half second) should be sufficient to appropriately generate a correctly encoded reference picture. However, two GOPs are preferred to assure proper quality. Even though it is desired to use the same quantizer values as originally used, in practice there may be minor deviations from the original quantizer values and therefore, two GOPs are preferred.

[0212] In addition to original periods of the video being used from 4m 58.5s to 4m 59s and 10m 0s to 10m 1s, Figure 33C illustrates additional original video from 4m 59s to 5m 0s. This video is not essential to preventing decoding artifacts and therefore, the one half second to determine the last P-frame could be used immediately before the 5m 0s mark (e.g. from 4m 59.5s to 5m 0s). However, the 4m 59s to 5m 0s period is added to the new video to eliminate any chance of perception by the operator that the manual changing of quality has had an effect beyond the edit boundary. This mis-perception is due to the fact that the coding order for frames at the beginning of an MPEG GOP is IBB but the frames are displayed as BBI. The one second period is encoded using the same quantizer values as were used during the original encoding of that time period. The manner of obtaining the previously used quantizer values is explained below. If the one second period was not added from 4m 59s to 5m 0s, the substitute encoded video illustrated in Figure 33D would begin at 5m 0s.

[0213] The useful data resulting from the encoding of the video in Figure 33C is illustrated in Figure 33D as the substitute data, S, which runs from 4m 59s through 10m 1s. This substitute data, S, is then substituted into the original encoded data illustrated in Figure 33A which results in the final bitstream as illustrated in Figure 33E.

[0214] The unencoded and encoded information illustrated in Figure 33A is video in the preferred embodiment, but can be audio, or other types of information which use previous and/or subsequent information to decode the encoded information.

[0215] The new section of video in Figure 33B is obtained from any video source and may be video generated by the temporal or spatial editing process described above. The original video in Figure 33C from 4m 59s to 5m 0s and 10m 0s through 10m 1s is the original video having the same quantizer values used for the original encoding. This video at the ends of the new video N in Figure 33C is simply obtained using the temporal editing process and setting the quality of the end sections to protected as described in the temporal manual editing section, meaning that the quantization values are to remain the same. Alternatively, the video in Figure 33B may be completed unrelated to the video in Figure 33A and may have a different length. Additionally, it is possible to use the invention to cut out a section of video and not add new video. In this case, no video would exist in Figure 33B and the 'N' segment and its corresponding encoding would be absent from Figures 33C-33E. As long as the last P-frame before the edit segment is properly constructed and one half to one second (one or two GOPs) of the original video is added to the end of the new video to be encoded and encoded with the same quantizer scales used in the original encoding, no visible decoding artifacts will occur in any situation.

[0216] Figure 34 is a flowchart illustrating the process described with respect to Figures 33A-33E. After starting, step 780 encodes the video into the compressed format to generate the encoded video illustrated in Figure 33A. Step 782 determines the new section of video which is in an unencoded format which is to be substituted into the encoded video. This new section of video is illustrated as N in Figures 33B and 33C and may represent scenes related to, unrelated to, or have a different time period than the original frame. The start and end point of the video to be substituted are 5m 0s and 10m 0s, respectively. However, the actual start and end points of the final encoding to be substituted, may be different, as illustrated in Figure 33D.

[0217] Step 784 adds periods of unencoded video to the ends of the encoded new sections of video for the reasons

LINKPLAY EXHIBIT 1004
Page 1648 of 4822

described above in order to prevent decoding artifacts at the edit points 5m 0s and 10m 0s. These added periods are from 4m 59s through 5m 0s and 10m 0s through 10m 1s, although the period from 4m 59s to 5m 0s is not essential.

[0218] Step 786 determines the P-frame immediately before the edit point. As the video being substituted in Figure 33E is from 4m 50s to 10m 1s, the last P-frame immediately before 4m 50s needs to be determined. This P-frame is determined by running the encoder for the time period from 4m 58.5s to 4m 50s with the quantizer values used to originally encode the P-frame. The quantizer values are determined by retrieving the original quantizer values from the macroblock log file having the format illustrated in either Figure 10A or 10B. However, the quantizer values for each macroblock of an entire movie consume a large amount of storage space and it may not be desirable to store the quantizer values. As an alternative, the quantizer values can simply be created using are the rate control status pertaining to the video buffers for the I, P, and B frames, the target bits for the picture shown in the picture layer log file of Figure 9B as allocated bit, the activity level of the picture, stored as mean_activity in Figure 9C and the activity level of the macroblocks which are computed by the encoder during encoding. The rate control status is stored in the picture layer log file as illustrated at the bottom of Figure 9C as S1_i, S2_i, S1_p, S2_p, S1_b, and S2_b which stores the rate control status information for the I, P, and B video buffers respectively. One of ordinary skill in the video encoding art, based on the teachings contained herein, could determine the quantizer values for the desired P-frame without undue experimentation.

[0219] As an alternative manner of determining the quantizer values of the reference frame, and/or the encoded reference frame itself, the encoded version of the reference frame in the encoded reference stream can be decoded in a known manner. Further, any other method of obtaining the needed reference frame can be performed.

[0220] After the reference P-frame is obtained, the video is encoded in step 788 while maintaining the same quantizer levels as the original encodings of the end sections. The quantizer levels are obtained as described above. The resulting encoded video is illustrated in Figure 33D.

[0221] Step 790 then substitutes the newly encoded video into the originally encoded video and the final product, as illustrated in Figure 33E is obtained. This encoded video containing substituted encoded video has minimal decoding artifacts at the edit points.

2. Re-Encoding in a Closed Group of Pictures Mode

[0222] As an alternative to the procedure described in Figures 33A-34, it is possible to perform a different procedure for the encoding of the beginning of the new video while still reducing decoding artifacts. In this procedure, the manner of treating the trailing end of the video is the same as described above with respect to Figures 33A-34.

[0223] As an example of this alternative embodiment, use Figure 33A as the original encoded video and Figure 33B as the unencoded video to be substituted into Figure 33A. No original unencoded video is added at the beginning edge of Figure 33B, and as described for Figure 33C, two GOPs (one second) of corresponding original unencoded video are added to the end of Figure 33B and the video to be encoded is as illustrated in Figure 35A.

[0224] If the encoder encodes the video of Figure 35A in the closed GOP mode for the first GOP of the video while increasing the bits for the B-frames without a second reference picture, the visible decoding artifacts at the start edit point will be eliminated. The encoding of the end portion using the same quantizer levels as used for the original encoding from 10m 0s to 10m 1s is the same as described above with respect to Figures 33A-34, and the resulting encoded video is illustrated in Figure 35B. The closed GOP mode is described in the MPEG-2 standard at section 6.3.8. The closed Group Of Picture mode flag is set to 1 which indicates that encoded B-frames have been encoded using only backward prediction and the reference frame of the previous GOP is not used. However, as one of the reference frames normally used for the bidirectional frames is not available, the B-frames will have reduced quality, unless a procedure is performed to eliminate the reduced quality.

[0225] In order to solve the problem of reduced quality of B-frames encoded in the closed GOP mode, the number of bits for the B-frames having only one direction of prediction is increased so that the quality of the frame is not substantially reduced. As the B-frame has only one direction of prediction, the number of bits is preferably increased to approximately the number of bits used for ordinary P-frames. However, the number of bits may vary, depending on the desired picture quality.

[0226] The encoded video of Figure 35B can be directly substituted into the previously encoded video, as illustrated in Figure 35C.

[0227] A flowchart describing the process of using the closed GOP mode is illustrated in Figure 36. In Figure 36, the video is encoded into the compressed format in step 800 and the new section of video which is to be substituted into the encoded video is determined in step 802 in a similar manner as described with respect to steps 780 and 782. Next, step 804 adds GOPs as needed (one or two GOPs) of unencoded original video to the trailing end of the new section of video.

[0228] Step 786 then encodes the video. The first GOP of the new video is encoded in the closed Group Of Pictures mode and the number of bits for the bidirectionally predicted frames which are missing the reference frame is increased

LINKPLAY EXHIBIT 1004
Page 1649 of 4822

to approximately the number of bits used for single direction predicted frames, P-frames. If the total number of bits for the encoded video is to stay the same, the extra bits needed for the bidirectional frame must be removed from other frames. This may be accomplished above as described in the sections on spatial and/or temporal manual editing. Next the remainder of the new section of video is normally encoded. Last the added period at the trailing end of the new video is encoded using the same quantizer values as were used during the original encoding of the added period to prevent decoding artifacts at the end point of the edit segment. Finally, the newly encoded video is substituted into the originally encoded video in step 808.

[0229] By either of the processes described above and illustrated in Figures 34 and 36, the final version of the encoded video containing the substitute encoded sections has a reduced number of decoding artifacts and therefore, the visible video glitches are reduced, increasing the quality of video as compared to if no correction steps were taken to prevent the decoding artifacts. The examples described above relate to edit points at boundaries of GOPs. However, the above teachings can be applied to edits which are not at GOP boundaries.

[0230] The process to reduce decoding artifacts is carried out using workstations 10, 30, and 40 along with the playback VTR 51 and the video encoder 50. However, other hardware alternatives are possible and the above aspect of the invention may be implemented using any general purpose computer or specific hardware to perform the process. The calculations used to determine the edit points and to determine the video to be added at the end of the new section of video to be encoded are performed either in workstation 10 or workstation 30.

IX. RATE-QUANTIZER MODELING USED FOR ENCODING

[0231] To accurately estimate the number of bits that will be needed to represent a series of macroblocks resulting from an encoding process, a frame can be pre-encoded using several quantization levels that are representative of normal quantization levels for a frame. The frame then can be actually encoded using the representative quantization levels as well as other quantization levels. In the preferred embodiment, illustrated in Figure 1A, workstation 10 loads predetermined quantization levels into the video encoding device 50. The video encoding device 50 then performs the pre-encoding. The most accurate method of generating a quantization model would involve encoding each frame at each possible quantization level and recording the statistics on the resulting number of bits used. However, this process would require n passes over the raw video where n is the total number of quantization levels used to encode the raw video. As this method is prohibitively long, this invention provides an improved method, which generates similar results to the optimal method in a fraction of the time needed for the conventional method. The method involves assigning various parts of plural frames with different quantization levels such that a representative bit-rate versus quantization level average can be calculated for each of the plural frames during a pre-encoding phase. This saves n-1 pre-encodings for each frame while generating accurate quantization level versus bit-rate statistics. These statistics can then be interpolated to estimate the number of bits to be used by a macroblock whose quantization level versus bit-rate was not determined in the pre-encoding process. In two alternate embodiments, the method pre-encodes a majority of frames from the raw video or all the frames of the raw video.

[0232] Figure 37A shows a frame of raw video that has two different quantization levels assigned on a macroblock by macroblock basis in the pre-encoding process so that a two point quantization level versus bit-rate line is generated for plural frames and the results are stored for the plural frames. In alternate embodiments, the quantization levels could be assigned in groups or strips of macroblocks. Figure 37B illustrates the uniform distribution of four quantization levels across a frame where the quantization levels are assigned in different positions in the frame to avoid statistical anomalies based on position. In the illustrated frame, for each row, j, the first quantization level in the row is $q_{j \bmod 4}$. This results in a four point quantization level versus bit-rate curve. Figure 37C shows the same process as Figure 37B, but the quantization levels are assigned in a block fashion. Although a single pass of this process produces an accurate quantization level versus bit-rate curve, in an alternate embodiment, this process is repeated a number of times across the plural frames to generate more accurate statistics that approach the statistics that would be gathered by analyzing each frame at each quantization level.

[0233] Figure 38 illustrates the general process of determining the quantization level versus bit-rate characteristics of plural frames. The process starts in step 850 and continues in step 852 by obtaining video signals which represent the raw video to be pre-encoded. In step 854, the process signifies that the first frame, i=1, where i is the frame index, is to be pre-encoded. Continuing to step 856, the quantization levels for $frame_{i=1}$ are assigned uniformly across the frame to prevent statistical anomalies. In step 858, $frame_{i=1}$ is pre-encoded using the quantization levels assigned in step 856. The results of step 858 are determined in step 860 such that the bit-rates for each quantization level assigned in step 856 is recorded for $frame_{i=1}$. In step 862, the method determines whether or not all of the plural frames have been pre-encoded. Since only one frame has been pre-encoded, the process continues to step 866 where it designates the second frame as the frame to be encoded by adding one to the frame index, i. Control is then returned to step 856 which assigns quantization levels to $frame_{i=2}$ The process of assigning, pre-encoding and determining the bit-rates in steps 856, 858, 860 and 862 are continued for all remaining frames. When step 862 determines that all of the plural frames have been

LINKPLAY EXHIBIT 1004
Page 1650 of 4822

pre-encoded, the process ends by moving to step 864. When step 864 has been reached, a quantization level versus bit-rate curve is approximated by using the quantization levels assigned in step 856 and their corresponding bit-rates.

[0234] Having generated quantization level versus bit-rate curves for plural frames of a raw video, the pre-encoding process moves on to an encoding process by which parts of the raw video are encoded to fit on a digital storage medium, i.e., a compact disc. One, plural, a majority or all of the raw video segments may be encoded in this second phase. If the raw video segments are to be encoded automatically, an encoder uses an activity measure of each macroblock combined with a corresponding quantization level versus bit-rate curve to determine quantization levels to be assigned to macroblocks of each frame in the segments. Each frame in the segments are then encoded and their resultant number of bits are compared with an estimated number of bits based on the quantization levels assigned. In cases where the resultant number of bits varies greatly from the expected number of bits, the automatic encoder attempts to compensate for the variation by changing the quantization levels of the next set of macroblocks to be encoded. This sequential process allows an automatic encoder to encode segments of the entire raw video to match a target number of bits.

[0235] In addition, the quantization level versus bit-rate curves of later frames in a video sequence can be used to regulate quality throughout a resultant encoded digital video. As is shown in Figure 39, Frame N is known to generate fewer bits than Frame M for the same quantization levels. Therefore, Frame M requires more bits to generate the same quality image as in Frame N. When Frame N is encoded at a desired quality without using all its allocated bits, the encoder can carry the extra bits forward to be used in encoding Frame M which requires more bits per unit time for the same quality. This capability is not available in systems with only a single frame look ahead buffer when M>N+1.

X. COMBINING AUDIO, VIDEO, AND SUB-PICTURE DIGITAL SIGNALS USING DATA STRUCTURES

[0236] The above portion of this patent specification is primarily concerned with the techniques used during encoding of audiovisual information to generate encoded and compressed digital audiovisual signals. However, in order for the audio and video information to be combined, stored, and subsequently decoded in order to recreate or reconstruct the audiovisual information, such as on a television, it is necessary to combine and interleave the audio, video and other information which has been encoded so that it can be stored and subsequently decoded. The combining of the audio, visual, and other information has been briefly described above as a formatting process which occurs in the workstation 20 illustrated in Figure 1A. Further details of the formatting process and resulting data structures and signals will now be given.

[0237] Turning now to Figure 1A, the MPEG video and encoded audio are stored in the disk 21. Additionally, sub-picture information which includes subtitles and/or graphical representations which are displayed and/or overlaid on top of the video are earlier encoded by a separate workstation, for example, and the encoded sub-picture information is stored on a digital storage medium such as a floppy disk, tape, or other type of disk. This stored sub-picture information is read by one of the digital storage devices 22, combined in a formatting process with the encoded audio and video in disk 21, and formatted into a single data stream by workstation 20. The output of the formatting process is stored on the disk 21 and then written to a digital tape of the digital storage devices 22. The tape is then used to create optical discs by known optical disc manufacturing methods. Accordingly, not only is the invention related to the encoding of the audio and visual data but is also related to a process and system for combining encoded, audio, video, and sub-picture data into a data stream, the digital storage medium and data structures thereon which store the formatted audio, video, and sub-picture information, and is also related to a decoding process and a decoder which separates and reconstructs the audio, video, and sub-picture information for display to a user or consumer. It is to be noted that the data structures on the optical disc or within the formatted data pertain to specific electronic structural elements which impart a physical organization on the information stored in memory. These specific electronic structural elements are stored in an optical disc which can be used for a video on demand system or which are decoded for a user or consumer using a digital video disc player.

[0238] Before describing the specific features of the format structure of the disc or other storage medium for the encoded audiovisual information, known standards on which the invention improves will be described. ISO/IEC 13818-1 describes system aspects of MPEG video and audio. ISO/IEC 13818-1, referred hereinafter as the MPEG system description, describes the basic multiplexing approach for video and audio streams. These streams are broken up into packets called Packetized Elementary Stream (PES) packets. These packets from different streams including audio, and video streams have a common time base and are combined into a single stream. Also described in the MPEG system description is the use of a transport stream which combines one or more programs with one or more independent time bases into a single stream. The MPEG system description describes the use of Presentation Time-Stamps (PTS) which are used to synchronize multiple elementary streams. The time-stamps are generally in units of 90kHz and are used in conjunction with a System Clock Reference (SCR), the Program Clock Reference (PCR) and an optional Elementary Stream Clock Reference (ESCR). A complete description of the details of how the data is formatted according to the MPEG system description are found in ISO/IEC 13818-1.

[0239] In addition to using a formatting process based on ISO/IEC 13818-1, the data is also formatted and stored

LINKPLAY EXHIBIT 1004
Page 1651 of 4822

according to ISO 9660, 1988, Information Processing - Volume and file structure of CD-ROM for information interchange. This standard describes how the disc volume (not loudness but disc content) and the file structure is organized.

[0240] The preferred embodiment, at the current time, of the optical disc which is used to store the audio, video, and sub-picture data is a single layered double-sided disc having a capacity of 5 gigabytes per side with a total storage capacity of 10 gigabytes. Future formats will allow for a multi-layer disc in order to increase storage capacity, and in addition to using a read-only disc, write-once and write-over technology can also be used. Other aspects of the disc which may be used with the present invention are describe in U.S. patent application 08/304,849 filed September 13, 1994, and entitled "Optical Disk and Optical Disk Apparatus" (US 5 459 712 A). The outside diameter of the disc is preferably 120 mm.

[0241] Figure 40 illustrates the structure of the formatted output of workstation 20 which is ultimately included on an optical disc. This structure 900 includes a lead-in area 902. The lead-in area contains individual fields which controls subsequent processing of the data read from the disc. Exemplary lead-in information included in lead-in area 900 is illustrated in Figure 3 of U.S. Patent 5,400,077 to Cookson et al.

[0242] The system area 904 and volume management information 906 comply with ISO 9660. The volume management information 906 includes a primary volume descriptor, a volume descriptor set terminator, directory records, and path table records. The primary volume descriptor contains basic information of the disc volume such as a descriptor type indicating whether the disc is a video disc of audio disc, a standard identifier, indicating to what standard the disc conforms with, a system identifier, a volume identifier, the volume size, logical block size, path table size, path table size, etc. These and other fields of the system area and volume management information are described in detail in ISO 9660 and a description thereof is omitted for sake of brevity. Additionally, there is a supplementary volume descriptor, if desired, which conforms to ISO 9660.

[0243] The volume descriptor set terminator contains the volume descriptor type, the standard identifier, and the volume descriptor version. The directory records contain various directory information of the disc in accordance with ISO 9660. The path table records contain an L-type path table and an M-type path table, as described in ISO 9660.

[0244] The disc information file 908 includes further information about the content of the disc and menu information which a user may choose when playing the optical disc. Details of the disc information file 908 are described with respect to Figure 41.

[0245] Each disc contains a minimum of one data file 910 up to a maximum of 99 data files. For example, if there are two different audiovisual programs on the disc, the disc information file 908 would contain the information necessary to generate a menu for the user to choose which of the data files the user wish to use. The data files are described in detail with respect to Figures 42-56. The lead-out area 916 but may contain other information describing the disc, how processing is performed, or information concerning related discs.

[0246] Figure 41 illustrates the disc information file 908 of Figure 40. The disc information file 908 includes file management information 920 and menu data 922 which is information making up a menu with which a user interacts. The menu data includes video data, audio data, and sub-picture data. The file management information 920 includes a file management table 924, disc structure information 926, menu structure information 928, and a menu cell information table 930 which includes a plurality of menu cell information fields 932 and 934.

[0247] The file management table 924 includes information related to every aspect of the disc information file. Included within the file management table 924 are a file name which describes the same contents as the corresponding file identifier in the corresponding directory record of the volume management information 906 for identifying a file name. The file identifier identifies the type of file which in this case is a disc structure information file. There is a field which describes the size of the file using the number of logical blocks. Throughout this writing, the term "logical block" is used as defined in ISO 9660. There is an entry describing the size of the file management table using a number of logical blocks, a start address of the disc structure information 926 using the relative logical block number from the head of the file, a start address of the menu structure information 928 described using the relative logical block number from the head of the file, a start address of the menu cell information table 930 using the relative logical block number from the head of the file. If there is no menu data, the value for this field will be zero. There is a field which describes the start address of the menu data 922 using the relative logical block number from the head of the file. When no menu data exists, this entry is zero.

[0248] The file management table 924 also contains a field which describes video attributes of video for the menu data. This information includes whether the video compression mode was MPEG-1 or MPEG-2, the frame rate of the video (either 29.97 frames per second or 25 frames per second), a field indicating whether the display aspect ratio is 3/4 or 9/16, and a display mode field indicating both pan-scan and letter box modes are permitted, indicating that the pan-scan mode is permitted but the letterbox mode is prohibited, or indicating that the pan-scan mode is prohibited and the letter box mode is permitted.

[0249] Similar to the field describing the video attributes, there is also a field describing the audio stream in the file management table 924. Included within this field is an indication of the audio coding mode including whether the audio was encoded according to Dolby AC-3, whether the audio is MPEG audio, or whether the audio is linear PCM audio (16

LINKPLAY EXHIBIT 1004
Page 1652 of 4822

bits at 48 kHz). There is also an indicator of whether the audio is monaural, stereo, or digital surround.

[0250] The file management table 924 further contains a field which describes the sub-picture stream attributes for the menu data 922. This field indicates a run-length sub-picture coding mode will be used for the sub-pictures, as described below. The file management table 924 also contains a field describing 16 sets of color pallets used in all sub-pictures for the menu. The pallet numbers from 1 through 16 contain a luminance signal Y, a color difference signal Cr=R-Y, and a color difference signal Cb=B-Y.

[0251] The disc structure information 926 includes configuration information of the video and audio files stored on the disc. The disc structure information 926 includes a file name which describes the same contents as the corresponding file identifier in the directory record which is used for identifying a file name. There is a file identifier field which identifies the file as a disc structure information file. There is a field for indicating the number of data files on the disc such as data files 910 and 914 illustrated in Figure 40. There is also a field indicating the type of each of the data files including whether the file contains both video and audio information or only audio information, and whether or not the file is a karaoke file.

[0252] The disc structure information 926 also contains a field describing the sub-picture and audio information in the data file. Indicated within this field are the number of sub-picture channels. Each sub-picture channel may be used to display different sub-picture information such as different sub-picture languages. There is also an indication of the number of audio streams, an indication of the language code of each sub-picture channel in the order of channel number, a sequential description of the language code of the audio stream in the order of the audio stream number, and a description of parental management and angle information of the files. The parental management information is used to describe the parental level of a particular sequence of information to be displayed. It is the maximum parental level of specific cells contained within a sequence. The levels vary from 1 through 5 and the higher the level, the more restriction over what is heard and seen. Details of the sequence and cells and parental management scheme are explained below. There is also a field which describes the number of angles of a video sequence. For example, if a music video had three different camera views throughout the entire music video, the user may select which particular camera angle is desired to be viewed. There is also a field indicating the number of programs in a sequence.

[0253] The menu structure information 928 contains the start cell number of each menu type. A display sequence includes a plurality of cells. One sequence can contain up to 256 cells, for example. A cell is an integral number of GOPs (group of pictures) between a playback information (PBI) pack, to be described later, and a pack immediately before the next PBI pack. Data for video is separated into cells as a record unit in which the data should be reproduced successively. A cell contains the information for display and is divided depending on its purpose. It is desirable that the first audio pack and the first sub-picture pack in a cell include a presentation time-stamp (PTS) close to a PTS in the video head pack recorded behind a PBI pack at the head of a cell. Different cells are displayed depending on the desired angle and parental restriction of the program. As an example, suppose there was a section of a movie which was inappropriate for younger audiences. There can be two sequences sharing cells, depending on the parental management level which may be viewed. For example, for unrestricted viewing, cells 1, 2, 3 and 5 may be viewed for a particular sequence. For a restricted level of viewing, the cells viewed for the sequence may be 1, 2, 4 and 5.

[0254] Regarding the different angles, it is possible to have each video recorded at plural camera angles which proceed simultaneously and whose time required for reproduction is substantially equal. Each angle is constituted by one cell called an angle cell. A set of plural angle cells which precede simultaneously is called an angle block. If an angle block is defined in plural locations in the sequence, the number of angle cells and the camera angle of each angle number must be the same through the angle blocks.

[0255] A menu cell is the cell constituting the menu screen which informs the user of the contents of each video or audio file in a volume. A set of menu cells is recorded together in the disc information file as data for the menu. The menu cell types, also referred to as menu types, are classified into the title menu, a program menu, an audio menu, a sub-picture menu, and an angle menu. A menu screen includes video for the menu. Items for selection are displayed on the screen of reproduction equipment are recorded as sub-picture data of the menu cell.

[0256] The menu cell information table 930 is a table on which plural menu cell information such as 932 and 934 in Figure 41 are stored. One menu screen includes one menu cell. The menu cell information includes a plurality of fields. A first field indicates whether or not copying of information of the disc is permitted, the level of parental management, the menu cell type such as whether the menu is a title menu, program menu, audio menu, sub-picture menu, or angle menu, and the language code of the menu. Each language is assigned a code and the specific code appearing in the language code sub-field indicates the language used with the menu.

[0257] There is also a field which describes the start number of selection items displayed by the sub-picture on the menu screen and the number of the selection items. The selection item start number describes the smallest number of selection items on the menu screen, there is a field which indicates whether or not a next page of the menu exists, and a number which describes the number of items between 1 and 16 which may be selected. The menu cell information also includes the lower 32 bits of the system clock reference (SCR) in the start pack of the menu cell, a field which describes the start address of the menu cell with a relative logical block number from the head of the file, and a field

LINKPLAY EXHIBIT 1004
Page 1653 of 4822

which describes the number of logical blocks constituting the menu cell.

[0258] The menu data 922 contains the actual information which is displayed for the menu. This information is constructed in the same manner as the information in the data files 910 and 914 in Figure 40 is constructed and accordingly, an in depth description of each aspect the menu data 922 is omitted to prevent redundant descriptions. The items which are selected on the menu screen are formed by sub-picture information. The processing of menu selections by a disc player is performed using a microprocessor executing computer code. When a menu selection for a title is made, the result of the selection is reproduced from the beginning of the file or sequence number corresponding to the selected title number. For program menu selections, the selected program is reproduced from the beginning of the selected program. For audio selections from the menu, the audio stream is set to correspond to the selected audio stream. Similarly, for sub-picture menu selections the sub-picture channel number is set equal to the selected sub-picture channel. Last, an angle selection from the menu changes the angle cell number to equal the selected number. Other information, as desired, may be included in and selected through menus.

[0259] The data files such as the data file 910 of Figure 40 contain two sections, as illustrated in Figure 42; data file management information 940 and the actual data 942 used to display audio, video, and sub-picture information. The data file management information 940 includes four major components; the data file management table 944, the sequence information table 946, the cell information table 948, and a data search map 950.

[0260] The data file management table 944 includes information related to every aspect of the data file 910. This information includes a field for the file name which is the same contents as the corresponding file identifier in the directory record for identifying the file name, the file identifier which indicates that the file contains both audio and video information, the size of the file as defined by the number of logical blocks, the size of the file management table, the number of sequences for the file, the number of cells for the file, the number of disc structure information (DSI) packs, a start address of the sequence information table 946, a start address of the cell information table 948, a start address of the data search map, a start address of the data 942, a field describing the video attributes of the data 942 including the video compression mode such as MPEG-1 or MPEG-2, the frame rate including whether the frame rate is 29.97 frames per second or 25 frames per second, a display aspect ratio indicating whether the display aspect ratio is 3/4 or 9/16, or whether the display mode permits one or both of pan-scan and letterbox format.

[0261] The data file management information further includes a field describing the number of audio streams in the file, the audio stream attributes such as the audio coding mode, the audio mode such as monaural, stereo, or digital surround, the audio type which indicates whether or not there is a specific language included, and a field for a specific code which indicates the code number of the language.

[0262] The data file management table 944 further includes a field describing the number of sub-picture channels, a field describing the sub-picture channel attributes such as a field which indicates run-length coding of the sub-pictures is used and whether or not there is a specified language and the language which is specified. Additionally, there is a field indicating the Y, Cr, and CB colors for 16 color pallets used in all sub-picture channels of the file. There is also a field describing the size of the sequence information table, a field describing the smallest cell number and number of consecutive cells which follows, and the range of cell numbers used within one sequence of the file. The cell numbers used within one sequence should be included in a set of numbers figured out by letting 32 consecutive numbers make a set, and placing the smallest cell number to be the first number of the set. This field includes a sub-field including the number of cells used in the sequence, and the smallest cell number used in the sequence.

[0263] The sequence information table 946 includes a plurality of sequence information entries 952 and 954. A sequence is the order in which cells within the range specified by this sequence are reproduced selectively. There are two types of sequences; completion and connection. A completion-type sequence is terminated after reproducing itself. Connection-type sequences are sequentially reproduced. The sequence information number such as sequence information 1, indicated by reference number 952, and sequence information i, indicated by 954, is equal to the sequence number and is numbered in the described order in the sequence information table beginning with 1. The sequence information entries of the sequence information table are described in the order of completion-type sequences or connection-type sequences. Each sequence information contains field indicating the sequence type, the number of included programs, the number of included cells, the sequence reproduction time, the number of connection-type sequences, the connectable sequence number, and sequence control information.

[0264] The sequence type field of the sequence information describes copy and parental management of the sequence. There is a sub-field indicating whether copying is permitted or prohibited, a sub-field specifying the level of parental management from 1-5, a sub-field indicating the sequence type which may be one of a completion-type sequence, a connection-type head sequence, a connection-type midsequence or a connection-type end sequence. There is also an application type sub-field which indicates whether or not the sequence is for use is a karaoke application.

[0265] The number of included programs field describes the number of programs, up to 100 in a sequence. The number of included cells field describes the number of total cells in a particular sequence, up to 256 cells in total. The sequence or reproduction time describes the total reproduction time of this sequence in units by hour, minute, second, and video frame.

LINKPLAY EXHIBIT 1004
Page 1654 of 4822

[0266] The number of connection-type sequences field includes the number of sequences which can be connected immediately after them, the maximum number being 8. The connectable sequence number field describes sequence numbers of connectable sequences and the level for parental management of the sequence for the numbers specified. The selection number of connectable sequences are assigned from one in the described order. This field contains sub-fields for the parental management number, and a sub-field for the sequence number. The last field of the sequence information is the sequence control information which describes the cells included in the sequence in the order of reproduction. This field is also used to describe angle blocks which is a set of plural cells. The reproduction time of those cells is substantially equal. Each reproduction section is constituted by one cell. An angle block includes nine cells at most with the first cell numbered 1. The sub-fields of the sequence control information include a program number which defines a program as a reproduction unit combining plural cells which are sequentially reproduced. There is a sub-field for cell reproduction control which indicates whether to proceed to the following cell sequentially, to pause, or whether the end of a sequence has been reached, there is a block mode sub-field which indicates if a block is not a configuration cell, whether there is a first cell of block configuration, whether it is a cell in a block configuration, or a final cell in a block configuration. There is a block type sub-field which indicates if there is not a block or whether there is an angle block. Last, there is a sub-field which indicates the cell numbers of the cells to be reproduced.

[0267] The data file management information 940 further includes a cell information table 948 containing a plurality of cell information entries 956 and 958. Each cell information entry includes cell type information which indicates whether copying is permitted or prohibited, and the level of parental management. There is also a field indicating cell reproduction time which describes the total cell reproduction time of the cell by hour, minute, second and video frame. There is a field which describes the lower 32 bits of the system clock reference described in the first pack of the cell, a field which describes the address at the cell's starting point with the relative logical block number from the beginning of the file, and a field which describes the number of logical blocks included in the cell.

[0268] The last section of the data file management information 940 is the data search map 950. The data search map includes a plurality of pointers of data search information packs 960 and 962. These pointers, referred to as playback information (PBI) pack pointers, give the address of the playback information pack which exists in the data 942. In order to perform a fast forward or fast reverse mode and allow viewing of the information, it is best to rely on intra pictures in the MPEG-2 video sequence. These intra pictures are located using playback information packs within the data 942 whose addresses are contained in the data search map. A further description of the PBI pack is given below.

[0269] The data 942 of the data file 910, illustrated in Figure 43, includes interleaved packs of playback information (PBI), video, sub-picture, and audio information. For example, in Figure 43, the data includes playback information 940, video information 942 and 944, sub-picture information 946, audio information 948 etc. The structure of the information within the data 942, and presentation timing complies with the program stream defined in the MPEG system description (ISO/IEC 13818-1). However, the playback information and sub-picture information and the manner in which this information is stored, encoded, and decoded constitute part of the present invention.

[0270] The various information making up the data 942 in Figure 43 includes information packs. Exemplary packs 970A and 970B are illustrated in Figures 44A and 44B. Each pack comprises a pack header 972A or 972B, and a packet 982A or 982B which includes video, audio, sub-picture, or playback information data.

[0271] The structure of pack 970A is used when the packet 982A occupies between 2034 and 2027 bytes. The stuffing field 980A allows 1 to 8 bytes of stuffing to bring the total size of the pack to 2048 bytes. When the packet for video, audio, sub-picture, or playback information is less than 2027 bytes, a pack structure 970B as illustrated in Figure 44B is used which has a stuffing 980B of one byte and a packet for padding 984 which makes the total number of bytes for the packets 982B and 984B to be 2034 bytes. The packet headers 972A and 972B each contain a pack start field 974, a system clock reference 976, a multiplex (MUX) rate field 978 and a stuffing field 980. The packets in Figure 44A and 44B are constructed in accordance with ISO/IEC 13818, and described in §2.5.3.4 thereof pertaining to the semantic definition of fields in a program stream pack.

[0272] Figure 45 illustrates a playback information (PBI) pack. The pack header 972 is constructed as described with respect to Figures 44A and 44B. The system header 986 of Figure 45 is constructed in conformance with the requirements of the program stream system header described in ISO/IEC 13818-1. The semantic definition of fields within the system header is described in §2.5.3.6 of ISO/IEC 13818-1.

[0273] The packet header 988 is constructed in accordance with the fields set forth in the semantic definition of fields in PES packet described at §2.4.3 of ISO/IEC 13818-1. However, only fields up to the presentation time-stamp are needed in the packet header 988.

[0274] The playback information pack further includes a sub-stream ID field 990. This is an 8 bit field which indicates the identification of the sub-stream. The different identifications which may be used are a sub-picture stream, a video blanking information (VBI) stream, an AC-3 stream, or a linear PCM stream. It is also possible to include other stream types such as an MPEG audio sub-stream. As the sub-stream ID 990 precedes the data of the video blanking information 992, the sub-stream ID is set to indicate that the information which follows is a video blanking information stream. The data of the video blanking information 992 may be constructed in accordance with ISO/IEC 13818-1 and conform to the

LINKPLAY EXHIBIT 1004
Page 1655 of 4822

semantics required for ISO/IEC 13818-1.

[0275] Also included within the playback information pack illustrated in Figure 45 is a packet header 994 for a data search information packet which includes the data of the data search information 996. The packet header 994 includes a packet start code prefix having a 24 bit value of 000001h, a stream identification indicating that the data search information 996 does not conform to previously defined standards, and a field indicating a packet length. As the data search information 996 is not disclosed in the MPEG standard, the stream ID indicates that the information which follows is a private type of data stream.

[0276] The specific content of the data of the data search information 996 illustrating Figure 45 is shown in Figure 46. The data search information 996 includes general information 1000, angle address information 1020, highlight information 1022, synchronous reproduction information 1024, and PBI pack address information 1036.

[0277] The general information 1000 is illustrated in further detail in Figure 47. As previously stated, a purpose for the data search information is to allow a quick search of the digitally encoded video and specifically allows a rapid location of an intra picture for trick-modes executed during video decoding. Therefore, the playback information pack illustrated in Figure 46 is the head pack of each group of pictures (GOP) and allocated immediately before the video head pack starting with an I-picture. The general information 1000 includes a time-stamp of the GOP 1002 which is the reproduction start time of an I-picture. Next, there is a system clock reference (SCR) of the PBI. This describes the lower 32 bits of the SCR described in the PBI pack header. 1006 indicates the end address of the I-picture and describes by the relative logical block number from the beginning of the file, the address of the pack on which the final data of the I-picture is recorded. The cell number is stored in field 1008 and describes the cell number to which the GOP belongs. Field 1010 describes the parental management information including the level of restriction and also whether or not copying is permitted. Last, field 1012 describes the PBI pack address which is the address of this PBI pack using the relative logical block number from the head of the file.

[0278] Field 1020 of the data search information 996 illustrated in Figure 46 is the angle address information. The angle address information includes address information of other angles. The angle address information includes nine fields, each pertaining to a different angle cell number. For each field in the angle address information 1020, if a cell to which a specific PBI pack belongs constitutes an angle block, the field in the angle address information describes the address of the PBI pack for the angle cell number which has the closest video start presentation time-stamp not exceeding the video start presentation time-stamp of the corresponding data search information. If no angle cell exists or if an angle block is not constituted, the angle address information for a specific cell is set to zero.

[0279] The third section of the data search information 996 is the highlight information 1022. The highlight information describes the position of selection items on the menu screen and their color and contrast to be changed when selected. This information is used only when the cell is part of a menu. The highlight information includes a field describing the selection item start number and the number of items displayed by the sub-picture on the menu screen. The specific sub-fields of the selection item start number and number of items include a sub-field which describes the smallest number of selection items on the menu screen, a sub-field which describes whether or not a next page of the menu exists, and a sub-field which describes the number of selection items on the menu screen. The second field of the highlight information includes position, color, and contrast of the selection item. This field describes a rectangular display area for each selection item on the menu screen and the corresponding color and contrast to be changed, when selected. The rectangular display area is defined in an X-Y coordinate plane. Information in this field describing the position, color, and contrast of the selection item include defining the rectangular area of the menu, and information describing how the various types of pixels are changed when selected. As will be described in more detail later, a sub-picture allows for four different types of pixels. There are first and second emphasis pixels which may be used to display text or other information, pattern pixels which are used to display patterns such as line drawings or other graphical patterns, and background pixels. When these four different types of information are highlighted or selected, it is necessary to know the color which will be chosen when the item is highlighted and also the contrast of the pixel, when highlighted. Therefore, there are eight sub-fields within the highlight information describing the color and contrast of the four different types of pixels when they are selected.

[0280] The fifth item within the data search information 996 is the synchronous reproduction information 1024. The synchronous reproduction information 1024 is illustrated in detail in Figure 48. The purpose of the synchronous reproduction information is to be able to find the start time and address information of the audio and sub-picture which is synchronized with the video data. The synchronous reproduction information 1024 illustrated in Figure 48 contains the audio pack address 1026 which is the target of the presentation time-stamp of the audio. The most significant bit of this field indicates whether the audio pack is located before or after this PBI. The presentation time-stamp (PTS) of the audio 1028 describes the PTS of the audio packet having the reproduction start time just behind the reproduction start time of the I-picture with the relative presentation time-stamp from the video start PTS. The presentation time-stamp of the audio allows for up to eight audio streams to have the presentation time-stamp stored therefore. Additionally, there may be a sub-field indicating whether or not the audio packet comprising audio frames which is reproduced during the GOP. If there is no audio for the picture, it will not be necessary to store or decode the presentation time-stamp of the audio.

LINKPLAY EXHIBIT 1004
Page 1656 of 4822

As there are up to eight audio channels, each audio channel has corresponding address and time stamp fields.

[0281] Data structure 1024 also includes a sub-picture pack address 1030 which describes the address, relative to the PBI pack, of the sub-picture pack at issue. The presentation time-stamp of the sub-picture describes synchronization of corresponding sub-picture information. There are sub-fields which indicates whether or not a sub-picture unit which is to be reproduced during the GOP reproduction exists, whether or not the reproduction start time of the sub-picture is before the video presentation time-stamp, the production start time of a sub-picture unit reproduced during the GOP reproduction, and the reproduction termination time of a sub-picture unit reproduced during the GOP reproduction. As up to 32 sub-pictures which may be included into a single data stream, each sub-picture which exists has a corresponding address and time-stamp field.

[0282] The last section illustrated in Figure 46 of the data search information 996 is the PBI pack address information. The PBI pack address information comprises addresses of other PBI packs within the file and allows other PBI packs to be quickly located. The PBI pack address information describes the address of the surrounding PBI packs using relative logical block numbers. The PBI pack address information includes two fields which describe the next and previous PBI pack addresses, respectively. There are fifteen fields which define, using the relative logical block number from the current PBI pack address, the address of the PBI pack which belongs to the GOP immediately ahead of the GOP with its accumulated reproduction time being between n x 0.5 seconds and (n+1) x 0.5 seconds, and when more than one PBI pack falls in this range, selecting the PBI pack closest to n x 0.5 seconds, $1 \leq n \leq 15$.

[0283] There are fifteen fields which define, using the relative logical block number from the current PBI pack address, the address of the PBI pack which belongs to the GOP with its accumulated reproduction time being between n x 0.5 seconds and (n+1) x 0.5 seconds, and when more than one PBI pack falls in this range, selecting the PBI pack closest to n x 0.5 seconds, $-15 \leq n \leq -1$.

[0284] There are two fields which define, using the relative logical block number from the current PBI pack address, the address of the PBI pack which belongs to the GOP immediately ahead of the GOP with its accumulated reproduction time being between n x 0.5 seconds and [n x 0.5 + 1] seconds, and when more than one PBI pack falls in this range, selecting the PBI pack closest to n x 0.5 seconds, n=20 or n=60.

[0285] There are two fields which define, using the relative logical block number from the current PBI pack address, the address of the PBI pack which belongs to the GOP with its accumulated reproduction time being between n x 0.5 seconds and [n x 0.5 + 1] seconds, and when more than one PBI pack falls in this range, selecting the PBI pack closest to n x 0.5 seconds, n=-20 or n=-60.

[0286] The structure of a video pack 1040 is illustrated in Figure 49. The video pack contains a pack header 972 constructed in accordance with the headers 972A or 972B illustrated in Figures 44A and 44B. The packet header 1046 for the video is constructed in accordance with ISO/IEC 13818-1 as is the video data 1048. However the video data 1048, represented as MPEG video signals is preferably encoding in accordance with the techniques described above.

[0287] An MPEG encoded group of pictures is illustrates as 1050 in Figure 50. The manner of breaking up this GOP 1050 into a plurality of video packs 1040 to generate the video stream 1060 is illustrates at the bottom portion of Figure 50. The group of pictures 1050 is broken up into appropriate units to fit into the video packs. The last video pack 1040G, video pack j, does not take up the full 2048 bytes used for each of the packs and therefore, stuffing is used to make the last video pack j equal to 2048 bytes.

[0288] Figure 51A illustrates a video pack for MPEG audio. This video pack 1080 includes a pack header 972, a packet header 1084 constructed in conformance with the MPEG requirements, and the encoded MPEG audio data 1086.

[0289] Figure 51B illustrates an audio pack 1090 constructed in conformance with either linear PCM audio or Dolby AC-3 audio. For the packet illustrated in Figure 51B, the pack header 972 is arranged as described above, the packet header 1094 is constructed in conformance with the MPEG requirements with the stream_id field set to indicate a private audio data stream corresponding either to linear PCM or AC-3. The sub-stream id 1096 indicates whether the audio data is for liner PCM or AC-3 (or MPEG). The audio frame pointer 1098 includes a number of frames header which indicates the number of audio frame headers within the audio packet and also includes a first access unit pointer which indicates the head of the first access unit of the audio frame using a relative byte number from the last byte of the pointer. It is standard for linear PCM audio to be broken into audio frames, each frame including a frame header and frame data. The audio frame pointer 1098 may similarly be used with dolby AC-3. The audio data 1100 is constructed in accordance with the corresponding audio encoding method.

[0290] Figure 52 illustrates an encoded audio stream 1100. This audio stream is converted into a stream of audio packs 1120 comprising individual packs 1122. Each pack is 2048 bytes in length and the last audio pack 1122G, audio pack j, is adjusted by adding stuffing bits in order to bring its length to 2048 bytes.

[0291] A novel feature of the present invention is the use of sub-pictures and the interleaving of sub-picture packs with audio and video information. Sub-picture information allows the display of any graphical information and allows for overlapping the sub-picture information over the MPEG video. A major difference between the sub-picture information used by the present invention and conventional closed caption subtitles is that the sub-picture information is bit mapped graphics whereas closed captioning transmits character codes to display characters within character sets which are

LINKPLAY EXHIBIT 1004
Page 1657 of 4822

stored in the decoder. Accordingly, conventional closed captioning is limited by the character set within the decoder. However, as the sub-picture information is bit mapped, any type of characters including foreign language characters may be displayed without the need to store character sets within the decoder for each language.

[0292] Each display screen of sub-picture information is called a sub-picture unit. There is a sub-picture unit 1140 illustrated in Figure 53. A sub-picture unit contains one screen of bit mapped pixel data and this screen of pixel data is displayed across a plurality of video frames. The sub-picture unit 1140 includes a sub-picture unit header 1142. The sub-picture unit header 1142 includes a field describing the size of the sub-picture unit and the start address of the display control sequence table 1146, relative to the number of bytes from the head of the sub-picture unit.

[0293] The pixel data 1144 is run-length compressed bit mapped information. The pixels may be background pixels which form the background of the displayed information, pattern pixels which allow the display patterns within the sub-picture including graphical line drawings, and two types of emphasis pixels which allow graphics or characters to be displayed which have two different attributes. The bit mapped information may be created using conventional graphics generating techniques. The run length compression is performed in accordance with the following rule. If one to three pixels of the same kind follow, enter the number of the pixels in the first two bits and the pixel data in the following two bits. The four bits are considered a unit. If four to fifteen pixels of the same kind follow, specify zero in the first two bits and enter the number of pixels in the following four bits and the pixel data in the next two bits. The eight bits are considered to be one unit. If sixteen to sixty-three pixels of the same kind follow, specify zero in the first four bits and enter the number of the pixels in the following six bits and the pixel data in the next two bits. The twelve bits are considered to be one unit. If sixty-four to 255 pixels of the same kind follow, specify zero in the first six bits and enter the number of the pixels in the following eight bits and the pixel data in the next two bits. The sixteen bits are considered to be one unit. If the same pixels follow to the end of a line, specify zero in the first fourteen bits and describe the pixel data in the following two bits. The sixteen bits are considered to be one unit. If the byte alignment does not result when the description for pixels on one line is completed, insert dummy data of four bits for adjustment. It is also possible to use other compression mechanisms to represent the image data, as desired. For example, JPEG or GIFF formats may be used to represent the sub-picture images.

[0294] The display control sequence table 1146 contains a plurality of display control sequences 1146A, 1146B, 1146C, etc., which describe the chronological order in which control sequences occur. The display control sequence table contains information regarding how the information in the sub-picture unit is displayed. For example, words forming a single sub-picture unit may appear one-by-one or in groups over time, or their colors may change, which is useful when using sub-pictures to display karaoke information.

[0295] Each display control sequence (DCSQ) allows for commands changing the pixel data to be executed over a time period during which the sub-picture unit is displayed. The first field of a display control sequence is a display start time. This field describes the execution start time of the display control command included in the DCSQ having a relative PTS from the PTS described in the sub-picture packet with a sub-picture unit header. From the first video frame after the described execution start time, display control is started according to the DCSQ display control start time. The second field of each DCSQ is the address of the following display control sequence. This field describes the start address of the following DCSQ with the relative byte number from the first sub-picture unit. If no following DCSQ exists, this field describes the start address of this DCSQ with a relative byte number from the first sub-picture unit. Thereafter, the DCSQ contains one or more display control commands. These commands allow the attributes and display of the pixel data to be controlled and changed. The commands include a command to forcedly start the display of a sub-picture unit, irrespective of the on/off state of the sub-picture. For example, if a user uses a menu to turn off sub-pictures, it is possible to override the user's setting by setting this command. Another command starts the display of an updated sub-picture unit. This command must appear once in each DCSQ. There are commands to set the colors and contrast of the above-described four types of pixels including the two types of emphasis pixels, the pattern pixel, and the background pixel. A command exists which sets the rectangular area display and position of the pixel data making up the sub-picture. This command allows the setting of the upper and lower X and Y coordinates. There is also a command used to set the head address of the pixel data used for displaying. This command allows the head address of both the top and bottom field to be entered. In the extended fields, each head address of the pixel data for the top field and bottom field with the relative byte number from the head of the unit is used. The first pixel data represents the first pixel on the line. This command must be used at least in the first DCSQ, DCSQO.

[0296] There is a command which may appear in a DCSQ to change the color and contrast of the pixel data. This command is not to be used when the highlight information of the data search information is used. The command to change the color and contrast of pixel data includes pixel control data to control the color and contrast of the pixel data during display. The content of the control described in the pixel control data is executed for every video frame after the specified start time and continued until new pixel control data is encountered or a new sub-picture unit is encountered. The pixel control data includes line control information used to specify lines on which the same change is made. Also, pixel control information can be used to specify a position on a line at which the change is made. The line control command allows the changing of the start line number, the number of change points, and the change termination line

LINKPLAY EXHIBIT 1004
Page 1658 of 4822

number. The pixel control information includes the change start pixel number, and the new color and contrast for the emphasis pixels 1 and 2, the pattern pixels, and the background pixels. There is also a command which indicates the end of a display control sequence. Each DCSQ must end with this command.

[0297] Over a time period of video frames such as for a movie, there are many different sub-picture units which may be used. The sub-picture units are broken into sub-picture packs 1150, as illustrated in Figure 54. The sub-picture pack contains a pack header 972, as described above, a packet header which conforms to the MPEG system requirements, a sub-stream ID which indicates that the following data is sub-picture data, and the sub-picture data 1158 itself.

[0298] The sub-picture units such as sub-picture unit 1140 is divided into sub-picture packs 150, as illustrated in Figure 55. The last sub-picture pack 1050G contains padding to make it 2048 bytes in length.

[0299] Figure 56 illustrates the manner in which successive sub-picture units are displayed. When the time equals to the time of a PTS described in a new sub-picture packet, the sub-picture unit currently being displayed (1160) is cleared (1162) and when the time specified by the display control sequence of the next sub-picture unit occurs, that sub-picture is displayed (1164). Between 1160 and 1162, there is a dividing line between the sub-picture unit n and the sub-picture unit n+1. This dividing line occurs at the PTS described in the packet header of the sub-picture unit n+1. The actual display time for sub-unit n+1 is set forth in the DCSQ of sub-picture n+1.

[0300] The creation of the pixel data used for the sub-picture information may be performed using a computer to create bit maps of the information which is desired to be displayed. The process of creating bit-mapped information is well-known in the art.

[0301] Figures 40-56 emphasize the data structures used to encode the audio, video, and sub-picture information. However, by the disclosure of the data structures in Figures 40-56, one of ordinary skill in the art of MPEG encoding and in particular with the assistance of the MPEG system description ISO/IEC 13818-1, could format encoded audio, video, and sub-pictures into the data structures. Similarly, with the knowledge of the structures in which the data is stored, the decoding of the data structures to generate the video, audio, and sub-picture information can be performed by one of ordinary skill in the art.

[0302] An exemplary video decoder which constitutes part of the present invention includes an optical disc reader which reads data stored on the optical disc. The information which is read is parsed using conventional parsing techniques in order to decode the information. The video, audio, sub-picture, and PBI packs must all be decoded. The video may be decoded using commercially available MPEG decoders as are the audio packs decoded using conventional decoders. The sub-picture information is decoded by constructing sub-picture units from the sub-picture packs in an inverse manner as they were encoded. Specifically constructed hardware or a general purpose microprocessor programmed using software coding may be used to decode the sub-picture information. The playback information packs, as illustrated in Figure 45 contain data search information 996. The data search information is processed using a programmed microprocessor and is different from the audio, video, and sub-picture decoders. As the menu functions are also performed by a programmed microprocessor, the highlight information 1022 included within the data search information and all other menu functions are performed by the programmed microprocessor and not by the sub-picture decoder. Data structures, encoding and decoding process are described in Japanese Patent Applications 7-81298 (EP 737 009 A), and 7-85693 (EP 737 975 A).

[0303] The encoding system of the present invention allows the control of quality within pictures. This control is important as the people creating the final video product will be able to closely control the final product of the encoding system. Therefore, even though digital video encoding results in decoding artifacts which are not present in analog video, the final digital product will be quite acceptable. The teachings in any of the above sections are applicable to the other sections.

[0304] The determinations, calculations, and steps of the present invention may be conveniently implemented using a conventional general purpose digital computer programmed according to the teachings of the present invention, as will be apparent to those skilled in the computer art. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

[0305] The invention may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

[0306] The present invention includes a computer program product which is a storage medium including instructions which can be used to program a computer to perform a process of the invention. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, or any type of media which is suitable for storing electronic instructions. The present invention further includes a computer program product which is a storage medium including encoded data output by the present invention stored on any of the above described media suitable for storing electronic instructions or data.

LINKPLAY EXHIBIT 1004
Page 1659 of 4822

Claims

1. A method for processing video signals included in video frames to determine characteristics of the video frames, comprising the steps of :

   a) obtaining video signals included in plural video frames ;
   b) assigning plural quantization levels to one of said plural video frames such that different parts of said plural frames are assigned different quantization levels;
   c) pre-encoding said one of said plural video frames using said plural quantization levels to obtain plural bit-rates corresponding to each of said plural quantization levels ;
   d) determining bit-rates for each of said plural quantization levels resulting from said pre-encoding step ;
   e) repeating steps b-d sequentially for plural succeeding video frames ; and
   f) encoding a plurality of said plural video frames after determining the bit-rates for each of said plural video frames;
   g) determining a target number of bits for said plurality of said plural video frames ;
   h) assigning a first set of quantization levels for each frame in said plurality of said plural video frames using said target number of bits and the plural determined bit-rates;
   i) encoding one video frame of said plurality of said plural video frames into an encoded frame with a resultant number of bits ;
   j) estimating an encoded number of bits for a succeeding video frame ;
   k) adjusting the first set of quantization levels for said succeeding video frame based on said target number of bits, said bit-rates for succeeding video frames, the estimated encoded number of bits and the resultant number of bits ; and
   l) repeating steps i-k sequentially for plural succeeding video frames.

2. The method of Claim 1, wherein the step of obtaining video signals comprises obtaining said video signals from a video camera.

3. The method of Claim 1, further comprising the step of storing the plural determined bit-rates.

4. The method of Claim 1, wherein the step of assigning comprises assigning said plural quantization levels uniformly across each of said plural video frames.

5. The method of Claim 1, wherein the step of assigning comprises assigning two quantization levels uniformly across each of said plural video frames in a checkerboard pattern.

6. The method of Claim 1, wherein the step of assigning comprises assigning four quantization levels, q_1, q_2, q_3, q_4, uniformly across each of said plural video frames such that a first quantization level of each row, j, is $q_{j \bmod 4}$.

7. The method of Claim 1, wherein the step of determining a target number of bits comprises determining the target number of bits automatically.

8. The method of Claim 1, wherein the step of determining a target number of bits comprises determining the target number of bits through user-intervention.

9. The method of Claim 1, wherein the step of assigning further comprises assigning said first set of quantization levels based on an activity measure of plural macroblocks of corresponding video frames.

10. The method of Claim 1, wherein the step e) repeats for a majority of video frames in said plural video frames.

11. The method of Claim 1, wherein the step e) repeats for all video frames in said plural video frames.

12. A system for processing video signals included in video frames to determine characteristics of video frames, comprising :

    means for obtaining video signals included in plural video frames ;
    means for assigning plural quantization levels to a first plurality of said plural frames of the video signals such that different parts of said plural frames are assigned different quantization levels;

LINKPLAY EXHIBIT 1004
Page 1660 of 4822

means for pre-encoding each video frame in said first plurality of said plural video frames using said quantization levels to obtain plural bit-rates corresponding to each of said plural quantization levels ;
means for determining said plural bit-rates for each of said plural quantization levels for each video frame in said first plurality of said plural video frames generated by said means for pre-encoding ; and
means for encoding a second plurality of said plural video frames after determining said plural bit-rates for each video frame in said first plurality of said plural frames;
means for determining a target number of bits for said second plurality of said plural video frames ;
and
means for assigning a first set of quantization levels for each frame in said second plurality of said plural video frames using said target number of bits and the plural determined bit-rates;
means for encoding one video frame of said second plurality of said plural video frames into an encoded video frame with a resultant number of bits ;
means for estimating an encoded number of bits for a succeeding video frame ; and
means for adjusting the first set of quantization levels for said succeeding video frame bases on said target number of bits, said bit-rates and the estimated encoded number of bits for succeeding video frame and the resultant number of bits of said one of said second plurality of frames.

13. The system of Claim 12, wherein said means for obtaining comprises means for obtaining said video signals from a video camera.

14. The system of Claim 12, further comprising means for storing the determined bit-rates.

15. The system of Claim 12, wherein said means for assigning comprises means for assigning said plural quantization levels uniformly across each of said plural video frames.

16. The system of Claim 12, wherein said means for assigning comprises means for assigning two quantization levels across each of said plural frames in a checkerboard pattern.

17. The system of Claim 12, wherein said means for assigning comprises means for assigning four quantization levels, q_1, q_2, q_3, q_4, uniformly across each of said plural video frames such that a first quantization level of each row, j, is $q_{j \bmod 4}$.

18. The system of Claim 14, wherein the means for determining a target number of bits comprises means for determining the target number of bits automatically.

19. The system of Claim 14, wherein the means for determining a target number of bits comprises means for determining the target number of bits through user-intervention.

20. The system of Claim 14, wherein the means for assigning further comprises means for assigning said first set of quantization levels based on an activity measure of plural marcroblocks of corresponding video frames.

21. The system of Claim 12, wherein the means for assigning comprises means for assigning plural quantization levels to a majority of said plural video frames.

22. The system of Claim 12, wherein the means for assigning comprises means for assigning plural quantization levels to all of said plural video frames.

Patentansprüche

1. Verfahren zum Verarbeiten von Videosignalen, die in Videobildern beinhaltet sind, um Eigenschaften der Videobilder zu bestimmen, die folgenden Schritte aufweisend:

a) Erlangen von Videosignalen, die in mehreren Videobildern beinhaltet sind;
b) Zuordnen von mehreren Quantisierungsebenen einem von mehreren Videobildern derart, dass unterschiedliche Teile der mehreren Bilder unterschiedlichen Quantisierungsebenen zugeordnet sind;
c) Vorcodieren des einen der mehreren Videobilder unter Verwendung der mehreren Quantisierungsebenen zum Erlangen mehrerer Bitraten entsprechend jedem der mehreren Quantisierungsebenen;

LINKPLAY EXHIBIT 1004
Page 1661 of 4822

d) Bestimmen von Bitraten für jede der mehreren Quantisierungsebenen, die aus dem Vorcodierungsschritt resultieren;
e) Wiederholen der Schritte b-d sequenziell für mehrere aufeinanderfolgende Videobilder; und
f) Codieren einer Mehrzahl der mehreren Videobilder nach Bestimmen der Bitraten für jedes der mehreren Videobilder;
g) Bestimmen einer Zielanzahl von Bits für die Mehrzahl der mehreren Videobilder;
h) Zuordnen eines ersten Satzes aus Quantisierungsebenen für jedes Bild in der Mehrzahl der mehreren Videobilder unter Verwendung der Zielanzahl von Bits und der mehreren bestimmten Bitraten;
i) Codieren eines Videobilds der Mehrzahl der mehreren Videobilder in ein codiertes Bild mit einer resultierenden Anzahl von Bits;
j) Schätzen einer codierten Anzahl von Bits für ein nachfolgendes Videobild;
k) Einstellen des ersten Satzes Quantisierungsebenen für das nachfolgende Videobild basierend auf der Zielanzahl von Bits, den Bitraten für nachfolgende Videobilder, der geschätzten codierten Anzahl von Bits und der resultierenden Anzahl von Bits; und
l) Wiederholen der Schritte i-k sequenziell für mehrere aufeinanderfolgende Videobilder.

2. Verfahren gemäß Anspruch 1, wobei der Schritt zum Erlangen von Videosignalen Erlangen der Videosignale von einer Videokamera umfasst.

3. Verfahren gemäß Anspruch 1, ferner umfassend den Schritt zum Speichern der mehreren bestimmten Bitraten.

4. Verfahren gemäß Anspruch 1, wobei der Schritt zum Zuordnen Zuordnen der mehreren Quantisierungsebenen einheitlich über jedes der mehreren Videobilder umfasst.

5. Verfahren gemäß Anspruch 1, wobei der Schritt zum Zuordnen Zuordnen von zwei Quantisierungsebenen einheitlich über jedes der mehreren Videobilder in einem Schachbrettmuster umfasst.

6. Verfahren gemäß Anspruch 1, wobei der Schritt zum Zuordnen Zuordnen von vier Quantisierungsebenen q_1, q_2, q_3, q_4 einheitlich über jedes der mehreren Videobilder derart umfasst, das eine erste Quantisierungsebene von jeder Reihe j $q_{j \bmod 4}$ ist.

7. Verfahren gemäß Anspruch 1, wobei der Schritt zum Bestimmen einer Zielanzahl von Bits automatisches Bestimmen der Zielanzahl von Bits umfasst.

8. Verfahren gemäß Anspruch 1, wobei der Schritt zum Bestimmen einer Zielanzahl von Bits Bestimmen der Zielanzahl von Bits durch Benutzerintervention umfasst.

9. Verfahren gemäß Anspruch 1, wobei der Schritt zum Zuordnen ferner Zuordnen des ersten Satzes Quantisierungsebenen basierend auf einer Aktivitätsmessung mehrerer Makroblöcke entsprechender Videobilder umfasst.

10. Verfahren gemäß Anspruch 1, wobei der Schritt e) sich für eine Mehrzahl von Videobildern in den mehreren Videobildern wiederholt.

11. Verfahren gemäß Anspruch 1, wobei der Schritt e) sich für alle Videobilder in den mehreren Videobildern wiederholt.

12. System zum Verarbeiten von Videosignalen, die in Videobildern beinhaltet sind, zum Bestimmen von Eigenschaften von Videobildern, aufweisend:

Mittel zum Erlangen von Videosignalen, die in mehreren Videobildern beinhaltet sind;
Mittel zum Zuordnen mehrerer Quantisierungsebenen einer ersten Mehrzahl der mehreren Bilder der Videosignale derart, dass unterschiedliche Teile der mehreren Bilder unterschiedlichen Quantisierungsebenen zugeordnet sind;
Mittel zum Vorcodieren jedes Videobilds in der ersten Mehrzahl der mehreren Videobilder unter Verwendung der Quantisierungsebenen zum Erlangen mehrerer Bitraten entsprechend jeder der mehreren Quantisierungsebenen;
Mittel zum Bestimmen der mehreren Bitraten für jede der mehreren Quantisierungsebenen für jedes Videobild in der ersten Mehrzahl der mehreren Videobilder, die durch die Mittel zum Vorcodieren erzeugt werden; und
Mittel zum Codieren einer zweiten Mehrzahl der mehreren Videobilder nach Bestimmen der mehreren Bitraten

LINKPLAY EXHIBIT 1004
Page 1662 of 4822

für jedes Videobild in der ersten Mehrzahl der mehreren Bilder;
Mittel zum Bestimmen einer Zielanzahl aus Bits für die zweite Mehrzahl der mehreren Videobilder; und
Mittel zum Zuordnen eines ersten Satzes Quantisierungsebenen für jedes Bild in der zweiten Mehrzahl der mehreren Videobilder unter Verwendung der Zielanzahl von Bits und der mehreren bestimmten Bitraten;
Mittel zum Codieren eines Videobilds der zweiten Mehrzahl der mehreren Videobilder in ein codiertes Videobild mit einer resultierenden Anzahl von Bits;
Mittel zum Schätzen einer codierten Anzahl von Bits für ein nachfolgendes Videobild; und
Mittel zum Einstellen des ersten Satzes Quantisierungsebenen für das nachfolgende Videobild basierend auf der Zielanzahl von Bits, den Bitraten und der geschätzten codierten Anzahl von Bits für das nachfolgende Videobild und der resultierenden Anzahl von Bits des einen der zweiten Mehrzahl von Bildern.

13. System gemäß Anspruch 12, wobei das Mittel zum Erlangen Mittel zum Erlangen der Videosignale von einer Videokamera umfasst.

14. System gemäß Anspruch 12, wobei ferner umfassend Mittel zum Speichern der bestimmten Bitraten.

15. System gemäß Anspruch 12 wobei das Mittel zum Zuordnen Mittel zum Zuordnen der mehreren Quantisierungsebenen einheitlich über jedes der mehreren Videobilder umfasst.

16. System gemäß Anspruch 12, wobei das Mittel zum Zuordnen Mittel zum Zuordnen von zwei Quantisierungsebenen einheitlich über jedes der mehreren Videobilder in einem Schachbrettmuster umfasst.

17. System gemäß Anspruch 12, wobei das Mittel zum Zuordnen Mittel zum Zuordnen von vier Quantisierungsebenen q_1, q_2, q_3, q_4 einheitlich über jedes der mehreren Videobilder derart umfasst, das eine erste Quantisierungsebene von jeder Reihe j $q_{j \bmod 4}$ ist.

18. System gemäß Anspruch 14, wobei das Mittel zum Bestimmen einer Zielanzahl von Bits Mittel zum automatischen Bestimmen der Zielanzahl von Bits umfasst.

19. System gemäß Anspruch 14, wobei das Mittel zum Bestimmen einer Zielanzahl von Bits Mittel zum Bestimmen der Zielanzahl von Bits durch Benutzerintervention umfasst.

20. System gemäß Anspruch 14, wobei das Mittel zum Zuordnen ferner Mittel zum Zuordnen des ersten Satzes Quantisierungsebenen basierend auf einer Aktivitätsmessung mehrerer Makroblöcke entsprechender Videobilder umfasst.

21. System gemäß Anspruch 12, wobei das Mittel zum Zuordnen Mittel zum Zuordnen mehrerer Quantisierungsebenen einer Mehrheit der mehreren Videobilder umfasst.

22. System gemäß Anspruch 12, wobei das Mittel zum Zuordnen Mittel zum Zuordnen mehrerer Quantisierungsebenen allen der mehreren Videorahmen umfasst.

Revendications

1. Procédé pour traiter des signaux vidéo inclus dans des trames vidéo afin de déterminer des caractéristiques des trames vidéo, comprenant les étapes consistant à :

a) obtenir des signaux vidéo inclus dans plusieurs trames vidéo ;
b) affecter plusieurs niveaux de quantification à une desdites plusieurs trames vidéo de sorte que des parties différentes desdites plusieurs trames se voient affecter des niveaux de quantification différents ;
c) pré-encoder ladite une desdites plusieurs trames vidéo en utilisant lesdits plusieurs niveaux de quantification afin d'obtenir plusieurs débits binaires correspondant à chacun desdits plusieurs niveaux de quantification ;
d) déterminer des débits binaires pour chacun desdits plusieurs niveaux de quantification résultant de ladite étape de pré-encodage,
e) répéter les étapes b à d séquentiellement pour plusieurs trames vidéo suivantes ; et
f) encoder une pluralité desdites plusieurs trames vidéo après avoir déterminé les débits binaires pour chacune desdites plusieurs trames vidéo ;

LINKPLAY EXHIBIT 1004
Page 1663 of 4822

g) déterminer un nombre de bits cible pour ladite pluralité desdites plusieurs trames vidéo ;
h) affecter un premier ensemble de niveaux de quantification pour chaque trame de ladite pluralité desdites plusieurs trames vidéo en utilisant ledit nombre de bits cible et les plusieurs débits binaires déterminés ;
i) encoder une trame vidéo de ladite pluralité desdites plusieurs trames vidéo en une trame encodée avec un nombre de bits résultant ;
j) estimer un nombre de bits encodé pour une trame vidéo suivante ;
k) ajuster le premier ensemble de niveaux de quantification pour ladite trame vidéo suivante sur la base dudit nombre de bits cible, desdits débits binaires pour des trames vidéo suivantes, du nombre de bits encodé estimé et du nombre de bits résultant ; et
l) répéter les étapes i à k séquentiellement pour plusieurs trames vidéo suivantes.

2. Procédé selon la revendication 1, dans lequel l'étape consistant à obtenir des signaux vidéo comprend l'obtention desdits signaux vidéo à partir d'une caméra vidéo.

3. Procédé selon la revendication 1, comprenant en outre l'étape consistant à stocker les plusieurs débits binaires déterminés.

4. Procédé selon la revendication 1, dans lequel l'étape consistant à affecter comprend l'affectation desdits plusieurs niveaux de quantification uniformément à travers chacune desdites plusieurs trames vidéo.

5. Procédé selon la revendication 1, dans lequel l'étape consistant à affecter comprend l'affectation de deux niveaux de quantification uniformément à travers chacune desdites plusieurs trames vidéo selon un motif en damier.

6. Procédé selon la revendication 1, dans lequel l'étape consistant à affecter comprend l'affectation de quatre niveaux de quantification, q_1, q_2, q_3, q_4, uniformément à travers chacune desdites plusieurs trames vidéo de sorte qu'un premier niveau de quantification de chaque rangée, j, soit $q_{j \bmod 4}$.

7. Procédé selon la revendication 1, dans lequel l'étape consistant à déterminer un nombre de bits cible comprend la détermination du nombre de bits cible automatiquement.

8. Procédé selon la revendication 1, dans lequel l'étape consistant à déterminer un nombre de bits cible comprend la détermination du nombre de bits cible par l'intermédiaire d'une intervention de l'utilisateur.

9. Procédé selon la revendication 1, dans lequel l'étape consistant à affecter comprend en outre l'affectation dudit premier ensemble de niveaux de quantification sur la base d'une mesure d'activité de plusieurs macroblocs de trames vidéo correspondantes.

10. Procédé selon la revendication 1, dans lequel l'étape e) se répète pour une majorité de trames vidéo desdites plusieurs trames vidéo.

11. Procédé selon la revendication 1, dans lequel l'étape e) se répète pour toutes les trames vidéo desdites plusieurs trames vidéo.

12. Système pour traiter des signaux vidéo inclus dans des trames vidéo afin de déterminer des caractéristiques des trames vidéo, comprenant :

des moyens pour obtenir des signaux vidéo inclus dans plusieurs trames vidéo ;
des moyens pour affecter plusieurs niveaux de quantification à une première pluralité desdites plusieurs trames des signaux vidéo de sorte que des parties différentes desdites plusieurs trames se voient affecter des niveaux de quantification différents ;
des moyens pour pré-encoder chaque trame vidéo de ladite première pluralité desdites plusieurs trames vidéo en utilisant lesdits niveaux de quantification afin d'obtenir plusieurs débits binaires correspondant à chacun desdits plusieurs niveaux de quantification ;
des moyens pour déterminer lesdits plusieurs débits binaires pour chacun desdits plusieurs niveaux de quantification pour chaque trame vidéo de ladite première pluralité desdites plusieurs trames vidéo générée par lesdits moyens pour pré-encoder ; et
des moyens pour encoder une seconde pluralité desdites plusieurs trames vidéo après avoir déterminé lesdits plusieurs débits binaires pour chaque trame vidéo de ladite première pluralité desdites plusieurs trames ;

LINKPLAY EXHIBIT 1004
Page 1664 of 4822

des moyens pour déterminer un nombre de bits cible pour ladite seconde pluralité desdites plusieurs trames vidéo ; et
des moyens pour affecter un premier ensemble de niveaux de quantification pour chaque trame de ladite seconde pluralité desdites plusieurs trames vidéo en utilisant ledit nombre de bits cible et les plusieurs débits binaires déterminés ;
des moyens pour encoder une trame vidéo de ladite seconde pluralité desdites plusieurs trames vidéo en une trame vidéo encodée avec un nombre de bits résultant ;
des moyens pour estimer un nombre de bits encodé pour une trame vidéo suivante ; et
des moyens pour ajuster le premier ensemble de niveaux de quantification pour ladite trame vidéo suivante sur la base dudit nombre de bits cible, desdits débits binaires et du nombre de bits encodé estimé pour la trame vidéo suivante et du nombre de bits résultant de ladite une de ladite seconde pluralité de trames.

13. Système selon la revendication 12, dans lequel lesdits moyens pour obtenir comprennent des moyens pour obtenir lesdits signaux vidéo à partir d'une caméra vidéo.

14. Système selon la revendication 12, comprenant en outre des moyens pour stocker les débits binaires déterminés.

15. Système selon la revendication 12, dans lequel lesdits moyens pour affecter comprennent des moyens pour affecter lesdits plusieurs niveaux de quantification uniformément à travers chacune desdites plusieurs trames vidéo.

16. Système selon la revendication 12, dans lequel lesdits moyens pour affecter comprennent des moyens pour affecter deux niveaux de quantification à travers chacune desdites plusieurs trames selon un motif en damier.

17. Système selon la revendication 12, dans lequel lesdits moyens pour affecter comprennent des moyens pour affecter quatre niveaux de quantification, q_1, q_2, q_3, q_4, uniformément à travers chacune desdites plusieurs trames vidéo de sorte qu'un premier niveau de quantification de chaque rangée, j, soit $q_{j \bmod 4}$.

18. Système selon la revendication 14, dans lequel les moyens pour déterminer un nombre de bits cible comprennent des moyens pour déterminer le nombre de bits cible automatiquement.

19. Système selon la revendication 14, dans lequel les moyens pour déterminer un nombre de bits cible comprennent des moyens pour déterminer le nombre de bits cible par l'intermédiaire d'une intervention d'un utilisateur.

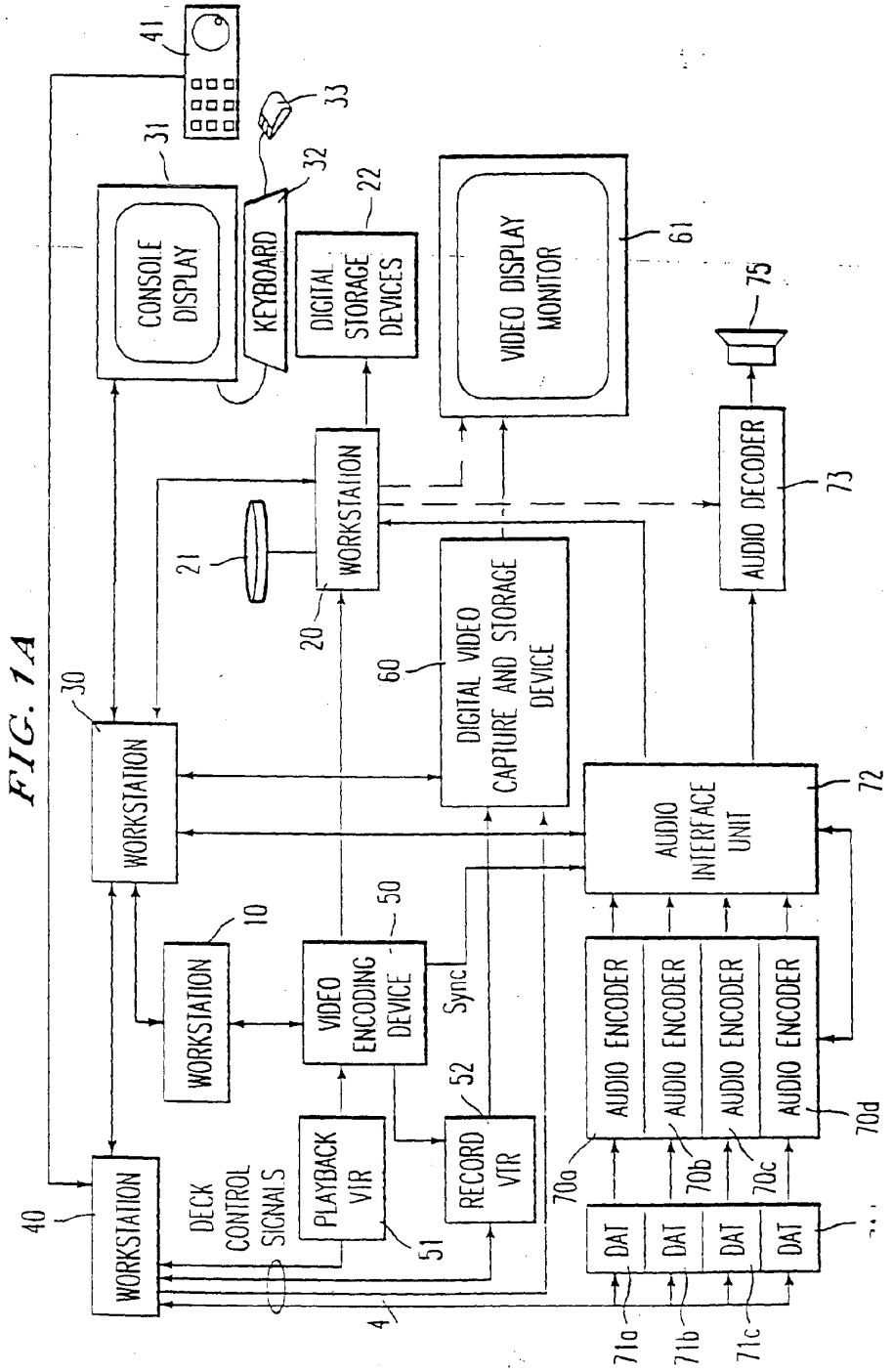
20. Système selon la revendication 14, dans lequel les moyens pour affecter comprennent en outre des moyens pour affecter ledit premier ensemble de niveaux de quantification sur la base d'une mesure d'activité de plusieurs macro-blocs de trames vidéo correspondantes.

21. Système selon la revendication 12, dans lequel les moyens pour affecter comprennent des moyens pour affecter plusieurs niveaux de quantification à une majorité desdites plusieurs trames vidéo.

22. Système selon la revendication 12, dans lequel les moyens pour affecter comprennent des moyens pour affecter plusieurs niveaux de quantification à la totalité desdites plusieurs trames vidéo.

LINKPLAY EXHIBIT 1004
Page 1665 of 4822

FIG. 1A



LINKPLAY EXHIBIT 1004
Page 1666 of 4822

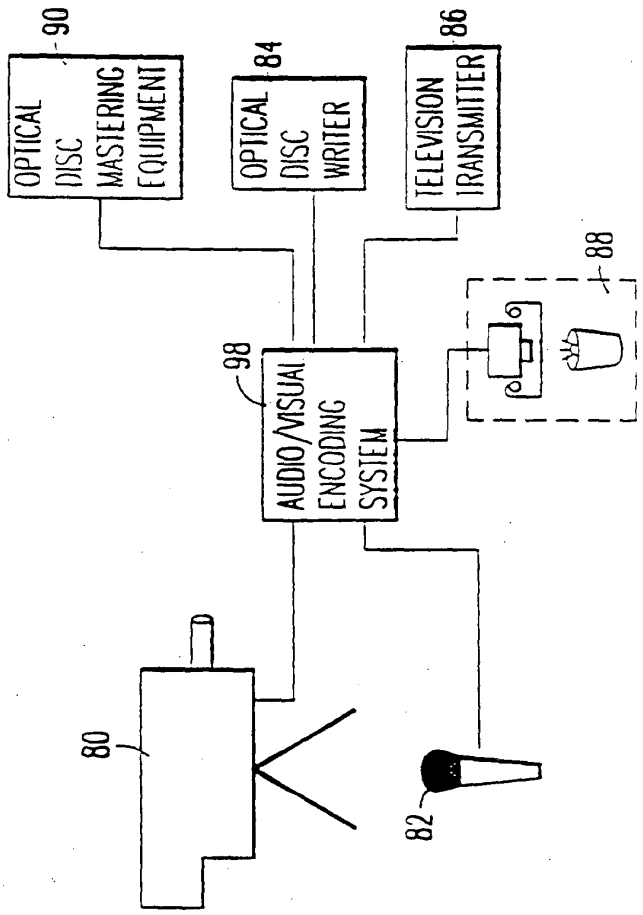
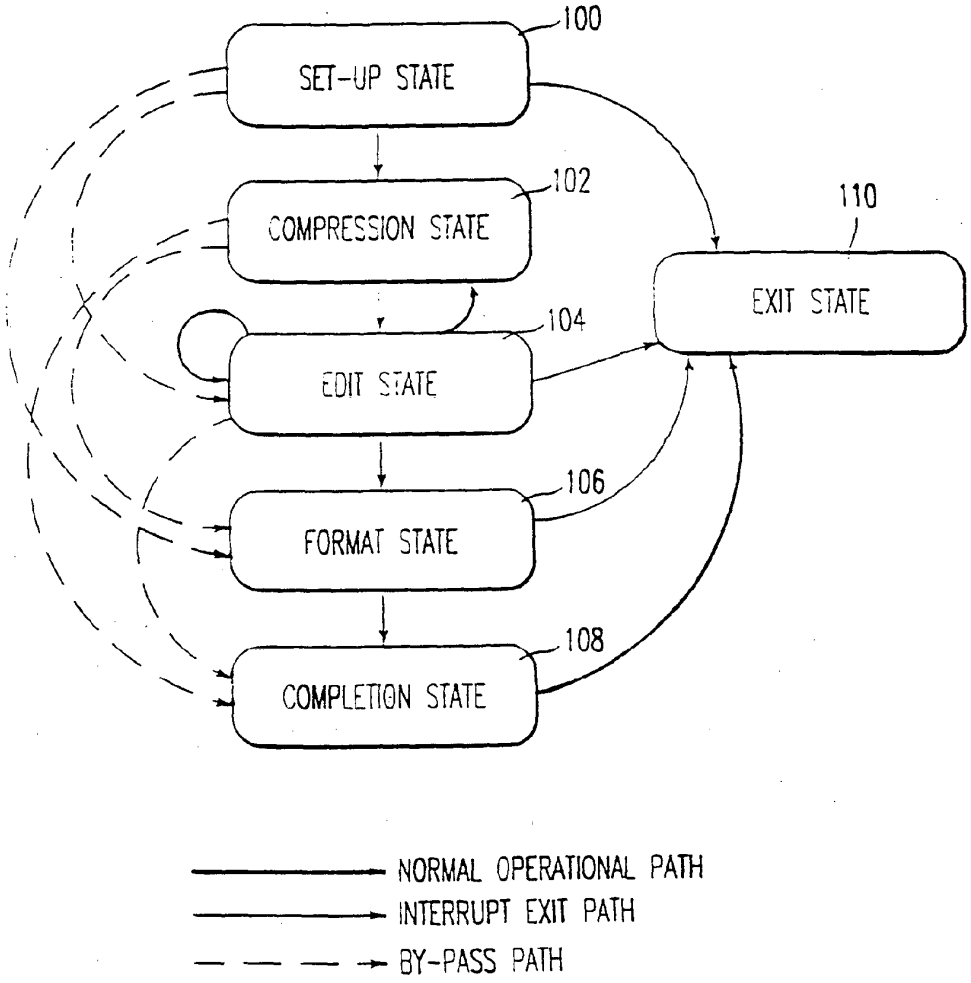


FIG. 1B

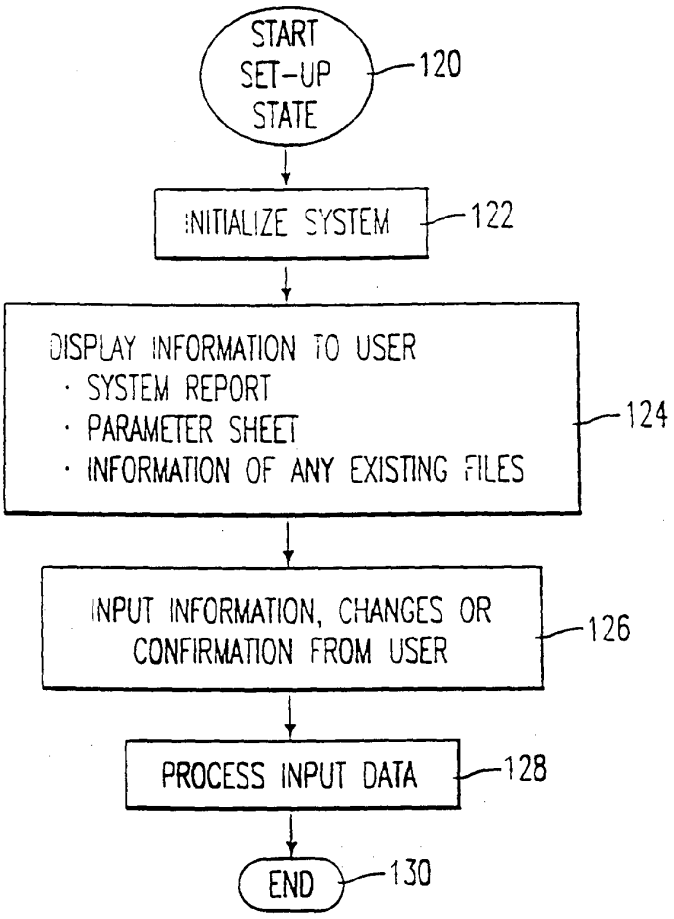
LINKPLAY EXHIBIT 1004
Page 1667 of 4822

FIG. 2



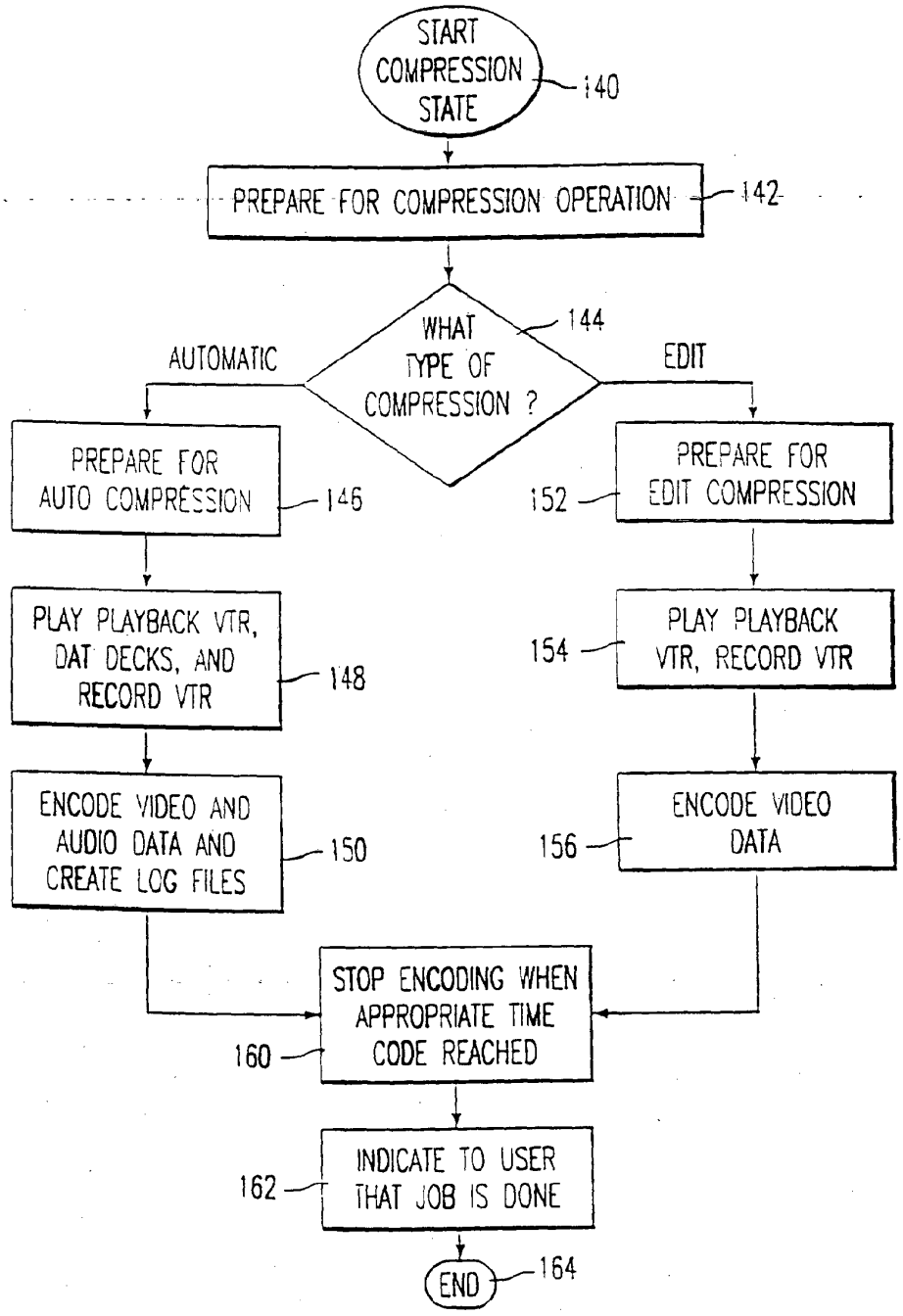
LINKPLAY EXHIBIT 1004
Page 1668 of 4822

FIG. 3



LINKPLAY EXHIBIT 1004
Page 1669 of 4822

FIG. 4



LINKPLAY EXHIBIT 1004
Page 1670 of 4822

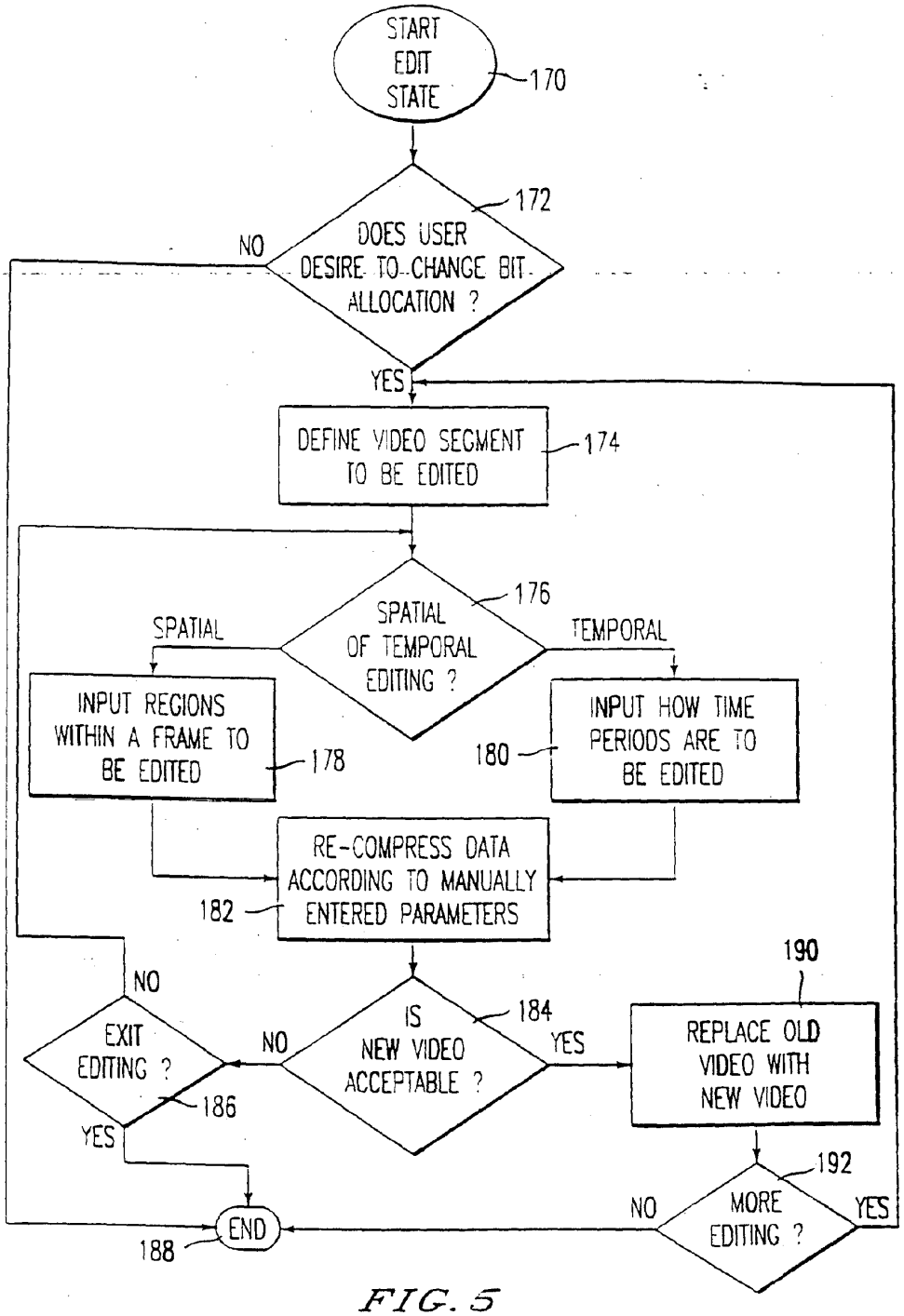
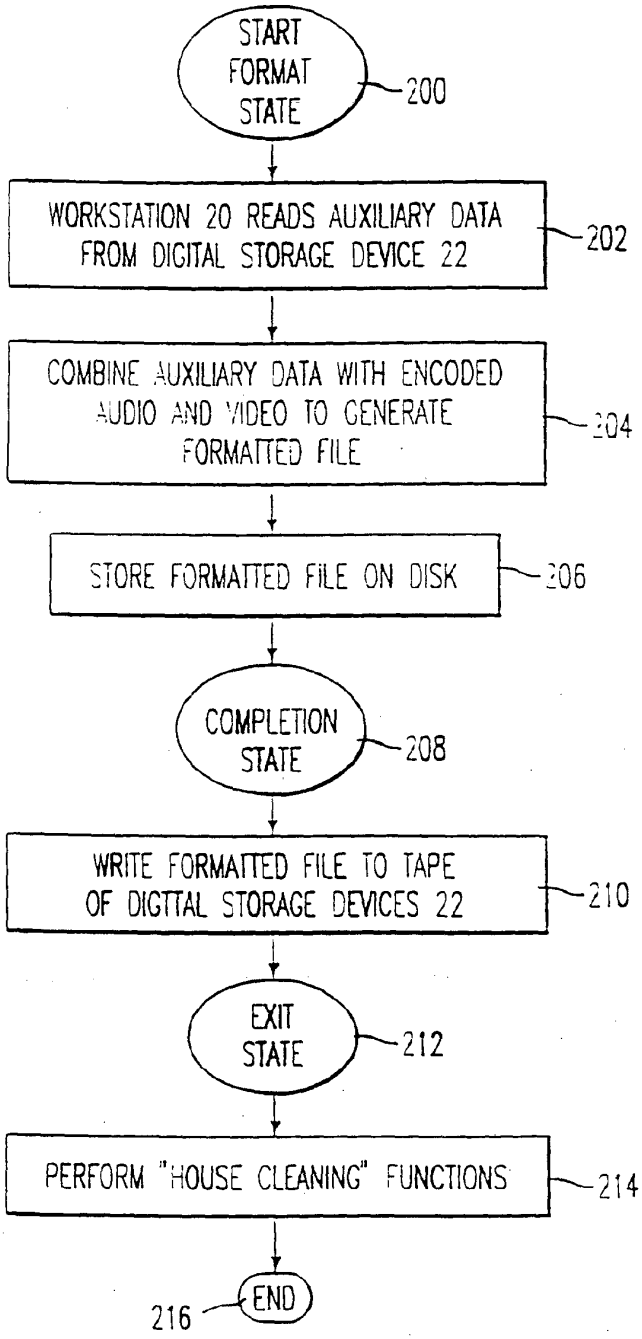

START EDIT STATE
170
172
DOES USER DESIRE TO CHANGE BIT ALLOCATION ?
NO
YES
DEFINE VIDEO SEGMENT TO BE EDITED
174
176
SPATIAL OF TEMPORAL EDITING ?
SPATIAL
TEMPORAL
INPUT REGIONS WITHIN A FRAME TO BE EDITED
178
180
INPUT HOW TIME PERIODS ARE TO BE EDITED
RE-COMPRESS DATA ACCORDING TO MANUALLY ENTERED PARAMETERS
182
190
IS NEW VIDEO ACCEPTABLE ?
184
NO
YES
REPLACE OLD VIDEO WITH NEW VIDEO
EXIT EDITING ?
186
NO
YES
192
MORE EDITING ?
NO
YES
END
188

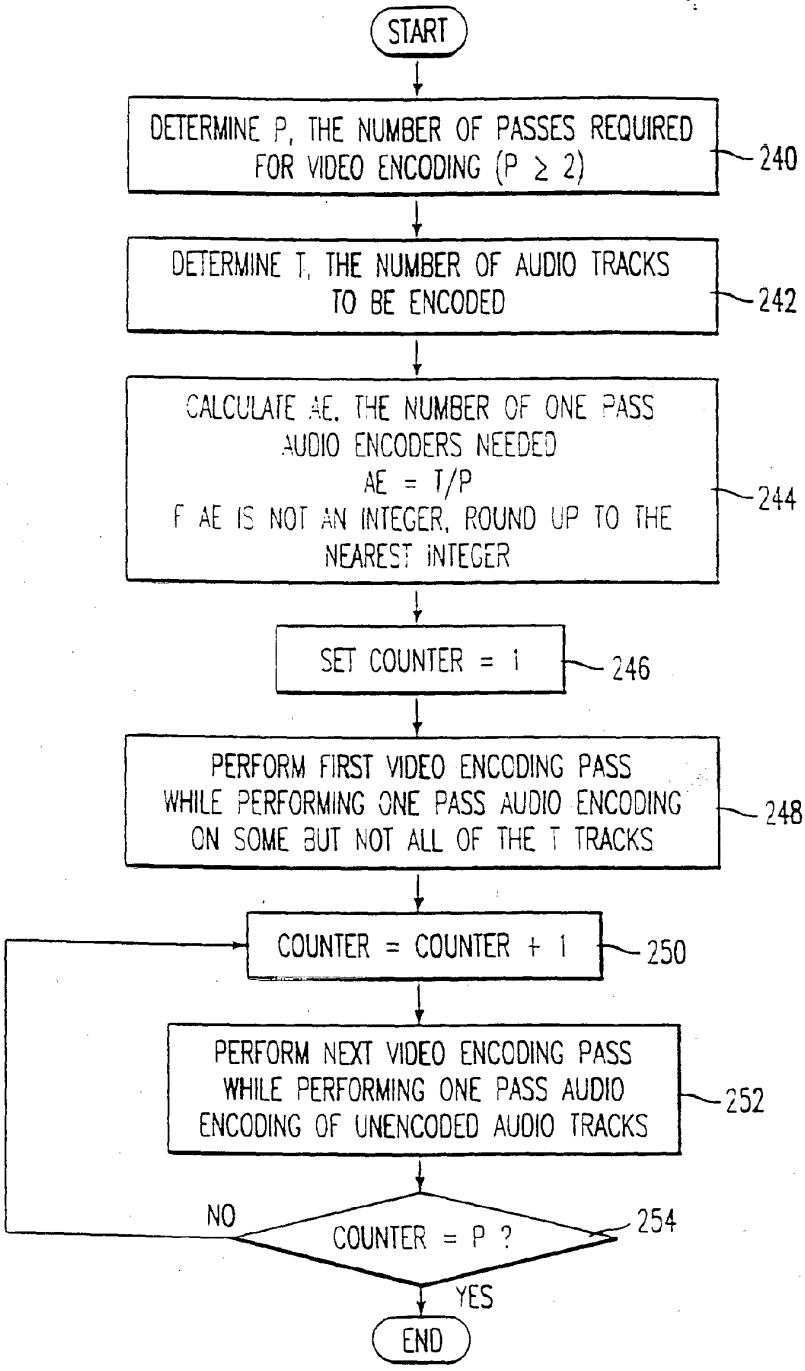
FIG. 5

FIG. 6



LINKPLAY EXHIBIT 1004
Page 1672 of 4822

FIG. 7



LINKPLAY EXHIBIT 1004
Page 1673 of 4822

	bits	bit field	contents
version	8	[31..28]	parameter format version
begin_tc	24	[23..0]	begin timecode
end_tc	24	[31..8]	end timecode
ntsc_pal	1	[7..7]	ntsc:0,pal:1
low_delay	1	[6..6]	lowdelay:1,else:0
enable_3_2	1	[5..5]	use 3 2 pull down; 1,else:0
MPEG1_2	1	[4..4]	MPEG1:0,MPEG2:1
MEmode	1	[3..3]	fixed:0,adaptive:1
Mmax	2	[2..1]	maximum size of M
cbr_vbr	1	[0..0]	CBR:0,VBR:1
MB_hsize	6	[31..26]	MB width of the picture
MB_vsize	6	[25..20]	MB height of the picture
initial_delay	15	[19..6]	initial buffer fullness
word_boundary2	5	[5..0]	reversed for word boundary
bit_rate	24	[31..8]	average bit rate
m	2	[7..6]	If M is variable, M is 0.
n	6	[5..0]	If N is variable, N is 0.
VBV_size	24	[31..8]	VBV buffer size in bits
word_boundary4	8	[7..0]	reserved for word boundary
number_of_coded_picture	32	[31..0]	number of total coded pictures

TO FIG. 8B

FIG. 8A

LINKPLAY EXHIBIT 1004
Page 1674 of 4822

FROM FIG. 8A

intra_q1[64]	8x64	[7..0]	intra q-matrix 1
intra_q2[64]	8x64	[7..0]	intra q-matrix 2
non_intra_q1[64]	8x64	[7..0]	non-intra q-matrix 1
non_intra_q2[64]	8x64	[7..0]	non-intra q-matrix 2 (zigzag scan order)
total_bits[2]	32x2	[31..0]	total number of the bitstream most significant 32 bits ([0]) least significant 32 bits ([1])
max_bit_rate	24	[31..8]	maximum bit rate for VBR mode
word_boundary5	8	[7..0]	reserved for word boundary
Ki	8	[31..24]	rate control constant
Kp	8	[23..16]	rate control constant
Kb	8	[15..8]	rate control constant

TO FIG. 8C

FIG. 8B

LINKPLAY EXHIBIT 1004
Page 1675 of 4822

FROM FIG. 8B

	bits	bit field	contents
rc_type	8	[7..0]	rate control type
Ki2	8	[31..24]	rate control constant
Kp2	8	[23..16]	rate control constant
Kb2	8	[15..8]	rate control constant
rc_type2	8	[7..0]	rate control type2
display_hsize	14	[31..18]	display horizontal size
display_vsize	14	[17..4]	display vertical size
word_boundary75	4	[3..0]	reserved for word boundary
run_mode	2	[31..30]	running mode 0:loop control 1:timecode control 2:timecode start and loop ctrl
loops	18	[29..12]	specified loops in loop control
pulldown_load_mode	2	[11..10]	the way to get pulldown phase 0:auto, 1:from log, 2:from data file
scene_change_adp_enable	1	[9..9]	1 : scene change adaptation on
sequence_header_mode	1	[8..8]	1 : before every GOP header
pan_scan_enable	1	[7..7]	1 : enable
MB_parameter_mode	1	[6..6]	1 : full, 0 : short
buffer_check_enable	2	[5..4]	0 : disable 1-3 : enable

TO FIG. 8D

FIG. 8C

EP 0 742 674 B1

LINKPLAY EXHIBIT 1004
Page 1676 of 4822

FROM FIG. 8C

frame_memory_channel	2	[3..2]	0 : CH1, 1 : CH2, 2 : through
timecode_check_enable	1	[1..1]	1 : enable
word_boundary76	1	[0..0]	reserved for word boundary
frame_memory_file_num	5	[31..27]	frame memory file number
initial_pulldown_th	18	[26..9]	initial pulldown threshold
word_boundary77	9	[8..0]	reserved for word boundary
scene_change_threshold	32	[31..0]	for scene change adaptation
pulldown_phase_threshold_file	8x256		file name for pulldown phase
MB_ctrl_file	8x256		file name for MB control
pan_scan_file	8x256		file name for pan-scan date
VBR_parameter_file	8x256		file name for VBR parameter
M_sequence_file	8x256		file name for M-sequence

FIG. 8D

LINKPLAY EXHIBIT 1004
Page 1677 of 4822

	bits	bit field	contents
time_code	24	[31..8]	timecode
picture_coding_type	2	[7..6]	picture coding type
top_field_first	1	[5..5]	top field first
repeat_first_field	1	[4..4]	repeat first field
display_repeat_field	1	[3..3]	repeat number of displaying picture
load_intra_qmat	1	[2..2]	load intra quantiser matrix
load_non_intra_qmat	1	[1..1]	load non-intra quantiser matrix
pan_scan	1	[0..0]	use picture_display_ extension()
input_order	32	[31..0]	input order of the coded picture
prefil_coeff	4	[31..28]	prefilter coefficients index
nr_coeff	2	[27..26]	NR coefficients index
mc_rom_type	2	[25..24]	mc_rom_type index
rlc_zero_type	2	[23..22]	rlc_zero_type index
vbv_occupancy	21	[21..1]	VBV occupancy Bn• in MPEG2 DIS
MEmode_type	1	[0..0]	interlace:0, non-interlace:1
forward_ae	23	[31..9]	ME error for forward reference
mean_qscale	8	[8..1]	mean qscale x 4
processing_type	1	[0..0]	4:2:2 → 4:2:0 filter type 0:field / 1:frame

FIG. 9A

TO FIG. 9B

EP 0 742 674 B1

LINKPLAY EXHIBIT 1004
Page 1678 of 4822

FROM FIG. 9A

backward_ae	23	[31..9]	NE error for backward reference
intra_q_matrix_type	1	[8..8]	intra quantizer matrix type
non_intra_q_matrix_type	1	[7..7]	non-intra quantizer matrix type
intra_dc_precision	2	[6..5]	intra DC precision
q_scale_type	1	[4..4]	qscale type
concealment_MVs	1	[3..3]	concealment MV
intra_vlc_format	1	[2..2]	intra VLC format
alternate_scan	1	[1..1]	alternate scan
progressive_frame	1	[0..0]	progressive frame
allocated_bit	21	[31..11]	allocated bit for the picture
temporal_reference	10	[10..1]	temporal reference
scene_change	1	[0..0]	1 : scene change
generated_bit_picture	21	[31..11]	generated bit for the picture
word boundary6	11	[10..0]	reserved for word boundary
bits_sequence_header	12	[31..20]	If exists for the picture, sequence header bit
GOP_number	20	[19..0]	GOP number
bits_GOP_header	12	[31..20]	If exists for the picture, GOP header bit

TO FIG. 9C

FIG. 9B

EP 0 742 674 B1

LINKPLAY EXHIBIT 1004
Page 1679 of 4822

FROM FIG. 9B

stuffing_size	20	[19..0]	zero stuffing size of the picture
bits_picture_header	12	[31..20]	picture header bit
bits_MB_header	20	[19.0]	slice header, MB header, DC, and EOB bit
word_boundary10	2	[31..30]	reserved for word boundary
mean_activity	30	[29..0]	mean activity
mc_error	32	[31..0]	MC error
coding_error	32	[31..0]	coding error
S1_I	32	[31..0]	rate control status
S2_I	32	[31..0]	rate control status
S1_P	32	[31..0]	rate control status
S2_P	32	[31..0]	rate control status
S1_B	32	[31..0]	rate control status
S2_B	32	[31..0]	rate control status
Qsum	32	[31..0]	sum of Q
reserved(4)	32	[31..0]	

FIG. 9C

LINKPLAY EXHIBIT 1004
Page 1680 of 4822

MB parameters	bits	bit field	contents
word_boundary0	2	[31..30]	reserved for word boundary
MB_type	5	[29..25]	macroblock type
MC_type	2	[24..23]	MC type
DCT_type	1	[22..22]	DCT type
header_length	8	[21..14]	MB header, DC bits
code_length	14	[13..0]	RLC bits
word_boundary2	1	[31..31]	reserved for word boundary
MB_act	15	[30..16]	MB activity
mc_error_MB	16	[15..0]	MC error
q_scale	5	[31..27]	quantizer scale index
reserved	27	[26..0]	reserved for word boundary

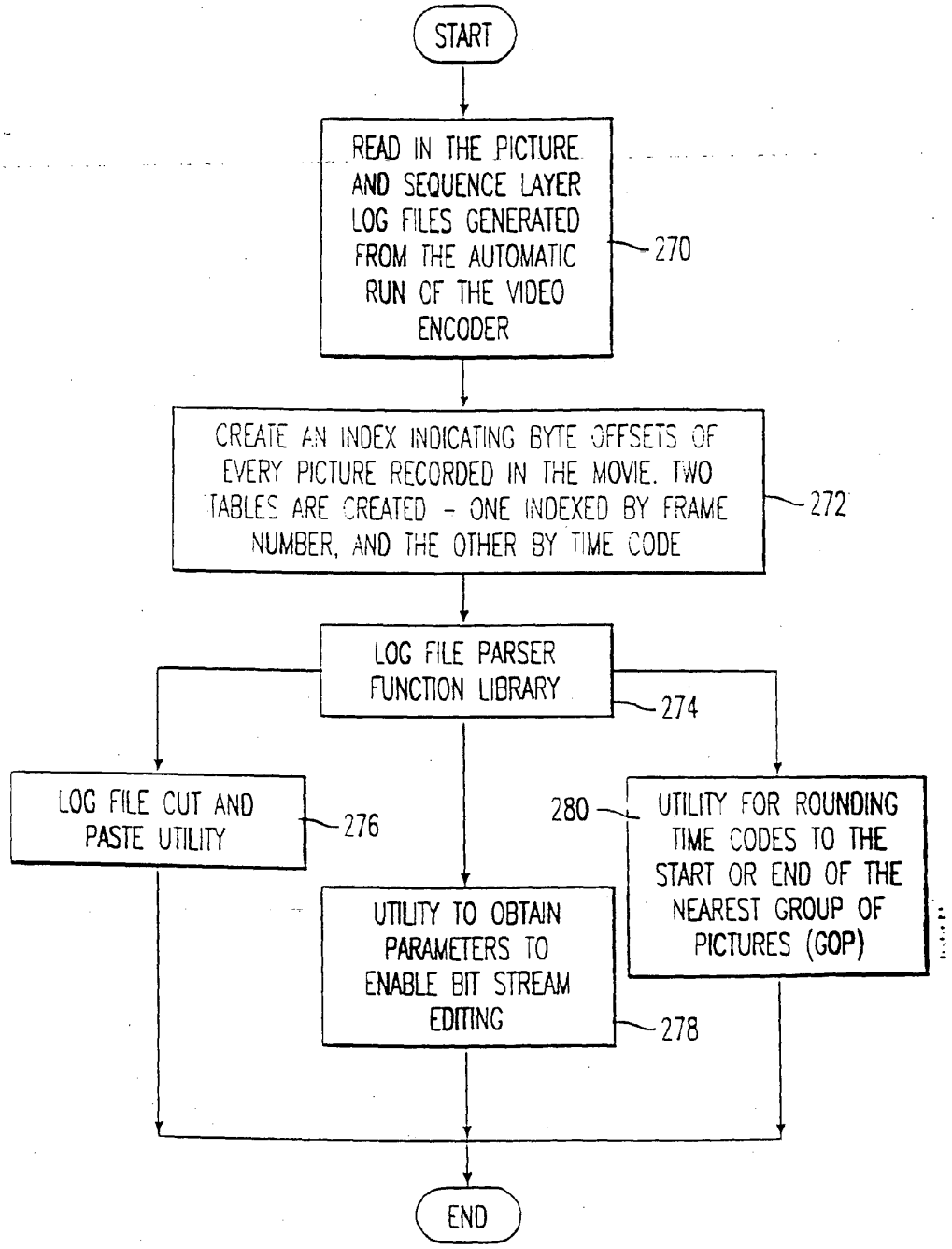
FIG. 10A

MB parameters	bits	bit field	contents
MB_type	5	[31..27]	macroblock type
header_length	8	[26..19]	MB header, DC bits
code_length	14	[18..5]	RLC bits
q_scale	5	[4..0]	quantizer scale index

FIG. 10B

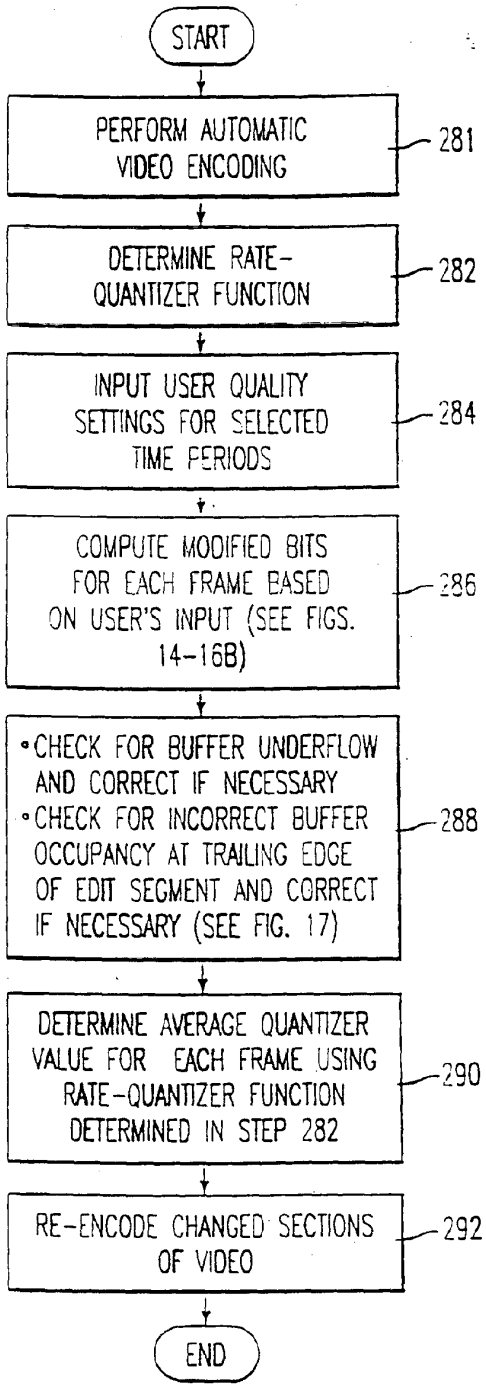
LINKPLAY EXHIBIT 1004
Page 1681 of 4822

FIG. 11



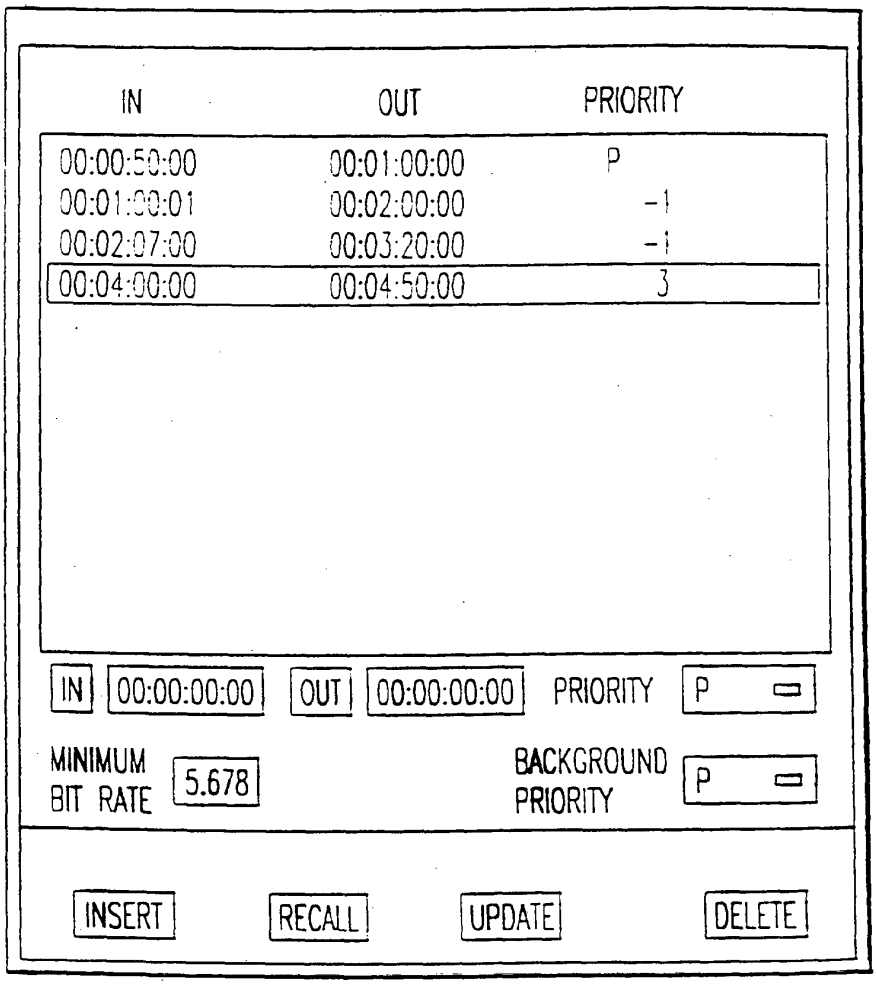
LINKPLAY EXHIBIT 1004
Page 1682 of 4822

FIG. 12



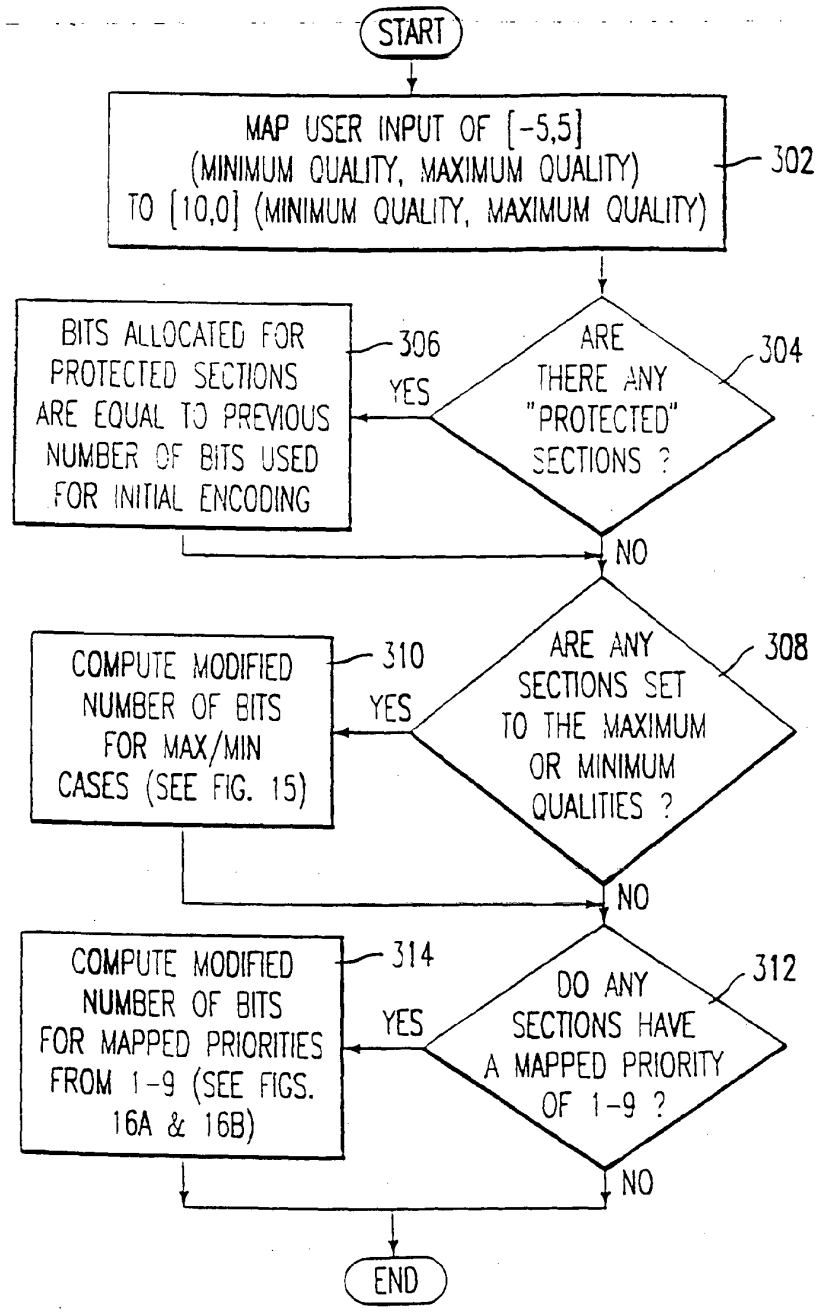
LINKPLAY EXHIBIT 1004
Page 1683 of 4822

FIG. 13



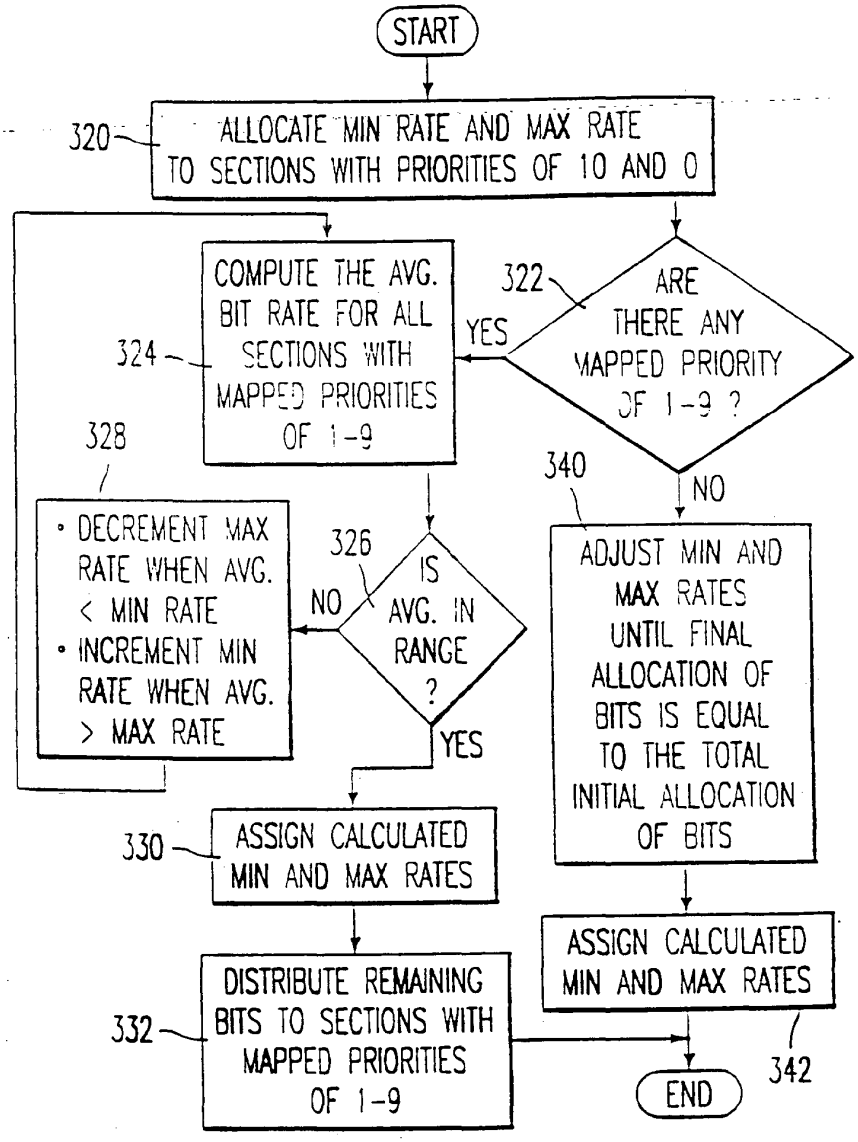
LINKPLAY EXHIBIT 1004
Page 1684 of 4822

FIG. 14



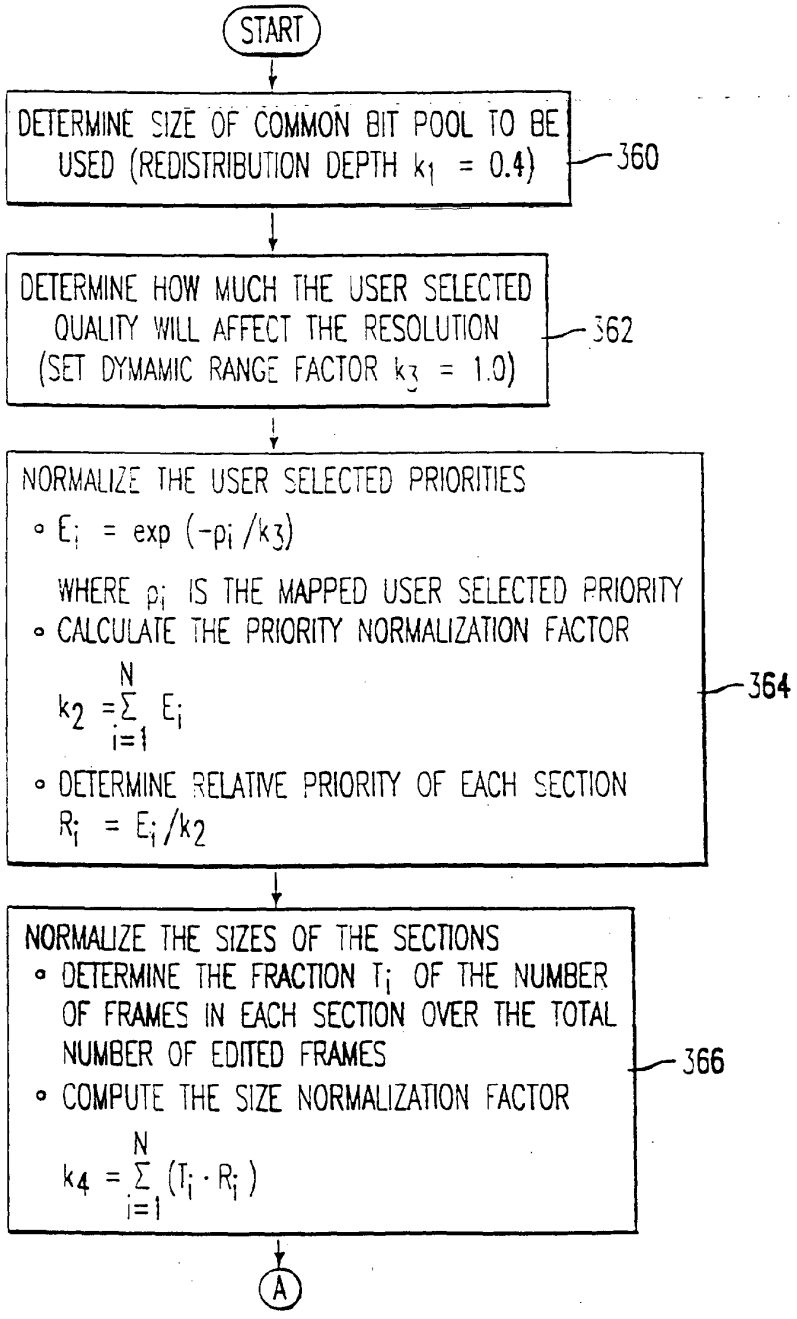
LINKPLAY EXHIBIT 1004
Page 1685 of 4822

FIG. 15



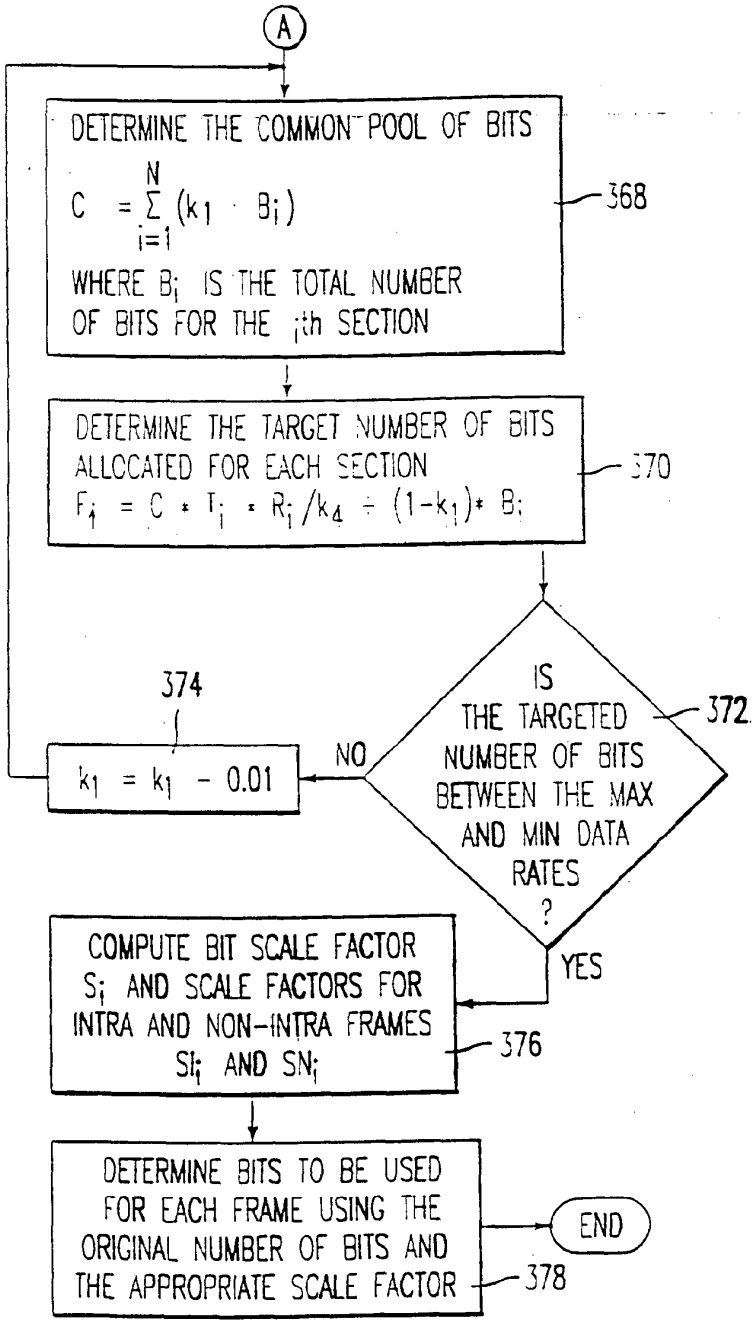
LINKPLAY EXHIBIT 1004
Page 1686 of 4822

FIG. 16A



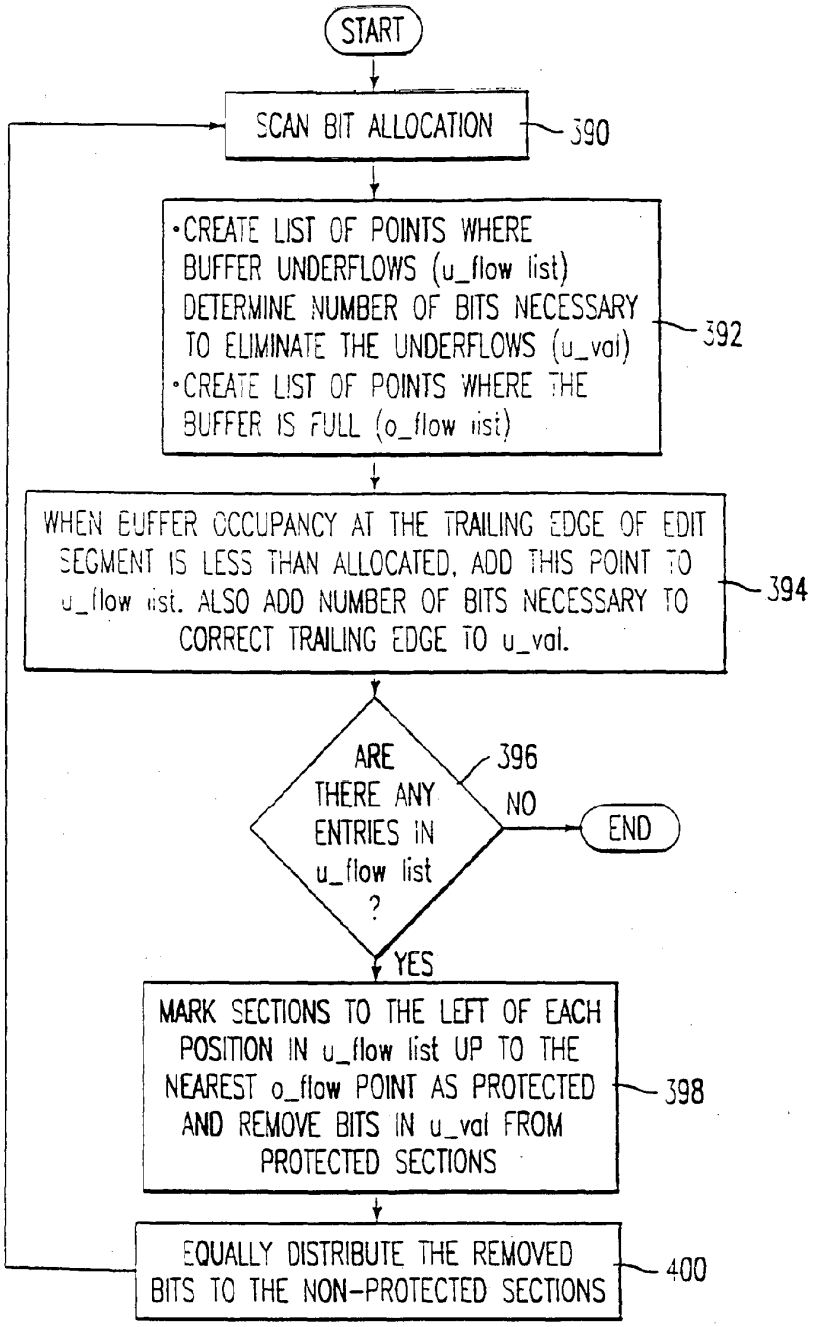
LINKPLAY EXHIBIT 1004
Page 1687 of 4822

FIG. 16B



LINKPLAY EXHIBIT 1004
Page 1688 of 4822

FIG. 17



LINKPLAY EXHIBIT 1004
Page 1689 of 4822

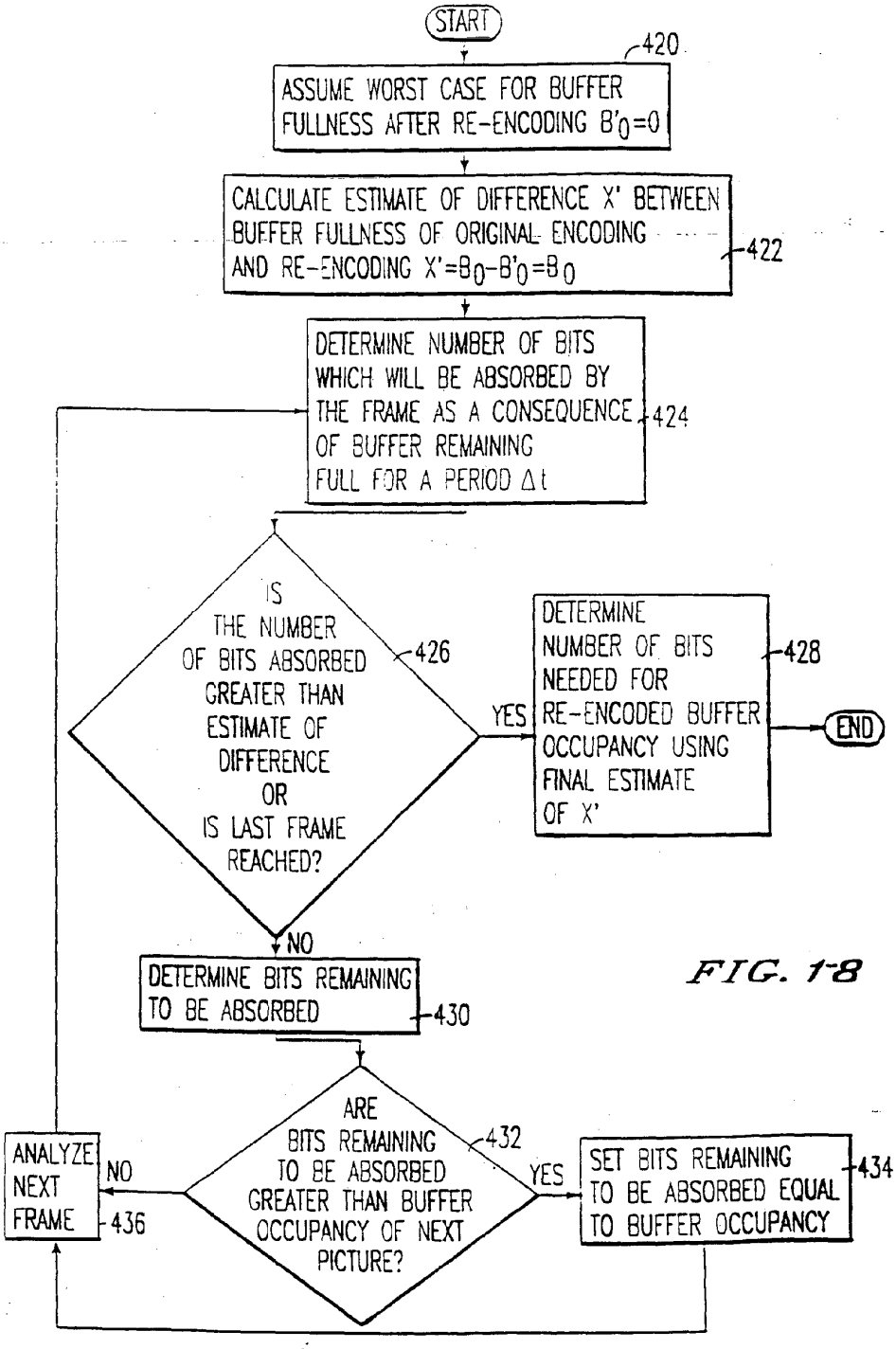


FIG. 18

LINKPLAY EXHIBIT 1004
Page 1690 of 4822

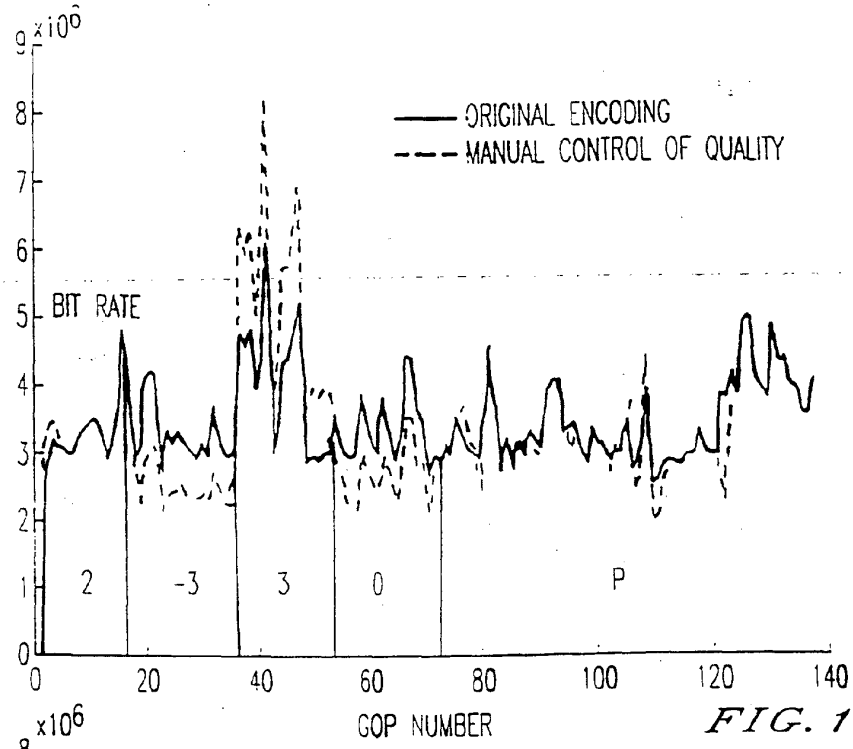


FIG. 19A

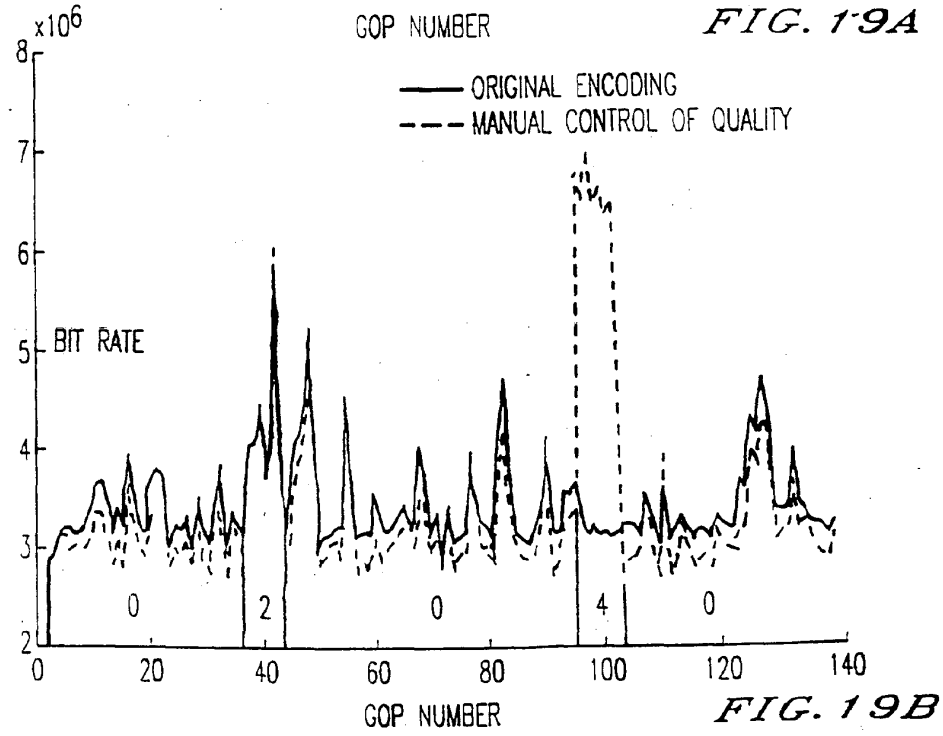


FIG. 19B

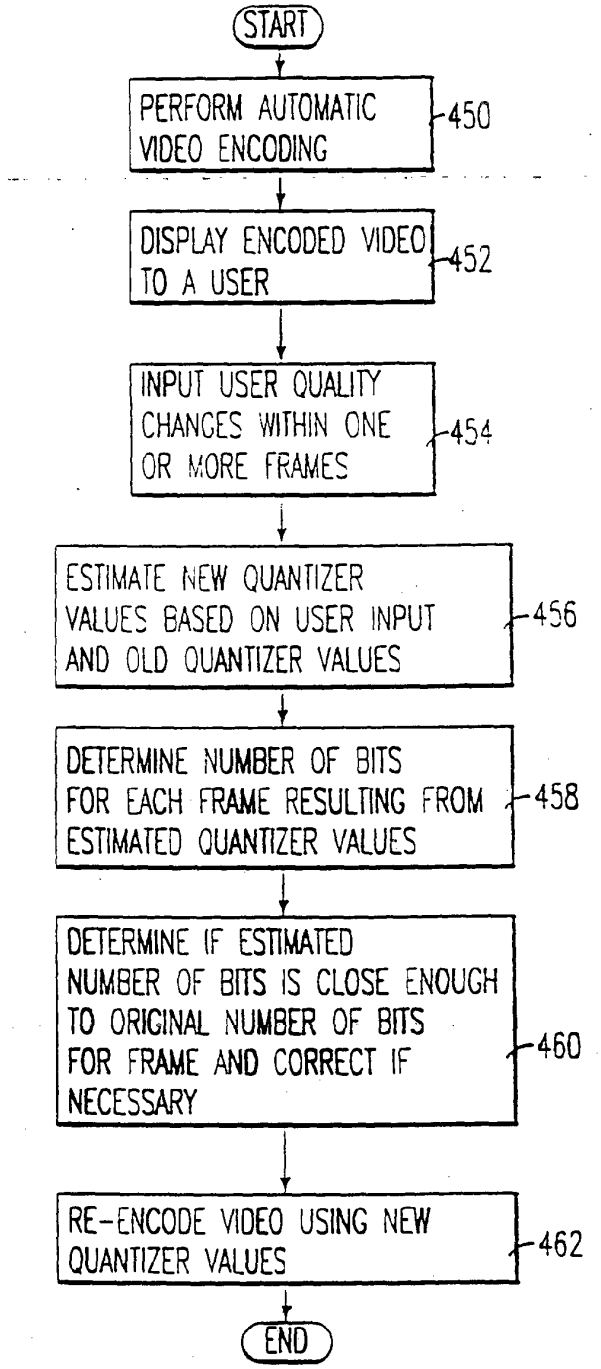


FIG. 20

LINKPLAY EXHIBIT 1004
Page 1692 of 4822

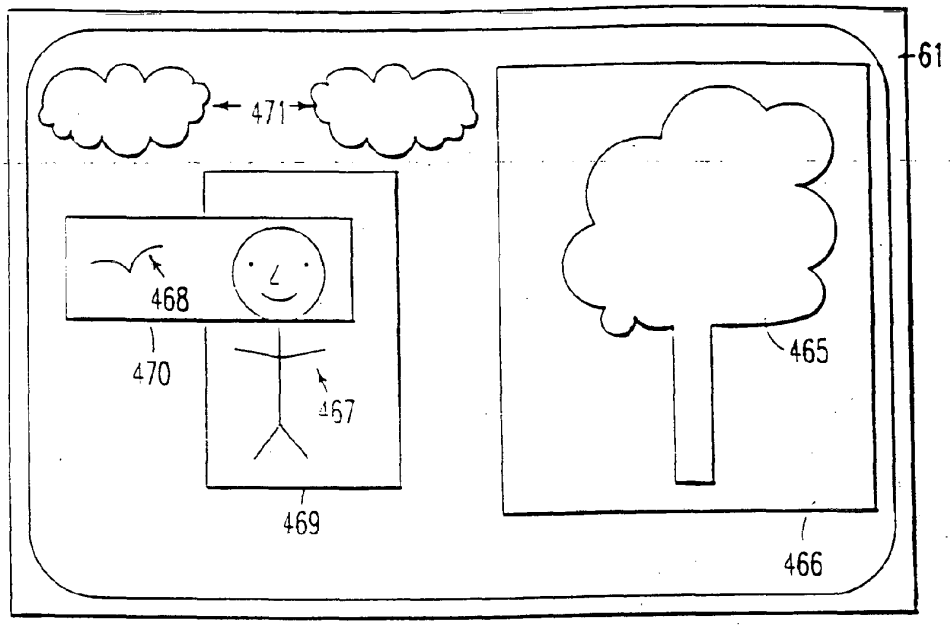


FIG. 21

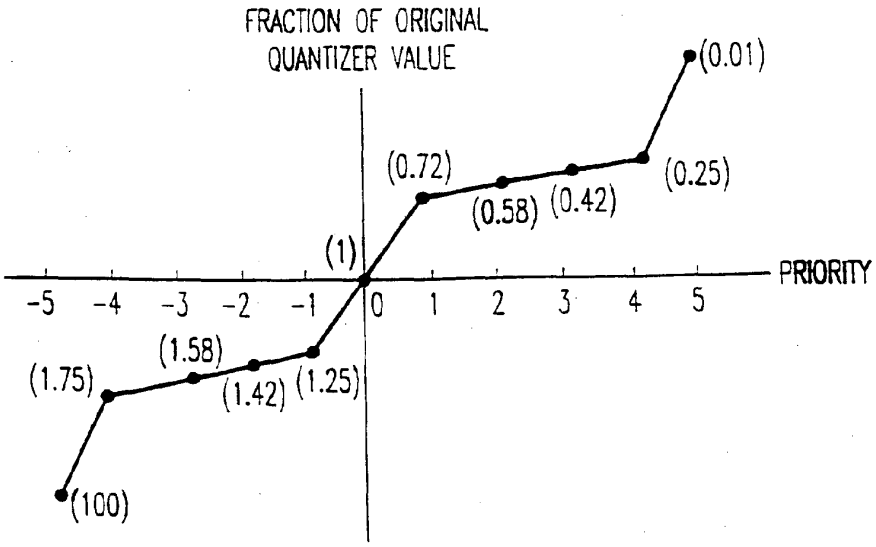


FIG. 22

LINKPLAY EXHIBIT 1004
Page 1693 of 4822

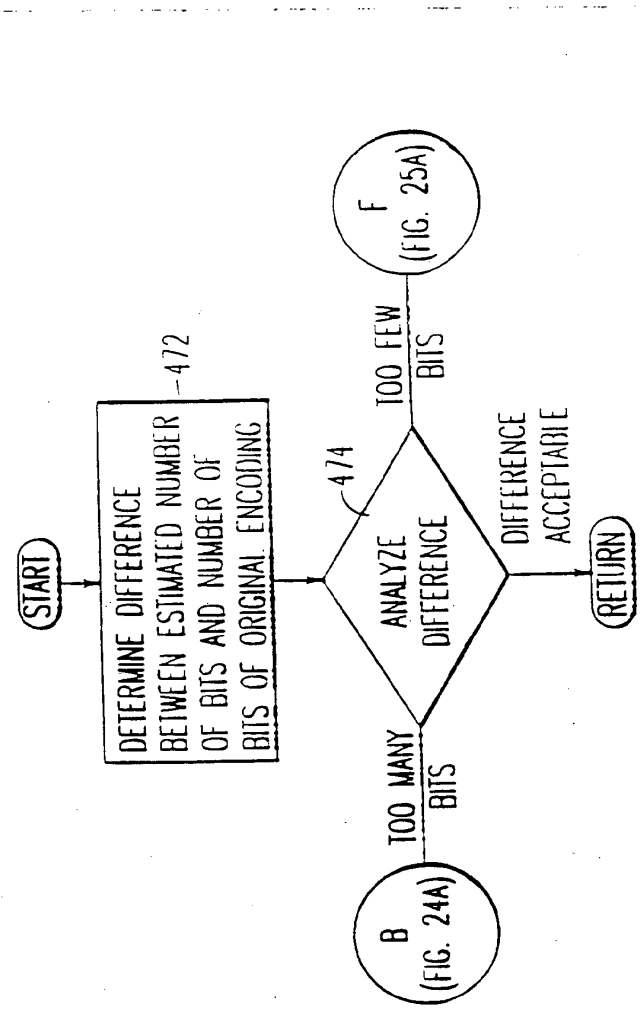

START
DETERMINE DIFFERENCE BETWEEN ESTIMATED NUMBER OF BITS AND NUMBER OF BITS OF ORIGINAL ENCODING
472
474
ANALYZE DIFFERENCE
TOO FEW BITS
F (FIG. 25A)
TOO MANY BITS
B (FIG. 24A)
DIFFERENCE ACCEPTABLE
RETURN

FIG. 23

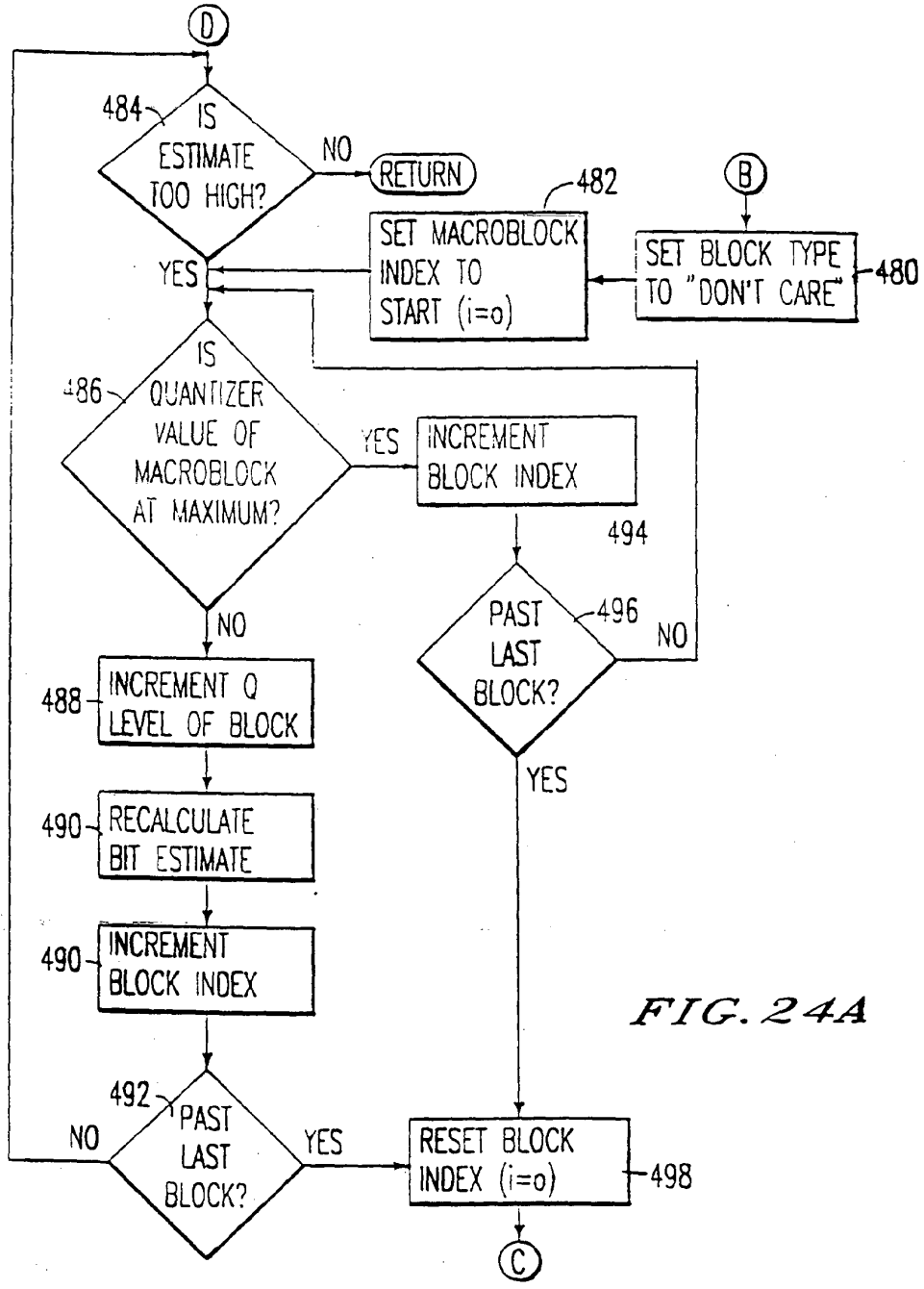

D
484
IS ESTIMATE TOO HIGH?
NO
RETURN
482
B
SET MACROBLOCK INDEX TO START (i=o)
SET BLOCK TYPE TO "DON'T CARE"
480
YES
IS QUANTIZER VALUE OF MACROBLOCK AT MAXIMUM?
486
YES
INCREMENT BLOCK INDEX
494
PAST LAST BLOCK?
496
NO
NO
INCREMENT Q LEVEL OF BLOCK
488
RECALCULATE BIT ESTIMATE
490
INCREMENT BLOCK INDEX
490
YES
PAST LAST BLOCK?
492
NO
YES
RESET BLOCK INDEX (i=o)
498
C

FIG. 24A

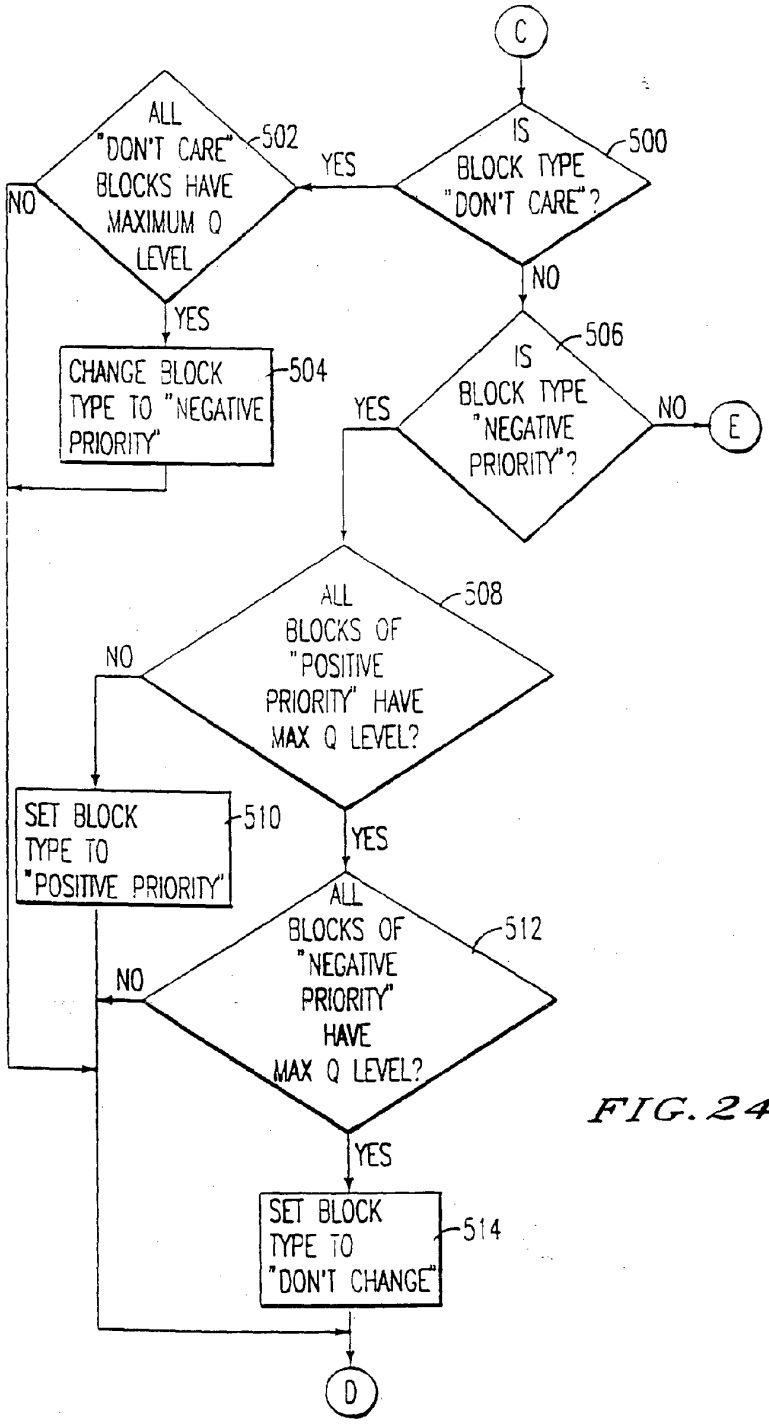
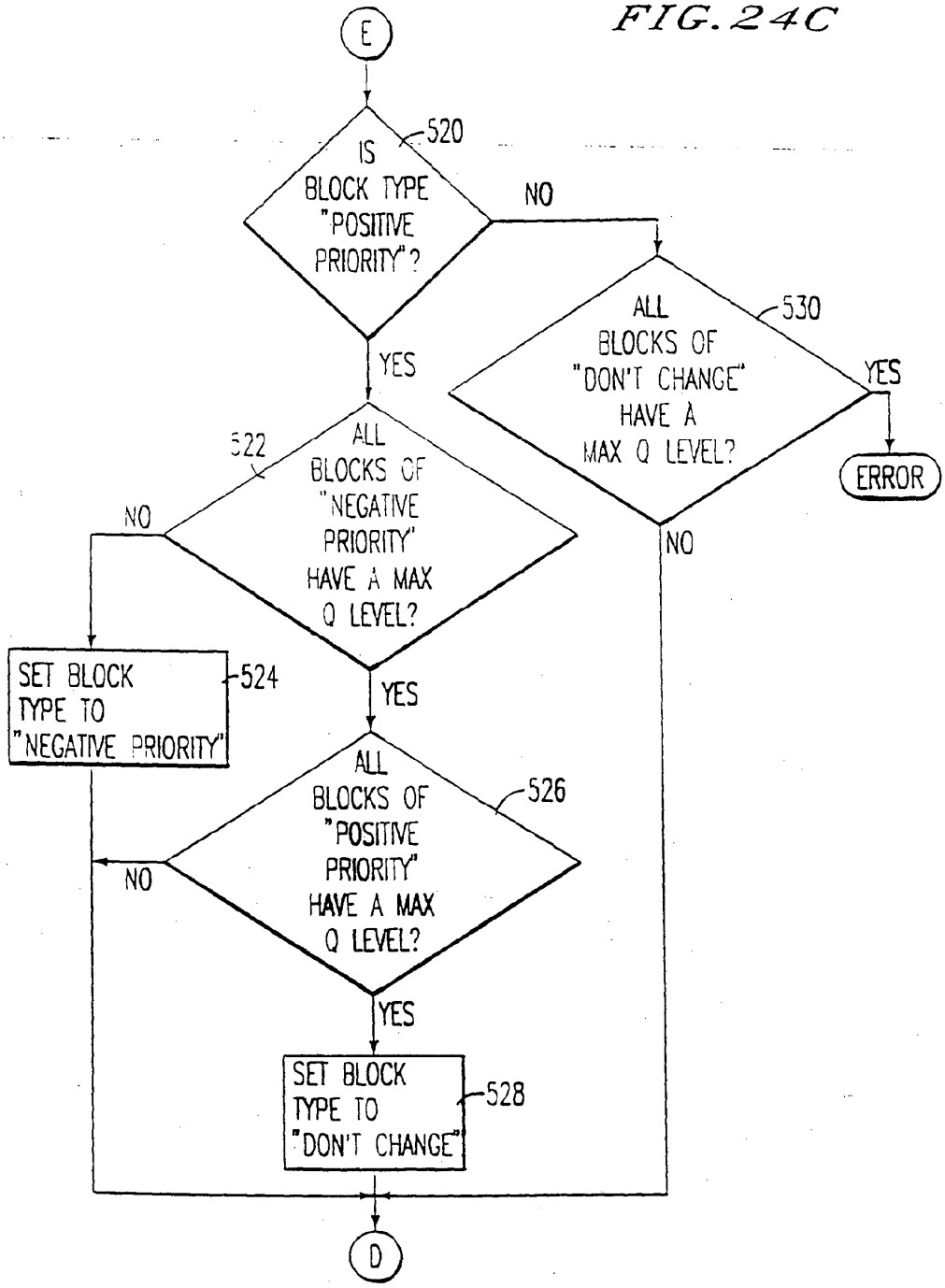


FIG. 24B

LINKPLAY EXHIBIT 1004
Page 1696 of 4822

FIG. 24C



LINKPLAY EXHIBIT 1004
Page 1697 of 4822

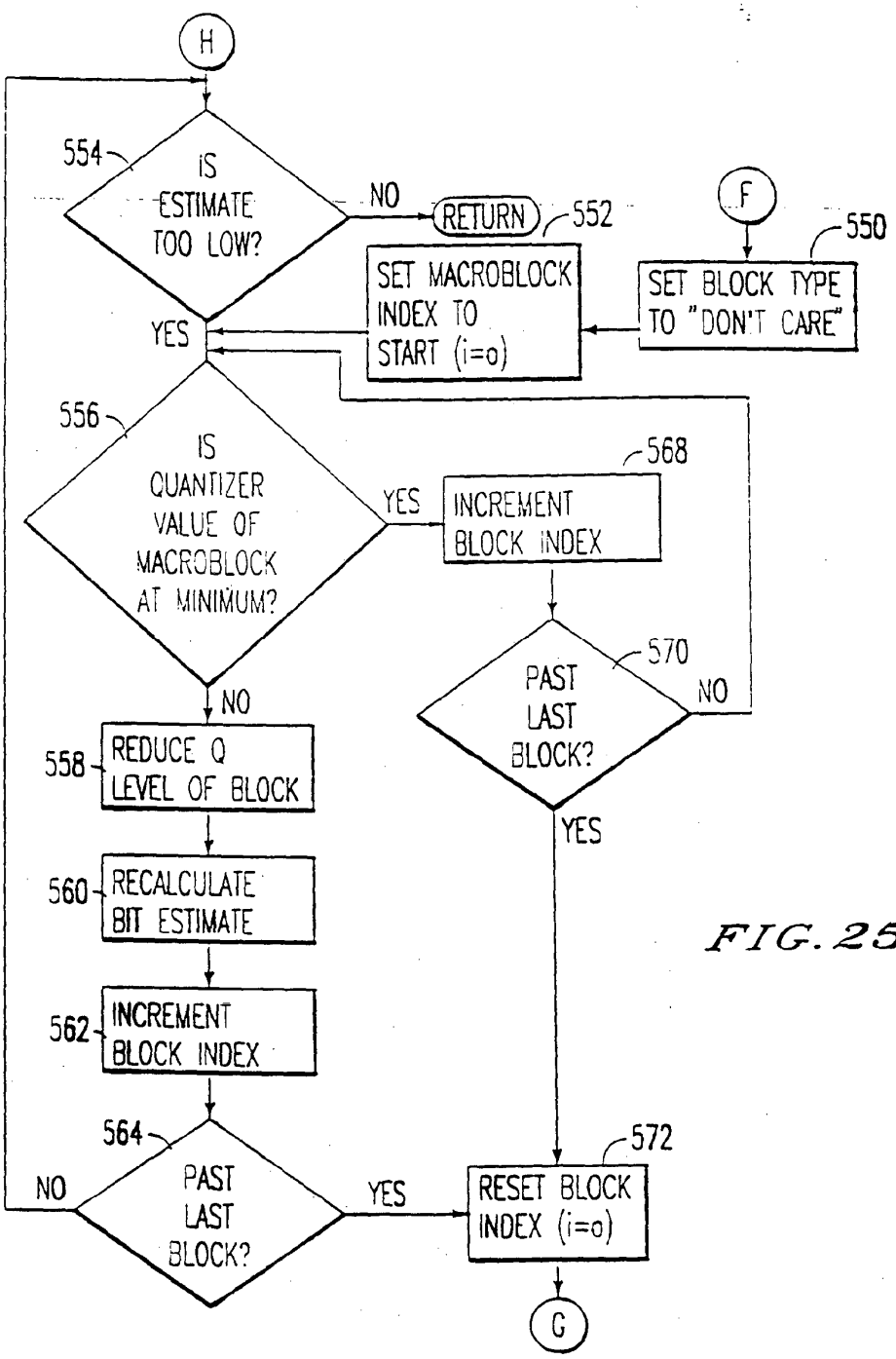

H
554
IS ESTIMATE TOO LOW?
NO
RETURN
552
F
550
SET MACROBLOCK INDEX TO START (i=o)
SET BLOCK TYPE TO "DON'T CARE"
YES
556
IS QUANTIZER VALUE OF MACROBLOCK AT MINIMUM?
YES
568
INCREMENT BLOCK INDEX
570
PAST LAST BLOCK?
NO
YES
NO
558
REDUCE Q LEVEL OF BLOCK
560
RECALCULATE BIT ESTIMATE
562
INCREMENT BLOCK INDEX
564
PAST LAST BLOCK?
NO
YES
572
RESET BLOCK INDEX (i=o)
G

FIG. 25A

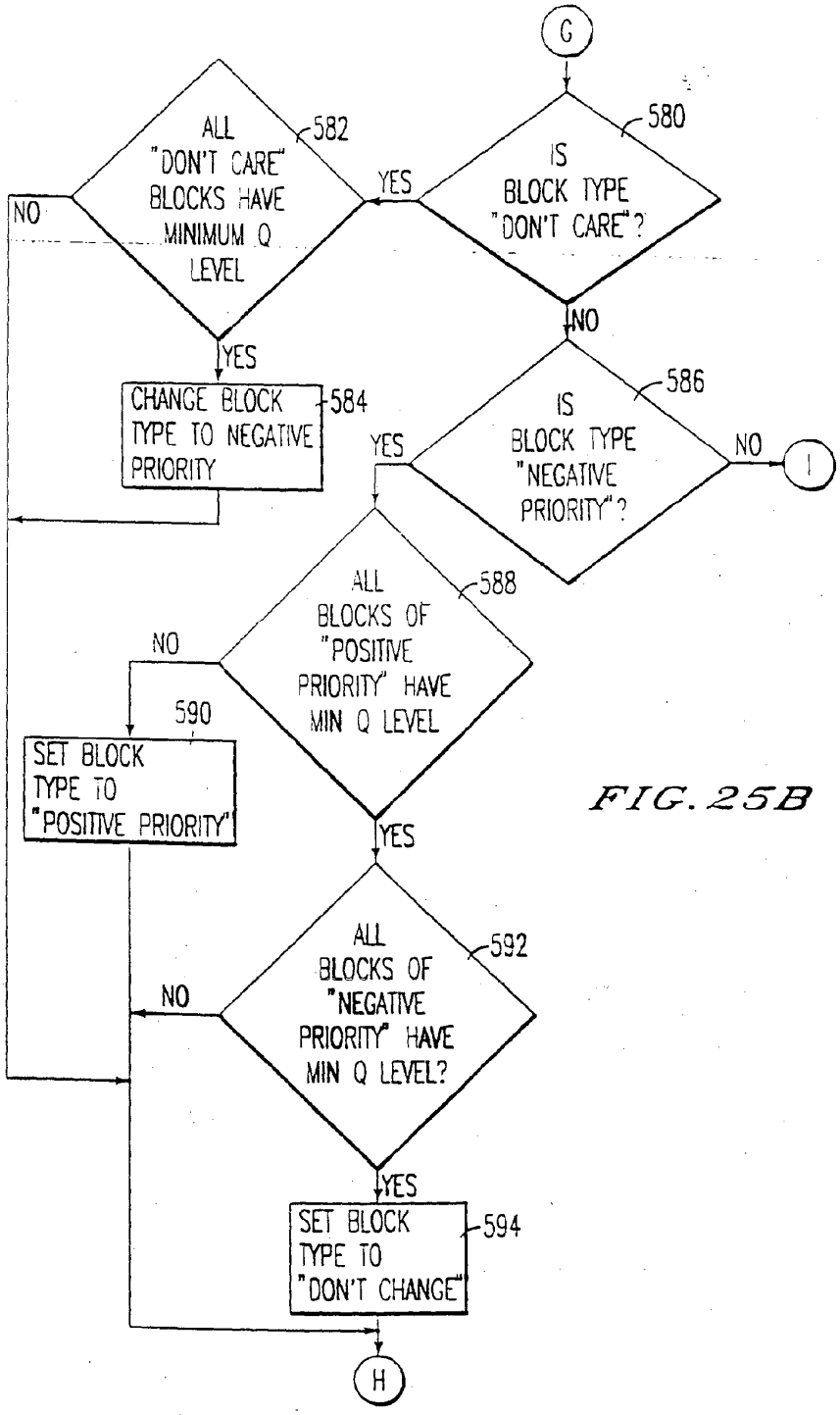
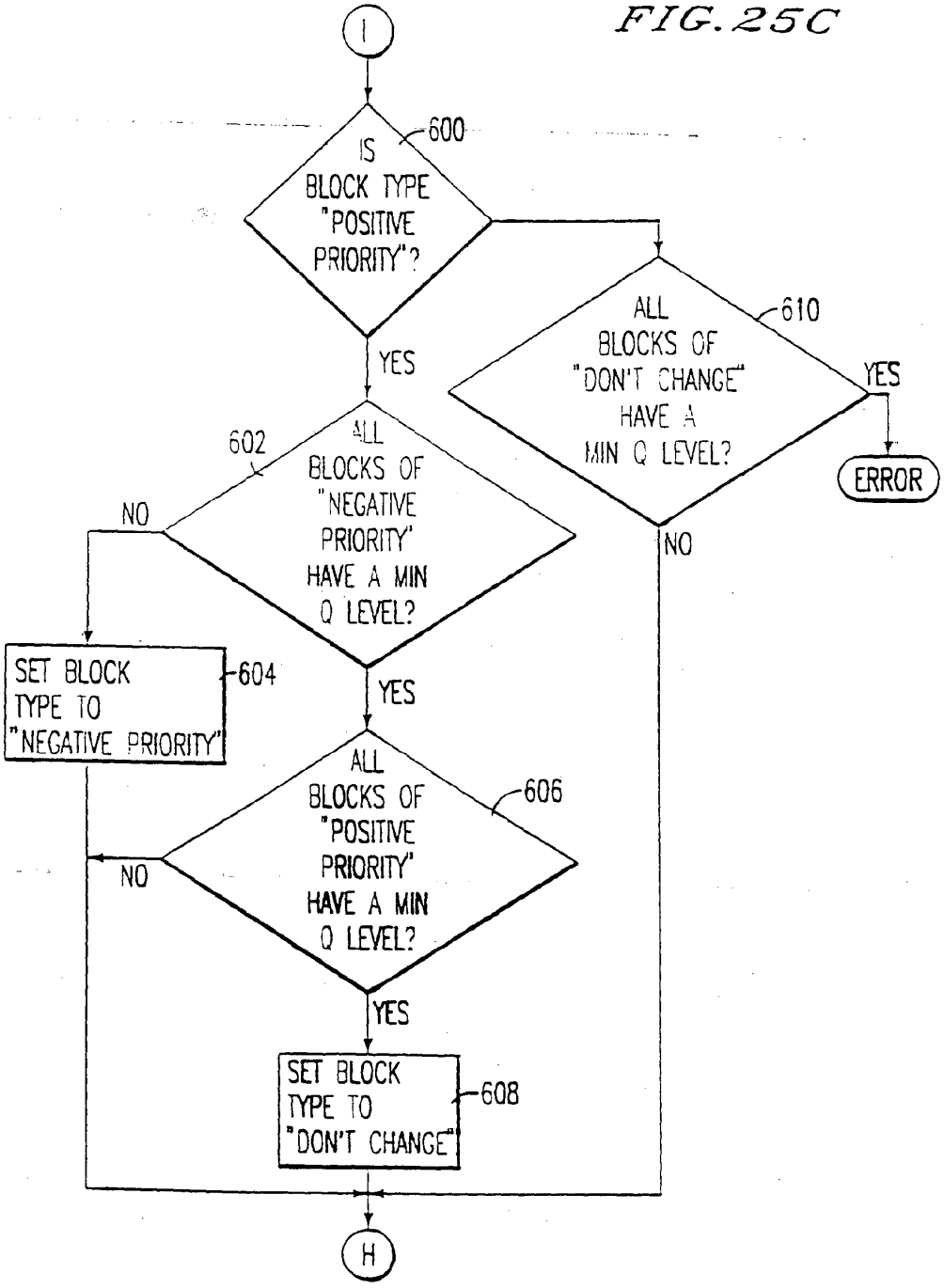

G
IS BLOCK TYPE "DON'T CARE"?
580
YES
ALL "DON'T CARE" BLOCKS HAVE MINIMUM Q LEVEL
582
NO
YES
CHANGE BLOCK TYPE TO NEGATIVE PRIORITY
584
NO
IS BLOCK TYPE "NEGATIVE PRIORITY"?
586
YES
NO
I
ALL BLOCKS OF "POSITIVE PRIORITY" HAVE MIN Q LEVEL
588
NO
590
SET BLOCK TYPE TO "POSITIVE PRIORITY"
YES
ALL BLOCKS OF "NEGATIVE PRIORITY" HAVE MIN Q LEVEL?
592
NO
YES
SET BLOCK TYPE TO "DON'T CHANGE"
594
H

FIG. 25B

FIG. 25C


I
IS BLOCK TYPE "POSITIVE PRIORITY"?
600
YES
ALL BLOCKS OF "NEGATIVE PRIORITY" HAVE A MIN Q LEVEL?
602
NO
SET BLOCK TYPE TO "NEGATIVE PRIORITY"
604
YES
ALL BLOCKS OF "POSITIVE PRIORITY" HAVE A MIN Q LEVEL?
606
NO
YES
SET BLOCK TYPE TO "DON'T CHANGE"
608
ALL BLOCKS OF "DON'T CHANGE" HAVE A MIN Q LEVEL?
610
YES
ERROR
NO
H

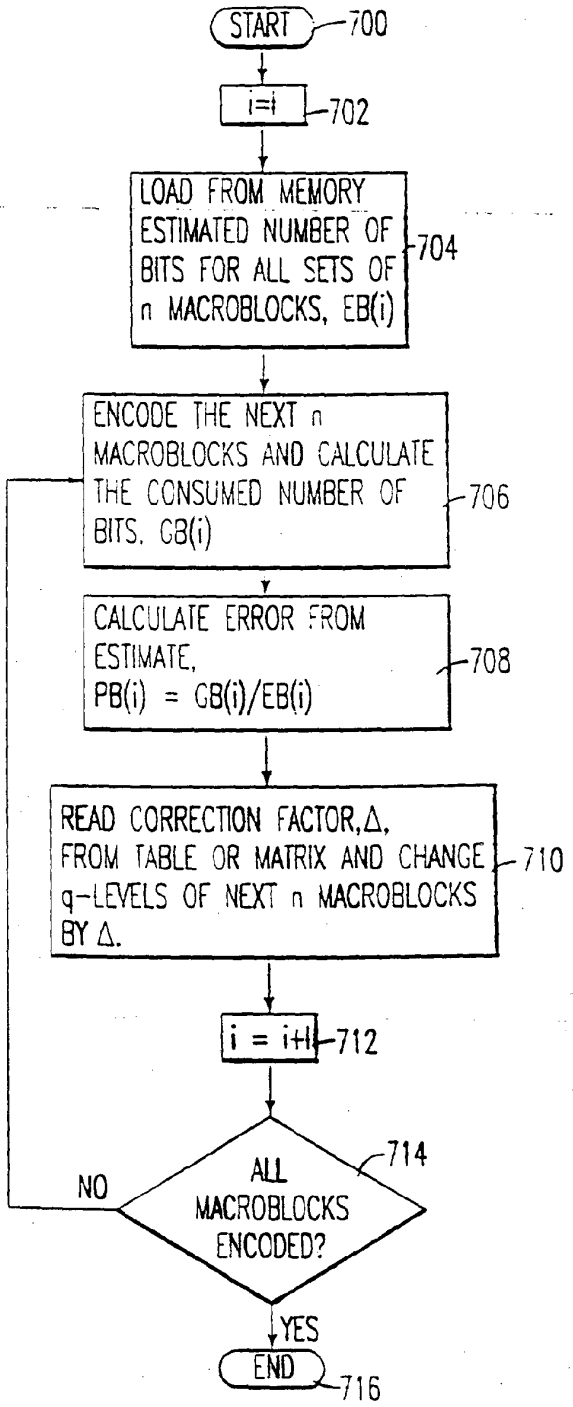


FIG. 26

LINKPLAY EXHIBIT 1004
Page 1701 of 4822

$q_{i,j}$	$q_{i+1,j}$		$q_{i+n-2,j}$	$q_{i+n-1,j}$	$q_{i+n,j}$
$q_{i,j+1}$	$q_{i+1,j+1}$				$q_{i+n,j+1}$

FIG. 27A

$q_{i,j}$	$q_{i+1,j}$			$q_{i+n,j}$
$q_{i,j+1+\Delta 1}$	$q_{i+1,j+1+\Delta 1}$			$q_{i+n,j+1+\Delta 1}$

FIG. 27B

$q_{i,j}$	$q_{i+1,j}$			
$q_{i,j+1+\Delta 1}$	$q_{i+1,j+1+\Delta 1}$			$q_{i+n,j+1+\Delta 1}$
$q_{i,j+2+\Delta 2}$	$q_{i+1,j+1+\Delta 2}$			$q_{i+n,j+2+\Delta 2}$

FIG. 27C

$q_{i,j}$	$q_{i+1,j}$			$q_{i+n,j}$
$q_{i,j+1}$	$q_{i+1,j+1}$			$q_{i+n,j+1}$
$q_{i,j+2+\Delta 1}$	$q_{i+1,j+2+\Delta 1}$			$q_{i+n,j+2+\Delta 1}$
$q_{i,j+3+\Delta 1}$	$q_{i+1,j+3+\Delta 1}$			$q_{i+n,j+3+\Delta 1}$

FIG. 27D

LINKPLAY EXHIBIT 1004
Page 1702 of 4822

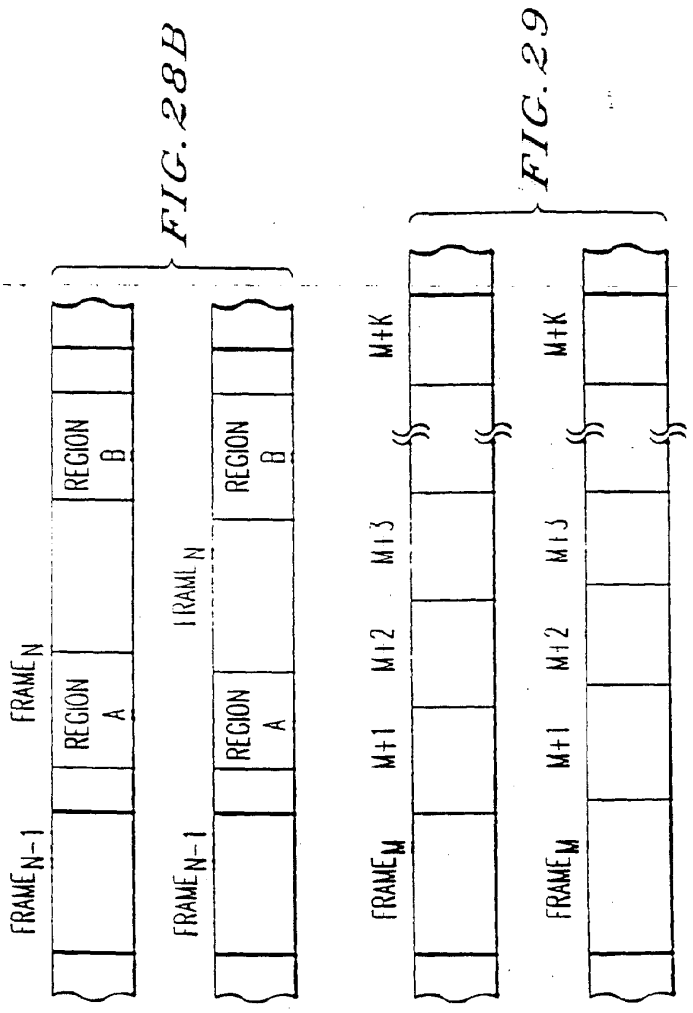


FIG. 28B

FIG. 29

FIG. 28A

LINKPLAY EXHIBIT 1004
Page 1703 of 4822

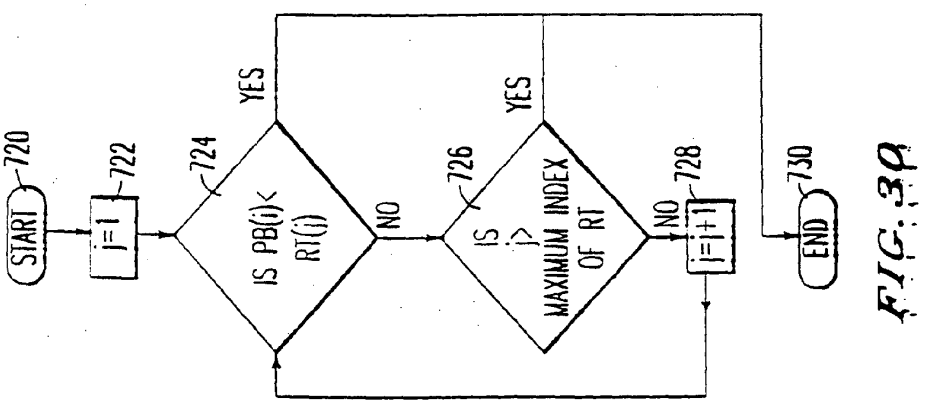


FIG. 30

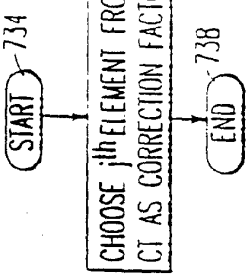


FIG. 31

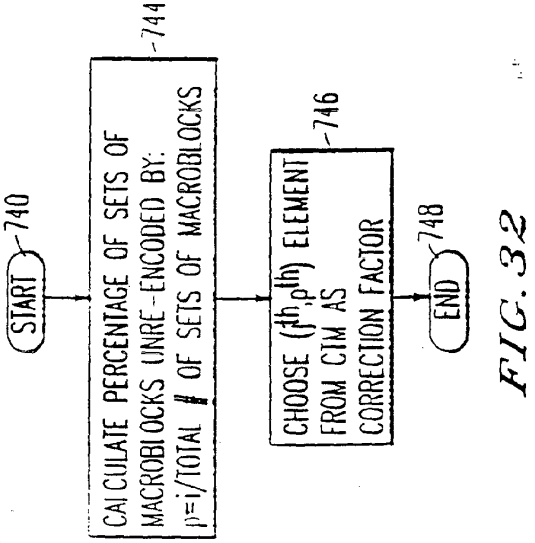


FIG. 32

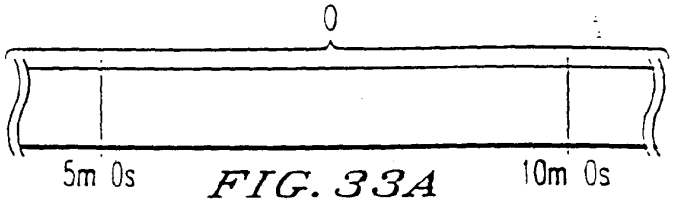


FIG. 33A

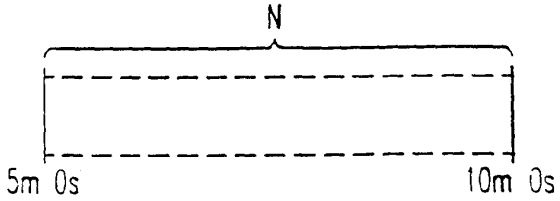


FIG. 33B

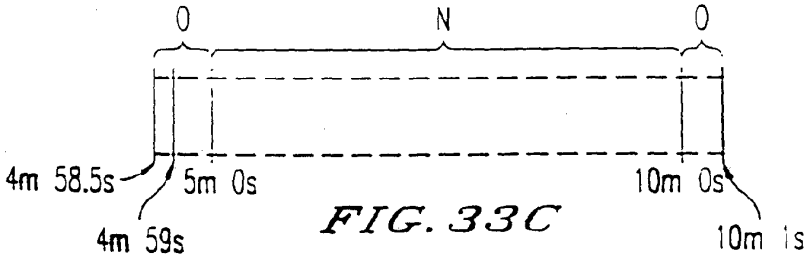


FIG. 33C

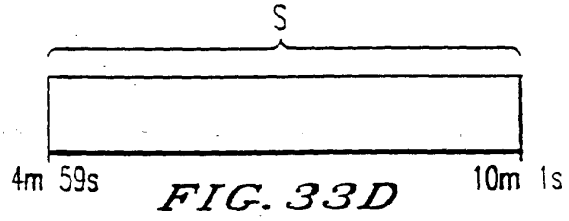


FIG. 33D

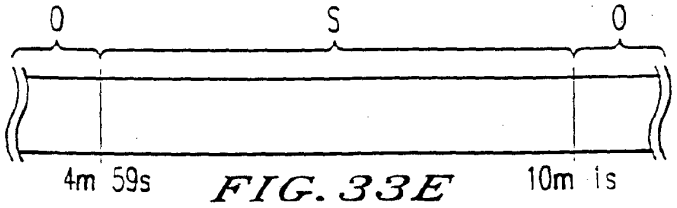


FIG. 33E

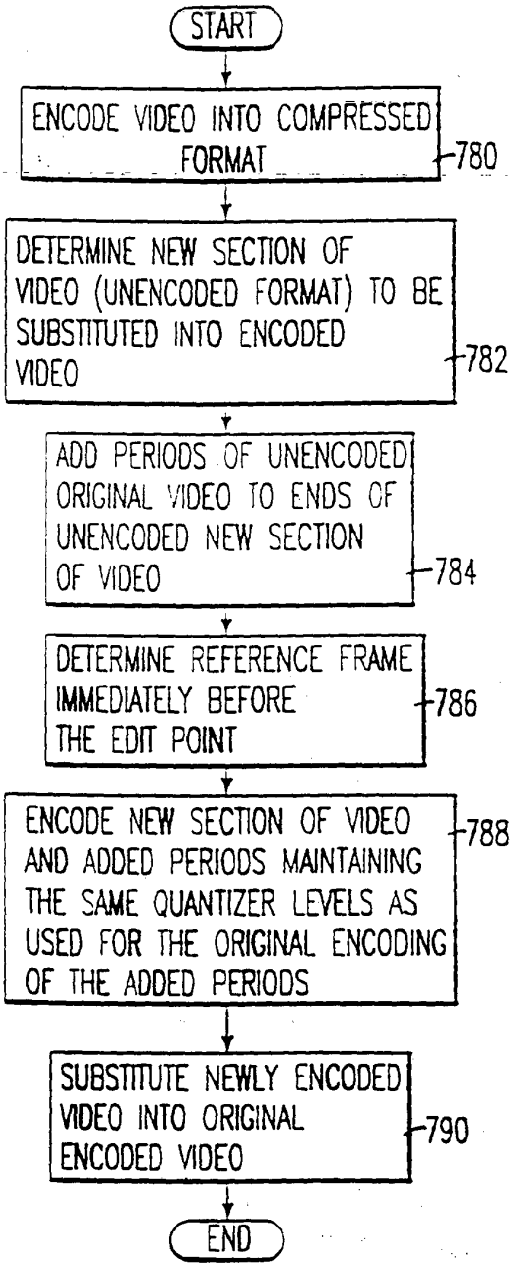


FIG. 34

LINKPLAY EXHIBIT 1004
Page 1706 of 4822

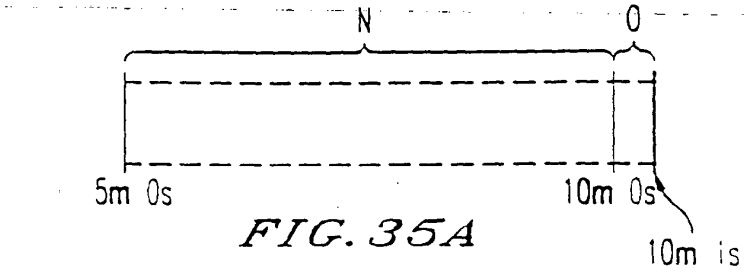


FIG. 35A

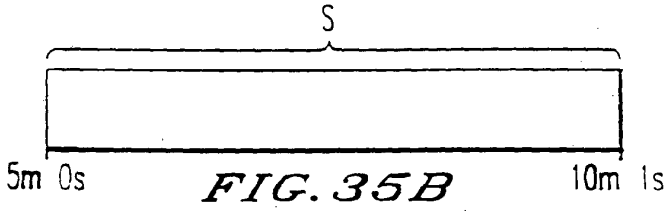


FIG. 35B

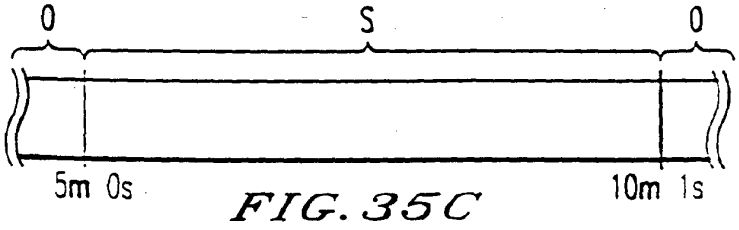


FIG. 35C

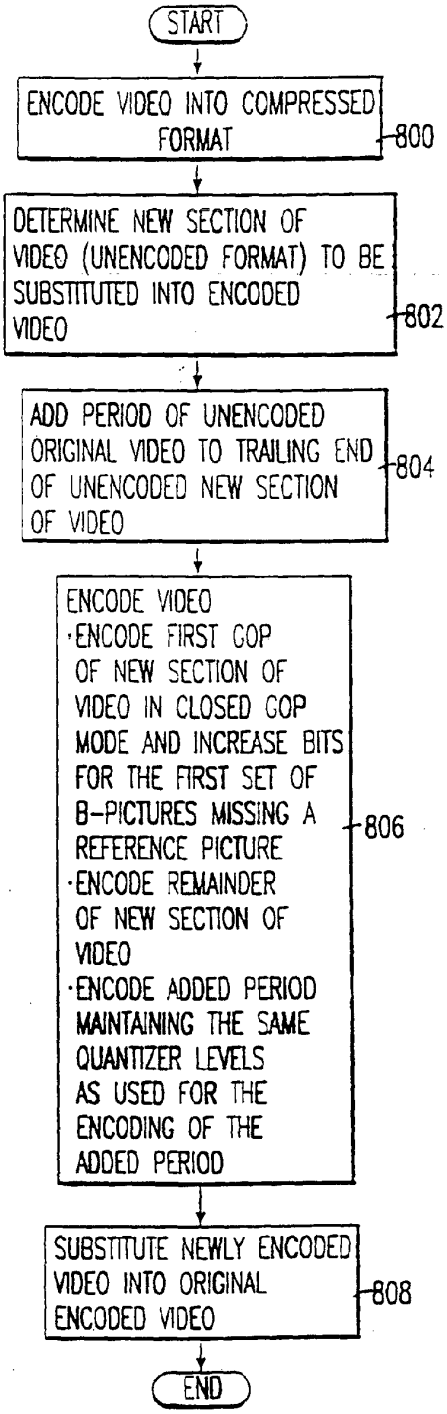


FIG. 36

LINKPLAY EXHIBIT 1004
Page 1708 of 4822

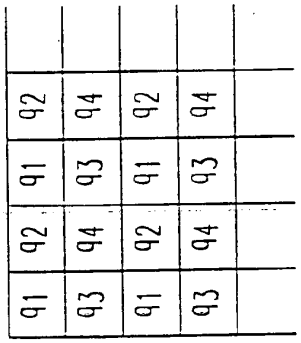


FIG. 37C

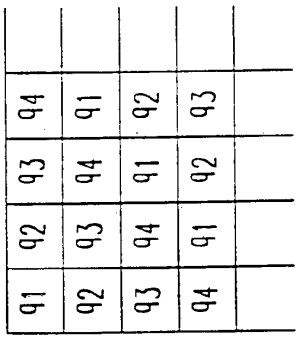


FIG. 37B

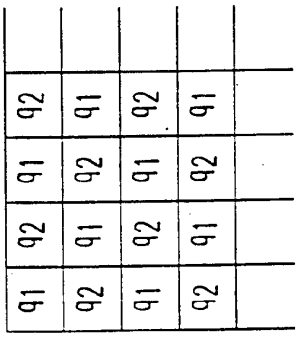


FIG. 37A

LINKPLAY EXHIBIT 1004
Page 1709 of 4822

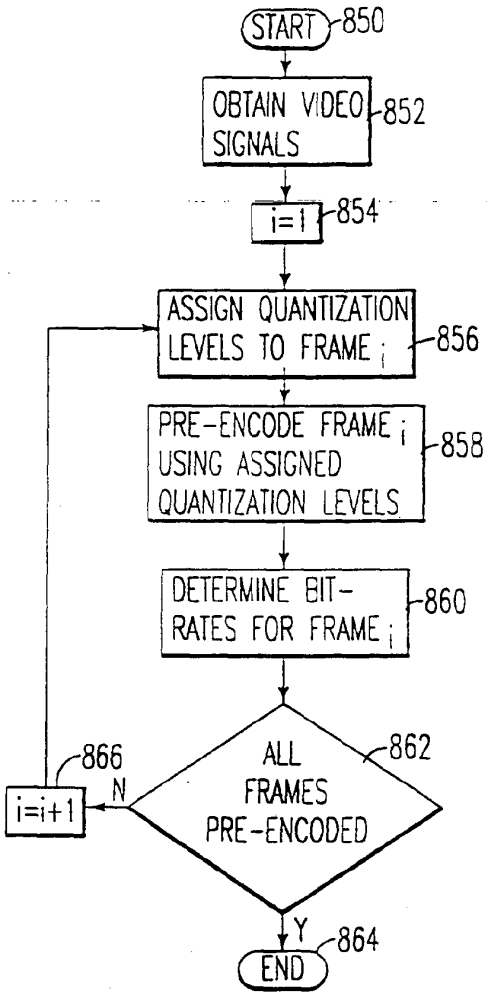


FIG. 38

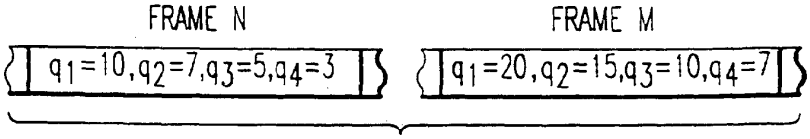


FIG. 39

LINKPLAY EXHIBIT 1004
Page 1710 of 4822

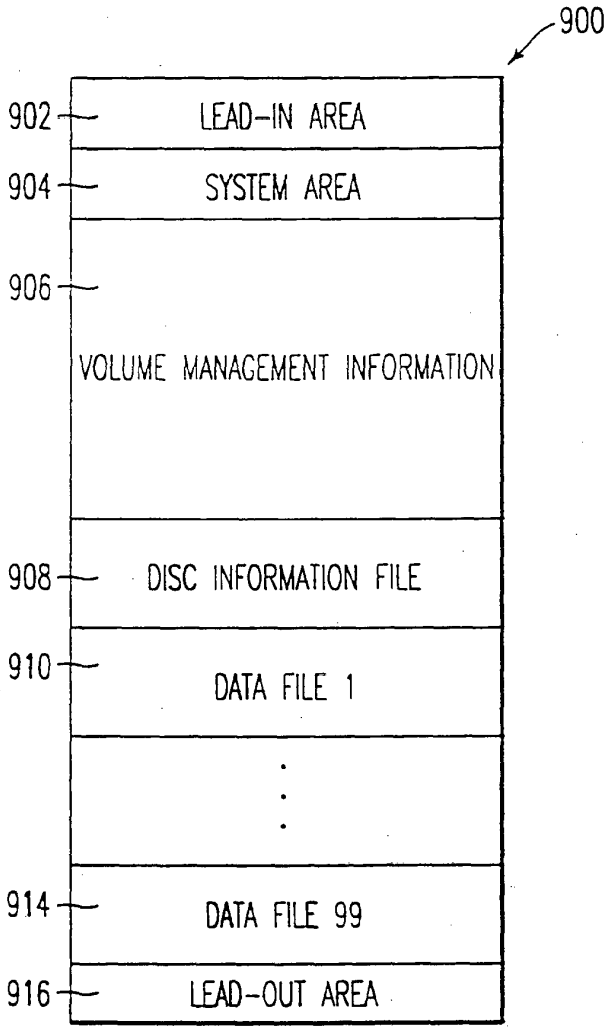


FIG. 40

LINKPLAY EXHIBIT 1004
Page 1711 of 4822

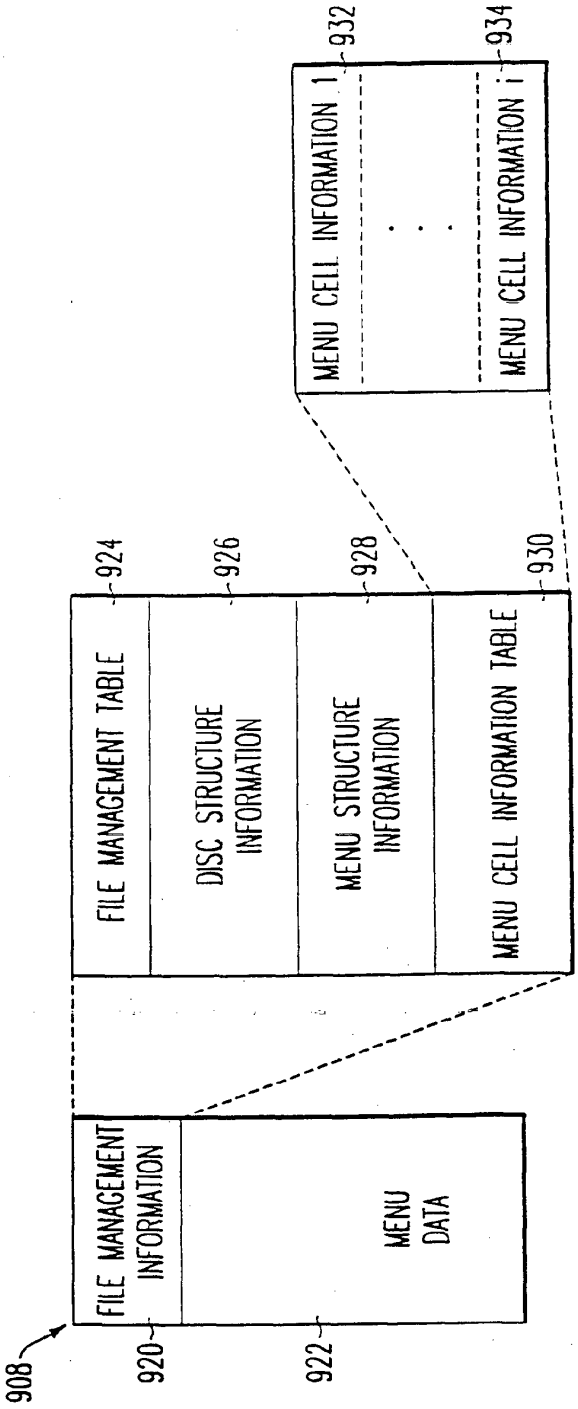


FIG. 41

LINKPLAY EXHIBIT 1004
Page 1712 of 4822

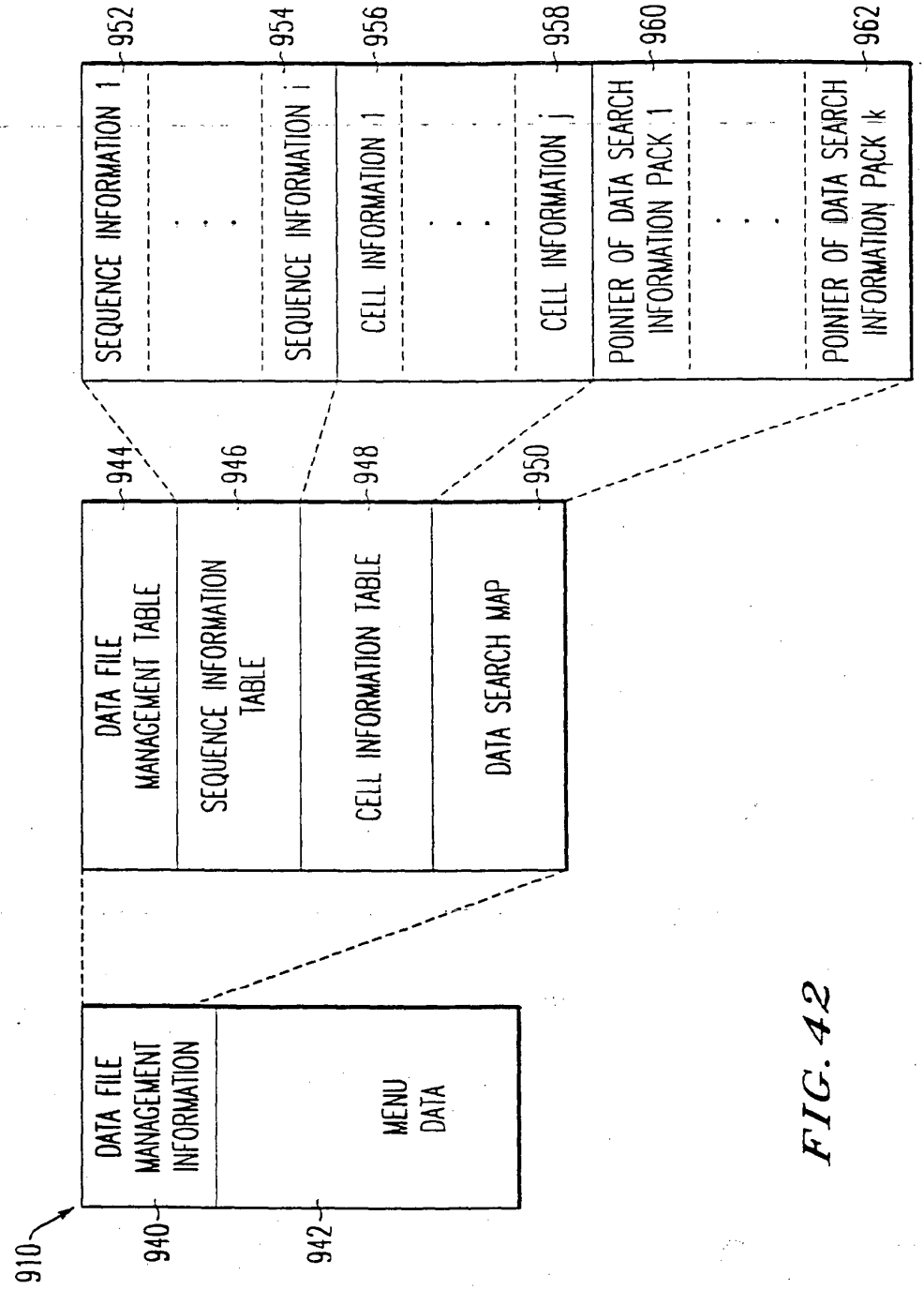


FIG. 42

LINKPLAY EXHIBIT 1004
Page 1713 of 4822

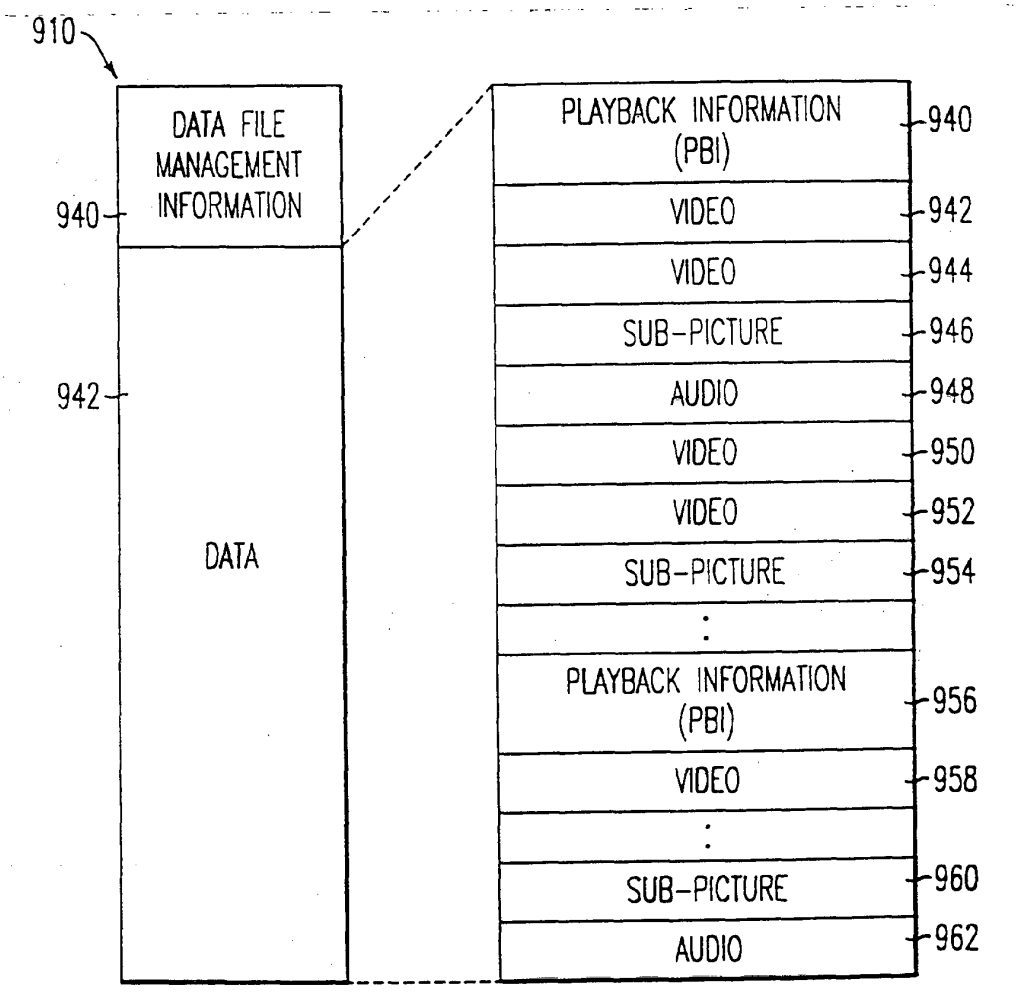


FIG. 43

LINKPLAY EXHIBIT 1004
Page 1714 of 4822

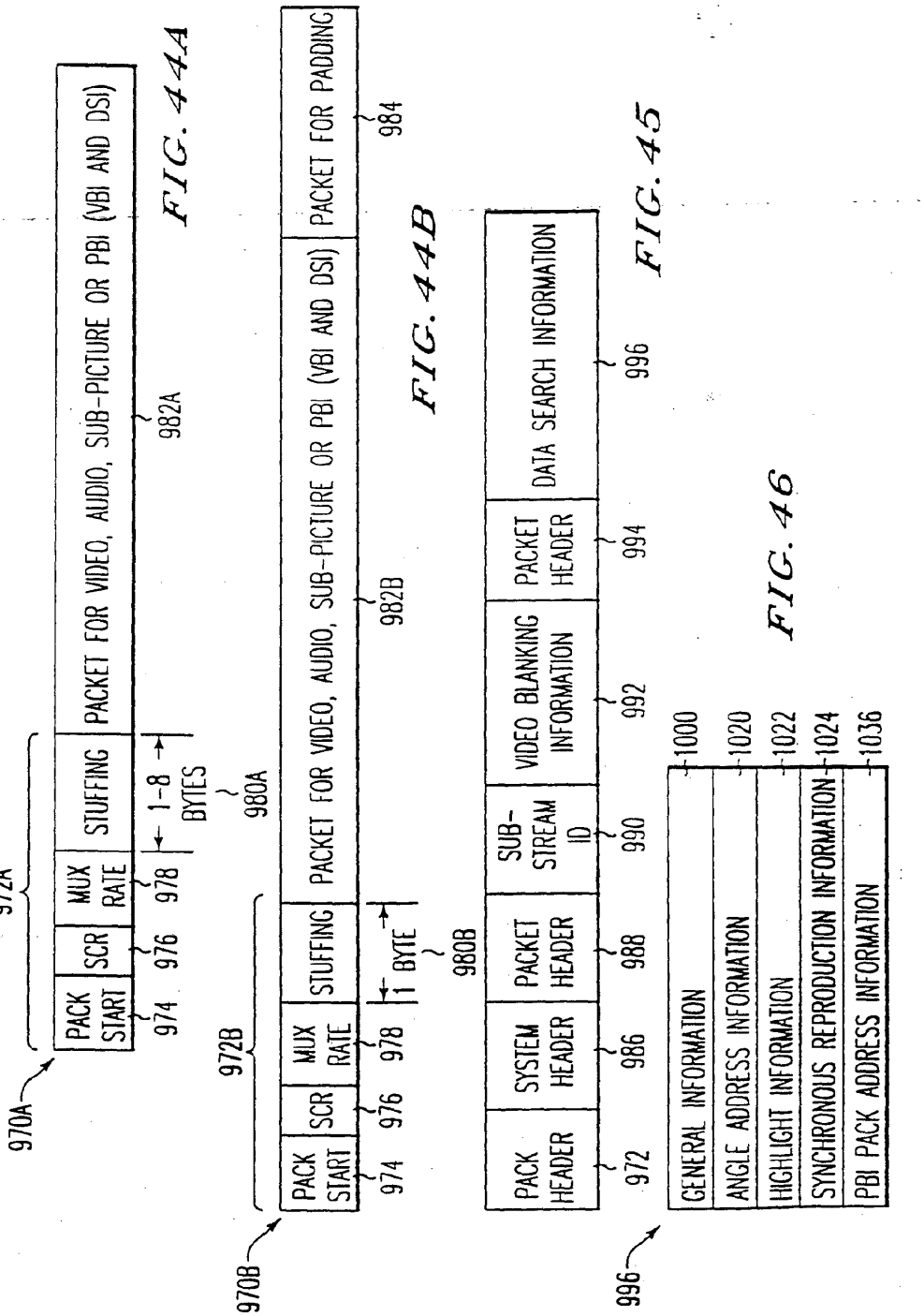


FIG. 44A

FIG. 44B

FIG. 45

FIG. 46

LINKPLAY EXHIBIT 1004
Page 1715 of 4822

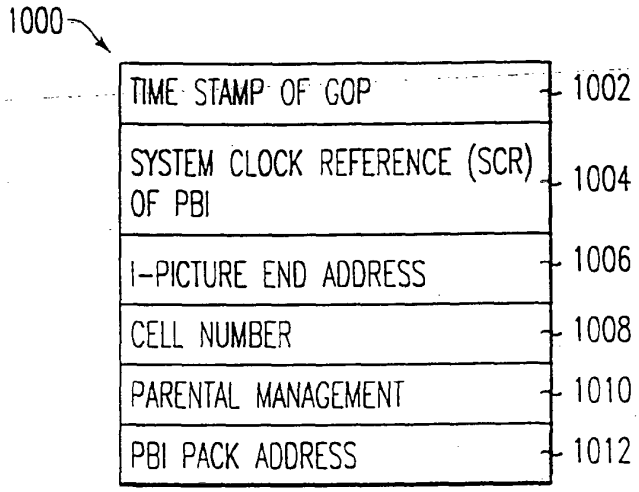


FIG. 47

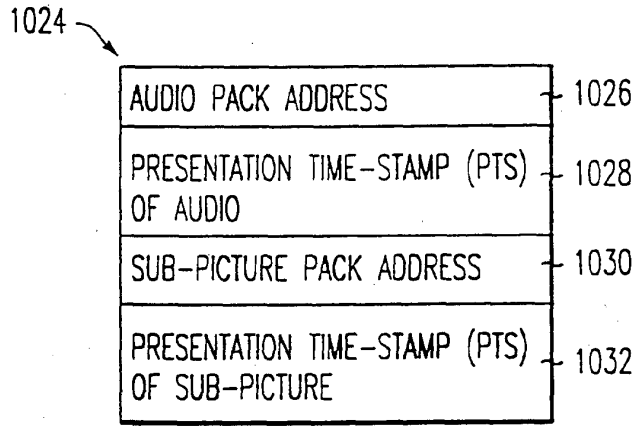


FIG. 48

LINKPLAY EXHIBIT 1004
Page 1716 of 4822

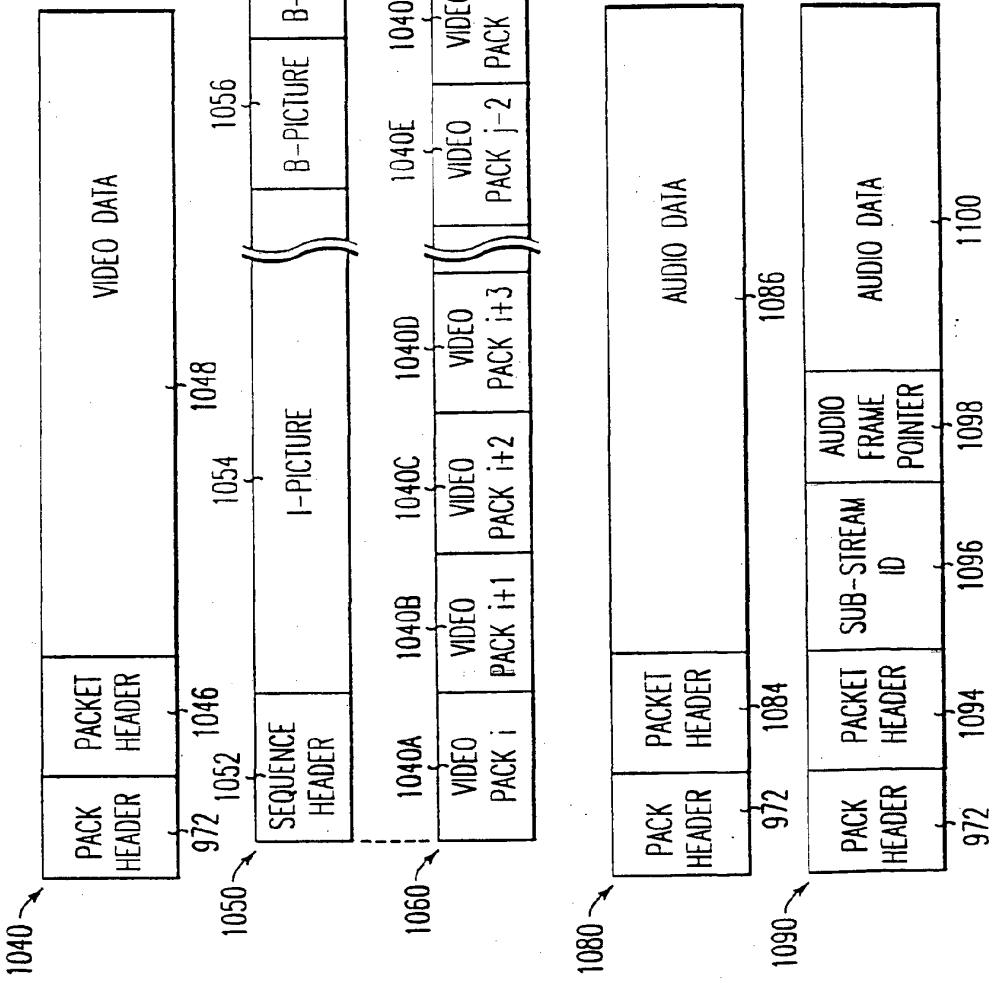


FIG. 49

FIG. 50

FIG. 51A

FIG. 51B

LINKPLAY EXHIBIT 1004
Page 1717 of 4822

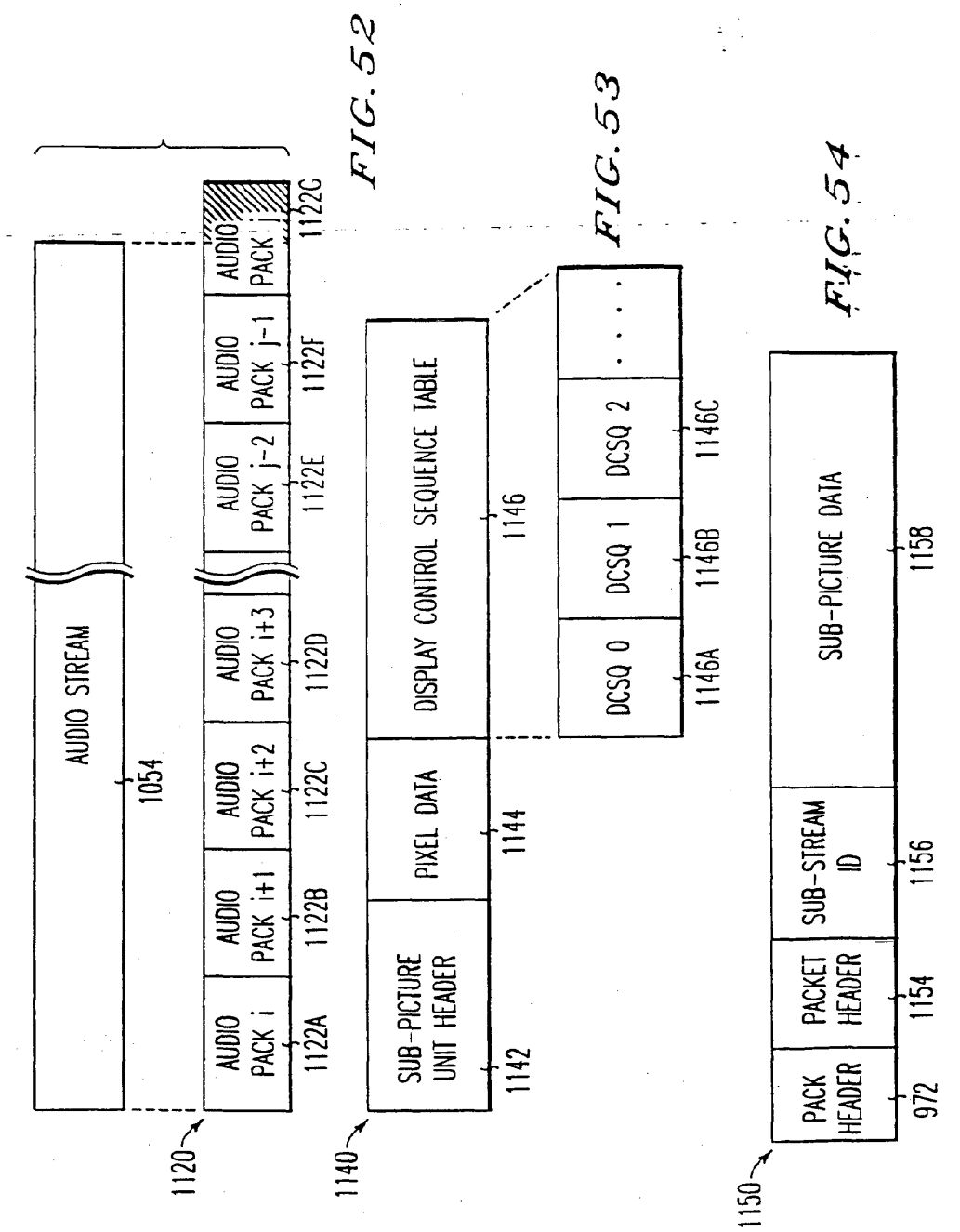


FIG. 52

FIG. 53

FIG. 54

LINKPLAY EXHIBIT 1004
Page 1718 of 4822

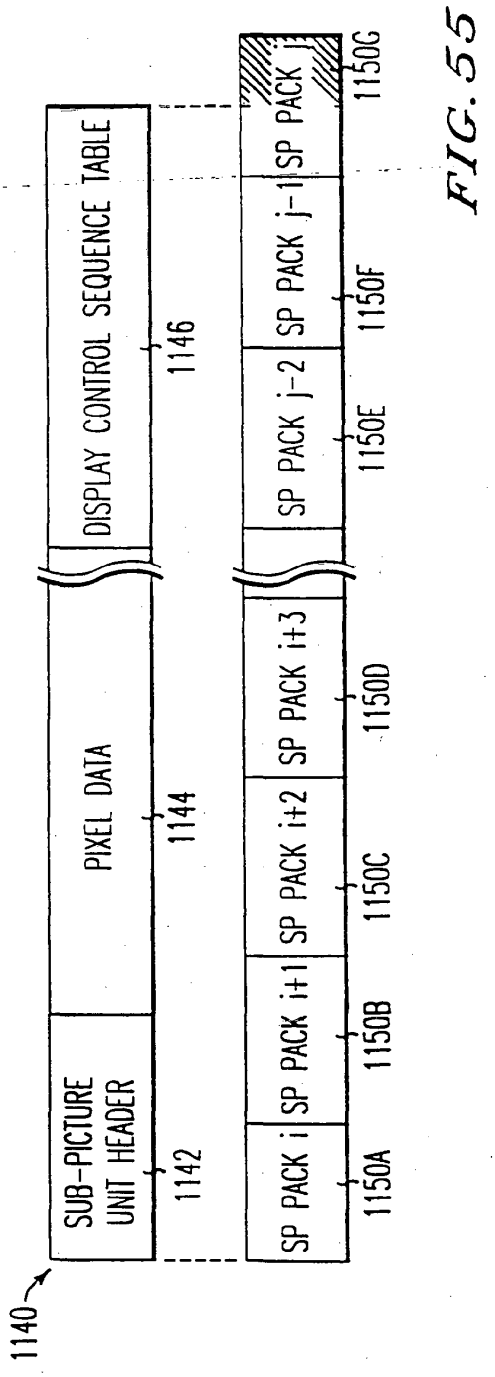


FIG. 55

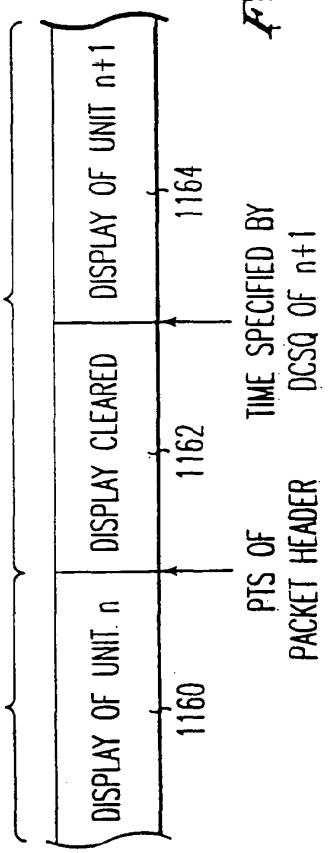


FIG. 56

LINKPLAY EXHIBIT 1004
Page 1719 of 4822

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 43801495 A **[0001]**
- WO 9703443 A **[0001]**
- WO 9636173 A **[0001]**
- WO 9635999 A **[0001]**
- WO 9636170 A **[0001]**
- WO 9635998 A **[0001]**
- EP 0597647 A **[0009]**
- JP 6326435 A **[0030]**
- JP 8186821 A **[0030]**
- US 30484994 A **[0240]**
- US 5459712 A **[0240]**
- US 5400077 A, Cookson **[0241]**
- JP 7081298 A **[0302]**
- EP 737009 A **[0302]**
- WO 785693 A **[0302]**
- EP 737975 A **[0302]**

Non-patent literature cited in the description

- **REININGER et al.** Joint Multi-Channel Bit-Rate Control for Multiplexed MPEG Video Using Rate-Distortion Curves. *Princeton Conference,* 1994 **[0006]**
- **RAMCHANDRAN et al.** Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Disclosures. *IEEE Transactions on Image Processing,* September 1994, vol. 3 (5), 533-545 **[0102]**
- **RAMCHANDRAN et al.** Bit Allocation for Dependent Quantization with Applications to MPEG Video Coders. *IEEE ICASSP,* 1993, V-381-V-384 **[0102]**
- **REININGER.** Joint Multi-channel Bit-Rate Control for Multiplexed MPEG Video using Rate-Distortion Curves. *Princeton video conference,* 1994 **[0102]**

LINKPLAY EXHIBIT 1004
Page 1720 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 772 374 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
07.05.1997 Bulletin 1997/19

(51) Int. Cl.[6]: **H04R 3/04**, H04R 29/00

(21) Application number: **96203005.2**

(22) Date of filing: **29.10.1996**

(84) Designated Contracting States:
DE DK FR GB

(30) Priority: **02.11.1995 DK 1124/95**

(71) Applicant: **BANG & OLUFSEN A/S**
DK-7600 Struer (DK)

(72) Inventor: **Pedersen, Jan Abildgaard**
Prof. Holstlaan 6, 5656 AA Eindhoven (NL)

(74) Representative: **Schoonheijm, Harry Barend**
INTERNATIONAAL OCTROOIBUREAU B.V.,
Prof.Holstlaan 6
5656 AA Eindhoven (NL)

(54) **Method and apparatus for controlling the performance of a loudspeaker in a room**

(57) A method for controlling the performance of a loudspeaker in a room, by determining in a first acoustic environment suitable parameters defining a first complex transfer function based on a measurement of the resultant movement of a loudspeaker driver diaphragm and the associated force, arising from the sound field in the room, acting on it, and in a second acoustic environment the same or different parameters of a second complex transfer function by measuring the loudspeaker driver relating to the room. The ratio between the real parts of the first and second transfer functions is used to define the performance of a correcting filter, included in the signal chain to the loudspeaker driver.

EP 0 772 374 A2

Printed by Rank Xerox (UK) Business Services
2.14.3/3.4

Description

The invention relates to a method and apparatus for controlling the performance of a loudspeaker in a room.

The actual performance of a loudspeaker is known to be highly dependent on the acoustics of the actual listening room and the actual loudspeaker position within this room. In particular the performance of a loudspeaker will change very noticeably when it is in proximity to the boundaries of the room. This is caused by the loading of the room on the loudspeaker as a radiator, or in other words due to the changing radiation resistance. A change of listener position changes the perceived performance of the loudspeaker, in particular due to early reflections and standing waves. However some boundary effects are universal in the room, in particular in the bass frequency range, and hence the perception of this range is less influenced by the listener position.

Loudspeaker designers experience this fact by having to make a compromise when optimizing the timbre of the loudspeaker so that the perceived sound will be acceptable under a number of different conditions, i.e. different room acoustics, loudspeaker positions, and listening positions. Even though making this compromise, the designer cannot ensure that the customer will always experience the intended quality. Thus, the listener will experience a performance of the loudspeaker that depends on the acoustic properties of the actual listening room and the position chosen for both loudspeaker and listener. There is a risk that an expensive loudspeaker which performs very well in the shop, will turn out performing badly or at least disappointingly when placed in a different environment and/or in a different position.

In order to compensate for this problem it is known to fit a switch in the cross-over filter unit in the loudspeaker in order that the bass response may be modified to suit a particular placement of the loudspeaker. At best, this must be considered a poor compromise, and if at all possible, the precise adjustment will be dependent on a measurement of the room characteristics. Some automatic systems are based on measuring the transfer function from the input of the loudspeaker to an omnidirectional microphone, placed at the preferred listening position or a number of representative positions. An equalizing filter is then inserted so that the resulting transfer function approximates a target function, which e.g. can be flat in the frequency range of interest. A major problem of such systems is the sensitivity to changes in the position of the sound source as well as the receiver. If the position of the loudspeaker or the listener is changed after calculating the equalizing filter, the effects can be severe colouration, pre-echoes, etc. Another problem of such systems is the choice of a suitable target function, where a flat function may not be found to be optimal.

It has in the present invention been realized that since all the involved acoustic phenomena's are considered to be linear, what is actually compensated through the apparently sensible procedures discussed above is the superposition of several phenomena, such as standing waves/natural frequencies of the room, early reflections, reverberation and the reduction of angular space angle due to the boundary effect, and it is considered that this is the reason why the known procedures will only function for one listening position.

It is the purpose of the invention to provide a method and apparatus for controlling the performance of a loudspeaker in a room in order that it becomes independent of the placement of the loudspeaker. This is obtained in a method according to the invention which is particular in that in a first acoustic environment the movement, e.g. velocity, of the diaphragm of the loudspeaker driver and the force, arising from the sound field, acting on it are determined by measuring suitable parameters, defining thereby a first complex transfer function, that in a second acoustic environment a second complex transfer function is determined by measuring the same or different parameters of the loudspeaker driver, relating to the room, that the ratio between the real parts of the first and second transfer functions is used to define the performance of a correcting filter, that the filter is applied in the signal chain to the loudspeaker driver.

The invention is based on the realization that there is a strong link between the way the loudspeaker sounds, in particular in the bass range, and its radiation resistance as a function of frequency, being the real part of the radiation impedance. Implementing the invention for a loudspeaker has proved to significantly increase the certainty that the customer will always experience the quality intended by the loudspeaker designer. This is achieved by measuring the radiated power output, radiation resistance or any similar physical parameter, e.g. real part of the acoustic wave impedance near the diaphragm, when the loudspeaker is placed in the actual position and comparing this to a reference measurement. More precisely this is obtained in that the loudspeaker in a first step is put in a reference room environment where it performs to a standard to be determined, and during which a reference radiated power output (real, i.e. active) or reference radiation resistance of a driver as a function of frequency is measured, and in that in a second step the loudspeaker is put in its room of usage where its attendant radiated power output or radiation resistance is measured, the ratio between the said real (active) power outputs or radiation resistances respectively being used to define the transfer function of a correcting filter in order to obtain said standard of performance determined in said reference room environment, and that in a third step said correcting filter is introduced in the electrical signal path to the driver. In principle a multi-driver loudspeaker should have each driver subjected to such a measurement, however one or several may be selected as representative. At the time of measurement of one particular driver or a group of drivers, the other drivers may either be short-circuited, disconnected or connected to the

LINKPLAY EXHIBIT 1004
Page 1722 of 4822

signal.

When the loudspeaker is placed in a position which is not identical to the reference position/room, the bass performance changes. However, the method according to the invention is able to detect a major part of this change in the acoustic environment of the loudspeaker and to correct accordingly. Switching on and off an apparatus working according to the principles of the invention can lead to dramatic changes of the bass performance of the loudspeaker depending on how different the actual position and room are from the reference conditions. If a loudspeaker is designed to operate away from the walls of a room, then when placing such a loudspeaker close to a corner of the listening room, the bass performance becomes boomy, coloured, and the sound pressure level increases. In such a situation the apparatus according to the principles of the invention corrects the timbre in such a way that the perceived timbre is almost the same as in the reference position. The effect of the apparatus in this situation has been described by listeners as quite startling. The bass performance then was not plagued by the rumble which is traditionally a characteristic of a corner position, and the bass performance becomes more even and neutral without becoming "thin". In a corner position this is perceived as a dramatic improvement of the bass performance.

An advantageous embodiment is particular in that the loudspeaker is permanently fitted with measurement means, the ratio between reference and use measurements being used to define the parameters of the correcting filter. This enables a measurement to be initiated by a user or in the event that some predefined conditions are met, e.g. power up of the apparatus. This measurement cycle could be performed using a dedicated measuring signal, e.g. obtained from a particular Compact Disc.

A further advantageous embodiment of the invention is particular in that the loudspeaker is permanently fitted with measurement means, and the complex transfer function, which corresponds to the situation during usage, is continuously measured during operation of the apparatus. The ratio between reference and usage measurements being used to define the parameters of the correcting filter. This means that the loudspeaker will be automatically and continuously adaptable to any new listening room environment, e.g. using the played music as the stimuli when measuring the complex transfer functions. In this case the transfer function in the usage situation is continuously measured, and e.g. a digital signal processor in the signal chain calculates and performs the filtering which provides a sound from the loudspeaker which is very similar to the sound in the reference position/room and which presumably was judged positively during the design of the loudspeaker.

A further advantageous embodiment is particular in that the listening room is divided into zones of e.g. 30 cm by 30 cm, each having a correction filter transfer function assigned to it, and that information on the particular zone is fed to the correcting filter in the electrical signal path to the loudspeaker. By this means it is possible to accomodate a number of typical placements of a loudspeaker and to obtain a large degree of the improvement according to the invention, without having to perform a measurement.

A simpler arrangement is obtained by instructing the user to activate switches according to a schematic showing various typical placements of a loudspeaker in a room. This functions in practice, provided the loudspeaker is of the same type as the loudspeaker used in the reference environent.

An apparatus according to the invention is particular in that it comprises a filter, the transfer function of which is controllable by electronic/numerical signals, said signals being obtained from a unit which determines the ratio between a stored reference radiation resistance or active power output (real) as a function of frequency and a measured radiation resistance or active power output (real) in the usage situation. This ratio basically defines the amplitude response of the correction filter, and various filter implementations, e.g. minimum phase can be obtained from this. However various operation might be performed to modify the ratio before implementation, e.g. smoothing, convolution, frequency limiting, correction limiting, logarithm, exponential, multiplication, addition etc. and combinations of these. For instance, defining the amplitude response of the correction filter as the square root of the ratio seems to be a reasonable choice.

The invention will be further described in the following with reference to the drawing, in which

Fig. 1 shows the electrical, mechanical and acoustical signal paths associated with a loudspeaker placed in a room,
Fig. 2 shows a loudspeaker with a driver and measuring transducers, and
Fig. 3 shows a schematic of how the correction filter can be inserted in the signal chain according to one embodiment of the invention.

By way of example Fig. 1 shows the signal path and transfer functions relating to a loudspeaker in a room. The electrical signal from the source is fed to a power amplifier A which drives the loudspeaker which is designated B and comprises the electrical and mechanical parts of the loudspeaker driver unit and the acoustic influence of the cabinet enclosure. The output from the loudspeaker is transformed by the transfer function C from the acceleration of the diaphragm to the sound pressure in front of the diaphragm which may be measured by a microphone D as one example of how to obtain the force, arising from the sound field, acting on the diaphragm. An accelerometer E for example may measure the diaphragm acceleration directly. At point 1 the source signal is provided, at point 2 the electrical input signal to the loudspeaker driver is available, point 3 refers to the acceleration of the diaphragm of the loudspeaker, and at point 4 the sound pressure at some pre-

LINKPLAY EXHIBIT 1004
Page 1723 of 4822

determined and fixed point in front of the driver is available. After being converted by the microphone D an electrical signal representing the sound pressure is available at point 5, and correspondingly, an electrical signal representing the membrane acceleration is available at point 6.

Fig. 2 shows one embodiment of the invention where the loudspeaker B with one of a multitude of possible placements of a microphone D and an accelerometer E.

Fig. 3 shows how a measurement of the radiation resistance of the loudspeaker is used when calculating the filter F, which is switched into the signal path. The signal processing may occur through any means available to the skilled person, the result will be a linear predistortion of the signal to the power amplifier in order that the loudspeaker provides an excitation of the listening room so that the perceived sound is a good approximation to the quality determined during the design phase. The advantage of making the measurement continuous is that the system will automatically compensate e.g. for an influx of listeners or a changed placement of furniture or the loudspeaker placement itself, which disturbs the sound distribution in the room. Such a disturbance is now compensated so that the perceived sound is essentially unchanged.

Claims

1. A method for controlling the performance of a loudspeaker in a room, **characterized in** that in a first acoustic environment the resultant movement of the loudspeaker driver diaphragm and the associated force, arising from the sound field in the room, acting on it are determined by measuring suitable parameters defining a first complex transfer function, that in a second acoustic environment a second complex transfer function is determined by measuring the same or different parameters of the loudspeaker driver relating to the room, that the ratio between the real parts of the first and second transfer functions is used to define the performance of a correcting filter, that the filter is applied in the signal chain to the loudspeaker driver.

2. A method for controlling the performance of a loudspeaker in a room, in particular in the low frequency range, according to claim 1, **characterized in** that the loudspeaker in a first step is put in a reference room environment where it performs to a standard to be determined, and during which a reference radiated power output (real, i.e. active), reference radiation resistance (acoustic or mechanical) of a driver or any similar physical parameter, e.g. real part of the acoustic wave impedance near the diaphragm of the driver, as a function of frequency is measured, and in that in a second step the loudspeaker is put in its room of use where a usage radiated power output (real, i.e. active), usage radiation resistance of the same driver or any similar physical parameter, e.g. real part of the acoustic wave impedance near the diaphragm of the same driver, is measured, the ratio between the real part of said power outputs (active), radiation resistances or any similar physical parameters, e.g. real parts of the acoustic wave impedances near the diaphragm of the driver, respectively being used to define the transfer function of a correcting filter in order to obtain said standard of performance determined in said reference room environment, and that in a third step said correcting filter is introduced in the electrical signal path to the driver.

3. A method according to claim 1 and 2 , **characterized in** that the loudspeaker is permanently fitted with measurement means, the ratio between reference and usage measurements being used to define the parameters of the correcting filter.

4. A method according to claim 1 and 2, **characterized in** that the loudspeaker is permanently fitted with measurement means and is continously measuring the second complex transfer function, the ratio between reference and usage measurements being used to define the parameters of the correcting filter.

5. A method according to claim 3, **characterized in** that the measurement means are activated by a user or in the event that some predefined conditions are met, e.g. power up of the apparatus.

6. A method according to claim 1 and 2, **characterized in** that the listening room is divided into zones of e.g. 30 cm by 30 cm, each having a correcting filter transfer function assigned to it, and that information on the particular zone is fed to the correcting filter in the electrical signal path to the loudspeaker.

7. An apparatus for performing the method according to claim 1 and 2, **characterized in** that it comprises a filter, the transfer function of which is controllable by electronic/numerical signals, said signals being obtained from a unit which determines the ratio between a stored reference radiation resistance or active power output (real) or wave resistance near the driver as a function of frequency and a measured radiation resistance or active power output (real) or wave resistance near the driver in the usage situation. This ratio basically defines the amplitude response of the correction filter, and various filter implementations, e.g. minimum phase can be obtained from this. However various operations might be performed to modify the ratio before implementation, e.g. smoothing, convolution, frequency limiting, correction limiting, logarithm, exponential, multiplication, addition etc. and combinations of these. For instance, defining the

LINKPLAY EXHIBIT 1004
Page 1724 of 4822

amplitude response of the correction filter as the square root of the ratio seems to be a reasonable choice.

8. A method according to claim 3 or 4 or 5 or 6, **characterized in** that a multi-driver system, e.g. 2 woofers and 1 tweeter, should have each driver subjected to a measurement according to claim 1 and 2. However one or several may be selected as representative. At the time of measurement of one particular or a group of drivers, the other drivers may either be short-circuited, disconnected or connected to the signal. Each driver may have individual filters implemented or some groups may have a common filter implemented.

LINKPLAY EXHIBIT 1004
Page 1725 of 4822

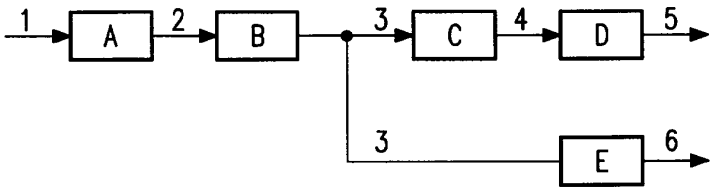


FIG. 1

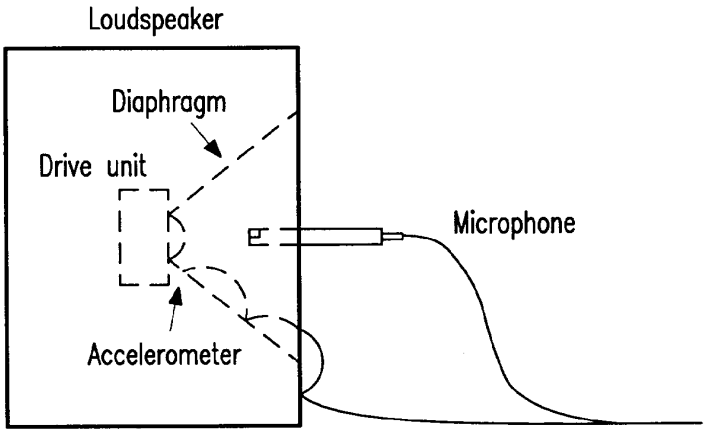


FIG. 2

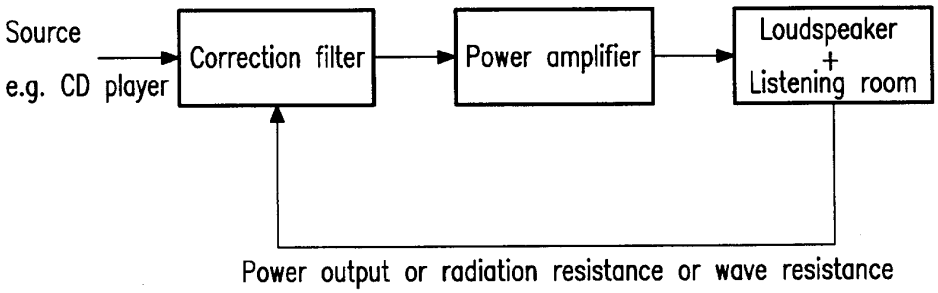


FIG. 3

LINKPLAY EXHIBIT 1004
Page 1726 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 895 427 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.02.1999 Bulletin 1999/05

(51) Int Cl.[6]: **H04N 7/52**

(21) Application number: **98305634.2**

(22) Date of filing: **15.07.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **28.07.1997 US 901090**

(71) Applicant: **Sony Electronics Inc.**
Park Ridge, New Jersey 07656-8003 (US)

(72) Inventors:
- **Tan, Yew-Koon**
  San Jose, California 95134 (US)
- **Yu, Gong-san**
  Milpitas, California 95035 (US)

(74) Representative: **Cotter, Ivan John et al**
D. YOUNG & CO.
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Audio-video synchronizing

(57) A method and apparatus for synchronizing playback of audio and video frames from a program source associates an audio presentation time stamp ("PTS") value with an output audio frame. Selected ones of audio and video data packets include respective audio and video PTS values representing desired playback times of the respective audio and data associated therewith. The selected ones of the audio data packets further include audio frame numbers representing a number of output frames of audio to be played back between the selected ones of the audio data packets. The method comprises storing the audio and video PTS values in respective audio and video PTS tables (302, 304) during an audio demultiplexing process. In addition, the audio frame numbers are stored in frame counters (309) in association with respective PTS values during the demultiplexing process. Thereafter, the process sequentially decodes the audio and video input data to produce respective frames of audio and video which are presented to the user. With the presentation of each audio and video frame, the respective audio and video frame counters (309) are selectively decremented. Upon detecting one of the audio frame counters having a zero value, the audio PTS value for that zero value audio frame counter is retrieved. Thereafter, the playback of the audio and video frames is selectively modified so that frames of audio and video are played back in synchronization.

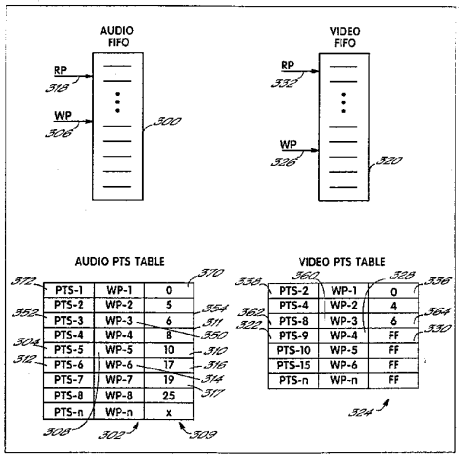


FIG. 4

EP 0 895 427 A2

LINKPLAY EXHIBIT 1004
Page 1727 of 4822

Description

[0001] This invention relates to audio-video synchronizing. A preferred form of implementation of the invention described hereinbelow relates to the digital processing of video to be displayed on a television screen and, more particularly, to digitally synchronizing the audio and video being output to a video display.

[0002] Almost all televisions manufactured today are capable of interfacing with different sources of program materials, for example a VCR, a digital versatile or video disk ("DVD") player, cable, DSS, etc., that provide audio signals for creating sounds and associated video input signals for creating screen displays. Some of those sources provide digital audio and video input signals in accordance with the Moving Picture Expert Group MPEG-2 audio/video digital compression standard. Thus, contemporary televisions and/or DVD systems preferably have the capability of processing compressed digital input signals and providing digital output signals representing the desired images. Most often, those digital signals are converted to analog signals for use by known analog television display units.

[0003] The implementation of digital signal processing for providing a video display and associated audio from an audio-video source of program material presents numerous design challenges that were not encountered in the prior processing of analog audio and video signals. For example, with digital signal processing, the audio signals are separated from the video signals; and the audio and video are processed independently. However, the playback of the audio and video must be synchronized, so that there is a coordinated and coherent reproduction of the desired audio and video provided by the source of program material.

[0004] The program source preferably provides the audio and video data in respective data packets in an "MPEG-2" format. Each of the audio and video data pockets is received from the source of video material in a continuous data stream. Each packet of video data includes a header block followed by a data block. The data block may include any number, for example one to twenty, of frames of video data that may include a full field of video data or be a coded group of pictures that includes its own header block identifying the picture type and display order. The header block for a video data packet includes control information, for example, the identity of the format of the video data, the type of compression, if used, picture size, display order, and other global parameters. The audio data packet has a header block that again identifies the format of the audio data with instructions relating to how the audio data is to be decoded and processed to provide desired enhancements, if applicable. Following the header block, the audio data packet includes an audio data block that has any number of blocks or frames of audio data, for example, from one to approximately twenty blocks.

[0005] Selected ones of the header blocks of the audio and video data packets include a presentation time stamp ("PTS") value which is a time stamp that is applicable to that data packet. The PTS value is a time reference to a system time clock that was running during the creation or recording of the audio and video data. A similar system time clock is also running during the playback of the audio and video data, and if the audio and video data are played back at the times represented by their presentation time stamps, the audio and video data will be presented to the user in the desired synchronized manner. Therefore, the PTS is used to synchronize the presentation or playback of the audio and video data.

[0006] During the decoding of the audio data, it normally must be decompressed, reconstructed and enhanced in a manner consistent with the source of program material and the capabilities of the sound reproduction system. In some applications, audio data packets may contain up to six channels of raw audio data. Depending on the number of channels the sound reproduction systems can reproduce, for example, from two to six, the sound reproduction system selectively uses the channels of raw audio data to provide a number of channels of audio which are then stored in an audio FIFO.

[0007] The decoding of the video data normally requires decompression, conversion of partial frames into full frames and the recognition of full frames. Simultaneously with the decoding process, the frames of audio and video data are being output, that is, played back to the user; and that playback must be synchronized such that the frames of audio and video present a coordinated and coherent presentation.

[0008] As will be appreciated from the foregoing, demultiplexing the audio and video data pockets is a complex process of deconstructing the data packets and storing the necessary decoding instructions as well as the content data itself to permit the decoding and playback of the data in a synchronized manner. In accordance with one known technique the audio and video content data or raw data is stored in respective audio and video first-in, first-out ("FIFO") memories. The FIFOs have write and read pointers that are controlled by a memory controller, which, in turn, is under the general control of a CPU. The write pointers are driven as a function of the requirements of the demultiplexing process, which sequentially delivers data to each of the FIFOs. The read pointers are driven as a function of independent and parallel decoding processes, which sequentially read data from the FIFOs. In addition to loading the raw data into the FIFOs, the demultiplexing process sequentially writes the associated PTS values, if present, into memory locations in respective audio and video PTS tables. To associate the PTS values with data in the FIFOs, in addition to a PTS value, the location in the respective FIFO of the first byte of data received after the PTS, is typically written into the respective audio and video PTS table.

[0009] While the audio and video data is being loaded

LINKPLAY EXHIBIT 1004
Page 1728 of 4822

into the FIFO memories by the demultiplexing process, audio and video data is simultaneously and in parallel being read from the respective FIFOs during audio and video decoding and playback processes. While both are occurring, a supervisory process must monitor the time synchronization of the video and audio data being produced by the video and audio decoding processes. In the known technique described above, this is done by relating the read pointers in the FIFOs, as they are driven by the decoding processes, to the memory locations stored in the PTS tables. When the read pointer is sufficiently close to a stored location associated with a PTS, it can be determined that the PTS identifies the current time of the associated decoding process. PTS values identified in this manner may be compared to determine whether one decoding process is ahead or behind of another.

[0010] Unfortunately, this approach has distinct disadvantages which arise from the fact that, during the audio and video decoding processes, the read pointers for the respective FIFOs are automatically and continuously driven by decoding processes interacting directly with the memory controller independent of any instructions from the CPU. This must be the case because the entire process of demultiplexing audio and video data, as well as decoding and outputting the data must occur continuously in a synchronized manner.

[0011] The above-described technique for synchronizing the audio and video decoding and playback processes presents a significant challenge, due to delays inherent in the interaction of the supervisory process with the various decoding processes. Considering, for example, the decoding of audio data, assume that the audio decoder delivers a start audio frame interrupt to the CPU running the supervisory process each time decoding of an audio frame commences. At the start of an audio frame, the supervisory process must associate the data currently being read from the audio FIFO with its corresponding presentation time stamp ("PTS"), that is, the PTS value that was loaded in the audio PTS table when the data currently being read was written into the audio FIFO. Theoretically, if the location of the read pointer at the beginning of the audio frame is known, that location can be compared with the write pointer locations that were stored in the PTS table during the demultiplexing process. If a correspondence between the current read pointer location and a stored write pointer location can be found, then the PTS associated with the stored write pointer corresponds to the PTS of the audio data currently being identified by the read pointer. If the PTS value for the data being read can accurately be determined, then the decoding and playback processes may be instructed in the known manner to skip or repeat frames in order to provide a synchronized output of frames of the audio and video.

[0012] However, there are two conditions which may result in the above process, on occasion, failing to achieve synchronization in the playback of the audio and video data. The first condition arises because the CPU running the supervisory process must time share between supervision of the audio and video decoding process, and the demultiplexing process. Accordingly, the CPU must respond to each supervised process using prioritized interrupt based communication scheme. Further, the CPU communicates with the memory controller and other functional units over a shared, time multiplexed communication bus or channel. Therefore, when a start audio frame interrupt is received by the CPU, it may not be processed immediately because the CPU is processing other interrupts of an equal or higher priority. Further, even if the CPU services the start audio frame interrupt immediately, it must then communicate with the memory controller over the time multiplexed bus. Access to the bus is arbitrated and the CPU may not have the highest priority. However, during the delay in first, processing the start audio frame interrupt and second, communicating with the memory controller over the time multiplexed bus, the decoding process for the audio data FIFO continues to read audio data from the FIFO. Therefore, when the start audio frame interrupt is serviced by the CPU and the CPU is able to communicate with the memory controller, the location of the audio data FIFO read pointer obtained by the CPU will normally be different from its location when the start audio frame interrupt was initially received by the CPU. Thus, when the CPU responds to the interrupt and obtains the current read pointer for the audio FIFO from the memory controller, the read pointer will no longer have the value it had at the time the interrupt was generated. Therefore, the result of the delay is some inaccuracy in the identity of read pointer location obtained by the CPU.

[0013] The second condition of concern is that the audio packets being read may be small and processed very quickly. Therefore, the PTS table may have two PTS entries with FIFO location values that are very close. Hence, when an inaccurate read pointer location is compared to the write pointer location values in the PTS table, an incorrect PTS entry may be associated with the start of the audio frame being decoded. Thus, there will arise a small loss of synchronization between the presentation of the audio and video frames. A single occurrence of the above event might not be detectable by the user. However, if the time to service start audio frame interrupt is longer and the processing time required for the audio data packet is very short, or if several such audio data packets occur successively, the loss of synchronization may be greater. Furthermore, the loss of synchronization in the audio process is cumulative of losses of synchronization in other decoding processes. Thus, accumulated loss of synchronization can occur to the point where the loss of synchronization is disturbing or perceptible to the user.

[0014] Consequently, in a system such as that described above, there is a need to improve the association of stored PTS values (stored, for example, in PTS tables) with the audio and video data being read from

LINKPLAY EXHIBIT 1004
Page 1729 of 4822

storage (for example from respective FIFO memories) during the decoding process.

[0015] First to third aspects of the present invention are set forth in claims 1, 32 and 33 hereof, respectively.

[0016] A preferred form of implementation of the present invention described hereinbelow provides a method and apparatus for improving the synchronization of the playback of the audio and video frames from a program source. The preferred form of implementation of the invention has the further advantage of providing a wide range of flexibility with respect to how the synchronization is implemented. Either the playback of the audio or the playback of the video may be used as a master to which the other is synchronized. In another embodiment, the system time counter is the master and the playback of the frames of audio and video are independently synchronized to the system time counter.

[0017] In accordance with the principles of the present invention and in accordance with the described embodiments, a method is described for associating an audio presentation time stamp ("PTS") value with an output audio frame that is a part of the sequence of audio frames that are derived from demultiplexing and decoding audio input data in respective audio data packets. The frames of audio are played back with frames of video being a part of a sequence of frames of video derived by multiplexing and decoding video data in respective video data packets. Selected ones of the audio and video data packets include respective audio and video PTS values representing desired playback times of the respective audio and data associated therewith. The selected ones of the audio data packets further include audio frame numbers representing a number of output frames of audio to be played back between the selected ones of the audio data packets. The method comprises the steps of first storing the audio and video PTS values in respective audio and video PTS tables during an audio demultiplexing process. In addition, the audio frame numbers are stored in frame counters in association with respective PTS values during the demultiplexing process. Thereafter, the method sequentially decodes the audio and video input data to produce respective frames of audio and video which are presented to the user. With the presentation of each audio frame, the audio frame counters are selectively decremented upon detecting one of the audio frame counters having a zero value, the audio PTS value for that zero value audio frame counter is retrieved. Thereafter, the playback of the audio and video frames is adjusted so that they are played back in synchronization.

[0018] In one aspect of the invention, the playback of the frames of audio is selected to be the master with which the playback of frames of video is synchronized. In that mode, the process establishes an audio clock extension for a system time counter. The audio clock extension has a value equal to the difference between audio PTS value associated with the zero audio frame counter and the current value of a system time counter. Thereafter, the current state of the system time counter is adjusted by the audio clock extension, thereby bringing the system time counter into synchronization with the playback of frames of audio.

[0019] In another aspect of the invention, video PTS values contained in selected ones of the video data packets are stored during a demultiplexing process. Thereafter, at the start of a video decoding process, a number of video frames corresponding to the duration of the video decoding process is stored in respective video frame counters. Upon presenting each of the frames of video, selected ones of the video frame counters are decremented. Upon detecting a video frame counter having a zero value, the system checks to determine whether the video PTS value associated with the zero value video frame counter is equal to the current state of the system time counter plus the audio clock extension. If there is a correspondence between the video PTS value and the value of the system time counter plus the audio clock extension, the audio and video frames are being presented in a synchronized manner.

[0020] In accordance with the further aspect of the invention, if the video PTS values is greater than the value of the current state of the system time counter plus the audio clock extension, the video frames are being presented too fast, and a frame of video is repeated to bring it into closer synchronization with the presentation of the frames of audio.

[0021] In a still further aspect of the invention, the system determines whether the video PTS value associated with the zero value frame counter is less than the value of the current state of the system time counter plus the audio clock extension. That condition indicates that the frames of video are being presented too slowly with respect to the presentation of the frames of audio, and a subsequent frame of video is skipped, thereby bringing the presentation of the frames of video into closer synchronization with the presentation of the frames of audio.

[0022] In accordance with the further embodiment of the invention, the presentation of frames of video is selected to be the master with which the presentation of the frames of audio is synchronized. In that embodiment, a video clock extension to the system time counter is determined as a function of the difference of the video PTS value associated with the video frame counter having a zero value and the current state of the system time counter. In this embodiment, in a manner similar to that described above, an audio PTS value associated with an audio frame counter having a zero value is compared to the current state of the system time counter plus the video clock extension as determined from the video PTS value. If the audio PTS value greater than the current state of the system time counter plus the video clock extension, the audio is being presented too fast with respect to the frames of video; and therefore, a frame of audio is repeated to bring the presentation of audio and video into closer synchronization.

LINKPLAY EXHIBIT 1004
Page 1730 of 4822

[0023] Similarly, in a further aspect of this embodiment, if it is determined that the audio PTS value associated with the zero audio frame counter is less than the current state of the system time counter plus the video clock extension, the audio is being presented too slow with respect to the video and a subsequent frame of audio is skipped, thereby bringing the audio and video presentations into closer synchronization.

[0024] In a still further embodiment of the invention, neither the playback of audio frames or the playback of video frames is chosen as a master, and thus, no clock extension is determined. Instead the playback of audio and video is compared to the system time counter, and both the audio and the video are independently adjusted by repeating or skipping frames as required to maintain a correspondence to the system time counter. The above methods provide significant flexibility in synchronizing the playback of audio and video frames.

[0025] The invention will now be further described, by way of illustrative and nonlimiting example, with reference to the accompanying drawings, in which:

[0026] Fig. 1 is a schematic block diagram of only those portions of a television utilizing a digital audio-video processor in accordance with the principles of the present invention.

[0027] Fig. 2 is a flow chart illustrating the steps of a portion of the demultiplexing process executed by the demultiplexer in accordance with the principles of the present invention.

[0028] Fig. 3 is a flow chart illustrating the steps of a portion of the demultiplexing process executed by the CPU in accordance with the principles of the present invention.

[0029] Fig. 4 is a schematic block diagram illustrating audio and video FIFOs and tables within the memory of the processor utilized in executing the audio and video processing as illustrated in Fig. 2.

[0030] Fig. 5 is a flow chart illustrating the steps of a portion of an audio decoding process executed by the CPU in accordance with the principles of the present invention.

[0031] Fig. 6 is a flow chart illustrating the steps of a portion of a video decoding process executed by the CPU in accordance with the principles of the present invention.

[0032] Fig. 7 is a flow chart illustrating the steps of portion of a process executed by the CPU for playing back output frames of audio data in accordance with the principles of the present invention.

[0033] Fig. 8 is a flow chart illustrating the steps of a portion of a process executed by the CPU for playing back output frames of video data in accordance with the principles of the present invention.

[0034] Referring to Fig. 1, a television 18 contains a digital audio/video processor 20 that receives on input 22 a continuous serial stream of data packets containing the audio and video information required to create the desired video images. The audio and video data may be provided by any one of several devices, for example, a VCR or DVD player, cable, a DSS receiver, a menu display commanded within the television 18, etc; and the video information may be supplied in any one of several different formats. Normally, the audio and video data packets are provided in the MPEG-2 format.

[0035] The audio and video data packets are received and demultiplexed continuously in independent parallel data streams. Further, the decoding and playback of output frames of audio and video data is also performed continuously in parallel data streams independent of the demultiplexing processes. Further, demultiplexing is a process that varies significantly in real time, depending on the nature of audio and video data being received. In addition, the number of video frames to be presented and their order of presentation cannot be determined from the raw video data being received. The creation of video frames and their order of presentation is a function of the decoding process and is determined primarily by the control data in the header portion of the video data packet. Similarly, the raw audio data being received in the data packet bears little resemblance to the audio data output and presented, and the frames of audio data to be presented are created during the decoding process of the audio data.

[0036] Finally, it should be noted that output audio frames can be of any length in real time, and further, several audio frames may be associated with single video frame, or in contrast, a single audio frame may be presented during video produced by several video frames. However, it is required that the frames of audio and video be played back in a synchronized manner to provide a coordinated and coherent presentation to the user. To facilitate the coordination of the presentation of the frames of audio and video data, selected ones of the audio and video data packets contain a presentation time stamp "PTS", which is a time reference to a system counter that was running during the creation or recording of the audio and video data. A similar system time counter is running in the CPU 26 during the decoding and playback of the frames of audio and video, and audio and video PTS tables are created during the demultiplexing process. If there is perfect synchronization, when the frames of audio and video are output, their stored PTS values will be the same as the current state of the system time counter. However, the differences in the processing of the audio and video data in separate parallel bit streams does not facilitate such precise timing control. Therefore, for that and other reasons, the playback of the frames of audio and video may lose synchronization with the system time counter. The net result is a loss of synchronization in the audio and video being played back to the user.

[0037] Fig. 2 is a flow chart illustrating the general operation of the demultiplexer 19 of Fig. 1. At 202, the input 22 to the multiplexer 19 continuously receives an input bit stream of data containing in random order, audio, video and subpicture data packets. The header block of

LINKPLAY EXHIBIT 1004
Page 1731 of 4822

data is extracted at 204, and video data packets are identified at 206. A video demultiplexing interrupt is provided at 208 to the CPU 26 (Fig. 1); and at 210, the video data is sequentially stored in a video data first-in, first-out ("FIFO") buffer 320 (Fig. 4) in memory 24. In a similar process, audio data packets are identified at 212, and an audio demultiplexing interrupt is provided at 214 to the CPU 26. At 216, the video data is sequentially stored in a video data FIFO buffer 300 (Fig. 4) in memory 24 (Fig. 1). Subpicture data packets are identified at 218, and a subpicture demultiplexing interrupt is provided at 220 to the CPU 26. At 222, the subpicture data is sequentially stored in a subpicture FIFO buffer (not shown) in memory 24. While the demultiplexer 19 performs other processing, it does not relate to the synchronization of the audio and video and therefore, will not be further described.

[0038] The demultiplexing process continues in the CPU 26 as illustrated in Fig. 3, and Fig. 4 is a schematic representation of how various portions of audio and video data are partitioned in the memory 24 (Fig. 1). In addition to the audio and video data FIFOs 300, 320 (Fig. 4), the memory 24 includes audio and video PTS tables 302, 324. Referring to Fig. 3, at 250, the CPU 26 services the interrupt from the demultiplexer 19 and determines at 252 whether the interrupt is for an audio PTS interrupt. If so, at 254, the PTS value in the header block of the audio data packet is loaded into audio PTS table 302, for example, at location 304. Further, the location of the write pointer 306 of the FIFO 300 associated with the location of the first byte of audio data loaded in the FIFO 300 is stored in the audio PTS table 302, for example, at location 308.

[0039] As previously mentioned, a PTS is only provided with selected audio data packets, however, those same audio data packets also contain, in their header data, a field having an audio frame number representing the number of frames of audio data to be output between the current PTS in the current audio data packet and the next audio data packet containing a PTS value. Further, the audio PTS table 302 includes frame counters, for example, memory locations at 309. And, during the demultiplexing of an audio data packet having a PTS value, that PTS value is loaded into an appropriate memory location, for example, memory location 304. Further, the number of audio frames until the next PTS value is added to the count in the corresponding frame counter, that is, frame counter 310, and the sum is written into the frame counter associated with the next PTS, that is, frame counter 316. For example, if, in the header block for the PTS loaded in memory location 304, the number of frames to the next audio data packet with a PTS is 7, that number, 7, is added to the current count, 10, in frame counter 310; and the sum, 17, is written the counter location 316 to be associated with the next audio data packet having a PTS value.

[0040] As will subsequently be explained, while the audio demultiplexing process is being executed, the process of decoding and playing back output audio frames is also running simultaneously and in parallel with the demultiplexing process. Further, with the output of each audio frame, the audio playback process decrements all of the frame counters in the audio PTS table 302. Therefore, when the count for the next PTS is loaded into location 316, the count loaded into location 310 most probably has been decremented from its original value. Consequently, as will be appreciated, the values in the frame counters 309 of the audio PTS table 302 are continuously changing as audio data packets are being demultiplexed and audio data is written into the FIFO 300, and in a simultaneously running parallel process, the audio data is also being read from the FIFO 300, decoded into output audio frames and output to the user.

[0041] If, at 256 of Fig. 3, the CPU 26 determines whether the interrupt is for a video PTS. If so, at 258, the video PTS value is sequentially stored in video PTS table 324 (Fig. 4) at, for example, location 322. Further, the location of the write pointer 326 of the video FIFO 320 when it stores the first byte of video data for that packet is written into video PTS table 324, for example, at location 328. The video PTS table 324 also has a frame counter, and during the demultiplexing process, the CPU sets a frame counter location, for example, location 330, associated with the PTS in location 322, to a non-functional value, for example, FF in hexadecimal notation.

[0042] If, at 260 of Fig. 3, the CPU determines that the interrupt is for a subpicture PTS; at 262, the CPU stores the subpicture PTS into an internal register (not shown). A PTS value is included with every subpicture data packet, and the corresponding subpicture is output when the system time clock equals the stored PTS. As subpicture synchronization is not as critical as audio/video synchronization, processing of the subpicture data will not be discussed further.

[0043] After servicing a demultiplexing interrupt, the demultiplexing process as described with respect to Figs. 1-4 continues in a similar manner. Raw audio data in the next audio data packet is sequentially loaded into the audio FIFO 300. If the next audio data packet does not have a PTS in its header block, a PTS entry is not made to the audio PTS table 302. If, however, the next audio data packet contains a PTS, that PTS is written into table 302, for example, at location 312. The write pointer location at which the write pointer 306 loads the first audio data for the current audio data packet into FIFO 300 is loaded into location 314. In addition, the number of frames of audio data between the current PTS and the next PTS is added to the count in counter location 316 and loaded into the frame counter location 317. Further, with each successive video data packet, the raw video data is sequentially loaded in the video FIFO 320; and if appropriate, PTS values are loaded in the video PTS table 324 with the respective write pointer locations. The demultiplexing process of the audio, vid-

LINKPLAY EXHIBIT 1004
Page 1732 of 4822

eo and subpicture data packets proceeds in the same sequential manner to load the respective FIFOs and PTS tables with the appropriate data.

[0044] As will be appreciated, during the demultiplexing process, data is written into the respective audio and video FIFOs 300, 320 as a function of the requirements of the demultiplexing process. Further, during the demultiplexing process, the PTS values are disassociated with their respective audio and video data and stored in the respective PTS tables 302, 324. In simultaneous, independent and parallel processes, the audio and video data is read from the FIFOs 300, 320, decoded in respective audio and video decoders 25, 23 (Fig. 1) and output to the user. During the decoding process, the read pointers 318, 332 of the respective audio and video FIFOs 300, 320 are being moved automatically and continuously by a controller in the memory 24; and hence, the read pointers are not normally controlled by specific instructions from the CPU. In order to synchronize the playback of frames of audio and video data, during the decoding process, the streams of audio and video data being read from the respective FIFOs 300, 320 must be reassociated with the appropriate PTS values stored in the audio and video PTS tables 302, 324.

[0045] The audio decoding process is initiated by the audio decoder 25 (Fig. 1) detecting the extraction of the head block at 202 of Fig. 2. The audio decoder 25 then provides an audio decode interrupt to the CPU 26 and begins the process of decoding the next audio data packet. When the audio decoder 25 provides the audio decode interrupt to the CPU 26, the continuously moving read pointer 318 (Fig. 4) is identifying the current location from which the audio decoder is reading data from the FIFO 300. However, the CPU 26 must process a great number of interrupts, each having a different priority and a different real time requirement. Therefore, if the CPU 26 is processing interrupts having a higher priority than the audio decoder interrupt, there is a delay before the audio decoder interrupt can be serviced. Further, the CPU communicates with a memory controller in memory 24 and other functional units over a shared, time multiplexed communication bus or channel 29. Access to the bus is arbitrated, and the CPU may not have the highest priority. However, during the delay in first, processing the start audio frame interrupt and second, communicating with the memory controller over the time multiplexed bus 29, the audio decoder 25 is continuing to read audio data from the audio FIFO 300, and the read pointer 318 is continuously moving.

[0046] Eventually, at 400 of Fig. 5, the CPU services the audio decode interrupt; and at 402, the CPU 26 reads the current location value of the read pointer 318. However, the current location of the read pointer 318 is different from the location of the read pointer 318 when the audio decode interrupt was first provided to the CPU 26. Next, at 404, the CPU 26 scans all of the write pointer locations stored in the audio PTS table 302 to find a write pointer value that is closest to the current read pointer value, for example, at location 350 (Fig.4). The closest write pointer value is then subtracted at 408 from the current read pointer value, and the difference is tested at 410 against a maximum allowable difference value. The maximum allowable difference value is a function of the expected motion of the read pointer 318 during a maximum time delay of the CPU 26 in responding to the audio decode interrupt at 400.

[0047] If, at 410, the difference between the closest write pointer value in location 350 and the current read pointer value is equal to or greater than the maximum allowable difference value, it is concluded that there is no PTS value, including the PTS value in location 352 of table 302, associated with the audio data currently being read by the read pointer 318. If, at 410, the difference is less the maximum allowable difference value, the PTS value in location 352 associated with the identified write pointer value in location 350 is determined to be associated with the audio data being currently read by read pointer 318.

[0048] Then, at 412, the value in the frame counter 354 associated with the closest read pointer value in location 350 is evaluated to determine whether it is approximately equal to the duration or time, that is, the length, of the audio decoding process as measured in terms of number of audio frames. The real time required to decode frames of audio data can be predicted with reasonable accuracy. Further, that period of decoding time can also be calibrated or dimensioned in terms of the number of audio frames of data being decoded. Since the frame counters are decremented with each frame of audio data being output, it can be expected that the frame counter value will be equal to the number of audio frames of decoder delay. Therefore, if the value in the counter location 354 of the PTS table 302 is approximately equal to the expected delay in decoding the current audio data, at the end of that decoding process, the audio data being played back should correspond to the PTS value in the table 302 that has a zero counter value.

[0049] However, it is possible that during the decoding process a bit error occurs, and therefore, at 412, it will be determined that the counter value is not approximately equal to the number audio frames of the duration of the audio decoding process. In that event, at 414, the counter value in location 354 is modified and set to the number of audio frames of decoder delay. Hence, with each iteration through the process, any erroneous value in one of the frame counters can be detected and corrected. Alternatively, in addition to changing the value in the one frame counter, the values in the other frame counters below location 354 in the audio PTS table 302 are modified by the same incremental modification that was made to the counter location 354. In summary, during the multiplexing process, the PTS value was disassociated from the audio data in the FIFO 300; but the CPU responds to interrupts during the decoding by re-establishing the association between audio data being

LINKPLAY EXHIBIT 1004
Page 1733 of 4822

read from the FIFO 300 and PTS values in the audio PTS table 302.

[0050] Referring to Fig. 6, the video decoding process runs independently, continuously and in parallel with the audio decoding process as well as the video demultiplexing process. The audio decoder 25 (Fig. 1) initiates the audio decoding process; however, in contrast, the CPU 26 initiates the video decoding process. At 500 of Fig. 6, the CPU detects video header that was extracted from the video data packet at 204 of Fig. 2. Thereafter, the video decoding process is functionally similar to the audio decoding process, and at 502, the CPU must read the current location of the read pointer 332 (Fig. 4) of the video FIFO 320. As previously described, this operation is subject to delays encountered in the CPU 26 (Fig. 1) accessing the memory controller in memory 24 over the multiplexed communication bus 29 connected therebetween. Therefore, the read pointer 332 associated with the video data FIFO 320 will have moved from the point in time at which the CPU provided an instruction to read the pointer location and the point in time at which the current read pointer is identified by the CPU. Thereafter, at 506, the CPU finds the closest write pointer in the video PTS table 324, for example, the write pointer in location 360 corresponding to the value of the read pointer location. The difference between the values of current read pointer location and closest write pointer location is determined at 508, and at 510, that difference is compared to a maximum value. As with the audio decoding process, the maximum value is a function of the delay in the CPU 26 determining the current location of the read pointer 332.

[0051] If the difference is equal to or exceeds the maximum value it is concluded that the video data associated with the current value of the read pointer does not have a corresponding PTS value stored in the video table 324. However, if the difference between the values of the closest write pointer and the current read pointer values is less than the maximum value, the PTS value in location 362 associated with the closest write pointer value in location 360 is determined to be associated with the video data being read by the read pointer 332. Further, at 512, the counter value at location 364 of the video PTS table 324 is set to a value equal to the delay in the video decoding process as measured in terms of the number of frames of video data being decoded. As with the audio data, the real time required to decode the video data is predictable with reasonable accuracy. Further, the time required to process the video data associated with the current location of the read pointer can be calibrated or dimensioned in units corresponding to the number of frames of video data Therefore, the number frames of video data expected to be processed through the video decoder 23 prior to the video data associated with the current position of the read pointer being played back from the video decoder is loaded in the counter location 364 of the video PTS table 324. Thereafter, at 514, the CPU 26 provides a command to the video decoder 23 to initiate the video decoding process.

[0052] As described above, audio and video data is being continuously read from the respective audio and video FIFOs 300, 320 (Fig. 4); and the read data is being continuously but independently processed in parallel by the respective audio and video decoders 25, 23 (Fig. 1). During the audio decoding process, the audio decoder 25 will create an audio frame that is ready to be output to the user. At that time, the audio decoder 25 provides an audio frame interrupt to the CPU 26 and thereafter, immediately proceeds to start the playback of the audio frame. Referring to Fig. 7, the CPU 26 at 600 services the audio frame interrupt, and at 602, proceeds to decrement by one all of the frame counters 309 in the audio PTS table 302 (Fig. 4). Next at 604, the CPU 26 determines whether any frame counter state is equal to zero. If a zero value is found, for example, in counter location 370, the CPU 26 then at 606 retrieves the audio PTS value associated with the zero counter value, for example, from memory location 372.

[0053] When a counter state is found to be zero, the CPU 26 at 608, then determines whether the audio has been selected to be the master. There are several different approaches to providing audio/video synchronization. As earlier described, a system time counter is running in the CPU 26 during the playback of the audio and video frames. If there is perfect synchronization, when the frames of audio and video are output, their stored PTS values will be the same as the current state of the system time counter. However, during playback, either one of the audio or video streams of output frames may lose synchronization with the system time counter and hence, with each other. One approach to synchronization is to choose the audio to be a master and synchronize the video to the audio. A second approach is to choose the video to be a master and synchronize the audio to the video. A third approach is to synchronize both the audio and video to the system time counter. There are perceived advantages and disadvantages to each approach depending on the source of program material, the format of the program material, other capabilities of the audio/video processor, etc. Normally, the selection of an audio, video or no master is a system parameter that may be selected by the manufacturer of the television unit. Alternatively, the selection could also be provided to the user. In other applications, the master could be dynamically chosen by the audio/video processor.

[0054] If at 608 of Fig. 7, the CPU 26 determines that the audio is the master, the CPU creates at 610 an extension or offset for the system time counter equal to the difference between the current value of the system time counter and the PTS value associated with the zero counter value, that is, for example, the PTS value in table location 372. This system time clock extension is used in synchronizing video, as discussed below. If at 608, the CPU determines that the audio is not the master, then at 612, the CPU determines whether the PTS

LINKPLAY EXHIBIT 1004
Page 1734 of 4822

value associated with the PTS table counter having a zero value is within a certain range of the current value of the system counter added to the current system time clock extension. If video master mode is used, the video playback process will create and update values for the system time clock extension (in a manner similar to that described above at 608 with respect to the audio master). However, if neither the audio or the video is chosen as master, the system time clock extension will have a zero value. In step 612, the selected PTS is compared to the current value of the system time clock, added to the system time clock extension, plus a constant, limit factor. For the audio, the magnitude of the limit factor is approximately equal to the delay of the CPU in responding the PTS interrupt as well as the delay in communicating with the memory controller plus an acceptable degree of error in audio-video synchronization.

[0055] If at 612, the PTS value from the PTS table 302 is greater than the current value of the system time counter plus its extension and plus the limit factor, that means that the audio frames are being played back faster or ahead of when the system is expecting them to be played back. Thus, there is an apparent loss of synchronization between the audio frame and the video frame being presented. To correct that loss of synchronization, the CPU 26 at 614 provides instructions to the audio decoder 25 to repeat an audio frame. Repeating an audio frame while permitting the video frames to proceed through their natural progression permits the audio frames to slide back in time and regain their synchronization with the video frames.

[0056] If the comparison at 612 is negative, the CPU 26 then at 616 determines whether the PTS value from the PTS table 302 is less than the current value of the system time counter plus its extension and minus the limit factor. If so, that means that the audio frames are being played back slower or behind of when the system time counter is expecting them to be played back. To correct that loss of synchronization, the CPU 26 at 618 provides instructions to the audio decoder 25 to skip an audio frame. Skipping the audio frame while permitting the video frames to proceed through their natural progression permits the audio frames to slide forward in time and regain their synchronization with the video frames.

[0057] Referring to Fig. 8, except for its initiation, the video frame output process is similar to the audio frame output process of Fig. 7. At 700, the CPU 26 services an interrupt created from the raster scanning process which controls the playback of the video frames. After servicing the interrupt at 700, the CPU 26 then at 702 initializes various internal registers for video playback. Next, at 704, the CPU 26 decrements all of the frame counters, for example, frame counters 336, 364, etc.; but the CPU does not decrement the video frame counters having an FF value. Then at 708, the CPU determines whether any of those frame counters have a zero value. If a zero counter value is found, for example, at location of 336, then at 710, the CPU retrieves that video PTS value.

[0058] If at 712 of Fig. 8, the CPU 26 determines that the video is the master, the CPU creates at 714 an extension or offset for the system time counter equal to the difference between the current value of the system time counter and the video PTS value associated with the zero counter value, that is, for example, the PTS value in table location 338. If at 712, the CPU determines that the video is not the master, then at 716, the CPU determines whether the video PTS value associated with the PTS table counter having a zero value is within a certain range of the current value of the system counter. At 716, the CPU determines whether the video PTS value from the PTS table location 338 is greater than the current value of the system time counter plus its extension and plus a limit factor. In the case of video, the limit factor is chosen to be approximately equal to one-half the duration of the shortest video frame. If the comparison at step 716 is true, that means that the video frames are being played back faster or ahead of when the system is expecting them to be played back. To correct that loss of synchronization, the CPU 26 at 718 provides instructions to the video decoder 23 to repeat a video frame. Repeating the video frame while permitting the audio frames to proceed through their natural progression permits the video frames to slide back in time and regain their synchronization with the audio frames.

[0059] If the comparison at 716 is false, the CPU 26 then at 720 determines whether the PTS value from the PTS table location 338 is less than the current value of the system time counter plus its extension and minus the limit factor. If so, that means that the video frames are being played back slower or behind of when the system time counter is expecting them to be played back. To correct that loss of synchronization, the CPU 26 at 722 provides instructions to the video decoder 23 to skip a video frame. Skipping the video frame while permitting the audio frames to proceed through their natural progression permits the video frames to slide forward in time and regain their synchronization with the audio frames.

[0060] Thus, the audio and video output processes operate in parallel continuously and independently to provide synchronized audio and video presentations to the user. During that process, the PTS of the video frame being presented is compared to the PTS of the currently presented audio frame; and if any loss of synchronization is detected, a video frame is repeated or skipped to regain synchronization.

[0061] The system described above has a significant advantage over the prior system previously described herein. With the prior system, delays in servicing the audio PTS interrupt in combination with audio data packets that could be processed very quickly had the potential of selecting the wrong audio PTS value from the audio PTS table. In such a case, a wrong audio PTS value would be associated with the audio data being read from

LINKPLAY EXHIBIT 1004
Page 1735 of 4822

the audio data FIFO. With the prior system, the counters in the audio PTS table 302 operate the same as the video counters described with respect to Fig. 6 herein. Thus, the audio PTS table counter locations 309 are, during the audio decoding process loaded with a number of audio frames corresponding to the audio decoder time. Therefore, in the prior system if the wrong audio PTS value were chosen from the PTS table, after the decoding delay, that erroneous choice would cause an erroneous frame counter value to be stored in table 302. Further, that erroneous PTS value would then be used in an attempt to synchronize the playback of the audio and video and might result in a perceptible loss of synchronization between the audio and video being presented to the user.

[0062] With the present system, the known number of audio frames provided in the header information of the audio data packet is used in generating the frame counters in the audio PTS table 302 along with an associated PTS value as part of the demultiplexing process. Further, during the decoding process with the output of each audio frame, the frame counter is decremented, and therefore, the number of audio frames being decoded and output with respect to each PTS value in the audio PTS table 302 is accurately tracked during the decoding process of each audio data packet. This is a far more accurate approach to associating a PTS value with decoded data being output, than the prior method of comparing memory pointers to identify a PTS value. Accordingly, with the present system, memory pointer comparisons are only used to check whether the audio PTS table counter values are approximately equal to the audio decoder delay to correct bit errors should any occur.

[0063] With the present system, by using the audio frame number from the header to lock the PTS values of the decoding process to the audio frame counter in the CPU, the coordination of the audio PTS values with demultiplexed audio data can be maintained very accurately throughout the playback process. The header portion of the video data packet does not contain similar information with respect to the number of video frames to be expected between PTS values. However, with the present system, since the audio frame presentation is very precise, adjusting the audio and video synchronization pursuant to the processes of Figs. 7 and 8 provides a highly synchronized audio and video presentation to the user that conforms very closely to the desired playback.

[0064] While the invention has been illustrated by the description of a preferred embodiment and while the embodiment has been described in considerable detail, there is no intention to restrict nor in any way limit the scope of the amended claims to such detail. Additional advantages and modifications will readily appear to those who are skilled in the art. For example, the header block of an audio data packet that has a PTS value also includes the number of bytes of audio data between successive audio PTS values, in other words, the number of bytes of audio data in the current data packet plus the number of bytes of audio data in successive audio data packets until the next data packet having an audio PTS value. This data can be stored in the write pointer location instead of storing the write pointer locations in the audio and video PTS tables 302, 324.

[0065] In Fig. 4, the frame counter values are created by adding the number of audio frames from the header block of the audio data packet to the value in the prior counter and the sum is written into the counter associated with the current PTS value. Further, all of the frame counter values are decremented by one with the output of each audio frame. Alternatively, the number of audio frames from the the header block of the audio data packet may be written into the frame counter; however in this case, only the currently active counter is decremented in response to the output of an audio frame. When that counter reaches zero, then the next counter is decremented.

[0066] As described, the audio/video synchronization is normally performed with respect to the system time counter in the system. The system time counter is a counter that, when operated in the playback of audio and video data, provides counter states that are directly related to PTS values. The synchronization may be performed using any other counter or clock that has respective counter or clock states that are correlated to the PTS values. Therefore, the invention in its broadest aspects is not limited to the specific details shown and described. Consequently, departures may be made from the details described herein without departing from the scope of the claims which follow.

Claims

1. A method for associating an audio presentation time stamp ("PTS") value with an output frame of audio being a part of a sequence of frames of audio derived by demultiplexing and decoding audio input data in respective audio data packets, the output frames of audio being played back with output frames of video being a part of a sequence of frames of video are derived by multiplexing and decoding video data in respective video data packets, selected ones of the audio and video data packets include respective audio and video PTS values representing desired playback times of the respective audio and video data associated therewith, the selected ones of the audio data packets further including audio frame numbers representing a number of output frames of audio to be played back between the selected ones of the audio data packets, the method comprising the steps of:

    storing the audio and video PTS values contained in the selected ones of the respective au-

LINKPLAY EXHIBIT 1004
Page 1736 of 4822

dio and video data packets during respective audio and video demultiplexing processes;
storing audio frame numbers in respective audio frame counters during the audio demultiplexing process, each of the audio frame counters being associated with one of the stored audio PTS values;
sequentially decoding the audio and video data in the selected ones of the respective audio and video data packets to produce the frames of audio and video, respectively;
providing a simultaneous playback of the frames of audio and video to the user;
selectively decrementing the audio frame counters in response to presenting each of the frames of audio;
detecting one of the audio frame counters having a zero value; and if so,
retrieving the audio PTS value corresponding to the one of the audio frame counters; and
selectively modifying the playback of the frames of audio and video to synchronize the presentation of the audio and video to the user.

2. The method of claim 1 further comprising after the step of retrieving the audio PTS value, the steps of:

providing a audio clock extension for a system time counter approximately equal to a difference between an audio PTS value associated with the one of the audio frame counters and a current value of the system time counter; and
adjusting a current state of the system time counter by the audio clock extension, thereby bringing the system time counter in synchronization with the playback of the frames of audio.

3. The method of claim 2 further comprising the steps of:

storing at the start of a video decoding process for the selected ones of the video data packets, a number of video frames in respective video frame counters, the number of video frames representing a time duration approximately equal to a duration of the video decoding process;
decrementing all of the video frame counters in response to presenting each of the frames of video;
detecting one of the video frame counters having a zero value; and if so,
retrieving the video PTS value corresponding to the one of the video frame counters; and
determining a video PTS value associated with the one of the video frame counters to be approximately equal to the current state of the system time counter plus the audio clock extension, thereby determining that the frames of audio and video are in synchronization.

4. The method of claim 3 further comprising the steps of:

determining in response to the one of the video frame counters having a zero value that the frames of video are currently being presented to the user too fast and out of synchronization with the frames of audio currently being presented to the user; and
repeating a frame of video to bring the frames of video into closer synchronization with the presentation of the frames of audio.

5. The method of claim 4 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be greater than the current state of the system time counter plus the audio clock extension, thereby indicating that the frames of video are being presented too fast with respect to the presentation of the frames of audio.

6. The method of claim 5 further comprising the steps of:

determining in response to the one of the video frame counters having a zero value that the frames of video are currently being presented to the user too slowly and out of synchronization with the frames of audio currently being presented to the user; and
skipping a frame of video to bring the frames of video into closer synchronization with the presentation of the frames of audio.

7. The method of claim 6 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be less than the current state of the system time counter plus the audio clock extension, thereby indicating that the frames of video are being presented too slowly with respect to the presentation of the frames of audio.

8. The method of claim 7 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be less than the current state of the system time counter plus the audio clock extension minus a limit factor, the limit factor being determined as a function of the delays in decoding the audio data.

9. The method of claim 8 further comprising the step of detecting the selection of an audio master synchronization determining a synchronization of the playback of frames of video to the playback of

LINKPLAY EXHIBIT 1004
Page 1737 of 4822

frames of audio.

10. The method of claim 1 further comprising the steps of:

storing at the start of a video decoding process for the selected ones of the video data packets, a number of video frames in respective video frame counters, the number of video frames representing a time duration approximately equal to a duration of the video decoding process;
decrementing all of the video frame counters in response to presenting each of the frames of video;
detecting one of the video frame counters having a zero value; and if so,
retrieving the video PTS value corresponding to the one of the video frame counters;
providing a video clock extension for the system time counter approximately equal to a difference between a video PTS value associated with the one of the video frame counters and a current value of the system time counter; and
adjusting a current state of the system time counter by the video clock extension, thereby bringing the system time counter in synchronization with the playback of the frames of video.

11. The method of claim 10 further comprising the steps of:

determining in response to the one of the audio frame counters having a zero value that the frames of audio are currently being presented to the user too fast and out of synchronization with the frames of video currently being presented to the user; and
repeating a frame of audio to bring the frames of audio into closer synchronization with the presentation of the frames of video.

12. The method of claim 11 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be greater than the current state of the system time counter plus the video clock extension, thereby indicating that the frames of audio are being presented too fast with respect to the presentation of the frames of video.

13. The method of claim 12 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be greater than the current state of the system time counter plus the video clock extension plus a limit factor, the limit factor being determined as a function of the video frame length.

14. The method of claim 13 further comprising the steps of:

determining in response to the one of the audio frame counters having a zero value that the frames of audio are currently being presented to the user too slowly and out of synchronization with the frames of video currently being presented to the user; and
skipping a frame of audio to bring the frames of audio into closer synchronization with the presentation of the frames of video.

15. The method of claim 14 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be less than the current state of the system time counter plus the video clock extension, thereby indicating that the frames of audio are being presented too slowly with respect to the presentation of the frames of video.

16. The method of claim 15 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be less than the current state of the system time counter plus the video clock extension minus a limit factor, the limit factor being determined as a function of the video frame length.

17. The method of claim 16 wherein the limit factor is approximately equal to approximately one-half of a duration of a shortest video frame.

18. The method of claim 17 further comprising the step of detecting the selection of a video master synchronization determining a synchronization of the playback of frames of audio to the playback of frames of video.

19. The method of claim 11 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be greater than the current state of the system time counter, thereby indicating that the frames of audio are being presented too fast with respect to the presentation of the frames of video.

20. The method of claim 19 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be greater than the current state of the system time counter plus a limit factor, the limit factor being determined as a function of the video frame length.

21. The method of claim 20 further comprising the steps of:

determining in response to the one of the audio

LINKPLAY EXHIBIT 1004
Page 1738 of 4822

frame counters having a zero value that the frames of audio are currently being presented to the user too slowly and out of synchronization with the frames of video currently being presented to the user; and
skipping a frame of audio to bring the frames of audio into closer synchronization with the presentation of the frames of video.

22. The method of claim 21 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be less than the current state of the system time counter, thereby indicating that the frames of audio are being presented too slowly with respect to the presentation of the frames of video.

23. The method of claim 22 further comprising the step of determining an audio PTS value associated with the one of the audio frame counters to be less than the current state of the system time counter minus a limit factor, the limit factor being determined as a function of the video frame length.

24. The method of claim 23 wherein the limit factor is approximately equal to approximately one-half of a duration of a shortest video frame.

25. The method of claim 24 further comprising the steps of:

determining in response to the one of the video frame counters having a zero value that the frames of video are currently being presented to the user too fast and out of synchronization with the frames of audio currently being presented to the user; and
repeating a frame of video to bring the frames of video into closer synchronization with the presentation of the frames of audio.

26. The method of claim 25 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be greater than the current state of the system time counter, thereby indicating that the frames of video are being presented too fast with respect to the presentation of the frames of audio.

27. The method of claim 26 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be greater than the current state of the system time counter plus a limit factor, the limit factor being determined as a function of the delays in decoding the audio data.

28. The method of claim 27 further comprising the steps of:

determining in response to the one of the video frame counters having a zero value that the frames of video are currently being presented to the user too slowly and out of synchronization with the frames of audio currently being presented to the user; and
skipping a frame of video to bring the frames of video into closer synchronization with the presentation of the frames of audio.

29. The method of claim 28 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be less than the current state of the system time counter, thereby indicating that the frames of video are being presented too slowly with respect to the presentation of the frames of audio.

30. The method of claim 29 further comprising the step of determining a video PTS value associated with the one of the video frame counters to be less than the current state of the system time counter minus a limit factor, the limit factor being determined as a function of the delays in decoding the audio data.

31. The method of claim 32 further comprising the step of detecting the selection of a no master synchronization for which no clock extension is determined or used and both the video and audio are independently synchronized to the system time counter.

32. A method for synchronizing a play back of frames of audio and video output data, the frames of audio and video output data being derived from demultiplexing and decoding audio and video data contained in respective audio and video data packets, selected ones of the audio and video data packets containing presentation time stamp PTS values, and the selected ones of the audio data packets containing audio frame numbers, each of the audio frame numbers representing a number of frames of audio to be played back between the selected ones of the audio data packets, the method comprising:

sequentially storing during an audio demultiplexing process of the audio data packets containing PTS values, audio data in respective audio FIFO memory locations;
sequentially storing PTS values in first memory locations in an audio PTS memory table during the first demultiplexing process;
storing in second memory locations in the audio PTS table during the first demultiplexing process, write pointer locations associated with the data written into the video FIFO memory locations for the selected video data packets con-

LINKPLAY EXHIBIT 1004
Page 1739 of 4822

taining a PTS value;
storing audio frame numbers in counter memory locations in the PTS table during the first demultiplexing process;
initiating an audio decoding process simultaneously with the a start audio frame interrupt to a CPU;
acquiring a read pointer location of the audio FIFO in response to the start audio frame interrupt;
finding a write pointer location in the audio PTS table approximately equal to the read pointer location;
decoding the audio data being read from the audio FIFO;
presenting a successive audio output frame to the user;
decrementing all of the counter memory locations in the audio PTS table by one;
detecting any of the counter locations having a zero value; and if so.
adjusting a current state of a system time counter to the PTS value in the audio PTS table associated with the counter having a zero value.

33. A digital video processor for synchronizing a play back of output frames of audio and video data derived from demultiplexing and decoding audio and video data contained in audio and video data packets and selected ones of the audio and video data packets containing a presentation time stamp value, the video processor comprising:

audio and video FIFO memories for sequentially storing the audio and video data contained the respective audio and video data packets;
an audio PTS memory table having

first memory locations for sequentially storing the PTS values in the selected ones of the audio data packets,
second memory locations for storing values representing write pointer locations associated with the data written into the audio FIFO memory location in the selected audio data packets containing a PTS value, and
third memory locations functioning as counter locations for storing audio frame numbers, each representing a number of frames of audio to be played back until an occurrence of a subsequent PTS value; and

a video PTS memory table having

first memory locations for sequentially storing the PTS values in the selected ones of the video data packets,
second memory locations for storing values representing write pointer locations associated with the data written into the video FIFO memory locations for the selected video data packets containing a PTS value, and
third memory locations functioning as counter locations for storing a value representing a number of frames of video approximately equal to a delay time required for decoding a frame of video data prior to its play back to a user.

LINKPLAY EXHIBIT 1004
Page 1740 of 4822

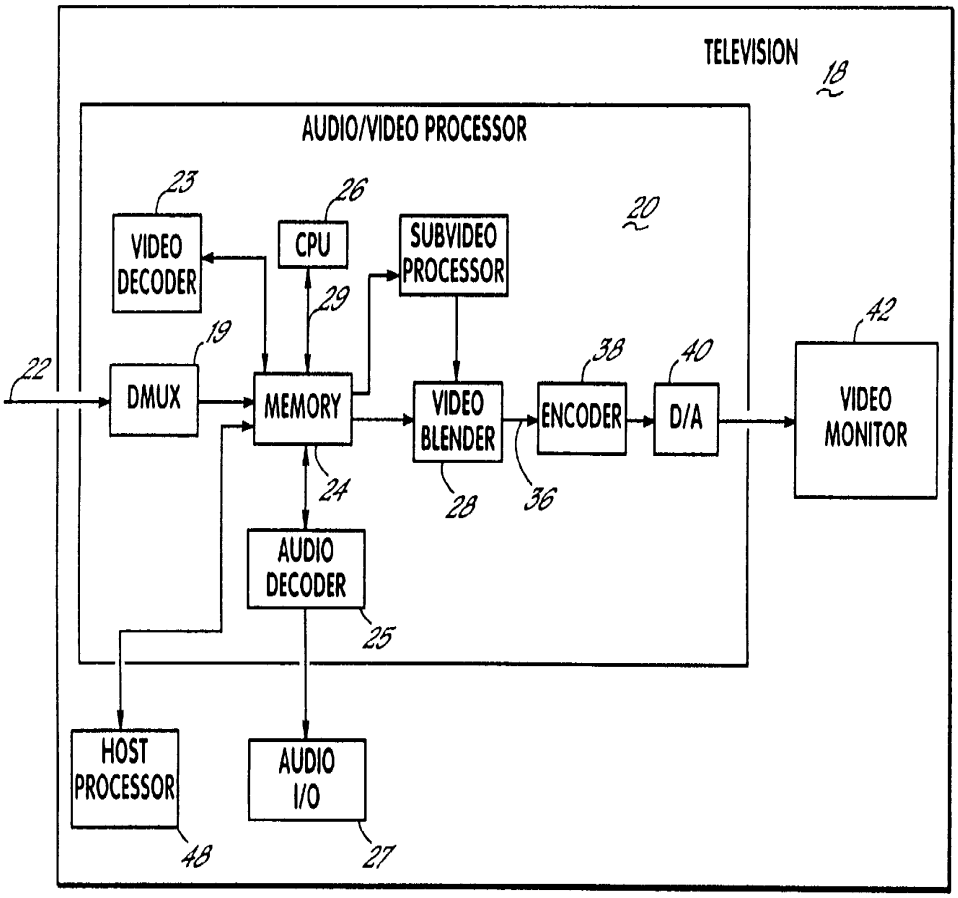


FIG. 1

EP 0 895 427 A2

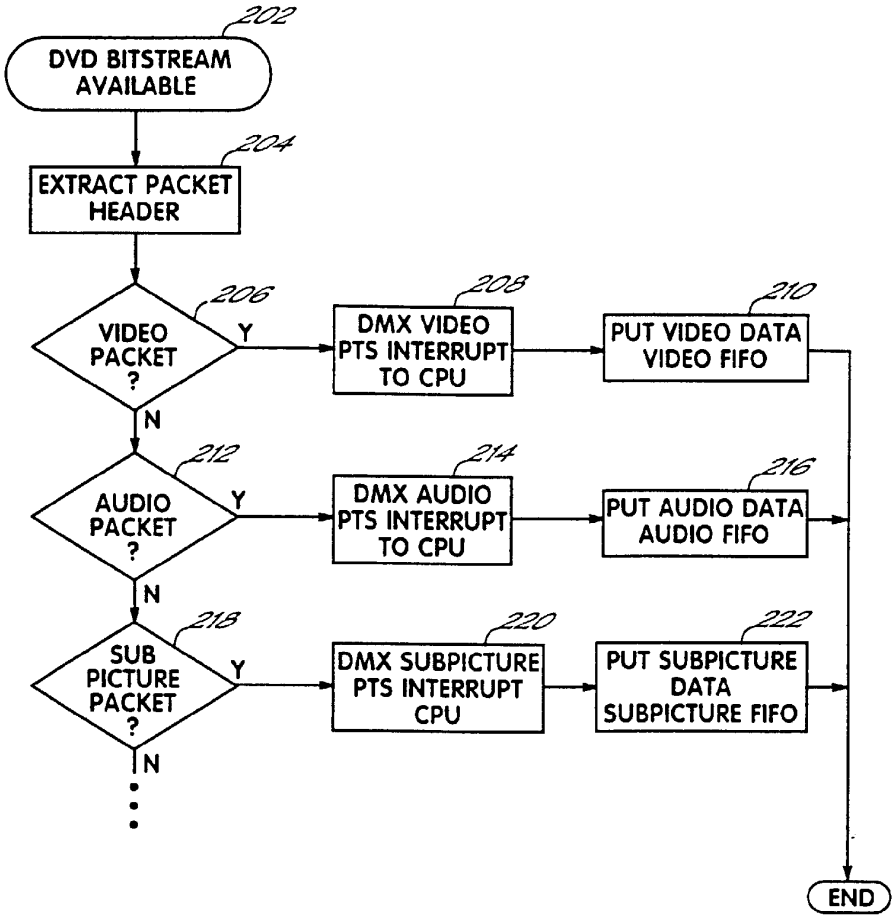


FIG. 2

LINKPLAY EXHIBIT 1004
Page 1742 of 4822

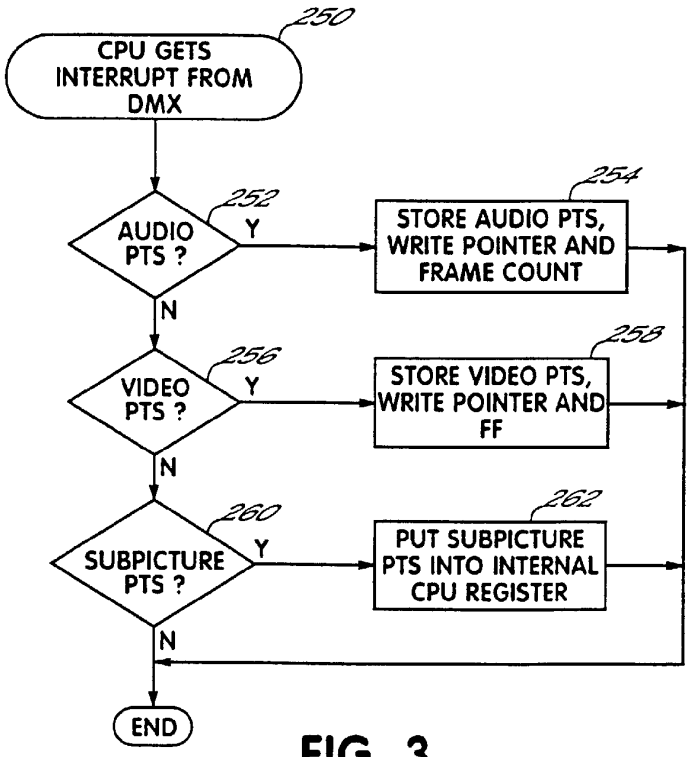


FIG. 3

LINKPLAY EXHIBIT 1004
Page 1743 of 4822

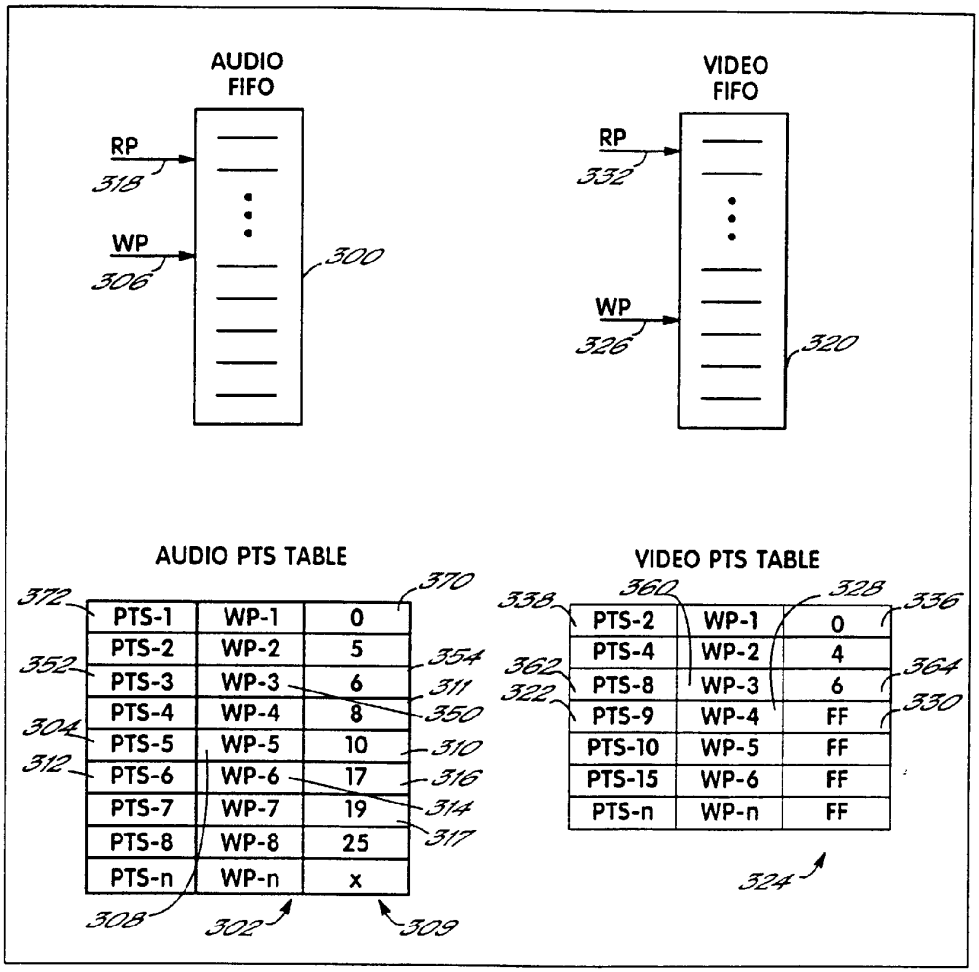


FIG. 4

LINKPLAY EXHIBIT 1004
Page 1744 of 4822

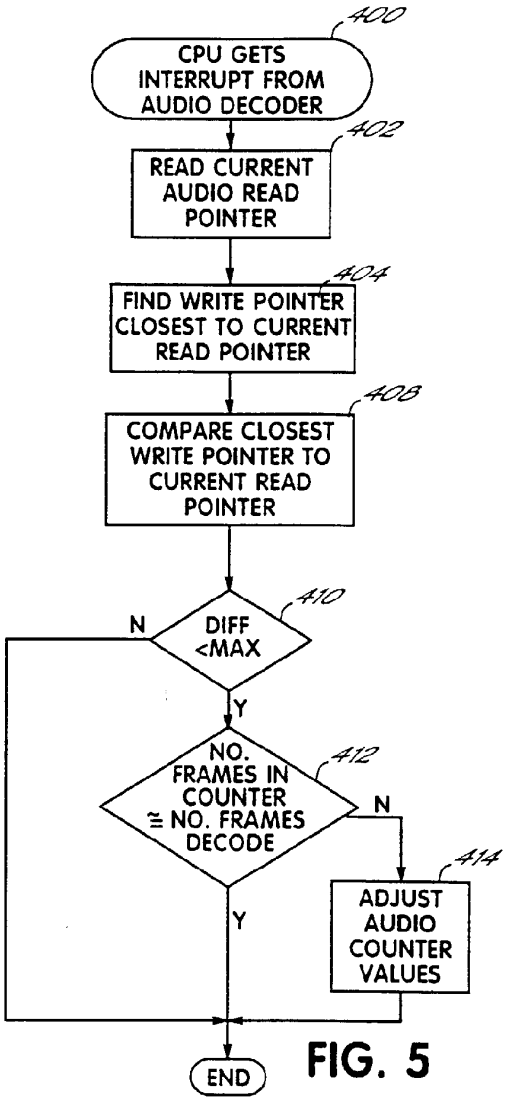


FIG. 5

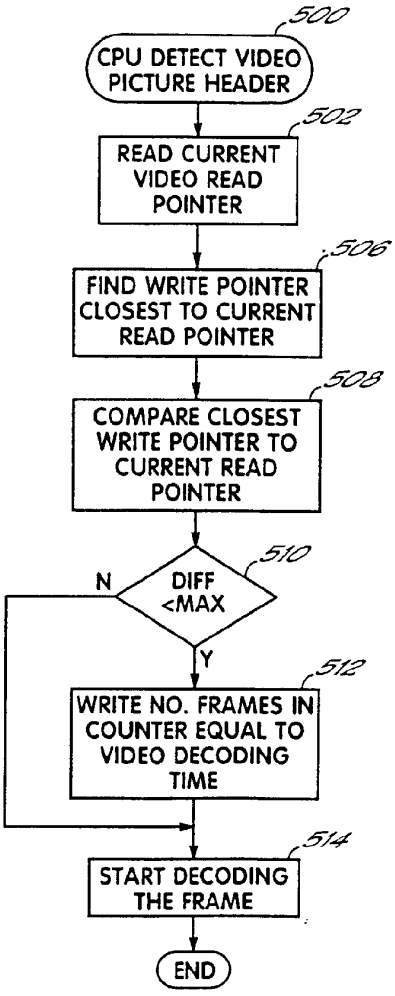


FIG. 6

LINKPLAY EXHIBIT 1004
Page 1745 of 4822

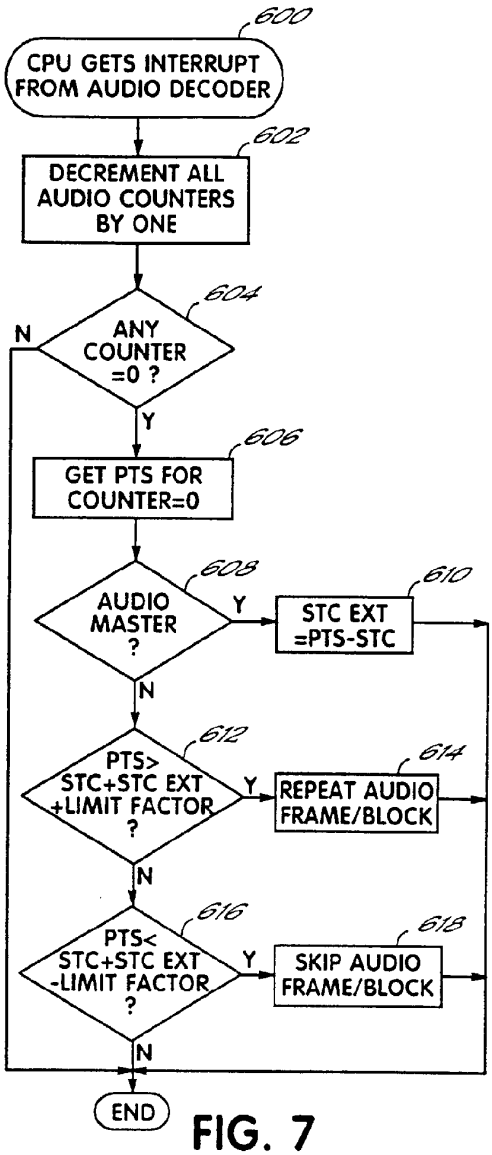


FIG. 7

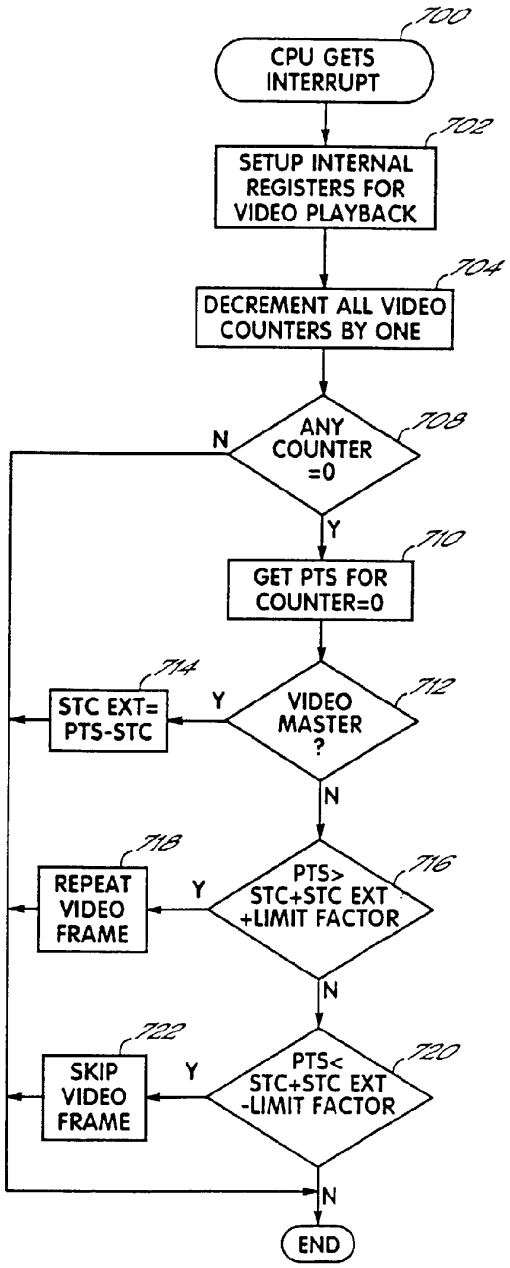


FIG. 8

LINKPLAY EXHIBIT 1004
Page 1746 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 111 527 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
27.06.2001 Bulletin 2001/26

(51) Int Cl.[7]: **G06F 17/60**, H04L 29/06

(21) Application number: **00310244.9**

(22) Date of filing: **17.11.2000**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **19.11.1999 JP 33023599**

(71) Applicant: **KABUSHIKI KAISHA TOSHIBA**
Kawasaki-shi, Kanagawa-ken 210-8572 (JP)

(72) Inventors:
- **Kizu, Sachio, Kabushiki Kaisha Toshiba**
  Tokyo 105-8001 (JP)
- **Natsubori, Shigeyasu,**
  Kabushiki Kaisha Toshiba
  Tokyo 105-8001 (JP)
- **Imai, Toru, Kabushiki Kaisha Toshiba**
  Tokyo 105-8001 (JP)
- **Amano, Shinya, Kabushiki Kaisha Toshiba**
  Tokyo 105-8001 (JP)
- **Muranaga, Tetsuro, Kabushiki Kaisha Toshiba**
  Tokyo 105-8001 (JP)
- **Takabatake, Yoshiaki,**
  Kabushiki Kaisha Toshiba
  Tokyo 105-8001 (JP)

(74) Representative: **Midgley, Jonathan Lee**
Marks & Clerk
57-60 Lincoln's Inn Fields
GB-London WC2A 3LS (GB)

(54) Communication method for data sychronization processing and electronic device therefor

(57) At member registration step, a master transmits a certificate and a priority to a slave. The certificate indicates that the slave receiving it belongs to the same data synchronization group as the master. The priority is used to solve a data conflict. At data synchronization step, after it is determined that members belong to the same data synchronization group by the certificate, data synchronization processing is performed between arbitrary members in which an input device and a display device such as wrist watch are limited. Data synchronization does not require a specific server.

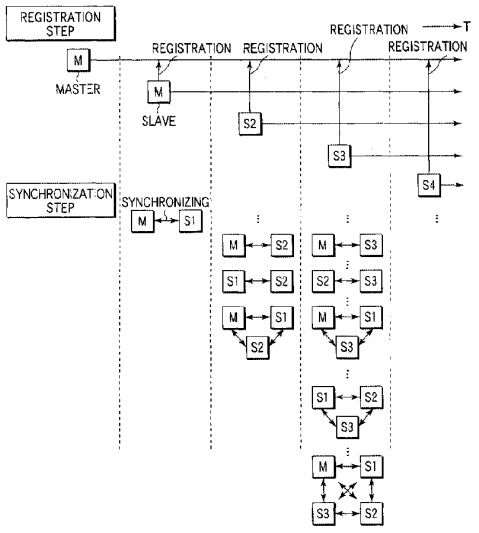


FIG. 1

EP 1 111 527 A2

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The present invention relates to a communication method for synchronously processing data between a plurality of devices and an electronic device therefor. In particular, the present invention relates to a communication method and an electronic device that have been modified so as to synchronize data in a wireless or wired network such that a network configuration is dynamically changed.

[0002] In recent years, apart from a personal computer, in a portable information processing terminal such as a portable cellular phone, a palm top computer or a wrist watch, personal data such as a telephone number (address) directory, schedule data, a task list, an E-mail can be managed and stored. Using this function, data is shared by using duplicate data in order to use the same data at home or outside as well as in office between electronic devices such as a personal computer or a portable information processing terminal. As a technique for keeping coincidence of a plurality of shared data, a data synchronization processing is known. For example, in the case where an attempt is made to share personal data such as an address directory or a schedule between a personal computer and a portable information processing terminal, duplication of data is created, and original data is held in the host and the duplicate data is held in each device other than the host. In the case where a change occurs with these items of data, the changed data is exchanged between these devices, whereby data consistency is maintained. This method is referred to as data synchronization.

[0003] As a typical computer program for data synchronization, Intelli Sync (Intelli Sync User's Guide Rev 3.1 Puma TECHNOLOGY 1999) is known. This enables data synchronization between personal computers or between a personal computer and a portable information processing terminal. This program enables data synchronization between two personal computers, one personal computer and one portable information processing terminal, or one personal computer and two portable information processing terminals.

[0004] In the case of this system, however, a device serving as a server must be participated in data synchronization processing. In the case where three or more devices share data, there applies a restriction that data synchronization cannot be performed between two devices other than server. Therefore, for example, in the case where a personal computer connected to a network is used as a server, thereby causing two portable information processing terminals to include the duplicate data, even if an attempt is made to use duplicate data after updated in one portable information processing terminal by the other portable information processing terminal outside, data synchronization processing cannot be performed by these portable information processing terminals other than server. Stated another way, the conventional data synchronization processing is performed between the server and the client in a one-to-one manner. It is not possible to perform the data synchronization processing in a one-to-plural manner.

[0005] Furthermore, in general, it is required to provide communication settings for a data synchronization party, settings regarding type of synchronization data, or setting of a method of solving a problem during data conflict. These settings are conventionally provided by a user operating an input device of a personal computer or portable information processing terminal through the setting screen displayed on a display device of the personal computer or portable information processing terminal and inputting data. This means that at least the display device and input device of the portable terminal device are required. Therefore, in the conventional method, because of a restriction specific to device such as a wrist watch including device size, for example, it is very difficult to provide settings of data synchronization in a device having only a limited input device or display device.

[0006] Accordingly, it is an object of the present invention to provide a communication method and electronic device that enables data synchronization between arbitrary devices without requiring a server.

[0007] According to the present invention, there is provided a communication method for synchronizing data between a plurality of arbitrary devices using a wireless or wired network, the communication method comprising a member registration step and a data synchronization step. In the member registration step, a device specified as a master by a user transmits to a device specified as a slave by the user a certificate indicating that the slave belongs to a data synchronization group that performs a data synchronization processing and a priority to be used when the slave performs the data synchronization processing, and then, the slave is registered as a member of the data synchronization group. In the data synchronization step, it is determined using the certificate whether or not one slave and one or more other slaves capable of making wireless or wired communication belong to the same data synchronization group, and the data synchronization processing is performed through wireless or wired communication in accordance with the priority between the slaves which are determined to belong to the same data synchronization group.

[0008] In this communication method, apart from the data synchronization step of actually processing data synchronization, the member registration step is provided for the purpose of specifying a target device for data synchronization. At this member registration step, the certificate and priority are transmitted from a device specified as the master to a device specified as the slave, whereby the slave can be registered as a member of the data synchronization group. The certificate and priority are provided to each member, whereby data synchronization can be easily performed between slaves even if the master is absent, as long as these devices

LINKPLAY EXHIBIT 1004
Page 1748 of 4822

are registered in the same data synchronization group.

[0009] In this case, as to whether or not the slaves capable of making wireless or wired communication belong to the same group, it is determined by exchanging the certificate between the slaves. Thus, even if a master does not exist, data synchronization can be performed reliably and safely between the slaves. In the case where a conflict occurs during data synchronization processing between slaves, such conflict can be solved by the priority assigned to each slave at the member registration step. Therefore, each slave may hold only the certificate and priority assigned to each slave itself, making it unnecessary to set data or the like concerning a communication party for data synchronization. Hence, a device having only its limited input device and display device can be used as a device targeted for synchronization. Further, as in a wired or wireless network such that a network configuration is dynamically changed, even in a case of use under environment that makes it difficult to provide data for statically specifying a communication party for data synchronization, a device that is such data synchronization party can be reliably specified.

[0010] The data synchronization step can be performed between one master and one or more arbitrary slaves having a certificate indicating that these slaves belong to the same group as well as between slaves.

[0011] In addition, there is no need to specify a master and a slave by devices, and an arbitrary device can be specified as a master or slave of the data synchronization group by data targeted for data synchronization. One device may function as a slave of the data synchronization group for one set of data or may function as a master of the data synchronization group for another set of data. Alternatively, one device may function as a slave of two data synchronization groups corresponding to two sets of different data.

[0012] In the case where a master and a slave are thus specified individually by data targeted for synchronization, the member registration step is also executed between a device specified as a master of a data synchronization group for the data targeted for data synchronization processing and a device specified as a slave of the master. Thus, for transmission of the certificate and priority as well, a master that is a transmission source and a slave that is a transmission destination are set by data, and the certificate and priority are set by data accordingly.

[0013] In addition to the aforementioned member registration step and data synchronization step, a master privilege transfer step may be provided, whereby only the master's function can be transferred to another device while the function and configuration of the current data synchronization group are maintained intact. Hence, a newly purchased device can be used as a master, and a slave can be temporarily used as a master outside.

[0014] In addition, it is preferable that a priority exchange step of exchanging a priority between slaves be further provided. In this manner, the priority registered in the member registration step can be changed later, and data synchronization processing can be achieved more flexibly between slaves.

[0015] The master privilege transfer step and priority exchange step as well are preferably performed after it has been determined that there does not exist any device other than two devices set in a mode for executing the above steps.

[0016] In order to easily change a member of the data synchronization group or the like, it is preferable to further provide a member registration releasing step of releasing a master or a slave from the data synchronization group. The slave releasing step may be performed by that slave device solely. However, the master releasing step is desirably executed together with the member registration releasing step of a slave communicable with the master. In this manner, a plurality of devices including a master can be released from the data synchronization group with one operation.

[0017] In addition, a conventional data synchronizing method assumes one-to-one cable connection, one-to-one connection with the server by modem, or one-to-one connection with the server by infrared-ray radiation. This is because a communication party for data synchronization can be uniquely specified.

[0018] However, if the data synchronization is performed in a bus type wired network or a wireless network free of directivity, unlike a case of one-to-one connection, it is required to uniquely identify a party for data synchronization from the devices in a network by any method. In addition, in the case of a wireless or wired network such that a network configuration is dynamically changed, the network address of each device is not fixed, and a network address is often assigned during network connection or the like. Thus, the network address of a communication party cannot be used as static identification information for specifying a communication party for data synchronization. Therefore, in a wire or wireless network such that a configuration is dynamically changed as well, it is required to introduce a new technology for making it possible to uniquely identify a party for data synchronization.

[0019] Although it is not presumed in one-to-one connection, in a bus type wired network or a wireless network free of directivity, there is a possibility that a device other than device for data synchronization, for example, a malicious third party exists in the same network, especially in a wireless network. Thus, it is required to ensure data security, and at the same time, to prevent data leak due to computer hacking or the like. In particular, in a wireless network, there is high possibility that a device is merely disposed within the reach of wireless communication, whereby the device can be easily participated in the network. Thus, data security is important. In addition, without maliciousness, there is a danger that data contained in a device other than that targeted for

LINKPLAY EXHIBIT 1004
Page 1749 of 4822

synchronization, the device existing within the reach of wireless communication, is mistakenly or unintentionally changed by data synchronization processing between devices via wireless communication.

[0020] According to the present invention, such an undesired data synchronization is prevented from being occurred since the party for data synchronization can be easily identified by exchanging the certificate between the devices.

[0021] This summary of the invention does not necessarily describe all necessary features so that the invention may also be a sub-combination of these described features.

[0022] The invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a view illustrating a member registration step and a data synchronization step that are employed in a communication method according to one embodiment of the present invention;
FIG. 2 is a view showing an example of data set of each device employed according to the present embodiment;
FIG. 3 is a block diagram depicting the configuration of each device employed according to the present embodiment;
FIG. 4 is a view showing an example when functions provided to each device employed according to the present embodiment are classified by communication mode;
FIG. 5 is a view showing master and slave data exchange during data synchronization member registration according to the present embodiment;
FIG. 6 is a flow chart showing procedures for master processing during data synchronization member registration according to the present embodiment;
FIG. 7 is a flow chart showing procedures for slave processing during data synchronization member registration according to the present embodiment;
FIG. 8 is a view showing an example of communication mode during certificate data exchange and synchronizing process in the present embodiment;
FIG. 9 is a view showing another example of communication mode during certificate data exchange and synchronizing process in the present embodiment;
FIG. 10 is a view showing an example of processing at the time of data conflict in the present embodiment;
FIG. 11 is a flow chart showing procedures for master declaration processing in the present embodiment;
FIG. 12 is a flow chart showing procedures for slave processing in data synchronization member release processing according to the present embodiment;
FIG. 13 is a flow chart showing procedures for master processing in data synchronization member release processing according to the present embodiment;
FIG. 14 is a view showing master and slave data exchange in master privilege transfer processing according to the present embodiment;
FIG. 15 is a flow chart showing procedures for master processing during master privilege transfer processing according to the present embodiment;
FIG. 16 is a flow chart showing procedures for slave processing during master privilege transfer processing according to the present embodiment;
FIG. 17 is a view showing data exchange between devices during priority exchange in the present embodiment;
FIG. 18 is a flow chart showing procedures for device processing in priority exchange processing according to the present embodiment;
FIG. 19 is a timing chart showing set mode and execute mode applied to the present embodiment;
FIG. 20 is a view showing an example of an interface of a wrist watch to which the communication method according to the present embodiment is applied; and
FIG. 21 is a view showing an example of an interface of a portable information terminal to which the communication method according to the present embodiment is applied.

[0023] A preferred embodiment of a communication method for data synchronization processing according to the present invention will now be described with reference to the accompanying drawings.

First Embodiment

[0024] Hereinafter, an embodiment of a method and apparatus for synchronously processing data according to the present invention will be described with reference to the accompanying drawings.

Data Synchronization Group

[0025] Data synchronization denotes processing for, in the case where part of duplicated data is asynchronously updated independently relevant to plural sets of the duplicated data, entirely synchronizing the plural sets of these duplicated data. For example, assume that a set of data (data set) that a device A has is duplicated in a device B. Synchronization processing corresponds to processing for synchronizing data sets of the devices A and B each other. If a partial data field of the data set that the device A has is changed, a partial data field of the data set that the device B has is updated during next synchronization processing.

[0026] If the same data field of the data sets of the devices A and B is changed during and after the previous synchronization processing, in order to synchronize

LINKPLAY EXHIBIT 1004
Page 1750 of 4822

these data fields, it is required to select which of the data field values has a higher priority. When the data in the same data field is changed with a plurality of devices during or after synchronization processing, it is referred to as data conflict.

[0027] In the present invention, unlike a conventional method, data synchronization processing between information devices is not performed by one-to-one processing between a server and a client. This processing is performed based on a data synchronization group constructed exclusively for synchronization processing. Of the information devices capable of making data communication wirelessly or using a wire, a group configured between devices that performs data synchronization is referred to as a data synchronization group (or network). Namely, a data synchronization group composed of devices that should perform data synchronization processing corresponds to a data synchronization group. In this case, such data synchronization group may not be always configured by devices, and an arbitrary data synchronization group can be configured by each data set that enables data synchronization. Namely, each information device can configure the data synchronization group by the number of data sets that can be managed.

[0028] In addition, information devices may be a device employed for limited use such as wrist watch or may be a device having a plurality of functions such as personal computer. These information devices are not restricted by their sizes or the types or functions of input/output devices or display devices and the like. The data synchronization group is configured by a plurality of information devices. A device configuring one data synchronization group is referred to as a member of the data synchronization group.

[0029] A data synchronization group is a permanent concept. Data synchronization processing is performed by a member of the data synchronization group that is configured in advance, unlike a case in which a data synchronization group is configured every time data synchronization processing is performed.

Data Synchronization Group Communication

[0030] A communication method in a data synchronization group comprises a member registration step and a data synchronization step. FIG. 1 shows an example.

[0031] The member registration step is a step to be performed when a member is added to a data synchronization group. At this member registration step, a certificate and a priority are transmitted from a device specified as a master by a user to a device that the user want to newly register as a slave. The certificate indicates that a device receiving it belongs to the same data synchronization group (data synchronization network) as the master. That is, the certificate corresponds to a group ID or the like. The priority is used when a device receiving it performs data synchronization processing, and is employed to solve data conflict.

[0032] The data synchronization step is a step that performs data synchronization processing between a plurality of arbitrary members configuring a data synchronous group.

[0033] In FIG. 1, M denotes a master of a data synchronization group. The master M is a core of the data synchronization group. Only one master exist in one data synchronization group, and has a privilege of adding a member to the data synchronization group. S_1 to S_4 denote slaves of the data synchronization group. One or more slaves exist in one data synchronization group. Data synchronization member registration is performed between the slave and the master M in the data synchronization group, whereby the slave becomes a member of the data synchronization group.

[0034] A master and a slave may not be always specified by devices, and an arbitrary device can be specified as a master or slave of the data synchronization group corresponding to the data sets. Thus, one device may function as a slave of the data synchronization group relevant to one set of data or may function as a master of the data synchronization group relevant to another set of data. Alternatively, one device may function as each of the slaves of the two data synchronization groups corresponding to two different types of data.

Member Registration Step

[0035] The member registration step is executed by making one-to-one communication between a device that is a master of a data synchronization group and a slave that newly participates in the data synchronization group. Two members configure the data synchronization group at a stage at which a first slave S_1 is registered in the master M. Subsequently, the member registration step is sequentially performed similarly between the master M and each of the slaves S_2 to S_4 to be newly registered, whereby the number of members configuring the data synchronization group increases one by one. The member registration step is not performed every data synchronization. Once a device is registered as the member, such registered device is continuously a member of the data synchronization group unless the device has been released explicitly. Data synchronization processing can be executed any time between the members in the same data synchronization group.

Data Synchronization Step

[0036] At the data synchronization step, data is synchronized between the members that belong to a data synchronization group. Here, all the members may perform data synchronization processing at one time or arbitrary members may perform data synchronization processing. In addition, the master M may be or may not be participated in data synchronization processing. Ac-

LINKPLAY EXHIBIT 1004
Page 1751 of 4822

cording to the present invention, data synchronization processing can be performed between slaves. A plurality of synchronizing processes configured by other members can be performed independently at an arbitrary time, and a plurality of these synchronizing processes may be performed simultaneously in another place.

[0037] For example, as shown in FIG. 1, in the case where only the master M and the slave S_1 configure a data synchronization group, the data synchronization step is executed between these two devices, namely between the master M and the slave S_1. Alternatively, in a state in which a slave S_2 is newly registered, the data synchronization step can be executed between the master M and slave S_1, between the master M and slave S_2, between the slave S_1 and the slave S_2, or between the master M and the slaves S_1 and S_2. Hereinafter, the synchronization step is executed similarly between arbitrary two devices irrespective of the master or slave.

Member Configuration

[0038] Now, the data and functional structures set to each of the master and slave devices will be described with reference to FIG. 2 and FIG. 3.

[0039] Set data for setting the data synchronization group that each device stores, includes master/slave specification data, identification data, certificate data, priority data, and a counter value.

[0040] The master/slave specification data is like a flag, and its initial value is blanked. When a device is registered as a master or slave, the corresponding flag is set.

[0041] The identification data denotes data for uniquely specifying a member in one data synchronization group, and is used as a key for generating certificate data. The synchronization group is defined by data type, and thus, the identification data differs depending on type of data set in one device. The identification data is generated in each device when one device is registered as a slave of a data synchronization group or when a device is operated as a master by a master declaration described later.

[0042] The certificate data corresponds to the aforementioned certificate, and is employed to certify that one device belongs to one data synchronization group.

[0043] For example, during data synchronization assume that it is authenticated by a public key encryption system as to whether or not devices belong to the same data synchronization group and encryption of synchronization data after authenticated. The identification data and the certificate data are employed as follows in an authentication and encryption system.

[0044] First, each of the master and slave generates a secret key and a public key. A pair of these secret key and public key corresponds to the identification data of each device. When the identification data is transmitted to another device, only the public key that is part of the identification data is transmitted. During data synchronization group registration, when a slave passes a public key as the identification data to a master, the master returns this public key of the slave as a certificate after the public key has been encrypted by own secret key. In this manner, the slave has own secret key and public key; and a certificate certified by the master and the public key of the master. Here, this certificate and the public key of the master are referred to as certificate data.

[0045] During actual authentication and encryption, when a device performing authentication receives a certificate from the counterpart, this certificate is decrypted by own public key of the master. If the certificate is correctly decrypted, it means that the device of the counterpart belongs to the same data synchronization group. Then, the public key of the counterpart is extracted from the decrypted data, and the synchronization data is encrypted by the extracted public key of the counterpart, and is transmitted, whereby synchronization data can be transmitted safely.

[0046] The certificate data may certify reliably that each device belongs to a data synchronization group. The data may be generated by other well known various methods without being limited to the aforementioned method.

[0047] The priority data is used to solve conflict of data synchronization as described previously. The data is used to determine a precedence (namely, priority) for solving such data conflict.

[0048] The counter value is data used to count the number of slaves registered by the master and determine the priority of that slave. The highest priority of the master is "1". The counter value is counted up every time a slave is registered, and the priorities "2", "3", ... are assigned to the slaves in order of registration. This counter is used by the master only.

[0049] As described above, the data synchronization group is configured by each data set. Thus, the master/slave specification data, certificate data, identification data, priority data, and counter value are provided and managed by each data set. That is, in a device having a plurality of data sets, the certificate data, identification data, priority data, and counter value are not managed in one-dimensional manner, and are manages in two-dimensional manner, namely, by each data set, as shown in FIG. 2. FIG. 2 shows an example when a device belongs to three data synchronization groups corresponding to three types of data sets, i.e., "address directory", "time", and "E-mail", respectively. In this case, in a device having these three types of data sets, whether the device operates as a master or slave by each data set is managed. The counter value is valid only for a data set of the data synchronization group that operates a master. In addition, the aforementioned member registration step is performed by data synchronization group, the certificate data, identification data, priority data, and counter value are independent of each other by each data set, respectively. The secret key and public

LINKPLAY EXHIBIT 1004
Page 1752 of 4822

key configuring identification data are generated by the member registration step by employing random numbers, for example. The value of the identification data is continuously maintained until it has been removed from the data synchronization group in which a device is registered as a member at the member registration step.

[0050] In this manner, in the case where set data on a data synchronization group is managed by each data set, when one performs each of the functions including data synchronization member registration, data synchronization processing, master declaration, data synchronization member releasing, master privilege transfer, and priority exchange, a step is executed such that a user can select a target data set from among the data sets in the row direction shown in FIG. 2 in order to specify what is a data set targeted for processing.

[0051] FIG. 3 shows the functional configuration of each device. This network configuration is provided as an example when a device can be configured as a master and a slave.

[0052] As shown in FIG. 3, each device includes a communication device 11, a communication controller 12, a device input/output device 13, a device input/output controller 14, an authentication unit 15, a data synchronization group setting and management unit 16, a data synchronization processing unit 17, and a synchronous data storage device 18.

[0053] The communication device 11 makes communication with another device via a wireless or wired communication. This device 11 transmits data to another device, and receives data from such another device. As a wireless communication, there can be used an electric wave communication, free of directivity such as "Bluetooth", for example. In addition, a bus type LAN such as "Ethernet" can be used as a wired network. In either of the wireless and wired communication, the network configuration may not be statically determined. The present invention is applicable to a network such that the network configuration is dynamically changed.

[0054] The communication controller 12 controls data reception and/or transmission. The controller 12 includes a function for checking data and a request received by the communication device 11; a function for transmitting and receiving a processing request and response from the device input/output controller 14; a function for making a data transmission and reception request for the communication device 11; and a function for transmitting and receiving a request and a response among the authentication unit 15, data synchronization group setting and management unit 16, and data synchronization processing unit 17, respectively.

[0055] The device input/output device 13 is a user interface, and is an input device or a display device of own device.

[0056] The device input/output controller 14 controls data request and transmission and/or reception relevant to the device input/output device 13. The controller 14 includes a function for checking data and a request received by the device input/output device 13; a function for making a processing request for the communication controller 12; a function for receiving the processing request from the communication controller 12; a management function for writing, reading, deleting data relevant to the synchronization data storage device 18; and a function making a request for and a response to the data synchronization group setting and management unit 16.

[0057] The authentication unit 15 authenticates certificate data received from another device in response to an authentication request from the communication controller 12 by correlating the data with certificate data of the own device, the data being owned by a certificate data management controller 161 through the data synchronization group setting and management unit 16. The unit 15 determines whether or not another device belongs to its data synchronization group, i.e., whether or not both of the own and other devices belong to the same data synchronization group.

[0058] The data synchronization group setting and management unit 16 generally manages settings of the data synchronization group. The unit 16 includes a certificate data management controller 161 and a priority management controller 162. The data synchronization group setting and management unit 16 manages own settings by using the certificate data management controller 161 and the priority management controller 162. The unit 16 transfers data or a processing request to the communication controller 12, authentication unit 15, device input/output controller 14, data synchronization processing unit 17, priority management controller 162, and certificate data management controller 161.

[0059] The certificate data management controller 161 has a region for storing certificate data certified by a master of the own device and a region for storing identification data that is a source of the certificate data. The controller 161 manages certificate data and identification data for generating such certificate data. The controller 161 includes functions for generating, deleting, and setting the certificate data and the identification that is a source of the certificate data according to a request from the data synchronization group setting and management unit 16.

[0060] The priority management controller 162 has a region for storing priority data on own device; a region for storing a counter value changed when a priority is issued in a master; and a counter device. The controller 162 includes a function for processing requests for generating, setting, deleting, and transferring priority data from the data synchronization group setting and management unit 16; and a function for resetting and incrementing the counter device.

[0061] The data synchronization controller 17 performs synchronization processing for a data set targeted for synchronization (synchronization data). The controller 17 includes a function for reading, writing, and deleting data relevant to the synchronization data storage device 18; a function for solving data conflict in view of the

LINKPLAY EXHIBIT 1004
Page 1753 of 4822

priority of own device and that of another synchronization device; and a function for processing a request from the communication controller 12 and delivering data.

[0062] The synchronization data storage device 18 stores synchronization data, and has a region for storing data.

[0063] The configuration of this device is provided as an example when a device can be configured as a master and a slave. This configuration may include a device (member) that can be configured as a slave only or a device (member) that can be configured as a master only. In this case, there can be provided a configuration in which functions or configurations required for a slave and a master are eliminated. For example, a certificate data producing function and a counter device and its control function and the like may be provided to only a device that can operate as a master, and is not required for a device that functions as a slave only.

[0064] Now, data synchronization group management and operation will be described here.

[0065] First, all functions owned by each device in a data synchronization group will be described with reference to FIG. 4.

[0066] A device that can configure a data synchronization group includes a total of six functions. The device includes a function to be processed without making communication with another device; a function requiring one-to-one communication between one device and another device; and a function requiring communication between one device to a plurality of devices (actually, made of a plurality of one-to-one communications). The "master declaration" function is a stand alone function that is processed without making communication with another device. The "data synchronization member registration", "master privilege transfer", and "priority exchange" functions are processed through one-to-one communication. The "data synchronization" function is performed between two or more devices. The "data synchronization member releasing" function is a stand along function in a slave, and a function that is processed through communication with one or more slaves in a master. Hereinafter, these functions will be described in detail. These functions can be performed by data synchronization group. When a function is read, it is required to specify an identification code of the data synchronization group.

(1) Data Synchronization Member Registration

[0067] In the case where a new member is added to a data synchronization group, "data synchronization member registration" processing is performed. This processing is adopted to achieve the aforementioned member registration step. The processing is performed by making data communication between a master of a data synchronization group and a member to be newly added as a slave. The master and newly registered member must be communicable with each other via a wireless or wired communication channel while power is supplied. The user calls the "data synchronization member registration" function each in the master and the member to be registered. Subsequently, each of the master and the member to be registered is placed in a member registration mode. In order to enter the member registration mode, the user interaction and exclusive control for another device are used such that the user holds down the registration button in each of the master and the new member, for example, and registration is permitted only when only two devices in the member registration mode exist, and security authentication during registration is provided. It is possible to determine whether or not only two devices in the member registration mode exist by checking a response to a broadcast from the master, for example. This prevents accidental registration of a device owned by a malicious third party within the reach of electric waves.

[0068] The effective communication range means that a device is within the reach of wireless communication in a wireless network. In a wired network, this range means a limited network area such as sub-network, for example, in which broadcasting is effective. Namely, in the range which has a possibility that a third party receives the communication, under the condition that there exists any device other than only two devices, i.e., a device targeted for member registration and a master, which are set in the member registration mode, data synchronization member registration is started.

[0069] Next, a function for registering a data synchronization member will be described in detail. In FIG. 5, there is shown master and slave communication processing to be performed together with synchronization member registration.

[0070] First, both of the master M and the slave S transmit to the counterpart a request for registering a data synchronization member, and then, the master notifies its identity to the slave. In this state, the master has all set data, but the slave has only identification data. When the slave receives the notification, it transmits the identification data to the master. The master transmits to the slave the certificate data for this slave generated using the identification data and the priority data, and communication processing is completed.

[0071] Now, the processing step in each device will be described in detail. In FIG. 6, there is shown the step of master processing when a master device registers a slave device as a member of the data synchronization group. A description is given with reference to FIG. 3.

[0072] First, when a user call of the "data synchronization member registration" function is transmitted to the device input/output controller 14 (step S11), the current setting of the data synchronization group (whether the own device is a master or slave) is determined at the data synchronization group setting and management unit 16 (step S12). If the own device is set as a master by a master declaration function described later, the user is prompted for a request for executing regis-

LINKPLAY EXHIBIT 1004
Page 1754 of 4822

tration processing. When it is set as a slave, processing is terminated. In the case where nothing is set (i.e., blank data is specified), the device operates as a newly registered member described later (FIG. 7).

[0073] In the case where the own device is set as a master, when the device input/output device 13 receives a request for executing registration processing from the user (step S13), the communication device 11 transmits a request for registering a data synchronization member in a broadcasting manner (step S14). Then, the communication device 11 waits for a registration request from a new member. Although not shown, in the case where a registration request is not made for a predetermined time, processing terminates with time out. When a registration request from a new member is received (step S15), the registered device determines whether or not only two devices requires the registration, i.e., the own device and a new member (step S16). Namely, after a request for registering a data synchronization member has been transmitted between the own device and the counterpart, if only one counterpart transmits such request for registering a data synchronization member, it is determined that only two device, i.e., the own device and new member participate the registration.

[0074] In the case where any device other than only two devices, i.e., the own device and new member participate the registration (NO at the step S16), processing is terminated. In the case where only two devices participate the registration (YES at step S16), the communication device 11 notifies its identity to the new member (step S17). Then, when identification data that is a source of certificate data from the new member is received by the communication device 11 (step S18), the data synchronization group setting and management unit 16 generates the certificate data and the priority data to be passed to the new member, based on this data (steps S19 and S20). Further, the value of the counter that counts the number of registered devices is incremented by one (step S21).

[0075] Last, the communication device 11 transmits the generated certificate data and the priority data to the new member (steps S22 and S23). Although the counter value is used as the priority data, the priority may be determined in any system as long as a precedence can be defined by generating different values every issue, thereby making it possible to uniquely judge the priority among these generated values.

[0076] Now, in FIG. 7, there is shown the step of processing a new member when a desired device (member) newly participating in a data synchronization group is registered as a data synchronization group member by a master device. A description will be described with reference to FIG. 3.

[0077] When the device input/output device 13 receives the user request for "data synchronization member registration", as in the case with FIG. 6, the current data synchronization group setting is determined by the data synchronization group setting and management unit 16. In the case where the own device is set as a master, the aforementioned "data synchronization member registration" (FIG. 6) is performed. In the case where the own device is set as a slave, processing terminates. If a data synchronization group is not set (blank data is specified as a master or slave), the user is prompted for request for executing registration processing. When the user makes a request for executing registration processing to the device input/output device 13 (step S31), the communication device 11 receives a registration request from the master (step S32), and transmits such registration request to the master (step S33). Although not shown, in the case where there exists no response from the master within a predetermined period of time, processing terminates with time out. After the registration request has been transmitted to the master, the registered device determines whether or not only two devices, i.e., the own device and master participate the registration (step S34). In the case where only the two devices participate the registration, the own device waits for the counterpart master notification. In this manner, it is determined that the counterpart is a master (step S35), the data synchronization setting and management unit 16 generates identification data that is a source of the certificate data (step S36), and transmits the data to the master (step S37).

[0078] When the communication device 11 waits for the certificate data and priority data from the master, and receives them, the certificate data and priority data are set through the data synchronization group setting and management unit 16, and registration processing completes (steps S38 and S39).

[0079] In registration processing of a data synchronization member, it is required to share a data set of the same type between the master issuing the certificate data and priority and the new member to be registered. In the case where an attempt is made to register data sets of their different types, it is required to disable registration. This collation step can be performed by checking whether or not identifiers 01, 02, and 03 are assigned in advance to an address directory, a time, and schedule, for example, during "transmission of a request for registering a data synchronization member" (step S14) and during "transmission of a request for registering a data synchronization member" (step S33), transmitting these identifiers between the master and the new member, and checking whether or not the identifiers are identical to each other.

[0080] Once it is guaranteed that a master and a slave that belong to a data synchronization group and a slave have a data set of the same type, it is authenticated as to whether or not the master and the slave belong to the same data synchronization group when another function is used. Thus, there is no possibility that data sets of the different types coexist. Even in a device having a plurality of data sets, the data sets in that device configure, respectively, other completely independent data synchronization groups, the data sets of different types

LINKPLAY EXHIBIT 1004
Page 1755 of 4822

and processing do not coexist.

(2) Data synchronization processing

[0081] Now, data synchronization processing will be described here.

[0082] When a master in a data synchronization group transmits certificate data and priority data to each member by data synchronization member registration processing, actual data synchronization processing can be performed between arbitrary members (even without intervening a master) by using at least these two sets of data.

[0083] During synchronization processing, the members belonging to the same data synchronization group first authenticate their identities each other, and then, perform data synchronization. Sets of data that do not conflict with each other are merged, respectively. In the case where a conflict occurs with some sets of data, the data with the highest priority is employed as data for solving a conflict.

[0084] As shown in FIG. 8, assume that a data synchronization group is configured by five members A, B, C, D, and E, wherein the member A is a master.

[0085] In the case where data synchronization processing is performed when all the members are present, one member transmits own certificate data to another member, receives certificate data from the counterpart, and authenticates their identity with each other. In transmission and authentication of certificate data, there is provided a method for all members transmitting/receiving certificate data to/from all the counterparts, thereby individual members using their received certificate data to authenticate their identities. Namely, among the members A to E, certificate data transmission/reception and authentication may be performed according to the number of combinations of different two members.

[0086] Alternatively, as shown in FIG. 9, certificate data transmission/reception and authentication may be performed only between a member with the highest priority (the member A here) and another member. Namely, there is provided a method in which the pairs of members such as A and B, A and C, and A and D, and A and E (the member A serves as a server), thereby performing authentication. After each of the members A to E has been authenticated to be a member of the same data synchronization group, in actual data synchronization processing, a member with the highest priority (the member A here) transmits/receives synchronization data in a data field to/from all other members, checks a conflict, solves such conflict based on the priority, and changes data so that the member can sequentially process the synchronization data in the next data field.

[0087] A conflict during data synchronization processing occurs, as shown in FIG. 10, in the case where the members A, C, and E have not been changed relevant to a data field 'x' but data owned by the members B and D is updated since previous synchronization processing. Means for solving such data conflict is to employ data owned by the member B with the highest priority. This is because, although the member A has the highest priority of the five, data is changed in the members B and D only. Even if data is not changed, a conflict during synchronization processing may be solved using the data owned by the member A with the highest priority.

[0088] Next, assume that only the members A, B, and C in this data synchronization group perform data synchronization. Such situation can occur when the members D and E are not turned ON or when only the members A, B, and C are carried to the outside, and data synchronization is performed outside. Authentication and data synchronization are performed in the same manner as described above. Then, assume that data owned by the members B and C conflict with each other (that is, assume that the data in the same data field owned by the members B and C has been updated during and after the previous synchronization). Such conflict is solved by using the data of a member having the highest priority of the members B and C. As a result, the data owned by the members A, B, and C are synchronized with each other. Assume that the remaining members D and E performs data synchronization in another place. As is the case with the members A, B, and C, if a conflict occurs, such conflict is solved by using the data with the highest priority of the members D and E, and the members D and E undergoes data synchronization.

[0089] As described above, data synchronization processing can be performed between slaves as well as between one master and one or more arbitrary slaves, each of which has a certificate indicating that they belong to the same data synchronization group that performs data synchronization processing. In this case as well, certificate data is transmitted between devices capable of making wireless or wired communication, whereby it is determined that these devices belong to the same data synchronization group. Thus, data synchronization can be performed safely by using only the certificate data without having specific data for uniquely specifying the counterpart.

[0090] Data synchronization processing can be performed as follows, for example.

(a) A device set in a data synchronization processing mode transmits a data synchronization processing request in a broadcasting manner to other devices that exist within a communicable range;

(b) A certificate is transmitted between these devices, and it is authenticated as to whether or not they belong to the same data synchronization group each other; and

(c) When the above devices belong to the same data synchronization group, data synchronization of the corresponding data sets is executed in accordance with each priority.

LINKPLAY EXHIBIT 1004
Page 1756 of 4822

[0091] In the case of data synchronization processing, unlike registration processing of a data synchronization member as described in FIG. 6 and FIG. 7, there does not apply a restriction that only two devices, i.e., a master and a member targeted for new registration. Data synchronization processing can be performed between two or more arbitrary devices that exist within a communicable range provided they belong to the same data synchronization group. All the devices to be synchronized may be set in a data synchronization processing mode; only one device may be set in a data synchronization processing mode by the user, and another device receiving a data synchronization processing request from that device may automatically enter the data synchronization processing mode according to that request.

[0092] In addition, in actual data synchronization processing, the user may specify a data set targeted for data synchronization processing in the data sets for which data synchronization groups are set. Alternatively, it is determined as to whether devices belong to the same data synchronization group relevant to all the data sets for which the data synchronization groups are set, whereby these synchronizing processes are executed simultaneously relevant to the data sets which belong to the same data synchronization group. Authentication data exists by each data set for which the data synchronization group is set. Thus, all the authentication data is exchanged each other, for example, thereby making it possible to check a data synchronization group to which both of the counterpart device and the own device belong. Of course, the identifier assigned to each data set is transmitted, whereby a data synchronization group to be authenticated may be specified.

(3) Master Declaration

[0093] Now, a "master declaration" function for setting the own device as a master of a data synchronization group will be described here.

[0094] In the case where a data synchronization group is newly constructed, it is required to determine a member as a master of the network. The "master declaration" function is provided to achieve this. The "master declaration" can provide setting such that only a member having a data set, which does not belong to any data synchronization group, can be a master of a data synchronization group that the member will construct. That is, a device that has already been set as a slave cannot be changed to a master. This setting is provided by a stand alone, and communication with another member is not performed.

[0095] In the existing data synchronization group, the "master declaration" cannot be performed by changing to a master a member that does not belong to the data synchronization group or a member that has been set as a slave or temporarily executing "data synchronization member releasing" for a slave to remove the slave from the data synchronization group, thereafter adding it as a master again. This is required to ensure security. In the latter case, there is defined another new data synchronization group in which a member performing "master declaration" is set as a master. The "master declaration" is used any time only when a new data synchronization group is constructed.

[0096] In FIG. 11, there is shown the processing step of "master declaration" in which a new member is set as a master of a new data synchronization group by the "master declaration".

[0097] When the user first makes an operation for calling the "master declaration" function for a data set through the device input/output device 13 (step S41), the data synchronization group setting and management unit 16 checks the current setting of the own device in the data synchronization group (step S42). In the case where the device has been set as a master or slave, processing is terminated. If no data synchronization group is defined (i.e., if blank data is specified as a master or slave), the data synchronization group setting and management unit 16 generates identification data on a data synchronization group (step S43), and generates authentication data based on the generated identification data (step S44).

[0098] Further, in the data synchronization group setting and management unit 16, a priority counter is reset, and the initial value of the counter is set as priority data indicative of own priority (steps S45 and S46). Then, the counter is incremented, and processing is terminated.

(4) Data Synchronization Member Releasing

[0099] "Data synchronization member releasing" is performed when a registered device is released from a data synchronization group. This operation differs depending on a master and a slave, and thus, a description will be given separately.

[0100] First, a description will be given with respect to a case in which a slave is released from a data synchronization group. This function is used when a slave participating in a data synchronization group is used while the slave is excluded from the member of the synchronization group or when a slave device is discarded. The "data synchronization member releasing" is a stand-alone operation, and communication with a master or another device is not required. The existing data synchronization group is not affected by this function. The priority data owned by the released slave will be vacant because it is not reused in "data synchronization member registration". However, this does not mean that consistency of the data synchronization group is not destroyed.

[0101] In FIG. 12, there is shown the processing step when "data synchronization member releasing" is performed in a slave.

[0102] First, when a user make a request for "data synchronization member releasing" for the device input/

LINKPLAY EXHIBIT 1004
Page 1757 of 4822

output device 13 relevant to a data set (step S51), the data synchronization group setting and management unit 16 determines the current setting of the own device in the data synchronization group (step S52). When the own device is a slave, the data synchronization group setting and management unit 16 deletes the corresponding priority data, certificate data, and identification data (steps S53, S54, and S55). This releasing is performed by each synchronization group, and thus, network identification data is also specified when the releasing function is called.

[0103] In the case where a master is released from a data synchronization group, if a slave currently communicable with the master exists when "data synchronization member releasing" is performed, this slave is also released from the data synchronization group. This is because the master generally dismembers a slave without transferring its master privilege when the master desired to release the slave from the data synchronization group. In addition, only a communicable slave is released from the data synchronization group simultaneously because it is not assumed that all members in the data synchronization group are always present by the fact that the master cannot grasp all the members in the data synchronization group and the fact the slave is placed in another place due to purchase of a new device, discarding, faults, theft or the like. In this case, a slave that does not exist on the spot still participates in the previous data synchronization group, and data synchronization can be performed between the slaves that do not exist on the spot. However, in this case, since a master does not exist any longer, a data synchronization group member cannot be increased by newly performing data synchronization member registration.

[0104] In FIG. 13, there is shown the processing step when "data synchronization member releasing" is performed in a master. When a user makes a request for "data synchronization member releasing" relevant to a data set to the device input/output device 13, as described in FIG. 12, the data synchronization group setting and management unit 16 determines the current data synchronization group. In the case where the own device is a master, this device searches for a currently communicable slave (step S61). This may be determined according to a response from the slave relevant to data transmitted in a broadcasting manner. Then, the own device transmits a request for "data synchronization member releasing" and the master certificate data to a slave within the currently communicable range (steps S62 and S63).

[0105] The slave receiving this request performs authentication by using the master certificate data, and then, resets own setting as in "data synchronization member releasing" in the slave.

[0106] The master is then deletes the priority data, certificate data, identification data, and counter values of the own device, the counter is reset, and processing is terminated (steps S64 to S67).

(5) Master Privilege Transfer

[0107] The "master privilege transfer" function allows a data synchronization group to continuously exist intact in the existing data synchronization group, and allows the current master to transfer only the master's privilege to another member (slave). This function is used when one attempt to transfer a master privilege to another device while maintaining the functions and configuration of the data synchronization group, including cases in which where one attempts to specify a newly purchased device as a master of the existing data synchronization group; a master device is discharged; one attempts to change the current master device to a new device; and, when a desk top personal computer that is not mobile is specified as a master, one attempt to temporarily perform "data synchronization member registration" outside. This means that the slave and master privileges are exchanged in the existing data synchronization group. Alternatively, the master privilege may be transferred to a member that is not included in a data synchronization group by automatically exchange the privileges in combination with "data synchronization member registration".

[0108] In this process, the "master privilege transfer" function is called by a master and each slave device that transfers the master privilege. In a method similar to the "data synchronization member registration", the set data held by the master is transferred to a slave, and the set data held by the slave is transferred to the master. The data held by each of the master and slave is exchanged intact. Individual members do not have data concerning another device each other, and thus, these individual members do not affect those in another data synchronization group. Thus, processing can be performed safely and simply as in the "data synchronization member registration".

[0109] Now, the "master privilege transfer" function will be described in detail,

[0110] Here, master and slave communication processing to be performed together with the "master privilege transfer" processing will be described with reference to FIG. 14.

[0111] First, both of the master and the slave transmit a master privilege transfer request to each other. When both of them receive this request, they transmit certificate data for device authentication. After authentication processing, the master transmits the identification data, priority data, and counter value to the slave. Alternatively, the slave transmits the priority data and identification data, and communication processing is completed.

[0112] Now, the processing steps in each device will be described in detail. FIG. 15 shows the processing steps in a master device having the "master privilege transfer" function. A description will be given with reference to FIG. 3.

[0113] When the device input/output device 13 receives the "master privilege transfer" request from the

LINKPLAY EXHIBIT 1004
Page 1758 of 4822

user (step S71), the data synchronization group setting and management unit 16 judges the current setting of the own device in the data synchronization group (step S72). When the device is a master, the device input/output device 13 continuously waits for a request for executing processing from the user. When the device input/output device 13 receives a request for executing processing (step S73), the communication device 11 transmits a request for master privilege transfer processing in a broadcasting manner (step S74). Then, the communication device 11 waits a request for such transfer processing from the counterpart device. When such request from the counterpart device arrives (step S75), the communication device 11 and the communication controller 12 determine whether or not only the own device and the counterpart device are executable the master privilege transfer processing (step S76). In the case of affirmative determination (YES at the step S76), the communication device 11 transmits certificate data to the counterpart device (step S77). The sequence of the steps of transferring certificate data and determining the number of executable devices may be reversed. When the communication device 11 receives certificate data from the counterpart device (step S78), the authentication unit 15 and data synchronization group setting and management unit 16 authenticate the counterpart device, and determines that the counterpart device exists in the same data synchronization group as the master (step S79). If authentication fails, processing terminates (NO at the step S79). When authentication is successful (YES at the step S79), the data synchronization group setting and management unit 16, communication controller 12, and communication device 11 transmit the identification data, priority data, and counter value owned by the own device to the slave (steps S80, S81, and S82). Then, the communication device 11 receives the priority data and identification data from the slave (steps S83 and S84). The data synchronization group setting and management unit 16 sets the priority data, certificate data, and identification data received from the slave as own priority data, certificate data, and identification data (step S85), and resets the counter. Then, processing terminates (step S86). In this manner, a device that has been a master is specified as a slave device.

[0114] The timings of transmitting the priority data, counter value, certificate data, and identification data owned by the own device (steps S77, S80, S81, and S82) and those of receiving the priority data, certificate data, and identification data from a slave (steps S78, S83, and S84) may be in accordance with an arbitrary sequence.

[0115] Slave operations are generally similar to master operation. The processing steps are shown in FIG. 16.

[0116] When the device input/output device 13 receives a "master privilege transfer" request from the user, as described in FIG. 15, the data synchronization group setting and management unit 16 judges the current setting of the own device in the data synchronization group. In the case where this setting is not provided, processing terminates. When the device is a master, the aforementioned master operation is performed. In the case where it is determined as a slave, the device input/output device 13 waits for a request for executing master privilege transfer processing from the user. When the user executes master privilege transfer processing for the device input/output device 13 (step S91), the communication device 11 receives a master privilege transfer request from a master (step S92), and transmits the master privilege transfer request to the master (step S93). Then, at the step S94, the communication device 11 and the communication controller 12 check the number of devices that are executable for master privilege transfer processing. In the case where only one device other than own device is executable (YES at the step S94), it goes to the following steps.

[0117] First, the certificate data is exchanged with the master (steps S95 and S96). Then, the authentication unit 15 and the data synchronization group setting and management unit 16 authenticates the counterpart (step S97). After authentication, in the case where the communication counterpart is not a master of the data synchronization group to which the own device belong (NO at the step S97), processing terminates. In the case where the counterpart is the master (YES at the step S97), the priority data, counter value, and identification data owned by the master is received from the communication device 11 (steps S99, S100, and S101). Then, after the priority data and identification data owned by the own device have been transferred from the communication device 11 (steps S101 and S102), the data synchronization setting and management unit 16 sets the received master's identification data, certificate data, priority data, and counter value as own data (steps S103 and S104), and processing terminates.

[0118] The data synchronization group master can be set by each data set. In the case where a master device has a plurality of data sets, the master privilege transfer processing may be performed for only set data concerning a specific data set specified by the user. Of course, if a slave device specified as a transfer destination has a plurality of data sets, the master privileges concerning all the data sets owned by the master device can be transferred to one slave device in all.

(6) Priority Exchange

[0119] A "priority exchange" function exchanges the priorities owned by two slaves with each other. If a conflict occurs during data synchronization, such conflict is solved by using data on a member with its higher priority. The priority is assigned from the master in registration order. Thus, there can occur a case in which the user wants to cause the data on a slave registered later to precede the data on a slave registered previously. Al-

LINKPLAY EXHIBIT 1004
Page 1759 of 4822

though the priority may be selected during data synchronization member registration, the master or slave requires a display interface for selecting the priority. Here, devices are numbered in registration order for the purpose of simplification of processing so that priorities can be exchanged. At the time of "priority exchange" as well, as is the case with "data synchronization member registration" or "master privilege transfer", processing is performed only when only two devices are set.

[0120] Now, the communication processing step when the priorities are exchanged between slaves will be described with reference to FIG. 17.

[0121] First, slaves that want to exchange priorities transmit a priority exchange request to each other. When one slave receives such request, both of them transmit certificate data to each other. After it has been determined that the slaves belong to the same data synchronization group, both exchange priority data, and communication processing terminates.

[0122] In FIG. 18, there is shown the processing step of "priority exchange" in each slave device that exchanges the priority.

[0123] When the device input/output device 13 receives a "priority exchange" request from the user (step S101), the data synchronization group setting and management unit 16 judges the current setting of the own device in the data synchronization group. In the case where the setting is not provided or in the case of a master, processing terminates. In the case of a slave, the device input/output device 13 waits for a request for executing processing from the user. When the device input/output device 13 receives a request for executing processing (step S103), the communication device 11 transmits a request for processing "priority exchange" in a broadcasting manner, and waits for such request for processing "priority exchange" from the counterpart that exchange the priority. Then, when the communication device 11 receives a request for processing "priority exchange" from the counterpart (step S105), the communication controller 12 determines the number of devices that are currently executable for "priority exchange" (step S105) by using the data from the communication device 11 (step S106). In the case where only one device other than the own device is executable for "priority exchange" (YES at the step S106), the data synchronization group setting and management unit 16, communication controller 12, and communication device 11 receive and/or transmit certificate data from and/or to each other (steps S107 and S108). If three or more devices are executable within a communicable range (NO at the step S106), processing is terminated.

[0124] After certificate data has been transmitted and/or received, the authentication unit 15 and the data synchronization group setting and management unit 16 authenticate the communication counterpart based on the authentication data received at the communication device 11 (step S109). When it is authenticated that the counterpart device is a member of the same synchronization group (YES at the step S109), the own device transmits own priority data, and receives the counterpart's priority data (steps S110 and S111). Then, the data synchronization group setting and management unit 16 rewrites its own priority with the received priority, and terminates processing (step S112).

[0125] In the case where a device functions as a slave of the data synchronization group, the device having a plurality of data sets, the processing shown in FIG. 18 may be performed for a data set in which priority exchange is specified by the user.

(7) Method for Setting a Device

[0126] During each of the "data synchronization member registration", "priority exchange", and "master privilege transfer" processes, it is required to ensure safety of data communication to be performed together with setting. Therefore, prior to exchange actually set data, two devices targeted for setting are moved to be requested for executing processing. Then, after it is determined as to whether or not only two devices are requested for executing processing in a communicable range, it is required to determine whether or not such setting is continued. A series of these processes will be described in more detail.

[0127] In connection to these processes, each device has a set mode and an execute mode. The set mode serves to specify call functions including "data synchronization member registration", "priority change", and "master privilege transfer". The execute mode serves to start actual processing of a function called by the set mode, and make data communication with the set counterpart as required. Therefore, this execute mode means a request for executing processing.

[0128] In the case where a user attempts to perform "data synchronization member registration", "priority change", or "master privilege transfer", as shown in FIG. 19, the user moves two devices targeted for setting (devices 1 and 2) to the set mode. Then, each device determines which function is called, and determines whether or not the called function is valid in current setting of the own device. For example, in the case where the user attempts to perform "data synchronization member registration", the user determines that the called function is the "data synchronization member registration", and determines whether or not the own device is a mater, slave or none. When the called function is valid, it can go to the execute mode. User's operation is required for the current mode to go to the execute mode.

[0129] The user simultaneously moves two devices to the execute mode relevant to the same function (strictly, an error Δt of about 1 second is permitted). At a stage before communication with the counterpart is established in the execute mode, each device waits for data indicating the execute mode from another device. After each device has received the data, each device transmits it to a transmission channel. During this data trans-

LINKPLAY EXHIBIT 1004
Page 1760 of 4822

mission, a data broadcasting transmission may be repeated with predetermined time intervals, for example. When each device receives data indicating the execute mode of the counterpart, after it is determined that there exists no other device that enters this mode and the devices 1 and 2 communicate with each other, communication of set data is started. In communication of the set data, data is encrypted to ensure security by using an encryption system such as the aforementioned public key encryption system. In the case where there exists any other device entering the execute mode, communication regarding the set data is not performed, and processing is interrupted and terminated.

[0130] A method of configuring a data synchronization group and the functions of each device has been described hereinbefore. Now, a specific example of providing devices configuring a data synchronization group will be described here.

[0131] FIG. 20 shows an example when a system for this communication method is applied to a wrist watch, and FIG. 21 shows an example when this system is applied to a portable information processing terminal.

[0132] In the wrist watch shown in FIG. 20, sections associated with data synchronization are two buttons that exist at the left of a character board and four display sections at the upper part of a clock display section. The two buttons include an upper left button employed to instruct a request for executing each function for setting a data synchronization group and for instructing the start of data synchronization processing; and a lower left button for selecting each function for setting a data synchronization group. The display sections include a mode display section for displaying each function for setting a data synchronization group; a display section for displaying a character M or S that represents a master or slave; a display section indicating a numeric value of priority data; and a display section indicating whether or not a data synchronization group is currently set. For a description of use, procedures for executing two typical functions, "data synchronization member registration" and "data synchronization processing" will be discussed here.

[0133] First, the "data synchronization member registration" will be described here. The lower left button is used to call a "data synchronization member registration" mode. The mode display section displays that the "data synchronization member registration" function has been called. When this call is valid, the user holds down the upper left button, and executes a request for "data synchronization member registration." In this example, if this wrist watch is registered as a member by a given master, the similar procedures are performed in the master as well. When the aforementioned "data synchronization member registration" processing is successful while both devices execute the request, registration processing completes. In the case of "data synchronization processing", the upper left button is first pressed, whereby a function for starting synchronization processing is executed. If one or more devices belong to the same synchronization group that exists in a communicable range, and that can execute synchronization processing, data synchronization processing is actually started. The start of data synchronization processing may be executed by any one device calling the corresponding function or may be executed by all the devices calling the data synchronization processing start functions simultaneously, all of these devices being actually targeted for data synchronization processing at this time.

[0134] A data set that the wrist watch has is generally limited to time data (clock data). Thus, a device targeted for data synchronization processing with the wrist watch is a device that belongs to a data synchronization group concerning time data (clock data) in the devices that exist in a range communicable with the wrist watch.

[0135] In FIG. 21, there is shown an example of the screen for setting data synchronization of a portable information processing terminal. Shown here is an example of the screen for setting data synchronization concerning time data. This screen is composed of a "set menu" for calling each function provided at the upper left part; a display section that is set to display the current settings; and an execute button for starting data synchronization processing, the button being provided at the lower left. In the case where an attempt is made to perform data synchronization with a device that exists in the range communicable with the portable information processing terminal, belongs to the same data synchronization group capable of executing synchronization processing, a synchronization button at the lower right is pressed, whereby the data synchronization processing is called. In addition, the "data synchronization member registration" is selected from the set menu, whereby the data synchronization member registration screen is displayed. The execute button provided on this screen is held down, whereby the function for data synchronization member registration can be executed in the same manner as in the foregoing example of the wrist watch.

[0136] As has been described above, according to the communication method of the present embodiment, there is no need to provide set data concerning a communication counterpart for data synchronization. Basically, data synchronization processing can be performed safely and reliably by a certificate (certificate data) and a priority. Thus, data synchronization processing can be performed between arbitrary slaves without requiring a master. In particular, the present invention is preferable to application to a device that makes wireless communication outside. Moreover, synchronization processing of a data set that belongs to the same data synchronization group is automatically executed with another device that exists in a communicable range by instructing the start of synchronization processing. Therefore, in a device having only a limited user interface such as wrist watch or portable cellular phone, the device being difficult in setting concerning a communi-

LINKPLAY EXHIBIT 1004
Page 1761 of 4822

cation counterpart as well, data synchronization can be easily performed by data synchronization group such that a group member is dynamically changed.

[0137] For example, although each embodiment has been described singly, a plurality of embodiments may be combined with each other as required. The communication method according to the present embodiment can be achieved by a computer program. Thus, the computer program is recorded in a computer readable recording medium such as CD-ROM, and then, is introduced into the computer or the computer program is recorded into the recording medium such as ROM, and then, is merely incorporated into an electronic device, making it possible to achieve an effect similar to the present embodiment using a general electronic device.

[0138] As has been described above, according to the present invention, it is possible to provide a communication method and electronic device that enables data synchronization between arbitrary devices without requiring a server. It is possible to perform data synchronization with a device that does not have a limited input device and a display device. It is possible to make communication for data synchronization safely between arbitrary devices in a wired or wireless network such that a network configuration dynamically changes. In exchanging set data between devices, only specific two devices perform setting operations simultaneously after the presence of two set devices has been guaranteed, whereby setting can be done safely.

Claims

1. A data synchronization method characterized by comprising:

   transmitting from a master device to a slave device a certificate indicating that the slave device belongs to a data synchronization group to which the master device belongs and a priority to be used for solving conflict of data, thereby registering the slave device as a member of the data synchronization group to which the master device belongs; and
   determining whether or not a first slave device and a second slave device which is capable of communicating with the first slave device belong to the same data synchronization group by using the certificate, and performing data synchronization between the first slave device and the second slave device based on the priority if the first slave device and the second slave device belong the same data synchronization group.

2. The method according to claim 1, characterized in that said registering is performed after it is confirmed that there is no other device than said master device and said slave device, which other device is capable of communicating with said master device and is set in a registration mode.

3. The method according to claim 1, characterized in that said data synchronization is performed between one master device and one or more slave devices which belong to the same synchronization group or between at least two slave devices which belong to the same synchronization group.

4. The method according to claim 1, characterized in that said master device and slave device store plural types of data and belong to plural data synchronization groups defined for each of said plural types of data.

5. The method according to claim 4, characterized in that said registering is performed by transmitting said certificate and said priority from the master device belonging to a given synchronization group defined for a given type of data to the slave device belonging to the given synchronization group, said certificate and said priority being set depending on the given type of data.

6. The method according to claim 1, characterized by further comprising transmitting data required to operate as the master device from said master device to a slave device which is targeted to transfer a master privilege, thereby transferring the master privilege to the slave device.

7. The method according to claim 6, characterized in that said master privilege transferring is performed after it is confirmed that there is no other slave device than said master device and said slave device, which other slave device is capable of communicating with said master device and is set in a master privilege transfer mode.

8. The method according to claim 1, characterized by further comprising exchanging the priority between slave devices that belong to the same data synchronization group.

9. The method according to claim 8, characterized in that said priority exchanging is performed after it is confirmed that there is no other slave device than said master device and said slave device, which other slave device is capable of communicating with said master device and is set in a priority exchanging mode.

10. The method according to claim 1, characterized by further comprising releasing said master device or said slave device from the data synchronization group, characterized in that, if said master device

LINKPLAY EXHIBIT 1004
Page 1762 of 4822

is released, said master device transmits a releasing instruction to a slave device which is capable of communicating with said master device to release said slave device as well as said master device.

11. A computer readable recording medium storing a computer program for data synchronization between plural devices, the program characterized by comprising:

first program code means causing a master device transmit to a slave device a certificate indicating that the slave device belongs to a data synchronization group to which the master device belongs and a priority to be used for solving conflict of data, thereby registering the slave device as a member of the data synchronization group to which the master device belongs; and

first program code means causing a first slave device and a second slave device to determine whether or not the first slave device and the second slave device which is capable of communicating with the first slave device belong to the same data synchronization group by using the certificate, and perform data synchronization between the first slave device and the second slave device based on the priority if the first slave device and the second slave device belong the same data synchronization group.

12. The computer readable recording medium according to claim 11, characterized in that said registering is performed after it is confirmed that there is no other device than said master device and said slave device, which other device is capable of communicating with said master device and is set in a registration mode.

13. The computer readable recording medium according to claim 11, characterized in that said data synchronization is performed between one master device and one or more slave devices which belong to the same synchronization group or between at least two slave devices which belong to the same synchronization group.

14. The computer readable recording medium according to claim 11, characterized in that said data synchronization group is formed of a type of data.

15. The computer readable recording medium according to claim 14, characterized in that said registering is performed by transmitting said certificate and said priority from the master device set by a user for a given type of data to the slave device set by the user for the given type of data, said certificate and said priority being set depend on the type of data.

16. The computer readable recording medium according to claim 11, characterized in that said program further comprises third program code means causing said master device to transmit data required to operate as the master device to a slave device which is targeted to transfer a master privilege, thereby transferring the master privilege to the slave device.

17. The computer readable recording medium according to claim 16, characterized in that said master privilege transferring is performed after it is confirmed that there is no other slave device than said master device and said slave device, which other slave device is capable of communicating with said master device and is set in a master privilege transfer mode.

18. The computer readable recording medium according to claim 11, characterized in that said program further comprises fourth program code means for causing slave devices that belong to the same data synchronization group to exchanging the priority.

19. The computer readable recording medium according to claim 18, characterized in that said priority exchanging is performed after it is confirmed that there is no other slave device than said master device and said slave device, which other slave device is capable of communicating with said master device and is set in a priority exchanging mode.

20. The computer readable recording medium according to claim 1, characterized in that said program further comprises fifth program code means for causing said master device or said slave device to be released from the data synchronization group, and if said master device is released, causing said master device to transmit a releasing instruction to a slave device which is capable of communicating with said master device to release said slave device as well as said master device.

21. An electronic device characterized by comprising:

a master unit, when an own device is set by a user as a master, configured to transmit to a slave device a certificate indicating that the slave device belongs to a data synchronization group to which the master device belongs and a priority to be used for solving conflict of data, thereby registering the slave device as a member of the data synchronization group to which the master device belongs;

a slave unit, when the own device is set by the user as the slave, configured to receive from

LINKPLAY EXHIBIT 1004
Page 1763 of 4822

the master unit the certificate and the priority; and

a data synchronization unit configured to determine whether or not the own device and another device which is capable of communicating with the own device belong to the same data synchronization group by using the certificate, and perform data synchronization between the own device and the other device based on the priority if the own device and the other device belong the same data synchronization group.

LINKPLAY EXHIBIT 1004
Page 1764 of 4822

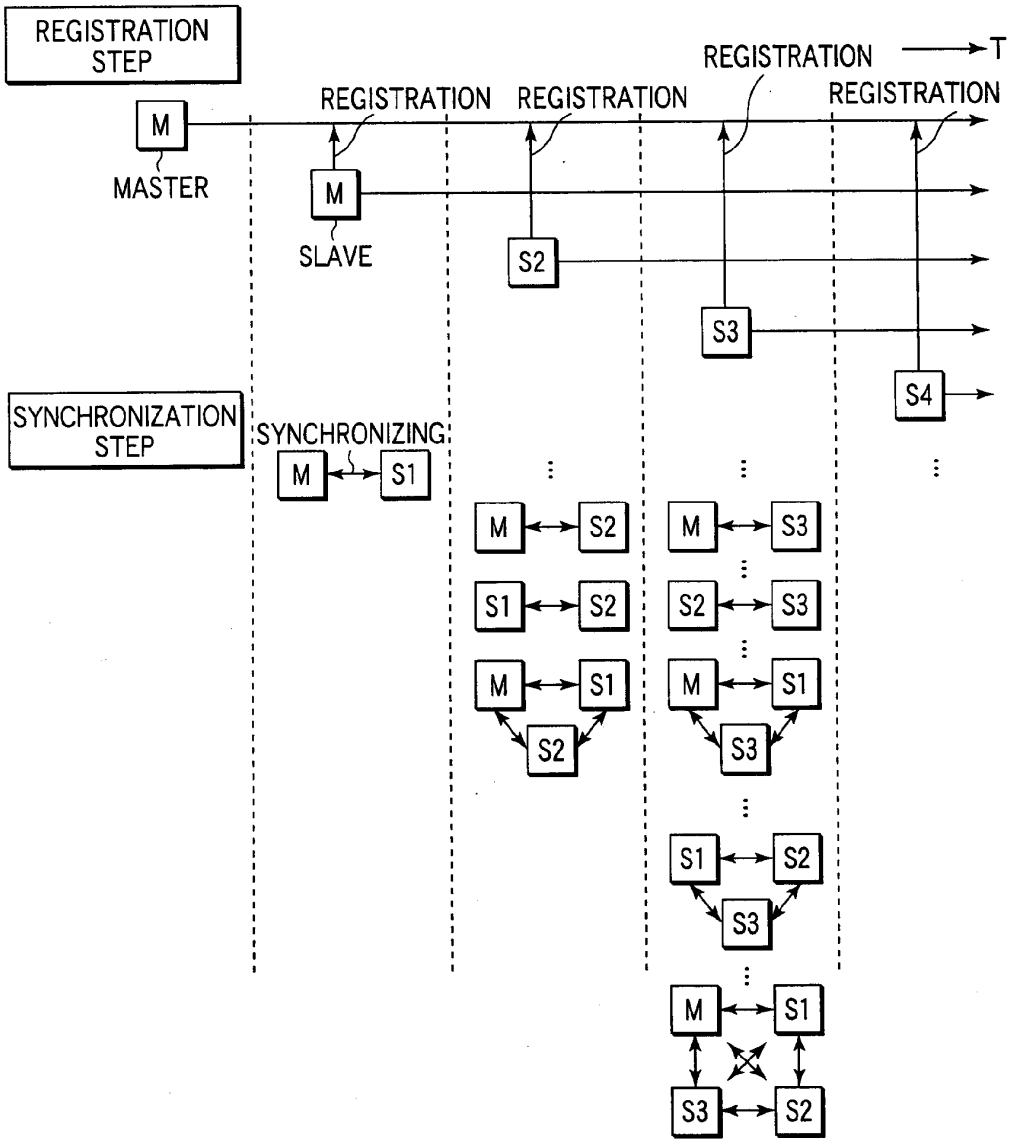


FIG. 1

LINKPLAY EXHIBIT 1004
Page 1765 of 4822

TYPE OF DATA SET	MASTER/ SLAVE	PRIORITY DATA	CERTIFICATE DATA	IDENTIFICATION DATA	COUNTER VALUE
ADDRESS DIRECTORY	MASTER	1	01101110	10101010	12
TIME	SLAVE	5	00111010	01010101	———
SCHEDULE	———	———	———	———	———
TASK LIST	———	———	———	———	———
E-MAIL	SLAVE	8	11010011	11110100	———
⋮	⋮	⋮	⋮	⋮	⋮

FIG. 2

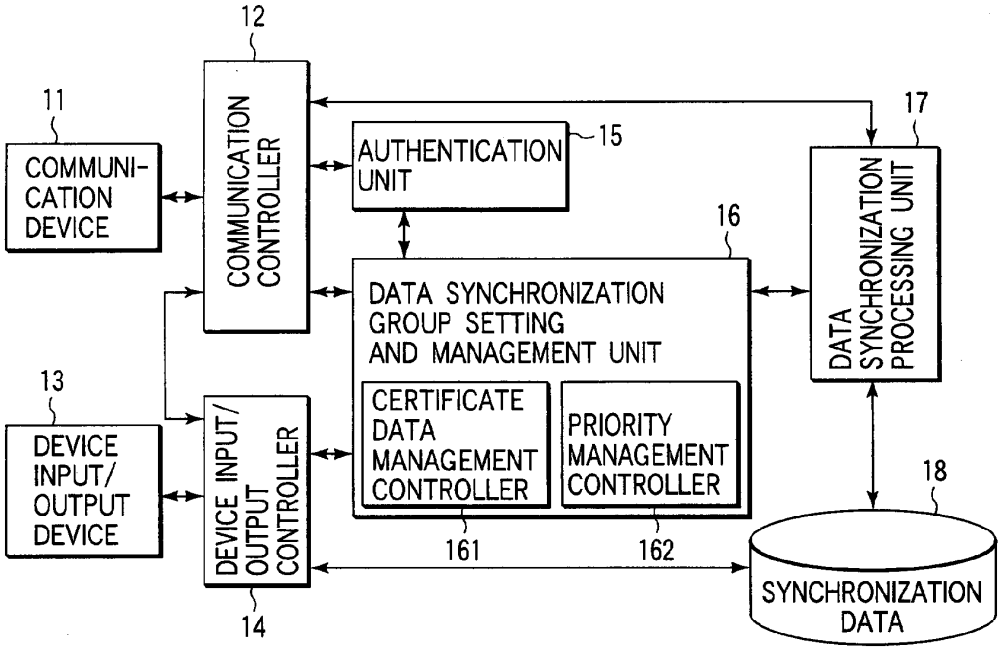


FIG. 3

LINKPLAY EXHIBIT 1004
Page 1766 of 4822

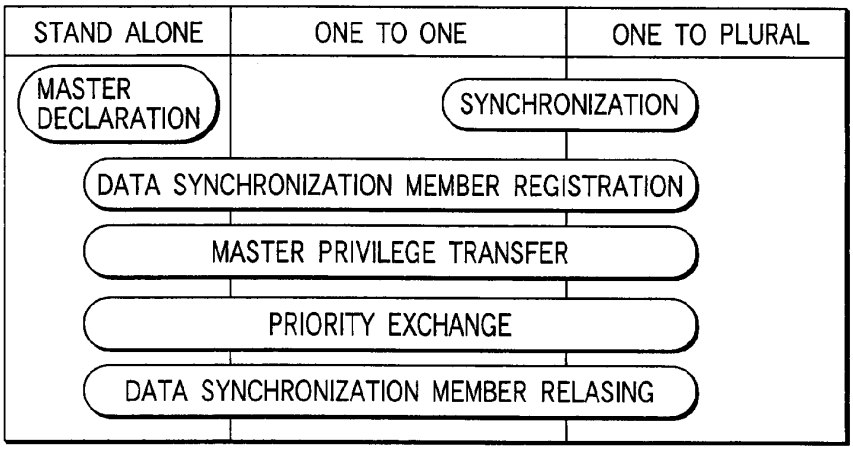


FIG. 4

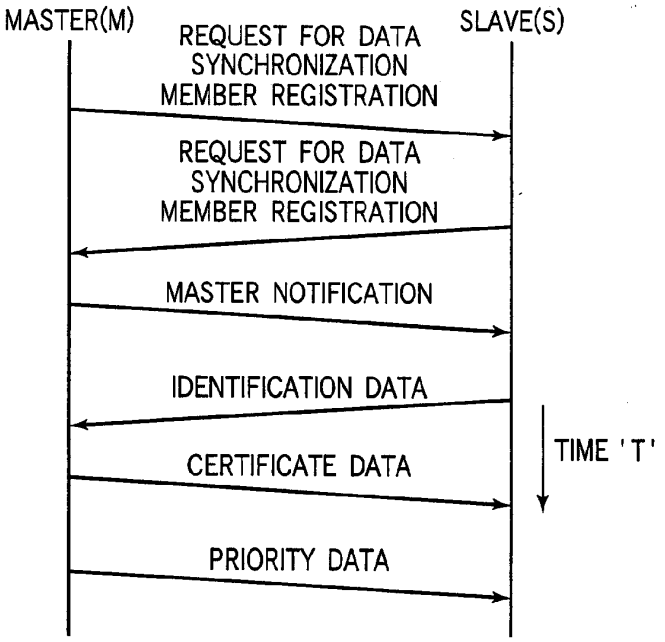


FIG. 5

LINKPLAY EXHIBIT 1004
Page 1767 of 4822

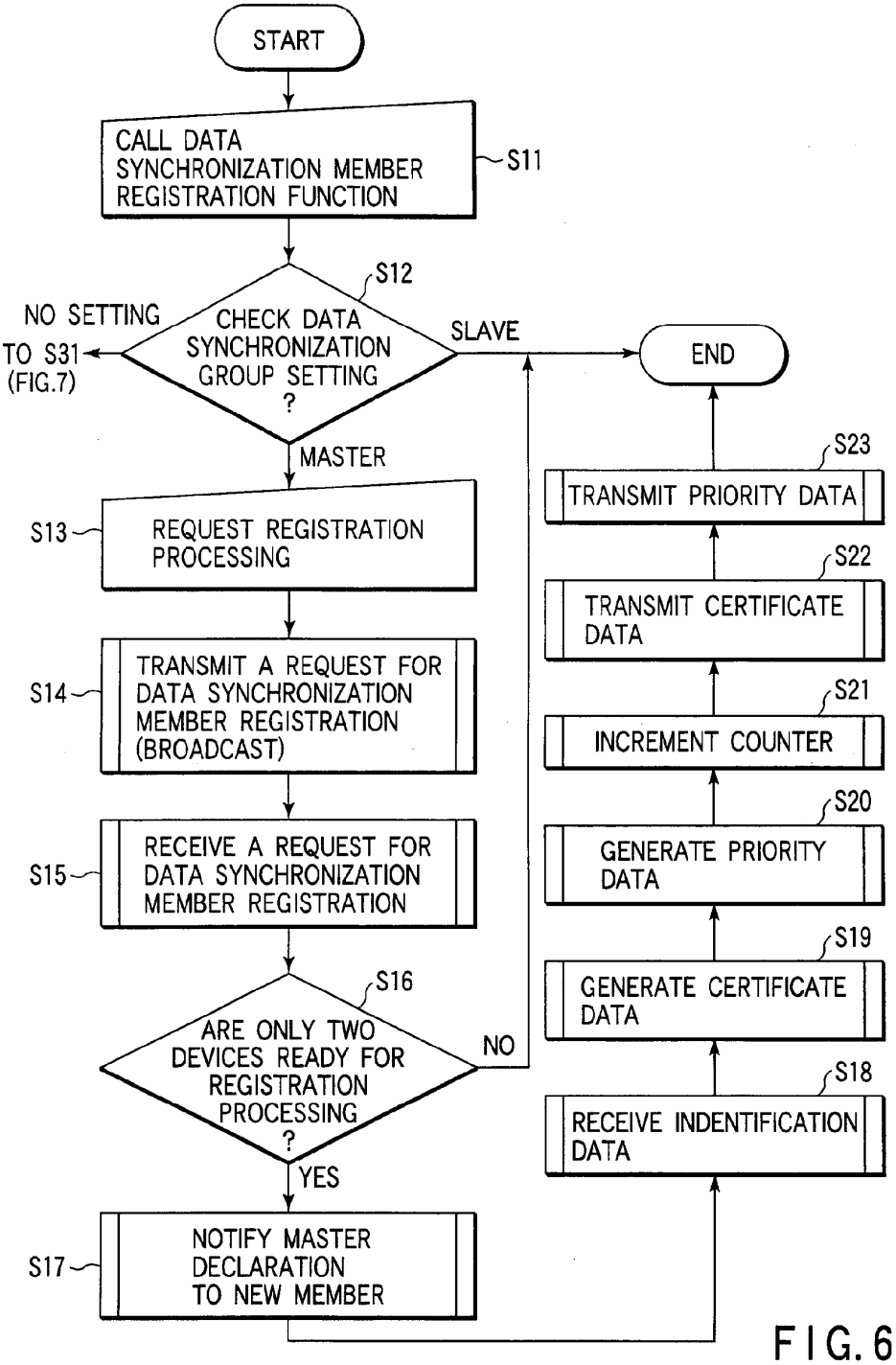


FIG. 6

LINKPLAY EXHIBIT 1004
Page 1768 of 4822

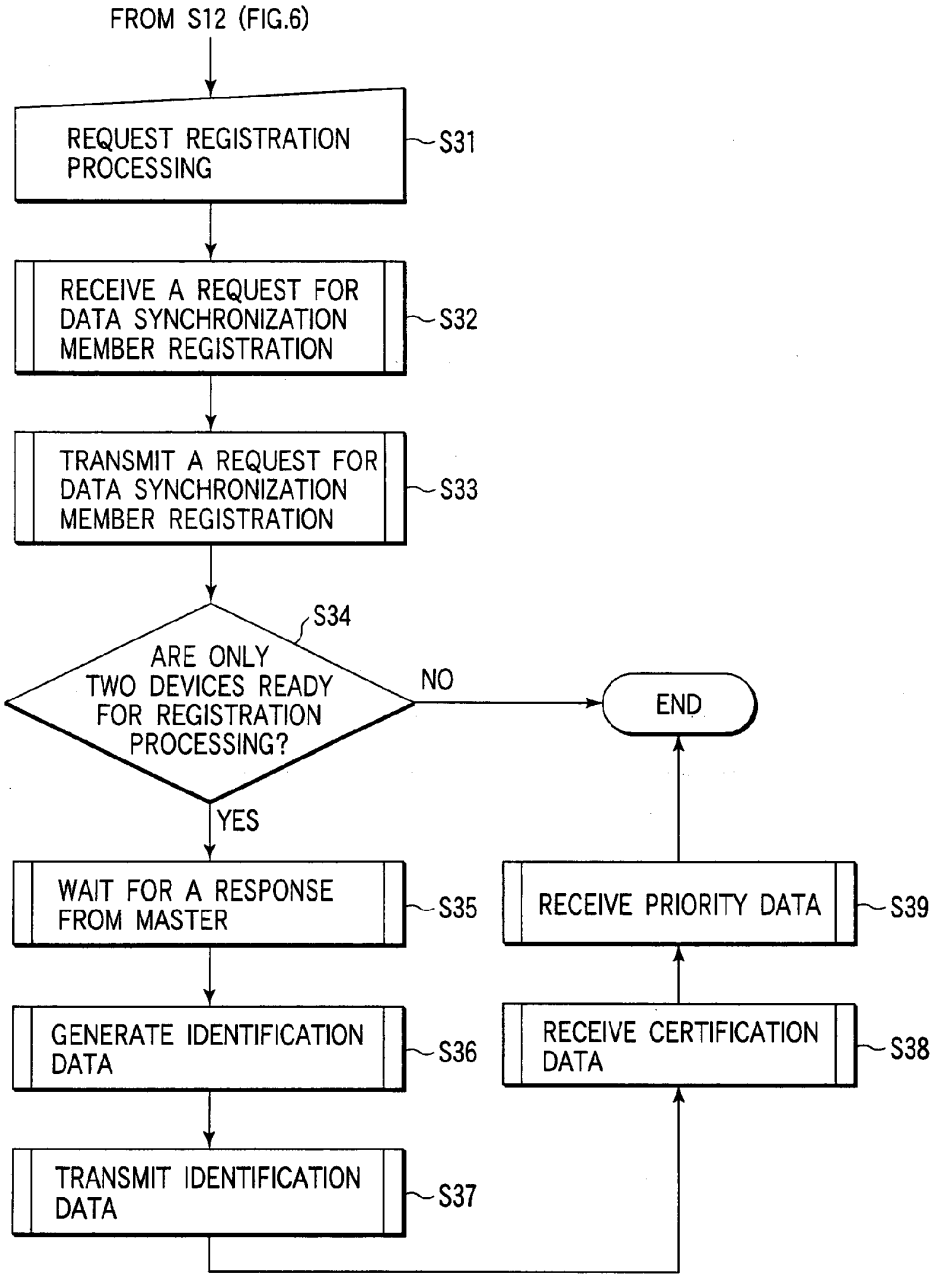


FIG. 7

LINKPLAY EXHIBIT 1004
Page 1769 of 4822

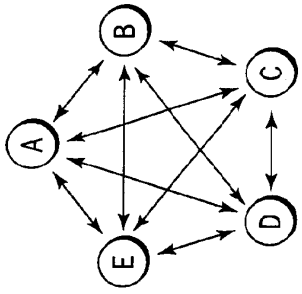


FIG. 8

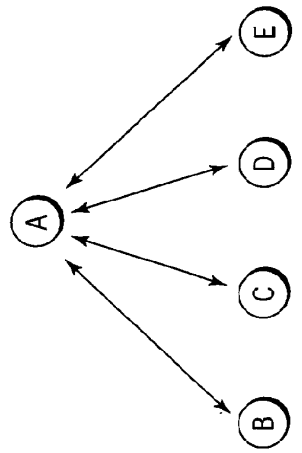


FIG. 9

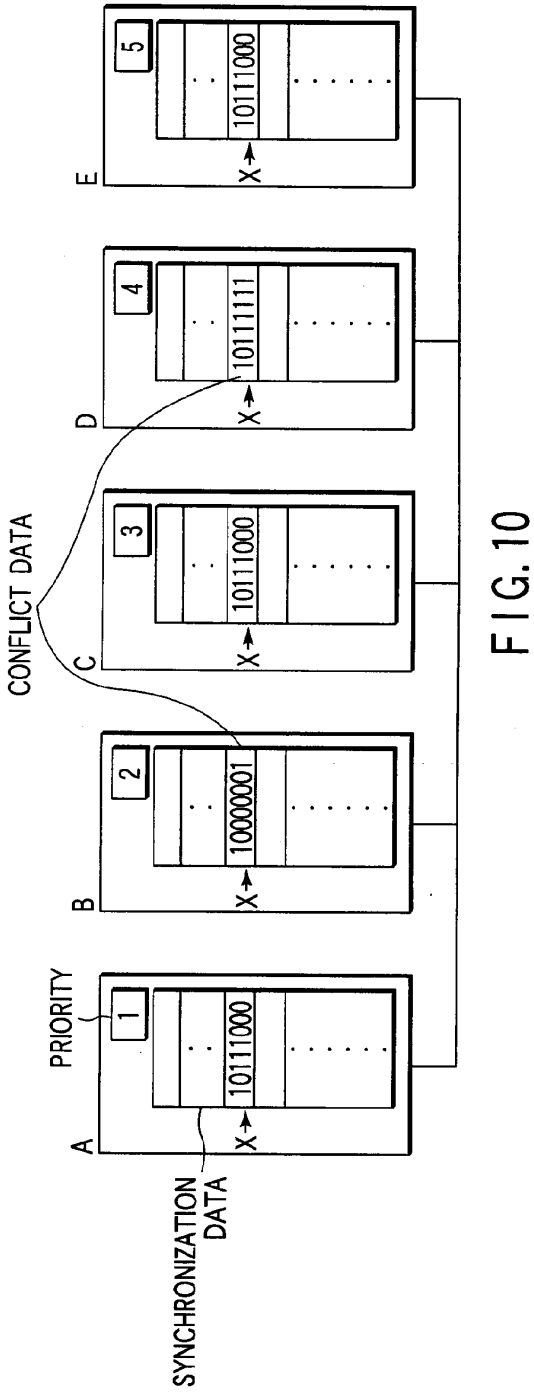


FIG. 10

LINKPLAY EXHIBIT 1004
Page 1770 of 4822

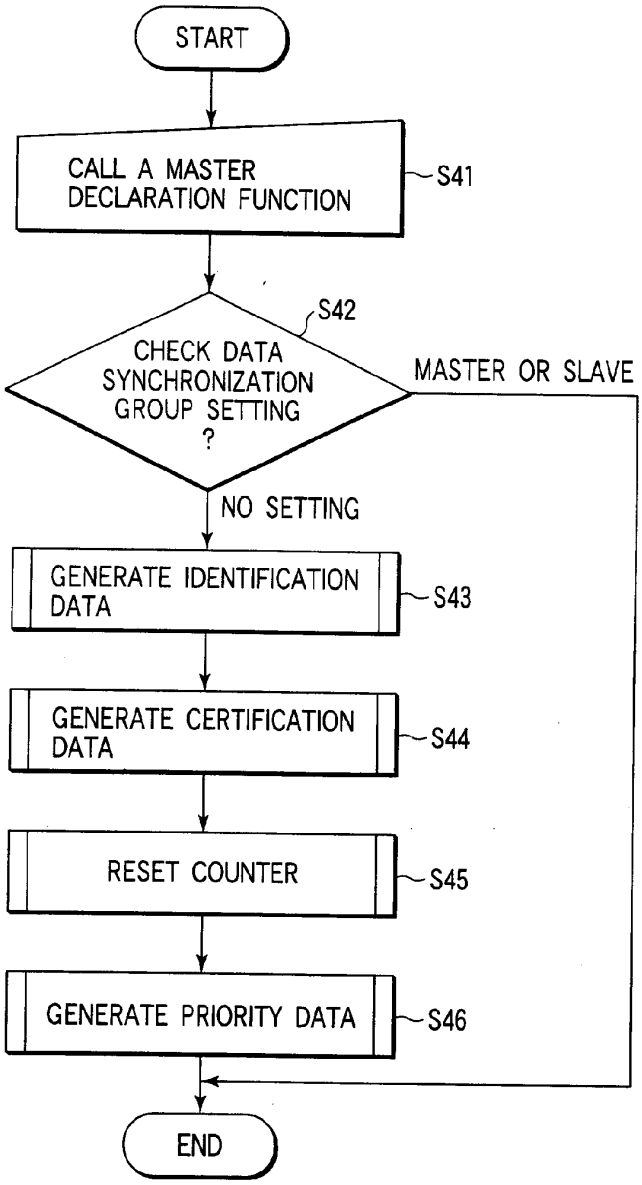


FIG. 11

LINKPLAY EXHIBIT 1004
Page 1771 of 4822

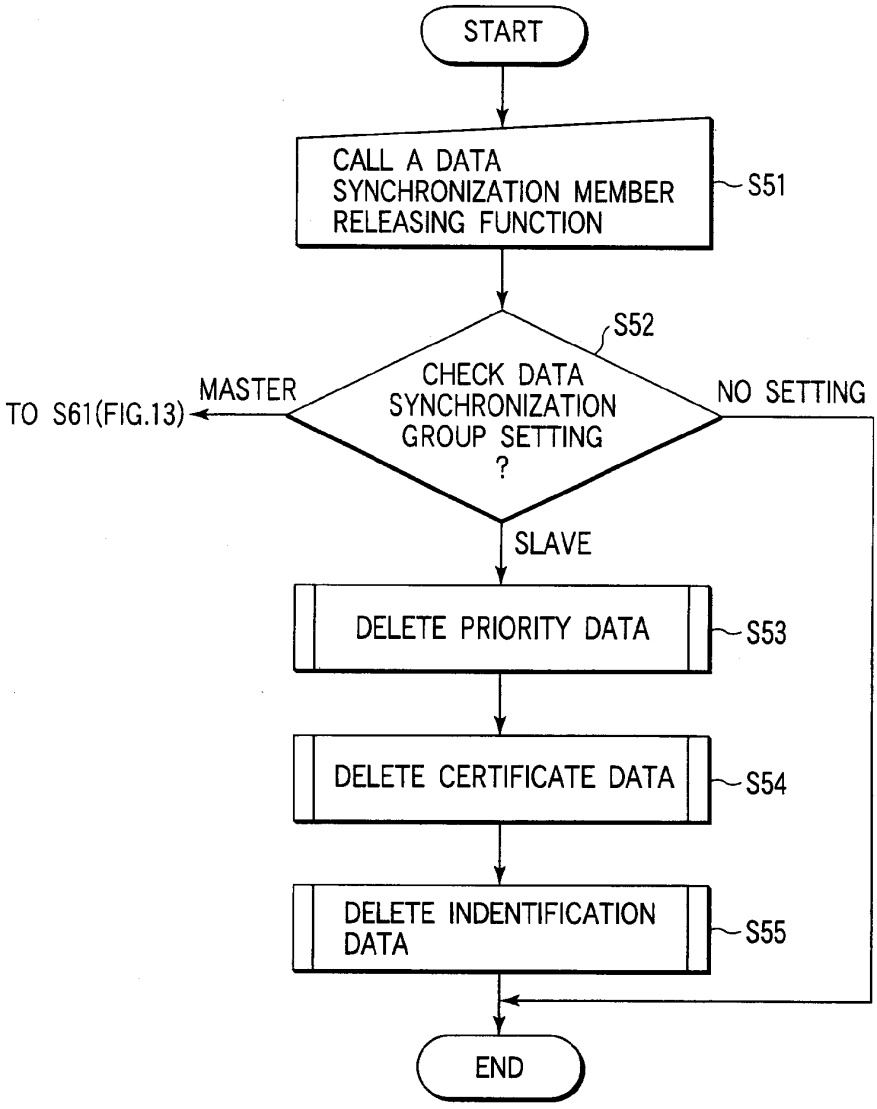


FIG. 12

LINKPLAY EXHIBIT 1004
Page 1772 of 4822

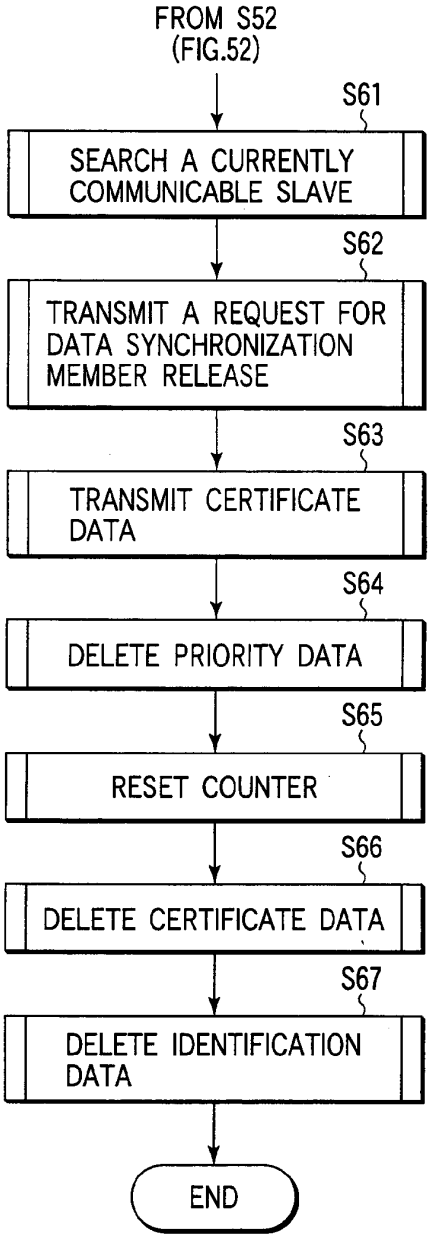


FIG. 13

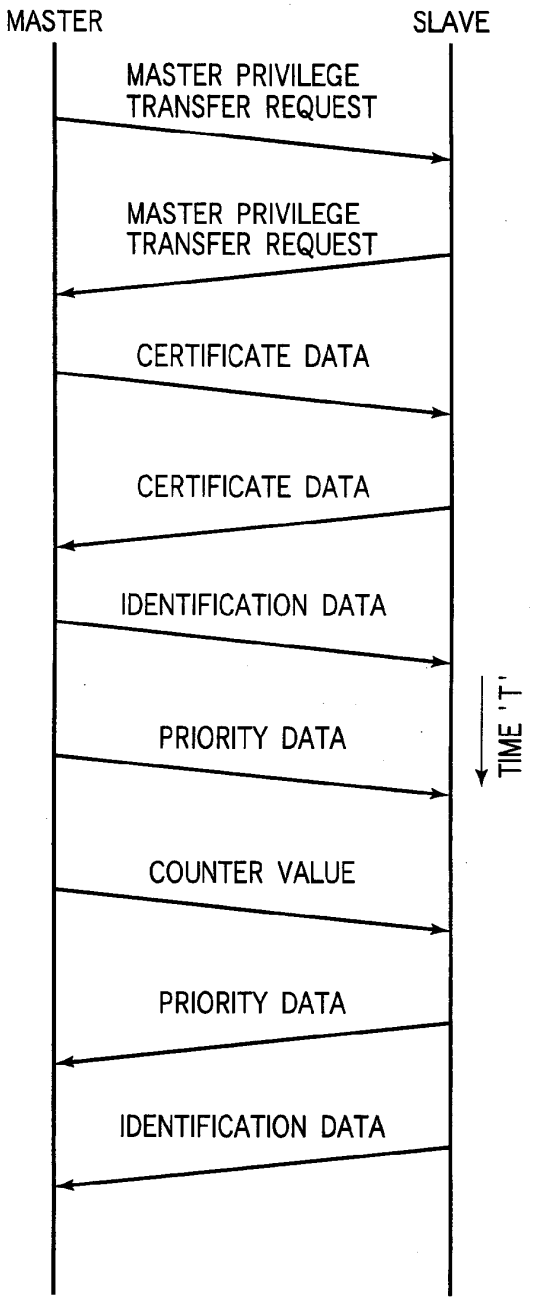


FIG. 14

LINKPLAY EXHIBIT 1004
Page 1773 of 4822

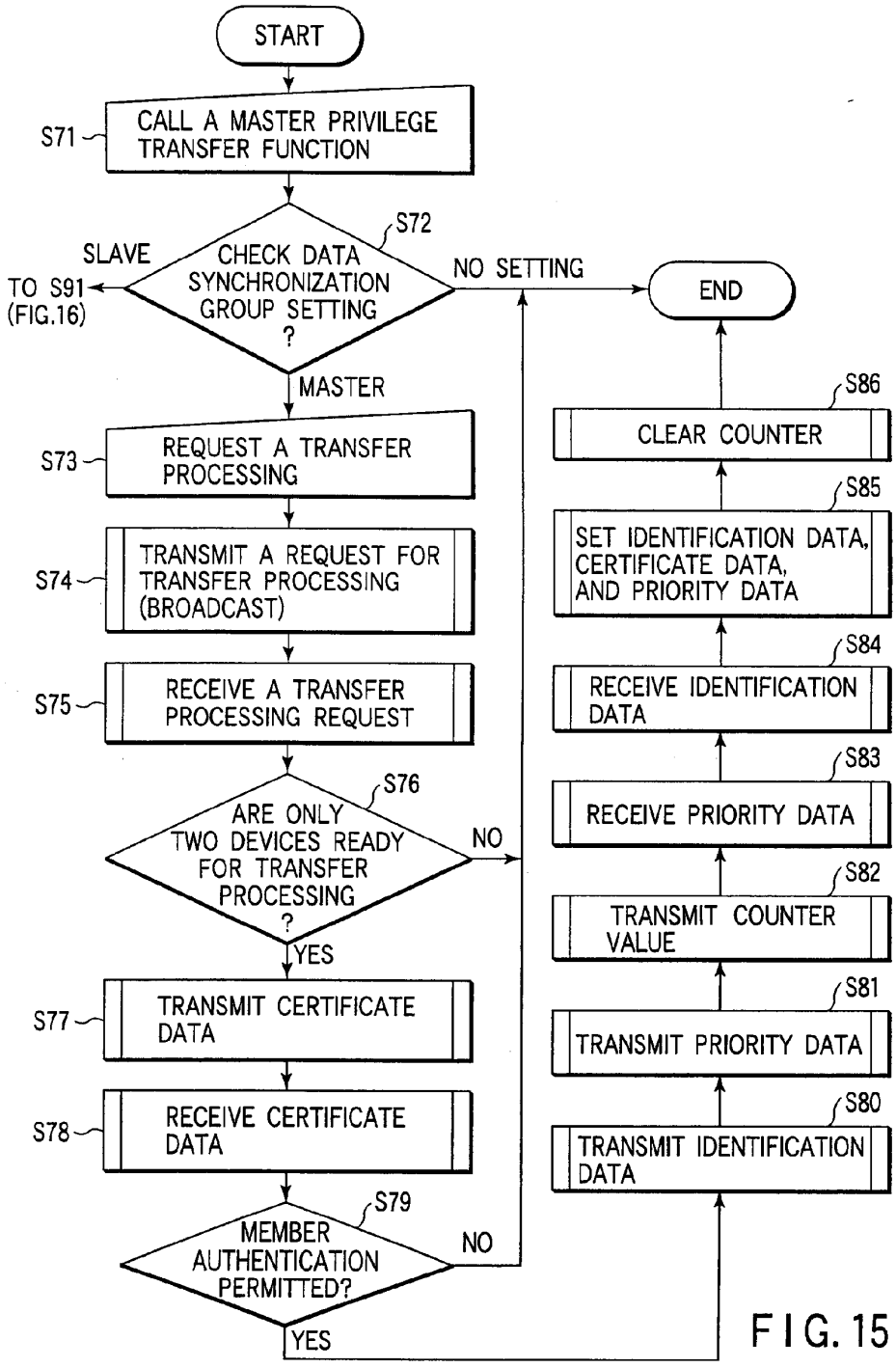


FIG. 15

LINKPLAY EXHIBIT 1004
Page 1774 of 4822

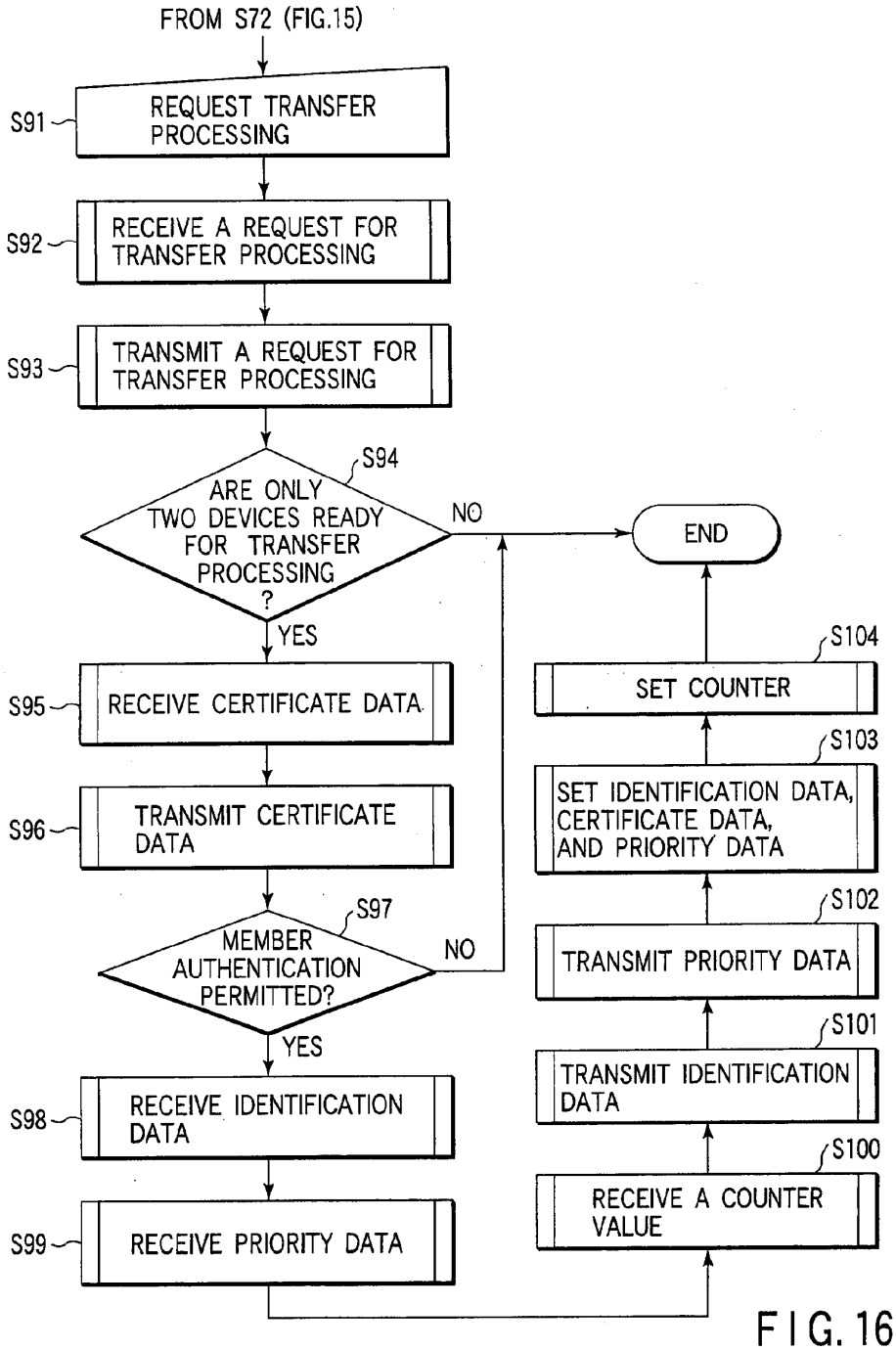


FIG. 16

LINKPLAY EXHIBIT 1004
Page 1775 of 4822

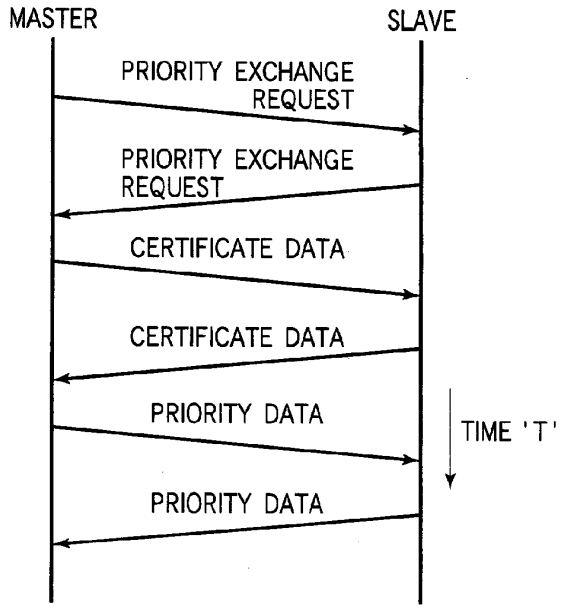


FIG. 17

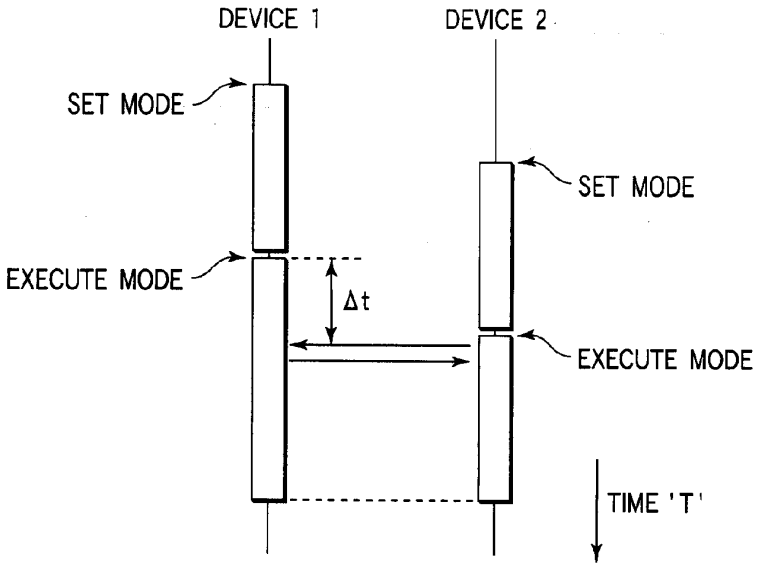


FIG. 19

LINKPLAY EXHIBIT 1004
Page 1776 of 4822

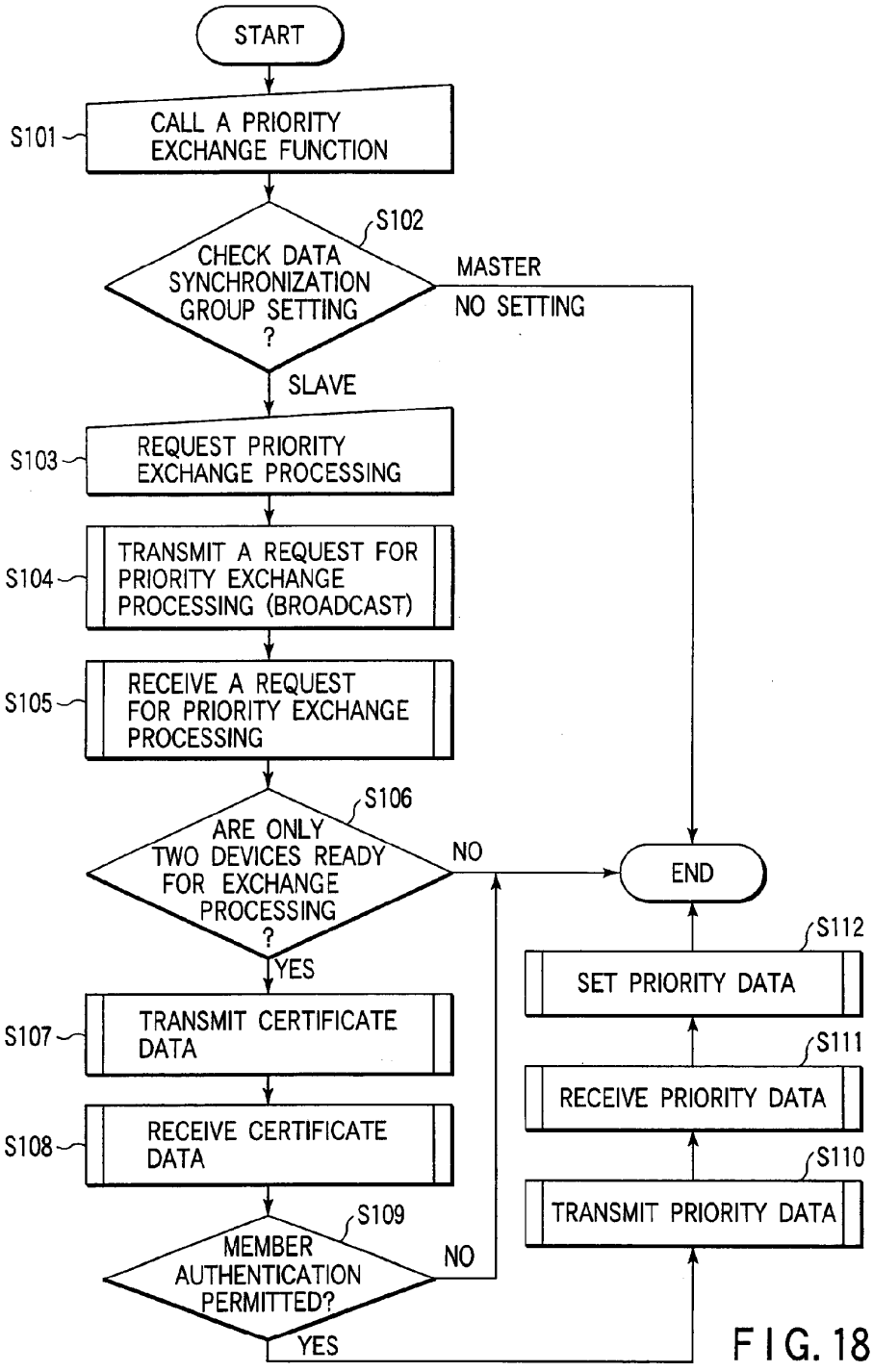


FIG. 18

LINKPLAY EXHIBIT 1004
Page 1777 of 4822

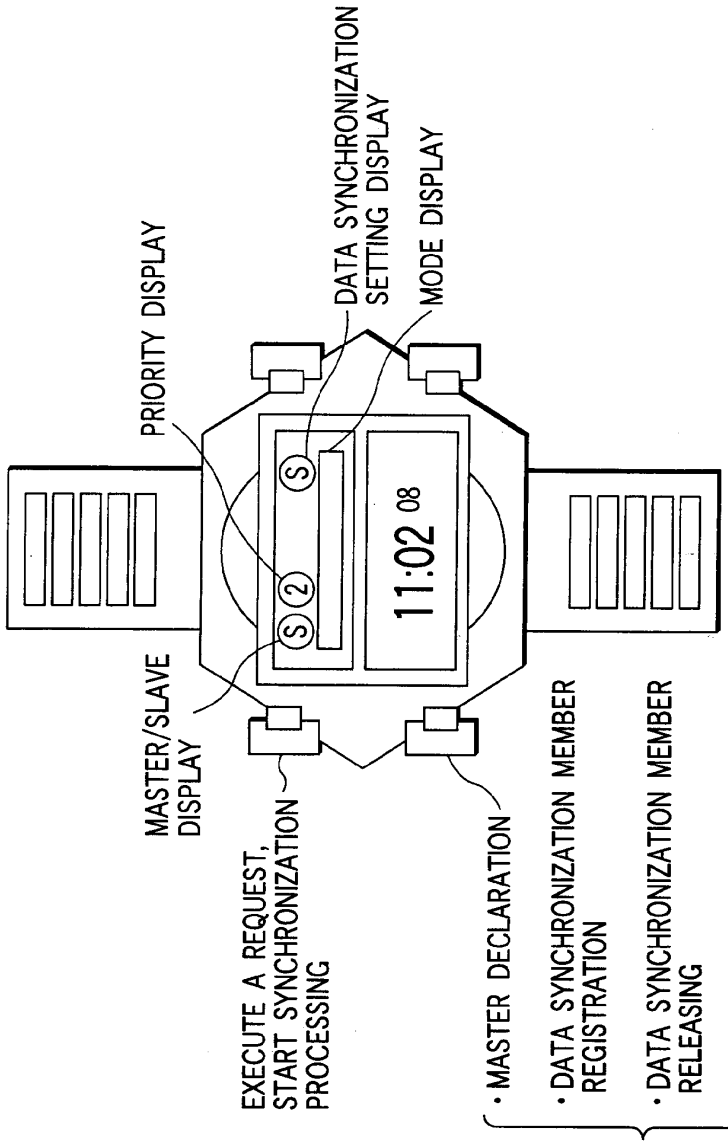


FIG. 20

LINKPLAY EXHIBIT 1004
Page 1778 of 4822

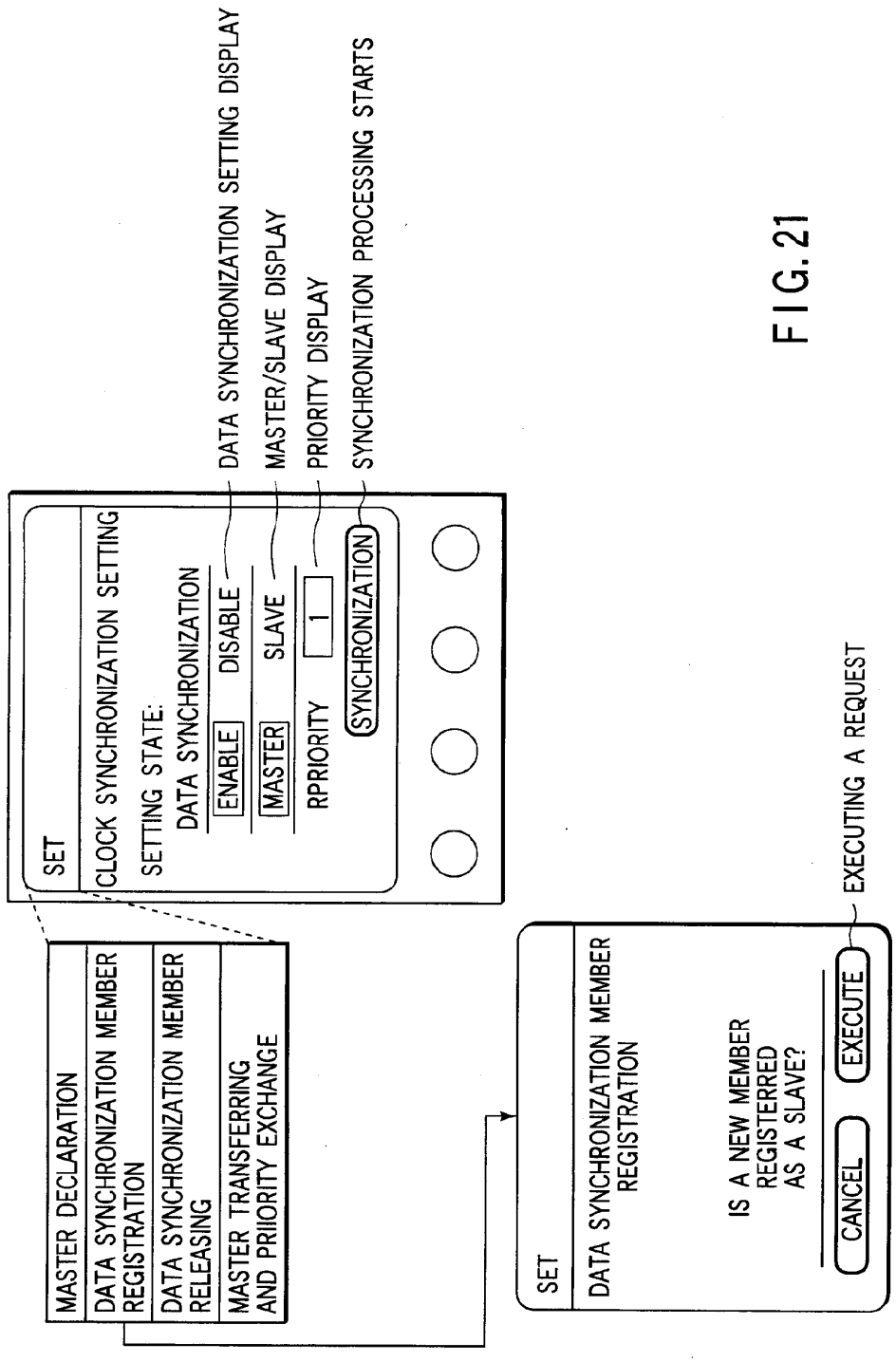


FIG. 21

LINKPLAY EXHIBIT 1004
Page 1779 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 122 931 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
08.08.2001 Bulletin 2001/32

(51) Int Cl.[7]: **H04L 29/06**

(21) Application number: **01102060.9**

(22) Date of filing: **30.01.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **07.02.2000 US 499701**

(71) Applicant: **SHARP KABUSHIKI KAISHA**
Osaka 545-8522 (JP)

(72) Inventor: **Ishii, Atsushi**
Vancouver, Washington 98665 (US)

(74) Representative: **MÜLLER & HOFFMANN**
Patentanwälte
Innere Wiener Strasse 17
81667 München (DE)

(54) Real-time media content synchronization and transmission in packet network apparatus and method

(57) A system for transmitting and synchronizing real-time multimedia content includes a multimedia server (62) for generating a multimedia packet (64) having multimedia audio/visual information therein in a play sequence, and for transmitting the multimedia packet; a packet-based communication network (66) connected to the multimedia server (62) for receiving the multimedia packets therefrom; the network having multiple routers (68, 70) therein to route the multimedia packets to plural destinations; means for inserting total delay information (TDI) into the multimedia packets, wherein the TDI includes total end-to-end delay (TED) and cumulative network delay (CND); and plural multimedia receivers (80, 82, 84) for receiving the multimedia packets having TDI therein, wherein each multimedia receiver (80, 82, 84) includes a buffer for storing received multimedia packets, a sequencing means for retrieving stored multimedia packets from the buffer and sequencing the multimedia packets in the play sequence, and a playing mechanism for playing the retrieved, sequenced multimedia packets at time TED after the multimedia packet has been transmitted by the multimedia server (62). A method of transmitting and synchronizing real-time multimedia content includes loading multimedia audio/visual into plural multimedia packets (64); inserting total delay information (TDI) into at least one of the multimedia packets (64), wherein TDI includes total end-to-end delay (TED) and cumulative network delay (CND); transmitting the multimedia packets to plural multimedia receivers (80, 82, 84) over a packet-based network (66) in a play sequence; buffering the multimedia packets in each multimedia receiver (80, 82, 84); recalling the multimedia packets in play sequence; and playing the multimedia packets (64) at each receiver (80, 82, 84) at time TED.

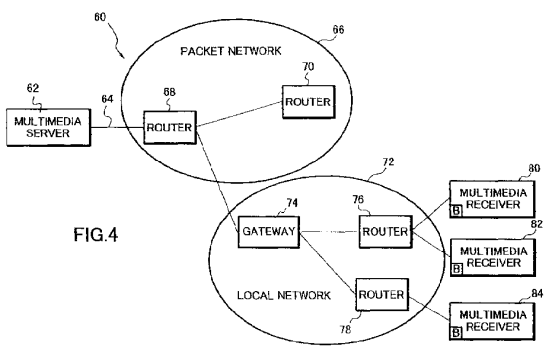


FIG.4

EP 1 122 931 A2

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

[0001] This invention relates to multimedia networks, and specifically to the synchronization of multicast distributed media content when played by multiple multimedia devices, e.g., monitors and audio systems.

Background of the Invention

[0002] In some situations, such as at conventions and in exhibition halls, it is required to play the same video contents synchronized at multiple video monitors. Generally, it is also required to synchronize audio and video between separately located monitors and speakers. Current real-time media transmission technology allows transferring an audio/video stream over non-guaranteed quality of service (QoS) packet networks with reasonable quality and at a lower cost than provided by higher quality networks. Thus, there will be a demand for playing multimedia contents, delivered to a plurality of receivers over packet network, in synchronized manner.

[0003] Fig. 1 depicts a prior art multimedia content delivery system, generally at 10, which uses a packet network. A multimedia server 12 generates a stream of multimedia packets 14 that contain multimedia audio/video contents. Normally, server, or source, 12 generates multimedia packets at a constant rate (packet rate); *e.g.*, one packet every 20 ms. Packet size depends on the multimedia codec, which may result in fixed size or variable sized packets. The generated multimedia packets are then transferred through a packet network 16, having plural routers, or nodes, 18, 20, 22, 24, 26, 28 and 30 therein, and are delivered to a plurality of multimedia receivers, 32, 34, 36 and 38. The mechanism for delivery to multiple devices is well known as "multicast." All the receivers that wish to receive multimedia contents from the server join a multicast group. The routers in packet network 16 are aware of the multicast group, and transfer received multimedia packet to multiple routes. Each receiver is equipped with a speaker and/or a video display, and each plays the contents in the received multimedia packet in real-time. The multimedia receivers may buffer received multimedia packets for some duration in order to absorb the variance in delay caused by the network, and to playback multimedia packets in right order.

[0004] In conventional packet network, such as an IP network, packets are queued in each router. How long a packet stays in a queue depends on the amount of traffic that a particular router is handling. Hence, each multimedia packet delivered to each multimedia receiver has an unpredictable delay. As a result, the audio/video contents may be played at multiple locations without synchronization.

[0005] As previously noted, real-time multimedia transmission technology currently allows transferring an audio/video stream over non-guaranteed quality of service (QoS) packet networks with reasonable quality. As mobile Internet access increases, with its data transfer over bandlimited communication channels, such as wireless data networks, the technology to transmit real-time multimedia packets requiring minimum bandwidth becomes essential and of increases importance.

[0006] A packet-based real-time media transmission system is illustrated in Fig. 2, generally at 40. A multimedia source, or server, 42, periodically transmits multimedia packets 44 having audio/video multimedia contents to a packet network 46. In packet network 46, each multimedia packet is routed by a plurality of network nodes, or routers, 48, 50, 52 and 54, and delivered to a receiver 56. Each multimedia packet includes timing information that is used to set timing for playback at a receiver 56. The multiple routers and receivers of Fig. 1 are not shown in Fig. 2 for the sake of clarity.

[0007] Packet network 46 causes a variable length delay in the delivery of each multimedia packet. In addition, network 46 does not guarantee the delivery of the multimedia packets in the same order as transmitted. Thus, a receiver is normally equipped with a buffer to handle delay variances and/or disorders in multimedia packet receipt. However, due to the real-time property of the contents, the receiver will discard a received multimedia packet if the delay in its arrival exceeds a certain value, or if the multimedia packet does not arrive by the time it is to be played.

[0008] Fig. 3 shows an example of packets received and played back by the receiver, where each packet contains the timing information. Line I depicts seven multimedia packets transmitted in sequential order by server 42. Line II depicts the order of receipt of the multimedia packets at a receiver. Line III depicts a play order by the receiver. The figure depicts that the multimedia packet with timing information 3 is discarded due to an excessive delay. Basically, the routers in the prior art packet network are not aware of delays in packets transmission and receipt, nor are they aware of disordering of packet transmission. Thus, each router re-directs every received real-time multimedia packet to the next router or the receiver, even if the multimedia packet may not have any chance of being played at the receiver. If a physical link between some two routers, or the physical link between the receiver and the adjacent router has limited bandwidth, the transmission of the multimedia packets which will be discarded at the receiver possibly wastes substantial portion of valuable bandwidth on the link. Delay and disordering is illustrated by the variable routing of multimedia packets from router 48 through router 50, router 52 and router 54, or directly from router 48 to router 54.

[0009] U.S. Patent No. 4,506,358, for *Time stamping for a packet switching system*, to Montgomery, granted March

LINKPLAY EXHIBIT 1004
Page 1781 of 4822

19, 1985, describes a method of measuring the network delay in a packet network by including an information field in each packet that indicates the cumulative network delay, time stamping each packet's arrival and re-transmission in each network node, and updating the information field in each network node for each packet by adding the delay occurring in each node, *i.e.*, re-transmission time - arrival time.

[0010] U.S. Patent No. 4,530,091, for *Synchronization of real-time clocks in a packet switching system*, to Crockett, granted July 16, 1985, describes a means to synchronize a remote real-time clock to a standard real-time clock via an X.25 (CCITT) network, but does not teach or suggest how to synchronize a multimedia stream, and does not consider network delay caused in a packet network.

[0011] U.S. Patent No. 4,918,687 for *Digital packet switching networks*, to Bustini *et al.*, granted April 17, 1990, describes a method to avoid consecutive packet loss by clipping voice packets whose contents indicate whether the packet is eligible for clipping, if required, in a node queue within the network.

[0012] U.S. Patent No. 5,432,790 for *Method for allocating internodal link bandwidth in a packet oriented communication network to guarantee delay quality-of-service*, to Hluchyj *et al.*, granted July 11, 1995, describes a method to allocate bandwidth on internodal links such that worst case maximum delays and worst case average delays are guaranteed while providing statistical gains in maintaining predetermined end-to-end delay QoS.

[0013] U.S. Patent No. 5,444,709 for *Protocol for transporting real time data*, to Riddle, granted August 22, 1995, describes a method to maintain the isochronous nature of the real-time data transmission by discarding data that is not timely received.

[0014] U.S. Patent No. 5,450,394 for *Delay monitoring of telecommunication networks*, to Gruber *et al.*, granted September 12, 1995, describes a means to measure a round-trip delay of network transit time, however, the delay which must be known for multimedia transmission and synchronization is a one-way network delay.

[0015] U.S. Patent No. 5,557,724 for *User interface, method, and apparatus selecting and playing channels having video, audio, and/or text streams*, to Sampat *et al.*, granted September 17, 1996, describes a user interface in a computer system to playback multimedia streams.

[0016] U.S. Patent No. 5,594,660 for *Programmable audio-video synchronization method and apparatus for multimedia systems*, to Sung *et al.*, granted January 14, 1997, describes a means to playback an audio and video stream synchronously to a single network-based receiver.

[0017] U.S. Patent No. 5,598,352 for *Method and apparatus for audio and video synchronizing in MPEG playback systems,* to Rosenau *et al.*, granted January 28, 1997, describes a means to playback an audio and video stream synchronously to a single network-based receiver.

[0018] U.S. Patent No. 5,623,483 for *Synchronization system for networked multimedia streams,* to Agrawal *et al.*, granted April 22, 1997, describes a means for a receiving device to determine the total end-to-end delay from a statistical network delay distribution and the synchronization of the two media streams from different sources through the provision of a synchronization clock located in each server and in each receiver.

[0019] U.S. Patent No. 5,629,936 for *Control of consecutive packet loss in a packet buffer*, to Lee *et al.*, granted May 13, 1997, describes a method for dropping packets to control consecutive packet loss based on a desired packet gap.

[0020] U.S. Patent No. 5,675,573 for *Delay-minimizing system with guaranteed bandwidth delivery for real-time traffic*, to Karol *et al.*, granted October 7, 1997, describes a bandwidth allocation mechanism based on individual guaranteed bandwidth requirements associated with each traffic flow.

[0021] U.S. Patent No. 5,682,384 for *Apparatus and methods achieving multiparty synchronization for real-time network application,* to Zarros, granted October 28, 1997, describes a means to playback contents from multiple sources synchronously to a single receiver on a network.

[0022] U.S. Patent No. 5,768,527 for *Device, system and method of real-time multimedia streaming,* to Zhu *et al.*, granted June 16, 1998, describes a method to recover lost packets by retransrnission.

[0023] U.S. Patent No. 5,802,060 for *Data network and a process for real time transmission of data packets in a data network*, to Fischbach *et al.*, granted September 1, 1998, describes a method to prioritize packet transmissions for stations whose network access is controlled by token exchanges.

[0024] U.S. Patent No. 5,815,634 for *Stream synchronization method and apparatus for MPEG playback system,* to Daum *et al.*, granted September 29, 1998, describes a means to playback an audio and video stream synchronously to a single network-based receiver.

[0025] U.S. Patent No. 5,844,600 for *Methods, apparatus, and systems for transporting multimedia conference data streams through a transport network,* to Kerr, granted December 1, 1998, describes a means for audio/video conferencing over an ATM network. Audio streams are combined in a network-based mixer and distributed to terminals. At each terminal, the received audio and video streams are played synchronously, however, there is no provision for synchronous play at multiple terminals on a network.

[0026] U.S. Patent No. 5,874,997 for *Measuring and regulating synchronization of merged video and audio data*, to Haigh, granted February 23, 1999, describes a means to playback an audio and video stream synchronously to a single, network-based receiver.

LINKPLAY EXHIBIT 1004
Page 1782 of 4822

[0027] The Internet Engineering Task Force (IETF) publishes requests for comments (RFC) and comments filed in response thereto.

[0028] RFC 1889: A real-time end-to-end protocol utilizing existing transport layers for data that has real-time properties.

[0029] RFC 2508: Compressing IP/UDP/RTP Headers for Low-Speed Serial Links

[0030] RFC 1144: Compressing TCP/IP Headers for Low-Speed Serial Links,

[0031] Similar to RFC 2508, but for TCP/IP packets.

Summary of the Invention

[0032] A system for transmitting and synchronizing real-time multimedia content includes a multimedia server for generating a multimedia packet having multimedia audio/visual information therein in a play sequence, and for transmitting the multimedia packet; a packet-based communication network connected to the multimedia server for receiving the multimedia packets therefrom; the network having multiple routers therein to route the multimedia packets to plural destinations; means for inserting total delay information (TDI) into the multimedia packets, wherein the TDI includes total end-to-end delay (TED) and cumulative network delay (CND); and plural multimedia receivers for receiving the multimedia packets having TDI therein, wherein each multimedia receiver includes a buffer for storing received multimedia packets, a sequencing means for retrieving stored multimedia packets from the buffer and sequencing the multimedia packets in the play sequence, and a playing mechanism for playing the retrieved, sequenced multimedia packets at time TED after the multimedia packet has been transmitted by the multimedia server. A method of transmitting and synchronizing real-time multimedia content includes loading multimedia audio/visual into plural multimedia packets; inserting total delay information (TDI) into at least one of the multimedia packets, wherein TDI includes total end-to-end delay (TED) and cumulative network delay (CND); transmitting the multimedia packets to plural multimedia receivers over a packet-based network in a play sequence; buffering the multimedia packets in each multimedia receiver, recalling the multimedia packets in play sequence; and playing the multimedia packets at each receiver at time TED.

[0033] An object of the invention is to provide synchronized A/V output at a plurality of receivers over and multimedia carrying network.

[0034] Another object of the invention is to prevent transmissions of any multimedia packets which will not be used by a receiver on the network.

[0035] These and other objects and advantages of the invention will become more fully apparent as the description which follows is read in conjunction with the drawings.

Brief Description of the Drawings

[0036] Fig. 1 depicts a prior art multimedia content delivery system.

[0037] Fig. 2 depicts a packet-based real-time multimedia content transmission system.

[0038] Fig. 3 is a block diagram of packet receipt and playback.

[0039] Fig. 4 depicts a LAN multimedia content delivery system constructed according to the invention.

[0040] Fig. 5 depicts a multimedia content delivery system wherein the video and audio streams are separately routed.

Detailed Description of the Preferred Embodiment

[0041] Synchronization and transmission of multicast multimedia, particularly over packet-based systems is an efficient and inexpensive techniques for distributing information. Prior art systems simply do not consider the problems associated with synchronizing and transmitting a multimedia data stream, from one or more sources, over a packet-based system, to multiple receivers. Although there exist a number of technologies that enable a plurality of correlated media streams, transmitted from different locations, to be played at single receiver in a synchronized manner, the known prior art does not allow media streams generated by a single source, or multiple sources, to be multicast over a packet-based network to be played synchronously at multiple locations. This invention provides a system and method to transmit and synchronously play the same audio/video stream at the different locations, and to eliminate transmission of a multimedia packet when it is apparent that the multimedia packet will not be played at its destination.

[0042] Referring now to Fig. 4, a content synchronized multimedia transmission system constructed according to the invention is depicted generally at 60. System 60 includes a multimedia server 62, which has the means to insert TED and CND information, which is defined and described later herein, into each multimedia packet 64 that passes therethrough. A packet network 66 includes two routers, 68, 70, although many more routers are likely present in such a generalized network. Router 68 is connected to a local area network (LAN) 72, which includes a gateway 74 and

LINKPLAY EXHIBIT 1004
Page 1783 of 4822

routers 76, 78. Multimedia receivers 80, 82 and 84 are connecter to LAN 72.

[0043] The method of the invention includes loading multimedia information into a series of packets 64, referred to herein as multimedia packets. Multimedia server 62 inserts fields of total end-to-end delay (TED) and cumulative network delay (CND) in each multimedia packet, referred to herein as total delay information (TDI). TED indicates the time from when the multimedia packet is transmitted from the server to when the multimedia packet is to be played at a receiver, whereas CND indicates the actual latency caused by routers in the communications network before the multimedia packet is delivered to a receiver. Multimedia server 62 determines the value of TED from network statistics. When transmitting a multimedia packet, server 62 sets the value of jitter to CND. The jitter is obtained by

$$\text{time interval of transmissions of two consequent packets - packet rate.} \quad (1)$$

For example, if the packet rate is 20 ms and a packet is actually transmitted 30 ms after the previous packet, jitter is 10 ms and the CND of the packet is set to 10 ms.

[0044] Each router in the network, regardless of whether the router is located in packet network 66 or LAN 72, measures the queuing delay of each multimedia packet caused by that router. The queuing delay is obtained from time stamps of reception and transmission of the multimedia packet. When transmitting the multimedia packet, each router updates CND by adding the queuing delay to the current CND value. If the router needs to route a received multimedia packet to multiple routes, *i.e.,* multicast, such as router 68 multicasting to router 70 and gateway 74, the multimedia packet is duplicated, and the queuing delay for the duplicated multimedia packet is handled separately from the original. There may be more than one duplicated packet, and the queuing delay is placed in each duplicated packet.

[0045] Each multimedia receiver includes a buffer "B" to store incoming multimedia packets, and recalls a multimedia packet from the buffer to play the content of a multimedia packet at (TED-CND) after its reception. Thus, a multimedia packet delivered to the multiple multimedia receivers is played at TED after transmitted at the server, regardless of the actual network delay caused in the route to each receiver.

[0046] As described in the previous example, CND contains only a queuing delay, and does not include physical latencies caused by the physical transmission between two routers. The physical latency is assumed to be small and is therefore negligible. Alternatively, a router may be provided with a known physical latency, which may be added to the queuing delay for increased accuracy.

[0047] If all the receivers are aware of the packet rate, performing the procedures described above for the first multimedia packet of the media stream is sufficient, because a receiver receiving a multimedia packet with TED and CND knows when to play the multimedia packets which follow, however, where the first multimedia packet containing TED and CND is "lost," *i.e.*, not received by one or more receivers, the server may be required to include TED and CND in the first several multimedia packets. Alternatively, the server may be required to include TED and CND until it receives an acknowledgement that a multimedia packet having TED and CND has been received by all of the receivers belonging to the multicast group.

[0048] As mentioned earlier, multimedia server 62 determines the value of TED, based on network statistics. TED may be provided to the server by a network administrator, or measured dynamically. The actual determination of TED is beyond of the scope of this invention, however, U.S. Patent No. 5,623,483 provides a technique to determine TED from the statistical distribution of network delay. The overall solution of U.S. Patent No. 5,623,483, however, is that the server and the receivers have synchronized clocks. The server specifies, in each packet, the time at which the receiver is to play the packet content. However, it is unrealistic to expect that the servers and receivers of a packet network maintain precisely the same time on their synchronized clocks.

[0049] It is foreseeable that situations may arise where queuing delays and physical latencies add up to the point where a multimedia packet reaches a receiver, and is "late." In the case of prior art systems, audible and visual discontinuities are perceptible to a user, without any means to correct this situation. The system and method of the invention have provisions to function in the case where one of the receivers detects a CND in a multimedia packet which exceeds the TED for the multimedia packet. In this case, the multimedia packet cannot be played synchronously, because the time for play of the contents of the multimedia packet has already passed. The receiver includes means to request a larger TDI, and sends a message to the server with the value CND-TED. The server then increases the value of TED, and thus TDI, for following multimedia packets.

[0050] In the embodiment shown in Fig. 4, server 62 and all the routers included in packet network 66 and LAN 72 for the multicast group supports TED and CND. As depicted in Fig. 4, all of the multimedia receivers in the multicast group are in a local area network (LAN), however, a multimedia receiver may also be connected to a router in packet network 66. A multimedia server does not have to include special hardware or software to insert TDI into the multimedia packets, as described above. Instead, a gateway, which connects the LAN to the (public) packet network, generates TDI from TED and CND for multimedia packets received from the server. The gateway must be capable of generating the value of TED, and must be supplied with the packet rate. The routers and the receivers in the LAN may perform

LINKPLAY EXHIBIT 1004
Page 1784 of 4822

the same function. Consequently, the invention may be implemented even if the entire packet network does not support the method presented here. Put another way, the means for inserting TDI may be provided by a multimedia server, a router, or a gateway.

[0051] Another embodiment, illustrated in Fig. 5, generally at 90, includes a multimedia server 92, which generates a video stream 94 and an audio stream 96, a packet network 98, having routers 100, 102, 104, 106 and 108, and a video monitor 110 and a speaker 112. Video stream 94 and audio stream 96 are to be played separately yet synchronously at video monitor 110 and at speaker 112, respectively, which are connected to packet network 98. The destination addresses of video packets and audio packets are the addresses for the video monitor and for the speaker, respectively. Server 92 includes means to insert TDI, including TED and CND, in both video and audio packets. The routers perform the same procedure as shown in the other embodiments.

[0052] The method of U.S. Patent No. 4,506,358 is different from the invention described herein in that the present invention employs an information field in each packet of a type neither taught nor suggested in the prior art, that indicates the time when the audio/video content in the packet should be played. This enables the same content to be played synchronously at multiple receivers, without the necessity of synchronized server/receiver clocks.

[0053] This invention also maximizes the efficiency of bandwidth on physical links among routers and receivers by preventing a router from transmitting multimedia packets that will not be played at the final destination, *i.e.*, a media player, or receiver, as in the case where CND > TED.

[0054] Non-viable multimedia packets may be discarded if it is determined that they are not need, or if it is determined that they will not be used. The following may be used as a stand-alone technique, or in conjunction with the synchronization technique previously described herein. A router places a time stamp and sequence number in every received multimedia packet transferred between the media source and the receiver. Let:

$t_{ts}[n]$: the time stamp of the multimedia packet with sequence number n, and

$t_p[n]$: the timing information included in the multimedia packet with sequence number n, indicating when the content in this multimedia packet is to be played at the receiver. This information provides relative timing information among series of multimedia packets and does not indicate the absolute time.

[0055] At the receipt of the first multimedia packet, having sequence number = n_0, the router stores $t_{ts}[n_0]$ and $t_p[n_0]$ in memory and re-directs the multimedia packet to the next router or the receiver. Now, let

$$T_u = t_u[n_0] \text{ and} \qquad (2)$$

$$T_p = t_p[n_0]. \qquad (3)$$

[0056] At the receipt of every following received multimedia packet with sequence number n, the router examines the values of $t_{ts}[n]$ and $t_p[n]$, and, if

$$t_{ts}[n] - T_{ts} < tp[n] - T_p + D, \qquad (4)$$

where D denotes the maximum TED allowance, including the buffering delay in the receiver, the router re-directs the multimedia packet and sets

$$T_{ts} = t_{ts}[n] \text{ and} \qquad (5)$$

$$T_p = t_p[n]; \qquad (6)$$

and determines that the multimedia packet is a viable packet. Otherwise, the router discards the multimedia packet.

[0057] The value $t_{ts}[n] - T_{ts}$ is the difference between the arrival time of the multimedia packet of interest and the latest successfully re-directed multimedia packet at the router. Similarly, $t_p[n] - T_p$ indicates the time interval at play back between these two multimedia packets. Thus, if

LINKPLAY EXHIBIT 1004
Page 1785 of 4822

$$t_{ts}[n] - T_{ts} > t_p[n] - T_p + D, \quad (7)$$

the multimedia packet with sequence number n has already consumed the entire TED allowance. In this case, there is no chance for the multimedia packet to be played at the receiver, because the multimedia packet will not arrive at the receiver before or at the time it is to be played. By eliminating unnecessary transmission of such a multimedia packet, referred to herein as a non-viable packet, the bandwidth on the physical link between this router and the next entity, regardless of whether the next entity is a router or a receiver, may be utilized efficiently. The router thus includes means to determine whether a multimedia packet is a viable multimedia packet, in which case the multimedia packet is transmitted; or whether the multimedia packet is a non-viable multimedia packet, in which case the multimedia packet is discarded.

[0058] The value of D at any router may be provided by: (1) being hard-coded on the node's software or hardware; (2) being manually configured by a system administrator; or (3) being provided in every real-time multimedia packet by the multimedia server. The value is determined by the required QoS and must be greater than the buffering delay in the receiver. D will be a small value for delay-sensitive applications, such as two-way voice conversation. The receiver may negotiate the value of D with the media source using common communication protocols.

[0059] The transmission method of the invention functions with IP/UDP/RTP header compressor, specified in RFC 2508, wherein the header information of real-time multimedia packets is compressed by a differential coding scheme. The compressor, which may be implemented in a transmitting portion of a router, provides a means to identify a specific real-time media stream between a media source and a receiver, among other traffic. Because of differential coding, the compressor is required to examine the timing information in the header of real-time multimedia packets in any packet-based network, and the extra processing power required to implement the method of the invention is minimal.

[0060] Thus, a system and method for transmitting and synchronizing real-time, multimedia packet-based content over a packet-based network has been disclosed. Although several variations of the system and method of the invention have been disclosed, it will be appreciated that further variations and modification may be made thereto without departing from the scope of the invention as defined in the appended claims.

Claims

1. A system (60) for transmitting and synchronizing real-time multimedia content, comprising:

   a multimedia server (62) for generating a multimedia packet (64) having multimedia audio/visual information therein in a play sequence, and for transmitting said multimedia packet (64);
   a packet-based communication network (66) connected to said multimedia server (62) for receiving said multimedia packets therefrom; said network having multiple routers (68, 70) therein to route said multimedia packets (64) to plural destinations;
   means for inserting total delay information (TDI) into said multimedia packets (64), wherein said TDI includes total end-to-end delay (TED) and cumulative network delay (CND); and
   plural multimedia receivers (80, 82, 84) for receiving said multimedia packets (64) having TDI therein, wherein each multimedia receiver (80, 82, 84) includes a buffer for storing received multimedia packets, a sequencing means for retrieving stored multimedia packets from said buffer and sequencing said multimedia packets in said play sequence, and a playing mechanism for playing the retrieved, sequenced multimedia packets at time TED after said multimedia packet (64) has been transmitted by said multimedia server.

2. The system (60) of claim 1 wherein said means for inserting is located in said multimedia server (62).

3. The system (60) of claim 1 wherein said means for inserting is located in a router (68, 70).

4. The system (60) of claim 1 which further includes a local area network (LAN) (72) located between said packet-based network (66) and at least some of said multimedia receivers (80, 82, 84), wherein said LAN (72) includes a gateway (74).

5. The system (60) of claim 4 wherein said means for inserting is located in said gateway (74).

6. The system (60) of claim 1 wherein at least one of said routers (68, 70) includes duplication means for duplicating multimedia packets (64), wherein said means for inserting is located in said router (68, 70), and wherein said router

LINKPLAY EXHIBIT 1004
Page 1786 of 4822

(68, 70) inserts TDI into each duplicated multimedia packet.

7. The system (60) of claim 1 which further includes means to determine whether a multimedia packet (64) is a viable packet or whether a multimedia packet is a non-viable packet, and wherein said non-viable packets are discarded.

8. The system (60) of claim 1 wherein said multimedia receivers (80, 82, 84) includes means to request a larger TDI from said means for inserting.

9. A method of transmitting and synchronizing real-time multimedia content, comprising:

loading multimedia audio/visual into plural multimedia packets (64);
inserting total delay information (TDI) into at least one of the multimedia packets (64), wherein TDI includes total end-to-end delay (TED) and cumulative network delay (CND);
transmitting the multimedia packets (64) to plural multimedia receivers (80, 82, 84) over a packet-based network (66) in a play sequence;
buffering the multimedia packets (64) in each multimedia receiver (80, 82, 84);
recalling the multimedia packets (64) in play sequence; and
playing the multimedia packets (64) at each receiver (80, 82, 84) at time TED.

10. The method of claim 9 wherein said inserting includes inserting TDI in a multimedia server (62).

11. The method of claim 9 wherein said inserting includes inserting TDI in a router (68, 70) located in the packer-based network (66).

12. The method of claim 9 wherein said inserting includes inserting TDI in a gateway (74) located in a local area network (LAN) (72) which is located between the packet-based network (66) and a multimedia receiver (80, 82, 84).

13. The method of claim 9 which includes duplicating a multimedia packet (64) for multicasting to plural multimedia receivers (80, 82, 84), and wherein said duplicating includes inserting TDI into at some of the duplicated multimedia packets.

14. The method of claim 9 which includes discarding a multimedia packet (64) if at any time during said transmitting CND>TED.

15. The method of claim 14 which includes increasing the value of TED if CND>TED.

16. The method of claim 9 which includes determining whether a multimedia packet (64) is a viable packet or a non-viable packet, and which further include discarding any non-viable packets.

17. A method of transmitting and synchronizing real-time multimedia content, comprising:

determining whether a multimedia packet (64) is a viable packet or a non-viable packet; and
discarding any non-viable packets.

18. The method of claim 17 which further includes:

loading multimedia audio/visual into plural multimedia packets (64);
inserting total delay information (TDI) into at least one of the multimedia packets (64), wherein TDI includes total end-to-end delay (TED) and cumulative network delay (CND);
transmitting the multimedia packets (64) to plural multimedia receivers (80, 82, 84) over a packet-based network (66) in a play sequence;
discarding a multimedia packet (64) if at any time during said transmitting CND>TED;
buffering the multimedia packets (64) in each multimedia receiver (80, 82, 84);
recalling the multimedia packets (64) in play sequence; and
playing the multimedia packets (64) at each receiver (80, 82, 84) at time TED.

19. The method of claim 18 wherein said inserting includes inserting TDI in a multimedia server (62).

LINKPLAY EXHIBIT 1004
Page 1787 of 4822

20. The method of claim 18 wherein said inserting includes inserting TDI in a router (68, 70) located in the packet-based network (66).

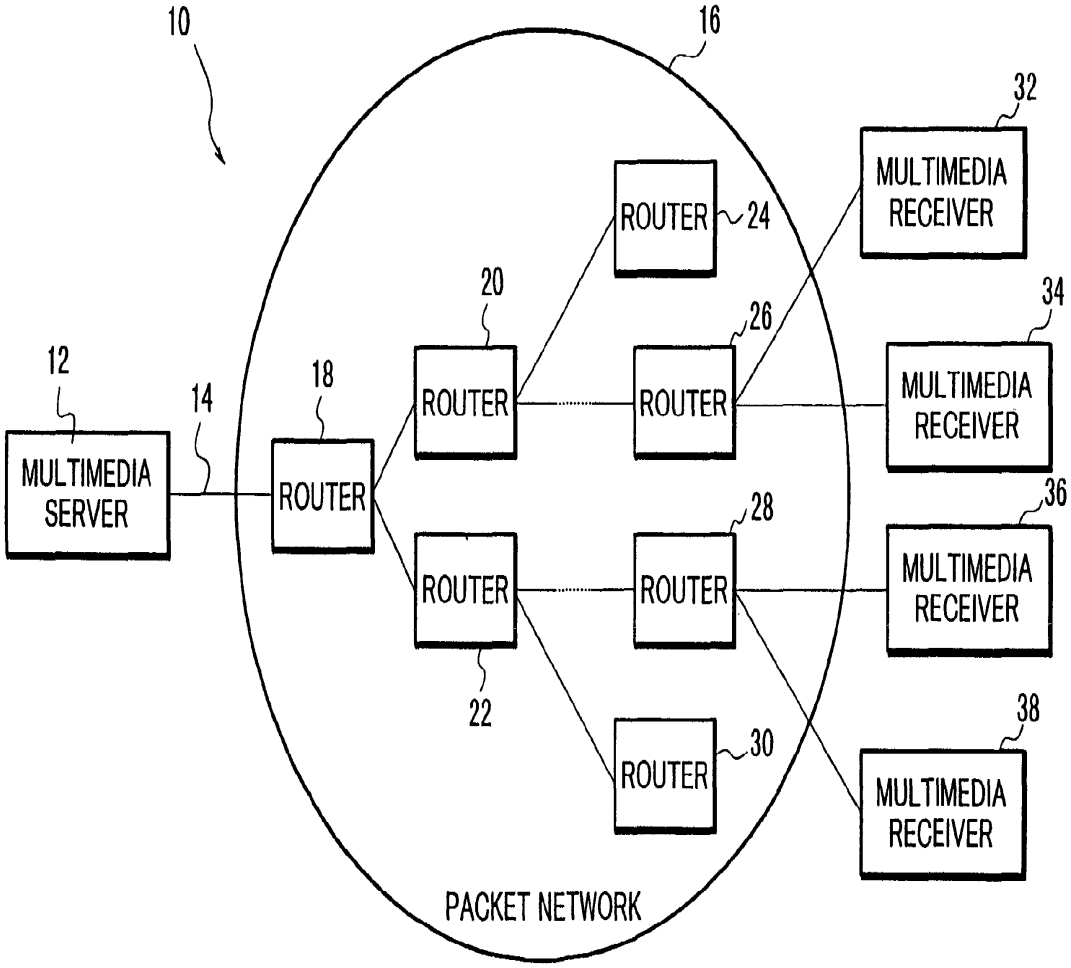
21. The method of claim 18 wherein said inserting includes inserting TDI in a gateway (74) located in a local area network (LAN) (72) which is located between the packet-based network (66) and a multimedia receiver (80, 82, 84).

22. The method of claim 18 which includes duplicating a multimedia packet (64) for multicasting to plural multimedia receivers (80, 82, 84), and wherein said duplicating includes inserting TDI into at some of the duplicated multimedia packets.

23. The method of claim 18 which includes increasing the value of TED if CND>TED.

LINKPLAY EXHIBIT 1004
Page 1788 of 4822

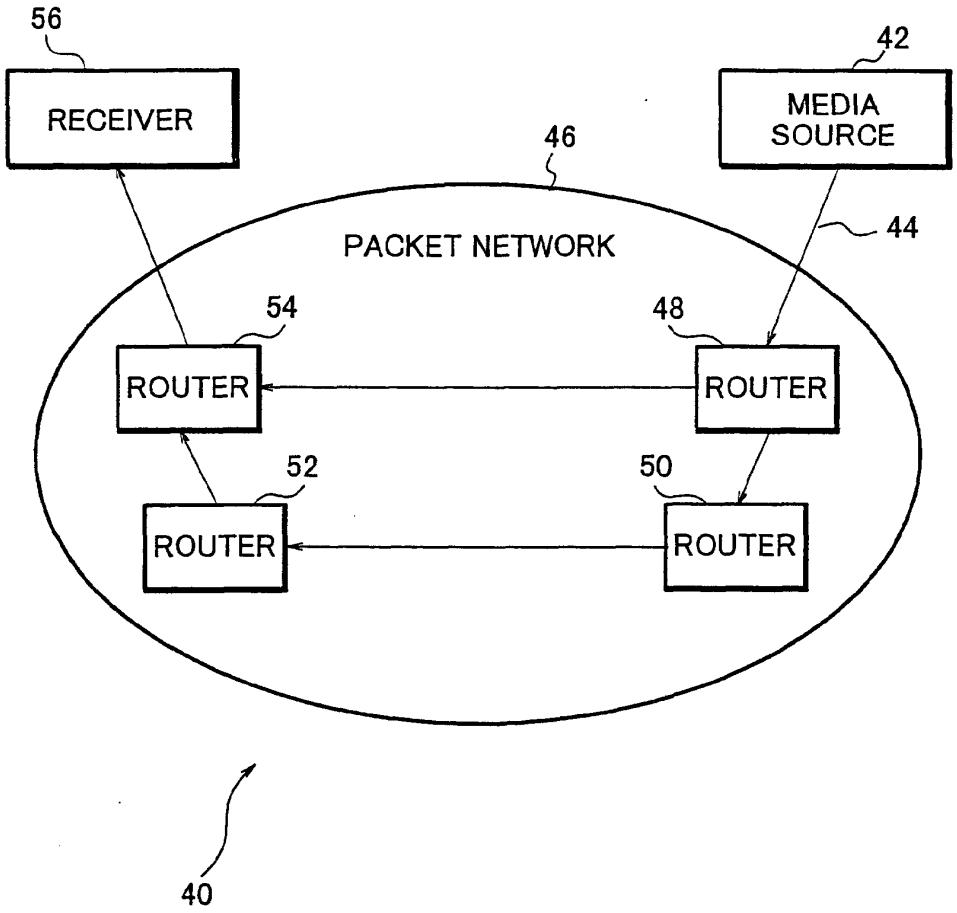
FIG.1
(PRIOR ART)



EP 1 122 931 A2

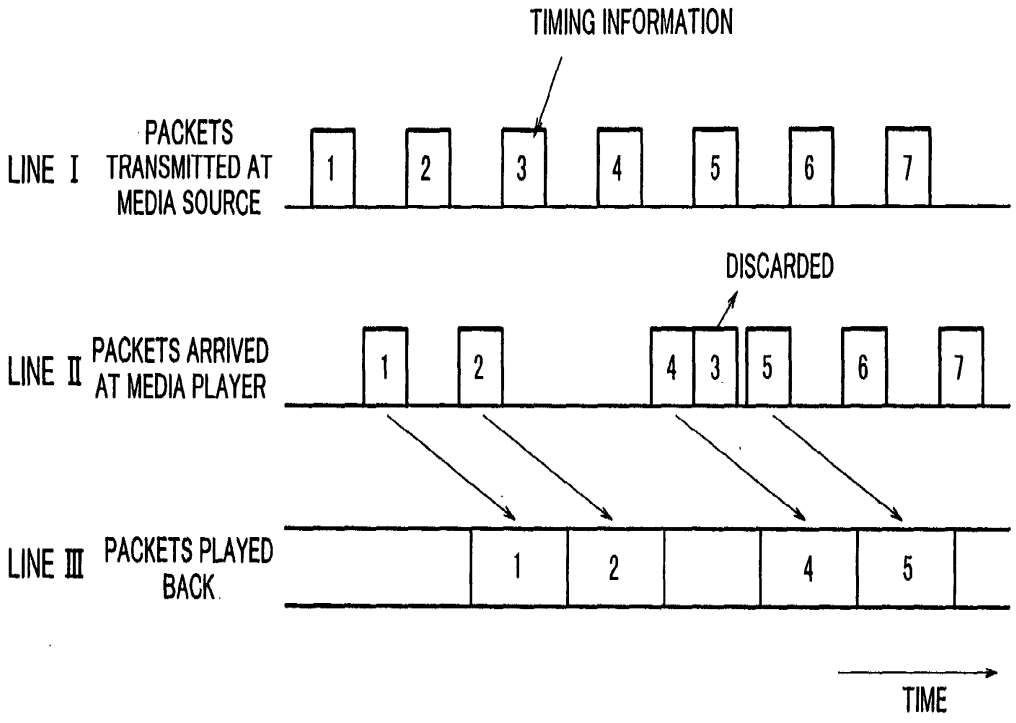
LINKPLAY EXHIBIT 1004
Page 1789 of 4822

FIG.2



LINKPLAY EXHIBIT 1004
Page 1790 of 4822

FIG.3


TIMING INFORMATION
LINE I
PACKETS TRANSMITTED AT MEDIA SOURCE
1
2
3
4
5
6
7
DISCARDED
LINE II
PACKETS ARRIVED AT MEDIA PLAYER
1
2
4
3
5
6
7
LINE III
PACKETS PLAYED BACK
1
2
4
5
TIME

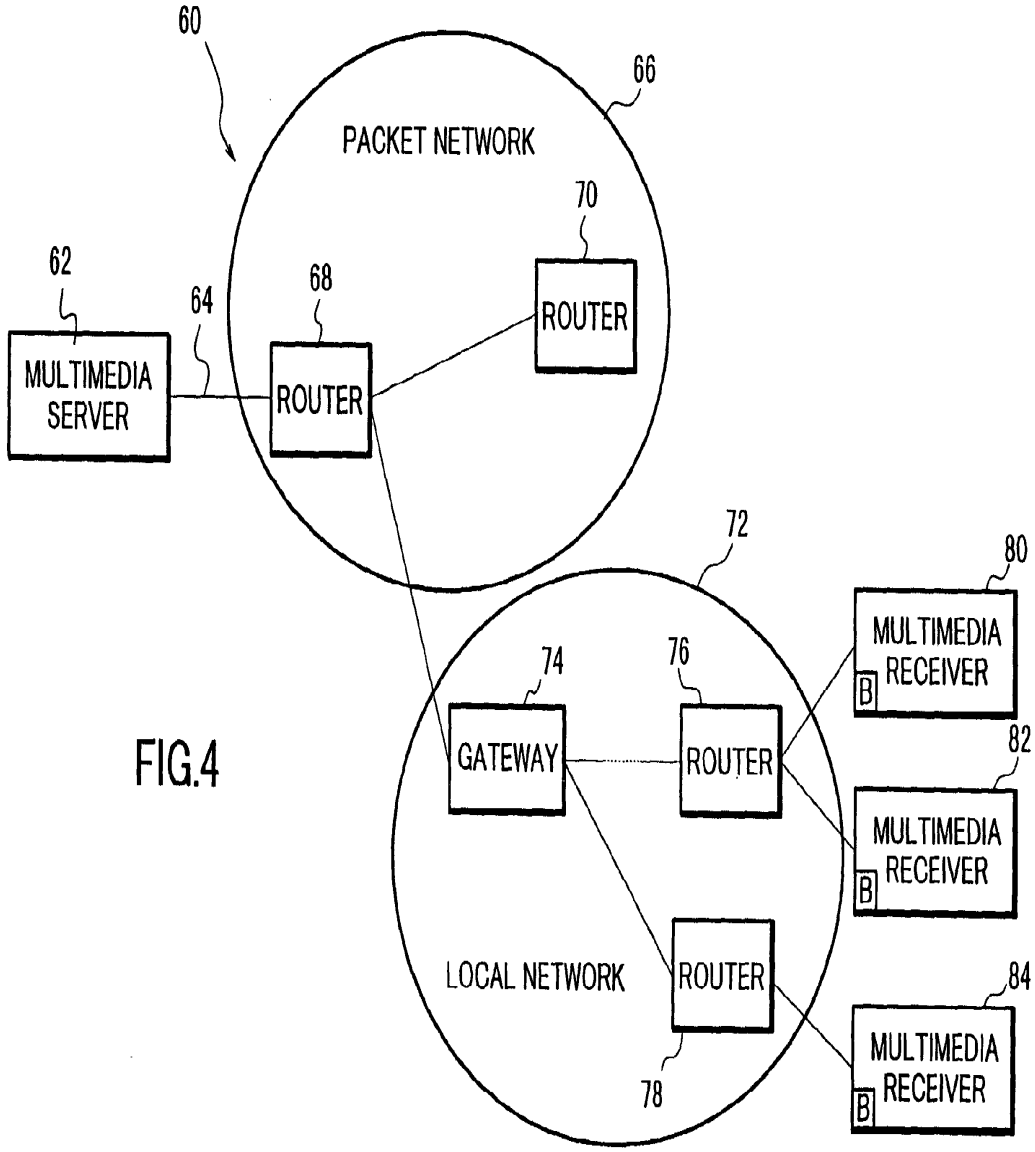
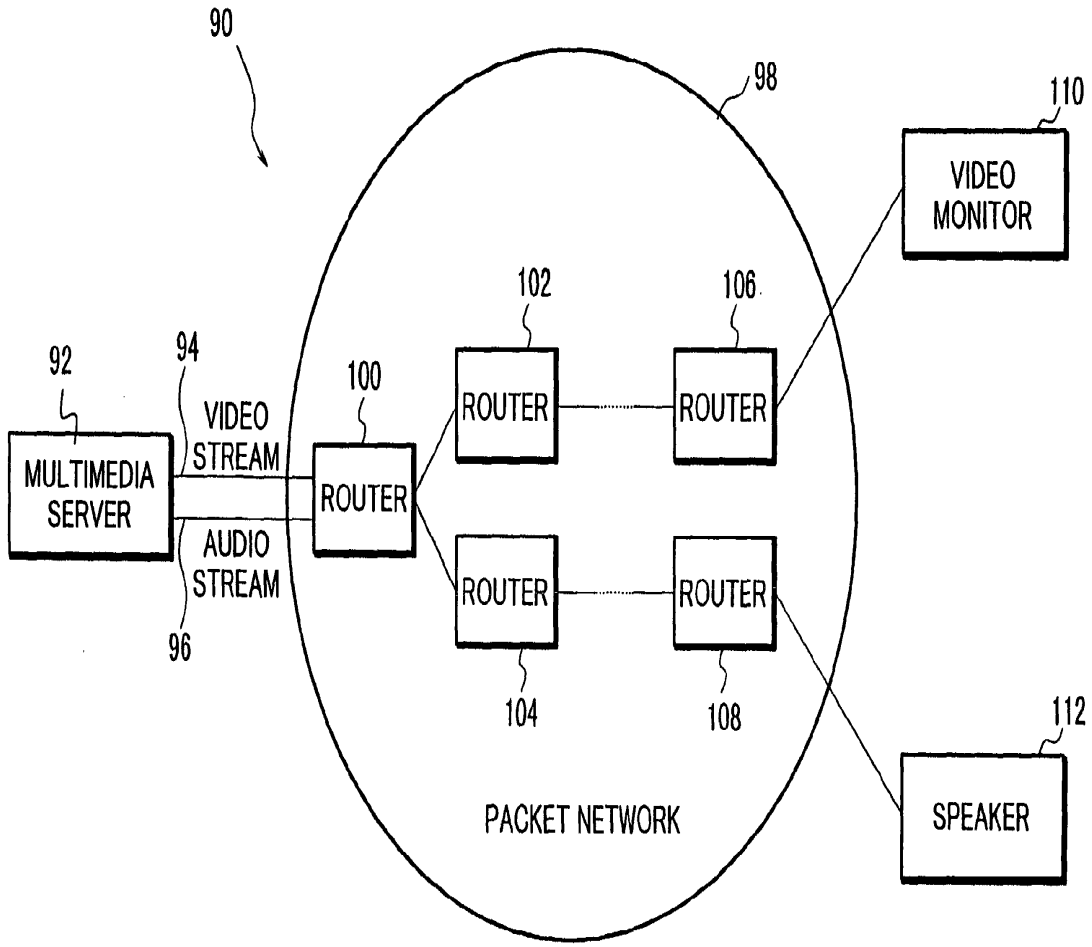

60
66
PACKET NETWORK
70
62
68
64
ROUTER
MULTIMEDIA SERVER
ROUTER
72
80
MULTIMEDIA RECEIVER
B
74
76
GATEWAY
ROUTER
82
MULTIMEDIA RECEIVER
B
LOCAL NETWORK
ROUTER
84
MULTIMEDIA RECEIVER
B
78

FIG.4

FIG.5



EP 1 122 931 A2

LINKPLAY EXHIBIT 1004
Page 1793 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 312 188 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
26.09.2007 Bulletin 2007/39

(21) Application number: **01960879.3**

(22) Date of filing: **09.08.2001**

(51) Int Cl.:
H04L 12/64 *(2006.01)*

(86) International application number:
PCT/GB2001/003595

(87) International publication number:
WO 2002/017579 (28.02.2002 Gazette 2002/09)

(54) **AUDIO DATA PROCESSING**

AUDIODATENVERARBEITUNG

TRAITEMENT DE DONNEES AUDIO

(84) Designated Contracting States:
DE FR GB

(30) Priority: **25.08.2000 EP 00307348**

(43) Date of publication of application:
21.05.2003 Bulletin 2003/21

(73) Proprietor: **BRITISH TELECOMMUNICATIONS public limited company**
London EC1A 7AJ (GB)

(72) Inventors:
- **RADENKOVIC, Milena**
  East Midlands NG1 5DU (GB)
- **GREENHALGH, Christopher, Michael**
  Nottingham NG11 6DR (GB)

(74) Representative: **Nash, Roger William et al**
BT Group Legal
Intellectual Property Department
PP C5A
BT Centre
81 Newgate Street
London EC1A 7AJ (GB)

(56) References cited:
EP-A- 1 024 638

- **KYEONG-YEOL YU, JONG-HOON PARK, JONG-HYEONG LEE: "Linear PCM Signal Processing for Audio Processing Unit in Multipoint Video Conferencing System" IEE, July 1998 (1998-07), pages 549-553, XP002159070 KOREA**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

EP 1 312 188 B1

Description

[0001] This invention relates to audio data processing and in particular concerns a method, system and software application program for a real-time audio service for processing audio data streams transmitted over a communications network.

[0002] Developers of real-time audio services have for some time recognised the need for scalability in terms of the number of simultaneous listeners to an audio stream. Applications such as video on demand, on-line lectures and Internet radio require an audio stream to be broadcast from a single source to potentially many listeners. Various techniques have been proposed for reducing the bandwidth required for such broadcasts, most notably network multicasting including Internet multicasting, for example. By comparison however, support for many simultaneous speakers has not been fully developed. Some telephony and CSCW (Computer Supported Co-operative Work) applications such as audio and video-conferencing do support several simultaneous speakers, but the focus has been mostly on small groups. In addition, where the possibility of simultaneous speaking has been acknowledged applications usually prevent it through the use of floor control and "push-to-talk" concepts that enforce turn-taking and ensure that only one participant speaks at a time.

[0003] Communication protocols often discount the possibility of multiple simultaneous speakers. The Internet standard Real Time Protocol (RTP) (IETF RFC 1889) is the standard packet format used for continuous media traffic, such as audio streams, on the Internet. RTP includes sophisticated algorithms to control the amount of management traffic placed on the network, but assumes that audio traffic will not be a problem: "For example, in any audio conference the data [audio] traffic is inherently self-limiting because only one or two people will speak at a time ...", (RFC 1889, section 6.1). Similarly, the multicast Internet backbone (MBone), which provides the Internet's wide-area multicast capabilities, does not actively support simultaneous speakers. The Mbone guidelines for use and resources assume that each audio session will not have more than one active speaker at a time. Telecommunication applications are now emerging that support on-line events involving large groups of people. For example, collaborative virtual environments (CVEs) support interactive social events such as multi-player games and inhabited television for large on-line communities. Speech is the primary communication medium for on-line social interaction and so real-time audio services are vital for these new applications. Indeed, speech is arguably the most critical aspect of any real-time collaborative application. Research has shown that problems with other aspects of collaborative systems such as video, shared tools, or 3D graphics, can be resolved or compensated for by speech.

[0004] Analysis of patterns of audio activity in CVE applications, for example virtual teleconferencing, has revealed significant periods when several participants are simultaneously generating audio traffic. In relatively focused applications such as tele-conferencing, audio activity is best approximated by a model of people transmitting audio at random, rather than deliberately avoiding overlapping speech. Indeed, overlapping audio, including speech and other sounds, is likely to be a basic requirement for many CSCW applications

[0005] Unfortunately, there are many disadvantages associated with real-time audio services that allow many simultaneous speakers. Real-time audio is bandwidth intensive, particularly for applications that support large numbers of users. In addition, packet-loss due to network congestion can severely reduce the intelligibility of received audio streams.

[0006] Providing an audio service for many simultaneous speakers is a significant problem since each speaker independently introduces a new audio stream that has to be accommodated by the network and that also has to be processed by each recipient's audio receiver. While multicasting reduces the bandwidth required to distribute a single audio stream to many listeners, it does not address the problem of many simultaneous speakers.

[0007] Audio services are known in which peer-to-peer architectures provide each user with an independent audio stream from each respective speaker. By receiving independent audio streams each listener can create their own personal audio mix. The mix can be tailored to the user's own audio equipment capabilities; the audio streams can be spatialised according to associated speaker locations within a shared virtual space; and the streams can be individually controlled, for example allowing the volume of particular speakers to be raised or lowered according to their relative importance, for example. Peer-to-Peer audio services can be implemented using unicast or multicast protocols.

[0008] The peer-to-peer approach is very demanding in terms of network resources, particularly bandwidth, and can easily flood the network with traffic. With underlying unicast protocols, which are typically used today for wide-area communication, the resulting number of audio streams is of the other n^2, where n is the number of simultaneous users, that is to say for the case in which all users are sending audio data simultaneously. With underlying multicast protocols, which are still experimental over wide-area networks, this reduces to the order n, but no lower.

[0009] With conventional networks such as the Internet, once the traffic in any part of the network exceeds the local capacity then that part of the network becomes congested. This congestion causes speech to become delayed (due both to direct delay and also jitter) and increasingly broken and disjointed (due to audio data packets being lost in transit). The peer-to-peer approach is also very demanding in terms of the processing to be done by each listener's terminal, which must be capable of receiving, decoding, processing, mixing and playing out all of the available audio streams. User terminals may

LINKPLAY EXHIBIT 1004
Page 1795 of 4822

become overloaded, causing problems similar to an overloaded network.

[0010] Audio services which use total mixing are also known. In the total mixing approach each audio stream is sent to a central audio server that mixes the audio streams into a single stream that is then redistributed to each listener. This approach requires that each listener and surrounding network handles only one audio stream. Total mixing, however, prevents each listener from creating their own personal mix and the central server becomes a potential bottleneck in the system, since it and its nearby network still have to receive, mix and distribute n audio streams. Other drawbacks associated with total mixing include increased delay due to additional processing, reduced audio fidelity, additional hardware requirements and management complexity. In this respect total mixing is only appropriate for relatively simple applications where resources are limited. Total mixing is not appropriate for high fidelity applications such as home entertainment and tele-conferencing where more resources are generally available and where audio spatialisation is more important, that is to say where separate audio streams are required. An example of total mixing is described by one of the present inventors in "Inhabited TV: Multimedia Broadcasting from Large Scale Collaborative Virtual World", Fracta Universitatisi, Ser. Electronics and Energetics, 13(1) IEEE, ISBN 0-7803-5678-X.

[0011] Another example of an audio service that supports many simultaneous speakers is described in "Diamond Park and Spline: A Social Virtual Reality System With 3D Animation, Spoken Interaction, and Runtime Extendibility", PRESENCE, 6(4), 461-481, 1997 MIT. This paper describes a system which allow users with low bandwidth connections to access an audio-graphical multi-user virtual world. In this approach one or more low-bandwidth users connect to a specialised access server that interfaces to a main system which is configured for peer-to-peer multicast audio. The access server or servers deal with and mix all of the available audio streams. It is also known from EP-A-1 024 638 to direct terminals to adjust their coding algoritms based on system bandwidth usage. According to an aspect of the present invention there is provided a method of processing audio data streams transmitted in a communications network; said method comprising the steps of:-

i) receiving a plurality of audio data streams transmitted from one or more audio data stream transmitters distributed in the network; and,

ii) processing data relating to at least one respective network resource parameter to determine respective network resources available for subsequent communication of said audio data streams to at least one respective audio data stream receiver in the network; and,

iii) comparing said available resources with respective network resource requirements necessary for communicating said audio streams to at least one respective audio data stream receiver in the network; and,

iv) determining whether to mix selected audio data streams prior to transmission in response to said comparison.

[0012] The term "network resources" used herein refers to any component in a communications network necessary for communicating audio data streams to potential recipients, including but not limited to network communication links, network audio processors or mixers, audio stream transmitters and receivers and user terminals, for example.

[0013] By comparing available network resources with network resource requirements necessary for communicating audio streams the above method enables audio mixing decisions to be made dynamically in response to changing network conditions or application requirements. In this way the number of data streams transmitted can be controlled so that network traffic can be optimised according to available network resources. This aspect of the invention is particularly relevant for dynamic virtual environment applications involving varying numbers of active participants engaged in various activities and running over dynamic networks where congestion and delay may change rapidly. The method enables mixing decisions to be made that adapt to changing network conditions and application requirements so as to optimise the conflicting requirements of audio quality and traffic management.

[0014] Preferably the method further comprises the step of:-

v) processing two or more audio streams in response to said comparison to provide at least one mixed audio data stream for subsequent transmission in said network. Thus, if a decision is made to mix selected audio data streams the traffic introduced into the network can be reduced. The amount of processing required by neighbouring data stream receivers can also be reduced. Mixing enables two or more selected audio data streams to be combined to reduce overall network congestion without significantly affecting the intelligibility of the mixed audio streams when received by a user, for example in a similar way that stereo audio signals can be combined for playback on a non-stereo enabled output device such as a hand held radio receiver having a single loudspeaker.

[0015] Conveniently, steps ii) and iii) comprise the steps of:-

determining a current value for the or each respective network resource parameter; and,

comparing the or each respective current resource parameter value with a respective minimum resource threshold value necessary for communicat-

LINKPLAY EXHIBIT 1004
Page 1796 of 4822

ing said unmixed audio data streams to the or each respective receiver. In this way the available network resources can be compared with pre-determined minimum resource requirements necessary for transmitting or processing all the unmixed data streams.

[0016] In preferred embodiments, said minimum resource threshold value is determined according to at least one pre-defined quality of service parameter. In this way mixing decisions can be made according to pre-determined quality of service requirements.

[0017] Preferably, said network is a packet switched network and said pre-defined quality of service parameter is defined by a maximum packet loss rate. Thus, a minimum threshold value may be determined according to an acceptable packet loss rate associated with a codec used to encode respective audio streams. For instance, a maximum acceptable packet loss rate for an audio codec may be 15%.

[0018] Conveniently, one network resource parameter relates to available network bandwidth for transmission of said audio data streams to the or each respective receiver. In this way mixing decisions can be determined according to the bandwidth available for transmitting the audio streams to a next audio stream receiver in the network. This provides for efficient use of network bandwidth and readily enables a maximum number of mixed or unmixed audio data streams to be transmitted without causing congestion in the network.

[0019] Preferably, said available bandwidth capacity is determined by user specific quality of service requirements. In this way bandwidth resources can be allocated or reserved for use according to user specified quality of service requirements. The allocation or reservation of bandwidth may be controlled by different charging tariffs associated with the quality of service required. In this way a user may specify a quality of service requirement of say 3 x 64kb/s audio channels in which case selected audio streams will be mixed by the network when more than three separate audio streams are to be transmitted by the network. In this way available bandwidth may be considered as allocated or reserved bandwidth.

[0020] Conveniently, one network resource parameter relates to receiver processing characteristics. Thus, mixing decisions can be determined according to the characteristics of respective receivers. In this way separate audio streams being sent to a receiver having a low processing capability or capacity can be mixed in the network so that the number of audio streams the receiver receives is reduced.

[0021] In preferred embodiments, said audio data streams are selected for mixing according to predetermined criteria. This enables selection criteria to be used to determine which of the received audio data streams should be mixed.

[0022] In one way, said audio streams are mixed according to audio stream content. For instance, in a virtual environment one or more audio streams may be more significant in terms of audio content than others and the loss of audio spatialisation experienced by a recipient will be less if the less significant audio streams are mixed in preference to the more significant ones.

[0023] In another way, said audio streams are mixed according to recipient requirements. In this way mixing can be determined by the recipient's own requirements, for instance the extent of audio spatialisation required.

[0024] In a further way, said audio streams are mixed according to respective audio stream sources. In this way audio streams from related sources can be mixed, for instance it may be desirable to mix all audio streams associated with a particular group of participants in an audio conference or virtual environment.

[0025] In a yet further way, said audio streams are mixed according to receiver capabilities. In this way audio streams may be mixed according to the capabilities of the receiver. For example, a receiver may comprise a full 3-D audio system capable of recreating fully spatialised studio quality audio where mixing considerations are important for recreating spatialised audio. Alternatively a receiver may comprise a simple stereo audio system where mixing considerations are less important.

[0026] According to another aspect of the invention there is provided a software program arranged to process audio data streams according to the above mentioned method.

[0027] According to a further aspect of the invention there is provided a system for processing audio data streams transmitted in a communications network; said system comprising:-

i) a receiver for receiving a plurality of audio data streams transmitted from one or more audio data stream transmitters distributed in the network; and,
ii) a processor for processing data relating to at least one respective network resource parameter to determine respective network resources available for subsequent transmission of said audio data streams in the network; and,
iii) a comparator for comparing said available resources with respective network resource requirements necessary for transmission of said audio streams to at least one respective audio data stream receiver in the network; and,
iv) determining means for determining whether to mix selected audio data streams prior to transmission in response to said comparison.

[0028] The invention will now be described by way of example only with reference to the accompanying drawings in which:-

Figure 1 is a schematic representation of a network used for implementing an embodiment of the present invention;
Figure 2 is a schematic representation of a logical

LINKPLAY EXHIBIT 1004
Page 1797 of 4822

network topology for part of the network of Figure 1;
Figure 3 shows a modular block diagram of an audio data processor for processing audio data streams transmitted in the network of Figures 1 and 2;
Figure 4 is a flowchart showing steps involved in implementing an embodiment of the invention;
Figure 5 is a schematic representation of a distributed processing system for simulating network conditions;
Figure 6 is a schematic representation of one implementation of the distributed processing system of Figure 5 used for determining network performance characteristics;
Figure 7 is a graphical representation of simulated network performance characteristics showing packet loss rates for different audio distribution strategies; and,
Figure 8 is a graphical representation similar to Figure 7 showing the different audio stream distribution characteristics for different audio distribution strategies.

[0029] An example of an IP communications network for implementing one arrangement of the invention is shown in Figure 1. In Figure 1 a plurality of user terminals 100 are connected to the Internet 102 via respective Internet access servers 106 which are each connected to an audio processor 104. The audio processors 104 are each capable of mixing a plurality of audio data streams received over the network. Although only two audio processors 104 are shown in the network of Figure 1 in practice any number of audio processors may be provided and distributed throughout the network for receiving, mixing or transmitting respective audio data streams. The audio data streams are typically transmitted using the Internet standard data stream transfer protocol Real Time Protocol (RTP) and underlying multicast transport protocols although unicast could be used also.

[0030] Referring to Figure 2, logically the network of Figure 1 can be considered to comprise a plurality of audio data stream sources 200, audio data stream sinks 202 and audio mixer 204 type components. Thus, each terminal 100 and audio processor 104 may comprise at least one source, sink or mixer component and the network of Figure 1 may comprise a plurality of these components linked together by audio data streams 206. In Figure 2 a simple logical network topology is shown in which the source and sink components are positioned at the terminal nodes of the network and the mixer components are positioned at the non-terminal nodes. In the network of Figure 2 each source 200 transmits one or more audio data streams to a respective parent mixer node 204. The mixer nodes may then forward the received audio data stream or streams directly to the other connected nodes, mix selected ones of the received audio data streams and then forward the respective mixed data streams, or perform a combination of these two actions. Each sink eventually receives each of the data streams either in the original un-mixed state or part of a new mixed stream. As will be described in greater detail later, each mixer component may additionally create multiple mixes from arbitrary subsets of the received data streams and transmit these mixes instead of or in addition to the audio data streams it would otherwise forward to other nodes in the network. For example, each mixer component may select to mix only some of its incoming audio data streams and forward the resultant partial mix on to other connected nodes.

[0031] In the arrangement of Figure 3 the audio data stream processors are integrated with selected IP multicast enabled routers 300 distributed in the network of Figure 1. Each audio processor 104 comprises an audio data stream receiver 302 for receiving audio data streams transmitted over the IP network, an audio mixer component 304 for mixing selected audio data streams and an audio data stream transmitter 306 for forwarding mixed and unmixed audio data streams to other audio processors 104 or terminals 100 in the IP network. The audio data stream receiver 302 and transmitter 306 comprises software components of the type implemented in audio applications such as Real Inc's Real Player or Microsoft Inc's Media Player, for example IP sockets, packet buffers, packet assemblers etc. The audio mixer component 306 is provided for mixing two or more audio data streams received at the processor 104. The audio mixer component is arranged to average selected audio data streams to create a single combined data stream comprising the same number of bits as each unmixed data stream.

[0032] Each audio processor is further provided with a congestion monitor 308 for monitoring congestion on the respective communication transmission links connecting the audio processor to other parts of the network. In one arrangement the congestion monitor 308 utilises Real Time Control Protocol (RTCP) control messages or the like received from respective audio stream receivers in the network to determine congestion levels on respective transmission links. These control messages provide the audio processors with information relating to data stream congestion on respective neighbouring transmission links and audio processors. Selected audio processors, typically those on the edge of the network connecting terminals 100 to the Internet, are also provided with a database 310 containing data relating to recipient terminal equipment characteristics including audio data processing and playback capabilities, network connection types and speeds, other recipient specific data including current tariff data for determining an appropriate quality of service to be provided and user specific audio data stream mixing profiles comprising user defined mixing preferences and other data relating to user specific mixing policies.

[0033] Mixing may occur at many of the audio processor mixer nodes in the network of Figure 1. When an audio processor mixes selected audio streams the number of audio data streams to be transmitted to neigh-

LINKPLAY EXHIBIT 1004
Page 1798 of 4822

bouring audio processor or terminal nodes is reduced and this also reduces the amount of processing required at the neighbouring nodes. Distributing the audio processor mixer nodes throughout the network as described enables the overall audio mixing task to be shared and mixing bottlenecks avoided. Distributed processing not only provides for scalability, say in terms of the maximum number of simultaneous speakers allowed in an audio conference, but also enables audio processors to monitor and respond to changing local network conditions in heterogeneous environments such as the Internet. Since mixing reduces the quality of the audio streams being mixed the audio processors are arranged to limit the amount of mixing they perform within the constraints of the available network resources so that end user recipients receive as many separate audio streams as possible within prevailing network resource constraints. Thus, the audio data processors are arranged to adapt to changing network conditions so that at one extreme peer-to-peer communication is possible, that is when available network resources are adequate for communicating all the received audio data streams to respective neighbouring nodes. At the other extreme, where available network resources are severely restricted, say due to network congestion, all audio data streams are mixed together to provide a single data stream for communication to the neighbouring nodes.

[0034] Mixing decisions are made dynamically by the audio data processors according to software program logic stored in audio processor memory and executed by the audio processors.

[0035] Referring now to the flowchart of Figure 4, network conditions such as available bandwidth, packet loss rate and delay are monitored by the congestion monitor 308 in step 400 to determine current values for respective network resource parameters associated with transmission links and neighbouring processor and terminal nodes located in the network. Current values for variable network resource parameters including but not limited to available bandwidth on respective transmission links and processing capacity and delay of respective processor nodes are stored in the data base 310 in step 402. Current values for other more stable network resource parameters including but not limited to terminal node processing capability and capacity and user specific mixing preferences and quality of service requirements are also stored in the processor databases 310. The more stable resource parameters may be monitored in step 400 in the same way as the variable resource parameters or by periodic polling the resources for current parameter values.

[0036] In step 404 the receiver 302 monitors relevant unicast or multicast communication channels for incoming audio data streams and determines in step 406 whether any streams are being received at the audio processor node. If audio data streams are being received the audio processor proceeds to step 408 where an appropriate algorithm determines the network resources necessary for transmitting the received data streams to the next relevant node or nodes in the network. If audio data streams are not being received monitoring continues and control passes back to step 400. In step 408 network resources necessary for forwarding all the received data streams to each respective node in the relevant network distribution tree are determined and in step 410 the respective available resources for transmitting the data streams are determined. In step 412 the current values of the relevant network resource parameters are compared with respective resource requirements necessary for communicating the received audio streams to the respective next network nodes comprising audio data stream receivers. If the available resources match those required for subsequent communication of the data streams all the data streams are transmitted onwards in step 420 to the relevant next nodes. However, if the audio processors determine in step 412 that there are insufficient resources selected mixing occurs. For instance, if there are only three communications channels available on a particular transmission link and four separate data streams are received, at least one pair of data streams are mixed prior to subsequent transmission by the respective audio processor. Similarly, if a recipient terminal node is only capable of processing two data streams simultaneously only two sets of mixed streams are provided for transmission to that terminal. Further, if a terminal node is designated as having a pre-determined quality of service as defined by a user selected tariff there may be insufficient bandwidth allocated for forwarding all the streams without mixing so that mixing will occur even if the network has sufficient bandwidth resources available on the relevant link or links to the user terminal node. In step 414 the difference between available and required resources is determined so that the number of streams to be mixed can be determined. The audio data stream processors select appropriate data streams to be mixed in step 416. The selected streams are mixed in step 418 and then transmitted by the appropriate audio processor transmitter 306 in step 420. Audio data stream selection in step 416 may be based on any number of considerations relevant to network, application or user requirements, for example.

[0037] The following discussion concerns mixing considerations that may be relevant in CSCW environments which support many simultaneous speakers.

[0038] One consideration for mixing may concern the roles of speakers. Some real-time CSCW applications assign participants to different roles within an event. For example, early experiments in inhabited television differentiated between performers, inhabitants and viewers. Performers are part of the core content of an on-line TV show whereas inhabitants are active within the virtual world, but typically receive a broadcast mix created by a director. These roles are complemented by differences in the technologies used to access the real-time event. Performers typically use professional studio-quality equipment, with fully spatialised 3D audio. Inhabitants may use commodity PCs, equipped with headphones.

LINKPLAY EXHIBIT 1004
Page 1799 of 4822

Viewers on the other hand may use conventional television sets, equipped with multiple loudspeaker surround sound audio systems. Roles or so called "layers of participation" can determine mixing policy. For instance, it may be appropriate to ensure that performers are heard with the maximum possible audio quality. Thus, as network congestion increases the audio streams for inhabitants might be mixed together first, with the performers streams being kept separate for as long as possible. Other CSCW applications may also benefit from defining layers of participation and using these to prioritise audio sources for mixing.

[0039] Another consideration for mixing may concern the roles of listeners or recipients. Mixing can also be prioritised according to listener requirements. Roles or layers of participation can also define the different ways in which listeners take part in an event, although many participants will be both speakers and listeners in an event. For example, an active inhabitant may benefit from fully spatialised audio that provides clues to support navigation and conversation management. A passive viewer with a surround-sound audio system may benefit from a mix that clearly separates the key performers, but where their accurate location in the world is less important. In the case of inhabitants it may be important to maintain the separation of streams from nearby participants, whereas for viewers it may be appropriate to maintain the separation of key performers only.

[0040] A further mixing consideration may concern the grouping of audio sources. CSCW applications often group participants in some pre-determined way. It is often appropriate to mix audio streams from one coherent group. For example, avatars in a CVE may have gathered together to form definable and separate groups. Audio streams from each group could be mixed to form a single stream that could be spatialised to the average position of the group as a whole in the CVE. In addition, some CSCW applications calculate levels of mutual awareness among participants which may provide a more dynamic basis for mixing respective audio streams.

[0041] Another mixing consideration may concern voice characteristics. The timbre of voices or other audio sources may be useful for determining which streams to mix. For example, it may be appropriate to mix a high and a low pitch voice into a single stream so that a listener can readily separate them when hearing the mixed stream.

[0042] Patterns of activity within a multiple speaker environment may also determine mixing decisions. For example, audio data streams from participants whose speech rarely overlaps could be mixed together

[0043] A number of more practical concerns may also shape affect mixing decisions. Mixing may depend upon aspects of the available communication network, including its topology (i.e. shape and structure), underlying bandwidth, regional variations, or transitory congestion. Mixing decisions may also depend on the available computing resources, for example the number and capability of available mixer components, how many are positioned within the network, and how heavily loaded they are. Mixing decisions may also consider the current and past states of the system. For example, the transition from one choice of mixed streams to another may be noticeable to users, and potentially undesirable.

[0044] It is clear from the above discussion that the process of selecting data streams for optimal mixing is a complex task and will very often be application specific. For instance, different applications may have different mixing requirements. In particular applications may have different ways of assigning priorities to audio streams to determine the order in which streams are gradually mixed together as network resources become scarce. Low priority streams will be mixed before higher priority streams. These requirements may also vary between different phases of the same application as a session progresses. In particular, priorities may change as participants take on different roles or move to different locations. For example, a virtual football game with the crowd (audience) will not have the same mixing requirements as a virtual shopping mall, a virtual education application (on-line demonstration, lecture etc), or an on-line television drama in a virtual world.

[0045] In a virtual football game for example, audio priority will be higher for the referee say, than the other players. In this respect, the system will avoid mixing the referee's audio stream with the respective players audio streams unless this becomes absolutely necessary due to network resource limitations. On the other hand audio streams from a crowd of spectators will have a lower priority since each spectator will not need to be heard individually. It may be sufficient to mix all the streams from the same "stand" or group of collocated spectators and spatialise only the resulting stream at the receiver. In this way mixing is based on the roles of the speakers, that is to say, audio streams from more important speakers are forwarded and less important speakers are mixed

[0046] In a virtual shopping mall, mixing policy may be based on the collocation and mutual awareness of the participants. For example, participants may want to receive separate audio streams from other collocated participants or other more important participants of whom they are more aware of than others. The remainder of the of the group can be mixed together or divided into smaller groups which are mixed separately, similar to the different stands in the football game example.

[0047] In a virtual lecture environment there is likely to be one lecturer, may be a few demonstrators and many mutually aware students. In this environment mixing can be based on speaker roles. For instance, the lecturer's and demonstrators' audio streams may be forwarded so that the other participants can process the individual streams on receipt, where the respective students audio streams will be mixed. The present inventors have implemented and tested the invention in a distributed processing system simulator 500 shown in Figure 5. In the distributed processing system two end user systems

LINKPLAY EXHIBIT 1004
Page 1800 of 4822

502a and 502b are shown on a first local area network 506 and two end systems 502c and 502d on a second local area network 512. The system 500 may comprise any number of end user systems 502 depending on the networks being simulated. Each end user system comprises a respective virtual world client 514 for accessing a shared virtual world generated by a virtual world server 516 on LAN 512. Each end user system is also provided with a local audio server 518 that is interfaced to respective audio hardware (not shown) so that users can speak to each other within the virtual world environment. Each client 514 controls the local audio server 518 for the respective end system and uses information in the virtual world to determine how the audio server should transmit, receive and process audio streams, for example according to the positions of other users in the virtual world.

[0048] For peer-to-peer audio, each user's audio server 518 sends an audio data stream directly to all the other audio servers in the system using underlying unicast or multicast protocols. An audio processor 104 is provided on each LAN for mixing selected data streams received from the connected audio servers. The audio processors are both controlled directly by the virtual world server 516 and are connected together by means of a WAN simulator 520. The local audio servers 518 and audio processors 104 together define an audio distribution tree as shown by the dashed lines 522 in Figure 5. Each audio processor is capable of receiving audio data streams from the audio servers on its respective LAN and transmitting these streams to the remote audio processor on the other LAN. The audio processors are arranged to adapt to changing network conditions so that instead of three separate audio streams being forwarded from say end user systems 502a, 502b and 502c to 502d the audio processors 104 can mix respective data streams so that end user systems 502c and 502d receive a single stream comprising a mixture of the streams from end user systems 502a and 502b and a separate audio stream from each other, for example.

[0049] The inventors evaluated the effectiveness of dynamic mixing by investigating the effect of network congestion on audio quality. Two quantifiable aspects of audio quality were considered, the level of packet loss experienced and the degree of audio stream spatialisation, that is the number of separate audio streams delivered to an end system. These two criteria were chosen since they both relate to the end user's perceived experience of the system and they can also be readily determined from measurements of the system 500, for example the number of packets being sent per second. The first measure, the level of packet loss experienced, is the primary determinant of whether a network audio stream will be intelligible to the user and therefore of any use at all. Audio codecs that encode audio streams in 40ms to 80ms packets and utilise silence substitution for packet loss recovery typically become unintelligible if 15% or more of the packets are lost during transmission. Other factors such as delay or jitter are of secondary importance by comparison. The second measure, the degree of spatialisation, is the primary distinguishing feature between the peer-to-peer, fully mixed and partially mixed approaches. For instance, research has shown that spatialised audio is a key factor in providing users with a sense of presence in a virtual environment.

[0050] Referring to Figure 6 which shows a system 600 configured from the system of Figure 5 for evaluating the effect of network congestion on audio quality. In Figure 6 the LANs 506 and 512 are assumed to be generally congestion free high bandwidth networks connected via a lower bandwidth shared WAN 520 which is prone to congestion. Six end

[0051] user systems 502 are provided on LAN 506 for simulating network usage. A single end user system is provided on LAN 512. A WAN simulation tool is provided for simulating network delays and bandwidth restrictions for a limited bandwidth WAN connection. An additional application 602 is provided for introducing controlled levels of competing traffic onto the simulated WAN connection 520 in order to create network congestion.

[0052] All packets on the system 600 are monitored and analysed to classify the packets and to measure the number of audio streams in transit and the amount of competing traffic. Packet loss experienced by the audio streams in transit is measured by matching the number of packets leaving LAN 506 to those arriving on LAN 512.

[0053] The following strategy was used to evaluate the system of Figure 6:-

- Six simulated users on LAN 506 continuously sending audio data to a single user on LAN 512.
- Virtual WAN bandwidth limit of 500 Kbit/second corresponding to just over seven audio streams for 8KHz, 8bit, Ulaw, mono audio encoding. A WAN buffer size of 250 Kbits.
- Eight levels of competing (congestion-inducing) traffic: 0, 70, 140, 210, 280, 350, 420, and 490 Kbits/s.
- Three audio distribution strategies: forward all audio streams without mixing (equivalent to peer-to-peer multicast), mix all audio streams before forwarding (equivalent to total mixing at LAN 1), and mix a dynamic subset of audio streams (parial mixing).

[0054] In the final distribution strategy dynamic or partial mixing was used to keep the packet loss rate below 15% whilst maintaining the maximum number of separate audio streams.

[0055] The experimental results are shown in Figures 7 and 8. Figure 7 shows the effect that increasing levels of congestion has on the packet loss rate experienced for each of the three audio distribution strategies. The peer-to-peer approach (line 700 in the graph of Figure 7) experiences increasing levels of packet loss as competing traffic increases. The packet loss rate exceeds 15%

LINKPLAY EXHIBIT 1004
Page 1801 of 4822

at 210 Kbits/s of additional traffic. Full-mixing (line 702 in the drawing) uses the minimum bandwidth throughout, and only starts to experience congestion when the competing traffic reaches 490 Kbits/s. Distributed partial mixing (line 704 in the drawing) gives higher loss rates than full mixing, but much lower rates than all-forwarding peer-to-peer, and maintains its loss rate below 15% even with 490 Kbit/s of competing traffic (as for full mixing).

[0056] Figure 8 shows the number of separate audio streams being transmitted to a listener on the end user system on LAN 512. For all-forwarding (line 800 in the graph of Figure 8), six streams are always transmitted by the LAN 506, however none of these arrive in any useful form when competing traffic levels exceed 210 Kbits/s. For total mixing (line 802 in Figure 8), one stream is always sent. Dynamic mixing (line 804 in Figure 8) lies between these two extremes. With no congestion, six distinct steams are transmitted over the WAN connection from the WAN 506 to the WAN 512. As competing traffic, and hence congestion increases, dynamic mixing reduces the number of distinct streams by mixing more audio streams together. When competing traffic levels reach 490 Kbits/s dynamic mixing falls back to total mixing, with only a single stream sent over the WAN.

[0057] The above described investigation demonstrates that dynamic or distributed partial mixing combines the benefits of both peer-to-peer and total mixing audio services. With sufficient bandwidth, the system operates like a peer-to-peer system, delivering independent audio streams to each listener, giving maximum individual flexibility and control over what users hear. As bandwidth becomes restricted the distributed partial mixing scheme moves incrementally towards a totally mixed (minimum bandwidth) service, thereby preserving a useful level of audio communication under a wide range of network conditions.

[0058] More generally, the distributed partial mixing approach has the following distinctive benefits: it is adaptive, reacting to network congestion in a way that peer-to-peer systems cannot; it supports dynamic load balancing between different distributed components of the audio service; it readily supports heterogeneous networks and different end user terminal capabilities; and it is adapative to varying application requirements.

Claims

1. A method of processing audio data streams transmitted in a communications network; said method comprising the steps of:-

   i) receiving a plurality of audio data streams transmitted from one or more audio data stream transmitters distributed in the network (404); and
   characterized by
   ii) processing data relating to at least one respective network resource parameter to determine respective network resources available for subsequent communication of said audio data streams to at least one respective audio data stream receiver in the network (408); and,
   iii) comparing said available resources with respective network resource requirements necessary for communicating said audio streams to at least one respective audio data stream receiver in the network (410, 412); and,
   iv) determining whether to mix selected audio data streams prior to transmission in response to said comparison (418, 420).

2. A method according to claim 1 further comprising the step of:-

   v) processing two or more audio streams in response to said comparison to provide at least one mixed audio data stream for subsequent transmission in said network.

3. A method according to claim 1 or claim 2 wherein steps ii) and iii) comprise the steps of:-

   determining a current value for the or each respective network resource parameter, and
   comparing the or each respective current resource parameter value with a respective minimum resource threshold value necessary for communicating said unmixed audio data streams to the or each respective receiver.

4. A method according to claim 3 wherein said minimum resource threshold value is determined according to at least one pre-defined quality of service parameter.

5. A method according to claim 4 wherein said network is a packet switched network and said pre-defined quality of service parameter is defined by a maximum packet loss rate.

6. A method according to any preceding claim wherein one network resource parameter relates to available network bandwidth for transmission of said audio data streams to the or each respective receiver.

7. A method according to claim 6 wherein said available bandwidth capacity is determined by user specific quality of service requirements.

8. A method according to any preceding claim wherein one network resource parameter relates to receiver processing characteristics.

9. A method according to any preceding claim wherein said audio data streams are selected for mixing according to predetermined criteria.

LINKPLAY EXHIBIT 1004
Page 1802 of 4822

10. A method according to claim 9 wherein said audio streams are mixed according to respective audio stream content.

11. A method according to claim 9 wherein said audio streams are mixed according to respective recipient requirements.

12. A method according to claim 9 wherein said audio streams are mixed according to respective audio stream sources.

13. A method according to claim 9 wherein said audio streams are mixed according to respective receiver audio data stream processing capabilities.

14. A software program component for processing audio data streams transmitted in a communications network; said program being arranged to execute the following steps when run on a computer:

i) receive a plurality of audio data streams transmitted from one or more audio data stream transmitters distributed in the network; and **characterized by** :
ii) process data relating to at least one respective network resource parameter to determine respective network resources available for subsequent transmission of said audio data streams in the network; and,
iii) compare said available resources with respective network resource requirements necessary for transmission of said audio streams to at least one respective audio data stream receiver in the network; and,
iv) determine whether to mix selected audio data streams prior to transmission in response to said comparison.

15. A system for processing audio data streams transmitted in a communications network; said system comprising:-

i) a receiver (301) for receiving a plurality of audio data streams transmitted from one or more audio data stream transmitters distributed in the network; and, **characterized by**
ii) a processor (104) for processing data relating to at least one respective network resource parameter to determine respective network resources available for subsequent transmission of said audio data streams in the network; and,
iii) a comparator for comparing said available resources with respective network resource requirements necessary for transmission of said audio streams to at least one respective audio data stream receiver in the network; and,
iv) determining means for determining whether to mix selected audio data streams prior to transmission in response to said comparison.

Patentansprüche

1. Verfahren zur Verarbeitung von Audiodatenströmen, die in einem Kommunikationsnetzwerk übertragen werden, wobei das Verfahren die folgenden Schritte umfasst:

i) Empfangen von mehreren Audiodatenströmen (404), die von einem oder von mehreren in dem Netzwerk verteilten Audiodatenstromsendern gesendet werden, und
gekennzeichnet ist durch
ii) Verarbeiten von Daten (408), die sich auf zumindest einen jeweiligen Netzwerkressourcenparameter zum Ermitteln entsprechender Netzwerkressourcen beziehen, die für eine nachfolgende Übertragung der Audiodatenströme an zumindest einen jeweiligen Audiodatenstromempfänger des Netzwerks verfügbar sind, und
iii) Vergleichen der verfügbaren Ressourcen mit dem jeweiligen Netzwerkressourcenbedarf (410, 412), der für die Übertragung der Audiodatenströme an zumindest einen jeweiligen Audiodatenstromempfänger des Netzwerks erforderlich ist, und
iv) Bestimmen, ob ausgewählte Audiodatenströme vor der Übertragung als Reaktion auf den Vergleich zu mischen sind (418, 420).

2. Verfahren nach Anspruch 1, das ferner den Schritt umfasst zum

v) Verarbeiten von zwei oder mehr Audiodatenströmen als Reaktion auf den Vergleich, um für die nachfolgende Übertragung in dem Netzwerk zumindest einen gemischten Audiodatenstrom herzustellen.

3. Verfahren nach Anspruch 1 oder 2, worin die Schritte ii) und iii) die Schritte zum
Bestimmen eines aktuellen Werts für den oder für jeden der jeweiligen Netzwerkressourcenparameter und zum
Vergleichen des oder eines jeden der jeweiligen aktuellen Netzwerkressourcenparameterwerte mit einem entsprechenden Minimalressourcenschwellwert umfassen, der zur Übertragung der nicht gemischten Audiodatenströme an den oder jeden der jeweiligen Empfänger erforderlich ist.

4. Verfahren nach Anspruch 3, worin der Minimalressourcenschwellwert zumindest einem vorgegebenen Dienstgüteparameter entsprechend festgelegt wird.

LINKPLAY EXHIBIT 1004
Page 1803 of 4822

5. Verfahren nach Anspruch 4, worin das Netzwerk als Datenpaketvermittlungsnetzwerk ausgebildet ist und der vorgegebene Dienstgüteparameter durch eine maximale Paketverlustrate bestimmt ist.

6. Verfahren nach einem der vorhergehenden Ansprüche, worin ein Netzwerkressourcenparameter auf die verfügbare Netzwerkbandbreite zur Übertragung der Audiodatenströme an den oder jeden der jeweiligen Empfänger bezogen ist.

7. Verfahren nach Anspruch 6, worin die verfügbare Bandbreitenkapazität über benutzerspezifische Dienstgütevorgaben bestimmt wird.

8. Verfahren nach einem der vorhergehenden Ansprüche, worin ein Netzwerkressourcenparameter auf eine Empfängerverarbeitungseigenschaft bezogen ist.

9. Verfahren nach einem der vorhergehenden Ansprüche, worin die Audiodatenströme nach einem vorgegebenen Kriterium zum Mischen ausgewählt werden.

10. Verfahren nach Anspruch 9, worin die Audiodatenströme nach dem jeweiligen Inhalt der Audiodatenströme gemischt werden.

11. Verfahren nach Anspruch 9, worin die Audiodatenströme nach den jeweiligen Empfängervoraussetzungen gemischt werden.

12. Verfahren nach Anspruch 9, worin die Audiodatenströme entsprechend den jeweiligen Audiodatenstromquellen gemischt werden.

13. Verfahren nach Anspruch 9, worin die Audiodatenströme entsprechend dem jeweiligen Vermögen der Empfänger zur Verarbeitung von Audiodatenströmen gemischt werden.

14. Datenverarbeitungsprogrammkomponente zum Verarbeiten von Audiodatenströmen, die in einem Kommunikationsnetzwerk übertragen werden, wobei das Programm dazu ausgebildet ist, bei einer Ausführung auf einer Datenverarbeitungsanlage die folgenden Schritte auszuführen:

i) Empfangen von mehreren Audiodatenströmen, die von einem oder von mehreren in dem Netzwerk verteilten Audiodatenstromsendern gesendet werden, und
gekennzeichnet ist durch
ii) Verarbeiten von Daten, die sich auf zumindest einen jeweiligen Netzwerkressourcenparameter zum Ermitteln entsprechender Netzwerkressourcen beziehen, die für eine nachfolgende Übertragung der Audiodatenströme in dem Netzwerks verfügbar sind, und
iii) Vergleichen der verfügbaren Ressourcen mit dem jeweiligen Netzwerkressourcenbedarf, der für die Übertragung der Audiodatenströme an zumindest einen jeweiligen Audiodatenstromempfänger des Netzwerks erforderlich ist, und
iv) Bestimmen, ob ausgewählte Audiodatenströme vor der Übertragung als Reaktion auf den Vergleich zu mischen sind.

15. Vorrichtung zur Verarbeitung von Audiodatenströmen, die in einem Kommunikationsnetzwerk übertragen werden, wobei die Vorrichtung umfasst:

i) einen Empfänger (301) zum Empfangen von mehreren Audiodatenströmen, die von einer oder von mehreren in dem Netzwerk verteilten Audiodatenstromsendern gesendet werden, und
gekennzeichnet ist durch
ii) einen Prozessor (104) zum Verarbeiten von Daten, die sich auf zumindest einen jeweiligen Netzwerkressourcenparameter zum Ermitteln entsprechender Netzwerkressourcen beziehen, die für eine nachfolgende Übertragung der Audiodatenströme in dem Netzwerks verfügbar sind, und
iii) einen Komparator zum Vergleichen der verfügbaren Ressourcen mit dem jeweiligen Netzwerkressourcenbedarf, der für die Übertragung der Audiodatenströme an zumindest einen jeweiligen Audiodatenstromempfänger des Netzwerks erforderlich ist, und
iv) eine Bestimmungseinrichtung zum Bestimmen, ob ausgewählte Audiodatenströme vor der Übertragung als Reaktion auf den Vergleich zu mischen sind.

Revendications

1. Procédé de traitement de trains de données audio transmis dans un réseau de communication, ledit procédé comprenant les étapes suivantes :

i) la réception d'une pluralité de trains de données audio émis depuis un ou plusieurs émetteurs de trains de données audio répartis dans le réseau (404), et **caractérisé par** :
ii) le traitement de données se rapportant à au moins un paramètre de ressource de réseau respective pour déterminer des ressources de réseau respectives disponibles pour une communication ultérieure desdits trains de données audio à au moins un récepteur de trains de données audio respectif dans le réseau (408), et
iii) la comparaison desdites ressources disponi-

LINKPLAY EXHIBIT 1004
Page 1804 of 4822

bles aux exigences de ressources de réseau respectives nécessaires pour communiquer lesdits trains de données audio à au moins un récepteur de trains de données audio respectif dans le réseau (410, 412), et
iv) la détermination du mélange ou non de trains de données audio sélectionnés avant la transmission, en réponse à ladite comparaison (418, 420).

2. Procédé selon la revendication 1, comprenant en outre l'étape suivante :

v) le traitement de deux trains de données audio ou plus en réponse à ladite comparaison pour procurer au moins un train de données audio mélangé en vue d'une transmission ultérieure dans ledit réseau.

3. Procédé selon la revendication 1 ou la revendication 2, dans lequel les étapes ii) et iii) comprennent les étapes consistant à :

déterminer une valeur en cours pour le ou chaque paramètre de ressource de réseau respectif, et
comparer la ou chaque valeur de paramètre de ressource en cours respectif à une valeur de seuil de ressource minimale respective nécessaire pour communiquer lesdits trains de données audio non mélangés au récepteur ou à chaque récepteur respectif.

4. Procédé selon la revendication 3, dans lequel ladite valeur de seuil de ressource minimale est déterminée conformément à au moins un paramètre de qualité de service prédéfini.

5. Procédé selon la revendication 4, dans lequel ledit réseau est un réseau commuté par paquets et ledit paramètre de qualité de service prédéfini est défini par un taux de perte de paquets maximal.

6. Procédé selon l'une quelconque des revendications précédentes, dans lequel un paramètre de ressource de réseau se rapporte à une bande passante de réseau disponible pour la transmission desdits trains de données audio au récepteur ou à chaque récepteur respectif.

7. Procédé selon la revendication 6, dans lequel ladite capacité de bande passante disponible est déterminée par les exigences de qualité de service spécifiques à un utilisateur.

8. Procédé selon l'une quelconque des revendications précédentes, dans lequel un paramètre de ressource de réseau se rapporte à des caractéristiques de traitement de récepteur.

9. Procédé selon l'une quelconque des revendications précédentes, dans lequel lesdits trains de données audio sont sélectionnés en vue d'un mélange conformément à des critères prédéterminés.

10. Procédé selon la revendication 9, dans lequel lesdits trains de données audio sont mélangés conformément à des contenus des trains de données audio respectifs.

11. Procédé selon la revendication 9, dans lequel lesdits trains de données audio sont mélangés conformément à des exigences de destinataires respectives.

12. Procédé selon la revendication 9, dans lequel lesdits trains de données audio sont mélangés conformément à des sources de trains de données audio respectives.

13. Procédé selon la revendication 9, dans lequel lesdits trains de données audio sont mélangés conformément à des capacités de traitement de trains de données audio de récepteurs respectifs.

14. Composant de programme logiciel destiné à traiter des trains de données audio transmis dans un réseau de communication, ledit programme étant conçu pour exécuter les étapes suivantes quand il est exécuté sur un ordinateur :

i) la réception d'une pluralité de trains de données audio émis depuis un ou plusieurs émetteurs de trains de données audio répartis dans le réseau, et **caractérisé par** :
ii) le traitement de données se rapportant à au moins un paramètre de ressource de réseau respectif pour déterminer des ressources de réseau respectives disponibles en vue d'une transmission ultérieure desdits trains de données audio dans le réseau, et
iii) la comparaison desdites ressources disponibles à des exigences de ressources de réseau respectives nécessaires à la transmission desdits trains de données audio à au moins un récepteur de trains de données audio respectif dans le réseau, et
iv) la détermination du mélange ou non de trains de données audio sélectionnés avant la transmission, en réponse à ladite comparaison.

15. Système de traitement de trains de données audio transmis dans un réseau de communication, ledit système comprenant :

i) un récepteur (302) destiné à recevoir une pluralité de trains de données audio émis depuis

LINKPLAY EXHIBIT 1004
Page 1805 of 4822

un ou plusieurs émetteurs de trains de données audio répartis dans le réseau, et **caractérisé par** :

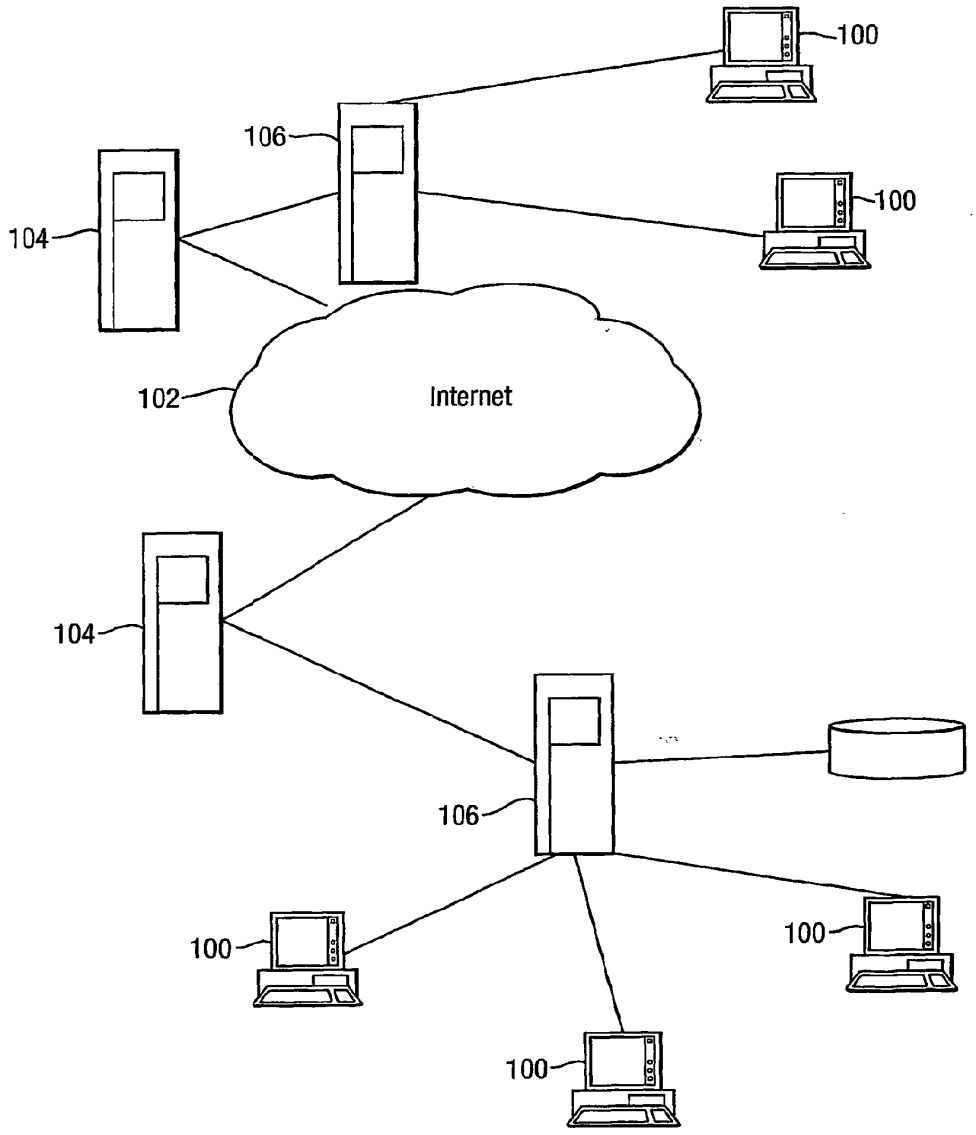
ii) un processeur (104) destiné à traiter des données se rapportant à au moins un paramètre de ressource de réseau respectif pour déterminer des ressources de réseau respectives disponibles en vue d'une transmission ultérieure desdits trains de données audio dans le réseau, et

iii) un comparateur destiné à comparer lesdites ressources disponibles auxdites exigences de ressources de réseau respectives nécessaires pour la transmission desdits trains de données audio à au moins un récepteur de trains de données audio respectif dans le réseau, et

iv) un moyen de détermination destiné à déterminer s'il faut mélanger des trains de données audio sélectionnés avant la transmission, en réponse à ladite comparaison.

LINKPLAY EXHIBIT 1004
Page 1806 of 4822

Fig.1.



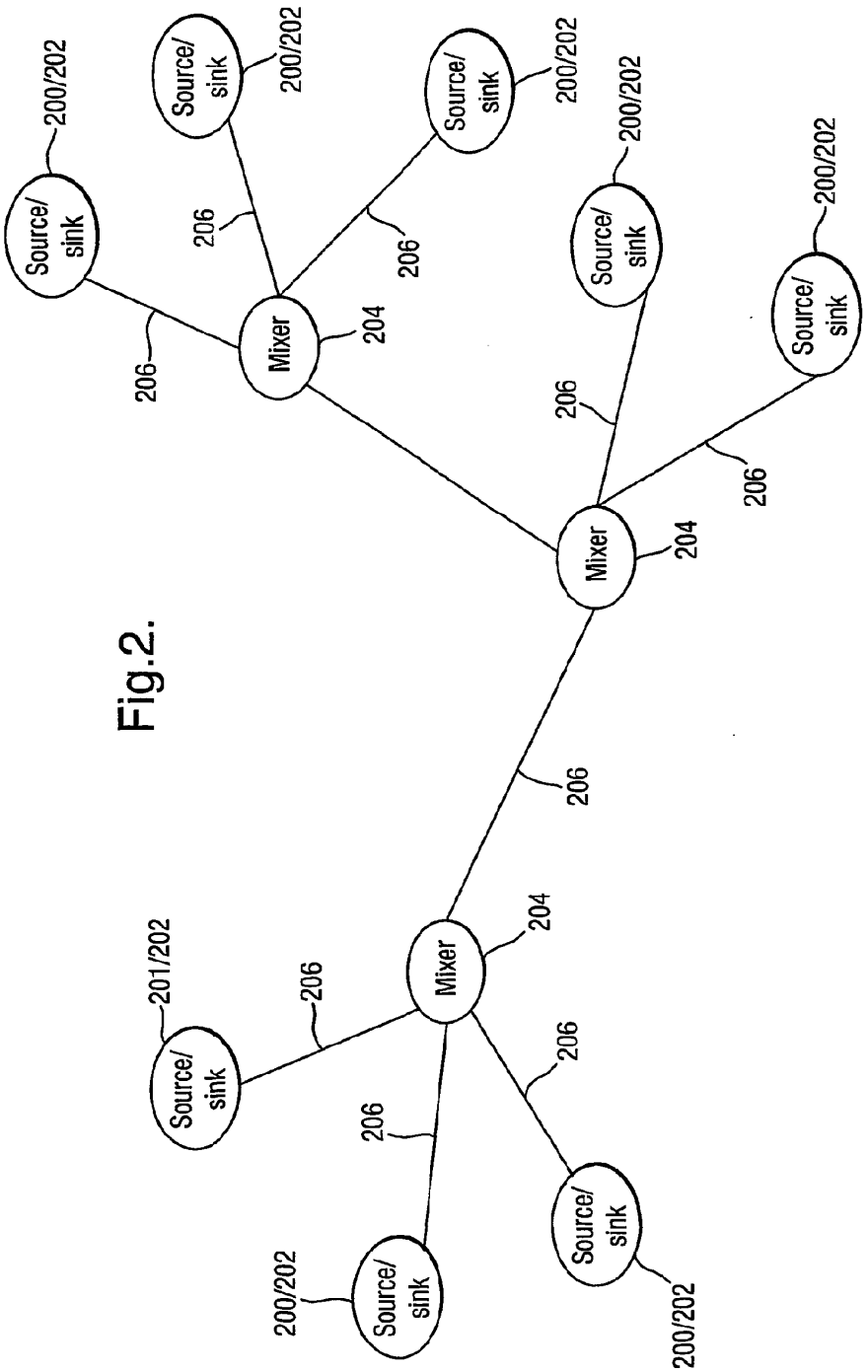
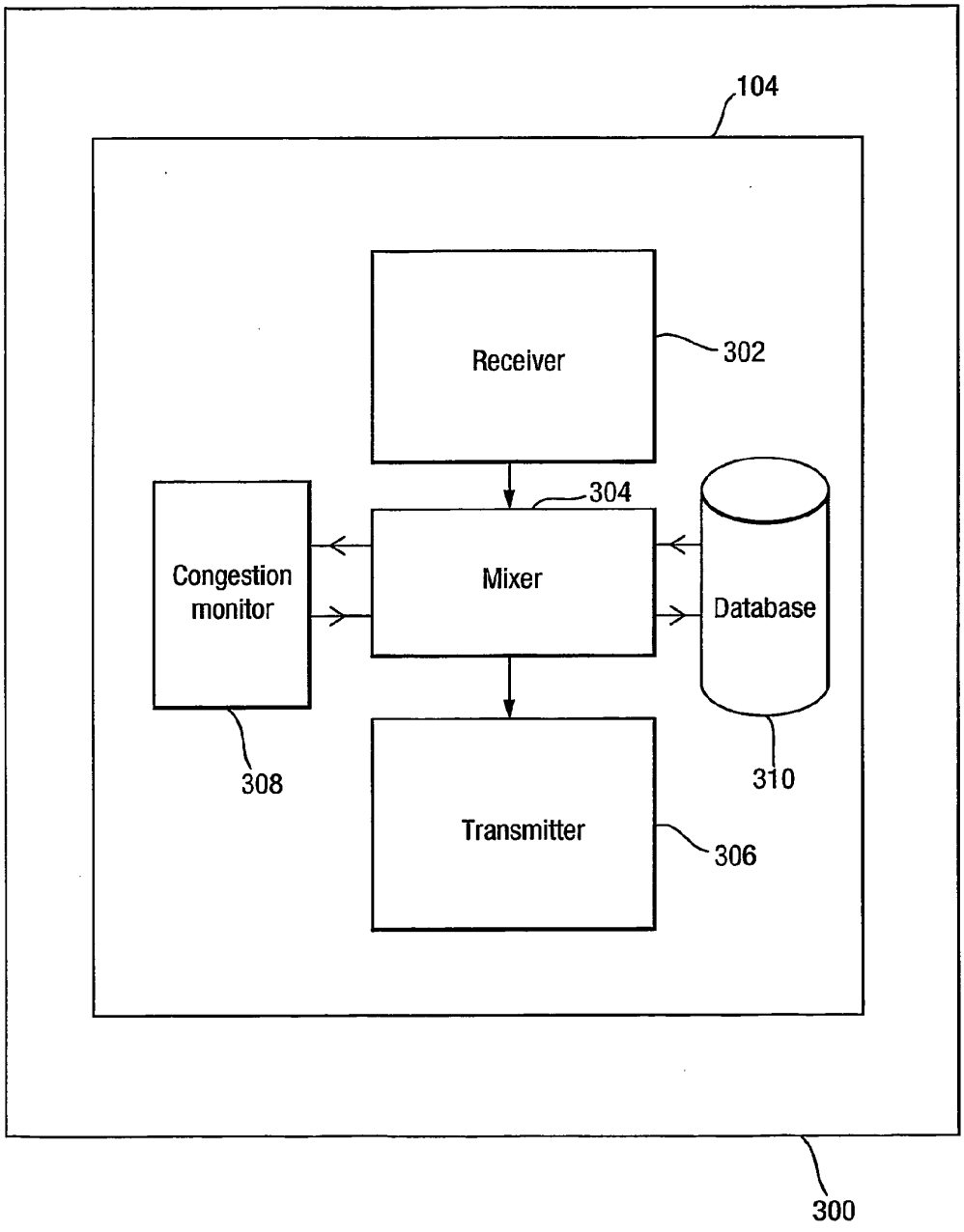


Fig.2.

Fig.3.


104
Receiver
302
304
Congestion monitor
Mixer
Database
308
310
Transmitter
306
300

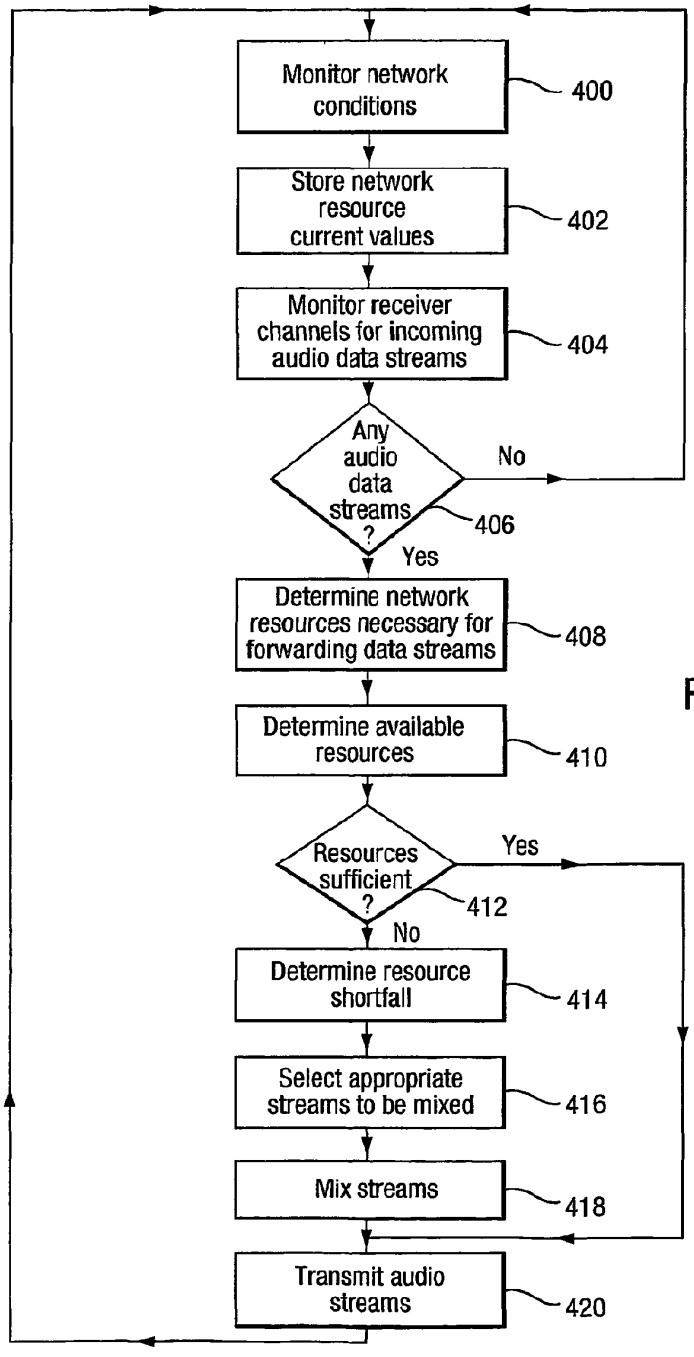

Monitor network conditions
400
Store network resource current values
402
Monitor receiver channels for incoming audio data streams
404
Any audio data streams ?
406
No
Yes
Determine network resources necessary for forwarding data streams
408
Determine available resources
410
Resources sufficient ?
412
Yes
No
Determine resource shortfall
414
Select appropriate streams to be mixed
416
Mix streams
418
Transmit audio streams
420

Fig.4.

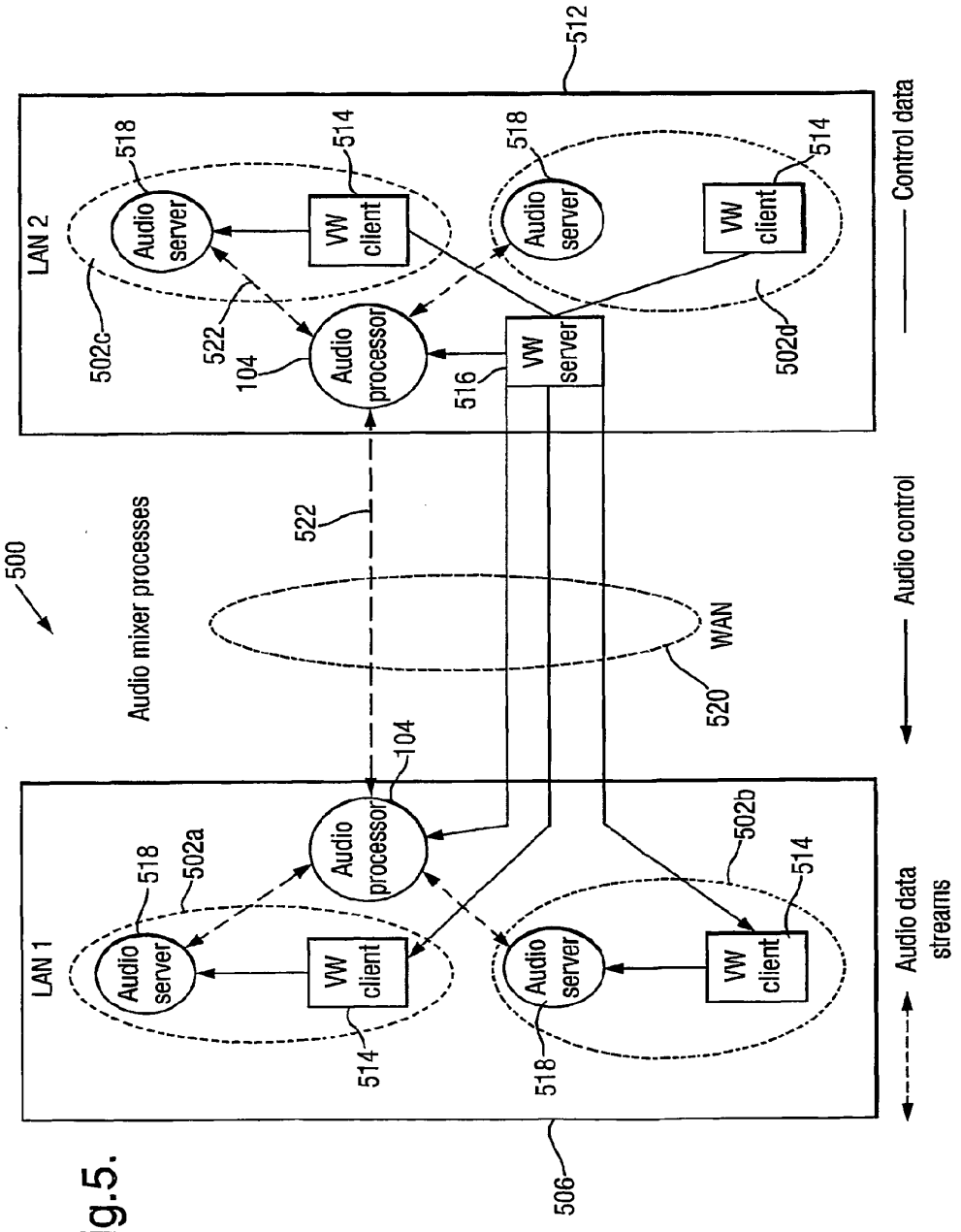


Fig.5.

LINKPLAY EXHIBIT 1004
Page 1811 of 4822

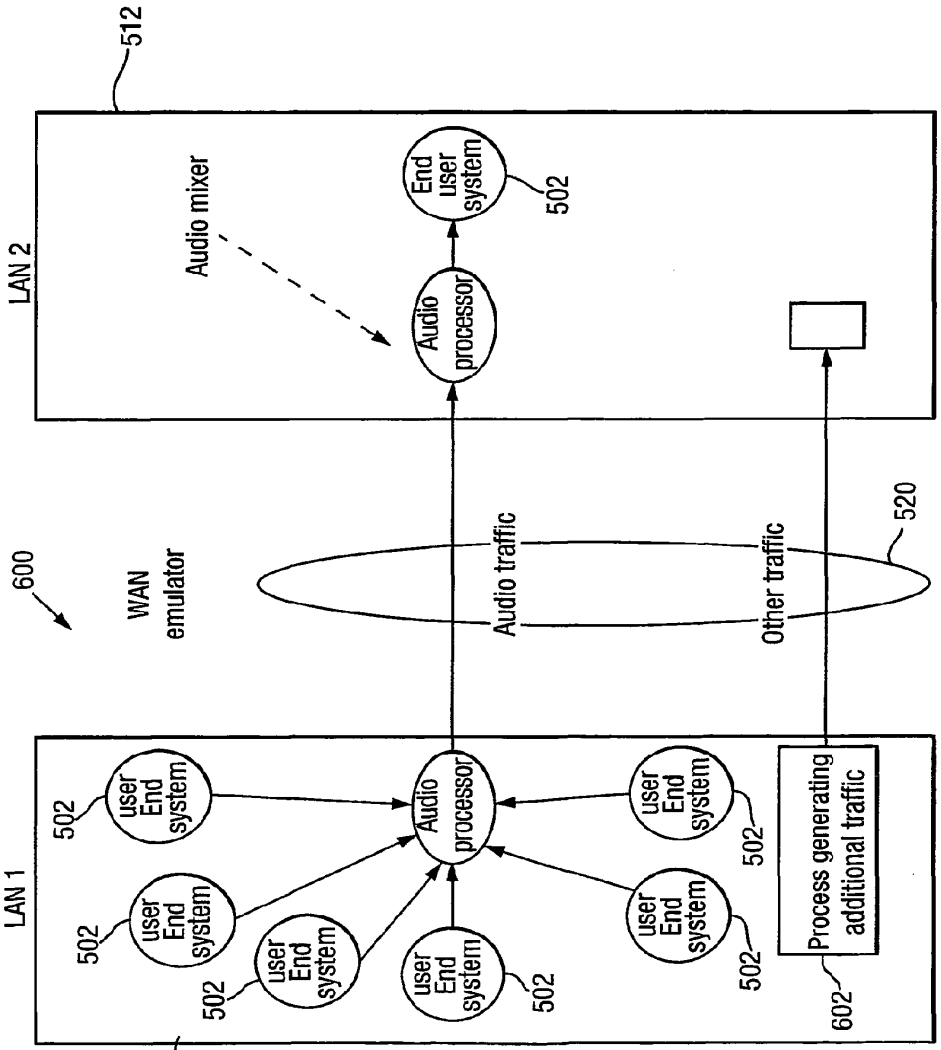
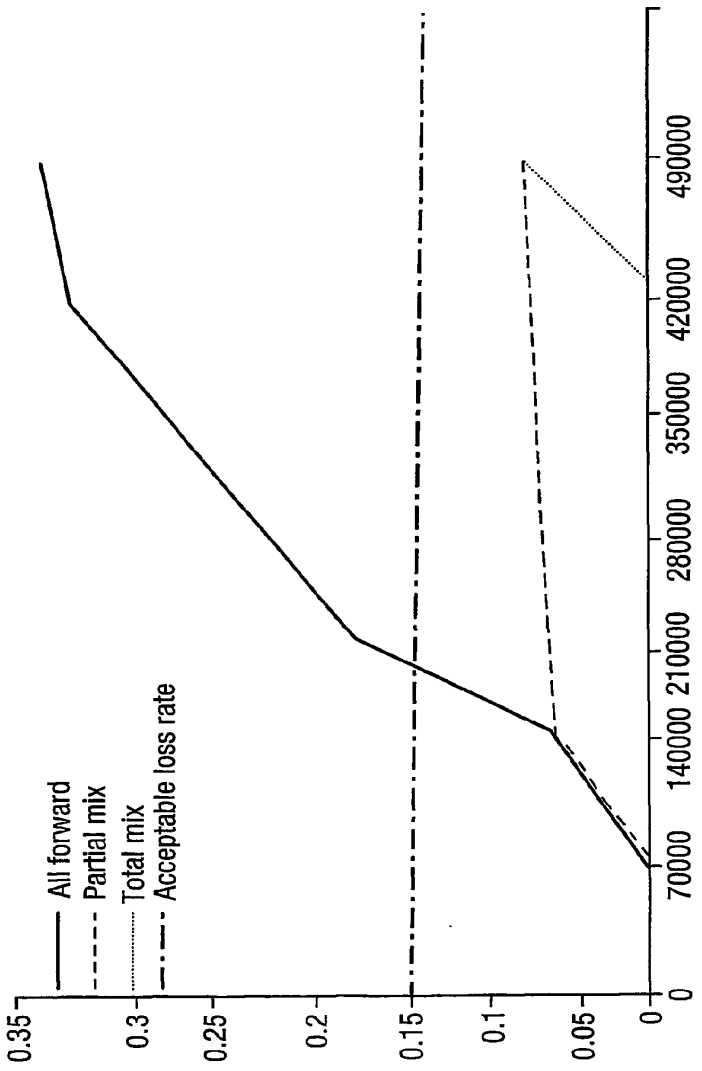

512
LAN 2
Audio mixer
End user system
502
Audio processor
600
WAN emulator
Audio traffic
Other traffic
520
LAN 1
user End system
502
Audio processor
Process generating additional traffic
602
506

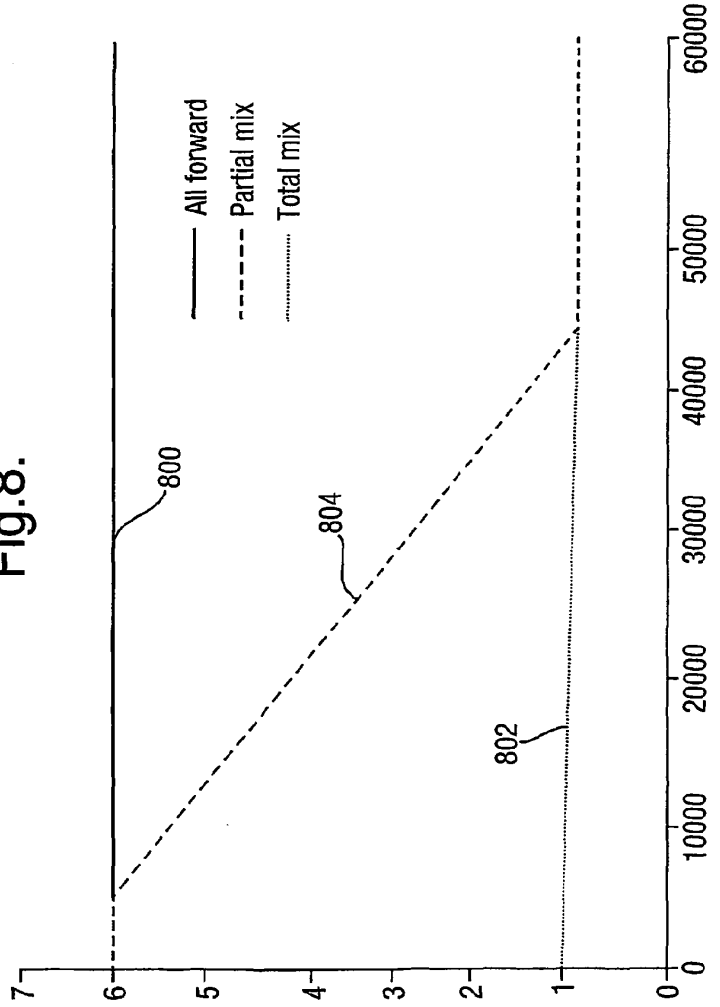
Fig.6.

Fig.7.



LINKPLAY EXHIBIT 1004
Page 1813 of 4822

Fig.8.


All forward
Partial mix
Total mix
800
804
802
0
10000
20000
30000
40000
50000
60000
0
1
2
3
4
5
6
7

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- EP 1024638 A **[0011]**

Non-patent literature cited in the description

- Diamond Park and Spline: A Social Virtual Reality System With 3D Animation, Spoken Interaction, and Runtime Extendibility. *PRESENCE,* 1997, vol. 6 (4), 461-481 **[0011]**

LINKPLAY EXHIBIT 1004
Page 1815 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 389 853 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
18.02.2004 Bulletin 2004/08

(51) Int Cl.[7]: **H04L 12/28**

(21) Application number: **02018385.1**

(22) Date of filing: **14.08.2002**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **Sony International (Europe) GmbH**
10785 Berlin (DE)

(72) Inventors:
- **Stadelmeier, Lothar,**
  c/o Sony International(Europe
  70327 Stuttgart (DE)
- **Dawidowsky, Frank,**
  c/o Sony International (Europe)
  70327 Stuttgart (DE)
- **Hoefflinger, Jens,**
  c/o Sony International (Europe)
  70327 Stuttgart (DE)

(74) Representative: **Müller - Hoffmann & Partner Patentanwälte,**
Innere Wiener Strasse 17
81667 München (DE)

Remarks:
Amended claims in accordance with Rule 86 (2) EPC.

(54) Bandwidth oriented reconfiguration of wireless ad hoc networks

(57) A first wireless ad hoc network with a certain available amount of bandwidth comprises a plurality of wireless terminals communicating via wireless connections with each other. Each wireless connection requires a certain amount of the available amount of bandwidth. A problem occurs, if a wireless terminal wants to set up a new wireless connection requiring bandwidth or requires more bandwidth for an existing wireless connection and the required amount of bandwidth exceeds the left amount of the available amount of bandwidth of the first wireless ad hoc network. In a state of the art wireless ad hoc network the requested bandwidth cannot be provided and a user application may not be possible. According to the invention, however, the required amount of bandwidth can be provided by splitting the first wireless ad hoc network such that a second wireless ad hoc network is spawned which operates in a different communication channel and thus provides additional bandwidth. The second wireless ad hoc network comprises wireless terminals of the first wireless ad hoc network and/or new wireless terminals.

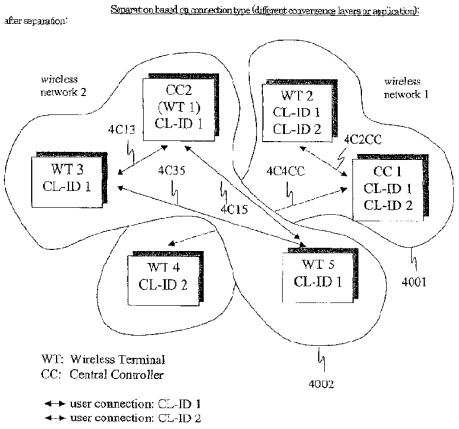


Figure 4B

EP 1 389 853 A1

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The invention is related to wireless ad hoc networks which comprise a central controller and a set of wireless terminals (also referred to as wireless devices). In particular, the invention is related to a method and means for providing additional bandwidth for a wireless ad hoc network if the available bandwidth is not sufficient.

[0002] A wireless ad hoc network is a collection of autonomous nodes or wireless terminals that communicate with each other by forming a radio network and maintaining connectivity in a decentralized manner. Since the nodes communicate over wireless connections, they have to contend with the effects of radio communication, such as noise, fading, and interference. In addition, the connections typically have less bandwidth than in a wired network. The network topology is in general dynamic, because the connectivity among the nodes may vary with time due to node departures, new node arrivals, and the possibility of having mobile nodes.

[0003] For a variety of home multimedia applications and business applications, it is important to establish networks, preferably wireless networks, for exchanging data and messages between different devices that are part of the network. In a typical business application scenario, a mobile terminal gets services over a fixed corporate or public infrastructure. In an exemplary home application scenario, a low-cost and flexible networking is supported to interconnect wireless digital consumer devices. Such devices can for example be multimedia devices which typically require high bit rates and are therefore using a lot of the available amount of bandwidth of a wireless ad hoc network.

[0004] The ETSI (European Telecommunications Standards Institute) Project BRAN (Broadband Radio Access Networks) has defined the standard HIPERLAN/2 (High Performance Radio Local Area Network), which provides high-speed multimedia communications between different broadband core networks and wireless (mobile) terminals. HIPERLAN/2 provides a flexible platform for a variety of business and home applications that can support a set of bit rates up to 54 Mbit/s. The HIPERLAN/2 standard is an example how data can be transmitted between different devices in a wireless ad hoc network. Another standardization body for wireless ad hoc networks is IEEE 802.11. The invention is not limited to wireless networks according to the HIPERLAN/2 and IEEE 802.11 standard, though.

[0005] A problem occurs within a state of the art wireless ad hoc network, if a wireless terminal requests more bandwidth than the left amount of bandwidth of the wireless ad hoc network. The terminal may request more bandwidth e.g. because it wants to setup a new wireless connection or requires more bandwidth for an existing wireless connection with another wireless terminal of the network. More bandwidth may also be required if a new wireless terminal wants to join the wireless ad hoc network and wants to set up wireless connections. In such a case, if the wireless ad hoc network runs according to the above mentioned HIPERLAN/2 standard or IEEE 802.11 standard, a wireless terminal may not be provided with the requested amount of bandwidth. Therefore, a certain user application may not be possible.

[0006] Fig. 1 shows a state of the art wireless ad hoc network 100 comprising three wireless terminal-devices, i.e. a first wireless terminal-device WT10, a second wireless terminal-device WT20, and a third wireless terminal-device WT30, and a terminal-device central controller CC100 that controls the wireless terminal-devices. Further, there are three wireless device-connections, a first wireless device-connection C23 between the second wireless terminal-device WT20 and the third wireless terminal-device WT30, a second wireless device-connection C31 between the first wireless terminal-device WT10 and the third wireless terminal-device WT30 and a third device-connection C1CC between the first wireless terminal-device WT10 and the terminal-device central controller CC100. The wireless device-connections require a certain amount of the available bandwidth of the wireless ad hoc network 100. In the example the first wireless device-terminal WT10 wants to establish a new wireless device-connection C21 with the second wireless terminal-device WT20. However, the required bandwidth for this new wireless device-connection C21 exceeds the left amount of bandwidth of the total available bandwidth of the wireless ad hoc network. Therefore, in the state of the art wireless ad hoc network 100, if the network is operated according to HIPERLAN/2 standard, the setup of the new wireless device-connection C21 fails, or, if the network is operated according to IEEE 802.11 standard, an application on the new wireless device-connection C21 is not possible. A similar situation would occur if more bandwidth would be required for an existing wireless device-connection and the required amount of bandwidth exceeds the left available bandwidth of the wireless ad hoc network. If for example the second wireless terminal-device WT20 requires more bandwidth for the first wireless device-connection C23 and the required amount of bandwidth exceeds the left amount of the total available bandwidth, an application may not be possible. Moreover, data packets transmitted within the wireless ad hoc network may get lost.

[0007] Therefore, it is an object of the invention to provide a method and means for providing additional bandwidth for a wireless ad hoc network in case the amount of bandwidth of the wireless ad hoc network is not sufficient to establish new wireless connections or to provide more bandwidth for existing wireless connections.

[0008] The object of the invention is solved by a method for providing additional bandwidth for a wireless ad hoc network according to claim 1, by a wireless terminal according to claim 7, by a central controller according

LINKPLAY EXHIBIT 1004
Page 1817 of 4822

to claim 9, and by a wireless ad hoc network according to claim 11. Preferred embodiments thereof are respectively defined in the respective following subclaims.

[0009] The method to provide additional bandwidth for a wireless ad hoc network comprising a plurality of wireless terminals operating in a certain communication channel (e.g. a RF channel at a certain frequency, an IR or optic channel) with a certain amount of available bandwidth according to the present invention comprises a step of splitting said wireless ad hoc network such that at least one new wireless ad hoc network is spawned, if more bandwidth than said certain amount of available bandwidth is required by said plurality of wireless terminals, wherein after the split of said wireless ad hoc network at least one wireless terminal of said wireless ad hoc network and/or one or more new wireless terminals belong(s) to said at least one new wireless ad hoc network, and said at least one new wireless ad hoc network is operating in a respective different communication channel.

[0010] In other words, if a first wireless ad hoc network requires more bandwidth, a second (new) wireless ad hoc network is spawned (generated). This second wireless ad hoc network is operated in a different communication channel than the first wireless ad hoc network, therefore providing additional bandwidth. Wireless terminals of the first wireless ad hoc network or new wireless terminals may be moved to the second wireless ad hoc network. It may also be possible, that new wireless terminals are moved to the first wireless ad hoc network and wireless terminals of the first wireless ad hoc network are moved to the second wireless ad hoc network. The splitting of a wireless ad hoc network (spawning of a new wireless ad hoc network) can also be seen as a bandwidth oriented reconfiguration of a wireless ad hoc network.

[0011] The wireless terminal of a wireless ad hoc network controlled by a central controller of said wireless ad hoc network according to the present invention comprises a receiving means adapted to receive a requesting command (SPAWN_NETWORK) from the central controller of the wireless ad hoc network, i.e. the first wireless ad hoc network, indicating certain operating conditions for the wireless terminal, a condition checking means to check if the wireless terminal can be operated under said certain conditions, and a sending means that sends out a confirmation command (SPAWN_NETWORK_ACK), if the condition checking means determines that the wireless terminal can be operated under said certain conditions, i.e. in the new (second) wireless ad hoc network.

[0012] The central controller of a (first) wireless ad hoc network comprising a plurality of wireless terminals according to the present invention comprises a splitting means that controls the splitting of said (first) wireless ad hoc network, wherein a new (second) wireless ad hoc network is spawned that comprises at least one of said plurality of wireless terminals and/or one or more new wireless terminals.

[0013] The wireless ad hoc network according to the present invention comprises a plurality of wireless terminals according to the present invention and a central controller according to the present invention.

[0014] Preferably, said (first) wireless ad hoc network and the splitting of said wireless ad hoc network are controlled by a central controller of said (first) wireless ad hoc network that decides which wireless terminals of said (first) wireless ad hoc network and/or which new wireless terminals are moved to said at least one new wireless ad hoc network (second wireless ad hoc network), wherein the decision is based on certain separation criteria, and said central controller determines a new central controller for said at least one new wireless ad hoc network (second wireless ad hoc network).

[0015] Further, said (first) wireless ad hoc network and said at least one new wireless ad hoc network (second wireless ad hoc network) might be operated according to the IEEE802.11 or ETSI BRAN HIPERLAN/2 standard.

[0016] It is further advantageous, if said certain separation criteria assure that wireless terminals that have the same convergence layers, and/or are of the same application layer, and/or have the same connection, and/or provide the same or corresponding functions are not separated into different wireless ad hoc networks.

[0017] Preferably, said certain separation criteria further assure that wireless terminals with certain connections that should not be interrupted are not moved to said at least one new wireless ad hoc network.

[0018] Further, the invention provides advantageous new commands in order to spawn said at least one new wireless ad hoc network (second wireless ad hoc network), i.e. a requesting command (SPAWN_ NETWORK) that is sent to a request wireless terminal to ask this request wireless terminal to move to said at least one new ad hoc wireless network (second wireless ad hoc network), and a confirmation command (SPAWN_ NETWORK_ACK) that is used by a request wireless terminal to signal that it can move to said at least one new ad hoc wireless network (second wireless ad hoc network).

[0019] Preferably, said certain conditions define if said request wireless terminal can operate as a central controller of a wireless ad hoc network, if said request wireless terminal can operate in a certain communication channel, and a moment in time at which said request wireless terminal should operate in said certain communication channel.

[0020] Further, advantageously a wireless terminal stops using its entire wireless connections (radio link control - RLC) the moment it sent out said confirmation command (SPAWN_NETWORK_ACK), moves to one of said at least one new wireless ad hoc network, waits until it receives a start command (RLC_CC_START_ OPERATION) sent out by a central controller, and then starts using its wireless connections according to the information

LINKPLAY EXHIBIT 1004
Page 1818 of 4822

(e.g. time stamp) provided by said start command (RLC_CC_START_OPERATION). It may also be possible that within said requesting command (SPAWN_ NETWORK) a time stamp is transmitted and a wireless terminal stops its entire connections at the moment in time indicated by this time stamp. Thus, all wireless terminals that are moved to said at least one new wireless ad hoc network (second ad hoc wireless network) stop their radio link control at the same time.

[0021] The invention and advantageous details thereof will be explained by way of an exemplary embodiment thereof in the following with reference to the accompanying drawings in which

Fig. 1 shows an example of a state of the art wireless ad hoc network where the setup of a new connection fails because there is not enough available bandwidth to establish the connection;

Fig. 2A shows an example of a wireless ad hoc network with different user connections wherein a wireless terminal requires more bandwidth and a wireless terminal wants to join the network;

Fig. 2B shows the splitting based on application level of the wireless ad hoc network of Fig. 2A according to the invention to provide more bandwidth;

Fig. 3 shows a flow chart illustrating the process of splitting a wireless ad hoc network;

Fig. 4A shows an example of a wireless ad hoc network with wireless connections with different convergence layers, i.e. different connection types, wherein a wireless terminal requires more bandwidth to set up a new wireless connection;

Fig. 4B shows the splitting based on connection type of the wireless ad hoc network of Fig. 4A according to the invention to provide more bandwidth; and

Fig. 5 shows a message sequence chart (MSC) for the ETSI BRAN HIPERLAN/2 standard that describes the process of splitting the wireless ad hoc network of Fig. 4A and 4B, wherein new commands according to the invention are used.

[0022] Figs. 2A and 2B show an example of a wireless ad hoc network according to the invention. In Fig. 2A, the first original wireless network 200 comprises four wireless terminals, i.e. a first wireless terminal WT1, a second wireless terminal WT2, a third wireless terminal WT3, and a fourth wireless terminal WT4, and a central controller CC. There are three wireless connections (user connections), i.e. a first wireless connection 2C13 (connecting the first wireless terminal WT1 and the third wireless terminal WT3), a second wireless connection 2C12 (connecting the first wireless terminal WT1 and the second wireless terminal WT2), and a third wireless connection 2C4CC (connecting the fourth wireless terminal WT4 and the central controller CC). In the example, the second wireless terminal WT2 requests the setup of a fourth wireless connection 2C23 with the third wireless terminal WT3 requiring a certain amount of bandwidth. Further, a fifth wireless terminal WT5 wants to join the first original wireless network 200 requesting a fifth wireless connection 2C35 with the third wireless terminal WT3 requiring a certain amount bandwidth. The central controller CC therefore checks, if all conditions for establishing the new connections, i.e. the fourth wireless connection 2C23 and the fifth wireless connection 2C35, are met, i.e. especially the central controller CC checks, if there is enough bandwidth available. In the example, there is not enough bandwidth available to establish the new connections. If the network would run according to state of the art HIPERLAN/2 standard the required connections could not be established, or if the network would run according to IEEE 802.11 standard a desired user application would not be possible.

[0023] As can be seen in Fig. 2B, according to the invention, the first original wireless ad hoc network 200 is split up into two wireless ad hoc networks such that a first new wireless ad hoc network is spawned. In Fig. 2B, wireless ad hoc network 2001 corresponds to the first original wireless ad hoc network 200 of Fig. 2A, further, a first new wireless ad hoc network 2002 can be seen. Each wireless ad hoc network operates in a different RF channel. It should be mentioned, that the invention is not limited to RF channels. It may be used for a variety of different communication channels, e.g. IR or other optic channels. If the RF channels of the wireless ad hoc network 2001 and the first new wireless ad hoc network 2002 have the same available amount of bandwidth, the total amount of bandwidth available for the wireless terminals is doubled in comparison to the first original wireless ad hoc network 200 of Fig. 2A. Here, the wireless ad hoc network 2001 operates in the same RF channel as the wireless ad hoc network 200. However, it is also possible, that the RF channel of wireless ad hoc network 2001 is changed, wherein the channel is different from that of the first new wireless ad hoc network 2002.

[0024] As can be seen in Fig. 2B, each wireless ad hoc network, i.e. wireless network 2001 and the first new wireless ad hoc network 2002, has a central controller. Wireless ad hoc network 2001 has a first central controller CC1 and wireless ad hoc network 2002 has a second central controller CC2. The first central controller CC1 is the original central controller CC of the first original wireless ad hoc network 200 of Fig. 2A. The second

LINKPLAY EXHIBIT 1004
Page 1819 of 4822

central controller CC2 is the original second wireless terminal WT2 of Fig. 2A, i.e. the second wireless terminal WT2 now has additional controlling functions to function as the second central controller CC2 of the first new wireless ad hoc network 2002.

[0025] A wireless terminal can function as a central controller. In this case the wireless terminal performs additional controlling functions. That means, a central controller can be seen as a wireless terminal with additional controlling functions. Controlling functions include the management of the network and its connections, for example the association of new wireless terminals. However, there may also be some wireless terminals that are not capable of performing addition controlling functions, i.e. they cannot be used as central controllers. If a network is split up the new central controller of the new wireless ad hoc network can therefore only be a wireless terminal with additional controlling functions. The original central controller can assign the new central controller, since the capabilities of the different wireless terminals with respect to the controlling functions are exchanged during the association process, i.e. the login of a wireless terminal to a network.

[0026] In Fig. 2B wireless ad hoc network 2001 comprises the fourth wireless terminal WT4 and the second central controller CC2. The second central controller CC2 is the central controller CC of the first original wireless ad hoc network 200 of Fig. 2A. Further, the wireless network 2001 comprises the third wireless connection 2C4CC between the fourth wireless terminal WT4 and the second central controller CC2.

[0027] In the following description, if the reference symbol of a wireless terminal, central controller, and/or a connection changes due to a split of a network, the corresponding reference symbol before the split will be given in parenthesis, e.g. the second central controller CC2 in Fig. 2B will be written as second central controller CC2 (CC).

[0028] In Fig. 2B the first new wireless ad hoc network 2002 comprises the first wireless terminal WT1, the third wireless terminal WT3, the fifth wireless terminal WT5 and the second central controller CC2 (WT2), which is the second wireless terminal WT2 of the first original wireless network 200 of Fig. 2A now functioning as a central controller. Further, the first new wireless network 2002 comprises the first wireless connection 2C13, the second wireless connection 2C12, the fourth wireless connection 2C23, and the fifth wireless connection 2C35. The fifth wireless connection 2C35 between the third wireless terminal WT3 and the fifth WT5 and the fourth wireless connection 2C23 between the second wireless terminal WT2 and the third wireless terminal WT3 are the requested connections that can now be established because there is now enough bandwidth available due to the splitting of the first original wireless network 200 of Fig. 2A.

[0029] In the example of Fig. 2B the criteria used for separating (splitting), i.e. the decision which wireless terminals are placed in which wireless network, is based on the functionality of the different wireless terminals (functional or application level based separation). Examples for different functionalities of wireless terminals are video communication and audio communication on the one hand, and data transfer (PC to PC data communication) on the other hand. In the example of Figs. 2A and 2B, the first wireless terminal WT1, the second wireless terminal WT2, the third wireless terminal WT3 and the fifth wireless terminal WT5 are PC-type devices. The fourth wireless terminal WT4 and the first central controller CC1 are video and audio devices (VCR, TV, loudspeaker, ...). Therefore, the video and audio devices, i.e. the fourth wireless terminal WT4 and the first central controller CC1 are left in wireless ad hoc network 2001 and the PC-type devices, i.e. the first wireless terminal WT1, the second wireless terminal WT2, the third wireless terminal WT3 and the fifth wireless terminal WT5 are put into the first new wireless ad hoc network 2002. There is no connection between wireless ad hoc network 2001 and the first new wireless ad hoc network 2002.

[0030] There may be other criteria according to which wireless terminals should be separated during the splitting process. Examples for such criteria are:

- Connection type (supported convergence layers). In HIPERLAN/2 it is possible to have multiple convergence layers (e.g. Ethernet, IEEE 1394) in parallel on top of one wireless connection. It may be Important not to separate wireless terminals connected by a wireless connection with the same convergence layer.

- Application layer specific criteria. In IEEE 802.11 and HIPERLAN/2 standard, an application layer may have specific requirements, e.g. communication on specific port numbers, applications, or protocols. These may be criteria to perform network separation, e.g. wireless terminals with application layers communicating on the same port number should not be separated.

- Communication on the same connection. It may be important not to separate wireless terminals that are communicating on the same connection(s).

- Functionality of devices (e.g. in IEEE 1394: Video devices, Audio devices, ...). It may be important that devices that can potentially communicate with each other are not separated. For example a video outputting device should maybe not separated from a video reproducing device (cf. example of Fig. 2B).

[0031] Another separation criterion may be the interruption of connections. During the move of a wireless terminal to a new (spawned) wireless ad hoc network, the transfer of data may not be possible for a short mo-

LINKPLAY EXHIBIT 1004
Page 1820 of 4822

ment. If the wireless terminal has an application requiring a certain quality of service, e.g. for a real time application, it should therefore eventually not be moved to the new wireless ad hoc network. If such a wireless terminal stays in the original wireless ad hoc network, the data transfer may not be interrupted, i.e. the quality of service can be assured even during a split of a network.

[0032] Fig. 3 shows a flow chart illustrating the process of splitting a wireless ad hoc network. In the original state S301 a wireless ad hoc network is established. In a following step S302 a wireless device (wireless terminal, WT) wants to set up a new connection. The wireless device therefore informs the central controller of the wireless ad hoc network about this desire to set up a new connection.

[0033] In the next step S303 the wireless terminal that wants to set up a new connection gets the feedback from the central controller that the left amount of bandwidth is insufficient to establish the new connection. The central controller therefore checks thereafter in step S304 if another channel is available. If this is not the case, the first final state S311 is reached which indicates that the application is not possible.

[0034] If another channel is available, in step S305, the central controller checks if a network split is possible. Therefore, it is checked which devices of the network need to remain in the same wireless ad hoc network and which can be moved to the new wireless network. This means the above mentioned criteria are checked and two sets of wireless terminals are formed such that wireless terminals that cannot be separated according to the respective criteria are kept within the same set, i.e. the same wireless network. It is then determined which wireless terminals are separated from the wireless ad hoc network to form a new wireless ad hoc network. (In the following these wireless terminals are referred to as splitting wireless terminals). If a network split is not possible, the first final state S311 is reached, i.e. the application is not possible. If, however, in step S305, the central controller determines that a network split is possible, in step 306, all splitting wireless terminals are informed about the split. The splitting wireless terminals are provided with the following information:

- Each splitting wireless terminal is informed whether its role in the new wireless ad hoc network is the role of a regular wireless terminal or the role of the central controller of the new network.
- The splitting devices are also informed about the splitting time, i.e. the moment in time at which the split of the network will be performed.
- Further, the RF frequency of the new wireless ad hoc network is transmitted to the splitting device such that a splitting device may restart its wireless connections after the split at the indicated frequency.

In a following step S307 the radio link control (RLC) of all affected devices, i.e. all splitting devices, is stopped.

[0035] Thereafter, in step S308 the central controller then transfers all connection information to the central controller of the new wireless ad hoc network, and the new central controller of the new wireless ad hoc network activates the RLC of all devices in the new network at the splitting time in step S309. In a second final state S310 there then exists a new wireless ad hoc network under control of the new central controller. In parallel there exists a wireless network (original wireless ad hoc network) that comprises all wireless terminals that have not been moved to the new wireless ad hoc network and the original central controller. It may also comprise new wireless terminals.

[0036] Fig. 4A and 4B show another example of a wireless ad hoc network that is split up according to the invention. Here, the separation of wireless terminals is based on the connection type of the different wireless terminals, i.e. wireless terminals with the same connection type are preferably not separated into different ad hoc networks. In the example here, a connection type refers to a convergence layer in the sense of the HIPERLAN/2 standard. This means, wireless terminals with the same convergence layers are not separated. As mentioned above in HIPERLAN/2 networks, it is possible to have different convergence layers (CL) on top of the data link control (DLC) layers and physical (PHY) layers. Specific convergence layers can be for example Ethernet. IEEE 1394, IP, ATM, or UMTS. This makes HIPERLAN/2 a multi-network air interface. There may be exceptions, i.e. it may be possible that wireless terminals with the same connection type are admitted to be separated. An exception is made, if the wireless terminals are not communicating with each other, i.e. they have no connections with each other.

[0037] Fig. 4A shows a second original wireless ad hoc network 400 before the separation according to the Invention. It comprises the first wireless terminal WT1, the second wireless terminal WT2, the third wireless terminal WT3, the fourth wireless terminal WT4, and the fifth wireless terminal WT5, and the central controller CC. Further, it comprise a sixth wireless connection 4C13, a seventh wireless connection 4C2CC, an eighth wireless connection 4C35, and a ninth wireless connection 4C4CC. The first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5 have wireless connections with the same first convergence layer CL-ID1, i.e. the sixth wireless connection 4C13 and the eighth wireless connection 4C35 have the first convergence layer CL-ID1. The fourth wireless terminal WT4 has a wireless connection with a different convergence layer, i.e. a second convergence layer CL-ID2. The second wireless terminal WT2 and the central controller CC have the seventh wireless connection 4C2CC with both convergence layers, i.e. the first con-

LINKPLAY EXHIBIT 1004
Page 1821 of 4822

vergence layer CL-ID1 and the second convergence layer CL-ID2. Now, the first wireless terminal WT1 wants to setup a tenth wireless connection 4C15 with the first convergence layer CL-ID1 with the fifth wireless terminal WT5 requiring a certain amount of bandwidth. However, the left amount of bandwidth within the wireless network is not sufficient. Therefore, according to the invention, the network is split up, i.e. a new wireless ad hoc network is spawned. It should be mentioned again, that if the network would run according to state of the art HIPERLAN/2 standard the setup of the tenth wireless connection 4C15 would fail, or if the network would run according to IEEE802.11 standard, an application would not be possible.,.

[0038] The central controller CC now determines, which wireless terminals are moved to the second new wireless ad hoc network and which remain in the second original network. The two separated wireless networks can be seen in Fig. 4B. The wireless ad hoc network 4001 in Fig. 4B corresponds to the second original wireless ad hoc network 400 of Fig. 4A, further a second new wireless ad hoc network 4002 is spawned during the splitting process. As mentioned above, in the example of Fig. 4A and 4B, the separation of wireless terminals is based on connection type, I.e. on the respective convergence layers. Since the first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5 are communicating over wireless connections with the same first convergence layer CL-ID1, these wireless terminals are not separated. Further, the second wireless terminal WT2, the fourth wireless terminal WT4 and the central controller CC have connections with each other with the second convergence layer CL-ID2 and are therefore not separated. As mentioned above, the second wireless terminal WT2 and the central controller CC additionally have the second convergence layer CL-ID1 on top of the seventh wireless connection 4C2CC. However, they have no connection with the first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5, and can therefore be separated from these wireless terminals.

[0039] As can be seen in Fig. 4B, the tenth wireless connection 4C15 between the fifth wireless terminal WT5 and the second new central controller CC2 (WT1) could now be established because of the split of the network there is now enough bandwidth available. The second new central controller CC2 is the original first wireless terminal WT1 of the second original wireless network in Fig. 4A.

[0040] Fig. 5 shows a message sequence chart (MSC) for the ETSI BRAN HIPERLAN/2 standard for the separation (splitting) process according to the invention. The MSC in Fig. 5 describes the separation process of the wireless ad hoc network in Figs. 4A and 4B. The MSC in Fig. 5 shows the first wireless terminal WT1, the second wireless terminal WT2, the third wireless terminal WT3, the fourth wireless terminal WT4, the fifth wireless terminal WT5, and the central controller CC of the second original wireless ad hoc network 400 of Fig. 4A.

[0041] In the initial state S501 the first wireless terminal WT1, the second wireless terminal WT2, the third wireless terminal WT3, the fourth wireless terminal WT4, and the fifth wireless terminal WT5 are associated to the central controller CC as shown in Fig. 4A. As mentioned above the first wireless terminal WT1 now requires more bandwidth to set up the tenth wireless connection 4C15 with the fifth wireless terminal WT5. Therefore, in step S502, the first wireless terminal WT1 sends the standard HIPERLAN/2 message "RLC_Setup" to the central controller CC indicating the required setup of the new connection requiring more bandwidth. In the following step S503, the central controller CC sends back the HIPERLAN/2 standard command "RLC_Release: lack of resources" to the first wireless terminal WT1 indicating that no bandwidth is available to set up a new connection with the required amount of bandwidth. The lack of bandwidth causes the central controller CC to check if parts of the network can be separated to a new network. This means the above mentioned criteria are checked. In the present example the criteria for the separation are the convergence layers of the wireless terminals associated to the central controller CC. As explained above with reference to the Figs. 4A and 4B, the first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5 have the same convergence layers and are therefore not separated and the second wireless terminal WT2, the fourth wireless terminal WT4, and the central controller CC are not separated.

[0042] Once, the central controller CC has determined that a split is possible and which wireless terminals are moved to the new wireless ad hoc networks (splitting wireless terminals), the central controller CC sends a command "Spawn_Network" to the splitting wireless terminals, here, the first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5. The command "Spwan_Network " is defined as follows:

SPAWN NETWORK

[0043] functionality: a central controller (CC) informs wireless terminals (WTs) to change to a new wireless network

parameters:

channel	center frequency of the new network
CC_or_MT	defines role of the device in the new network (wireless terminal or new central controller)
start_mac_frame	defines point of time when devices belong to new network and are controlled by the new central controller CC.

LINKPLAY EXHIBIT 1004
Page 1822 of 4822

[0044] A wireless terminal that receives a command "Spawn_Network" from a central controller sends back a command "Spawn_Network_ack" to the central controller indicating whether it can change to the new network or not. The command "Spawn_Network_ack" is defined as follows:

SPAWN NETWORK ACK

[0045] functionality: wireless terminals acknowledge SPAWN_NETWORK command parameters:

ACCEPT_ACTION	indicates if the wireless terminal can change to the new network or rejects the separation. The rejection lets the central controller look for new separation possibilities or cancel the separation process.

[0046] As soon as a command "Spawn_Network_ack" is sent out by a wireless terminal indicating that the wireless terminal can change to a new network (the parameter ACCEPT_ACTION is set accordingly), the wireless terminal stops its radio link control (RLC). When the time indicated by the command "start_mac_frame" arrives, the wireless terminal moves to the new network, i.e. it sets all its parameters (e.g. frequency) according to the values indicated by the command "SPAWN_NETWORK". It then waits for a wake-up command "RLC_CC_START_ OPERATION" sent out by the new central controller of the new wireless ad hoc network to actually start its RLC.

[0047] Now, in the example of Fig. 5, in step S504, the command "Spawn_Network" is sent to the first wireless terminal WT1. Since the first wireless terminal WT1 will become the second new central controller CC2, it is the designated central controller CC-CAND (central controller candidate) of the new wireless ad hoc network. Therefore, the parameter "CC_or_MT" of the command "Spawn_Network" indicates that the first wireless terminal WT1 becomes a central controller. In the following step S505 the first wireless terminal WT1 sends back the command "Spawn_Network_ack" to the central controller CC indicating that it can change to the new network and can be the central controller of the new network. Then, in step S506, the central controller sends the command "Spawn_Network" to the third wireless terminal WT3. Within this Spawn_Network command the parameter "CC_or_MT" indicates that the third wireless terminal WT3 remains a wireless terminal, i.e. does not become a central controller. Then, in step S507, the third wireless terminal WT3 sends back a command "Spawn_Network_ack" to the central controller CC indicating that it can change to the new network. In the following step, i.e. step S508, the central controller CC sends the command "Spawn_Network" to the fifth wireless terminal WT5. The parameter "CC_or_MT" again indicates that the fifth wireless terminal WT5 remains a wireless terminal in the new network. The fifth wireless terminal WT5 sends back the command "Spawn_ Network_ack" in step S509 to the central controller CC indicating that it can change to the new network.

[0048] As mentioned above, when a wireless terminal sends out a command "Spawn_Network_ack" it stops its RLC. Therefore, in the respective steps S505, S507, and S509 the first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5 stopped their RLC. Therefore, in the internal state S510 the RLC of the affected devices, i.e. the first wireless terminal WT1, the third wireless terminal WT3, and the fifth wireless terminal WT5 are stopped.

[0049] Thereafter, in step S511, the central controller CC sends out the HIPERLAN/2 standard command "RLC_TRANS_CC_DATA" to the central controller candidate CC-CAND of the second new wireless ad hoc network (the first wireless terminal WT1). With this command the central controller CC transmits all connection-relevant data to the first wireless terminal WT1 (central controller candidate CC-CAND of the second new network). It is therefore possible to maintain all direct link connections that shall be moved to the new network without the need of a new setup procedure. After the central controller candidate CC-CAND of the new network, here the first wireless terminal WT1, is aware of all network-relevant data it sends the standard HIPERLAN/2 command "RLC_CC_START_OPERATION" to all wireless terminals of the new network. Besides other information within the command "RLC_CC_START_ OPERATION" also a time stamp is transmitted indicating at which time the wireless terminal that receives the command may start its RLC. This way, all wireless terminals of the new network start their RLC at the same moment in time.

[0050] In the next step, i.e. in step S512, the central controller candidate CC-CAND (WT1) sends a command "RLC_TRANS_DATA_ACK" of the HIPERLAN/2 standard to the central controller to acknowledge the reception of the command "RLC_TRANS_CC_DATA".

[0051] In the following step S513, the central controller candidate CC_CAND (WT1) sends the HIPERLAN/2 standard command "RLC_CC_START_OPERATION" to the third wireless terminal WT3. In step S514 the central controller candidate CC_CAND (WT1) then sends the HIPERLAN/2 standard command "RLC_CC_ START_OPERATION" to the fifth wireless terminal WT5.

[0052] In step S515, when the time indicated by the time stamps of the commands "RLC_CC_START_ OPERATION" transmitted to the third wireless terminal WT3 and the fifth wireless terminal WT5 in steps S513 and S514 arrives, the RLC of the third wireless terminal WT3 and the fifth wireless terminal WT5 and the RLC of the second new central controller CC2, i.e. CC-CAND (WT1), are enabled.

[0053] In the final state S516 the separation process is finished and the second new wireless ad hoc network

LINKPLAY EXHIBIT 1004
Page 1823 of 4822

is running as shown in Fig. 4B.

[0054] In the following the invention is summarized:

[0055] A first wireless ad hoc network with a certain available amount of bandwidth comprises a plurality of wireless terminals communicating via wireless connections with each other. Each wireless connection requires a certain amount of the available amount of bandwidth. A problem occurs, if a wireless terminal wants to set up a new wireless connection requiring bandwidth or requires more bandwidth for an existing wireless connection and the required amount of bandwidth exceeds the left amount of the available amount of bandwidth of the first wireless ad hoc network. In a state of the art wireless ad hoc network the requested bandwidth cannot be provided and a user application may not be possible. According to the invention, however, the required amount of bandwidth can be provided by splitting the first wireless ad hoc network such that a second wireless ad hoc network is spawned which operates in a different communication channel and thus provides additional bandwidth. The second wireless ad hoc network comprises wireless terminals of the first wireless ad hoc network and/or new wireless terminals.

Claims

1. A method to provide additional bandwidth for a wireless ad hoc network operating in a certain communication channel with a certain amount of available bandwidth

   comprising a plurality of wireless terminals

   characterized in that

   said wireless ad hoc network is split up such that at least one new wireless ad hoc network is spawned, if more bandwidth than said certain amount of available bandwidth is required by said plurality of wireless terminals,

   wherein after the split of said wireless ad hoc network at least one wireless terminal of said wireless ad hoc network and/or one or more new wireless terminals belong(s) to said at least one new wireless ad hoc network, and

   said at least one new wireless ad hoc network is operating in a respective different communication channel.

2. The method according to claim 1,

   characterized in that

   said wireless ad hoc network and the splitting of said wireless ad hoc network are controlled by a central controller of said wireless ad hoc network that decides which wireless terminals of said wireless ad hoc network and/or which new wireless terminals are moved to said at least one new wireless ad hoc network,

   wherein the decision is based on certain separation criteria, and

   said central controller determines a new central controller for said at least one new wireless ad hoc network.

3. The method according to claim 1 or 2,

   characterized in that

   said wireless ad hoc network and said at least one new wireless ad hoc network are operated according to the IEEE802.11 or ETSI BRAN HIPERLAN/2 standard.

4. The method according to anyone of the preceding claims,

   characterized in that

   said certain separation criteria assure that wireless terminals that

   - have the same convergence layers, and/or
   - are of the same application layer, and/or
   - have the same connection, and/or
   - provide the same or corresponding functions

   are not separated into different wireless ad hoc networks.

5. The method according to anyone of the preceding claims,

   characterized in that

   said certain separation criteria assure that wireless terminals with certain connections that should not be interrupted are not moved to said at least one new wireless ad hoc network.

6. The method according to anyone of the preceding claims,

   characterized by

   providing new commands in order to spawn said at least one new wireless ad hoc network,

   wherein a requesting command (SPAWN_NETWORK) is sent to a request wireless terminal to ask this request wireless terminal to move to said at least one new ad hoc wireless network,

   and a confirmation command (SPAWN_NETWORK_ACK) is used by a request wireless terminal to signal that it can move to said at least one new ad hoc wireless network.

7. The method according to claim 6,

   characterized in that

   a wireless terminal stops using its entire wireless connections the moment it sent out said confirmation command (SPAWN_NETWORK_ACK),

   moves to one of said at least one new wireless ad hoc network,

   waits until it receives a start command (RLC_CC_START_OPERATION) sent out by a central controller, and

   then starts using its wireless connections ac-

LINKPLAY EXHIBIT 1004
Page 1824 of 4822

cording to the information provided by said start command (RLC_CC_START_OPERATION).

8. A wireless terminal of a wireless ad hoc network controlled by a central controller of said wireless ad hoc network

characterized by

a receiving means adapted to receive a requesting command (SPAWN_NETWORK) from the central controller indicating certain operating conditions for the wireless terminal,

a condition checking means to check if the wireless terminal can be operated under said certain conditions, and

a sending means that sends out a confirmation command (SPAWN_NETWORK_ACK), if the condition checking means determines that the wireless terminal can be operated under said certain conditions.

9. The wireless terminal according to claim 8

characterized in that

said certain conditions define

if said wireless terminal can operate as a central controller of a wireless ad hoc network,

a certain communication channel at which said wireless terminal is able to operate, and/or

a moment in time at which said wireless terminal shall operate in said certain communication channel and at which it may be controlled by a different central controller.

10. A central controller of a wireless ad hoc network comprising a plurality of wireless terminals,

characterized by

a splitting means that controls the splitting of said wireless ad hoc network, wherein a new wireless ad hoc network is spawned that comprises at least one of said plurality of wireless terminals and/or one or more new wireless terminals.

11. The central controller according to claim 10,

characterized in that

the splitting means comprises

a sending means that sends out requesting commands (SPAW-N_NETWORK) to wireless terminals,

a receiving means that receives confirmation commands (SPAWN_NETWORK_ACK), and

an operating means that decides which of said plurality of wireless terminals and/or of said new wireless terminals may be moved to said new wireless ad hoc network and determines a wireless terminal of said plurality of wireless terminals and/or of said new wireless terminals that becomes the central controller of said new wireless ad hoc network.

12. A wireless ad hoc network comprising

a plurality of wireless terminals according to claim 7 or 8, and

a central controller according to claim 9 or 10.

Amended claims in accordance with Rule 86(2) EPC.

1. A method to provide additional bandwidth for a wireless ad hoc network operating in a certain communication channel with a certain amount of available bandwidth comprising a plurality of wireless terminals, **characterized in that**

- said wireless ad hoc network is split up such that at least one new wireless ad hoc network is spawned, if more bandwidth than said certain amount of available bandwidth is required by said plurality of wireless terminals,
- wherein after the split of said wireless ad hoc network at least one wireless terminal of said wireless ad hoc network and/or one or more new wireless terminals belong(s) to said at least one new wireless ad hoc network, and
- said at least one new wireless ad hoc network is operating in a respective different communication channel,
- wherein the decision which of said wireless terminals of said wireless ad hoc network and/or which of said at least one wireless terminal are moved to said at least one new wireless ad hoc network is based on certain separation criteria,
- which certain separation criteria assure that wireless terminals that
  - have the same convergence layers, and/or
  - are of the same application layer, and/or
  - have the same connection, and/or
  - provide the same or corresponding functions are not separated into different wireless ad hoc networks.

2. The method according to claim 1, **characterized in that**

said wireless ad hoc network and the splitting of said wireless ad hoc network are controlled by a central controller of said wireless ad hoc network that decides which wireless terminals of said wireless ad hoc network and/or which new wireless terminals are moved to said at least one new wireless ad hoc network,

wherein the decision is based on said certain separation criteria, and

said central controller determines a new central controller for said at least one new wireless ad hoc network.

3. The method according to claim 1 or 2, **charac-**

LINKPLAY EXHIBIT 1004
Page 1825 of 4822

terized in that

said wireless ad hoc network and said at least one new wireless ad hoc network are operated according to the IEEE802.11 or ETSI BRAN HIPERLAN/2 standard.

4. The method according to anyone of the preceding claims, **characterized in that**

said certain separation criteria assure that wireless terminals With certain connections that should not be interrupted are not moved to said at least one new wireless ad hoc network.

5. The method according to anyone of the preceding claims, **characterized by**

providing new commands in order to spawn said at least one new wireless ad hoc network,

wherein a requesting command (SPAWN_NETWORK) is sent to a request wireless terminal to ask this request wireless terminal to move to said at least one new ad hoc wireless network,

and a confirmation command (SPAWN_NETWORK_ACK) is used by a request wireless terminal to signal that it can move to said at least one new ad hoc wireless network.

6. The method according to claim 5, **characterized in that**

a wireless terminal stops using its entire wireless connections the moment it sent out said confirmation command (SPAWN_NETWORK_ACK),

moves to one of said at least one new wireless ad hoc network,

waits until it receives a start command (RLC_CC_START_OPERATION) sent out by a central controller, and

then starts using its wireless connections according to the information provided by said start command (RLC_CC_START_OPERATION).

7. A wireless terminal of a wireless ad hoc network controlled by a central controller of said wireless ad hoc network **characterized by**

a receiving means adapted to receive a requesting command (SPAWN_NETWORK) from the central controller indicating certain operating conditions for the wireless terminal,

a condition checking means to check if the wireless terminal can be operated under said certain conditions, and

a sending means that sends out a confirmation command (SPAWN_NETWORK_ACK), if the condition checking means determines that the wireless terminal can be operated under said certain conditions.

8. The wireless terminal according to claim 7, **characterized in that**

said certain conditions define

if said wireless terminal can operate as a central controller of a wireless ad hoc network,

a certain communication channel at which said wireless terminal is able to operate, and/or

a moment in time at which said wireless terminal shall operate in said certain communication channel and at which it may be controlled by a different central controller.

9. A central controller of a wireless ad hoc network comprising a plurality of wireless terminals, **characterized by**

a splitting means that controls the splitting of said wireless ad hoc network, wherein a new wireless ad hoc network is spawned that comprises at least one of said plurality of wireless terminals and/or one or more new wireless terminals.

10. The central controller according to claim 10, **characterized in that**

the splitting means comprises

a sending means that sends out requesting commands (SPAW-N_NETWORK) to wireless terminals,

a receiving means that receives confirmation commands (SPAWN_NETWORK_ACK), and

an operating means that decides which of said plurality of wireless terminals and/or of said new wireless terminals may be moved to said new wireless ad hoc network and determines a wireless terminal of said plurality of wireless terminals and/or of said new wireless terminals that becomes the central controller of said new wireless ad hoc network.

11. A wireless ad hoc network comprising

a plurality of wireless terminals according to claim 7 or 8, and

a central controller according to claim 9 or 10.

LINKPLAY EXHIBIT 1004
Page 1826 of 4822

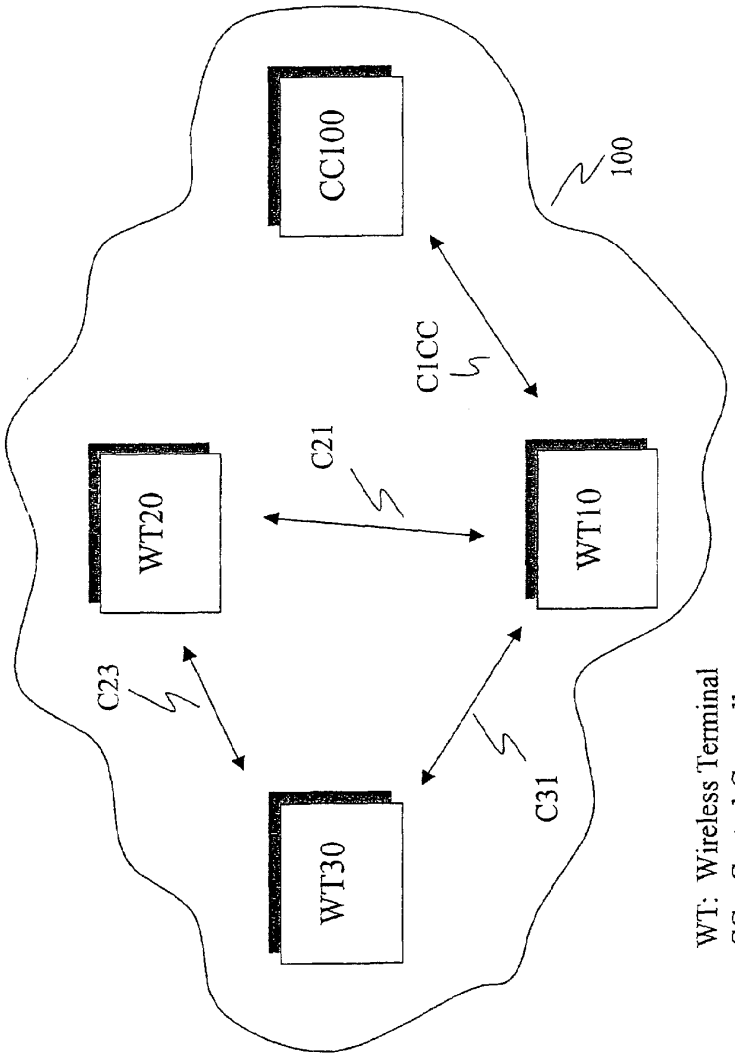


Figure 1

LINKPLAY EXHIBIT 1004
Page 1827 of 4822

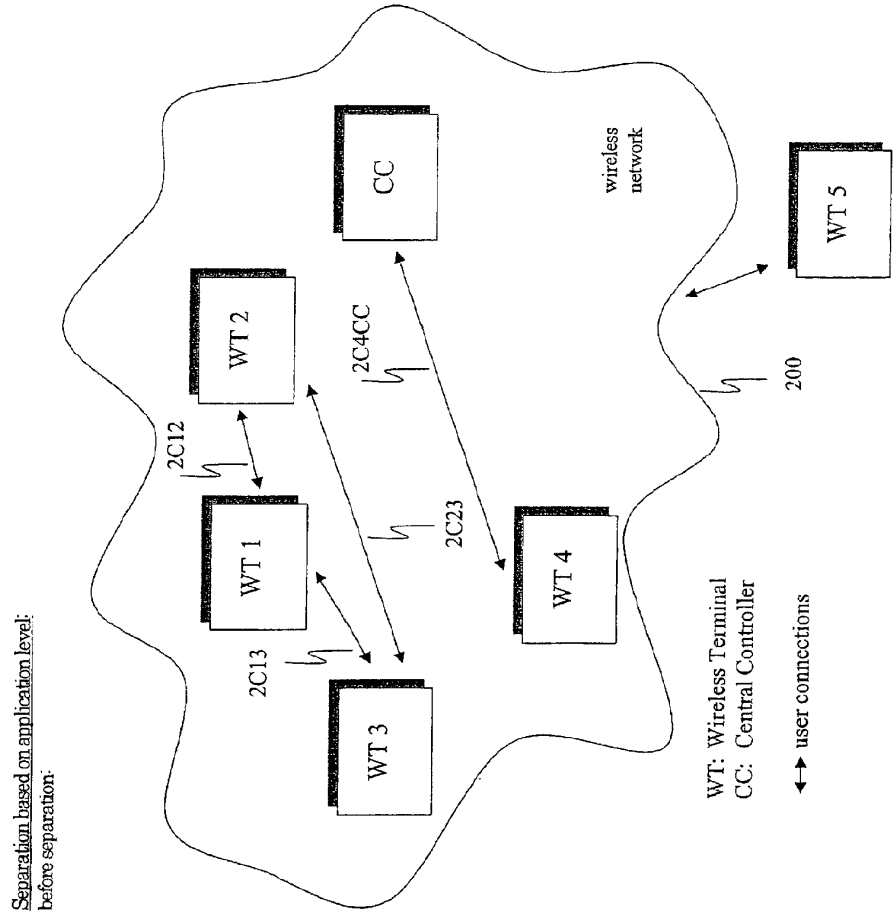


Figure 2A

LINKPLAY EXHIBIT 1004
Page 1828 of 4822

Separation based on application level:
after separation:

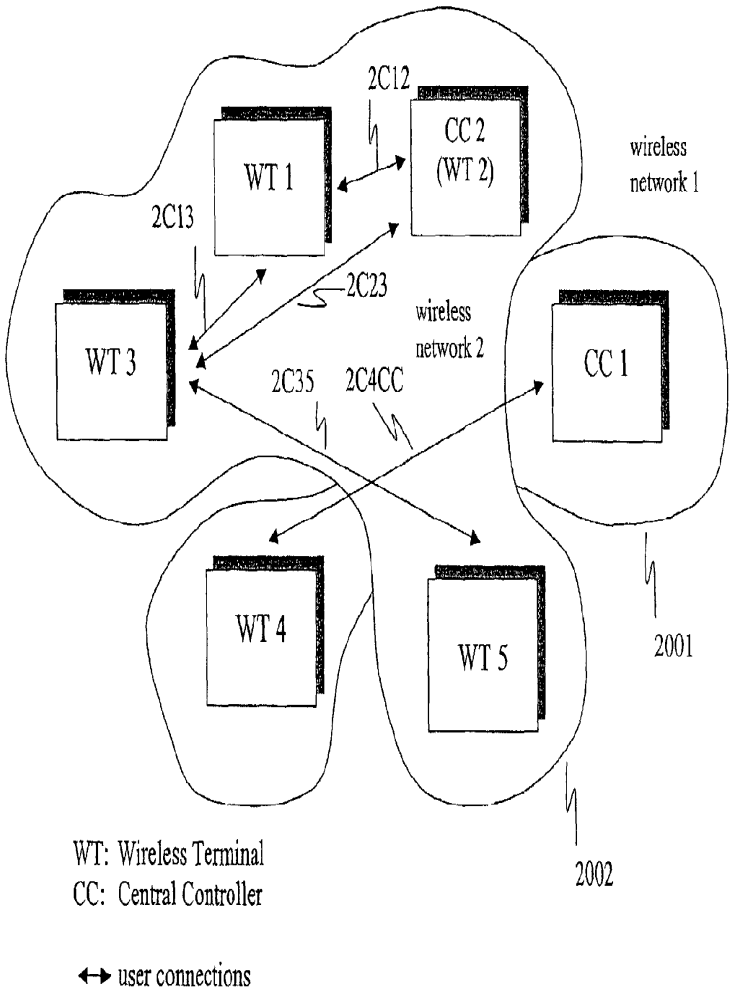


Figure 2B

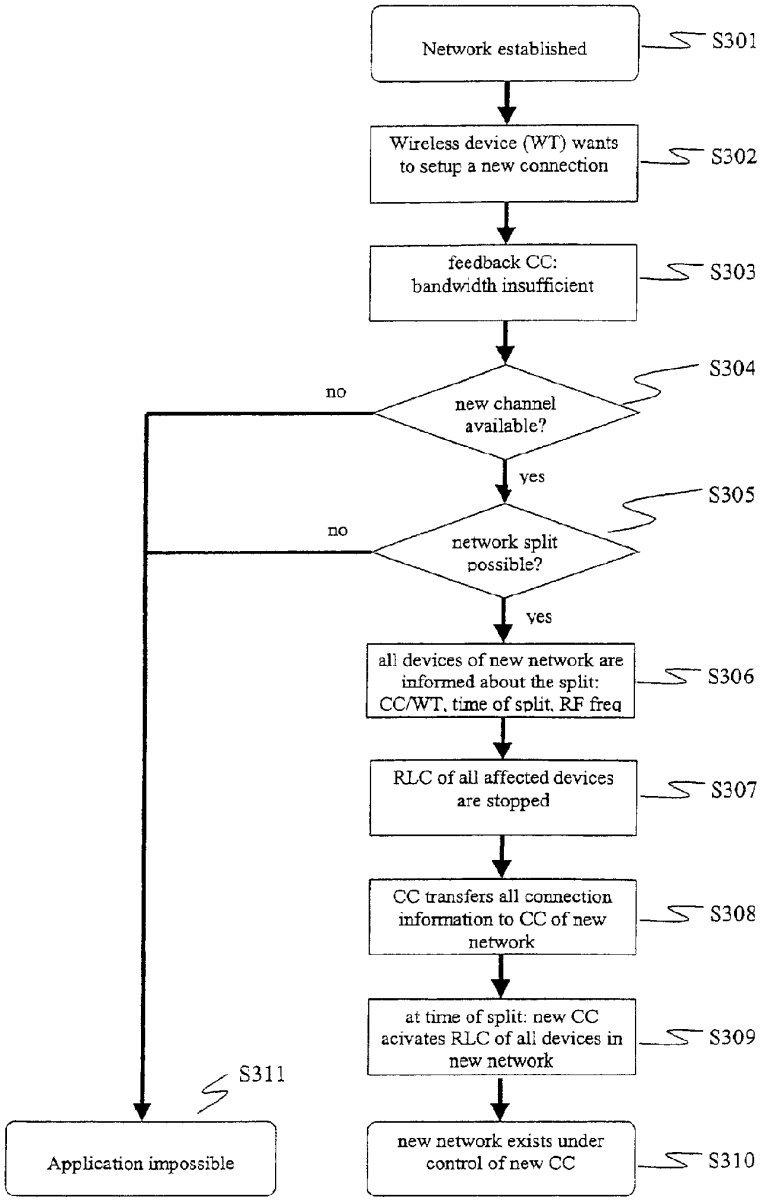


Figure 3

LINKPLAY EXHIBIT 1004
Page 1830 of 4822

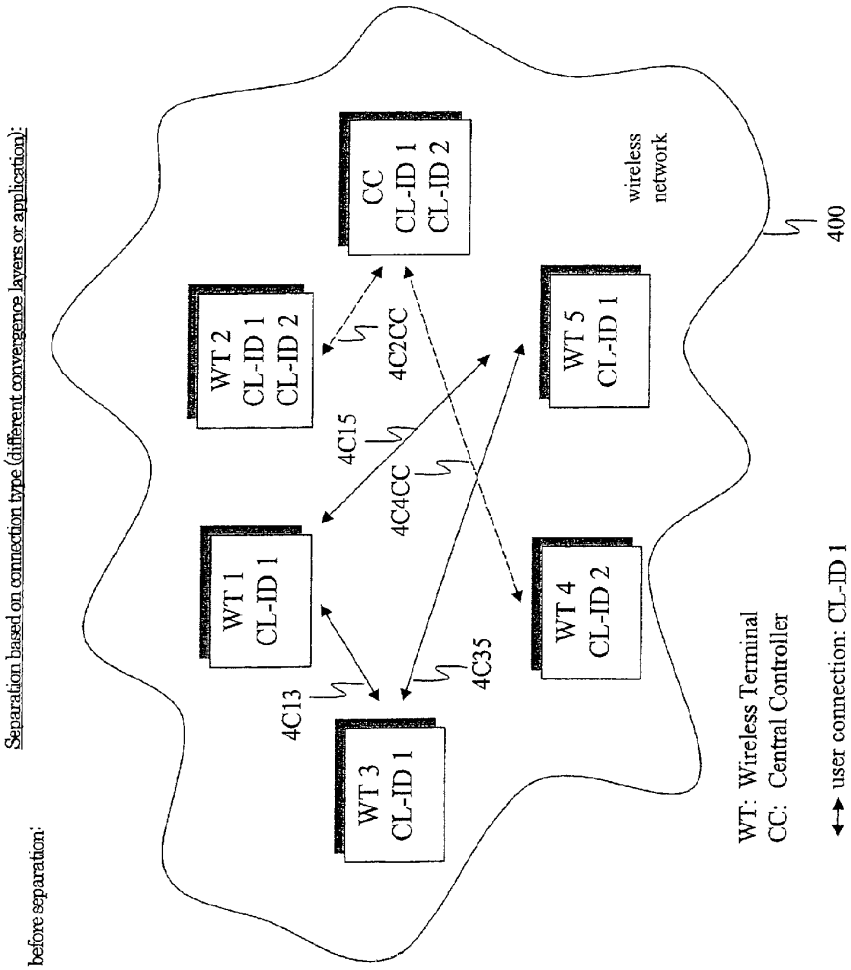


Figure 4A

LINKPLAY EXHIBIT 1004
Page 1831 of 4822

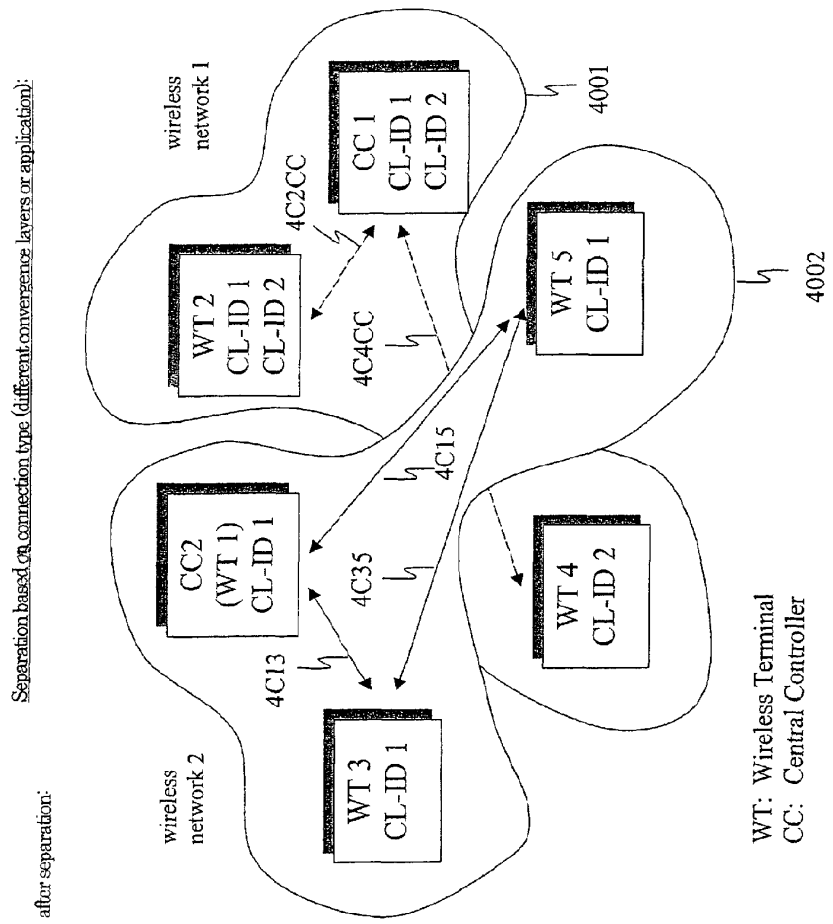
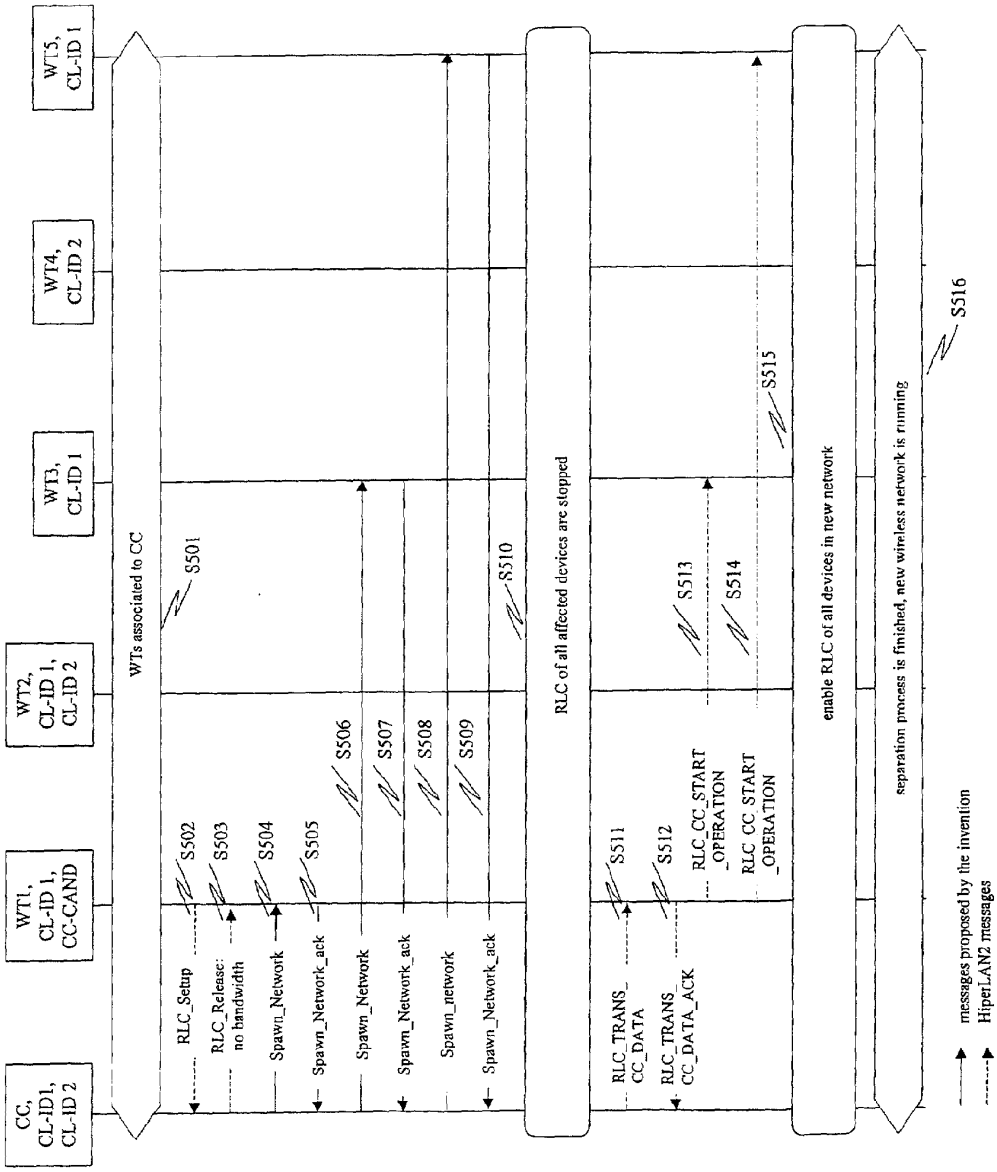


Figure 4B

LINKPLAY EXHIBIT 1004
Page 1832 of 4822

Figure 5



LINKPLAY EXHIBIT 1004
Page 1833 of 4822

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 01 8385

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	DE 100 53 809 A (PHILIPS CORP INTELLECTUAL PTY) 8 May 2002 (2002-05-08) * column 1, line 1 - line 37 * * column 2, line 55 - line 67 * * column 3, line 9 - line 65 * * column 4, line 5 - line 60 * * column 7, line 5 - line 39 * ---	1-12	H04L12/28
X	WO 99 14897 A (ERICSSON TELEFON AB L M) 25 March 1999 (1999-03-25) * page 1, line 1 - line 18 * * page 3, line 3 - line 22 * * page 5, line 13 - page 6, line 5 * * page 15, line 14 - page 16, line 5 * * page 23, line 6 - line 23 * ---	1-12	
A	VILLELA D ET AL: "Virtuosity: Programmable resource management for spawning networks" COMPUTER NETWORKS, ELSEVIER SCIENCE PUBLISHERS B.V., AMSTERDAM, NL, vol. 36, no. 1, June 2001 (2001-06), pages 49-73, XP004304884 ISSN: 1389-1286 * abstract * * page 49, left-hand column, line 1 - page 52, left-hand column, line 30 * * page 55, right-hand column, line 34 - page 56, left-hand column, line 36 * * page 58, left-hand column, line 1 - page 61, left-hand column, line 30 * --- -/--	1-12	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	29 January 2003	Heinrich, D

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

LINKPLAY EXHIBIT 1004
Page 1834 of 4822

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 01 8385

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	CAMPBELL A T ET AL: "SPAWNING NETWORKS" IEEE NETWORK, IEEE INC. NEW YORK, US, July 1999 (1999-07), pages 16-29, XP000875025 ISSN: 0890-8044 * abstract * * page 16, right-hand column, line 19 - page 17, right-hand column, line 34 * * page 18, right-hand column, line 53 - page 19, right-hand column, line 34 * * page 20, right-hand column, line 36 - page 21, left-hand column, line 40 * * page 24, right-hand column, line 12 - page 26, left-hand column, line 36 * * page 27, left-hand column, line 56 - right-hand column, line 2 * ---	1-12	
A	US 2002/018448 A1 (AMIS ALAN DEWAYNE ET AL) 14 February 2002 (2002-02-14) * page 1, paragraphs 2,4 * * page 3, paragraph 44 * * page 6, paragraph 108 * -----	1-12	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	29 January 2003	Heinrich, D

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

LINKPLAY EXHIBIT 1004
Page 1835 of 4822

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 01 8385

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29-01-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE 10053809 A	08-05-2002	DE 10053809 A1	08-05-2002
		AU 1057202 A	15-05-2002
		WO 0237770 A2	10-05-2002
		EP 1249106 A2	16-10-2002
		US 2002194384 A1	19-12-2002
WO 9914897 A	25-03-1999	AU 9099498 A	05-04-1999
		BR 9812226 A	18-07-2000
		CA 2304189 A1	25-03-1999
		CN 1278974 T	03-01-2001
		EE 200000166 A	16-04-2001
		EP 1016241 A2	05-07-2000
		JP 2001517021 T	02-10-2001
		NO 20001378 A	12-05-2000
		WO 9914897 A2	25-03-1999
US 2002018448 A1	14-02-2002	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

LINKPLAY EXHIBIT 1004
Page 1836 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 410 686 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
26.03.2008 Bulletin 2008/13

(21) Application number: **02720929.5**

(22) Date of filing: **07.02.2002**

(51) Int Cl.:
H04S 3/00 *(2006.01)*

(86) International application number:
PCT/US2002/003619

(87) International publication number:
WO 2002/063925 (15.08.2002 Gazette 2002/33)

(54) **AUDIO CHANNEL TRANSLATION**

AUDIOKANALÜBERSETZUNG

MODULATION DE CANAL AUDIO

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

(30) Priority: **07.02.2001 US 267284 P**

(43) Date of publication of application:
21.04.2004 Bulletin 2004/17

(73) Proprietor: **DOLBY LABORATORIES LICENSING CORPORATION**
San Francisco
California 94103 (US)

(72) Inventor: **DAVIS, Mark, Franklin**
San Francisco, CA 94103 (US)

(74) Representative: **Hoffmann, Eckart**
Patentanwalt,
Bahnhofstrasse 103
82166 Gräfelfing (DE)

(56) References cited:
EP-A- 1 001 549
EP-A- 1 054 575
WO-A-99/51063
US-A- 6 009 179

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

EP 1 410 686 B1

Description

TECHNICAL FIELD

[0001] The invention relates to audio signal processing. More particularly the invention relates to translating M audio input channels representing a soundfield to N audio output channels representing the same soundfield, wherein each channel is a single audio stream representing audio arriving from a direction, M and N are positive whole integers, and M is at least 2.

BACKGROUND ART

[0002] Although humans have only two ears, we hear sound as a three dimensional entity, relying upon a number of localization cues, such as head related transfer functions (HRTFs) and head motion. Full fidelity sound reproduction therefore requires the retention and reproduction of the full 3D soundfield, or at least the perceptual cues thereof. Unfortunately, sound recording technology is not oriented toward capture of the 3D soundfield, nor toward capture of a 2D plane of sound, nor even toward capture of a 1D line of sound. Current sound recording technology is oriented strictly toward capture, preservation, and presentation of zero dimensional, discrete channels of audio.

[0003] Most of the effort on improving fidelity since Edison's original invention of sound recording has focused on ameliorating the imperfections of his original analog modulated-groove cylinder/disc media. These imperfections included limited, uneven frequency response, noise, distortion, wow, flutter, speed accuracy, wear, dirt, and copying generation loss. Although there were any number of piecemeal attempts at isolated improvements, including electronic amplification, tape recording, noise reduction, and record players that cost more than some cars, the traditional problems of individual channel quality were arguably not finally resolved until the singular development of digital recording in general, and specifically the introduction of the audio Compact Disc. Since then, aside from some effort at further extending the quality of digital recording to 24bits/96 kHz sampling, the primary efforts in audio reproduction research have been focused on reducing the amount of data needed to maintain individual channel quality, mostly using perceptual coders, and on increasing the spatial fidelity. The latter problem is the subject of this document.

[0004] Efforts on improving spatial fidelity have proceeded along two fronts: trying to convey the perceptual cues of a full sound field, and trying to convey an approximation to the actual original sound field. Examples of systems employing the former approach include binaural recording and two-speaker-based virtual surround systems. Such systems exhibit a number of unfortunate imperfections, especially in reliably localizing sounds in some directions, and in requiring the use of headphones or a fixed single listener position.

[0005] For presentation of spatial sound to multiple listeners, whether in a living room or a commercial venue like a movie theatre, the only viable alternative has been to try to approximate the actual original sound field. Given the discrete channel nature of sound recording, it is not surprising that most efforts to date have involved what might be termed conservative increases in the number of presentation channels. Representative systems include the panned-mono three-speaker film soundtracks of the early 50's, conventional stereo sound, quadraphonic systems of the 60's, five channel discrete magnetic soundtracks on 70mm films, Dolby surround using a matrix in the 70's, AC-3 5.1 channel sound of the 90's, and recently, Surround-EX 6.1 channel sound. "Dolby", "Pro Logic" and "Surround EX" are trademarks of Dolby Laboratories Licensing Corporation. To one degree or another, these systems provide enhanced spatial reproduction compared to monophonic presentation. However, mixing a larger number of channels incurs larger time and cost penalties on content producers, and the resulting perception is typically one of a few scattered, discrete channels, rather than a continuum soundfield. Aspects of Dolby Pro Logic decoding are described in U.S. Patent 4,799,260. Details of AC-3 are set forth in "Digital Audio Compression Standard (AC-3)," Advanced Television Systems Committee (ATSC), Document A/52, December 20, 1995 (available on the World Wide Web of the Internet at www.atsc.org/Standards/A52/a_52.doc). See also the Errata Sheet of July 22, 1999 (available on the World Wide Web of the Internet at www.dolby.com/tech/ATSC_err.pdf.

Insights Underlying Aspects of the Present Invention

[0006] The basis for recreating an arbitrary distribution in a source-free wave medium is provided by a theorem by Gauss that stipulates that a wave field within some region is completely specified by the pressure distribution along the boundary of the region. This implies that re-creation of the sound field in a concert hall within the confines of a living room is possible by conceptually placing the living room, walls impermeable to sound, within the concert hall, then electronically rendering the walls sonically transparent by festooning the outside of the walls with an infinite number of infinitesimal microphones, each connected with suitable amplification to a corresponding loudspeaker just inside the wall. By interposing a suitable recording medium between microphones and speakers, a complete, if impractical, system of accurate 3D sound reproduction is realized. The only remaining design task is to render the system practical.

LINKPLAY EXHIBIT 1004
Page 1838 of 4822

[0007] A first step toward practicality can be taken by noting the signal of interest is bandlimited, at about 20 kHz, permitting the application of the Spatial Sampling theorem, a variant of the more common Temporal Sampling theorem. The latter holds that there is no loss of information if a continuous bandlimited temporal waveform is discretely sampled at a rate at least twice the highest frequency of the source. The former theory follows from the same considerations to stipulate that the spatial sampling interval must by at least twice as dense as the shortest wavelength in order to avoid information loss. Since the wavelength of 20 kHz in air is about 3/8", the implication is that an accurate 3D sound system can be implemented with an array of microphones and loudspeakers spaced no more than 3/16" apart. Extended over all surfaces of a typical 9'x12' room, this works out to about 2.5 million channels, a considerable improvement over an infinite number, but still impractical at this time. Still, it establishes the basic approach of using an array of discrete channels as spatial samples, from which the sound field can be recovered by application of appropriate interpolation.

[0008] Once the sound field is characterized, it is possible in principle for a decoder to derive the optimal signal feed for any output loudspeaker. The channels supplied to such a decoder will be referred to herein variously as "cardinal," "transmitted," and "input" channels, and any output channel with a location that does not correspond to the position of one of the cardinal channels will be referred to as an "intermediate" channel. An output channel may also have a location coincident with the position of a cardinal input channel.

[0009] It is therefore desirable to reduce the number of discrete channel spatial samples, or cardinal channels. One possible basis for doing so is the fact that, above 1500 Hz, the ear no longer follows individual cycles, only the critical band envelope. This might allow channel spacing commensurate with 1500 Hz, or about 3". This would reduce the total for the 9'x12' room to about 6000 channels, a useful saving of about 2.49 million channels compared to the previous arrangement.

[0010] In any case, further reduction in the number of spatial sampling channels is theoretically possible by appeal to psychoacoustic localization limits. The horizontal limit of resolution, for centered sounds, is about 1 degree of arc. The corresponding limit of vertical resolution is about 5 degrees. If this density is extended appropriately around a sphere, the result will still be a few hundred to a few thousand channels.

[0011] EP 1 054 575 A discloses a process for translating two audio input channels representing a soundfield to eight audio output channels representing the same soundfield, wherein each channel is a single audio stream representing audio arriving from a direction. A matrix generates, from the two input channels the eight output channels by a process that includes determining a measure of the correlation of the two input channels and the level interrelationships of the two input channels.

[0012] EP 1 001 549 discloses an encoding/decoding system having, on the encoding side, a mix and matrix circuit for translating six audio input channels representing a soundfield, wherein each input channel is a single audio stream representing audio arriving from a direction, to six intermediate signals. Each intermediate signal is associated with two or three input channels. On the decoding side this known system has another mix and matrix circuit for translating the six intermediate signals to six audio output channels respectively corresponding to the six input channels. The mix and matrix circuit on the encoding side includes: an adder which adds a first and a second one of the input channels to a first intermediate signal, a subtracter which subtracts the second input channel from the first input channel to generate a second intermediate signal, a combination of an adder, a 1/2 divider, and a subtracter which processes a third input channel, a fourth input channel, and a fifth input channel into a third intermediate signal, another adder which adds the fourth input channel and the fifth input channel to a fourth intermediate signal, a subtracter which subtracts the fifth input channel from the fourth input channel to generate a fifth intermediate signal, and a combination of a multiplier and a subtracter which processes the third input channel and the sixth input channel into the sixth intermediate signal.

[0013] It is an object of the invention to solve the problem of how to approximate, in a practical way, an actual original sound field for presentation to multiple listeners in a living room or a commercial venue like a movie theatre.

DISCLOSURE OF THE INVENTION

[0014] This object is achieved by a process as claimed in claim 1 and its preferred embodiment as claimed in claim 2.

[0015] In accordance with the present invention, a process translates M audio input channels representing a soundfield to N audio output channels representing the same soundfield, wherein each channel is a single audio stream representing audio arriving from a direction, M and N are positive whole integers, and M is at least two. One or more sets of output channels are generated, each set having one or more output channels. Each set is associated with two or more spatially adjacent input channels and each output channel in a set is generated by a process that includes determining a measure of the correlation of the two or more input channels and the level interrelationships of the two or more input channels.

[0016] In one aspect of the present invention, multiple sets of output channels are associated with more than two input channels and the process determines the correlation of input channels with which each set of output channels is associated according to a hierarchical order such that each set or sets is ranked according to the number of input channels with which its output channel or channels are associated, the greatest number of input channels having the highest ranking, and the processing processes sets in order according to their hierarchical order. Further according to an aspect of the

LINKPLAY EXHIBIT 1004
Page 1839 of 4822

present invention, the processing takes into account the results of processing higher order sets.

[0017] The playback or decoding aspects of the present invention assume that each of the M audio input channels representing audio arriving from a direction was generated by a passive-matrix nearest-neighbor amplitude-panned encoding of each source direction (i.e., a source direction is assumed to map primarily to the nearest cardinal channel or channels), without the requirement of additional side chain information (the use of side chain or auxiliary information is optional), making it compatible with existing mixing techniques, consoles, and formats. Although such source signals may be generated by explicitly employing a passive encoding matrix, most conventional recording techniques inherently generate such source signals (thus, constituting an "effective encoding matrix"). The playback or decoding aspects of the present invention are also largely compatible with natural recording source signals, such as might be made with five real directional microphones, since, allowing for some possible time delay, sounds arriving from intermediate directions tend to map principally to the nearest microphones (in a horizontal array, specifically to the nearest pair of microphones).

[0018] A decoder or decoding process according to aspects of the present invention may be implemented as a lattice of coupled processing modules or modular functions (hereinafter, "decoding modules"), each of which is used to generate one or more output channels (or, alternatively, control signals usable to generate one or more output channels) from the two or more of the closest spatially adjacent cardinal channels associated with the decoding module. The output channels represent relative proportions of the audio signals in the closest spatially adjacent cardinal channels associated with the particular decoding module. As explained in more detail below, the decoding modules are loosely coupled to each other in the sense that modules share nodes and there is a hierarchy of decoding modules. Modules are ordered in the hierarchy according to the number of cardinal channels they are associated with (the module or modules with the highest number of associated cardinal channels is ranked highest). A supervisory routine function presides over the modules so that common node signals are equitably shared and higher-order decoder modules may affect the output of lower-order modules.

[0019] Each decoder module may, in effect, include a matrix such that it directly generates output signals or each decoder module may generate control signals that are used, along with the control signals generated by other decoder modules, to vary the coefficients of a variable matrix or the scale factors of inputs to or outputs from a fixed matrix in order to generate all of the output signals.

[0020] Decoder modules emulate the operation of the human ear to attempt to provide perceptually transparent reproduction. Each decoder module may be implemented as either a wideband or multiband structure or function, in the latter case with either a continuous filterbank, or a block-structure, for example, a transform-based processor, using, for example, the same essential processing in each band.

[0021] Although the basic invention relates generally to the spatial translation of M input channels to N output channels, wherein M and N are positive whole integers and M is at least two, another aspect of this invention is that the quantity of speakers receiving the N output channels can be reduced to a practical number by judicious reliance upon virtual imaging, that is the creation of perceived sonic images at positions in space other than where a loudspeaker is located. The most common use of virtual imaging is in the stereo reproduction of an image part way between two speakers, by panning a mono signal between the channels. Virtual imaging is not considered a viable technique for group presentation with a sparse number of channels, because it requires the listener to be equidistant from the two speakers, or nearly so. In movie theatres, for example, the left and right front speakers are too far apart to obtain useful phantom imaging of a center image to much of the audience, so, given the importance of the center channel as the source of much of the dialog, a physical center speaker is used instead.

[0022] However, as the density of the speakers is increased, a point will be reached where virtual imaging is viable between any pair of speakers for much of the audience, at least to the extent that pans are smooth; with sufficient speakers, the gaps between the speakers are no longer perceived as such. Such an array has the potential to be nearly indistinguishable from the 2 million array derived earlier.

[0023] In order to test aspects of the present invention, we deployed a horizontal array of 5 speakers on each wall, 16 total allowing for common corner speakers, plus a ring of 6 speakers above the listener at a vertical angle of about 45 degrees, plus a single speaker directly above, total 23, plus a subwoofer/LFE channel, total 24, all fed from a PC set up for 24-channel playback. Although by current parlance this system might be referred to as a 23.1 channel system, for simplicity it will be referred to as a 24-channel system herein.

[0024] FIG. 1 is a top plan view showing schematically an idealized decoding arrangement in the manner of the just-described test arrangement. Five wide range horizontal cardinal channels are shown as squares 1', 3', 5', 9' and 13' on the outer circle. A vertical channel, perhaps derived from the five wide range cardinals via correlation or generated reverberation, or separately supplied, is shown as the broken square 23' in the center. The twenty-three wide range output channels are shown as numbered filled circles 1-23. The outer circle of sixteen output channels is on a horizontal plane, the inner circle of six output channels is forty-five degrees above the horizontal plane. Output channel 23 is directly above one or more listeners. Five two-input decoding modules are illustrated as arrows 24-28 around the outer circle, connected between each pair of horizontal cardinal channels. Five additional two-input vertical decoding modules are illustrated as arrows 29-33 connecting the vertical channel to each of the horizontal cardinals. Output channel 21, the

LINKPLAY EXHIBIT 1004
Page 1840 of 4822

elevated center rear channel, is derived from a three-input decoding module illustrated as arrows between output channel 21 and cardinal channels 9, 13 and 23. Thus, each module is associated with a respective pair or trio of closest spatially adjacent cardinal channels. Although the decoding modules represented in FIG. 1 have three, four or five output channels, a decoding module may have any reasonable number of output channels. An output channels may be located intermediate to one or more cardinal channels or at the same position as a cardinal channel. Thus, in the FIG. 1 example, each of the cardinal channel locations is also an output channel. Each input channel is shared by multiple decoding modules.

[0025] As will be discussed, a design goal of this invention is that the playback processor should be capable in concept of working with an arbitrary number and arrangement of speakers, so the 24-channel array will be used as an illustrative but non-unique example of the density and arrangement required to achieve a convincing continuum perceived soundfield according to one aspect of the invention.

[0026] The desire to be able to use a large, and possibly user-selectable, number of presentation channels raises the question of the number of discrete channels, and/or other information, that must be conveyed to the playback processor in order for it to derive, at least as one option, the twenty four channels described above. Obviously, one possible approach is simply to transmit twenty four discrete channels, but aside from the fact that it would likely be onerous for content producers to have to mix that many separate channels, and for a transmission medium to convey as many channels, it is preferred not to do so, as the 24-channel arrangement is merely one of many possible, and it is desired to allow for more or fewer presentation channels from a common transmitted signal array.

[0027] One way to recover output channels is to use formal spatial interpolation, a fixed weighted sum of the transmitted channels for each output, assuming the density of such channels is sufficiently great to allow for that. However, this would require from thousands to millions of transmitted channels, analogous to the use of a multihundred-tap FIR filter to perform temporal interpolation of a single signal. Reduction to a practical number of transmitted channels requires the application of psychoacoustic principles and more aggressive, dynamic interpolation from far fewer channels, still leaving unanswered the question of just how many channels are needed to impart the percept of a complete soundfield.

[0028] This question was addressed by an experiment performed by the present inventor some years ago, and recently replicated by another. The basis for the earlier experiment, at least, was the observation that conventional 2-channel binaural recording is capable of reproducing a realistic left/right image spread, but results in erratic front/back localization, owing in part to the imperfection of any HRTF employed, and the lack of head motion cues. To circumvent this drawback, a dualbinaural (4-channel) recording was made, using two pairs of directional microphones spaced commensurate to the size of the human head. One pair faced forward, the other to the rear. The resulting recording was played over four speakers spaced close to the head, to mitigate acoustic cross coupling effects. This arrangement provided realistic left/right timing and amplitude localization cues from each pair of speakers, plus unambiguous front/back information from the corresponding discrete positions of the microphones and speakers. The result was a singularly compelling surround sound presentation that lacked only a viable representation of height information. A recent experiment of another added a center front channel and two height channels, and was reported to be similarly realistic, perhaps even enhanced by the addition of height information.

[0029] Therefore, from both psychoacoustic considerations and empirical evidence, it appears that the relevant perceptual information can be conveyed in perhaps 4 to 5 "binaural-like" horizontal channels, plus perhaps one or more vertical channels. However, the signal crossfeed characteristic of binaural channel pairs makes them unsuitable for direct playback to a group via loudspeakers, since there is very little separation at midrange and low frequencies. So rather than introducing the crossfeed at the encoder (as is done for a binaural pair) only to have to undo it in the decoder, it is simpler and more direct to keep channels isolated, and to mix output channel signals from the nearest transmitted channels. Not only does this allow for direct playback through a like number of speakers without a decoder, if desired, plus optional downmix to fewer channels with a passive matrix decoder, but it essentially corresponds to the existing standard arrangement of 5.1 channels, at least in the horizontal plane. It is also largely compatible with natural recordings, such as might be made with five real directional microphones, since, allowing for some possible time delay, sounds arriving from intermediate directions will tend to map principally to the nearest microphones (in a horizontal array, specifically to the nearest pair of microphones).

[0030] Thus, from a perceptual standpoint, it should be possible for a channel translation decoder to accept a standard 5.1 channel program and convincingly present it through an arbitrary number of horizontally arrayed speakers, including the sixteen horizontal speakers of the twenty-four-channel array described earlier. With the addition of a vertical channel, such as is sometimes proposed for a digital cinema system, it should be possible to feed the entire twenty-four-channel array with individually derived, perceptually valid signals that together impart a continuum soundfield percept at most listening positions. Of course, if there is access to the fine grain source channels at the encoding site, additional information about them might be used to actively alter the encode matrix scale factors to pre-compensate for decoder limitations, or might simply be included as additional side-chain (auxiliary) information, perhaps similar to the coupling coordinates used in AC-3 (Dolby Digital) multichannel coding, but perceptually, such extra information should not be necessary; and practically, requiring the inclusion of such information is undesirable. The intended operation of the channel translation decoder is not limited to operation with 5.1 channel sources, and may use fewer or more, but there is at least some

LINKPLAY EXHIBIT 1004
Page 1841 of 4822

justification to the belief that credible performance can be obtained from 5.1 channel sources.

[0031] This still leaves unanswered the question of just how to extract the intermediate output channels from a sparse array of transmitted channels. The solution proposed by one aspect of the present invention is to exploit again the notion of virtual imaging, but in a somewhat different way. It was previously noted that virtual imaging is not viable for group presentation with sparse speaker arrays because it required the listener to be nearly equidistant from each speaker. But it will work, after a fashion, for a listener who is fortuitously so placed, allowing the percept of intermediate phantom channels for signals that have been amplitude panned between the nearest real output channels. It is therefore proposed in one aspect of the present invention that the channel translation decoder consist of a series of modular interpolating signal processors, each in effect emulating an optimally placed listener, and each functioning in a manner analogous to the human auditory system to extract what would otherwise be virtual images from amplitude-panned signals, and feed them to real loudspeakers; the speakers preferably arrayed densely enough that natural virtual imaging can fill in the remaining gaps between them.

[0032] In general, each decoding module derives its inputs from the nearest transmitted cardinal channels, which, for example, for a canopy (overhead) array of speakers may be three or more cardinal channels. One way of generating output channels involving more than two cardinal channels might be to employ a series of pair-wise operations, with, e.g., outputs of some pair-wise decoding modules feeding the inputs of other modules. However, this has two drawbacks. One is that cascading decoding modules introduces multiple cascaded time constants, resulting in some output channels responding more quickly than others, causing audible position artifacts. The second drawback is that pair-wise correlation alone can only place intermediate or derived output channels along the line between the pair; use of three or more cardinals removes this restriction. Consequently, an extension to common pair-wise correlation has been developed to correlate three or more output signals; this technique is described below.

[0033] Horizontal localization in the human ear is predicated primarily upon two localization cues: interaural amplitude differences and interaural time differences. The latter cue is only valid for signal pairs in near time alignment, $\pm$ 600 microseconds or so. The practical effect is that phantom intermediate images will only occur at positions corresponding to a particular left/right amplitude difference, assuming the common signal content in the two real channels is correlated, or nearly so. (Note: two signals can have cross correlation values that span from +1 to -1. Fully correlated signals (correlation = 1) have the same waveform and time alignment, but may have different amplitudes, corresponding to off-center image positions.) As the correlation of a signal pair diminishes below 1, the perceived image will tend to spread, until, for two uncorrelated signals, there will be no intermediate image, only separate and distinct left and right images. Negative correlations are usually treated by the ear as similar to uncorrelated signal pairs, although the two images may appear to be spread wider. The correlations are carried out on a critical band basis, and above about 1500 Hz, the critical band signal envelopes are used instead of the signals themselves, to save human computational requirements (MIPS).

[0034] Vertical localization is a little more complex, relying on HRTF pinna cues and dynamic modulation of the horizontal cues with head motion, but the final effect is similar to horizontal localization with respect to panned amplitudes, cross correlation, and corresponding perceived image position and fusion. Vertical spatial resolution is, however, less precise than horizontal resolution, and does not require as dense an array of cardinal channels for adequate interpolation performance.

[0035] An advantage of using directional processors that emulate the operation of the human ear is that any imperfections or limitations of the signal processing should be perceptually masked by like imperfections and limitations of the human ear, allowing for the possibility that the system will be perceived as nearly indistinguishable from the original full continuum presentation.

[0036] Although the present invention is designed to make effective use of however many or few output channels are available (including playback via as many loudspeakers as there are input channels with no decoding, and passive mixdown to fewer channels, including mono, stereo and surround compatible Lt/Rt), it is preferably intended to employ a large and somewhat arbitrary, but nonetheless practical number of presentation channels/loudspeakers, and use as source material a similar or smaller number of encoded channels, including existing 5.1 channel surround tracks, and possible next-generation 11- or 12-channel digital cinema soundtracks.

[0037] Implementations of the present invention desirably should exhibit four principles: error containment, dominant containment, constant power, and synchronized smoothing.

[0038] Error containment refers to the notion that, given the likelihood of decoding errors, the decoded position of each source should be in some reasonable sense near its true, intended direction. This mandates a certain degree of conservatism in decoding strategy. Faced with the prospect of more aggressive decoding accompanied by possibly greater spatial disparity in the event of errors, it is usually preferable to accept less precise decoding in exchange for assured spatial containment. Even in situations in which more precise decoding can confidently be applied, it may be unwise to do so if there is a likelihood that dynamic signal conditions will require the decoder to ratchet between aggressive and conservative modes, resulting in audible artifacts.

[0039] Dominant containment, a more constrained variant of error containment, is the requirement that a single well-defined dominant signal should be panned by the decoder to only nearest neighbor output channels. This condition is

LINKPLAY EXHIBIT 1004
Page 1842 of 4822

necessary to maintain image fusion for dominant signals, and contributes to the perceived discreteness of a matrix decoder. While a signal is dominant, it is suppressed from other output channels, either by subtracting it from the associated cardinal signals, or by directly applying to other output channels matrix coefficients complementary to those used to derive the dominant signal ("anti-dominant coefficients/signal").

[0040] Constant power decoding requires not only that the total decoded output power be equal to the input power, but also equates the input/output power of each channel and directional signal encoded in the conveyed cardinal array. This minimizes gain-pumping artifacts.

[0041] Synchronized smoothing applies to systems with signal dependent smoothing time constants, and requires that if any smoothing network within a decoding module is switched to a fast time constant mode, all other smoothing networks within the module be similarly switched. This is to avoid having a newly dominant directional signal appear to slowly fade/pan from the previous dominant direction.

DESCRIPTION OF THE DRAWINGS

[0042]

FIG. 1 is a schematic drawing showing a top plan view of an idealized decoder arrangement.

BEST MODE FOR CARRYING OUT THE INVENTION

Decoding Module

[0043] Because encoding any source direction is assumed to map primarily to the nearest cardinal channels, channel translation decoding is based on a series of semi-autonomous decoding modules which in a general sense recover output channels, particularly intermediate output channels, each usually from a subset of all the transmitted channels, in a fashion similar to the human ear.

[0044] In a fashion analogous to the human ear, the operation of the decoding module is based on a combination of amplitude ratios, to determine the nominal ongoing primary direction, and cross correlation, to determine the relative width of the image.

[0045] Using control information derived from the amplitude ratios and cross correlation, the processor then extracts output channel audio signals. Since this is best done on a linear basis, to avoid generation of distortion products, the decoder forms weighted sums of cardinal channels containing the signal of interest. (As explained below, it may also be desirable to include information about non-neighbor cardinals in the calculation of the weighted sum.) This limited but dynamic form of interpolation is more commonly referred to as matrixing. If, in the source, the desired signal is mapped (amplitude panned) to the nearest M cardinal channels, then the problem is one of M:N matrix decoding. In other words, the output channels represent relative proportions of the input channels.

[0046] Especially in the case of two-input decoding modules, this is much like the issue addressed by active 2:N matrix decoders, such as the now classic Dolby Pro Logic matrix decoder, with pairwise decoding module inputs corresponding to the Lt/Rt encoded signals

[0047] Note: The outputs of a 2:N matrix decoder are sometimes referred to as cardinal channels. This document, however, uses "cardinal" to refer to the input channels of the channel translation decoder.

[0048] There is, however, at least one significant difference between prior art active 2:N decoders and the operation of a decoding module according to the present invention. While the former use left/right amplitudes to indicate left/right position, as postulated as well for the channel translation decoder, they also use interchannel phase to indicate front/back position, relying specifically on the ratio of sum/difference of the Lt/Rt encoded channels.

[0049] There are two problems with such active 2:N decoder arrangements. One is that fully correlated (frontal), but off-center signals, for example, will result in a sum/difference ratio of less than infinity, incorrectly indicating a less-than-full-frontal position (similarly for full anti-correlated off-center rear signals). The result is a somewhat warped decoding space. The second drawback is that the positional mapping is many-to-one, introducing inherent decoding errors. For example, in a 4:2:4 matrix system, an uncorrelated Left-In and Right-In signal pair with no Front-In or Rear-In will map to the same net, uncorrelated Lt/Rt pair as will an uncorrelated Front-InBack-In pair, with no Left-In/Right-In, or for that matter from all four inputs uncorrelated. The decoder, faced with an uncorrelated Lt/Rt signal pair, has no choice but to "relax the matrix", that is use a passive matrix that distributes sound to all output channels. It is incapable of decoding to a simultaneous Left-Out/Right-Out only, or Front-Out/Rear-Out only signal array.

[0050] The underlying problem is that the use of interchannel phase to code front/back position in N:2:N matrixing systems runs counter to the operation of the human ear, which does not use phase to judge front/back position. The present invention works best with at least three non-collinear cardinal channels, so that front/back position is indicated by the assumed directions of the cardinal channels, without assigning different directions depending on their relative

LINKPLAY EXHIBIT 1004
Page 1843 of 4822

phases or polarities. As such, a pair of uncorrelated or anti-correlated channel translation cardinal signals unambiguously decodes to isolated cardinal-output channel signals, with no intermediate signal and no "rearward" direction indicated. (This, by the way, avoids the unfortunate "center pileup" effect in active 2:N decoders, in which uncorrelated Left-In and Right-In signals are presented with reduced separation because the decoder feeds sum and difference of these signals to center and surround channels.) Of course, it is possible in principle to spatially expand a Lt/Rt signal pair by cascading a 2:N decoder, N = 4, or 5, with an N:M channel translation system, but in that case any limitations of the 2:N decoder, such as center pileup, will be carried over to the channel multiplied outputs. It is also possible to combine these functions into a channel translation decoder configured to accept 2-channel Lt/Rt signals and, in such cases, modify its behavior to interpret negative correlation signals as having rearward orientation, leaving the rest of the processing largely intact. However, even in that case, decoding ambiguities resulting from having only two transmitted channels would remain.

[0051] Thus, each decoding module, especially those with two input channels, resembles a prior art active 2:N decoder, with the front/back detection disabled or modified, and an arbitrary number of output channels. Of course, it is a mathematical impossibility to use matrixing to uniquely extract a larger number of channels from a smaller number, as this basically involves N linear equations with M unknowns, M greater than N. Therefore, it is to be expected that the decoding module may at times exhibit less than perfect channel recovery in the presence of multiple active source direction signals. However, the human auditory system, limited to using just two ears, will tend to be subject to the same limitations, allowing the system to be perceived as discrete, even with all channels operating. Isolated channel quality, with other channels muted, is still a consideration to accommodate listeners that may be situated near one speaker.

[0052] To be sure, the ear is operating on a frequency-dependent basis, but given that most sonic images will be similarly correlated at all frequencies, together with the successful empirical experience with Pro Logic decoders as a wideband system, it is to be expected that a wideband channel translation system may also be capable of satisfactory performance in some applications. Multiband channel translation decoding should also be possible, using similar processing on a band-by-band basis, and using the same encoded signal in each case, so the number and bandwidth of individual bands can be left as a free parameter to the decoder implementer. Although multiband processing is likely to require higher MIPS than wideband processing, the computational demands may not be that much higher if the input signals are divided into data blocks and the process is carried out on a block basis.

[0053] Before describing an algorithm usable by the decoding modules of the present invention, consideration is first given to the problem of shared nodes.

Shared Nodes

[0054] If the cardinal channel groups used by the decoding modules were all independent, then the decoding modules themselves could be independent, autonomous entities. Such is, however, not usually the case. A given transmitted channel will in general share separate output signals with two or more neighboring cardinal channels. If independent decoding modules are used to decode the array, each will be influenced by output signals of neighboring channels, resulting in possibly serious decoding errors. In effect, two output signals of neighboring decoding modules will "pull", or gravitate, toward each other, because of the increased level of the common cardinal node containing both signals. If, as is likely to be the case, the signals are dynamic, so too will be the amount of interaction, leading to signal dependent dynamic positioning errors of a possibly highly objectionable nature. This problem does not arise with Pro Logic and other active 2:N decoding, since they use only a single, isolated channel pair as the decoder input.

[0055] Thus, it is necessary to compensate for the "shared node" effect. One possible way to do so would be to subtract one recovered signal from the common node before trying to recover the output signal of an adjacent decoding module sharing the common node. This is often not possible, so as a fall back, each decoding module estimates the amount of common output signal energy present at its input channels, and a supervisor routine then informs each module of its neighbors' output signal energy estimates.

Pair-wise calculation of common energy

[0056] For example, suppose cardinal channel pair A/B contains a common signal X along with individual, uncorrelated signals Y and Z:

$$A = 0.707X + Y$$

$$B = 0.707X + Z$$

LINKPLAY EXHIBIT 1004
Page 1844 of 4822

where the scalefactors of $0.707 = \sqrt{0.5}$ provide a power preserving mapping to the nearest neighbor cardinal channels.

$$RMS\,Energy(A) = \int A^2 \partial t = \overline{A^2} = \overline{(.707X + Y)^2} = \overline{(0.5X^2 + 0.707XY + Y^2)}$$

$$= 0.5\overline{X^2} + 0.707\overline{XY} + \overline{Y^2}$$

Because X and Y are uncorrelated,

$$\overline{XY} = 0$$

So:

$$\overline{A^2} = 0.5\overline{X^2} + \overline{Y^2}$$

i.e., Because X and Y are uncorrelated, the total energy in cardinal channel A is the sum of the energies of signals X and Y. Similarly:

$$\overline{B^2} = 0.5\overline{X^2} + \overline{Z^2}$$

Since X, Y, and Z are uncorrelated, the averaged cross-product of A and B is:

$$\overline{AB} = 0.5\overline{X^2}$$

[0057] So, in the case of an output signal shared equally by two neighboring cardinal channels which may also contain independent, uncorrelated signals, the averaged cross-product of the signals is equal to the energy of the common signal component in each channel. If the common signal is not shared equally, *i.e.,* it is panned toward one of the cardinals, the averaged cross-product will be the geometric mean between the energy of the common components in A and B, from which individual channel common energy estimates can be derived by normalizing by the square root of the ratio of the channel amplitudes. Actual time averages are computed with a leaky integrator having a suitable decay time constant, to reflect ongoing activity. The time constant smoothing can be elaborated with nonlinear attack and decay time options, and in a multiband system, may be scaled with frequency.

Higher order calculation of common energy

[0058] In order to derive the common energy of decoding modules with three or more inputs, it is necessary to form averaged cross-products of all the input signals. Simply performing pairwise processing of the inputs will fail to differentiate between separate output signals between each pair of inputs and a signal common to all.

[0059] Consider, for example, three cardinal channels, A, B, and C, made up of uncorrelated signals W, Y, Z, and common signal X:

$$A = X + W$$

LINKPLAY EXHIBIT 1004
Page 1845 of 4822

$$B = X + Y$$

$$C = X + Z$$

[0060] If the average cross-product is calculated, all terms involving combinations of W, Y, and Z will cancel, as in the second order calculation, leaving the average of $_X{}^3$:

$$\overline{ABC} = \overline{X^3}$$

[0061] Unfortunately, if X is a zero mean time signal, as expected, then the average of its cube is zero. Unlike averaging X^2, which is positive for any nonzero value of X, X^3 has the same sign as X, so the positive and negative contributions will tend to cancel. Obviously, the same holds for any odd power of X, corresponding to an odd number of module inputs, but even exponents greater than 2 can also lead to erroneous results; for example four inputs with components (X, X, -X, -X) will have the same product/average as (X, X, X, X).

[0062] This problem has been resolved by employing a variant of the averaged product technique. Before being averaged, the sign of the each product is discarded by taking the absolute value of the product. The signs of each term of the product are examined. If they are all the same, the absolute value of the product is applied to the averager. If any of the signs are different from the others, the negative of the absolute value of the product is averaged. Since the number of possible same-sign combinations may not be the same as the number of possible different-sign combinations, a weighting factor comprised of the ratio of the number of same to different sign combinations is applied to the negated absolute value products to compensate. For example, a three-input module has two ways for the signs to be the same, out of eight possibilities, leaving six possible ways for the signs to be different, resulting in a scale factor of 2/6 = 1/3. This compensation causes the integrated or summed product to grow in a positive direction if and only if there is a signal component common to all inputs of a decoding module.

[0063] However, in order for the averages of different order modules to be comparable, they must all have the same dimensions. A conventional second-order correlation involves averages of two-input multiplications and hence of quantities with the dimensions of energy or power. Thus the terms to be averaged in higher order correlations must be modified also to have the dimensions of power. For a kth order correlation, the individual product absolute values must therefore be raised to the power 2/k before being averaged.

[0064] Of course, regardless of the order, the individual input node energies of a module, if needed, can be calculated as the average of the square of the corresponding node signal, and need not be first raised to the kth power and then reduced to a second order quantity.

Shared nodes: Neighbor levels

[0065] By using averaged squares and modified cross-products of cardinal channel signals, the amount of common output channel signal energy can be estimated. The above example involved a single interpolation processor, but if one or more of the A/B(/C) nodes were common to another module with its own common signal component, uncorrelated with any other signals, then the averaged cross-product computed above would not be affected, making the calculation inherently free of any image pulling effects. (Note: if the two output signals are not uncorrelated, they will tend to pull the decoders some, but should have a similar effect on the human ear, so again system operation should remain faithful to human audition.)

[0066] Once each decoding module has computed the estimated common output channel signal energy at each of its cardinal nodes, the supervisor routine function can inform neighboring modules of each others' common energy, at which point the extraction of the output channel signals can proceed as described below. The calculation of the common energy used by a module at a node must take into account the hierarchy of possibly overlapping modules of different order, and subtract the common energy of a higher order module from the estimated common energy of any lower order module sharing the same nodes.

[0067] For example, suppose there are two adjacent cardinal channels A and B, representing two horizontal directions, plus a cardinal channel C, representing a vertical direction, and further suppose the existence of an intermediate or derived output channel representing an interior direction (i.e. one within the limits of A, B and C), with signal energy X^2. The common energy of a three-input module, with inputs (A, B, C), will be X^2, but so will the common energy of two-

LINKPLAY EXHIBIT 1004
Page 1846 of 4822

input modules (A, B), (B, C), and (A, C). If the common energy of A-connected modules (A, B, C), (A, B), and (A, C) is simply added, the result is $3X^2$, instead of X^2. In order for the calculation of common node energy to be correct, the common energy of each higher order module is first subtracted from the estimate of the common energy of each overlapping lower-level module, so the common energy X^2 of higher order module (A, B, C) is subtracted from the common energy estimates of the two two-input modules, resulting in 0 in each case, and making the net common energy estimate at node A equal to $X^2 + 0 + 0 = X^2$.

Output Channel Signal Extraction

[0068] As has been noted, the process of recovering the ensemble of output channels from the transmitted channels in a linear fashion is basically one of matrixing, that is forming weighted sums of the cardinal channels to derive output channel signals. The optimal choice of matrix scale factors is generally signal dependent. Indeed, if the number of currently active output channels is equal to the number of transmitted channels (but representing different directions), making the system exactly constrained, it is mathematically possible to compute an exact inverse of the effective encoding matrix, and recover isolated versions of the source signals. Even if the number of active output channels is greater than the number of cardinals, it may still be possible to compute a matrix pseudo-inverse.

[0069] Unfortunately, there are problems with this approach, not the least of which is that it is computationally demanding, especially on a multiband basis, and oriented toward high accuracy floating point implementation. Even though intermediate signals are assumed to be panned to nearest neighbor cardinal channels, a mathematical inverse or pseudo-inverse of the effective encoding matrix will in general involve contributions from all cardinal channels to each output channel, because of the node sharing effect. If there are any imperfections in the decoding, as indeed there inevitably will be, a cardinal channel signal could be reproduced from an output channel far removed from it spatially, which is highly undesirable. In addition, pseudo-inverse calculations tend to produce minimum-RMS-energy solutions, which maximally spread the sound around, providing minimum separation; this is quite the opposite of the intention.

[0070] So, in order to implement a practical, fault-tolerant decoder in which spatial decoding errors are inherently contained, the same modular structure as was used for signal detection is employed for signal extraction.

[0071] Following are details of the extraction process by which output signals are recovered by a decoding module. Note that the effective position of each output channel connected to the module is assumed to be indicated by the amplitude ratio that would otherwise be needed to pan a signal to that physical location, i.e., the ratio of the effective matrix encoding coefficients corresponding to that direction. To avoid divide-by-zero problems, ratios are typically calculated as the quotient of one channel's matrix coefficient over the RMS sum of all of that input channels' matrix coefficients (usually 1). For example, in a two-input module with inputs L and R, the energy ratio used would be the L energy over the sum of the L and R energies ("L-ratio"), which has a well-behaved range of 0 to 1. If the two-input decoding module has five output channels with effective encoding matrix coefficient pairs of (1.0, 0), (0.89, 0.45), (0.71, 0.71), (0.45, 0.89) and (0, 1.0), the corresponding L-ratios are 1.0, 0.89, 0.71, 0.45, and 0, since each scale factor pair has an RMS sum of 1.0.

[0072] From the signal energy of each input node (cardinal channel) of the decoding module is subtracted any node-sharing signal energy claimed by neighboring decoding modules, resulting in normalized input signal power levels used for the remainder of the calculation.

[0073] The dominant direction indicator is calculated as the vector sum of the cardinal directions, weighted by the relative energy. For a two input module, this simplifies to being the L-ratio of the normalized input signal power levels.

[0074] The output channels bracketing the dominant direction are determined by comparing the dominant direction L-ratio of step two, to the L-ratios of the output channels. For example, if the L-ratio of the above five-output-decoding-module inputs is 0.75, the second and third output channels bracket dominant signal direction, since $0.89 > 0.75 > 0.71$.

[0075] Panning scale factors to map the dominant signal onto the nearest bracketing channels are calculated from the ratio of the anti-dominant signal levels of the channels. The anti-dominant signal associated with a particular output channel is the signal that results when the corresponding decoding module's input signals are matrixed with the output channel's anti-dominant matrix scale factors. An output channel's anti-dominant matrix scale factors are those scale factors with RMS sum = 1.0 which result in zero output when a single dominant signal is panned to the output channel in question. If an output channel's encode matrix scale factors are (A, B), then the anti-dominant scale factors of the channel are just (B, -A).

Proof

[0076] If a single dominant signal is panned to an output channel with encode scale factors (A, B), then the signal must have amplitudes (kA, kB), k the overall amplitude of the signal. Then the anti-dominant signal for that channel is (kA * B - kB*A)=0.

[0077] So, if a dominant signal consists of two-input module input signals (x(t), y(t)) with input amplitudes normalized to RMS=1 (X, Y), the extracted dominant signal will be dom(t) = Xx(t) + Yy(t). If the position of this signal is bracketed

LINKPLAY EXHIBIT 1004
Page 1847 of 4822

by output channels having matrix scale factors (A, B) and (C, D) respectively, the dominant signal scale factor scaling dom(t) for the former channel will be:

$$SF(A,B) = sqrt\ ((DX - CY) / ((DX - CY) + (BX - AY))),$$

while the equivalent dominant signal scale factor for the latter channel will be:

$$SF(C,D) = sqrt\ ((BX - AY) / ((DX - CY) + (BX - AY))).$$

[0078] As the dominant direction is panned from one output channel to the other, these two scale factors move in opposite directions between zero and one with constant power sum.

[0079] The anti-dominant signal is calculated and panned with suitable gain scaling to all non-dominant channels. The anti-dominant signal is a matrixed signal lacking any of the dominant signal. If the inputs to a decoding module are (x (t), y(t)) with normalized amplitudes (X, Y), the dominant signal is Xx(t)+Yy(t) and the anti-dominant signal is Yx(t)-Xy (t), irrespective of the positions of the non-dominant output channels.

[0080] In addition to the dominant/anti-dominant signal distribution, a second signal distribution is calculated, using the "passive" matrix, which is basically the output channel matrix scale factors already discussed, scaled to preserve power.

[0081] The cross correlation of the decoding module input signals is calculated as the averaged cross-product of the input signals divided by the square root of the product of the normalized input levels.

[0082] Returning to details of the extraction process, the final output signals are then calculated as a weighted crossfade sum of the dominant and passive signal distributions, using the decoding module's input signal cross-correlation to derive the crossfade factor. For correlation=1, the dominant/anti-dominant distribution is used exclusively. As the correlation diminishes, the output signal array is broadened by cross-fading to the passive distribution, reaching completion at a low positive value of correlation, typically 0.2 to 0.4, depending on the number of output channels connected to the decoding module. As the correlation falls further, toward zero, the passive amplitude output distribution is progressively bowed outward, reducing the output channel levels, emulating the response of the human ear to such signals.

Vertical Processing

[0083] Most of the processing described so far applies to the extraction of output channel signals from neighboring cardinal channels, regardless of the direction of the output and cardinal channels. However, because of the horizontal orientation of the ears, human auditory localization tends to be less sensitive to interchannel correlation in the vertical direction than horizontally. To remain faithful to the operation of the human ear, it may be desirable to relax the correlation constraint in interpolation processors using vertically-oriented input channels, such as processing the correlation signal with a warping function before otherwise applying it. However, it may be that use of the same processing as for horizontal channels will not involve any audible penalty, which will simplify the structure of the overall decoder.

[0084] Strictly speaking, vertical information includes both sound from above and below, and the decoder structure described will work equally well with either, but in practice there is little natural sound normally perceived as coming from below, so such processing and channels can probably be omitted without seriously compromising the perceived spatial fidelity of the system.

[0085] That notion may have practical significance in the application of channel translation to existing 5.1 channel surround material, which, of course, lacks any vertical channel. However, it may contain vertical information, such as fly-overs, which are panned across many or all of the horizontal channels. Thus, it should be possible to extract a virtual vertical channel from such source material, by looking for correlations among non-neighboring channels or groups of channels. Where such correlations exist, they will usually indicate the presence of vertical information from above, rather than below the listener. In some instances, it may also be possible to derive virtual vertical information from a reverberation generator, perhaps keyed to a model of the intended listening environment. Once the virtual vertical channel is extracted or derived from the 5.1-channel source, the expansion to larger numbers of channels, such as the 24-channel arrangement described earlier, can proceed as if a real vertical channel had been supplied.

LINKPLAY EXHIBIT 1004
Page 1848 of 4822

Directional Memory

[0086] One respect in which the operation of the decoding module control generation described above is similar to a 2:N active decoder such as a Pro Logic decoder is that the only "memory" in the process is in the smoothing networks, which derive the basic control signals. At any one point in time, there is only one dominant direction and one value of input correlation; and signal extraction proceeds directly from these signals.

[0087] However, particularly in complex acoustical environments (like the archetypal cocktail party), the human ear exhibits a certain degree of positional memory, or inertia, in that a briefly dominant sound from a given direction that is clearly localized will result in other, less distinctly localizable sounds from that general direction to be perceived as coming from the same source.

[0088] It is possible to emulate this effect in the decoding modules (and, indeed, in Pro Logic decoding as well) by adding an explicit mechanism to keep track of recently dominant directions and, during intervals of directionally ambiguous signal conditions, weight the output signal distribution toward recently dominant directions. This can enhance the perceived reproduced discreteness and stability of complex signal arrays.

Modified Correlation and Selective Channel Mixing

[0089] As described, the spreading determination of each decoding module is based on the coincident cross correlation of its input signals. This may underestimate the amount of output signal content under some conditions. This will occur, for example, with a naturally recorded signal in which non-centered directions have slightly different arrival times, along with unequal amplitudes, resulting in a reduced correlation value. The effect may be exaggerated if wide-spaced microphones are used, with commensurately elongated interchannel delays. To compensate, the correlation calculation can be extended to cover a range of interchannel time delays, at the expense of slightly higher processing MIPS requirements. Also, since the neurons on the auditory nerve have an effective time constant of about 1 msec., more realistic correlation values may be obtained by first smoothing the rectified audio with a smoother having a 1 msec. time constant.

[0090] In addition, if a content producer has an existing 5.1 channel program with strongly uncorrelated channels, the evenness of the spread when processed with a channel translation decoder can be increased by slightly mixing adjacent channels, thereby increasing the correlation, which will cause the channel translation decoding module to provide a more even spread among its intermediate output channels. Such mixing can be done selectively, for example leaving the center front channel signal unmixed, to preserve the compactness of the dialog track.

Loudness Compression/Expansion

[0091] When the encoding process involves mixing a larger number of channels to a smaller number, there is a potential for clipping of the encoded signal if some form of gain compensation is not provided. This problem exists as well for conventional matrix encoding, but is potentially of greater significance for channel translation, because the number of channels being mixed to a given output channel is greater. To avoid clipping in such cases, an overall gain scale factor is derived by the encoder and conveyed in the encoded bitstream to the decoder. Normally, this value is 0 dB, but can be set to a nonzero attenuating value by the encoder to avoid clipping, with the decoder providing an equivalent amount of compensating gain.

[0092] If the decoder is used to process an existing multichannel that lacks such a scale factor program (e.g., an existing 5.1 channel soundtrack), it could optionally use a fixed scale factor with an assumed value (presumably 0 dB), or apply an expansion function based on signal level and/or dynamics, or possibly make use of available metadata, such as a dialog normalization value, to adjust the decoder gain.

[0093] The present invention and its various aspects may be implemented in analog circuitry, or more probably as software functions performed in digital signal processors, programmed general-purpose digital computers, and/or special purpose digital computers. Interfaces between analog and digital signal streams may be performed in appropriate hardware and/or as functions in software and/or firmware.

Claims

1. A process for translating M audio input channels (1', 3', 5', 9', 13') representing a soundfield to N audio output channels (1-23) representing the same soundfield, wherein each channel is a single audio stream representing audio arriving from a direction, M and N are positive whole integers, and M is a positive integer equal to two or more, **characterized by** comprising
   a plurality of decoding modules each associated with two or more spatially adjacent input channels (1', 3', 5', 9', 13'), wherein each input channel is shared among multiple modules, and each module either

LINKPLAY EXHIBIT 1004
Page 1849 of 4822

- includes a matrix that generates, from the associated two or more input channels (1', 3', 5', 9', 13'), one or more output channels (1-23) each constituting a subset of said N channels, by a process that includes determining a measure of the correlation of the two or more input channels (1', 3', 5', 9', 13') and the level interrelationships of the two or more input channels (1', 3', 5', 9', 13'), or
- generates, from the associated two or more input channels (1', 3', 5', 9', 13') by a process that includes determining a measure of the correlation of the two or more input channels (1', 3', 5', 9', 13') and the level interrelationships of the two or more input channels (1', 3', 5', 9', 13'), control signals that are used, along with control signals generated by other decoder modules, to vary the coefficients of a variable matrix to generate all of the output channels (1-23), or
- generates, from the associated two or more input channels (1', 3', 5', 9', 13') by a process that includes determining a measure of the correlation of the two or more input channels (1', 3', 5', 9', 13') and the level interrelationships of the two or more input channels (1', 3', 5', 9', 13'), control signals that are used, along with control signals generated by other decoder modules to vary the scale factors of inputs to or outputs from a fixed matrix to generate all of the output channels (1-23).

2. The process according to claim 1 wherein the modules are hierarchically ordered according to their number of input channels (1', 3', 5', 9', 13') and a supervisor communicates with the modules to control the sharing of input signals in accordance with their hierarchical ordering.

Patentansprüche

1. Verfahren zur Umsetzung von M Audioeingangskanälen (1', 3', 5', 9', 13'), die ein Schallfeld darstellen, auf N Audioausgangskanäle (1-23), die dasselbe Schallfeld darstellen, wobei jeder Kanal ein einzelner Audiostrom ist, der aus einer Richtung ankommendes Audio darstellt, M und N positive ganze Zahlen sind und M eine positive ganze Zahl gleich zwei oder mehr ist, **dadurch gekennzeichnet, dass** es enthält:

eine Vielzahl von Decodiermodulen, von denen jedes mit zwei oder mehr räumlich benachbarten Eingangskanälen (1', 3', 5', 9', 13') verknüpft ist, wobei jeder Eingangskanal von mehreren Modulen gemeinsam benutzt wird, und jedes Modul entweder

- eine Matrix einschließt, die aus den verknüpften zwei oder mehr Eingangskanälen (1', 3', 5', 9', 13') einen oder mehr Ausgangskanäle (1-23), von denen jeder eine Untergruppe der N Kanäle darstellt, durch ein Verfahren erzeugt, das die Bestimmung eines Maßes der Korrelation der zwei oder mehr Eingangskanäle (1', 3', 5', 9', 13') und der Pegelzusammenhänge der zwei oder mehr Eingangskanäle (1', 3', 5', 9', 13') einschließt, oder
- aus den verknüpften zwei oder mehr Eingangskanälen (1', 3', 5', 9', 13') durch ein Verfahren, das die Bestimmung eines Maßes der Korrelation der zwei oder mehr Eingangskanäle (1', 3', 5', 9', 13') und der Pegelzusammenhänge der zwei oder mehr Eingangskanäle (1', 3', 5', 9', 13') einschließt, Steuersignale erzeugt, die zusammen mit von anderen Decodiermodulen erzeugten Steuersignalen benutzt werden, um die Koeffizienten einer variablen Matrix zu verändern, um die Gesamtheit der Ausgangskanäle (1-23) zu erzeugen, oder
- aus den verknüpften zwei oder mehr Eingangskanälen (1', 3', 5', 9', 13') durch ein Verfahren, das die Bestimmung eines Maßes der Korrelation der zwei oder mehr Eingangskanäle (1', 3', 5', 9', 13') und der Pegelzusammenhänge der zwei oder mehr Eingangskanäle (1', 3', 5', 9', 13') einschließt, Steuersignale erzeugt, die zusammen mit von anderen Decodiermodulen erzeugten Steuersignalen benutzt werden, um die Skalierungsfaktoren der Eingaben in oder der Ausgaben aus einer festen Matrix zu verändern, um die Gesamtheit der Ausgangskanäle (1-23) zu erzeugen.

2. Verfahren nach Anspruch 1, bei welchem die Module hierarchisch nach ihrer Anzahl von Eingangskanälen (1', 3', 5', 9', 13') geordnet sind und eine Kontrollinstanz mit den Modulen kommuniziert, um die gemeinsame Benutzung von Eingangssignalen entsprechend ihrer hierarchischen Ordnung zu steuern.

Revendications

1. Procédé permettant de convertir M canaux d'entrée audio (1', 3', 5', 9', 13') représentant un champ acoustique en N canaux de sortie audio (1-23) représentant le même champ acoustique, dans lequel chaque canal est un flux

LINKPLAY EXHIBIT 1004
Page 1850 of 4822

audio unique représentant des sons provenant d'une direction, M et N sont des entiers positifs et M est un entier positif égal à deux ou plus, **caractérisé en ce qu'**il comprend :

une pluralité de modules de décodage chacun étant associé à deux canaux d'entrée adjacents dans l'espace ou plus (1', 3', 5', 9', 13'), dans lequel chaque canal d'entrée est partagé entre de multiples modules, et chaque module :

- comporte une matrice qui génère, à partir des deux canaux d'entrée associés ou plus (1', 3', 5', 9', 13'), un canal de sortie ou plus (1-23) chacun constituant un sous-ensemble desdits N canaux, par un procédé qui comporte la détermination d'une mesure de la corrélation entre les deux canaux d'entrée ou plus (1', 3', 5', 9', 13') et le niveau d'interrelation entre les deux canaux d'entrée ou plus (1', 3', 5', 9', 13'), ou
- génère à partir des deux canaux d'entrée associés ou plus (1', 3', 5', 9', 13'), par un procédé qui comporte la détermination d'une mesure de la corrélation entre les deux canaux d'entrée ou plus (1', 3', 5', 9', 13') et le niveau d'interrelation entre les deux canaux d'entrée ou plus (1', 3', 5', 9', 13') des signaux de commande qui sont utilisés, avec des signaux de commande générés par d'autres modules de décodeur, afin de modifier les coefficients d'une matrice variable pour générer tous les canaux de sortie (1-23), ou
- génère, à partir des deux canaux d'entrée ou plus (1', 3', 5', 9', 13') par un procédé qui comporte la détermination d'une mesure de la corrélation entre les deux canaux d'entrée ou plus (1', 3', 5', 9', 13') et le niveau d'interrelation entre les deux canaux d'entrée ou plus des signaux de commande qui sont utilisés, avec des signaux de commande générés par d'autres modules décodeurs pour modifier les facteurs d'échelle des entrées et/ou des sorties d'une matrice fixe pour générer tous les canaux de sortie (1-23).

2. Procédé selon la revendication 1, dans lequel les modules sont classés par ordre hiérarchique en fonction de leur nombre de canaux d'entrée et un superviseur communique avec les modules pour commander le partage des signaux d'entrée en fonction de leur ordre hiérarchique.

LINKPLAY EXHIBIT 1004
Page 1851 of 4822

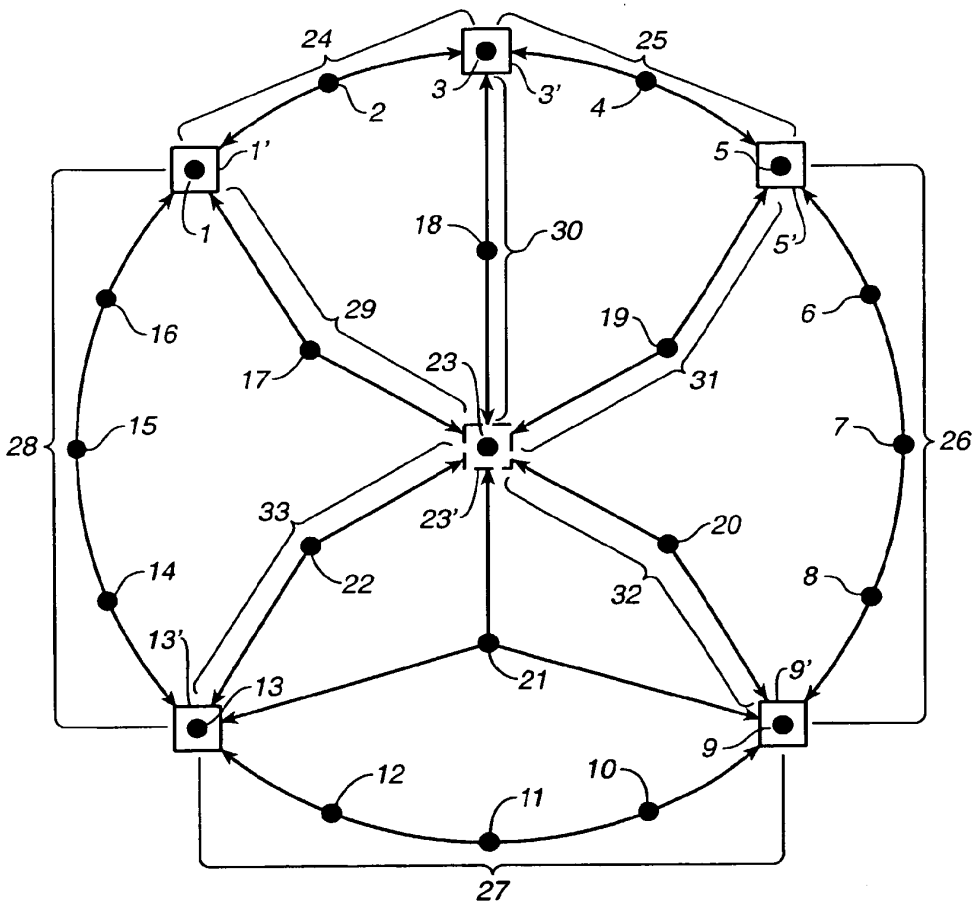

24
25
3
3'
2
4
1'
5
1
18
30
5'
16
29
19
6
17
23
31
28
15
7
26
33
23'
20
14
22
32
8
13'
21
9'
13
9
12
10
11
27

FIG._1

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 4799260 A **[0005]**
- EP 1054575 A **[0011]**
- EP 1001549 A **[0012]**

LINKPLAY EXHIBIT 1004
Page 1853 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 416 687 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
02.08.2006 Bulletin 2006/31

(51) Int Cl.:
H04L 12/56 *(2006.01)* ***H04L 12/40*** *(2006.01)*

(21) Application number: **03256829.7**

(22) Date of filing: **29.10.2003**

(54) **Communication protocol for controlling transfer of temporal data over a bus between devices in synchronization with a periodic reference signal**

Kommunikationsprotokoll für die Übertragungssteuerung von zeitlichen Daten über einen Bus zwischen Geräten in Synchronisation mit einem periodischen Referenzsignal

Protocole de communication pour contrôler le transfert de données temporelles sur un bus entre dispositifs en synchronisme avec un signal de référence périodique

(84) Designated Contracting States:
DE FR GB NL

(30) Priority: **01.11.2002 US 286215**

(43) Date of publication of application:
06.05.2004 Bulletin 2004/19

(73) Proprietor: **AVID TECHNOLOGY, INC.**
Tewksbury, MA 01876 (US)

(72) Inventors:
- **Wallace, Ron**
  Lexington, Massachusetts 02420 (US)
- **Der, Harry**
  Westford, Massachusetts 01886 (US)
- **Corbett, Martin**
  Westford, Massachusetts 01886 (US)
- **Fetters, Terrence**
  Tewksbury, Massachusetts 01876 (US)

(74) Representative: **Kazi, Ilya et al**
Mathys & Squire
120 Holborn
London EC1N 2SQ (GB)

(56) References cited:
WO-A-00/60478
US-A- 5 915 130
US-A1- 2002 141 439
US-B1- 6 418 150

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

EP 1 416 687 B1

Description

[0001] Peripheral devices that process video and audio data commonly are connected to a host computer either to perform functions not performed by the host computer or to convert the video and audio data between digital and analog formats. The connection between the peripheral device and the host computer is typically provided by a bus over which communication is performed using packets. Example types of a bus include, a parallel bus, such as a peripheral component interconnect (PCI) bus, a serial bus, such as an IEEE-1394 compliant bus, or a computer network, such as Ethernet.

[0002] Video and audio data are typically in a standard format, such as the format described by "Specifications of Consumer-Use Digital VCR's using 6.3mm magnetic tape" of the HD Digital VCR Conference dated December 1994, also called the "Blue Book," or by SMPTE 314M-1999 or by IEC-61834, which describe video and audio data in a format that is commonly called "DV." Video data may be compressed or uncompressed. Audio data typically is uncompressed.

[0003] An example peripheral device that processes video and audio data is a transcoder. A transcoder typically receives compressed digital video data and audio data, such as DV, over a serial bus, such as an IEEE-1394 compliant bus, and processes the data for output. In particular, the video data is converted into an analog video signal for output to a video device, such as a video monitor or camcorder. The audio data is converted into an audio signal for output to an audio device, such as a set of speakers. Such a transcoder also may receive an input analog video signal and audio signal and generate compressed digital video data and audio data, such as DV, that is transferred to the host computer over a serial bus, such as an IEEE-1394 compliant bus.

[0004] Many systems currently are available that transfer video and audio data between a peripheral device, such as a camcorder or a transcoder, and a host computer. The host computer may be used for any of a number of purposes, such as video and audio editing. When using DV, DV commonly is transferred between the host computer and a peripheral device using an IEEE-1394 compliant bus. A standard is defined for transporting DV streams over an IEEE-1394 compliant bus, called IEC-61883. In an editing system that edits stored DV and outputs DV over an IEEE-1394 bus using IEC-61883, the host first decompresses the DV, performs editing operations on decompressed data, and then compresses the results back into DV before transferring DV over the IEEE-1394 bus. This process requires resources of the host computer to be used for compression, thus limiting the real-time processing capabilities of the host computer. Thus, a DV version of the entire program often is created and stored in a data file before it is transferred from the computer to another device. Further, to produce video in any output format other than DV, such as an analog video signal, the DV version also must be decompressed first. Decompression of DV and encoding of the results in an analog video signal generally is performed by a transcoder.

[0005] An IEEE-1394 compliant device transfers data based on a local time base using what is called a bus timer clock. As a result, each device attached to an IEEE-1394 bus has its own time base, which generally is faster or slower than the time base on other devices on the same bus. Although the IEC-61883 standard provides a technique for synchronizing these clocks, different time bases generally exist on different devices on the bus due to jitter and phase differences between the clocks. A consequence of the different time bases is that a host either transmits more data packets than a transcoder can consume or transmits too few data packets to the transcoder, causing a frame to be repeated or dropped. To address this problem, a host is required to periodically send empty data packets and to use a presentation time stamp (the SYT field) in a packet of each frame. The presentation time stamp designates to the transcoder a point in time when the frame should be displayed. However, the time stamp is generated by a sender using the sender's bus timer clock, but is used by the recipient using the recipient's bus timer clock. Thus, even if the number and frequency of the transmission of the empty data packets is calculated carefully in advance of transmission, on average, the empty data packets merely extend the period of time before a frame is dropped or repeated. The use of empty data packets does not solve the problem created by the different time bases on sending and receiving devices. Another problem with time stamps is that not all commercially available devices support use of time stamps. Some devices do not send timestamps, and some devices do not attempt to process received timestamps. Because frames may be dropped or repeated using these techniques, such transcoders cannot be used to reliably generate a synchronous video signal for playback on a video monitor or for recording to tape, or to synchronize playback of audio and video material with an external synchronized video device.

[0006] Another method is known from US-A-6418150 which calculates the frequency offset between a receiving and a source device and adjust the frame packets rate accordingly.

[0007] Current transcoder designs also do not enable a host to connect to another device through the transcoder in a manner that allows the host to communicate directly to the other device, yet allow the transcoder to modify data in packets sent to the other device.

[0008] A communication protocol is provided to allow a device that receives a periodic reference signal to control the flow of temporal data between devices over a bus to maintain synchronization of the temporal data with the periodic reference signal. Flow control messages are sent between devices, allowing a device that receives the periodic reference signal to control how and when the data is sent over the bus. For example, a peripheral device that receives a periodic

LINKPLAY EXHIBIT 1004
Page 1855 of 4822

reference signal may send flow control messages to a host computer to control the flow of data from the host computer to the peripheral device. Different types of temporal data, such as audio and video data, may be sent as separate data streams. Using the IEEE-1394 protocol, the flow control messages are sent using an asynchronous protocol, whereas temporal data are sent using an isochronous protocol.

[0009] Uncompressed video and audio data are transferred from the host computer over a serial bus to the peripheral device, where the peripheral device performs operations on the data, such as compressing the video data or generating an analog video signal. Video and audio data may be transferred as separate streams. By transferring uncompressed video data, the host computer does not need to consume resources to compress the video data, thus allowing those resources to be used for more creative operations on the video and audio data. Further, by transferring uncompressed video data, the peripheral device can generate video data in many output formats without first decompressing the video data. The peripheral device also may output video and audio data using a standard protocol over a standard bus. For example, the peripheral device may generate DV from the uncompressed audio and video data, and output the DV using IEC-61883 over an IEEE-1394 compliant bus.

[0010] A peripheral device that synchronizes transfer of data from the host computer with a periodic reference signal also can be used to synchronize data transfer to another device, such as a transcoder or camcorder or deck, that is connected to the peripheral device. For example, the peripheral device may encode the uncompressed audio and video data received from the host computer as a video signal. The peripheral device then outputs the video signal to the other device. The other device may provide the periodic reference signal. For example, if the other device is a camera, a composite video output signal from the camera can be used as the periodic reference signal. As a result, transfer of data from the host to the peripheral device and then to the camera is synchronized to the camera.

[0011] Such a peripheral device also allows a real time digital cut to tape to be performed. In particular, by transferring uncompressed data from the host computer to the peripheral device during playback in synchronization with a periodic reference signal, the host computer can process effects in real time. The peripheral device may generate a DV output from the uncompressed data. This DV output can be provided over an IEEE-1394 compliant bus to a recording device such as a DV deck or camcorder. If this DV deck or camcorder has a composite video output signal, then this output signal from the deck or camcorder may be used as the periodic reference signal. The peripheral device also may act as a bridge between the host computer and the other device. In particular, the peripheral device may implement a protocol over the serial bus between the host computer and the peripheral device and a protocol over another serial bus between the peripheral device and the other device. After receiving a packet at one port, the peripheral device determines if the packet is directed to a device connected on the other port or if the packet is directed to the peripheral device or if the packet is to be processed by the peripheral device to create one or more new packets to be output to the device connected on the other port. For example, the peripheral device may decompress compressed video data received from the host and output data packets that include uncompressed video data to the other device. From the perspective of the host, the peripheral device behaves as if it were the other device. From the perspective of other device, the data packets appear to be from a host device.

Fig. 1 is a block diagram of a system in which a host computer is connected to a peripheral device.
Fig. 2 is a block diagram of a system in which a host computer is connected to a peripheral device that is synchronized with another video device.
Fig. 3 is a more detailed block diagram of a mode of operation of the peripheral device as in Fig. 1.
Fig. 4 is a more detailed block diagram of another mode of operation of the peripheral device as in Figs. 1 and 2.
Fig. 5 is a more detailed block diagram of another mode of operation of the peripheral device as in Figs. 1 and 2.
Fig. 6 is a block diagram of a mode of operation using two peripheral devices that both receive data from a host computer.
Fig. 7 is a block diagram of a peripheral device for the mode of operation shown in Fig. 3.
Fig. 8 is a block diagram of a peripheral device for the mode of operation shown in Fig. 4.
Fig. 9 is a block diagram of a peripheral device for the mode of operation shown in Fig. 5.
Figs. 10A-10B are a more detailed block diagram of a peripheral device incorporating the modes of operation illustrated in Figs. 7-9.
Fig. 11 illustrates an example synchronization packet format.
Fig. 12 illustrates an example preamble packet format.
Fig. 13 illustrates an example stream descriptor format.
Fig. 14 is a table of stream identifiers.
Fig. 15 illustrates an example data packet format.
Fig. 16 illustrates an example null packet format.

[0012] Fig. 1 illustrates a system in which a host computer 100 is connected to a peripheral device 102 by a bus 104, such as an IEEE-1394 compliant bus. The host computer typically includes storage for storing video data and audio

LINKPLAY EXHIBIT 1004
Page 1856 of 4822

data in data files using a file system of the operating system of the computer. An editing system is provided on the host computer to allow editing of a program using the video data and audio data with effects to be performed on the video data. Such an editing system typically represents a program as a sequence of clips, wherein a clip is a reference to a data file containing temporal media, and a range within the data file. A playback system also is provided for playing back the edited program in full motion at full resolution. The playback system allows an editor to view the program that has been edited. If the video data is compressed, the playback system decompresses the video data, and performs any effects on the decompressed video data.

[0013] The bus may be any type of bus over which communication is performed using packets and which can transmit data for temporal data, such as video data and audio data, at a rate faster than the sample rate of the temporal data, such as the frame rate of the video data. The host computer includes a bus interface for sending packets to and receiving packets from the device over the bus. A buffer in memory commonly is connected to transfer data to the bus interface for transmission as a plurality of packets. This buffer will receive, for example, data from the playback application for transfer to the device. A driver, a form of software that acts as a controller, for the bus interface also is on the host computer and allows the host computer to implement communication protocols, such as the protocol described herein, over the bus.

[0014] The host computer may receive data over the bus from the peripheral device. The host computer may send data over the bus to the peripheral device. The host computer and peripheral device may exchange several kinds of data including video data, audio data and other temporal data and other content. Video data may be either compressed or uncompressed, and may be combined with or separate from audio data and other related time-based media. Audio data typically is uncompressed, but may be compressed. The invention is not limited to any particular format of video data or audio data or other temporal data.

[0015] Data is transferred over the bus 104 in the form of packets. A packet is a unit of transmission of data that includes both data and routing information, such as an address of a node on the bus. A packet also may indicate a type of the packet if the communication protocol allows different types of packets to be transmitted. A packet also may include error correction codes and other data. The form and content of each type of packet typically is specified as part of a communication protocol that is described in more detail below.

[0016] The peripheral device may have several inputs 106 to receive data in any of a number of formats from a source, such as video and audio data from a tape deck. The peripheral device also may have several outputs 108 to send data in any of a number of formats to a receiver, such as video and audio data to a monitor and speakers or to a tape deck. For example, the peripheral device may send DV to a DV device, such as a DV camcorder or DV tape deck, or may receive DV from a DV device. Examples of other video and audio input and output formats include, but are not limited to, analog formats such as composite video, component video (such as YCrCb and YUV video) and S-Video, and digital formats, both compressed and uncompressed, both standard and proprietary, such as MPEG-2, SMPTE-125M, SMPTE-260M, SMPTE-264M, SMPTE-265M, SMPTE-267M, SMPTE-274M, and SMPTE-279M, among others. The peripheral device also may internally generate or receive from an external source a periodic reference signal 110 for use in controlling data transfer from the host. This periodic reference signal has a rate that corresponds to the desired rate of the temporal data, such as a field or frame rate of video, that is output by the peripheral device. The periodic reference signal may be generated by an accurate crystal clock generation circuit. Such a circuit may be internal to the peripheral device or external to the peripheral device. The periodic reference signal also may be generated by using a video decoder that decodes a received composite video signal to generate a vertical synchronization signal (VSYNC), a horizontal synchronization signal (HSYNC), a pixel clock (such as 27 MHz for NTSC/PAL standard definition video) and a burst phase signal. Such a composite video signal used for this purpose is commonly called a genlock, RS170, black burst or house sync signal, and may or may not include active video. The signals output from the decoder may be provided as inputs to a video encoder to generate output video that is in synchronization with the received composite video signal.

[0017] Each field boundary or frame boundary, or other designated boundary in time that can be derived from the periodic reference signal, may be used to define a data interval. A data interval is a period of time within the temporal data that corresponds to the periodic reference signal, such as a frame or field of video data, or audio or metadata that corresponds to the frame or field of video data. The communication protocol described below utilizes a data interval as a basis for transfer of data from the host to the device to synchronize the data with the periodic reference signal.

[0018] The peripheral device may be used to capture or playback multiple streams of audio and video data for the host computer. For capture, the peripheral device receives data from one or more of its inputs and transfers the received data to the host computer over the bus. For example, the peripheral device may receive a DV stream, which it transfers directly to the host computer. The peripheral device may receive an analog video and audio stream, which it then converts to a DV stream that is transferred to the host. In the playback mode, the peripheral device receives a stream of video and audio data from the host computer over the bus. This stream may be converted into any of a number of formats for output to another device. For example, the peripheral device may convert the received stream into an analog video and audio signal or into a DV stream for transmission to a recording device or to a monitor and speakers.

[0019] Referring to Fig. 2, the peripheral device 202 can synchronize data transfer over the bus 204, from the host

LINKPLAY EXHIBIT 1004
Page 1857 of 4822

computer 200 to another device 212 that is connected (via inputs 206 or outputs 208) to the peripheral device 202, to a periodic reference signal 210 that is received by both the peripheral device 202 and the other device 212. The periodic reference signal may be generated by the peripheral device 202, the other device 212, or may come from another source.

[0020] Examples of other devices 212 that may be connected to the peripheral device 202 include but are not limited to a transcoder, video display, tape deck, camera or video processing equipment. For example, the peripheral device may receive uncompressed video and audio data from the host computer. The peripheral device may generate from this data an analog video and audio signal that is synchronized with the periodic reference signal and that is output to the other device.

[0021] The various functions of the peripheral device also may be integrated within the other device to produce a single device that operates to provide the combined functions. Similarly, the various functions of the other device may be integrated within the peripheral device to produce a single device that operates to provide the combined functions. Thus, a transcoder, video display, tape deck, camera or other video processing equipment could incorporate the functions of the peripheral device as described herein. Three example operational modes for such a peripheral device include uncompressed playback mode, digital cut mode, and capture mode, as will be described in connection with Figs. 3-5.

[0022] In uncompressed playback mode, shown in Fig. 3, the host system 300 passes a stream of uncompressed video and audio and/or device control commands to the peripheral device 302 over the bus 304. For playback during editing, the uncompressed stream may be fractional frame (such as ¼-frame) or full-frame, depending on a user-defined setting. For digital cut mode discussed below, full-frame mode is the default setting. The peripheral device may include a resizer to scale fractional-frame video to full-frame video. The peripheral device may generate an analog video and audio output 306 using the uncompressed stream. A video window on the desktop of the host computer may play in synchronization with the outputs of the peripheral device. The peripheral device may create a DV or other compressed output 308 by encoding the uncompressed stream using an encoder on the peripheral device. By providing sufficient buffering, the compressed output may be kept in synchronization in spite of latencies incurred by encoding the uncompressed stream on the peripheral device.

[0023] In this mode, editing, playback and creation of output video and audio with real-time effects can be provided. In particular, because the host does not compress the video and audio stream after application of a real-time effect, the uncompressed output is transferred in real time during playback to the peripheral device, which in turn can encode the uncompressed stream for output. Such operation also allows a real time digital cut to tape to be performed. In particular, by transferring uncompressed data from the host computer to the peripheral device during playback in synchronization with a periodic reference signal, the host computer can process effects in real time. The peripheral device may generate a DV output from the uncompressed data. This DV output can be provided over an IEEE-1394 compliant bus to a recording device such as a DV deck or camcorder. If this DV deck or camcorder has a composite video output signal, then this output signal from the deck or camcorder may be used as the periodic reference signal.

[0024] Playback of video may occur in one or more of a desktop window on the host computer, composite or S-Video outputs 306 on the peripheral device, and a DV output 308 via a IEEE-1394 compliant bus on the peripheral device, or on other outputs of the peripheral device. Playback of audio may occur through one or more of the analog outputs, DV outputs or digital outputs (not shown).

[0025] In another mode shown in Figure 4, a digital cut of a compressed stream rendered by the host computer 400 and/or device control commands is provided to the peripheral device 402 over the bus 404. In this mode, any effects are rendered and the rendered images are compressed at the host computer and stored in the data files before the transfer. The compressed stream is read by the host computer 400 from its data files and transmitted to peripheral device 402. The compressed stream may be transmitted to the peripheral device using the synchronized protocol described below. The synchronized protocol is used to transfer the DV stream for playback on a synchronized device such as a television monitor.

[0026] Playback of video may occur in one or more of a desktop window on the host computer, composite or S-Video outputs 406 on the peripheral device, and a DV output 408 via the IEEE-1394 compliant bus on the peripheral device, or on other outputs of the peripheral device. Playback of audio may occur through one or more of the analog outputs, DV outputs or digital outputs (not shown). The analog outputs are created by having the peripheral device decode the incoming compressed stream using a decoder on the peripheral device, then encoding the decoded data using a video encoder. To perform a digital cut to a DV tape, the peripheral device performs device control operations on the DV device (not shown) attached to the DV output 408, and outputs the DV data directly as received from the host.

[0027] In this mode the compressed output from the peripheral device plays in synchronization with the video in the desktop window on the host computer. By providing sufficient buffering in the host, the latencies arising from decoding the DV stream and encoding the analog outputs on the peripheral device may be overcome to synchronize the analog outputs with the video in the desktop window.

[0028] A third mode of operation is capture, shown in Figure 5. Capture employs one of several different input sources (506, 508, 510) of the peripheral device 502, such as composite video, S-Video video, and DV video and audio inputs. DV device control by the peripheral device 502 is available if a DV input device is used.

LINKPLAY EXHIBIT 1004
Page 1858 of 4822

[0029] If the input device is DV, then the DV stream is passed through the peripheral device directly to the host 500 over the bus 504. The DV stream may be transferred to the host system using the synchronized protocol described below. During transfer of DV data to the host computer, the peripheral device may decode the DV stream and encode the decoded data to generate an analog video and audio outputs (not shown).

[0030] If the input signal is composite or S-Video, or some other format, then the input signal is converted into a digital format, such as DV, before being transmitted to the host over the IEEE-1394 compliant bus. In this mode the analog video and audio signals also may be output through the corresponding analog video and audio outputs (not shown). Output of DV format video over the other IEEE-1394 compliant bus output 508 to an external device also may be active or inactive.

[0031] In all of these modes, the peripheral device may act as a bridge between the host computer and the other device. In other words, the peripheral device implements a protocol over the first serial bus for communication between the host computer and the peripheral device. The peripheral device implements the same protocol over the second serial bus for communication between the peripheral device and the other device. The peripheral device communicates information from the other device to the host computer to allow the host computer to communicate packets directed to the other device using the protocol over the first serial bus. The peripheral device also communicates information from the host computer to the other device to allow the other device to communicate packets directed to the host computer using the protocol over the second serial bus.

[0032] In general, the peripheral device receives messages from a host at one port of the peripheral device, translates the message, then send a message to the other device from another port of the peripheral device. The translation involves at least translating any node address in the message so that it is properly directed to the other device through the other port. Translation may also involve other processing to be performed on the data in the packet such as conversion from one data format to another.

[0033] Conversely, the peripheral device receives messages from the other device at one port of the peripheral device, translates the message, then sends a message to the host from another port of the peripheral device. Again, the translation involves at least translating any node address in the message so that it is properly directed to the host through the other port, and may involve converting data formats or other processing of the data in the message.

[0034] To implement such functionality, a separate process for managing communication on each port is executed by a microprocessor on the peripheral device. When a message is received on one port, the process for that port invokes a translator process which translate the message and sends the translated message to the other process for the other port.

[0035] Whether packets flow from the host to the other device or from the other device to the host, packets received by the host, peripheral device or other device are locally acknowledged using a low level protocol, such as the link layer protocol. However, application level acknowledgements are passed from the host to the other device, and from the other device to the host, through the peripheral device using the node translation described above. Thus, the peripheral device appears to be a virtual host from the perspective of the other device, and appears to be a virtual device from the perspective of the host.

[0036] This bridge functionality will now be described in more detail for four types of communication: from host to peripheral device, from peripheral device to other device, from other device to peripheral device and from peripheral device to host.

[0037] If a packet is received by the peripheral device from the host computer on one port, that port locally acknowledges receipt of the packet using, for example, a link layer or other low level protocol. The packet is processed to determine if the packet is directed to the peripheral device or the other device. If the packet is directed to the peripheral device, the packet is processed at the peripheral device. If the packet is directed to the other device, the packet is first processed by the peripheral device to create one or more new packets that are in turn sent to the other device.

[0038] If a packet is sent to the other device by the peripheral device by one port, the other device locally acknowledges receipt of the packet by using a link layer or other low level protocol message.

[0039] If a packet is received by the peripheral device from the other device by one port, that port locally acknowledges receipt of the packet, for example using a link layer or other low level protocol. The packet is processed to determine if the packet is directed to the host computer or to the peripheral device. If the packet is directed to the peripheral device, the packet is processed at the peripheral device. If the packet is directed to the host computer, the packet is processed by the peripheral device to create one or more new packets that are in turn transferred to the host computer.

[0040] If a packet is sent to the host by the peripheral device by one port, the host locally acknowledges receipt of the packet by using a link layer or other low level protocol message.

[0041] The local acknowledgement of packets using the low level protocol is different from acknowledgement of application level messages by the other device or by the host. If the other device is to respond to a message received from the host through the peripheral device, this response is made using an application level protocol, such as AV/C, and is handled as a message from the other device that is directed to the host through the peripheral device. Similarly, if the host is to respond to a message received from the other device through the peripheral device, this response is made using an application level protocol, such as AV/C, and is handled as a message from the host that is directed to

LINKPLAY EXHIBIT 1004
Page 1859 of 4822

the other device through the peripheral device.

[0042] Two peripheral devices also may be connected to receive data from the same host, as illustrated in Fig. 6. In particular, a first peripheral device 600 and a second peripheral device 610 both receive a periodic reference signal 620. The first peripheral device 600 communication with a host 630 over a bus 640. The second peripheral device 610 is another node on the same bus 640. Using the command protocol described below the host computer selects one of the peripheral devices and directs it not to send flow control messages. The host computer then replies only to flow control messages from the other peripheral device. In this manner, both of the peripheral devices can provide synchronized outputs from the same data stream sent by the host computer.

[0043] Example hardware that supports each of the modes of operation described above will now be described in connection with Figs. 7-10A-B.

[0044] In uncompressed playback mode, shown in Figure 7, the peripheral device operates as a frame buffer and DV compression device. Uncompressed audio and video data is sent by the bus interface, shown as a IEEE-1394 transceiver 706, on the host 700 over the IEEE-1394 compliant bus 704 to the peripheral device 702. The bus interface, shown as a IEEE-1394 transceiver 708, on the peripheral device 702 receives the data, and the data is stored in the media hub 710, which includes a frame buffer. The video data is then reformatted as an ITU/CCIR 656-video stream by a CCIR 656 formatter 712, which provides the reformatted data to both the DV encoder 714 as well as a video encoder 716. An I2S formatter 718 reformats the audio data into an I2S audio stream for transport to both an audio encoder 720 and the DV encoder 714. The DV stream output by DV encoder 714 may be provided to an IEEE-1394 formatter 722 to produce IEEE-1394 packets that may be output over another IEEE-1394 compliant bus 726 by another IEEE-1394 transceiver 724.

[0045] To prevent dropping or adding video frames in the video provided to the CCIR 656 formatter 712 during playback of video over the IEEE-1394 compliant bus 704, the peripheral device controls the flow of data from the host computer 700 to the peripheral device 702. This flow control also allows the peripheral device to generate and output a synchronous video signal using the analog encoder 716 that is synchronized with an internal or external periodic reference signal. A communication protocol to implement this flow control is described in more detail below.

[0046] In digital capture mode, shown in Fig. 8, the peripheral device 802 operates as both a buffer and a decoder. Digital video and audio data, such as DV, is received from a device (not shown) and is passed to the host computer 800 for processing and storage on a storage device accessed by the host computer, such as a disk system. The received data, if compressed, also may be decompressed to produce output audio and video for monitoring. More particularly, in Fig. 8, data, such as DV, is received through an IEEE-1394 interface 806, is buffered and sent by another IEEE-1394 interface 808 over the bus 804 to the IEEE-1394 interface 810 at the host 800.

[0047] The received data also may be deformatted by IEEE-1394 deformatter 812 and input to a decoder 814, such as a DV decoder, to be decompressed. The decompressed data then can be encoded for output to other devices. For example, the decompressed data may be input to an analog video encoder 816 and the audio data may be input to an audio encoder 818. The outputs of the encoders can be applied to a monitor and speakers for monitoring of the capture process. Because the timing on the analog video encoder 816 is derived from a local time base and is asynchronous to the incoming DV data, the analog encoder 816 may either repeat a frame of video or drop a frame of video depending on whether the incoming DV data steam is running slower or faster than the video encoder.

[0048] In analog capture mode, shown in Fig. 9, the peripheral device 902 converts input analog video and associated audio into digital format, and transfers the digital data to the host 900 over the bus 904. For example, an analog video decoder 906 converts the input analog video into a digital ITU 656-video data stream. The digital video data is sent to a encoder 908, such as a DV encoder, for compression. Received audio is connected to an audio encoder/decoder 910 for conversion to an I2S digital audio format. The audio data also may be provided to encoder 908 for compression. The audio encoder/decoder 910 also may provide audio outputs for monitoring the audio during capture. A video encoder 912 also may provide a video output for monitoring the video during capture. Encoded video data from encoder 908 is the formatted by an IEEE-1394 formatter 914, which prepares it for transmission by the bus interface, shown as IEEE-1394 transceiver 916. The host receives the data over the bus 904 through the bus interface, shown as IEEE-1394 transceiver 918.

[0049] A block diagram of a peripheral device that has these modes of operation is shown in Figs. 10A-10B. In Figs. 10A-10B, the peripheral device 1000 operates in two basic modes: capture and playback. In capture mode, digital video and audio data, such as DV, may be received by an IEEE-1394 compliant interface 1002 and sent to the host over an IEEE-1394 interface 1004 to a corresponding interface 1051 on the host. Analog audio and video also may be received and converted to a digital format and then sent to the host over the interface 1004. In playback mode video and audio data is received from the host through interface 1004 and is output as analog video and audio and/or is output as digital video and audio data through the IEEE-1394 compliant interface 1002. The IEEE-1394 compliant interface 1002 provides an interface through port 1045 to external devices, such as cameras and decks, whereas the IEEE-1394 compliant interface 1004 provides an interface to the host computer. The interfaces 1002 and 1004 can be implemented, for example, using a physical layer device and a link layer device. A Texas Instruments TSB12LV32 Link-Layer Controller and a TSB41LV01 Cable Transceiver/Arbiter can be used to achieve a 400 megabit per second nominal transmission

LINKPLAY EXHIBIT 1004
Page 1860 of 4822

rate over the IEEE-1394 bus.

[0050] Left and right audio inputs and outputs 1040, 1041, 1042, 1043 for audio processing provide inputs to an audio encoder/decoder 1006, which may be implemented, for example, using a Philips UDA1345TS, which supports 32, 44.1 and 48kHz rates. Other chips supporting other rates can be used. To keep the audio locked to the video during capture, the audio clock is derived from the input video decoder 1008. For playback or output the audio clock is derived from the analog video encoder 1010.

[0051] A headphone audio output 1044 may be provided through an encoder 1012, which may be implemented, for example, using a Philips UDA1320ATS audio digital to audio converter (DAC), which supports 32, 44.1 and 48kHz rates. Other chips supporting other rates can be used. A separate DAC may be used for the headphone to allow adjustment to the headphone volume without affecting the main audio outputs.

[0052] The peripheral device 1000 may receive analog video through inputs 1046 and 1047, such as composite and S-video inputs. These inputs are received by a decoder 1008 which converts the input signal into digital video data. The decoder 1008 may be implemented, for example, by a Philips SAA7114H video decoder, which supports NTSC, NTSC-j and PAL analog video formats. Other decoders supporting the same or other formats may be used. If connected to either a composite or S-Video source, this decoder converts the incoming analog signal into a standard ITU 656 YUV 4:2:2 digital video data stream. This decoder also provides support for luminance offset control, luminance contrast control, chrominance saturation control and chrominance hue control.

[0053] The peripheral device 1000 also may output analog video through outputs 1048 and 1049, such as composite and S-video outputs provided by an encoder 1010. The encoder 1010 may be implemented, for example, by an Analog Devices ADV7176 encoder. The encoder 1010 accepts a YUV 4:2:2 digital video stream and real time control stream information from the genlock circuit 1018, buffered through a media hub 1028, to generate synchronized analog composite and S-Video outputs. The ADV7176 device encodes analog video and provides several functions for processing the output video, such as luminance filters (Low-Pass/Notch/Extended) control, subcarrier frequency and phase control and luminance delay control.

[0054] The peripheral device 1000 also may include a compression/decompression device (codec) 1014, such as a DV codec. This codec may be implemented, for example, using a Divio NW701 DV25 codec for both decompressing and compressing a DV data stream. In compression, the codec 1014 receives an ITU 656 YUV 4:2:2 digital video stream and an I2S audio stream and compresses it to produce a DV compressed video stream. In decompression, the codec 1014 receives a DV compressed video stream and produces both an ITU 656 YUV 4:2:2 digital video stream and an I2S audio stream. The NW701 codec also provides NTSC (4:1:1), PAL (4:2:0) and PAL (4:1:1) processing, audio support for 48, 44.1 and 32 kHz (12 and 16 bit), and optional low data-rate modes, including 3.0, 2.4, 1.8, 1.5 and 1.0 Mbytes/sec. The codec 1014 uses a memory 1016 for storing data temporarily during processing. Such a memory 1016 may be implemented, for example, using a 256Kx32 DRAM.

[0055] The peripheral device 1000 also may receive a periodic reference signal (genlock) through an input 1050. This signal is received by a video genlock device 1018, which may be implemented, for example, using a Philips SAA7113 decoder. This decoder supports NTSC, NTSC-j and PAL analog video formats. If connected to a composite genlock source it extracts the vertical synchronization, horizontal synchronization and color burst phase information and provides this information to the media hub 1028, which can be used to lock the output timing of the ADV7176 video encoder 1010 to the genlock source.

[0056] Controls may be provided on the peripheral device to allow an individual to adjust the various controls of the input decoders and output encoders, particularly to adjust levels of the analog video signals, for audio muting and for adjusting gain levels of the audio signals.

[0057] A frame buffer controller, resizer and data router, herein called a "media hub" 1028, and associated buffer 1030 provide four basic functions on the peripheral device 1000. First, they buffer both uncompressed and compressed data streams between the inputs and the outputs. Second, they provide a switching or multiplexing function that routes data among the inputs, outputs, encoders and decoders. Third, they de-format incoming IEEE-1394 compliant data streams and ITU-656 compliant video streams. Fourth, they format out going IEEE-1394 compliant data streams and ITU-656 compliant video streams. The media hub 1028 may be implemented, for example, using a field programmable gate array (FPGA) with a local memory, such as an SDRAM as its associated buffer 1030.

[0058] The functions of the media hub 1028 are performed in connection with a microcontroller subsystem 1020. The primary function of the microcontroller subsystem is to provide an abstraction layer for software commands received over the host IEEE-1394 compliant interface 1004 and for controlling the IEEE-1394 compliant interface 1002. The microcontroller subsystem also provides support for setting up and controlling the video and audio encoders and decoders 1006, 1008, 1010, 1012 and 1018. The microcontroller subsystem 1020 includes a microprocessor 1022, which may be implemented, for example, using a Motorola MPC5307 microprocessor. The microprocessor 1022 operates in accordance with instructions stored in the flash memory 1024 and static random access memory 1026. In particular, an actual operating system with a real-time kernel can be provided for the microcontroller, such as the NUCLEUS operating system from Accelerated Technology, Inc., the embedded systems division of Mentor Graphics and located in Mobile,

LINKPLAY EXHIBIT 1004
Page 1861 of 4822

Alabama.

[0059] Operation of the microcontroller will now be described in more detail. The microcontroller receives commands from the host through a command protocol described in more detail below. These commands allow the host to set the mode of operation of the peripheral device. These modes include settings for the expected audio and video format for input and output. Example format mode settings include PAL and NTSC for video and 48KHz, 44.1KHz or other rate for audio. For these modes, the host may designate a format to the microcontroller. The microcontroller in turn sets various registers of the input and output devices to handle the desired format.

[0060] Other modes indicate whether the host is receiving data from or providing data to the peripheral device. Some example modes include, but are not limited to:

- DV input to peripheral device and DV output to host
- DV input to peripheral device and uncompressed output to host
- Composite input to peripheral device and DV output to host
- Composite input to peripheral device and uncompressed output to host
- Uncompressed input from host and composite output
- Uncompressed input from host and DV output
- DV input from host and composite output
- DV input from host and DV output.

[0061] In a mode in which DV is input to the peripheral device and DV is output to the host computer, packets are received through interface 1002 and stored in the memory 1030 by media hub 1028. When a packet becomes available in the memory 1030, the media hub 1028 notifies the interface 1004. Interface 1004 then transfers the packet to the host. In this mode the device acts as a bridge.

[0062] In a mode in which DV is input to the peripheral device and uncompressed data is output to the host computer, packets are received through interface 1002 and stored in the memory 1030 by media hub 1028. Packets are allowed to accumulate in the memory 1030 until a frame of image data is received. The media hub 1028 then directs the DV codec 1014 to read the frame of data from memory 1030 into its memory 1016. The DV codec then generates an uncompressed frame. The uncompressed frame is placed in another portion of the memory 1030. When an uncompressed frame is available in the memory 1030, the media hub 1028 notifies the interface 1004. The interface 1004 then transfers the image as multiple packets to the host. In this mode the device acts as a bridge.

[0063] In a mode in which a composite input is received by the peripheral device and DV is output to the host computer, the decoder 1008 produces digital video information which is stored in the memory 1030 by the media hub 1028. When a frame of image data has been received, the media hub 1028 directs the DV encoder 1014 to read the frame of data from memory 1030 into its memory 1016. The DV encoder then generates DV and places the data in another portion of the memory 1030. The media hub 1028 notifies the interface 1004 that data is available. The interface 1004 then transfers the DV image as multiple packets to the host.

[0064] In a mode in which a composite input is received by the peripheral device and uncompressed data is output to the host computer, the decoder 1008 produces digital video information which is stored in the memory 1030 by the media hub 1028. As image data is received, the media hub 1028 directs the interface 1004 that data is available for transfer. The interface 1004 then transfers packets of uncompressed data to the host.

[0065] In a mode in which the peripheral device receives uncompressed data from the host and outputs a composite analog video signal, the media hub 1028 directs packets received through interface 1004 to be stored in the buffer 1030. As data becomes available in memory 1030, the media hub 1028 directs data to the encoder 1010 for output.

[0066] In a mode in which the peripheral device receives uncompressed data from the host and outputs DV to another device, the media hub 1028 directs packets received through interface 1004 to be stored in the buffer 1030. As a frame of uncompressed data becomes available in the memory 1030, the media hub 1028 notifies the DV codec 1014 which reads the data into its buffer 1016. The DV codec then generates DV which is stored into the memory 1030. The media hub 1028 then notifies the interface 1002 that DV data is available. The interface 1002 reads the data and transfers it as packets through the output 1045. In this mode the device acts as a bridge.

[0067] In a mode in which the peripheral device receives DV input from the host and provides a composite analog video output signal, the interface 1004 receives packets of DV data that are stored in the memory 1030. When a frame of data is available, the media hub 1028 notifies the DV codec 1014, which then reads the data into its memory 1016. The DV codec then produces uncompressed video data which is stored in memory 1030. The media hub 1028 the provides the uncompressed video data to the analog encoder 1010 for output.

[0068] In a mode in which the peripheral device receives DV input from the host and outputs DV to another device, the interface 1004 receives packets of DV data that are stored in memory 1030 by media hub 1028. The media hub 1028 notifies the interface 1002 when packets are available for output. The interface 1002 then reads the packets from the memory 1030. In this mode the device acts as a bridge.

LINKPLAY EXHIBIT 1004
Page 1862 of 4822

[0069] During each of the modes described above, corresponding audio input and output may be provided. Also, a composite or S-video output may be provided by the peripheral device to allow monitoring by an external monitor. Each device 1002, 1014 and 1004 may have a direct memory access (DMA) engine to access the memory 1030.

[0070] In some modes, the uncompressed video data may be received or output in 1/4 frame size. In these modes, the media hub 1028 also performs a resize function.

[0071] In some modes, multiple outputs may be provided by the peripheral device. For example, the peripheral device may output DV, composite and S-video using an uncompressed input from the host. Instead of DV, the peripheral device may input or output other digital video formats, whether compressed or uncompressed. Instead of composite, the peripheral device may input or output other analog video formats. Multiple outputs also may be provided by the peripheral device, in which operations from the above described modes would be performed in parallel.

[0072] The peripheral device also may include a bus connector connected to a data path to the media hub 1028 to allow other devices to connect to the media hub to transfer audio and video data through the peripheral device. This bus connector could be used to allow other processing devices both to provide data to and to receive data from the peripheral device. For example, such a device could receive a high definition video signal and convert it into a compressed data stream which is then transferred to the peripheral device for input to the host computer.

[0073] A communication protocol for controlling communication between the host computer and the peripheral device will now be described. This protocol is divided into two parts, a data transport protocol and a control protocol. The data transport protocol defines how temporal data, such as video, audio and associated metadata, are transferred from the host system to the peripheral device, or from the peripheral device to the host system. The data transport protocol includes both data format and synchronization rules. The control protocol defines how the host system queries and controls the states, modes, and operating parameters of the peripheral device. These protocols are described in more detail below.

[0074] The communication protocol is built on parts of the IEEE-1394 high-speed serial digital bus standard and conforms with the following standards documents, IEEE 1394-1995 Specification IEEE 1394 A Specification. The communication protocol is designed to work with standard IEEE-1394 compliant hardware bus controllers. However, for certain video data formats (e.g.: uncompressed, full-frame, 2-field, 4:2:2) the host and peripheral device controllers support a 400Mb data rate.

[0075] The protocol used to control external devices (such as DV camcorders and VTRs) is known as AV/C, or Audio/Video Control. The specification and command set are defined in the following documents: AV/C Digital Interface Command Set, General Specification, and Version 3.0AV/C Tape Recorder/Player Subunit Specification, Version 2.1. An example data transport protocol will now be described in more detail. In this example, the data-transport protocol is used to transmit streams of video and audio over an IEEE-1394 compliant bus between a host system and a peripheral device. This protocol is sufficiently flexible to allow for expansion for transmission of metadata as well.

[0076] In the following description of the protocol, the host computer is called the host and the peripheral device is called the device. Depending on the mode in use, the host can transmit (for digital cut mode) or receive (for capture mode). The transmitting device is called the talker and the receiving device is called the listener. The term host should be understood to indicate any apparatus that can send or receive data. The term device should be understood to indicate any device that can send or receive data that also processes a periodic reference signal.

[0077] Data transmission is performed in intervals, called data intervals, that correspond to a rate of the periodic reference signal to which temporal data is to be synchronized. For example, if the periodic reference signal corresponds to video, the data interval corresponds to an image, typically a frame, of video data. Example frame intervals include, but are not limited to, 29.97 FPS (for NTSC (525-line) systems) and 25 FPS (for PAL (625-line) systems).

[0078] Data is transmitted in the form of a series of concurrent streams, each identified by a unique stream identifier defined by the protocol. During transmission, streams can be started and stopped at any sample corresponding to a data interval, such as a frame boundary in the video data. The streams of data are broken down into segments based on the data interval in effect. Because of the nature of IEEE-1394 protocol, each data stream is further divided into a series of one or more packets for transmission during a given data interval.

[0079] For each data interval, the talker can choose to transmit any combination of streams that the listening device accommodates. For example, assuming that the listening device can accept two fields of video and two channels of audio, the talker could choose to just send audio or just video, or nothing at all. The listener takes a default action if a stream is missing. For example, if one or more audio streams are missing, which may occur if the audio channel is muted, then the listener fills silence on those channels. If the video data is omitted, which may occur in off speed playback, then the listener might repeat the last frame it received.

[0080] A data transmission includes several types of packets. A "synchronization packet" is sent from the device to the host indicating that a new data interval has begun. The talker then may send data. In one embodiment, a "preamble packet" is sent from the talker to indicate to the listener which streams' will be transmitted during the data interval, in what order they will be sent, and how many packets will be used to transport each stream. Zero or more "data packets" then are sent from the talker, based on the information given in the preamble. The talker then waits for the next data

LINKPLAY EXHIBIT 1004
Page 1863 of 4822

interval to send more data. For example, the talker may send zero or more "null packets" to fill bus cycles on an IEEE-1394 interface until the next data interval begins. This sequence of packets is repeated for each data interval.

[0081] Preamble, data, synchronization, and null packets may use the streaming packet modes of the IEEE-1394 protocol. The IEEE-1394 specification allows for two types of streaming packets: isochronous and asynchronous. For both types, the packet format is the same, utilizing the same transaction code (tcode = 0xA). Asynchronous stream transmissions are similar to isochronous transmissions, but there are the following differences. Asynchronous stream packets are transmitted during the asynchronous bus time, whereas isochronous stream packets are transmitted during the isochronous bus interval. Asynchronous stream packets are limited to a maximum payload of 2048 bytes, whereas isochronous stream packets may contain as many as 4096 bytes.

[0082] The synchronization packet is sent by the peripheral device to indicate that a data interval has begun. The peripheral device generates the synchronization packet in response to the periodic reference signal. More particularly, a synchronization packet is sent at the beginning of each data interval corresponding to the periodic reference signal. The beginning of a data interval may be detected, for example, by monitoring the periodic reference signal. An interrupt or other signal may be issued to the microcontroller of the device when then beginning of a data interval is detected.

[0083] An example format for a synchronization packet 1100, as an asynchronous packet, is shown in Figure 11, and has a format that corresponds to a Write Request packet. The shaded fields indicate those defined by the IEEE-1394 standard. The "destination_ID" field 1102 includes a 16-bit address of the node that is to receive the packet. This field includes a destination_bus_ID (upper 10 bits) and a destination_node_ID (lower 6 bits). To broadcast the synchronization packet to all nodes on the local bus, the destination_bus_ID is set to 3FFh and the destination_node_ID is set to 3Fh. The "tl" field 1104 is the transaction label which is a field that is set by the host. The "rt" field 1106 is the retry code, specifying whether the packet is a first attempt (00b) or a successive retry of the same command (01b through 11b). The "tcode" field 1108 specifies the 1394 transaction code for the packet and is set to "1h", which indicates that this packet is a Write Request packet. The "pri" field 1110 represents a priority code in the IEEE-1394 standard, but is not used. The "source_ID" field 1112 is the address of the node that is sending the packet. The "destination_offset" field 1114 specifies the 48-bit address location on the target that is being accessed. A specific register may be set aside in each possible destination for receiving synchronization packets. This field 1114 shall contain the address of that register. The "device_status" field 1116 is the data sent by the packet and includes information regarding the reception of the previous frame, and the general status of the device. A field that enables error detection and correction ("header_CRC" 1118) also may be included.

[0084] The preamble packet is transmitted by the talker at the beginning of each data interval as a single packet. It may be transmitted as an isochronous packet. This packet specifies which streams will be transmitted during the data interval, and how many packets to expect for each stream. The talker transmits a preamble packet for a data interval even if no stream is to be sent. An example format for a preamble packet 1200 is shown in Figure 12. The shaded fields indicate those defined by the IEEE-1394 standard. The "data_length" field 1202 includes the length in bytes of the payload section of the isochronous packet. The "tag" field 1204 includes the value 00b which indicates that the data is unformatted. The "channel" field 1206 includes a value that indicates the channel assigned to the talker and listener for transmission of this packet. The "tcode" field 1208 includes the value 1010b which indicates that this packet is a streaming packet. The "sy" field 1210 includes the value 0001b to indicate that this packet is a preamble packet. The "transfer rate" field 1212 indicates the rate at which the stream data is arriving in comparison to its real time rate, and is represented by one byte. In other words, it specifies how many frames will be transmitted during this frame time. Example transfer rates are 1x, 2x and 4x real time. The "stream count" field 1214 is an unsigned integer that indicates how many streams will be transmitted per frame during the current data interval. This value is represented by one byte. A number of stream descriptors 1216 follows this quadlet equal to the count specified by the product of the transfer rate field and the stream count field. The list of stream descriptors specifies which streams will be transmitted during this data interval, in what order they will be transmitted, and how many packets are used for each stream. The preamble packet also may include fields that enable error detection and correction ("header_CRC" 1218 and "data_CRC" 1220).

[0085] An example format for stream descriptor 1300 is shown in Figure 13. The "stream ID" field 1302 identifies the type of stream, for example using an 8-bit unsigned integer. The "packet count" field 1304 specifies how many packets will be used to transmit the stream, for example using a 12-bit unsigned integer. Bits in fields marked as "stream-specific-data" in the preamble and stream descriptors are set to zero for the video fields, video frames, system and auxiliary streams. For each audio stream, this value is an unsigned 12-bit integer representing the number of audio samples in the data interval. The total number of data packets which will follow the preamble is the sum of all of the "packet count" fields in the stream descriptor list.

[0086] Example values for stream identifiers for the "stream ID" field are illustrated in the table 1400 of Fig. 14. The stream identifiers in this example do not indicate the format of the data itself (e.g.: uncompressed or DV or JFIF). Instead, the format of the data is established before transmission begins using the control protocol. In this table, each row represents a stream identifier. The "value" column 1402 represents the value that is used as a "stream ID" in a stream descriptor packet. The "name" 1404 and "description" 1404 columns serve to further describe the stream identifier and

LINKPLAY EXHIBIT 1004
Page 1864 of 4822

may be used to reference the stream identifier. The following table illustrates example definitions for preambles for different types of video data, specifically DV25, DV50, IMX and uncompressed data, including a list of descriptors for each type.

name	transfer-rate	stream-count	descriptors
DV25/DV50 @ 1X	1	1	VFRAME
DV25/DV50@ 2X	2	1	VFRAME VFRAME
DV25 @ 4X	4	1	VFRAME VFRAME VFRAME VFRAME
IMX @ 1X	1	10	SYSTEM VFRAME A1 A2 A3 A4 A5 A6 A7 A8
IMX @ 2X	2	10	SYSTEM VFRAME A1 A2 A3 A4 A5 A6 A7 A8 SYSTEM VFRAME A1 A2 A3 A4 A5 A6 A7 A8
uncompressed two	1	10	VFIELD1

LINKPLAY EXHIBIT 1004
Page 1865 of 4822

(continued)

name	transfer-rate	stream-count	descriptors
field			VFIELD2 A1 A2 A3 A4 A5 A6 A7 A8

[0087] Data packets convey actual audio, video and metadata. An example format for a data packet 1500 is shown in Figure 15. This format corresponds to either an asynchronous packet or an isochronous packet. The data packet format is a basic IEEE-1394 streaming packet. One frame of data for a single stream is broken up into a series of packets of this format. The shaded fields indicate those defined by the IEEE-1394 standard. The "data_length" field 1502 includes the length in bytes of the payload section 1512 of the packet. The "tag" field 1504 includes the value 00b which indicates that the data is unformatted. The "channel" field 1506 includes a value that indicates the channel assigned to the talker and listener for transmission of this packet. The "tcode" field 1508 includes the value 1010b which indicates that this packet is a streaming packet. The "sy" field 1510 includes the value 0000b to indicate that this packet is a data packet. The data packet also may include fields that enable error detection and correction ("header_CRC" 1514 and "data_CRC" 1516).

[0088] The payload 1512 of a data packet includes 1 to 1024 data quadlets (meaning 4 bytes of data) for an isochronous packet. There are 1 to 512 data quadlets for an asynchronous packet. In general, a talker formats the data for a data interval into N packets of data of the same fixed size to fit the payload of the data packet. This size can be the maximum size allowed by the transmission mode being used to efficiently use bandwidth. For example, in the IEEE-1394a standard, this maximum may be either 2,048 or 4,096 bytes. If the length of the data is not evenly divisible by this quantity, then it is permissible to generate a short packet at the end of a stream.

[0089] For example, assume an uncompressed, full frame of NTSC 4:2:2 YUV format video data is transmitted using isochronous mode of IEEE-1394. The frame is transmitted as two fields, there are (240 lines * 720 pixels * 2 bytes-per-pixel =) 345,600 bytes per field. To send a single field in 4096-byte packets, the number of packets used is computed as: 345,600 bytes per field / 4096 bytes per packet = 84 packets with a remainder 1536 bytes, indicating that 84 4096-byte packets and one 1536-byte packet will be transmitted.

[0090] It is also permissible for a talker to transmit a short packet at the beginning of a stream if there are memory page alignment constraints. For instance, if the talker is generating the packets for a stream out of a host buffer that is not aligned on a 4096-byte boundary, then each packet in the stream would cross a page boundary, resulting in poor direct memory access performance. To remedy this, the talker can generate a short packet at the beginning in order to insure that the remaining packets are generated out of buffers that are aligned on 4096-byte boundaries. For example, if the memory buffer for a field to transmit is located at address 119403A4H, then the next 4096-byte boundary is at address 11941000H. That means that there are 119403A4H - 11941000H = C5CH, or 3,164 (decimal) bytes, until the next page boundary. Thus, the first packet would be 3,164 bytes generated out of address 119403A4H, the second packet would be 4,096 bytes generated out of address 11941000H, the third packet would be 4,096 bytes generated out of address 11942000H, and so on.

[0091] The host may transmit null packets for any unused isochronous cycle. During data transmission, one or more null packets may be sent in the data intervals between the transmission of data for each data interval. An example format for a null packet 1600 is shown in Figure 16. This format corresponds to either an asynchronous packet or an isochronous packet. The shaded fields indicate those defined by the IEEE-1394 standard. The "data_length" field 1602 may include any value, including the value 0. If the data_length is zero, then the payload portion 1614 and error correction data 1616 are omitted from the packet. If the data length is non-zero, then dummy data is in the payload 1614 following the header with a number of quadlets corresponding to the value specified in field 1602. The "tag" field 1604 includes the value 00b which indicates that the data is unformatted. The "channel" field 1606 includes a value that indicates the channel assigned to the talker and listener for transmission of this packet. The "tcode" field 1608 includes the value 1010b which indicates that this packet is a streaming packet. The "sy" field 1610 includes the value 0000b. An optional payload portion 1614 of this packet, if present, is ignored by the listener. Fields that enable error detection and correction ("header_CRC" 1612 and "data_CRC" 1616) may be provided.

LINKPLAY EXHIBIT 1004
Page 1866 of 4822

[0092] Sequencing and timing of packet transmissions for host-to-device and device-to-host transfers using this protocol will now be described.

[0093] As noted above, to transmit data from a host to a device, a host and device first set up parameters of the communication using the control protocol described below. The host then waits, for example by sending null packets, until a data interval begins. These null packets are ignored by the device. At the beginning of the next data interval, a synchronization packet is sent from the device to the host indicating that a new data interval has begun. After receiving the synchronization packet, the host may send data. In one embodiment, a preamble packet is sent from the host to indicate to the device which streams will be transmitted during the data interval, in what order they will be sent, and how many packets will be used to transport each stream. The device processes the preamble packet to set up appropriate parameters of the device. Data packets then are sent by the host. These data packets are then received and processed by the device. After the data for the data interval has been sent, the host then waits for the next data interval. For example, the host may send null packets to fill bus cycles until the next data interval begins. This sequence of packets is repeated for each data interval.

[0094] As noted above, to transmit data from a device to a host, a host and device first set up parameters of the communication using the control protocol described below. The host then waits, for example by sending null packets, until a data interval begins. These null packets are ignored by the device. At the beginning of the next data interval, a synchronization packet is sent from the device to the host indicating that a new data interval has begun. After sending the synchronization packet, the device may send data. In one embodiment, a preamble packet is sent from the device to indicate to the host which streams will be transmitted during the data interval, in what order they will be sent, and how many packets will be used to transport each stream. The host processes the preamble packet to set up appropriate parameters. Data packets then are sent by the device. These data packets are then received and processed by the host. After the data for the data interval has been sent, the device then waits for the next data interval. For example, the device may send null packets to fill bus cycles until the next data interval begins. This sequence of packets is repeated for each data interval.

[0095] In addition to the data transport protocol, a control protocol is provided. Two types of control information are conveyed by the control protocol: commands which control the modes and parameters of the peripheral device hardware (herein called "break-out box (BOB) control"), and commands which control the operation of a DV camcorder or VTR attached to the peripheral device's IEEE-1394 compliant port (called device control).

[0096] BOB control commands may use asynchronous packets in the IEEE-1394 protocol. Device control commands may use the AV/C command set, an industry standard for controlling camcorders and VTRs. AV/C commands intended for the external device are forwarded directly to that device by the peripheral device hardware, and responses are returned to the host in a similar manner. The AV/C commands are defined in several documents, which include: AV/C Digital Interface Command Set General Specification (version 3.0) and AV/C Tape Recorder/Player Subunit Specification. Thus, for device control commands, the communication protocol complies with the existing IEC 61883-1 and IEEE-1394 Trade Association AV/C specifications. The peripheral device relays commands and responses between the host and external device as though the host were directly interacting with the external device.

[0097] BOB control employs a command set that includes command sequences that are transmitted over the IEEE-1394 compliant bus between the host and the peripheral device. Each command sequence includes a command frame transmitted from the host to the peripheral device, and a corresponding response frame transmitted from the peripheral device to the host. Command frames include a command code and optional parameters. Response frames include a success code and optional data values requested by the command.

[0098] Using the IEEE-1394 protocol, a Write Data Block Request format packet is used, having a "tcode" of "1h". The destination offset specified by a packet designates the registers specified for receiving the commands and responses in this protocol. The payload of such a packet includes up to 1024 bytes. The first two bytes of the payload includes a fixed value that represents that the packet includes a BOB control command or response. The next two bytes indicate the packet length. Two more bytes provide the information, such as parameters for a command, status of a command or other information. An additional byte is reserved. The last byte of the payload conveys status information and determines whether the packet is part of a command or a response, and whether it is part of a single- or multi-packet command.

[0099] The BOB control allows the host to control and query the modes, status and parameters of the peripheral device. The various commands depend on the implementation of the peripheral device, its modes and available registers for status information and control settings. To simplify operation of multiple devices connected to the same bus and same periodic reference signal, the BOB control protocol should allow for a command to instruct the peripheral device not to send flow control messages in response to frame boundaries defined by the periodic reference signal.

[0100] The foregoing describes an example device and communication protocol for communication between the device and a host over a serial bus that allows control of the flow of data transferred between the host and the device so that synchronization to a periodic reference signal can be achieved. The protocol involves transferring flow control messages between the peripheral device and the host computer, allowing the peripheral device to control how the host computer sends the uncompressed audio and video data. The audio and video data may be sent as separate streams. Using the

LINKPLAY EXHIBIT 1004
Page 1867 of 4822

IEEE-1394 protocol, the flow control messages are sent using an asynchronous protocol, whereas uncompressed video and audio data are sent using an isochronous protocol.

[0101] Further, separate streams of uncompressed video and audio data are transferred from the host computer over a serial bus to the peripheral device, where the peripheral device performs operations on the data such as encoding the data into a standard format, such as DV and MPEG, or generating an analog video signal. Decoding of standard format data is not required to generate such an analog video signal, resulting in less processing. Further, when the host computer is used to process digital video and audio data for editing audiovisual programs, the resources on the host computer are not used for encoding video data. Therefore the host computer resources can be used for more creative operations on the video and audio data. The peripheral device also may output standard format data using a standard protocol over a standard bus. For example, the peripheral device may generate DV format data from the uncompressed audio and video data, and output the DV format data using a standard DV transport protocol over an IEEE-1394 compliant serial bus.

[0102] The peripheral device also can be used to synchronize an output of another device, such as a transcoder connected to the peripheral device via a serial bus such as an IEEE-1394 compliant bus, to a periodic reference signal received by the peripheral device. For example, the peripheral device may perform DV encoding of the uncompressed audio and video data received from the host computer. The peripheral device then transfers this DV format data over the serial bus using a standard protocol to the other device. The operation of this other transcoder may be genlocked to a periodic reference signal provided to the peripheral device.

[0103] The host may be configured to receive a stream of audio from the peripheral device during transfer of an audio-video stream to the peripheral device. Such an audio stream would be data received through the peripheral device which may be used to implement audio dubbing or punch-in. In this embodiment, the audio-video stream is sent from the host to the peripheral device as a stream of isochronous packets. The audio data for dubbing or punch-in is sent from the peripheral device to the host as a stream of asynchronous packets. The peripheral device may mix the audio stream with the audio in the received audio video stream for playback through or output to another device.

[0104] For example, referring to Figs. 10A and 10B, audio codec 1006 receives input audio (at 1040, 1041) and produces digital audio data for storage in the media hub 1028. Meanwhile, the IEEE-1394 interface 1004 receives the audio-video data stream and provides it to the media hub 1028. After receipt of a frame of data, the media hub 1028 mixes the received audio data from codec 1006 and from interface 1004 and outputs the mixed data to the codec 1006 and HeadPhone DAC 1012 to generate audio outputs. The microprocessor 1022 is instructed to periodically retrieve audio data from the media hub 1028 to send the data as multiple asynchronous packets to the host over the interface 1051.

[0105] It also should be understood that the host computer and peripheral device as described herein may be any kind of source and recipient connected over a bus to communicate temporal data in synchronization with a periodic reference signal.

[0106] It also should be understood that the host computer may store computer program instructions on a computer readable medium that, when executed by the host computer, cause the host computer to transfer data from the host computer to the peripheral device or to cause the host computer to receive data from the peripheral device. Similarly, the controller 1020 (Figs. 10A-10B) on the device may accessed stored computer program instructions to cause the device to perform a variety of operations.

[0107] Having now described an example embodiment, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by way of example only. Numerous modifications and other embodiments are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention. Embodiments will be further described and illustrated by way of the following numbered clauses:

1. A method for transferring data from a first device to a second device over a bus, performed by the second device for each data interval defined by a periodic reference signal processed by the second device, and comprising:

sending a synchronization packet over the bus to the first device for the data interval in response to the periodic reference signal;
receiving a plurality of packets over the bus from the first device after receipt by the first device of the synchronization packet, wherein the packets include data for the data interval;
repeating the steps of sending and receiving for each data interval defined by the periodic reference signal.

2. The method of clause 1, wherein the plurality of packets comprises:

a preamble packet indicating a number of data packets to be sent; and
one or more data packets.

3. The method of clause 2, wherein the data comprises a plurality of streams of data, and wherein the preamble packet indicates streams to be transmitted during the interval, an order of transmission of the data streams, and a

LINKPLAY EXHIBIT 1004
Page 1868 of 4822

number of packets used for each stream.

4. The method of clause 2, wherein the plurality of packets further comprises zero or more null packets after the one or more data packets for a remainder of the data interval.

5. The method of clause 2, wherein data packets comprises video data packets.

6. The method of clause 2, wherein data packets comprises audio data packets.

7. The method of clause 2, wherein data packets comprises video and audio data packets.

8. The method of clause 1, wherein the plurality of packets comprises zero or more null packets after the one or more data packets for a remainder of the data interval.

9. The method of clause 1, wherein the data interval corresponds to a frame of video data.

10. The method of clause 1, wherein the data interval corresponds to a field of video data.

11. The method of clause 1, wherein the bus is a serial bus.

12. The method of clause 11, wherein the serial bus is an IEEE-1394 compliant bus.

13. The method of clause 12, wherein the synchronization packet is an asynchronous packet.

14. The method of clause 12, wherein the plurality of packets are isochronous packets.

15. The method of clause 1, wherein the periodic reference signal is received by the second device.

16. The method of clause 1, wherein the periodic reference signal is generated by the second device.

17. The method of clause 1 further comprising:

receiving from the first device a command to start transfer of the data; and receiving from the first device zero or more null packets over the bus until a first synchronization packet for a first data interval is sent by the second device.

18. The method of clause 1, further comprising:

receiving the data in a buffer at the second device; and
outputting the data from the second device in synchronization with the periodic reference signal.

19. The method of clause 18, wherein outputting the data comprises encoding and outputting an analog video signal.

20. The method of clause 18, wherein outputting comprises generating DV format video data.

21. A method for transferring data from a first device to a second device over a bus, performed by the second device for each data interval defined by a periodic reference signal processed by the second device, and comprising:

sending a synchronization packet over the bus to the first device for the data interval in response to the periodic reference signal;
receiving a plurality of packets over the bus from the first device after receipt by the first device of the synchronization packet, wherein the packets include data for the data interval;
after receiving the plurality of packets, waiting for a subsequent data interval defined by the periodic reference signal.

22. A device for connection to source of data over a bus, comprising:

means for processing a periodic reference signal defining a plurality of data intervals,

LINKPLAY EXHIBIT 1004
Page 1869 of 4822

means for communicating a synchronization packet over the bus to the source of data for each data interval in response to the periodic reference signal;
means for accepting a plurality of packets over the bus from the source of data after receipt by the source of data of the synchronization packet, wherein the packets include data for the data interval; and
means for repetitively operating the means for communicating and the means for accepting for each data interval defined by the periodic reference signal.

23. A device for connection to source of data over a bus, comprising:

a bus interface for sending packets to and receiving packets from the source of data over the bus;
a controller having an input for receiving a signal related to a periodic reference signal defining a plurality of data intervals, and having an output for providing instructing the bus interface to send a synchronization packet over the bus to the source of data for each data interval in response to the periodic reference signal; and
a buffer connected to receive data for the data interval from a plurality of packets received through the bus interface over the bus from the source of data after receipt by the source of data of the synchronization packet.

24. A method for transferring data from a first device to a second device over a bus, comprising, for each data interval defined by a periodic reference signal processed by the first device:

the first device sending a synchronization packet over the bus to the second device for the data interval;
the first device sending a plurality of packets over the bus to the second device after sending the synchronization packet, wherein the packets include data for the data interval; and
the first device waiting until the next data interval according to the periodic reference signal.

25. The method of clause 24, wherein the plurality of packets comprises:

a preamble packet indicating a number of data packets to be sent; and
one or more data packets.

26. The method of clause 25, wherein the data comprises a plurality of streams of data, and wherein the preamble packet indicates streams to be transmitted during the interval, an order of transmission of the data streams, and a number of packets used for each stream.

27. The method of clause 25, wherein waiting comprises sending zero or more null packets over the bus to the second device for a remainder of the data interval.

28. The method of clause 25, wherein data packets comprises video data packets.

29. The method of clause 25, wherein data packets comprises audio data packets.

30. The method of clause 25, wherein data packets comprises video and audio data packets.

31. The method of clause 24, wherein waiting comprises sending zero or more null packets over the bus to the second device for a remainder of the data interval.

32. The method of clause 24, wherein the data interval corresponds to a frame of video data.

33. The method of clause 24, wherein the data interval corresponds to a field of video data.

34. The method of clause 24, wherein the bus is a serial bus.

35. The method of clause 34, wherein the serial bus is an IEEE-1394 compliant bus.

36. The method of clause 35, wherein the synchronization packet is an asynchronous packet.

37. The method of clause 35, wherein the plurality of packets are isochronous packets.

38. The method of clause 24, wherein the periodic reference signal is received by the first device.

LINKPLAY EXHIBIT 1004
Page 1870 of 4822

39. The method of clause 24, wherein the periodic reference signal is generated by the first device.

40. The method of clause 24, further comprising:

receiving from the second device a command to start transfer of the data; and
sending zero or more null packets over the bus to the second device until a first synchronization packet for a first data interval is sent.

41. The method of clause 24, further comprising:

the first device receiving the data in synchronization with the periodic reference signal into a buffer; and the first device transferring the data from the buffer over the bus to the second device.

42. The method of clause 41, wherein receiving the data comprises receiving and decoding an analog video signal.

43. The method of clause 41, wherein receiving comprises receiving DV format video data.

44. A device for connection to a recipient of data over a bus, comprising:

means for receiving a periodic reference signal defining a plurality of data intervals,
means for communicating a synchronization packet over the bus to the recipient of data for each data interval in response to the periodic reference signal;
means for transferring a plurality of packets over the bus to the recipient of data after receipt by the recipient of data of the synchronization packet, wherein the packets include data for the data interval; and
means for repetitively operating the means for communicating and the means for transferring for each data interval defined by the periodic reference signal.

45. A device for connection to a recipient of data over a bus, comprising:

a bus interface for sending packets over the bus connected to the recipient of data;
a buffer having an input for receiving data and connected to provide data to the bus interface for transmission as a plurality of packets over the bus to the recipient of data; and
a controller having an input for receiving a signal related to a periodic reference signal defining a plurality of data intervals, and having an output for providing an indication to the bus interface to send a synchronization packet over the bus to the recipient of data for each data interval in response to the periodic reference signal.

46. A system for connection to a host computer over a bus, comprising:

a peripheral device comprising:

means for receiving a periodic reference signal defining a plurality of data intervals,
means for receiving video data for the plurality of data intervals;
means for communicating a synchronization packet over the bus to the host computer for each data interval in response to the periodic reference signal;
means for transferring a plurality of packets over the bus to the host computer after receipt by the host computer of the synchronization packet, wherein the packets include video data for the data interval; and
means for repetitively operating the means for communicating and the means for transferring for each data interval defined by the periodic reference signal; and

a video device, comprising:

an input for receiving and synchronizing with the periodic reference signal; and
means for providing the video data to the peripheral device in synchronization with the periodic reference signal.

47. A system for connection to a host computer over a bus, comprising:

a peripheral device comprising:

LINKPLAY EXHIBIT 1004
Page 1871 of 4822

means for receiving a periodic reference signal defining a plurality of data intervals,
means for communicating a synchronization packet over the bus to the host computer for each data interval in response to the periodic reference signal;
means for accepting a plurality of packets over the bus from the host computer after receipt by the host computer of the synchronization packet, wherein the packets include video data for the data interval; and
means for repetitively operating the means for communicating and the means for transferring for each data interval defined by the periodic reference signal; and
means for providing video data for the plurality of data intervals; and

a video device, comprising:

an input for receiving and synchronizing with the periodic reference signal; and
means for receiving the video data provided by the peripheral device in synchronization with the periodic reference signal.

48. A method for transferring data from a first device to a second device over a bus, performed by the first device for each data interval defined by a periodic reference signal processed by the second device, comprising:

receiving a synchronization packet over the bus from the second device for the data interval;
after receiving the synchronization packet, sending a plurality of packets over the bus to the second device, wherein the packets include data for the data interval;
after sending the plurality of packets for the data interval, waiting until receipt of a synchronization packet over the bus from the second device for a subsequent data interval.

49. The method of clause 48, wherein the plurality of packets comprises:

a preamble packet indicating a number of data packets to be sent; and one or more data packets.

50. The method of clause 49, wherein the data comprises a plurality of streams of data, and wherein the preamble packet indicates streams to be transmitted during the interval, an order of transmission of the data streams, and a number of packets used for each stream.

51. The method of clause 49, wherein waiting comprises sending zero or more null packets over the bus to the second device for a remainder of the data interval.

52. The method of clause 49, wherein data packets comprises video data packets.

53. The method of clause 49, wherein data packets comprises audio data packets.

54. The method of clause 49, wherein data packets comprises video and audio data packets.

55. The method of clause 48, wherein waiting comprises sending zero or more null packets over the bus to the second device for a remainder of the data interval.

56. The method of clause 48, wherein the data interval corresponds to a frame of video data.

57. The method of clause 48, wherein the data interval corresponds to a field of video data.

58. The method of clause 48, wherein the bus is a serial bus.

59. The method of clause 58, wherein the serial bus is an IEEE-1394 compliant bus.

60. The method of clause 59, wherein the synchronization packet is an asynchronous packet.

61. The method of clause 59, wherein the plurality of packets are isochronous packets.

62. The method of clause 48, further comprising:

LINKPLAY EXHIBIT 1004
Page 1872 of 4822

sending to the second device a command to start transfer of the data; and
sending zero or more null packets over the bus to the second device until a first synchronization packet for a first data interval is received.

63. An apparatus for transferring data to a device over a bus, comprising:

means for receiving a synchronization packet over the bus from the device for each data interval defined by a periodic reference signal processed by the device;
means, operative after receiving the synchronization packet, for sending a plurality of packets over the bus to the device, wherein the packets include data for the data interval; and
means, operative after sending the plurality of packets for the data interval, for waiting until receipt of a synchronization packet over the bus from the device for a subsequent data interval.

64. An apparatus for transferring data to a device over a bus, comprising:

a bus interface for sending packets to and receiving packets from the device over the bus;
a buffer connected to transfer data to the bus interface for transmission as a plurality of packets; and
a controller comprising:

means for processing synchronization packets received by the bus interface from the device for each data interval defined by a periodic reference signal;
means, operative after receiving the synchronization packet, for causing a plurality of packets to be transferred by the bus interface over the bus between the apparatus and the device, wherein the packets include data for the data interval; and
means, operative after transferring the plurality of packets for the data interval, for waiting until receipt of a synchronization packet over the bus from the device for a subsequent data interval.

65. A computer program product, comprising:

a computer readable medium;
computer program instructions stored on the computer readable medium that, when executed by a host computer, cause the host computer to perform a method for transferring data from the host computer to a peripheral device over a bus, performed by the host computer for each data interval defined by a periodic reference signal and comprising:

receiving a synchronization packet over the bus from the peripheral device for the data interval;
after receiving the synchronization packet, sending a plurality of packets over the bus to the peripheral device, wherein the packets include data for the data interval;
after sending the plurality of packets for the data interval, waiting until receipt of a synchronization packet over the bus from the peripheral device for a subsequent data interval.

66. A method for transferring data from a first device to a second device over a bus, performed by the second device for each data interval defined by a periodic reference signal processed by the first device, comprising:

receiving a synchronization packet over the bus from the first device for the data interval;
receiving a plurality of packets over the bus from the first device after receiving the synchronization packet, wherein the packets include data for the data interval; and
after receiving the plurality of packets, waiting until receipt of a synchronization packet over the bus from the first device for a subsequent data interval.

67. A computer program product, comprising:

a computer readable medium;
computer program instructions stored on the computer readable medium that, when executed by a host computer, cause the host computer to perform a method for transferring data from a peripheral device to the host computer over a bus, comprising, for each data interval defined by a periodic reference signal processed by the peripheral device:

LINKPLAY EXHIBIT 1004
Page 1873 of 4822

receiving a synchronization packet over the bus from the peripheral device for the data interval;
receiving a plurality of packets over the bus from the peripheral device after receiving the synchronization packet, wherein the packets include data for the data interval; and
after receiving the plurality of packets, waiting until receiving a synchronization packet over the bus from the peripheral device for a subsequent data interval.

68. A method for transferring data from a host computer to a peripheral device over a bus, comprising:

the peripheral device sending a synchronization packet over the bus to the host computer for each data interval according to a periodic reference signal;
the host computer sending a plurality of packets over the bus to the peripheral device after receipt of the synchronization packet, wherein the packets include data for the data interval; and
after sending the plurality of packets for the data interval, the host computer waiting until a synchronization packet for a subsequent data interval is received.

69. The method of clause 68, wherein the plurality of packets comprises:

a preamble packet indicating a number of data packets to be sent; and
one or more data packets.

70. The method of clause 69, wherein the data comprises a plurality of streams of data, and wherein the preamble packet indicates streams to be transmitted during the interval, an order of transmission of the data streams, and a number of packets used for each stream.

71. The method of clause 69, wherein waiting comprises sending zero or more null packets after the one or more data packets for a remainder of the data interval.

72. The method of clause 69, wherein data packets comprises video data packets.

73. The method of clause 69, wherein data packets comprises audio data packets.

74. The method of clause 69, wherein data packets comprises video and audio data packets.

75. The method of clause 68, wherein waiting comprises sending zero or more null packets after the one or more data packets for a remainder of the data interval.

76. The method of clause 68, wherein the data interval corresponds to a frame of video data.

77. The method of clause 68, wherein the data interval corresponds to a field of video data.

78. The method of clause 68, wherein the bus is a serial bus.

79. The method of clause 78, wherein the serial bus is an IEEE-1394 compliant bus.

80. The method of clause 79, wherein the synchronization packet is an asynchronous packet.

81. The method of clause 79, wherein the plurality of packets are isochronous packets.

82. The method of clause 68, wherein the periodic reference signal is received by the peripheral device.

83. The method of clause 68, wherein the periodic reference signal is generated by the peripheral device.

84. The method of clause 68, further comprising:

the host computer sending a command to start transfer of the data; and
the host computer sending zero or more null packets over the bus until a first synchronization packet for a first data interval is sent by the peripheral device.

LINKPLAY EXHIBIT 1004
Page 1874 of 4822

85. The method of clause 68, further comprising:

the peripheral device receiving the data in a buffer; and
the peripheral device outputting the data in synchronization with the periodic reference signal.

86. The method of clause 85, wherein outputting the data comprises encoding and outputting an analog video signal.

87. The method of clause 85, wherein outputting comprises generating DV format video data.

88. A system comprising:

a host computer;
a peripheral device connected to the host computer via a bus;

wherein the peripheral device comprises means for sending a synchronization packet over the bus to the host computer for each data interval according to a periodic reference signal;
wherein the host computer comprises means for sending a plurality of packets over the bus to the peripheral device after receipt of the synchronization packet, wherein the packets include data for the data interval, and means, operative after sending the plurality of packets for the data interval, for waiting until a synchronization packet for a subsequent data interval is received.

89. A system comprising:

a host computer comprising:

storage for storing compressed video data and associated audio data;
an editing system for editing a program using the video data and the audio data and effects to be performed on the video data;
a playback system for playing back the edited program in full motion and at full resolution, including means for decompressing the compressed video data and means for processing the effects on the decompressed video data during playback;
a peripheral device connected to the host computer via a bus, wherein the peripheral device comprises:

means for controlling flow of decompressed video and audio data over the bus from the host computer to synchronize flow with a periodic reference signal;

wherein the host computer comprises:

means for sending the processed decompressed video with effects and the audio data from the playback system during playback over the bus to the peripheral device in response to flow control by the peripheral device;

wherein the peripheral device further comprises:

means for compressing the decompressed video and audio data to create a compressed data stream during playback; and
means for outputting the compressed data stream to a recording device synchronized with the periodic reference signal during playback.

90. A peripheral device for providing a bridge between a host computer and another device, wherein the peripheral device connects to the host computer with first serial bus and connects to the other device with a second serial bus, wherein the peripheral device comprises:

means for implementing a protocol over the first serial bus for communication between the host computer and the peripheral device
means for implementing the protocol over the second serial bus for communication between the peripheral device and the other device;
means for communicating information from the other device to the host computer to allow the host computer to

LINKPLAY EXHIBIT 1004
Page 1875 of 4822

communicate packets directed to the other device using the protocol over the first serial bus;
means for communicating information from the host computer to the other device to allow the other device to communicate packets directed to the host computer using the protocol over the second serial bus;
means for processing packets received over the first serial bus, including:

means for determining if the packet is directed to the other device and for transferring the packet to the other device;
means for determining if the packet is directed to the peripheral device and for processing the packet at the peripheral device; and
means for determining if the packet is to be processed by the peripheral device but is directed to the other device, and for processing the packet to create one or more new packets and for transferring the one or more new packets to the other device; and
means for processing packets received over the second serial bus, including:

means for determining if the packet is directed to the host computer and for transferring the packet to the host computer;
means for determining if the packet is directed to the peripheral device and for processing the packet at the peripheral device; and
means for determining if the packet is to be processed by the peripheral device but is directed to the host computer, and for processing the packet to create one or more new packets and for transferring the one or more new packets to the host computer.

Claims

1. A method for transferring temporal data from a first device (100) to a second device (102) over a bus, performed by the second device for each data interval defined by a periodic reference signal (110) processed by the second device, and comprising:

sending a synchronization packet over the bus to the first device for the data interval in response to the periodic reference signal;
receiving a plurality of packets over the bus from the first device after receipt by the first device of the synchronization packet, wherein the packets include data for the data interval;
repeating the steps of sending and receiving for each data interval defined by the periodic reference signal.

2. The method of claim 1, wherein the plurality of packets comprises:

a preamble packet indicating a number of data packets to be sent; and
one or more data packets.

3. The method of claim 2, wherein the data comprises a plurality of streams of data, and wherein the preamble packet indicates streams to be transmitted during the interval, an order of transmission of the data streams, and a number of packets used for each stream.

4. The method of any preceding claim, wherein the plurality of packets further comprises zero or more null packets after the one or more data packets for a remainder of the data interval.

5. The method of any preceding claim, wherein data packets comprises video data packets.

6. The method of any preceding claim, wherein data packets comprises audio data packets.

7. The method of any preceding claim wherein data packets comprises video and audio data packets.

8. The method of any preceding claim, wherein the data interval corresponds to a frame of video data.

9. The method of any of claims 1 to 7, wherein the data interval corresponds to a field of video data.

10. The method of any preceding claim, wherein the bus is a serial bus.

LINKPLAY EXHIBIT 1004
Page 1876 of 4822

11. The method of claim 10, wherein the serial bus is an IEEE-1394 compliant bus.

12. The method of any preceding claim, wherein the synchronization packet is an asynchronous packet.

13. The method of any preceding claim, wherein the plurality of packets are isochronous packets.

14. The method of any preceding claim, wherein the periodic reference signal is received by the second device.

15. The method of any of claims 1 to 13, wherein the periodic reference signal is generated by the second device.

16. The method of any preceding claim further comprising:

receiving from the first device a command to start transfer of the data; and
receiving from the first device zero or more null packets over the bus until a first synchronization packet for a first data interval is sent by the second device.

17. The method of any preceding claim, further comprising:

receiving the data in a buffer at the second device; and
outputting the data from the second device in synchronization with the periodic reference signal.

18. The method of claim 17, wherein outputting the data comprises encoding and outputting an analog video signal.

19. The method of claim 17, wherein outputting comprises generating DV format video data.

20. A device (102), for connection to source of temporal data over a bus, comprising:

a bus interface (104) for sending packets to and receiving packets from the source (100) of data over the bus;
a controller having an input for receiving a signal related to a periodic reference signal (110) defining a plurality of data intervals, and having an output for providing instructing the bus interface to send a synchronization packet over the bus to the source of data for each data interval in response to the periodic reference signal; and
a buffer connected to receive data for the data interval from a plurality of packets received through the bus interface over the bus from the source of data after receipt by the source of data of the synchronization packet.

21. A system for transferring temporal data comprising:

a host computer (100);
a peripheral device (102) connected to the host computer via a bus (104);

wherein the peripheral device comprises means for sending a synchronization packet over the bus to the host computer for each data interval according to a periodic reference signal (110);
wherein the host computer comprises means for sending a plurality of packets over the bus to the peripheral device after receipt of the synchronization packet, wherein the packets include data for the data interval, and means, operative after sending the plurality of packets for the data interval, for waiting until a synchronization packet for a subsequent data interval is received.

Patentansprüche

1. Verfahren zum Übertragen von zeitlichen Daten von einer ersten Vorrichtung (100) an eine zweite Vorrichtung (102) über einen Bus, das von der zweiten Vorrichtung für jedes Datenintervall durchgeführt wird, das von einem periodischen Bezugssignal (110) definiert wird, das von der zweiten Vorrichtung verarbeitet wird, und wobei das Verfahren Folgendes umfasst:

Senden eines Synchronisationspakets über den Bus an die erste Vorrichtung für das Datenintervall als Antwort auf das periodische Bezugssignal;
Empfangen von mehreren Paketen über den Bus von der ersten Vorrichtung, nachdem das Synchronxeations-paket von der ersten Vorrichtung empfangen wurde, wobei die Pakete Daten für das Datenintervall enthalten;

LINKPLAY EXHIBIT 1004
Page 1877 of 4822

Wiederholen der Schritte des Sendens und Empfangens für jedes Datenintervall, das von dem periodischen Bezugssignal definiert wird.

2. Verfahren nach Anspruch 1, wobei die mehreren Pakete Folgendes umfassen:

ein Präambelpaket, das eine Anzahl zu sendender Datenpakete anzeigt; und
ein oder mehrere Datenpakete.

3. Verfahren nach Anspruch 2, wobei die Daten mehrere Datenströme umfassen und wobei das Präambelpaket während des Intervalls zu übermittelnde Ströme, eine Reihenfolge der Übermittlung der Datenströme und eine Anzahl von Paketen, die für jeden Strom verwendet werden, anzeigt.

4. Verfahren nach einem der vorhergehenden Ansprüche, wobei die mehreren Pakete weiterhin null oder mehr Nullpakete nach dem einen oder den mehreren Datenpaketen für einen Rest des Datenintervalls umfasst.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei Datenpakete Videodatenpakete umfassen.

6. Verfahren nach einem der vorhergehenden Ansprüche, wobei Datenpakete Audiodatenpakete umfassen.

7. Verfahren nach einem der vorhergehenden Ansprüche, wobei Datenpakete Video- und Audiodatenpakete umfassen.

8. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Datenintervall einem Videodatenrahmen entspricht.

9. Verfahren nach einem der Ansprüche 1 bis 7, wobei das Datenintervall einem Videodatenfeld entspricht.

10. Verfahren nach einem der vorhergehenden Ansprüche, wobei der Bus ein serieller Bus ist.

11. Verfahren nach Anspruch 10, wobei der serielle Bus ein IEEE-1394-konformer Bus ist.

12. Verfahren, nach einem der vorhergehenden Ansprüche, wobei das Synchronisationspaket ein asynchrones Paket ist.

13. Verfahren nach einem der vorhergehenden Ansprüche, wobei die mehreren Pakete isochrone Pakete sind.

14. Verfahren nach einem der vorhergehenden Ansprüche, wobei das periodische Bezugssignal von der zweiten Vorrichtung empfangen wird.

15. Verfahren nach einem der Ansprfiche 1 bis 13, wobei das periodische Signal von der zweiten Vorrichtung erzeugt wird.

16. Verfahren nach einem der vorhergehenden Ansprüche, das weiterhin Folgendes umfasst:

Empfangen eines Befehls, die Übertragung der Daten zu starten, von der ersten Vorrichtung und
Empfangen von null oder mehr Nullpaketen von der ersten Vorrichtung über den Bus, bis von der zweiten Vorrichtung ein erstes Synchronisationspaket für ein erstes Datenintezvall gesendet wird.

17. Verfahren einem der vorhergehenden Ansprüche, das weiterhin Folgendes umfasst:

Empfangen der Daten in einem Pufferspeicher an der zweiten Vorrichtung und
Ausgeben der Daten von der zweiten Vorrichtung in Synchronisation mit dem periodischen Bezugssignal.

18. verfahren nach Anspruch 17, wobei das Ausgeben der Daten das Kodieren und Ausgeben eines analogen Videosignals umfasst.

19. Verfahren nach Anspruch 17, wobei das Ausgeben das Erzeugen von Videodaten im DV-Format umfasst.

20. Vorrichtung (102) zur Verbindung mit einer Quelle von zeitlichen Daten über einen Bus, die Folgendes umfasst:

LINKPLAY EXHIBIT 1004
Page 1878 of 4822

eine Busschnittstelle (104) zum Senden von Paketen an die Datenquelle (100) und Empfangen von Paketen von der Datenquelle über den Bus;
einen Controller mit einem Eingang zum Empfangen eines Signals, das mit einem periodischen Bezugssignal (110) in Beziehung steht, das mehrere Datenintervalle definiert, und mit einem Ausgang zum Liefern einer Azxweisung an die Busschnittstelle, für jedes Datenintervall als Antwort auf das periodische Bezugssignal ein Synchronisationspaket über den Bus an die Datenquelle zu senden; und
einen Pufferspeicher, der angeschlossen ist, um Daten für das Datenintervall von mehreren Paketen zu empfangen, die mittels der Busschnittstelle über den Bus von der Datenquelle empfangen wurden, nachdem das Synchronisationspaket von der Datenquelle empfangen wurde.

21. System zum übertragen von zeitlichen Daten, das Folgendes umfasst:

einen Hostrechner (100);
ein Peripheriegerät (102), das über einen Bus (104) an den Hostrechner angeschlossen ist,

wobei das Peripheriegerät Mittel zum Senden eines Synchronisationspakets über den Bus an den Hostrechner für jedes Datenintervall gemäß einem periodischen Bezugssignal (110) umfasst;
wobei der Hostrechner Mittel zum Senden von mehreren Paketen über den Bus an das Peripheriegerit nach Empfang des Synchronisationspakets, wobei die Pakete Daten für das Datenintervall enthalten, und Mittel, die nach dem Senden der mehreren Pakete für das Datenintervall funktionsfähig sind, zum Warten, bis ein Synchronisationspaket für ein folgendes Datenintervall empfangen wird, umfasst.

Revendications

1. Procédé servant à transférer des données temporelles à partir d'un premier dispositif (100) vers un second dispositif (102) sur un bus, exécuté par le second dispositif pour chaque intervalle de données défini par un signal de référence périodique (110) traité par le second dispositif, et comprenant :

l'envoi d'un paquet de synchronisation sur le bus vers le premier dispositif pour l'intervalle de données en réponse au signal de référence périodique ;
la réception d'une pluralité de paquets sur le bus à partir du premier dispositif après réception par le premier dispositif du paquet de synchronisation, dans lequel les paquets comprennent des données pour l'intervalle de données ;
la répétition des étapes d'envoi et de réception pour chaque intervalle de données défini par le signal de référence périodique.

2. Procédé selon la revendication 1, dans lequel la pluralité de paquets comprend :

un paquet de préambule indiquant un nombre de paquets de données à envoyer ; et un ou plusieurs paquets de données.

3. Procédé selon la revendication 2, dans lequel les données comprennent une pluralité de flux de données, et dans lequel le paquet de préambule indique les flux à transmettre pendant l'intervalle, un ordre de transmission des flux de données et un nombre de paquets utilisés pour chaque flux.

4. Procédé selon l'une quelconque des revendications précédentes, dans lequel la pluralité de paquets comprend en outre zéro ou plusieurs paquets nuls qui suivent le ou les plusieurs paquets de données pour l'intervalle de données restante.

5. Procédé selon l'une quelconque des revendications précédentes, dans lequel les paquets de données comprennent des paquets de données vidéo.

6. Procédé selon l'une quelconque des revendications précédentes, dans lequel les paquets de données comprennent des paquets de données audio.

7. Procédé selon l'une quelconque des revendications précédentes, dans lequel les paquets de données comprennent des paquets de données audio et vidéo.

LINKPLAY EXHIBIT 1004
Page 1879 of 4822

8. Procédé selon l'une quelconque des revendications précédentes, dans lequel l'intervalle de données correspond à une trame de données vidéo.

9. Procédé selon l'une quelconque des revendications 1 à 7, dans lequel l'intervalle de données correspond à un champ de données vidéo.

10. Procédé selon l'une quelconque des revendications précédentes, dans lequel le bus est un bus en série.

11. Procédé selon la revendication 10, dans lequel le bus en série est un bus FireWire conforme.

12. Procédé selon l'une quelconque des revendications précédentes, dans lequel le paquet de synchronisation est un paquet asynchrone.

13. Procédé selon l'une quelconque des revendications précédentes, dans lequel la pluralité de paquets sont des paquets isochrones.

14. Procédé selon l'une quelconque des revendications précédentes, dans lequel le signal de référence périodique est reçu par le second dispositif.

15. Procédé selon l'une quelconque des revendications 1 à 13, dans lequel le signal de référence périodique est généré par le second dispositif.

16. Procédé selon l'une quelconque des revendications précédentes, comprenant en outre :

la réception d'une commande à partir du premier dispositif pour commencer le transfert des données ; et
la réception de zéro ou plusieurs paquets nuls à partir du premier dispositif sur le bus jusqu'à ce qu'un premier paquet de synchronisation soit envoyé par le second dispositif pour un premier intervalle de données.

17. Procédé selon l'une quelconque des revendications précédentes, comprenant en outre :

la réception des données dans une mémoire tampon au niveau du second dispositif ; et
la sortie des données à partir du second dispositif en synchronisation avec le signal de référence périodique.

18. Procédé selon la revendication 17, dans lequel la sortie des données comprend le codage et la sortie d'un signal vidéo analogue.

19. Procédé selon la revendication 17, dans lequel la sortie comprend la génération de données vidéo dans un format DV.

20. Dispositif (102) pour la connexion à une source de données temporelles sur un bus, comprenant :

une interface de bus (104) servant à envoyer des paquets vers la source de données (100) et à recevoir des paquets à partir de celle-ci sur le bus ;
un contrôleur ayant une entrée pour recevoir un signal lié à un signal de référence périodique (110) définissant une pluralité d'intervalles de données, et ayant une sortie pour fournir l'instruction à l'interface de bus d'envoyer un paquet de synchronisation sur le bus vers la source de données pour chaque intervalle de données en réponse au signal de référence périodique ; et
une mémoire tampon connectée pour recevoir des données pour l'intervalle de données à partir d'une pluralité de paquets reçus à travers l'interface de bus sur le bus à partir de la source de données après réception par la source de données du paquet de synchronisation.

21. Système servant à transférer des données temporelles comprenant :

un ordinateur hôte (100) ;
un dispositif périphérique (102) connecté à l'ordinateur hôte via un bus (104) ;

dans lequel le dispositif périphérique comprend le moyen d'envoyer un paquet de synchronisation sur le bus vers l'ordinateur hôte pour chaque intervalle de données selon un signal de référence périodique (110) ;
dans lequel l'ordinateur hôte comprend le moyen d'envoyer une pluralité de paquets sur le bus vers le dispositif

LINKPLAY EXHIBIT 1004
Page 1880 of 4822

périphérique après réception du paquet de synchronisation, dans lequel les paquets comprennent des données pour l'intervalle de données, et le moyen, opérationnel après l'envoi de la pluralité de paquets pour l'intervalle de données, d'attendre jusqu'à ce qu'un paquet de synchronisation soit reçu pour un intervalle de données ultérieur.

LINKPLAY EXHIBIT 1004
Page 1881 of 4822

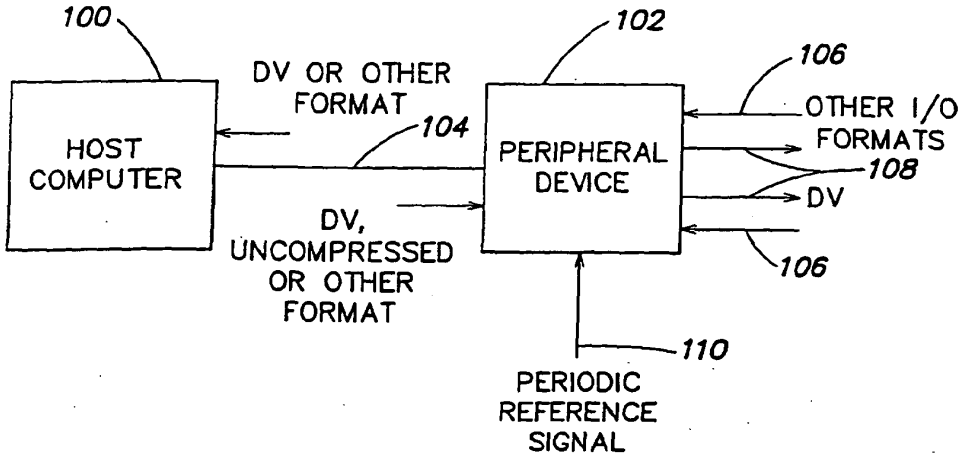


FIG. 1

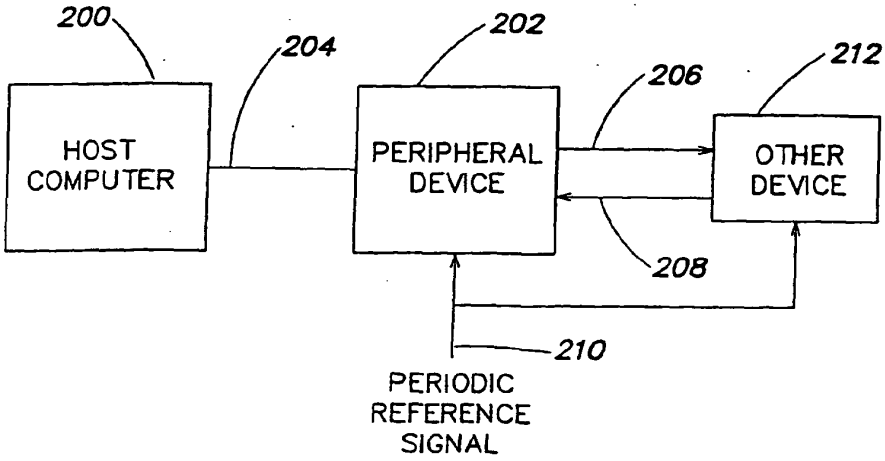


FIG. 2

LINKPLAY EXHIBIT 1004
Page 1882 of 4822

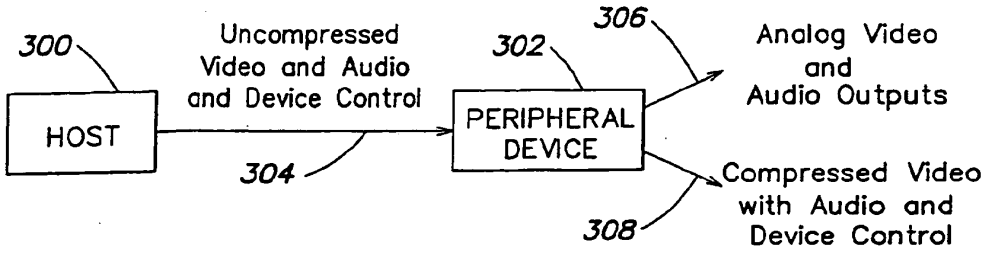


FIG. 3

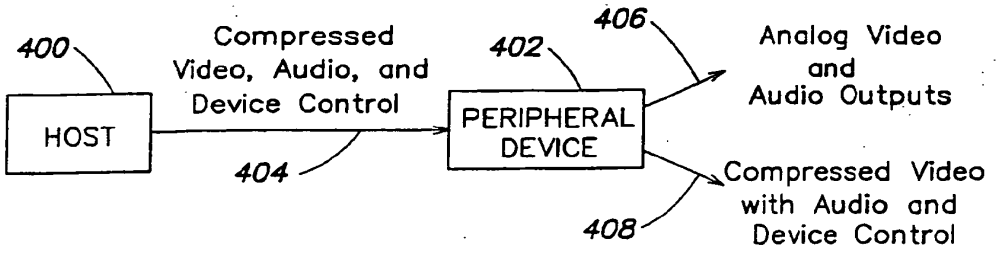


FIG. 4

LINKPLAY EXHIBIT 1004
Page 1883 of 4822

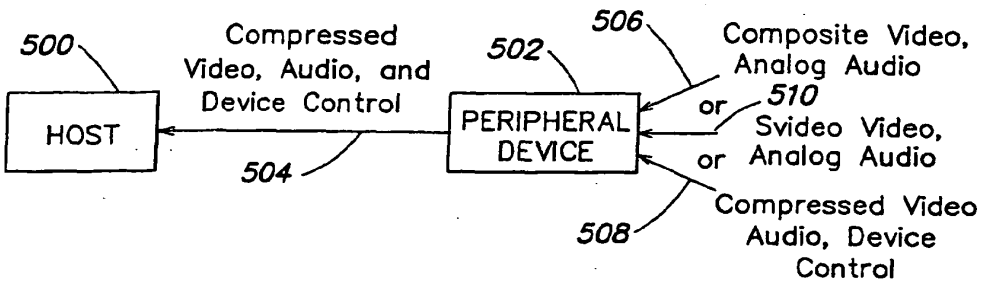

500
HOST
Compressed
Video, Audio, and
Device Control
504
502
PERIPHERAL
DEVICE
506
Composite Video,
Analog Audio
or
510
Svideo Video,
or Analog Audio
508
Compressed Video
Audio, Device
Control

FIG. 5

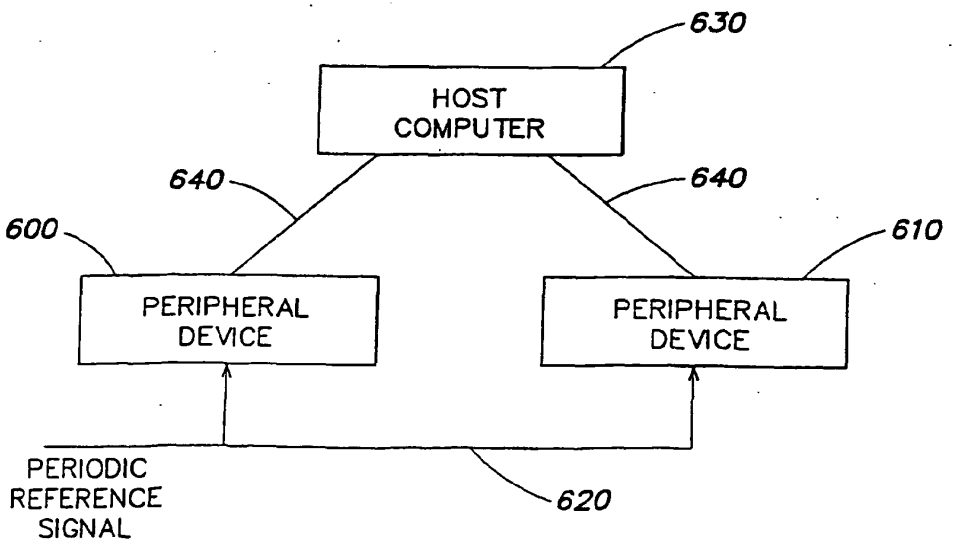

630
HOST
COMPUTER
640
640
600
610
PERIPHERAL
DEVICE
PERIPHERAL
DEVICE
PERIODIC
REFERENCE
SIGNAL
620

FIG. 6

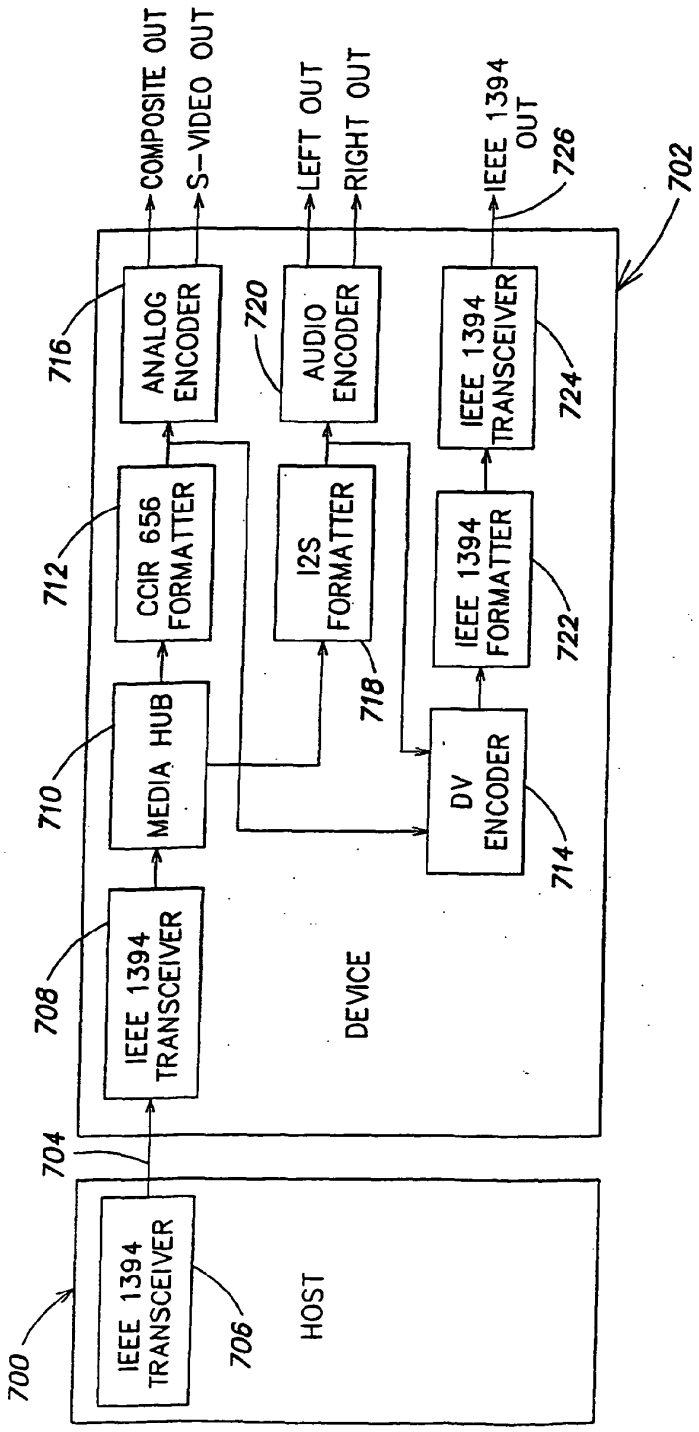


FIG. 7

LINKPLAY EXHIBIT 1004
Page 1885 of 4822

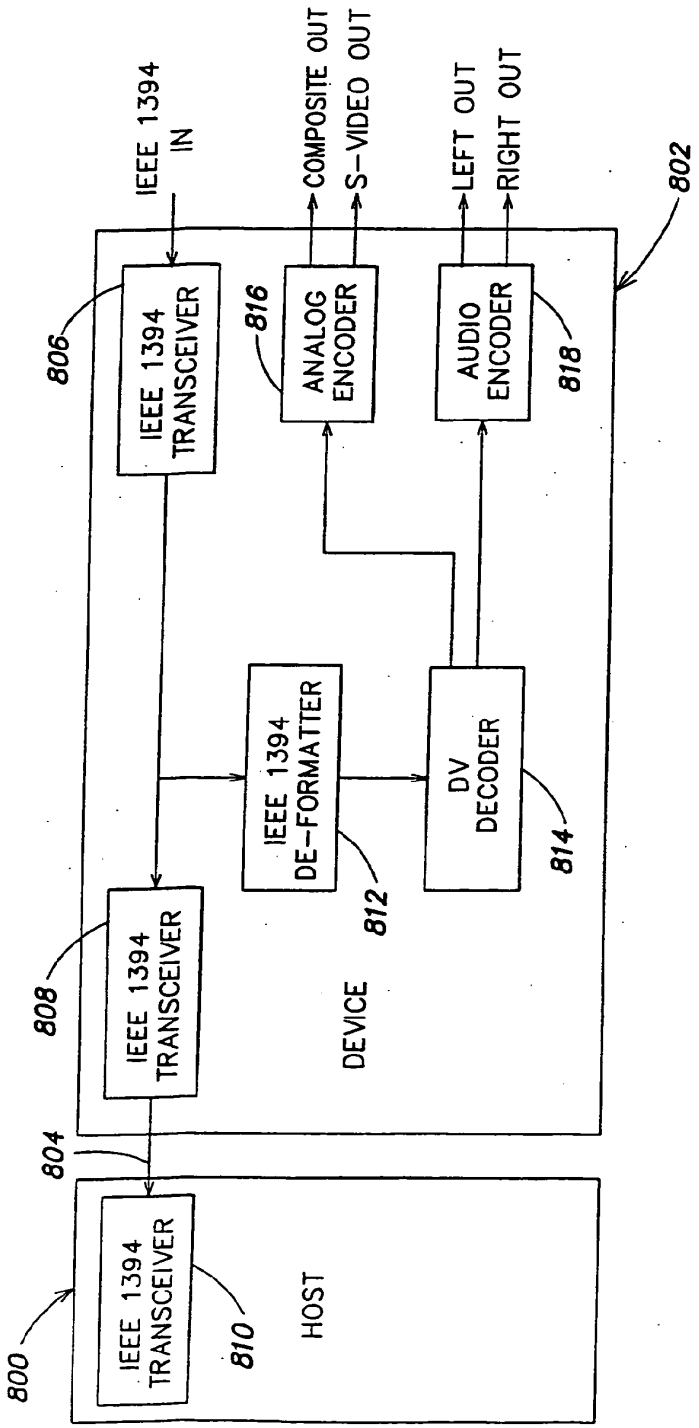

IEEE 1394
IN
COMPOSITE OUT
S-VIDEO OUT
LEFT OUT
RIGHT OUT
802
806
IEEE 1394
TRANSCEIVER
816
ANALOG
ENCODER
AUDIO
ENCODER
818
IEEE 1394
DE-FORMATTER
812
DV
DECODER
814
808
IEEE 1394
TRANSCEIVER
DEVICE
804
800
IEEE 1394
TRANSCEIVER
810
HOST

FIG. 8

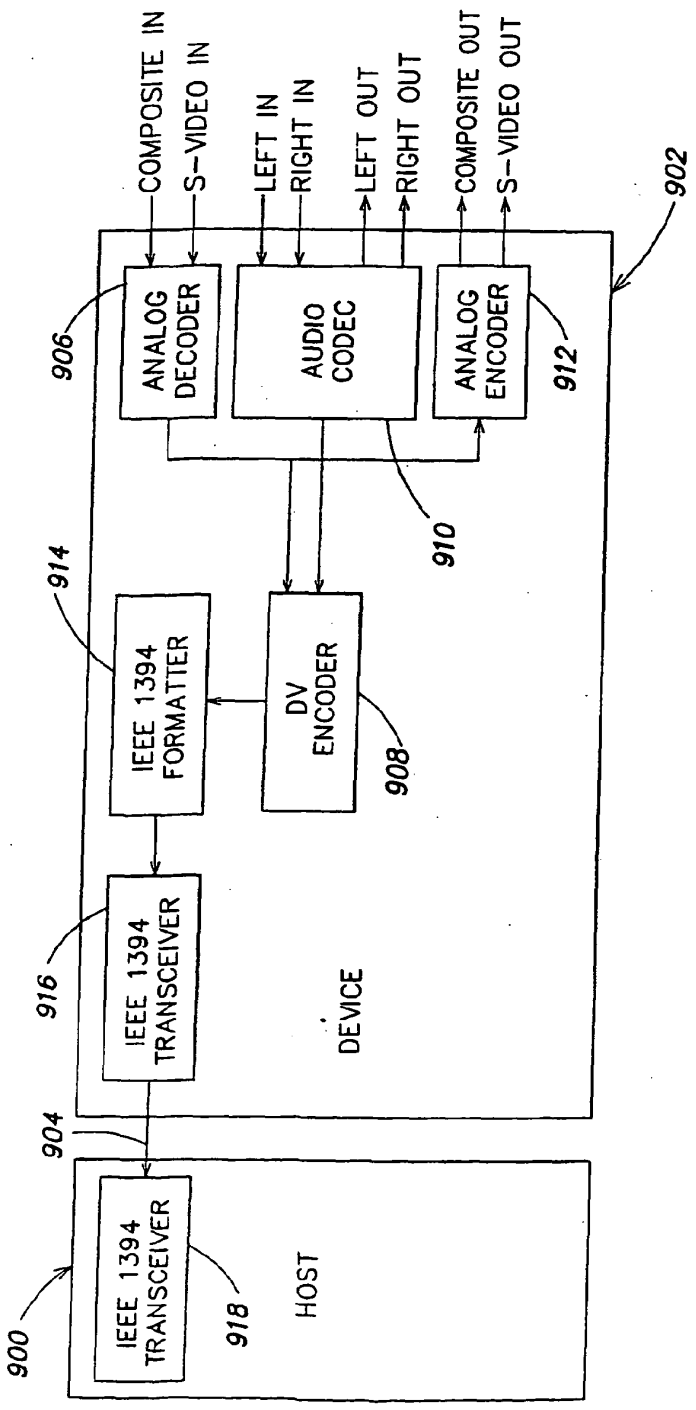

COMPOSITE IN
S-VIDEO IN
LEFT IN
RIGHT IN
LEFT OUT
RIGHT OUT
COMPOSITE OUT
S-VIDEO OUT
902
906
ANALOG DECODER
AUDIO CODEC
ANALOG ENCODER
912
910
914
IEEE 1394 FORMATTER
DV ENCODER
908
916
IEEE 1394 TRANSCEIVER
DEVICE
904
900
IEEE 1394 TRANSCEIVER
918
HOST

FIG. 9

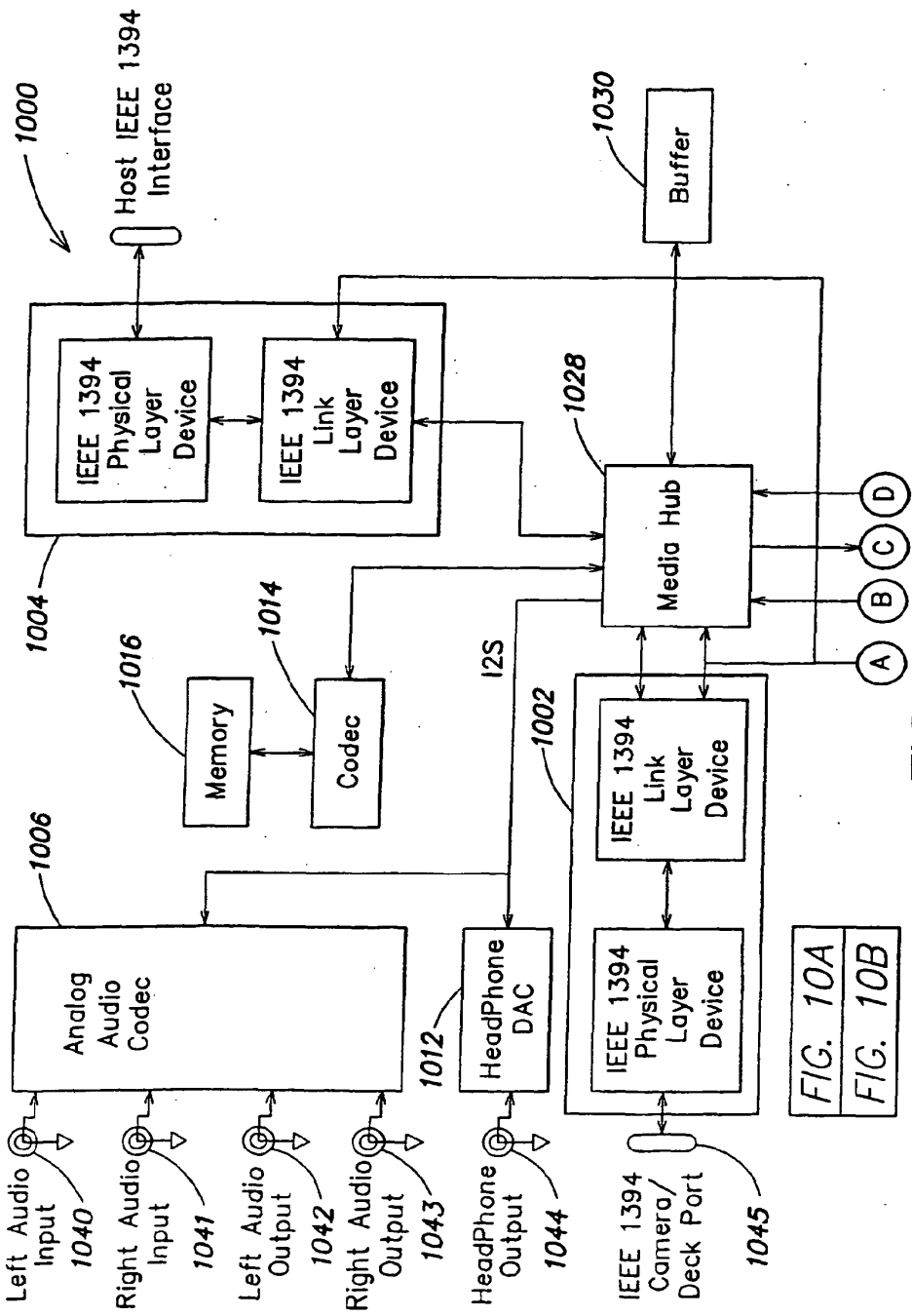

1000
Host IEEE 1394 Interface
1030
Buffer
IEEE 1394 Physical Layer Device
IEEE 1394 Link Layer Device
1028
1004
Media Hub
A
B
C
D
1016
Memory
1014
Codec
I2S
1002
IEEE 1394 Link Layer Device
1006
Analog Audio Codec
1012
HeadPhone DAC
IEEE 1394 Physical Layer Device
FIG. 10A
FIG. 10B
Left Audio Input 1040
Right Audio Input 1041
Left Audio Output 1042
Right Audio Output 1043
HeadPhone Output 1044
IEEE 1394 Camera/ Deck Port 1045

FIG. 10A

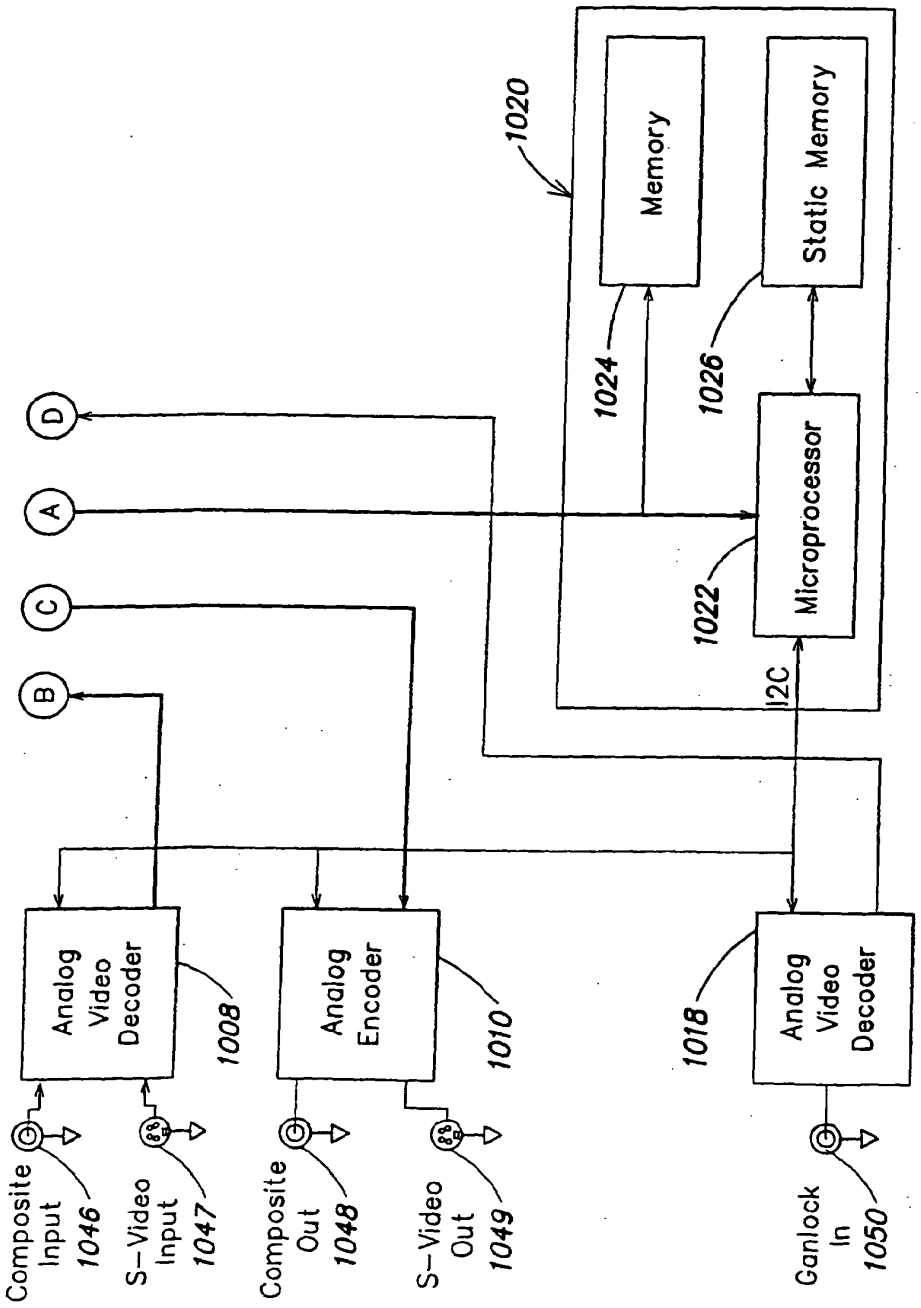


FIG. 10B

LINKPLAY EXHIBIT 1004
Page 1889 of 4822

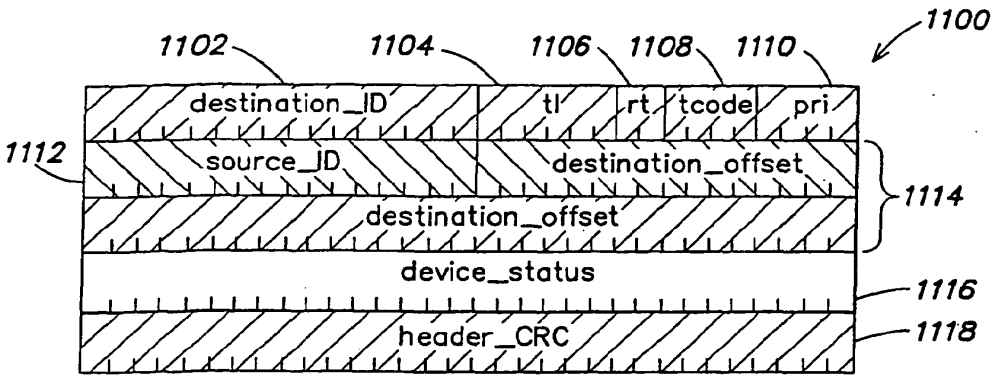


FIG. 11

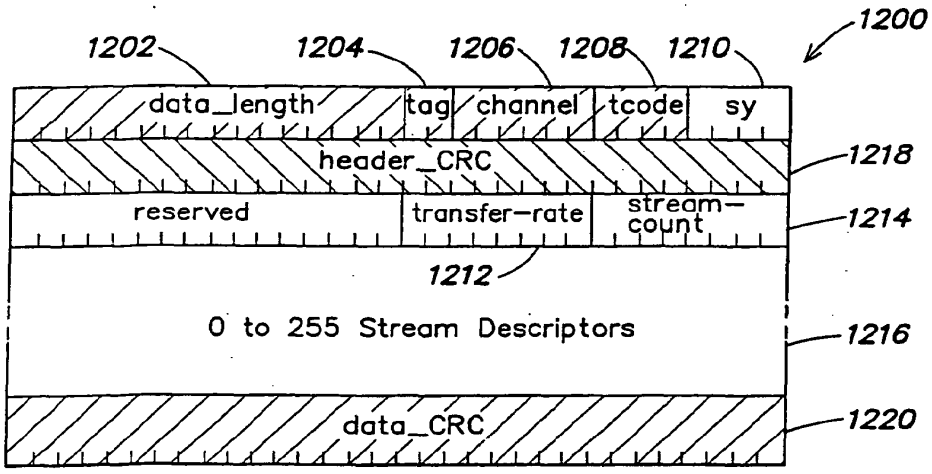


FIG. 12

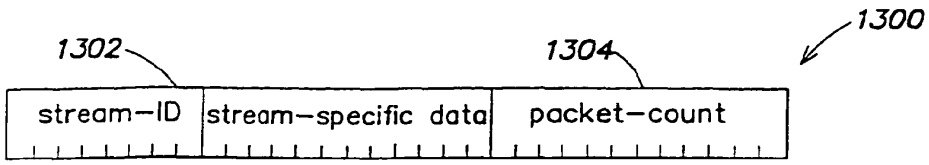


FIG. 13

LINKPLAY EXHIBIT 1004
Page 1890 of 4822

1400

1404 NAME	1402 VALUE	1406 DESCRIPTION
VFIELD1	1	field 1 video data
VFIELD2	2	field 2 video data
VFRAME	3	full frame video data
SYSTEM	32	system data
AUXILIARY	33	auxiliary data
A1	64	A1 audio data
A2	65	A2 audio data
A3	66	A3 audio data
A4	67	A4 audio data
A5	68	A5 audio data
A6	69	A6 audio data
A7	70	A7 audio data
A8	71	A8 audio data

FIG. 14

LINKPLAY EXHIBIT 1004
Page 1891 of 4822

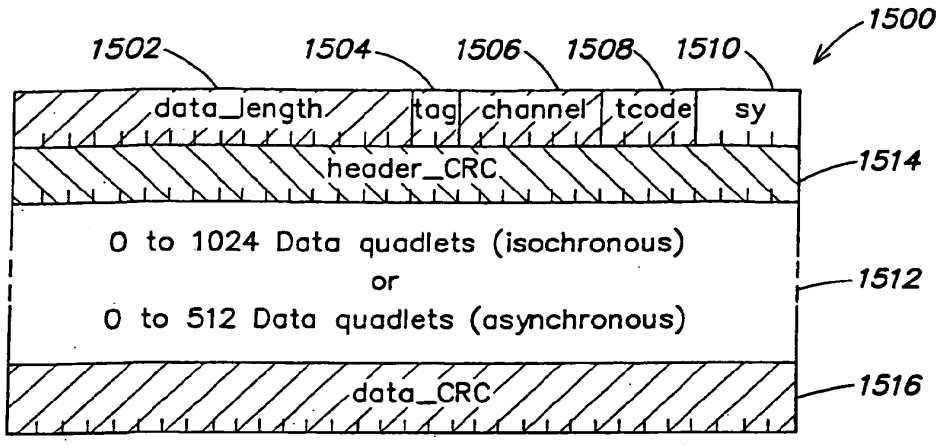


FIG. 15

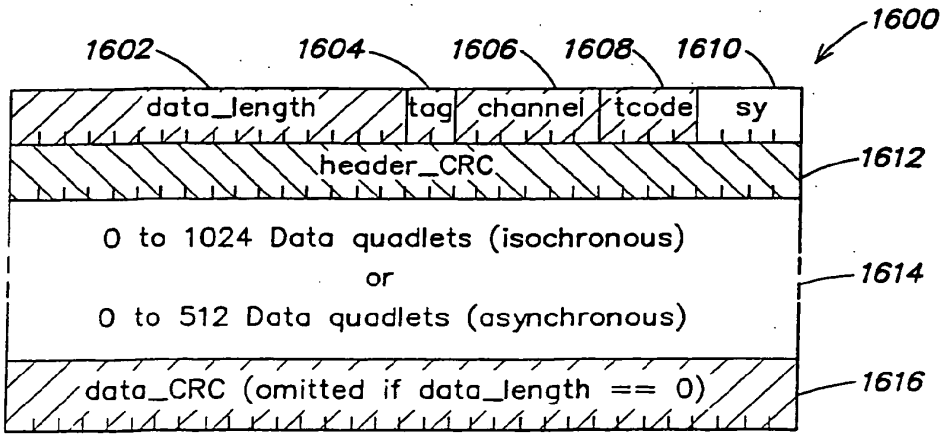


FIG. 16

LINKPLAY EXHIBIT 1004
Page 1892 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 517 464 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
23.03.2005 Bulletin 2005/12

(51) Int Cl.[7]: **H04H 1/00**, H04L 12/28

(21) Application number: **04020609.6**

(22) Date of filing: **31.08.2004**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PL PT RO SE SI SK TR
Designated Extension States:
AL HR LT LV MK

(30) Priority: **03.09.2003 US 500055 P**
12.12.2003 US 734706
08.01.2004 US 754912

(71) Applicant: **Phoenix Gold International, Inc.**
9300 North Decatur Portland, OR 97203 (US)

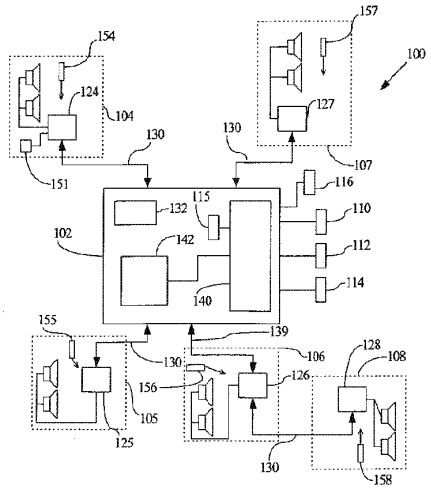
(72) Inventors:
- **Devin, Craig, c/o Devin Design & Development**
90277 Redondo Beach, CA (US)
- **White, Dorian**
90025 Los Angeles, CA (US)
- **Williams, Steve Phoenix Gold International, Inc.**
97203 Portland, OR (US)
- **MacBeth, Roland c/o Phoenix Gold Intern.,Inc.**
97203 Portland, OR (US)

(74) Representative: **Lang, Johannes, Dipl.-Ing.**
Bardehle Pagenberg Dost Altenburg Geissler,
Postfach 86 06 20
81633 München (DE)

(54) **Digital audio distribution system**

(57) A digital audio distribution network includes a hub configured to receive analog or other audio signals from one or more audio sources. The hub includes analog-to-digital converters that produce digital audio signals and a remote control input that receives control data from a remote control device. An audio processor combines the digital audio signals and the control data for delivery to one or more audio zones as a digital audio distribution signal. The audio processor also receives a digital audio distribution signal and produces an analog audio signal for delivery to one or more speakers. The audio processor also decodes control data from the input digital audio distribution signal and directs the decoded control data to an infrared repeater. The hub also delivers electrical power to additional network hubs or other network elements using a cable that also conducts one or more digital audio distribution signals. A remote control processor is configured to learn additional remote control commands, and a serial interface is provided for network configuration and troubleshooting.

FIG. 1



EP 1 517 464 A2

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD

[0001] The disclosure pertains to digital audio distribution systems.

BACKGROUND

[0002] Residential audio systems typically provide audio content from a variety of sources. A typical system offers AM/FM radio, analog and digital audio tape, compact discs, and a variety of digital file formats. Thus, audio system users have available a broad spectrum of audio materials. Audio systems adapted to this variety of audio sources can require corresponding components such as AM/FM tuners, CD players, satellite television or set top box audio feeds, DVD audio, SuperCD format, computer streamed audio from the internet, etc, and can be large and expensive. In addition, system users frequently customize audio installations for a particular setting to, for example, accommodate room acoustics or permit convenient selection of favorite audio sources.

[0003] While sophisticated, versatile audio components are available, system users still encounter significant difficulties in obtaining the intended audio programming. Audio components are typically situated in and configured for a particular location, and require the user to remain in this location to enjoy the intended audio programming. In addition, sophisticated audio components can represent a considerable expense, and users are typically unwilling to configure additional audio components for other locations. Thus, while an audiophile may have a state-of-the-art audio system in a designated area in her residence, she typically must still resort to listening to a portable audio system or lower fidelity component such as a clock radio or intercom equipped with CD player or FM tuner boom box while in other areas. Therefore, methods and systems for audio distribution are needed.

SUMMARY

[0004] Digital audio networks comprise a hub configured to receive a plurality of stereo signals and combine the stereo signals with control data to produce a digital audio network signal. An output connector is configured to communicate the digital audio network signal from the hub to additional network elements downstream or upstream along an output cable. In representative examples, the output connector is also configured to deliver electrical power to the output cable in accordance with, for example, standard IEEE 802.3AF usage guidelines. In additional examples, the output connector is configured to deliver the digital audio network signal to a first pair of conductors in the output cable and electrical power to at least a second pair of conductors in the output cable. In representative examples, the output connector is configured to deliver the digital audio network signal to a first pair of conductors in a CAT 5 cable, and to deliver electrical power to at least a second pair of conductors in the CAT 5 cable. In additional examples, the hub includes an input connector configured to receive a digital audio network signal. In additional examples, an infra-red control input is configured to receive the control data. In further examples, a digital signal processor is configured to process stereo audio data associated with at least one of the stereo signals.

[0005] Controllers for a digital audio network comprise a keypad configured for entry of at least one audio source identifier or configuration parameter for at least one audio playback output or a plurality of output locations grouped as a zone. An audio processor is configured to receive an input digital audio distribution signal that includes digital audio data and control data, and produce an output digital audio distribution signal based on the entry of the at least one audio source identifier or configuration parameter. In specific examples, a display is configured to display the at least one audio source identifier or configuration parameter. In other examples, an infrared sensor is configured to receive control data from a modulated infrared emitter and the audio processor is configured to include the control data from the modulated infrared emitter in the output digital audio distribution signal.

[0006] Audio network devices comprise an infrared sensor configured to receive control data from a modulated infrared emitter, and an audio processor configured to produce a digital audio distribution signal based on the received control data. In representative examples, an input is configured to receive at least one audio input signal, wherein the audio processor is configured to produce a digital audio signal based on the input audio signal, and the digital audio distribution signal is based on the digital audio signal. In additional examples, a user input is configured to select at least one audio signal to be decoded from the digital audio distribution signal. In further examples, a tone generator is configured to produce an audio tone, wherein the audio processor is configured to include the audio tone in the digital audio distribution signal. In additional examples, an input audio transducer is in communication with the audio processor and configured to generate an audio signal. The audio processor is configured to include the audio signal generated by the input audio transducer in the digital audio distribution signal.

[0007] Audio distribution methods comprise receiving at least one audio signal from an audio signal source and digitizing the at least one audio signal to produce a digital audio signal. The digital audio signal is combined with control data associated with the audio signal source to produce a digital audio distribution signal. In additional examples, the control data is received from a remote control device. In other examples, the digital audio distribution signal is transmitted to an audio zone. An ana-

D+M-PA_0000697

LINKPLAY EXHIBIT 1004
Page 1894 of 4822

log audio signal is extracted from the digital audio distribution signal, and the extracted analog signal is delivered to an audio output device. In other examples, the extracted analog signal is configured based on the control data.

[0008] Audio distribution methods comprise receiving a plurality of digital audio signals and receiving control data. A digital audio distribution signal is produced based on a combination of selected digital audio signals and the control data. In other examples, the digital audio distribution signal is transmitted and the control data is extracted from the digital audio distribution signal and delivered to at least one audio source. In additional examples, the control data is associated with at least one of source volume, loudness, or muting. This control data is separated and routed to an on-board microprocessor for interpretation, routing and digital signal processing control functions. In other examples, the control data is associated with identification of a particular digital audio signal for delivery to a selected zone. In other representative examples, the method includes delivering electrical power with the transmitted digital audio distribution signal.

[0009] In additional examples, systems are configured to recognize previously undefined control commands received from, for example, an IR remote control, notify a system user of such receipt, and initiate a programming or "learning" sequence to associate a selected function with the undefined command, thereby defining the command. A remote control processor can be included that is in communication with a library of commands, or that can process new commands that are not in the command library. In other examples, network elements are configured to include program material in a digital audio distribution signal, and to recognize control commands to be executed remotely for inclusion in the digital audio distribution signal. In other examples, a communication interface is provided for communication with an external device such as a PDA or a personal computer or a touch panel controller for network installation, maintenance, and updates.

[0010] These and other features and advantages are described below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011]

FIG. 1 is schematic diagram of a digital audio distribution network.

FIG. 2A is schematic diagram of connection panel for a digital audio distribution network hub.

FIG. 2B is a schematic block diagram of an eight stereo channel digital audio distribution network hub.

FIG. 3 is schematic diagram of a portion of a network hub illustrating analog-to-digital conversion of two stereo audio signals.

FIGS. 4A(1)-4A(3) contain a schematic diagram of a portion of a network hub configured to receive digitized stereo audio signals and produce a digital audio network distribution signal that includes digital audio data for a plurality of stereo sources and control data associated with control of the stereo sources.

FIG. 4B is a schematic diagram of a portion of a network hub that includes status inputs for a plurality of audio sources and output connections for IR emitters associated with control of audio sources and network elements.

FIG. 5 is a schematic diagram of a portion of a network hub configured to produce analog audio signals based on digital audio signals received from a digital signal processor.

FIGS. 6A-6C contain a schematic diagram of a keyboard module for a digital audio distribution network.

FIGS. 7A-7B are schematic diagrams of front panels of representative user control modules.

FIG. 8 is a schematic diagram of a front panel of a remote power module.

FIG. 9 is a schematic diagram of a front panel of an input/output module.

FIGS. 10A(1)-10A(3) is a schematic diagram of user control module that includes a processor configured to learn remote control commands and a serial interface.

FIG. 10B is a block diagram illustrating processing of command received from a remote control.

FIG. 11 is a schematic diagram of a digital audio distribution network arranged in a point-to-point configuration.

FIG. 12 is a schematic diagram of a digital audio distribution network arranged in a star configuration.

FIG. 13 is a schematic diagram of another representative digital audio distribution network.

FIGS. 14A-14C contain a schematic diagram of a hub that includes a serial interface and a processor configured to learn control commands.

FIGS. 15A-15B contain a schematic diagram of a keypad module configured to communicate with a digital signal processor.

FIG. 16 is a schematic diagram of an IR output module configured to control IR emitters based on a processed IR signal.

DETAILED DESCRIPTION

[0012] With reference to FIG. 1, a representative audio distribution network 100 includes a hub 102 configured to communicate with representative zones 104-108. The hub 102 is configured to receive audio signals and other signals from external sources 110, 112, 114 such as, for example, CD players, DVD players, tun-

D+M-PA_0000698

LINKPLAY EXHIBIT 1004
Page 1895 of 4822

ers, satellite decoders, intercom systems, alarm systems, intercoms microphones, etc. The hub 102 can be configured to accept, for example, sixteen audio data channels (eight stereo sources) as inputs, and can transmit eight audio data channels (four stereo sources) to the zones 104-108, but fewer or more channels can be provided. The zones 104-108 are typically associated with areas in which audio signals are to be received. In a residential installation, the zones can correspond to various rooms in a residence (media room, dining room, family room) or locations (deck, pool, workshop, back yard). One or more audio sources such as a representative source 115 can be included in the hub 102, and digital audio sources such as a representative digital source 116 can be used. In addition, sources can be situated in one or more zones such as a source 151 situated in the zone 104.

[0013] As shown in FIG. 1, the zones 104-108 are provided with digital audio distribution (DAD) processors 124-128, respectively, that are interconnected with cables 130, typically so-called Category 5 ("Cat 5") cable. Cat 5 cables include eight conductors arranged in four pairs of conductors. As described below, the DAD processors of the zones can be different, and DAD processors can be connected via another DAD processor to the hub 102 or to other hubs. The DAD processors are typically configured to receive a digital audio distribution signal and decode/encode digital audio signals and control data as described below.

[0014] The hub 102 also receives and routes infrared remote (IR) or other control information from remote controllers 154-158 in the zones 104-108, respectively, to and from system components such as the DAD processors 124-128 and the sources 110, 112, 114, 116, 151. The IR information is typically transmitted as control data packets that are combined with digital audio signals for full duplex or bi-directional delivery over the cables 130. For example, a DAD processor in a selected zone can be configured to receive a command from an infrared or other wireless remote controller, and deliver the command to the hub 102. In this manner, users can control remote sources as well as control other DAD network elements.

[0015] The hub 102 also includes a power supply 132 configured to provide power to the digital audio distribution system via the cables 130. In a representative example, the power supply is configured to operate with input voltages of between about 90 VAC to 260 VAC and produce an output of about 24 VDC at 2.5 Amps.

[0016] A digital audio processor 140 is configured to receive analog audio signals from the sources 110, 112, 114 and produce corresponding digital signals. In a representative example, the processor 140 is based on a so-called ADAT chipset available from ALESIS SEMICONDUCTOR and is configured to convert the eight channels of analog input (four stereo pairs) to a single digital signal, typically a 24-bit, 48 kHz digital signal. Operation of this chipset is described in U.S. Patent 5,297,181. The processor 140 also receives IR information and other control information, and combines the control information with the digital audio signal. The combined audio/control signal is transmitted over a pair of conductors, and return (or duplex) information is received on a second pair of conductors. Power (e.g., 24 VDC) and ground are carried on the third and fourth pair of conductors, respectively. The hub 102 is also configured to sense input signal voltages that can be used to verify the status of a source component as ON or OFF based on, for example, detecting that a signal is present for greater than 2 seconds.

[0017] The digital audio processor 140 is configured to implement digital signal processing to provide user-selectable and customizable audio processing. The hub 102 and the digital audio processor 140 can be configured to provide user-selectable audio or control functions such as Auto Level in which audio signal amplitudes of signals from different sources are adjusted via intelligent compression. Auto Loudness can also be selected in which magnitudes of audio signal portions in selected frequency bands, typically low and high frequency bands, are adjusted based on an audio volume level. Typically, low and high frequency signal components are increased at low audio volume levels. Multiband parametric equalization can also be provided in which, for example, parameters for five frequency bands can be selected for parametric equalization for a selected zone or zones. The frequency bands can be user-selectable, and adjustable boost/cut and "Q" can be provided at selected frequencies or frequency bands. Muting during power up/power down, and temporary muting for one or more selected zones can also be provided.

[0018] The hub 102 can be configured to have a default configuration such as, for example, a power-on configuration. In addition, the hub 102 can include a display 142 and an input interface for controlling and configuring audio delivery for one or more of the zones. For example, upon initialization, audio inputs 1-4 can be identified in the display 142 as INPUT1-INPUT4, respectively. During a configuration process, the inputs can be assigned other names such as "CD Player," "Tuner," "Tape," or other names. These assignments can be stored for display in subsequent system initializations. These assignments can also be displayed by DAD network elements situated in other zones by combining name assignment data with audio data in the digital audio distribution signal.

[0019] The hub 102 can also include a tone generator and can be configured to receive an alarm signal or to be in communication with a doorbell switch. In response to the alarm signal or the doorbell switch, a tone generated by the tone generator can be delivered to one or more zones and to override previously selected programming. In addition, intercom components can be included in the network 100 for operation under control of the hub 102.

[0020] With reference to FIG. 2A, connections to a

D+M-PA_0000699

LINKPLAY EXHIBIT 1004
Page 1896 of 4822

hub such as the hub 102 can be provided with a connection panel 200 that includes source modules 201, 207, 213, 219 that are configured to receive and output stereo audio signals as left/right (L/R) pairs. The source module 201 includes a left input connector 202, a right input connector 203, and associated L/R output connectors 204, 205, respectively. Various connector types can be used, but in one example, so-called RCA connectors are used. The source modules 207, 213, 219 can be similarly configured. The source modules 201, 207, 213, 219 are associated with respective trigger inputs 206, 212, 218, 224 that are configured to receive "trigger signals" that indicate that the associated source is turned on. In addition, the source module inputs are sampled by a digital audio processor to determine if the associated audio input pair is active (i.e., is receiving audio), and a STATUS LED 272 is used to indicate if any source module inputs are active. A determination of activity can be based on, for example, presence of an audio signal for a predetermined interval, such as 2 seconds. The panel 200 includes a coaxial connector 232 or other connector that can be configured to receive electrical power from an external power supply.

[0021] A remote source module 240 includes left and right output connectors 242, 244 and can be associated with a selected source module to provide, for example, subwoofer output or speaker output, or to receive a mono output associated with a stereo input. An RJ45 connector 250 is configured to communicate audio data and control data in a digital audio distribution signal and electrical power to and from additional network components located in one or more zones. An additional RJ45 connector 252 is provided for expansion or connection to a digital bus. A universal serial bus (USB) port 256 can also be provided for software revisions or for network control. A 4-conductor connector 258 is provided for a remote IR receiver for zones for which a system controller is unavailable. A master/slave switch 262 can be configured to select a selected hub as an originating source for a system clock. A green LED 268 is configured to indicate that power is available to the hub, a red LED 270 is configured to indicate data transmission on all RJ45 connections, and the blue LED 272 is configured to indicate that a voltage is present at a Status Input, typically 5-15 VDC. Connectors 281-284 are configured to receive infrared repeaters for transmission of IR-controller based commands. The connectors are typically 3.5 mm phone jacks, but other types of connector can be used. These IR repeaters are generally wired in parallel.

[0022] With reference to FIG. 2B, a hub configured for eight stereo sources (sixteen mono channels) typically includes A-D converters 285-288 that are configured to receive two analog stereo inputs and produce corresponding stereo digital signals that are delivered to a controller 289. The stereo digital signals typically include both left and right channel audio in a single digital signal. The controller 289 is configured to receive network control commands or commands associated with audio or other sources via an infrared sensor unit 290 that can be configured to receive commands from a conventional IR remote controller. The controller 289 combines the stereo digital signals with control data and delivers to produce a digital audio network signal that is delivered to an output 291. The controller 289 is also configured to receive a digital audio network signal from an input 292 and to produce processed audio signals and a word clock signal that are delivered to a processed audio output 293. Operation of the hub is indicated with an LED display panel 294 that indicates audio channel status and the presence of audio data, as well as producing an IR signal based on a processed IR signal produced by the controller 289. Analog stereo output signals are produced by D-A converters 294-297 and can be based on audio signals provided to the A-D converters 285-289 or on audio signals associated with a digital audio network signal received at the input 292, or both.

[0023] Referring to FIG. 3, a portion of a network hub is configured to receive audio signals from two stereo sources at a left source 1 (L1) input 302, a right source 1 (R1) input 304, a left source 2 (L2) input 306, and a right source 2 (R2) input 308. Alternatively, monophonic audio signals can be provided to one or more of these source inputs. The source inputs are connected to signal conditioners 303, 305, 307, 309 that process the audio input signals for delivery to respective direct outputs 310-313 as processed analog signals. In addition, processed audio signals from the L1 input 302 and the R1 input 304 are delivered as differential signals to an analog-to-digital converter 320, and processed audio signals from the L2 input 306 and the R2 input 308 are delivered as differential signals to an analog-to-digital converter 322. Corresponding digital signals are delivered to A-D outputs 326, 328, respectively. A word clock (WDCLK) input 330 is configured to receive a clock signal for synchronization of the digital audio output signals produced by the A-D converters 320, 322.

[0024] With reference to FIGS. 4A(1)-4A(3), a representative network hub includes input/output connectors 402, 404 configured to receive and transmit digital audio distribution network signals and deliver electrical power to other remote or local network elements. The I/O connectors 402, 404 are typically RJ45 connectors that are configured so that pin pairs 1, 2 are configured to output digital audio distribution network signals that include audio data for up to four stereo pairs and control data, pins 3, 6 are configured to receive a digital audio distribution network signal that includes digital audio data for up to four stereo pairs and control data, and pin pairs 4, 5 and 7, 8 are configured for DC power and ground, respectively. Digital audio distribution network signals are transmitted to and from the I/O connectors 402, 404 via respective signal conditioners 406, 408 and are delivered to respective digital audio decoders 410, 412 that are configured to produce up to four digital outputs corresponding to each of the stereo pairs presented to the

D+M-PA_0000700

LINKPLAY EXHIBIT 1004
Page 1897 of 4822

I/O connectors 402, 404. The decoded digital audio signals are delivered to a complex programmable logic device (CPLD) 414 such as, for example, a XILINX XC9572 or other logic device. The digital audio decoder 410 is also configured to produce a clock signal (WDCLK) that is delivered to the digital audio decoder 412 and to a WDCLK output 411, but in other examples, the decoder 412 can be configured to produce a clock signal or a clock signal can be provided from another network component.

[0025] Digital stereo inputs 420 are configured to receive digital stereo signals produced by, for example, A/D converters as shown in FIG. 3. Such digital stereo signals are delivered to the CPLD 414 along with status signals for up to eight stereo channels received from one or more status signal inputs 422. Rotary encoders 424, 426 are configured to provide control signals to encoder inputs of the CPLD 414. The CPLD 414 is configured to deliver selected digital stereo signals to a digital signal processor (DSP) 428 that delivers processed digital audio signals to processed digital outputs (PDOUT) 430, 432. Referring to FIG. 5, the processed digital outputs are in communication with respective D-A converters 502, 504 that produce associated left channel and right channel analog audio signals. The left channel audio signals are delivered to signal conditioners 510, 511 and output connectors 520, 521, and the right channel audio signals are delivered to signal conditioners 512, 513 and output connectors 522, 523.

[0026] Referring again to FIG. 4, a microprocessor 450 is configured to receive control data extracted from an input digital audio distribution network signal by the audio decoders 410, 412. This control data is generally associated with IR remote control signals and is processed for delivery to a processed IR output 452. The audio decoders 410, 412 also communicate with a Data Present output 453 to indicate that a digital audio distribution network signal is available for decoding. The microcontroller 450 is configured to selectively provide a mute signal to the decoders 410, 412 based on control data extracted from the digital audio distribution network signal or otherwise provided. The microprocessor is also configured to communicate with and control signal processing by the DSP 428. An EEPROM 454 is in communication with the microprocessor 450, and a power conditioner 456 is also provided. An encoder 417 is configured to combine digital audio signals received from the CPLD 414 with control data for delivery as a digital audio distribution signal to pins 1, 2 of connector 404.

[0027] With reference to FIG. 4B, set 470 on input connectors is configured to receive signals associated with a power on/power off condition of up to eight audio sources. A status output (STATUS[1..8]) 471 is configured to deliver associated status signals to, for example, the CPLD 414. Based on the power on/off signals and a data present signal delivered to a DATA PRESENT input 472, LEDS 474-476 are turned on. A processed IR input receives a processed IR signal for delivery to a set 480 of connectors for delivery to IR repeaters that can be situated to deliver IR signals to, for example, IR remote control inputs of one or more audio sources.

[0028] Referring to FIGS. 6A-6C, a network keypad 600 includes audio signal conditioners 602, 604 and respective audio decoders 610, 612 that are in communication with respective input/output connectors 601, 603. Decoded digital audio (typically as a combined left/right digital signal) is delivered to a CPLD 614 and control data is delivered to a microprocessor 616. An audio encoder 618 is configured to deliver a digital audio distribution signal to the signal conditioners 602, 604 and to the connectors 601, 603. Audio data can be selected for inclusion the output signal using encoders 620, 622. Keypad switches 631, 632, 633, 634 are configured for control of audio volume, selection of a particular zone, muting, and source selection, respectively. An encoder 636 is configured for selection of zones or audio sources, or other audio network features. The microprocessor 616 is configured to deliver a processed IR control signal to an LED array 640 that is used to, for example, communicate infrared remote control signals to one or more audio sources. These control signals are typically based on conventional infrared remote control signal protocols. The microprocessor is also in communication with an IR sensor 642 that is configured to receive IR control signals from an IR remote controller and transmit associated electrical control signals. A connector 644 can be provided for communication with a DSP module.

[0029] With reference to FIG. 7, a keypad-based controller 700 includes an LCD display, typically a 128 by 64 back or side illuminated LCD. The controller 700 typically includes two IDC, RJ45, or other connectors for Cat5/Cat5e cable for connection to a DAD network, and is configured to receive electrical power from, for example, four of the eight conductors available in a Cat5 cable. The controller 700 is configured to fit into a standard 2-gang electrical workbox and includes a cover 702 that can be attached to the workbox with clips, screws, or other mounting mechanisms. The controller 700 is also configured to receive an amplifier module to power speakers as well.

[0030] The controller 700 includes user perimeter input buttons 704-709 and central input buttons 722-726. The input buttons can be configured to select listening sources, name the sources, define the routing of sources, set parametric equalization and auto-leveling compression and limiting, as well as adjust volume and balance in a local listening area (a local zone) or in remote zones. A display unit 730 includes a display area configured to display source names or other data, and regions 731-733 configured to display audio power, balance, tuner signal strength and other audio network and audio source information. The controller 700 is generally configured so that switching between audio sources produces a default audio volume level, typically -30 dB, unless an auto level function is activated so that audio output level is unchanged. If the input buttons are un-

D+M-PA_0000701

LINKPLAY EXHIBIT 1004
Page 1898 of 4822

used for a predetermined time such as, for example, 10 seconds, a currently selected audio source name is displayed such as CD, DVD, tuner etc. on the display 730. The controller 700 also includes an IR input 712.

[0031] With reference to FIG. 7B, an alternative representative controller 750 includes general purpose input buttons 751-755 as well as illuminated buttons 762, 764 associated with selection of auto audio leveling and loudness, respectively. Pressing such buttons toggles the corresponding function from on to off or off to on. If the function is active, the associated button can remain illuminated. Source selection buttons 766-769 permit selection of a source for audio delivery or assignment of configuration parameters. These buttons function similarly to the input buttons 751-755.

[0032] Controllers such as the representative controllers 700, 750 can be used as follows. For example, in response to pressing and holding an "ENTER" button for 1 sec., messages such as "Press UP and DOWN buttons to adjust Volume" or "Press LEFT and RIGHT buttons to adjust balance" are presented on a display. Pressing a "SOURCE" button allows an installer or user to select a source and assign the source to multiple destinations and create a zone from these destinations. The display can present messages such as, for example, "Use the UP / DOWN buttons to select a source," "Use the RIGHT / LEFT buttons to select destinations" and "Press ENTER to link source/destination." A selected input button can be designated as, for example, an "Auto Volume" button to adjust all audio sources to produce a similar volume in a selected zone or zones, or a "Remote Off" button to turn off a remote input from an I/O module in a selected zone. Transmission from selected I/O Modules in other zones can also be inhibited. One or more of the input buttons can serve as a menu button for user selection of functions such as equalization adjustments and assignment of names to rooms and zones as well as configuring zones. An IR sensor in a controller is configured to receive IR commands from a remote control for transmission over the network back to one or more source components, as well as commands associated with volume control or balance in one or more zones, zone muting, and switching a zone or the system on or off.

[0033] With reference to FIG. 8, a power module 800 is configured for connection to a digital audio distribution network using IDC, RJ45, or other connectors and CAT-5 or CAT-5e cable, or other cables. The power module 800 also includes a coaxial or other input connector to receive supplemental power from an external power source (not shown in FIG. 8). The power module 800 includes a green LED indicator 802 configured to indicate that an additional power supply connected to the power module 800 is providing power. Supplemental power is provided to the network using selected conductors in the CAT-5 or other cable. The power module 800 is typically included in a network installation to compensate power losses produced by, for example, power distribution over system cables. The power module 800 also provides, for example, 24-bit, 48 kHz digital audio transmission. In a particular example, the power module 800 is configured to receive 24 V DC from the supplemental power source. The power module 800 is conveniently configured for mounting in a conventional electrical box. In other examples, a power module includes an internal power supply. The power module can be configured to act as a hub as well, and can include IR sensors, D-A or A-D converters, DSP, and other functions.

[0034] With reference to FIG. 9, an input/output (I/O) module 900 for a DAD network includes a connector configured to receive a CAT-5 or CAT-5e cable. The I/O module 900 also includes left and right stereo inputs 902, 903 configured to receive an analog signal from a remote source, typically a source located in a zone associated with the I/O module 900. The I/O module 900 includes a digital audio processor that is configured to convert the analog signal to a corresponding digital signal that can be included in a digital audio network distribution signal that is transmitted to other zones. The I/O module 900 also includes left and right channel stereo outputs 904, 905 configured for delivery of an audio signal associated with a selected source to a local amplifier (i.e., an amplifier associated with a selected zone) or directly to speakers or headphones. The inputs and outputs can be configured to use so-called RCA connectors or other types of connectors. A power supply input connector 908 is also provided if additional electrical power is needed due to, for example, a large number of DAD network components and an associated large power demand, or power losses associated with deliver of power to the I/O module 900 via long cable lengths. The I/O module 900 also includes a connector 910 (e.g., a 3.5 mm phone jack) to deliver a control signal to an IR emitter for control of a source in the local zone from a remote zone. A similar connector 912 is configured as a 2 V trigger input to receive a signal from a local signal source that indicates that the local signal source is operational. Data associated with the trigger input is included in the digital audio network distribution signal that is transmitted by the I/O module 900. For convenience, the I/O module 900 can be configured to fit in a standard electrical box. The I/O module 900 can also include a DSP or other processor, one or more internal audio sources, or be configured to receive digital audio directly from one or more audio sources.

[0035] Another example of a user control module is illustrated in FIGS. 10A(1)-A(3). The user control module is similar to that shown in FIGS. 6A-6C, but includes a processor 1002 configured to communicate with an IR emitter and an IR detector via a jack 1004 and a connector 1006, respectively. The processor can be, for example, an SP8001 Serial Port Remote Control Integrated Circuit with "Learning" produced by Innotech Systems Inc. Such a processor includes a library of remote control codes for remote controls made by many manufacturers and can be operated to acquire or "learn" ad-

D+M-PA_0000702

LINKPLAY EXHIBIT 1004
Page 1899 of 4822

ditional codes that can be associated with activation of various keys/buttons on a remote control. In addition, such a processor can be configured to automatically search for a correct code. In addition, such a processor can store three digit ("ISI") codes for several devices. The processor 1002 also includes outputs associated with LEDs 1008, 1009 for evaluating progress in learning codes. In a particular example, a code is learned by providing a button code for a virtual button to be associated with a code and the code to be learned is directed from a remote control to an IR detector and to the processor 1002. Confirmation that the code is learned is obtained by monitoring the LEDs 1008, 1009. With such a processor, commands associated with arbitrary remote control codes can be implemented without hardware modification. The user control module (or a network hub or other network element) can recognize control codes received via IR, wireless, or otherwise and convert or translate control codes for control of a selected device such as a program source or a network element. Such a processor can be associated with other network elements such as network hubs or input/output modules or other network locations. Some commands can be executed directly at the user control module (volume changes, muting) while other commands are directed to other network locations. For example, a command associated with a pause function is directed to the program source to be paused. In another example, a command is associated with selection of a different program source, and execution of such a command can be based on selection of particular audio data from a digital audio data distribution signal, without communication with the program source. In some cases, the digital audio data signal does not include the selected program material, and the command is directed appropriately in the network.

[0036] FIG. 10B illustrates processing of a command received by a control module from a remote control such as an IR remote. In a step 1050, a command is received and in a step 1052 the command is checked to determine if the command is a known command. If the command is not recognized as a known command, an error can be posted or control passed to a command learning procedure 1054. If the command is recognized, the command can be evaluated to determine if the command can be executed by the control module in a step 1056. If a "local" command is received, the command is executed in a step 1060. Otherwise, the command is distributed by the network in a step 1062 to other network locations in the digital audio data for execution at an appropriate location.

[0037] The user control module of FIG. 10A also includes a serial interface 1012 that can be configured to receive or transmit network configuration parameters such as, for example, program source information, equalization settings, zone definitions, or other settings. The serial interface 1012 can be implemented based on, for example, RS-232, universal serial bus, FIREWIRE (IEEE 1394), or other communication protocol. As shown in FIG. 10A, serial output is delivered to and receive from a DB9 connector 1014. Using the serial interface 1012, network settings can be stored, evaluated, or copied to duplicate a particular network configuration. In addition, the serial interface 1012 can be used for communication with an additional touch panel controller, a personal computer, a personal digital assistant, or other control system. Such a control system can include a display that permits visual presentation of the network configuration and network parameters. The serial interface 1012 can also be configured to communicate with a touch panel controller using a wireless interface. In addition, the serial interface 1012 can be used to reprogram or update an installed network to, for example, learn additional commands or to add additional features such as room monitoring, door chime functionality, or a security system interface if not provided at installation. The serial interface 1012 can also be used to establish a remote connection to the network using, for example, a phone line and a modem for remote control, repair, and evaluation. Alternatively, the serial interface 1012 can be configured to communicate over the Internet.

[0038] In other examples, audio distribution systems can be configured to distribute audio program material or other program material to one or more zones that can be configured as one or more rooms or other regions in a user's residence or business, or other location. As noted above, program material can be situated in one or more zones and made available to all zones or to selected zones, and while a central location for generation of source material can be used, such a location is not necessary. A system can be arranged in a point-to-point configuration (a "daisy chain") or in a star configuration. A typical point-to-point configuration is illustrated in FIG. 11. An equipment rack 1100 includes various program sources such as a first tuner 1101, a CD player 1102, a cassette player/recorder 1103, a second tuner 1104, and a slot 1105 for an additional source. Analog audio signals from these program sources are delivered to a hub 1106 that is in communication with a user interface module 1108 situated in, for example, a kitchen. The user interface module 1108 is typically configured to control program material delivered to speakers 1110, 1111 and to receive and transmit digital data to/from the hub 1106 and a user interface 1114 situated in a study or other room or zone. Digital data is communicated between the user interface 1114 and a user interface 1116 situated in master bedroom or other zone. The user interface 1116 is in communication with an input/output unit 1118 that is configured to communicate analog program material to and from a television 1120. The input/output unit is in communication with a user interface 1122 situated at a patio or other zone or location. As shown in FIG. 11, each zone/room includes speakers that a configured to receive analog program material from the associated user interface. Category 5e cable and RJ45 connectors are typically used to interconnect the user interface modules and input/output modules,

D+M-PA_0000703

LINKPLAY EXHIBIT 1004
Page 1900 of 4822

but other types of cables can be used, or one or more of the system modules can include a wireless interface so that cables are not needed. In addition to program material, command, control, and program information can be communicated from zone to zone or room to room in the digital audio data signal. Typically each of the digital network components (such as, for example, input/output modules, user interfaces, or hubs) can be associated with a device identifier that can be set with, for example, one or more encoder wheels.

[0039] With reference to FIG. 12, a digital audio network 1200 arranged in a star configuration includes user interface modules 1202, 1204, 1206, 1208 that are in communication with an expansion module 1210. The expansion module 1210 is in communication with a hub 1214 and a set of program material sources 1212. As shown in FIG. 12, each user interface module is associated with a single user location (patio, bedroom, study etc.), but in other examples, one or more user interface modules can be associated with a collection of rooms or other locations that form a zone. A master bedroom zone 1216 include an input/output module 1218 to, for example, communicate an audio portion of television programming to the network 1200 or to supply audio programming from, for example, any of the set of sources 1212 to a television 1220.

[0040] FIG. 13 illustrates another example network that includes an equipment rack 1302 configured to retain, for example, AM/FM tuners 1304, 1305, a satellite tuner 1306, CD and DVD players/recorders 1307, 1308, a videocassette recorder 1309, a cable television box 1310, and a cassette player/recorder 1311. Hubs 1312, 1314 are configured for distribution to respective point-to-point networks. Zone 1320 includes a power amplifier 1322 that processes analog audio signals for delivery to speakers 1324, 1325.

[0041] Other configurations of network components include a hub that includes a processor 1402 configured to learn remote control commands and a serial interface 1404 as shown in FIGS. 14A-14C. FIGS. 15A-15B illustrates an additional keypad controller configuration. FIG. 16 illustrates an IR output module that includes jacks 1601-1608 configured to communicate IR emitter drive signals based on processed IR received at an input 1610. In the above examples, any zone can typically be configured to receive and transmit control data to, for example, control remote program sources (e.g., turn on or off a CD player in another room, or select an FM channel at a remote tuner) or to control the network (e.g., select parameters for audio processing, or select audio sources for delivery to a selected zone). Control data is conveniently provided using infra-red emitting diodes and detectors, and an IR remote in one zone can select and control program material delivery in other zones.

[0042] Typically, network components and modules are interconnected using so-called Cat 5 or similar cables using RJ45 connectors. These components and modules are configured so that crossover cables are unnecessary, and all network cables can be configured as "straight-through" cables.

[0043] User interface modules can be conveniently arranged as wall mounted key pads with an LCD or other display that is configured to show, for example, the source currently being played, other available sources, equalization and loudness settings, volume, tone and other settings. In addition, a user interface can be configured to show a source material identifier such as a performer or song name and this information included in a digital audio data distribution signal. The user interface module can also be configured to indicate user preferences for tone, equalization, loudness etc. as a function of volume or other parameter. Typically a user interface module is configured to receive an amplifier module so that the user interface can directly drive one or more speakers. Amplifier power is typically received at the user interface module via the associated Cat 5 cable. In addition, user interface modules are generally configured to include a "last source recall" so that upon activation of a single button, a recently played source is returned to a previous configuration. For example, a particular FM station can be selected via "one touch" and played with previous volume, tone, equalization, or other parameters. In addition, a user interface module can be configured so that audio levels associated with all or a selected number of program sources are played at a common audio level.

[0044] Other system modules include power input modules to provide additional power, system expanders, and infrared (IR) repeaters configured to reproduce IR commands sent from remote zones to a program material source. An IR repeater is typically configured to be situated at an IR receiver at the program source.

[0045] For some program sources, program material details such as composer, performer, selection length, playlists, or other details are available, and such information can be provided in the digital audio distribution signal. For example, program information concerning channel selection and current program material obtained from a satellite tuner can be distributed. In addition, text messages can be distributed.

[0046] Only selected specific examples of audio networks and the use of these networks is described herein, but it will be apparent that such networks permit users to conveniently select, process, control, and deliver audio signals from one or more sources to one or more destinations or zone. For example, a user located in a first zone can direct a command to an IR remote to select a remote audio source for audio delivery to the first zone (or other zones), and control the audio source by selecting, for example, CD track, FM tuner frequency, or other audio source configuration. A network hub in the first zone is response to the IR remote, and processes a remote control command for the network. In some cases, the remote control command is relayed to the audio source or other destination. For example, selection of an audio source is based on network hub configuration,

D+M-PA_0000704

LINKPLAY EXHIBIT 1004
Page 1901 of 4822

while audio source control (e.g., initiating playing of a CD or selecting a CD track) is based on delivery of a command to an audio source.

[0047] In the examples above, examples of particular network elements have been described with respect to associated network functions. However, any network element can be configured as a network controller or a user input device, and can serve as a hub. Such assignments of functionality to particular network elements in the examples is for convenience in explanation only. In addition, in the examples, digital audio distribution signals are transmitted and received on respective assigned pairs of an 8-conductor cable while the remaining four conductors are used to provide electrical power. In other examples, signals and power can be arranged on additional or fewer conductors, and the used of 8-conductor CAT-5 cable is for convenience only. In addition, analog or other signals associated with audio, video or other media can be transmitted or received using some cable conductors. In some examples, system functions are implemented in hardware or software, but generally functions can be implemented in any combination of hardware and software, or using only hardware or software. In other examples, program parameters can be stored at some or all distribution points. For example, parameters for functions such as automatic loudness control can be stored for frequently used or favorite sources and distribution zones. Thus, it will be apparent that the examples described above can be modified in arrangement and detail without departing from the scope of the disclosure. We claim all that is encompassed by the appended claims.

Claims

1. A digital audio network, comprising:

a hub configured to receive a plurality of stereo signals and combine the stereo audio signals with control data to produce a digital audio network signal; and
an output connector configured to communicate the digital audio distribution signal; and
an interface configured to communicate at least one network parameter.

2. The digital audio network of claim 1, wherein the hub is configured to include program material information in the digital audio distribution signal.

3. The digital audio network of claim 1, wherein the output connector is configured to deliver the digital audio distribution signal to a pair of conductors.

4. The digital audio network of claim 1, further comprising an interface configured to communicate at least one network parameter.

5. The digital audio network of claim 1, wherein the output connector is configured to deliver the digital audio distribution signal to a first pair of conductors in a CAT-5 cable, and to deliver electrical power to at least a second pair of conductors in the CAT-5 cable.

6. The digital audio network of claim 1, further comprising an input connector configured to receive a digital audio distribution signal.

7. The digital audio network of claim 1, further comprising a remote control processor configured to process remote control commands.

8. The digital audio network of claim 1, further comprising a display configured to present program material information.

9. A controller for a digital audio network, comprising:

a keypad configured for entry of at least one audio source identifier or a configuration parameter for at least one audio zone;
an audio processor configured to receive an input digital audio network distribution signal that includes digital audio data and control data, and produce an output digital audio network distribution signal based on the entry of the at least one audio source identifier or configuration parameter; and
a serial interface configure to communicate at least one of the at least one audio source identifier or the configuration parameter for the at least one zone.

10. The controller of claim 9, further comprising a display configured to display the at least one audio source identifier or configuration parameter.

11. The controller of claim 9, further comprising a remote control processor configured to identify an unknown remote control command and to indicate entry into a command acquisition mode.

12. The controller of claim 11, wherein the audio processor is configured to include the control data associated with the unknown remote control command into the output digital audio network distribution signal after command acquisition.

13. An audio network device, comprising:

an infrared sensor configured to receive a remote control command;
a control data processor that includes an indicator configured to indicate if the received re-

D+M-PA_0000705

LINKPLAY EXHIBIT 1004
Page 1902 of 4822

mote control command is undefined and acquire an associated command definition; and
an audio processor configured to produce a digital audio distribution signal based on the command definition.

14. The audio network device of claim 13, further comprising:

an input configured to receive at least one audio input signal, wherein the processor is configured to produce a digital audio signal based on the input audio signal and the digital audio distribution signal is based on the digital audio signal.

15. The audio network device of claim 14, further comprising a user input configured to select at least one audio signal to be decoded from the digital audio distribution signal.

16. The audio network device of claim 14, further comprising an interface configured to communicate at least one network parameter.

17. The audio network device of claim 16, wherein the interface is a serial interface.

18. An audio distribution method, comprising:

receiving at least one audio signal from an audio signal source;
digitizing the at least one audio signal to produce a digital audio signal; and
combining the digital audio signal with control data received from a remote control to produce a digital audio distribution signal.

19. The audio distribution method of claim 18, further comprising identifying a command received from a remote control as a new command.

20. The audio distribution method of claim 19, further comprising associating a network function with the new command.

21. An audio distribution method, comprising:

receiving a plurality of digital audio signals;
receiving control data;
determining a location for execution of a command associated with at least a portion of the control data; and
producing a digital audio distribution signal based on a combination of selected digital audio signals and the command location.

22. The method of claim 21, further comprising executing a command associated with the control data without inclusion of the command in the digital audio distribution signal.

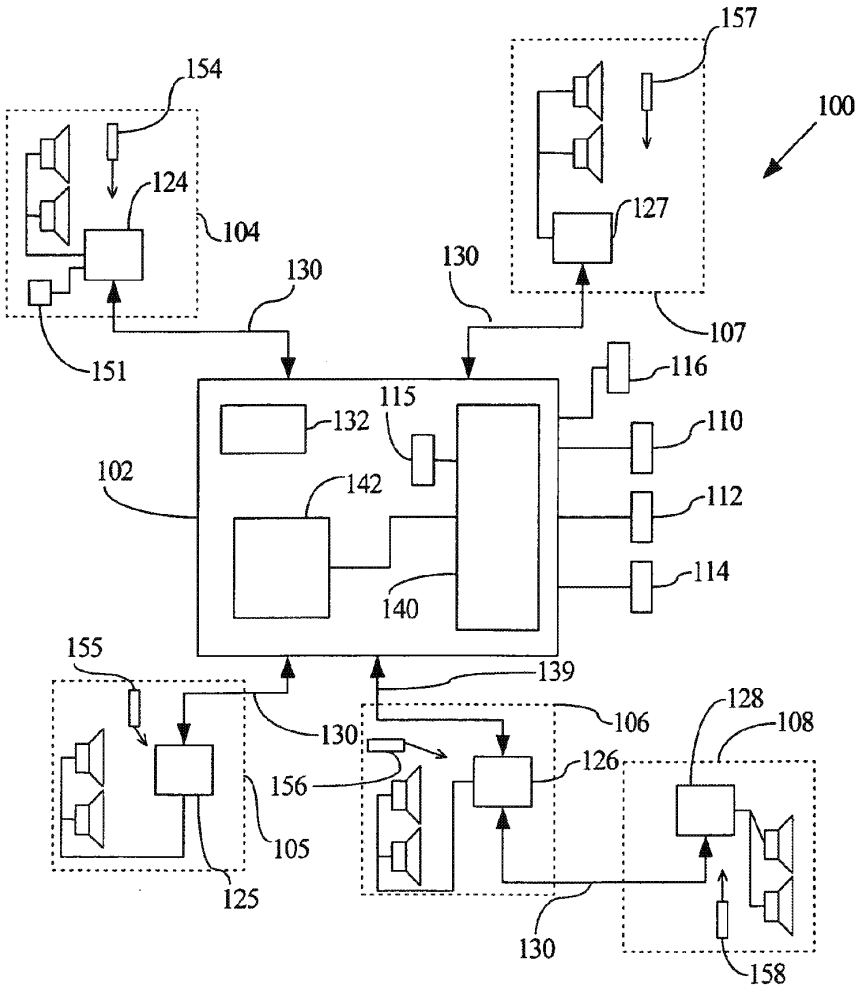
23. The method of claim 21, further comprising including the command in the digital audio distribution signal.

25. The method of claim 22, wherein the control data is associated with at least one of volume, loudness, equalization, balance, or muting.

D+M-PA_0000706

LINKPLAY EXHIBIT 1004
Page 1903 of 4822

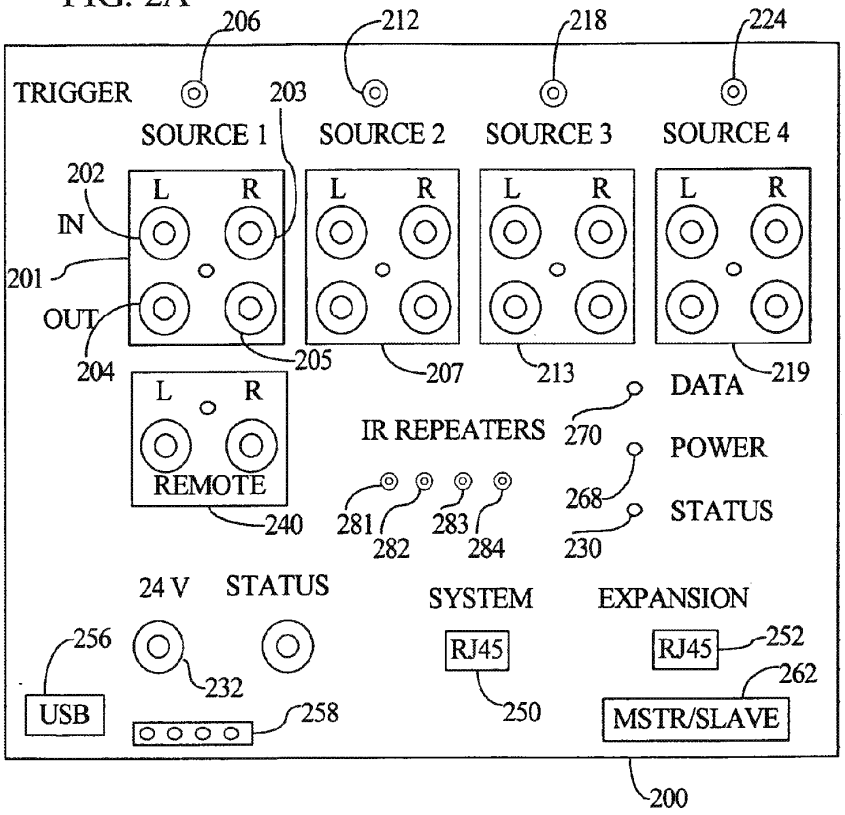
FIG. 1



D+M-PA_0000707

LINKPLAY EXHIBIT 1004
Page 1904 of 4822

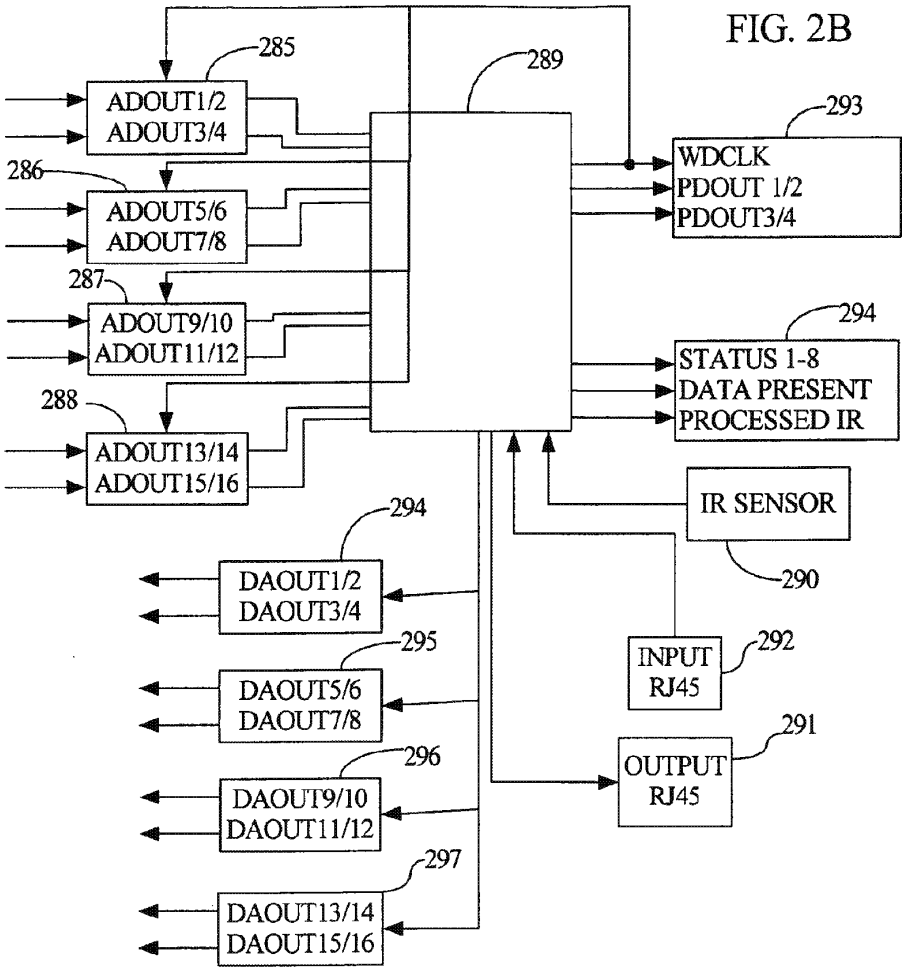
FIG. 2A



D+M-PA_0000708

LINKPLAY EXHIBIT 1004
Page 1905 of 4822

FIG. 2B



D+M-PA_0000709

LINKPLAY EXHIBIT 1004
Page 1906 of 4822

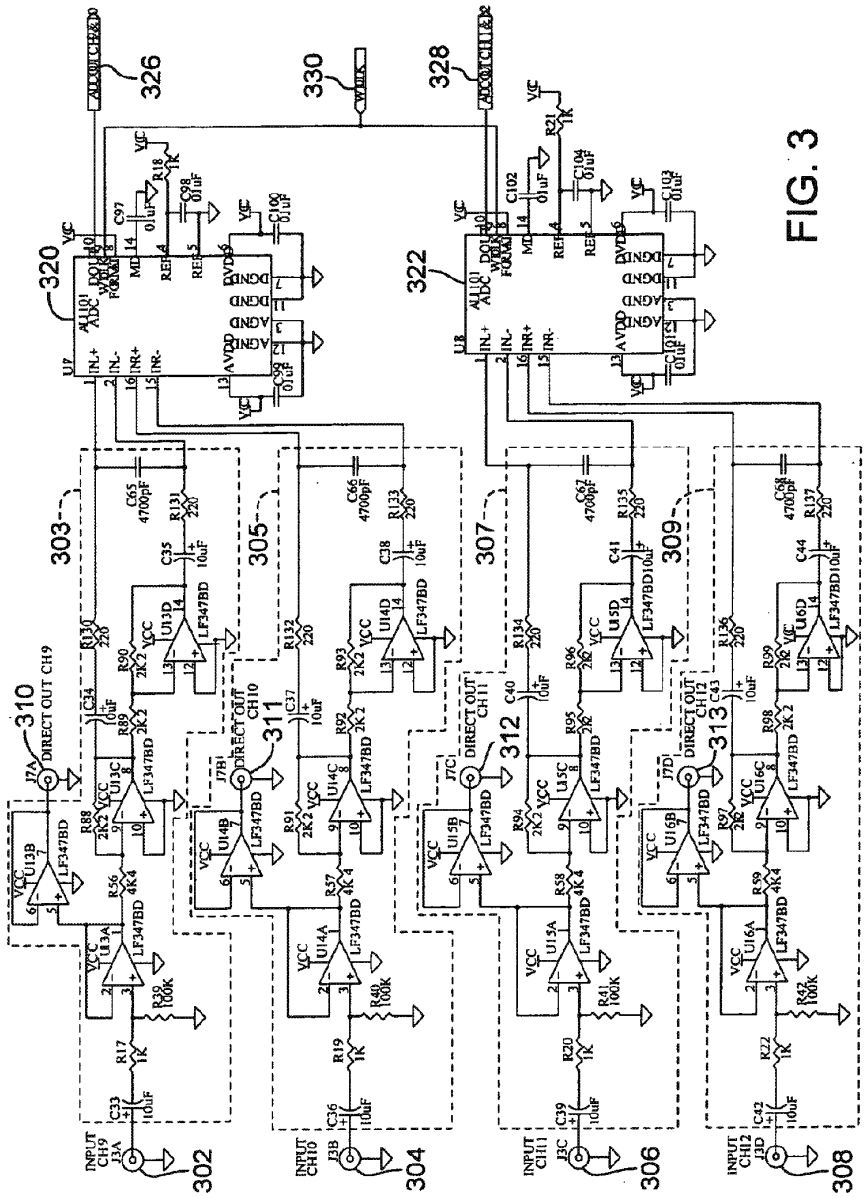


FIG. 3

D+M-PA_0000710

LINKPLAY EXHIBIT 1004
Page 1907 of 4822

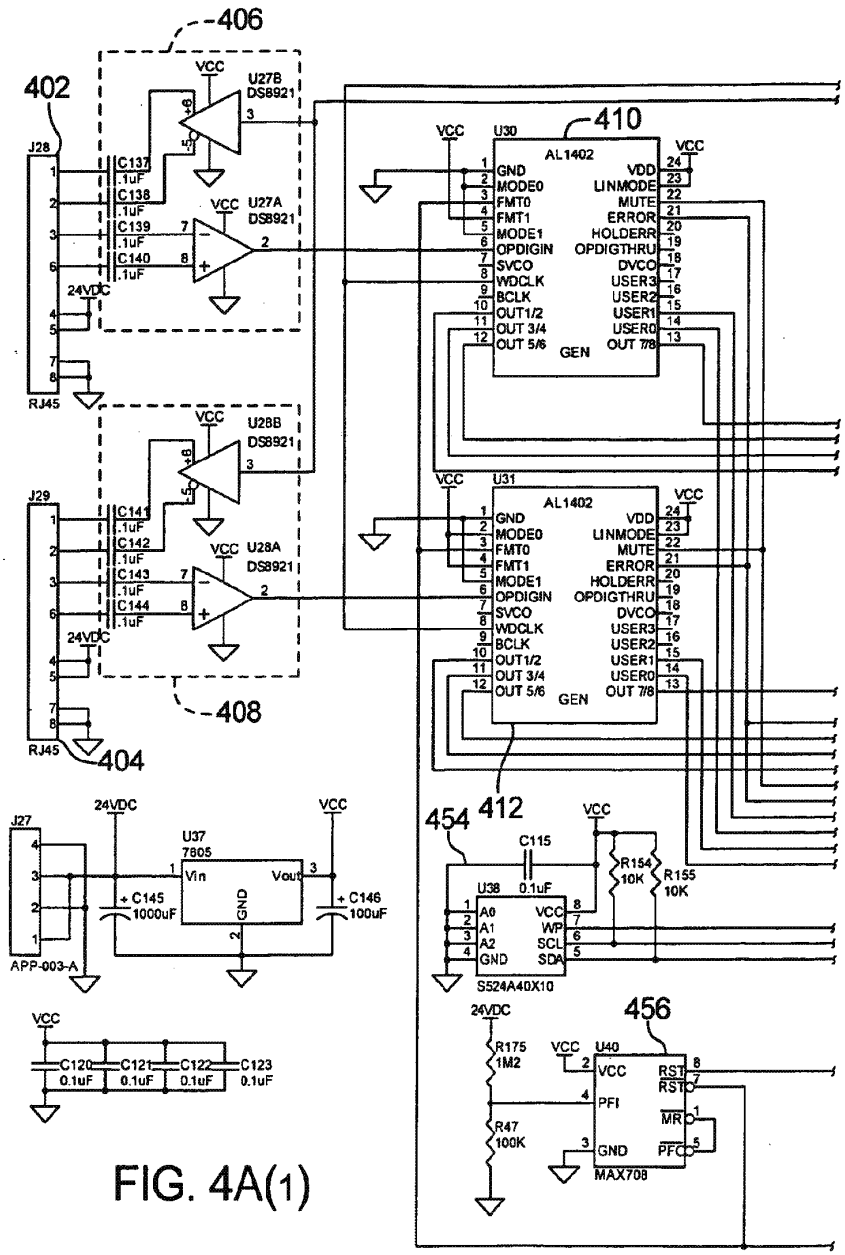


FIG. 4A(1)

D+M-PA_0000711

LINKPLAY EXHIBIT 1004
Page 1908 of 4822

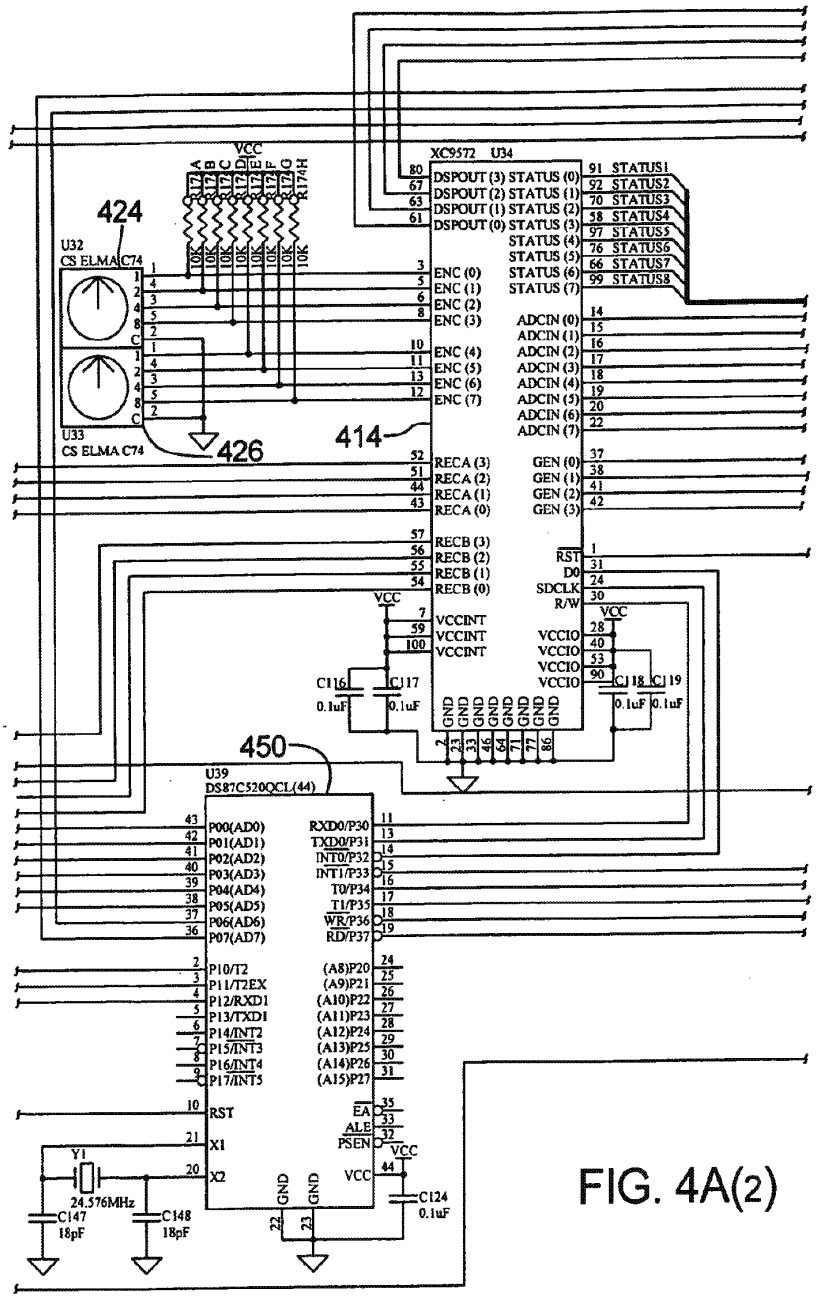


FIG. 4A(2)

D+M-PA_0000712

LINKPLAY EXHIBIT 1004
Page 1909 of 4822

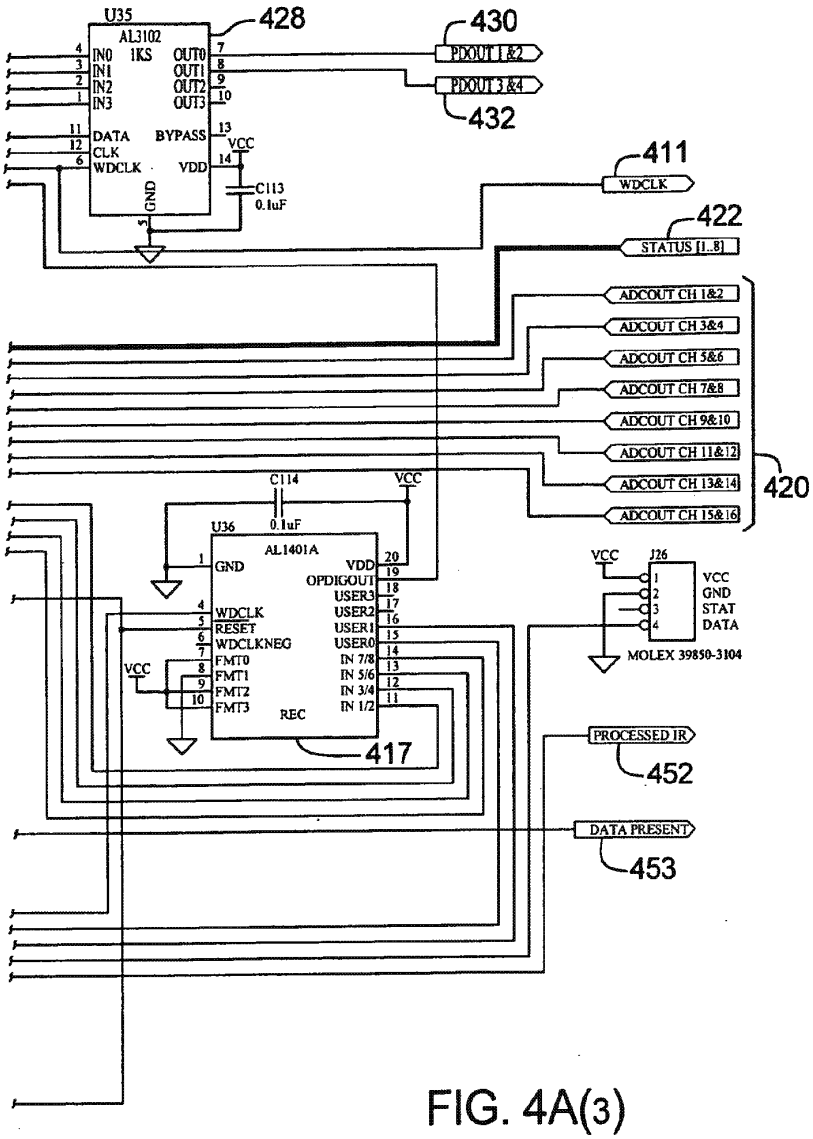


FIG. 4A(3)

D+M-PA_0000713

LINKPLAY EXHIBIT 1004
Page 1910 of 4822

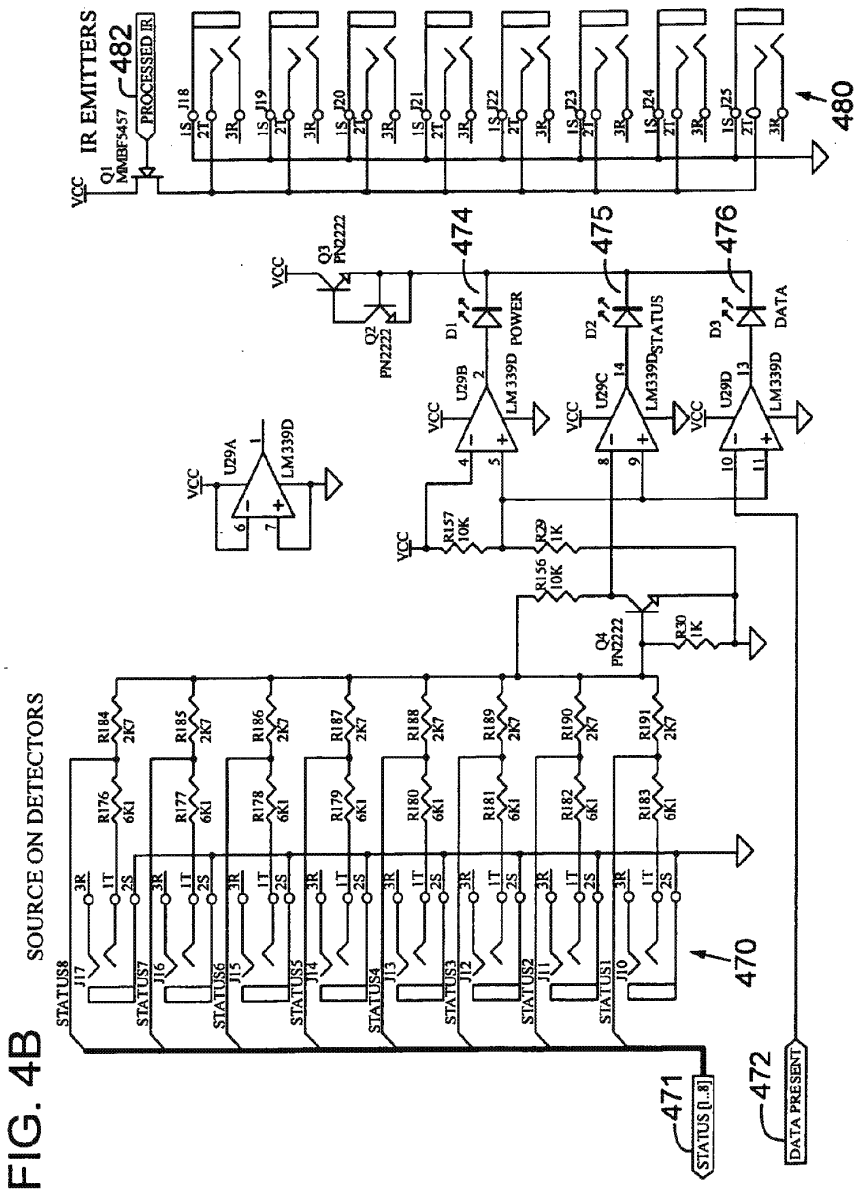


FIG. 4B

D+M-PA_0000714

LINKPLAY EXHIBIT 1004
Page 1911 of 4822

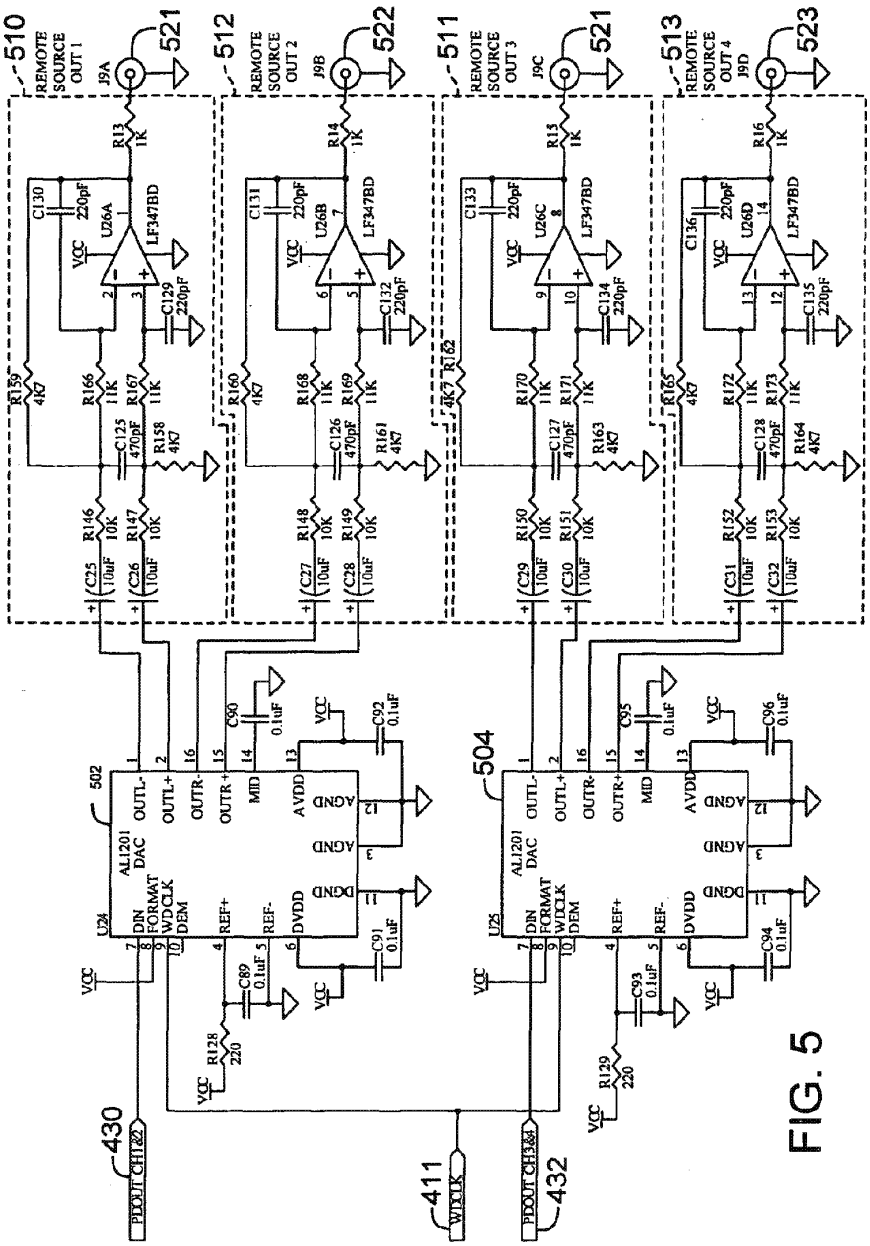
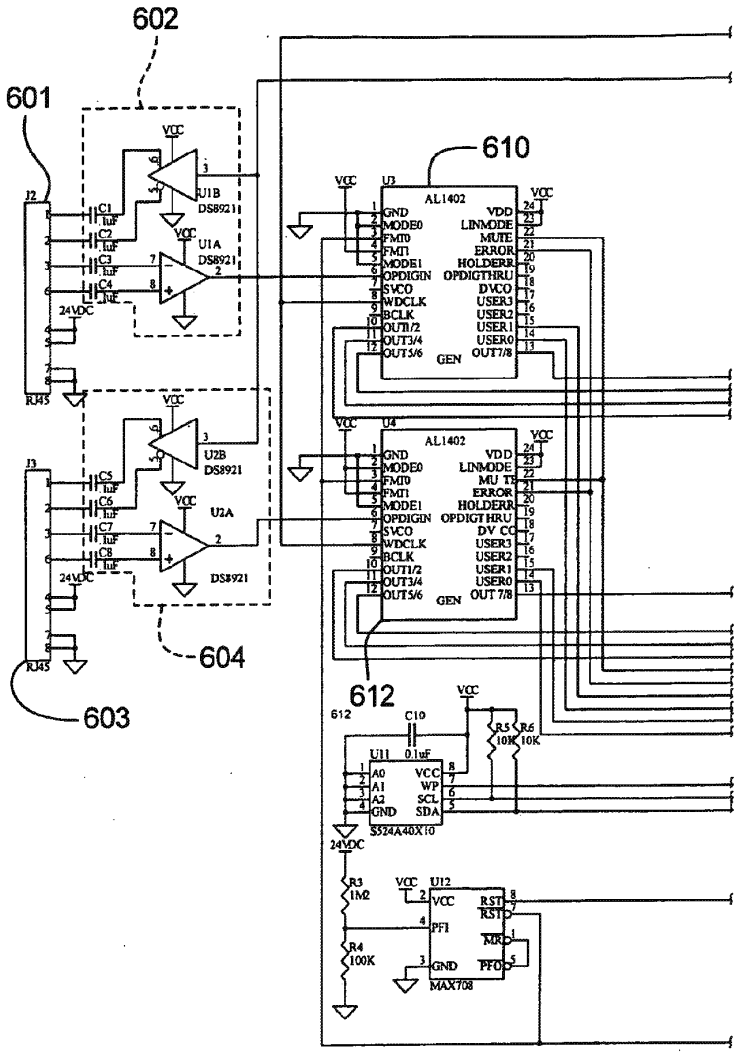


FIG. 5

D+M-PA_0000715

LINKPLAY EXHIBIT 1004
Page 1912 of 4822

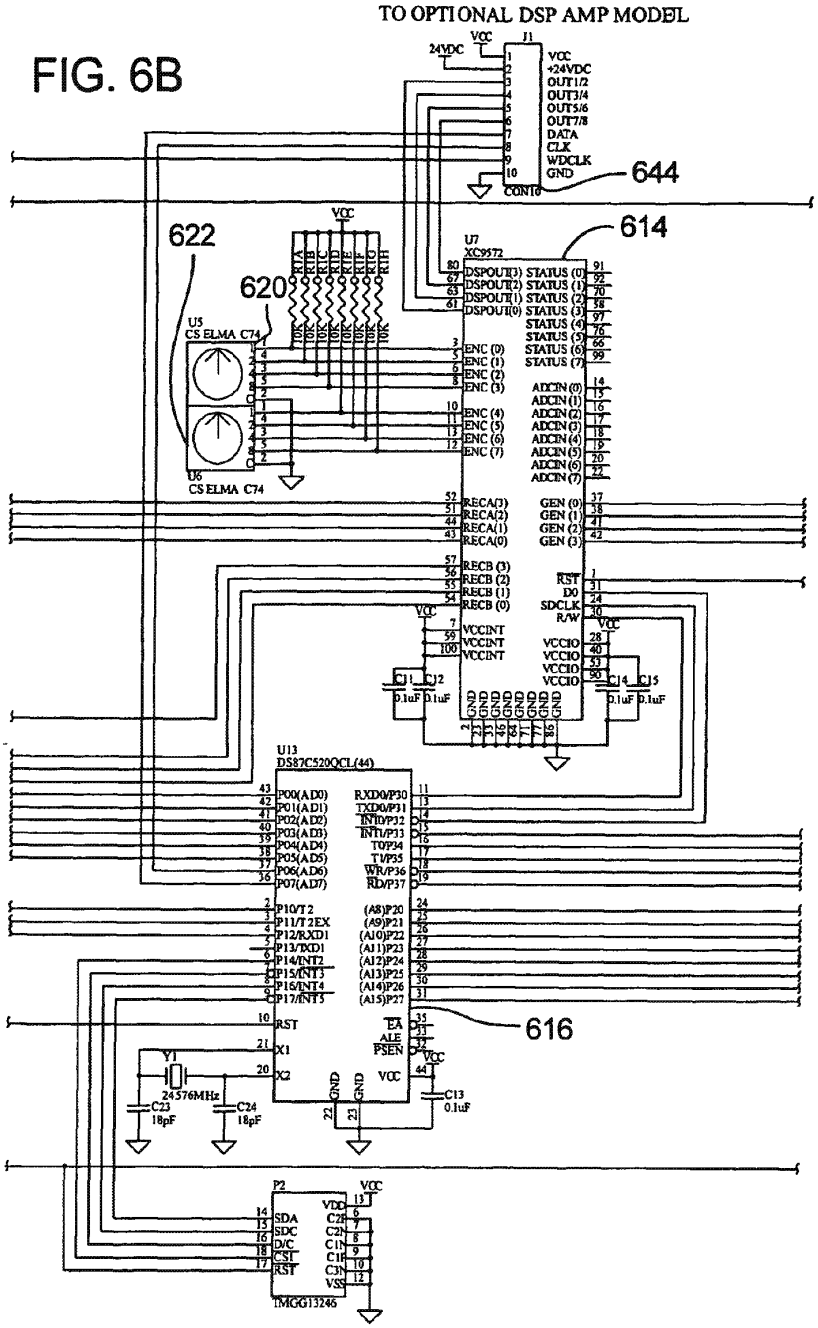
FIG. 6A



D+M-PA_0000716

LINKPLAY EXHIBIT 1004
Page 1913 of 4822

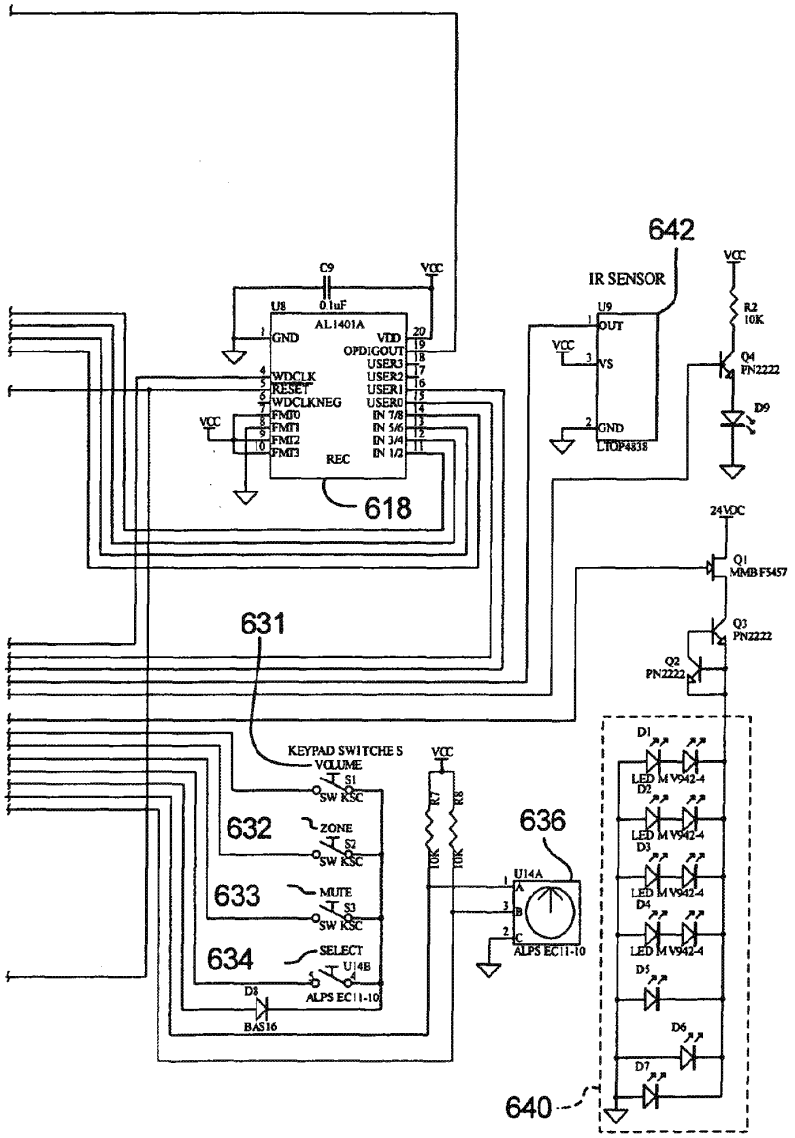
FIG. 6B



D+M-PA_0000717

LINKPLAY EXHIBIT 1004
Page 1914 of 4822

FIG. 6C



D+M-PA_0000718

LINKPLAY EXHIBIT 1004
Page 1915 of 4822

FIG. 7A

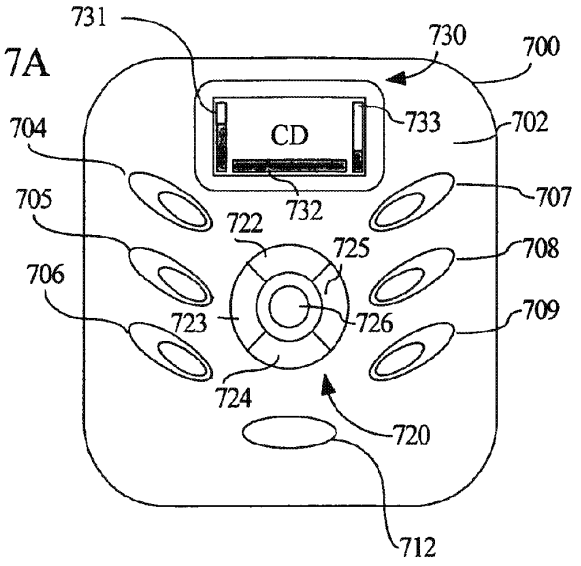
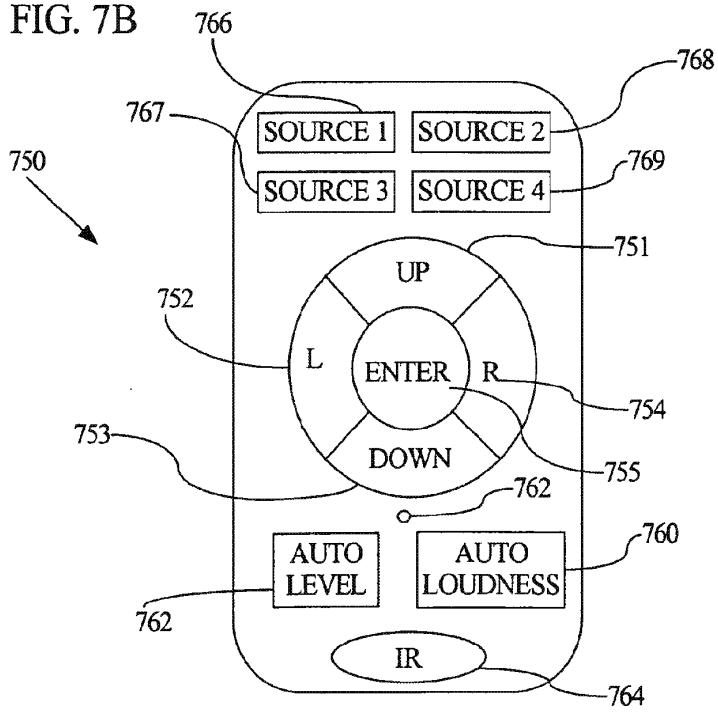


FIG. 7B



D+M-PA_0000719

LINKPLAY EXHIBIT 1004
Page 1916 of 4822

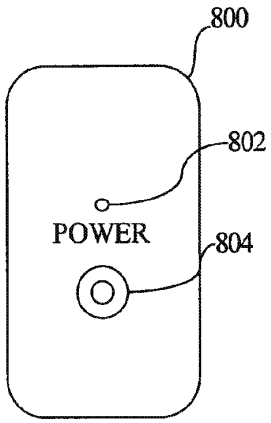


FIG. 8

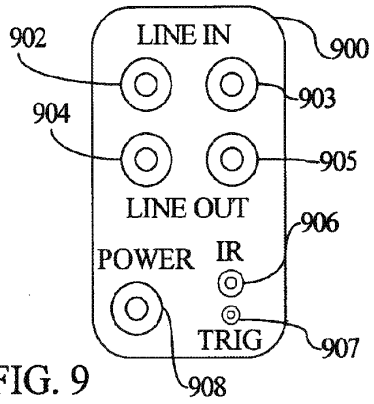


FIG. 9

D+M-PA_0000720

LINKPLAY EXHIBIT 1004
Page 1917 of 4822

D+M-PA_0000721

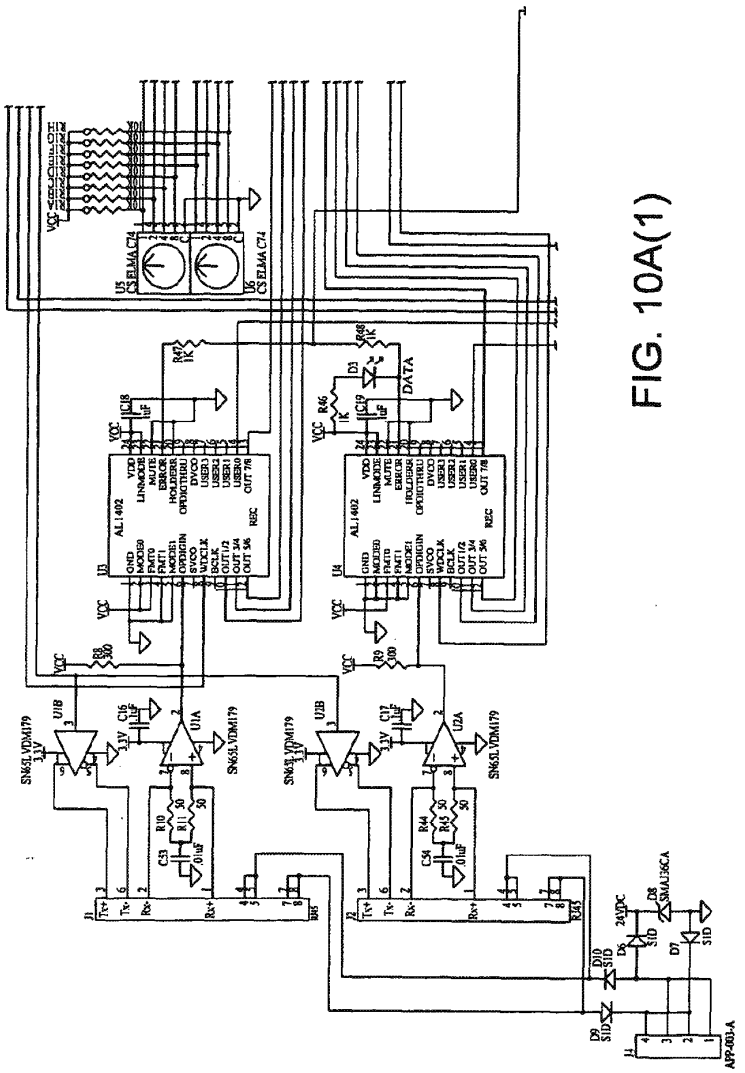


FIG. 10A(1)

FIG. 10A(2)

D+M-PA_0000722

FIG. 10A(3)

D+M-PA_0000723

LINKPLAY EXHIBIT 1004
Page 1920 of 4822

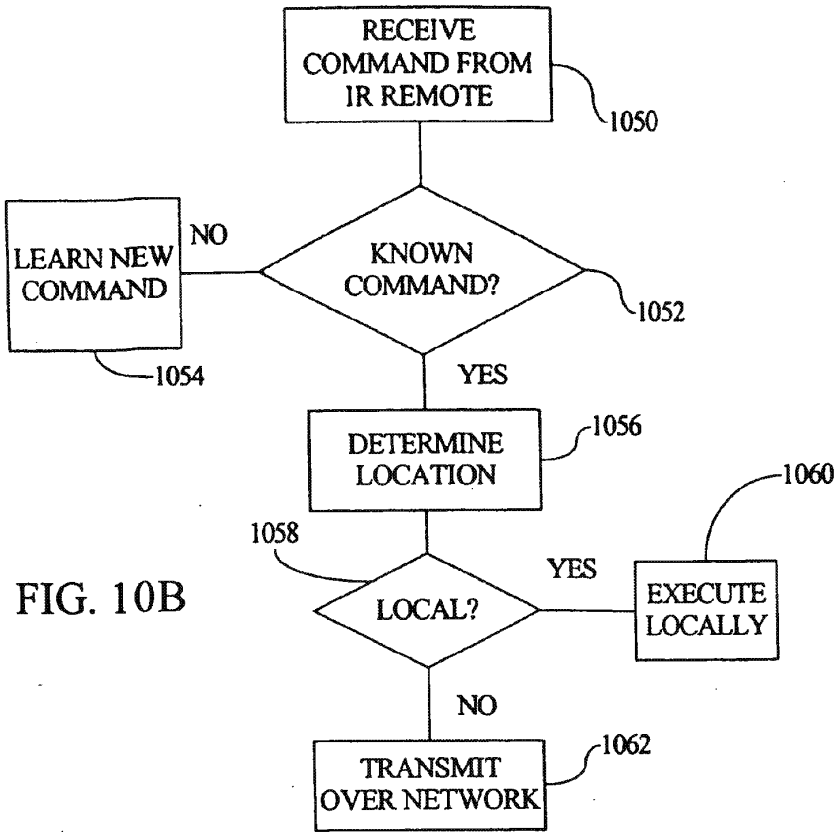


FIG. 10B

D+M-PA_0000724

LINKPLAY EXHIBIT 1004
Page 1921 of 4822

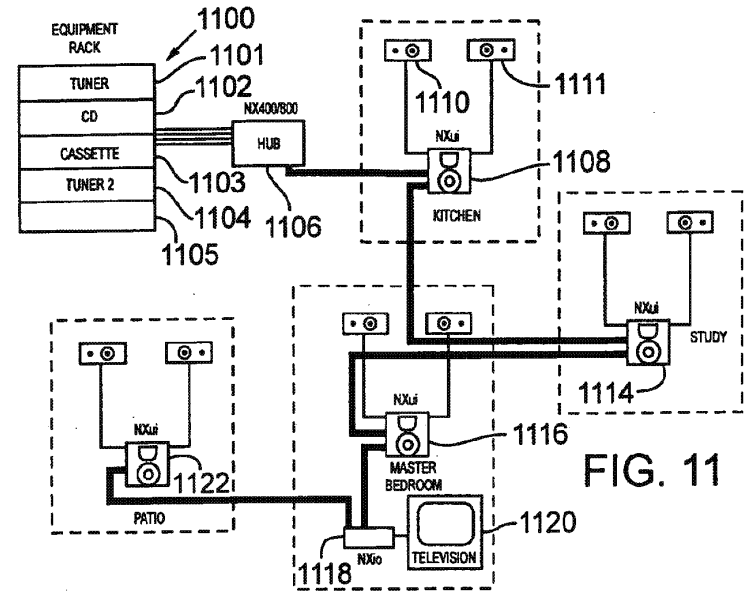


FIG. 11

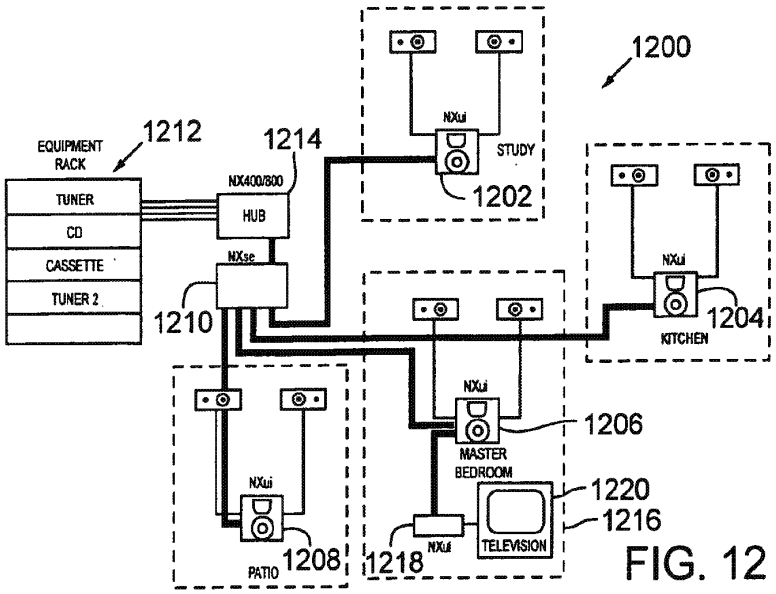


FIG. 12

D+M-PA_0000725

LINKPLAY EXHIBIT 1004
Page 1922 of 4822

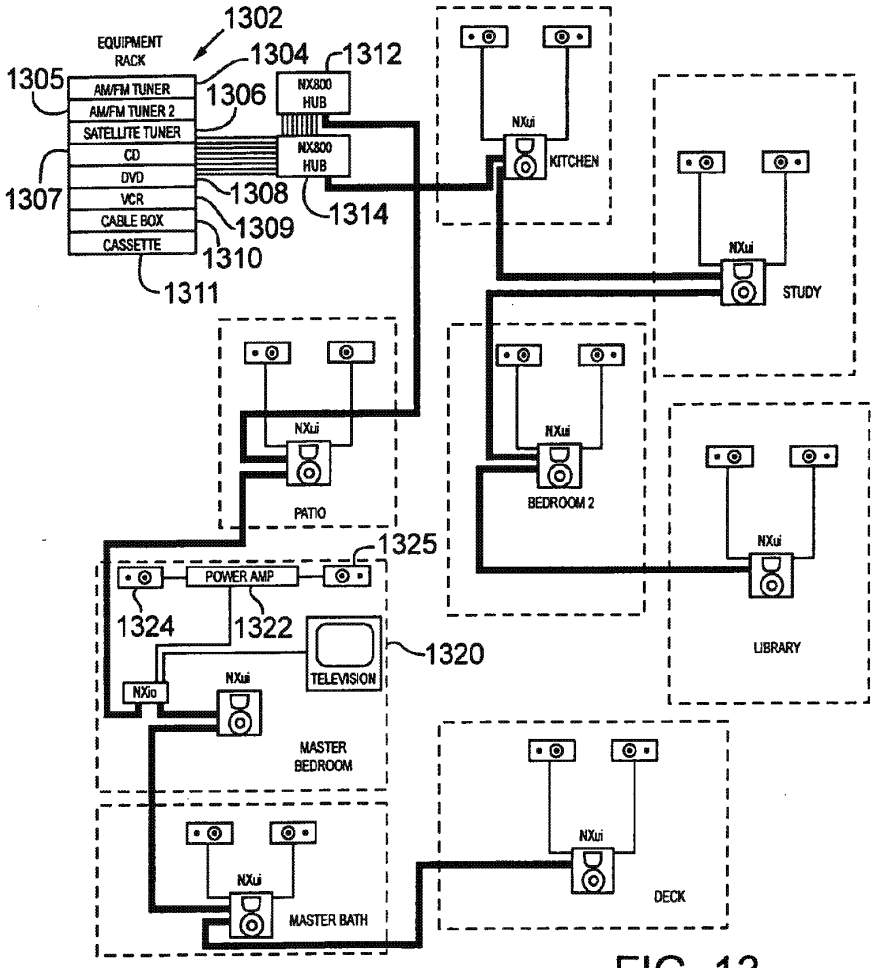


FIG. 13

D+M-PA_0000726

LINKPLAY EXHIBIT 1004
Page 1923 of 4822

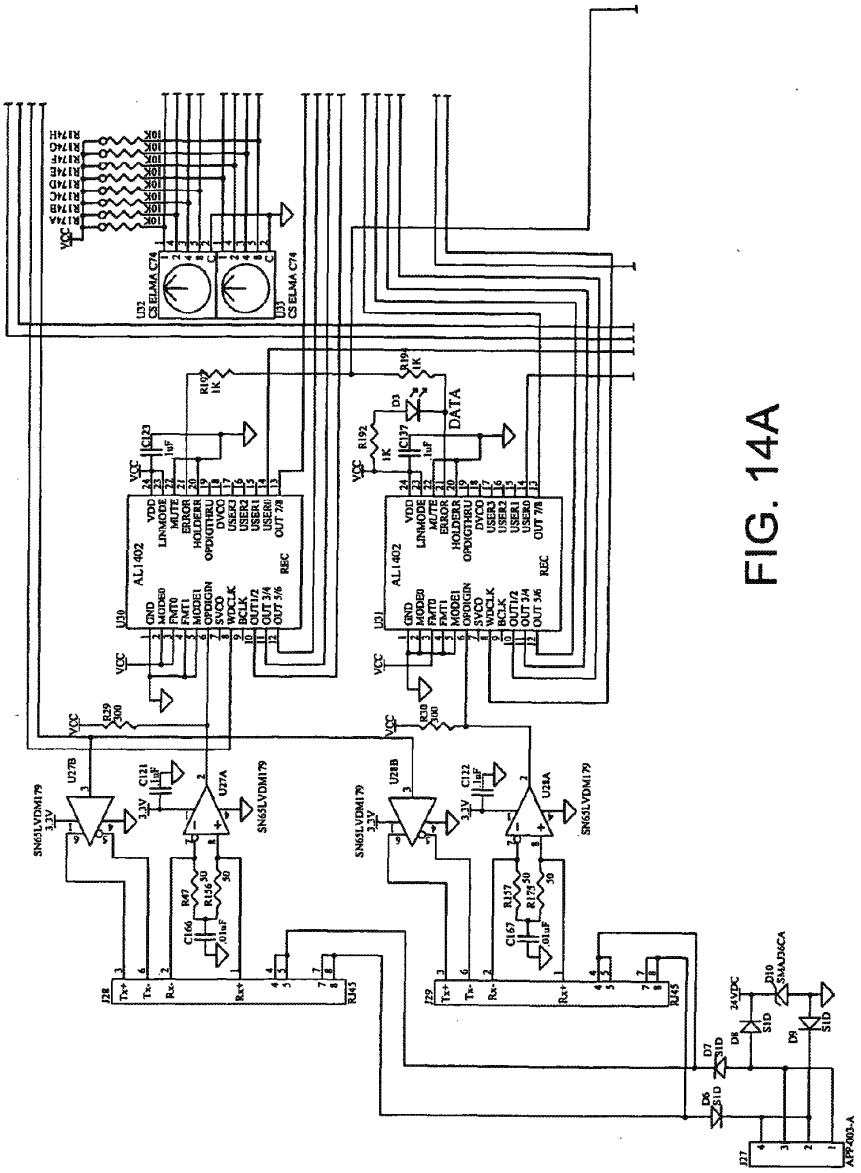


FIG. 14A

D+M-PA_0000727

LINKPLAY EXHIBIT 1004
Page 1924 of 4822

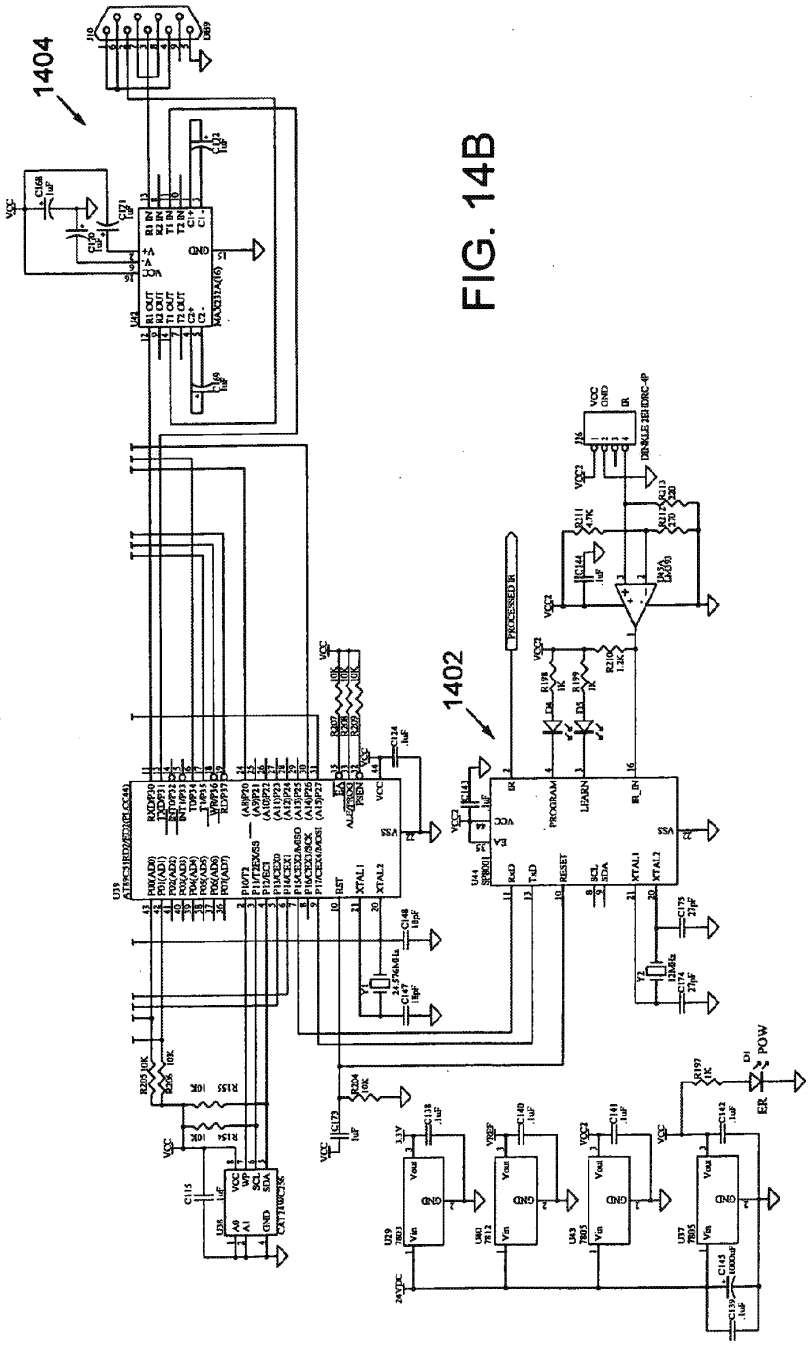


FIG. 14B

D+M-PA_0000728

LINKPLAY EXHIBIT 1004
Page 1925 of 4822

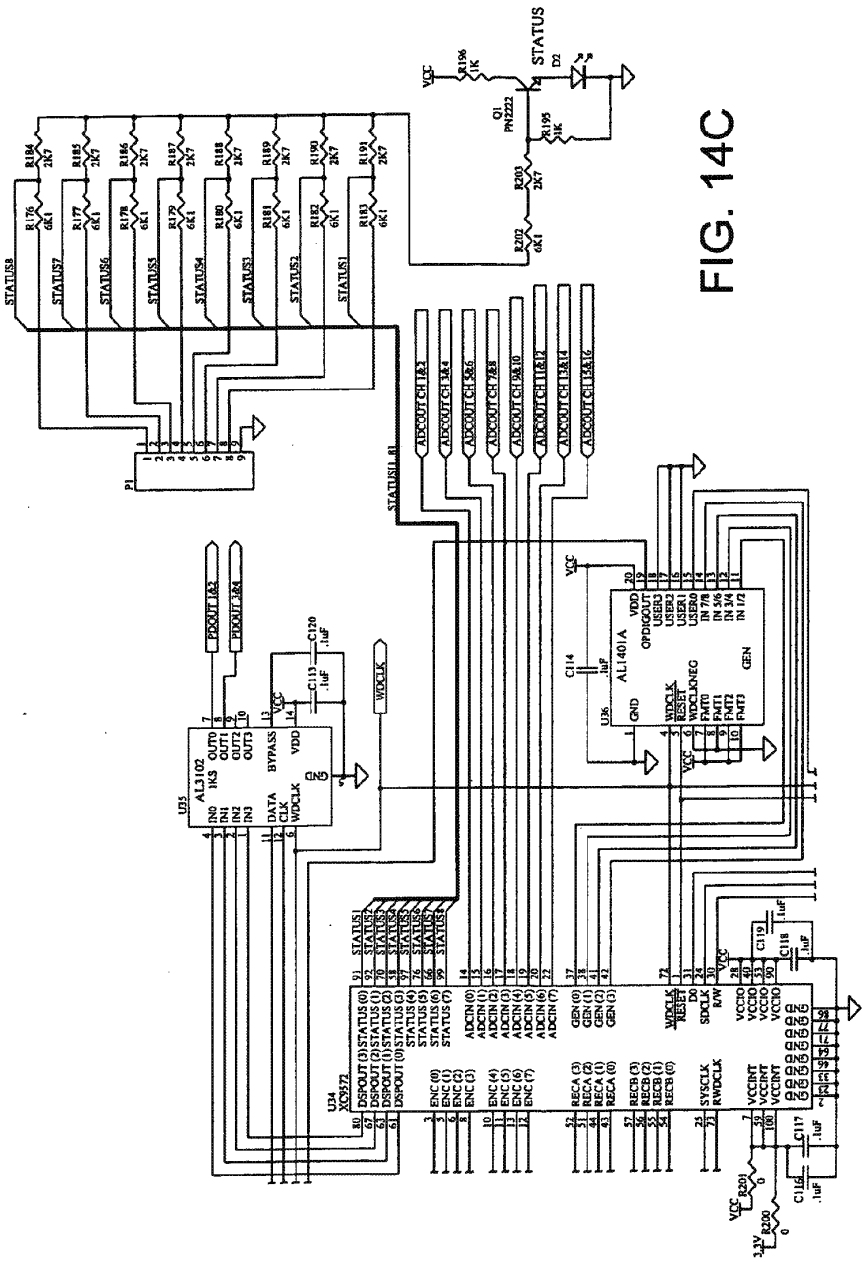


FIG. 14C

D+M-PA_0000729

LINKPLAY EXHIBIT 1004
Page 1926 of 4822

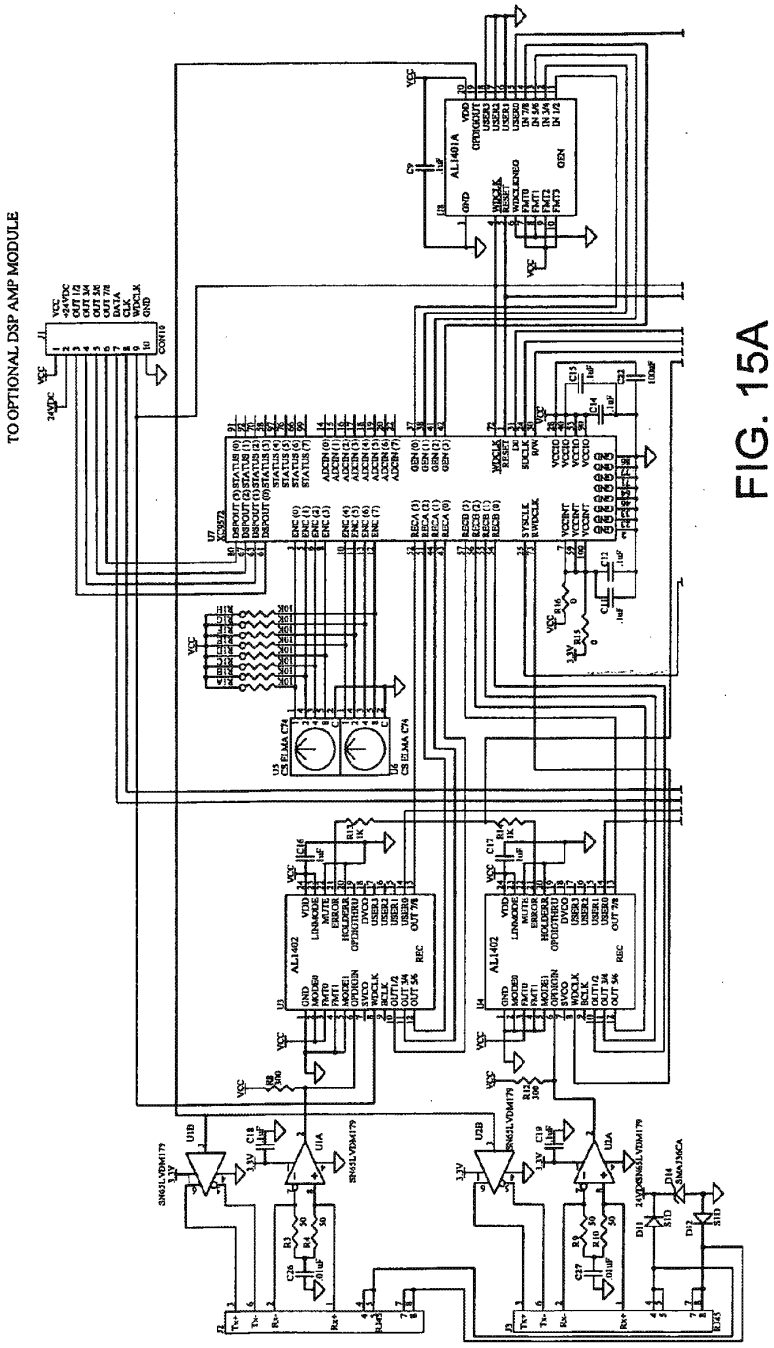


FIG. 15A

D+M-PA_0000730

LINKPLAY EXHIBIT 1004
Page 1927 of 4822

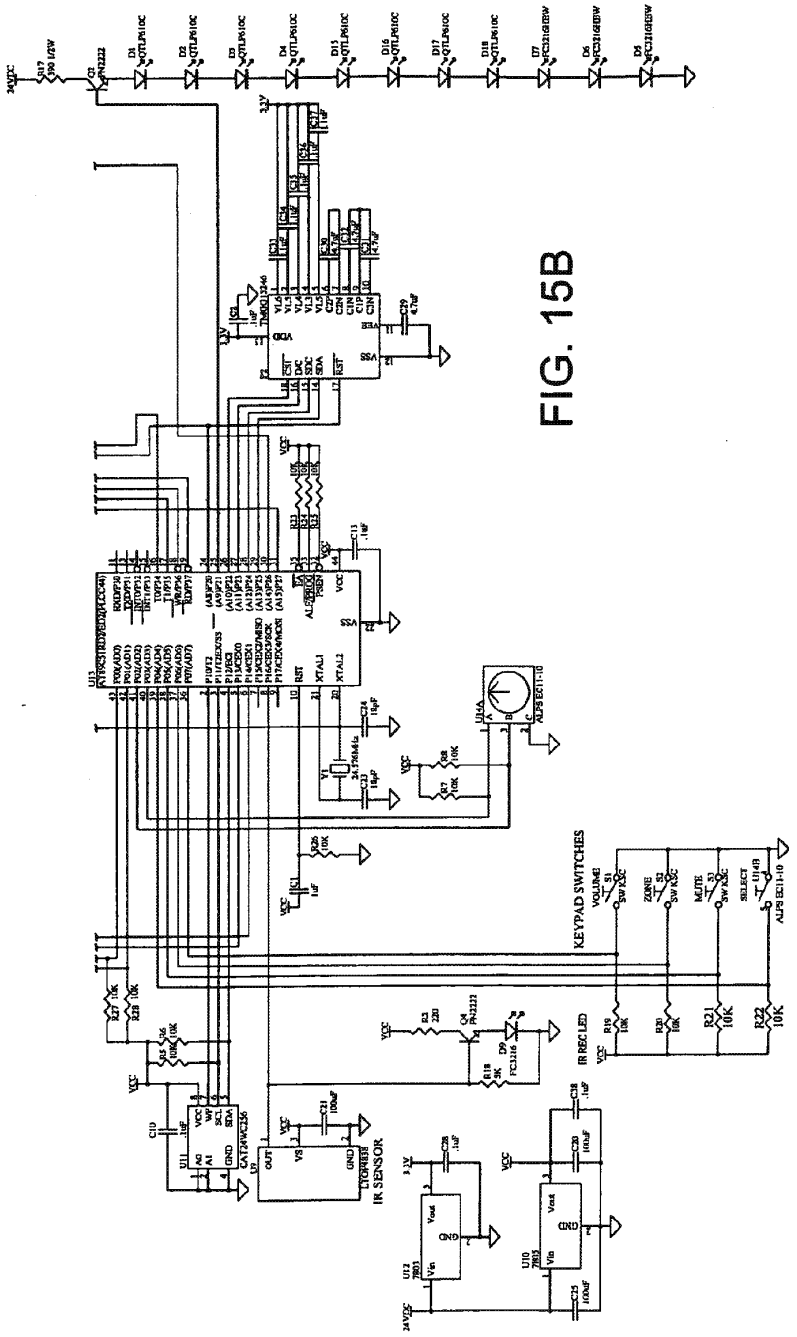


FIG. 15B

D+M-PA_0000731

LINKPLAY EXHIBIT 1004
Page 1928 of 4822

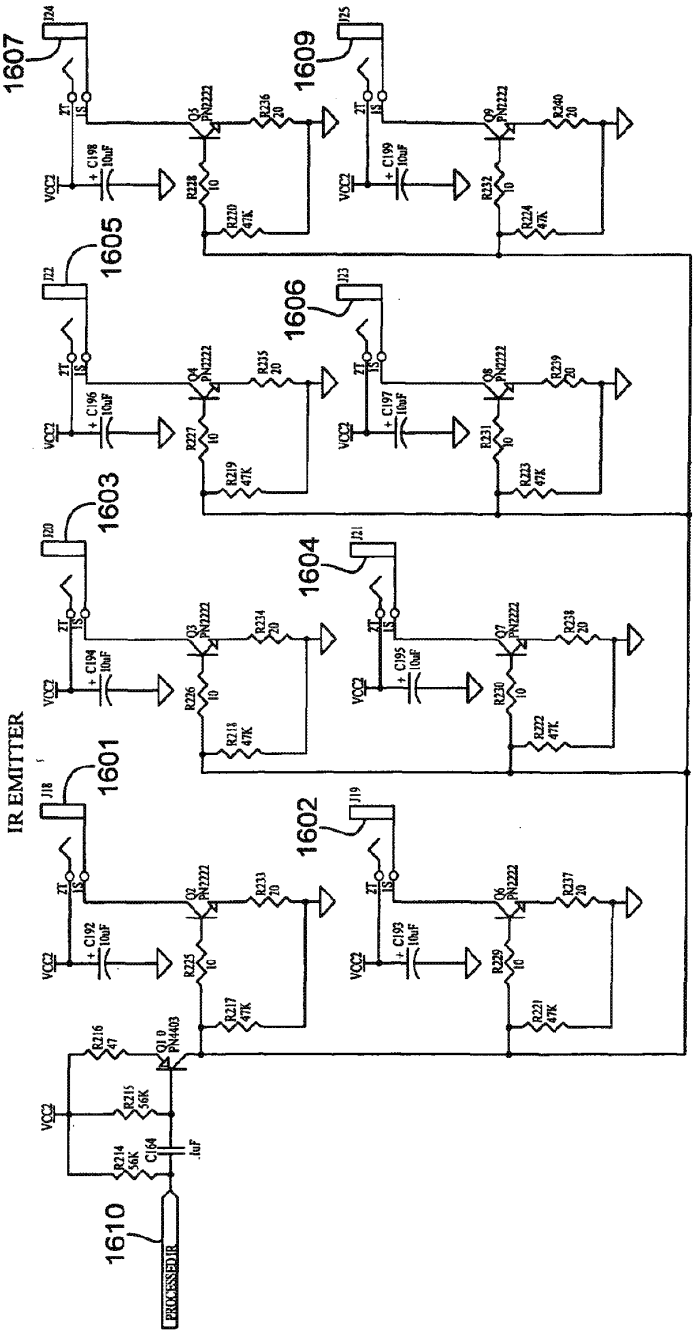


FIG. 16

D+M-PA_0000732

LINKPLAY EXHIBIT 1004
Page 1929 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 043 381 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.04.2009 Bulletin 2009/14

(51) Int Cl.:
H04R 1/02 *(2006.01)* *H04R 5/02* *(2006.01)*

(21) Application number: **08165831.2**

(22) Date of filing: **03.10.2008**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR
Designated Extension States:
AL BA MK RS

(30) Priority: **28.09.2007 DK 200701404**

(71) Applicant: **BANG & OLUFSEN A/S**
7600 Struer (DK)

(72) Inventors:
- **Nielsen, Kaj**
  7790 Thyholm (DK)
- **Jensen, Ole**
  7800 Skive (DK)

(74) Representative: **Tellefsen, Jens J. et al**
Patrade A/S,
Fredens Torv 3A
8000 Aarhus C (DK)

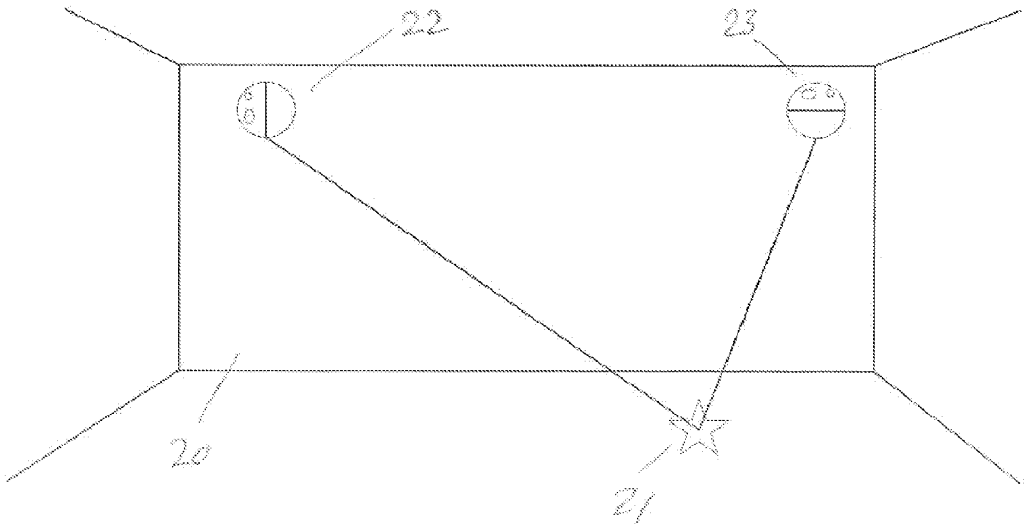
Remarks:
A request for re-establishment of rights in respect of the twelve-month period from the date of filing of the first application has been granted (Art.87(1) and Art. 122 EPC).

(54) A method and a system to adjust the acoustical performance of a loudspeaker

(57) The invention relates to a method and a system for optimizing the performance of a loudspeaker system as perceived by user.

The physical part of the speaker system is adjustable which enables the speaker units to be placed in positions that imply maximal acoustical performance. This is obtained by combined means to: 1) bring the speaker in a specified position and 2) apply accordingly position dependant individual acoustical filters inserted in the signal path to each individual speaker.

Figure 5



EP 2 043 381 A2

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

[0001] The present invention relates to improvements in sound experience as perceived by the user in a listening room. Specifically - but not exclusively the means to obtain the increased performance is by adjusting the speaker position according to the listening position of the user and accordingly adjust individual filter means in the signal path to the loudspeaker.

Background of the Invention

[0002] When a loudspeaker is placed within an enclosed space the timbre of the loudspeaker as perceived by a listener is highly affected by the acoustical properties of the space.
Traditionally built-in loudspeaker in walls, ceiling's and floors has a very poor performance due to the difficult radiation properties out from the speaker units.
[0003] The designer of the sound reproduction system usually wishes to give the listener the same intended listening experience regardless of the acoustical properties of the listening space.
[0004] Attempts of providing systems which addresses this desire has been suggested in the art, see for example JP58218294, JP58170195 and JP57116495.
From JP58218294 is known a system where the mid and high range speaker units are arranged in a convex rotatable member relative to the loudspeakers surface. By rotating the member, the sound emitting angle due to the convex shape of the member from the mid and high range speakers, relative to the speaker's surface may be altered, in order to improve the overall sound emission.
In JP58170195 is disclosed a system where the loudspeaker type may be changed by rotating a member. The mid and high range loudspeaker units are mounted in the rotatable member. Furthermore by rotating the member a bus reflex port may be opened or closed in the front surface of the loudspeaker, thereby changing the properties of the loudspeaker.
Finally in JP57116495, the tweeter speaker unit of a loudspeaker construction, has been mounted such that it may tilt relative to a woofer speaker. In this manner it is possible to adjust the tweeters emission relative to the woofers emission, simply by directing the tweeters sound emission in a different direction.
[0005] With the current invention the loudspeaker system may adapt dynamically to the listeners position in a room.
[0006] Thus the object of the invention is to provide a loudspeaker system concept with dynamic properties such that not only the sound properties are optimised but also the listeners position is taken into account such that it is possible to create an optimum listening experience for the listener regardless of the listeners position in the room.
[0007] This object is achieved by a speaker system according to independent claim 1.
[0008] In further embodiments according to the dependent claims further advantageous objects are achieved, for example by:

- Including one or more speaker units,
- the speaker units are mounted into a frame that is rotate able and tilt able,
- individual filters may be adapted according to the individual speaker positions,
- positioning the speakers according to the listener position may be user commanded and/or automatically controlled, and
- active loudspeakers are applied for efficiency.

Description of the Invention

[0009] In the following, preferred embodiments of the invention will be described with reference to the drawing wherein

Fig. 1	illustrates a top view of the invention with the speaker units in the default position.
Fig. 2 a & b	illustrates a top view of the invention with the speaker units in two different positions.
Fig. 2 c & d	illustrates a side view of the invention with the speaker units in two different positions.
Fig. 3	illustrates the coordinate systems related to rotation- and tilt positions.
Fig. 4	illustrates the signal path including filters, amplifiers and two speaker units.
Fig. 5	illustrates how to set of speaker systems adjust according to the listing position in a room.
Fig. 6	illustrates the control system of the invention.

[0010] In a first aspect, the invention relates to:
[0011] A speaker system, where the speaker units mounted in the speaker system may be configured into different

LINKPLAY EXHIBIT 1004
Page 1931 of 4822

positions to increase the acoustical performance perceived by the user, where said speaker system contains a mechanical structure consisting of four members:

- a first member(1), where said first member may have any geometrical shape, where said member has a first surface and defining a circular innermost periphery of said first member;
- a second member(2), where said second member may be arranged within the circular periphery provided on the first member(1), such that the second member may be rotate able around an axis perpendicular to, or close to being perpendicular to, the plane of the surface of the first member(1), and with the second member(2) fully or partly having a material surface;
- a third member(3) being an integrated part of the second member(2) or being a separate object mounted on top of the second member(2), and where said third member (3) includes one or more speaker units(5,6) mounted to deliver acoustical output directed away from the surface of the third member (3);
- a fourth member (4), where said fourth member is support structure including a hinge for the third member (3) and enables the third member(3) to be tilt able around an axis horizontal to, or close to being horizontal to, the plane of the surface of the second member(2) and with the fourth member (4) attached to the second member(2).

[0012] In a preferred embodiment the motorized means (31, 32, 33) control the position of the loudspeaker units (5, 6) in the speaker system. The motor means controls the rotating of the second member (2) and controls the tilting of the third member (3).

[0013] Standard means are applied to control the motors and to determine the position of each motor relative to a polar coordinate system for each of the movements, i.e. the rotation(8) of the second object(2) and the tilt(9) of the third object (3).

[0014] In one aspect the required position of the speakers are obtained by passing information of the X, Y coordinates related to the rotation of the second member to the second member motor means and by passing information of the X, Y coordinates related to the tilting of the third member to the third member motor means.

[0015] A predefined set of X, Y parameters for each of the motor means may be applied as the initial setting of the second- and third members upon start up of the speaker system. The same set of parameters may be applied as a default position that may be returned to during operation, thus to act as a Reset position for the loudspeakers.

[0016] In another aspect the rotation and the tilting of the loudspeaker units may be controlled by the user via a command from a wireless control device. Standard means (e.g. Infrared, Radio Frequency, Bluetooth or alike) may be applied for the communication from/to the remote terminal.
Specific X, Y coordinate values may be transferred to the speaker system and/or relative commands like: 'rotate left; rotate right; tilt up; tilt down; reset or similar commands.
Other functional related commands like 'bigger'; 'smaller'; 'light'; 'heavy' may be sent from a remote device for further mapping into specific X, Y parameter by the speaker system.

[0017] In the preferred embodiment this flexible addressing speaker units positions are that:

- the X, Y coordinates of the second member position may be forced to given values, and thus forced to a given position of the speaker, specified from an external control device, and
- the X, Y coordinates of the third member position may be forced to given values, and thus forced to a given position of the speaker, specified from an external control device.

[0018] In yet another aspect the position of the speaker units of one speaker system may automatically be oriented towards a detected location of a source signal in a room. This room location may be the listing position. Standard means are applied to generate the source signal; e.g. an infrared signal from a control device may be applied, and issued from the device upon a user command. The speaker system has a built in infrared receiver and detects the infrared beam. In a sequential flow of operation each of the motor means, i.e. the rotation motor means and the tilt motor means, are positioned from their respective min. to the max. positions in combinations to detect the maximum of the received IR signal. Alternatively to the IR signal and - detection method, an RF signal and - detection method or a sound signal and - detection method may be applied.

[0019] In the preferred embodiment these automated speaker units positions are where: the position of the second member and the position of the third member are determined from a detected maximum and or a detected minimum in a signal beam issued from a source located in the listening position of the user towards the speaker system.

[0020] In yet another aspect individual correction filters are applied in the signal path of every speaker according to the orientation/position of the speaker; this to maximize the sound quality as perceived by the listener. A predefined set of filter attributes are available for a set of defined X, Y positions related rotation and related to tilting of the loudspeaker units.

[0021] In the preferred embodiment these correction filter attributes may be used dynamically and consists of:

LINKPLAY EXHIBIT 1004
Page 1932 of 4822

- a predefined set of filter attributes are applied as the default correction filters to be in the signal path for each of the one or more speakers, and
- a predefined plurality of individual set of filter attributes are related to different speaker positions for each of the one or more speakers, and
- a predefined set of filter attributes are selected and applied as the actual correction filter to be in the signal path for each of the one or more speakers, where the set of filter attributes are selected based on the X,Y position of the speaker.
- a predefined plurality of individual set of filter attributes related to speaker positions for each of the one or more speakers may be replaced with data from an external source.

[0022] In the preferred embodiment these correction filter attributes are related to the X, Y position of the rotation means and to the X, Y position of the tilt means in a simple index table in a discrete manner:

[0023] If X, Y position is in the interval pos1 -> pos2 Then apply filter set1

Rotate position	Tilt position	Filter	Index
Rpos1 -> Rpos2	Tpos1 -> Tpos2	Fset1	1
	Tpos2 -> Tpos3	Fset2	2
	+++	+++	//
	TposN-> TposM	FsetN	N
Rpos2 -> Rpos3	Tpos1 -> Tpos2	Fset1.1	N+1
	Tpos2 -> Tpos3	Fset2.1	N+2
	+++	+++	//
	TposN-> TposM	FsetN.1	N+N
			//
Rpos3 -> Rpos4	Tpos1 -> Tpos2	Fset1.2	2N+1
	Tpos2 -> Tpos3	Fset2.2	2N+2
	+++	+++	//
	TposN-> TposM	FsetN.2	//
+++			
RposP -> RposQ	Tpos1 -> Tpos2	Fset1.p	//
	Tpos2 -> Tpos3	Fset2.p	//
	+++	+++	//
	TposN-> TposM	FsetN.p	//

[0024] The table above illustrates the mapping of positions into filter index:

- a predefined plurality of individual set of X, Y coordinates are related to the second member position and where a predefined plurality of individual set of other X, Y coordinates are related to the third member position and where an index is related to a set of corresponding filter attributes that applies in that specific speaker position, and
- a predefined plurality of individual set of X, Y coordinates are related to the second member position and where a predefined plurality of individual set of other X, Y coordinates are related to the third member position and where an index is related to a set of corresponding filter attributes that applies in that specific speaker position and where the complete set of definitions may be replaced with data from an external source.

[0025] In figure 1 the invention is displayed with the main elements:

LINKPLAY EXHIBIT 1004
Page 1933 of 4822

- The basic structure as a first element (1) may act as the fixture of other parts of the construction. The outer shape of this element is primarily determined by the requirement set up for the end user product, into which the invention will be a part of.
- A circular cavity in the first element is the room for the second (2) and the third (3) elements. Those elements are mounted so they have the freedom to rotate. According to requirements the allowed rotation may be whatever subset of the full rotation capability of 0 -> 360 deg. Preferable the axis of rotation is perpendicular to the surface of the first element(1).
- The third element (3) is attached to the rotational part of the construction (2) via a hinge (4) or alike, that allows the third member (3) to tilt. According to requirements the allowed tilting may be whatever subset of the full movement capability of 0 -> 90 deg. Preferable the axis of tilting is parallel to the surface of the second element (2).
- The loudspeaker units (5, 6) are mounted into the third element (3) such that the acoustical output is radiated away from the surface of the third element.
  One or more speaker units may be implemented according to product requirements. In a preferred embodiment two speakers are applied: one for high frequency sound radiation and one for medium frequency sound radiation.
- As part of the subsystem that detects a listening position (37) the detector of an infrared receiver (7) may be integrated into the surface of the third element. Alternative means may be implemented applying RF- or sound signal technologies.

[0026] Figure 2 illustrates different positions of the third element. The positions are examples of how the loudspeaker speaker units (5, 6) may be oriented towards a specific position in a listening room:

- In figure 2.a the second - and third element is rotated approximately - 45 deg. relative to the initial position in figure 1.
- In figure 2.b the second - and third element is rotated approximately - 90 deg. relative to the initial position in figure 1.
- In figure 2.c the second - and third element is rotated approximately - 90 deg. relative to the initial position in figure 1.
- In figure 2.d the second - and third element is rotated approximately - 90 deg., and the third element is tilted approximately - 45 deg. relative to the initial position in figure 1.

[0027] Figure 3 illustrates how ordinary X, Y coordinate systems defined the position of the speaker units (5, 6). A specific position may be defined by X, Y coordinates (8, 9) or an angle relative to the x-axis (8', 9'):

- Figure 3.a display an example of the rotation of the second - and third element into a position (8).
- Figure 3.b display an example of the tilting of the third element into a position (9).

[0028] Figure 4 illustrates how one of more speaker units is configured:

- Compensation filters (12, 12') are inserted into the signal path (11, 11'). The parameters determining the characteristics of a filter may be replaced dynamically according to an actual position of the speaker units. Individual filters are applied per speaker unit (14, 14').
- Individual amplifiers (13, 13') are applied per speaker unit (14, 14'). Active speakers may be applied. For efficiency reasons, i.e. to optimize physical size and power consumption versus output power performance, a technology like ICEpower from Bang & Olufsen may be used.

[0029] Typical applications of the speaker system as outlined may be into a stand alone loudspeaker or to be applied as an in-wall loudspeaker, or as an in-ceiling loudspeaker or as an in-floor loudspeaker, or as a loudspeaker in a home appliance and in media systems.

[0030] In figure 5 it is illustrated how two in-wall speaker systems (22, 23) are located in a room (20). One of the loudspeaker systems (22) is vertically aligned towards the listening position (21) and the other of the loudspeaker systems (23) is horizontally aligned towards the listening position (21).

[0031] In a room with a plurality of loudspeaker systems according to the invention, the filters to be applied dynamically per loud speaker unit in all systems may be configured according to different methods based on:

- Positions commanded externally from a user remote control device (36).
- A detected listener position, as deduced from an incoming source signal beam, e.g. an infrared signal (37).
- A detected position of other loudspeakers in the room, as deduced from an incoming sound signal received by the actual speaker system, e.g. a sound signal issued by another speaker unit in the room and received via a microphone (34') by the actual speaker system.

[0032] Figure 6 display the controller (30) of the loudspeaker system of one embodiment according to the invention:

LINKPLAY EXHIBIT 1004
Page 1934 of 4822

- The sound signal (40, 11) is via the filters (12, 12') and the amplifiers connected to the loud speaker units (14, 14').
- The dynamic load and reload of the filter attributes is supported by data sourced in a table (39).
- The support of the filter management that is related to actual X, Y positions of the speaker units is supported by data sourced in a table (38).
- Archived filter attributes may be updated or modified by a reload af new data from an external source (35). The load may via a wired data link or via a wireless data link and sourced e.g. from a PC.
- A subsystem (37) supports the detection of the listening position by detecting the maximum of an incoming source signal e.g. infra red light beam. The system detects the maximum by analyzing different combinations of the rotation position and the tilt position.
- A subsystem (34) supports the detection of position of other loudspeakers in the room, as deduced from an incoming sound signal received via a microphone (34') by the actual speaker system.
- A subsystem (31) controls the motor that supports the rotational movement (33) and the tilt movement (33'). Standard means are applied for the control and the position of each of the motors, one to detect the X, Y position of the rotation (32) and one to detect the X,Y position of the tilt (32').

Claims

1. A speaker system, where the speaker units mounted in the speaker system may be configured into different positions to increase the acoustical performance perceived by the user, where said speaker system contains a mechanical structure consisting of four members:

   - a first member, where said first member may have any geometrical shape, where said member has a first surface and defining a circular innermost periphery of said first member;
   - a second member, where said second member may be arranged within the circular periphery provided on the first member, such that the second member may be rotate able around an axis perpendicular to, or close to being perpendicular to, the plane of the surface of the first member, and with the second member fully or partly having a material surface;
   - a third member being an integrated part of the second member or being a separate object mounted on top of the second member, and where said third member includes one or more speaker units mounted to deliver acoustical output directed away from the surface of the third member;
   - a fourth member, where said fourth member is support structure including a hinge for the third member and enables the third member to be tilt able around an axis horizontal to, or close to being horizontal to, the plane of the surface of the second member(2) and with the fourth member (4) attached to the second member(2).

2. A speaker system according to claim 1 wherein motor means controls the rotating of the second member and controls the tilting of the third member.

3. A speaker system according to claim 2 wherein the position of the speakers are obtained by passing information of X, Y coordinates related to the rotation of the second member to the second member motor means and by passing information of X, Y coordinates related to the tilting of the third member to the third member motor means.

4. A speaker system according to claim 3, wherein a predefined set of filter attributes are applied as the default correction filters to be in the signal path for each of the one or more speakers.

5. A speaker system according to claim 4, wherein one of the predefined set of X, Y coordinates defines the initial position of the second member.

6. A speaker system according to claim 4, wherein one of the predefined set of X, Y coordinates defines the initial position of the third member.

7. A speaker system according to claim 6, wherein the X, Y coordinates of the second member position may be forced to given values, and thus forced to a given position of the speaker, specified from an external control device.

8. A speaker system according to claim 6, wherein the X, Y coordinates of the third member position may be forced to given values, and thus forced to a given position of the speaker, specified from an external control device.

9. A speaker system according to claim 8, wherein a predefined plurality of individual set of X, Y coordinates are related

LINKPLAY EXHIBIT 1004
Page 1935 of 4822

to the second member position and where a predefined plurality of individual set of other X, Y coordinates are related to the third member position and where an index is related to a set of corresponding filter attributes that applies in that specific speaker position.

10. A speaker system according to claim 9, wherein a predefined plurality of individual set of X, Y coordinates are related to the second member position and where a predefined plurality of individual set of other X, Y coordinates are related to the third member position and where an index is related to a set of corresponding filter attributes that applies in that specific speaker position and where the complete set of definitions may be replaced with data from an external source.

11. A speaker system according to claim 10, wherein the position of the second member and the position of the third member are determined from a detected maximum and or a detected minimum in a signal beam issued from a source located in the listening position of the user towards the speaker system.

12. A speaker system according to one or more of the claims above, wherein a predefined plurality of individual set of filter attributes are related to different speaker positions for each of the one or more speakers.

13. A speaker system according to one or more of the claims above, wherein a predefined set of filter attributes are selected and applied as the actual correction filter to be in the signal path for each of the one or more speakers, where the set of filter attributes are selected based on the X,Y position of the speaker.

14. A speaker system according to one or more of the claims above, wherein a predefined plurality of individual set of filter attributes related to speaker positions for each of the one or more speakers may be replaced with data from an external source.

15. Use of the speaker system according to any of the claims 1 to 14, implemented as a stand alone loudspeaker.

16. Use of the speaker system according to any of the claims 1 to 14, implemented as a built in loudspeaker, to be applied as in-wall loudspeaker, and/or as in-ceiling loudspeaker and/or as in-floor loudspeaker.

17. Use of the speaker system according to any of the claims 1 to 14, implemented as a loudspeaker in a home appliance and/or in media systems.

LINKPLAY EXHIBIT 1004
Page 1936 of 4822

Figure 1

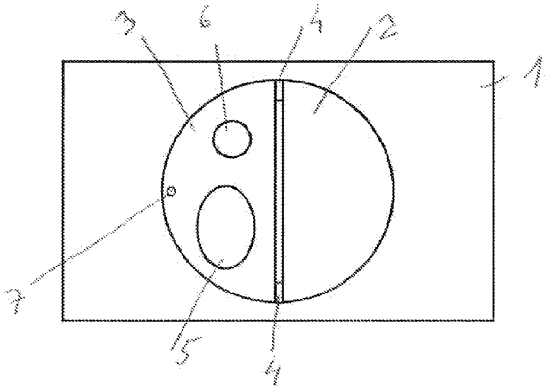


Figure 2

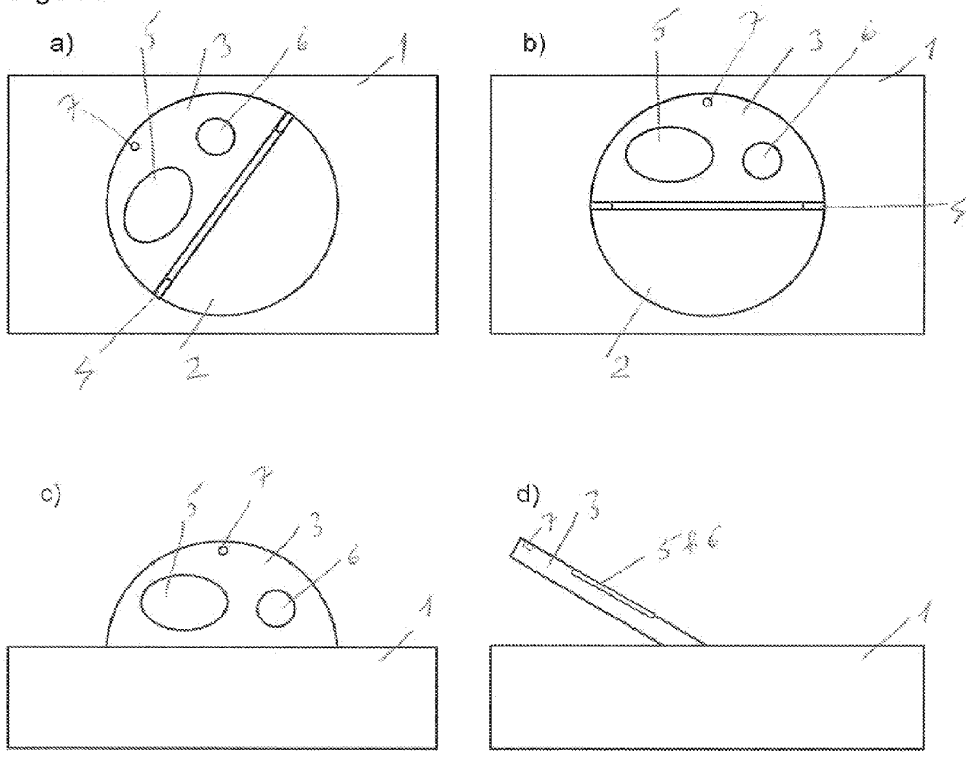


Figure 3

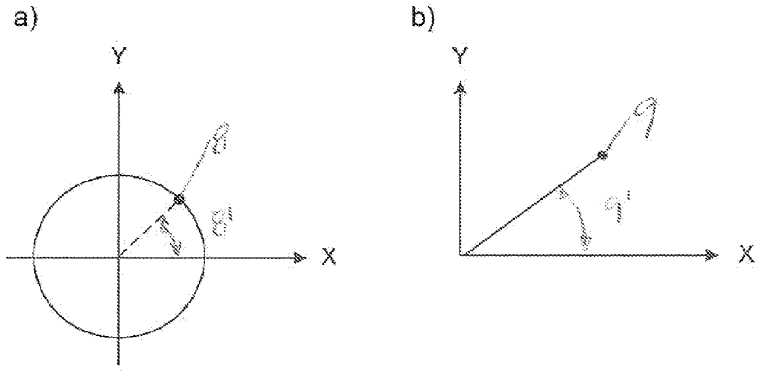


Figure 4

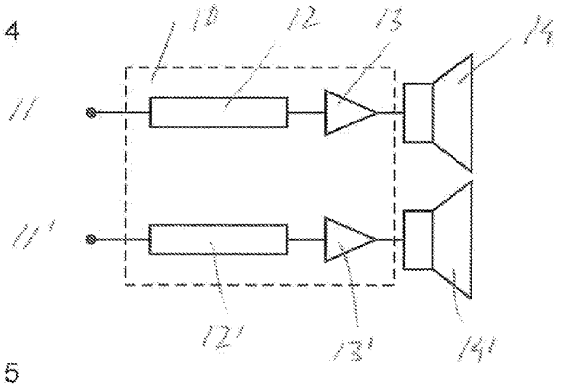
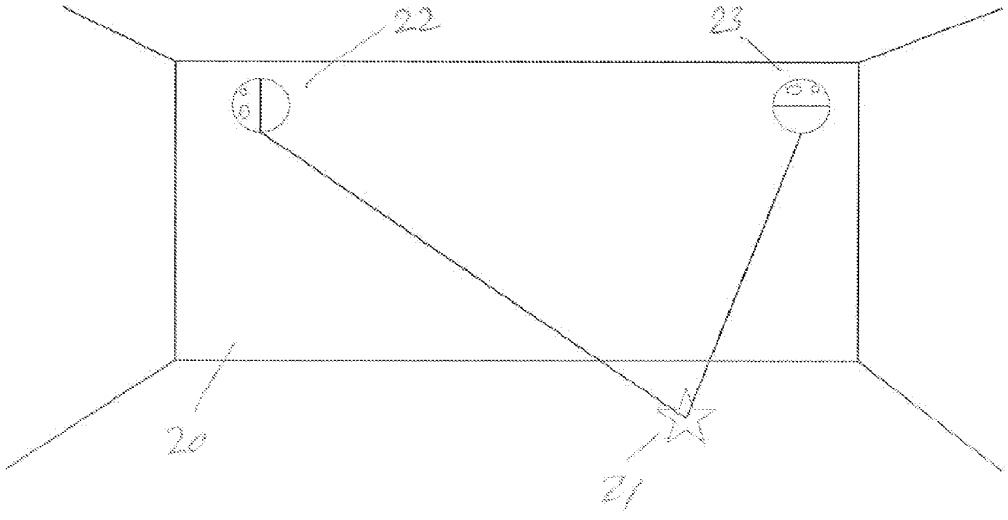
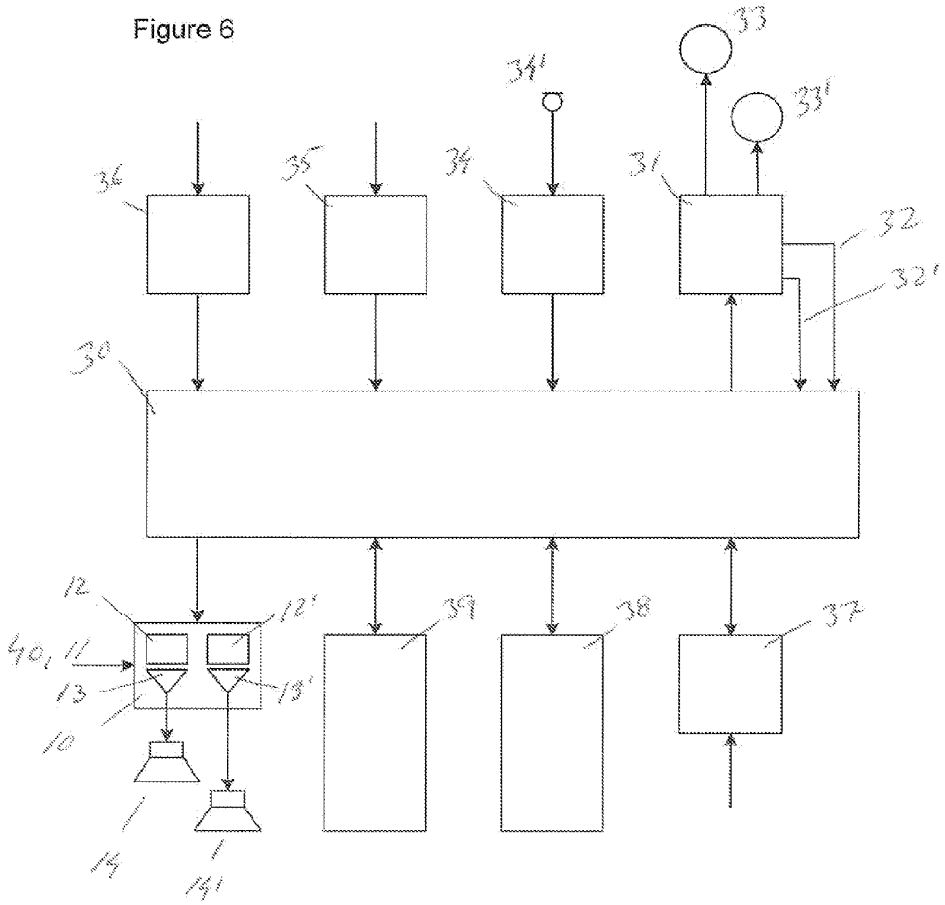


Figure 5



LINKPLAY EXHIBIT 1004
Page 1938 of 4822

Figure 6



REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- JP 58218294 B **[0004] [0004]**
- JP 58170195 B **[0004] [0004]**
- JP 57116495 B **[0004] [0004]**

LINKPLAY EXHIBIT 1004
Page 1940 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 161 950 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
10.03.2010 Bulletin 2010/10

(51) Int Cl.:
H04S 3/00 *(2006.01)*

(21) Application number: **09169687.2**

(22) Date of filing: **08.09.2009**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO SE SI SK SM TR
Designated Extension States:
AL BA RS

(30) Priority: **08.09.2008 DK 200801258**
11.03.2009 DK 200900336
22.09.2008 DK 200801317

(71) Applicant: **BANG & OLUFSEN A/S**
7600 Struer (DK)

(72) Inventors:
- **Fredsgaard, Lars**
  7600, Struer (DK)
- **Lydolf, Morten**
  7500, Holstebro (DK)

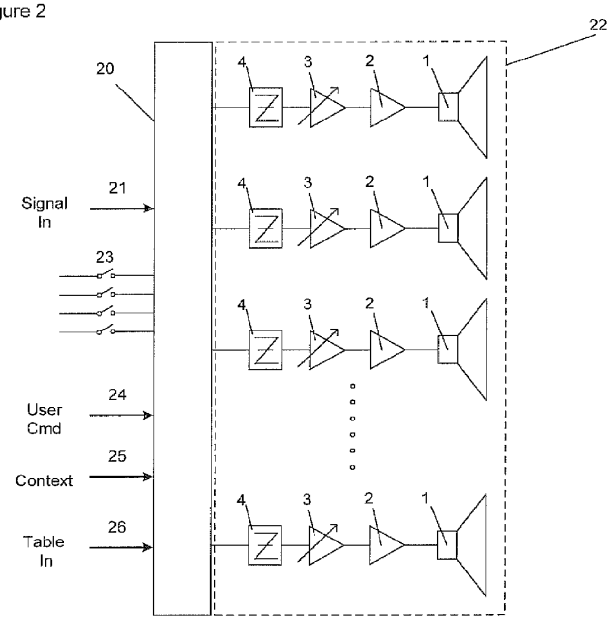
(74) Representative: **Tellefsen, Jens J.**
Patrade A/S,
Fredens Torv 3A
8000 Aarhus C (DK)

(54) Configuring a sound field

(57) The present invention relates to a method and system for dynamic reconfiguring an audio reproduction system comprising two or more sound channels. The premises of the reconfiguring are being based on the position of one or more listeners and the one or more sound channels in the reproduction system. The feature obtained is a sound distribution into individual zones of sound fields.

The reconfiguring means are based on a digital signal processor that configures the sound channels by adjusting the equalization, the delay and the gain according to individual listener's position in a room/space.

Figure 2



Printed by Jouve, 75001 PARIS (FR)

EP 2 161 950 A2

Description

Field of the Invention

[0001] The present invention relates to a method and system for dynamic reconfiguring an audio reproduction system comprising two or more sound channels. The premises of the reconfiguring are being based on the position of one or more listeners and the one or more sound channels in the reproduction system. The feature obtained is a sound distribution into individual zones of sound fields.

[0002] The invention includes digital signal processing, sound transducers, filters and amplifier configurations to be applied in surround sound systems and traditional stereophonic system, and to be applied in any domain like a home, in a vehicle, in a boat, in an airplane, in an office or in any other private or public domain, this being airports, shopping centres, sports arenas and theatres, i.e. in a PA system.

Background of the Invention

[0003] It is a well known problem in prior art loudspeaker systems operating in closed rooms/spaces e.g. a car, that the sound expired by the user may vary according to the listeners position in a space relative to the loudspeaker system transducers.

[0004] Furthermore systems have been developed for simplistic sound distribution, for example for use inside rooms or cars. The systems detect the configuration of the space and the presence of listeners, and reconfigure the sound output according to the number of listeners and their position relative to the loudspeakers. Such systems are disclosed in US2007/0211908, US2004/0240676 and DE102004022379.

Object of the Invention

[0005] Thus to obtain a certain perceived quality level of a loudspeaker system, e.g. in a car, the individual sound channels incorporating one or more loudspeaker modules must be calibrated and adjusted individually and according to the number of persons in the car, and their position in the space e.g. their seated positions.

[0006] This principle may be applied in any type of room like airplanes, boats, theatres, arenas, and shopping centres and alike.

[0007] In the public domain applications it would be a huge advantage to direct the sound fields primarily into zones where people are located and /or seated, this to avoid acoustical reflections from distributed speakers in areas where no people are located.

Description of the Invention

[0008] The invention discloses a surround sound system where different sound settings may be enabled via digital signal processing means controlling the individual sound channels parameters e.g. the equalization (filters), the delay, the gain (amplification). The premise for the control is based upon the listener's position in the listening room/space.

[0009] In a more advanced embodiment of the invention the control and sound setting may work in a contextual mode of operation to accommodate for:

- the position of the one or more listeners;
- a functional mode of operation, e.g. adjust the sound system settings to be in movie mode;
- the type of music, e.g. rock;
- the time of the day; e.g. morning/day/night/Christmas etc.

[0010] The invention discloses a solution that in an automatically mode of operation may fulfil the stated requirements and solve the related technical problems.

[0011] In summary the invention includes:

- sensor means to detect a listener position;
- a mode of operation related to a user position or a required function;
- Information available according to context (place, time and music content);
- multi channel sound system, e.g. two channels stereo, a 5.1 surround sound system;
- digital controlled sound system e,g. digital control of gain, equalization and delay;
- active speakers including amplifiers and filters for each loudspeaker transducer.

LINKPLAY EXHIBIT 1004
Page 1942 of 4822

[0012] The digital control of the sound system is based on standard means for:

- gain: adjust the signal by a certain level, i.e. + /- xx dB, e.g. +0.1 dB;
- delay: the signal is delayed by a specific time, i.e. yy ms, e.g. 100 ms;
- EQ: the signal is filtered according to the Finite Impulse Response principle (FIR) or the signal is filtered according to the Infinite Impulse Response principle (iiR); a number of coefficients are specified; the number of parameters typically to be from 1 -> 1000.

[0013] The invention may be implemented in different embodiments, in which the alternative reconfiguring procedures may be implemented as:

- a traditional algorithm in terms of a sequential software program in which the values of the resulting adjustment parameters are embedded in the code itself; the algorithm validates according to a specified system structure of an actual loudspeaker configuration;
- a table based concept in which one or more tables defines the attributes to be applied per loudspeaker channel versus the mode of operation, the listener position and other context related information (e.g. time).

[0014] In a first aspect of the invention a method for automatic reconfiguring an audio reproduction system comprising two or more sound channels where the reconfiguring has the steps:

- determine the physical position of one or more listeners,
- apply the physical position as information to select a set of predefined or calculated (in situ) sound parameters, the set of parameters to include settings for equalization, gain and delay per sound channel,
- optionally apply other context related information to select a set of predefined sound parameters,
- optionally apply other user commanded information to select a set of predefined sound parameters, and
- provide the set of parameters per sound channel accordingly.

[0015] The fundamental goal of the invention is to enhance the perceived sound quality seen from a listener prospective. Regardless of the physical position of the loudspeakers, a virtual sound field may be generated and directed so it appears to be anywhere the listener may desire. This is obtained by adjusting the gain, the delay and the equalization parameters of the individual sound channels in a multichannel sound system.

[0016] In another aspect the reconfiguring process validates the position of the one or more users, and/or functional premises and/or context related premises and or content related premises as part of the reconfiguring of the one or more sound channels that constitutes the configured loudspeaker system, and where the validation prioritizes the position of one or more listeners.

[0017] According to specific requirements of a sound providing system miscellaneous features may be offered and controlled automatically and/or commanded by the user. The features may be used individually and/or mixed in a way that accommodates the user's demands. A predefined combination of sound settings per channel and the functional options supports the processing. Examples of functional options are:

- seat priority: driver, all, back, front, centre, left, right etc.;
- a persons physical position: in a room, in a car, in a PA domain etc.;
- zone priority: a zone includes one or more specific sound channels;
- mode priority: movie, music, speech;
- channel mode: stereo, surround sound 5.1, or other multi channel system of any order;
- group of people: one or more listeners in a zone;
- the time of an event.

[0018] The reconfiguring process is provided as one or more algorithms to provide the calculation of each of the values of the sound channel adjustment variables.

[0019] In a preferred embodiment the reconfiguring process is provided as a table with relations provided to be accessed by the digital signal processor to provide each of the values of the sound channel adjustment with variables.
The saved attributes and key parameters are loaded into the reconfiguring means supported by electronic means, means that is connected wirelessly or connected by wire to the audio reproduction system.

[0020] In a third aspect of the invention the reconfiguring process provides the settings of the sound parameters; gain, equalization and delay for one sound channel, to be applied in one sound field zone related to the physical position of first group of people including one or more listeners.
To accommodate for the handling of more zones including different groups of people the reconfiguring process provides

LINKPLAY EXHIBIT 1004
Page 1943 of 4822

the settings of the sound parameters; gain, equalization and delay for one sound channel, to be applied in one sound field zone related to the physical position of one or more other groups of people including one or more listeners.

[0021] In yet another aspect a listener position is detected via standard sensor means like switches or infrared detectors or strain gauges or temperature sensitive detectors.

[0022] The configuring process is to be automatically executed and/or controlled by the audio amplifier means, including a digital signal processor, which drives the loudspeaker system. The reconfiguring means is embedded into an audio reproduction system.

[0023] In a preferred embodiment, the reconfiguring is controlled via a table mapping the mode of operation to adjustment parameters for each speaker in every channel or just relevant channels. The adjustment parameters are e.g. but not limited to: equalization, delay and gain.

[0024] The table may be represented as one or more data set as most appropriate to the digital controller unit. E.g. one data set may contain the relations among:

- listener position,
- loud speaker channel #,
- parameter settings (EQ, delay, and gain).

[0025] Another data set may contain the functional/mode related information like:

- functional settings (movie, audio only),
- loud speaker channel #,
- parameter settings (EQ, delay, and gain).

[0026] This table concept is considered to be a data driven control systems, and enables for easy updates of the functional behaviour of a specific system simply by loading alternative data set into the controller.

[0027] In addition, the invention includes a constraint Solver, which comprises a table with digital data representing the constraints of the passenger(s) position(s) and related acoustical adjustment, attributes and corresponding variable values.

The constraint solver processing enables an arbitrary access mode to information with no order of sequences required. This is an important advantage over prior art tree-like programming structure, where the starting point is given by the root of the tree. Especially, when finding ways from nodes within the tree-like program to a specific target location in the tree, it may be obscure with regard to which way to chose, whether backwards or forwards, and the final solution may be a new start from the root.

[0028] According to the invention, the product configuration domain table is organized as relations among variables in the general mathematical notation of 'Disjunctive Form':

AttribVariable 1.1 and AttribVariable 1.2 and AttribVariable 1.3 and Attrib Variable 1.n
Or AttribVariable 2.1 and AttribVariable 2.2 and AttribVariable 2.3 and Attrib-Variable 2.n
Or ....
Or ....
Or AttribVariable m.1 and AttribVariable m.2 and AttribVariable m.3 and AttribVariable m.n

[0029] For example, AttribVariable 1.1 may be defining a passenger seat position, AttribVariable 1.1 a speaker transducer unit, AttribVariable 1.3 a speaker system/subsystem and AttribVariable 1.n gain value for the transducer unit. In another example, AttribVariable 2.n may be a reference to another table.

[0030] An alternatively definition term is the 'Conjunctive Form':

AttribVariable 1.1 or AttribVariable 1.2 or AttribVariable 1.3 or AttribVariable 1.n
And AttribVariable 2.1 or AttribVariable 2.2 or AttribVariable 2.3 or AttribVariable 2.n
And ....
And ....
And AttribVariable m.1 or AttribVariable m.2 or AttribVariable m.3 or AttribVariable m.n

[0031] With this method of defining the problem domain, it becomes a multi-dimensional state space enabling equal and direct access to any point in the defined set of solutions. The term multidimensional has to be understood as a contrast to a tree-like programming structure, which is two-dimensional.

[0032] According to the invention, the product configuration function proceeds by finding the result of the interrogation in the set of allowed and possible combinations in one or more Configuration Constraint Tables. According to definitions

LINKPLAY EXHIBIT 1004
Page 1944 of 4822

made in the configuration constraint tables, the result might be:

- a list of variables useful in the application e.g. gain, and filter setting i.e. equalization, for the one or more transducers;
- a list of variables useful in the application e.g. delay for the one or more transducers;
- a list of variables useful in the application to configure individual sound domains to related to zone of sound of targeted to one or more users.

[0033] Alternative configurations may be evaluated by the constraint solver, the alternatives being one or more of the defined set of legal combination in the constraint table.
The example illustrates alternative solutions for x, y when y is given:
[0034] The: If x Then y may be represented as x -> y
[0035] With the truth table:

x	y	x->y
0	0	1
0	1	1
1	0	0
1	1	1

Given the premise x is '1' the conclusion y is '1'.
Given the conclusion y is '1' the premises for x may be '1' or '0'.

Thus, when y is '1' the alternative for x are '1' or '0'.
[0036] Represented in a constraint table the x -> y has the legal combinations of x, y:

x	y
0	0
0	1
1	1

[0037] Table entries in a constraint table may be combined into legal/illegal combination such that all the well known logical operators known from the Boolean algebra will be included as required. The logical operators being: AND, OR, NOT, XOR, Logical Implication (->), Logical Bi-implication (=).
[0038] An example of a table definition is displayed below:

Mode or position	Speaker channel #	Other (e.g. time, content)	Equalization EQ	Delay	Gain
Driver	1		p-val-d1	q-val-d1	r-val-d1
	2		p-val-d2	q-val-d2	r-val-d2
	.....		.....	.....	.....
	n		p-val-dn	q-val-dn	r-val-dn
Front	1		p-val-f1	q-val-f1	r-val-f1
	2		p-val-f2	q-val-f2	r-val-f2
	.....		.....	.....	.....
	n		p-val-fn	q-val-fn	r-val-fn
Rear	1		p-val-r1	q-val-r1	r-val-r1
	2		p-val-r2	q-val-r2	r-val-r2

LINKPLAY EXHIBIT 1004
Page 1945 of 4822

(continued)

Mode or position	Speaker channel #	Other (e.g. time, content)	Equalization EQ	Delay	Gain
	.....		.....	.....	.....
	n		p-val-rn	q-val-rn	r-val-rn
All	1		p-val-a1	q-val-a1	r-val-a1
	2		p-val-a2	q-val-a2	r-val-a2
	....		.....	.....	.....
	n		p-val-an	q-val-an	r-val-an
Movie	1		p-val-m1	q-val-m1	r-val-m1
	2		p-val-m2	q-val-m2	r-val-m2
	.....		.....	.....	.....
	n		p-val-mn	q-val-mn	r-val-mn
.....	.....				
Other	1		p-val-o1	q-val-o1	r-val-o1
	2		p-val-o2	q-val-o2	r-val-o2
	.....		.....	.....	.....
	n		p-val-on	q-val-on	r-val-on

[0039] Thus in an aspect of the invention the reconfiguring means is embedded into an audio reproduction system. Thus the system becomes the controller of the re- configuring process that provides a reconfiguration of the system itself.

Description of the Drawing

[0040] In the following, preferred embodiments of the invention will be described with reference to the drawings wherein:

Figure 1 displays components that constitute examples of loudspeaker module configurations, that represent one sound channel configuration;
Figure 2 displays an example of an embodiment of the invention displaying a plurality of sound channels;

[0041] Figure 1 display components that constitute examples of loudspeaker module configurations related to one sound channel.
A loudspeaker transducer unit (1) is the fundamental means that transforms the electrical signal to the sound waves produced by movements of the membrane of the transducer unit. Any standard transducer unit may be applied in the invention.
The characteristics of each loudspeaker transducer unit are determined by measurement and/or from specifications delivered by the supplier of the unit.
[0042] In the preferred embodiment an audio reproduction system comprising active sound transducers including an amplifier (2) is provided for each transducer unit (1). This type of amplifier is e.g. the technology ICEpower from Bang & Olufsen DK.
[0043] In a high quality audio reproduction system a dedicated filter means, an equalizer, (3) is provided per amplifier (2). The means (3) provides a frequency dependent amplification to control the overall gain, which may be regulated up or down as required. Means for down regulation may be as simple as adjustment of a resistive means serial connected to the loudspeaker module.
[0044] To control the sound distribution into individual zones of sound fields the sound delay among channels must be controlled. In a preferred embodiment the delay (4) is controlled individually per sound channel.
[0045] Figure 2 displays the example of an audio reproduction system comprising loudspeaker modules configured

LINKPLAY EXHIBIT 1004
Page 1946 of 4822

in a multi channel concept e.g. a 5.1 surround sound system.
In fact any number of channels (22) may be handled; having a corresponding table complex related according to the actual functional system requirements.

[0046] The audio reproduction system may be an audio amplifier (20), including a digital signal processor, for controlling and driving the loudspeaker channels (22), and to provide a signal (21) that's amplified and filtered and delayed accordingly.

[0047] Input from sensor means (23) determines the mode and functional operation of the control system and may be of a different nature e.g. but not limited to: simple switch, accelerometer, infrared detection, thermal sensing etc.
In the preferred embodiment the sensors primarily detects the position of a user in a room/space, the position to be related to the sound field that's relevant for the one or more users /listeners.
E.g. in a vehicle the detection of a user's position is where the individual user is seated. The detection in this system may be via a simple switch in every seat, or via switches related to the security belts.

[0048] Additional control signals may be received as user commands (24), e.g. a command to 'enable automatic sound control' and a context related information (25), e.g. information about the time of the day and/or the type of music actual in play. The additional control signal information to be used as input into the digital signal processor to control mode and functional operation of the sound system.

[0049] In a preferred embodiment the reconfigure means is imbedded into the controller of the audio sound system. Thus the audio sound system is the master having the digital signal processor means (20) that initiates, controls and applies the reconfiguring process. Alternative table definitions may be loaded into the digital signal processor (20) from external means e.g. a laptop/PC (26).

[0050] The invention may be applied in relation to automatically adjusting an audio sound system to a maximum level of quality taking into account the position of individual listeners in dedicated zone of sound fields.

[0051] The invention includes sound channel configurations to be applied in surround sound systems and traditional stereophonic system, and to be applied in any domain like a home, in a vehicle, in an airplane in a boat, in an office or in any other public domain and alike.

Claims

1. A method for dynamic and automatic reconfiguring an audio reproduction system comprising two or more sound channels where the reconfiguring has the steps:

   - determine the physical position of one or more listeners,
   - apply the physical position as information to select a set of predefined or calculated sound parameters, the set of parameters to include settings for equalization, gain and delay per sound channel and/or per loudspeaker transducer,
   - optionally apply other context related information to select a set of predefined sound parameters,
   - optionally apply other user commanded information to select a set of predefined sound parameters, and
   - provide the set of parameters per sound channel and/or per loudspeaker transducer accordingly,

   and where the reconfiguring method is **characterized by** a constraint solver, which constraint solver finds one or more legal combinations among defined legal combinations predefined in a constraint table.

2. A method according to claim 1, where the reconfiguring process validates the position of the one or more users, and/or functional premises and/or context related premises and/or content related premises as part of the reconfiguring of the one or more sound channels that constitutes the configured loudspeaker system, and where the validation prioritizes the position of the one or more listeners.

3. A method according to any preceding claim where the reconfiguring process provides the settings of the sound parameters; gain, equalization and delay for one sound channel, to be applied in one sound field zone related to the physical position of a first group of people including one or more listeners.

4. A method according to claim 3 where the reconfiguring process provides the settings of the sound parameters; gain, equalization and delay for one sound channel, to be applied in one sound field zone related to the physical position of one or more other groups of people including one or more listeners.

5. A method according to claim 1, where a listener's position is detected via sensor means like switches or infrared detectors or strain gauges or temperature sensitive detectors.

LINKPLAY EXHIBIT 1004
Page 1947 of 4822

6. A method according to any preceding claim, where the reconfiguring means is embedded into an audio reproduction system.

7. A method according to claim 1,**characterised in that** the constraint table comprises digital data representing the constraints of the

    1) Passenger(s) position(s), and related acoustical adjustment, attributes and corresponding variable values, and/or
    2) Passenger(s) position(s)" loud speaker channel number, parameter settings (EQ, delay, gain) and/or
    3) Functional settings, such as movie,audio only, loudspeaker channel number, parameter settings (EQ, delay, gain).

8. A method according to claim 1, **characterised in that** the constraint table is configured as a multi dimensional state space, defining relations among variables in either disjunctive form or conjunctive form.

9. A audio system comprising means for carrying out the method according to any of claims 1 to 8, **characterised in that** the automatic reconfiguring is applied on an audio reproduction system comprising two or more sound channels, each channel being active and with individual means for equalization and amplification, the system to constitute an audio reproduction equipment in a room, in a vehicle, in an airplane, in a boat, in a PA system and alike.

LINKPLAY EXHIBIT 1004
Page 1948 of 4822

Figure 1

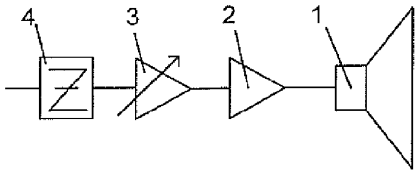
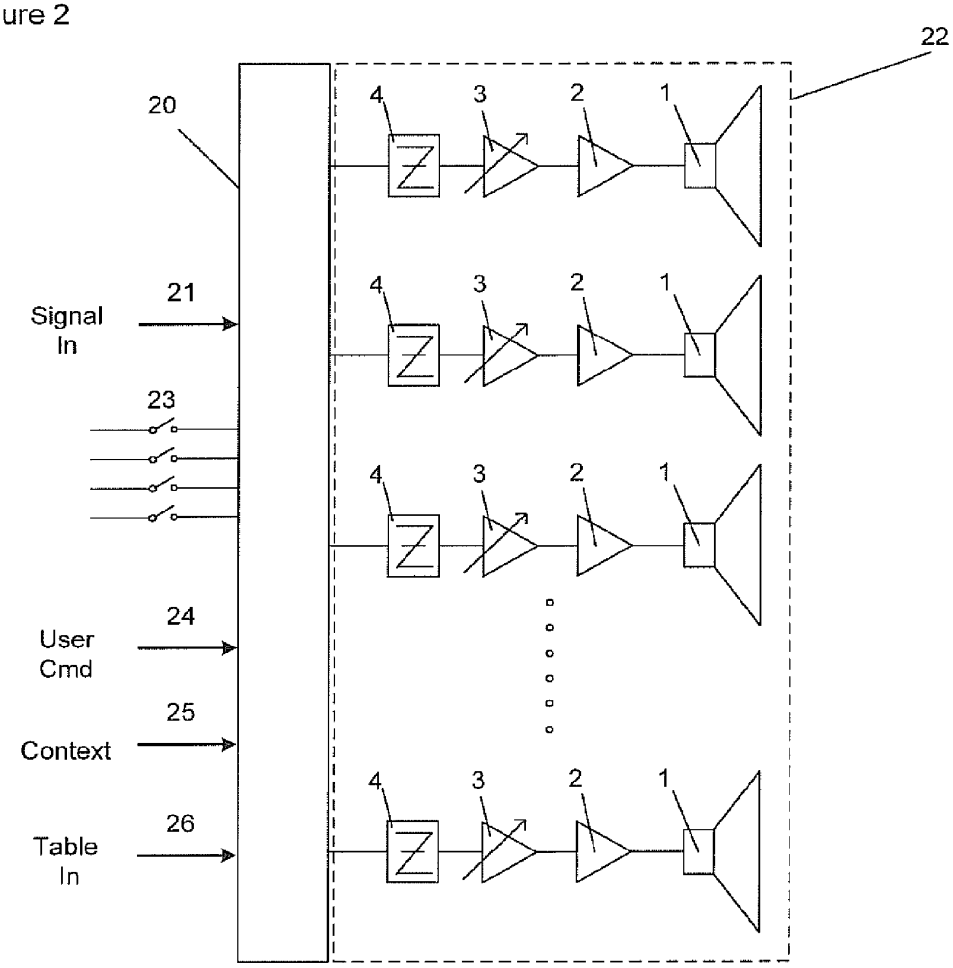


Figure 2



LINKPLAY EXHIBIT 1004
Page 1949 of 4822

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20070211908 A **[0004]**
- US 20040240676 A **[0004]**
- DE 102004022379 **[0004]**

LINKPLAY EXHIBIT 1004
Page 1950 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 591 617 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
18.06.2014 Bulletin 2014/25

(21) Application number: **10788313.4**

(22) Date of filing: **06.12.2010**

(51) Int Cl.:
H04S 7/00 *(2006.01)*

(86) International application number:
PCT/EP2010/068934

(87) International publication number:
WO 2012/003894 (12.01.2012 Gazette 2012/02)

(54) **ADAPTIVE SOUND FIELD CONTROL**

ADAPTIVE KLANGFELDSTEUERUNG

COMMANDE ADAPTATIVE DE CHAMP ACOUSTIQUE

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **09.07.2010 DK 201000609**

(43) Date of publication of application:
15.05.2013 Bulletin 2013/20

(73) Proprietor: **Bang & Olufsen A/S**
7600 Struer (DK)

(72) Inventor: **PEDERSEN, Jan Abildgaard**
DK-7500 Holstebro (DK)

(74) Representative: **Edlund, Fabian**
Awapatent AB
Södra Hamngatan 37-41
P.O. Box 11 394
404 28 Göteborg (SE)

(56) References cited:
EP-A2- 0 772 374
EP-A2- 2 161 950
WO-A1-00/21331
WO-A1-2005/109954
WO-A1-2007/076863
US-B1- 6 760 451

- **ABILDGAARD PEDERSEN JAN ET AL: "Natural Timbre in Room Correction Systems", AES CONVENTION 122; MAY 2007, AES, 60 EAST 42ND STREET, ROOM 2520 NEW YORK 10165-2520, USA, 1 May 2007 (2007-05-01), XP040508184,**
- **ABILDGAARD PEDERSEN ET AL: "Loudspeaker-Room Adaptation for a Specific Listening Position using Information about the Complete Sound Field", AES CONVENTION 121; OCTOBER 2006, AES, 60 EAST 42ND STREET, ROOM 2520 NEW YORK 10165-2520, USA, 1 October 2006 (2006-10-01), XP040507831,**
- **Jan Abildgaard Pedersen: "Adjusting a loudspeaker to its acoustic environment", AES 115th Convention , 5880, 10 October 2003 (2003-10-10), pages 1-13, XP002627369, New York Retrieved from the Internet: URL:http: //www.aes.org/e-lib/browse.cfm?el ib=12471 [retrieved on 2011-03-10]**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

EP 2 591 617 B1

Description

TECHNICAL FIELD

[0001] The invention relates generally to a method and a system for controlling loudspeakers in a room and more specifically to such methods and systems optimised for use in small and/or closed enclosures, such as the cabin of a vehicle, for instance a car.

BACKGROUND OF THE INVENTION

[0002] Methods and systems having the capability to adjust a speaker system to its environment are known within the art. Such methods and systems comprise for instance single point room equalization, obtained by measuring sound pressure in one point near the listening position and subsequent direct inverse filtering. Another prior art procedure comprises adjusting a loudspeaker to its acoustic environment and controlling the acoustic energy in the sound field. Still another procedure comprises adjusting the loudspeaker according to the listening position.

SUMMARY OF THE INVENTION

[0003] The invention is defined by the appended claims.

[0004] The present invention differentiates from the prior art at least with respect to the following aspects:

- Adjustment of the performance in the full audible frequency range.
- According to the invention there are provided a number of individual options for adjustment, such as:
- Acoustic power output (distributed energy);
- Room gain, which may vary according to the actual room design;
- Optimal listening position;
- Listening position combined with supplemental measurements in several positions;
- Real-time adaption, by applying a subset of measurement points from the complete set of possible measurement points;
- Pre-shaped adjustments according to predefined knowledge data.

[0005] A specific embodiment of the method and system according to the invention will be described in detail in the following, but it is understood that other embodiments, for instance of the various functional entities used in the method and system would also be possible and would fall within the scope of the present invention which is defined by the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present invention will be better understood by reading the following detailed description of an embodiment of the invention in conjunction with the drawings, wherein:

Figure 1 shows the three main elements (the acoustic power output from the loudspeakers, the room gain and the listening position interface) of the present invention;

Figure 2 shows sampling of energy or power in a sound field in a room;

Figure 3 shows the measurement of room gain in a room or enclosure by using a special measurement loudspeaker that could be placed at any position in the room or enclosure;

Figure 4 shows an example of a special measurement loudspeaker placed in a car at one of the listening positions. The measurement loudspeaker is fitted with special means to allow easy and reproducible measurement of sound

LINKPLAY EXHIBIT 1004
Page 1952 of 4822

pressure in two positions in the close near field of the loudspeaker diaphragm;

Figure 5 shows measuring the listening position interface at a particular listening position; and

Figure 6 shows real time adaptation that tracks changes of the energy or power in the sound field based on measurements of sound pressure at a subset of random positions, for instance three positions as shown in this figure.

DETAILED DESCRIPTION OF THE INVENTION

[0007] In the following, a specific embodiment of the invention will be described in detail, but it is understood that this embodiment only constitutes an example of how the general concepts of the present invention can be implemented in practice.

[0008] A basic concept of the present invention is that the total transmission path from the electrical input terminal(s) of the loudspeaker(s) to the sound pressure generated at the ears of a listener in an enclosure can be described to constitute three main elements: (i) the acoustic power emitted from the one or more loudspeakers into the enclosure when the one or more loudspeakers are provided with electrical input signals, (ii) the influence of the acoustic properties of the enclosure itself on the sound field created in the enclosure, i.e. on the distribution of sound pressure and hence on average sound pressure generated in the enclosure by the one or more loudspeakers emitting acoustic power into the enclosure, and (iii) the coupling or interface between the created sound field in the enclosure and a listener located at a given position in the enclosure. This is illustrated with reference to figure 1 that shows the three elements that participate in creating a sound field 10 in an enclosure 1 such as a listening room or automobile cabin. An example of how these three elements can be measured in practice is illustrated with reference to figures 2 through 6.

[0009] As shown in figure 1, one or more loudspeaker drive units 2, 3, 4 emit a total acoustic power into the sound field 10 depending on the coupling to the sound field 10 from each driver position and the interaction between the drive units. Depending on the acoustic properties of the listening room/enclosure, more or less power averaged sound pressure will be created in the room/enclosure for a given acoustic power output into the sound field. In order to characterize the influence of the room on the sound field generated herein, the quantity room gain is introduced. Room Gain is here defined as the ratio between the squared power averaged sound pressure in a room and the acoustic power into this room.

[0010] A listener's access to the sound field 10, i.e. his ability to receive sound energy from a sound field and at the specific place herein in which he is located, depends on the coupling to the sound field at the actual listening position. A listener is in figure 1 schematically indicated by the head 6. This coupling can be described by the radiation resistance at the listening position, for instance measured by placing a loudspeaker (14 in figures 3, 4 and 5) in the listening position and measuring the radiation resistance seen by this loudspeaker into the sound field.

[0011] Below follows a detailed description of methods for calculating the above three contributions to the total transmission path and how to determine an electronic filter for obtaining a specific target transmission from the electrical input to the loudspeakers to the ears of the listener.

[0012] Figure 2 shows how the energy or power in a sound field can be measured by power averaging a number of measured sound pressures at random positions 9 scattered across the entire space in both x, y and z directions (length, width and height directions). It is understood that although the figures represent the sound field in only two dimensions, the actual field will of course be three-dimensional, and measurement points generally should be distributed throughout the three-dimensional space. Equation I below shows how to calculate the power averaged sound pressure. The number of random positions must be higher than 1 and in a preferred embodiment it will be 9 positions, but good results can be obtained with 4 positions.

$$p_{channel\_average} = \sqrt{\frac{\sum_{i=1}^{N} \left| p_{channel\_i}(f) \right|^2}{N}} \qquad (1)$$

[0013] $p_{channel\_average}$ is the power averaged sound pressure for a given content channel, e.g. front left channel. $p_{channel\_i}$ is the measured transfer function from the input 5 to a power amplifier (not shown)to the sound pressure at the i'th random position, for instance the one designated by 9' in figure 2.

[0014] Figure 3 illustrates a method of measuring the room gain in a room/enclosure by using a special measurement loudspeaker 14 that can be placed anywhere in the room.

[0015] After placing the special measurement loudspeaker 14 at a chosen position in the room/enclosure as shown

LINKPLAY EXHIBIT 1004
Page 1953 of 4822

in figure 3, sound pressure at a number of random positions I 1 is measured while emitting acoustic signals through the measurement loudspeaker 14. Equation 2 shows how to calculate the power averaged sound pressure $p_{RG\_average}$ of the sound pressures at these random positions.

$$p_{RG\_average} = \sqrt{\frac{\sum_{i=1}^{N} |p_{RG\_i}(f)|^2}{N}} \quad (2)$$

[0016] $p_{RG\_average}$ is the power averaged sound pressure when emitting acoustic signals through the special measurement loudspeaker 14 (the abbreviation RG stands for Room Gain). Figure 4 shows a practical example of such a special measurement loudspeaker mounted at a specific location in an automobile cabin. p_{RG_i} is the transfer function between the inputs to the power amplifier to the measured sound pressure at the i'th random position. (It should be noted that throughout this specification, all generated sound pressures are normalised with respect to the input voltage to the respective amplifiers. It is hence correct to designate for instance the quantity p_{RG_i} as a transfer function).

[0017] While keeping the measurement loudspeaker 14 in the same position (see figure 3), the transfer function from the input of the power amplifier to the sound pressure at two positions 12 and 13 differently spaced from the loudspeaker diaphragm but both in the very near field of the diaphragm is then measured. The distance from the diaphragm to the closest position 12, in which position the sound pressure p_1 is measured, could be e.g. 4 cm while the distance to the other position, in which position the sound pressure p_2 is measured, could be e.g. 9 cm. Equation 3 shows how to calculate the mechanical radiation resistance, $R_{m,s} = Re(Zm,r)$, from the two sound pressures p_1 and p_2 in the near field of the measurement loudspeaker 14.

$$R_{m,r} = \mathrm{Re}\left(j \cdot g \cdot f \cdot \frac{p_1}{p_1 - p_2} \right) \quad (3)$$

[0018] g is a real valued constant describing the scaling due to geometry etc. But g is unimportant in the application and can be set to 1.

[0019] Figure 4 displays one example of a special measurement loudspeaker 14 placed in a car at one of the listening positions. The measurement loudspeaker 14 is fitted with special means 15 to allow easy and reproducible measurement of sound pressure in two positions in the close near field of the loudspeaker diaphragm.

[0020] After calculating the mechanical radiation resistance from equation 3 it is possible to calculate the acoustic power output using equation 4.

$$Pa_{RG\_actual} = \frac{1}{2} \cdot R_{m,r} \cdot |v_{RG}|^2 \quad (4)$$

[0021] v_{RG} is the diaphragm velocity of the measurement loudspeaker, which can be estimated from the difference between the two near field sound pressures p_1 and p_2 as given in equation 5. As for the constant g mentioned above, h is also a real valued constant, which is a function of geometry, out which can be set to I here.

$$v_{RG} = h \cdot \frac{1}{j\omega} \cdot (p_1 - p_2) \quad (5)$$

[0022] The Room Gain in the actual room/enclosure can now be calculated from the definition given by equation 6.

$$Room\ Gain = \frac{p^2_{RG\_average}}{Pa_{RG\_actual}} \quad (6)$$

LINKPLAY EXHIBIT 1004
Page 1954 of 4822

[0023] Figure 5 shows how the special measurement loudspeaker 14 is used to determine the listening position interface at a particular listening position. This measurement can be repeated for as many listening positions as desired. Individual filters can then be designed for each listening position or an average filter can be designed. According to a specific embodiment of the invention, the listening position interface filter is left out, whereby a global equalizing can then be designed. The listening position interface and the corresponding part of the speaker equalization filter, Listener(f), is hence to be regarded as an option.

[0024] The listening position interface is obtained from a similar measurement of two sound pressures at positions 12 and 13, respectively, in near field of the measurement loudspeaker. Again equation 3 can be used to calculate the mechanical radiation resistance from the two measured sound pressures.

Adaptive Sound Field Control Equalization Filters:

[0025] The total equalization filter is split into two main components: a speaker equalization filter and a listening position interface equalization filter as described in equation 7.

$$ASFC(f) = Speaker(f) \bullet Listener(f) \tag{7}$$

[0026] The speaker equalization filter ensures generation of a desired sound field in the room/enclosure while the listening position interface equalization filter compensates for the actual access to the generated sound field.

[0027] The total acoustic power output from the loudspeaker driver(s) for a given content channel can be calculated from the power averaged sound pressure and the measured Room Gain by using equation 8.

$$Pa_{channel\_actual} = \frac{p^2_{channel\_average}}{Room\ Gain} \tag{8}$$

[0028] Then the speaker equalization filter can be calculated from equation 9.

$$Speaker(f) = \sqrt{\frac{Pa_{Adaptive\ Target}(f)}{Pa_{channel\_actual}(f)}} \tag{9}$$

[0029] $Pa_{Adaptive\ Target}$ is a desired target function for the acoustic power output, which may be fixed, selectable or adaptive as a function of the measured total acoustic power output. This target function may also include the difference between a desired Room Gain and the actual Room Gain measured in the room/enclosure. A desired Room Gain may also be fixed, selectable or adaptive to the actual room. According to one embodiment of the invention, the desired Room Gain is equal to the actual Room Gain, i.e. preserving the natural Room Gain in the actual room. An alternative embodiment is to use a desired Room Gain equal to the Room Gain in a particular room of choice, e.g. an IEC 268-13 standard listening room.

[0030] A procedure of pre-shaping and aligning the response of the loudspeakers in the room/enclosure may precede all the measurements described in this document. This is equivalent to designing the basic loudspeaker in a particular room different or equal to the room of usage before applying this procedure for adapting to the actual room/enclosure of usage. E.g. a general trend of decreasing level with increasing frequency can be imposed onto the loudspeaker system before starting the described procedure for adapting the loudspeaker to the room/enclosure. Another example is performing dedicated equalization on a specific loudspeaker driver to correct known problems with that loudspeaker driver, which may be independent of the actual room of usage. Also specific equalization in the actual room/enclosure of usage may be applied before the room adaptation procedure is started.

[0031] Equation 10 gives the listening position interface equalization filter target.

LINKPLAY EXHIBIT 1004
Page 1955 of 4822

$$Listener(f) = \sqrt{\frac{R_{m,r,reference\ listening\ position}(f)}{R_{m,r.actual\ listener\ postion}(f)}} \quad (10)$$

[0032] The radiation resistance in a reference listening position, $R_{m,r,reference\_listening\_position}$ can be measured in a listening position where the access to the sound field is considered to be optimal. E.g. a listening position where the step of pre-shaping and equalization before the procedure of adapting to the room/enclosure is optimized may be used for this reference. A preferred embodiment is, however, to use $R_{m,r,reference\_listening\_position}$ equal to the radiation resistance in free field because in free field, a listener has equal access to all frequencies. The theoretical radiation resistance in free field for a point source is frequency squared multiplied by a constant depending on geometry.

[0033] Equation 11 now gives the complete equalization filter for an Adaptive Sound Field Control (ASFC) system according to the described embodiment of the invention.

$$ASFC(f) = \sqrt{\frac{Pa_{Adaptive\ Target}(f) \bullet R_{m,r,reference\ listening\ position}(f)}{Pa_{channel\_actual}(f) \bullet R_{m.r.actual\ listener\ postion}(f)}} \quad (11)$$

[0034] A special case is when no specific listening position is considered, i.e. equalization globally valid to the whole room/enclosure. Then the total equalization filter reduces to the speaker equalization filter, as in equation 12.

$$ASFC(f) = \sqrt{\frac{Pa_{Adaptive\ Target}(f)}{Pa_{channel\_actual}(f)}} \quad (12)$$

Real time adaptation

[0035] Another aspect of the present invention is the possibility of real time adaption to any changes of loudspeaker drivers and acoustic properties of the room/enclosure. Loudspeaker drivers are known to change as a function of aging and environment changes like temperature and humidity. Also playing loud music will heat the loudspeaker drive units and in particular the voice coils, which will also change the characteristics of the drive units,

[0036] Changing the acoustic properties of the room/enclosure can occur for numerous reasons, for instance as a consequence of how many people are present in the room/enclosure, opening or closing a window/door, opening or closing the roof in a car, fitting more or less furniture/baggage etc. into the room/enclosure. Using room adaptation according to the present invention provides a new possibility of tracking such changes in real time. The whole procedure can be repeated under different configurations, e.g. open/closed windows, and the corresponding equalization filters can be stored and selected manually or automatically when the same configuration happens during use. Alternatively, the whole or a part of the procedure can be performed during actual use in the actual configuration.

[0037] According to a preferred embodiment of the present invention, a subset of the random positions is chosen that has been proved to contain the major part of information obtained by the full set of random positions. Experiments show that a subset of 3 positions can be selected to track the changes due to changes in loudspeaker drive units and/or acoustic properties in the room/enclosure. This is shown in figure 6.

[0038] The three positions are sufficient when combining with the knowledge obtained in the full set of random positions in an initial design or calibration step. The initial or calibration step may be a collection of measurements in different configurations that can be manually selected or automatically selected when the configuration changes.

[0039] Figure 6 shows a real time adaptation tracking changes from measurement of sound pressure at a subset of random position, e.g. three positions.

[0040] In order to be able to carry out the adaptation in real time, i.e. during actual operation of for instance an automobile, those parts of the filter that are determined by means of the dedicated measurement loudspeaker 14 either located at the described position used for determining room gain RG(f) or for determining the listening position interface,

LINKPLAY EXHIBIT 1004
Page 1956 of 4822

are preferably pre-recorded during an initial or calibration phase, as the measurement of p1 and p2 in front of the measurement loudspeaker and/or the use of a measurement loudspeaker at different positions in the enclosure would be difficult or in some situations even impossible to implement during actual operation. Hence, according to a preferred embodiment of the invention, in an initial or calibration phase these quantities are determined for all relevant conditions in the particular enclosure, whereas the acoustic power output from the one or more loudspeakers 2, 3, 4 can be estimated during actual operation, for instance by means of the simplified three-microphone setup shown in figure 6, which only applies three microphones, whereas a larger number of microphones (or measurement positions) is typically used in the initial or calibration phase. With reference to figures 2 and 3, respectively, nine measurement positions are used in this phase.

[0041] For instance, an event can activate the adaption procedure to take place. The event is a trigger to start the adaption algorithm and to perform the measurements, with the three microphones for instance, the calculations and the adjustments accordingly. The trigger may be of different types according to actual product requirements e.g. but not limited to: a) automatically when the car is entered or the engine is started, or b) upon a specific control command given by the user.

[0042] The command given by the user may be e.g. a pressure on button, a touch on a touch panel, a spoken command or a given gesture from a hand or the like.

[0043] The information about the mode of the adaption procedure may be displayed, with an indication of the transition from off -> initiated -> completed -> active. The type of display may e.g. be a LCD screen, an OLED panel, an LED or the like.

[0044] In the preferred embodiment, a combined key input device (switch button) and display device (an LED with multiple colour mode indications) constitute the user interface to the adaption procedure.

Claims

1. A method for determining an equalization filter for one or more loudspeakers provided in an enclosure, such as a listening room or an automobile cabin, the method comprising the steps of:

   (i) providing said one or more loudspeakers (2, 3, 4) with an audio input signal (5) whereby a sound field (10) is generated in the enclosure (1), and determining a power averaged sound pressure emitted from the one or more loudspeakers (2, 3, 4) into said enclosure (1) by power averaging a number of measured sound pressures at a first set of positions (9) in the enclosure (1);
   (ii) determining a room gain RG(f) of the enclosure (1), by

   - placing a measurement sound source (14) at a measurement position in said enclosure (1);
   - determining a power averaged sound pressure emitted by said measurement sound source (14) into said enclosure by power averaging a number of measured sound pressures at a second set of positions (11);
   - measuring sound pressures p_1 and p_2, respectively, at two positions (12) and (13), respectively, in the near field of the measurement sound source (14);
   - based on the measured sound pressures at said two positions (12, 13), calculating a mechanical radiation resistance in the near field of the measurement sound source (14);
   - based on said mechanical radiation resistance, calculating the acoustic power output from the measurement sound source (14); and
   - calculating said room gain RG(f) of the enclosure (1) to the generated sound field (10) as the ratio between the square of the power averaged sound pressure emitted by the measurement sound source (14) and said acoustic power output from the measurement sound source (14).

   (iii) forming a speaker equalization filter (Speaker(f)) as the square root of a ratio between a target acoustic power output and the actual acoustic power output from the loudspeaker driver(s), which actual acoustic power output is calculated as the square of the power averaged sound pressure divided by the room gain RG(f);
   (iv) determining said equalization filter as said speaker equalization filter (Speaker(f)).

2. The method according to claim 1, further comprising:

   - determining a listening position interface equalization filter (Listener (f)) for a specific listening position in the sound field by:

   placing a measurement sound source (14) at this specific listening position in said enclosure, and measuring sound pressures p_1 and p_2, respectively, at two positions (12) and (13), respectively, in the near

LINKPLAY EXHIBIT 1004
Page 1957 of 4822

field of the measurement sound source (14)
based on the measured sound pressures at said two positions (12, 13), calculating a mechanical radiation resistance in the near field of the measurement sound source (14);
determining the listening position interface equalization filter as the square root of a ratio between the mechanical radiation resistance determined at a reference listening position and the mechanical radiation resistance determined at said specific listening position; and

- determining said equalization filter as the product of said speaker equalization filter (Speaker(f)) and said listening position interface equalization filter (Listener(f)).

3. A method according to claim 2, wherein said reference listening position is a position in free field.

4. A method according to any of the preceding claims, comprising the steps of:

(i) during an initial or calibration phase, measuring said room gain RG(f) and optionally the listening position interface equalization filter for different conditions of said enclosure and storing the measurement results in storage means;
(ii) either during said initial or calibration phase determining said power average of measured sound pressures for one or more of said different conditions of said enclosure and storing the measurement results in storage means;
or during said initial or calibration phase determining said power average of measured sound pressures for one condition, such as a reference condition, of said enclosure and storing the measurement result in storage means and during actual use of the loudspeakers (2, 3, 4) in the enclosure carrying out measurements of said sound pressures for up-dating the power averaged sound pressures, thereby to carry out up-dating of characteristics of said equalization filter to correspond to changed conditions of the enclosure.

5. A method according to claim 4, wherein the number N_u of measurement positions used for real-time updating of the filter characteristic is less than the corresponding number N of measurement positions used in the initial or calibration phase.

6. A method according to claim 5, wherein N_u is equal to 3.

7. A system for determining an equalization filter for one or more loudspeakers provided in an enclosure, such as a listening room or an automobile cabin, the system comprising:

(a) one or more microphones for measuring sound pressure at a first set of positions (9) and a second set of positions (11) in a sound field (10) in the enclosure;
(b) a measurement sound source (14) that can be placed at chosen positions in said sound field (10) in said enclosure (1);
(c) means for determining a power averaged sound pressure emitted from the one or more loudspeakers (2, 3, 4) into said enclosure (1) by power averaging a number of measured sound pressures at said first set of positions (9);
(d) means for determining a room gain RG(f) of the enclosure (1) by:

determining a power averaged sound pressure emitted by said measurement sound source (14) into said enclosure by power averaging a number of measured sound pressures at said second set of positions (11);
calculating a mechanical radiation resistance in the near field of the measurement sound source (14) based on measured sound pressures p_1 and p_2 at two positions (12, 13) in the near field of the measurement sound source (14);
calculating the acoustic power output from the measurement sound source (14) based on said mechanical radiation resistance; and
calculating said room gain RG(f) of the enclosure (1) to the generated sound field (10) as the ratio between the square of the power averaged sound pressure emitted by the measurement sound source (14) and said acoustic power output from the measurement sound source (14),

(e) means for forming a speaker equalization filter (Speaker(f)) as the square root of a ratio between a target acoustic power output and the actual acoustic power output from the loudspeaker driver(s), which actual acoustic power output is calculated as the square of the power averaged sound pressure divided by the room gain RG(f);

LINKPLAY EXHIBIT 1004
Page 1958 of 4822

(f) means for determining said equalization filter characteristic as said speaker equalization filter (Speaker(f)).

8. The system in claim 7, further comprising means for determining a listening position interface equalization filter (Listener (f)) for a specific listening position in the sound field by:

calculating a mechanical radiation resistance in the near field of the measurement sound source (14) based on measured sound pressures at two positions (12, 13) in the near field of the measurement sound source (14) placed in a specific listening position in said enclosure, and
determining the listening position interface equalization filter as the square root of a ratio between the mechanical radiation resistance determined at a reference listening position and the mechanical radiation resistance determined at said specific listening position; and
wherein said means for determining said equalization filter are arranged to determine said equalization filter as the product of said speaker equalization filter (Speaker(f)) and said listening position interface equalization filter (Listener(f)).

9. A system according to claim 7 or 8, further comprising means for storing said speaker equalization filter, Speaker(f), and optionally said listening position interface equalization filter, Listener(f) corresponding to different conditions of said enclosure (1), or means for storing those quantities that are derived from measurements of said sound pressures in the enclosure and that are used for determining the speaker equalization filter, Speaker(f), and a listening position interface equalization filter, Listener(f).

10. A system according to claim 9, further comprising a set of microphones that can carry out real time measurements of sound pressure during use of the enclosure and use these measurements together with appropriately stored quantities to determine real time changes of characteristics of said equalization filter or said speaker equalization filter, Speaker(f), and optionally a listening position interface equalization filter, Listener(f).

11. A system according to claim 9 or 10 further comprising means for triggering said real time changes of equalization filter characteristics, for instance when the acoustic properties of the enclosure change, or when the acoustic properties of the loudspeakers (2, 3, 4) that create the acoustic field (10) in the enclosure (1) change.

12. A system according to claim 11, wherein said triggering is performed manually by an operator, such as by the driver or a passenger in a car.

13. A system according to claim 11, wherein said triggering is performed automatically, for instance by opening of a window in a car or when the number of persons in the cabin is changed.

14. The system according to any one of claims 7 to 13, wherein said enclosure is the cabin of vehicle, such as a car.

15. Use of the system according to any of the claims 7 to 14 for implementing the method according to any of the claims 1 to 6.

Patentansprüche

1. Verfahren zum Bestimmen eines Entzerrungsfilters für einen oder mehrere Lautsprecher, die in einer Umfassung, z.B. in einem Hörsaal oder der Kabine eines Automobils, bereitgestellt werden, wobei das Verfahren die folgenden Schritte umfasst:

(i) Versehen des einen oder der mehreren Lautsprecher (2, 3, 4) mit einem Audioeingangssignal (5), wodurch ein Klangfeld (10) in der Umfassung (1) erzeugt wird, und Bestimmen eines über die Leistung gemittelten Schalldrucks, der aus dem einen oder den mehreren Lautsprechern (2, 3, 4) in die Umfassung (1) emittiert wird, durch Mitteln einer Anzahl gemessener Schalldrücke an einer ersten Gruppe von Positionen (9) in der Umfassung (1) über die Leistung;
(ii) Bestimmen einer Raumverstärkung (Room Gain) RG(f) der Umfassung (1) durch

- Anordnen einer Messtonquelle (14) an einer Messposition in der Umfassung (1);
- Bestimmen eines über die Leistung gemittelten Schalldrucks, der aus der Messtonquelle (14) in die Umfassung emittiert wird, durch Mitteln einer Anzahl gemessener Schalldrücke an einer zweiten Gruppe von

LINKPLAY EXHIBIT 1004
Page 1959 of 4822

Positionen (11) über die Leistung;
- Messen von Schalldrücken *p1* bzw. *p2* an zwei Positionen (12) bzw. (13) im Nahbereichsfeld der Messtonquelle (14);
- Berechnen eines mechanischen Strahlungswiderstands im Nahbereichsfeld der Messtonquelle (14) auf der Grundlage der gemessenen Schalldrücke an den zwei Positionen (12, 13);
- Berechnen der akustischen Leistungsausgabe aus der Messtonquelle (14) auf der Grundlage des mechanischen Strahlungswiderstands und
- Berechnen der Raumverstärkung RG(f) der Umfassung (1) für das erzeugte Klangfeld (10) als das Verhältnis zwischen dem Quadrat des über die Leistung gemittelten Schalldrucks, der aus der Messtonquelle (14) emittiert wird, und der akustischen Leistungsausgabe aus der Messtonquelle (14);

iii) Bilden eines Lautsprecher-Entzerrungsfilters (Lautsprecher(f)) als die Quadratwurzel aus einem Verhältnis zwischen einer angestrebten akustischen Leistungsausgabe und der tatsächlichen akustischen Leistungsausgabe aus dem (den) Lautsprechertreiber(n), wobei die tatsächliche akustische Leistungsausgabe als das Quadrat des über die Leistung gemittelten Schalldrucks, dividiert durch die Raumverstärkung RG(f), berechnet wird;
iv) Bestimmen des Entzerrungsfilters als der Lautsprecher-Entzerrungsfilter (Lautsprecher(f)).

2. Verfahren nach Anspruch 1, ferner umfassend:

- Bestimmen eines Hörpositions-Schnittstellen-Entzerrungsfilters (Hörer(f)) für eine spezielle Hörposition in dem Klangfeld durch:

Anordnen einer Messtonquelle (14) an dieser speziellen Hörposition in der Umfassung und
Messen von Schalldrücken *p1* bzw. *p2* an zwei Positionen (12) bzw. (13) im Nahbereichsfeld der Messtonquelle (14);
Berechnen eines mechanischen Strahlungswiderstands im Nahbereichsfeld der Messtonquelle (14) auf der Grundlage der gemessenen Schalldrücke an den zwei Positionen (12, 13);
Bestimmen des Hörpositions-Schnittstellen-Entzerrungsfilters als die Quadratwurzel eines Verhältnisses zwischen dem mechanischen Strahlungswiderstand, bestimmt an einer Referenz-Hörposition, und dem mechanischen Strahlungswiderstand, bestimmt an der speziellen Hörposition; und

- Bestimmen des Entzerrungsfilters als das Produkt aus dem Lautsprecher-Entzerrungsfilter (Lautsprecher(f)) und dem Hörpositions-Schnittstellen-Entzerrungsfilter (Hörer(f)).

3. Verfahren nach Anspruch 2, wobei die referenz-Hörposition eine Position in einem freien Feld ist.

4. Verfahren nach einem der vorhergehenden Ansprüche, welches die folgenden Schritte umfasst:

(i) Messen der Raumverstärkung RG(f) und gegebenenfalls des Hörpositions-Schnittstellen-Entzerrungsfilters für verschiedene Bedingungen der Umfassung und Speichern der Messergebnisse in einem Speichermittel während einer Anfangs- oder Kalibrierungsphase;
(ii) entweder Bestimmen des Leistungsmittelwerts gemessener Schalldrücke für eine oder mehrere der verschiedenen Bedingungen der Umfassung und Speichern der Messergebnisse in einem Speichermittel während der Anfangs- oder Kalibrierungsphase oder Bestimmen des Leistungsmittelwerts gemessener Schalldrücke für eine Bedingung, z.B. eine Referenzbedingung, der Umfassung und Speichern der Messergebnisse in einem Speichermittel während der Anfangs- oder Kalibrierungsphase und Durchführen von Messungen der Schalldrücke zum Aktualisieren der über die Leistung gemittelten Schalldrücke während der tatsächlichen Verwendung der Lautsprecher (2, 3, 4) in der Umfassung, um dadurch eine Aktualisierung von Eigenschaften des Entzerrungsfilters durchzuführen, damit diese veränderten Bedingungen der Umfassung entsprechen.

5. Verfahren nach Anspruch 4, wobei die Anzahl N_u an Messpositionen, die für die Echtzeitaktualisierung der Filtereigenschaften verwendet werden, geringer als die entsprechende Anzahl N an Messpositionen ist, die in der Anfangs- oder Kalibrierungsphase verwendet werden.

6. Verfahren nach Anspruch 5, wobei N_u gleich 3 ist.

7. System zum Bestimmen eines Entzerrungsfilters für einen oder mehrere Lautsprecher, die in einer Umfassung, z. B. einem Hörsaal oder in der Kabine eines Automobils, bereitgestellt werden, wobei das System das Folgende

LINKPLAY EXHIBIT 1004
Page 1960 of 4822

umfasst:

(a) ein oder mehrere Mikrofone zum Messen des Schalldrucks an einer ersten Gruppe von Positionen (9) und einer zweiten Gruppe von Positionen (11) in einem Klangfeld (10) in der Umfassung;
(b) eine Messtonquelle (14), welche an ausgewählten Positionen in dem Klangfeld (10) in der Umfassung (1) angeordnet sein kann;
(c) ein Mittel zum Bestimmen eines über die Leistung gemittelten Schalldrucks, der aus dem einen oder den mehreren Lautsprechern (2, 3, 4) in die Umfassung (1) emittiert wird, durch Mitteln einer Anzahl gemessener Schalldrücke an der ersten Gruppe von Positionen (9) über die Leistung;
(d) ein Mittel zum Bestimmen einer Raumverstärkung RG(f) der Umfassung (1) durch:

Bestimmen eines über die Leistung gemittelten Schalldrucks, der durch die Messtonquelle (14) in die Umfassung emittiert wird, durch Mitteln einer Anzahl gemessener Schalldrücke an einer zweiten Gruppe von Positionen (11) über die Leistung;
Berechnen eines mechanischen Strahlungswiderstands im Nahbereichsfeld der Messtonquelle (14) auf der Grundlage der gemessenen Schalldrücke p_1 und p_2 an zwei Positionen (12, 13) im Nahbereichsfeld der Messtonquelle (14);
Berechnen der akustischen Leistungsausgabe aus der Messtonquelle (14) auf der Grundlage des mechanischen Strahlungswiderstands und
Berechnen der Raumverstärkung RG(f) der Umfassung (1) für das erzeugte Klangfeld (10) als das Verhältnis zwischen dem Quadrat des über die Leistung gemittelten Schalldrucks, der durch die Messtonquelle (14) emittiert wird, und der akustischen Leistungsausgabe aus der Messtonquelle (14),

(e) ein Mittel zum Bilden eines Lautsprecher-Entzerrungsfilters (Lautsprecher(f)) als die Quadratwurzel aus einem Verhältnis zwischen einer angestrebten akustischen Leistungsausgabe und der tatsächlichen akustischen Leistungsausgabe aus dem (den) Lautsprechertreiber(n), wobei die tatsächliche akustische Leistungsausgabe als das Quadrat des über die Leistung gemittelten Schalldrucks, dividiert durch die Raumverstärkung RG(f), berechnet wird;
(f) ein Mittel zum Bestimmen der Entzerrungsfiltereigenschaft als der Lautsprecher-Entzerrungsfilter (Lautsprecher(f)).

8. System nach Anspruch 7, ferner umfassend ein Mittel zum Bestimmen eines Hörpositions-Schnittstellen-Entzerrungsfilters (Hörer(f)) für eine spezielle Hörposition in dem Klangfeld durch:

Berechnen eines mechanischen Strahlungswiderstands im Nahbereichsfeld der Messtonquelle (14) auf der Grundlage von gemessenen Schalldrücken an zwei Positionen (12, 13) im Nahbereichsfeld der Messtonquelle (14), die an einer speziellen Hörposition in der Umfassung angeordnet sind; und
Bestimmen des Hörpositions-Schnittstellen-Entzerrungsfilters als die Quadratwurzel eines Verhältnisses zwischen dem mechanischen Strahlungswiderstand, bestimmt an einer Referenz-Hörposition, und
dem mechanischen Strahlungswiderstand, bestimmt an der speziellen Hörposition; und
wobei das Mittel zum Bestimmen des Entzerrungsfilters dafür eingerichtet ist, den Entzerrungsfilter als das Produkt aus dem Lautsprecher-Entzerrungsfilter (Lautsprecher(f)) und dem Hörpositions-Schnittstellen-Entzerrungsfilter (Hörer(f)) zu bestimmen.

9. System nach Anspruch 7 oder 8, ferner umfassend ein Mittel zum Speichern des Lautsprecher-Entzerrungsfilters, Lautsprecher(f), und gegebenenfalls des Hörpositions-Schnittstellen-Entzerrungsfilters, Hörer(f), entsprechend verschiedenen Bedingungen der Umfassung (1) oder ein Mittel zum Speichern jener Größen, die aus Messungen der Schalldrücke in der Umfassung abgeleitet werden und die zum Bestimmen des Lautsprecher-Entzerrungsfilters, Lautsprecher(f), und eines Hörpositions-Schnittstellen-Entzerrungsfilters, Hörer(f), verwendet werden.

10. System nach Anspruch 9, ferner umfassend eine Gruppe von Mikrofonen, welche Echtzeitmessungen des Schalldrucks während der Verwendung der Umfassung durchführen können und diese Messungen zusammen mit geeignet gespeicherten Größen verwenden können, um Echtzeitveränderungen von Eigenschaften des Entzerrungsfilters oder des Lautsprecher-Entzerrungsfilters, Lautsprecher(f), oder gegebenenfalls eines Hörpositions-Schnittstellen-Entzerrungsfilters, Hörer(f), zu bestimmen.

11. System nach Anspruch 9 oder 10, ferner umfassend ein Mittel zum Auslösen der Echtzeitveränderungen von Entzerrungsfiltereigenschaften, zum Beispiel wenn sich die akustischen Eigenschaften der Umfassung ändern oder

LINKPLAY EXHIBIT 1004
Page 1961 of 4822

wenn sich die akustischen Eigenschaften der Lautsprecher (2, 3, 4) ändern, welche das Klangfeld (10) in der Umfassung (1) erzeugen.

12. System nach Anspruch 11, wobei das Auslösen manuell durch einen Operator durchgeführt wird, z.B. durch den Fahrer oder einen Passagier in einem Automobil.

13. System nach Anspruch 11, wobei das Auslösen automatisch durchgeführt wird, zum Beispiel durch Öffnen eines Fensters in einem Automobil oder wenn sich die Anzahl der Personen in der kabine ändert.

14. System nach einem der Ansprüche 7 bis 13, wobei die Umfassung die Kabine eines Fahrzeugs, z.B. eines Automobils, ist.

15. Verwendung des Systems nach einem der Ansprüche 7 bis 14 zum Realisieren des Verfahrens nach einem der Ansprüche 1 bis 6.

Revendications

1. Dispositif de détermination d'un filtre d'égalisation pour un ou plusieurs haut-parleurs fournis dans une enceinte, comme une salle d'écoute ou une cabine automobile, le procédé comprenant les étapes de :

(i) la fourniture d'un signal d'entrée audio (5) aux dits un ou plusieurs haut-parleurs (2, 3, 4) de manière à générer un champ de son (10) dans l'enceinte (1), et la détermination d'une pression de son de puissance moyenne émise à partir de l'un ou plusieurs haut-parleurs (2, 3, 4) dans ladite enceinte (1) en calculant la moyenne de puissance d'un nombre de pressions de son mesurées à un premier ensemble de positions (9) dans l'enceinte (1) ;
(ii) la détermination d'un gain de salle, RG(f), de l'enceinte (1) par

- le placement d'une source de son de mesure (14) à une position de mesure dans ladite enceinte (1) ;
- la détermination d'une pression de son de puissance moyenne émise par ladite source de son de mesure (14) dans ladite enceinte par le calcul de la moyenne de puissance d'un nombre de pressions de son mesurées à un deuxième ensemble de positions (11) ;
- la mesure de pressions de son p_1 et p_2 respectivement à deux positions (12) et (13) respectivement dans le champ proche de la source de son de mesure (14) ;
- sur la base des pressions de son mesurées aux dites deux positions (12, 13), le calcul d'une résistance de rayonnement mécanique dans le champ proche de la source de son de mesure (14) ;
- sur la base de ladite résistance de rayonnement mécanique, le calcul de la sortie de puissance acoustique de la source de son de mesure (14) ; et
- le calcul dudit gain de salle, RG(f), de l'enceinte (1) dans le champ de son généré (10) en tant que rapport entre le carré de la pression de son de puissance moyenne émise par la source de son de mesure (14) et ladite sortie de puissance acoustique de la source de son de mesure (14) ;

(iii) la formation d'un filtre d'égalisation de haut-parleurs (Speaker(f)) en tant que racine carrée d'un rapport entre une sortie de puissance acoustique cible et la sortie de puissance acoustique réelle du ou des pilotes de haut-parleurs, ladite sortie de puissance acoustique réelle étant calculée en tant que carré de la pression de son de puissance moyenne divisé par le gain de salle, RG(f) ;
(iv) la détermination dudit filtre d'égalisation en tant que ledit filtre d'égalisation de haut-parleurs (Speaker(f)).

2. Procédé selon la revendication 1, comprenant en outre :

- la détermination d'un filtre d'égalisation d'interface de position d'écoute (Listener(f)) pour une position d'écoute spécifique dans le champ de son par :

le placement d'une source de son mesure (14) à cette position d'écoute spécifique dans ladite enceinte, et la mesure de pressions de son p_1 et p_2 respectivement à deux positions (12) et (13) respectivement dans le champ proche de la source de son de mesure (14),
sur la base des pressions de son mesurées aux dites deux positions (12, 13), le calcul d'une résistance de rayonnement mécanique dans le champ proche de la source de son de mesure (14) ;

LINKPLAY EXHIBIT 1004
Page 1962 of 4822

la détermination du filtre d'égalisation d'interface de position d'écoute en tant que racine carrée d'un rapport entre la résistance de rayonnement mécanique déterminée à une position d'écoute de référence et la résistance de rayonnement mécanique déterminée à ladite position d'écoute spécifique ; et

- la détermination dudit filtre d'égalisation en tant que produit dudit filtre d'égalisation de haut-parleurs (Speaker(f)) et dudit filtre d'égalisation d'interface de position d'écoute (Listener(f)).

3. Procédé selon la revendication 2, dans lequel ladite position d'écoute de référence est une position dans un champ libre.

4. Procédé selon l'une quelconque des revendications précédentes, comprenant les étapes de :

(i) au cours d'une phase initiale ou d'étalonnage, la mesure dudit gain de salle RG(f) et facultativement le filtre d'égalisation de l'interface de position d'écoute pour différentes conditions de ladite enceinte et la mémorisation des résultats de mesure dans un moyen de mémorisation ;
(ii) au cours de ladite phase initiale ou d'étalonnage, la détermination de ladite moyenne de puissance de pressions de son mesurées pour une ou plusieurs desdites conditions différentes de ladite enceinte et la mémorisation des résultats de mesure dans le moyen de mémorisation ; ou
au cours de ladite phase initiale ou d'étalonnage, la détermination de ladite moyenne de puissance de pressions de son mesurées pour une condition, comme une condition de référence, de ladite enceinte, et la mémorisation du résultat de mesure dans le moyen de mémorisation et, au cours de l'utilisation réelle des haut-parleurs (2, 3, 4) dans l'enceinte, l'exécution de mesures desdites pressions de son pour mettre à jour les pressions de son de puissance moyenne, afin d'effectuer de ce fait la mise à jour de caractéristiques dudit filtre d'égalisation pour correspondre à des conditions changées de l'enceinte.

5. Procédé selon la revendication 4, dans lequel le nombre N_u de positions de mesure utilisées pour la mise à jour en temps réel des caractéristiques de filtre est inférieur au nombre correspondant N de positions de mesure utilisées dans la phase initiale ou d'étalonnage.

6. Procédé selon la revendication 5, dans lequel N_u est égal à 3.

7. Système de détermination d'un filtre d'égalisation pour un ou plusieurs haut-parleurs fournis dans une enceinte, comme une salle d'écoute ou une cabine automobile, le système comprenant :

(a) un ou plusieurs microphones pour mesurer une pression de son à un premier ensemble de positions (9) et un deuxième ensemble de positions (11) dans un champ de son (10) dans l'enceinte ;
(b) une source de son de mesure (14) qui peut être placée à des positions choisies dans ledit champ de son (10) dans ladite enceinte (1) ;
(c) un moyen pour déterminer une pression de son de puissance moyenne émise à partir de l'un ou plusieurs haut-parleurs (2, 3, 4) dans ladite enceinte (1) en calculant la moyenne de puissance d'un nombre de pressions de son mesurées au dit premier ensemble de positions (9) ;
(d) un moyen pour déterminer un gain de salle, RG(f), de l'enceinte (1) par:

la détermination d'une pression de son de puissance moyenne émise par ladite source de son de mesure (14) dans ladite enceinte par le calcul de la moyenne de puissance d'un nombre de pressions de son mesurées au dit deuxième ensemble de positions (11) ;
le calcul d'une résistance de rayonnement mécanique dans le champ proche de la source de son de mesure (14) sur la base de pressions de son mesurées p_1 et p_2 à deux positions (12, 13) dans le champ proche de la source de son de mesure (14) ;
le calcul de la sortie de puissance acoustique de la source de son de mesure (14) sur la base de ladite résistance de rayonnement mécanique ; et
le calcul dudit gain de salle, RG(f), de l'enceinte (1) dans le champ de son généré (10) en tant que rapport entre le carré de la pression de son de puissance moyenne émise par la source de son de mesure (14) et ladite sortie de puissance acoustique de la source de son de mesure (14) ;

(e) un moyen pour former un filtre d'égalisation de haut-parleurs (Speaker(f)) en tant que racine carrée d'un rapport entre une sortie de puissance acoustique cible et la sortie de puissance acoustique réelle du ou des pilotes de haut-parleurs, ladite sortie de puissance acoustique réelle étant calculée en tant que carré de la

LINKPLAY EXHIBIT 1004
Page 1963 of 4822

pression de son de puissance moyenne divisé par le gain de salle, RG(f) ;
(f) un moyen pour déterminer ladite caractéristique de filtre d'égalisation en tant que ledit filtre d'égalisation de haut-parleurs (Speaker(f)).

8. Système selon la revendication 1, comprenant un moyen pour déterminer un filtre d'égalisation d'interface de position d'écoute (Listener(f)) pour une position d'écoute spécifique dans le champ de son par :

le calcul d'une résistance de rayonnement mécanique dans le champ proche de la source de son de mesure (14) sur la base des pressions de son mesurées aux dites deux positions (12, 13) dans le champ proche de la source de son de mesure (14) placée à une position d'écoute spécifique dans ladite enceinte, et
la détermination du filtre d'égalisation d'interface de position d'écoute en tant que racine carrée d'un rapport entre la résistance de rayonnement mécanique déterminée à une position d'écoute de référence et la résistance de rayonnement mécanique déterminée à ladite position d'écoute spécifique ; et
dans lequel ledit moyen pour déterminer ledit filtre d'égalisation est agencé pour déterminer ledit filtre d'égalisation en tant que produit dudit filtre d'égalisation de haut-parleurs (Speaker(f)) et dudit filtre d'égalisation d'interface de position d'écoute (Listener(f)).

9. Système selon la revendication 7 ou 8, comprenant en outre un moyen pour mémoriser ledit filtre d'égalisation de haut-parleurs, Speaker(f), et facultativement ledit filtre d'égalisation d'interface de position d'écoute, Listener(f) correspondant à différentes conditions de ladite enceinte (1), ou un moyen pour mémoriser ces quantités qui sont dérivées de mesures desdites pressions de son dans l'enceinte et qui sont utilisées pour déterminer le filtre d'égalisation de haut-parleurs, Speaker(f), et un filtre d'égalisation d'interface de position d'écoute, Listener(f).

10. Système selon la revendication 9, comprenant en outre un ensemble de microphones qui peuvent effectuer des mesures en temps réel de pressions de son au cours d'une utilisation de l'enceinte et utiliser ces mesures avec des quantités mémorisées de manière appropriée pour déterminer des changements en temps réel de caractéristiques dudit filtre d'égalisation ou dudit filtre d'égalisation de haut-parleurs, Speaker(f), et facultativement un filtre d'égalisation d'interface de position d'écoute, Listener(f).

11. Système selon la revendication 9 ou 10, comprenant en outre un moyen pour déclencher lesdits changements en temps réel de caractéristiques de filtre d'égalisation, par exemple lorsque les propriétés acoustiques de l'enceinte changent, ou lorsque les propriétés acoustiques des haut-parleurs (2, 3, 4) qui créent le champ acoustique (10) dans l'enceinte (1) changent.

12. Système selon la revendication 11, dans lequel ledit déclenchement est effectué manuellement par un opérateur, comme par le conducteur ou un passager d'une voiture.

13. Système selon la revendication 11, dans lequel ledit déclenchement est effectué automatiquement, par exemple en ouvrant une vitre d'une voiture ou lorsque le nombre de personnes dans la cabine est changé.

14. Système selon l'une quelconque des revendications 7 à 13, dans lequel ladite enceinte est la cabine d'un véhicule, comme une voiture.

15. Utilisation du système selon l'une quelconque des revendications 7 à 14 pour mettre en oeuvre le procédé selon l'une quelconque des revendications 1 à 6.

LINKPLAY EXHIBIT 1004
Page 1964 of 4822

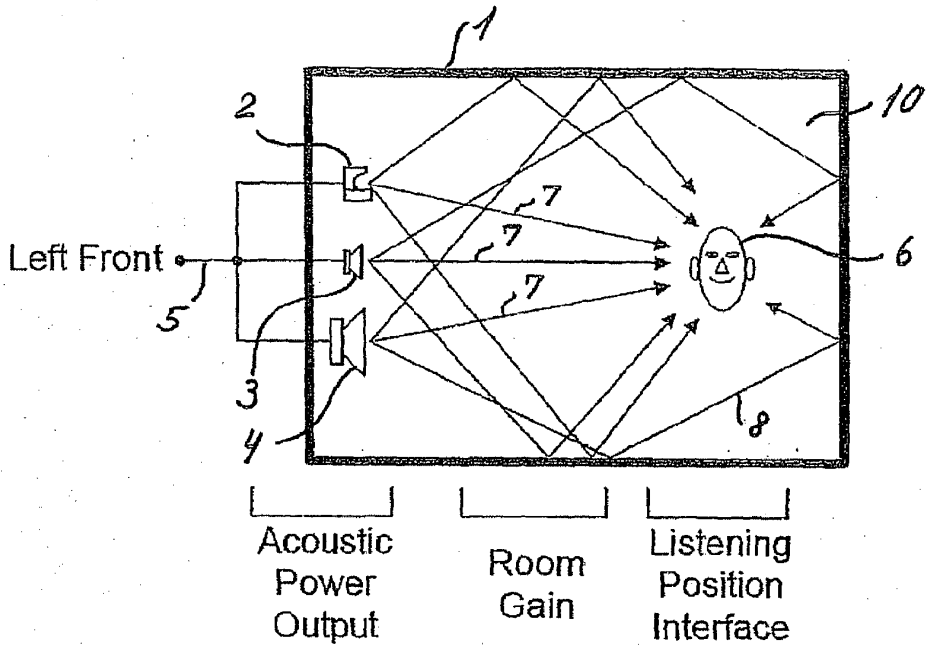


Fig. 1

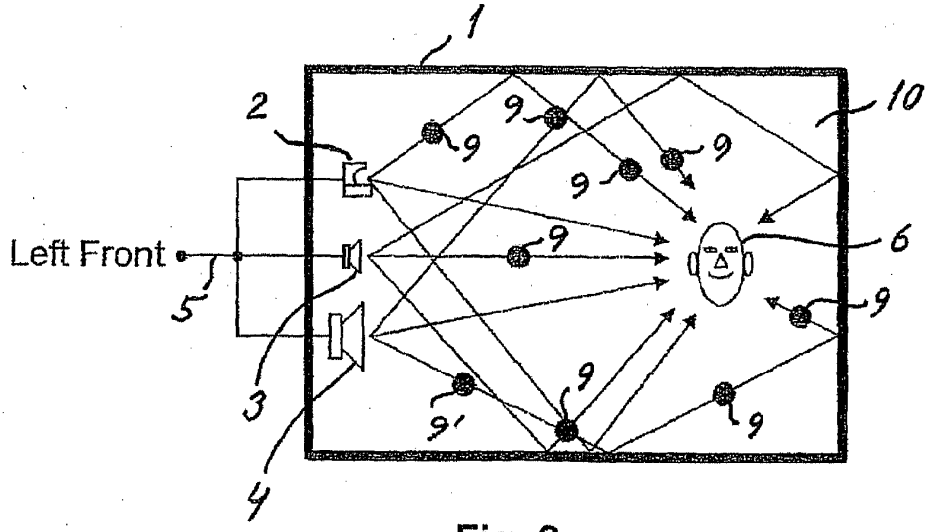


Fig. 2

LINKPLAY EXHIBIT 1004
Page 1965 of 4822

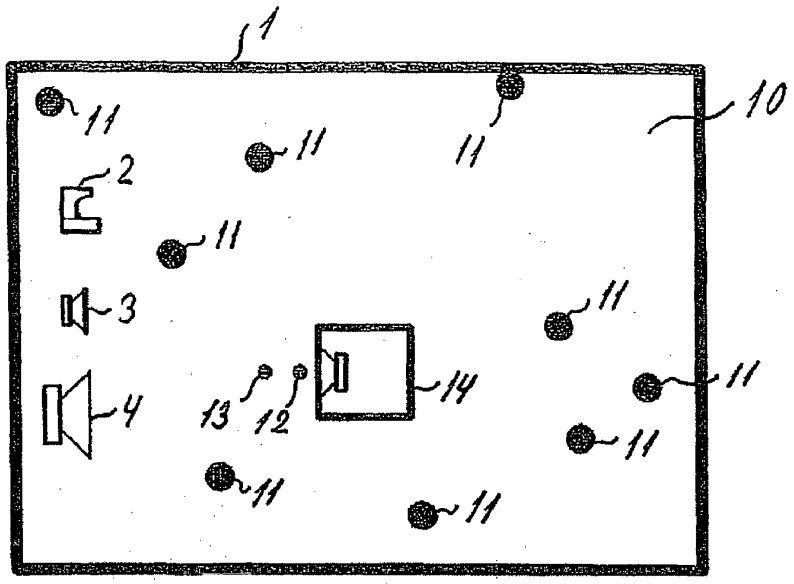


Fig. 3

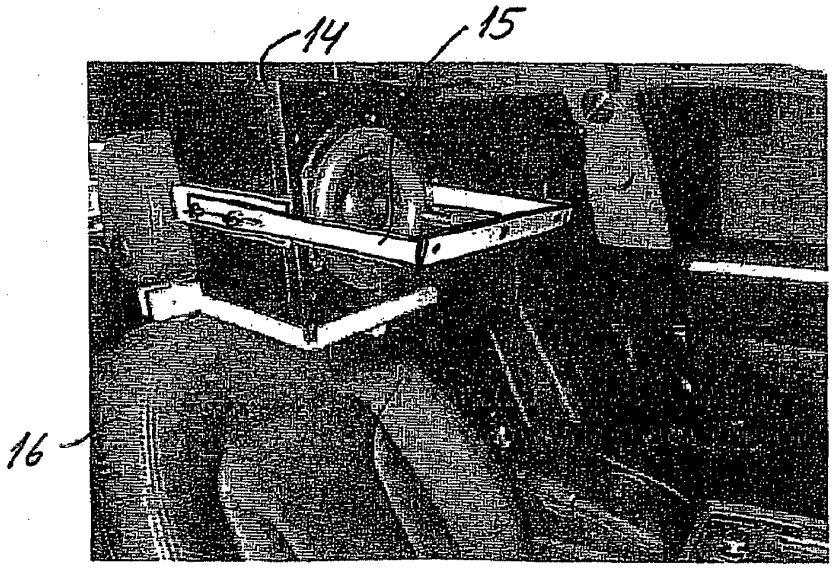


Fig. 4

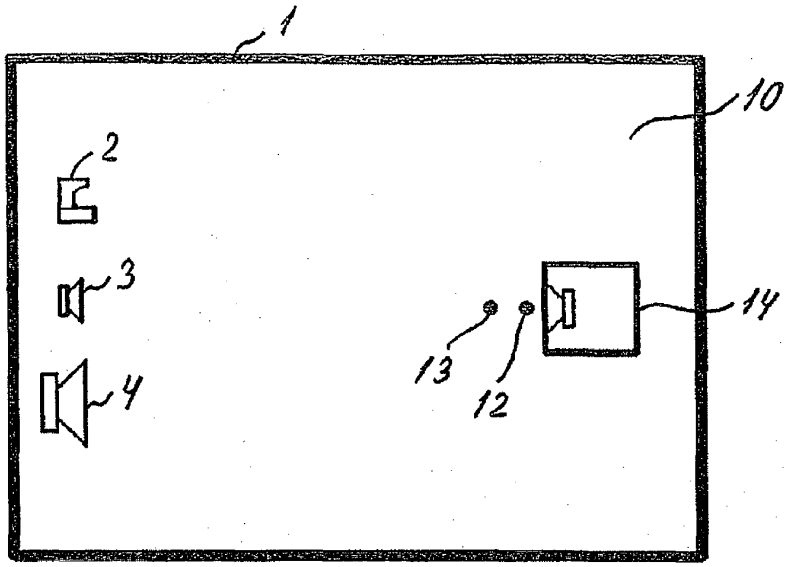


Fig. 5

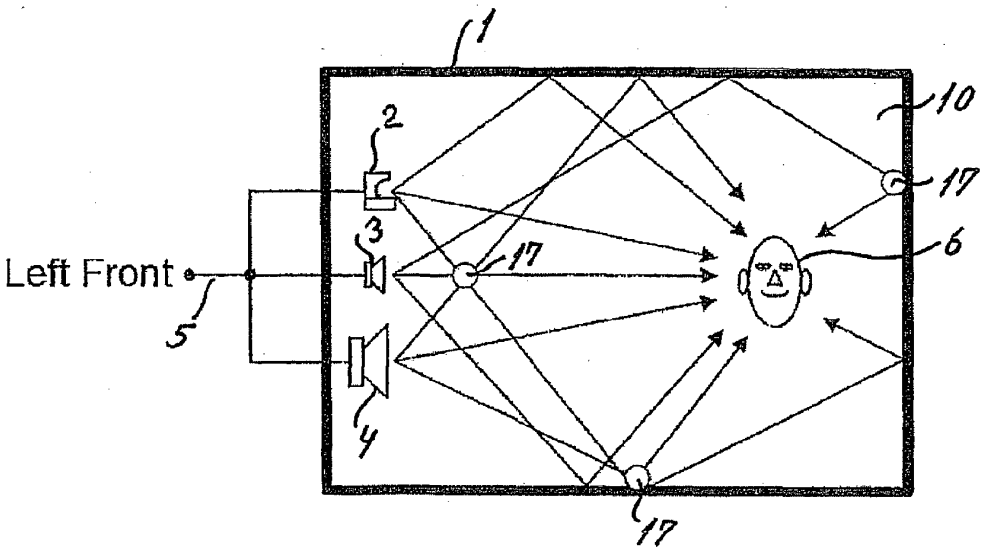


Fig. 6

LINKPLAY EXHIBIT 1004
Page 1967 of 4822

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 713 281 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.04.2014 Bulletin 2014/14

(51) Int Cl.:
G06F 15/16 *(2006.01)* ***G06F 9/48*** *(2006.01)*

(21) Application number: **13184747.7**

(22) Date of filing: **02.07.2004**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PL PT RO SE SI SK TR

(30) Priority: **28.07.2003 US 490768 P**
01.04.2004 US 816217

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
13155130.1 / 2 648 111
04778540.7 / 1 654 614

(71) Applicant: **Sonos, Inc.**
Santa Barbara, California 93101 (US)

(72) Inventor: **Millington, Nicholas A. J.**
Santa Barbara, CA 93103 (US)

(74) Representative: **EIP**
Fairfax House
15 Fulwood Place
London WC1V 6HU (GB)

Remarks:
•This application was filed on 17-09-2013 as a divisional application to the application mentioned under INID code 62.
•Claims filed after the date of filing of the application (Rule 68(4) EPC).

(54) System and method for synchronizing operations among a plurality of independently clocked digital data processing devices

(57) A system is described for maintaining synchrony of operations among a plurality of devices that have independent clocking arrangements. The system includes a task distribution device that distributes tasks to a synchrony group comprising a plurality of devices that are to perform the tasks distributed by the task distribution device in synchrony. The task distribution device distributes each task to the members of the synchrony group over a network. Each task is associated with a time stamp that indicates a time, relative to a clock maintained by the task distribution device, at which the members of the synchrony group are to execute o the task. Each member of the synchrony group periodically obtains from the task distribution device an indication of the current time indicated by its clock, determines a time differential between the task distribution device's clock and its respective clock and determines therefrom a time at which, according to its respective clock, the time stamp indicates that it is to execute the task.

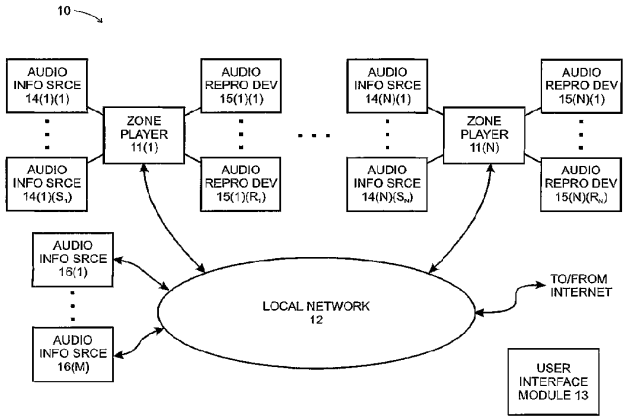


FIG. 1

Printed by Jouve, 75001 PARIS (FR)

EP 2 713 281 A1

Description

FIELD OF THE INVENTION

[0001] The present invention relates generally to the field of digital data processing devices, and more particularly to systems and methods for synchronizing operations among a plurality of independently-clocked digital data processing devices. The invention is embodied in a system for synchronizing operations among a plurality of devices, in relation to information that is provided by a common source. One embodiment of the invention enables synchronizing of audio playback as among two or more audio playback devices that receive audio information from a common information source, or channel.

[0002] More generally, the invention relates to the field of arrangements that synchronize output generated by a number of output generators, including audio output, video output, combinations of audio and video, as well as other types of output as will be appreciated by those skilled in the art, provided by a common channel. Generally, the invention will find utility in connection with any type of information for which synchrony among independently-clocked devices is desired.

BACKGROUND OF THE INVENTION

[0003] There are a number of circumstances under which it is desirable to maintain synchrony of operations among a plurality of independently-clocked digital data processing devices in relation to, for example, information that is provided thereto by a common source. For example, systems are being developed in which one audio information source can distribute audio information in digital form to a number of audio playback devices for playback. The audio playback devices receive the digital information and convert it to analog form for playback. The audio playback devices may be located in the same room or they may be distributed in different rooms in a residence such as a house or an apartment, in different offices in an office building, or the like. For example, in a system installed in a residence, one audio playback device may be located in a living room, while another audio playback device is be located in a kitchen, and yet other audio playback devices may be located in various bedrooms of a house. In such an arrangement, the audio information that is distributed to various audio playback devices may relate to the same audio program, or the information may relate to different audio programs. If the audio information source provides audio information relating to the same audio program to two or more audio playback devices at the same time, the audio playback devices will generally contemporaneously play the same program. For example, if the audio information source provides audio information to audio playback devices located in the living room and kitchen in a house at the same time, they will generally contemporaneously play the same program.

[0004] One problem that can arise is to ensure that, if two or more audio playback devices are contemporaneously attempting to play back the same audio program, they do so simultaneously. Small differences in the audio playback devices' start times and/or playback speeds can be perceived by a listener as an echo effect, and larger differences can be very annoying. Differences can arise because for a number of reasons, including delays in the transfer of audio information over the network. Such delays can differ as among the various audio playback devices for a variety of reasons, including where they are connected into the network, message traffic and other reasons as will be apparent to those skilled in the art.

[0005] Another problem arises from the following. When an audio playback device converts the digital audio information from digital to analog form, it does so using a clock that provides timing information. Generally, the audio playback devices that are being developed have independent clocks, and, if they are not clocking at precisely the same rate, the audio playback provided by the various devices can get out of synchronization.

SUMMARY OF THE INVENTION

[0006] The invention provides a new and improved system and method for synchronizing operations among a number of digital data processing devices that are regulated by independent clocking devices. Generally, the invention will find utility in connection with any type of information for which synchrony among devices connected to a network is desired. The invention is described in connection with a plurality of audio playback devices that receive digital audio information that is to be played back in synchrony, but it will be appreciated that the invention can find usefulness in connection with any kind of information for which coordination among devices that have independent clocking devices would find utility.

[0007] In brief summary, the invention provides, in one aspect, a system for maintaining synchrony of operations among plurality of devices that have independent clocking arrangements. The system includes a task distribution device that distributes tasks to a synchrony group comprising a plurality of devices that are to perform the tasks distributed by the task distribution device in synchrony. The task distribution device distributes each task to the members of the synchrony group over a network. Each task is associated with a time stamp that indicates a time, relative to a clock maintained by the task distribution device, at which the members of the synchrony group are to execute the task. Each

LINKPLAY EXHIBIT 1004
Page 1969 of 4822

member of the synchrony group periodically obtains from the task distribution device an indication of the current time indicated by its clock, determines a time differential between the task distribution device's clock and its respective clock and determines therefrom a time at which, according to its respective clock, the time stamp indicates that it is to execute the task.

[0008] In one embodiment, the tasks that are distributed include audio information for an audio track that is to be played by all of the devices comprising the synchrony group synchronously. The audio track is divided into a series of frames, each of which is associated with a time stamp indicating the time, relative to the clock maintained by an audio information channel device, which, in that embodiment, serves as the task distribution device, at which the members of the synchrony group are to play the respective frame. Each member of the synchrony group, using a very accurate protocol, periodically obtains the time indicated by the audio information channel device, and determines a differential between the time as indicated by its local clock and the audio information channel device's clock. The member uses the differential and the time as indicated by the time stamp to determine the time, relative to its local clock, at which it is to play the respective frame. The members of the synchrony group do this for all of the frames, and accordingly are able to play the frames in synchrony.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1 schematically depicts an illustrative networked audio system, constructed in accordance with the invention;
FIG. 2 schematically depicts a functional block diagram of a synchrony group utilizing a plurality of zone players formed within the networked audio system depicted in FIG. 1 ;
FIG. 2A schematically depicts two synchrony groups, illustrating how a member of one synchrony group can provide audio information to the members of another synchrony group;
FIG. 3 depicts an functional block diagram of a zone player for use in the networked audio system depicted in FIG. 1 ; and
FIG. 4 is useful in understanding an digital audio information framing methodology useful in the network audio system depicted in FIG. 1.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

[0010] FIG. 1 depicts an illustrative network audio system 10 constructed in accordance with the invention. With reference to FIG. 1, the network audio system 10 includes a plurality of zone players 11(1) through 11(N) (generally identified by reference numeral 11(n)) interconnected by a local network 12, all of which operate under control of one or more user interface modules generally identified by reference numeral 13. One or more of the zone players 11(n) may also be connected to one or more audio information sources, which will generally be identified herein by reference numeral 14(n)(s), and/or one or more audio reproduction devices, which will generally be identified by reference numeral 15(n)(r). In the reference numeral 14(n)(s), index "n" refers to the index "n" of the zone player 11(n) to which the audio information source is connected, and the index "s" (s=1,...,S_n) refers to the "s-th" audio information source connected to that "n-th" zone player 11(n). Thus, if, for example, a zone player 11(n) is connected to four audio information sources 14(n)(1) through 14(n)(4), the audio information sources may be generally identified by reference numeral 14(n)(s), with S_n=4. It will be appreciated that the number of audio information sources S_n may vary as among the various zone players 11(n), and some zone players may not have any audio information sources connected thereto. Similarly, in the reference numeral 15(n)(r), index "n" refers to the index "n" of the zone player 11(n) to which the audio reproduction device is connected, and the index "r" (r=1,...,R_n) refers to the "r-th" audio information source connected to that "n-th" zone player 11(n). In addition to the audio information sources 14(n)(s), the network audio system 10 may include one or more audio information sources 16(1) through 16(M) connected through appropriate network interface devices (not separately shown) to the local network 12. Furthermore, the local network may include one or more network interface devices (also not separately shown) that are configured to connect the local network 12 to other networks, including a wide area network such as the Internet, the public switched telephony network (PSTN) or other networks as will be apparent to those skilled in the art, over which connections to audio information sources may be established.

[0011] The zone players 11(n) associated with system 10 may be distributed throughout an establishment such as residence, an office complex, a hotel, a conference hall, an amphitheater or auditorium, or other types of establishments as will be apparent to those skilled in the art or the like. For example, if the zone players 11(n) and their associated audio information source(s) and/or audio reproduction device(s) are distributed throughout a residence, one, such as zone player 11(1) and its associated audio information source(s) and audio reproduction device(s) may be located in a living

LINKPLAY EXHIBIT 1004
Page 1970 of 4822

room, another may be located in a kitchen, another may be located in a dining room, and yet others may be located in respective bedrooms, to selectively provide entertainment in those rooms. On the other hand, if the zone players 11(n) and their associated audio information source(s) and/or audio reproduction device(s) are distributed throughout an office complex, one may, for example, be provided in each office to selectively provide entertainment to the employees in the respective offices. Similarly, if the zone players 11(n) and associated audio information source(s) and/or audio reproduction device(s) are used in a hotel, they may be distributed throughout the rooms to provide entertainment to the guests. Similar arrangements may be used with zone players 11(n) and associated audio information source(s) and/or audio reproduction device(s) used in an amphitheater or auditorium. Other arrangements in other types of environments will be apparent to those skilled in the art. In each case, the zone players 11(n) can be used to selectively provide entertainment in the respective locations, as will be described below.

[0012] The audio information sources 14(n)(s) and 16(m) may be any of a number of types of conventional sources of audio information, including, for example, compact disc ("CD") players, AM and/or FM radio receivers, analog or digital tape cassette players, analog record turntables and the like. In addition, the audio information sources 14(n)(s) and 16(m) may comprise digital audio files stored locally on, for example, personal computers (PCs), personal digital assistants (PDAs), or similar devices capable of storing digital information in volatile or non-volatile form. As noted above, the local network 12 may also have an interface (not shown) to a wide area network, over which the network audio system 10 can obtain audio information. Moreover, one or more of the audio information sources 14(n)(s) may also comprise an interface to a wide area network such as the Internet, the public switched telephony network (PSTN) or any other source of audio information. In addition, one or more of the audio information sources 14(n)(s) and 16(m) may comprise interfaces to radio services delivered over, for example, satellite. Audio information obtained over the wide area network may comprise, for example, streaming digital audio information such as Internet radio, digital audio files stored on servers, and other types of audio information and sources as will be appreciated by those skilled in the art. Other arrangements and other types of audio information sources will be apparent to those skilled in the art.

[0013] Generally, the audio information sources 14(n)(s) and 16(m) provide audio information associated with audio programs to the zone players for playback. A zone player that receives audio information from an audio information source 14(n)(s) that is connected thereto can provide playback and/or forward the audio information, along with playback timing information, over the local network 12 to other zone players for playback. Similarly, each audio information source 16(m) that is not directly connected to a zone player can transmit audio information over the network 12 to any zone player 11(n) for playback. In addition, as will be explained in detail below, the respective zone player 11(n) can transmit the audio information that it receives either from an audio information source 14(n)(s) connected thereto, or from an audio information source 16(m), to selected ones of the other zone players 11(n'), 11(n"),... (n not equal to n', n",...) for playback by those other zone players. The other zone players 11(n'), 11(n"),... to which the zone player 11(n) transmits the audio information for playback may be selected by a user using the user interface module 13. In that operation, the zone player 11(n) will transmit the audio information to the selected zone players 11(n'), 11(n"),... over the network 12. As will be described below in greater detail, the zone players 11(n), 11(n'), 11(n"),... operate such that the zone players 11(n'), 11(n"),... synchronize their playback of the audio program with the playback by the zone player 11(n), so that the zone players 11(n), 11(n'), 11(n") provide the same audio program at the same time.

[0014] Users, using user interface module 13, may also enable different groupings or sets of zone players to provide audio playback of different audio programs synchronously. For example, a user, using a user interface module 13, may enable zone players 11(1) and 11(2) to play one audio program, audio information for which may be provided by, for example, one audio information source 14(1)(1). The same or a different user may, using the same or a different user interface module 13, enable zone players 11 (4) and 11(5) to contemporaneously play another audio program, audio information for which may be provided by a second audio information source, such as audio information source 14(5)(2). Further, a user may enable zone player 11(3) to contemporaneously play yet another audio program, audio information for which may be provided by yet another audio information source, such as audio information source 16(1). As yet another possibility, a user may contemporaneously enable zone player 11(1) to provide audio information from an audio information source connected thereto, such as audio information source 14 (1) (2), to another zone player, such as zone player 11(6) for playback.

[0015] In the following, the term "synchrony group" will be used to refer to a set of one or more zone players that are to play the same audio program synchronously. Thus, in the above example, zone players 11(1) and 11(2) comprise one synchrony group, zone player 11(3) comprises a second synchrony group, zone players 11 (4) and 11(5) comprise a third synchrony group, and zone player 11(6) comprises yet a fourth synchrony group. Thus, while zone players 11(1) and 11(2) are playing the same audio program, they will play the audio program synchronously. Similarly, while zone players 11(4) and 11(5) are playing the same audio program, they will play the audio program synchronously. On the other hand, zone players that are playing different audio programs may do so with unrelated timings. That is, for example, the timing with which zone players 11(1) and 11(2) play their audio program may have no relationship to the timing with which zone player 11 (3), zone players 11(4) and 11(5), and zone player 11(6) play their audio programs. It will be appreciated that, since "synchrony group" is used to refer to sets of zone players that are playing the same audio program

LINKPLAY EXHIBIT 1004
Page 1971 of 4822

synchronously, zone player 11(1) will not be part of zone player 11(6)'s synchrony group, even though zone player 11(1) is providing the audio information for the audio program to zone player 11(6).

[0016] In the network audio system 10, the synchrony groups are not fixed. Users can enable them to be established and modified dynamically. Continuing with the above example, a user may enable the zone player 11(1) to begin providing playback of the audio program provided thereto by audio information source 14(1)(1), and subsequently enable zone player 11(2) to join the synchrony group. Similarly, a user may enable the zone player 11 (5) to begin providing playback of the audio program provided thereto by audio information source 14(5)(2), and subsequently enable zone player 11(4) to join that synchrony group. In addition, a user may enable a zone player to leave a synchrony group and possibly join another synchrony group. For example, a user may enable the zone player 11 (2) to leave the synchrony group with zone player 11(1), and join the synchrony group with zone player 11(6). As another possibility, the user may enable the zone player 11(1) to leave the synchrony group with zone player 11(2) and join the synchrony group with zone player 11(6).In connection with the last possibility, the zone player 11(1) can continue providing audio information from the audio information source 14(1)(1) to the zone player 11(2) for playback thereby.

[0017] A user, using the user interface module 13, can enable a zone player 11(n) that is currently not a member of a synchrony group to join a synchrony group, after which it will be enabled to play the audio program that is currently being played by that synchrony group. Similarly, a user, also using the user interface module 13, can enable a zone player 11(n) that is currently a member of one synchrony group, to disengage from that synchrony group and join another synchrony group, after which that zone player will be playing the audio program associated with the other synchrony group. For example, if a zone player 11(6) is currently not a member of any synchrony group, it, under control of the user interface module 13, can become a member of a synchrony group, after which it will play the audio program being played by the other members of the synchrony group, in synchrony with the other members of the synchrony group. In becoming a member of the synchrony group, zone player 11(6) can notify the zone player that is the master device for the synchrony group that it wishes to become a member of its synchrony group, after which that zone player will also transmit audio information associated with the audio program, as well as timing information, to the zone player 11(6).As the zone player 11(6) receives the audio information and the timing information from the master device, it will play the audio information with the timing indicated by the timing information, which will enable the zone player 11(6) to play the audio program in synchrony with the other zone player(s) in the synchrony group.

[0018] Similarly, if a user, using the user interface module 13, enables a zone player 11(n) associated with a synchrony group to disengage from that synchrony group, and if the zone player 11(n) is not the master device of the synchrony group, the zone player 11(n) can notify the master device, after which the master device can terminate transmission of the audio information and timing information to the zone player 11(n). If the user also enables the zone player 11(n) to begin playing another audio program using audio information from an audio information source 14(n)(s) connected thereto, it will acquire the audio information from the audio information source 14(n)(s) and initiate playback thereof. If the user enables another zone player 11(n') to join the synchrony group associated with zone player I 11(n), operations in connection therewith can proceed as described immediately above.

[0019] As yet another possibility, if a user, using the user interface module 13, enables a zone player 11(n) associated with a synchrony group to disengage from that synchrony group and join another synchrony group, and if the zone player is not the master device of the synchrony group from which it is disengaging, the zone player 11(n) can notify the master device of the synchrony group from which it is disengaging, after which that zone player will terminate transmission of audio information and timing information to the zone player 11(n) that is disengaging. Contemporaneously, the zone player 11(n) can notify the master device of the synchrony group that it (that is, zone player 11(n)) is joining, after which the master device can begin transmission of audio information and timing information to that zone player 11(n). The zone player 11(n) can thereafter begin playback of the audio program defined by the audio information, in accordance with the timing information so that the zone player 11(n) will play the audio program in synchrony with the master device.

[0020] As yet another possibility, a user, using the user interface module 13, may enable a zone player 11(n) that is not associated with a synchrony group, to begin playing an audio program using audio information provided to it by an audio information source 14(n)(s) connected thereto. In that case, the user, also using the user interface module 13 or a user interface device that is specific to the audio information source 14(n)(s), can enable the audio information source 14(n)(s) to provide audio information to the zone player 11(n). After the zone player 11(n) has begun playback, or contemporaneously therewith, the user, using the user interface module 13, can enable other zone players 11(n'), 11(n''),... to join zone player 11(n) 's synchrony group and enable that zone player 11(n) to transmit audio information and timing information thereto as described above, to facilitate synchronous playback of the audio program by the other zone players 11(n'), 11(n'')....

[0021] A user can use the user interface module 13 to control other aspects of the network audio system 10, including but not limited to the selection of the audio information source 14(n)(s) that a particular zone player 11(n) is to utilize, the volume of the audio playback, and so forth. In addition, a user may use the user interface module 13 to turn audio information source(s) 14(n)(s) on and off and to enable them to provide audio information to the respective zone players 11(n).

LINKPLAY EXHIBIT 1004
Page 1972 of 4822

[0022] Operations performed by the various devices associated with a synchrony group will be described in connection with FIG. 2, which schematically depicts a functional block diagram of a synchrony group in the network audio system 10 described above in connection with FIG. 1. With reference to FIG. 2, a synchrony group 20 includes a master device 21 and zero or more slave devices 22(1) through 22 (G) (generally identified by reference numeral 22(g)), all of which synchronously play an audio program provided by an audio information channel device 23. Each of the master device 21, slave devices 22(g) and audio information channel device 23 utilizes a zone player 11(n) depicted in FIG. 1, although it will be clear from the description below that a zone player may be utilized both for the audio information channel device for the synchrony group 20, and the master device 21 or a slave device 22(g) of the synchrony group 20. As will be described below in more detail, the audio information channel device 23 obtains the audio information for the audio program from an audio information source, adds playback timing information, and transmits the combined audio and playback timing information to the master device 21 and slave devices 22(g) over the network 12 for playback. The playback timing information that is provided with the audio information, together with clock timing information provided by the audio information channel device 23 to the various devices 21 and 22(g) as will be described below, enables the master device 21 and slave devices 22(g) of the synchrony group 20 to play the audio information simultaneously.

[0023] The master device 21 and the slave devices 22(g) receive the audio and playback timing information, as well as the clock timing information, that are provided by the audio information channel device 23, and play back the audio program defined by the audio information. The master device 21 is also the member of the synchrony group 20 that communicates with the user interface module 13 and that controls the operations of the slave devices 22(g) in the synchrony group 20. In addition, the master device 21 controls the operations of the audio information channel device 23 that provides the audio and playback timing information for the synchrony group 20. Generally, the initial master device 21 for the synchrony group will be the first zone player 11(n) that a user wishes to play an audio program. However, as will be described below, the zone player 11(n) that operates as the master device 21 can be migrated from one zone player 11(n) to another zone player 11(n'), which preferably will be a zone player that is currently operating as a slave device 22(g) in the synchrony group.

[0024] In addition, under certain circumstances, as will be described below, the zone player 11(n) that operates as the audio information channel device 23 can be migrated from one zone player to another zone player, which also will preferably will be a zone player that is currently operating as a member of the synchrony group 20. It will be appreciated that the zone player that operates as the master device 21 can be migrated to another zone player independently of the migration of the audio information channel device 23. For example, if one zone player 11(n) is operating as both the master device 21 and the audio information channel device 23 for a synchrony group 20, the master device 21 can be migrated to another zone player 11(n') while the zone player 11(n) is still operating as the audio information channel device 23. Similarly, if one zone player 11(n) is operating as both the master device 21 and the audio information channel device 23 for a synchrony group 20, the audio information channel device 23 can be migrated to another zone player 11(n') while the zone player 11(n) is still operating as the master device 21. In addition, if one zone player 11(n) is operating as both the master device 21 and the audio information channel device 23 for a synchrony group 20, the master device 21 can be migrated to another zone player 11(n') and the audio information channel device can be migrated to a third zone player 11(n").

[0025] The master device 21 receives control information from the user interface module 13 for controlling the synchrony group 20 and provides status information indicating the operational status of the synchrony group to the user interface module 13. Generally, the control information from the user interface module 13 enables the master device 21 to, in turn, enable the audio information channel device 23 to provide audio and playback timing information to the synchrony group to enable the devices 21 and 22(g) that are members of the synchrony group 20 to play the audio program synchronously. In addition, the control information from the user interface module 13 enables the master device 21 to, in turn, enable other zone players to join the synchrony group as slave devices 22(g) and to enable slave devices 22(g) to disengage from the synchrony group. Control information from the user interface module 13 can also enable the zone player 11(n) that is currently operating as the master device 21 to disengage from the synchrony group, but prior to doing so that zone player will enable the master device 21 to transfer from that zone player 11(n) to another zone player 11(n'), preferably to a zone player 11(n') that is currently a slave device 22(g) in the synchrony group 20. The control information from the user interface module 13 can also enable the master device 21 to adjust its playback volume and to enable individual ones of the various slave devices 22(g) to adjust their playback volumes. In addition, the control information from the user interface module 13 can enable the synchrony group 20 to terminate playing of a current track of the audio program and skip to the next track, and to re-order tracks in a play list of tracks defining the audio program that is to be played by the synchrony group 20.

[0026] The status information that the master device 21 may provide to the user interface module 13 can include such information as a name or other identifier for the track of the audio work that is currently being played, the names or other identifiers for upcoming tracks, the identifier of the zone player 11(n) that is currently operating as the master device 21, and identifiers of the zone players that are currently operating as slave devices 22(g). In one embodiment, the user interface module 13 includes a display (not separately shown) that can display the status information to the user.

LINKPLAY EXHIBIT 1004
Page 1973 of 4822

[0027] It will be appreciated that the zone player 11(n) that is operating as the audio information channel device 23 for one synchrony group may also comprise the master device 21 or any of the slave devices 22(g) in another synchrony group. This may occur if, for example, the audio information source that is to provide the audio information that is to be played by the one synchrony group is connected to a zone player also being utilized as the master device or a slave device for the other synchrony group. This will be schematically illustrated below in connection with FIG. 2A. Since, as noted above, the zone player 11(n) that is operating as the audio information channel device 23 for the synchrony group 20 may also be operating as a master device 21 or slave device 22(g) for another synchrony group, it can also be connected to one or more audio reproduction devices 15(n)(r), although that is not depicted in FIG. 2. Since the master device 21 and slave devices 22(g) are all to provide playback of the audio program, they will be connected to respective audio reproduction devices 15(n)(r). Furthermore, it will be appreciated that one or more of the zone players 11(n) that operate as the master device 21 and slave devices 22(g) in synchrony group 20 may also operate as an audio information channel device for that synchrony group or for another synchrony group and so they may be connected to one or more audio information sources 14(n)(s), although that is also not depicted in FIG. 2. In addition, it will be appreciated that a zone player 11(n) can also operate as a audio information channel device 23 for multiple synchrony groups.

[0028] If the audio information channel device 23 does not utilize the same zone player as the master device 21, the master device 21 controls the audio information channel device by exchanging control information over the network 12 with the audio information channel device 23. The control information is represented in FIG. 2 by the arrow labeled CHAN_DEV_CTRL_INFO. The control information that the master device 21 provides to the audio information channel device 23 will generally depend on the nature of the audio information source that is to provide the audio information for the audio program that is to be played and the operation to be enabled by the control information. If, for example, the audio information source is a conventional compact disc, tape, or record player, broadcast radio receiver, or the like, which is connected to a zone player 11(n), the master device 21 may merely enable the zone player serving as the audio information channel device 23 to receive the audio information for the program from the audio information source. It will be appreciated that, if the audio information is not in digital form, the audio information channel device 23 will convert it to digital form and provide the digitized audio information, along with the playback timing information, to the master device 21 and slave devices 22(g).

[0029] On the other hand, if the audio information source is, for example, a digital data storage device, such as may be on a personal computer or similar device, the master device 21 can provide a play list to the audio information channel device 23 that identifies one or more files containing the audio information for the audio program. In that case, the audio information channel device 23 can retrieve the files from the digital data storage device and provide them, along with the playback timing information, to the master device 21 and the slave devices 22(g). It will be appreciated that, in this case, the audio information source may be directly connected to the audio information channel device 23, as, for example, an audio information source 14(n)(s), or it may comprise an audio information source 16(m) connected to the network 12. As a further alternative, if the audio information source is a source available over the wide area network, the master device 21 can provide a play list comprising a list of web addresses identifying the files containing the audio information for the audio program that is to be played, and in that connection the audio information channel device 23 can initiate a retrieval of the files over the wide area network. As yet another alternative, if the audio information source is a source of streaming audio received over the wide area network, the master device 21 can provide a network address from which the streaming audio can be received. Other arrangements by which the master device 21 can control the audio information channel device 23 will be apparent to those skilled in the art.

The master device 21 can also provide control information to the synchrony group's audio information channel device 23 to enable a migration from one zone player 11(n) to another zone player 11(n'). This may occur if, for example, the audio information source is one of audio information sources 16 or a source accessible over the wide area network via the network 12. The master device 21 can enable migration of the audio information channel device 23 for several reasons, including, for example, to reduce the loading of the zone player 11(n), to improve latency of message transmission in the network 12, and other reasons as will be appreciated by those skilled in the art.

[0030] As noted above, the audio information channel device 23 provides audio and playback timing information for the synchrony group to enable the master device 21 and slave devices 22(g) to play the audio program synchronously. Details of the audio and playback timing information will be described in detail below in connection with FIGS. 3 and 4, but, in brief, the audio information channel device 23 transmits the audio and playback timing information in messages over the network 12 using a multi-cast message transmission methodology. In that methodology, the audio information channel device 23 will transmit the audio and playback timing information in a series of messages, with each message being received by all of the zone players 11(n) comprising the synchrony group 20, that is, by the master device 21 and the slave devices 22(g). Each of the messages includes a multi-cast address, which the master device 21 and slave devices 22(g) will monitor and, when they detect a message with that address, they will receive and use the contents of the message. The audio and playback timing information is represented in FIG. 2 by the arrow labeled "AUD+PBTIME INFO," which has a single tail, representing a source for the information at the audio information channel device 23, and multiple arrowheads representing the destinations of the information, with one arrowhead extending to the master device

LINKPLAY EXHIBIT 1004
Page 1974 of 4822

21 and other arrowheads extending to each of the slave devices 22(g) in the synchrony group 20. The audio information channel device 23 may make use of any convenient multi-cast message transmission methodology in transmitting the audio and playback timing information to the synchrony group 20. As will be described in detail in connection with FIG. 4, the audio and playback timing information is in the form of a series of frames, with each frame having a time stamp. The time stamp indicates a time, relative to the time indicated by a clock maintained by the audio information channel device 23, at which the frame is to be played. Depending on the size or sizes of the messages used in the selected multi-cast message transmission methodology and the size or sizes of the frames, a message may contain one frame, or multiple frames, or, alternatively, a frame may extend across several messages.

[0031] The audio information channel device 23 also provides clock time information to the master device 21 and each of the slave devices 22(g) individually over network 12 using a highly accurate clock time information transmission methodology. The distribution of the clock time information is represented in FIG. 2 by the arrows labeled "AICD_CLK_INF (M)" (in the case of the clock time information provided to the master device 21) and "AICD_CLK_INF (S_1)" through "AICD_CLK_INF (S_G)" (in the case of audio information channel device clock information provided to the slave devices 22(g)). In one embodiment, the master device 21 and slave devices 22(g) make use of the well-known SNTP (Simple Network Time Protocol) to obtain current clock time information from the audio information channel device 23. The SNTP makes use of a unicast message transfer methodology, in which one device, such as the audio information channel device 23, provides clock time information to a specific other device, such as the master device 21 or a slave device 22(g), using the other device's network, or unicast, address. Each of the master device 21 and slave devices 22(g) will periodically initiate SNTP transactions with the audio information channel device 23 to obtain the clock time information from the audio information channel device 23. As will be described below in more detail, the master device 21 and each slave device 22(g) make use of the clock time information to determine the time differential between the time indicated by the audio information channel device's clock and the time indicated by its respective clock, and use that time differential value, along with the playback time information associated with the audio information and the respective device's local time as indicated by its clock to determine when the various frames are to be played. This enables the master device 21 and the slave devices 22(g) in the synchrony group 20 to play the respective frames simultaneously.

[0032] As noted above, the control information provided by the user to the master device 21 through the user interface module 13 can also enable the master device 21 to, in turn, enable another zone player 11(n') to join the synchrony group as a new slave device 22(g). In that operation, the user interface module 13 will provide control information, including the identification of the zone player 11(n') that is to join the synchrony group to the master device 21. After it receives the identification of the zone player 11(n') that is to join the synchrony group, the master device 21 will exchange control information, which is represented in FIG. 2 by the arrows labeled SLV_DEV_CTRL_INF (S_1) through SLV_DEV_CTRL_INF (S_G) group slave control information, over the network 12 with the zone player I 11(n') that is identified in the control information from the user interface module 13. The control information that the master device 21 provides to the new zone player 11(n') includes the network address of the zone player 11(n) that is operating as the audio information channel device 23 for the synchrony group, as well as the multi-cast address that the audio information channel device 23 is using to broadcast the audio and playback timing information over the network. The zone player that is to operate as the new slave device 22(g') uses the multi-cast address to begin receiving the multi-cast messages that contain the audio information for the audio program being played by the synchrony group.

[0033] It will be appreciated that, if the zone player 11(n) that is operating as the master device 21 for the synchrony group 20 is also operating the audio information channel device 23, and if there are no slave devices 22(g) in the synchrony group 20, the audio information channel device 23 may not be transmitting audio and playback timing information over the network. In that case, if the new slave device 22(g') is the first slave device in the synchrony group, the zone player 11(n) that is operating as both the master device 21 and audio information channel device 23, can begin transmitting the audio and playback timing information over the network 12 when the slave device 22(g') is added to the synchrony group 20. The zone player 11(n) can maintain a count of the number of slave devices 22(g) in the synchrony group 20 as they join and disengage, and, if the number drops to zero, it can stop transmitting the audio and playback timing information over the network 12 to reduce the message traffic over the network 12.

[0034] The new slave device 22(g') added to the synchrony group 20 uses the network address of the audio information channel device 23 for several purposes. In particular, the new slave device 22(g') will, like the master device 21 (assuming the zone player 11(n) operating as the master device 21 is not also the audio information channel device 23), engage in SNTP transactions with the audio information channel device 23 to obtain the clock timing information from the audio information channel device 23. In addition, the new slave device 22(g') can notify the audio information channel device 23 that it is a new slave device 22(g') for the synchrony group 20 and provide the audio information channel device 23 with its network address. As will be described below, in one embodiment, particularly in connection with audio information obtained from a source, such as a digital data storage device, which can provide audio information at a rate that is faster than the rate at which it will be played, the audio information channel device 23 will buffer audio and timing information and broadcast it over the network 12 to the synchrony group 20 generally at a rate at which it is provided by the source. Accordingly, when a new slave device 22(g') joins the synchrony group 20, the playback timing information may indicate

LINKPLAY EXHIBIT 1004
Page 1975 of 4822

that the audio information that is currently being broadcast by the audio information channel device 23 using the multicast methodology is to be played back some time in the future. To reduce the delay with which the new slave device 22(g') will begin playback, the audio information channel device 23 can also retransmit previously transmitted audio and timing information that it had buffered to the new slave device 22(g') using the unicast network address of the slave device 22(g').

[0035] The master device 21 can also use the slave device control information exchanged with the slave devices 22(g) for other purposes. For example, the master device 21 can use the slave device control information to initiate a migration of the master from its zone player 11(n) to another zone player 11(n'). This may occur for any of a number of reasons, including, for example, that the master device 21 is terminating playback by it of the audio program and is leaving the synchrony group 20, but one or more of the other devices in the synchrony group is to continue playing the audio program. The master device 21 may also want to initiate a migration if it is overloaded, which can occur if, for example, the zone player 11(n) that is the master device 21 for its synchrony group is also operating as an audio information channel device 23 for another synchrony group.

[0036] The user can also use the user interface module 13 to adjust playback volume by the individual zone players 11(n) comprising the synchrony group. In that operation, the user interface module 13 provides information identifying the particular device whose volume is to be adjusted, and the level at which the volume is to be set to the master device 21. If the device whose volume is to be adjusted is the master device 21, the master device 21 can adjust its volume according to the information that it receives from the user interface module 13. On the other hand, if the device whose volume is to be adjusted is a slave device 22(g), the master device 21 can provide group slave control information to the respective slave device 22(g), to enable it to adjust its volume.

[0037] The user can also use the user interface module 13 to enable a synchrony group 20 to cancel playing of the track in an audio program that is currently being played, and to proceed immediately to the next track. This may occur, for example, if the tracks for the program is in the form of a series of digital audio information files, and the user wishes to cancel playback of the track that is defined by one of the files. In that case, when the master device 21 receives the command to cancel playback of the current track, it will provide channel device control information to the audio information channel device 23 so indicating. In response, the audio information channel device 23 inserts control information into the audio and playback timing information, which will be referred to as a "resynchronize" command. In addition, the audio information channel device 23 will begin transmitting audio information for the next track, with timing information to enable it to be played immediately. The resynchronize command can also enable playback of a track to be cancelled before it has been played. Details of these operations will be described below.

[0038] As noted above, there may be multiple synchrony groups in the network audio system 10, and further that, for example, a zone player 11(n) may operate both as a master device 21 or a slave device 22(g) in one synchrony group, and as the audio information channel device 23 providing audio and playback timing information and clock timing information for another synchrony group. An illustrative arrangement of this will be described in connection with FIG. 2A. With reference to FIG. 2A, that FIG. depicts elements of two synchrony groups, identified by reference numerals 20(1) and 20(2), respectively. For clarity, FIG. 2A does not show a number of elements, the presence of which would be evident from FIGS. 1 and 2 as described above. For example, FIG. 2A does not depict the audio information sources from which audio information is obtained for the synchrony groups or the audio reproduction devices that are used to produce sound for the master and slave devices, which are depicted in both FIGS. 1 and 2. In addition, FIG. 2A does not depict arrows that represent control information provided by the respective master devices to the slave devices in the respective synchrony groups, or to the audio information channel devices that provide audio and timing information for the respective synchrony groups, which are depicted in FIG. 2. In addition, FIG.2A does not depict the arrows that represent the clock timing information provided by the audio information channel devices to the respective members of the respective synchrony groups, which are also depicted in FIG. 2. As will be noted below, however, FIG. 2A does depict arrows representing the audio and playback timing information provided by the respective audio information channel devices for the respective synchrony groups 20(1), 20(2), to the master and slave devices comprising the respective synchrony groups 20(1), 20(2).

[0039] Each synchrony group 20(1), 20(2) comprises elements of a number of zone players. A functional block diagram of a zone player will be described below in connection with FIG. 3. Synchrony group 20(1) includes a master device 21(1) and "K" slave devices 22(1)(1) through 22(K)(1) (the index "1" in reference numeral 21(1) and the last index in reference numeral 22(1)(1) through 21(K)(1) corresponds to the index of the synchrony group 20(1) to which they belong) utilize zone players 11(1) and 11(K+1) respectively. Similarly, synchrony group 20(2) includes a master device 21(2) and "L" slave devices 22(1)(2) through 22(L)(2) that utilize zone players 11(K+2) through 11(K+L+2). In the illustrative arrangement depicted in FIG. 2A, both synchrony groups 20(1) and 20(2) are controlled by the user interface module 13, which can provide control information to, and receive status information from, the master devices 21(1) and 21(2) independently. It will be appreciated that separate user interface modules may be provided to provide control information to, and receive status information from, the respective master devices 21(1),21(2).

[0040] As noted above, the slave device 22(1)(2) in synchrony group 20(2) utilizes zone player 11(K+3). In the illustrative

LINKPLAY EXHIBIT 1004
Page 1976 of 4822

arrangement depicted in FIG. 2A, the audio information channel device 23(1) that provides audio and playback timing information to the master and slave devices 21(1), 22(1)(1),..., 22(K)(1) of synchrony group 20(1) also utilizes zone player 11(K+3). As noted above, this may occur if, for example, the audio information source that is to provide audio information to be played by the synchrony group 20(1) is connected to the zone player 11(K+3). Thus, when the master device 21(1) of synchrony group 20(1) exchanges channel device control information with the audio information channel device 23(1), it is effectively exchanging channel device control information with the zone player 11(K+3). Similarly, when the master and slave devices 21(1), 22(1)(1),..., 22(K)(1) of synchrony group 20(1) receive audio and playback timing information, as well as clock timing information, from the audio information channel device 23(1), they are effectively receiving the information from the zone player 11(K+3). FIG.2A depicts a multi-headed arrow representing audio and playback timing information transmitted by the zone player 11(K+3), as audio information channel device 23(1), to the master and slave devices 21(1), 22(1)(1),...,11(K)(1) comprising synchrony group 20(1).

[0041] On the other hand, in the illustrative arrangement depicted in FIG. 2A, the synchrony group 20(2) utilizes a zone player 11(K+L+3) as its audio information channel device 23(2). As with synchrony group 20(1), when the master device 21(2) of synchrony group 20(2) exchanges channel device control information with the audio information channel device 23(2), it is effectively exchanging channel device control information with the zone player 11(K+L+3). Similarly, when the master and slave devices 21(2), 22(1)(2),..., 22(L)(2) of synchrony group 20(2) receive audio and playback timing information, as well as clock timing information, from the audio information channel device 23(2), they are effectively receiving the information from the zone player 11(K+L+3). FIG. 2A depicts a multi-headed arrow representing audio and playback timing information transmitted by the zone player 11(K+3) as audio information channel device 23(2) to the master and slave devices 21(2), 22(1) (2),..., 22(L) (2) comprising synchrony group 20(2).

[0042] In the illustrative arrangement depicted in FIG. 2A, zone player 11(K+L+3), which is the audio information channel device 23(2) for synchrony group 20(2), is not shown as being either a master or a slave device in another synchrony group. However, it will be appreciated that zone player 11(K+L+3) could also be utilized as the master device or a slave device for another synchrony group. Indeed, it will be appreciated that the zone player that is utilized as the audio information channel device for synchrony group 20(2) may also be a zone player that is utilized as the master device 21(1) or a slave device 22(1) (1),..., 22(K)(1) in the synchrony group 20(1).

[0043] A zone player 11(n) that is utilized as a member of one synchrony group may also be utilized as the audio information channel device for another synchrony group if the audio information source that is to supply the audio information that is to be played by the other synchrony group is connected to that zone player 11(n). A zone player 11(n) may also be utilized as the audio information channel device for the other synchrony group if, for example, the audio information source is an audio information source 16(m) (FIG.1) that is connected to the network 12 or an audio information source that is available over a wide area network such as the Internet. The latter may occur if, for example, the zone player 11(n) has sufficient processing power to operate as the audio information channel device and it is in an optimal location in the network 12, relative to the zone players comprising the other synchrony group (that is the synchrony group for which it is operating as audio information channel device) for providing the audio and playback timing information to the members of the other synchrony group. Other circumstances under which the zone player 11(n) that is utilized as a member of one synchrony group may also be utilized as the audio information channel device for another synchrony group will be apparent to those skilled in the art.

[0044] As was noted above, the master device 21 for a synchrony group 20 may be migrated from one zone player 11(n) to another zone player 11 (n'). As was further noted above, the audio information channel device 23 for a synchrony group 20 may be migrated from one zone player 11(n) to another zone player 11(n'). It will be appreciated that the latter may occur if, for example, the audio information source that provides the audio program for the synchrony group is not connected to the zone player 11(n) that is operating as the audio information channel device 23, but instead is one of the audio information sources 16(m) connected to the network 12 or a source available over a wide area network such as the Internet. Operations performed during a migration of an audio information channel device 23 from one zone player 11(n) to another zone player 11(n') will generally depend on the nature of the audio information that is being channeled by the audio information channel device 23. For example, if the audio information source provides streaming audio, the zone player 11(n) that is currently operating as the audio information channel device 23 for the synchrony group 20, can provide the following information to the other zone player 11(n') that is to become the audio information channel device 23 for the synchrony group 20:

(a) the identification of the source of the streaming audio information,
(b) the time stamp associated with the frame that the zone player 11(n) is currently forming, and
(c) the identifications of the zone players that are operating as the master device 21 and slave devices 22(g) comprising the synchrony group 20.

After the zone player 11(n') receives the information from the zone player 11(n), it will begin receiving the streaming

LINKPLAY EXHIBIT 1004
Page 1977 of 4822

audio from the streaming audio information source identified by the zone player I 11(n), assemble the streaming audio information into frames, associate each frame with a time stamp, and transmit the resulting audio and playback timing information over the network 12. The zone player 11(n') will perform these operations in the same manner as described above, except that, instead of using the time indicated by its digital to analog converter clock 34 directly in generating the time stamps for the frames, the initial time stamp will be related to the value of the time stamp that is provided by the zone player 11(n) (reference item (b) above), with the rate at which the time stamps are incremented corresponding to the rate at which its (that is, the zone player 11(n')'s) clock increments. In addition, the zone player 11(n') will notify the zone players that are operating as the master device 21 and slave devices 22(g) of the synchrony group 20 that it is the new audio information channel device 23 for the synchrony group 20, and provide the multi-cast address that it will be using to multi-cast the audio and playback timing information, as well as its unicast network address. After the members of the synchrony group 20 receive the notification from the zone player 11(n') indicating that it is the new audio information channel device 23 for the synchrony group 20, they will receive the audio and playback timing information from the zone player 11(n') instead of the zone player 11(n), using the multi-cast address provided by the zone player 11(n'). In addition, they can utilize the zone player 11(n')'s unicast network address to obtain current time information therefrom. It will be appreciated that the zone player 11(n') will determine its current time in relation to the time stamp that is provided by the zone player 11(n) (reference item (b) above) or the current time information that it received from the zone player 11(n) using the SNTP protocol as described above.

[0045] Generally similar operations can be performed in connection with migrating the audio information channel device from one zone player 11(n) to another zone player 11(n') if the audio information is from one or more audio information files, such as may be the case if the audio information comprises MP3 or WAV files that are available from sources such as sources 16(m) connected to the network 12 or over from sources available over a wide area network such as the Internet, except for differences to accommodate the fact that the audio information is in files. In that case, the zone player 11(n) that is currently operating as the audio information channel device 23 for the synchrony group 20 can provide the following information to the zone player 11(n') that is to become the audio information channel device 23 for the synchrony group 20:

(d) a list of the audio information files containing the audio information that is to be played;
(e) the identification of the file for which the zone player 11(n) is currently providing audio and playback timing information, along with the offset into the file for which the current item of audio and playback timing information is being generated and the time stamp that the zone player 11(n) is associating with that frame, and
(f) the identifications of the zone players that comprise the master device 21 and slave devices 22(g) comprising the synchrony group 20.

After the zone player 11(n') receives the information from the zone player 11(n), it will begin retrieving audio information from the file identified in item (e), starting at the identified offset. In addition, the zone player 11(n') can assemble the retrieved audio information into frames, associate each frame with a time stamp and transmit the resulting audio and playback timing information over the network 12. The zone player 11(n') will perform these operations in the same manner as described above, except that, instead of using the time indicated by its digital to analog converter clock 34 directly in generating the time stamps for the frames, the value of the initial time stamp will be related to the time stamp that is provided by the zone player 11(n) (reference item (e) above), with the rate at which the time stamps are incremented corresponding to the rate at which its (that is, the zone player 11(n') 's) clock increments. In addition, the zone player 11(n') will notify the zone players that are operating as the master device 21 and slave devices 22(g) of the synchrony group 20 that it is the new audio information channel device 23 for the synchrony group 20, and provide the multi-cast address that it will be using to multi-cast the audio and playback timing information, as well as its unicast network address. After the members of the synchrony group 20 receive the notification from the zone player 11(n') indicating that it is the new audio information channel device 23 for the synchrony group 20, they will receive the audio and playback timing information from the zone player I 11(n') instead of the zone player I 11(n), using the multi-cast address provided by the zone player 11(n'). In addition, they can utilize the zone player 11(n')'s unicast network address to obtain current time information therefrom. It will be appreciated that the zone player I 11(n') will determine its current time in relation to the time stamp that is provided by the zone player 11(n) (reference item (b) above) or the current time information that it received from the zone player 11(n) using the SNTP protocol as described above. The zone player 11(n') will process successive audio information files in the list that it receives from the zone player 11(n) (reference item (d)).

[0046] Operations performed by the zone players 11(n) and 11(n') in connection with migration of the audio information channel device 23 for other types of audio information will be apparent to those skilled in the art. In any case, preferably, the zone player 11(n) will continue operating as an audio information channel device 23 for the synchrony group 20 for at least a brief time after it notifies the zone player 11(n') that it is to become audio information channel device for the synchrony group, so that the zone player 11(n') will have time to notify the zone players in the synchrony group 20 that it is the new audio information channel device 23 for the synchrony group.

LINKPLAY EXHIBIT 1004
Page 1978 of 4822

[0047] Before proceeding further in describing operations performed by the network audio system 10, it would be helpful to provide a detailed description of a zone player 11(n) constructed in accordance with the invention. FIG. 3 depicts a functional block diagram of a zone player 11(n) constructed in accordance with the invention. All of the zone players in the network audio system 10 may have similar construction. With reference to FIG. 3, the zone player 11(n) includes an audio information source interface 30, an audio information buffer 31, a playback scheduler 32, a digital to analog converter 33, an audio amplifier 35, an audio reproduction device interface 36, a network communications manager 40, and a network interface 41, all of which operate under the control of a control module 42. The zone player 11(n) also has a device clock 43 that provides timing signals that control the general operations of the zone player 11(n). In addition, the zone player 11(n) includes a user interface module interface 44 that can receive control signals from the user interface module 13 (FIGS. 1 and 2) for controlling operations of the zone player 11(n), and provide status information to the user interface module 13.

[0048] Generally, the audio information buffer 31 buffers audio information, in digital form, along with playback timing information. If the zone player 11(n) is operating as the audio information channel device 23 (FIG. 2) for a synchrony group 20, the information that is buffered in the audio information buffer 31 will include the audio and playback timing information that will be provided to the devices 21 and 22(g) in the synchrony group 20. If the zone player 11(n) is operating as the master device 21 or a slave device 22(g) for a synchrony group, the information that is buffered in the audio information buffer 31 will include the audio and playback timing information that the zone player 11(n) is to play.

[0049] The audio information buffer 31 can receive audio and playback timing information from two sources, namely, the audio information source interface 30 and the network communications manager 40. In particular, if the zone player 11(n) is operating as the audio information channel device 23 for a synchrony group 20, and if the audio information source is a source 14(n)(s) connected to the zone player I 11(n), the audio information buffer 31 will receive and buffer audio and playback timing information from the audio information source interface 30. On the other hand, if the zone player 11(n) is operating as the audio information channel device 23 for a synchrony group 20, and if the audio information source is a source 16(m) connected to the network 12, or a source available over the wide area network, the audio information buffer 31 will receive and buffer audio and playback timing information from the network communications manager 40. It will be appreciated that, if the zone player 11(n) is not a member of the synchrony group, the zone player 11(n) will not play this buffered audio and playback timing information.

[0050] On yet another hand, if the zone player 11(n) is operating as the master device 21 or a slave device 22(g) in a synchrony group, and if the zone player 11(n) is not also the audio information channel device 23 providing audio and playback timing information for the synchrony group 20, the audio information buffer 31 will receive and buffer audio and playback timing information from the network communications manager 40.

[0051] The audio information source interface 30 connects to the audio information source(s) 14(n)(s) associated with the zone player 11(n). While the zone player 11(n) is operating as audio information channel device 23 for a synchrony group 20, and if the audio information is to be provided by a source 14(n)(s) connected to the zone player 11(n), the audio information source interface 30 will selectively receive audio information from one of the audio information source(s) 14(n)(s) to which the zone player is connected and store the audio information in the audio information buffer 21. If the audio information from the selected audio information source 14(n)(s) is in analog form, the audio information source interface 30 will convert it to digital form. The selection of the audio information source 14(n)(s) from which the audio information source interface 30 receives audio information is under control of the control module 42, which, in turn, receives control information from the user interface module through the user interface module interface 44. The audio information source interface 30 adds playback timing information to the digital audio information and buffers the combined audio and playback timing information in the audio information buffer 21.

[0052] More specifically, as noted above, the audio information source interface 30 receives audio information from an audio information source 14(n)(s), converts it to digital form if necessary, and buffers it along with playback timing information in the audio information buffer 21. In addition, the audio information source interface 30 will also provide formatting and scheduling information for the digital audio information, whether as received from the selected audio information source 14(n)(s) or as converted from an analog audio information source. As will be made clear below, the formatting and scheduling information will control not only playback by the zone player 11(n) itself, but will also enable other zone players 11(n'), 11(n"),... that may be in a synchrony group for which the zone player 11(n) is the master device, to play the audio program associated with the audio information in synchrony with the zone player 11(n).

[0053] In one particular embodiment, the audio information source interface 30 divides the audio information associated with an audio work into a series of frames, with each frame comprising digital audio information for a predetermined period of time. As used herein, an audio track may comprise any unit of audio information that is to be played without interruption. On the other hand, an audio program may comprise a series of one or more audio tracks that are to be played in succession. It will be appreciated that the tracks comprising the audio program may also be played without interruption, or alternatively playback between tracks maybe interrupted by a selected time interval. FIG. 4 schematically depicts an illustrative framing strategy used in connection with one embodiment of the invention for a digital audio stream comprising an audio work. More specifically, FIG.4 depicts a framed digital audio stream 50 comprising a sequence of

LINKPLAY EXHIBIT 1004
Page 1979 of 4822

frames 51(1) through 51(F) (generally identified by reference numeral 51(f)). Each frame 51(f), in turn, comprises a series of audio samples 52(f)(1) through 52(f)(S) (generally identified by reference numeral 52(f)(s)) of the audio track. Preferably all of the frames will have the same number "S" of audio samples, although it will be appreciated from the following that that is primarily for convenience. On the other hand, it will be appreciated that, the number of audio samples may differ from "S"; this may particularly be the case if the frame 51(f) contains the last audio samples for the digital audio stream for a particular audio work. In that case, the last frame 51(F) will preferably contain samples 52(F)(1) through 52(F)(x), where "x" is less than "S." Generally, it is desirable that the number of samples be consistent among all frames 51(f), and in that case padding, which will not be played, can be added to the last frame 51(F).

[0054] Associated with each frame 51(f) is a header 55(f) that includes a number of fields for storing other information that is useful in controlling playback of the audio samples in the respective frame 51(f). In particular, the header 55(f) associated with a frame 51(f) includes a frame sequence number field 56, an encoding type field 57, a sampling rate information field 58, a time stamp field 60, an end of track flag 61, and a length flag field 62. The header 55(f) may also include fields (not shown) for storing other information that is useful in controlling playback. Generally, the frame sequence number field 56 receives a sequence number "f" that identifies the relative position of the frame 51(f) in the sequence of frames 51(1)..51(f)...51(F) containing the digital audio stream 50. The encoding type field 57 receives a value that identifies the type of encoding and/or compression that has been used in generating the digital audio stream. Conventional encoding or compression schemes include, for example, the well-known MP3 and WAV encoding and/or compression schemes, although it will be appreciated that other schemes may be provided for as well. The sampling rate information field 58 receives sampling rate information that indicates the sampling rate for the audio samples 52(f)(s). As will be apparent to those skilled in the art, the sampling rate determines the rate at which the zone player 11(n) is to play the audio samples 52(f)(s) in the frame, and, as will be described below, determines the period of the digital to analog converter clock 34.

[0055] The condition of the end of work flag 61 indicates whether the frame 51(f) contains the last digital audio samples for the audio track associated with the framed digital audio work 50. If the frame 51(f) does not contain the audio samples that are associated with the end of the digital audio stream 50 for a respective audio work, the end of work flag will be clear. On the other hand, if the frame 51(f) does contain the audio samples that are associated with the end of the digital audio stream 50 for a respective audio work, the end of work flag 61 will be set. In addition, since the number of valid audio samples 52(F)(s) in the frame 51(F), that is, the samples that are not padding, may be less than "S," the default number of audio samples in a frame 51(f), the length flag field 62 will contain a value that identifies the number of audio samples in 52(F)(s) in the last frame 51(F) of the audio work 50. If, as noted above, the frames have a consistent number "S" of samples, the samples 52(F)(x+1) through 52(F)(S) will contain padding, which will not be played.

[0056] The time stamp field 60 stores a time stamp that identifies the time at which the zone player 11(n) is to play the respective frame. More specifically, for each frame of a framed digital audio stream 50 that is buffered in the audio information buffer 21, the audio information source interface 30, using timing information from the digital to analog converter clock 34, will determine a time at which the zone player 11(n) is to play the respective frame, and stores a time stamp identifying the playback time in the time stamp field 60. The time stamp associated with each frame will later be used by the playback scheduler 32 to determine when the portion of the digital audio stream stored in the frame is to be coupled to the digital to analog converter 33 to initiate play back. It will be appreciated that the time stamps that are associated with frames in sequential frames 51(1), 51(2),..., 51(F), will be such that they will be played back in order, and without an interruption between the sequential frames comprising the digital audio stream 50. It will further be appreciated that, after a time stamp has been determined for the first frame, stored in frame 51(1), of a digital audio stream 50, the audio information source interface 30 can determine time stamps for the subsequent frame 51(2), 51(3),...., 51 (F) in relation to the number of samples "S" in the respective frames and the sample rate. The time stamps will also preferably be such that frames will be played back after some slight time delay after they have been buffered in the audio information buffer 21; the purpose for the time delay will be made clear below.

[0057] Returning to FIG. 3, in addition to dividing the digital audio information into frames, the audio information source interface 30 also aggregates and/or divides the frames 51(f) as necessary into packets, each of which will be of a length that would fit into a message for transmission over the network, and associates each packet with a packet sequence number. For example, if a packet will accommodate multiple frames 51(f), 51(f+1),... 51(f+y-1), it will aggregate them into a packet and associate them with a packet number, for example p(x). If the entire frames 51(f) and 51(f+y-1) was accommodated in packet p(x), where "x" is the sequence number, which will occur if the size of a packet is an exact multiple of the frame size, the next packet, p(x+1) will begin with frame 51(f+y) and will include frames 51(f+y),..., 51(f+2y-1). Subsequent packets p(x+2),.... will be formed in a similar manner. On the other hand, if the packet length will not accommodate an exact multiple of the frame size, the last frame in the packet will be continued at the beginning of the next packet.

[0058] If the audio information source interface 30 is aware of track boundaries, which may be the case if the tracks are divided into files, the packets will reflect the track boundaries, that is, the packets will not contain frames from two tracks. Thus, if the last frames associated with a track are insufficient to fill a packet, the packet will contain padding

LINKPLAY EXHIBIT 1004
Page 1980 of 4822

from the last frame associated with the track to the end of the packet, and the next packet will begin with the first frames associated with the next track.

[0059] In one embodiment, the audio information source interface 30 stores the packets in the audio information buffer 31 in a ring buffer. As is conventional, a ring buffer includes a series of storage locations in the buffer. Each entry will be sufficient to store one packet. Four pointers are used in connection with the ring buffer, a first pointer pointing to the beginning of the ring buffer, a second pointer pointing to the end of the ring buffer, an third "write" pointer pointing to the entry into which a packet will be written and a fourth "read" pointer pointing to the entry from which packet will be read for use in playback. When a packet is read from the ring buffer for playback, it will be read from the entry pointed to by the read pointer. After the packet has been read, the read pointer will be advanced. If the read pointer points beyond the end of the ring buffer, as indicated by the end pointer, it will be reset to point to the entry pointed to by the beginning pointer, and the operations can be repeated.

[0060] On the other hand, when the audio information source interface 30 stores a packet in the ring buffer, first determine whether the entry pointed to by the write pointer points to the same entry as the entry pointed to by the read pointer. If the write pointer points to the same entry as the entry pointed to by the read pointer, the entry contains at least a portion of a packet that has not yet been read for playback, and the audio information source interface 30 will delay storage of the packet until the entire packet has been read and the read pointer advanced. After the read pointer has been advanced, the audio information source interface 30 can store the packet in the entry pointed to by the write pointer. After the packet has been stored, the audio information source interface 30 will advance the write pointer. If the write pointer points beyond the end of the ring buffer, as indicated by the end pointer, it will be reset to point to the entry pointed to by the beginning pointer, and the operations can be repeated.

[0061] As noted above, the zone player 11(n) can operate both as an audio information channel device 23 and a member of the synchrony group 20 of which it is a member. In that case, the audio information buffer 31 can contain one ring buffer. On the other hand, the zone player 11(n) can operate as an audio information channel device 23 for one synchrony group 20(1) (FIG. 2A) and a member of another synchrony group 20(2). In that case, the audio information buffer 31 would maintain two ring buffers, one for the audio and timing information associated with synchrony group 20(1), and the other for the audio and timing information associated with synchrony group 20(2). It will be appreciated that, in the latter case, the zone player 11(n) will only use the audio and timing information that is associated with synchrony group 20(2) for playback.

[0062] The playback scheduler 32 schedules playback of the audio information that is buffered in the audio information buffer 31 that is to be played by the zone player 11(n). Accordingly, under control of the playback scheduler 32, the digital audio information that is buffered in the audio information buffer 21 that is to be played by the zone player 11(n) is transferred to the digital to analog converter 33 for playback. As noted above, if the zone player 11(n) is operating as an audio information channel device 23 for a synchrony group 20 for which it is not a member, the playback scheduler 32 will not schedule the digital audio information that is to be played by that synchrony group 20 for playback. The playback scheduler 32 only schedules the digital audio information, if any, that is buffered in the audio information buffer 31 that is associated with a synchrony group for which the zone player 11(n) is a member, whether as master device 21 or a slave device 22(g).

[0063] Essentially, the playback scheduler 32 makes use of the read pointer associated with the circular buffer that contains the audio and playback timing information that is to be played by the zone player 11(n). The playback scheduler 32 retrieves the packet information from the entry of the ring buffer pointed to by the read pointer, and then advances the ring pointer as described above. The playback scheduler 32 determines the boundaries of the frames in the packet and uses the time stamps in the time stamp fields 60 associated with the respective frame 51(f), along with timing information provided by the zone player 11(n)'s digital to analog converter clock 34, to determine when the respective frame is to be transferred to the digital to analog converter 33. Generally, when the time stamp associated with a buffered digital audio information frame corresponds to the current time as indicated by the digital to analog converter clock 34, the playback scheduler 32 will enable the respective frame to be transferred to the digital to analog converter 33.

[0064] The digital to analog converter 33, also under control of the digital to analog converter clock 34, converts the buffered digital audio information to analog form, and provides the analog audio information to the audio amplifier 35 for amplification. The amplified analog information, in turn, is provided to the audio reproduction devices 15(n)(r) through the audio reproduction device interface 36. The audio reproduction devices 15(n)(r) transform the analog audio information signal to sound thereby to provide the audio program to a listener. The amount by which the audio amplifier 35 amplifies the analog signal is controlled by the control module 42, in response to volume control information provided by the user through the user interface module 13.

[0065] The network communications manager 40 controls network communications over the network 12, and the network interface 41 transmits and receives message packets over the network 12. The network communications manager 40 generates and receives messages to facilitate the transfer of the various types of information described above in connection with FIG. 2, including the channel device control information, slave device control information, audio and playback timing information and the audio information channel device's clock timing information. In connection with the

LINKPLAY EXHIBIT 1004
Page 1981 of 4822

channel device control information and the slave device control information, the network communications manager 40 will generate messages for transfer over the network 12 in response to control information from the control module 42. Similarly, when the network communications manager 40 receives messages containing channel device control information and slave device control information, the network communications manager will provide the information to the control module 42 for processing.

[0066] With regards to the audio information channel device's clock timing information, as noted above, the master device 21 and slave devices 22(g) of the synchrony group 20 obtain the clock timing information from the audio information channel device 23 using the well-known SNTP. If the zone player 11(n) is operating as the audio information channel device 23 for a synchrony group, during the SNTP operation, it will provide its current time, particularly a current time as indicated by its digital to analog converter clock 34. On the other hand, if the zone player 11(n) is operating as the master device 21 or slave device 22(g) of a synchrony group 20, it will receive the clock timing information from the audio information channel device 23. After the respective device 21, 22(g) has obtained the audio information channel device's clock timing information, it will generate a differential time value AT representing the difference between the time T indicated by its digital to analog converter clock 34 and the current time information from the audio information channel device 23. The differential time value will be used to update the time stamps for the frames of the digital audio stream 50 (FIG. 4) that are received from the audio information channel device.

[0067] With regards to the audio and playback timing information, operations performed by the network communications manager 40 will depend on whether

(i) the audio and playback timing information has been buffered in the audio information buffer 31 for transmission, as audio information channel device 23, over the network 12 to the master device 21 and/or slave devices 22(g) of a synchrony group, or

(ii) the audio and playback timing information has been received from the network 12 to be played by the zone player 11(n) as either the master device 21 for a synchrony group or a slave device in a synchrony group.

[0068] It will be appreciated that the network communications manager 40 may be engaged in both (i) and (ii) contemporaneously, since the zone player 11(n) may operate both as the audio information channel device 23(1) for a synchrony group 20(1) (reference FIG. 2A) of which it is not a member, and a member of another synchrony group 20(2) for which another zone player 11(n') is the audio information channel device 23(2). With reference to item (i) above, after a packet that is to be transmitted has been buffered in the respective ring buffer, the network communications manager 40 retrieves the packet, packages it into a message and enables the network interface 41 to transmit the message over the network 12. If the control module 42 receives control information from the user interface module 13 (if the master device 21 is also the audio information channel device 23 for the synchrony group 20) or from the master device (if the master device 21 is not the audio information channel device 23 for the synchrony group 20) that would require the transmission of a "resynchronize" command as described above, the control module 42 of the audio information channel device 23 enables the network communications manager 40 to insert the command into a message containing the audio and playback timing information. Details of the operations performed in connection with the "resynchronize" command will be described below. As noted above, the "resynchronize" command is used if the user enables a synchrony group to terminate the playback of a track that is currently being played, or cancel playback of a track whose playback has not begun.

[0069] On the other hand, with reference to item (ii) above, if network interface 41 receives a message containing a packet containing frames of audio and playback timing information that the zone player 11(n) is to play either as a master device 21 or a slave device for a synchrony group 20, the network interface 41 provides the audio and playback timing information to the network communications manager 40. The network communications manager 40 will determine whether the packet contains a resynchronize command and, if so, notify the control module 42, which will enable operations to be performed as described below. In any case, the network communications manager 40 will normally buffer the various frames comprising the audio and playback timing information in the audio information buffer 31, and in that operation will generally operate as described above in connection with the audio information source interface 30. Before buffering them, however, the network communications manager 40 will update their time stamps using the time differential value described above. It will be appreciated that the network communications manager 40 will perform similar operations whether the messages that contain the packets were multi-cast messages or unicast messages as described above

[0070] The updating of the time stamps by the master device 21 and the slave devices 22(g) in the synchrony group 20 will ensure that they all play the audio information synchronously. In particular, after the network communications manager 40 has received a frame 51(f) from the network interface 41, it will also obtain, from the digital to analog converter clock 34, the zone player 11(n)'s current time as indicated by its digital to analog converter clock 34. The network communications manager 40 will determine a time differential value that is the difference between the slave device's current clock time, as indicated by its digital to analog converter 34, and the audio information channel device's time as indicated by the audio information channel device's clock timing information. Accordingly, if the master or slave

LINKPLAY EXHIBIT 1004
Page 1982 of 4822

device's current time has a value T_S and the audio information channel device's current time, as indicated by the clock timing information, has a value T_C, the time differential value $\Delta T=T_S-T_C$. If the current time of the master or slave device in the synchrony group 20, as indicated by its digital to analog converter clock 34, is ahead of the audio information channel device's clock time as indicated by the clock timing information received during the SNTP operation, the time differential value will have a positive value. On the other hand, if the master or slave device's current time is behind the audio information channel device's clock time, the time differential value ΔT will have a negative value. If the zone player 11(n) obtains clock timing information from the audio information channel device 23 periodically while it is a member of the synchrony group 20, the network communications manager 40 can generate an updated value for the time differential value AT when it receives the clock timing information from the audio information channel device 23, and will subsequently use the updated time differential value.

[0071] The network communications manager 40 uses the time differential value AT that it generates from the audio information channel device timing information and zone player 11(n)'s current time to update the time stamps that will be associated with the digital audio information frames that the zone player 11(n) receives from the audio information channel device. For each digital audio information frame that is received from the audio information channel device, instead of storing the time stamp that is associated with the frame as received in the message in the audio information buffer 21, the network communications manager 40 will store the updated time stamp with the digital audio information frame. The updated time stamp is generated in a manner so that, when the zone player 11(n), as a member of the synchrony group plays back the digital audio information frame, it will do so in synchrony with other devices in the synchrony group.

[0072] More specifically, after the zone player 11(n)'s network interface 41 receives a message containing a packet that, in turn, contains one or more frames 51(f), it will provide the packet to the network communications manager 40. For each frame 51(f) in the packet that the network communications manager 40 receives from the network interface 41, the network communications manager 40 will add the time differential value AT to the frame's time stamp, to generate the updated time stamp for the frame 51(f), and store the frame 51(f), along with the header 55(f) with the updated time stamp in the audio information buffer 31. Thus, for example, if a frame's time stamp has a time value T_F, the network communications manager 40 will generate an updated time stamp T^U_F having a time value $T^U_F=T_F+\Delta T$. Since time value T^U_F according to the slave device's digital to analog converter clock 34 is simultaneous to the time value T_F according to the audio information channel device's digital to analog converter clock 34, the zone player 11(n) device will play the digital audio information frame at the time determined by the audio information channel device 23. Since all of the members of the synchrony group 20 will perform the same operations, generating the updated time stamps T^U_F for the various frames 51(f) in relation to their respective differential time values, all of the zone players 11(n) in the synchrony group 20 will play them synchronously. The network communications manager 40 will generate updated time stamps T^U_F for all of the time stamps 60 in the packet, and then store the packet in the audio information buffer 31.

[0073] It will be appreciated that, before storing a packet in the audio information buffer 21, the network communications manager 40 can compare the updated time stamps T^U_F associated with the frames in the packet to the slave device's current time as indicated by its digital to analog converter clock 34. If the network communications manager 40 determines the time indicated by the updated time stamps of frames 51(f) in the packet are earlier than the zone player's current time, it can discard the packet instead of storing it in the audio information buffer 21, since the zone player 11(n) will not play them. That is, if the updated time stamp has a time value T^U_F that identifies a time that is earlier than the zone player's current time T_S as indicated by the zone player's digital to analog converter clock 34, the network communications manager 40 can discard the packet.

[0074] If the zone player 11(n) is operating as the master device 21 of a synchrony group 20, when the user, through the user interface module 13, notifies the zone player 11(n) that another zone player 11(n') is to join the synchrony group 20 as a slave device 22(g), the control module 42 of the zone player 11(n) enables the network communications manager 40 to engage in an exchange of messages, described above in connection with FIG. 2, to enable the other zone player 11(n') to join the synchrony group 20 as a slave device. As noted above, during the message exchange, the messages generated by the network communications manager 40 of the zone player 11(n) will provide the network communications manager of the zone player 11(n') that is to join the synchrony group 20 with information such as the multi-cast address being used by the audio information channel device 23 that is providing the audio program to the synchrony group 20, as well as a unicast network address for the audio information channel device 23. After receiving that information, the network communications manager and network interface of the zone player 11(n') that is to join the synchrony group 20 can begin receiving the multi-cast messages containing the audio program for the synchrony group, engage in SNTP transactions with the audio information channel device 23 to obtain the latter's current time, and also enable the audio information channel device 23 to send the zone player 11(n') frames 51(f) that it had previously broadcast using the unicast message transmission methodology as described above.

[0075] On the other hand, if the network communications manager 40 and network interface 41 of the zone player 11(n) receive a message over the network 12 indicating that it is to become a slave device 22(g) of a synchrony group for which another zone player 11(n') is the master device, the network communications manager 40 for zone player

LINKPLAY EXHIBIT 1004
Page 1983 of 4822

11(n) will provide a notification to the control module 42 of zone player 11(n). Thereafter, the control module 42 of zone player 11(n) can enable the network communications manager 40 of zone player 11(n) to perform the operations described above to enable it to join the synchrony group 20.

[0076] As noted above, the user, using user interface module 13, can enable the synchrony group to terminate playback of a track of an audio program that is currently being played. After playback of a track that is currently being played has been terminated, playback will continue in a conventional manner with the next track that has been buffered in the audio information buffer 31. It will be appreciated that that could be the next track that is on the original play list, or a previous track. In addition, the user can enable the synchrony group 20 to cancel playback of a track that it has not yet begun to play, but for which buffering of packets has begun in the synchrony group 20. Both of these operations make use of the "resynchronize" command that the master device 21 of the synchrony group 20 can enable the audio information channel device 23 to include in the multi-cast message stream that it transmits to the synchrony group 20. Generally, in response to receipt of the resynchronize command, the members of the synchrony group 20 flush the ring buffer containing the packets that they are to play in the future. In addition, if the members of the synchrony group provide separate buffers for their respective digital to analog converters 33, the members will also flush those buffers as well. After the audio information channel device transmits a packet containing the resynchronize command:

(i) in the case of the use of the resynchronize command to terminate playing of a track that is currently being played, the audio information channel device 23 will begin multi-casting packets for the next track, to begin play immediately, and will continue through the play list in the manner described above; and

(ii) in the case of the use of the resynchronize command to cancel play of a track for which buffering has begun, but which is to be played in the future, the audio information channel device 23 will begin multi-casting packets for the track after the track that has been cancelled, to be played beginning at the time the cancelled track was to begin play, and will also continue through the play list in the manner described above.

It will be appreciated that,

(a) in the first case (item (i) directly above), the resynchronize command can enable the read pointer to be set to point to the entry in the circular buffer into which the first packet for the next track will be written, which will correspond to the entry to which the write pointer points, but

(b) in the second case (item (ii) directly above), the resynchronize command can enable the write pointer for the circular buffer to be set to point to the entry that contains the first packet for the track whose play is being cancelled.

It will further be appreciated that, if a track is cancelled for which buffering has not begun, the resynchronize command will generally not be necessary, since the audio information channel device 23 for the synchrony group 20 need only delete that track from the play list.

[0077] Operations performed in connection with use of the resynchronize command to cancel playback of a track that is currently being played will be described in connection with Packet Sequence A below, and operations performed in connection with the use of the resynchronize command to cancel playback of a track that is has not yet begun to play, but for which buffering of packets has begun, will be described in connection with Packet Sequence B below.

Packet Sequence A

[0078]

(A1.0) [packet 57]
(A1.1) [continuation of frame 99]
(A1.2) [frame 100, time = 0:00:01, type = mp3 audio]
(A1.3) [frame 101, time = 0:00:02, type = mp3 audio]
(A1.4) [frame 102, time = 0:00:03, type = mp3 audio]

(A2.0) [packet 58]
(A2.1) [continuation of frame 102]
(A2.2) [frame 103, time = 0:00:04, type = mp3 audio]
(A2.3) [frame 104, time = 0:00:05, type = mp3 audio]
(A2.4) [frame 105, time = 0:00:06, type = mp3 audio]

LINKPLAY EXHIBIT 1004
Page 1984 of 4822

(continued)

(A3.0) [packet 59]
(A3.1) [continuation of frame 105]
(A3.2) [frame 106, time = 0:00:07, type = mp3 audio]
(A3.3) [frame 107, time = 0:00:08, type = mp3 audio]
(A3.4) [frame 108, time = 0:00:09, type = mp3 audio]
(A4.0) [packet 60]
(A4.1) [continuation of frame 108]
(A4.2) [frame 109, time = 0:00:10, type = mp3 audio]
(A4.3) [Resynchronize command]
(A4.4) [Padding, if necessary]

(A5.0) [packet 61]
(A5.1) [frame 1, time = 0:00:07, type = mp3 audio]
(A5.2) [frame 2, time = 0:00:08, type = mp3 audio]
(A5.3) [frame 3, time = 0:00:09, type = mp3 audio]
(A5.4) [frame 4, time = 0.00.10, type = mp3 audio]

(A6.0) [packet 62]
(A6.1) [continuation of frame 4]
(A6.2) [frame 5, time = 0:00:11, type = mp3 audio]
(A6.3) [frame 6, time = 0:00:12, type = mp3 audio]
(A6.4) [frame 7, time = 0:00:13, type = mp3 audio]

[0079] Packet Sequence A comprises a sequence of six packets, identified by packet 57 through packet 62, that the audio information channel device 23 multi-casts in respective messages to the members of a synchrony group 20. It will be appreciated that the series of messages that the audio information channel device 23 may multi-cast to the synchrony group 20 may include a messages prior to the packet 57, and may also include messages after packet 62. Each packet comprises a packet header, which is symbolized by lines (A1.0), (A2.0),...(A6.0) in Packet Sequence A, and will generally also include information associated with at least a portion of a frame. In the packets represented in Packet Sequence A, each packet includes information associated with a plurality of frames. Depending on the lengths of the packets, each packet may contain information associated with a portion of a frame, an entire frame, or multiple frames. In the illustration represented by Packet Sequence A, it is assumed that each packet may contain information associated with multiple frames. In addition, it is assumed that a packet does not necessarily contain information associated with an integral number of frames; in that case, a packet may contain information associated with a portion of a frame, and the next packet will contain the information associated with the rest of a frame.

[0080] The frames and associated header playback timing information contained in the various packets are symbolized by lines (A1.1), (A1.2),..., (A1.4), (A2.1),.... (A6.4) of Packet Sequence A. Thus, for example, line (A1.2) of packet 57 represents the one-hundredth frame, that is, frame 51(100) (reference FIG.4), of the track whose audio information is being transmitted in the sequence of packets that includes packet 57. The frame 51 (100) is to be played at an illustrative time, according to the audio information channel device's digital to analog converter clock, of "time=0:00:01," and the frame is encoded and/or compressed using the well-know MP3 encoding and compression methodology. In that case, the legend "time = 0:00:01" represents the time stamp that would be included in field 60 (FIG. 4) of the header associated with the frame 50(100) as multi-cast by the audio information channel device for the synchrony group. It will be appreciated that the playback time and encoding/compression methodology will be referred in the header 55(100) that is associated with the frame 51(100). It will also be appreciated that the header may also contain additional information as described above.

[0081] Similarly, line (A1.3) of packet 57 represents the one-hundred and first frame, that is, frame 51(101), of the track whose audio information is being transmitted in the sequence of packets that includes packet 57. The frame 51(101) is to be played at an illustrative time, according to the audio information channel device's digital to analog converter clock, of "0:00:02," and the frame is also encoded and/or compressed using the MP3 encoding and compression methodology. Line (A1.4) of packet 57 represents similar information, although it will be appreciated that, depending on the length of packet 57, the line may not represent the information for an entire frame 51(102) and/or its associated header. If the length of packet 57 is not sufficient to accommodate the information for the entire frame 51(102) and/or associated header, the information will continue in packet 58, as represented by line (A2.1) in Packet Sequence A. Similarly, if the

LINKPLAY EXHIBIT 1004
Page 1985 of 4822

length of packet 56 was not sufficient to contain the information for an entire frame 51(99) preceding frame 51(100), packet 57 (lines (A1.0) through 1.4) may contain any information from frame 51 (99) that packet 56 was unable to accommodate.

[0082] As noted above, when the master device 21 or a slave device 22(g) in the synchrony group 20 receives the packet 57, its respective network communications manager 40 will update the time stamps associated with the various frames 51(f) as described above before buffering the respective frames in the respective audio information buffer 31.

[0083] Packets 58 and 59 contain information that is organized along the lines described above in connection with packet 57.

[0084] Packet 60 also contains, as represented by lines (A4.1) and (A4.2), information that is organized along the lines of the information represented by lines (Ax.1) and (Ax.2) ("x" equals an integer) described above in connection with packets 57 through 59. On the other hand, packet 60 contains a resynchronize command, as represented by line (A4.3). Packet 60 also may contain padding, as represented by line 4.4, following the resynchronize command. As noted above, the master device 21 of a synchrony group 20 will enable the audio information channel device 23 that is providing audio information to the synchrony group 20 to multi-cast a message containing the resynchronize command when it receives notification from the user interface module 13 that the user wishes to cancel playback of a track that is currently being played. In the example depicted in Packet Sequence A, as will be described below, the audio information channel device 23 receives notification from the master device 21 that the user wishes to cancel playback of a track at a time corresponding to "time = 0:00:07" according to its digital to analog converter clock 34, and, in line (A4.3) of packet 60 it will provide the resynchronize command, followed by padding, if necessary.

[0085] As will be apparent from examining lines (A3.1) through (A3.4) of packet 59 and lines (A4.1) and (A4.2) of packet 60, although the audio information channel device 23 has received the notification from the synchrony group's master device 21 to multi-cast the resynchronize command at a time corresponding to "time = 0:00:07" according to the clock time indicated by its digital to analog converter clock 34, it (that is, the audio information channel device 23) has already multi-cast messages containing frames that are to be played at that time and subsequently. That is, the audio information channel device 23 has, multi-cast in packet 59, frames 51(106) through 51(108) that contain time stamps "time = 0:00:07," "time = 0:00:08" and "time = 0:00:09," respectively, and, in packet 60, in addition to the continuation of frame 51(108), frame 51(109) that contains time stamp "time = 0:00:10." (It will be appreciated that the times indicated by the illustrative time stamps are for illustration purposes only, and that in an actual embodiment the time stamps may have different values and differentials.)

[0086] As noted above, the audio information channel device 23 multi-casts a message containing a packet that, in turn, contains the resynchronize command when it receives the notification from the master device 21 to do so. In the example depicted in Packet Sequence A, the packet will be multi-cast when the audio information channel device's digital to analog converter clock time corresponds to"0:00:07." Subsequently, two things happen. In one, aspect, when the master device 21 and slave devices 22(g) receive the packet that contains the resynchronize command, they will stop playing the audio program that they are playing.

[0087] In addition, the audio information channel device 23 will begin transmitting frames containing audio information for the next track, including therewith time stamps immediately following the digital to analog converter clock time at which the packet including the resynchronize command was transmitted. Accordingly, and with further reference to Packet Sequence A, the audio information channel device 23 will multi-cast a message containing packet 61. As indicated above, packet 61 contains, as represented in lines (A5.1) through (A5.3), frames 51(1) through 51(3), which are the first three frames of the next track of the audio program that is to be played. It is also compressed and encoded using the MP3 encoding and compression scheme, and it is accompanied by time stamps "time = 0:00:07," "time = 0:00:08" and "time = 0:00:10." As noted above, the time stamp "time=0:00:07" corresponds to the clock time at which the audio information channel device 23 multi-casts the resynchronize command, and, when the master device 21 and slave devices 22(g) receive these frames, they would be expected to begin playing them very shortly, if not immediately after the audio information channel device 23 multi-casts the message containing the packet that, in turn, contains the resynchronize command. Packet 61 also includes at least a portion of the next frame, that is, frame 51(4), for that track. In addition, Packet Sequence A depicted above further includes a subsequent packet, namely, packet 62, that contains any necessary continuation of frame 51 (4), as well as three subsequent frames. If any additional packets are required for the track, as well as for subsequent tracks, they can be multi-cast in a similar manner.

[0088] As further noted above, the resynchronize command can also be used to cancel playing of one or more tracks for which playback has begun. This will be illustrated in connection with Packet Sequence B :

Packet Sequence B

[0089]

LINKPLAY EXHIBIT 1004
Page 1986 of 4822

(B1.0) [packet 157]
(B1.1) [continuation of frame 99]
(B1.2) [frame 100, time = 0:00:01, type = mp3 audio]
(B1.3) [frame 101, time = 0:00:02, type = mp3 audio]
(B1.4) [frame 102, time = 0:00:03, type = mp3 audio]

(B2.0) [packet 158]
(B2.1) [continuation of frame 102]
(B2.2) [frame 103, time = 0:00:04, type = mp3 audio]
(B2.3) [frame 104, time = 0:00:05, type = mp3 audio]
(B2.4) [frame 105, time = 0:00:06, type = mp3 audio]

(B3.0) [packet 159]
(B3.1) [continuation of frame 105]
(B3.2) [frame 106, time = 0:00:07, type = mp3 audio]
(B3.3) [track boundary notification]
(B3.4) [Padding, if necessary]

(B4.0) [packet 160]
(B4.1) [frame 1, time = 0:00:08, type = mp3 audio]
(B4.2) [frame 2, time = 0:00:09, type = mp3 audio]
(B4.3) [frame 3, time = 0:00:10, type = mp3 audio]

(B5.0) [packet 161]
(B5.1) [continuation of frame 3]
(B5.2) [frame 4, time = 0:00:11, type = mp3 audio]
(B5.3) [Resynchronize, after packet 159]
(B5.4) [Padding, if necessary]

(B6.0) [packet 162]
(B6.1) [frame 1, time = 0:00:08, type = mp3 audio]
(B6.2) [frame 2, time = 0:00:09, type = mp3 audio]
(B6.3) [frame 3, time = 0:00:10, type = mp3 audio]
(B6.4) [frame 4, time = 0:00:11, type = mp3 audio]

(B7.0) [packet 163]
(B7.1) [continuation of frame 4]
(B7.2) [frame 5, time = 0:00:12, type = mp3 audio]
(B7.3) [frame 6, time = 0:00:13, type = mp3 audio]
(B7.4) [frame 7, time = 0:00:14, type = mp3 audio]

[0090] Packet Sequence B comprises a series of seven packets, identified by packet 157 through 163 , that the audio information channel device 23 multi-casts to the members of a synchrony group 20. As with Packet Sequence A, it will be appreciated that the series of packets that the audio information channel device 23 may multi-cast to the synchrony group 20 may include packets prior to the packet 157, and may also include packets after packet 162. Each packet comprises a packet header, which is symbolized by lines (B1.0), (B2.0),... (B7.0) in Packet Sequence B. As in Packet Sequence A, each packet will also generally include information associated with at least a portion of a frame 51(f) along with its associated frame 55(f). As in the packets represented in Packet Sequence A, each packet includes information associated with a plurality of frames. Depending on the lengths of the packets, each packet may contain information associated with a portion of a frame, an entire frame, or multiple frames. Further, as with Packet Sequence A, it is assumed that each packet may contain information associated with multiple frames. In addition, it is assumed that a packet does not necessarily contain information associated with an integral number of frames; in that case, a packet

LINKPLAY EXHIBIT 1004
Page 1987 of 4822

may contain information associated with a portion of a frame, and the next packet will contain the information associated with the rest of a frame.

[0091] The structures of the packets represented by Packet Sequence B are similar to those described above in connection with Packet Sequence A, and will not be repeated here. Generally, Packet Sequence B illustratively contains a sequence of packets that represent at least portions of three tracks that may have been selected from, for example, a play list. In particular, packets 157 through 159 represent frames from a portion of one track, packets 160 and 161 represent frames from a second track and packets 162 and 163 represent frames from a third track. The play list indicated that the first, second and third tracks were to be played in that order. With particular reference to Packet Sequence B, it should be noted that line (B3.3) indicates that packet 159 includes an indication that that packet contains the last frame for the track, and line (B3.4) provides for padding to the end of the packet. The first frame of the next track begins in packet 160.

[0092] In connection with the use of the resynchronize command to cancel playback of a track, at least a portion of which the audio information channel device 23 has multi-cast to the members of the synchrony group, packet 161, in line (B5.3) represents a resynchronize command that indicates that resynchronization is to occur after packet 159, that is, immediately after the packet that contains the last frame of the first of the three tracks represented by the packets in Packet Sequence B. It should be noted that the resynchronize command is in the packet 161, while the resynchronization is to occur at packet 160, that is, the synchrony group is to not play the track starting with packet 160, but instead is to begin playing the track frames for which begin with the next packet, that is, packet 162. As with Packet Sequence A, in Packet Sequence B the audio information channel device 23, in packet 162 and 163, multi-casts frames whose time stamps indicate that they are to be played when the frames that were multi-cast in packets 160 and 161 were to be played. By use of the resynchronize command and specifying a packet in this manner, the audio information channel device can cancel playback of a track for which playback has not yet begun.

[0093] It will be appreciated that the resynchronize command is generally not necessary for cancelling play back of a track that the audio information channel device 23 has not started multi-casting to the synchrony group 20, since the audio information channel device 23 itself can re-order the play list to accommodate the cancellation.

[0094] The invention provides a number of advantages. In particular, the invention provides a network audio system in which a number of devices share information can reproduce audio information synchronously, notwithstanding the fact that packets, which may contain digital audio information, transmitted over the network to the various zone players connected thereto may have differing delays and the zone players operate with independent clocks. Moreover, although the invention has been described in connection with audio information, it will be appreciated that the invention will find utility in connection with any type of isochronous information for which synchrony among devices is desired. The system is such that synchrony groups are created and destroyed dynamically, and in such a manner as to avoid requiring a dedicated device as the master device.

[0095] It will be appreciated that a number of changes and modifications may be made to the network audio system 10 as described above. For example, although the invention has been described as providing that the audio information channel device 23 provides digital audio information to the members synchrony group 20 that has been encoded using particular types of encoding and compression methodologies, it will be appreciated that the audio information channel device 23 can provide digital audio information to various members of the synchrony group 20 that have been encoded and compressed using different types of encoding and compression methodologies, and, moreover, for which different sampling rates have been used. For example, the audio information channel device 23 may provide digital audio information to the master device 21 and slave devices 22(1) through 22(g_1) using the MP3 methodology at a specified sampling rate, the digital audio information for the same program to slave devices 22(g_1+1) through 22(g_2) using the WAV methodology at one specified sampling rate, and to slave devices 22(g_2+1) through 22(G) using the WAV methodology at another specified sampling rate. In that case, the audio information channel device 23 can specify the particular encoding and compression methodology that has been used in the encoding type field 57 associated with each frame and the sampling rate in the sampling rate field 58. Moreover, since the encoding and compression type and sampling rate are specified for each frame, the encoding and compression type and sampling rate can be changed from frame to frame. The audio information channel device 23 may use different multi-cast addresses for the different encoding and compression types and sampling rates, but it will be appreciated that that would not be required.

[0096] It will be appreciated that two advantages of providing that the encoding and compression methodology and the sampling rate is provided on a frame-by-frame basis, instead of on, for example, a track-by-track basis, is that that would facilitate a slave device joining the synchrony group 20 at a frame mid-track, without requiring, for example, the master device 21 or the audio information channel device 23 to notify it of the encoding and compression methodology or the sampling rate.

[0097] Another modification is that, instead of the network communications manager 40 of a member of a synchrony group 20 generating the updated time stamp T^u_F for a digital audio information frame by adding the time differential value AT to the time stamp T_F associated with a frame, the network communications manager 40 may instead generate the updated time stamp T^U_F by subtracting the differential time value AT from the member's current time T_S as indicated

LINKPLAY EXHIBIT 1004
Page 1988 of 4822

by the member's digital to analog converter clock 34 at the time at which the digital audio information is received. It will be appreciated, however, that there may be variable time delays in processing of messages by the slave device's network communications manager 40, and so it may be preferable to generate the time differential value AT using the time stamp T_F provided by the audio information channel device 23.

[0098] In addition, instead of the network communications manager 40 of a member of a synchrony group generating an updated time stamp to reflect the difference between the times indicated by the member's digital to analog converter clock and the audio information channel device's digital to analog converter clock, the network communications manager 40 can generate the time differential value AT and provide it to the member's playback scheduler 32. In that case, the member's network communications manager 40 can store each digital audio information frame along with the time stamp T_F as received from the master device in the audio information buffer 21. The playback scheduler 32 can utilize the time differential value ΔT, and the time stamps T_F associated with the digital audio information frames, to determine when the respective digital audio information frames are to be played. In determining when a digital audio information frame is to be played, the playback scheduler can add the time differential value to the time stamp T_F associated with the digital audio frame, and enable the digital audio frame to be coupled to the digital to analog converter 33 when the time indicated by the sum corresponds to the current time as indicated by the slave device's digital to analog converter clock 34. Alternatively, when the member's digital to analog converter clock 34 updates its current time T_S, the playback scheduler can generate an updated current time T'_S by subtracting the differential time value AT from the current time T_S, and using the updated current time T'_S to determine when to play a digital audio information frame.

[0099] As described above, the members of a synchrony group 20 periodically obtain the audio information channel device's current time value and uses the current time value that it receives from the audio information channel device to periodically update the time differential value AT that it uses in updating the time stamps associated with the various frames. It will be appreciated that, if the digital to analog converter clock(s) associated with the member(s) of a synchrony group 20 are ensured to have the same rate as the digital to analog converter clock, a member need only obtain the current time value from the audio information channel device once, at the beginning of playback.

[0100] As another alternative, if the zone players are provided with digital to analog converter clock 34 whose time and rate can be set by an element such as the network communications manager 40, when a zone player 11(n) is operating as a member of a synchrony group 20, its network communications manager 40 can use the various types of timing information that it receives from the audio information channel device 23, including the current time information and the playback timing information indicated by the time stamps that are associated with the various frames 51(f) comprising the audio and playback timing information that it receives, to adjust the synchrony group member's digital to analog converter clock's time value and/or the clock rate that it uses for playback.

[0101] If the clock's time value is to be adjusted, when the synchrony group member's network communications manager 40 initially receives the current time information from the audio information channel device 23 for the synchrony group 20, the network communications manager 40 can set the synchrony group member's digital to analog converter clock 34 to the current time value as indicated by the audio information channel device's current time information. The network communications manager 40 can set the clock 34 to the current time value indicated by the audio information channel device's current time information once, or periodically as it receives the current time information.

[0102] Alternatively or in addition, the synchrony group member's network communications manager 40 can use one or both of the current time information and/or the playback timing information in the time stamps associated with the respective frames 51(f) to adjust the clock rate of the clock 34 that it uses for playback. For example, when the synchrony group member's network communications manager 40 receives a frame 51(f_X) having a time stamp having a time value T_{fx}, it can generate the updated time value $T^U_{fx} = T_{fx} + \Delta T$ as described above, and store the frame
with the time stamp with the updated time value in the audio information buffer 30. In addition, since both the number of samples in a frame and the sampling rate, which determines the rate at which the frame is to be played, are known to the network communications manager 40, it can use that information, along with the updated time value T^U_{Fx} that is to be used for frame 51(f_X) to generate an expected updated time value T^E_{fx+1}, that is expected for the updated time stamp of the next frame 51(f_{x+1}). After the synchrony group member's network communications manager 40 receives the next frame 51(f_{x+1}), it can generate the updated time value T^U_{fx+1} and compare that value to the expected updated time value T^E_{fx+1}. If the two time values do not correspond, or if
the difference between them is above a selected threshold level, the clock that is used by the audio information channel device 23 to generate the time stamps is advancing at a different rate than the synchrony group member's digital to analog converter clock 34, and so the network communications manager 40 can adjust the rate of the digital to analog converter clock 34 to approach that of the clock used by the audio information channel device 23 so that the differential time value AT is constant. On the other hand, if the two time values do correspond, then the time differential value AT is constant, or the difference is below a threshold level, and the network communications manager 40 need not change the clock rate of the digital to analog converter clock 34. It will be appreciated that, if the clock rate is to be adjusted, the rate adjustment can be fixed, or it can vary based on, for example, the difference between the updated time value T^U_{fx+1}, and the expected

LINKPLAY EXHIBIT 1004
Page 1989 of 4822

updated time value T^{E}_{fx+1}.

[0103] It will also be appreciated that, if no rate adjustment is performed for one frame 51 (f_{X+1}), the synchrony group member's network communications manager 40 can generate an expected updated time value T^{E}_{fx+2} that is expected for the updated time stamp of the next frame 51(f_{X+2}) using the updated time value T^{U}_{Fx} determined for frame 51(f_X), along with the number of samples in a frame and the sampling rate, and compare the expected updated time value T^{E}_{fx+2} to the updated time value T^{U}_{fx+2} that it generates when it receives frame 51(f_{X+2}). At that point, if the network communications manager 41 determines that two time values do not correspond, or if the difference between them is above a selected threshold level, it can adjust the rate of the digital to analog converter clock 34. Similar operations can be performed if no rate adjustment is performed for several successive frames 51(f_{X+1}), 51(f_{X+2}),.... This will accommodate the possibility that the rate differential between the clock 34 and the clock used by the audio information channel device 23 in generating the time stamps have rates that differ by an amount sufficiently small that it cannot be detected using time stamps of two or more successive frames.

[0104] Instead or in addition to adjusting the clock rate as described above, the synchrony group member's network communications manager 40 can perform similar operations in connection with adjusting the clock rate in connection with the current time information that it receives from the audio information channel device 23.

[0105] Furthermore, although the network audio system 10 has been described such that the master device 21 of a synchrony group 20 can, in response to control information provided thereto by a user through the user interface module 13, provide a notification to a zone player 11(n) that it is to become a member of its synchrony group 20 as a slave device 22(g), it will be appreciated that the user interface module 13 can provide the notification directly to the zone player 11(n) that is to become a member of the synchrony group 20. In that case, the zone player 11(n) can notify the master device 21 that it is to become a slave device 22(g) in the synchrony group 20, after which the master device 21 can provide information regarding the synchrony group 20, including the multi-cast and unicast addresses of the audio information channel device and other information as described above.

[0106] Similarly, although the network audio system 10 has been described such that the master device 21 of a synchrony group 20 can, in response to control information provided thereto by a user through the user interface module 13, provide a command to a slave device 22(g) to enable the slave device 22(g) to adjust its volume, it will be appreciated that the user interface module 13 can provide control information directly to the slave device 22(g) to enable the slave device 22(g) to adjust its volume.

[0107] In addition, although the network audio system 10 has been described such that each frames 51(f) is associated with a frame sequence number (reference field 56, FIG. 4), it will be appreciated that, if the packets described above in connection with Packet Sequence A and Packet Sequence B are provided with packet sequence numbers, the frame sequence numbers need not be provided, since the packet sequence numbers can suffice for defining the frame sequencing.

[0108] Furthermore, although the network audio system 10 has been described such that the zone players 11(n) are provided with an audio amplifier 35 for amplifying the analog signal provided by the respective digital to analog converters 33, it will be appreciated that a zone player may be provided that does not itself include an audio amplifier. In that case, the analog signal may be coupled to an external amplifier for amplification as necessary before being provided to the audio reproduction device(s) 15(n)(r). It will be appreciated that a single zone player 11(n) may be provided with multiple audio amplifiers and audio reproduction device interfaces, and, if necessary, multiple digital to analog converters 33, to provide audio programs for corresponding numbers of synchrony groups.

[0109] Similarly, although the zone players 11(n) have been described such that they may be connected to one or more audio information sources, it will be appreciated that an audio information source may form part of and be integrated into a zone player 11(n). For example, a zone player may include a compact disk player, cassette tape player, broadcast radio receiver, or the like, that has been integrated into it. In addition, as noted above, an individual zone player 11(n) may be connected to multiple audio information sources and may contemporaneously operate as the audio information channel device 23 for multiple synchrony groups.

[0110] In addition, although FIG. 1 shows the network audio system 10 as including one user interface module 13, it will be appreciated that the system 10 may include a plurality of user interface modules. Each user interface module be useful for controlling all of the zone players as described above, or alternatively one or more of the user interface modules may be useful for controlling selected subsets of the zone players.

[0111] Moreover, it will be appreciated that, although the invention has been described in connection with audio information, it will be appreciated that the invention will find utility in connection with any type of information for which synchrony among devices connected to a network is desired.

[0112] As noted above, while a zone player 11(n) is operating as audio information channel device 23 for a synchrony group 20, when the zone player 11(n)'s audio information source interface 30 or network communications manager 40 stores digital audio information frames based on audio information from an audio information source 14(n)(s) in the audio information buffer 31, it will provide time stamps for the respective frames to schedule them for playback after some time delay after they have been buffered in the audio information buffer 31. The delay is provided so that, for other zone

LINKPLAY EXHIBIT 1004
Page 1990 of 4822

players 11(n'), 11(n''),... that are operating as members of a synchrony group, there will be sufficient time for the audio and playback timing information to be transferred over the network 12 to those other zone players 11(n'), 11(n''),... so that it can be processed and played by them at the appropriate time as described above. The time period that is selected for the time delay may be fixed or variable, and in either case may be based on a number of factors. If the time period selected for the time delay is fixed, it may be based on, for example, factors such as an estimate of the maximum latency in the network 12, the estimated maximum loading of the various components comprising the zone players 11(n), and other estimates as will be appreciated by those skilled in the art.

[0113] The time delay may be the same for audio information from all types of audio information sources, and may be constant over the entire period that the synchrony group 20 is playing an audio work. Alternatively, different time delays may be utilized based on various criteria. For example, if the audio information is to be played independently of information associated with other types of media, the time delay may be selected to be relatively long, on the order of a significant fraction of a second, or longer. On the other hand, if the audio information is to be played contemporaneously with, for example, video information, which may be supplied by, for example, a video disk, video tape cassette, over cable, satellite, or broadcast television, which may not be buffered or which may be displayed independently of the network audio system 10, it may be undesirable to provide for such a lengthy delay, since the time delay of the audio playback, in relation to the video display, may be noticeable. In that case, the zone player 11(n) may provide for a much shorter time delay. In one embodiment, the time delay provided for audio information to be played concurrently with video information is selected to be generally on the order of fifty milliseconds, which would barely, if at all, be perceptible to someone viewing the video. Other desirable time delays for information from other types of sources will be apparent to those skilled in the art.

[0114] As yet a further possibility, the zone player 11(n), when operating as an audio information channel device 23 for a synchrony group 20, can dynamically determine the time delay based on a number of conditions in the network audio system 10, including, for example, the message transfer latency in network 12, the loading of microprocessors or other components that are used in the various zone players 11(n'), 11(n''),... that may comprise a synchrony group 20, as well as other factors. For example, if the audio information channel device 23 determines that the latency in the network 12 has increased beyond a selected threshold, the audio information channel device 23 can adjust the delay to increase the likelihood that the members of the synchrony group 20 will be able to receive the packets and process the frames so that they will be able to play them at the appropriate times. Similarly, if the audio information channel device 23 is notified that a member of the synchrony group 20 to which it provides audio information requires additional time to receive and process the frames that it transmits, the audio information channel device 23 can adjust the delay accordingly. It will be appreciated that, to reduce or minimize possible discontinuities in the audio playback by the members of the synchrony group, the audio information channel device 23 can, instead of adjusting the time delay during a particular audio track, adjust the time delay between tracks, during silent periods of a track or otherwise as will be appreciated by those skilled in the art. In addition, the audio information channel device 23 can use conventional audio compression methodologies to facilitate a speeding up and/or slowing down of playback of an audio track while it is in the process of providing additional time delay. Generally, the members of the synchrony group 20 can provide notifications to the audio information channel device 23 if they determine that they will need an additional time delay, and the audio information channel device 23 can adjust the time delay in accordance with the notifications from the members of the synchrony group 20.

[0115] It will be appreciated that a system in accordance with the invention can be constructed in whole or in part from special purpose hardware or a general purpose computer system, or any combination thereof, any portion of which may be controlled by a suitable program. Any program may in whole or in part comprise part of or be stored on the system in a conventional manner, or it may in whole or in part be provided in to the system over a network or other mechanism for transferring information in a conventional manner. In addition, it will be appreciated that the system may be operated and/or otherwise controlled by means of information provided by an operator using operator input elements (not shown) which may be connected directly to the system or which may transfer the information to the system over a network or other mechanism for transferring information in a conventional manner.

[0116] The foregoing description has been limited to a specific embodiment of this invention. It will be apparent, however, that various variations and modifications may be made to the invention, with the attainment of some or all of the advantages of the invention. It is the object of the appended claims to cover these and such other variations and modifications as come within the true spirit and scope of the invention.

[0117] The following numbered clauses set out further exemplary embodiments of the present invention.

1. A system comprising plurality of devices, one of the devices operating as a task source device and at least one other device operating as a member of a synchrony group, A. the task source device being configured to distribute a series of tasks to the synchrony group, each task being associated with a time stamp indicating a time, relative to a clock maintained by the task source device, at which the devices comprising the synchrony group are to execute the respective task; B each member device being configured to: (i) periodically obtain from the task source device

LINKPLAY EXHIBIT 1004
Page 1991 of 4822

an indication of a current time value indicated by the task source device's clock, and (ii) determine, from the time stamp associated with each respective task and a time differential value representing a difference between the current time value indicated by the task source device's clock, and a current time value indicated by its respective clock, a time, relative to its respective clock, at which it is to execute the task.

2. A system as defined in clause 1 in which the synchrony group comprises a plurality of member devices.

3. A system as defined in clause 2 in which each device comprising a, member of the synchrony group is further configured to execute each task that it receives from the task source device at the determined time, thereby to facilitate execution by them of respective tasks in the series in respective timing relationships relative to the time indicated by the task source device's clock.

4. A system as defined in clause 3 in which the member devices are configured to execute respective tasks in synchrony.

5. A system as defined in clause 2 in which one of the member devices operates as a master device for the synchrony group, any other member devices comprising respective slave devices, the master device being configured to perform at least one type of synchrony group management operation in connection with the member devices comprising the synchrony group.

6. A system as defined in clause 5 further including a user interface module configured to control the master device's performance of said at least one type of synchrony group management operation, the master device being further configured to provide status information relating to the status of the synchrony group to the user interface module.

7. A system as defined in clause 6 in which the status information includes identifications of the devices comprising the synchrony group.

8. A system as defined in clause 6 in which the status information includes an identification of the task currently being executed.

9. A system as defined in clause 5, the system comprising at least one additional device, in which, in one type of synchrony group management operation, the master device is configured to enable the at least one additional device to join the synchrony group as a slave device.

10. A system as defined in clause 9 in which the task source device is configured to distribute tasks to the member devices using a selected multi-cast transmission methodology, the member devices being configured to buffer the tasks until they are to be executed, and further in which, when the at least one additional device joins the synchrony group as a slave device, the task source device is enabled to transmit at least one previously distributed task to the slave device using a selected unicast transmission methodology.

11. A system as defined in clause 5 in which, in at least one type of synchrony group management operation, the master device is configured to enable the task source device to join the synchrony group as a slave device, the task source device continuing to operate as the task source device for the synchrony group.

12. A system as defined in clause 5 in which, in at least one type of synchrony group management operation, the master device is configured to enable a slave device to disengage from the synchrony group, the disengaged slave device thereafter not being a member device in the synchrony group.

13. A system as defined in clause 5 in which, in at least one type of synchrony group management operation, the master device is configured to control the series of tasks to be distributed by the task source device.

14. A system as defined in clause 5 in which, in at least one type of synchrony group management operation, the master device is further configured to control execution by the member devices of the tasks that have been distributed by the task source device.

15. A system as defined in clause 14 in which the series of tasks includes a series of task sequences with each task sequence including a sub-set of the series of tasks, and further in which in at least one type of synchrony group management operation, the master device is configured to enable the member devices to terminate execution of a task sequence that is currently being executed.

16. A system as defined in clause 15 in which, in the synchrony group management operation enabling the member devices to terminate execution of a task sequence that is currently being executed, the master device enables the task source device to distribute a command to enable the member devices to terminate execution of the task sequence, the member devices being configured to, after receiving the command, terminate execution of the task sequence.

17. A system as defined in clause 14 in which the series of tasks includes a series of task sequences with each task sequence including a sub-set of the series of tasks, and further in which at least one type of synchrony group management operation, the master device is configured to enable the member devices to cancel execution of a task sequence that has been previously distributed, but for which execution has not begun.

18. A system as defined in clause 17 in which, in the synchrony group management operation enabling the member devices to cancel execution of a task sequence that has been previously distributed but for which execution has not begun, the master device enables the task source device to distribute a command to enable the member devices to cancel execution of the task sequence, the member devices being configured to, after receiving the command,

LINKPLAY EXHIBIT 1004
Page 1992 of 4822