IW 907159

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

January 09, 2003

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS OF:

APPLICATION NUMBER: *08/252,460*
FILING DATE: *May 31, 1994*
PATENT NUMBER: *5,734,862*
ISSUE DATE: *March 31, 1998*

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

PART (2) OF (3) PART(S)

P. SWAIN
Certifying Officer

08/252460 | 395 | 484 | Class | Subclass | ISSUE CLASSIFICATION

[redacted]

5734862

5734862

UTILITY SERIAL NUMBER	08/ 252460	PATENT DATE MAR 31 1998	PATENT NUMBER

SERIAL NUMBER	FILING DATE	CLASS	SUBCLASS	GROUP ART UNIT	EXAMINER
08/252,460	05/31/94	364	—	2301	

APPLICANTS

CHARLES J. KULAS, SAN FRANCISCO, CA.

CONTINUING DATA*******************
VERIFIED

P.H.

FOREIGN/PCT APPLICATIONS**********
VERIFIED

P.H.

FOREIGN FILING LICENSE GRANTED 07/06/94 ***** SMALL ENTITY *****

Foreign priority claimed ☐ yes ☒ no 35 USC 119 conditions met ☐ yes ☒ no Verified and Acknowledged P.H. Examiner's Initials	AS FILED	STATE OR COUNTRY	SHEETS DRWGS.	TOTAL CLAIMS	INDEP. CLAIMS	FILING FEE RECEIVED	ATTORNEY'S DOCKET NO.
		CA	16	7	4	$392.00	CJK1

ADDRESS

~~TOWNSEND AND TOWNSEND KHOURIE AND CREW~~
~~STEUART STREET TOWER~~
~~ONE MARKET PLAZA~~
~~SAN FRANCISCO, CA 94105~~

CHARLES J. KULAS
244 TEXAS ST
SAN FRANCISCO, CA 94107

TITLE

SYSTEM FOR SELECTIVELY BUFFERING AND DISPLAYING RELEVANT FRAMES FROM INTERLEAVING FRAMES ASSOCIATED WITH RESPECTIVE ANIMATION SEQUENCES STORED IN A MEDIUM IN RESPONSE TO USER SELECTION

v. 10-78)

PARTS OF APPLICATION FILED SEPARATELY		Applications Examiner

NOTICE OF ALLOWANCE MAILED		CLAIMS ALLOWED	
5/29/97	PO C. HUANG Assistant Examiner	Total Claims 9	Print Claim 5

ISSUE FEE			DRAWING		
Amount Due $645.00	Date Paid 6/23/97	Group Primary Examiner	Sheets Drwg. 17	Figs. Drwg. 19	Print Fig. 5
Label Area		PREPARED FOR ISSUE	ISSUE BATCH NUMBER A55		

WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 8/92)

ISSUE FEE IN FILE

PATENT APPLICATION SERIAL NO. 08/252460

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

~~140 WC 07/01/94 082524460~~ ~~1 201 392.00 CK CJK-1~~

~~140 WC 06/16/94 082524460~~ ~~1 201 392.00 CK CJK-1~~

140 WC 07/01/94 08252460 1 101 392.00 CK CJK-1

PTO-1556
(5/87)

TOWNSEND and TOWNSEND KHOURIE and CREW
Steuart Street Tower
One Market Plaza
San Francisco, CA 94105
(415) 543-9600

MAIL ROOM 31 MAY 1994 PAT. & TRADEMARK OFFICE

PATENT APPLICATION
COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D. C. 20231

08/252460

Atty. Docket CJK-1

"Express Mail" Label No. EF204755603US

Date of Deposit 5-31-94

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

By Charles J. Kulas

Sir:

Transmitted herewith for filing is the [X] patent application, [ ] continuation-in-part patent application of

Inventor:

For:

Enclosed are:

- [X] 16 sheet(s) of [ ] formal [ ] informal drawing(s).
- [ ] An assignment of the invention to ______.
- [ ] A [ ] signed [ ] unsigned Declaration & Power of Attorney.
- [X] A [X] signed [ ] unsigned Declaration.
- [ ] A Power of Attorney by Assignee with Certificate Under 37 C.F.R. Section 3.73(b).
- [X] A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27.
- [ ] A certified copy of a ______ application.
- [ ] Information Disclosure Statement under 37 CFR 1.97.
- [ ] Enclosed is a petition to extend time to respond in the parent application of the continuation-in-part application.
- [ ]

The filing fee has been calculated as shown below:

FOR:	NO. FILED (Col. 1)	NO. EXTRA (Col. 2)
BASIC FEE		
TOTAL CLAIMS	7 -20=	* 0
INDEP CLAIMS	4 -3=	* 1
[ ] MULTIPLE DEPENDENT CLAIM PRESENTED		

* If the difference in Col. 1 is less than zero, enter "0" in Col. 2

SMALL ENTITY RATE	SMALL ENTITY FEE		OTHER THAN A SMALL ENTITY RATE	OTHER THAN A SMALL ENTITY FEE
	$355	OR		$710
x11=	$	OR	x22=	$
x37=	$ 37	OR	x74=	$
+115=	$	OR	+230=	$
TOTAL	$392	OR	TOTAL	$

[X] A check for $392 is enclosed.
1 copies of this sheet are enclosed.

Telephone:
(415) 543-9600

Respectfully submitted,
Charles J. Kulas

Charles J. Kulas

Reg. No.: 35,809

16

392.0-261
08/252460

PATENT APPLICATION

see EXR AMDT #15/D for new title

~~SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED INTERACTIVE PRODUCTIONS~~

Inventor:
Charles J. Kulas

Charles J. Kulas
25 Capra Way #305
San Francisco, CA 94123
(415) 928-4375

MAIL DATE CANCELLED 21 JUN 1 1994 PAT & TRADEMARK OFF.

MAIL ROOM 21 MAY 31 1994 PAT. & TRADEMARK OFF.

08/252460

PATENT

See Amdt Exr #15/P for new title

~~SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED INTERACTIVE PRODUCTIONS~~

BACKGROUND OF THE INVENTION

This invention relates generally to displaying animation sequences in a computer system and specifically to displaying video data from a compact disc read only memory (CD-ROM) device in a computer system running an interactive application.

Interactive productions allow a user of a computer system to interact with movies, video or other displayed images while the images are being updated at a rapid rate. The purpose of these productions is to present useful information, educate or entertain the user. The ultimate goal of interactive technology is to make the user feel as though they are interacting with images on the screen so that, for example, characters or objects in a drama react to the users actions. The user's actions can affect characters, objects or other images on the display screen and change the course of the storyline.

One method for providing a high degree of interaction is to make the production completely computer generated. This means that the computer models a three dimensional world and calculates and displays the orientation of figures and objects on the screen. However, this approach is limited by today's technology because the computing power to fully calculate and render lifelike images, especially human figures, at resolutions approaching television quality in real time at video or film refresh rates is beyond the current technology for mass-marketed systems.

A different approach is to prerecord video, film or computer generated image sequences and play the prerecorded images, or frames, back at high speed. This achieves the resolution of television, or better, and is sufficiently lifelike to create a level of believability comparable to television. However, in this approach the user has a very limited amount of interactivity with the production since the user's ability to affect the story is limited to the small number of different "paths" of prerecorded image sequences that are branched to at predetermined decision points in the video or animation sequence. The use of any prerecorded sequences of images that are played back so as to achieve animation while allowing a user to interact with the images is referred to broadly here as "interactive video."

Interactive video productions typically use a compact disc read-only memory (CD-ROM) disc to store the images and a CD-ROM drive to retrieve images during playback. The CD-ROM disc stores information in a concentric spiral on optical media and is "read" or played back with a CD-ROM drive that uses a "read head" with a laser beam. The big problem with CD-ROM based interactive production is the break in continuity due to delays of about a half-second or more required to locate a desired branch path that is different from the current path that the drive's read head is tracking. Another problem is that CD-ROM based interactive video productions are severely limited in the number and types of ways that a user may interact with the video.

The length of time to access a different video path ("access time" or "seek time") depends upon the location of the different video path with respect to the current placement of the CD-ROM drive's read head. In order to access a given video sequence, a computer controller looks up the location of the sequence in an index and instructs the CD-ROM drive to access the new sequence by moving the read head to the beginning of the new sequence on the disc. Since the read head is moved by a mechanical mechanism it takes a comparatively long time to reposition the read head to a new point on the track to access the different video path.

The prior art uses caches to try to improve the performance of accessing data in a CD-ROM. The cache can be in the CD-ROM drive, in an interface card between the processor and the drive, in the memory of the computer system controlled by software or even on a hard disk or other storage medium. However, these caches only provide marginal improvement in access times where video is concerned because of the relatively small sizes of the caches compared to the data rate of the information coming off of the CD-ROM. Also, when a different path is branched to the information in the caches is usually useless since they don't contain the new data. The caches must be "purged" and loaded with new information.

While current CD-ROM drives are not adequate to provide sufficient interactivity in interactive video productions, they represent a huge installed base since hundreds of thousands have already been sold to consumers. Therefore, a system which eliminates the access time in CD-ROM based interactive videos without requiring modification of existing CD-ROM drives is desired. Further, a system that allows an interactive video production to use many paths to improve interactivity without suffering degraded performance is desired.

SUMMARY OF THE INVENTION

The invention uses a special formatting of a CD-ROM disc where portions of a first animation sequence are interleaved with portions of a second animation sequence on the CD-ROM disc itself. The invention also takes special advantage of the interleaved information during playback of the CD-ROM to completely eliminate the necessity of moving the read head in the CD-ROM drive while playing back an interactive video production. The invention uses several patterns of interleaved animations to provide vastly improved interactivity.

The invention works to special advantage with image compression techniques and with caches. Other advantages of the present invention will be apparent.

In one embodiment, the system of the present invention executes on a computer and includes a method of creating an interactive production on a CD-ROM. In the method, first and second animation sequences of digital frames are created and the animation sequences of frames are written to the CD-ROM by interleaving the frames of the first animation sequence with the frames of the second animation sequence to create the interactive production.

Another embodiment of the invention provides for playing back an interactive production created as described above. A computer system is used that includes a processor, user input device and display screen. The computer system is coupled to a CD-ROM drive. The following steps are performed under the control of the processor: continuously reading the interleaved frames from the CD-ROM; displaying only the frames of a first animation sequence on the display screen to play back a first animation; accepting signals from the user input device selecting a second animation sequence; and, in response to the signals from the user input device, displaying only the frames of the second animation sequence on the display screen to play back the second animation.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a prior art CD-ROM disc formatting;

?.H
5/27/97

Figs. 2, 2A, and 2B show ~~Fig. 2 shows~~ the formatting of a CD-ROM disc according to the present invention;

Fig. 3 is an illustration of a computer system suitable for use with the present invention;

Fig. 4 is an illustration of basic subsystems in the computer system of Fig. 3;

Fig. 5 is a block diagram of a computer system suitable for playing back data according to the present invention;

Fig. 6A shows a flowchart for a routine to select the current sequence;

Fig. 6B shows a flowchart for a routine to select and display frames according to the state of the current sequence variable;

Fig. 7 shows timelines used to illustrate simple path switching;

Fig. 8 shows timelines used to illustrate time stepped path switching;

Fig. 9 shows timelines used to illustrate a single point transition;

Fig. 10 illustrates multi point transitions;

Fig. 11 is a flowchart of a routine that obtains frames from the data path and stores the frames to a buffer;

Fig. 12 shows a routine for retrieving and displaying a frame from a buffer;

Fig. 13A is a first diagram of data structures;

Fig. 13B is a second diagram of data structures;

Fig. 13C is a third diagram of data structures; and

Fig. 14 shows a routine for clearing a frame buffer.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

First, various schemes for formatting sequences of interleaved frames on a CD-ROM are discussed in part I. Next, a buffering system for using the interleaved frames in order to eliminate access time is described in part II. Finally, specific interleaving patterns for providing improved interactivity are presented in part III.

I. Interleaved Formatting.

Fig. 1 illustrates the prior art formatting of a CD-ROM. Fig. 1 shows sequences of frame information such as sequences 40, 42 and 44. These sequences are made up of frames such as first frame 54 and last frame 56 of sequence A. For ease of discussion, the frames and sequences are referred to by their alphanumeric labels. Thus, sequence A starts with frame A1 and ends with frame AX.

The animation sequences, or merely "sequences," may be frames in video, film, computer graphic, or other formats and are of arbitrary length. Any ordered sequence of frames that are played back to cause a viewer to perceive animation is referred to here as a "sequence." Further, these formats may be mixed together on a CD-ROM.

As shown in Fig. 1, sequence A is made up of frames prefixed by A with a frame number suffix, such as frames A1, A2, A3, ... AX, with AX denoting the final frame in sequence A. Similarly, sequence B is made up of frames B1, B2, B3, ... BX and sequence Z has frames Z1, Z2, Z3, ... ZX. The sequences may be of different and arbitrary lengths and there may be an arbitrary number of different sequences on the CD-ROM up to the maximum capacity of the CD-ROM.

Fig. 1 shows a prior art CD-ROM disc (or merely "CD-ROM") 50. The CD-ROM has a starting point on its spiral track at 52. At the starting point the frames of sequence A have been written on the CD-ROM so that frames A1, A2, A3, ... AX appear as the spiral is traversed outward. Following AX is the first frame of the next sequence, namely, sequence B so that frames B1, B2, B3, ... BX follow frame AX.

Note that Fig. 1 is only a conceptual illustration for ease of discussion. In actuality, each frame is a rather large linear track of information that is stored on the CD-ROM as a series of binary digits of data. For example, one popular frame format is 640 x 480 pixels of resolution per frame. Assuming each pixel requires 24 bits of information, this comes to 7,372,800 bits of data, or 900 kilobytes (kB) per frame. In Fig. 1, a frame is illustrated as a small block wherein it is actually an elongated portion of a very thin spiral track on the CD-ROM. Also, each CD-ROM has many more "loops" in its spiral track than are shown in Fig. 1.

Fig. 2 shows the formatting of a CD-ROM according to the present invention.

In Fig. 2, sequence A and sequence B are written to the CD-ROM with their frames interleaved. Thus, the track for the CD-ROM of Fig. 2 begins with frames as follows: A1, B1, A2, B2, A3, B3, etc. At some point in the interleaved series, sequence B will end since it is shorter than sequence A. This is shown just before the start of the second loop by the sequence A120, B120, A121, BX, A122, A123, A124, ... Note that after the last frame, frame BX, in sequence B multiple frames from sequence A follow in a non-interleaved fashion. Interleaving does not have to be present for an entire sequence, such as sequence A, on the CD-ROM.

Another interleaving possibility is shown after the start of the third loop in the series of frames A500, C1, A501, C2, A502, ... This shows a sequence C interleaved with the remaining frames of sequence A. Thus, a sequence may be interleaved to end within another sequence or to start within another sequence. The first example is shown where sequence B ends within sequence A and the second example is shown where sequence C starts within sequence A.

Other interleaving schemes are possible. For example, sequence A ends partway through the third loop at frame AX. After frame AX the frames C98, C99, C100, D1, E1, C101, D2, E2, C102, ... are written. This series of frames shows two sequences, sequence D and sequence E, both starting within sequence C. The series thus includes three sequences interleaved equally. Any number of sequences may be interleaved within a series of frames. The limitation on the number of sequences that may be interleaved depends upon the transfer rate of information from the CD-ROM to the application program performing the display function as described below and the rate at which frames are updated on the display.

Interleaving need not be uniform. For example, Fig. 2 shows the series D3, D4, E3, C103, D5, D6, E4, CX about three-quarters of the way through the third loop. This shows two frames from sequence D written to the CD-ROM followed by a frame from sequence E and a frame from sequence C. Thus, many variations of interleaving schemes are possible with the present invention.

One or more interleaving schemes may be used on a single CD-ROM. Also, the prior art formatting of non-interleaved sequences may be included with instances of interleaving as shown in Fig. 2 by non-interleaved sequence Z at the end of the track.

A "frame" need not consist only of image data. In a preferred embodiment, the interleaving schemes are used for audio data as well as for image data. A portion of an audio track is included in each frame corresponding to a time interval occupied by the display of the frame image data upon playback. For example, at the video rate of 30 frames per second (fps) an audio portion of 1/30 second of speech, sound effects, music, etc., can be included adjacent to its corresponding frame. Other methods of associating audio data with interleaved frames are possible.

Interleaving may occur with amounts of information less that a complete frame. Portions of frames may be interleaved so that, for example, where a frame is 640 columns by 480 rows, single rows or groups of rows from a frame of a first sequence may be interleaved with single rows or groups of rows from a frame of a second sequence. Also, information other than frame data can be included in each of the "frames" shown in Fig. 2 so that other useful information may be associated with, or read conveniently while reading, frame information.

One type of data that is useful to keep associated with frame information is a frame "tag" or identification (ID) number. A tag is useful to decide which frames to display and which frames to "drop" during frame selection as discussed below. Other types of information such as multi-frame branching information, buffering information, etc., can be put on the CD-ROM within or adjacent to the frames, or between series' of frames on the CD-ROM.

Interleaving is also used advantageously with compression techniques. As is known in the art, compression techniques reduce the amount of data needed to represent information such as a frame. Popular compression techniques such as Motion Picture Experts Group (MPEG) currently achieve about 25:1 compression ratios thus reducing the amount of data to about one twenty-fifth of the original requirement. Compressed frames may be interleaved as described above. Similarly, where portions of frames are interleaved the frame portions may be compressed. Any information needed by decompression routines to later decompress the information can be stored with the compressed information within the frame on the CD-ROM, between frames or in a different portion of the CD-ROM.

Some compression schemes are "inter-frame." This means that the compression/decompression ("codec") scheme of one frame depends on the data in a different frame. Where inter-frame compression is used it is usually more advantageous to do the compression between frames in the same sequence. That is, where two sequences are interleaved one frame at a time the inter-frame codec is best achieved by skipping frames so that, to the codec routines, it is as if the frames are non-interleaved. However, some advantages may be found by performing inter-frame codec on adjacent frames in an interleaved series. This is especially true where the interleaved frames differ only slightly from each other even though they are from different sequences. This is a common occurrence in interactive video productions using branch points where the action in the scene branches from a single frame into two gradually differing sequences.

The CD-ROM track need not be a spiral in order for the present invention to function. Any track arrangement that allows continuous reading of data from the CD-ROM in one mode is suitable for use with the present invention. For example, the CD-ROM track could have many concentric circles instead of a continuous spiral. This is similar to the format for magnetic media floppy and hard disk drives. Floppy and hard drives suffer from a similar, although less pronounced, problem in access time since they, too, use mechanical mechanisms to move read heads when it becomes necessary to look for data in different video paths. The advantage of interleaving is that it allows multiple sequences of animation to be available during playback of the CD-ROM (or other similar media) without requiring the slow mechanical repositioning of the read head. Thus the present invention is adaptable to a floppy or hard disk by formatting the floppy or hard disk in the interleaved manner as described herein.

Frame interleaving may also be used without a CD-ROM as where animated sequences are transmitted over a cable, fiber optic line, or other data link, or by electromagnetic transmission. For example, where an interactive video production is transmitted via cable to a user's home computer or television (computer/tv) frame interleaving could be used by alternating the transmission of frames in the transmitted signal. In other words, instead of writing the information to a CD-ROM disc the information could be transmitted in the ordering shown in Fig. 2 over a transmission medium. The user interaction with respect to decision points and the operation of the playback software at the user's computer/tv would be the same as where the information is read from a CD-ROM disc that is local to the user's computer as described above.

Fig. 2A is a simple block diagram of a system suitable for use in creating a CD-ROM with an interleaved frame format such as the frame format shown in Fig. 2.

In Fig. 2A system 500 includes a frame source 502, control unit 504 and CD-ROM record device 506. Frame source 502 may be a hard disk or other media where the frames of at least two separate animation sequences are stored in a digital format. Control unit 504 is preferably a digital computer executing software allowing a user to specify the any of the interleaving patterns discussed above. In a preferred embodiment, boxes within control unit 504 in Fig. 2A are implemented as software processes executed by a central processing unit (CPU) in a digital computer, such as the computer system described below in connection with Figs. 3 and 4. The software processes use various computer system resources such as buses, I/O ports, etc. to accomplish their tasks as is known in the art.

Selection process 508 chooses between frames from sequence A and B according to the user's specifications. Selection process 508 sends each selected frame to compression process 510. Compression is used in a preferred embodiment of the invention but is not necessary to practice the present invention. Compression process 510 compresses each frame according to any of a variety of known compression techniques, e.g., MPEG. Once compressed, write process 514 writes each frame to CD-ROM record device 506 according to the order determined by selection process 508. Suitable CD-ROM recording devices are, e.g., Kodak PCD Writer 200 or Philips CDD-521.

Thus, system 500 is capable of creating a CD-ROM disc in any of the interleaved formats discussed above.

Fig. 2B shows flowchart 200 illustrating the basic steps in creating a CD-ROM with interleaved formatting. The method illustrated in the flowchart of Fig. 2B may be used with the system shown in Fig. 2A.

The routine of flowchart 550 is entered at step 552 where it is assumed that multiple animation sequences made up of frames are residing on a storage device that is used as the frame source. At step 554 the user specifies the sequences to interleave and the type of interleaving. Next, step 556 is executed where a check is made as to whether the interleaving is completed. Assuming that there are still more frames to interleave, execution of the routine proceeds to step 558 where the next frame is obtained from the frame source according to the interleaving specifications selected by the user at step 554, above.

Once step 558 completes, step 560 is executed to compress the frame. Again, compression is optional. Next, the frame is written to the CD-ROM recording device at step 562 and steps 556-562 are repeated as necessary until the interleaved series consisting of the sequences selected by the user is completely written to the CD-ROM disc inside the CD-ROM recording device. When there are no more frames to write, execution of the routine of flowchart 550 terminates at step 564.

II. Frame Selection During Playback

During the playback of an interactive video production according to the present invention interleaved frames are read continuously from the CD-ROM disc. Frames from a desired sequence are ~~alternately~~ displayed while frames from undesired sequences in the same series as the desired sequence are not displayed. I.e., they are "dropped." ~~displayed, or "dropped."~~ In an ideal system, the rate of frames transferred from the CD-ROM drive would be constant and be a direct multiple of the display "frame refresh rate" so that, for example, where there are 5 possible sequences in a system where frames are displayed every 1/30th second the CD-ROM transfers 5 frames every 1/30th second and one of the 5 frames is displayed. However, because of variations in the transfer rate, system resource conflicts, limitations of the processing power, etc., a real-world implementation of the present invention buffers the frames prior to their selection and display.

The following discussion describes how a computer system is used to play back the frames stored in an interleaved pattern on a CD-ROM as discussed above. First, an overview of hardware and software used in the playback of an interactive video production is presented. Next, a way to select and drop ~~dropping~~ frames is discussed followed by a discussion of a specific implementation.

A. Overview of Hardware and Software in the Data Path

Fig. 3 is an illustration of a computer system suitable for use with the present invention. Fig. 3 depicts but one example of many possible computer types or configurations capable of being used with the present invention. Fig. 3 shows computer system 1 including display device 3, display screen 5, cabinet 7, keyboard 9 and mouse 11. Mouse 11 and keyboard 9 are "user input devices." Other examples of user input devices are a touch screen, light pen, track ball, data glove, etc.

Mouse 11 may have one or more buttons such as buttons 13 shown in Fig. 3. Cabinet 7 houses familiar computer components such as CD-ROM drives, disk drives, a processor, storage means, etc. As used in this specification "storage means" includes any storage device used in connection with a computer system such as CD-ROM drives, disk drives, magnetic tape, solid state memory, bubble memory, etc. Cabinet 7 may include additional hardware such as input/output (I/O) interface cards for connecting computer system 1 to external devices such as external CD-ROM drive 15, or other devices (not shown) such as an optical character reader, external storage devices, other computers, etc.

Fig. 4 is an illustration of basic subsystems in computer system 1 of Fig. 3. In Fig. 4, subsystems are represented by blocks such as central processor 20, system memory 21, display adapter 22, monitor 23, etc. The subsystems are interconnected via a system bus 24. Additional subsystems such as a printer, keyboard, fixed disk and others are shown. Peripherals and I/O devices can be connected to the computer system by, for example, serial port 25. For example, serial port 25 can be used to connect the computer system to a modem or mouse input device. External interface 27 connects CD-ROM drive 28 to the bus. The interconnections via system bus 24 allow central processor 20 to communicate with each subsystem and to control the execution of instructions from system memory 21 or fixed disk 26, and the exchange of information between subsystems. Other arrangements of subsystems and interconnections are possible such as direct memory access (DMA) controllers, co-processing units, etc.

Fig. 5 is a block diagram of a computer system suitable for playing back video data according to the present invention.

In Fig. 5, system 100 includes ~~shows~~ the hardware and software through which information from CD-ROM 104 passes before it is displayed on display 120. In general, boxes with square corners in Fig. 5 denote hardware systems or components while ~~and~~ boxes with rounded corners denote software programs or processes (also referred to as "software components") executing in a computer system such as computer system 130. The processes reside in RAM 132 and are executed by CPU 124. The general flow of information is shown by lines with arrowheads and proceeds from left to right in Fig. 5. This flow of information from CD-ROM disc 104 to display screen 120 is referred to as a "data path." There are many possible configurations of hardware and software that will differ from that shown in Fig. 5. There are also a variety of data paths within any given configuration. The apparatus and methods of the present invention are adaptable for use at one or more points of any possible data path.

Information is read from CD-ROM 104, by controller 106. Controller 106 may include software and RAM and may, itself, be a computer system. Controller 106 transfers the information to CD-ROM bus interface 108. Bus interface 108 may be one of a variety of bus interfaces such as a small computer system interface ~~as small computer systems interface~~ (SCSI) adapter. Bus interface 108 transfers the information onto a bus within computer system 130 (not shown in Fig. 5) so that the data is available to software driver 110.

Driver 110 is usually accessed by operating system 112 to transfer data to application 114. However, some application programs bypass the operating system and access the driver directly. Also, some application programs incorporate their own drivers and may access I/O ports directly without going through an external driver such as driver 110.

Application 114 transfers data between itself and various buffers such as buffers 128. Application program 114 can also communicate with operating system 112 for many purposes such as storing information on disk drive 122, receiving signals from user input devices such as mouse 123 and keyboard 125, etc.

Other programs or processes, such as process 126, may also be executing in RAM 132 and in communication with application program 114. An example of a process is a so-called "ramdisk" that acts as an auxiliary cache or buffer and may transfer information with application 114. Another example of a process or application is a word processor, spreadsheet program, etc., that need not be in communication with application 114.

Application program 114 will typically access operating system 112 to write data to driver 116 in order to display information via video adapter 118 onto display screen 120. Again, information is transferred between driver 116 and video adapter 118 via a bus not shown in Fig. 5. Usually, CPU 124 is used to accomplish the transfer of information between driver 116 and adapter 118.

As mentioned above, each hardware system in Fig. 5 may include software as part of its operation. Software and hardware boxes may be added to or omitted from the data path shown in Fig. 5. Depending on compatibility versus speed considerations, an application program may not use the operating system to communicate with peripherals such as disk drive 122, CD-ROM drive 102 and display 120. Also, hardware systems may be employed in place of software processes if speed is more important than flexibility. For example, if DMA techniques are used by bus interface 108 then driver 110 is not needed and may be replaced by a buffer area in memory. In this case, a DMA controller is used to place data into the buffer area.

B. Selection of Frames

The present invention uses frame selection methods to discard frames during playback so that only frames of a desired sequence are displayed. These frame selection methods may be implemented in hardware or software at any point in the system of Fig. 5 from controller 106 to video adapter 118. In a preferred embodiment, the frame selection is made under the direction of application program 114. While this sacrifices speed and efficiency it allows the interactive video production to be compatible with standard operating systems such as Microsoft Windows 3.0 by Microsoft, Inc., or the Macintosh operating system by Apple Computer, Inc. A description of these operating systems may be found in Microsoft Windows User's Guide Version 3.0 by Microsoft Corporation, and Macintosh User's Guide, by Apple Computer, Inc. Performing frame selection at the application program level allows the features of the present invention to be transparent to existing computer systems, prevents conflicts among other software that may be executing in the computer system and is in keeping with standard computer hardware and software practices. Further, it allows the present invention to be practiced without modifying existing hardware or software that consumers may already own.

Figs. 6A and 6B show flowcharts of methods of the present invention to select frames from a series of frames read from a CD-ROM.

In general, the flowcharts in this specification illustrate one or more software routines executing in a computer system such as computer system 130 of Fig. 5. The routines may be implemented by any means as is known in the art. For example, any number of computer programming languages, such as "C", Pascal, FORTRAN, assembly language, etc., may be used. Further, various programming approaches such as procedural, object oriented or artificial intelligence techniques may be employed.

The steps of the flowcharts may be implemented by one or more software routines, processes, subroutines, modules, etc. It will be apparent that each flowchart is illustrative of merely the broad logical flow of the method of the present invention and that steps may be added to, or taken away from, the flowcharts without departing from the scope of the invention. Further, the order of execution of steps in the flowcharts may be changed without departing from the scope of the invention. Additional considerations in implementing the method described by the flowchart in software may dictate changes in the selection and

17

order of steps. For example, event handling may be by interrupt driven, polled, or other schemes. A multiprocessing or multitasking environment could allow steps to be executed "concurrently." For ease of discussion the implementation of each flowchart is referred to as if it is implemented in a single "routine".

Fig. 6A shows a flowchart for a routine 200 to select the "current sequence" according to user input. Routine 200 may be implemented by a variety of means as is known in the art, such as by using operating system routines to accept keyboard or mouse signals at predetermined times. The accepted signals are processed to set a variable to one of several states that indicate a current sequence such as sequence_A or sequence_B. In a preferred embodiment, routine 200 is invoked by an interrupt and entered at step 202 when a user input is detected. Routine 200 runs concurrently with the routine of Fig. 6B.

Routine 200 is greatly simplified. At step 204 user input is accepted. At step 206 the user input signals are used to set a variable indicating the current sequence. For example, at a decision point the user has the opportunity to enter a response that selects between two paths, or sequences of images. If the user inputs "yes" at the keyboard the value for sequence A is assigned to the current sequence variable. If the user does not enter "yes" the value for sequence B is assigned to the current sequence variable. After the current sequence has been updated, routine 200 exits at step 212.

Fig. 6B shows a flowchart for a routine 220 to select and display frames according to the state of the current sequence variable as set by routine 200. Routine 220 executes concurrently with routine 200 so that the current sequence is selectable while a sequence is being displayed. Routine 220 is also greatly simplified and may be implemented by a variety of means. A preferred implementation of key steps in routine 220 of Fig. 6B is discussed below.

Routine 220 is entered at step 222 where it is assumed that frame data is being read from a CD-ROM formatted in an interleaved manner as described above. At step 224 the next frame is obtained from the data path. Assuming routine 220 is implemented in an application program such as application program 114 of Fig. 5, the next frame will reside in a buffer such as buffers 128 after having been placed there by other routines in application 114 in concert with operating system 112, driver 110, bus interface 108, etc., as is known in the art. However, certain advantages may be realized by implementing routine 220 at different points in the data path as discussed below.

At step 226 a check is made as to whether the next frame is part of the current sequence as specified by the value of the current sequence variable set by routine 200. If so, execution proceeds to step 228 where the frame data is decompressed, if necessary. Next, the decompressed frame is displayed at step 230 so that the sequence is presented to the user as an animation. Execution returns to step 224 for subsequent frames.

If, at step 226, it is determined that the next frame is not part of the current sequence then execution proceeds back to step 224 where the next frame is obtained for similar checking. In short, only those frames that are part of the current sequence are decompressed and displayed so that only the selected sequence is presented to the user. The association of frames with sequences can be by tags as described above, where frames in a given sequence include an identifying symbol such as the letter A for frames in sequence A. Other methods of association are possible, such as where two sequences are interleaved frame-by-frame so that it is known, once the series starts, that alternating frames belong to the same sequence.

The selection methods of Figs. 6A and 6B allow an application program to have instant access to different sequences of frames on the CD-ROM disc. Thus, the time to switch between sequence A and sequence B is not limited by the time to mechanically reposition the read head in the CD-ROM drive and is instantaneous insofar as the frame rate is concerned since the first frame of a different path is displayed immediately after the current frame of the current path. Depending on the refresh rate of frames on the display, the transfer rate of data from the CD-ROM, the frame size and the compression and decompression schemes, any number of sequences may be interleaved and instantly selected during the playback of an interactive video while still maintaining visual continuity at high refresh rates of, e.g., 24 fps, 30 fps or more.

Where the method of Figs. 6A and 6B is implemented at the application level in, for example, application 114 of Fig. 5, each frame from the CD-ROM will most likely have to be read into a buffer such as buffers 128. In other words, even frames that will be dropped must be transferred from the CD-ROM disc to buffers in RAM. This is not a huge drawback since the CD-ROM drive and data path are designed to operate optimally in transferring continuous data to buffers. In a preferred embodiment the frames are also compressed. This means that dropped frames only impact the data transfer one twenty-fifth (assuming 25:1 compression) as much as they would have assuming no compression.

Much of the following discussion deals with "frame refresh intervals," or "intervals". An interval is merely the time between successive displaying of frames on the display screen and is used for convenience since, where the interval is fixed, describing the timing for an interval accurately describes the timing of the entire interactive production.

In a preferred embodiment the application program can make an instant decision on whether or not to decompress and display a frame that is only partially loaded into a buffer if the tag or ID number is placed at the beginning of frame data. For frames not part of the current sequence the application program, at step 226 of Fig. 6B, reads the tag from the first few bytes of the buffer and knows that the frame is not to be displayed. The application program is then free to do further processing while the ignored frame finishes being loaded into the buffer and the next frame in the selected sequence completes loading into the buffer. Other ways to optimize the execution of the application program and the selecting and dropping of frames at the application program level are possible.

The method of Figs. 6A and 6B may also be implemented in various other stages of the data path. For example, if the method is implemented in bus interface 108 it is possible to prevent frames that would otherwise be dropped from being loaded into RAM. This frees up bus time and wasted processing time in executing driver 110 and operating system 112 instructions. However, the drawback is that this requires a modification to the bus interface card unlike the software-only solution described above where the method is implemented by an application program.

Similar design tradeoffs are seen depending on where the method is implemented in the data path. For example, if the method is implemented in controller 106 the highest degree of efficiency is achieved since unused frames are dropped immediately and the rest of the data path sees only frames that are part of the current sequence and, therefore, will be displayed. This requires customizing the CD-ROM drive, however, and requires communicating parameters, such as the current sequence variable, from the application program to controller 106. The method could also be implemented in operating system 112, but this requires changing the operating system standard.

Other distinct advantages of the present invention are achieved where caching is used in the data path. In the example system of Fig. 5, look-ahead caching is typically used at several points.

Controller 106 typically includes a small cache to account for fluctuations in data transfer in reading CD-ROM 104. Caches also are usually present at bus interface 108, operating system 112 and application 114. Other caches could exist where the user's system includes, e.g., a ramdisk such as at application 126, or operating system caching to a hard disk such as disk drive 122. Additionally, CPU 124 typically implements on-chip caching. Other caches may exist at various points in the system such as within disk drive 122, itself.

The present invention makes optimal use of look-ahead caching wherever it's used since the data is always read from the CD-ROM sequentially. Thus, the transfer rate for information in the data path is at its maximum rate allowed by the various hardware and software components in the data path. The method of the present invention may be applied to the caches themselves by treating the caches as if they are buffers such as buffers 128 discussed in the above example. In other words, frame data is placed into a cache and a determination is made whether the frame is part of the current sequence. If not, the frame is dropped by purging the cache or merely reusing the data area of the cache occupied by the frame to store a subsequent frame.

Many specific implementations of the method of Figs. 6A and 6B are possible. For example, at the application level buffers can be used to contain the frames. The buffers may be updated with frames by a write routine while a read routine selects the proper frames for display and "passes over" or drops frames not in the selected current sequence. Such an implementation is described in detail in the next section of this specification.

The present invention has certain advantages when an interactive production is delivered to multiple users over a transmission channel as opposed to a CD-ROM drive. To illustrate this, the CD-ROM drive in Fig. 5 may simply be replaced by a transmission channel such as a wire cable, fiber optic link, electromagnetic broadcast, etc. Frames may then be transmitted over the channel in an interleaved pattern at a constant data rate.

The advantage of using the present system when data is sent over a transmission channel is that the data may be broadcast to multiple users at the same time without compromising the interactivity of the production, increasing the complexity of processing or increasing the bandwidth required to achieve interactivity. The data can be sent in real time, much like a standard television broadcast, while the user interacts with the transmitted frames by selecting the current sequence from the multiple sequences in the interleaved frames. This differs from other schemes providing interactivity where multiple channels are used for multiple users and there is no requirement to keep multiple channels synchronized to each other.

C. An Application Level Selection Implementation

Next, Figs. 11, 12 and 13A-C are discussed to describe an implementation of the present method using buffers at the application level.

Fig. 11 is a flowchart of a routine 240 that obtains frames from the data path and stores the frames to a buffer. Routine 240 operates on data structures such as those shown in Figs. 13A-C. In Fig. 13A pointers at 280 each point to a slot in buffer 282. Each slot in buffer 282 is large enough to contain a frame of data along with any associated information such as the frame tag and codec information. Buffer 282 has 8 slots numbered 0-7. The buffer size may be larger or smaller depending on the demands of the system and the interactivity requirements as discussed below. The number of pointers used may vary. Additional pointers are shown in Fig. 13A such as sequence_C and sequence_D pointers which do not have any list of slots associated with them. Therefore, these sequences are not available for the user to select during playback of the interactive production. However, they may become available at a later time when frames designated for these sequences are obtained from the data path and placed into the buffer.

Pointers 280 each point to the beginning of a linked list of slots although the list pointed to by a pointer may be empty, or null. Each slot in a given list includes a pointer to the next slot in the list. Thus, the slots in a list may be in arbitrary order with respect to the slot numbering. This is a so-called "singly" linked list. Refinements on this approach, such as by using back pointers between slots in the list to create a "doubly" linked list, and "tail pointers" to show the end of a list may be used. Many modifications to the present buffering scheme are possible. Also, many different ways of implementing the system of the present invention are possible, the use of buffers as described here being but one possible way.

Routine 240 of Fig. 11 is entered at step 242. Routine 240 is called often enough to obtain each frame from the data path without missing frames as they are read off of the CD-ROM disc at a more or less constant rate. At step 244 a check is made as to whether there is a free slot available in the buffer. Using the pointer scheme shown in Fig. 13A this merely requires looking at the pointer called "free" to see whether it points to a slot or not. If no slot is available, an error condition is flagged at step 258 and routine 240 exits at 256.

With the pointer values as shown in Fig. 13A, slots 0 and 4 are on the free list. In other words there are two free slots so execution proceeds from step 244 to step 246 in routine 240 of Fig. 11. At step 246 a frame is obtained from the data path. Obtaining a frame is dependent on the components used in the data path. One way to obtain a frame is to copy the frame from an auxiliary buffer used by operating system routines (or other hardware and software) after the routines have transferred the data from the CD-ROM. Alternatively, some systems allow a pointer to an area of memory, such as slot 0 of buffer 282, to be passed to hardware or software components whereupon the area of memory is filled with the data from the data path, e.g., by DMA transfer or by other means. In the latter case, the steps of flowchart 11 need to be modified in order to reflect the automatic placement of data into the buffer.

Assuming that the frame data must be copied from an auxiliary buffer, routine 240 of Fig. 11 executes step 248 to determine the animation sequence that the frame belongs to. In a preferred embodiment this is done by examining a tag or other identification associated with the obtained frame. At step 250 the frame is stored into a free slot. In the present example the first slot in the free list, slot 0, is used. Next, step 252 is executed in routine 240 so that slot 0 is added to the list for its associated sequence. Assuming the obtained frame belongs to sequence A, Fig. 13B shows the obtained frame added to the sequence_A list so that the list of slots is now 1, 3, 2, 5, 0. Step 254 in routine 240 removes slot 0 from the free list so that the free list now contains only slot 4. Finally, routine 240 of Fig. 11 is exited at step 256.

Fig. 12 shows a routine 260 for retrieving and displaying a frame from the buffer. In Fig. 12, routine 260 is entered at 262. Routine 260 is typically called after each frame refresh interval at a fixed frame rate where interactive video is used. However, routine 260 need only be called often enough to present smooth information to a user of the interactive production and the frame rate may be of varying rates in different productions and even within a given production.

At step 264 routine 260 checks the pointer for the list of slots of frames that corresponds to the current sequence. In the present example sequence A is the current sequence (as indicated by a current sequence variable, discussed above) so pointer sequence_A in Fig. 13B is checked. Step 266 determines whether or not the list pointed to by sequence_A is empty. If the list is empty then step 278 is executed which flags an error condition and the routine is exited. At this point in time, pointer sequence_A points to a list of slots 1, 3, 2, 5, 0 so step 268 is executed to decompress the next frame in the list, namely, the frame in slot 1.

After the frame in slot 1 is decompressed the frame information is displayed on the display screen at step 270. At step 272 the pointer sequence_A is set to point to the next frame in the list. Step 274 adds slot 1 to the free list so that it may be reused. At this point, the pointer and buffer arrangement is as shown in Fig. 13C.

Thus, the description above illustrates how frames from the data path are written to a buffer and read from the buffer and displayed.

A deficiency exists in the system described so far since, in a preferred embodiment, one frame per refresh interval is being displayed to the user while multiple frames are being read from the CD-ROM disc and stored to the buffer in any given frame refresh interval. This means that the buffer will quickly fill up.

Fig. 14 shows a routine 400 for clearing the buffer based on outdated frames or some other criterion. Routine 400 is entered at step 402 and is called frequently enough, e.g., once every refresh interval, to remove unneeded frames from the buffer. At step 404 a check is made as to whether there are enough free slots in the buffer. If so, the routine is exited at step 412. It is desirable to keep as many frames in the buffer as possible since this is more efficient use of buffer space and since there may be some use for frames, even outdated frames, by an application program.

At step 406 a check is made as to whether any outdated frames are in the buffer. Outdated frames are detected by including a frame number with each frame within the frame's tag. Frames from any sequence that have passed their opportunity for display will be discarded from the buffer by removing the frame from the list and adding the frame's slot to the free list. The discarding step is performed at step 408 of Fig. 14. Should none of the frames be outdated then a frame is discarded from the buffer according to some other criterion at step 410. For example, one criterion is to discard a frame that it is known will not be used because, e.g., the sequence has become unselectable at some time after the sequence frames were stored into the buffer. Other criterion are possible to free up a slot. For example, a slot may simply be discarded even though the need for the slot's frame at some later time may produce a "dropout" on the display.

In general, the design of a buffering scheme for frames at the application level is affected by many factors. In an ideal system, the frames are provided by the data path in perfect synchronization with the refresh rate so that no buffering is necessary. However, because of errors in hardware such as misreadings from the CD-ROM drive, sharing of resources such as waiting for a system bus to transfer data, limitations on the system such as having a single CPU executing several processes, etc., a buffer is necessary to account for fluctuations in the frame delivery rate of a data path. Also, buffering provides the ability for an interactive production to access multiple frames at any given frame interval to facilitate performing some of the operations discussed below in connection with interleaving techniques.

III. Interleaving Techniques

Next, different patterns of interleaved frames that greatly improve interactivity are discussed. Specific patterns that implement the following modes of interactivity are presented: Simple path switching, time stepped path switching, single branch point transitions and multi-branch point transitions.

A. Simple Path Switching

Fig. 7 shows three timelines corresponding to three different sequences of animation. The topmost timeline shows a sequence A that includes frames A1-A7. Only a portion of the timeline is shown since it continues off to the right. The display of each frame in sequence A is indicated along the timeline as a tic mark above each frame label. The interval 302 for displaying each frame depends upon the frame refresh rate. A typical rate is 30 fps for video or 24 fps for film. However, in contemporary computer productions this rate may vary. For ease of discussion we assume a constant desired frame rate of 30 fps. This means that the frame interval is 1/30th of a second. Thus, each frame is displayed on the display screen for about 1/30th of a second.

Timelines for sequence B and sequence C are similarly shown. Since sequence B and sequence C are also assumed to have frame rates of 30 fps, the tic marks for sequence B and C line up with the marks for sequence A.

CD-ROM track 300 is shown at the bottom of Fig. 7. CD-ROM track has all of the frames shown for sequences A, B and C on its single track in the interleaved fashion described above. CD-ROM track 300 is not a timeline but is a one-dimensional (or nearly so) spatial arrangement of frames from the sequences shown linearly in Fig. 7. In a preferred embodiment, track 300 plays back so that the frame rate of the data transfer from the CD-ROM drive to an application program is three times the 30 fps rate, or 90 fps. This can be achieved by a combination of codec schemes and drive transfer rate. Currently available codec schemes allow a 30 fps rate on a single speed CD-ROM drive. This means that triple speed CD-ROM drives now available will provide 90 fps, or three separate frame sequences when the system of the present invention is used. The number of allowable frame sequences will increase as drive speeds increase and codec schemes improve.

An interactive video production with an interleaved track as shown in Fig. 7 allows a user to instantly switch between any of the three sequences A, B and C. This is useful, for example, where each sequence focuses on a different character in a plot and the characters are in different locations. The user is able to select which character to observe. Note that the user is able to make the selection at any point in time so that, as opposed to traditional interactive video productions, the user is provided with unlimited interactivity insofar as observing different characters, moving to different locations, etc., is concerned. The switching between sequences occurs instantaneously. In other words, the frame rate of 30 fps is uninterrupted regardless of how much switching between sequences occurs. In a preferred embodiment, a short visual transition or "segue" is used so that a signal indicating the switch in sequences is provided to the user.

One problem with the simple path switching of Fig. 7 is that some codec schemes rely heavily on frame to frame similarities within a sequence in order to achieve good compression. When a frame's decompression uses data from a different frame it is called inter-frame compression as opposed to intra-frame compression which only uses the information within a given frame to compress and decompress the frame. Inter-frame compression has advantages in animation sequences since there are usually many similarities from one scene to the next if only part of the images in the scene are moving. However, in an interactive video production where the user is given the ability to switch randomly between different sequences the inter-frame compression scheme will suffer from inefficiencies.

To illustrate this, assume that frames A4 and A5 have been compressed using MPEG inter frame compression. This means that frame A5 will have decompression information associated with it that depends on the information from frame A4. However, if frame A5 is branched to from sequence B or C then A4 will never have been processed by the playback computer system so that decompression of frame A5 is impossible. Indeed, frame A4 may be compressed according to frame A3 and so on.

Fortunately, the MPEG codec scheme provides that intra-frame compression occurs quite regularly and relatively frequently compared to human response times. An intra-frame compression occurs about once every 0.4 seconds (assuming 30 fps). This translates to one of every 12 frames in a sequence being compressed purely based on information within the single frame. Once a user selects a branch to a different sequence, the playback system need only wait up to a <u>maximum</u> of 0.4 seconds until the next frame in the different sequence with intra-frame compression comes up in order to make the transition. This delay can be reduced with different codec schemes. Even with the 0.4

second delay there is an improvement over the 0.3 or greater average time to access information using traditional CD-ROM access methods. In practice, the access time in traditional CD-ROM drives is at its worst in interactive video productions since the long video sequences mean that the access time will almost always be close to its worst, at about 0.5 to 1 second for some drives.

For example, if frames A2-A6 are compressed using inter-frame compression and frames A1 and A7 are compressed using intra-frame compression then, should a user decide to transition from sequence B to sequence A just after frame B1 has been displayed, the playback system need only wait until frame A7 is available from the data path. This is a delay of only about 6/30ths of a second. This delay will be reduced if the data transfer rate from the CD-ROM is increased. In the present example, if the data transfer rate is doubled, that is, if 180 fps can be transferred from the CD-ROM either through improved transfer speed or codec schemes, then the worst-case delay will be halved to about 1/10 second.

Alternatively, the last twelve frames of every sequence could be kept in a buffer so that inter-frame decompression could always be carried out, although the time to compute decompression for twelve frames may be lengthy. Other solutions are to use only intra-frame compression techniques or to decompress all sequences on the fly, even those sequences not being used.

B. Time Stepped Path Switching

Time stepped path switching creates a problem in switching between different sequences when there are one or more same events that are present within two or more sequences. For example, in an interactive production where the user is able to be present in different rooms in a house in order to observe different characters there could be a ringing doorbell that is heard throughout the rooms. A second event could be someone announcing over an intercom system that they will get the door.

However, if these sequences are originally filmed or taped with live actors in real time (as opposed to computer generated characters) then it is unlikely that the spacing of the two events will occur in the same number of frames in each of the sequences. In other words, it is likely that there will be a difference in the two events of greater than 1/30 second. This could result in the user hearing a partial repeat of the announcement over the intercom should the user switch from a first sequence to a second sequence where the announcement came a fraction of a second later in the second sequence. While such a discrepancy could conceivably be corrected by shifting the audio track within the sequences, it may arise that characters or objects on the screen are affected in accordance with the sound events. For example, the announcement may be heard in the first sequence and then the user may switch to the room where the person making the announcement is seen speaking into the intercom. Other visual events could occur that need to be coordinated among the different sequences such as a flash of lightning that's visible in multiple rooms.

One approach to correct this is to use time stepped path switching patterns. This is a way to compensate for differences in the number of frames between two same events on different sequences. Time stepped path switching requires a faster data transfer rate than simple switching to allow for one or more sequences to "catch up" with other sequences.

Fig. 8 shows three timelines similar to those of Fig. 7. In Fig. 8, the timeline for sequence A has a first event, event 1, occurring at frame A11. Event 2 occurs at frame A18. In sequence B, event 1 occurs at frame B11 but event 2 occurs at frame B20. Thus, there is a two frame difference in the interval of the events in sequence B from sequence A. This small discrepancy is used here for ease of discussion. In practice much larger discrepancies can be handled by the system described here as will be apparent to one of ordinary skill in the art.

In sequence C, there are only 6 frames between the occurrences of events 1 and 2.

CD-ROM track 310 is shown at the bottom of Fig. 8. At interval 312 frames A9, B9 and C9 are stored, similarly to the format of Fig. 7. However, the bandwidth, or transfer rate, of track 310 is greater than that of Fig. 7 to allow at least one more frame per interval to be stored. Thus, the transfer rate of the playback of track 310 is at least 120 fps although the instant rate for interval 312 is only 90 fps since only three frames are stored in the interval.

At interval 314 event 1 has occurred. At interval 314 special use is made of the ability to store an extra frame by having frame B12 stored in the interval along with frames A11, B11 and C11. It can be seen that extra frames from sequence B are stored in four intervals of track 310 between events 1 and 2. This accounts for the greater number of frames in sequence B between the two events. Thus, frame B20 is available for playback at the position of event 2 on track 310. Similarly, for sequence A there are additional frames inserted so that, at event 2, frame A18 is available.

The requirement for increased bandwidth is calculated as the number of extra frames from all sequences (besides the shortest one) between the two events divided by the number of frames in the shortest sequence between the two events. This yields the number of extra frames per interval that must be stored. In the present example the calculation is (2+4) / 6 = 1. This is in accordance with the ability to store an extra frame per interval in track 310 when time stepped path switching is performed.

One problem with time stepped path switching is that a large number of frames may have to be buffered. Where frames for a given sequence are being transferred from the CD-ROM at faster than the display frame rate the additional frames must be held in a buffer until they can be used. Each time two frames from sequence B are obtained in an interval one of the frames must be stored. This, of course, assumes that sequence B is currently selected and is being displayed. Otherwise, the additional sequence B frames, indeed all sequence B frames, may simply be dropped according to the system described above. For short discrepancies between sequences this is not a problem. The current example, only requires that two frames from sequence B be buffered. However, where the discrepancy is two seconds, this method requires that 30 frames be buffered somehow. One possibility is to spool the buffered frames to a hard disk but this depends on disk storage availability. Another solution is to insure that such large discrepancies don't exist in the sequences when the sequences are filmed.

Once event 2 has been reached on track 310 it may be necessary to re-synchronize the three sequences. This can be done in a similar manner to the discussion above except that it is sequence C's along with sequence A's frames that are "doubled up" in an interval to catch up to sequence B's frame numbers. This is shown on track 310 subsequent to event 2. Again, buffering is necessary where multiple frames from a displayed sequence are available within a given interval. The three sequences are synchronized again at interval 314 and only three frames per interval need be stored thereafter.

The method presented in Fig. 8 is a general way to time shift two sequences with respect to each other. This may be useful for a variety of purposes in interactive productions. The additional frame capacity can be used for other purposes when not used for time stepped path switching patterns.

C. Single Point Transitions

A single point transition is shown in Fig. 9. A single point transition is similar to the simple path switching pattern of Fig. 7 except that multiple sequence data does not appear on track 340 until the decision point at 342, or frame A2, is reached. As before, track 340 represents a track on the CD-ROM that is played back at a transfer rate that allows multiple frames to be available during an interval corresponding to the interval used to display a frame from the current sequence on the display screen. The transfer rate does not have to be a multiple of the frame display rate. In a preferred embodiment, the data transfer rate allows three frames to be obtained within each interval. For convenience, each frame from sequence A is the start of what is referred to as an "interval" on track 340. Such as interval 344 at frame A3.

At decision point 342, the user is able to affect the interactive video by causing either of sequences B or C to be played. An example of a single point transition is where a character can perform one of two or more actions such as continuing to type a patent at a word processor, smashing the keyboard of the word processor or getting up and going home. In the example, the action of continuing to type is sequence A. The action of smashing the keyboard is sequence B and the action of getting up and going home is sequence C.

Though these actions are all very different, they start from a single common frame at A2. That is, frame A2 is the starting point for each of the three sequences beginning with A3, B3 and C3. Frame A2 could show, for example, the character sitting down and typing at a keyboard. Frames A3 onward in sequence A would continue to show the character sitting down and typing. Frame B3 would show the character beginning to clench his or her fists and subsequent frames in sequence B would show the fists being raised and then lowered violently into the keyboard. Similarly, frame C3 would be a first frame where the hands are coming to rest as the character rises.

Since each of the sequences branches from frame A2 the inter-frame compression problem of the simple path switching pattern discussed above is not a factor with single point transitions (or multi point transitions, discussed below). However, the user is limited in only affecting the interactive video at a specific instant in time. In other words, even if the user decides at some point prior to A3 that the character should get up and go home this effect will not be displayed until frame A3 is reached. In order for the user to believe that they are affecting the video the delay between the user selecting an action and having the action performed should not be more than a small fraction of a second, yet the user should be able to make the selection at any point within a relatively long period of time. Such are the features provided by multi point transitions, discussed next.

D. Multi Point Transitions

Multi point transitions allow a user to make a selection that switches to a different sequence over a relatively long period of time while having the image on the screen respond to the user's selection rapidly. The drawback of multi point transitions is that they require a high bandwidth meaning that a high transfer rate or superior codec scheme must be employed.

Fig. 10 illustrates a pattern to implement multi point transitions. In Fig. 10, sequences A, B and C are shown from top to bottom as in Fig. 9. Track 360 is a track on the CD-ROM. The decision point is at 362 where the user's actions determine which of sequences A, B or C will be chosen for subsequent playback as the current sequence. However, Fig. 10 also shows additional transition points at 364. During transition points 364 the user's actions will cause an immediate change in the frames being displayed. The additional transition points are shown in a broken area to indicate that the transition points may extend over an arbitrary length of the track. In this discussion we assume there are 5 transition points in addition to the decision point at 362.

Boxes at 366 and 368 show the number of frames to transition from a given frame in sequence A to sequences B and C, respectively. Thus, at frame A95 there are 20 frames that need to be generated in order to transition to frame B101 if the user selects sequence B at the time frame A95 is displayed. Similarly 18 frames are needed to go from frame A96 to frame B101, 19 frames are needed to go from frame A95 to frame C101, 15 frames are needed to go from frame A96 to frame C101, etc. Note that the number of frames needed to transition generally decreases as the decision point is neared although this isn't necessarily true. For example, where the transitions are as discussed above, i.e., the character is continuously typing in sequence A, in sequence B the character smashes the keyboard and in sequence C the character gets up and goes home, the characters hands may be closer to the clenched fist position in frame A97 in frame A98. This explains why 8 frames are needed to transition from frame A98 to sequence B while only 4 frames are needed to transition from frame A97 to sequence B.

Also, while the sequence of typing is fairly uniform (and boring) in that the character displays few, if any, major movements, such actions as the character moving a hand to do a shift operation to capitalize a letter, or to move a hand off of the keyboard entirely to pause for a second could require a sequence of frames to bring the hand back to the keyboard position in order to match up with the clenched fist frames of sequence B.

The total number of frames in the multi point transitions from sequence A to B or C is 10+9+4+8+5+8+5+9+5+2 = 65 frames. These must be written to the CD-ROM disc in the six intervals at 370. In order to achieve this density there must be an average of about 10.8 frames per interval. This is far more than the four frames per interval previously used in the time stepped path switching discussed above.

The density of 10.8 frames per interval can be reduced by only permitting transitions every other frame. This would expand the number of intervals by a factor of about 2 so that the frame density is about 5.4 frames per interval. The tradeoff is that the response to the user's input has a delay of 1/15 second. Expanding the number of intervals by 4 reduces the density to 2.7 frames per interval and makes the delay about 0.13 seconds. Re-synchronizing and time shifting of the transition sequences would be required along the lines of the discussion above in connection with time stepped path switching and Fig. 8.

In the foregoing specification, the invention has been described with reference to a specific exemplary embodiment thereof. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the appended claims. For example, various programming languages and techniques can be used to implement the disclosed invention. Also, the specific logic presented to accomplish tasks within the present invention may be modified without departing from the scope of the invention. Many such changes or modifications will be readily apparent to one of ordinary skill in the art. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense, the invention being limited only by the provided claims.

IN THE CLAIMS:

Sub D1

~~1.~~ A method for creating an interactive production on a CD-ROM, comprising the following steps:

creating a first animation sequence of digital frames;

creating a second animation sequence of digital frames; and

writing the first and second animation sequences of frames to the CD-ROM by interleaving the frames of the first animation sequence with the frames of the second animation sequence to create the interactive production.

2. The method of claim 1, wherein a computer system is used to play back the interactive production, wherein the computer system comprises a processor, user input device and display screen, wherein the computer system is coupled to a CD-ROM drive, the method further comprising the following steps performed under the control of the processor:

continuously reading the interleaved frames from the CD-ROM;

displaying only the frames of the first animation sequence on the display screen to play back the first animation;

accepting signals from the user input device selecting the second animation sequence; and

in response to the signals from the user input device, displaying only the frames of the second animation sequence on the display screen to play back the second animation.

D

2 ~~3~~. The method of claim 1 ~~2~~, further comprising the following steps:

prior to the "writing the first and second animation sequences" step, compressing one or more of the digital frames;

wherein the "writing the first and second animation sequences" step further includes the substep of writing the compressed digital frames to the medium ~~CD-ROM;~~ and decompressing a frame prior to displaying the frame on the display screen.

Sub B2

~~4.~~ A method for playing back an interactive production recorded on a CD-ROM, wherein the CD-ROM includes a first animation sequence of digital frames interleaved with frames from a second animation sequence of digital frames, wherein a computer system is used to play back the interactive production, wherein the computer system comprises a processor, user input device and display screen, wherein the computer system is coupled to a CD-ROM drive, the method further comprising the following steps performed under the control of the processor:

continuously reading the interleaved frames from the CD-ROM;

displaying only the frames of the first animation sequence on the display screen to play back the first animation;

accepting signals from the user input device selecting the second animation sequence; and

in response to the signals from the user input device, displaying only the frames of the second animation sequence on the display screen to play back the second animation.

5. An apparatus for playing back an interactive production stored on a CD-ROM, wherein the CD-ROM includes frames corresponding to a first animation sequence interleaved with frames corresponding to a second animation sequence to produce a series of frames wherein adjacent frames in the series correspond to different animation sequences, the apparatus comprising:

a computer system including a processor, user input device and display screen;

a CD-ROM drive coupled to the computer system for retrieving frames from the series of frames on the CD-ROM;

selection means responsive to signals from the user input device to output a select signal indicating the selection of the second sequence; and

displaying means coupled to the selection means for displaying on the display screen only frames corresponding to the first animation sequence, and, upon generation of the select signal, for displaying on the display screen only frames corresponding to the second animation sequence.

8 ~~6~~. The apparatus of claim 7 ~~5~~, wherein the frames in the series are compressed, the apparatus further comprising:

selection means responsive to signals from the user input device to output a select signal indicating the selection of the second sequence; and

decompression means for selectively decompressing frames in response to the select signal, wherein only frames corresponding to the first animation sequence are decompressed unless the select signal is present, in which case only frames corresponding to the second animation sequence are decompressed.

~~7~~. An apparatus for creating a CD-ROM disc with interleaved animation sequences, the apparatus coupled to a CD-ROM recording device for writing information to a CD-ROM disc, the apparatus including storage means, the storage means including first and second animation sequences each including a plurality of frames, the apparatus comprising:

first frame selection means for selecting frames from the first animation sequence in a predetermined order;

second frame selection means for selecting frames from a second animation sequence in a predetermined order;

combining means coupled to the first and second frame selection means for combining the selected frames in an interleaved manner; and

control means coupled to the CD-ROM recording device and coupled to the combining means, wherein the control means writes the selected frames combined in an interleaved manner to the CD-ROM disc.

08/252460

see Exr AMDT #15/D for new title

~~SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED INTERACTIVE PRODUCTIONS~~

ABSTRACT

A system for eliminating access time in CD-ROM based interactive video applications. A CD-ROM disc is formatted with multiple interleaved animation sequences. During playback, a user is able to select a sequence as the current sequence. Only those frames corresponding to the current sequence are displayed while other frames are dropped. The interleaved pattern of frames allows multiple sequences to be available from the CD-ROM drive without requiring repositioning of the CD-ROM drive's read head thereby eliminating access time. Specific patterns of interleaving that advantageously improve interactivity of an interactive production are described. A frame buffering implementation is described.

Attorney Docket No. CJK-1

DECLARATION

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED INTERACTIVE PRODUCTIONS** the specification of which X is attached hereto as Application Serial No. ______ and was amended on ___ (if applicable).

I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56. I claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign applications(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Country	Application No.	Date of Filing	Priority Claimed Under 35 USC 119
			Yes ____ No ____
			Yes ____ No ____

I claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, section 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial No.	Date of Filing	Status
		___ Patented ___ Pending ___ Abandoned
		___ Patented ___ Pending ___ Abandoned

1-00

Full Name of Inventor 1	Last Name **Kulas**	First Name **Charles**	Middle Name or Initial **J.**	
Residence & Citizenship	City **San Francisco**	State/Foreign Country **California**	Country of Citizenship **USA**	
Post Office Address	Post Office Address **25 Capra Way #305**	City **San Francisco**	State/Country **California/USA**	Zip Code **94123**
Full Name of Inventor 2	Last Name	First Name	Middle Name or Initial	
Residence & Citizenship	City	State/Foreign Country	Country of Citizenship	
Post Office Address	Post Office Address	City	State/Country	Zip Code
Full Name of Inventor 3	Last Name	First Name	Middle Name or Initial	
Residence & Citizenship	City	State/Foreign Country	Country of Citizenship	
Post Office Address	Post Office Address	City	State/Country	Zip Code

CA

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor 1	Signature of Inventor 2	Signature of Inventor 3
Charles J. Kulas **Charles J. Kulas**		
Date 5-31-94	Date	Date

DEC.MRG 1/93

Atty. Docket No. CJK-1

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) and 1.27(b)) - INDEPENDENT INVENTOR

Applicant or Patentee: Charles J. Kulas
Serial or Patent No.: ______
Filed or Issued: 5-31-94
Title: SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED INTERACTIVE PRODUCTIONS

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees to the Patent and Trademark Office regarding the invention entitled SYSTEM FOR ELIMINATING ACCESS TIME IN CD-ROM BASED INTERACTIVE PRODUCTIONS
described in:

[X] the specification herewith.
[ ] application Serial No. ______, filed ______.
[ ] Patent No. ______, issued ______.

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey or license any rights in the invention is listed below:*

[X] No such person, concern, or organization
[ ] Persons, concerns or organizations listed below*

*NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

NAME ______
ADDRESS ______

[ ] INDIVIDUAL [ ] SMALL BUSINESS CONCERN [ ] NONPROFIT ORGANIZATION

NAME ______
ADDRESS ______

[ ] INDIVIDUAL [ ] SMALL BUSINESS CONCERN [ ] NONPROFIT ORGANIZATION

NAME ______
ADDRESS ______

[ ] INDIVIDUAL [ ] SMALL BUSINESS CONCERN [ ] NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF INVENTOR: Charles J. Kulas	NAME OF INVENTOR:	NAME OF INVENTOR:
Charles J. Kulas Signature of Inventor	Signature of Inventor	Signature of Inventor
Date 5-31-94	Date:	Date:

08/252460

PRIOR ART

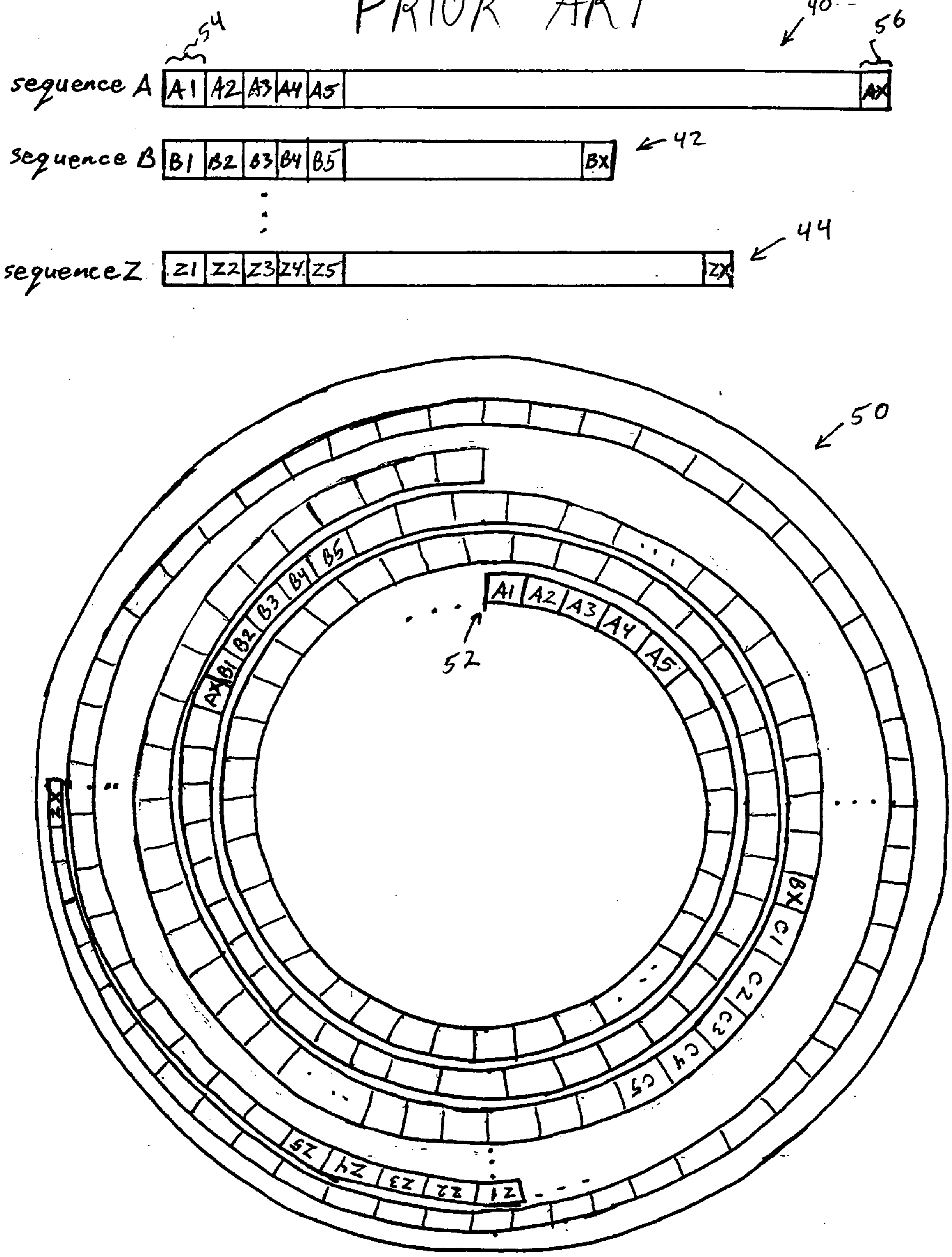


Fig. 1

08/252460

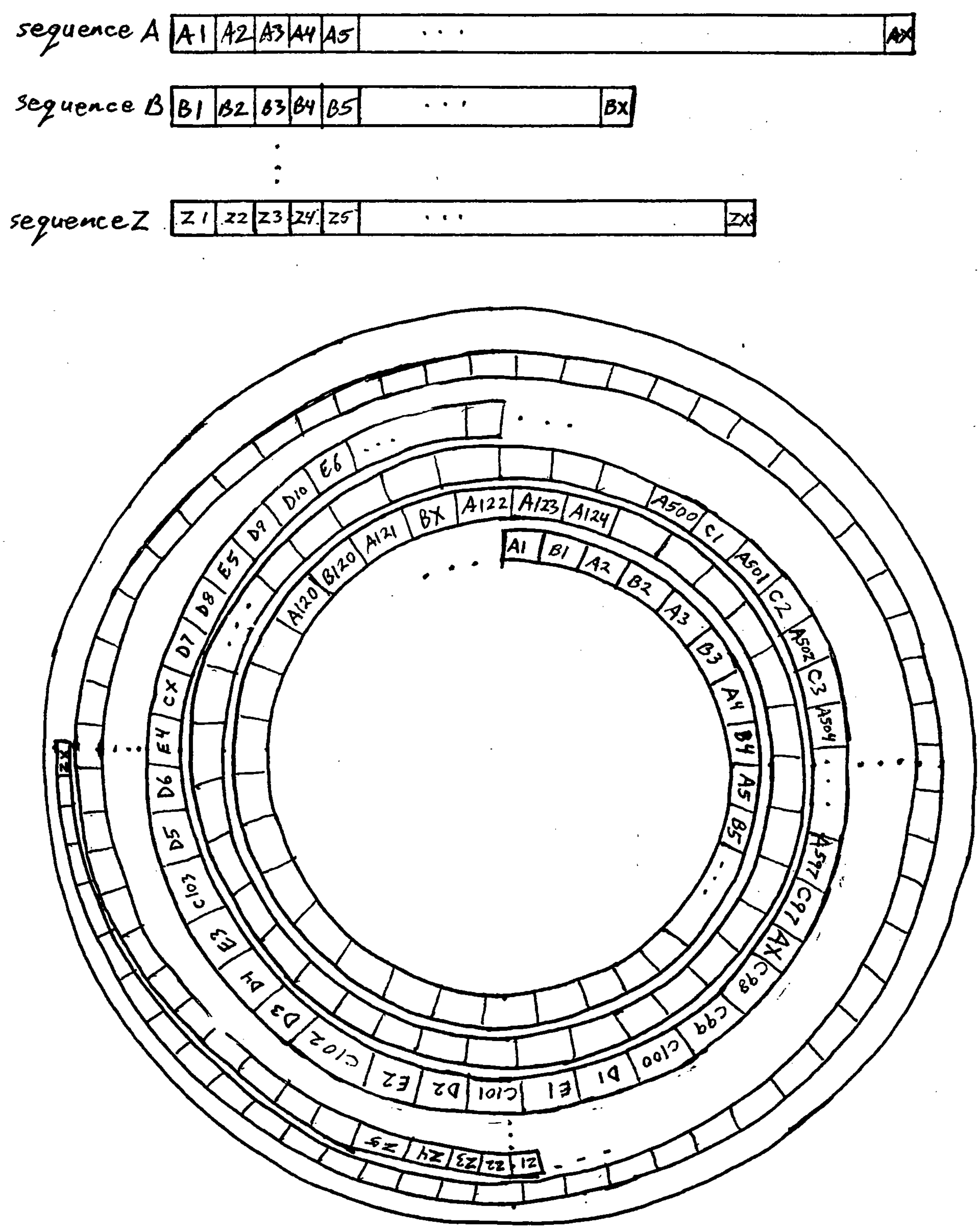


Fig. 2

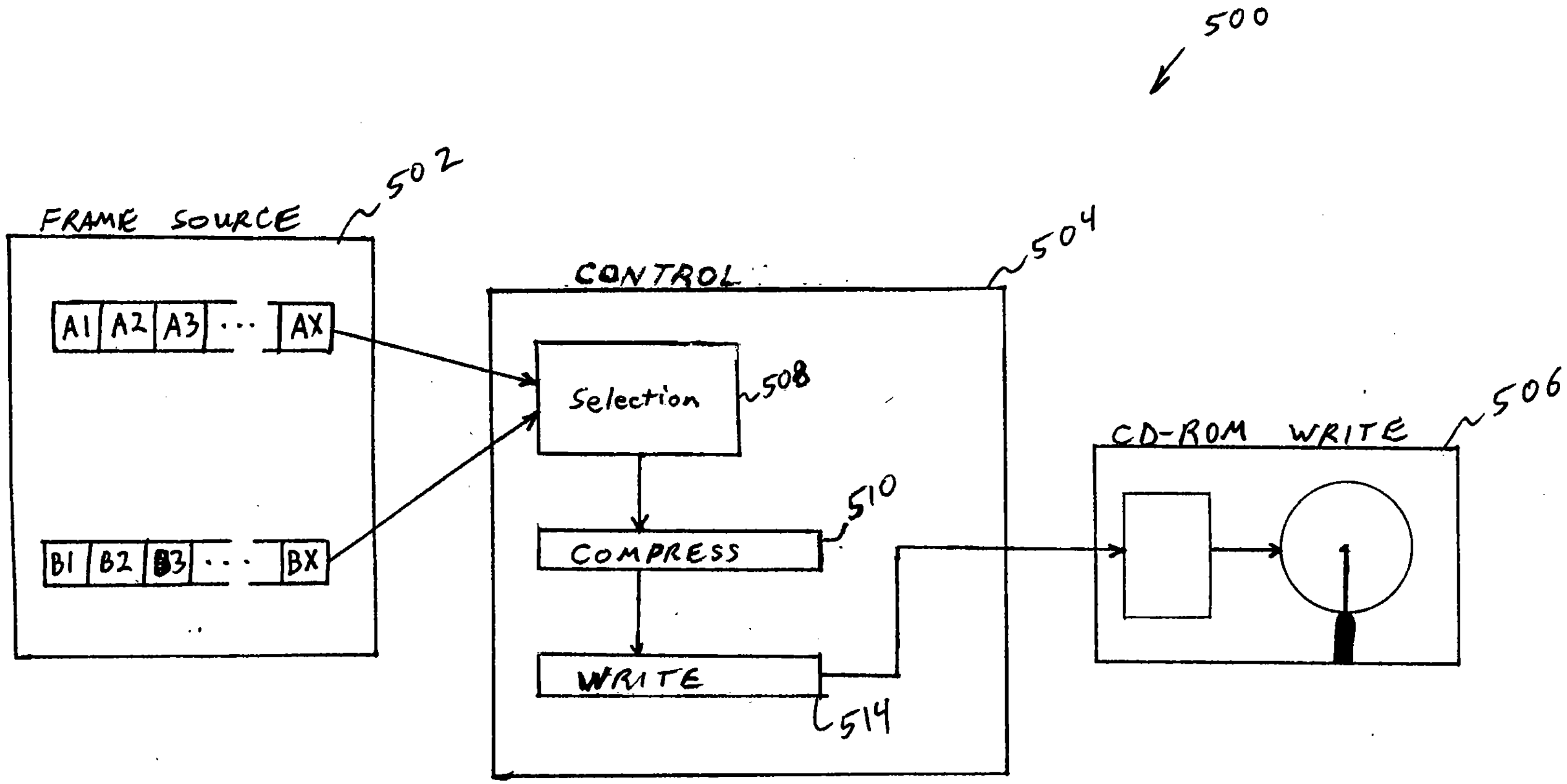


Fig. 2A

08/252460

08/252460

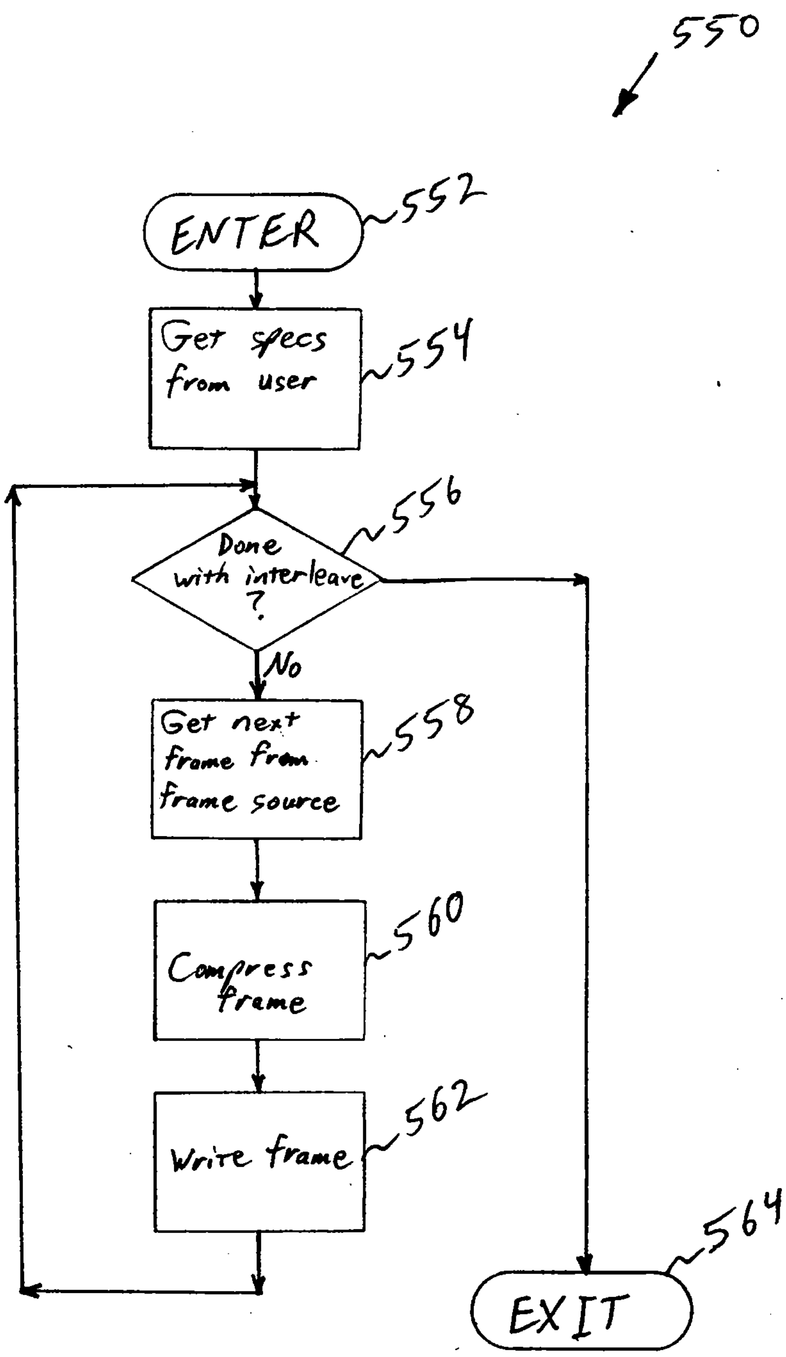


Fig. 2B

08/252460

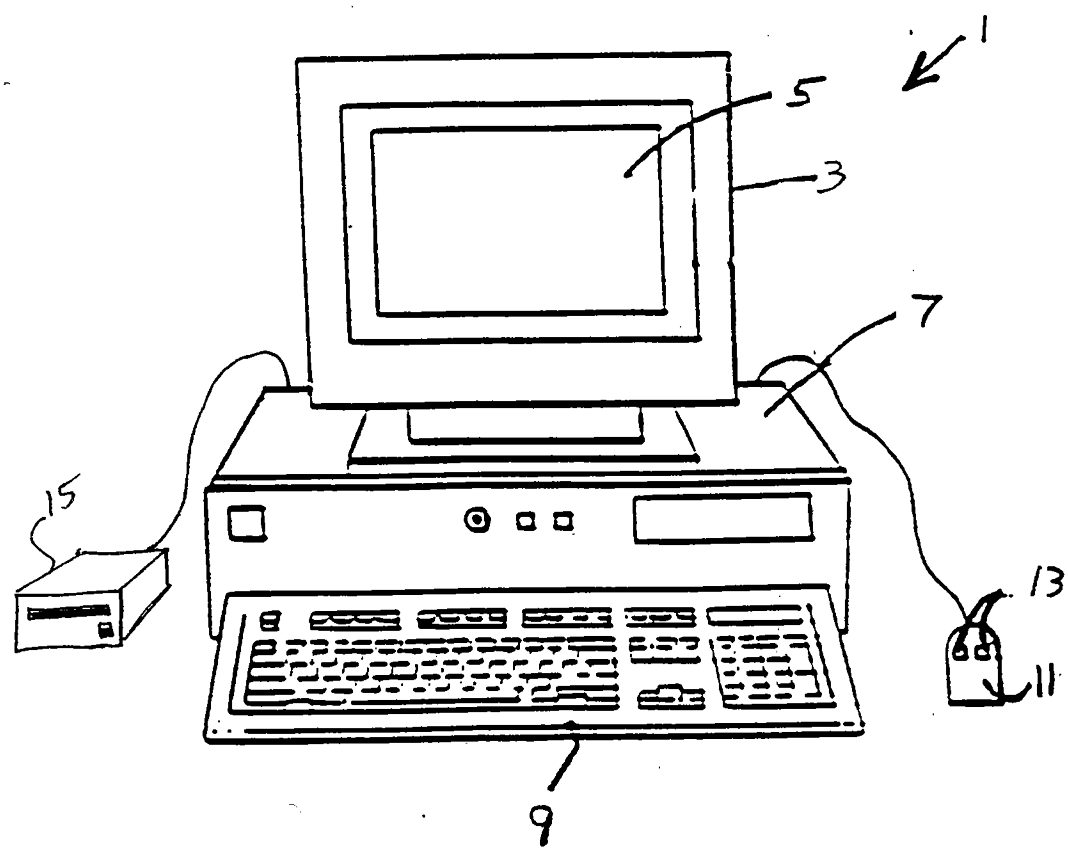


Fig. 3

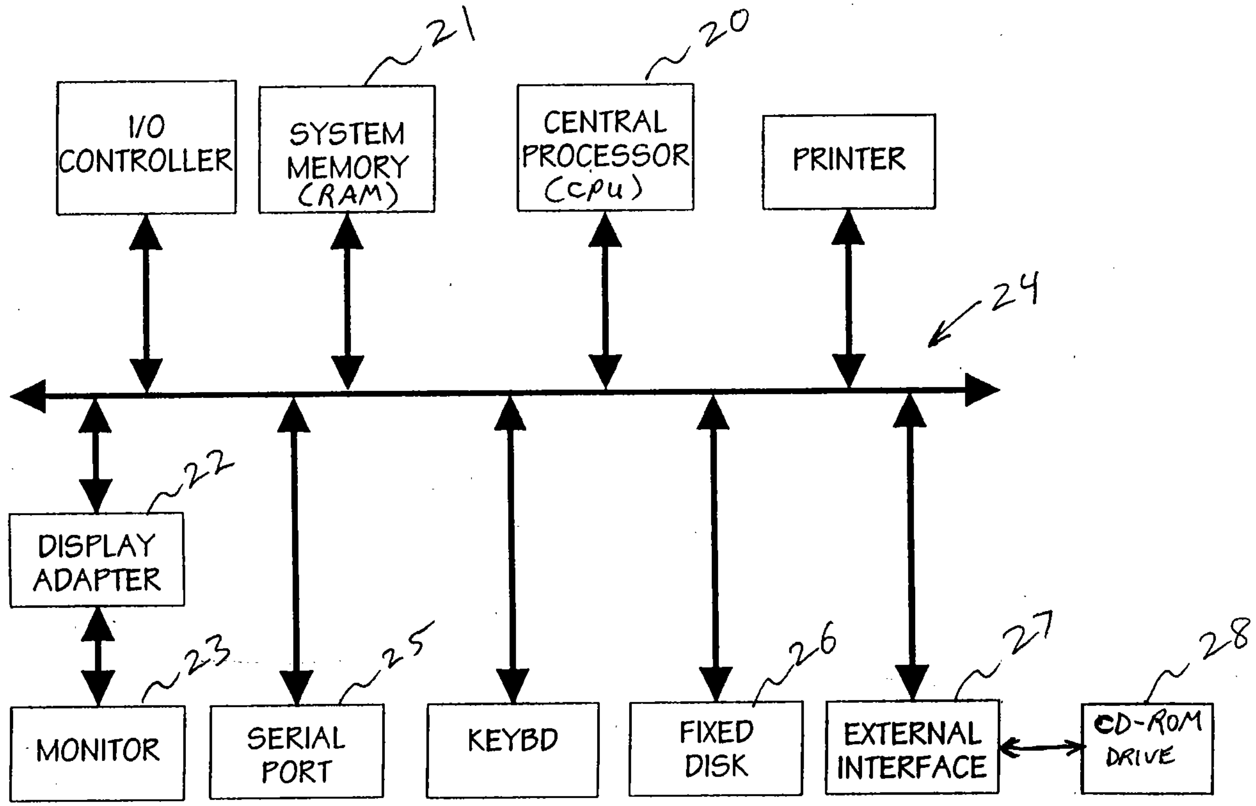


Fig. 4

08/252460

08/252460

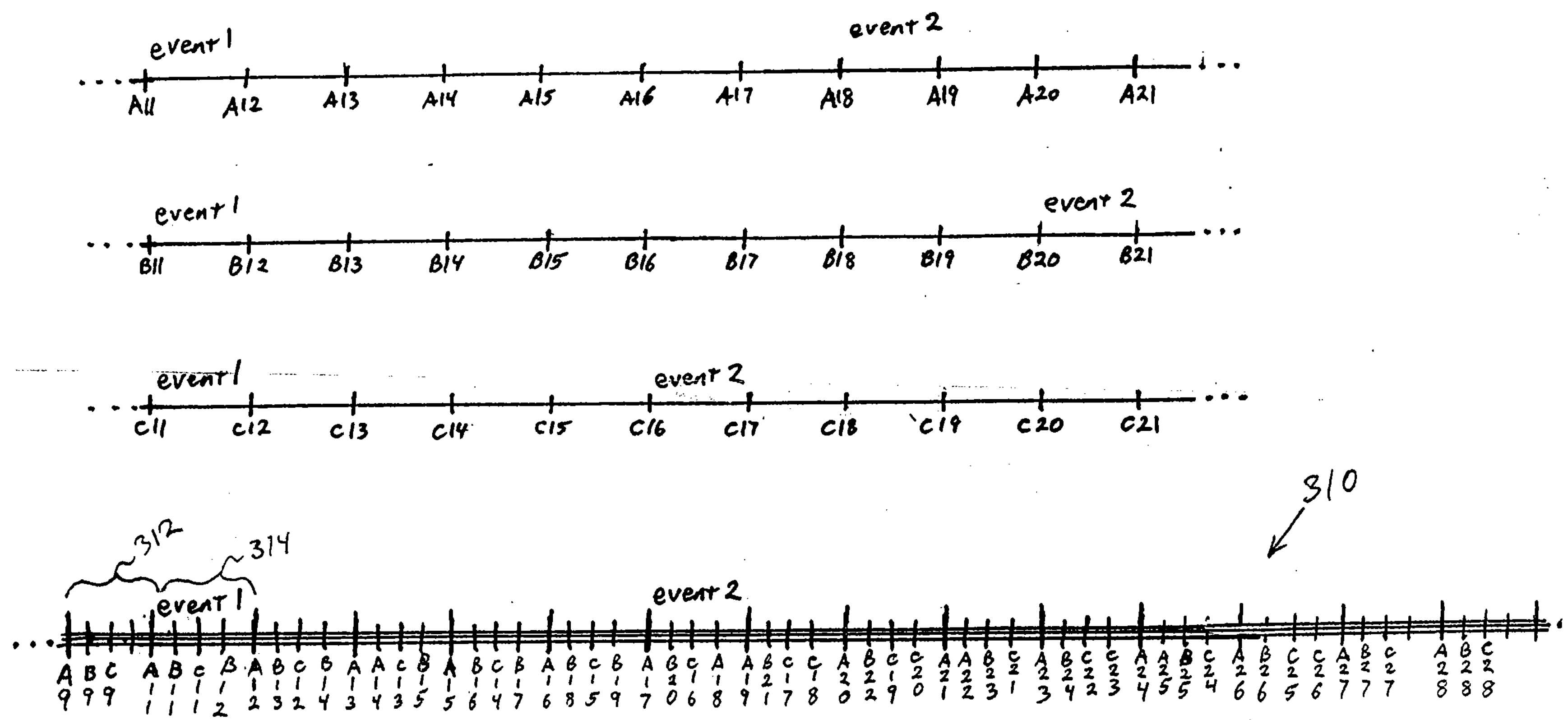


Fig. 8

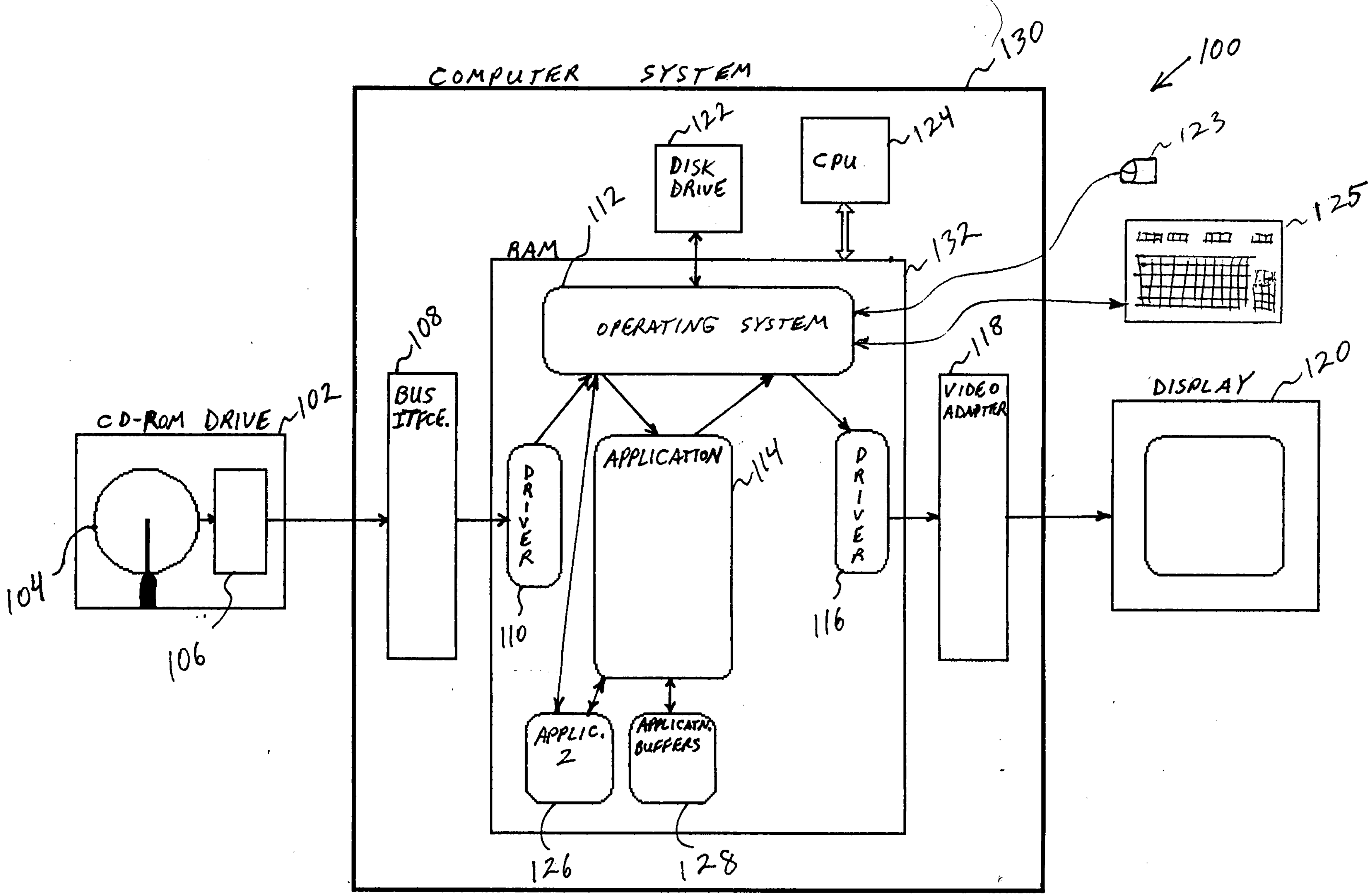


Fig. 5

08/252460

08/252460

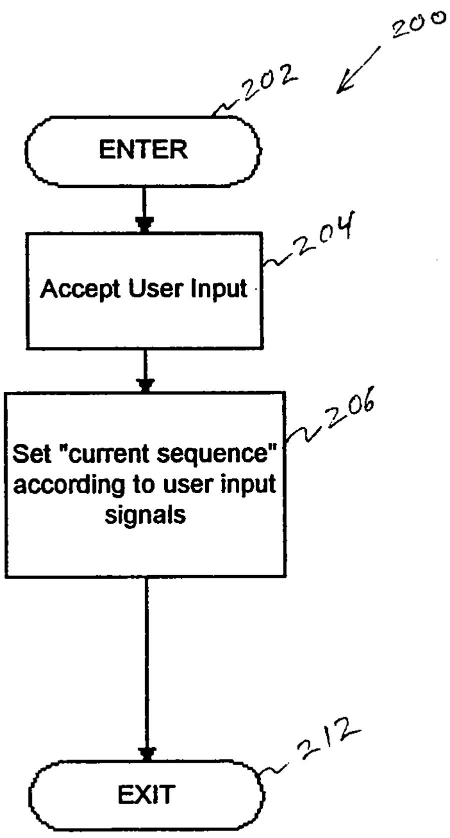


Fig. 6A

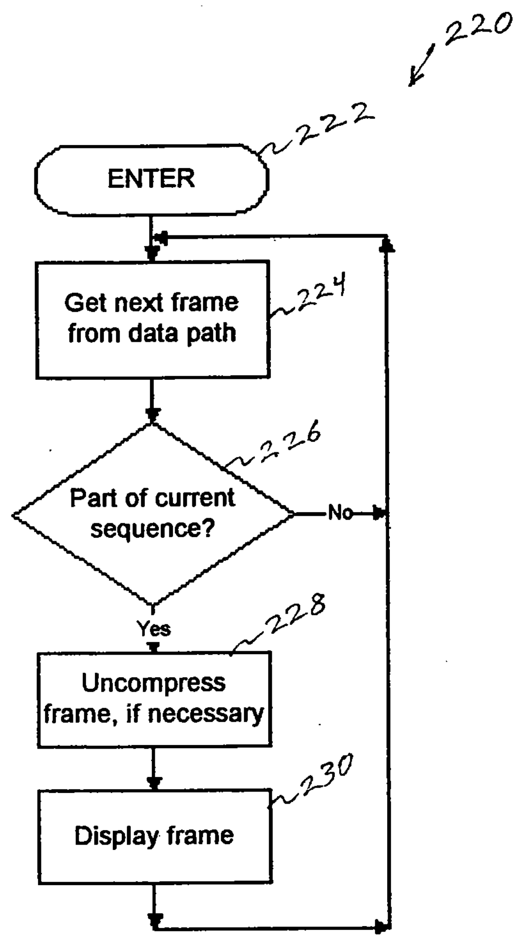


Fig. 6B

08/252460

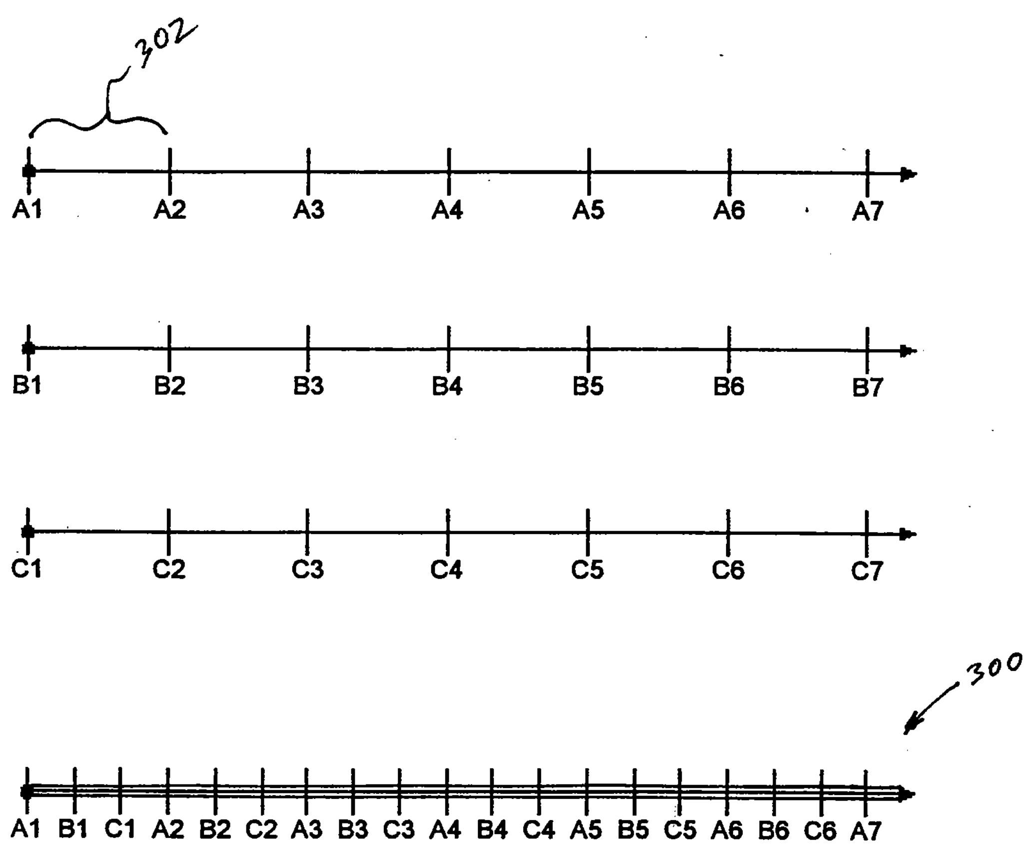


Fig. 7

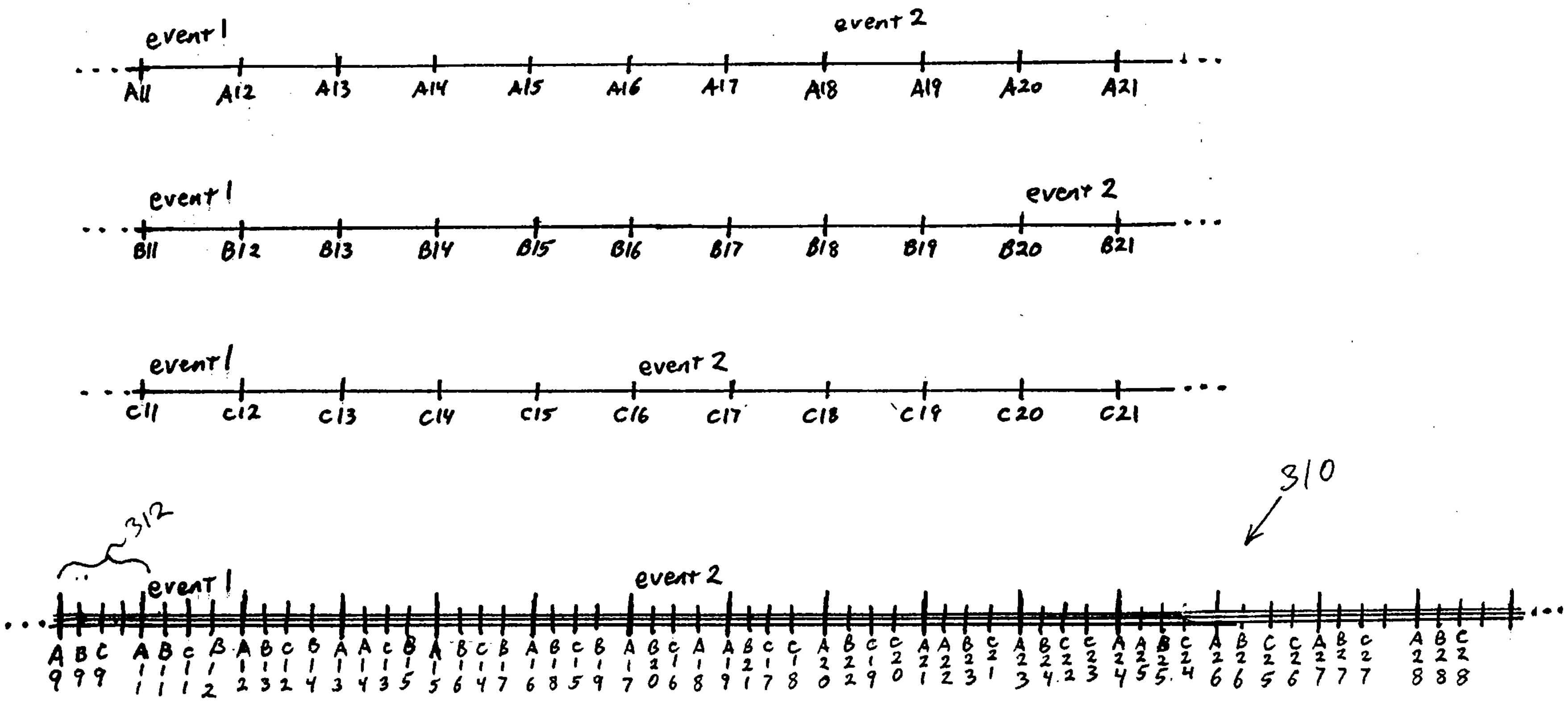
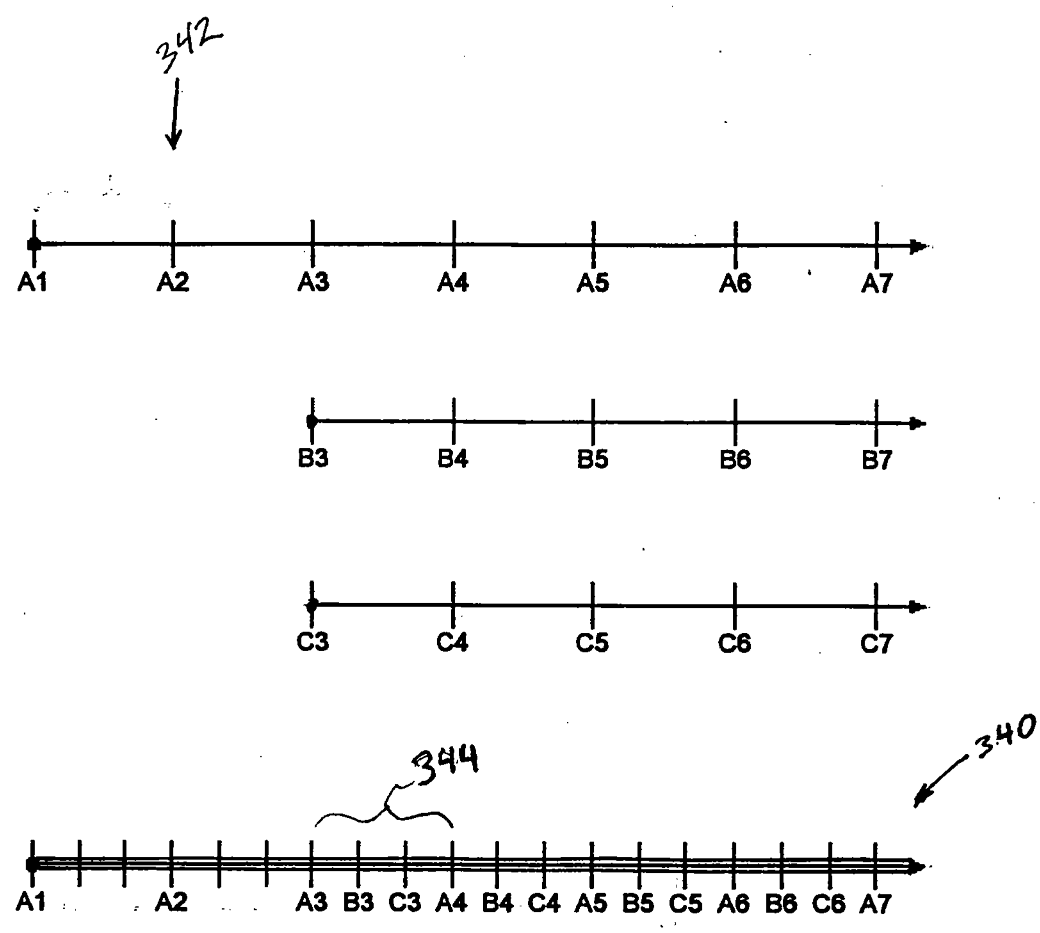


Fig. 8

08/252460



342
A1
A2
A3
A4
A5
A6
A7
B3
B4
B5
B6
B7
C3
C4
C5
C6
C7
344
340
A1
A2
A3
B3
C3
A4
B4
C4
A5
B5
C5
A6
B6
C6
A7

Fig. 9

08/252460

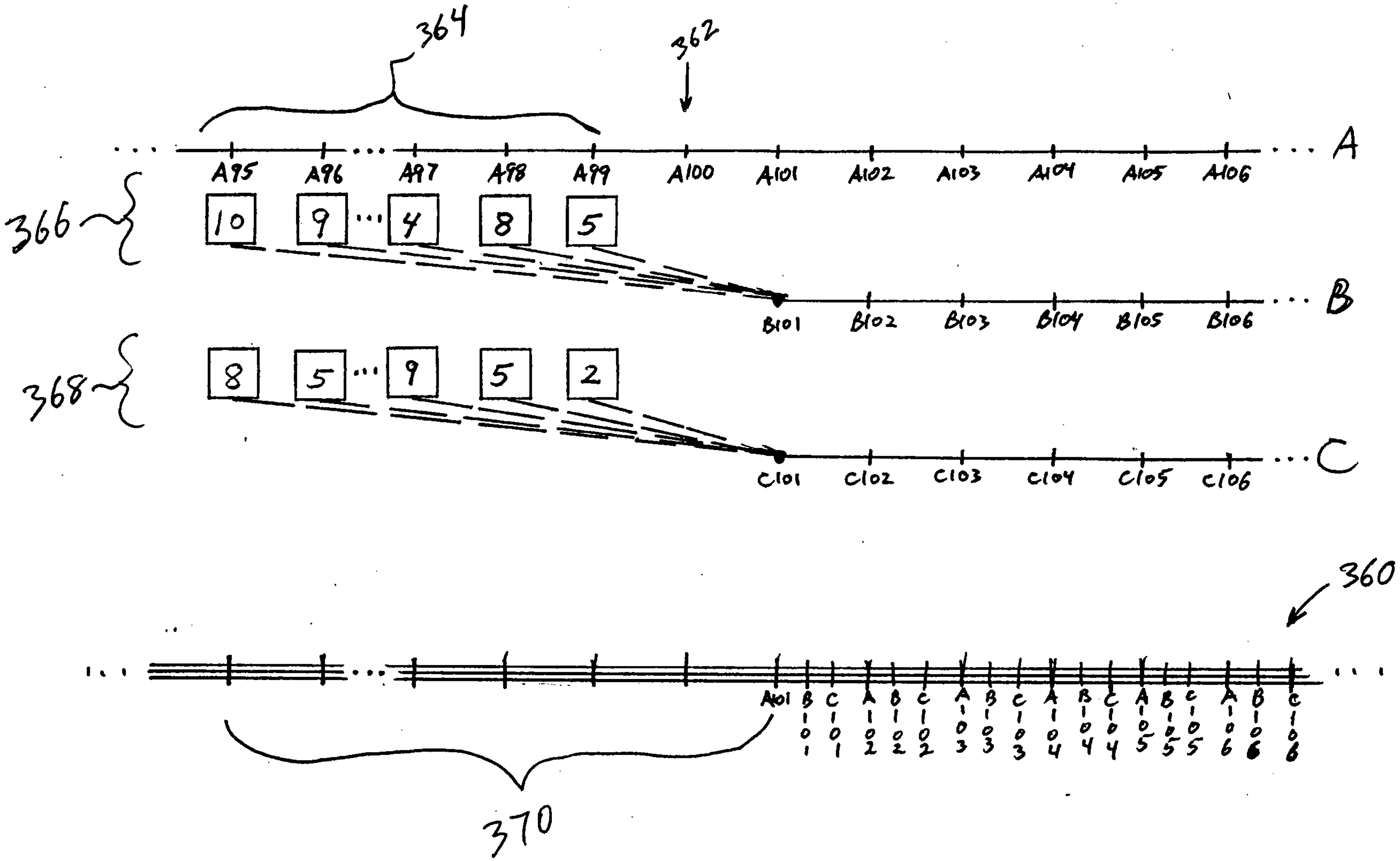


Fig. 10

08/252460

Write Frames to Buffer

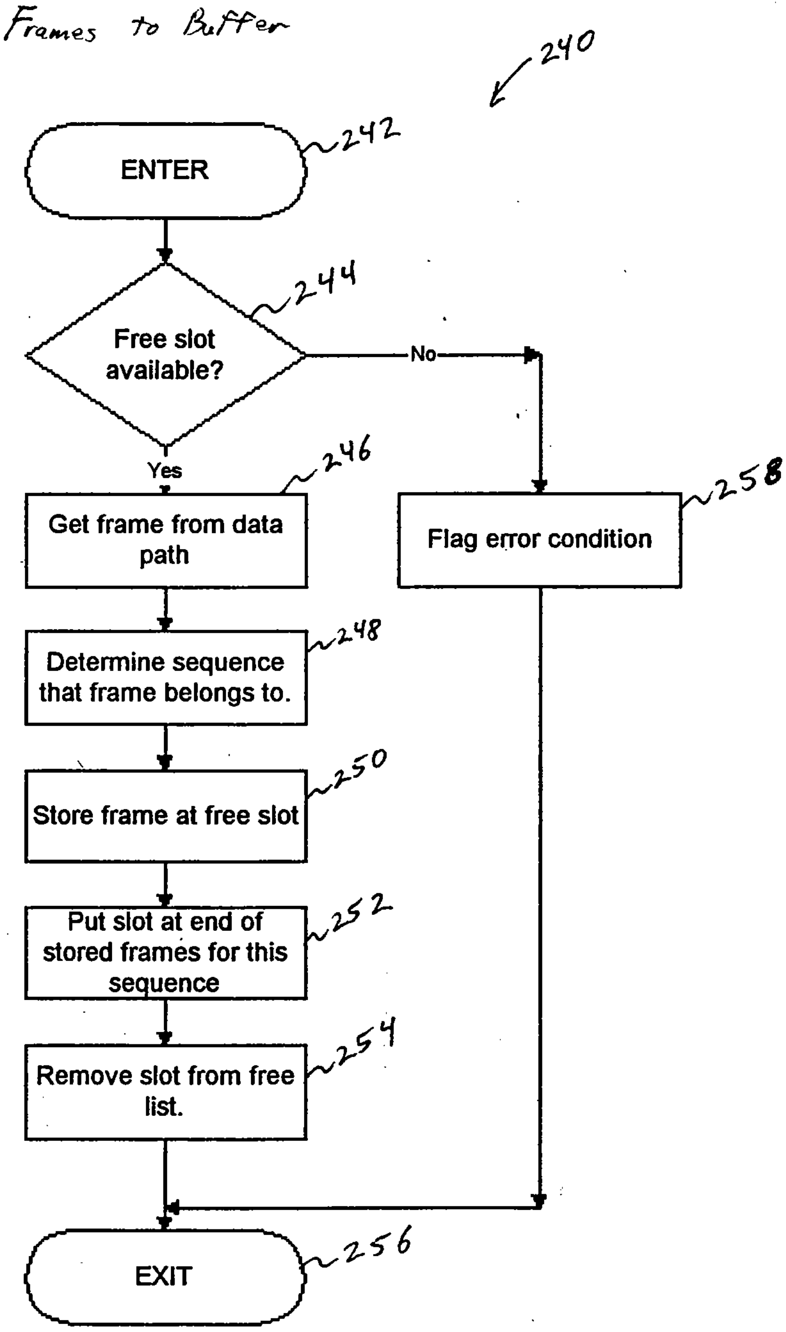


Fig. 11

08/252460

260

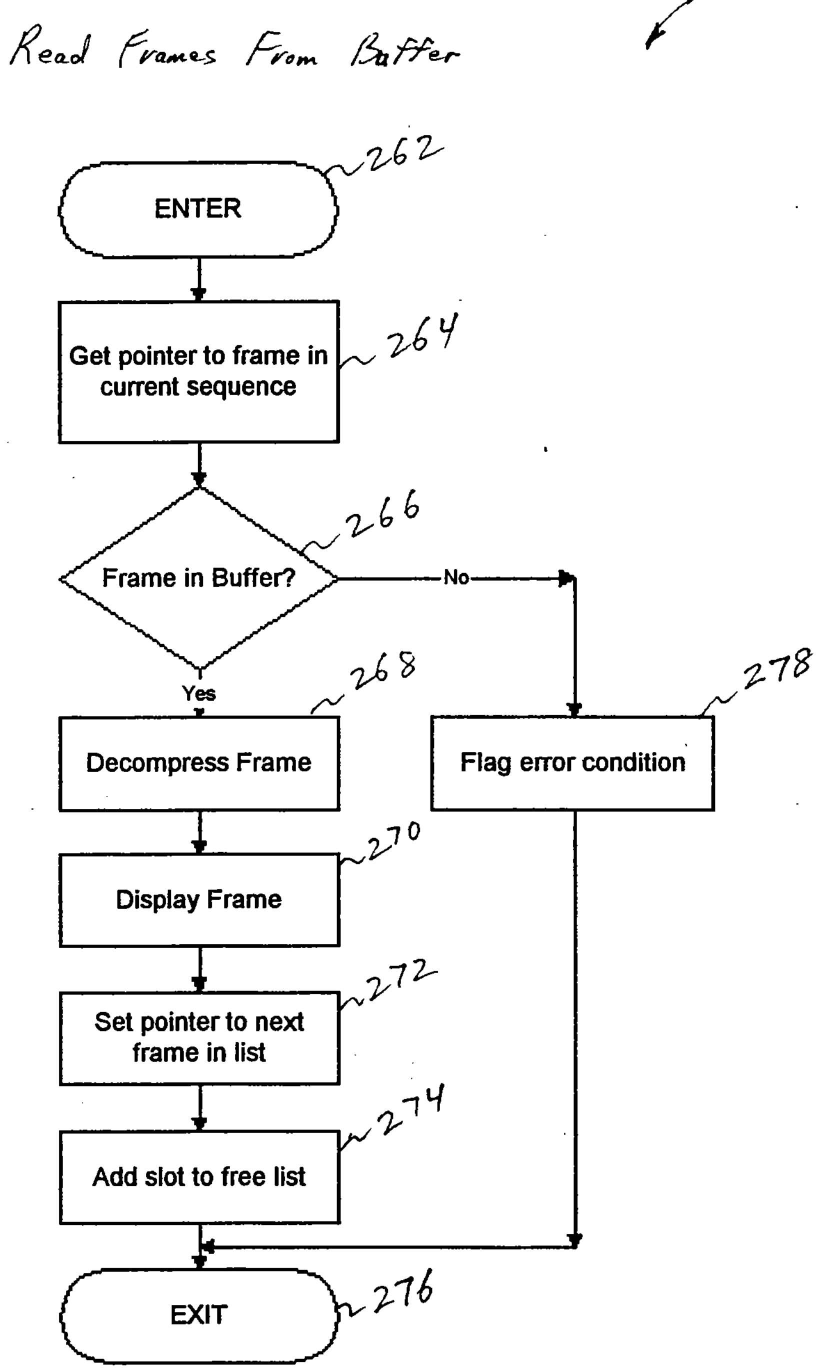


Fig. 12

08/252460

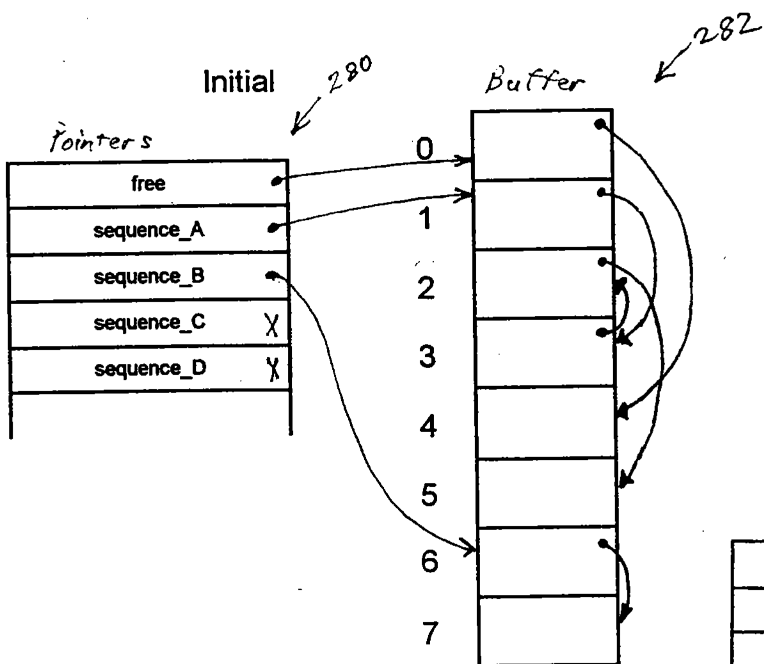


Fig. 13A

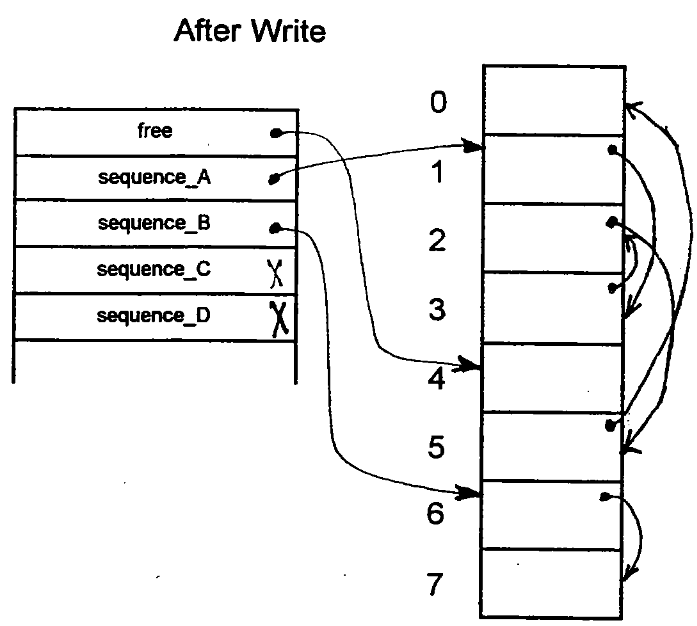


Fig. 13B

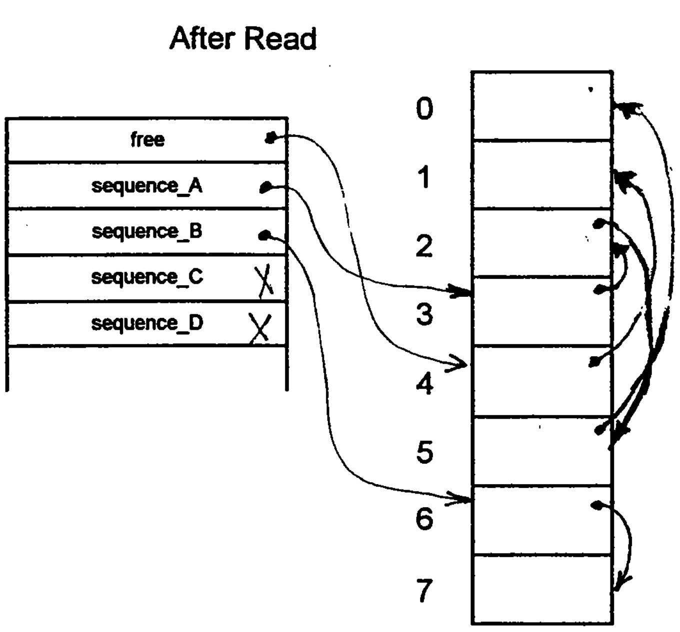


Fig. 13C

08/252460

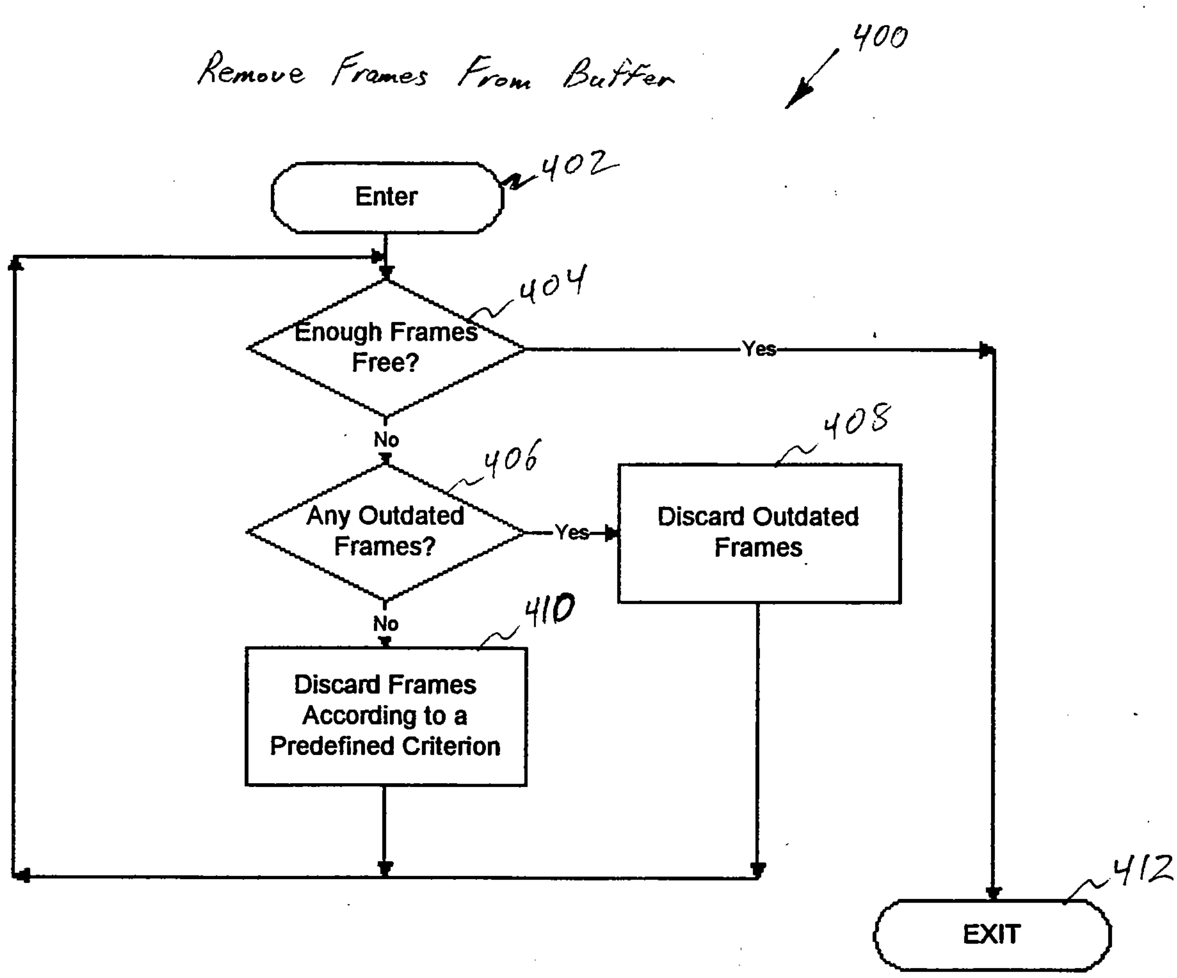


Fig. 14

PRINT OF DRAWINGS
AS ORIGINA FILED

395/275

08/252460

PRIOR ART

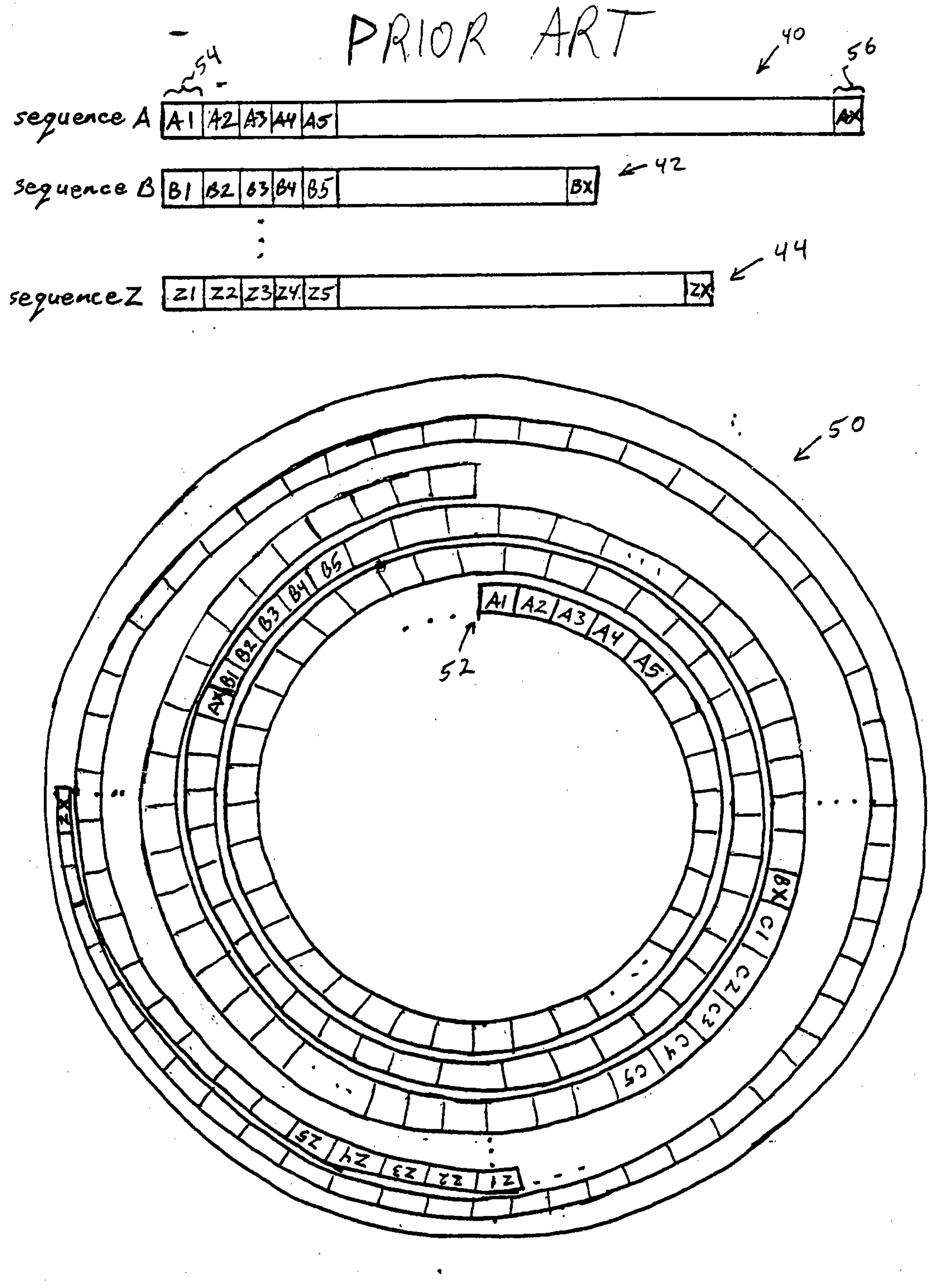


Fig. 1

PRINT OF DRAWINGS
AS ORIGINA FILED

08/252460

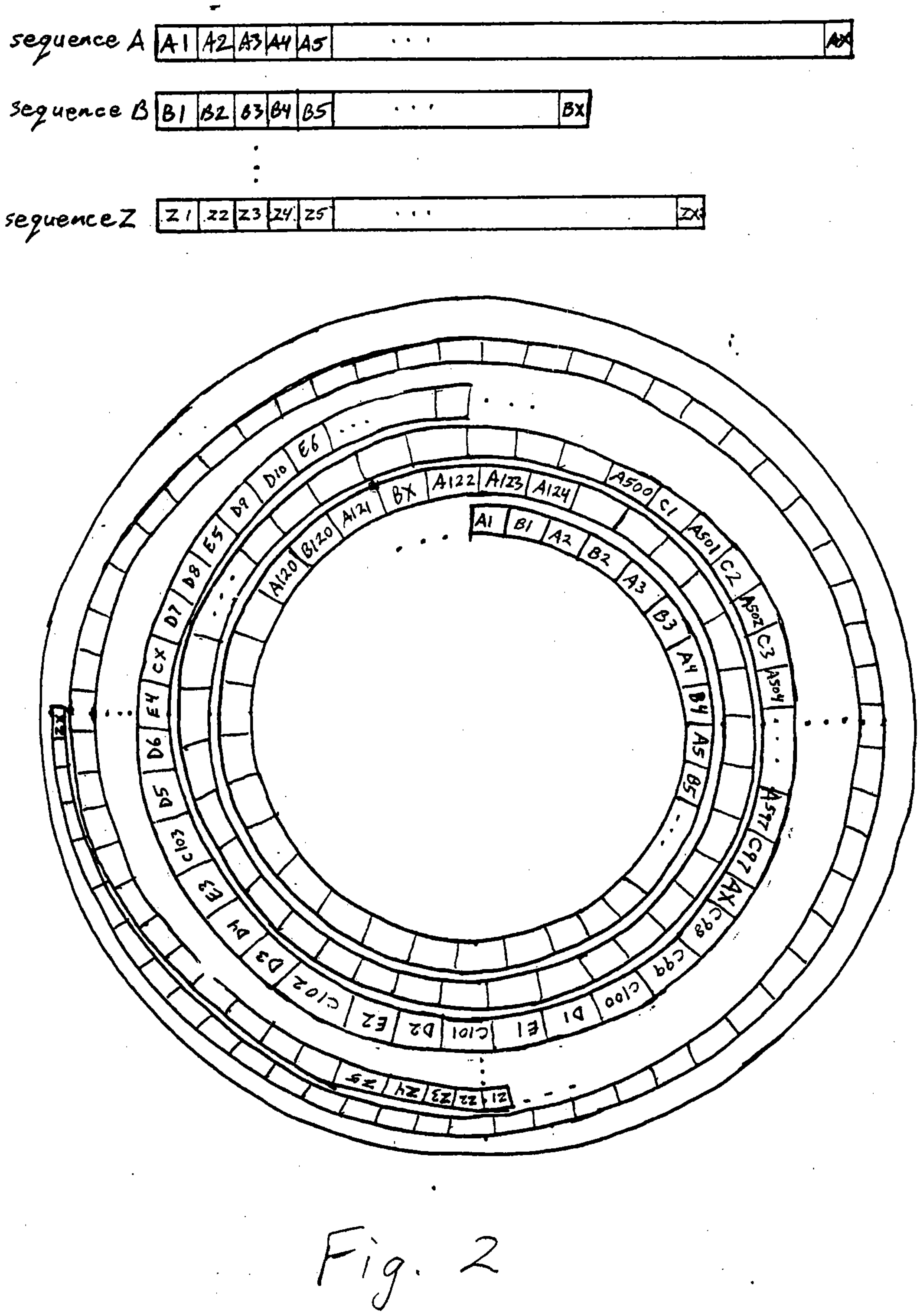


Fig. 2

PRINT OF DRAWINGS
AS ORIGINALLY FILED

08/252460

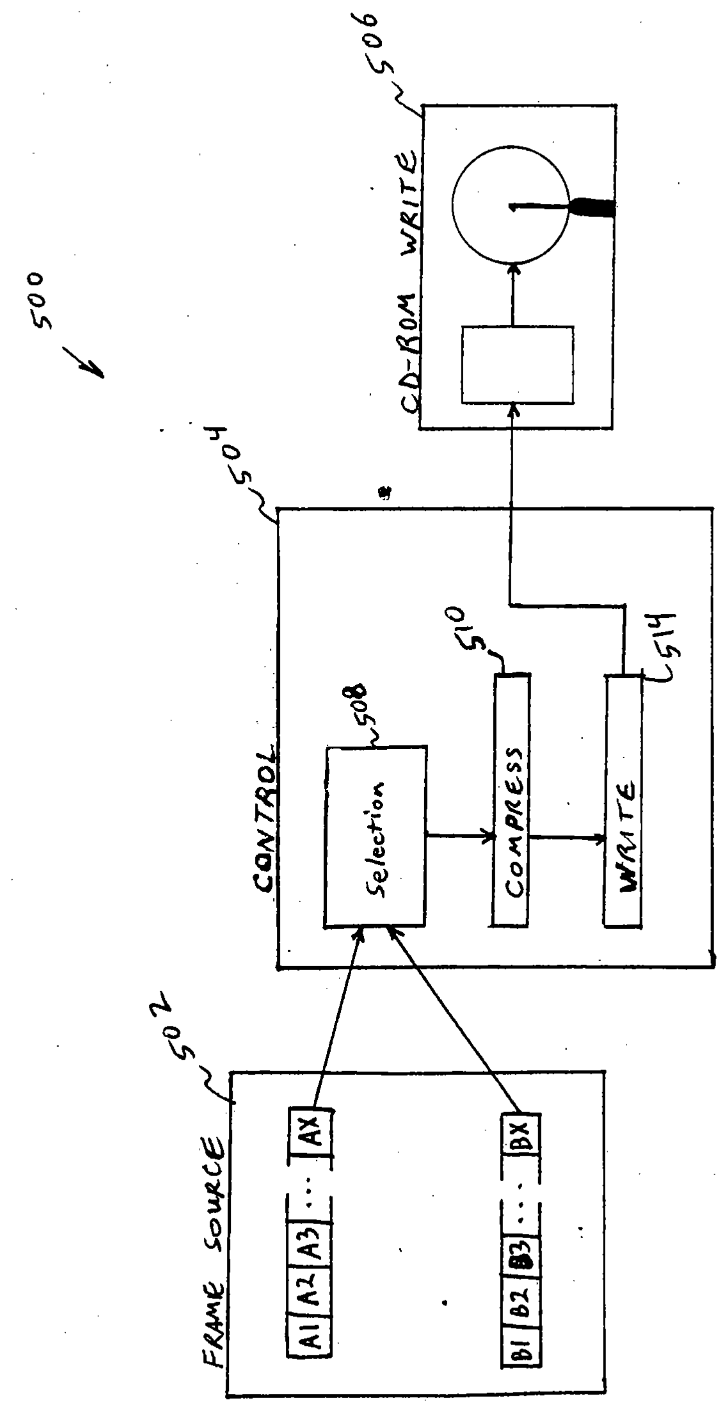


Fig. 2A

PRINT OF DRAWINGS
AS ORIGIN... FILED

08/252460

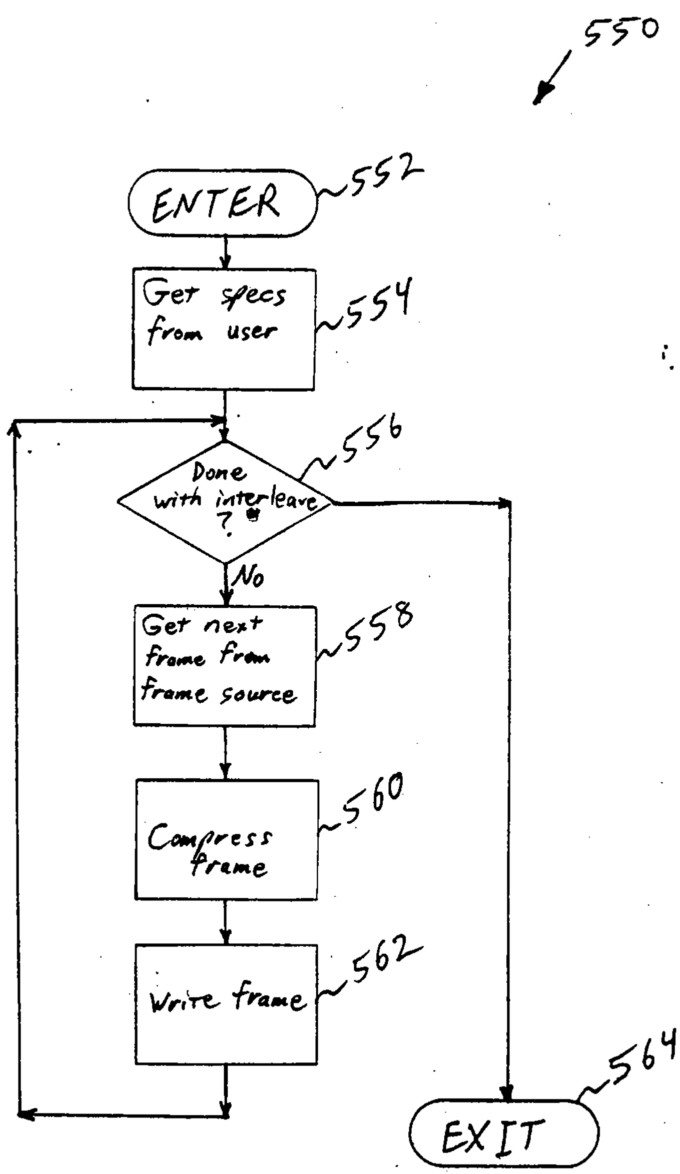


Fig. 2B

PRINT OF DRAWINGS
AS ORIGINAL FILED

08/252460

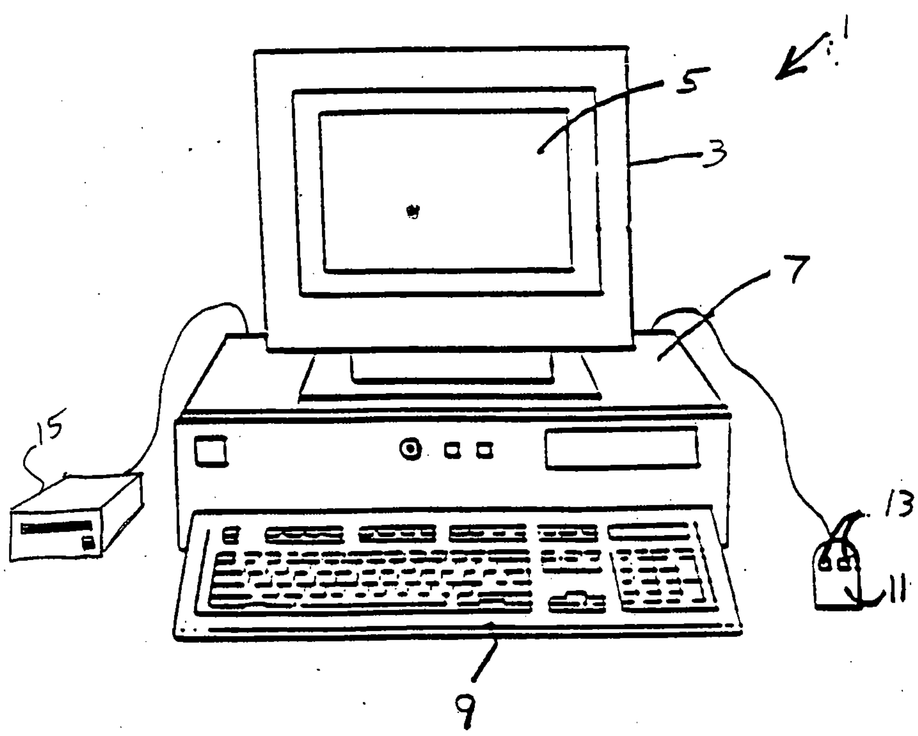


Fig. 3

PRINT OF DRAWINGS
AS ORIGIN/ / FILED

08/252460

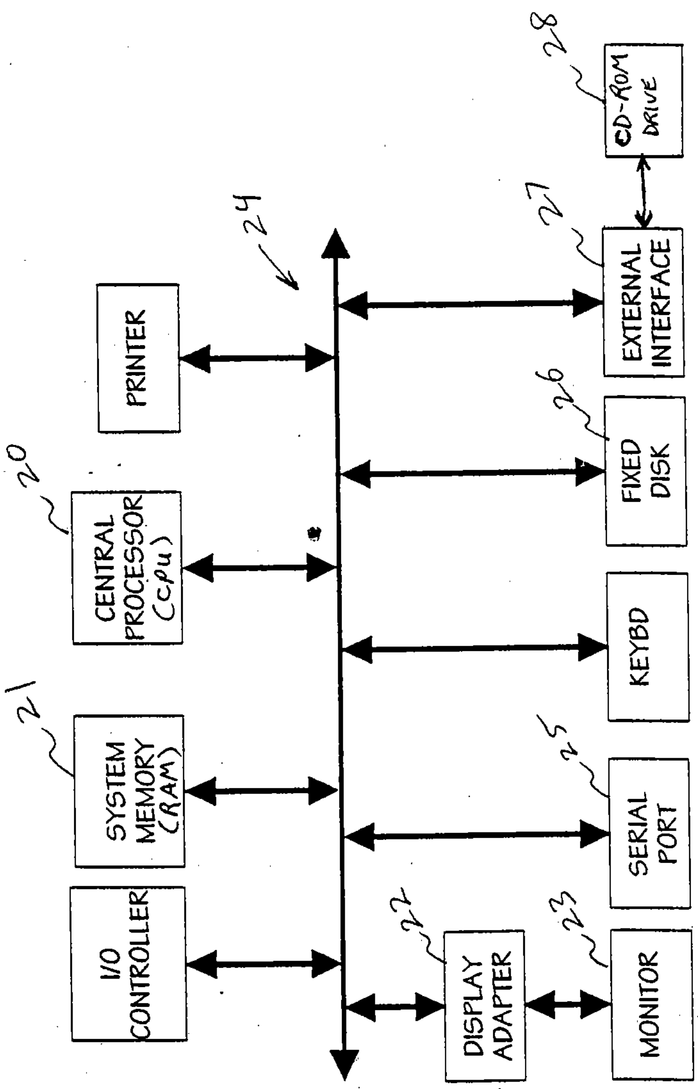


Fig. 4

PRINT OF DRAWINGS
AS ORIGINAL FILED

08/252460

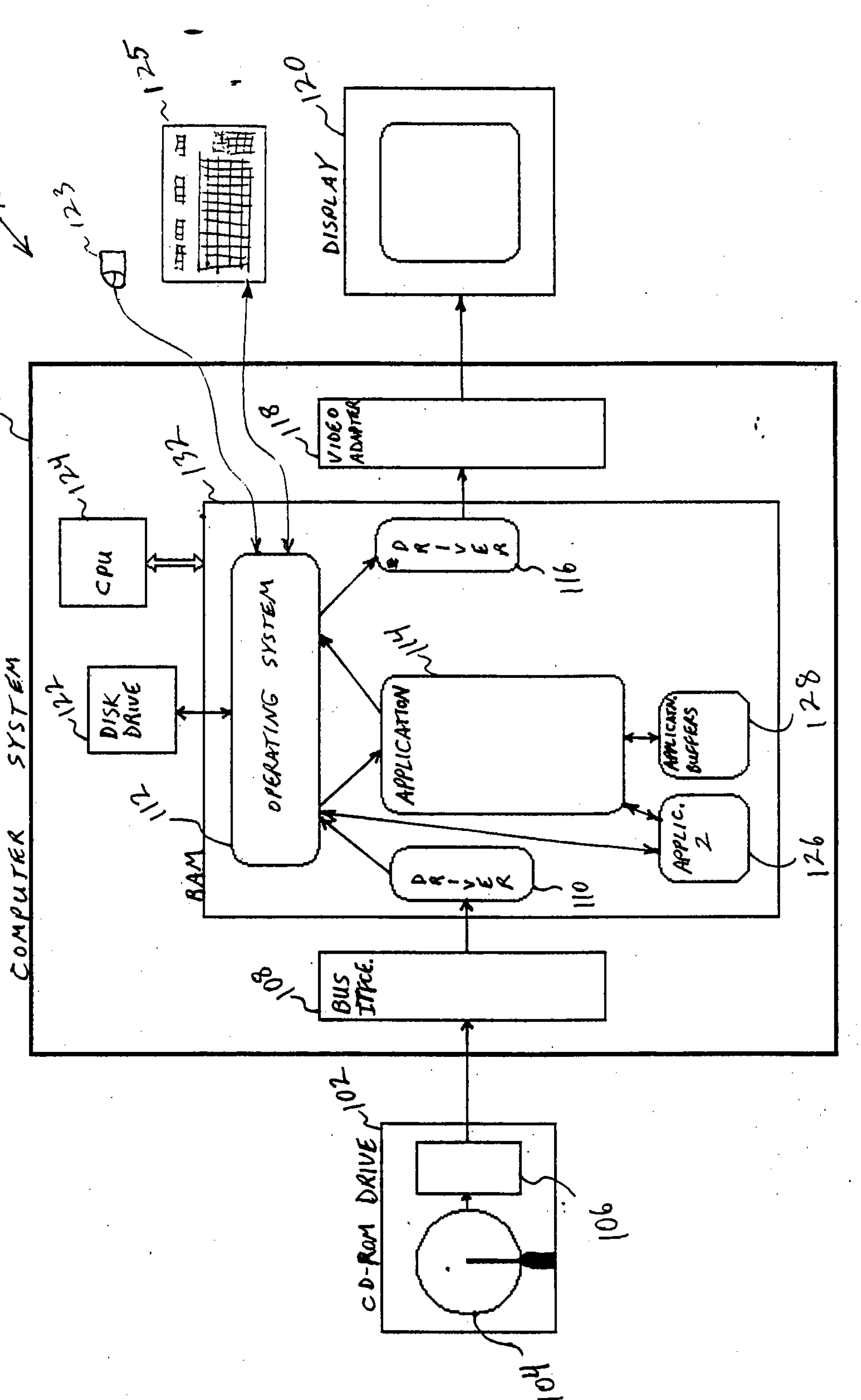


Fig. 5

PRINT OF DRAWINGS
AS ORIGIN[illegible] [illegible] FILED

08/252460

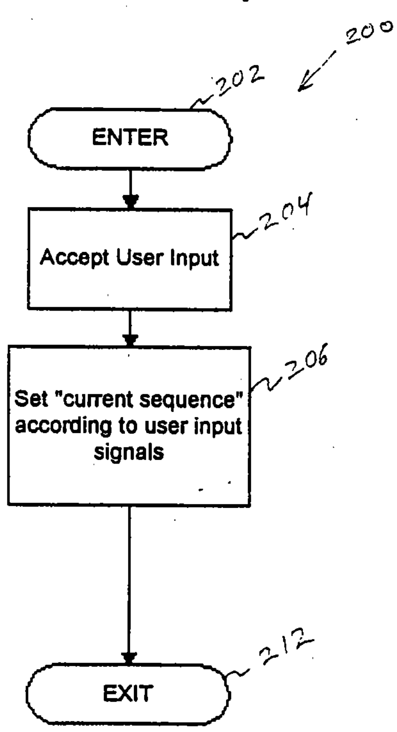


Fig. 6A

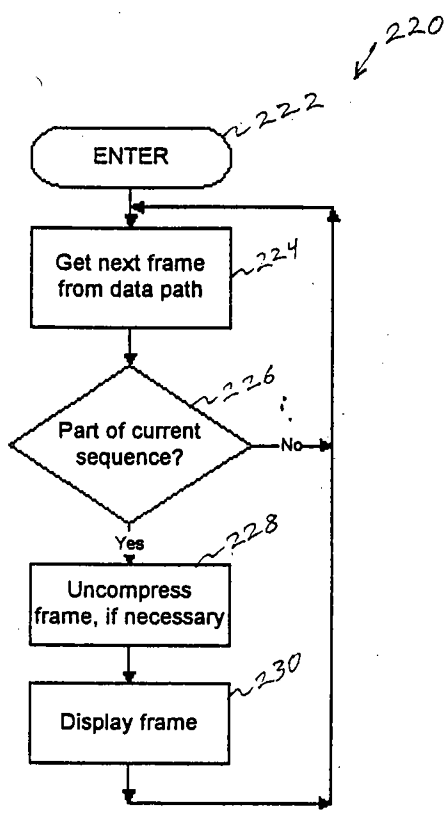


Fig. 6B

PRINT OF DRAWINGS
AS ORIGINALLY FILED

08/252460

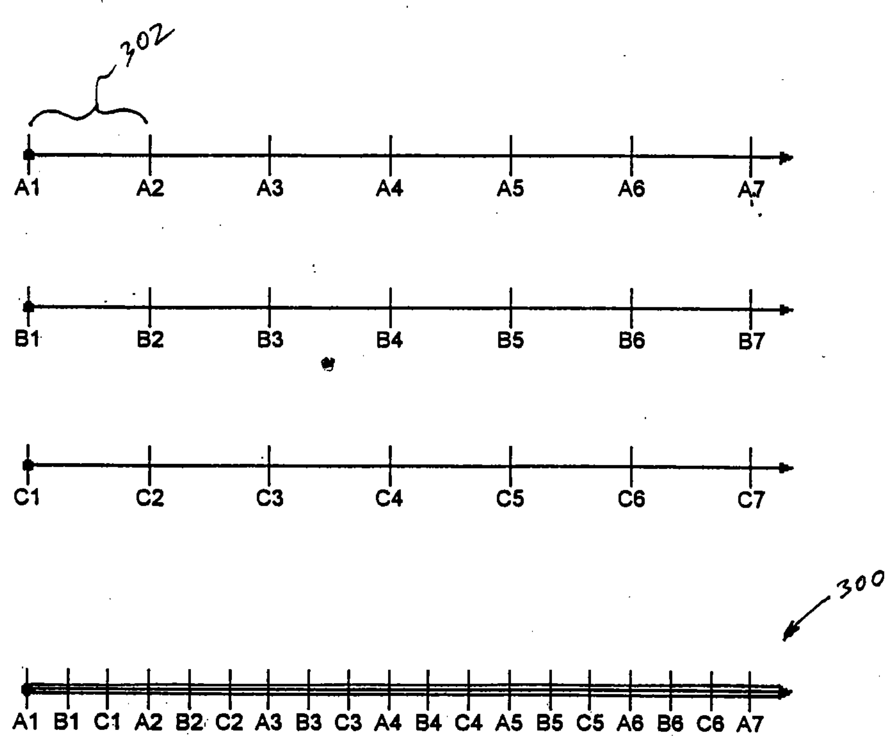


Fig. 7

PRINT OF DRAWINGS
AS ORIGINA FILED

08/252460

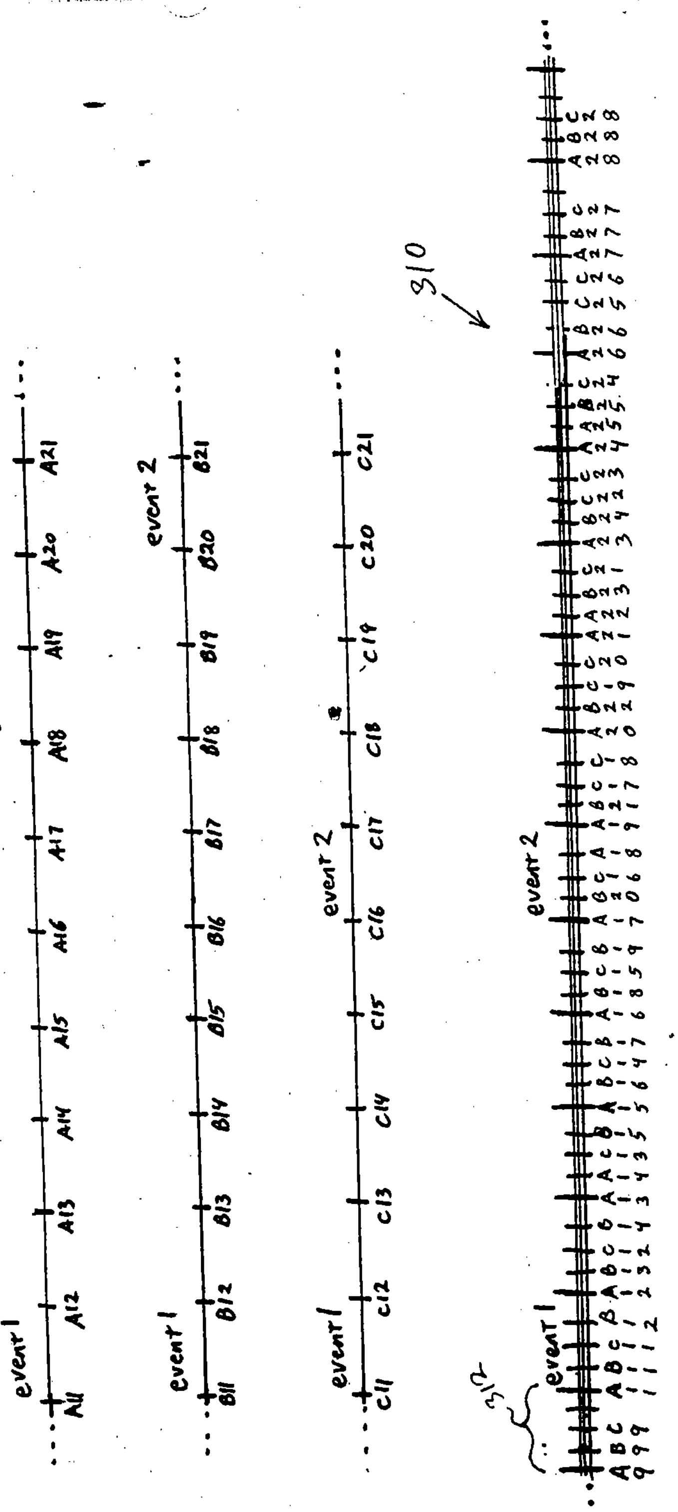


Fig. 8

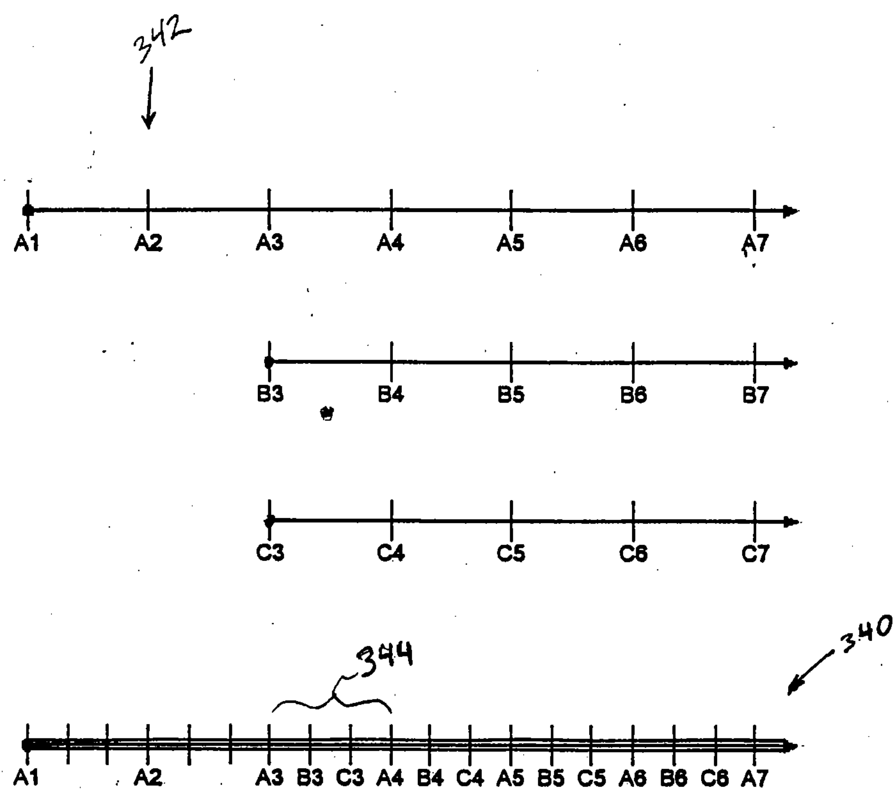

342
A1
A2
A3
A4
A5
A6
A7
B3
B4
B5
B6
B7
C3
C4
C5
C6
C7
344
340
A1
A2
A3 B3 C3 A4 B4 C4 A5 B5 C5 A6 B6 C6 A7

Fig. 9

PRINT OF DRAWINGS
AS ORIGINALLY FILED

08/252460

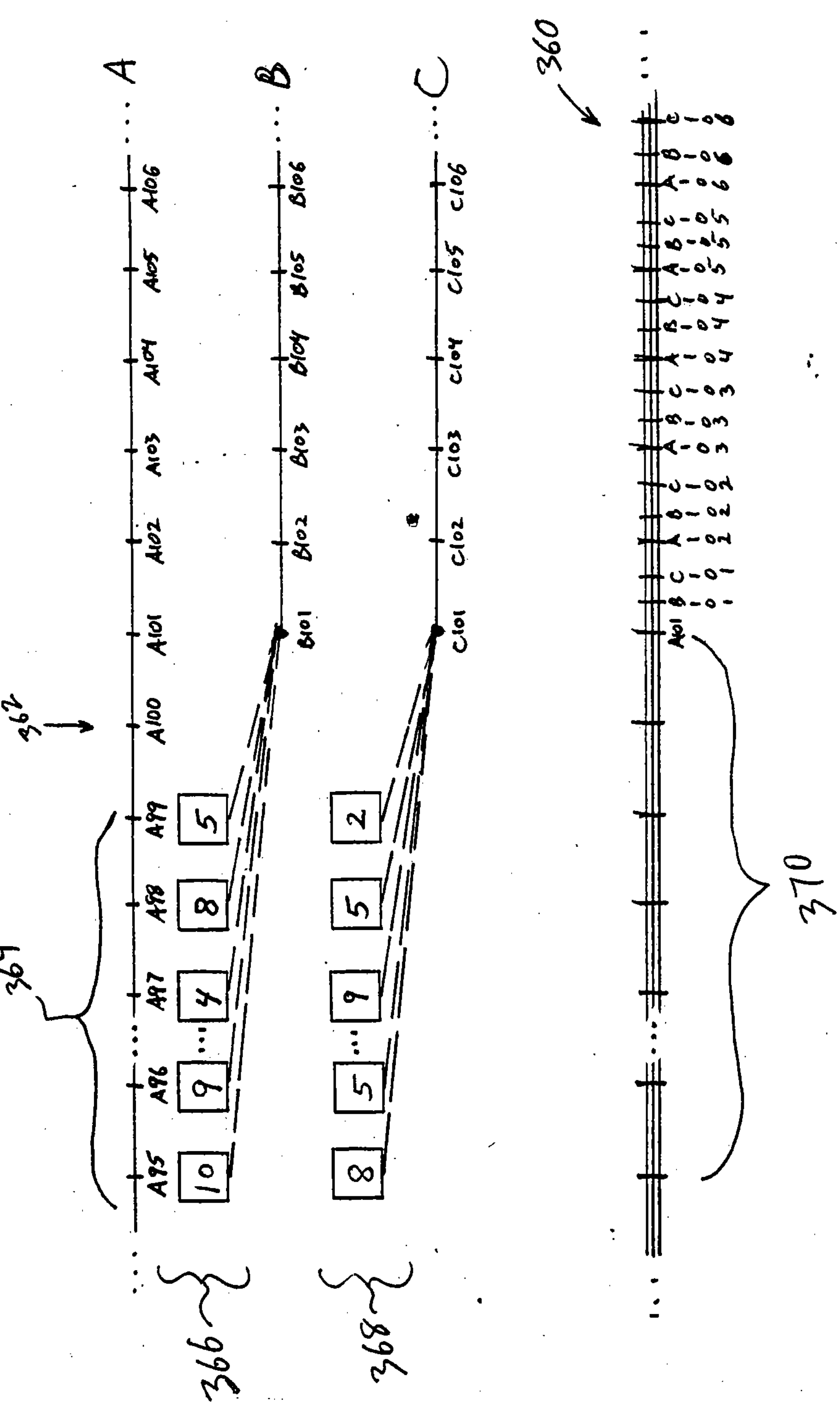


Fig. 10

PRINT OF DRAWINGS
AS ORIGINA FILED

08/252460

Write Frames to Buffer

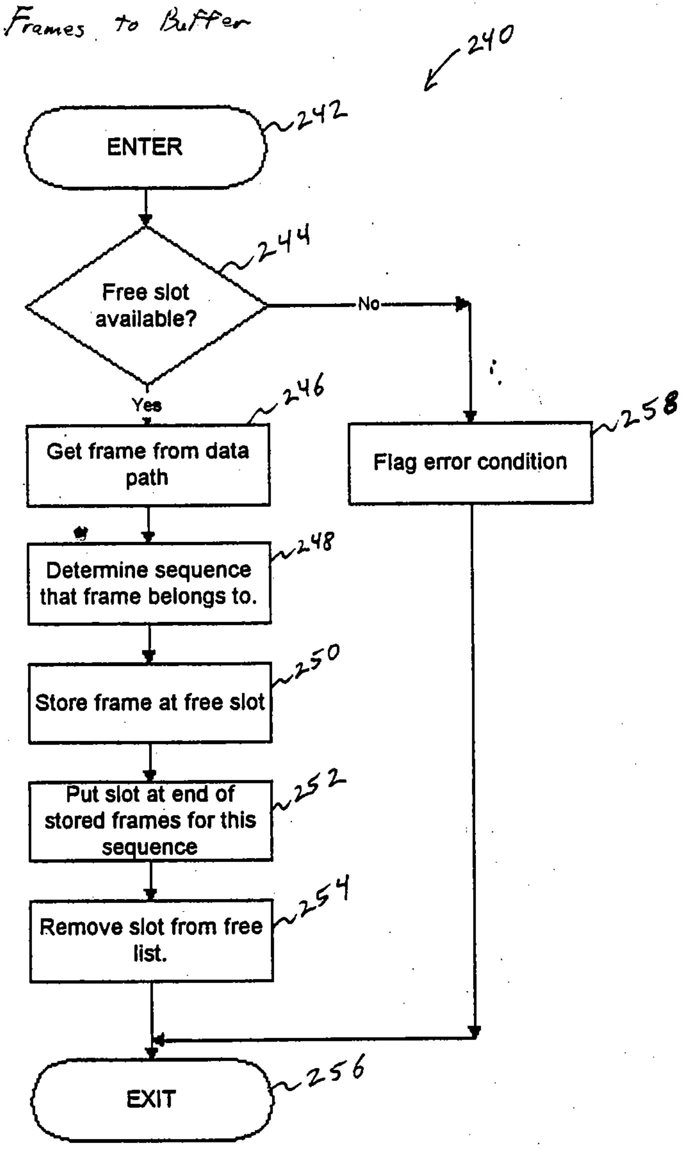


Fig. 11

PRINT OF DRAWINGS
AS ORIGINA FILED

08/252460

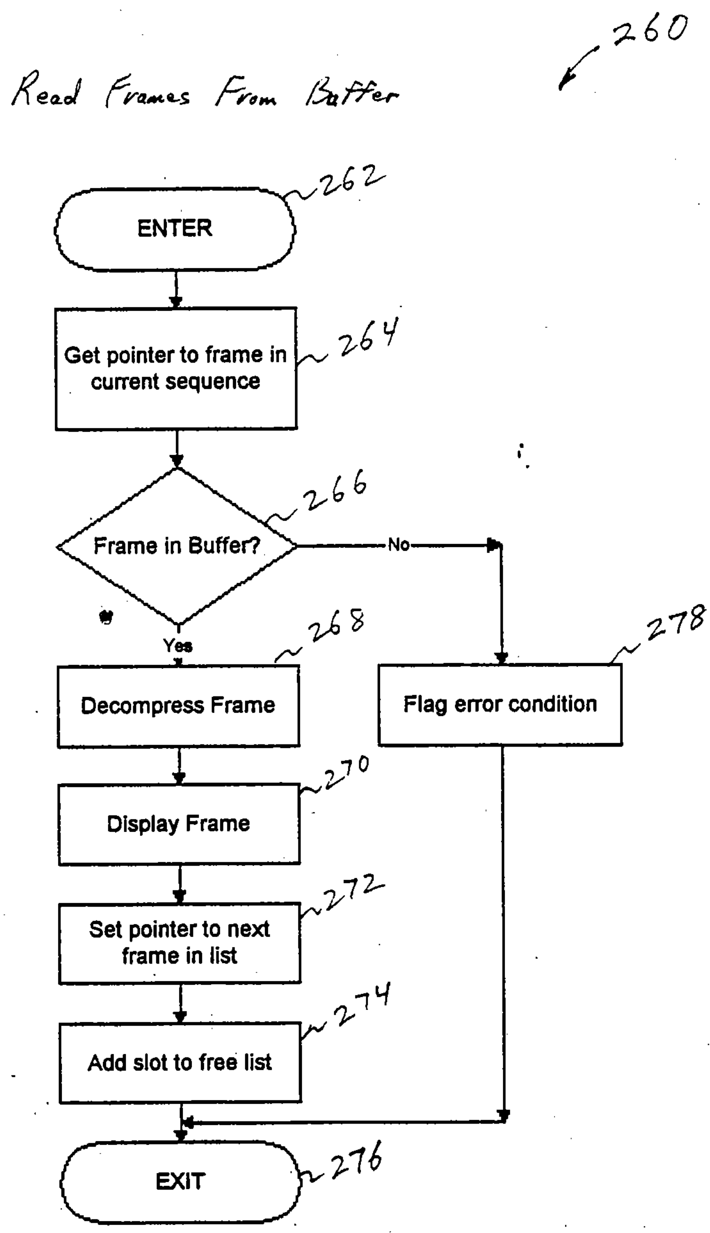


Fig. 12

PRINT OF DRAWINGS AS ORIGINALLY FILED

08/252460

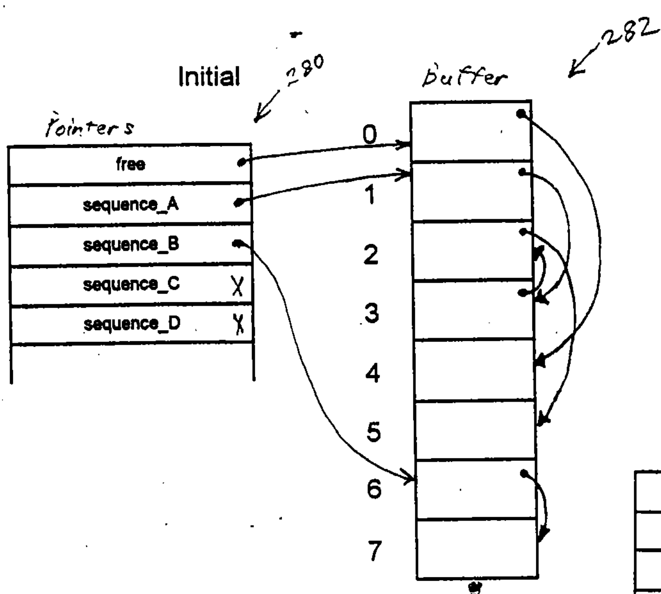


Fig. 13A

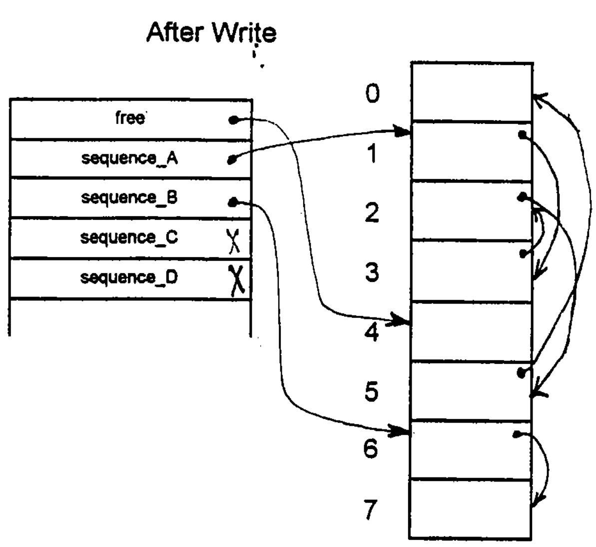


Fig. 13B

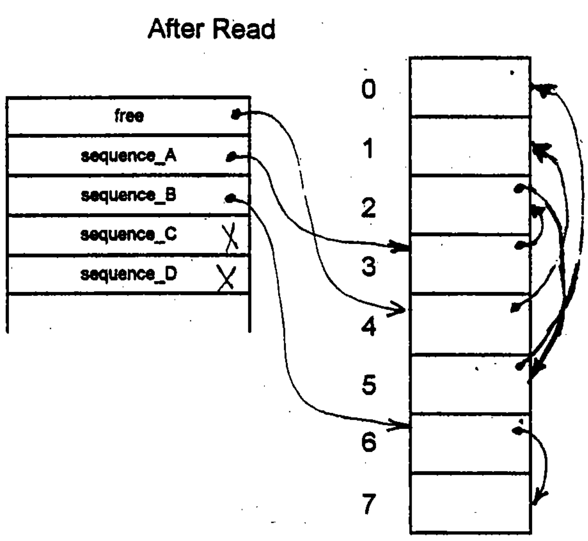


Fig. 13C

PRINT OF DRAWINGS
AS ORIGINA FILED

08/252460

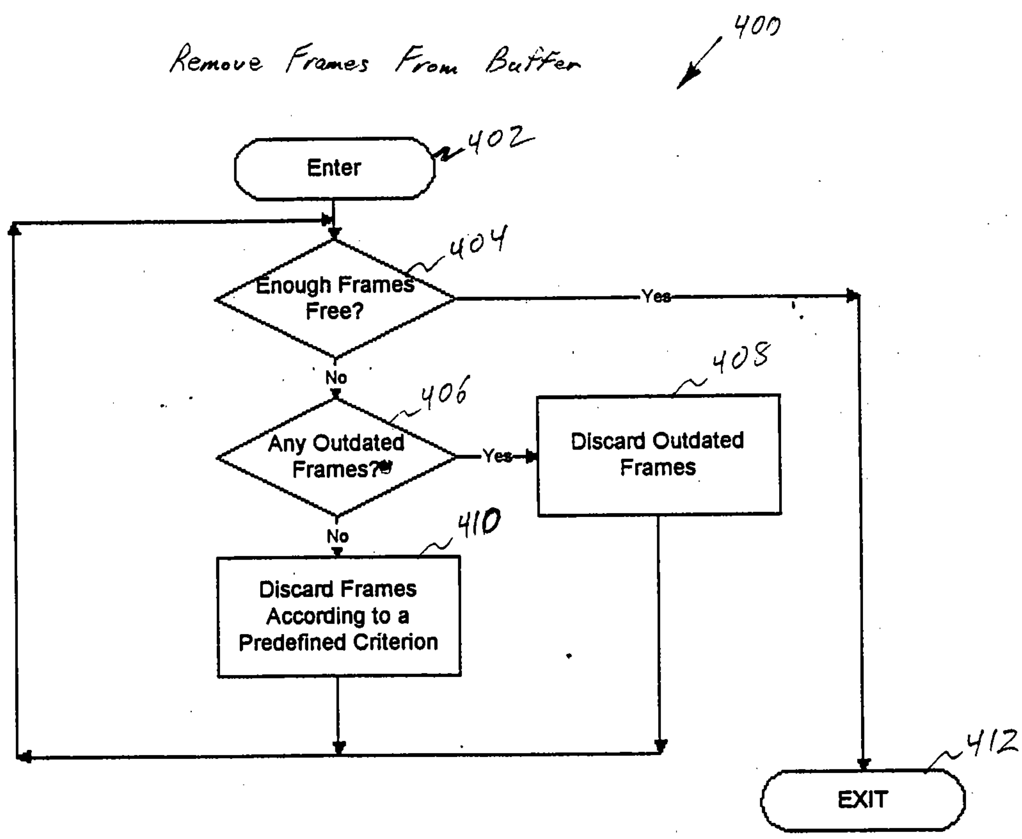


Fig. 14

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/252,460	05/31/94	KULAS	C CJK1

EXAMINER

B3M1/1005
TOWNSEND AND TOWNSEND KHOURIE AND CREW
STEUART STREET TOWER
ONE MARKET PLAZA
SAN FRANCISCO, CA 94105

ART UNIT	PAPER NUMBER
2317	2
2317	

DATE MAILED: 10/05/95

This is a communication from the examiner in charge of your application.
COMMISSIONER OF PATENTS AND TRADEMARKS

☒ This application has been examined ☐ Responsive to communication filed on________ ☐ This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), Ø days from the date of this letter.
Failure to respond within the period for response will cause the application to become abandoned. 35 U.S.C. 133

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☒ Notice of References Cited by Examiner, PTO-892.
2. ☒ Notice of Draftsman's Patent Drawing Review, PTO-948.
3. ☐ Notice of Art Cited by Applicant, PTO-1449.
4. ☐ Notice of Informal Patent Application, PTO-152.
5. ☐ Information on How to Effect Drawing Changes, PTO-1474.
6. ☐ ________.

Part II SUMMARY OF ACTION

1. ☒ Claims 1-7 are pending in the application.

   Of the above, claims ________ are withdrawn from consideration.

2. ☐ Claims ________ have been cancelled.
3. ☐ Claims ________ are allowed.
4. ☒ Claims 1-7 are rejected.
5. ☐ Claims ________ are objected to.
6. ☐ Claims ________ are subject to restriction or election requirement.
7. ☒ This application has been filed with informal drawings under 37 C.F.R. 1.85 which are acceptable for examination purposes.
8. ☐ Formal drawings are required in response to this Office action.
9. ☐ The corrected or substitute drawings have been received on ________. Under 37 C.F.R. 1.84 these drawings are ☐ acceptable; ☐ not acceptable (see explanation or Notice of Draftsman's Patent Drawing Review, PTO-948).
10. ☐ The proposed additional or substitute sheet(s) of drawings, filed on ________. has (have) been ☐ approved by the examiner; ☐ disapproved by the examiner (see explanation).
11. ☐ The proposed drawing correction, filed ________, has been ☐ approved; ☐ disapproved (see explanation).
12. ☐ Acknowledgement is made of the claim for priority under 35 U.S.C. 119. The certified copy has ☐ been received ☐ not been received ☐ been filed in parent application, serial no. ________; filed on ________.
13. ☐ Since this application apppears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11; 453 O.G. 213.
14. ☐ Other

1. Claims 1 - 7 are presented for examination.

2. This application has been filed with informal drawings which are acceptable for examination purposes only. Formal drawings will be required when the application is allowed. See Notice of Draftsperson's Patent Drawing Review form PTO 948.

Applicant is required to submit a proposed drawing correction in response to this Office action. However, correction of the noted defect can be deferred until the application is allowed by the examiner.

3. The disclosure is objected to because of the following informalities:

i. Page 31 line 10, "interval 314 event" is not in any of the drawings.

Appropriate correction is required.

4. The following is a quotation of 35 U.S.C. § 103 which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section

> 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.
>
> Subject matter developed by another person, which qualifies as prior art only under subsection (f) or (g) of section 102 of this title, shall not preclude patentability under this section where the subject matter and the claimed invention were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person.

5. Claims 1 - 7 are rejected under 35 U.S.C. § 103 as being unpatentable over Nguyen, patent no. 5,404,437 in view of Shusuke Hoshi, JP no. 02-121584.

6. As to claim 1, **Nguyen** teaches the invention substantially as claimed, including a method of:

creating a first animation sequence [20 of fig. 1];

creating a second animation sequence [20 of fig. 1];

Nguyen does not teach interleaving the frames of the animation sequences. **Shusuke** teaches another data processing system which performs interleaving on video data [abstract].

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of

Nguyen and Shusuke because they both directed the problem of data processing system. Shusuke's teaching would provide flexibility to Nguyen's teaching by allowing one to create special effects.

7. As to claim 2, a computer system comprising a processor, user input device, display screen, and couple to a CD-ROM are well known in the art. It is well known in the art to use the processor to continuously read the CD-ROM, displaying only relevant frames according to user input.

8. As to claim 3, compression and decompression scheme are well known in the art.

9. As to claims 4 - 7, the claims are rejected analogous to rejected claims 1 - 3, which were addressed supra.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Po Huang, whose telephone number is (703) 308-5230. The examiner can

normally be reached Monday through Friday from 8:00 AM to 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas C. Lee, can be reached at (703) 305-9717. The fax phone number for this Group is (703) 305-9564.

Any inquiry of a general nature of relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Po Huang
September 27, 1995

THOMAS C. LEE
SUPERVISORY PATENT EXAMINER
GROUP 230

Form PTO 948 (Rev. 10-93) U.S. DEPARTMENT OF COMMERCE - Patent and Trademark Office Application No. 252 460

NOTICE OF DRAFTSPERSON'S PATENT DRAWING REVIEW

PTO Draftpersons review all originally filed drawings regardless of whether they are designated as formal or informal. Additionally, patent Examiners will review the drawings for compliance with the regulations. Direct telephone inquiries concerning this review to the Drawing Review Branch, 703-305-8404.

The drawings filed (insert date) 5/31/94, are
A.___ not objected to by the Draftsperson under 37 CFR 1.84 or 1.152.
B. X objected to by the Draftsperson under 37 CFR 1.84 or 1.152 as indicated below. The Examiner will require submission of new, corrected drawings when necessary. Corrected drawings must be submitted according to the instructions on the back of this Notice.

1. DRAWINGS. 37 CFR 1.84(a): Acceptable categories of drawings:
Black ink. Color.
___ Not black solid lines. Fig(s)___
___ Color drawings are not acceptable until petition is granted.

2. PHOTOGRAPHS. 37 CFR 1.84(b)
___ Photographs are not acceptable until petition is granted.

3. GRAPHIC FORMS. 37 CFR 1.84 (d)
___ Chemical or mathematical formula not labeled as separate figure. Fig(s)___
___ Group of waveforms not presented as a single figure, using common vertical axis with time extending along horizontal axis. Fig(s)___
___ Individuals waveform not identified with a separate letter designation adjacent to the vertical axis. Fig(s)___

4. TYPE OF PAPER. 37 CFR 1.84(e)
___ Paper not flexible, strong, white, smooth, nonshiny, and durable. Sheet(s)___
___ Erasures, alterations, overwritings, interlineations, cracks, creases, and folds not allowed. Sheet(s)___

5. SIZE OF PAPER. 37 CFR 1.84(f): Acceptable paper sizes:
21.6 cm. by 35.6 cm. (8 1/2 by 14 inches)
21.6 cm. by 33.1 cm. (8 1/2 by 13 inches)
21.6 cm. by 27.9 cm. (8 1/2 by 11 inches)
21.0 cm. by 29.7 cm. (DIN size A4)
___ All drawing sheets not the same size. Sheet(s)___
___ Drawing sheet not an acceptable size. Sheet(s)___

6. MARGINS. 37 CFR 1.84(g): Acceptable margins:

Paper size

	21.6 cm. X 35.6 cm. (8 1/2 X 14 inches)	21.6 cm X 33.1 cm. (8 1/2 X 13 inches)	21 cm. X 27.9 cm. (8 1/2 X 11 inches)	21 cm. X 29.7 cm. (DIN Size A4)
T	5.1 cm. (2")	2.5 cm. (1")	2.5 cm. (1")	2.5cm.
L	.64 cm. (1/4")	.64 cm. (1/4")	.64 cm. (1/4")	2.5 cm.
R	.64 cm. (1/4")	.64 cm. (1/4")	.64 cm. (1/4")	1.5 cm.
B	.64 cm. (1/4")	.64 cm. (1/4")	.64 cm. (1/4")	1.0 cm.

Margins do not conform to chart above.
Sheet(s)___
___Top (T) ___Left (L) ___Right (R) ___Bottom (B)

7. VIEWS. 37 CFR 1.84(h)
REMINDER: Specification may require revision to correspond to drawing changes.
___ All views not grouped together. Fig(s)___
___ Views connected by projection lines. Fig(s)___
___ Views contain center lines. Fig(s)___

Partial views. 37 CFR 1.84(h)(2)
___ Separate sheets not linked edge to edge. Fig(s)___
___ View and enlarged view not labeled separately. Fig(s)___
___ Long view relationship between different parts not clear and unambiguous. 37 CFR 1.84(h)(2)(ii) Fig(s)___

Sectional views. 37 CFR 1.84(h)(3)
___ Hatching not indicated for sectional portions of an object. Fig(s)___
___ Hatching of regularly spaced oblique parallel lines not spaced sufficiently. Fig(s)___
___ Hatching not at substantial angle to surrounding axes or principal lines. Fig(s)___
___ Cross section not drawn same as view with parts in cross section with regularly spaced parallel oblique strokes. Fig(s)___
___ Hatching of juxtaposed different elements not angled in a different way. Fig(s)___

Alternate position. 37 CFR 1.84(h)(4)
A separate view required for a moved position.

Modified forms. 37 CFR 1.84(h)(5)
___ Modified forms of construction must be shown in separate views. Fig(s)___

8. ARRANGEMENT OF VIEWS. 37 CFR 1.84(i)
___ View placed upon another view or within outline of another. Fig(s)___
___ Words do not appear in a horizontal, left-to-right fashion when page is either upright or turned so that the top becomes the right side, except for graphs. Fig(s)___

9. SCALE. 37 CFR 1.84(k)
___ Scale not large enough to show mechanism without crowding when drawing is reduced in size to two-thirds in reproduction. Fig(s)___
___ Indication such as "actual size" or "scale 1/2" not permitted. Fig(s)___
___ Elements of same view not in proportion to each other. Fig(s)___

10. CHARACTER OF LINES, NUMBERS, & LETTERS. 37 CFR 1.84(l)
X Lines, numbers & letters not uniformly thick and well defined, clean, durable, and black (except for color drawings). Fig(s) 1-14

11. SHADING. 37 CFR 1.84(m)
___ Shading used for other than shape of spherical, cylindrical, and conical elements of an object, or for flat parts. Fig(s)___
___ Solid black shading areas not permitted. Fig(s)___

12. NUMBERS, LETTERS, & REFERENCE CHARACTERS. 37 CFR 1.84(p)
X Numbers and reference characters not plain and legible. 37 CFR 1.84(p)(l) Fig(s) 1-14
___ Numbers and reference characters used in conjuction with brackets, inverted commas, or enclosed within outlines. 37 CFR 1.84(p)(l) Fig(s)___
___ Numbers and reference characters not oriented in same direction as the view. 37 CFR 1.84(p)(l) Fig(s)___
___ English alphabet not used. 37 CFR 1.84(p)(2) Fig(s)___
✓ Numbers, letters, and reference characters do not measure at least .32 cm. (1/8 inch) in height. 37 CFR(p)(3) Fig(s)___

13. LEAD LINES. 37 CFR 1.84(q)
___ Lead lines cross each other. Fig(s)___
___ Lead lines missing. Fig(s)___
___ Lead lines not as short as possible. Fig(s)___

14. NUMBERING OF SHEETS OF DRAWINGS. 37 CFR 1.84(t)
___ Number appears in top margin. Fig(s)___
___ Number not larger than reference characters. Fig(s)___
___ Sheets not numbered consecutively, and in Arabic numerals, beginning with number 1. Sheet(s)___

15. NUMBER OF VIEWS. 37 CFR 1.84(u)
___ Views not numbered consecutively, and in Arabic numerals, beginning with number 1. Fig(s)___
___ View numbers not preceded by the abbreviation Fig. Fig(s)___
___ Single view contains a view number and the abbreviation Fig.
___ Numbers not larger than reference characters. Fig(s)___

16. CORRECTIONS. 37 CFR 1.84(w)
___ Corrections not durable and permanent. Fig(s)___

17. DESIGN DRAWING. 37 CFR 1.152
___ Surface shading shown not appropriate. Fig(s)___
___ Solid black shading not used for color contrast. Fig(s)___

1 of 2

FORM PTO-892 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	Serial No.	GROUP ART UNIT	ATTACHMENT TO PAPER NUMBER	
NOTICE OF REFERENCES CITED	08/252,460	2317		2
	APPLICANT(S) Charles J. Kulas			

U.S. PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	NAME	CLASS	SUB-CLASS	FILING DATE IF APPROPRIATE
	A	5,390,158	02/14/95	Furuhashi	369	47	09/29/93
	B	5,404,437	04/04/95	Nguyen	395	152	11/10/92
	C	5,428,731	06/27/95	Powers, III	395	154	05/10/93
	D	5,446,714	08/29/95	Yoshio et al.	369	48	07/15/93
	E						
	F						
	G						
	H						
	I						
	J						
	K						

FOREIGN PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUB-CLASS	PERTINENT SHTS. DWG	PP. SPEC
	L	01-287872	11/20/89	JP	Yoichiro Sako et al.	-----	-----		
	M	02-121584	05/09/90	JP	Shusuke Hoshi	-----	-----		
	N								
	O								
	P								
	Q								

OTHER REFERENCES (Including Author, Title, Date, Pertinent Pages, Etc.)

	R	"The muddy road to desktop video: Mac companies fudge facts in 'spec wars.' ", Digital Media Nov 10 1992, v2 n6 p13 (4).
	S	"CD-ROMs drive toward new standards.", Datamation, Feb 15 1993, v39 n4 p57 (3).
	T	"Multimedia: Sigma Designs unveils $299 video capture & play-back multimedia adapter for the PC.", EDGE: Work-Group Computing Report, June 14 1993, v4 n160 p12 (1).

EXAMINER Po Huang	09/27/95	Form 892BMR2107

* A copy of this reference is not being furnished with this office action.
See Manual of Patent Examining Procedure, section 707.05(a).

***** Computer Select, November 1993 : Articles *****

1

Journal: Digital Media Nov 10 1992 v2 n6 p13(4)

Title: The muddy road to desktop video: Mac companies fudge facts in 'spec wars.'
(the difficulties in implementing 640-by-480-pixel full-screen, full motion digital video on the Macintosh are exemplified by fudges in the New Video's EyeQ boards, SuperMac Technology's DigitalFilm dual boards and RasterOps MoviePak board)

Full Text:

Mac companies fudge facts in 'spec wars'

The quest for full-screen, full-motion digital video on the Macintosh rivals the search for the Holy Grail. It is a frenzied and sometimes downright nasty competition. To date, three desktop hardware companies actually claim to have captured the Grail: New Video -- the only one shipping a product -- SuperMac Technology and RasterOps.

Each has made claims that its product has the capability to capture, display and output 30 frames of video per second at "full screen." This is true on the surface, since all three products can display a full screen of video at full speed.

However, for most people in the computer industry full screen means 640x480 pixels, with no cheating. None of these companies' products -- at least in their current versions -- actually achieve capture at this resolution during capture and/or output.

What's so hard about full-screen, full-motion video?

Although computers use display screens that look an awful lot like television screens, one of the hardest things to do on a computer is to process and display full-frame, full-motion video images. The reason is that, at present, analog television and digital computers have remarkably little in common.

The desired result is to digitize, compress and decompress full-frame (640 pixels by 480 scan lines for NTSC video), full-motion (30 frames per second, or fps) video, and full CD-quality sound.

Most companies, especially those developing desktop products, use techniques such as interpolation and doubling the video scan lines to expand a lower-resolution image to fill a 640x480 window and achieve the illusion of 640x480 video.

What's the big deal? The first problem is that in order to avoid the appearance of flicker at the 30-fps refresh rate used for NTSC (the North American broadcast television standard), TV images are "interlaced": a TV first paints a "field" containing all the odd-numbered scan lines, then a second field containing all the even-numbered scan lines.

Computer monitors, which are no longer interlaced, have three choices for dealing with this problem.

First, they could use a memory buffer to interleave both the fields, then paint them both onto the screen at once. This is very expensive, because two memory buffers would be required: one for the frame in progress, and one for the frame right behind it. A control mechanism would have to ping-pong between the two buffers to continually refresh the monitor.

The second choice is to pick up one field of scan lines and simply paint every one of those scan lines twice, essentially throwing away half of the video information.

The third, which not many do, is to interpolate the missing scan lines, or take an average between Scan Line 1 and Scan Line 3, for example, and paint that as Scan Line 2.

Choosing samples. Those solutions deal with the interlacing problem. The other decision to make is the number of samples across each scan line. To maintain the 4:3 aspect ratio of a 480-line television image on a computer screen with square pixels requires 640 samples per scan line. This is more than twice the resolution of VHS videotape and one-and-a-half times the resolution of a laserdisc.

In the interest of reducing the amount of digital data, most vendors actually take 320 samples per scan line. They have to interpolate the missing data to blow this up to 640 pixels for a full 640x480 image display.

A true 640x480 digital television image displayed on a computer-quality monitor would be stunning -- far better than anything you see on your television set. An enlarged representation of 320x240 image -- especially one that has been compressed and decompressed -- will not look nearly as good. However, if the technology used is really clever, it may look as good as many TV images -- certainly good enough for a great many applications.

Taking 640 samples per scan line (as SuperMac appears to do) should yield a distinctly better picture. Ultimately, as the technology improves we will get "true" 640x480 digital video on computer screens, but in the meantime, purveyors of good-looking desktop video ought to be proud of themselves for the progress they've made to date.

Come on, come clean. New Video and RasterOps admit to doubling the scan line and interpolating the "square pixels" from interlaced NTSC video to produce full-screen video on a display device.

The only difference between these two and SuperMac, according to many sources outside of the company, is that SuperMac won't admit it's fudging. Steve Blank, SuperMac's vice president of marketing, maintains that the company's DigitalFilm board is powerful enough to sample 640x480's worth of pixels -- though others who've used the product say it is actually doing only 640x240. Blank refused any other comment on this article, and SuperMac did not return repeated telephone calls by press time.

In addition, SuperMac is publicly bad-mouthing RasterOps as an inferior solution. Not only is this bad karma and in bad taste, but these kinds of spitting matches are enormously confusing to the customer. It might be more productive for them all to address the real issues surrounding widespread adoption of digital video products: storage requirements, which to date are enormous, even with compression, and image quality, which in all cases is somewhat south of perfect.

What do they do, other than kvetch?

Products by all three companies do an excellent job of capturing analog video, converting it to digital, compressing it for storage or transmission, then decompressing it with a minimal number of artifacts so it can be displayed on a computer screen or output to videotape or television. When you consider the technical heroics that must be accomplished to do this using a desktop computer -- taking the analog standards of NTSC or PAL and actually figuring out a way to get things like scan lines and vertical blanking intervals sampled into digital format -- all three companies have earned great applause.

If they would stop mudslinging long enough to look at the desktop video market, they would realize that right now not one of the three has a competitor in its chosen segment.

New Video goes after presentation market

To be fair, New Video has not been much of a participant in the mud baths. Its EyeQ product - actually the first and only product to come to market at this point -- was shipped in July without fanfare, hissing or spitting. The company is young and does not yet have the marketing and sales power of either SuperMac or RasterOps.

Its product, designed for the Macintosh in either a two-board set for digitizing and playing back video ($4,495) or with one board for play-back only ($2,495), is targeted toward the multimedia presentation market, including networked video for education and training. Like the other two products, EyeQ sports the standard specs: It can capture, compress and display simultaneous video and audio. It can capture and display 30 frames of video per second and make it appear at full screen size without noticeable artifacts. It supports composite, S-Video or RGB input, as well as NTSC and PAL broadcast standards.

But EyeQ has some obviously superior design features that have snagged the interest of such companies as Apple (it was the only board Apple used for digital video demonstrations during its Worldwide Developers Conference) and other large multimedia developers.

For one, EyeQ has a DSP chip on board, which can sample 16-bit sound at CD-quality levels without requiring participation by the computer's central processor -- a real plus considering the sheer mass of audio and video files. It can also play back full-motion video even from slow data rate devices, including CD-ROM and networks, a feat none of its competitors has mastered.

In addition, EyeQ uses Intel's vastly improved new DVI programmable video processing technology, which, in turn, provides the product with the capability to support a variety of compression schemes and hardware platforms.

A prescient move. In what turns out was a prescient move, New Video decided to buck the past trend toward JPEG compression and built Intel's i750 video codec chip set into EyeQ. The i750 supports a variety of algorithms, including RTV (real-time video) and PLV (production-level video), JPEG for high-resolution still images and ADPCM for audio. It can easily be reprogrammed to support new algorithms as they're developed. Cases in point: New Video showcased a proprietary near-broadcast quality (60 fields per second) video compression algorithm during Macworld Boston this year, and Microsoft is using the DVI chip set in its Video for Windows product (see p. 19).

EyeQ also offers the allure of cross-platform playback capabilities, since it supports RTV 2.1, a cross-platform, scalable video algorithm supported under DVI. This technology enables video output from the Macintosh in the same file format supported by IBM's ActionMedia board, which in turn allows playback of EyeQ-created video on any computer equipped with ActionMedia.

One drawback to the existing EyeQ board set for many consumers is that it does not provide support for Adobe Premiere 2.0 and Diva Video-Shop, both popular with desktop video users. A free software upgrade that supports these products is on the way for those who already own the EyeQ hardware. EyeQ does support Quick-Time as well as applications such as HyperCard, IMC's Special Delivery and MacroMedia Director.

RasterOps takes a modular approach

The modular approach taken by RasterOps to digital video technology makes it particularly attractive to the professional video production market.

The company has created a family of products with the expectation that professional video editors will want to customize their digital video studios. This approach allows producers to take advantage of certain components they already own, which in turn helps them configure a system that suits their particular needs.

RasterOps's digital video is based on MoviePak, a product that is still in beta testing and not expected to be shipped until mid-November. Movie-Pak is a $1,999 add-on card that must be attached to one of the company's 24-bit video display boards.

MediaTime, one of RasterOps's two 24-bit cards that can drive a 13-inch monitor, also provides CD quality sound capture and playback. The card uses DigiDesign's AudioMedia chip set to sample 16-bit audio. The other RasterOps configurations require the use of an audio-digitizing card from a third party.

For output to videotape or to a TV monitor, RasterOps opted for an off-board encoding/decoding solution, thus offering professionals the chance to use high-end encoders/decoders.

What's required is either RasterOps's Video Expander, a $699 external controller box that enables video editors to output encoded video to a VCR or TV monitor with RGB pass-through, S-Video, and/or composite video connections, or a RasterOps-compatible encoder/decoder box. Video Expander has genlocking capabilities as well.

MoviePak uses a compression chip set from LSI Logic Corp. that supports a variety of ratios from 2:1 to 160:1. When it's first released, it will be bundled with Adobe Premiere 2.0 video editing software.

SuperMac offers a single-slot solution

The high-end video professionals moving down to the Macintosh and the multimedia presenters market are only part of the digital video puzzle. There is yet another segment left in the market, one that SuperMac's technology comfortably fills.

SuperMac's DigitalFilm product is a double-card, single-slot configuration and an external box that makes it an easy choice for computer users who have begun experimenting with digital video. It is what is known as a plug-and-play solution -- albeit an expensive one.

DigitalFilm, now in beta form, is expected to retail at $5,999. It's a NuBus card set that handles simultaneous video and audio, video compression and decompression, and video encoding and decoding, in a package that only requires a single Macintosh slot. SuperMac also bundles Adobe Premiere 2.0 video editing software with the DigitalFilm board set.

DigitalFilm fits into a Mac IIfx or, preferably, into one of Apple's Quadra series of computers. (Both RasterOps and New Video have similar minimum configuration requirements.) Although it can capture, playback and output video at the full-motion standard of 30 frames per second, DigitalFilm does not yet support true CD-quality (16-bit) audio on board, though the company says it will.

Compression and decompression is on board via C-Cube Microsystems' JPEG hardware for still images. SuperMac calls this "motion JPEG," though there is actually no such standard. JPEG was not designed for time-based media, either audio or video. That said, the on-screen image quality produced by SuperMac's products is excellent, even in beta. The card can handle a compression ratio of up to 70:1.

A piece of the pie for everyone

The irony of the spec wars, where vendors fudge the facts on paper to get a "perfect 10," is that these details ultimately don't matter. Desktop video customers look to the personal computer as an inexpensive tool for creating compelling prototypes for clients and storyboards for film and video projects. They do not yet use desktop technologies to produce final broadcast output.

Customers, in particular the professional video editors evaluating digital video products, look for image quality. They don't really care about the technical details of "full-motion video running at a resolution of 640x480," as long as the quality of the video is good and the storage requirements are somewhat reasonable, at least in comparison to other solutions.

They won't even notice. Hobbyists dabbling in desktop video aren't likely to challenge vendor claims. Even if they notice the difference, they aren't likely to care. The same disregard for specs holds true when you get to the presenters or business communicators. For the most part, they're seeking something that's easy to use and makes them look good in front of their bosses and peers. They don't want or need to know about square pixels and doubling scan lines. But this doesn't mean they shouldn't be told the truth.

Desktop video is not a war, nor should it be seen as an exercise in ego gratification. It is an industry and it appears that there are plenty of customers to go around -- at least there had better be, since there are many powerful and well-regarded companies like Radius that are just a few months behind in announcing these types of products themselves. It's a good bet that customers would like to see companies stop the irrelevant competitor-bashing and start figuring out how to fix the storage and image-quality problems that are really holding back digital video from widespread adoption.

Company: SuperMac Technology Inc.
RasterOps Corp.
New Video Corp.
Product: Apple Macintosh
New Video EyeQ
RasterOps MoviePak
SuperMac Technology Digital Film
Topic: Video Boards/Cards
Industry Analysis
System Design
Display Capacity
Functional Capabilities

Record#: 12 976 419.

*** End ***

***** Computer Select, November 1993 : Articles *****

(2)

Journal: Datamation Feb 15 1993 v39 n4 p57(3)

Title: CD-ROMs drive toward new standards.
(includes related article on recording standards for CD-ROM technology)

Author: Francis, Bob

Abstract: CD-ROM drive standards are still evolving. A majority of the CD-ROM drives now being marketed conform to the International Standards Organization's 9660 standard, which was established in 1988. The ISO 9660 closely simulates the MS-DOS style of naming documents. The ISO 9660 is sufficient for limited graphic use and conducting basic search and retrieval tasks. For information system managers, products with the CD-ROM XA standard may be a better choice. The XA is targeted at business applications such as multimedia and audio. CD-ROM XA drives also support still images by using a screen presentation format. Another CD-ROM standard exists for products such as those from North American Philips and Sony.

Full Text:

Just when you thought it was safe to plunk down money for a CD-ROM drive, along come new standards issues to consider.

Multimedia, image capture, full motion video and network data storage are tantalizing new technologies for Is managers looking to the future. And CD-ROM drives will be a key component in these technologies, to be sure. Yet for Is managers looking to invest in these systems over the next few years, there is plenty of concern over what stan dards will eventually dominate. That's particularly true of today's CD-ROM drives, since the standards a particular product supports may limit what can be done with it in the future.

Most of the CD-ROM drives sold today conform to the International Standards Organization's 9660 standard. Established in 1988, the standard closely approximates the MS-DOS style of naming files. For most basic text search and retrieval applications and for limited graphics use, the Iso 9660 standard is adequate. But as potential buyers examine future directions for the technology, they soon run into an alphabet soup of standards that are extensions to the ISO 9660. Among them are CDROM XA, a multimedia business standard; CD-Interactive, another multimedia standard aimed more at the consumer market; Multimedia CD (MMCD), a standard aimed at portable players; and multisession CD, a standard for inhouse CD-ROM recorders (see sidebar, "In Search Of A Recording Standard").

For IS managers, the XA standard may bear the most watching. CD-ROM XA is aimed at such professional applications as business audio and multimedia. Because they can interleave audio information with simultaneous data retrieval, CD-ROM XA drives can run multimedia applications more efficiently than ordinary Iso 9660 drives can.

CD-ROM XA drives support still images using a screen presentation format, but not full-motion video. True full-motion video standards are currently being considered, says Gess Shankar, director of technical research at CBIS Inc. in Norcross, Ga., a maker of network CD-ROM servers. According to Shankar, whatever full-motion video standard wins is likely to be compatible with the current CD-ROM XA standards.

One video standard that's already in sync with CD-ROM XA is the new Photo CD format from Eastman Kodak Co. of Rochester, N.Y. Kodak's Photo CD format stores photographs on CD-ROMs. These can then be viewed and manipulated on screen using either an XAcompatible drive or a special Photo CD player aimed at the consumer market.

When the Photo CD was announced in August 1992, Kodak also introduced software designed for business users. This includes Kodak Shoebox, an image database program that contains search and retrieval functions.

A handful of XA-compatible drives already on the market can take advantage of this software now. These include two drives from San Jose-based Sony Corp. of America: the CDU-31A-02 and the CDU-7211. Texel, in Santa Clara, has also introduced an XA drive, the DM5024. Cupertino, Calif.based Apple Computer Inc.'s CD-ROM drives also conform to the XA standard, and many other CD-ROM drives coming this year will also likely be xA-compatible.

Moreover, the new chip sets needed to implement XA compatibility don't add much cost to the final product. Texel's drive, for instance, lists for $599 with a Small Computer Systems Interface (scsi) port, cables and a caddy, well within range of many CDROM drives on the market.

CD-ROM XA should be more than enough to meet the demands of many future applications involving sound and perhaps some limited video capability. That's what Cummins Engine Co. Inc. of Columbus, Ind., has found. Cummins sends its parts catalog to its service centers via CD-ROM or microfilm, depending on which technology the center desires. According to William Seltzer, Cummins technology information director, the ordinary IsO 9660 drives he has are "more than adequate for what we're doing now and, if we go further, the price may be more important than standards," he says. Seltzer won't rule out XA drives or whatever future standards might emerge for applications now being planned, in cluding service manual CD-ROMS that incorporate training films on disc. The XA standard may set a good stake in the ground for corporate applications, but other standards will likely be important, as well. Companies considering using the new portable multimedia CD players from North American Philips Corp. and Sony should be aware that they conform to two different standards.

The Philips Portable CD-I 360 handheld multimedia CD player uses the CDInteractive standard. Some CD-ROM players may be sold as CD-I compatible, but at this time only the Philips players can be guaranteed to be compatible with CD-I software titles and also play XA-compatible discs. Sony's Multimedia CD Player, like the Philips device, is another portable multimedia player that can play XA discs. But Sony is pushing its MMCD standard as the preferred standard, rather than CD-I, for its portable player.

The main difference between XA and the two portable standards being promulgated is that XA is relatively plat* form independent. That is, it's supported by far more manufacturers than either the CD-I and MMCD formats. In addition, XA better handles database and textual environments more geared toward sophisticated search and retrieval. CD-I and MMCD target graphicsintensive applications, such as training programs, that don't use a great deal of cross references and don't require sophisticated searching techniques.

Nevertheless, CD-I and MMCD are finding some application in business. Caltex Petroleum Corp. in Dallas is using the Philips CD-1 360 as a sales force automation tool, allowing salespeople to carry manuals, technical reports and product specification sheets to user sites on disc. Trevor Ford, manager of training and development at Caltex, says the firm plans to publish its entire product catalog and other materials on two discs. "If the customer is interested in further information, we can connect the CD player to a printer and print out a sales sheet," Ford says.

Ford does not foresee any problems with standards, despite his early commitment to CD-I. He says that's because it's relatively easy to download the same information to other formats. The portability of the CD-I 360 was his primary reason for choosing Philips device for his sales force, not any preference for its CD-I standard.

Equipping everyone in an organization with his or her own CD-ROM drive is a luxury that few organizations can afford. Fortunately, a number of CDROM products have emerged that work over standard networks. CBIS, for instance, manufactures the CD-ROM Server 2000 series, a CD-ROM network server that allows up to 250 users on a NetBIOS network to simultaneously access up to 14 CD-ROM drives. Softwareonly solutions are also available. Online Computer Systems Inc. of Germantown, Md., offers the software necessary to network existing CD-ROM drives.

Not all networks can handle such products equally well, so caution is warranted. Judy Brown, manager of the computer resource center at Fox Valley Technical College in Appleton, Wis., found this out when she looked for a way to expand access to a CD-ROM technical database. None of the products she's seen is so far compatible with her company's network, IBM's LAN Server. "I suspect there will be a solution out there that will allow CD-ROM access on heterogeneous networks before long," she says hopefully.

Despite such glitches, current standards have evolved far enough to allow many to safely invest in CD-ROM technology. Cummins Engine's Seltzer counts himself in this camp. His firm committed itself early on to supporting the ISO 9660 standard, and it's proven to be an important factor in the catalog application project's success. "We have service bureaus in over 350 countries, so we have to make sure what we give them adheres to a worldwide standard if at all possible," he says. Otherwise, if the equipment breaks down and can't be repaired or replaced locally, it's of little use. "From the beginning of the project we believed in sticking to the standard, and we'll be looking at standards as we move on with other projects."

In Search Of A Recording Standard

If CD-ROM technology is really to take off as a corporate Is tool, most analysts agree that in-house recorders will have to be readily available and easy to use. That's not the case now, says Mike Peterson, market analyst at Peripheral Strategies Inc. in Santa Barbara, Calif.

A CD-ROM recorder, or CD-R, employs write-once CD technology that writes information on a special CD-ROM disc in a standard way. Most companies now rely on service bureaus to produce their CDROM titles for them. But, as the COSt Of CD-R technology comes down, more companies may choose to roll their own. "The companies are really just getting out of the gate with reasonably priced versions of these products," says Peterson.

The three big players in the game are JVC Information Products Co. of America in Santa Clara, with its Personal RomMaker For DOS; Sony Recording Media of America in Park Ridge, N.Y., with its CDw-900E; and Philips Consumer Electronics Co. in Knoxville, Tenn., offering the CDD 521Cw desktop CD recorder. Prices range from $7,995 for a Philips machine to $12,799 for the Jvc offering.

The software included with the recorders causes some of the variance in price, so it's obviously no small investment. But it is cheaper than the $20,000 + prices paid a few years back, notes Peterson. Blank CD-R discs are about $40 apiece, and each can hold from 540 to 630 megabytes of information. Recording takes about an hour for one disc.

Most CD recorders today force users to write all of the data to a CD-ROM in a single session. If you add more data to an unfilled disk at a later time, most CD-ROM drives won't be able to read them. It's another standards issue that's yet to be resolved. The current ISO 9660 standard does not allow for more than one table of contents on a disc. However, an extension to the ISO 9660 standard is expected to be approved before long covering CD-ROMS created over multiple sessions.

Several firms are offering packaged versions of these CD-ROM recorders that will make them easier to use. Meridian DataInc. of Scotts Valley, Calif., plans to repackage Philips' CD-ROM recorder with its software for use on Novell networks. Meridian's Netscribe 1000 is expected to be available in March. Dataware Technologies Inc. of Cambridge, Mass.. also sells a variety of software and hardware packages for CD-ROM recording.

Topic: CD-ROM
Disk drives
Standards
Trends

Record#: 13 472 114.

*** End ***

***** Computer Select, November 1993 : Articles *****

Journal: EDGE: Work-Group Computing Report June 14 1993 v4 n160 p12(1)
3

Title: Multimedia: Sigma Designs unveils $299 video capture & play-back multimedia adapter for the PC. (Sigma Designs Inc.'s WinMovie)

Full Text:

Sigma Designs Inc. Monday announced WinMovie, the lowest-priced, full-motion video capture and play-back board available in the market today.

WinMovie is optimized for creating Audio Video Interleave (AVI) movies under Microsoft Video for Windows and is capable of capturing video at up to 30 frames per second (fps) from any NTSC or PAL composite and S-Video device, such as a VCR, video camera or laser disc player. Video sequences captured by WinMovie can be edited and used for professional multimedia presentations, interactive training and educational materials.

"For less than $300 WinMovie allows anyone who owns a PC to experience the excitement of multimedia technology," said Pier Del Frate, director of marketing at Sigma Designs. "Multimedia enables better communication through computers by adding live action visuals to presentations and training tools. WinMovie was designed to make that process inexpensive and very end-user friendly."

WinMovie supports a maximum video capture and display resolution of up to 640 x 480, as well as 320 x 240, 240 x 180, 60 x 120, and 80 x 60 resolutions in 8-bit, 16-bit or 24-bit colors. WinMovie's interface allows the board to be fully compatible with all superVGA adapters and does not require a VGA feature connector. In addition, WinMovie is compatible with any PC 386/33 MHz or faster. Software capable of controlling all aspects of the video -- brightness, contrast, saturation, and hue -- is included with the adapter.

Microsoft "Video for Windows," which normally carries a $199 suggested list price, comes bundled for free with WinMovie. Video for Windows allows users to immediately create their own AVI movie library for multimedia presentations or entertainment. Video for Windows supports capturing full-motion video or single frames, synchronizing audio, and a variety of editing features -- cut, copy, paste, insert, and delete -- customizing video files for use with other applications.

Xing Technology "Picture Prowler" software utility is also bundled with WinMovie. Picture Prowler is an image database management program for Windows capable of creating thumbnail sketches of JPEG images and decompressing them for full screen viewing. AVI movies can also be played from within the Picture Prowler application by selecting a thumbnail icon of an AVI designated file.

For easy installation, Sigma Designs provides WinMovie DTV utility software with the WinMovie adapter. The WinMovie DTV utility features a familiar TV and VCR-style user interface which allows users to quickly install the WinMovie board and preview video, as well as view and play-back AVI files.

WinMovie will be available in mid-June 1993 for a suggested list price of $299. The package includes the WinMovie image-capture adapter, Microsoft "Video for Windows," Xing Technology "Picture Prowler," Sigma Designs WinMovie DTV utility, and complete documentation. Demonstrating Sigma Designs' commitment to quality, service and support, a five-year extended parts and labor warranty comes standard with WinMovie.

WinMovie rounds out Sigma Designs' numerous multimedia offerings which include WinStorm 24-bit video and 16-bit sound adapter; WinSound 16 CD-quality stereo sound adapter; WinStorm CD-ROM Upgrade Kit; WinSound 16 CD-ROM Upgrade Kit; as well as the recently announced Movie video and sound capture controller for Macintosh computers.

Sigma Designs, headquartered in Fremont, CA, is a leading manufacturer of high-resolution color, grayscale and monochrome display systems, as well as multimedia and video display adapters. Products are available for both IBM PC and Macintosh compatible computers and are designed for end-user productivity and comfort. All Sigma products are sold worldwide through a network of dealers and distributors. FMI: 800/845-8086 or 510/770-0100.

Type: product announcement
Company: Sigma Designs Inc.
Product: Sigma Designs WinMovie
Topic: Product Introduction
Graphics Boards/Cards
Multimedia technology

Record#: 13 967 065.

*** End ***

US005390158A

United States Patent [19]

Furuhashi

[11] Patent Number: 5,390,158

[45] Date of Patent: Feb. 14, 1995

[54] METHOD FOR RECORDING AND REPRODUCING COMPRESSED AUDIO PCM DATA ON AND FROM OPTICAL DISC USING ADJUSTABLE INTERLEAVING FACTOR

[75] Inventor: Makoto Furuhashi, Kanagawa, Japan

[73] Assignee: Sony Corporation, Tokyo, Japan

[21] Appl. No.: 128,950

[22] Filed: Sep. 29, 1993

Related U.S. Application Data

[62] Division of Ser. No. 976,900, Nov. 16, 1992.

[30] Foreign Application Priority Data

Nov. 22, 1991 [JP] Japan 3-333911

[51] Int. Cl.^{6} G11B 5/09; H04N 5/76

[52] U.S. Cl. 369/47; 369/59

[58] Field of Search 369/47, 48, 49, 59; 360/8, 32, 48

[56] References Cited

U.S. PATENT DOCUMENTS

4,829,497 5/1989 Sako et al. 369/48
5,224,087 6/1993 Maeda et al. 369/47

FOREIGN PATENT DOCUMENTS

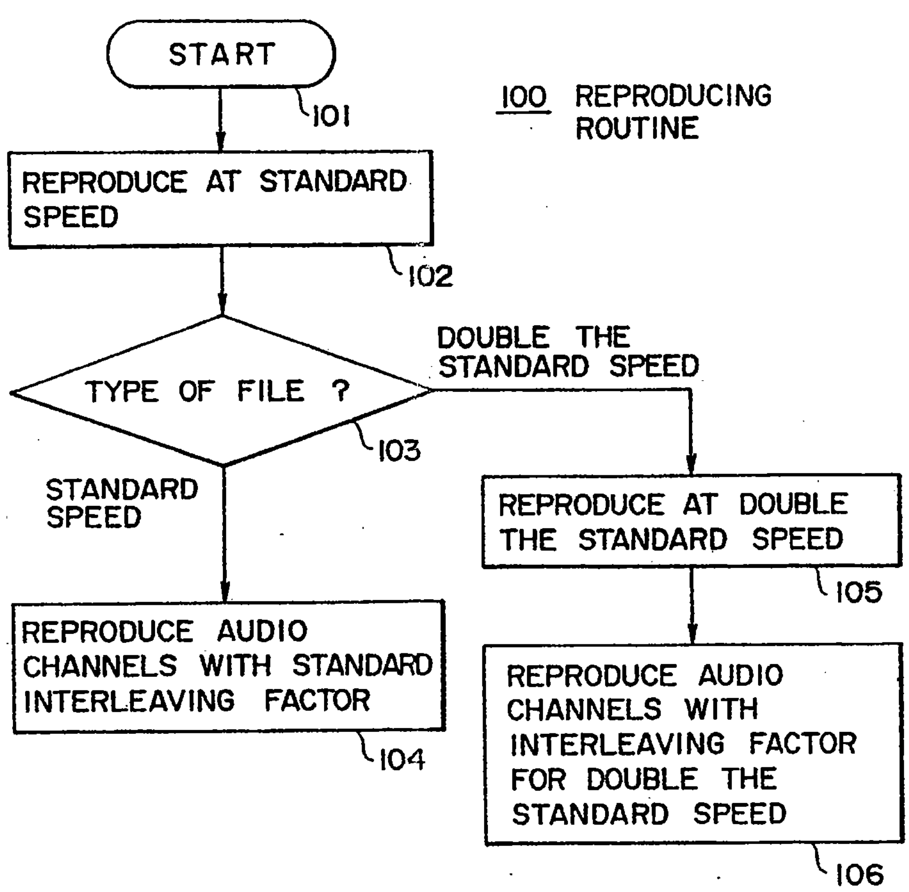
0233783 8/1987 European Pat. Off. .
0238194 9/1987 European Pat. Off. .
0440224 8/1991 European Pat. Off. .
0463183 1/1992 European Pat. Off. .
0464216 1/1992 European Pat. Off. .
9111002 1/1991 WIPO .
9111003 7/1991 WIPO .

Primary Examiner—Donald Hajec
Assistant Examiner—Won Tae C. Kim
Attorney, Agent, or Firm—Jay H. Maioli

[57] ABSTRACT

Compressed audio signal data are recorded on a CD-ROM disc at a data transfer rate higher than the standard rate. Information indicating the increased data transfer rate is recorded on the disc, and the compressed audio signal data as recorded are interleaved on the disc in accordance with; the increased data transfer rate.

6 Claims, 11 Drawing Sheets



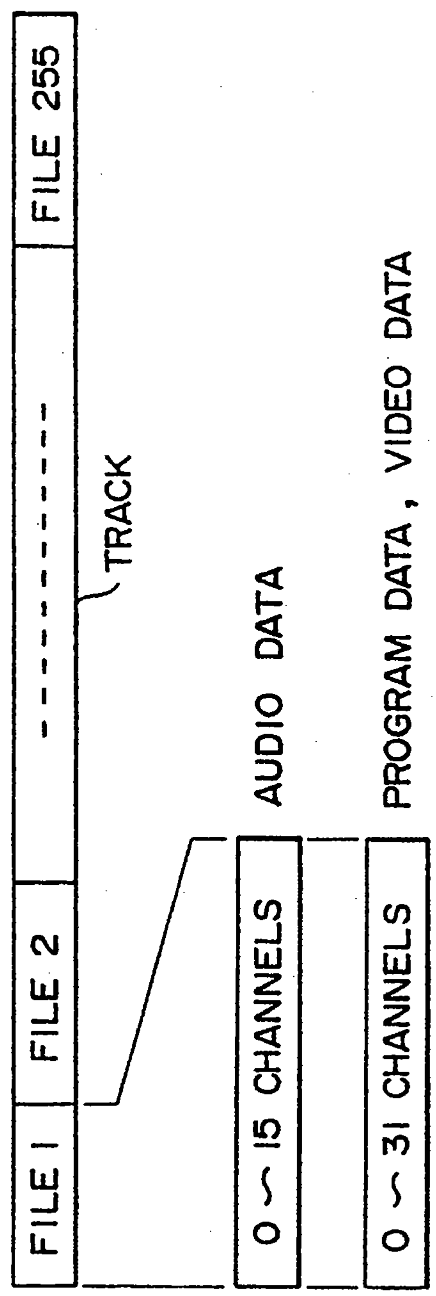


FIG. 1(A) RELATED ART

FIG. 1(B) RELATED ART

FIG. 1(C) RELATED ART

FIG. 2
RELATED ART

FORMAT	LEVEL AND SOUND SYSTEM	SAMPLING FREQUENCY	BIT COUNT	FREQUENCY BAND	INTERLEAVING FACTOR
a	STEREO	37.8 kHz	8	17 kHz	2
b	MONAURAL	37.8	8	17	4
c	STEREO	37.8	4	17	4
d	MONAURAL	37.8	4	17	8
e	STEREO	18.9	4	8.5	8
f	MONAURAL	18.9	4	8.5	16

FIG. 3
RELATED ART

BLOCK TRAIN →

FORMAT	LEVEL AND SOUND SYSTEM \ BLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	....
a	STEREO	O		O		O		O		O		O		O		O		O		
b	MONAURAL	O				O				O				O				O		
c	STEREO	O				O				O				O				O		
d	MONAURAL	O								O								O		
e	STEREO	O								O								O		
f	MONAURAL	O																O		

1/75 sec.

1 BLOCK

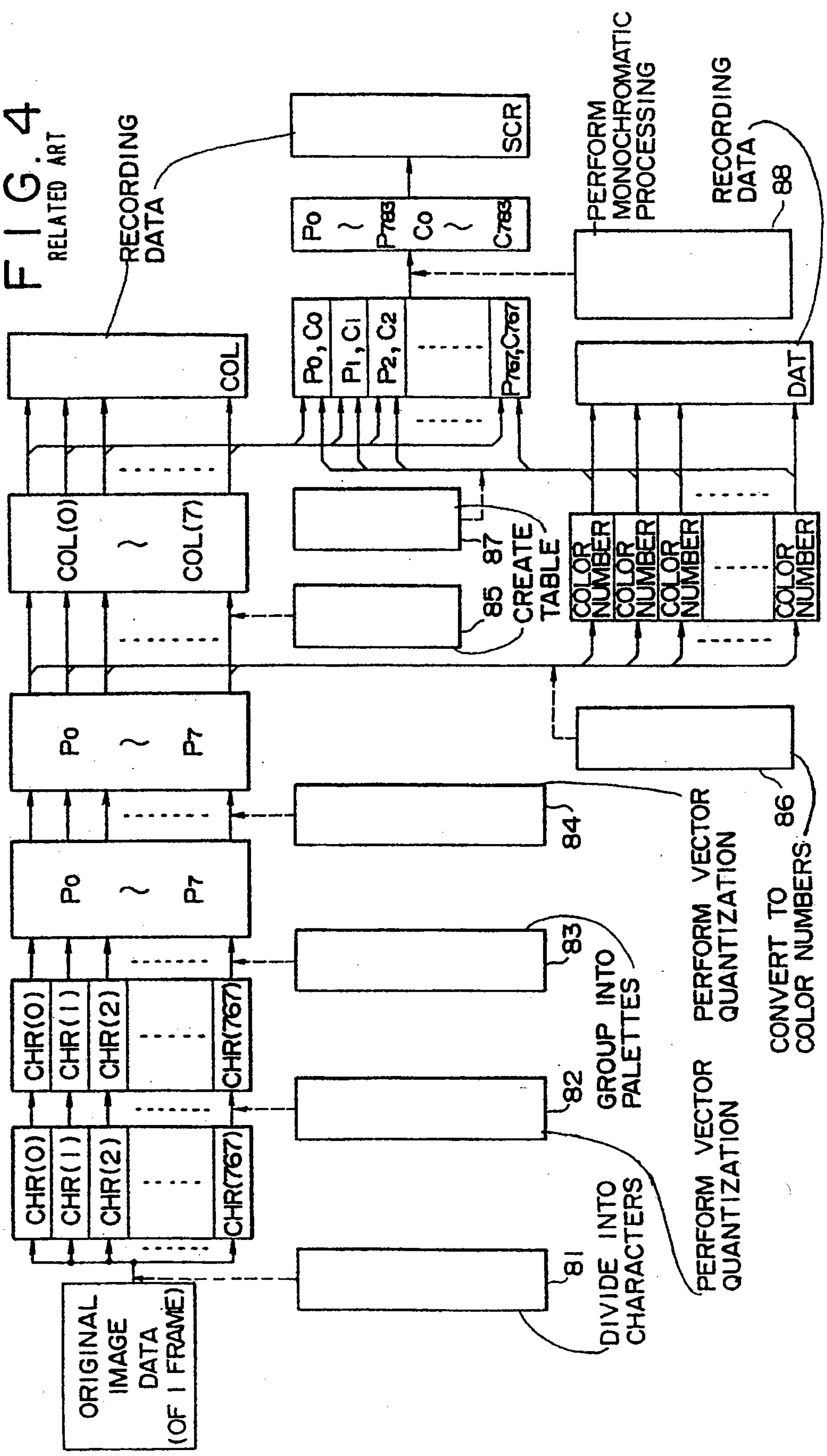


FIG. 4 RELATED ART

FIG. 5(A)
RELATED ART

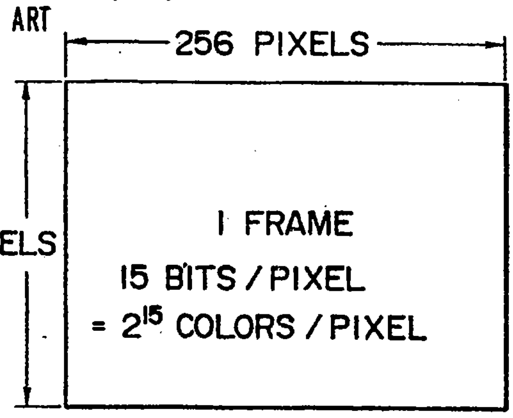


FIG. 5(B)
RELATED ART

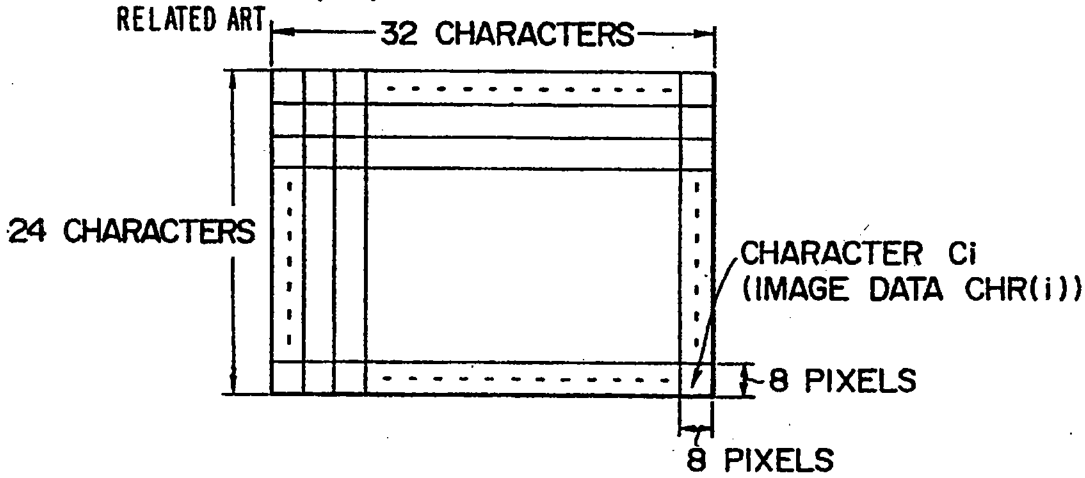


FIG. 5(C)
RELATED ART

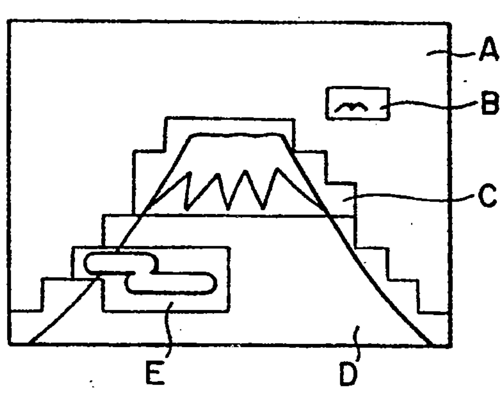


FIG. 6
RELATED ART

COLOR NUMBER	0	1	2	------	15
DATA OF 0-TH PALETTE	COLOR DATA (IN 15 BITS)	COLOR DATA	COLOR DATA	------	COLOR DATA

COLOR NUMBER TABLE COL(0)

COLOR NUMBER	0	1	2	------	15
DATA OF FIRST PALETTE	COLOR DATA	COLOR DATA	COLOR DATA	------	COLOR DATA

COLOR NUMBER TABLE COL(1)

COLOR NUMBER	0	1	2	------	15
DATA OF SEVENTH PALETTE	COLOR DATA	COLOR DATA	COLOR DATA	------	COLOR DATA

COLOR NUMBER TABLE COL(7)

FIG. 7
RELATED ART

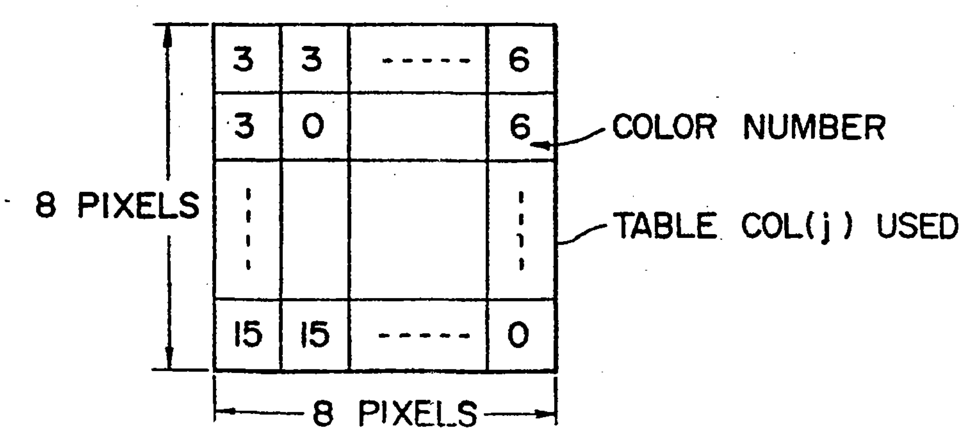


FIG. 8
RELATED ART

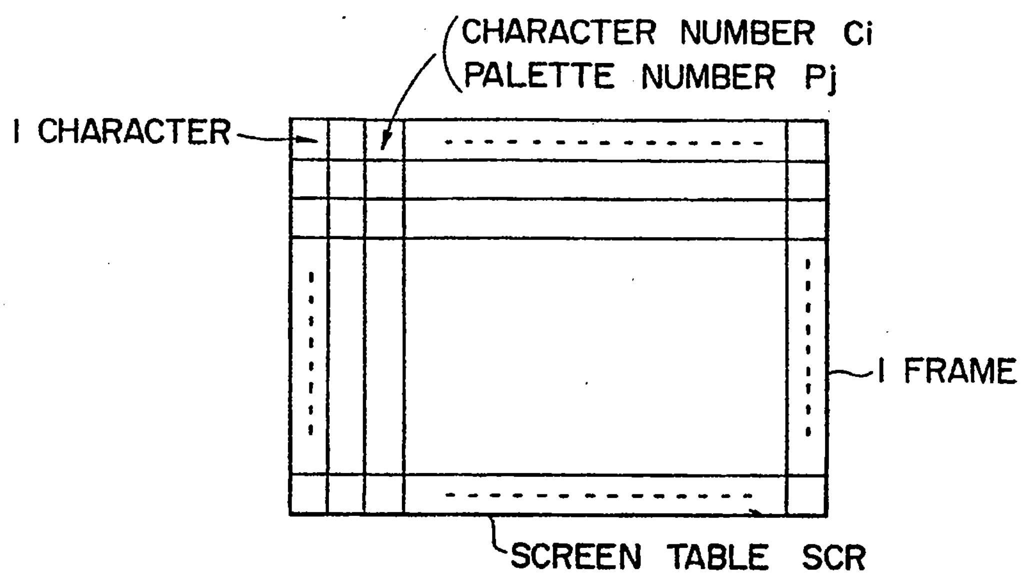


FIG. 9
RELATED ART

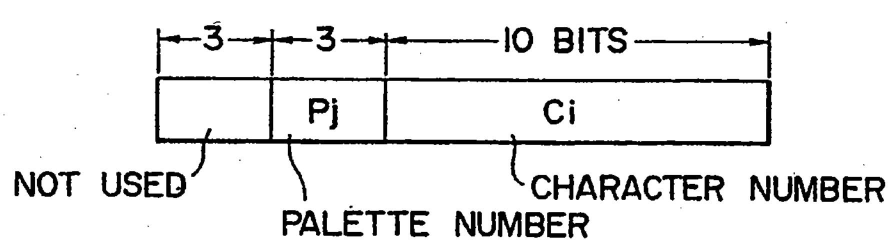
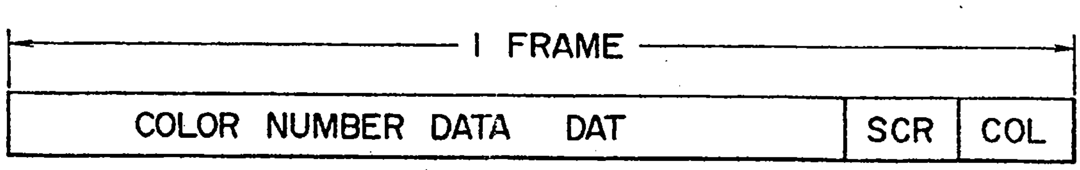


FIG. 10
RELATED ART



F I G. 11

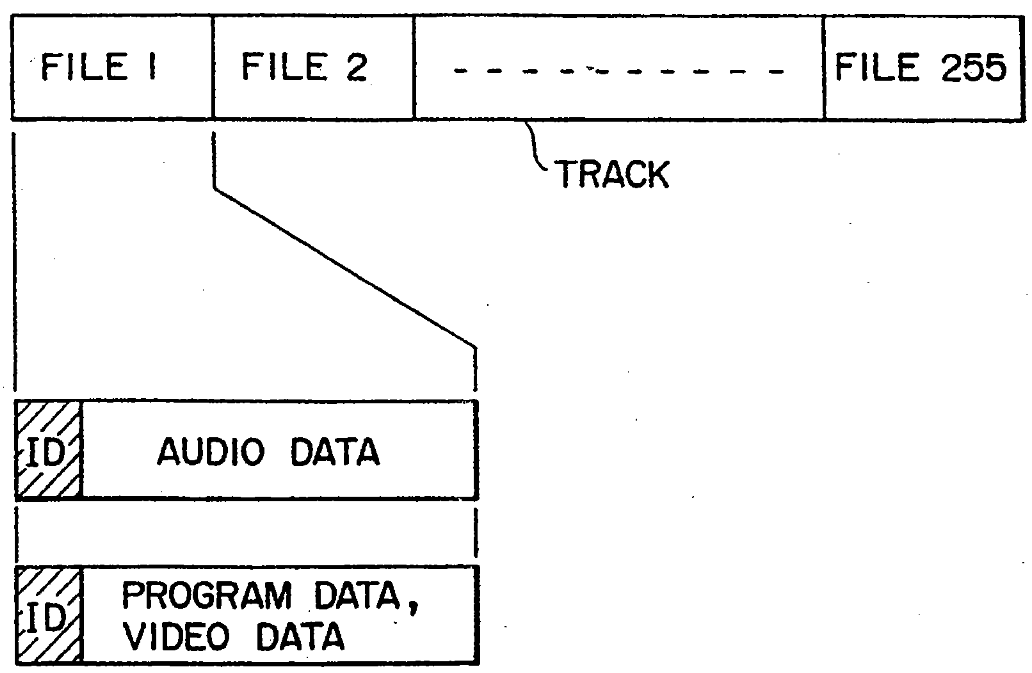
FORMAT	LEVEL AND SOUND SYSTEM \ BLOCK	BLOCK TRAIN →																																				
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	...
a	STEREO	O				O				O				O				O				O				O				O				O				O	
b	MONAURAL	O								O								O								O								O					
c	STEREO	O								O								O								O								O					
d	MONAURAL	O																O																O					
e	STEREO	O																O																O					
f	MONAURAL	O																																O					

1/150 sec.

1/75 sec.

1 BLOCK

F I G. 12(A)



F I G. 12(B)

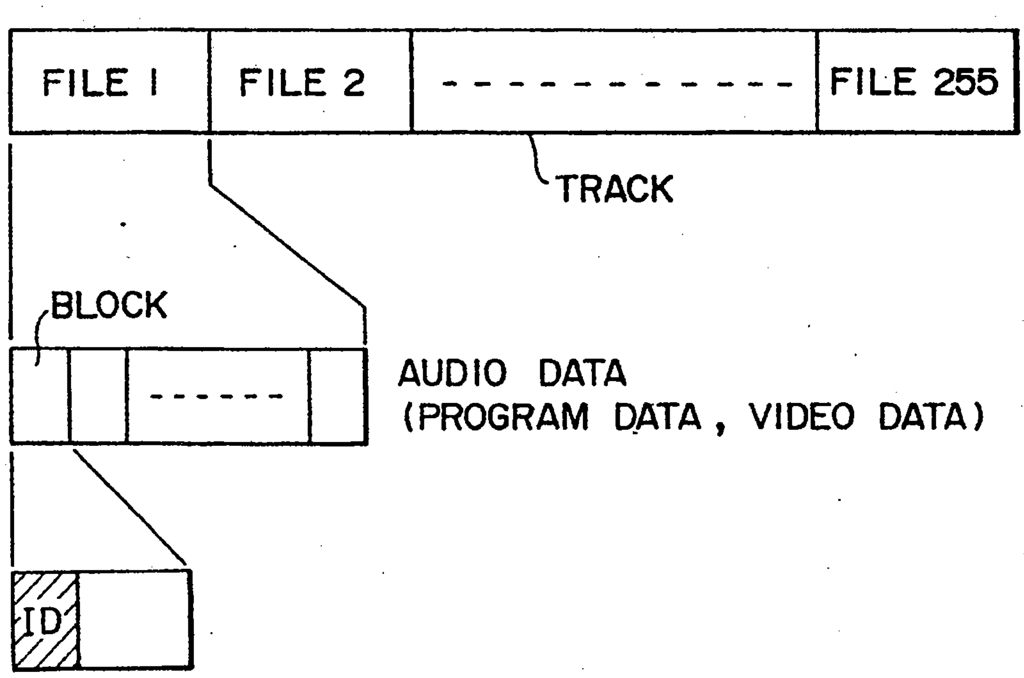


FIG. 13

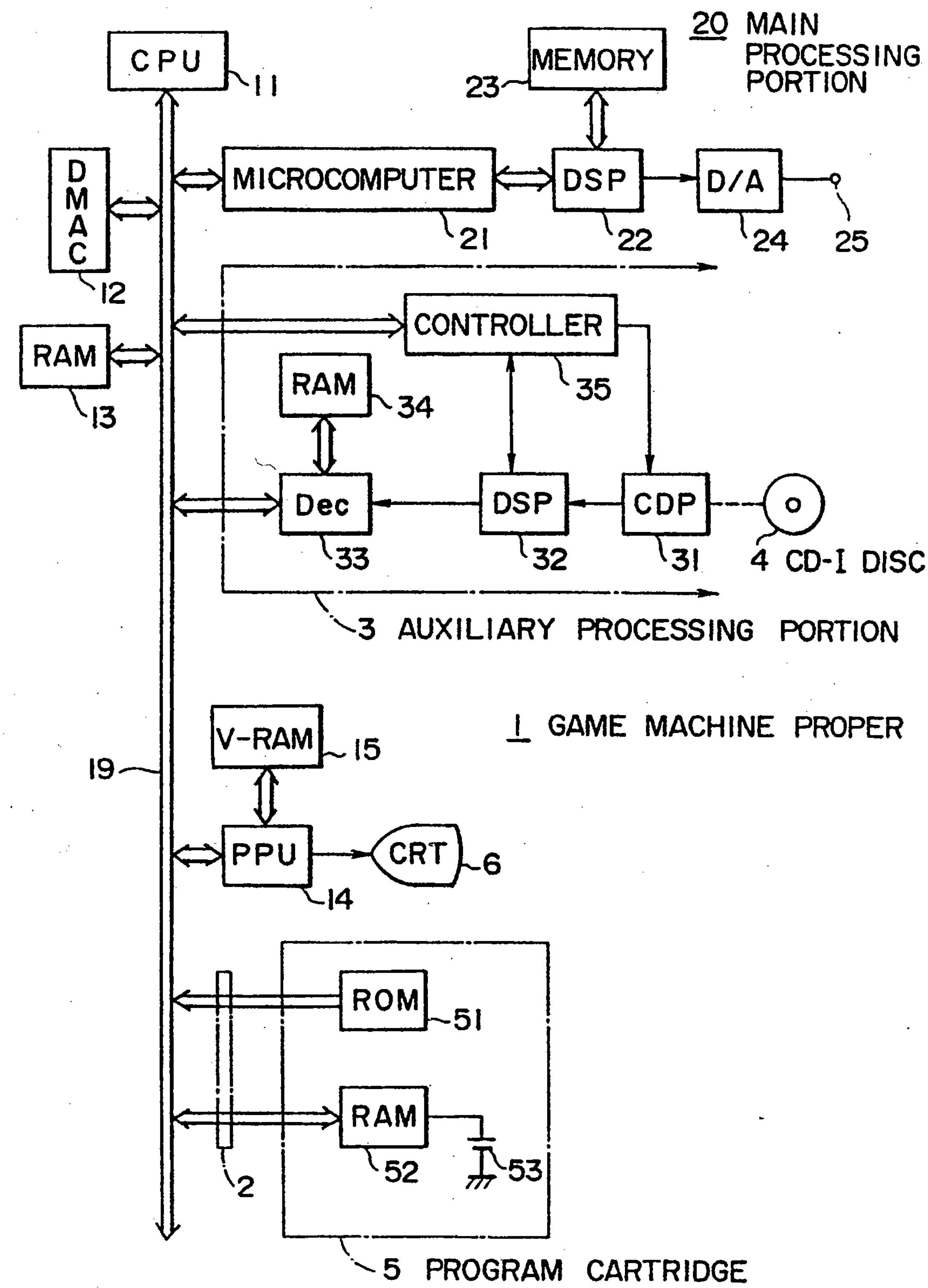
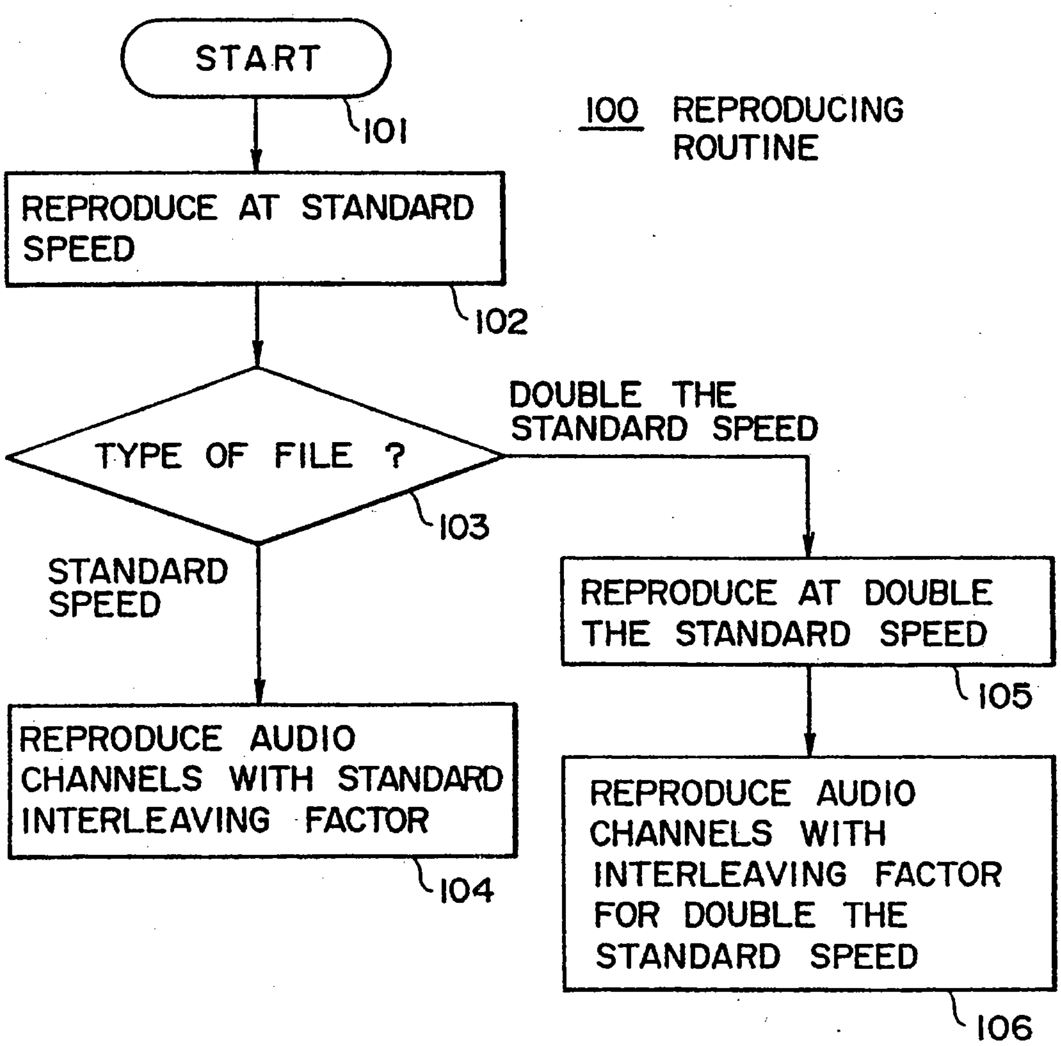

CPU
11
MEMORY
23
20 MAIN PROCESSING PORTION
DMAC
12
MICROCOMPUTER
21
DSP
22
D/A
24
25
RAM
13
CONTROLLER
35
RAM
34
Dec
33
DSP
32
CDP
31
4 CD-I DISC
3 AUXILIARY PROCESSING PORTION
1 GAME MACHINE PROPER
V-RAM
15
19
PPU
14
CRT
6
ROM
51
RAM
52
53
2
5 PROGRAM CARTRIDGE

FIG. 14


START
101
100 REPRODUCING ROUTINE
REPRODUCE AT STANDARD SPEED
102
TYPE OF FILE ?
103
DOUBLE THE STANDARD SPEED
STANDARD SPEED
REPRODUCE AT DOUBLE THE STANDARD SPEED
105
REPRODUCE AUDIO CHANNELS WITH STANDARD INTERLEAVING FACTOR
104
REPRODUCE AUDIO CHANNELS WITH INTERLEAVING FACTOR FOR DOUBLE THE STANDARD SPEED
106

METHOD FOR RECORDING AND REPRODUCING COMPRESSED AUDIO PCM DATA ON AND FROM OPTICAL DISC USING ADJUSTABLE INTERLEAVING FACTOR

This is a division of application Ser. No. 07/976,900, filed Nov. 16, 1992.

BACKGROUND OF THE INVENTION

Compact discs (CD's) on which digital audio signals such as those of music are recorded have gained widespread use. Also introduced are compact disc read-only memories (CD-ROM'S) accommodating not only music data but also image- and computer-related data. Today, there are signal formats extended from the CD-ROM such as CD-ROM-XA (CD-ROM extended architecture) and CD-I (CD-interactive). The CD-ROM-XA format is an extended CD-ROM format that minutely defines how audio and video data are to be recorded. The CD-I format is a format that emphasizes the interactive operation between the medium and users. The audio information complying with the CD-ROM-XA format and the CD-I format is recorded onto discs by adaptive differential pulse code modulation (ADPCM).

FIG. 1 (A) shows a typical data structure of a CD-ROM-XA disc. Illustratively, one track may record up to 255 files of a given size. When audio data are to be recorded, each file may be divided into 16 channels each for use independent of one another, as depicted in FIG. 1 (B). Where image data, text data or computer programs are to be recorded, each file may be divided into 32 channels each for use independent of one another, as indicated in FIG. 1 (C).

FIGS. 2 and 3 show respectively six signal formats and six recording formats in which to record audio data to each file of the CD-ROM-XA disc.

Format a is for stereo recording. With this format, the original stereo audio signal is subjected to A/D conversion at a sampling frequency of 37.8 kHz and with a quantization bit count of 16. The resulting audio data in digital format are compressed by ADPCM into eight bits per sample. As format a in FIG. 3 indicates, the compressed data are recorded on an interleaving basis to every other block in a plurality of blocks (sectors) making up each file.

With format b, the original monaural audio signal is subjected to A/D conversion at a sampling frequency of 37.8 kHz and with a quantization bit count of 16. The resulting audio data in digital format are compressed by ADPCM into eight bits per sample. As format b in FIG. 3 shows, the compressed data are recorded on an interleaving basis to every fourth block in a plurality of blocks constituting each file.

The same workings apply to formats c through f. With any of these formats, the original audio signal is encoded into ADPCM audio data. The resulting audio data are recorded on an interleaving basis in each of the formats c through f.

For normal use, the period per block is 1/75 sec., and the amount of data that may be recorded per block is 2,324 bytes.

Because audio data are recorded on an interleaving basis as depicted in FIG. 3, other data may be recorded to unused blocks. For example, where format a is used to record audio data in interleaving fashion, four-channel stereo recording is available. That is, four languages may be included in a single program. If image data are additionally recorded, the audio signal may be played back accompanied by image reproduction.

Typical image data that may be recorded on CD-ROM-XA discs will now be described with reference to FIGS. 4 through 10.

FIG. 4 is a flowchart of steps in which to record image data in units of frames. FIG. 5 (A) shows a typical image data screen construction for a single frame. The image data for one frame are made of 256 pixels (in traverse direction) by 192 pixels (in longitudinal direction). The red, green and blue for each pixel are represented by five bits each. Each pixel is constituted by 16 bits, i.e., 15 bits (5 bits×3) plus one dummy bit.

In step 81 of FIG. 4, the original image data of one frame shown in FIG. 5 (A) are divided into blocks each measuring 8×8 pixels (each block is called a character), as illustrated in FIG. 5 (B). That is, the image of one frame is split into 768 characters (32 ×24) CHR(0) through CHR(767). The data on one character CHR(i)-(i=0–767) are made of 128 bytes (8 pixels×8 pixels×16 bits).

In step 82, primary vector quantization is performed on the character data CHR(i). The vector quantization executed here involves quantizing the character data so that the number of pixel colors within the character will be limited to a maximum of four. Specifically, in this example, image data are created as follows: A three-dimensional color space is first considered in which the color components of red, green and blue are taken on three coordinate axes that intersect one another orthogonally. In this color space, the distances between the pixels are obtained. Then the pixels whose distances to one another are sufficiently short are grouped together so that the pixel colors within one character are limited to four or fewer representative colors.

The above vector quantization is allowed to continue immediately before a maximum quantization error Emax of each character (i.e., distance between representative color and pixel) is exceeded for each character. The process makes uniform the S/N ratios of all characters (C0 through C767) in each frame. The character data CHR(i) are quantized in this manner, thereby limiting the number of colors in each character to a maximum of four.

In step 83, the quantized character data CHR(i) are categorized into eight groups each comprising characters of like color tones (each group is called a palette). The character data CHR(i) for each character are grouped into one of eight palettes P0 through P7.

In the grouping above, the characters are simply grouped and their sequence is not altered. There is no need for a given palette Pj (j=0 to 7) to be made of contiguous character areas. That is, characters located in a scattered manner may constitute one palette. For example, as shown in FIG. 5 (C), areas A through E of like color tones may each form part of the palette Pj.

In step 84, secondary vector quantization is carried out on the character data CHR(i) grouped into each palette Pj. Even though each character Ci has four or fewer representative colors, the palette Pj comprising characters Ci may have more than 16 colors. If one palette Pj has more than 16 colors, the palette Pj is subjected to secondary vector quantization in the same manner as primary vector quantization so that the number of colors within the palette Pj will be limited to a maximum of 16. The character data CHR(i) belonging to each palette Pj are quantized into the color data (of

15 bits) about one of the 16 representative colors specific to that palette.

In step 85, color number conversion tables COL(j) are created for the palettes Pj. As shown in FIG. 6, the table COL(j) for each pallet Pj is a conversion table that contains the color data (in 15 bits) about the 16 representative colors specific to that palette as well as four-bit color numbers (0–15) for designating the color data. At this point, the character data CHR(i) are equal to one set of character data in the table COL(j) of the palette to which the character data CHR(i) belong.

In step 86, the character data CHR(i) about the respective colors following secondary vector quantization are converted to color numbers in the color number conversion table COL(j) of the palette Pj to which the character CHR(i) belong. The conversion is carried out by reference to the table COL(j). The pixels in each character are represented by two parameters: a four-bit color number designating each color, and data indicating the color number conversion table COL(j) to which the color number belongs.

In step 87, a screen table SCR is created. As illustrated in FIG. 8, the screen table SCR has a total of 768 addresses corresponding to 24×32 characters per frame of the original image data. Each address is two bytes long, as shown in FIG. 9. Of the two bytes, the low-order 10 bits constitute a number Ci indicating a character. The high-order three bits make up a palette number Pj designating the palette Pj to which the character data CHR(i) associated with the character number Ci belong.

In step 88, the character number Ci of the screen table SCR is shifted by 16 in the high-order direction, with character numbers 0 through 15 assigned to the color number of a monochromatic character Ci. For the monochromatic character Ci, the palette number Pj remains the palette number specific to that character but the character number Ci designates the color number. At this point, the address of the monochromatic character Ci in the table SCR indicates the character number Ci.

In the manner described, image data are converted per frame to a color number conversion table COL(j), to a screen table SCR, and to a four-bit color number for each pixel. In the description that follows, the color number conversion table COL(j) will be called the color number conversion table COL, and the four-bit color number for each pixel will be referred to as the color number data PAT. The data PAT, table SCR and table COL may illustratively be combined to form recording data RECD in the format shown in FIG. 10. The recording data are recorded on the CD-ROM-XA disc after undergoing a predetermined encoding process.

The amount of image data RECD recorded per frame is calculated as follows: Eight palettes Pj exist per frame. One palette has 16 colors, each color being represented by 16 bits (of which 1 bit is dummy). Thus the color number conversion table COL is made of a total of 256 bytes (=8 palettes×16 colors×16 bits). Since there are 768 characters and each character corresponds to two bytes, the screen table SCR amounts to 1,536 bytes (=768 characters×2 bytes). With the color number data DAT comprising four bits per pixel, the amount of data DAT per character is 32 bytes (4 bits ×64 pixels). Because the data about a monochromatic character Ci are already transmitted as part of the screen table SCR, it is not necessary to transmit the data anew. Thus if a quarter of the 768 characters per frame are monochromatic characters and the remaining three fourths are multi-colored characters, the total data amount is obtained by adding 256 bytes of the color number conversion table COL, 1,536 bytes of the screen table SCR, and 18,432 bytes of the color number data DAT (=32 bytes×768 characters×$\frac{3}{4}$). The sum is 20,224 bytes.

Where image data RECD are to be recorded on the CD-ROM-XA disc, all 32 channels may be used to record the data, as shown in FIG. 1 (C). At this time, the transmission rate of the CD-ROM-XA disc for the image data RECD is 174,300 bytes (=2,324 bytes×75 blocks) per second. In this example, the frame count per second is about 8.6 (=174,300 bytes divided by 20,224 bytes). Animated images are recorded and reproduced with this frame count.

As illustrated in FIG. 1, each file on the CD-ROM-XA disc may record audio data and image data mixedly in units of blocks. Where a computer game is executed from recordings on a CD-ROM-XA disc, background music (BGM) and sound effects may be reproduced concurrently to enhance the entertainment value of the game. Illustratively, format e in FIG. 2 or 3 is utilized so that one out of eight blocks will address audio data and the remaining seven will address image data RECD. In this manner, both music and animated images are reproduced at the same time.

Because seven out of eight blocks are used to record and reproduce image data RECD in the above example, the frame count per second for the image data RECD is about 7.5 (8.6 frames×$\frac{7}{8}$). That frame count is insufficient for the adequate display of animated images. This drawback is circumvented illustratively by raising the revolutions of the disc. Rotating at higher speeds, the disc increases the number of image frames per second for animation. Illustratively, doubling the standard revolutions of the disc doubles the rate of data reproduction therefrom. That is, with the standard disc rotating speed doubled, the frame count is about 15 (=7.5 frame×2) per second, which is sufficient for animated image reproduction. Descriptions of how data are recorded and reproduced to and from a disc at a rotating speed higher than usual are found illustratively in a pending international application, "Data Recording and Reproducing Method" (by Yoichiro Sako, Serial No. PCT/JP91/00054, filed Jan. 18, 1991).

As described, concurrent recording of audio data and animated image data on a disc should preferably be done by raising the disc revolutions (e.g., twice as fast as standard rotating speed). This scheme involves one disadvantage. That is, to double the disc revolutions requires lowering in half the frequency of the original audio signal as the signal is subjected to ADPCM for audio data preparation. Optimum encoding of the audio signal is disabled at this point, with the result that the sound quality of the reproduced audio signal deteriorates.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide an optical disc and a method for successfully reproducing audio signals in data compressed format from that disc rotating at a speed higher than usual.

In carrying out the invention and according to one aspect thereof, there is provided an optical disc on which an audio signal is recorded in the form of compressed data, the optical disc comprising: an identification area for identifying the data transfer rate of the optical disc; and an audio recording area in which compressed data are interleaved and arranged in accordance with the data transfer rate.

According to another aspect of the invention, there is provided a method for recording an audio signal on a disc in the form of compressed data, the method comprising the steps of: identifying the data transfer rate of compressed audio data; arranging the compressed audio data after interleaving thereof on the disc in accordance with the data transfer rate; and arranging identification data on the disc for identifying the data transfer rate of the recorded audio signal for use in data reproduction.

According to a further aspect of the invention, there is provided a method for reproducing a compressed audio signal from a disc, the method comprising the steps of: identifying the data transfer rate of a recorded audio signal; rotating the disc in accordance with the identified data transfer rate; and reproducing the compressed audio signal from the disc, the compressed audio signal being interleaved and arranged on the disc in accordance with the data transfer rate.

The invention establishes the interleaving factor for compressed audio data in accordance with the rotating speed of the disc which is greater than that of CD-ROM discs commonly used. For example, where the rotating speed of the disc is twice as high as the standard rotating speed of discs, the interleaving factor for compressed audio data is doubled for recording onto the disc.

The above and other related objects and features of the invention, as well as the novelty thereof, will clearly appear from the following description and from the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A–1C is a set of views depicting a typical track format of a disc;

FIG. 2 is a view listing typical signal formats of a disc;

FIG. 3 is a view listing typical recording formats of a disc;

FIG. 4 is a flowchart of key steps in which to record and process image data;

FIG. 5A–5C is a set of views illustrating a typical image data structure;

FIG. 6 is a view showing color number conversion tables;

FIG. 7 is a view depicting a typical pixel structure;

FIG. 8 is a view portraying how a typical screen table is constructed;

FIG. 9 is a view explaining how screen table data are typically structured;

FIG. 10 is a view showing a typical format of recorded data;

FIG. 11 is a view listing typical recording formats according to the invention;

FIG. 12A–12B is a set of views indicating the locations of ID codes that designate the data transfer rate of a disc according to the invention;

FIG. 13 is a schematic diagram of a disc reproducing circuit according to the invention; and

FIG. 14 is a flowchart of key steps in a disc reproducing routine according to the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 11 lists typical recording formats according to the invention. It is assumed that a CD-ROM-XA disc, used in the example below, rotates at a speed twice as high as the standard rotating speed of discs for recording and reproducing. In FIG. 11, formats a through f correspond to the formats a through f of FIGS. 2 and 3. That is, format a of the invention is also for stereo recording. With this format, the original stereo audio signal is subjected to A/D conversion at a sampling frequency of 37.8 kHz and with a quantization bit count of 16. The resulting audio data in digital format are compressed by ADPCM into eight bits per sample. The compressed audio data are recorded on an interleaving basis to every fourth block.

Format b is for monaural recording. With format b, the original monaural audio signal is subjected to A/D conversion at a sampling frequency of 37.8 kHz and with a quantization bit count of 16. The resulting audio data in digital format are compressed by ADPCM into eight bits per sample. As format b in FIG. 3 shows, the compressed audio data are recorded on an interleaving basis to every eighth block.

The same workings apply to formats c through f. The audio signal is encoded into ADPCM audio data in each of these signal formats shown in FIG. 2. The resulting audio data are recorded on an interleaving basis in each of the recording formats c through f of FIG. 11. That is, the block interleaving factor for audio data according to the invention is twice that of CD-ROM-XA discs that rotate at the standard speed.

When the rotating speed of the CD-ROM-XA disc is twice the standard rotating speed, the block period is 1/150 sec. As shown in FIG. 12 (A), the CD-ROM-XA disc with its rotating speed doubled has each of its files prefixed with an identification code indicating the disc rotating speed or data transfer rate. Alternatively, as depicted in FIG. 12 (B), each block is prefixed with an identification code indicating whether the rotating speed is standard or twice as high.

According to this recording format, the original audio signal to be recorded on a CD-ROM-XA disc rotating at double the standard rotating speed is encoded into audio data through the same ADPCM process as with a CD-ROM-XA disc rotating at the standard rotating speed. The audio data are recorded to each of the blocks. As the rotating speed or data transfer rate of the disc is enhanced, the interleaving factor for the compressed audio data is raised accordingly. This eliminates the need for specialized circuits for lowering the frequency of the audio signal in inverse proportion to the disc rotating speed.

Referring to FIG. 3, when seven blocks out of the eight are to be used to record image data RECD, the formats of the invention in FIG. 11 allow 15 blocks out of the sixteen to record the image data RECD. Because the number of animated image frames is increased in this manner, the resulting animation is seen moving naturally.

FIG. 13 is a schematic diagram of a disc reproducing circuit according to the invention, the circuit being incorporated in a computer game machine operating off a CD-ROM-XA disc.

In FIG. 13, the game machine proper 1 is based on a microcomputer system, wherein reference numeral 11 is a CPU; 12 is a DMAC (DMA controller); 13 is a RAM for work area and buffer use; 14 is a PPU (picture processing unit); and 15 is a video RAM.

The circuits 11 through 14 are connected to a system bus 19. The PPU 14 is connected to the video RAM 15 and a CRT display 6. The video RAM 15 has screen areas corresponding to at least two frames (2 screens). The image data of one such screen area are read out by the PPU 14 in synchronism with the vertical and horizontal scanning of the CRT display 6, the read-out image data being displayed on the CRT display 6. While the display is on, the image data of the image to be displayed next are written to another screen area. Part of the areas of the video RAM 15 is used as a work area for use by the PPU 14; part of them is also used as a palette area for use by the color number conversion table COL.

In the game machine proper 1, reference numeral 20 is a main processing portion that addresses audio data; 21 is a microcomputer for control of the main processing portion 20; 22 is a DSP (digital signal processor) dedicated to audio data processing; 23 is a cache memory for use with the DSP 22; 24 is a D/A converter; and 25 is an audio output terminal. The DSP 22 is connected to the system bus 19 via the microcomputer 21 and also to the D/A converter 24. The DSP 22 is further connected to the cache memory 23. An auxiliary processing portion 3 comprises a CD player 31 that permits the use of a CD-ROM-XA disc 4. Reference numeral 32 is a DSP; 33 is a CD-ROM decoder; 34 is a work area RAM for use by the CD-ROM decoder 33; and 35 is a controller. As described, the CD-ROM-XA disc 4 contains thereon audio data, image data RECD, programs for reproducing and processing the image data RECD, game programs, and the operating system (OS). The DSP 32 corrects errors in the reproduced signal from the CD player 31. At the same time, the DSP 32 separates from the reproduced signal user-oriented data such as image data RECD and control data such as track numbers. Based on the control data from the DSP 32 and on the data from the CPU 11, the controller 35 controls the CD player 31 so as to retrieve target data from the disc 4. Depending on whether the reproduced signal from the CD player 31 is a signal from the CD-ROM-XA disc 4 or from a CD-ROM disc, the decoder 33 performs error correction and other processing regarding the applicable disc.

In the example of FIG. 13, the auxiliary processing portion 3 includes a general-purpose DSP 44 that shoulders part of the processing burdens of the CPU 11. Although the auxiliary processing portion 3 is integrally attached to the game machine proper 1, the portion 3 may alternatively be provided as an adapter that is removably mounted on the game machine 1.

A program cartridge 5 contains a ROM 51 and a RAM 52. The ROM 51 contains programs and other data. The RAM 52, backed up by a battery 53, retains the data about a game interrupted halfway through so that the game will be resumed later where it was left off. When the program cartridge 5 is inserted into a slot 2 of the game machine proper 1, the ROM 51 and RAM 52 are connected to the bus 19 via connectors, not shown. If the program cartridge 5 alone is sufficient for executing a game or like computer-based program, the target game is started by setting the cartridge 5 in the slot 2 and by turning on or resetting power to the game machine 1.

Upon power-up, the CPU 11 executes the program in the ROM 51 to generate image data. The PPU 14 writes the generated image data to the video RAM 15. The image data in the video RAM are read out therefrom in synchronism with the horizontal and vertical scanning for graphic display onto the CRT display 6.

The CPU 11 loads audio data and a program for decoding the data into the microcomputer 21. In turn, the microcomputer 21 decodes the audio data into a digital audio signal which is converted from digital to analog format by the converter 24. The converted analog audio signal is output from the terminal 25.

Where the CD-ROM-XA disc 4 is used to execute a game or like computer-based program, the disc 4 is first set to the CD player 31 of the auxiliary processing portion 3. Then a system cartridge 5 (i.e., program cartridge) is set in the slot 2. The system cartridge 5 also has the ROM 51 and RAM 52. The ROM 51 contains an IPL (initial program loader). The RAM 52, as described, retains the data about a game interrupted halfway through so that the game will be resumed later where it was left off. The CD-ROM-XA disc 4 includes the OS (operating system) under which the disc 4 is run as well as the game program.

For operation, applying power causes the CPU 11 to execute the IPL in the ROM 51. This in turn causes the OS and game program to be loaded from the CD-ROM-XA disc 4 to the RAM 13. Then the CPU 11 executes the program loaded in the RAM 13, allowing the user to play the game.

FIG. 14 is a flowchart of key steps in a disc reproducing routine 100 for use with the CD-ROM-XA disc 4. The routine 100 is executed illustratively when the game program is loaded from the CD-ROM-XA disc 4 by the CPU 11 executing the IPL in the ROM 51.

With the IPL executed by the CPU 11, the processing of the CPU 11 goes from step 101 to step 102 of the routine 100. In step 102, the CD-ROM-XA disc 4 is set temporarily in standard rotating speed reproduction mode. In step 103, the identification code at the beginning of the file to be reproduced is checked to see what the code indicates: whether the file to be reproduced is to be read at the standard rotating speed, or at a higher rotating speed.

If the check in step 103 has revealed that the identification code indicates a standard rotating speed file, step 103 is followed by step 104. In step 104, the CD-ROM-XA disc 4 is reproduced at the standard rotating speed.

If the check in step 103 has revealed that the identification code indicates a file readable at a higher speed (e.g., double the standard rotating speed), step 103 is followed by step 105. In step 105, the CD-ROM-XA disc 4 is rotated illustratively at double the standard rotating speed. In step 106, the interleaving process for reproducing the audio data is carried out with double the interleaving factor according to FIG. 11.

Where one file mixedly contains both blocks for the standard rotating speed and blocks for double the standard rotating speed, each of the blocks is checked for its identification code. This allows each block to be identified with its applicable rotating speed and interleaving factor regarding the CD-ROM-XA disc 4.

When image data RECD of the CD-ROM-XA disc 4 are to be displayed as animated images, the image data RECD are reproduced and processed as follows:

(1) The CD player 31 reproduces one frame of image data RECD from the CD-ROM-XA disc 4. The image data RECD are fed from the CD player 31 to the DSP 32 and on to the decoder 33 for error correction and other processing. After error correction, the image data RECD are transferred through DMA by the DMAC 12 from the decoder 33 to the video buffer area of the RAM 13.

(2) Of the image data RECD transferred through DMA to the buffer area of the RAM 13, the color number data DAT are sent by the DMAC 12 through DMA transfer to the video RAM 15 via the PPU 14 during a vertical blanking period of the CRT display 6

(3) After the process (2) above, of the image data RECD sent through DMA transfer to the buffer area of the RAM 13, the screen table SCR is transferred through DMA by the DMAC 12 to the video RAM 15 via the PPU 14 during a vertical blanking period of the CRT display 6.

(4) Of the image data RECD sent through DMA transfer to the buffer area of the RAM 13, the color number conversion table COL is transferred by the DMAC 12 through DMA to the video RAM 15 via the PPU 14 during a horizontal blanking period of the CRT display 6.

(5) After the above processes, the PPU 14 converts the color number data PAT transferred in (2) above into pixel data of actual color by referencing the color number conversion table COL(j) in real time. At the same time, by referencing the screen table SCR, the PPU 14 writes the pixel data to the addresses corresponding to the original character positions in the video RAM 15.

(6) After one frame of pixel data is written to the video RAM 15 in the manner above, display areas of the video RAM 15 are switched. The area filled with the image data is rendered active, and the image in the active area is displayed on the CRT display 6.

(7) The process (1) above is reached again, and the processes (1) through (6) are repeated for each frame.

Thus the image data reproduced from the CD-ROM-XA disc 4 are sent consecutively from the RAM 13 through the PPU 14 to the video RAM 15. The PPU 14 retrieves and suitably processes the image data. As a result, the animated images based on the image data RECD from the CD-ROM-XA disc 4 are displayed on the CRT display 6.

When audio data from the CD-ROM-XA disc 4 are to be output as an audio signal to the terminal 25, the audio data are reproduced and processed as follows:

(1) The CD player 31 reproduces one block of audio data from the CD-ROM-XA disc 4. The audio data are supplied from the CD player 31 to the DSP 32 and on to the decoder 33 for error correction and other processing. After error correction, the audio data are sent by the DMAC 12 from the decoder 33 to the audio buffer area of the RAM 13 through DMA transfer.

(2) The audio data transferred through DMA to the buffer area of the RAM 13 are sent further to the memory 23 via the microcomputer 21 and DSP 22.

(3) With the process (2) above completed, the DSP 22 decodes the audio data in the memory 23 from ADPCM format back to the original audio data.

(4) The audio data decoded in the process (3) above are fed to the D/A converter 24. This allows the original audio signal to be output to the terminal 25.

(5) The process (1) is reached again. Thereafter, the processes(1) through (4) are repeated for each block of audio data.

The audio data thus reproduced from the CD-ROM-XA disc 4 are sent as continuous audio signals to the terminal 25.

For the example of CD-ROM-XA formats above, the rotating speed of the CD-ROM-XA disc is twice the standard rotating speed. Alternatively, the disc rotating speed or data transfer rate may be N (N>1) times the standard speed or standard rate. In the latter case, the standard interleaving factor for recording audio data on an interleaving basis need only be multiplied by a factor of N.

In the above example, the identification code indicating the rotating speed of the CD-ROM-XA disc 4 is attached to each file or to each block, as shown in FIG. 12. Alternatively, the identification code may be attached to the TOC (table of contents) area of the disc.

As many apparently different embodiments of this invention may be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.

What is claimed is:

1. A method of recording audio data on a disc, said method comprising the steps of:

rotating said disc at a predetermined speed to establish a predetermined data transfer rate;

compressing said audio data;

identifying said data transfer rate of said compressed audio data;

interleaving said compressed audio data to form interleaved data and recording said interleaved data on said disc as a recorded audio signal using an interleaving factor that increases with an increase in said data transfer rate; and

arranging identification data on said disc for identifying the data transfer rate of the recorded audio signal for use in data reproduction.

2. A method as claimed in claim 1 comprising the step of employing adaptive differential pulse code modulation to compress the audio data.

3. A method as claimed in claim 1 comprising the step of increasing said interleaving factor in direct proportion to an increase in said data transfer rate.

4. A method as claimed in claim 1 comprising the step of doubling said interleaving factor in response to a doubling of said data transfer rate.

5. A method of reproducing a compressed audio signal from a disc, said method comprising the steps of:

identifying a data transfer rate of an audio signal recorded on a disc;

rotating said disc at a speed that is directly proportional to the identified data transfer rate; and

reproducing said compressed audio signal from said disc, said compressed audio signal being interleaved and arranged on said disc with an interleaving factor that is directly proportional to said data transfer rate.

6. A method of reproducing a compressed audio signal as claimed in claim 5 comprising the step of employing adaptive differential pulse code modulation decompression to reproduce the audio signal.

* * * * *

US005404437A

United States Patent [19]

Nguyen

[11] Patent Number: 5,404,437

[45] Date of Patent: Apr. 4, 1995

[54] MIXING OF COMPUTER GRAPHICS AND ANIMATION SEQUENCES

[75] Inventor: Julien T. Nguyen, Redwood City, Calif.

[73] Assignee: Sigma Designs, Inc., Fremont, Calif.

[21] Appl. No.: 975,910

[22] Filed: Nov. 10, 1992

[51] Int. Cl.^{6} G06F 15/62

[52] U.S. Cl. 395/152; 395/153; 345/122

[58] Field of Search 395/152, 153, 154; 340/725, 734; 345/115, 116, 122, 196; 348/584, 589, 598, 600

[56] References Cited

U.S. PATENT DOCUMENTS

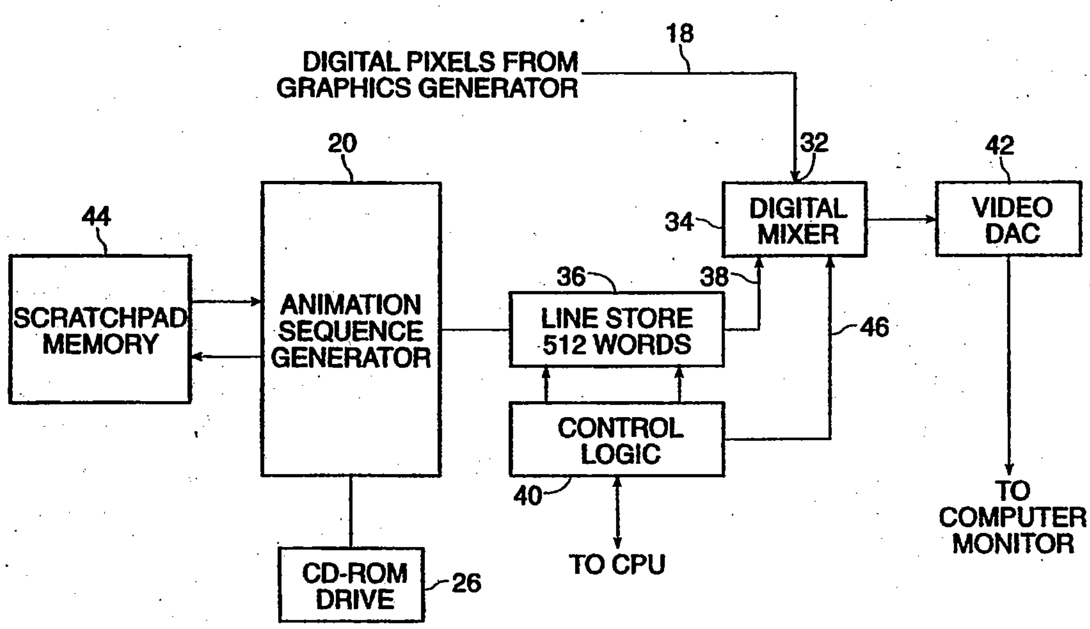
3,908,080	9/1975	Broadbent	358/343
4,498,098	2/1985	Stell	358/22
4,523,227	6/1985	Hust	358/105
4,580,165	4/1986	Patton et al.	358/148
4,626,837	12/1986	Priestly	348/584
4,680,622	7/1987	Barnes et al.	358/22
4,891,631	1/1990	Fredlund et al.	340/701
4,907,086	3/1990	Truong	358/183
4,951,229	8/1990	DiNicola et al.	340/725
4,994,912	2/1991	Lumelsky et al.	358/140
5,097,257	3/1992	Clough et al.	345/196
5,099,331	3/1992	Truong	340/734
5,138,307	8/1992	Tatsumi	340/734
5,185,665	2/1993	Okura et al.	348/589
5,218,431	6/1993	Gleicher et al.	358/13
5,243,447	9/1993	Bodenkamp et al.	345/116

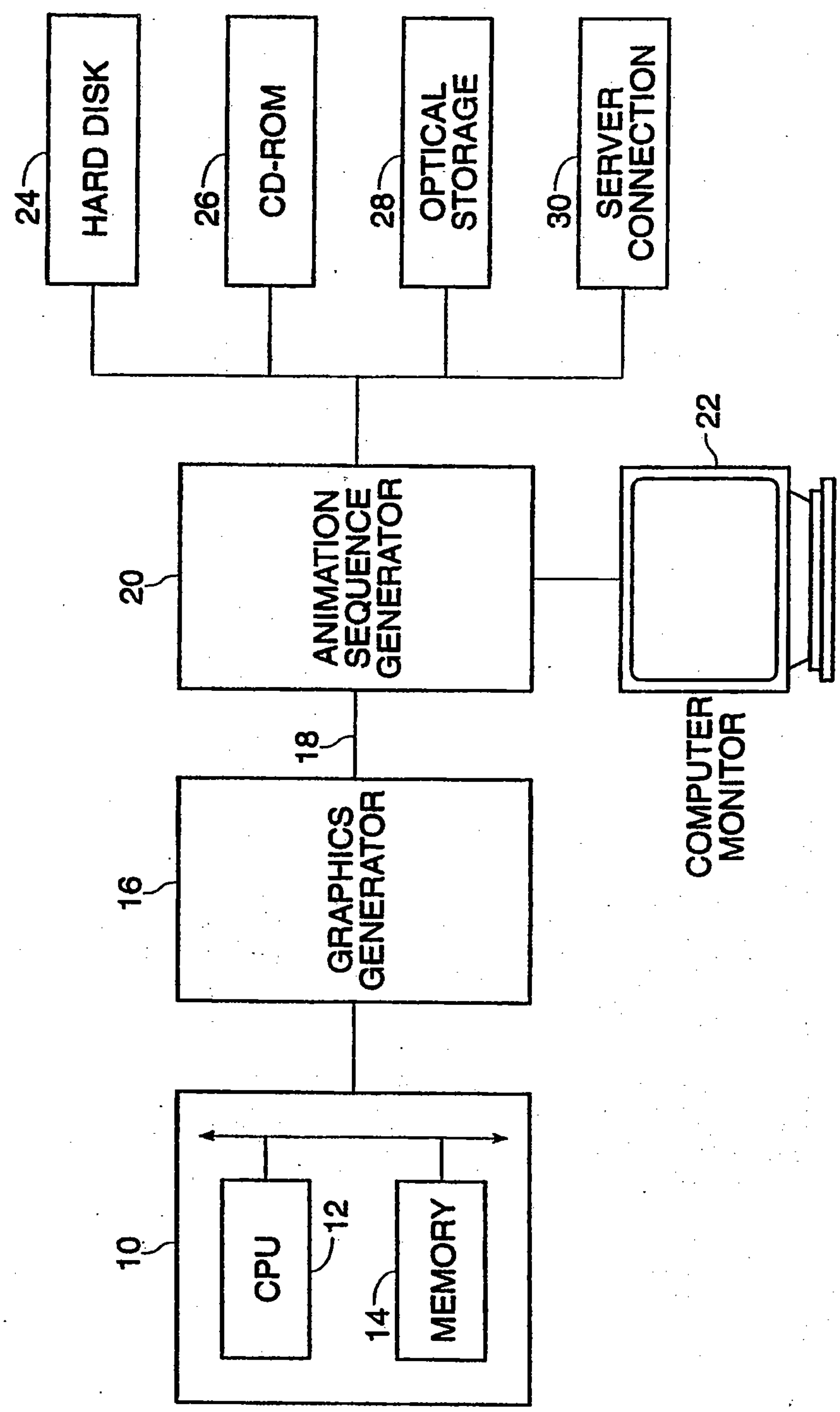
Primary Examiner—Heather R. Herndon
Assistant Examiner—N. Kenneth Burraston
Attorney, Agent, or Firm—D'Alessandro, Frazzini & Ritchie

[57] ABSTRACT

Apparatus and a method is disclosed for mixing animation sequences with computer graphics information for presentation on a computer display screen. The animation sequences may be stored in compressed format in a standard storage medium. An animation sequence generator retrieves these compressed animation sequences, decompresses them into pixel information, and sends them to a single line store for synchronization with the computer graphics pixel information. Pixels from the animation sequences and from the computer graphics generator are mixed, windowed and overlaid in a digital mixer. The output of the digital mixer is sent to a visual display system such as a video digital-to-analog converter driving a computer display monitor.

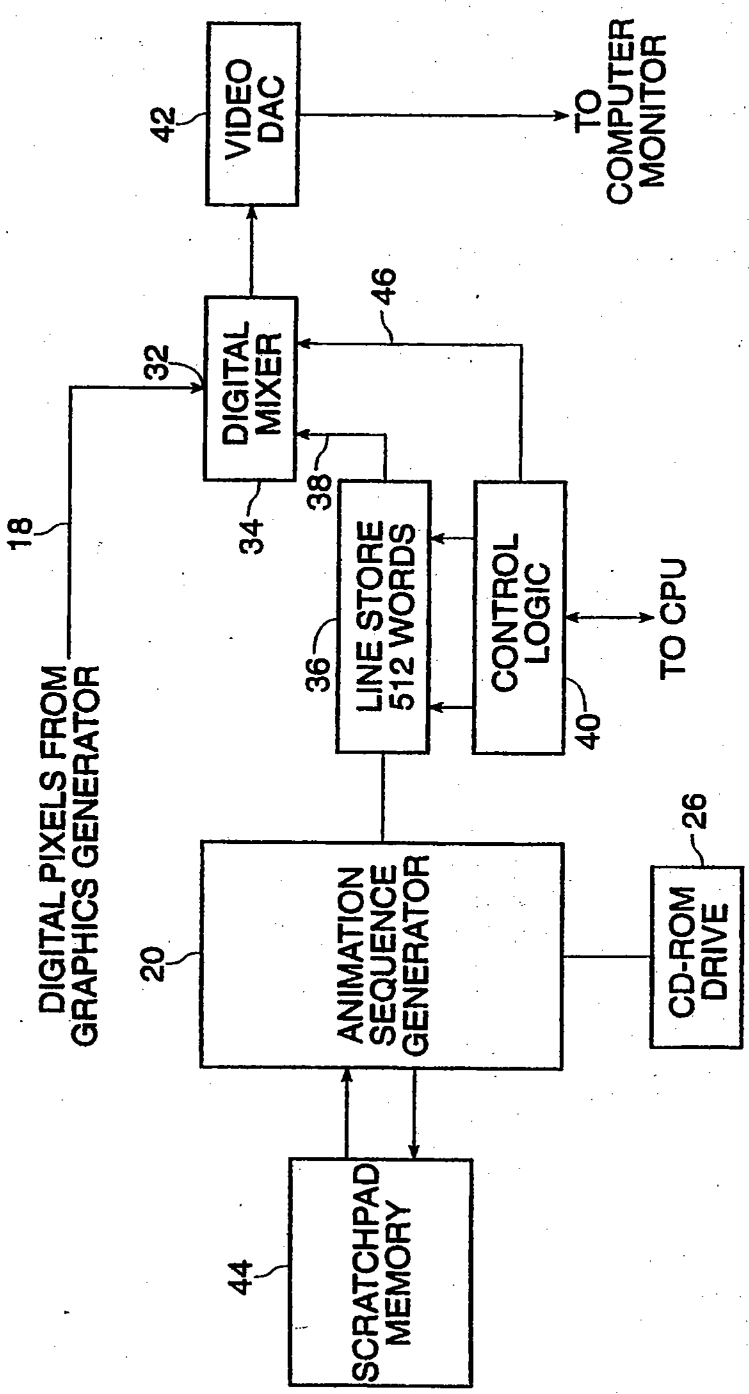
18 Claims, 9 Drawing Sheets





24
HARD DISK
26
CD-ROM
28
OPTICAL STORAGE
30
SERVER CONNECTION
20
ANIMATION SEQUENCE GENERATOR
22
COMPUTER MONITOR
18
16
GRAPHICS GENERATOR
10
CPU
12
14
MEMORY

FIG. 1



DIGITAL PIXELS FROM GRAPHICS GENERATOR
18
32
DIGITAL MIXER
34
VIDEO DAC
42
TO COMPUTER MONITOR
46
38
LINE STORE 512 WORDS
36
CONTROL LOGIC
40
TO CPU
ANIMATION SEQUENCE GENERATOR
20
CD-ROM DRIVE
26
SCRATCHPAD MEMORY
44

FIG. 2

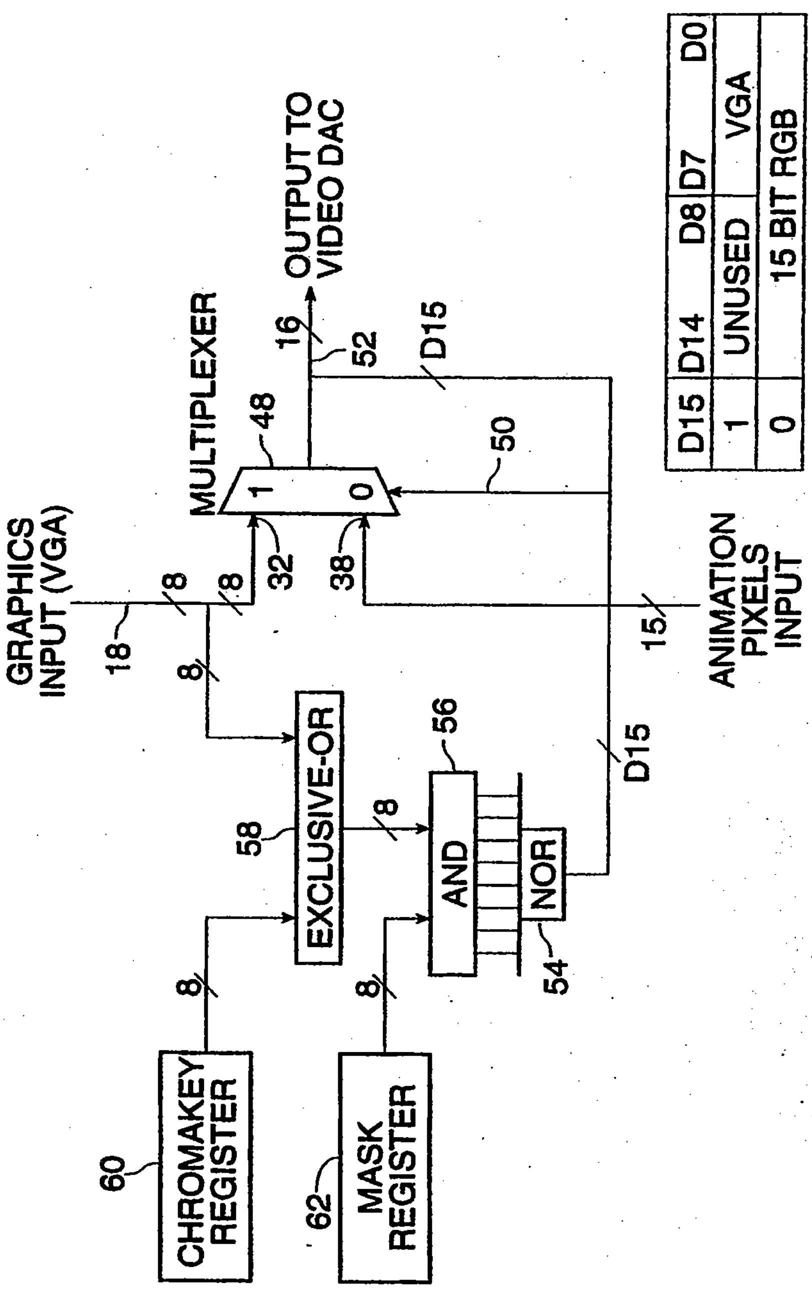
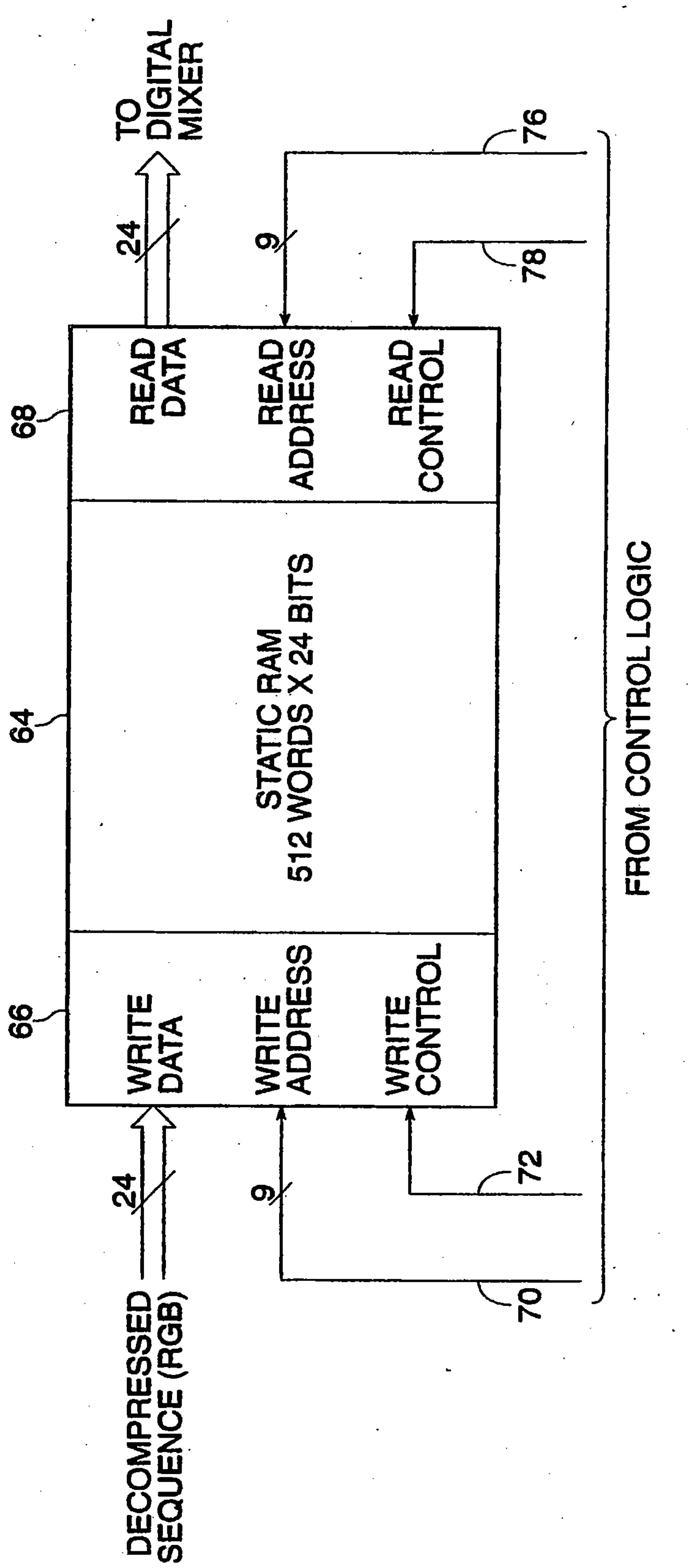

GRAPHICS INPUT (VGA)
18
8
8
8
32
38
MULTIPLEXER
48
1
0
16
52
OUTPUT TO VIDEO DAC
D15
50
15
ANIMATION PIXELS INPUT
EXCLUSIVE-OR
58
56
8
AND
NOR
54
D15
8
8
CHROMAKEY REGISTER
60
MASK REGISTER
62
D15 | D14 D8 | D7 D0
1 | UNUSED | VGA
0 | 15 BIT RGB

FIG. 3



TO DIGITAL MIXER
24
9
76
78
68
READ DATA
READ ADDRESS
READ CONTROL
64
STATIC RAM
512 WORDS X 24 BITS
66
WRITE DATA
WRITE ADDRESS
WRITE CONTROL
24
9
72
70
DECOMPRESSED SEQUENCE (RGB)
FROM CONTROL LOGIC

FIG. 4

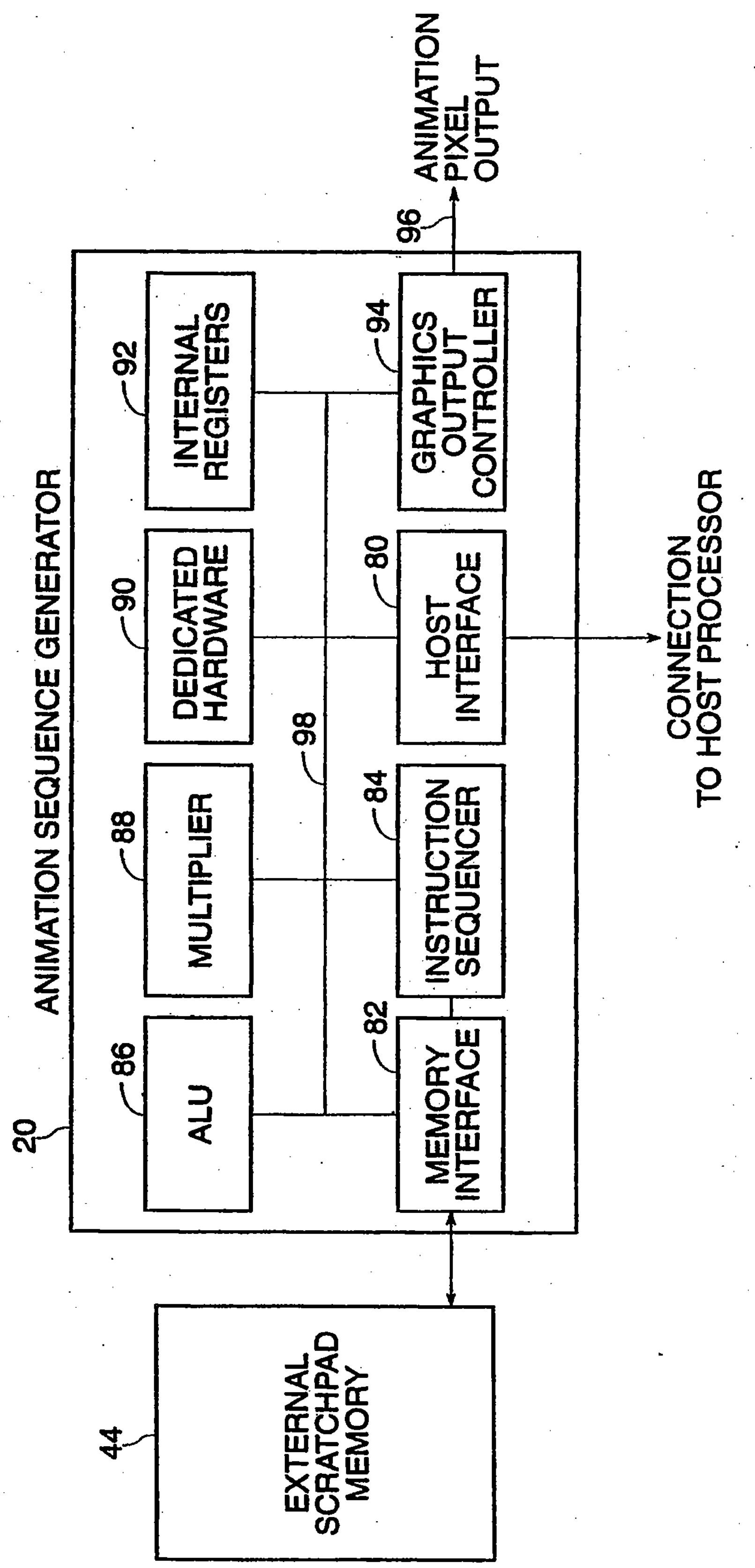

ANIMATION SEQUENCE GENERATOR
20
ALU
86
MULTIPLIER
88
DEDICATED HARDWARE
90
INTERNAL REGISTERS
92
MEMORY INTERFACE
82
INSTRUCTION SEQUENCER
84
HOST INTERFACE
80
GRAPHICS OUTPUT CONTROLLER
94
98
96
ANIMATION PIXEL OUTPUT
CONNECTION TO HOST PROCESSOR
EXTERNAL SCRATCHPAD MEMORY
44

FIG. 5

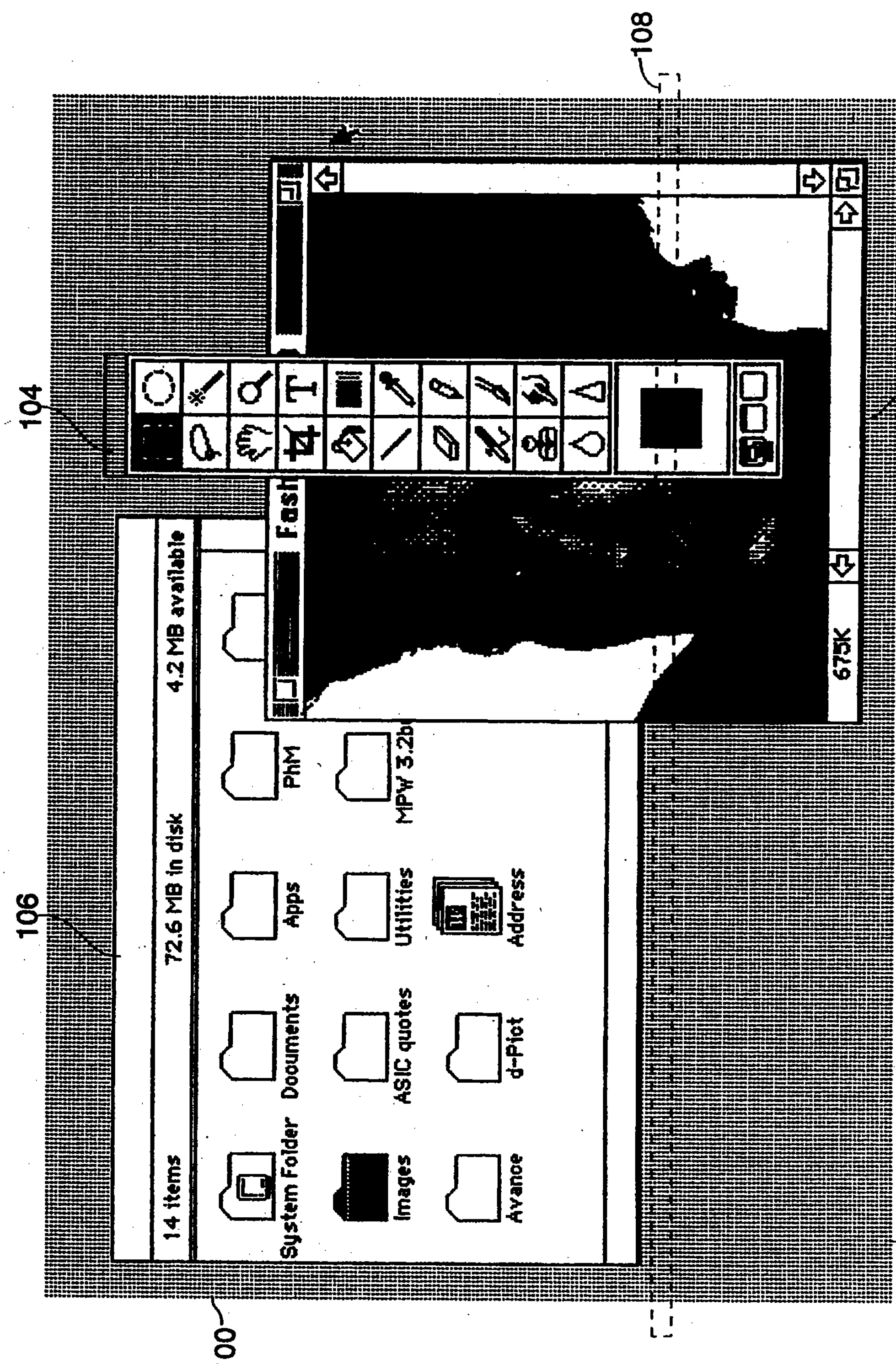

100
102
104
106
108
14 items
72.6 MB in disk
4.2 MB available
System Folder
Documents
Apps
PhM
Images
ASIC quotes
Utilities
MPW 3.2b
Avanoe
d-Pict
Address
Fash
675K

FIG. 6

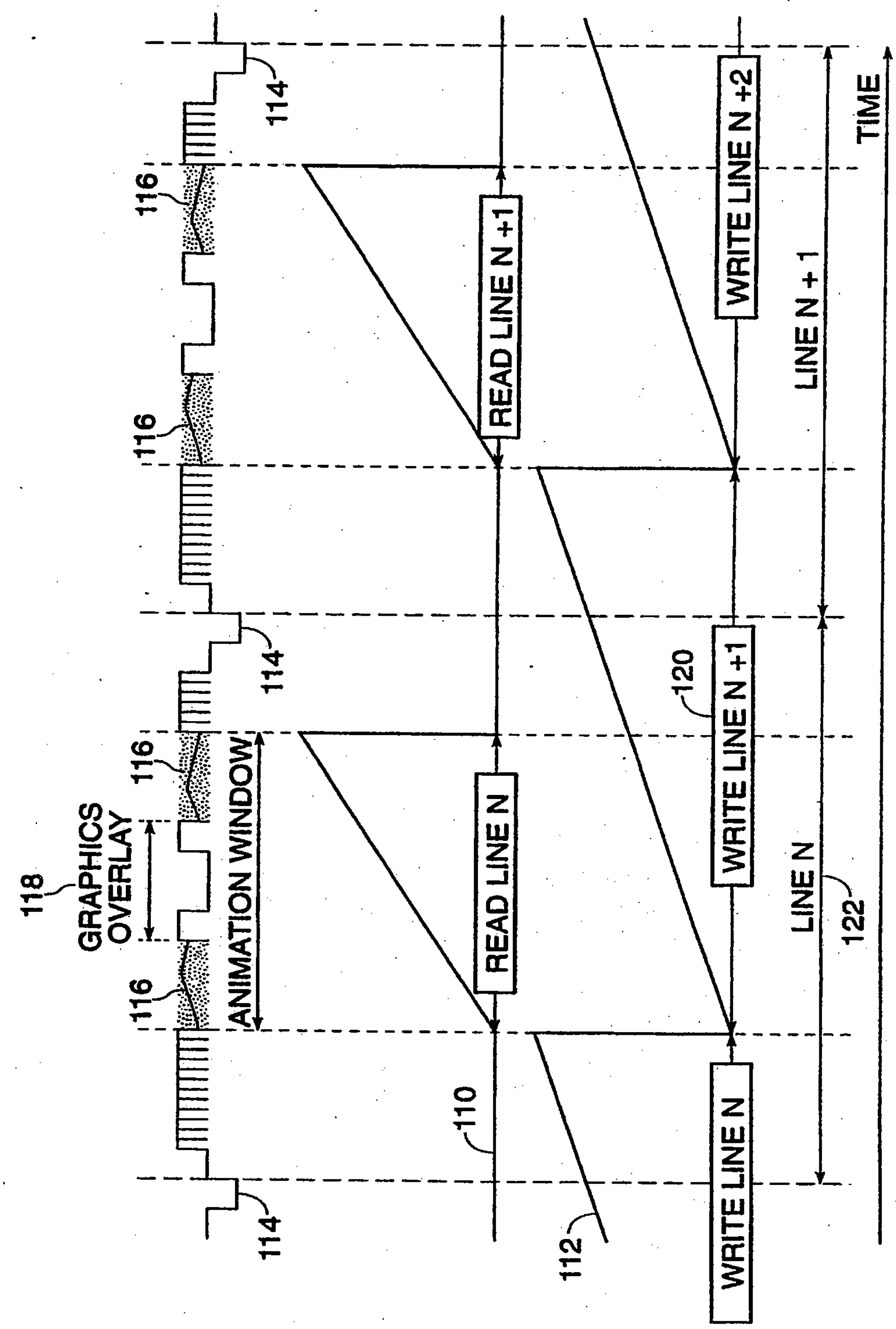

114
116
116
114
116
118
GRAPHICS OVERLAY
116
ANIMATION WINDOW
114
READ LINE N +1
READ LINE N
110
112
WRITE LINE N +2
120
WRITE LINE N +1
WRITE LINE N
LINE N + 1
LINE N
122
TIME

FIG. 7

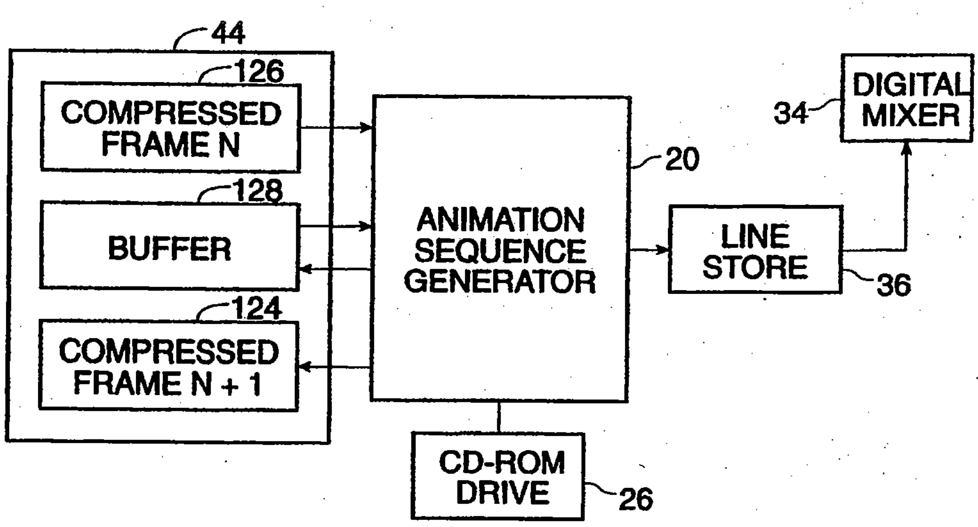

44
126
COMPRESSED FRAME N
128
BUFFER
124
COMPRESSED FRAME N + 1
ANIMATION SEQUENCE GENERATOR
20
CD-ROM DRIVE
26
LINE STORE
36
DIGITAL MIXER
34

FIG. 8A

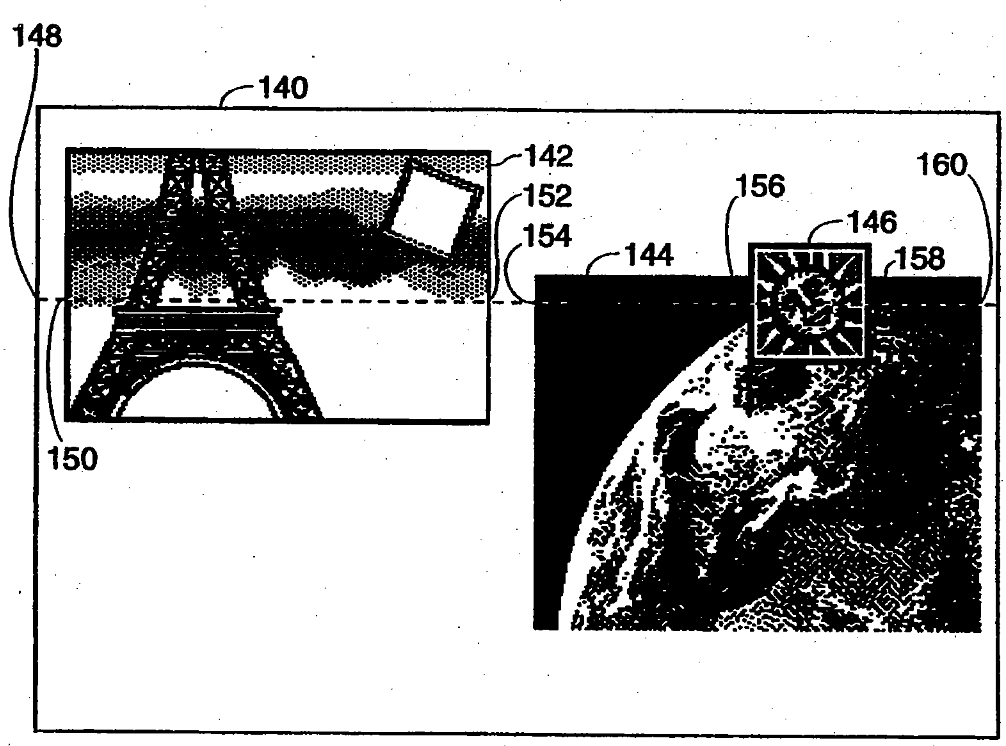

148
140
142
152
154
144
156
146
158
160
150

FIG. 9

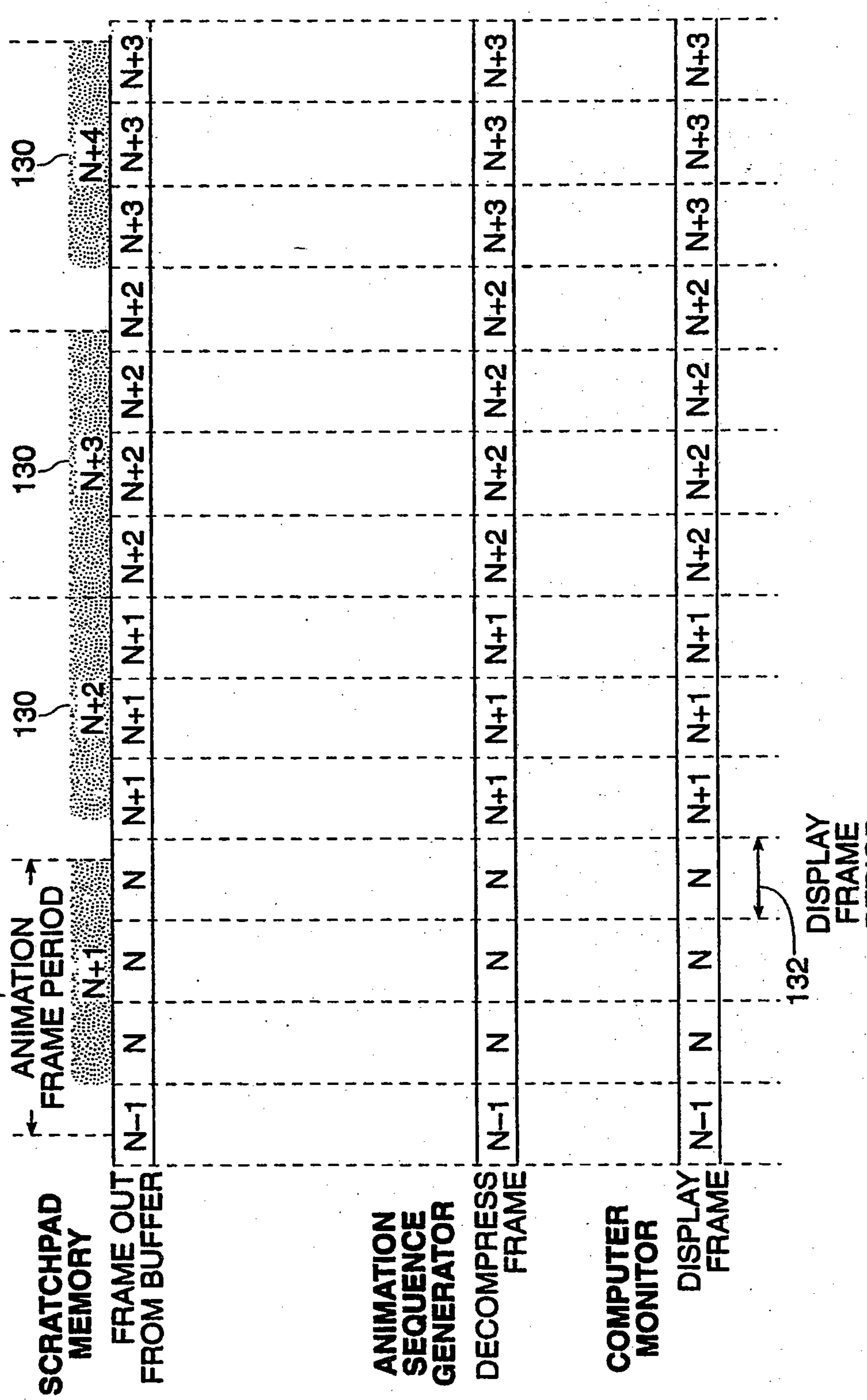

SCRATCHPAD MEMORY
FRAME OUT FROM BUFFER
ANIMATION SEQUENCE GENERATOR
DECOMPRESS FRAME
COMPUTER MONITOR
DISPLAY FRAME
ANIMATION FRAME PERIOD
130
N+1
N+2
N+3
N+4
N−1
N
DISPLAY FRAME PERIOD
132

FIG. 8B

MIXING OF COMPUTER GRAPHICS AND ANIMATION SEQUENCES

BACKGROUND OF THE INVENTION

1. Field Of The Invention

The present invention relates to the field of computer animation/graphics display systems. More particularly, the present invention relates to apparatus and methods for mixing animation sequences stored in compressed form in a standard digital storage system, and computer graphics generated by the computer employing a single line store to synchronize the decompressed animation sequences to the computer graphics.

2. The Prior Art

Many new multimedia applications require the presentation of animation sequences together with computer graphics on the same computer monitor. A dedicated animation sequence generator is used to decode the animation sequences stored in compressed form. Combining images delivered by the computer graphics generator and the animation sequence generator presents several problems.

The first problem relates to synchronization. The computer graphics generator outputs pixels at a certain pixel frequency (25 to 50 MHz), line frequency (35 kHz typically) and frame frequency (in excess of 70 Hz). In contrast, the animation sequence generator outputs pixels at typically a maximum frequency of 20 MHz, and a frame frequency of 25 or 30 Hz (a frame being typically 640×480 pixels). Before the computer graphics and the animation sequences can be combined, they have to be synchronized. Most of the time, the animation sequences have to be synchronized to the computer graphics. This is usually done by employing one or several frame stores, each frame store being implemented with dual port dynamic RAM (Video RAMs or VRAMs). This implementation results in a high product cost, since at least one megabyte of expensive VRAMs have to be used.

It is desirable for the animation images to occupy only part of the computer screen. Therefore, the animation images have to be resized dynamically in real time. This resizing usually requires one frame store, with its addition to the cost of the product. Moreover, resampling must be followed by filtering if a good quality animation image is to be displayed. Temporal filtering in particular requires simultaneous access to corresponding lines from consecutive frames. This usually requires n frame stores, if an n-order temporal filter is to be implemented. This results in a geometric increase of the memory requirement, and the cost of the product, as more sophisticated temporal filtering is implemented.

In some applications, it is desirable to simultaneously present on the computer screen several animation sequences mixed with the computer graphics. This usually requires the use of several animation sequence generators, thereby increasing considerably the cost of the product and the complexity of the hardware design.

BRIEF DESCRIPTION OF THE INVENTION

According to the present invention an apparatus for mixing animation sequences and computer graphics for display on a computer terminal display or the like includes a storage medium for storing information representing animation sequences. According to a presently preferred embodiment of the invention, animation sequences are stored in compressed format in a standard storage medium, such as a hard disk drive, a CD-ROM player, a readwrite optical drive, a server storage medium, etc. An animation sequence generator decompresses and stores information representing the pixel data for a predetermined number of frames of the animation sequence. A single line store is used to synchronize the animation sequence pixel data with the computer graphics pixel data. The mixed and synchronized pixel data is then presented to the display driver circuitry which drives the display.

The pixel data in the line store may be resampled and horizontally filtered, and can be temporally filtered by using n line stores if a n-order filter is desired. Finally, multiple animation sequences can be simultaneously delivered by the same animation sequence generator, by either downsizing each sequence or playing each sequence at lower frame rates.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating an animation sequence generator and overlay electronics according to the present invention connected in combination with a computer system.

FIG. 2 is a block diagram of the animation sequence generator and overlay electronics of FIG. 1 according to a presently preferred embodiment of the invention.

FIG. 3 is a block diagram of a digital mixer suitable for use in the present invention.

FIG. 4 is a block diagram illustrating a detailed implementation of a line store useful in the present invention.

FIG. 5 is a block diagram of a presently preferred embodiment of an animation sequence generator according to the present invention.

FIG. 6 is a depiction of an illustrative computer display with an animation sequence window shown thereon.

FIG. 7 is a timing diagram illustrating the relative timing of the read and write addresses for the line store, and the display line including the animation window position.

FIG. 8*a* is a block diagram illustrating the loading of animation frames into the scratchpad memory through the animation sequence generator

FIG. 8*b* is a diagram showing the relative timing of the scratchpad memory data, the animation sequence generator and line store data, and the display on the computer monitor.

FIG. 9 is a representation of the contents of a screen during a single frame wherein multiple animation windows are displayed.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Those of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled persons.

Referring first to FIG. 1, a block diagram is shown of a computer system incorporating an animation sequence generator according to the present invention. As shown in FIG. 1, a general computer 10, such as the IBM PC-AT or compatible machines, includes a central processing unit (CPU) 12, and some system random access memory (RAM) 14. The CPU is also connected to a computer graphics generator 16, such as a VGA or SVGA graphics generator, as is known in the art. The computer graphics generator has a digital output bus 18 which delivers 8 bits of pixel information (thereby enabling the presentation of at most 256 colors), the horizontal and vertical synchronization signals, and the pixel clock.

According to the present invention, the output 18 is connected to animation sequence generator and overlay electronics 20, which contains the electronics necessary to perform the mixing of the animation sequence pixels and the computer graphics pixels and deliver the composite signal to the computer display. Given the modular nature of personal computers, disclosure of the present invention will be made using an illustrative embodiment comprising a single circuit card including the electronics necessary to perform the mixing of the animation sequence pixels and the computer graphics pixels and generate the composite signal used to drive the computer display 22, but those of ordinary skill in the art will recognize that the circuit board distribution of the circuit elements disclosed herein is somewhat arbitrary and is largely a matter of design choice.

The animation sequences for use by animation sequence generator 20 may be stored in compressed form in any one of a number of different storage media, such as hard disk drive 24, a CD-ROM drive 26, a read/write optical storage 28, or a server through the server connection 30. Those of ordinary skill in the art will recognize that one or more of these devices may be easily interfaced with the other elements of the invention.

More specifically, and as may be seen from an examination of FIG. 2, a block diagram of the animation sequence generator 20 of FIG. 1, the digital output bus 18 from the computer graphics generator 16 is connected to input 32 of digital mixer 34. Digital mixer 34 may comprise a digital multiplexer having its select input controlled by a chromakey gating circuit, as will be more fully disclosed with reference to FIG. 4, and is used to mix and overlay pixels from the computer graphics generator 16 and the animation sequence generator 20. Pixels from the animation sequence generator 20 are buffered by the single line store 36. The output of the line store 36 is connected to input 38 of the digital mixer 34 so as to deliver animation-sequence pixels to the digital mixer 34 one line at a time. Control logic 40 operates under the control of CPU 12 in computer 10 to direct the operation of line store 36 and digital mixer 34, as will be further discussed herein.

The output of digital mixer 34 is connected to a video digital-to-analog converter (DAC) 42. Video DAC 42 converts the digital pixel information from the digital mixer 34 to analog signals, which are then fed to the computer monitor 22. The design and use of video DAC circuits are both well known in the art.

Finally scratchpad memory 44 is associated with animation sequence generator 20. Scratchpad memory 44 is used to store, among other things, animation sequence frames in compressed data form.

As an example, the system of FIG. 2 is shown coupled to CD ROM Drive 26, which supplies compressed data representing one or more animation sequences to animation sequence generator 20. They are buffered in the scratchpad memory 44. The animation sequence generator 20 reads blocks of compressed pixels from a particular frame from the scratchpad memory 44, decompresses these blocks into individual digital pixels. The decompressed sequence pixels are sent to line store 36.

In the implementation depicted in FIG. 2, line store 36 has a capacity of 512 words. Thus, in this embodiment, the maximum number of animation sequence pixels is then limited to 512 pixels per line. Those of ordinary skill in the art will recognize, however, that these pixels can be duplicated before they are sent to the digital mixer 34, in order that the desired number of pixels per line are delivered. For example, if 640 pixels of animation are to be displayed per line, and the animation sequence generator 20 sends only 320 pixels per line to the line store 36, the control logic 40 can force the line store to read the same pixel out of the line store twice.

Conversely, the decompressed pixels sent by the animation sequence generator 20 can also be decimated by the control logic 40. For example, if the animation sequence generator 20 delivers 320 pixels per line, and only 240 animation pixels are to displayed per line, the control logic then enables the writing of only 3 out of every 4 decompressed pixels, to decimate the pixel line count from 320 animation pixels to 240 animation pixels. The control logic 40 also selects the operation for the digital mixer 34 through control signals asserted on control bus 46. The output of digital mixer 34 is sent to the video DAC 42 which generates analog R, G, B signals to present to computer monitor 22.

Referring now to FIG. 3, a digital mixer 34 suitable for use in the present invention is shown in block diagram form. Multiplexer 48 has a first data input 32 connected to output bus 18 from graphics generator 16 and a second data input 38 connected to the output bus of line store 36. A control input 50 selects which of inputs 32 and 38 is passed to output 52, which forms the output of digital mixer 34.

The selection of the one of inputs 32 and 38 to pass to output 52 of multiplexer 48 is controlled by 8-input NOR gate 54, byte-wide AND gate 56, byte-wide XOR gate 58, Chromakey register 60 and Mask register 62. Chromakey register 60 is loaded by control logic 40 with a color value representing the color of a region which defines the portion of the video frame area to be occupied by the animation sequence. Typically, such a region is rectangular in shape (although it may be otherwise) and is defined by computer 10 as is well known in the art. The byte defining this color is compared with the pixel byte stream from output bus 18 of graphics generator 16 by byte-wide XOR gate 58. The individual bits of the output of the XOR gate 58 are compared with the value stored in mask register 62 in AND gate 56. The outputs of AND gate 56 are presented to 8-input NOR gate 54. If the value 00(Hex) is stored in mask register 56, the output of NOR gate 54 will be a logic one and the graphics generator pixels will be passed to the output of multiplexer 48. If, however, any value other than 00(Hex) is stored in chromakey register 60 and any value other than 00(Hex) is stored in Mask register 62, the output of NOR gate 54 will be a logic zero and the animation sequence pixels will be passed to the output of multiplexer 48.

According to the present invention, the animation sequence images are synchronized to the computer graphics images before the digital mixer, by using a single line store 36, instead of one or multiple complete frame stores as is customary. There are three requirements for synchronization. First, the animation pixel frequency must be equal to the computer graphics pixel frequency Fpixel. In addition, the animation pixel horizontal frequency must be equal to the computer graphics horizontal frequency Fh. Finally, the animation pixel vertical frequency must be equal to the computer graphics vertical frequency Fv.

Given Pa, the number of animation pixels to be displayed per horizontal line, the pixel frequency output from the animation sequence generator 20 Fa are written into the line store:

$$Fa = Pa*Fh$$

The pixels are read out of the line store at frequency Fpixel which is the computer graphics pixel frequency. In general Fa is much lower than Fpixel. This insures that the animation pixel frequency is equal to the computer graphics pixel frequency Fpixel. Since a complete line of animation pixels is written and read to and from the line store at frequency Fh, the animation pixel horizontal frequency is equal to the computer graphics horizontal frequency Fh.. Thus, the first two requirements are satisfied.

Animation sequences are generated at typically 24, 25 or 30 different frames per second, whereas the computer graphics frame rate varies from 60 frames per second to 76 frames per second. To synchronize the two different frame rates, the animation sequence generator 20 delivers the same frame multiple times so that the total number of animation frames per second is equal to the computer graphics frame rate. For example, to convert from 24 animation frames per second to 60 computer graphics frames per second, the 24 animation frames may be repeated in some manner, i.e., by repeating each odd numbered frame twice and each even numbered frame three times: A,A,B,B,B,C,C,D,D,-D,E,E,F,F,F, G,G,H,H,H,I,I,J,J,J,K,K,L,L,L,M,M,N,N,-N,O,O,P,P,P,Q,Q,R,R,R,S,S,T,T,T,U,U,V,V,V, W,W,X,X,X, for a total of sixty frames, where A through X represent frames 1–24, respectively of the animation sequence. Those of ordinary skill in the art will recognize that other similar schemes are possible. By having the animation sequence generator 20 repeat frame,, in such a selected pattern, the third requirement may be satisfied without the need to provide a complete frame store.

Referring now to FIG. 4, it is seen that line store 36 can be built as a static RAM 64 with separate write port 66 and read port 68. The write port 66 is connected to the output of the animation sequence generator 20. The data on write port is 24 bits wide, and is used to store RGB pixel information from the decompressed animation sequence. The write port address bus 70 is used to select one location in the static RAM to store the next pixel information. The write port control 72 is used to enable the write operation. Both write address bus 70 and write control 72 come from the control logic 40. Control logic 40 may comprise write and write address generators (typically counters), and read and write control signal generators (typically state machines and combinatorial logic). The read and write control signals are synchronized by external signals in a conventional manner. Digital pixel information is read from the Read Port 74. The read port 74 is connected one of the inputs of the digital mixer, and conveys the 24 bit RGB pixel information. The read port address bus 76 is used to select one location in the static RAM to read the next pixel information from. The read port control 78 is used to enable the read operation. Both read address bus 76 and read control 78 come from the control logic 40. Architectures like line store 36 are well known in the art. Those of ordinary skill in the art will recognize that a first-in, first-out (FIFO) buffer is equivalent to the line store memory for the purposes of the present invention and may be substituted therefor.

Referring now to FIG. 5, a block diagram of a presently preferred embodiment of animation sequence generator 20 is shown. Animation sequence generator 20 communicates with the host processor via host interface 80, which may comprise address and data latches, and host bus controllers, and controls communications with the host processor in a standard manner. The host processor generally sends animation sequences in compressed form to the animation sequence generator 20 through this port. Animation sequence generator 20 communicates with scratchpad memory 44 via memory interface 82, which may comprise row-address strobe (RAS) generators, column-address strobe (CAS) generators, address multiplexers, data buffers, and a refresh controller as is well known in the art.. Memory interface 82 is used to control external scratchpad memory 44, which is generally one or several megabytes of DRAM. Scratchpad memory 44 is used to store data or instructions.

The operation of animation sequence generator 20 is directed by instruction sequencer 86, which may comprise an instruction buffer and instruction decoder commonly employed in microprocessor systems. Instruction sequencer controls the operation of a conventional ALU (arithmetic logic unit) 86, used to perform general operations such as add, subtract, logical and, logical invert, logical or, magnitude comparison, a hardwired multiplier 88, used to perform very fast multiplications needed to perform image processing operations such as those found in decompression algorithms such as the standard JPEG and MPEG algorithms. Dedicated hardware 90 is used to perform special functions such as run-length decoding (Huffmann decoding), discrete cosine transform, and bit block transfer operations. Off-the-shelf, special-purpose integrated circuits from numerous manufacturers are readily available to perform these functions. Internal registers, comprising 32 words of 32 bits each, are used to store operands upon which ALU 86, the multiplier 88 and the special hardware 90 will operate. Finally, graphics output controller 94 is used to provide an interface for the decompressed animation pixel output to line store 36 on bus 96. The elements of animation sequence generator 20 are connected together by an internal bus 98 as is well known in the art.

In a typical operating sequence, the host processor 10 reads compressed animation sequences from a storage medium (hard disk, CD-ROM, server,etc), and sends those compressed sequences to the animation sequence generator 20 through the host interface port 80. The animation sequence generator 20 temporarily stores those compressed data in the external scratchpad memory 44 which is then used as a buffer for this information. This compressed information is then read from the external scratchpad memory 44 by the animation sequence generator 20, and decompressed according to standard decompression software, such as MPEG standard digital video decompression, stored in the same external scratchpad memory 44. This software processes the compressed data to yield decompressed pixel information in the ALU 86, multiplier 88, and dedicated hardware 90 are activated by the instruction sequencer 84 in response to specific programmable instructions stored in the external scratchpad memory 44. The decompressed pixels are then delivered through the graphics output controller port 96.

Referring now to FIG. 6, a typical computer display 100 includes several windows. A typical application is to show the animation sequence inside of a window, such as window 102. Objects can be overlaid on top of the animation sequence pixels, such as window 104, and objects, such as 106, can be underlaid behind the window 102. The overlay of computer graphics pixels and animation sequence pixels is controlled by the digital mixer 34 (FIG. 2). The digital mixer 34 can deliver to the video DAC 42 pixels from computer graphics generator 16 or from animation sequence generator 20. Pixel lines contained in area 108 of FIG. 4 will be examined in more detail.

FIG. 7 is a timing diagram showing synchronization of the animation pixel and line frequencies to the computer graphics pixel and line frequencies. The upper waveform of FIG. 7 comprises the amplitude of the analog video signal which is sent to the computer monitor. The middle waveform 110 is a representation of the read addresses asserted to the line store on write-address bus 70, where the increasing slope of the line symbolically indicates incrementing addresses. The lower waveform 112 is a representation of the write addresses asserted to the line store on read-address bus 76, where the increasing slope of the line also symbolically indicates incrementing addresses. The three waveforms are aligned to one another along the horizontal axis, which is time. Two display lines are described in FIG. 7: display line N and display line N+1. Each display line begins with a horizontal sync pulse, shown at reference numeral 114.

Referring now to FIGS. 6 and 7, the waveforms of FIG. 7 illustrate the portion of the display frame illustrated at reference numeral 108 in FIG. 5. Animation sequence window 102 on the computer screen shows the animation sequence delivered by the animation sequence generator 20 of FIG. 1. This animation window appears as a portion 98 of each display line. Note that computer graphics pixels 104 are overlaid on top of the animation pixels 102 to represent the region 104 of the display frame.

As shown in waveform 110 of FIG. 7, the animation pixels are read from the line store in consecutive address locations synchronously with the computer graphics pixels. This ensures the animation pixels are delivered to the digital mixer 34 at the same frequency as the computer graphics pixels. Even though the computer graphics pixel frequency can be high (typically 50 MHz), only the line store read operation has to be performed at that high frequency. As shown in waveform 112 of FIG. 6, the animation pixels coming from the animation sequence generator 20 may be written into the line store at a much lower frequency. In essence, the animation pixels for line N+1 can be written into the line store as soon as animation pixels for line N are being read out, and the writing operation can last until the first pixel for line N+1 is to be read out. The writing time interval (reference numeral 122) is equal to the display line time interval.

If Pa is the number of animation pixels to be displayed per line, and Fh is the display line frequency; then the animation pixels have to be written into the line store at a frequency

$$Fa=Pa*Fh.$$

Typically: Pa=340 pixels, Fh=35 kHz, and then Fa=340*35=11.9 MHz. Those of ordinary skill in the art will appreciate the difference between the write frequency (11.9 MHz) and the read frequency (50 MHz). The write frequency is in general much lower than the read frequency. As those of ordinary skill in the art will appreciate, the animation sequence generator 20 can operate at a lower frequency than the computer graphics pixel frequency, and the single line store 36 can resynchronize both pixel and line frequencies.

FIGS. 8*a* and 8*b* illustrate how the animation sequence frame rate may be synchronized to the computer graphics frame rate. Referring first to FIG. 8*a*, a block diagram of a portion of the apparatus of FIG. 2, the animation sequence generator 20 receives compressed information from the CD-ROM 26 and stores that information in the scratchpad memory 44. While the compressed frame N+1 is being stored in a buffer 124 of scratchpad memory 44, the compressed information for frame N is read out multiple times from the a buffer 126 storing compressed frame N to be decompressed by the animation sequence generator 20 and sent to the line store 36. The decompressed animation pixel information is stored in buffer area 128 for delivery to line store 36 in due course.

FIG. 8*b* is a graphic representation explaining the timing relationship between the animation frames and the computer graphics frames. FIG. 8*b* illustrates the timing relationships between the pixel information in the CD ROM drive 26 and the various buffers in scratchpad memory 44.

As shown in the top line of FIG. 8*b* from left to right, compressed information from frames (N+1), (N+2), and (N+3) are obtained from CD ROM drive 26 and are stored in the scratchpad memory 44. This occurs during the animation frame periods 130 (illustrated between the vertical lines; typically every 1/30 of a second). As will be understood by those of ordinary skill in the art and as may be seen in FIG. 8*b*, these data transfers may be performed in burst modes as time permits. The computer graphics display frame periods 132 is generally much faster, typically 1/70 of a second.

While compressed frame (N+1) is being stored in buffer 124 in scratchpad memory 44 during the first animation frame period illustrated in FIG. 8*b*, compressed frame (N) is read for decompression from buffer 126 to animation sequence generator 20 at the computer graphics frame rate 132. Because the computer graphics frame rate 132 is in general much shorter than the animation frame rate 130, the same compressed frame (N) is repeatedly read out to the animation sequence generator 20 which in turn decompresses the same frame multiple times and stored in buffer area 128, from which it is fetched, one line at a time, for delivery to line store 36. As previously mentioned, any frame may be displayed several times to synchronize the graphics and animation information. As illustrated in FIG. 8*b*, frame (N) is displayed 3 times, frame (N+1) is displayed 3 times, frame (N+2) is displayed 4 times, and so on. At the start of a new animation frame period, frame (N+1) is available in buffer 124 of the scratchpad memory. At the same time, decompressed frame (N) is being read out of buffer 128 to the animation sequence generator 20 to be decompressed and displayed. The next frame to be displayed is then (N+1), and the scratchpad memory starts buffering frame (N+2) from CD ROM 26.

This adaptive repetition of compressed animation frames converts the animation frame rate to the computer graphics frame rate, and therefore performs the required synchronization between the animation frame rate and the computer graphics frame rate.

When several sequences are simultaneously presented on the computer screen, the total number of visible pixels from all animation sequences must of course be less the total resolution of the computer screen: this means the animation sequence generator 20 never has to deliver more pixels per frame than can be displayed on the computer screen; therefore the animation sequence generator 20 can be shared in time between all the displayed animation sequences.

FIG. 9 is a representation of the contents of a screen 140 during a single frame wherein multiple animation windows are displayed. Three animation sequences 142, 144, and 146 are displayed simultaneously, each in its own window. Note that window 144 overlaps window 142. A scan line 146 comprises all three animation windows, as well as the computer graphics information. Scan line begins displaying graphics information at point 148. At point 150 it is seen to intersect first animation window 142, and then later at point 152 again comprises graphics information. At point 154 it first intersects second animation window 144 and then third animation window 146 at point 156, which overlaps the second animation window 144. It enters second animation window 144 again at point 158, and the graphics background again at point 160.

The animation sequence generator 20 stores in its scratchpad memory one compressed frame per animation sequence. At point 150, the animation sequence generator 20 decompresses the frame for first animation window 142; at point 154, the animation sequence generator 20 decompresses the frame for second animation window 144; at point 156, it decompresses the frame for third animation window 146, and resumes decompressing the frame for second animation window 144 at point 158. The animation sequence generator 20 has to decompress and assemble in the line store the pixels for all three animation sequences in the proper order and with proper overlay, during the duration of one computer screen scan line. Note again that because the maximum number of animation pixels from all animation sequences, visible in one scan line cannot be more than the scan line resolution, the animation sequence generator 20 can share its processing cycle between the animation sequences. However, because switching from decompressing one sequence to decompressing another sequence takes some time, the total number of pixels the animation sequence generator 20 can decompress per line is less when several sequences are visible, than when only one sequence is visible.

Given:

Th the duration of a computer screen scan line;

Pa the total number of animation pixels displayed in that line;

t the time required to decompress one pixel of a given sequence;

Ttrans the time required by the animation sequence generator 20 to change from decompressing one animation sequence to decompressing another animation sequence; then the maximum number n of different animation sequences that the animation sequence generator 20 can deliver per line is such that:

$$t*Pa+n*Ttrans<Th$$

While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art that many more modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.

What is claimed is:

1. In a computer system including a computer graphics generator producing pixels at a first selected frequency to compose lines at a second selected frequency to compose frames at a third selected frequency, said computer system further including a digital mixer for mixing, windowing, and overlaying computer graphics images and animation sequence images on a pixel-by-pixel basis for delivery to a visual display, apparatus for presenting animation information comprising pixels, lines, and frames from said animation sequence images at said first, second, and third selected frequencies in synchronization with pixels, lines, and frames from said computer graphics generator, said apparatus comprising:

first storage means for storing said animation information as compressed animation information;

means for retrieving and for temporarily storing said compressed animation in blocks comprising a selected number of frames of said compressed animation information;

means for decompressing and for temporarily storing amounts of said animation information comprising a selected number of pixels from a selected number of lines from a selected frame in a selected one of said blocks containing said compressed information, said selected frame being selected a number of consecutive times so as to synchronize with frames from said computer graphics generator at said selected third frequency; and

means for storing and presenting pixels from a selected number of lines of said amounts of said animation information to said mixer, said means for storing and presenting pixels operating to supply said pixels at first said selected frequency, and said lines at second said selected frequency in synchronization with pixels and lines from said computer graphics generator, said means for storing and presenting pixels comprising storage for fewer lines than comprise a frame of said information.

2. The apparatus of claim 1 wherein said means for storing and presenting pixels from a single line of said information comprises storage for a single line of pixels from said information.

3. The apparatus of claim 1 wherein said means for storing and presenting pixels from a single line of said information comprises a single line dual ported memory with separate read and write addressing circuits.

4. The apparatus of claim 1 wherein said first storage means comprises a compact disk drive.

5. The apparatus of claim 1 wherein said animation sequences comprise computer generated video images.

6. The apparatus of claim 1 wherein said animation sequences comprise realtime video images.

7. In a computer system including a computer graphics generator producing pixels at a first selected frequency to compose lines at a second selected frequency to compose frames at a third selected frequency, said computer system further including a digital mixer for

mixing, windowing, and overlaying computer graphics images and animation sequence images on a pixel-by-pixel basis for delivery to a visual display, apparatus for presenting animation information comprising pixels, lines, and frames from said animation sequence images from more than one animation sequence at said first, second, and third selected frequencies in synchronization with pixels, lines, and frames from said computer graphics generator, said apparatus comprising:

first storage means for storing said animation information from said more than one animation sequence as compressed animation information;

means for retrieving and for temporarily storing said compressed animation from said more than one animation sequence in blocks comprising a selected number of frames of said compressed animation information;

single means for decompressing and for temporarily storing amounts of said animation information comprising a selected number of pixels from a selected number of lines from a selected frame of each of said more than one animation sequences in a selected one of said blocks containing said compressed information, said selected frame being selected a number of consecutive times so as to synchronize with frames from said computer graphics generator at said selected third frequency; and

single means for storing and presenting pixels from a selected number of lines of said amounts of said animation information to said mixer, said means for storing and presenting pixels operating to supply said pixels at first said selected frequency, and said lines at second said selected frequency in synchronization with pixels and lines from said computer graphics generator, said means for storing and presenting pixels comprising storage for fewer lines than comprise a frame of said information.

8. The apparatus of claim 7 wherein said single means for storing and presenting pixels from a single line of said information comprises storage for a single line of pixels from said information.

9. The apparatus of claim 7 wherein said single means for storing and presenting pixels from a single line of said information comprises a single line dual ported memory with separate read and write addressing circuits.

10. The apparatus of claim 7 wherein said first storage means comprises a compact disk drive.

11. The apparatus of claim 7 wherein said animation sequences comprise computer generated video images.

12. The apparatus of claim 7 wherein said animation sequences comprise realtime video images.

13. A computer system for displaying animation sequences and graphics on a visual display as a set of pixels comprising a set of lines comprising a set of frames, said system comprising

an animation sequence generator, said animation sequence generator comprising an animation input port for accepting animation information as a set of blocks having a set of frames of said animation information;

a memory coupled to said animation sequence generator, said memory comprising less storage than for an entire frame of pixels;

a graphics generator, said graphics generator comprising a graphics output port for presenting pixels at a first selected frequency, lines at a second selected frequency, and frames at a third selected frequency;

a memory control element coupled to said memory, said memory control element comprising a read address port, a read control line, a write address port, and a write control line, said memory control element operating to read sequences of pixels from said memory at said first selected frequency, to read sequences of lines from said memory at said second selected frequency, and to write frames of said animation information to said memory at said third selected frequency;

a digital mixer coupled to said memory and to said graphics generator, said digital mixer comprising a mixer output port for presenting pixels, lines and frames in an output format suitable for visual display.

14. A computer system as in claim 13, wherein said animation sequence generator comprises

an animation memory for storing compressed animation information;

a processor for decompressing said compressed animation information and generating animation sequences having a set of frames of animation information; and

an animation output port for producing a selected sequence of said frames, said selected sequence comprising a set of frames having been selected a plurality of times to synchronize with frames from said graphics generator.

15. A computer system as in claim 13, comprising

an extended memory coupled to said animation sequence generator, said memory and said extended memory collectively comprising storage for a plurality of lines of pixels, said plurality of lines being less storage than an entire field; and

a temporal filter coupled to said memory and said extended memory.

16. A computer system as in claim 13, wherein said digital mixer comprises

a multiplexer coupled to said memory, to said graphics output port, and to said animation input port;

a control circuit coupled to said multiplexer, said control circuit coupled to said graphics output port and to at least one register, said control circuit disposed to generate a control signal which is a Boolean function of a content value of said at least one register and of a value presented by said graphics output port.

17. A computer system as in claim 16, wherein said control circuit comprises a chromakey register.

18. A computer system as in claim 16, wherein said control circuit is coupled to said mixer output port.

* * * * *

US005428731A

United States Patent [19]

Powers, III

[11] **Patent Number: 5,428,731**

[45] **Date of Patent: Jun. 27, 1995**

[54] **INTERACTIVE MULTIMEDIA DELIVERY ENGINE**

[75] Inventor: **John R. Powers, III**, Monte Sereno, Calif.

[73] Assignee: **Apple Computer, Inc.**, Cupertino, Calif.

[21] Appl. No.: **59,542**

[22] Filed: **May 10, 1993**

[51] **Int. Cl.**[6] **G09B 19/04**

[52] **U.S. Cl.** **395/154**; 395/155

[58] **Field of Search** 395/154, 161, 133, 152, 395/155

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,111,409 5/1992 Gasper et al. .... 395/154 X

OTHER PUBLICATIONS

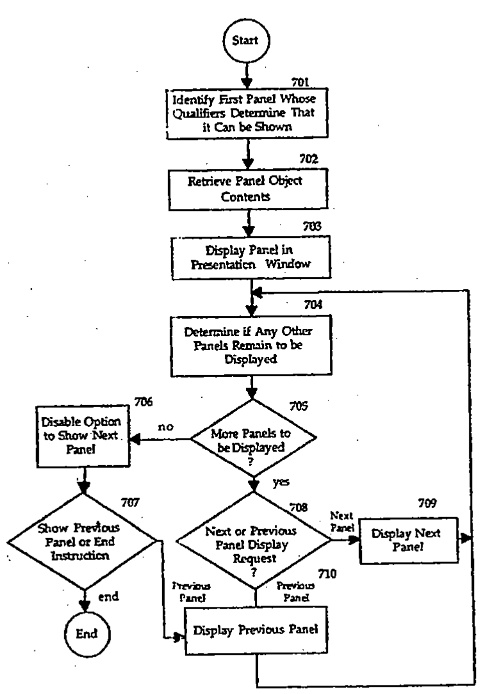
Hypercard User's Guide (Trademark of Apple Computer, Inc.), 1988, pp. XXII, 6-11, 15, 31-34, 46×95-104.
Mastering Wordperfect 5.1×5.2 for Windows (Trademark of Sybex Inc.), 1992, pp. 68-69.
Mastering Windows 3.1 (Trademark of Microsoft Corporation), 1992, pp. 31-42×856.
Winkler, D. and Kamins, S. *Hypertalk 2.0: The Book* Table of Contents and 258-261 (1990).
N. Yankelovich et al. "Intermedia: The Concept and Construction of a Seamless Information Environment" Computer, vol. 21, No. 1, Jan. 1988, Long Beach US, pp. 81-96.
S. Ueda "Hypertext System: TownsGear" Fujitsu-Scientific and Technical Journal, vol. 26, No. 3, 1990 Kawasaki JP pp. 187-196.

Primary Examiner—Heather R. Herndon
Assistant Examiner—Cliff Nguyen Vo
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

An engine for the interactive delivery and presentation of multimedia data. The delivery mechanism is premised on a topic and panel metaphor. A topic is comprised of a series of panels. Each panel further defines the multimedia data which will be used when the panel is displayed. Each panel may also contain control and qualifier elements which are used to determine the next panel to be displayed. Control elements are used to solicit information from the user. The information is then used by the qualifier elements in determining the next panel to be displayed. Each element is stored in a database as a discrete object. This allows sharing of objects amongst different panels or topics. Prior to delivery and presentation of the data, the topic must be set-up. Set-up occurs by retrieving all the necessary panel, control qualifier and data descriptor objects. The actual data associated with multimedia data objects is only retrieved when the data is to be displayed. The various qualifier objects are evaluated while panels are being viewed to determine the next panel to be displayed.

11 Claims, 15 Drawing Sheets



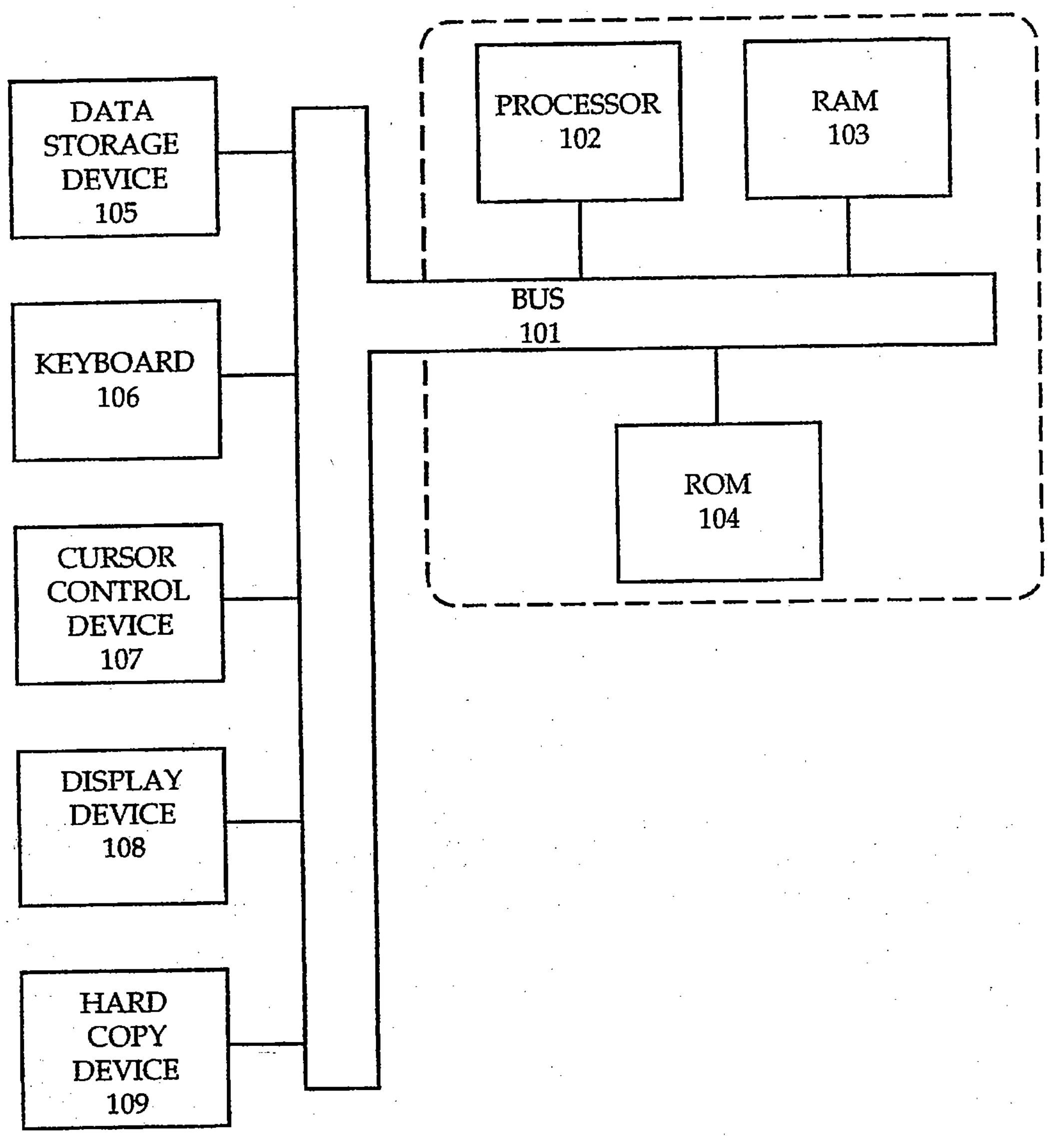

DATA
STORAGE
DEVICE
105
KEYBOARD
106
CURSOR
CONTROL
DEVICE
107
DISPLAY
DEVICE
108
HARD
COPY
DEVICE
109
PROCESSOR
102
RAM
103
BUS
101
ROM
104

Figure 1

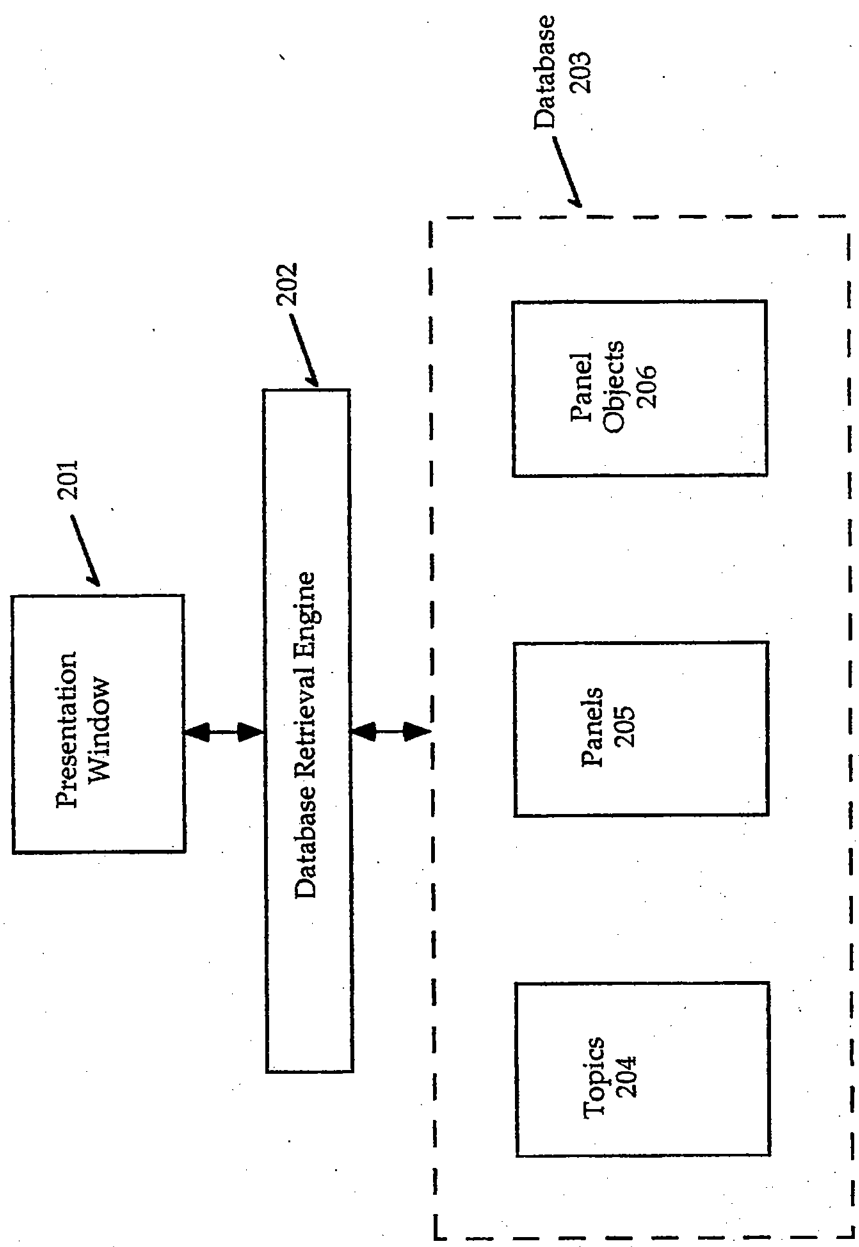

Presentation Window
201
Database Retrieval Engine
202
Database
203
Topics
204
Panels
205
Panel Objects
206

Figure 2

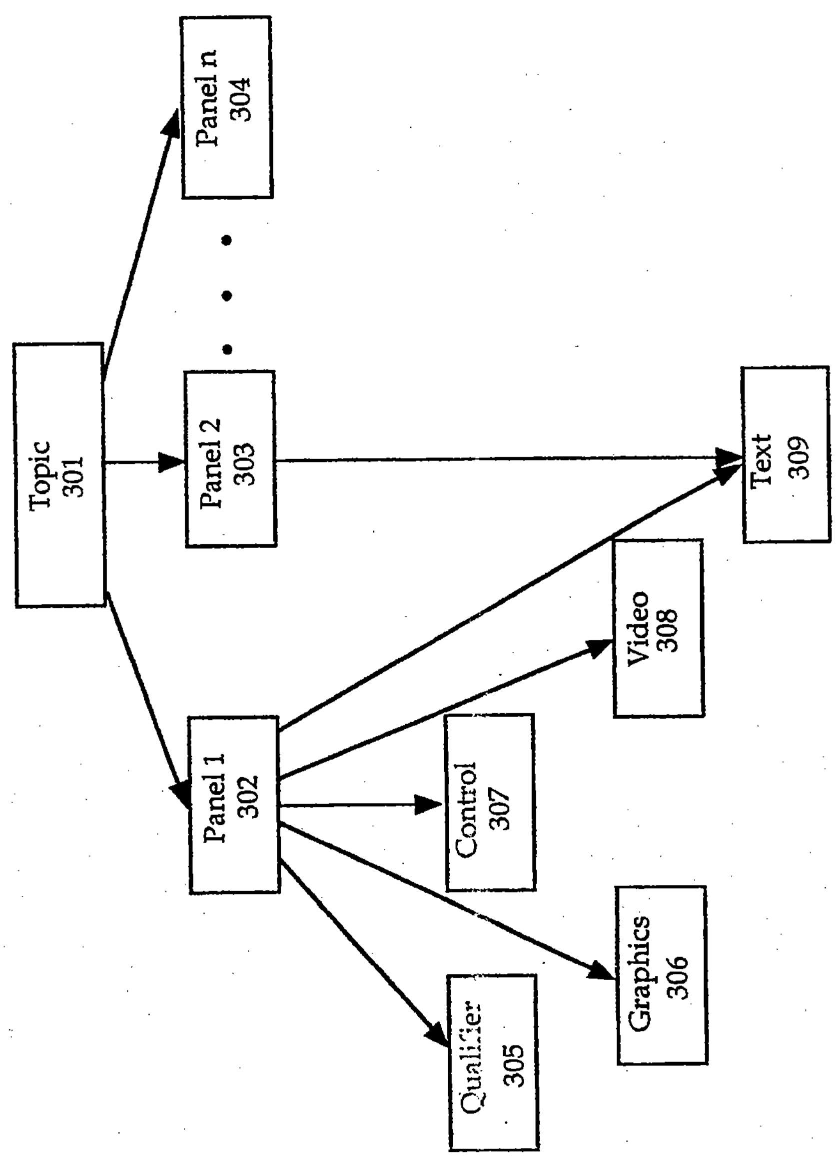

Topic
301
Panel 1
302
Panel 2
303
Panel n
304
Qualifier
305
Graphics
306
Control
307
Video
308
Text
309

Figure 3

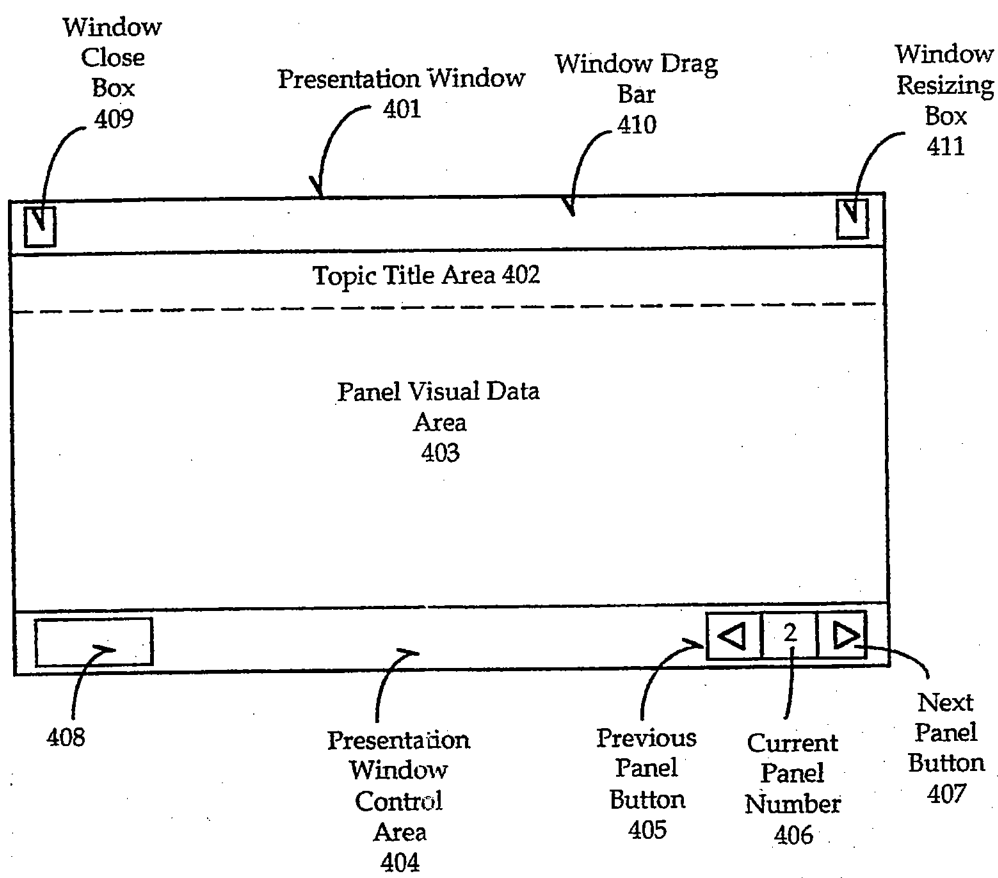

Window Close Box 409
Presentation Window 401
Window Drag Bar 410
Window Resizing Box 411
Topic Title Area 402
Panel Visual Data Area 403
2
408
Presentation Window Control Area 404
Previous Panel Button 405
Current Panel Number 406
Next Panel Button 407

Figure 4

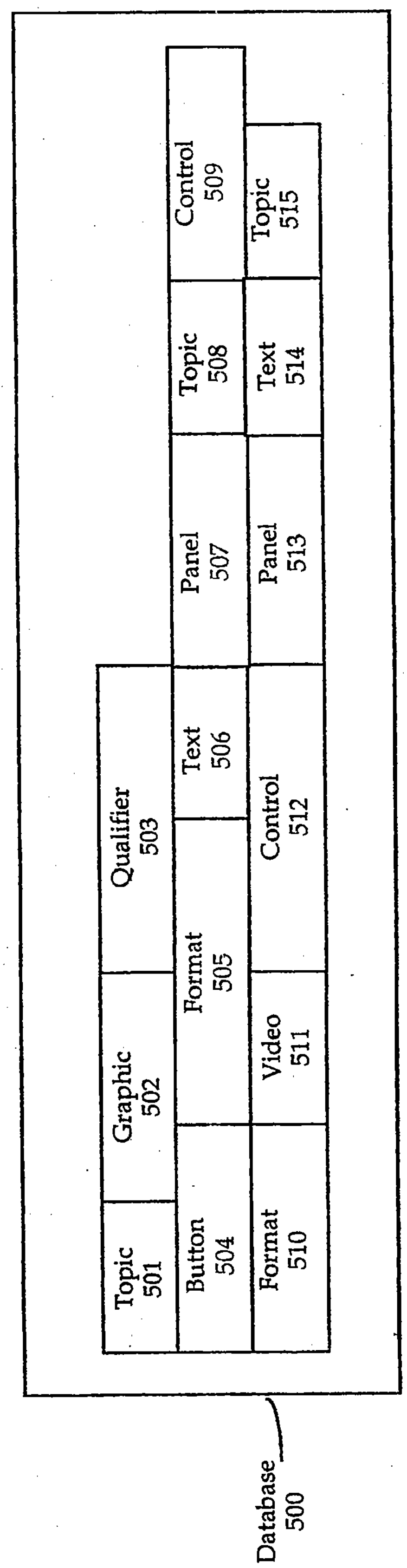

Topic 501
Graphic 502
Qualifier 503
Button 504
Format 505
Text 506
Panel 507
Topic 508
Control 509
Format 510
Video 511
Control 512
Panel 513
Text 514
Topic 515
Database 500

Figure 5a

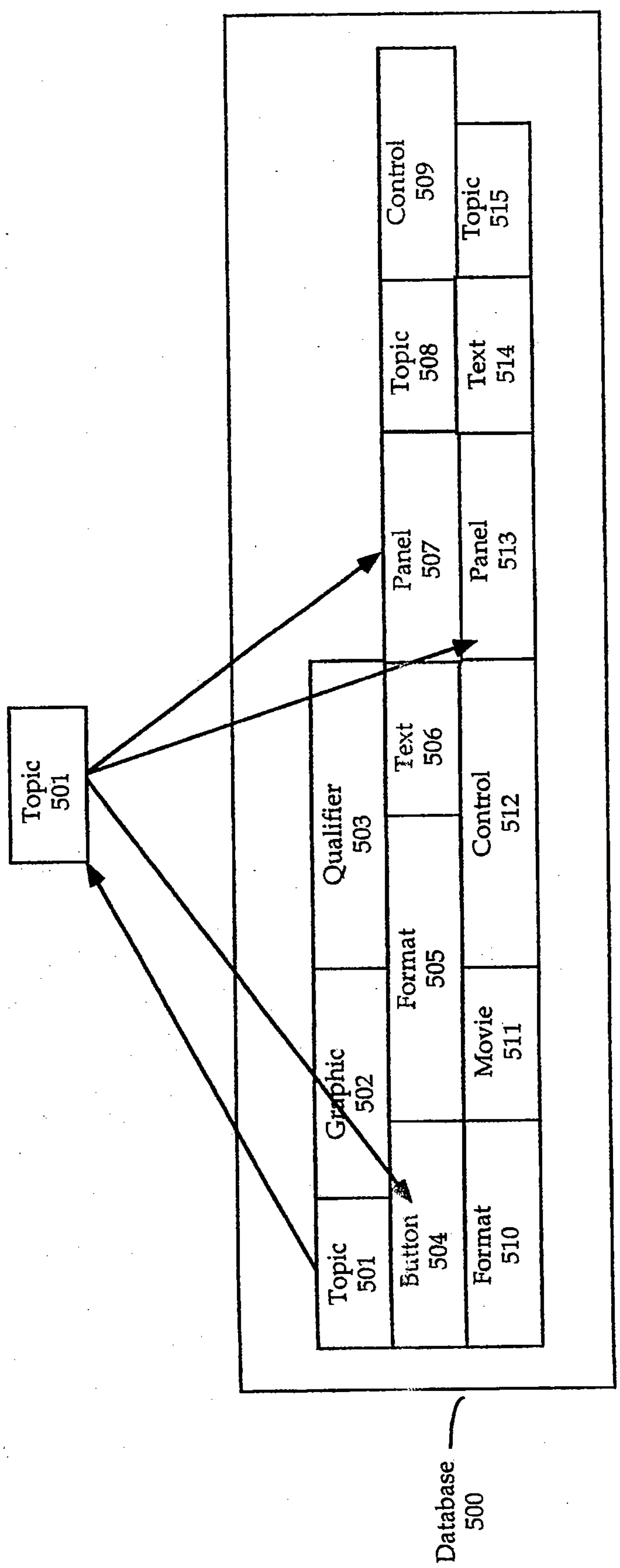

Topic 501
Topic 501
Graphic 502
Qualifier 503
Button 504
Format 505
Text 506
Panel 507
Topic 508
Control 509
Format 510
Movie 511
Control 512
Panel 513
Text 514
Topic 515
Database 500

Figure 5b

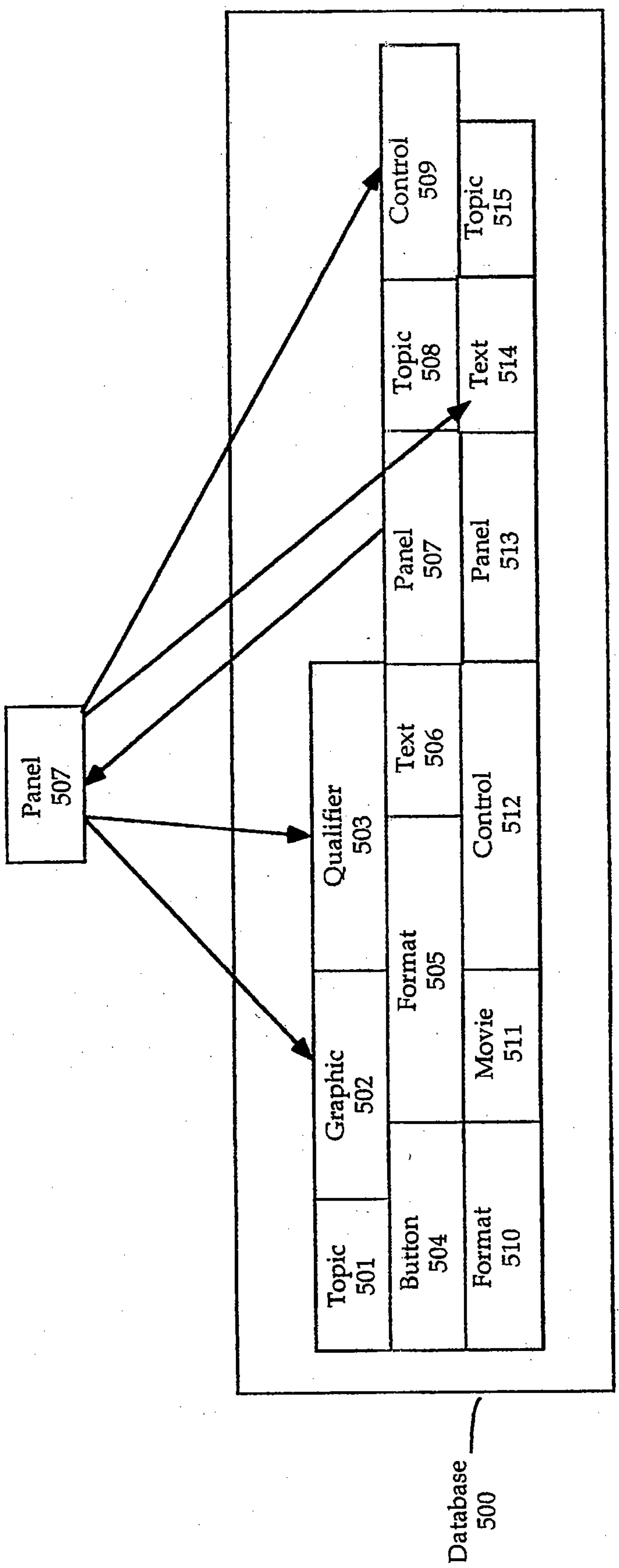

Panel
507
Topic
501
Graphic
502
Qualifier
503
Button
504
Format
505
Text
506
Panel
507
Topic
508
Control
509
Format
510
Movie
511
Control
512
Panel
513
Text
514
Topic
515
Database
500

Figure 5c

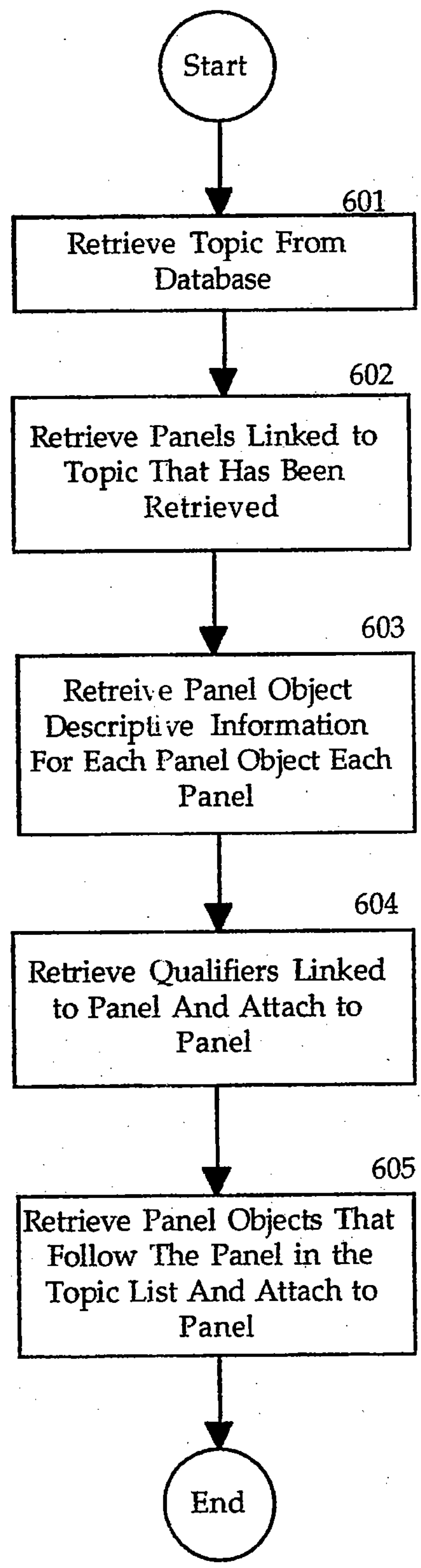
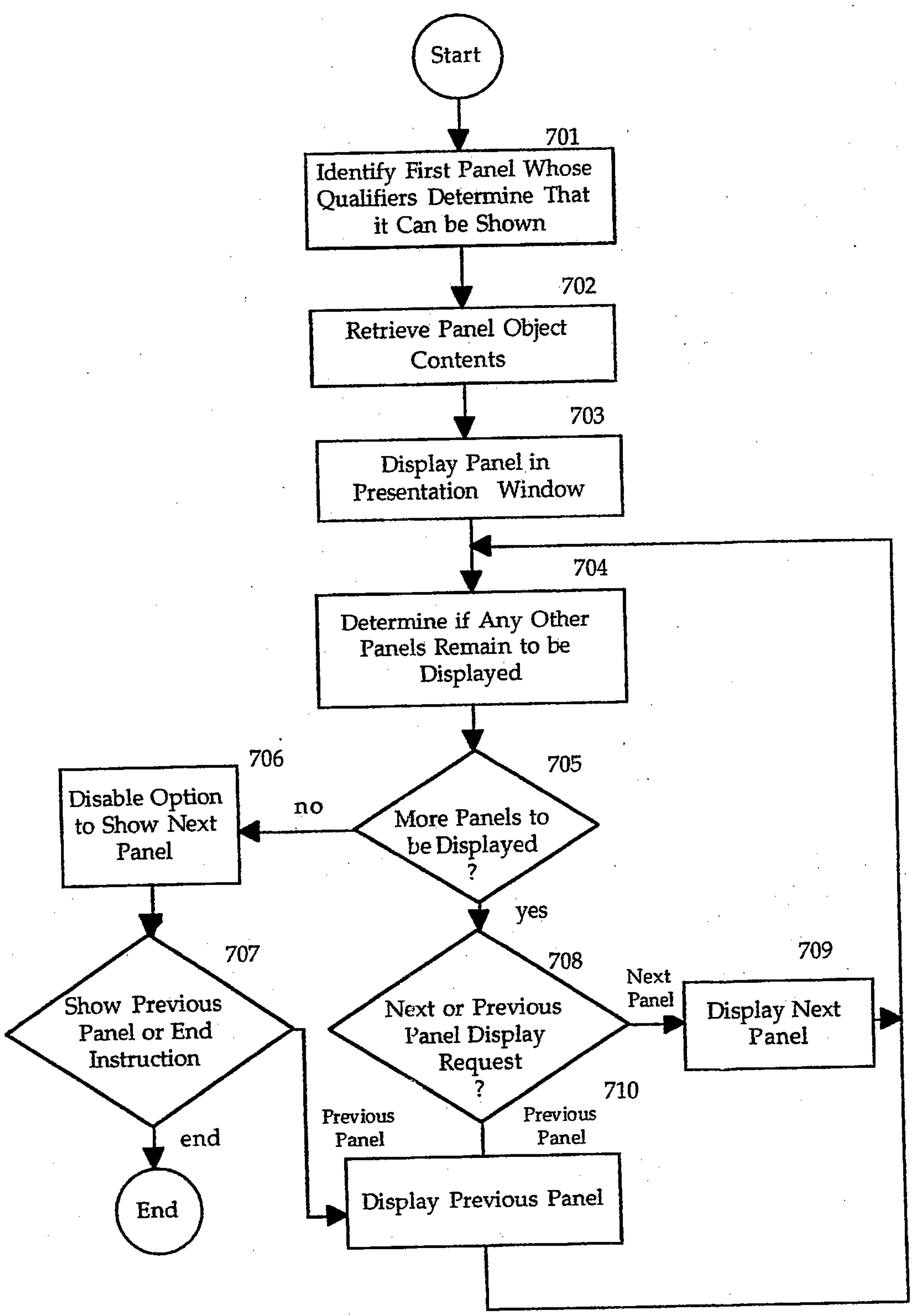
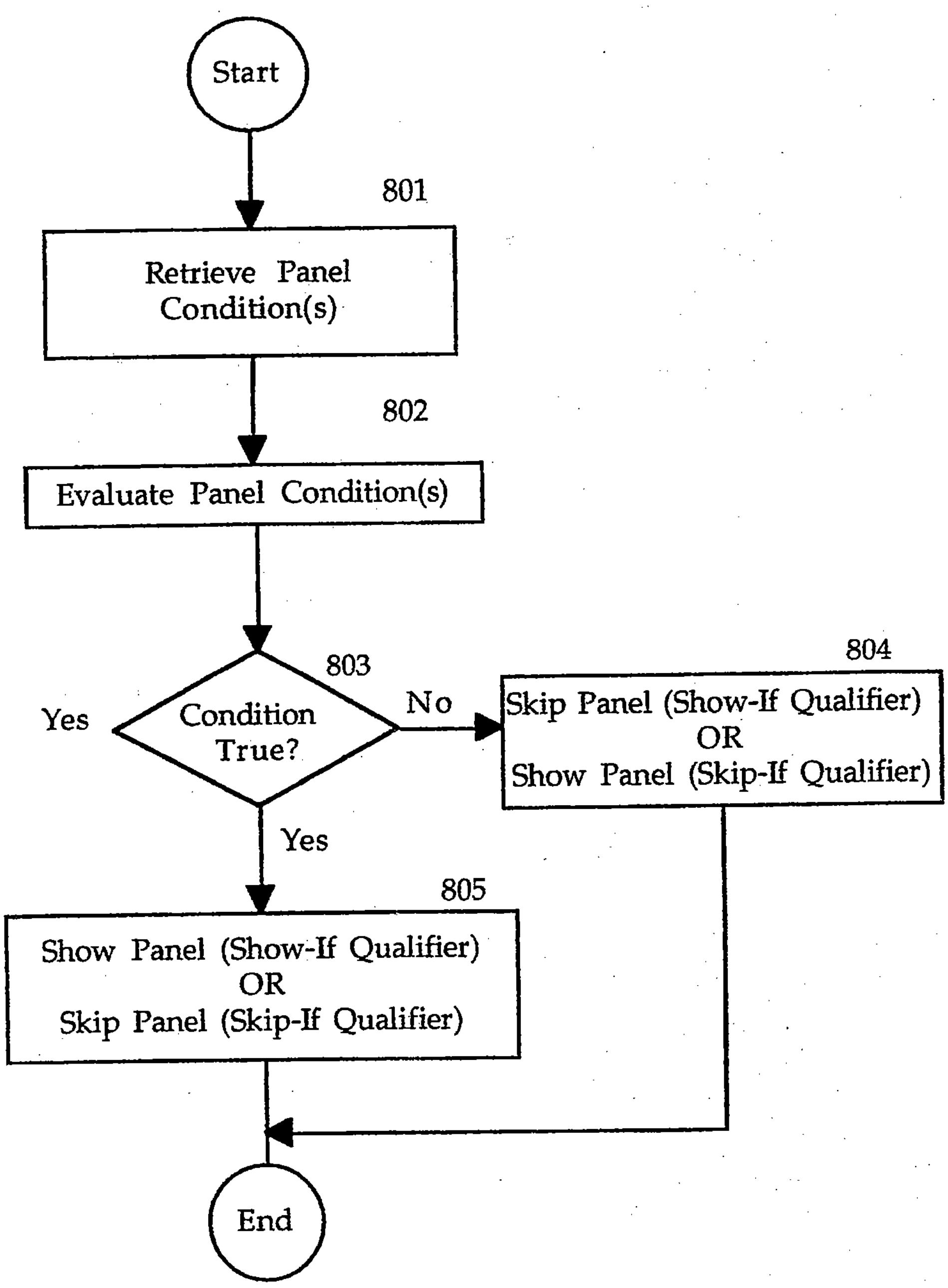

Start
601
Retrieve Topic From Database
602
Retrieve Panels Linked to Topic That Has Been Retrieved
603
Retreive Panel Object Descriptive Information For Each Panel Object Each Panel
604
Retrieve Qualifiers Linked to Panel And Attach to Panel
605
Retrieve Panel Objects That Follow The Panel in the Topic List And Attach to Panel
End

Figure 6



Start
701
Identify First Panel Whose Qualifiers Determine That it Can be Shown
702
Retrieve Panel Object Contents
703
Display Panel in Presentation Window
704
Determine if Any Other Panels Remain to be Displayed
705
More Panels to be Displayed ?
no
706
Disable Option to Show Next Panel
707
Show Previous Panel or End Instruction
end
End
yes
708
Next or Previous Panel Display Request ?
Next Panel
709
Display Next Panel
710
Previous Panel
Previous Panel
Display Previous Panel

Figure 7



Start
801
Retrieve Panel Condition(s)
802
Evaluate Panel Condition(s)
803
Condition True?
Yes
No
804
Skip Panel (Show-If Qualifier)
OR
Show Panel (Skip-If Qualifier)
Yes
805
Show Panel (Show-If Qualifier)
OR
Skip Panel (Skip-If Qualifier)
End

Figure 8

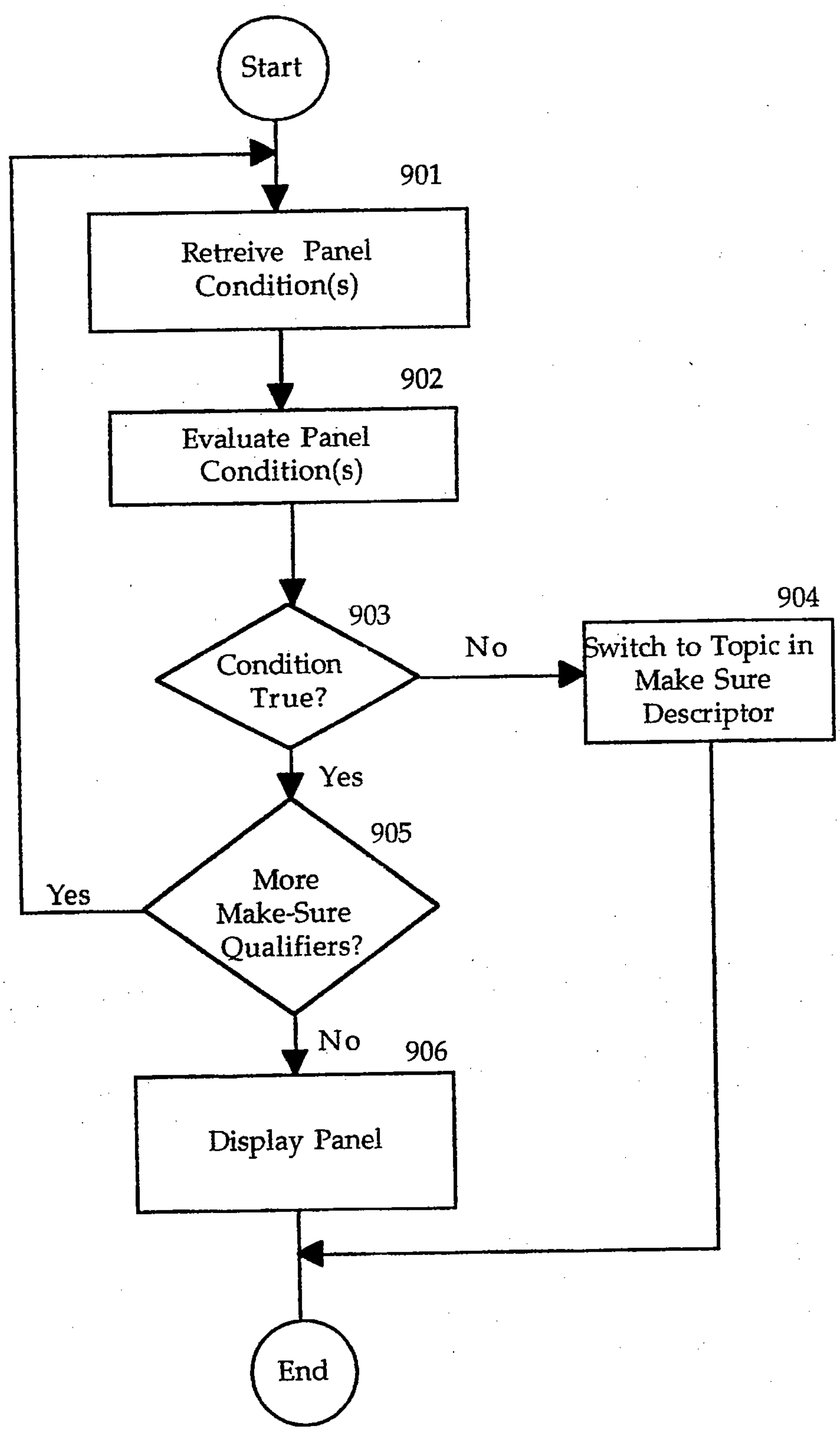

Start
901
Retreive Panel Condition(s)
902
Evaluate Panel Condition(s)
903
Condition True?
No
904
Switch to Topic in Make Sure Descriptor
Yes
905
More Make-Sure Qualifiers?
Yes
No
906
Display Panel
End

Figure 9

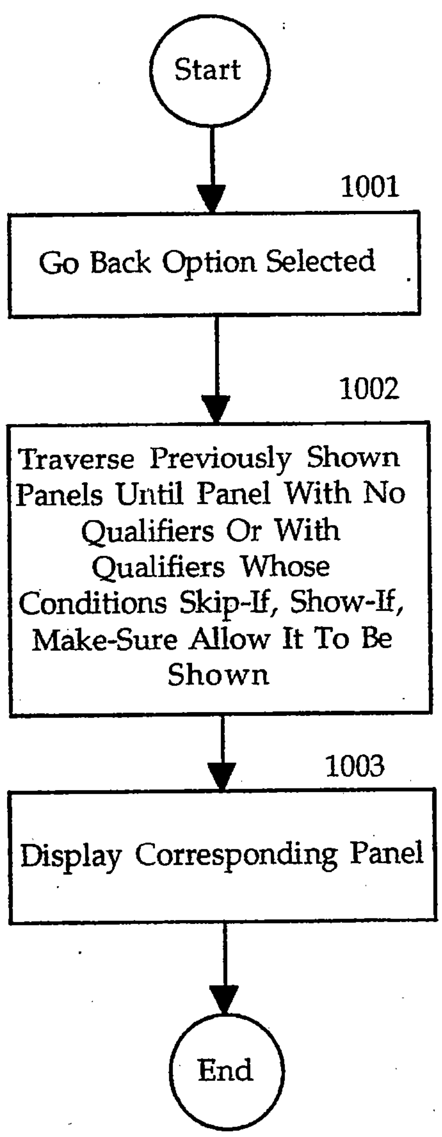

Start
1001
Go Back Option Selected
1002
Traverse Previously Shown Panels Until Panel With No Qualifiers Or With Qualifiers Whose Conditions Skip-If, Show-If, Make-Sure Allow It To Be Shown
1003
Display Corresponding Panel
End

Figure 10

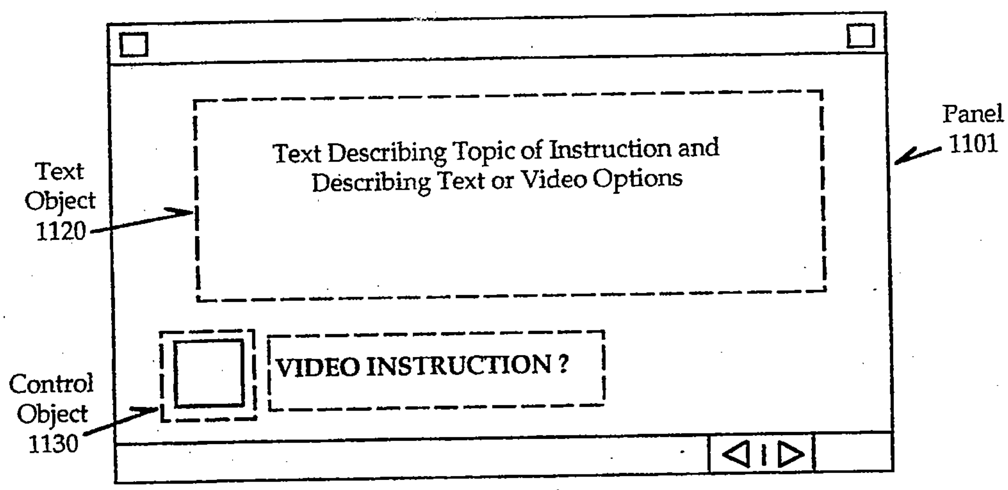


Figure 11a

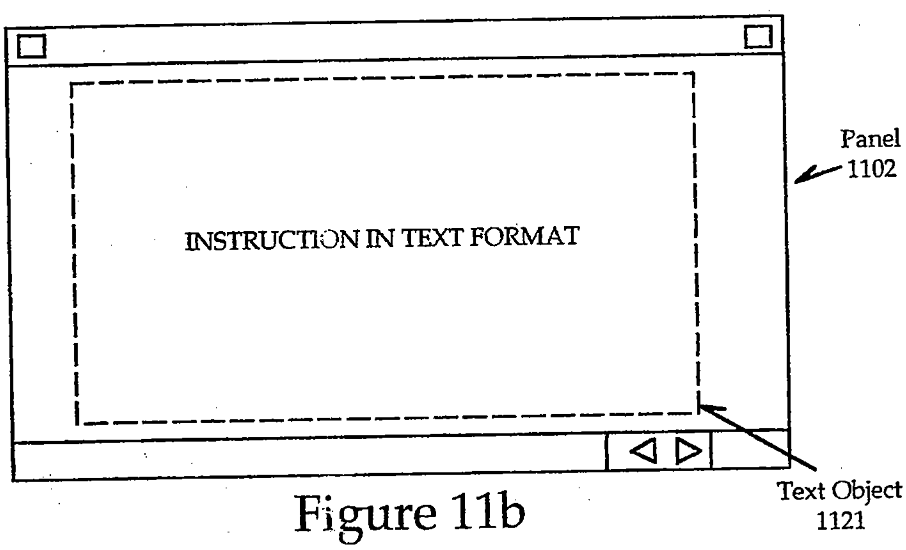


Figure 11b

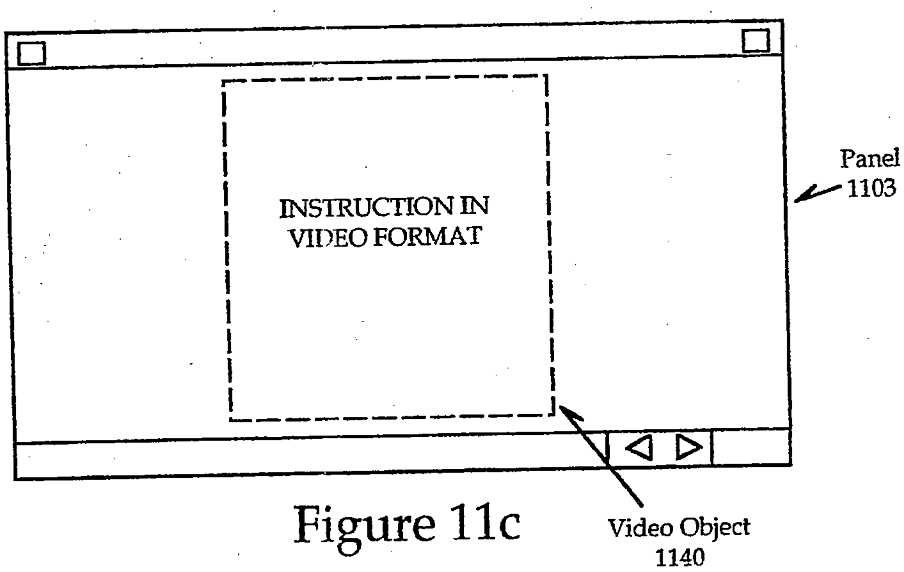


Figure 11c

LINE NO.	OBJECT DEFINITION				COMMENTS
1.	acPT		1100		This is the Topic object
2.	(text)		1120		Text content for Panel 1101
3.	(text)		1121		Text content for Panel 1102
4.	(video)		1140		Video content for Panel 1103
5.	ppUA		1101		Definition of Panel 1101
6.		pcTX		1120	Reference to text object for Panel 1101
7.		pcCI		1130	Reference to checkbox for Panel 1101
8.	pcCI		1130 = checkbox ='VIDEO INSTRUCTIONS?'		Definition of checkbox control object
9.	ppUA		1102		Definition of panel 1102
10.		pcTF		1121	Reference to text object for Panel 1102
11.		pqCS		1160	Reference to Skip-If qualifier
12.	pqCS		1160		Definition of Skip-If qualifier
13.		pcCI		1130	Reference to checkbox as a condition
14.	ppUA		1103		Definition of panel 1103
15.		pcMC		1140	Reference to video object for Panel 1103
16.		pqCN		1170	Reference to Show-If qualifier
17.	pqCN		1170		Definition of Show-If qualifier
18.		pcCI		1130	Reference to checkbox as a condition

Figure 12

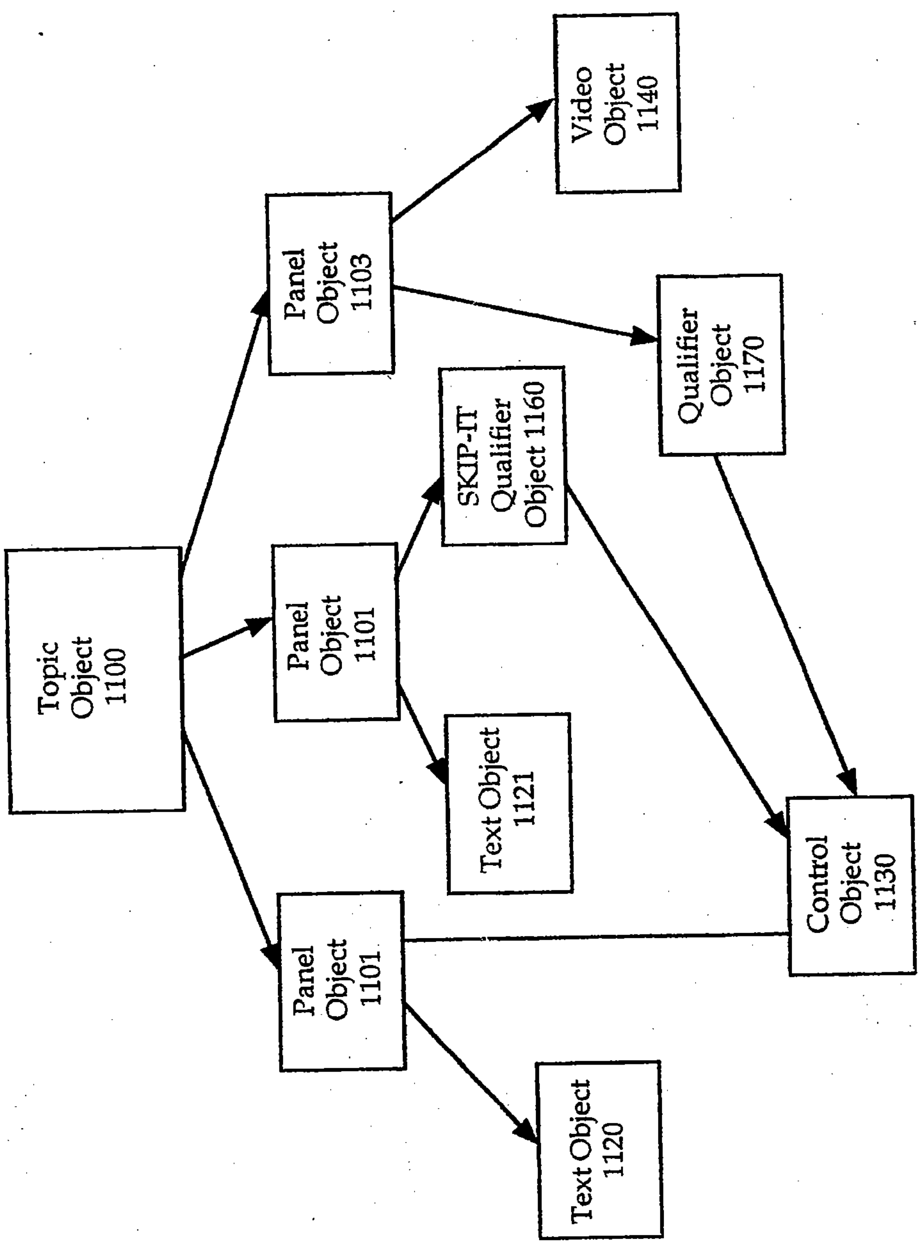

Topic Object 1100
Panel Object 1103
Panel Object 1101
Panel Object 1101
Video Object 1140
Qualifier Object 1170
SKIP-IT Qualifier Object 1160
Text Object 1121
Text Object 1120
Control Object 1130

Figure 13

INTERACTIVE MULTIMEDIA DELIVERY ENGINE

BACKGROUND OF THE INVENTION

1. Field of The Invention

The present invention relates to the field of storage and retrieval of data, in particular, to organizing and storing multimedia data for use in interactive data delivery and display systems such as those used for instructional or educational purposes.

2. Description of The Related Art

As use of computer systems become more widespread, the desire to utilize multimedia has become more immediate. Multimedia refers to the integrated use of text, graphics, video and audio information media's. In order to effectively use multimedia, mechanisms for delivering the data to the user must be developed. For example, in order to create an interactive instructional application, various screens or panels are created which contain the instructional information. The panels must be logically linked and displayed in an order corresponding to the state of the interactive instruction.

Two known systems provide interactive multimedia delivery capabilities; Macro Media Director and HyperCard®. MacroMedia Director, a product of Macro Media Inc., use a score, stage, and cast metaphor to deliver interactive multimedia. A score (timeline) is used to place cast members (multimedia objects) on the stage (display). Objects can be layered visually by their placement along the vertical axis of the score and temporally by their placement along the horizontal axis. What the user sees at any point in time is represented by a column in the score. The flow of delivery (i.e. instruction) is controlled by the score. Scripts can be attached to cast members and the score to alter the flow of delivery. Scripts require processing external to the flow of delivery. Further, subsequent alterations to the flow are cumbersome.

HyperCard, a product of Apple® Computer, Inc. uses a notecard and object metaphor to deliver interactive multimedia instruction. Visual objects are placed on cards. The cards are then ordered. The flow of delivery is controlled by the card order. Scripts can be attached to visual objects and cards to alter the flow of instruction. A HyperCard card is equivalent to a column in the Macromedia Director score. A HyperCard object is equivalent to a cast member in MacroMedia Director. Both elements in both products can have scripts attached.

The prior art systems have various deficiencies. First, the flow of data delivery may only be altered via scripting. It would be desirable to be able to alter the flow of instruction inherently within the design of the instruction. Further, there is no backtracking once a flow of instruction has commenced.

Accordingly, it would be desirable to have an interactive multimedia delivery system which overcomes the foregoing deficiencies of the prior art.

SUMMARY

A delivery engine for interactive presentation and display of multimedia data is disclosed. The delivery mechanism is premised on a topic and panel metaphor. A topic is comprised of panels. Each panel defines the multimedia data which will be used when the panel is displayed. Each panel may also contain control and qualifier elements which are used to determine the next panel to be displayed. Control elements are used to solicit information from the user which is then used by the qualifier elements in determining the next panel to be displayed.

Each element is stored in a database as a discrete object. This allows re-use of objects amongst different panels or topics. Prior to delivery of the data, the topic must be set-up. Set-up occurs by retrieving all the necessary panel, control qualifier and data descriptor objects. The actual data associated with data media objects is only retrieved when the data is to be displayed. The various qualifier objects are evaluated while panels are being viewed.

The database is generated via an authoring step. In the authoring step, the topic and the various elements are combined to create a flow for the presentation of the multimedia data. Once the authoring is completed, the elements are processed in a manner similar to compiling a software program, to create the database.

BRIEF DESCRIPTION OF THE FIGURE

FIG. 1 is a block diagram illustrating the components of a computer system as may be utilized in the currently preferred embodiment of the present invention.

FIG. 2 is a block diagram illustrating the components of the delivery engine of the currently preferred embodiment of the present invention.

FIG. 3 is a diagram which illustrates the logical relationship amongst the database objects of the currently preferred embodiment of the present invention.

FIG. 4 illustrates a Presentation Window as may be utilized in the currently preferred embodiment of the present invention "© 1993 Apple Computer, Inc." (17 U.S.C. 401).

FIG. 5*a* is a block diagram of a database as generated in the currently preferred embodiment of the present invention.

FIG. 5*b* illustrates the linkage between a topic and corresponding panels of the database of FIG. 5*a*.

FIG. 5*c* illustrates the linkage between a panel and corresponding visual objects and qualifiers of the database of FIG. 5*a*.

FIG. 6 is a flowchart which illustrates the steps retrieving objects for setting up a topic for delivery in the currently preferred embodiment of the present invention.

FIG. 7 is a flowchart which illustrates the steps retrieving objects for delivery of panels in the currently preferred embodiment of the present invention.

FIG. 8 is a flowchart which illustrates the steps taken during the evaluation of a Show_If or Skip_If qualifier in the currently preferred embodiment of the present invention.

FIG. 9 is a flowchart which illustrates the steps undertaken for evaluating a Make-Sure qualifier in the currently preferred embodiment of the present invention.

FIG. 10 is a flowchart which illustrates the steps taken by the delivery engine responsive to a user invoking a Go-Back option in the currently preferred embodiment of the present invention.

FIGS. 11a–11c represent an example of a topic comprising three panels that may be utilized in the currently preferred embodiment of the present invention "1993 Apple Computer, Inc." (17 U.S.C. 401).

FIG. 12 is a listing of object definitions for implementing the topic illustrated in the example of FIGS. 11a–11c.

FIG. 13 illustrates the linkage between the objects of the example of FIGS. 11–11c.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

This specification is related to the following specifications which are assigned to the same assignee, Apple Computer, Inc.:

Ser. No. 08/010,063, filed Jan. 27, 1993, entitled "Method and Apparatus For Providing A Help Based Window System Using Multiple Access Methods"; Ser. No. 08/010,061, filed Jan. 27, 1993, entitled "Method and Apparatus For Displaying And Scrolling Data In A Window-Based Graphic User Interface"; Ser. No. 08/010,064 filed Jan. 27, 1993, entitled "Method and Apparatus For Presenting Information In A Display Using Floating Windows"; and Ser. No. 08/010,062, filed Jan. 27, 1993, entitled "Method and Apparatus For Providing Visual Cues In A Graphic User Interface" all still pending.

An interactive multimedia instructional delivery engine is described. In the following description, numerous specific details are set forth such as coding examples, in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known circuits, for displaying text, graphics and video data have not been shown in detail in order not to unnecessarily obscure the present invention.

Overview of the Computer System of the Preferred Embodiment

The computer system of the preferred embodiment is described with reference to FIG. 1. The present invention is preferably implemented on a general purpose microcomputer such as one of the members of the Apple® Macintosh® family of computers. In any event, a computer system as may be utilized by the preferred embodiment generally comprises a bus or other communication means 101 for communicating information, a processing means 102 coupled with said bus 101 for processing information, a random access memory (RAM) or other storage device 103 (commonly referred to as a main memory) coupled with said bus 101 for storing information and instructions for said processor 102, a read only memory (ROM) or other static storage device 104 coupled with said bus 101 for storing static information and instructions for said processor 102, a data storage device 105, such as a magnetic disk and disk drive, coupled with said bus 101 for storing information and instructions, an alphanumeric input device 106 including alphanumeric and other keys coupled to said bus 101 for communicating information and command selections to said processor 102, a cursor control device 107, such as a mouse, track-ball, cursor control keys, etc., coupled to said bus 101 for communicating information and command selections to said processor 102 and for controlling cursor movement, and a display device 108 coupled to bus 101 for displaying textual, graphical or video output. Additionally, it is useful if the system includes a hardcopy device 109, such as a printer, for providing permanent copies of information. The hardcopy device 109 is coupled with the processor 102 through bus 101. The computer system of FIG. 1 may also include means for processing and re-playing audio data (not illustrated).

The computer system illustrated in FIG. 1 is one that generally provides a window based graphical user interface which is well known (for example the Apple Macintosh family of computers, available from Apple Computers, Inc.), and hence, a detailed description is not necessary herein. In operation, the window based user interface of the currently preferred embodiment is generated and displayed using software stored in the RAM 103 or ROM 104 and executed by the processor 102, as is known.

As will become apparent in the description below, various computer elements provide the functionality described. For example, a Presentation Window would be displayed on the display device 108. The processor 102 would provide processing resources for the object generation means, evaluation of qualifiers, and other processing means comprising the delivery engine. The generated database would be stored in the data storage device 105 and loaded into the RAM 103 during the delivery process.

In the currently preferred embodiment the multimedia delivery engine is implemented as part of a Help facility for providing assistance in the operation of the computer system. It will be apparent from the description below, that the present invention's features and attributes have application in a wide variety of areas, e.g. for interactive presentations, instruction, game development or reference guides. Accordingly, the present invention is not limited by the following description of the currently preferred embodiment, and it will be understood by those skilled in the art, that the present invention may be applied to a computer systems offered by a variety of manufacturers, and to systems having both instructional and non-instructional uses.

Overview of the Delivery Engine

The delivery engine provides for the presentation of information using multiple media's. The currently preferred embodiment is in a Help system which provides user instruction. The instruction is organized as a series of topics. Each topic contains general topic information and references to panels. Each panel contains references to visual media such as text, graphics, video, as well as control and qualifier objects. The topics, panels, visual media, and qualifiers are all encapsulated as individual entities in a database. That is they are all objects in a database. FIG. 2 illustrates the basic components of the delivery engine and a presentation window of the currently preferred embodiment. The presentation window is a pre-defined window having various display controls and which are used for displaying the panel's visual media data. Referring to FIG. 2, a presentation window

201 is coupled to a database retrieval engine 202. The presentation window of the currently preferred embodiment is described in greater detail below. The database retrieval engine 202 operates responsive to current and prior operator input to retrieve from database 203, the next data to be displayed.

The database 203 contains a plurality of objects comprised of topic objects 204, panel objects 205 and panel objects 206. Generally speaking, the various objects are defined as follows:

Topic	A series of Panels.
Panel	A collection of Panel Objects, Qualifiers and a display format which comprise what is seen by the user.
Panel Objects	Text, graphics and video data, controls, qualifiers and formats.
Controls	Standard Macintosh control items such as push-buttons, radio buttons and check boxes. Radio buttons and check boxes are used to get user input for conditional tests for qualifiers.
Qualifiers	Condition which are used to determine which panels will be displayed to the user.
Formats	Optional descriptors of how panel objects are displayed.
Movies	Animated graphical images with or without synchronized sound.

In the currently preferred embodiment, objects may be used by more than one other object. This notion of re-use is one of the aspects of what is commonly referred to as object oriented programming. For example, a single item of text data may be used by multiple panels. The objects listed above are not meant to be exhaustive or limiting. Other objects could be implemented without causing departure from the spirit and scope of the present invention.

In the currently preferred embodiment, the C++ programming language is used. C++ is used because of it's inherent capability to support object oriented programming. However, it would be apparent to one skilled in the art to use alternative programming languages such as ADA or Modula-2, would not cause departure from the spirit and scope of the present invention.

FIG. 3 illustrates the logical relationships between the objects in a database. Referring to FIG. 3 a topic 301 points to a series of panels, panel 1 302, panel 2, 303 and panel n 304. In FIG. 3, the panel 1 302 is linked to qualifier 305, graphics data 306, control 307, video data 308 and text data 309. Depending on the contents a particular panel may have none, one or multiples of the foregoing panel objects. Referring back to FIG. 3, the panel 2 303 is also linked to text data 309. Thus the text data 309 is displayed when either panel 1 302 or panel 2 303, is displayed.

Presentation Window

A presentation window as may be utilized in the currently preferred embodiment is illustrated in FIG. 4. Referring to FIG. 4, a presentation window 401 includes a topic title area 402 and a panel visual data area 403. The topic title area 402 will contain the name of the topic being viewed. The panel visual data area 403 will contain the visual data that is associated with the panel being viewed. It is the panel visual data area 403 to which the data delivery engine will provide data for display. In the currently preferred embodiment, the panel visual data area 403 is 340 pixels wide. The length is determined by the objects on the panel and the size of the screen. A panel with lots of text will be longer than a panel with little text. The panel consists of any number of graphic, text, prompt, control and video objects. The only limit to the number of objects on a panel is the RAM available.

The objects are drawn in the panel in the order specified in the panel object. As a result, objects later in the sequence can be made to overlay previous objects in the sequence. Text can be positioned in four different ways on a panel; tag, body, full panel and author positioned. Tag, body and full panel text are positioned by the delivery engine. Tag text is typically right justified in a bold font type. Body text is typically left justified with a normal font type. Full panel text is used when only text will appear on the panel. Additional formats may be created, stored in the database, and used. Text, graphic, video (movie) and control data positions are determined by parameters in the object definition stored in the database.

The presentation window 401 further contains a presentation window control area 404. The control area 404 provides means for traversing through the instruction path and includes a previous panel button 405 and a next panel button 407. Activation of these buttons, typically by a point and click operation, causes backward (button 405) and forward (button 407) traversal through the series of panels comprising the topic. A current panel number indicator 406 allows the user to know what panel that they are currently viewing. Up to three (3) additional controls may be added to the control area 404, e.g. the control button 408. The controls may be for altering the flow of the presentation of panels (e.g. switching topics.) These optional controls are stored in the database. The topic defines whether or not they should appear. The delivery engine retrieves and places the optional controls in the control area 404.

Finally, the presentation window 401 includes window control areas that are familiar to those who use window based graphical user interfaces. Window drag bar 410 allows the presentation window to be re-located via a cursor drag operation. Window re-sizing box 411 allows the size of the window to be shrunk/expanded as desired. Window close box 409 provides a quick means for closing the window and thus terminating the instruction (help) session.

Database Overview

The present invention operates responsive to initial user queries for data. Once the initial query is made, the present invention operates to retrieve data for display in a presentation window. The relationship of the various objects in the database is further illustrated in FIGS. 5a-5c.

Referring to FIG. 5a, a database 500 is illustrated with Topics 501, 508 and 515, Graphic object, 502, Qualifier Object 503, Button 504, Panels 507 and 513, Control object 509 and 512, Format objects 505 and 510 and Text Objects 506 and 514. FIG. 5b illustrates a linkage of Topic 501 with panels 507 and 513 and button 504. Finally, FIG. 5c illustrates the linkage of Panel 507 and graphic object 502, Qualifier 503, Control Object 509 and Text Object 514. The manner in which these linkages are made will become apparent in the description below.

Generating the Database

As described above, the database is comprised of a plurality of data objects. Generation of the database is essentially a two step process; authoring and object generation. This is analogous to writing and compiling a software program. Authoring is similar to writing a program or a script. In this case, authoring is used to generate the topic contents. The topic contents are then provided to an object generation means for creation of the objects for entry into the database. The object generation means performs a step that is similar to compilation. The database itself is analogous to an object file created during compilation.

In the currently preferred embodiment, various authoring tools may be used. In one instance the topic contents may be generated as a set of instructions. The topic contents may also be generated via a What You See Is What You Get (WYSIWIG) tool. In this case, the panels are generated as they would be displayed. Of course different object generation means would be used for each instance. Other techniques for creating the topic contents used to generate the database would be apparent to one skilled in the art and would not depart from the spirit and scope of the present invention.

The database is conceptually comprised of resources and data. The resources are the data objects. The data is the actual text, graphical or video data that is displayed. Data objects are retrieved by an object type and an identification number ("id"). In the currently preferred embodiment, the object type is a 4-character string and the object id is an integer number. Table A lists a set of "building-block" data objects that are available for the presentation window in the currently preferred embodiment of the present invention. The list is not meant to be exhaustive, but merely representative of commonly used objects.

TABLE A

DATA OBJECTS LIST

Category	Type	Label	Description
Topic	acPT	kResTopic	Topic script
Panel	ppUA	kResPanel	Panel
Content	pcTX	kResSTxt	Text
Content	pcTF	kResSTxtF	Full panel text
Content	pcTB	kResSTxtB	Body text
Content	pcTT	kResSTxtT	Tag text
Content	pcGA	kResGraphic	Graphic
Content	pcGF	kResGraphicF	Full panel graphic
Content	pcPR	kResPrompt	Prompt
Content	pcCI	kResCtrlItem	Control item
Content	pcMO	kResMovie	Movie
Content	pcMC	kResMovieC	Movie with controller
Qualifier	pqCN	kShowIf	Show panel if conditions are true.
Qualifier	pqCS	kSkipIf	Skip panel if conditions are true.
Qualifier	pqMS	kResMakeSure	Make sure the conditions are true before showing this panel. If not, show the Oops topic.
Qualifier	pqCN	kResCondition	Condition
Context	pxCA	kResContext	The specification for a context check.

In the currently preferred embodiment, each object is defined as a Macintosh resource. A Macintosh resource is code or data that may be used by various applications. Macintosh resources as described in the publication entitled "Inside Macintosh Volume 1" published by Addison-Wesley Publishing Company, Inc. The actual resource definition for many of the objects described in Table A is provided in Appendix A. These resource definitions are exemplary of how the resource may be defined and is not meant to be exhaustive.

Referring back to Table A, the first column Category indicates the general category of the data object, namely, Topic, Panel, Content, Action, Qualifier or Context. The second column Type lists the object type. The third column is the object label. The object label is an alternative way to reference the object and is used by the authoring tools and by the delivery engine implementation code (C++). Finally, the fourth column is a brief description of the use for the particular object.

Operation of the Delivery Engine

In the currently preferred of the present invention, elements are used only when they are needed. Before starting the instruction (displaying panels), the delivery engine must set-up the topic to be viewed. The steps for setting-up a topic are described with respect to the flowchart of FIG. 6. Referring to FIG. 6, the topic is retrieved from the database and is instantiated as an object, step **601**. By instantiating as an object it is meant to create an instance of its class. The topic object contains general topic information and a list of panels associated with the topic. Each panel that is linked to the topic (i.e. in the topic's list of panels) is then retrieved from the database and also instant jared as an object, step **602**. The retrieved panels each contain a list of panel objects. Each panel object is then retrieved from the database and instantiated as an object, step **603**. Only the panel object description itself is retrieved and instant jared, not the content of the panel object. For example, a text panel object contains a reference to the text content and the location or format of the text on the panel. The text content is not retrieved until the panel is displayed.

Next, each qualifier linked to the panel is retrieved and instantiated as an object, step **604**. The qualifier is attached to the panel object as a collaborator. A collaborator is an independent object that works to support, aid or assist another object. Finally, if any panel objects follow the panel in the topic list (as distinguished from the panel list), they are retrieved and instantiated as objects, step **605**. These objects are appended to the panel that was previously instantiated. This allows a panel descriptor to be modified within the topic. For example, prompts can be appended to a panel from the topic. At this point the topic of instruction is ready to be delivered.

The delivery of panels in the during the course of instruction is described with respect to the steps of the flowchart in Figure. Referring to FIG. 7, once all the panels and panel objects have been retrieved and instantiated, the delivery sequence begins with the first panel whose qualifiers determine that it can be shown or which has no qualifiers step **701**. The manner in which the qualifiers are examined is described below with reference to FIGS. 8 and 9. Here it is assumed that a first panel has been identified. The panel object contents are then retrieved, step **702**. As noted above, only the panel object descriptors were initially retrieved. The panel object contents are the text, graphics or video data that is to be displayed on the panel. Once the panel contents are retrieved, they are displayed in the panel display area of the presentation window, step **703**. In this display step, the display of the objects may be formatted by the author or by the delivery engine by the author's choice of object type for the data.

At this point, a panel is displayed and the author may provide the user with various options. One option that an author may provide to the user is a go-back option. In the go-back option the user may jump back to a previously displayed panel. The go-back option is described below in more detail with respect to FIG. 10. The other options are to traverse backwards or forwards through the current list of panels which is described as follows. These options are provided as part of the presentation window functionality (see FIG. 4). Referring back to FIG. 7, a determination is made by the delivery engine whether or not there are any more forward (next) panels to be displayed, steps 704, 705. This determination is made transparent to the user while they are examining the current panel. If no more panels are to be displayed in the forward direction, the "next" option is disabled, step 706. In this instance, the user options are limited to showing a previous panel or to end the instruction, step 707. If the user chooses to end instruction, the instruction is terminated. Otherwise, the previous panel is displayed, step 710.

Responding in the affirmative to the question of more panels to be displayed of step 705 causes the "next" and "previous" options to be enabled. The user must respond by selecting one of the options, step 708. If the "next" option is selected, the next eligible panel is displayed, step 709. The next eligible panel is identified by the state of the qualifiers associated with the next successive panels (or by the lack of qualifiers for a panel). The delivery engine then repeats the process by determining if any other panels remain to be displayed (step 704). If the previous option is selected, the previous panel is displayed, step 710 and the process repeats with the delivery engine determining if any other panels remain to be displayed (step 704).

From the foregoing description it is apparent that the delivery engine maintains a display list of panels that have been displayed. This display list facilitates the traversing of the panels using the go-back option and the make-sure qualifier (as will be described below).

Evaluation of Qualifiers

The qualifiers that may be associated with a panel facilitate the interactive nature in which the panels of a topic may be displayed. The qualifiers and the conditions referred to therein provide for control of its flow of panel display. Qualifiers are evaluated by the delivery engines before each panel is shown and in the background while the user is not interacting with the delivery engine. Qualifiers can be based on user controls within any panel, not necessarily the one being currently shown, or on a test of the user's context. For example, if the instruction is to display a short cut technique for performing a function or a long way technique for performing a function, a user may be given the option of deciding which way they want to learn to perform the function in a panel X. Assume that the panels for the instruction are linked sequentially as panels X through panels X+m. Further assume that the panels X+1 to X+n show the short cut technique, while the panels X+n +1 to X+m show the long way technique. If the user at panel X chooses the to proceed by learning the long way technique, the panels X+1 to X+n (corresponding to the short cut technique) would have qualifiers which would cause them to be skipped.

The evaluation for the show-if and skip-if qualifiers is described with reference to FIG. 8. Referring to FIG. 8, if the panel has a Show-If or Skip-If qualifier, retrieve the condition for the qualifier from the database and instantiate it as an object, step 801. The condition is then evaluated, steps 802, 803. The condition may consist of a user control, a context check, or another condition. These elements are paired with and/or logic (author's choice) and can be nested indefinitely. Hence, the Show-If or Skip-If qualifier is a binary decision tree. The tree is evaluated. If the eventual result is false, the panel will be skipped (Show-If) or shown (Skip-If), step 804. If the eventual result is true, the panel will be shown (Show-If) or skipped (Skip-If), step 805.

A third qualifier is the Make-Sure qualifier. The Make-Sure qualifier provides for the switching of topics (e.g. for remedial instruction or a notice that the user must go back to a prior step) and as the criteria checked for in a go-back operation. The former instance of the Make-Sure qualifier is described with reference to FIG. 9. First, the conditions for the qualifier are retrieved from the database and instantiated it as an object, step 901. The retrieved panel conditions are then evaluated, step 902 and 903. The condition is evaluated in the same way as the Show-If or Skip-If, but the result is used differently. If the result is false, another topic is automatically shown, step 904. The identification of which topic is to be shown is contained in the Make-Sure descriptor (See Appendix A for description). The first Make-Sure qualifier that yields a false result has its topic shown. If the result is true, then a check is made for additional Make-Sure qualifiers, step 905. If more Make-Sure qualifiers exist for the panel, they are evaluated per step 901. If there are no further Make-Surer qualifiers and all the Make-Sure qualifiers associated with the panel have evaluated to true, the panel may be shown, step 906.

The Make-Sure qualifier as used with the go-back option is described with reference to FIG. 10. Referring to FIG. 10, the user is presented with a "go-back" option by the author and the user elects to take it, step 1001. In this instance, the panels previously shown are searched in reverse order until a panel is found with no qualifiers or with qualifiers whose conditions (Skip-If, Show-IF, Make-Sure) allow it to be shown, step 1002. The panel is then displayed, step 1003. The Make-Sure qualifier allows an author to specify a set of conditions (a Make-Sure) which, if changed, automatically provide a warning. The engine can then search back to find where in the sequence the conditions were acceptable, allowing the user an opportunity to redo the conditions correctly.

Context checking is an optional element of a conditional operation. Note that a condition can specify one or more combinations of the following:

A panel control item such as a radio button or checkbox
A context-check specification
Other conditions.

A context-check specification is a resource (object) which, when evaluated in light of the user's context, results in a true or false. For example, checking for an open "window" is a context-check specification. Other context checks which can be performed in the currently preferred embodiment include:

Window <title> is open and front;
Window <title> is open but invisible;
Application <name> is open <running>;
Application <name> is open and in front.

Switching of Topics

As noted above, topics may be switched via the Make-Sure option of a panel. Moreover, in the currently preferred embodiment, multiple topics can be shown simultaneously. The author may elect to present a new topic while still displaying the old topic, close the old topic and replace it with a new one, or hide the old topic and show a new one (called an "Oops" topic). A hidden topic is automatically shown again when the "Oops" topic is closed. This is made possible by a displayed panel list maintained by the delivery engine.

Of course, whenever a topic switch the topic set-up described with reference to FIG. 6, must be performed.

Example

An example of the present invention is now provided. FIGS. **11***a*–**11***c* are the panels for a topic to be authored. In this example, instruction for performing certain operations on a computer system is being provided. Here, the instruction may alternatively be provided via textual instructions or via a video showing the instruction being carried out. So if the instruction was for closing a window, the user has the option of reading how to maneuver the cursor to point to a close box and then clicking a button on the cursor control device, or simply watching the action being performed.

Referring to FIG. **11***a*, a panel **1101** has two objects; a text object **1120** and a control object **1130**. The text object **1120** will contain the text which describes the topic of instruction. The control object **1130** is a checkbox by which a user may specify that they desire video instruction. FIG. **11***b* illustrates a panel **1102** which has one object namely a full panel text object **1121** which describes how to perform the desired operation. FIG. **11***c* illustrates a panel **1103** which has one object, namely a video object which is a move which shows the desired operation being carried out.

Assuming that topic set-up has occurred, the panel **1101** will be displayed. The user is presented with the option of having the instruction by text or video. Video instruction is specified by checking the VIDEO INSTRUCTION checkbox. Upon going to the next panel via the panel traversal operators of the presentation window, the panel **1102** will be displayed if the checkbox is not checked and the panel **1103** will be displayed if the checkbox is checked.

FIG. **12** contains pseudo-code exemplary of how the foregoing example may be encoded using the objects described in Table A above. Referring to FIG. **12**, line **1** contains the topic object definition. Lines **2**–**4** contain the definition of the panel object contents, i.e. the text and the video data. Line **5** starts the definition of panel **1101**. The objects associated with panel **1101** are on lines **6** and **7** (indented for clarity), a text object and checkbox object, respectively. Line **8** contains the definition for the checkbox control object. Note that whereas the checkbox object is referenced by panel **1101**, it is defined outside of the object. This allows the checkbox object to be utilized my multiple panels. Further note that the panel object contents may also be defined within the panel definition in which the object is used.

Line **9** begins the definition of panel **1102**. The panel **1102** includes a text object and Skip-If qualifier object, lines **10** and **11**, respectively. At line **12**, the qualifier object linked by panel **1102** is defined. Note that the Skip-If qualifier object links to the checkbox object as a condition (line **13**).

Line **14** begins the definition of panel **1103**. The panel **1103** includes a video object and a Show-If Qualifier object, lines **15** and **16**, respectively. Finally, the Show-If qualifier object is defined at lines **17**-**18**. As in the case of the Skip-If qualifier, the Show-If qualifier links to the checkbox object as a condition (line **18**).

FIG. **13** illustrates the linkages between the objects in the example. Referring to FIG. **13**, the topic **1100** would be linked to the panels **1101**, **1102** and **1103**. The panel **1101** would have links to the text **1120** and control **1130**. As is apparent, multiple objects will link to the control **1130**. The panel **1102** links to text object **1121** and Skip-IF qualifier **1160**. The qualifier **1160** in turn links to the control **1130**. Finally, the panel **1103** links to video object **1140** and to Show-If qualifier **1170**. The Show-If Qualifier in turn links to control **1130**.

During set-up of the topic, the objects are loaded as described above with reference to FIG. 6. Delivery of the panels would then occur in the manner as described with reference to FIGS. 7–10.

Thus, a method and apparatus for interactive delivery of multimedia data for display on a computer system, is described.

I claim:

1. A method for interactive delivery of multimedia data in a computer controlled display system, said method comprising the steps of:

a) authoring a topic, said topic comprised of a series of sequentially-linked panels, media elements, and flow control elements;

b) generating a database from said topic, said database comprised of a plurality of topic, panel, media and flow control objects;

c) setting-up said topic for display by retrieving said topic, panel, media and flow control objects from said database;

d) identifying a first panel to be displayed by performing the following steps:

i) examining a first panel in said series of sequentially linked panels to determine if said first panel has at least one flow control object;

ii) if said first panel in said series of sequentially linked panels does not have a said flow control object, identifying said first panel as the first panel to be displayed;

iii) if said first panel in said series of sequentially linked panels does have a flow control object, evaluating said flow control object:

iv) if said flow control object yields a first state, identifying said first panel as the first panel to be displayed:

v) if said flow control object yields a second state, skipping said panel: and

vi) continuing steps i)–v) for said series of sequentially linked panels until a panel is found which either has no flow control object or said flow control object yields said first state

e) retrieving the media content for said first panel to be displayed;

f) displaying said first panel;

g) identifying a second panel to be displayed by evaluating said flow control objects;

h) retrieving the media content for said second panel to be displayed; and

i) displaying said second panel.

2. The method as recited in claim **1** wherein said step of evaluating said flow control objects is further comprised of the steps of:

a) identifying conditional elements associated with said flow control elements;

b) determining logical values of said conditional elements;

c) performing any necessary logical operations on said logical values to derive a state for said flow control objects.

3. The method as recited in claim 1 wherein said step of identifying a second panel to be displayed by evaluating said flow control objects is further comprised of the steps of:

a) identifying flow control objects associated with a next panel in said series of sequentially linked panels;

b) identifying conditional elements associated with said flow control elements;

c) determining logical values of said conditional elements;

d) performing any necessary logical operations on said logical values to derive a state for said flow control objects;

e) repeating steps a)-d) until a panel having flow control objects in said first state is identified.

4. The method as recited in claim 1 wherein said flow control object further causes a topic switch when said flow control object is evaluated to be in said second state.

5. A computer display system having a central processing unit (CPU) coupled to a display, comprising:

topic authoring means for creating a topic, said topic comprised of a series of panels, each of said panels further comprised of panel elements;

database generation means for generating a database from a topic, said database comprised of a plurality of topics, panels and panel objects wherein said panel objects include video objects, text objects, graphics objects, qualifier objects and condition objects;

topic delivery means for delivering said panels of a topic for display in an order based on user input and further based on context wherein said topic delivery means includes object retrieval means for retrieving objects from said database for setting-up delivery of said topic, qualifier evaluation means for determining a logical state for said qualifier objects, and next panel determination means for determining the next panel to be displayed based on the logical state of said qualifier objects for succeeding panels; and

media content retrieval me,ms for retrieving media data corresponding to panel objects for display on said display.

6. The computer display system as recited in claim 5 wherein said topic delivery means is further comprised of:

topic switching means for switching topics responsive to a first qualifier object for a panel being displayed being in a first predetermined logical state.

7. The computer system as recited in claim 6 is further comprised of panel listing means for maintaining an ordered list of panels previously displayed.

8. The computer system as recited in claim 5 is further comprised of panel display formatting means for specifying the arrangement of display of media data on a panel.

9. The computer system as recited in claim 5 wherein said topic authoring means is a What You See Is What You Get application.

10. In a computer controlled display system, a method for interactive display of multimedia data comprising the steps of:

a) organizing multimedia data to be displayed into a topic comprised of a sequence of panels;

b) for each panel, specifying multimedia elements and flow control elements linked to said panel;

c) generating a database from said topic to create a set of topic objects, panel objects, multimedia data objects and flow control objects;

d) retrieving objects corresponding to a topic to be viewed; and for each panel in said sequence of panels perforating the steps of:

e) determining that a panel can be displayed based on the state of flow control objects linked to said panel including

i) identifying a panel display qualifier associated with a panel;

ii) determining the state of condition indicators associated with said panel display qualifiers:

iii) evaluating the state of said panel display qualifier based on the associated condition indicators:

iv) if said panel display qualifier is in a first state, identifying said panel as capable of display: and

v) if said panel display qualifier is not in said first state, identifying said panel as not capable of display; and

f) displaying the multimedia elements for said panel.

11. The method as recited in claim 10 wherein said flow control objects are comprised of panel display qualifiers and condition indicators, said step of determining that a panel can be displayed based on the state of flow control objects linked to said panel is further comprised of the steps of:

a) identifying a panel display qualifier associated with a panel;

b) determining the state of condition indicators associated with said panel display qualifiers;

c) evaluating the state of said panel display qualifier based on the associated condition indicators;

d) if said panel display qualifier is in a first state, identifying said panel as capable of display; and

e) if said panel display qualifier in not in said first state, identifying said panel as not capable of display.

* * * * *

US005446714A

United States Patent [19]

Yoshio et al.

[11] Patent Number: 5,446,714

[45] Date of Patent: Aug. 29, 1995

[54] DISC CHANGER AND PLAYER THAT READS AND STORES PROGRAM DATA OF ALL DISCS PRIOR TO REPRODUCTION AND METHOD OF REPRODUCING MUSIC ON THE SAME

[75] Inventors: **Junichi Yoshio; Yukio Taniai; Munetoshi Moriichi**, all of Tokorozawa, Japan

[73] Assignee: **Pioneer Electronic Corporation**, Tokyo, Japan

[21] Appl. No.: **91,881**

[22] Filed: **Jul. 15, 1993**

[30] **Foreign Application Priority Data**

Jul. 21, 1992	[JP]	Japan	4-193823
Jul. 21, 1992	[JP]	Japan	4-193824
Jul. 21, 1992	[JP]	Japan	4-193825

[51] **Int. Cl.**6 .... **G11B 7/00**

[52] **U.S. Cl.** **369/48**; 369/32; 369/36; 369/50; 369/58; 358/342

[58] **Field of Search** 369/47, 48, 32, 33, 369/36, 30, 50, 54, 58, 49; 358/342, 343, 341; 84/609, 601, 602, 603

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,796,099	1/1989	Compton	358/342
5,131,311	7/1992	Murakami et al.	84/609
5,157,643	10/1992	Suzuki	369/33
5,157,646	10/1992	Amemiya et al.	369/47
5,218,580	6/1993	Okamura et al.	369/2
5,228,021	7/1993	Sato et al.	369/58
5,245,600	9/1993	Yamauchi et al.	369/49
5,247,126	9/1993	Okamura et al.	84/609
5,268,889	12/1993	Furukawa et al.	369/47

FOREIGN PATENT DOCUMENTS

0383530 8/1990 European Pat. Off. .

Primary Examiner—W. R. Young
Assistant Examiner—Thang V. Tran
Attorney, Agent, or Firm—Young & Thompson

[57] **ABSTRACT**

The disc player accommodates a plurality of CD-I discs, to each of which program data required to reproduce information on the disc is recorded at a predetermined area thereof. The disc player has a disc changer to selectively supply each CD-I disc to a reproducing unit. The method of reproducing information of the disc player includes the steps of: reading the program data from the CD-I disc, and writing the read program data into a memory, with respect to all of the CD-I discs accommodated in the disc changer; and performing a reproduction operation with reference to the program data written in the memory at the time of the reproduction operation of the information on the CD-I disc.

7 Claims, 6 Drawing Sheets

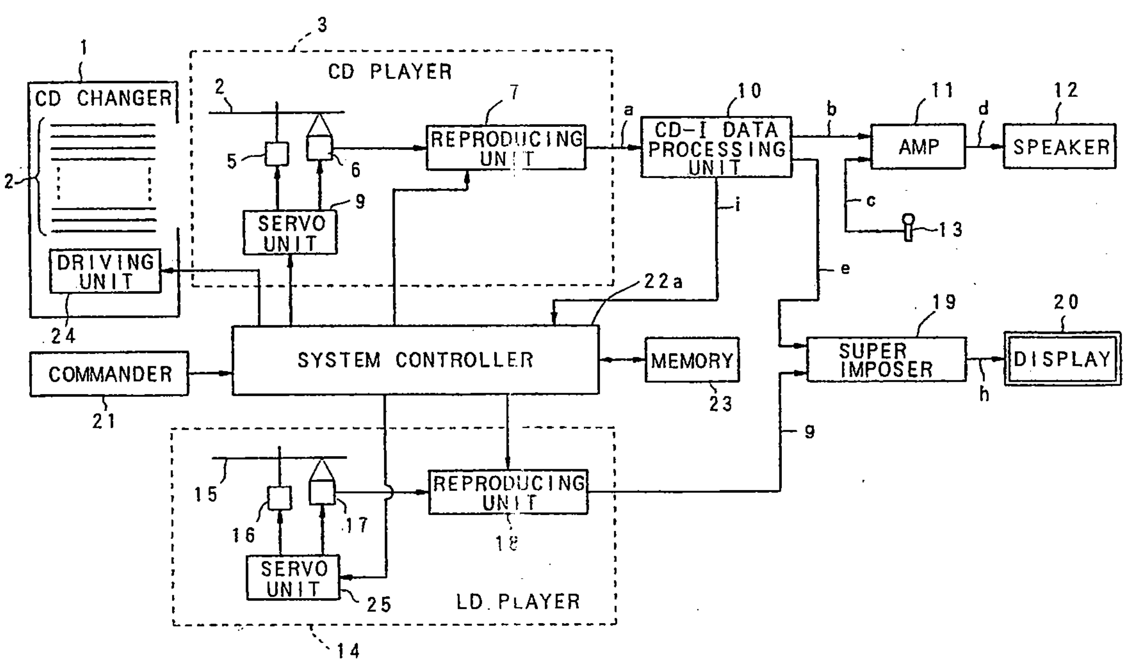


FIG. 1

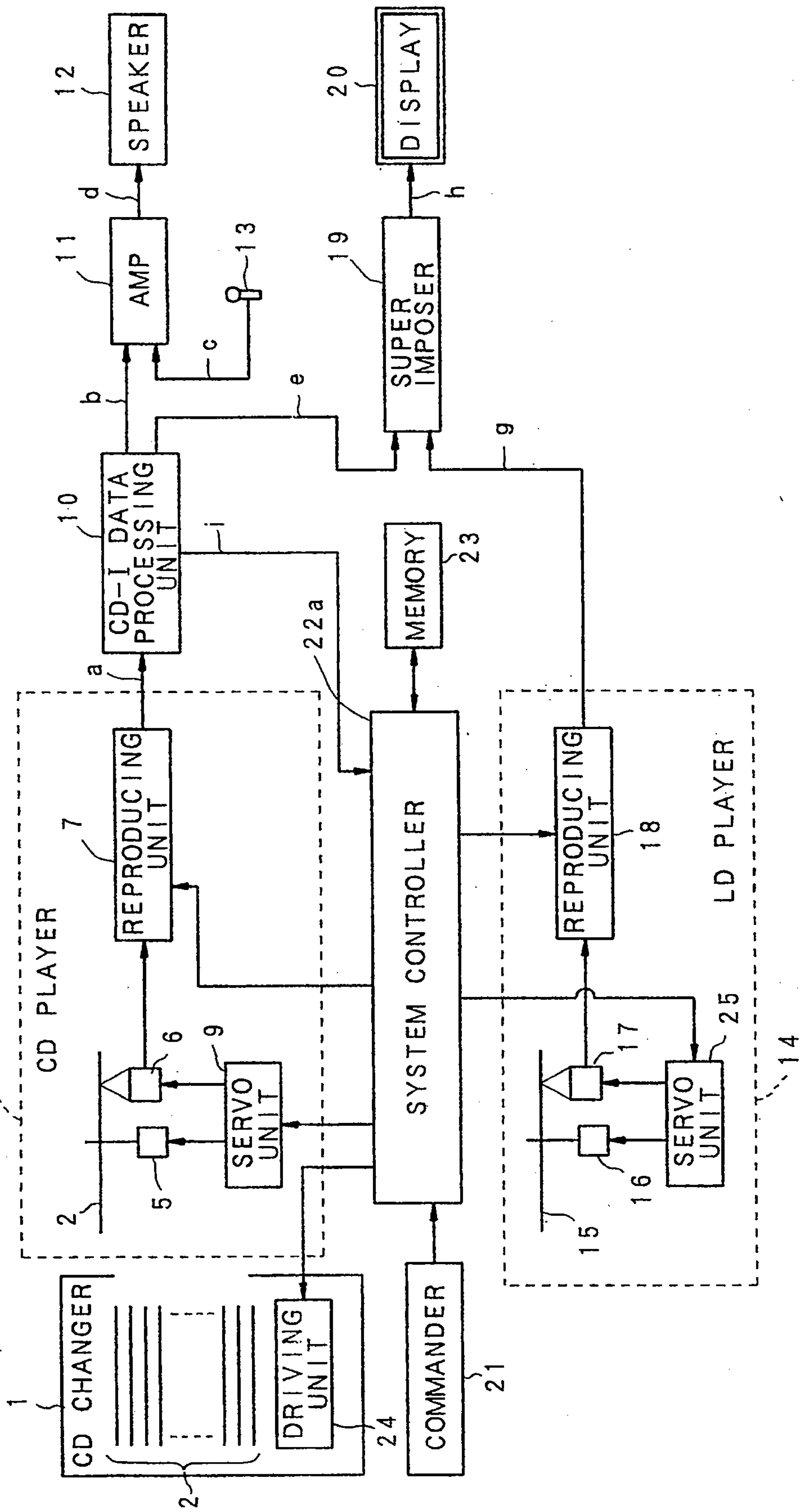

CD CHANGER
DRIVING UNIT
COMMANDER
CD PLAYER
REPRODUCING UNIT
SERVO UNIT
SYSTEM CONTROLLER
LD PLAYER
REPRODUCING UNIT
SERVO UNIT
CD-I DATA PROCESSING UNIT
MEMORY
AMP
SPEAKER
SUPER IMPOSER
DISPLAY
1
2
3
5
6
7
9
10
11
12
13
14
15
16
17
18
19
20
21
22a
23
24
25
a
b
c
d
e
g
h
i

FIG. 2

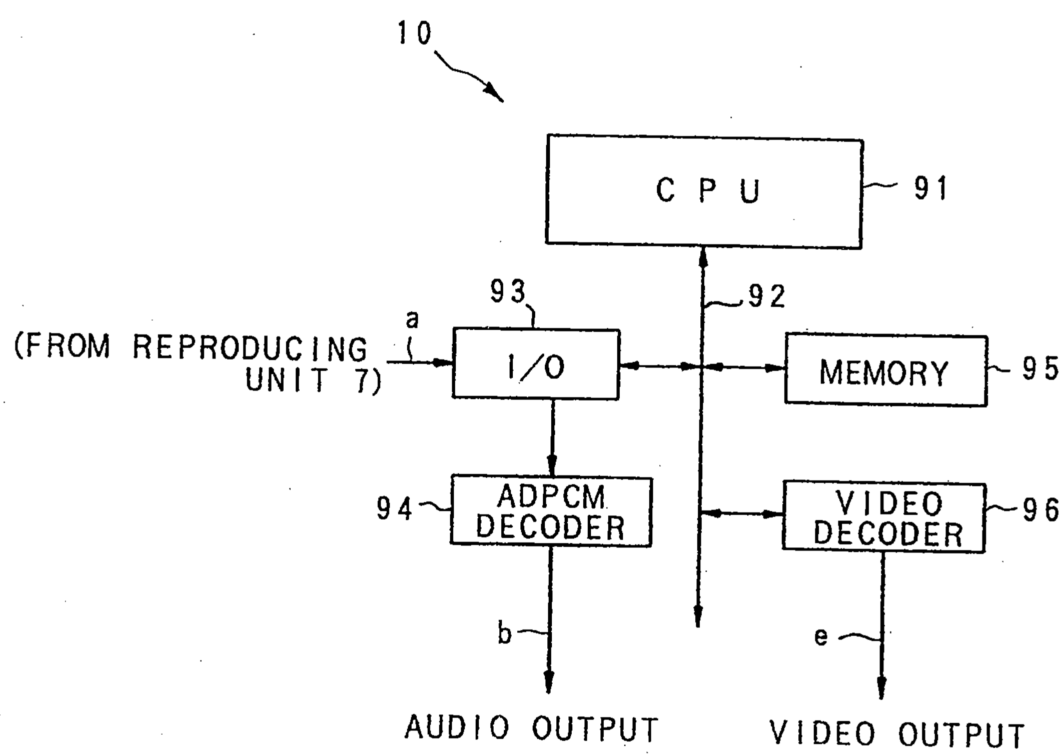
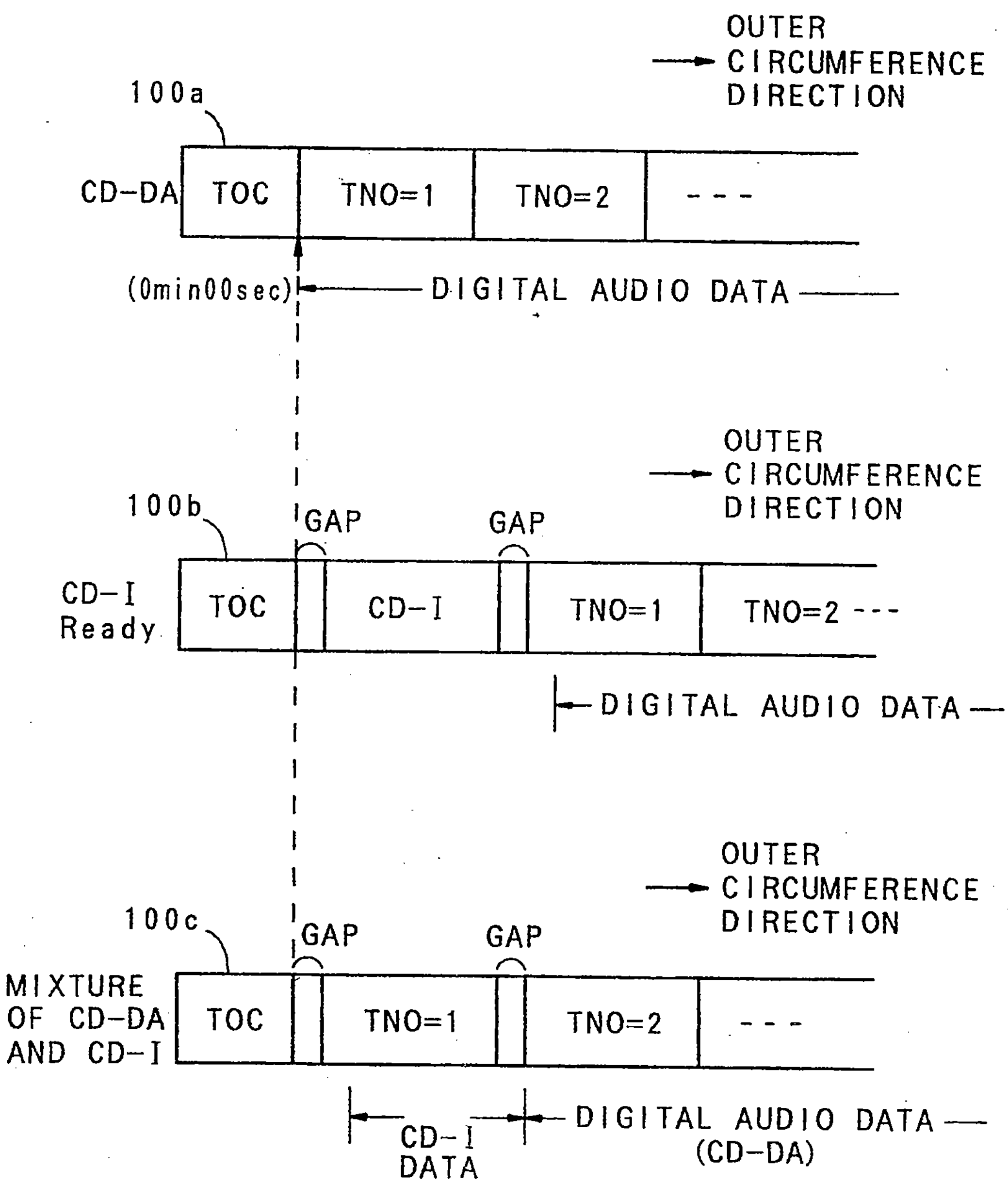

10
CPU
91
93
92
a
(FROM REPRODUCING UNIT 7)
I/O
MEMORY
95
94
ADPCM DECODER
VIDEO DECODER
96
b
e
AUDIO OUTPUT
VIDEO OUTPUT

FIG. 3


OUTER
CIRCUMFERENCE
DIRECTION
100a
CD-DA
TOC
TNO=1
TNO=2

(0min00sec)
DIGITAL AUDIO DATA
OUTER
CIRCUMFERENCE
DIRECTION
100b
GAP
GAP
CD-I
Ready
TOC
CD-I
TNO=1
TNO=2 ---
DIGITAL AUDIO DATA
OUTER
CIRCUMFERENCE
DIRECTION
100c
GAP
GAP
MIXTURE
OF CD-DA
AND CD-I
TOC
TNO=1
TNO=2

CD-I
DATA
DIGITAL AUDIO DATA
(CD-DA)

FIG. 4

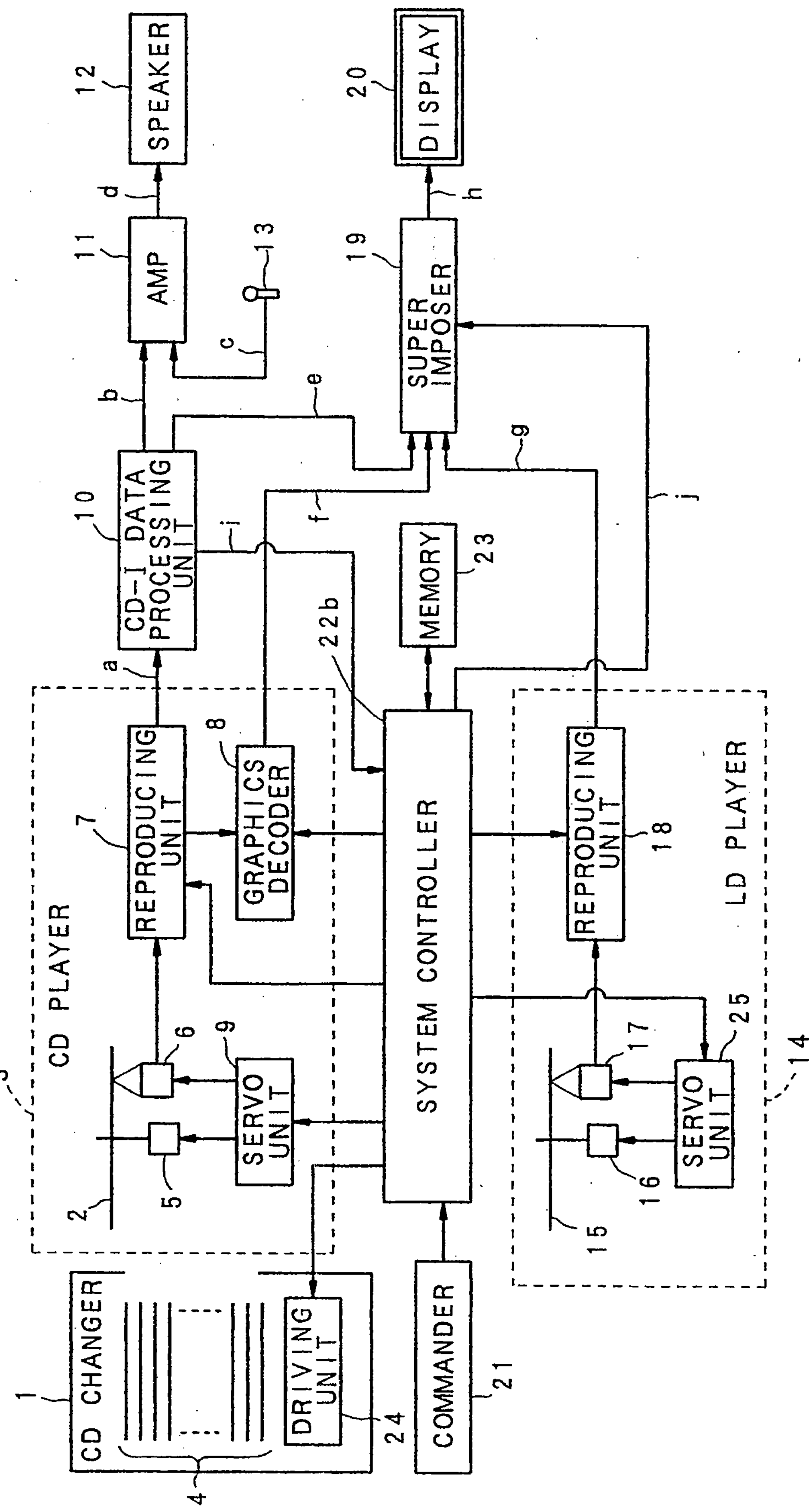

CD CHANGER
1
4
DRIVING UNIT
24
COMMANDER
21
CD PLAYER
3
2
5
6
SERVO UNIT
9
REPRODUCING UNIT
7
GRAPHICS DECODER
8
SYSTEM CONTROLLER
22b
a
CD-I DATA PROCESSING UNIT
10
i
b
AMP
11
c
13
d
SPEAKER
12
e
f
g
SUPER IMPOSER
19
h
DISPLAY
20
MEMORY
23
j
LD PLAYER
14
REPRODUCING UNIT
18
15
16
17
SERVO UNIT
25

FIG. 5

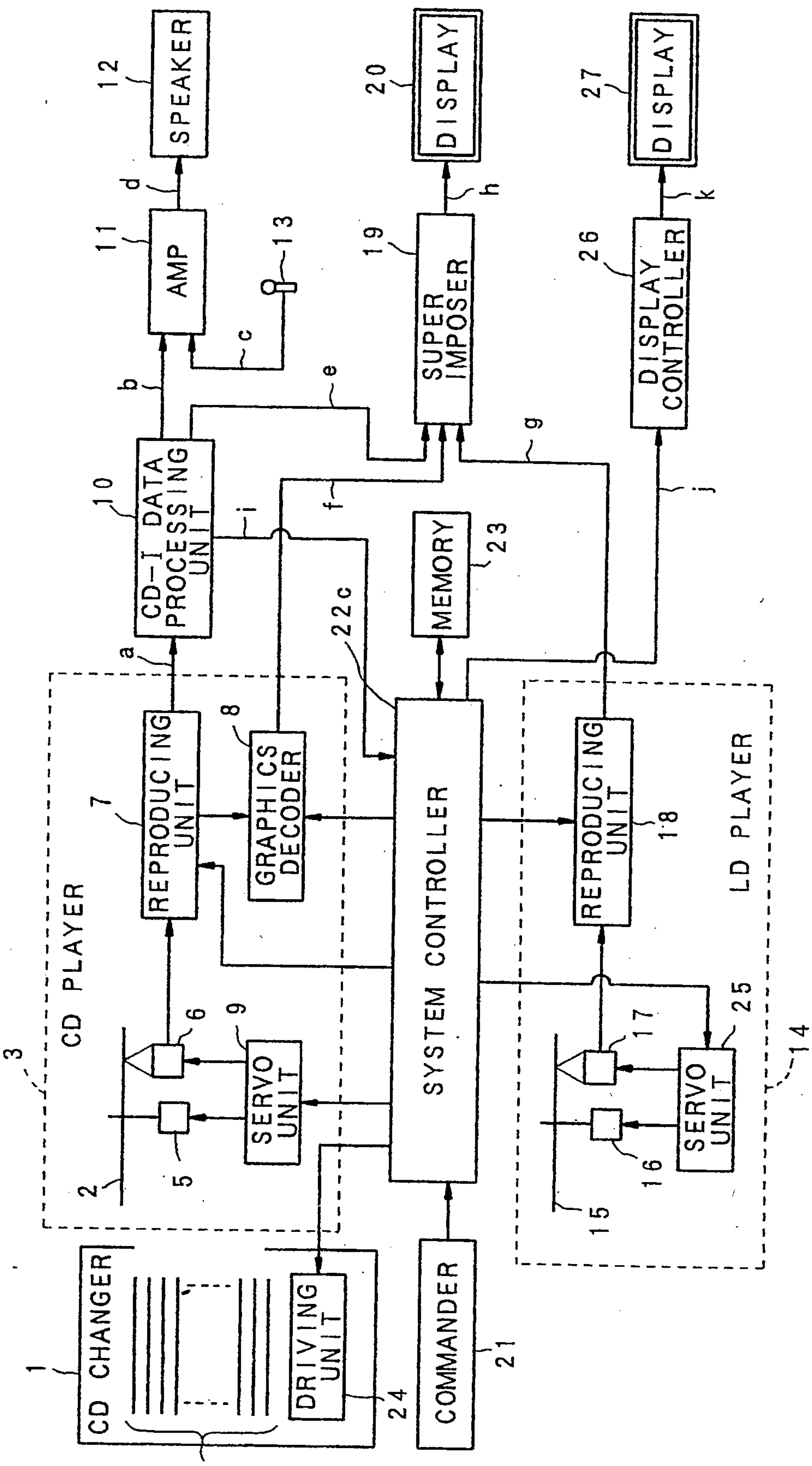
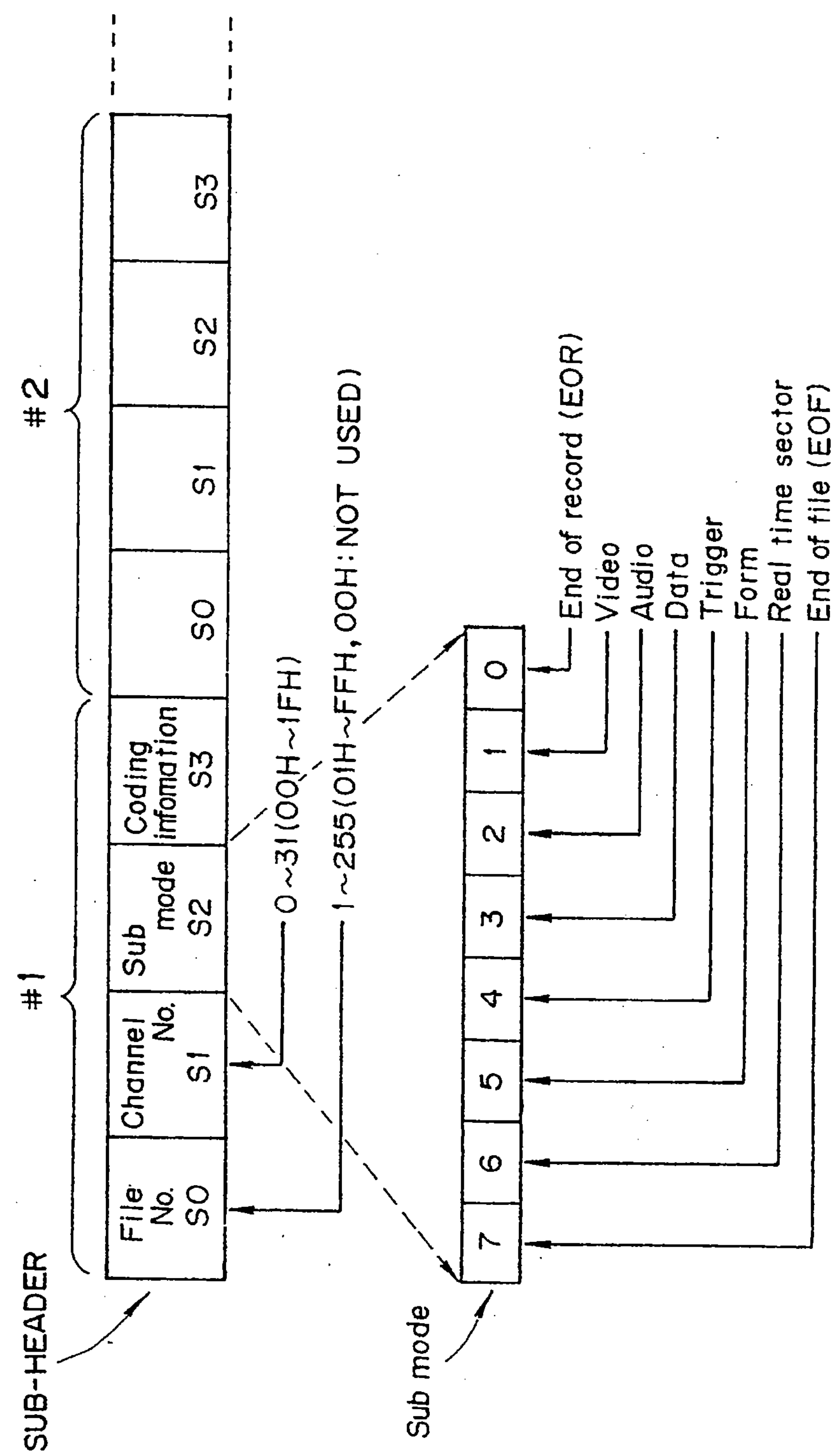

CD CHANGER
DRIVING UNIT
COMMANDER
CD PLAYER
SERVO UNIT
REPRODUCING UNIT
GRAPHICS DECODER
SYSTEM CONTROLLER
CD-I DATA PROCESSING UNIT
MEMORY
AMP
SPEAKER
SUPER IMPOSER
DISPLAY
DISPLAY CONTROLLER
DISPLAY
LD PLAYER
REPRODUCING UNIT
SERVO UNIT
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22c
23
24
25
26
27
a
b
c
d
e
f
g
h
i
j
k

FIG.6

DATA FORMAT OF SUB-HEADER



DISC CHANGER AND PLAYER THAT READS AND STORES PROGRAM DATA OF ALL DISCS PRIOR TO REPRODUCTION AND METHOD OF REPRODUCING MUSIC ON THE SAME

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention is related to a disc player and a method of reproducing information of the disc player.

More particularly, the present invention is related to a controlling method of a music accompaniment playing apparatus, which has a disc changer accommodating two or more CD-I (Compact Disc-Interactive) discs, on which "karaoke" (music accompaniment play) music is recorded by the CD-I format.

The present invention is also related to an improvement of the music accompaniment playing apparatus using a disc changer which accommodates many CD-Gs (CD graphics).

The present invention is further related to an improvement of displaying a title name of the music in the music accompaniment playing apparatus, which uses the disc changer accommodating the CD-G and the CD-I.

2. Description of the Related Art

A music accompaniment playing system is adapted to display words on a display device with a music accompaniment playing sound. Besides the music accompaniment playing apparatus using a LD (Laser Disc), which reproduces an animation as a BGV (Back Ground Video: background image), there is a music accompaniment playing apparatus by use of the CD-G (CD-Graphics). The graphics data is kept by the compact format in the CD-G.

The CD-G type music accompaniment playing apparatus as well as the LD music accompaniment playing apparatus, has a CD changer for storing or accommodating many CD-Gs. For example, about 300 discs of CD-Gs are accommodated in the CD changer. If it is converted into the number of songs, it is equivalent to about 3,000 songs. In the CD-G, word information is recorded as graphics data together with audio information of the music of the music accompaniment play. At the time of reproduction, this graphics data is reproduced with the audio information, and the character display of it is carried out on a picture plane of a TV monitor. A CD-G type music accompaniment playing apparatus is generally used while it is interlocked with the LD player prepared independently, and is adapted to superimpose the graphics data to the animation BGV data supplied from the LD player to display it at the time of reproduction.

In the above mentioned disc player and the reproducing method thereof, there is a first problem as follows.

Namely, according to the CD-I, the program for operating a CPU (Central Processor Unit) is stored on the disc itself. The disc system of a dialog type is built under a predetermined OS (Operating System) environment. The start address of the data built in the disc, the sequence about the control of the operation, etc. are described in this program.

It is possible to provide a music accompaniment playing system in the same manner as the conventional CD-G disc, by constructing the music data and the graphics data by use of the ADPCM (Adaptive Delta Pulse Code Modulation), as the data of such a CD-I disc.

However, in order to perform the performance operation by the CD-I disc, it needs to load in advance the program recorded in the disc, for each disc i.e. each time when the selected one disc is to be reproduced, so that, as the time for loading this, it is necessary about 30 seconds, including the time required for the loading operation of the disc, and for locking various servo-operations.

If such a CD-I disc is applied to the changer system, which stores about 300 discs, for example, it becomes necessary to load the program data for every operation of exchanging the discs. Thus, in order to shift to the next music, the waiting time for about 30 seconds is required, which becomes a serious problem when adapting it to the music accompaniment playing system, which is the first problem.

Further, in the aforementioned disc player and the reproducing method thereof, there is a second problem as follows.

Namely, the conventional CD-G type music accompaniment playing apparatus does not have a function which displays a music title in advance of the reproduction of the music. Generally, the music title to be reproduced is displayed on the picture plane of the TV monitor as the graphics data with a performance of the introductory Section (so-called introduction portion) for every song. In other words, the music title cannot be checked, until the reproduction has actually begun.

As a result, when the operator of the music accompaniment playing apparatus does operation mistakes, such as a music title input mistake, he notices it after a performance of the music actually starts. Furthers, there occurs a problem that One man has selected one music and another man selects the same music in succession.

Therefore, it may be proposed to employ a CD-I (CD-Interactive) as a disc which can record a music title data, in addition to the program information Of the music accompaniment play. The CD-I is adapted to record various kinds of program data other than the music data compressed by the ADPCM (Adaptive Delta Pulse Code Modulation).

However, it is not economically appropriate to exchange all of the existing CD-Gs installed in the wide-spread CD-G type music accompaniment playing apparatuses, to the CD-Is, resulting that the burden of the supplier of the music accompaniment playing discs as well as the user of the apparatus would be enormous, which is the second problem.

Further, in the aforementioned disc player and the reproducing method thereof, there is a third problem as follows.

Namely, since the conventional CD-G type music accompaniment playing apparatus does not have a function which displays a music title in advance of the reproduction of the music, and since the music title is displayed on the picture plane of the TV monitor with a performance of the introductory section for every song, when the reservations for a large number of desired musics of music accompaniment play are inputted, each music is continuously reproduced in the order of the input, according to the conventional technique, so that, the music title cannot be checked i.e. which music is nextly reproduced, until the reproduction is actually begun. This brings a problem that the preparation to sing must be done after starting the introduction portion of the music specified by the operator.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a disc player and a method of reproducing the information of the disc player, which makes it possible to access speedily a program required for the reproduction, at the time of an information reproducing operation accompanied by an exchanging operation of the CD-I discs.

It is another object of the present invention to provide a disc player and a method of reproducing information of the disc player, which can carry out the picture plane display of the program information name, such as a music title, in advance of the reproduction of the program information, while promoting an effective use of the existing discs.

It is another object of the present invention to provide a disc player and a method of reproducing information of the disc player, which can display the program information names, such as a title name of music etc., in precedence during the reproduction of the preceding program information in case that a plurality of pieces of program information, such as musics of music accompaniment play, are continuously reproduced, while promoting an effective practical use of the existing discs.

According to the present invention, the above object can be achieved by a first information reproducing method and a first disc player of the present invention.

In the first method, the information of a disc player accommodating a plurality of CD-I discs, to each of which program data required to reproduce information on the disc is recorded at a predetermined area thereof, is reproduced. The disc player has a disc changer to selectively supply each CD-I disc to a reproducing unit. The first method includes the steps of: reading the program data from the CD-I disc, and writing the read program data into a memory, with respect to all of the CD-I discs accommodated in the disc changer; and performing a reproduction operation with reference to the program data written in the memory at the time of the reproduction operation of the information on the CD-I disc.

The first disc player includes: a reproducing unit for reproducing information recorded on a CD-I disc, to which program data required to reproduce the information is also recorded at a predetermined area thereof; a disc changer for accommodating a plurality of CD-I discs and selectively supply each CD-I disc to the reproducing unit; a memory for storing the program data; and a controller for controlling the reproducing unit to read the program data from the CD-I disc and writing the read program data into the memory, With respect to all of the CD-I discs accommodated in the disc changer, and for performing a reproduction operation of the reproducing unit with reference to the program data written in the memory at the time of the reproduction operation of the information on the CD-I disc.

According to the first method and the first disc player of the present invention, the program data of all CD-I discs in the disc changer is read prior to the reproduction of the CD-I disc. And, this read program data is written into the memory. Subsequently, at the time of reproducing the CD-I disc which is supplied to the reproducing unit from the disc changer, the required program data is read from the memory. Then, the reproduction operation is performed with reference to this read program data.

In this manner, since the program data of each CD-I disc is transferred to memory in advance, it is not necessary to read the program data from the CD-I disc for each CD-I disc upon reproducing, and thus, it is possible to increase the speed of accessing the program data accompanied by the disc exchange, according to the first method and the first disc player of the present invention.

According to the present invention, the above object can be also achieved by a second information reproducing method and a second disc player of the present invention.

In the second method, the information of a disc player, is reproduced. The disc player accommodates a plurality of first format discs, each of which stores program information by a first format, and at least one second format disc, which stores program information name data of each of the first format discs by a second format different from the first format, and reproduces the program information. The second method includes the steps of: reading the program information name data from the second format disc and storing the program information name data into a memory; selecting one of the first format discs to be reproduced; and displaying the program information name data corresponding to the selected first format disc before reproducing the program information of the selected first format disc.

The second disc player includes: a plurality of first format discs, each of which stores program information by a first format; at least one second format disc, which stores program information name data of each of the first format discs by a second format different from the first format; a reproducing unit for selecting one of the first format discs and reproducing the program information of the selected first format disc; a memory for storing the program information name data; a displaying device for displaying a program information name; and a controlling device for reading the program information name data from the second format disc, storing it to the memory, and transferring the program information name data corresponding to the selected first format disc to the displaying device before the reproducing unit reproduces the program information of the selected first format disc.

Namely, according to one aspect of the present invention, the second disc player records the music title data of accompaniment play in the CD-G format, which is the first format disc, onto the CD-I, which is the second format disc, and it reads out the music title data from the CD-I at the time of installation of the music accompaniment playing apparatus etc. to store thus read music title data to the memory. Then, each time of reproducing the CD-G, it reads the corresponding music title of the music from the memory, and displays it on the picture plane of the TV monitor.

According to the second method and the second disc player of the present invention, the second format disc, on which each program information name data of a plurality of first format discs are collectively recorded, is prepared. From this second format disc, the program information name data is stored to the memory under the control at the time of installation etc. of the music accompaniment playing apparatus. And, in advance of the reproduction of the program information, the program information name data can be displayed on the displaying device.

Therefore, it is not necessary to replace all of the first format discs by the second format disc, but just by employing the second format disc of 1 disc or several discs, it is possible to operate the existing system with a function to display the program information name, according to the second method and the second disc player of the present invention.

According to the present invention, the above object can be also achieved by a third disc player and a third method of reproducing information of the disc player of the present invention.

In the third method, the information of a disc player for selecting one of a plurality of recording discs and reproducing program information included in the selected recording disc, is reproduced. The third method includes the steps of: reproducing one program information; and displaying an information title name of another program information to be nextly reproduced after said one program information, on a displaying device, at a predetermined time during a reproduction of said one program information.

The third disc player includes: a reproducing unit for selecting one of a plurality of recording discs and reproducing one program information included in the selected recording disc; a memory for storing an information title name of the program information of each of the recording discs; and a displaying device for displaying the information title name of another program information to be nextly reproduced after one program information at a predetermined time during a reproduction of one program information.

Namely, in case that a plurality of musics of music accompaniment play, are continuously reproduced, the present invention is adapted to display the title name of the music which is to be nextly reproduced, on a picture plane of a TV monitor during the reproduction of the preceding music at a predetermined time before the end, for example, according to one aspect of the present invention.

According to the third method and the third disc player of the present invention, at the predetermined time during the reproduction of one program information, for example, at the time just before the reproduction end of the program information etc., the information title name of another program information which is to be nextly reproduced, is displayed on the displaying means. Thereby, it becomes possible to know beforehand which program information will be nextly reproduced. For example, when the present invention is applied to the music accompaniment playing apparatus, it becomes possible to know beforehand that the music selected by the operator will be nextly reproduced during the reproduction of the preceding music.

The nature, utility, and further features of this invention will be more clearly apparent from the following detailed description with respect to preferred embodiments of the invention when read in conjunction with the accompanying drawings briefly described below.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram which shows a constitution of a disc player as a first embodiment according to the present invention;

FIG. 2 is a block diagram which shows a detailed example of a CD-I data processing unit in the first embodiment of FIG. 1;

FIG. 3 is a diagram which shows an example of a disc format employed in the first embodiment of FIG. 1;

FIG. 4 is a block diagram which shows a constitution of a disc player as a second embodiment of the present invention;

FIG. 5 is a block diagram which shows a constitution of a disc player as a third embodiment of the present invention; and

FIG. 6 is a diagram in which an example of data format of a sub-header of the CD-I is shown.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Nextly, the preferred embodiments of the present invention will be explained on the basis of the drawings.

First Embodiment

FIG. 1 shows a music accompaniment playing apparatus as a first embodiment according to the present invention.

In FIG. 1, the music accompaniment playing apparatus is provided with a CD changer **1**, a CD player **3**, a CD-I data processing unit **10**, an amplifier **11**, a speaker **12**, a microphone **13**, a LD player **14**, a superimposer **19**, a display device **20**, a commander **21**, a system controller *22a*, and a memory **23**.

The CD changer **1** accommodates a plurality of CDs **2**, including a CD-I disc, and is provided with a driving unit **24** for exchanging the CDs and selectively supplying one selected CD to the CD player **3** under the control of the system controller *22a*.

The CD player **3** is provided with a spindle motor **5**, an optical pickup **6**, a reproducing unit **7** and a servo unit **9**. The LD player is provided with a spindle motor **16**, an optical pickup **17**, a reproducing unit **18**, and a servo unit **25**.

The CD-I data processing unit **10** is provided with a CPU **91** (Central Processing Unit) **91**, a system bus **92**, an I/O input **93**, an ADPCM decoder **94**, a memory **95**, and a video decoder **96**, as shown in FIG. 2.

Here, before explaining the detailed construction of the system, the CD-I will be explained, which is the medium used in the present invention.

In the CD-I, the logical format, the computer and OS (Operating System) for controlling it, the method of compressing the audio signal and the video signal to the digital signal, and a portion of the specification as the product, are defined in a standard, which is called by common-name of "GREEN BOOK". According to this standard, all CD-I discs have interchangeability such that they are reproducible by all CD-I players. The CD-I system is defined such that it can use various audio and video information as an interactive multimedia system by use of a computer program.

In this manner, the CD-I is the first multi-media system which is considered as a system for public use toward the application for home and educational use.

The CD-I standard is defined on the basis of the CD-ROM standard. The CD-ROM standard is defined on the basis of the CD-DA standard (which is the general audio CD standard and has a common name of "RED BOOK"). Therefore, the physical specifications, such as a disc outside size and a weight, and the optical specification, follow the RED BOOK. The data format follows the YELLOW BOOK.

The format of the CD-I is specified on the basis of the mode 2 of the CD-ROM, and is different, by addition of the sub-header and the positions of the EDC, ECC etc., from the two formats (form **1** and form 2) and the CD-ROM format.

To the sub-header, 4 bytes of a file number, a channel number, a sub-mode and a coding information, are written twice as a countermeasure of the reading error. The file number is prepared in order to recognize the block when the block belonging to the same file is interleaved. The file number is stored to the directory mentioned later, corresponding to the file name. The block of the file number belongs to one certain file.

In the data format of the CD-I, each data form of the audio and video to be recorded, and the file format which constructs these data, are as following.

Namely, the audio data is recorded by the ADPCM method. There are three levels in this method. If the tone quality is decreased, the record capacity is increased, so that prolonged reproduction is enabled. In the data compression according to this method, the data of the prediction filter and the range used at the time of the reproduction, is recorded together with the compressed data.

The video data is recorded by the compression coding method of the various still pictures. The method suitable for the kind and the usage of the original image is employed. The DYUV is suitable for natural drawings. The CLUT and the RGB555 are suitable for so-called graphics drawings. The RL is also suitable for the graphics drawings and, since its rate of compression is higher, it is applicable also to graphics animations.

As for the file format, the CD-I format employs the hierarchical structure on the basis of a root directory, under which some layers of the sub-directories are layered, and a file is further located thereunder. Each subdirectory is also treated as a file.

From the files in this hierarchical structure, the target one is sought. A pass table is provided in order to make the seeking-time short. Moreover, the distributed subdirectory structure is employed. The disc label is written to the sector, whose physical address of the disc is 2 minutes 16 seconds block, in which the contents of the disc and the logical addresses of the pass table and the root directory, are written.

The seeking operation of the CD-I is performed as follows. Firstly, the disc label is accessed, and the address of the pass table is recognized. Then, the pass table is accessed, and the contents of the pass table are stored into the RAM. The pass table is an address table on which the sub-directory is arranged in the order of the parent-and-child relation and the name. Thus, the address of the sub-directory including the target file, is identified, so that the access there is performed. From this position, the continuous reading operation is performed, and the target file is obtained.

The CD-I disc, which is constructed by the above format, is then subjected to the reproduction in the CD-I system. The CD-I system is set to the GREEN BOOK.

The GREEN BOOK prescribes as for the OS, which is called as "CD-RTOS", and is constructed on the basis of the OS-9/68000.

In this manner, it is the CD-I system, which has the computer using the OS and the large-scaled CD-ROM, and includes the compressed audio and video signal format.

In FIG. 3, the format **100***a* shows an example of the format of CD-DA which is a so-called compact disc for audio, or CD-G. The format **100***b* shows an example of the format of CD-I Ready which is different form the formats of the CD-DA and CD-G. The format **100***c* shows a format which is, so to speak, a mixture of the CD-DA and the CD-I Ready, and is adapted to pseudo-function in the same manner as the CD-I Ready by recording the CD-I data at the area of TNO1 of the CD-DA. Identification of each of these discs is performed by detecting the identifying code recorded in each TOC area.

In the CD-I Ready format **100***b* shown in FIG. 3, there are provided a CD-I area and a FILE TOC (file track) area having a structure specialized as its application, through a gap area between the TOC area and the TNO (track number) 1. A volume descriptor VD is provided in the CD-I area. This volume descriptor VD is the description word which properly describes the contents of the information stored in the pertinent disc, and is read at the time of a reading start.

As shown in the format **100***c* of FIG. 3, as one form of the CD-I, for example, the TNO1 may be used as the CD-I track, and the track TNO2 and the tracks after the TNO 2 may be used as the CD-DA track.

In this manner, the CD-I Ready format **100***b* as a modification of the CD-I has the application program area and the data area between the TOC area situated in the lead-in area and the TNO1 in the program area, through the gap area.

The music accompaniment playing apparatus according to the present invention is constructed on the basis of the above mentioned CD-I format.

In FIG. 1, the CD changer **1** accommodates or stores a plurality of the above mentioned CD-Is **2**, and is adapted to selectively load each disc one by one to the CD player **3** by the driving unit **24** in correspondence with the change command. The CD changer **1** is operated by a control command of the system controller **22***a* mentioned later in detail. The CD player **3** reproduces the record information of the CD-I **2** selected from the CD changer **1**. Namely, for example, under a predetermined rotating condition by the spindle motor **5**, the loaded CD-I **2** is optically read by the pickup **6**, and the RF signal converted into the electric signal by the pickup **6** is demodulated by the reproducing unit **7**. The servo unit **9** controls the number of rotations of the spindle motor **5**, and the focusing and tracking controls of the pickup **6**. The CD-I data processing unit **10** is connected to the output of the reproducing unit **7**.

As shown in FIG. 2, the CD-I data processing unit **10** consists of a computer. By the program stored in the memory **95**, the CPU **91** carries out the decoding of the CD-ROM data in the CD-I **2**, performs the extraction of the data required for reproduction, and generates and outputs the audio data b and the video data e. The audio data b is sent to the amplifier **11**. The video data e is sent to the superimposer **19**.

The amplifier **11** combines the voice signal c from the microphone **13**, and the audio signal b from the CD-I data processing unit **10**, amplifies it to a predetermined power level, and outputs the composite signal d to the speaker **12**.

The LD player **14** is an apparatus for supplying the BGV which runs short in the CD-I **2**. For example, the BGV information edited by a music genre or other classification, is recorded in the LD **15**, such that, by the technique of specifying by an operator or an automatic selection, the BGV is demodulated by the reproducing unit **18**, and the BGV-data g is outputted.

The servo unit **25** controls the spindle motor **16** and the pickup **17** in the same manner as the servo unit **9**. These servo units **9** and **25** are controlled by the system controller **22***a*.

The above video data e from the CD-I data processing unit 10 and the BGV data g from there producing unit 18, are inputted into the superimposer 19. The superimposer 19 superimposes the character information to the BGV data g, generates a video signal h, and outputs it to the display device 20. The display device 20 displays thus given video signal h, on a picture screen.

The system controller 22a performs the system control of the music accompaniment playing apparatus including the CD player 3 and the LD player 14. Namely, the system controller 22a receives the command from the commander 21, and supplies various required control signals to each units, according to the command. The memory 23 is connected to the system controller 22a.

As for the memory 23, a SRAM (Static RAM) which is battery backup type, such as a NV (Non Volatile: non-volatility) RAM, is preferred. The memory 23 is a memory for storing the information recorded in the FILE TOC area of the CD-I 2. This information recorded in the FILE TOC area includes a program data required for reproduction of the disc. It has been mentioned that this information is repeatedly read each time of reproducing each disc.

Nextly, the main operation of the music accompaniment playing apparatus of the present embodiment will be explained in the above constitution.

Firstly, at the time of installation of this music accompaniment playing apparatus, for example, if a power supply is switched on, the music accompaniment playing apparatus enters the install mode. There are various items of operation at the time of the install mode. Here, only the operations directly related to the present invention, will be explained.

When entering the install mode, under the control of the system controller 22a, the CD changer 1 loads all of the stored CD-Is 2 to the CD player 3 one after another. The CD player 3 reads the FILE TOC of the loaded CD-I 2, demodulates the read data through the reproducing unit 7, and sends it to the CD-I data processing unit 10. The CD-I data processing unit 10 decodes the program data read from the FILE TOC, which is required for the reproduction, and sends the data i to the system controller 22a. The system controller 22a writes thus sent program data i into the memory 23. The above mentioned series of operation is performed with respect to all of the CD-Is 2 in the CD changer 1, so that the program data of all of the CD-Is 2 is collected and stored in the memory 23 in advance.

Nextly, when entering the usual music accompaniment play reproduction operation mode, when the music selecting command is given to the system controller 22a by the commander 21, the system controller 22a gives the necessary control signals to each units in the system such as the disc change command to the driving unit 24 of the CD changer 1, the reproducing operation control command to the CD player 3 and the LD player 14 and so on, according to the command.

Now, it is assumed that one CD-I 2 is selected and is loaded onto the CD player 3. At this time, conventionally, it is necessary to read the program data required for the reproduction of this CD-I 2 at first according to the prior art system. However, since, in the present invention, the program data is already written in the memory 23 at the time of the installation, there is no need to read it again, so that, just by reading out the corresponding data in the memory 23, it is possible to quickly enter the reproduction operation, and thus high speed access becomes possible. The same thing can be said as for all of the discs contained in the CD changer 1.

As described above in detail, according to the first embodiment of the present invention, the program data is read out from the CD-I disc at the time of install mode etc., and the operation of writing the read out program data into the memory is performed with respect to all of the CD-I discs contained in said disc changer. The reproduction operation is performed with reference to the program data written in the memory at the time of information reproduction operation of the CD-I disc. Accordingly, it turns out that the program of each CD-I disc is transferred to memory beforehand, so that there is no need to read the program from the disc for every CD-I disc, and it is possible to make the access speedy, which is accompanied by a disc change.

Further, the initial load operation of the CD-I disc can be simplified, and when applying it to the music accompaniment play, the operation of shifting to the next music can be quickly performed.

Second Embodiment

Nextly, the second embodiment of the present invention will be explained hereinbelow with referring to FIG. 4.

FIG. 4 shows a block diagram in case of applying the present invention to a music accompaniment playing apparatus. In FIG. 4, the same constitutional elements as those in FIG. 1, carry the same reference numerals, and the detailed explanations thereof are omitted.

In FIG. 4, the music accompaniment playing apparatus is provided with a graphics decoder 8, which is controlled by a system controller 22b. A plurality of the CD-Gs 4 as well as the CD-I 2, are accommodated in the CD disc changer 1, while the CD-I 2 is loaded from the CD changer 1 to the CD player 3 in the figure.

The CD changer 1 contains a plurality of discs (for example, 300 discs), and is adapted to selectively load each disc to the CD player 3 by the driving unit 24. As one manner of storing discs, a total of 300 discs including 299 discs of the existing CD-Gs 4 and just 1 discs of a newly added/exchanged CD-I 2, are stored therein, for example.

The CD-G 4 has a format by which graphics data is recorded in the sub code (8 bits) area (R to W) of each frame, in addition to the main data which consists of the music data.

On the other hand, unlike the CD-ROM, the CD-I 2 is not restricted as for the kind of data to be recorded into the data area, but is able to record character data etc. other than audio data.

The example of this format is shown in FIG. 3, as explained before in detail. In the present embodiment, the FILE TOC prepared as the above mentioned application, is used as a storing section for the information about all CD-Gs 4 accommodated in the CD changer 1, and the music title information of the music of music accompaniment play recorded in the program area of each CD-G 4.

In FIG. 4, the CD player 3 reproduces the record information of the CD-G 4 or the CD-I 2 selected from the CD changer 1. Namely, for example, under the predetermined rotation by the spindle motor 5, the loaded CD-I 2 is optically read by the pickup 6, and the RF signal converted into the electric signal by the pickup 6, is reproduced or demodulated by the reproducing unit 7. The servo unit 9 performs the number of rotations control of the spindle motor 5, the focusing and tracking controls of the pickup 6 and so on. Since the graphics data is included in the record information in case that the disc of the reading object is CD-G 4, the graphics decoder 8 is an apparatus for decoding the graphics data, to output graphics data f, which is the result of decoding. The CD-I data processing unit 10 is connected to the output of the reproducing unit 7.

In case that the disc of the reading object is the CD-I 2, the CD-I data processing unit 10 is adapted to extract the peculiar data i.e. the volume descriptor information VD, for example, and output the music title data e. In case that the disc of the reading object is either the CD-G 4 or the CD-I 2, the CD-I data processing unit 10 is adapted to extract the audio signal therefrom and output the audio signal b to the amplifier 11. An example of a detailed constitution of the CD-I data processing unit 10 is shown in FIG. 2.

The amplifier 11 combines the voice signal c from the microphone 13, and the audio signal b from the CD-I data processing unit 10, amplifies it to a predetermined power level, and outputs the composite signal d to the speaker 12.

The LD player 14 is an apparatus for supplying the BGV which runs short in the CD-G 4. The servo units 9 and 25 are subjected to the control of the system controller 22*b*.

The above explained music title data e from the CD-I data processing unit 10, the graphics data f from the graphics decoder 8 and the BGV data g from the reproducing unit 18, are inputted into the superimposer 19. The superimposer 19 superimposes these character information to the BGV data g, generates the video signal h and outputs it to the display device 20. The display device 20 displays thus given video signal h on a picture screen thereof. As a display mode, there may be such examples that the graphics data f (words) is synthesized on the BGV, the music title data e is superimposed on the BGV, and the music title data e is superimposed on the BGV of one color background.

The system controller 22*b* performs the system control of the music accompaniment playing apparatus including the CD player 3 and the LD player 14. Namely, the system controller 22*b* receives the command from the commander 21, and, according to the command, it supplies various required control signals to each unit. The memory 23 is connected to the system controller 22*b*.

The memory 23 is a memory for storing the music title data of the songs of music accompaniment play stored in the CD-I 2, as for all of the CD-Gs 4, at the time of installation of the music accompaniment playing apparatus. As the memory 23, for example, a semiconductor memory, such as a DRAM (Dynamic-RAM), may be used. Assuming that the music titles for 3,000 musics are to be stored, about 120 K bytes of memory capacity is required. Namely, in case of 3,000 songs, it becomes 2 bytes×20 character×3,000 musics=120,000 bytes as 20 characters are reserved per song. In addition to this, considering the data volume required to direct the address (the position and the track number on the disc) of the program of each disc, the memory 23 can be realized by use of a 256-K bytes DRAM on the whole. The origin of storing the music title data and the addresses, which are to be stored to the memory 23, is the CD-I 2.

The main operation of the present embodiment will be explained in the above constitution.

Firstly, by turning on the power supply, the operation enters the install mode at the time of installation of the music accompaniment playing apparatus. Although there are various items in the operations at the time of the install mode, only the operation directly related to the present invention, will be explained here.

When it enters the install mode, the CD-I 2 is selected from the CD changer 1 under the control of the system controller 22*b*, and is loaded to the CD player 3. The CD player 3 is controlled to read the volume descriptor VD from the loaded CD-I 2. The RF signal of the music title information read by the pickup 6, is demodulated by the reproducing unit 7, decoded through the CD-I data processing unit 10, and is taken into the system controller 22*b* as the music title information i. The system controller 22*b* writes the inputted music title information i to the memory 23. The installing operation as for the music title information is completed up to this point.

Nextly, at the time of the normal music accompaniment playing reproduction operation, when the music selecting command is given to the system controller 22*b* from the commander 21, the system controller 22*b* supplies the required control signal to each unit according to the given command. Namely, the system controller 22*b* controls the exchanging operation of the selected disc by the driving unit 24, the reproduction operations by the CD player 3 and the LD player 14, and so on. These series of operations are same as in the case of the conventional CD-G type music accompaniment playing apparatus.

However, according to the present invention, before beginning the reproduction of the CD player 3 and the LD player 14, the system controller 22*b* reads the music title information of the musics of music accompaniment play corresponding to the selected CD-G 4, from the memory 23, and the system controller 22*b* sends the music title information j to the superimposer 19 to display the information 3 on the display device 20. As a result, it brings such an advantageous effect that the title of the music of music accompaniment play to be reproduced from now, can be recognized, and thus, an unnecessary selection and a selection by mistake, can be checked before actually reproducing them. After checking the music title, it enters the reproduction mode in succession, as long as no cancellation command is given.

In the above mentioned explanation, the music title information j is transmitted to the display device 20 through the superimposer 19. However, it may be directly supplied to the display device 20. Therefore, the display device 20 in this case, may be a TV monitor set at the vicinity of the operator, independently from a TV monitor set at the audience seats, wherein the music title information 3 may be supplied only to this operator's TV monitor.

Moreover, although the above mentioned explanation is made as for the case where there is only a single CD-I 2 disc, the number of the CD-I may be two or more as occasion demands. As an example of such an occasion, there is an occasion where the existing CD-Gs are successively converted to the CD-I little by little, or an occasion where many discs are contained in one CD changer, so that the music title information are distributed to two or more CD-Is.

As described above, according to the second embodiment of the present invention, the program information name in each first format disc is recorded in the second format disc. At the time of installing the disc player etc., the program information name is transferred from the second format disc to the memory to store it. Prior to the reproduction of each CD-G, the program information name of each CD-G of the reproduction object, is displayed. Thus, there is no necessity to exchange whole first format discs. The program information name with respect to all discs can be displayed beforehand, just by preparing the second format disc or discs, which number is limited to the minimum degrees, according to the second embodiment of the present invention.

Nextly, third to fifth embodiments of the present invention will be explained.

Each of the third to fifth embodiments is constituted such that it displays the title name of the music accompaniment play in a preliminary announcement manner, for example, at the timing just before the reproduction end of the music which is reproduced in precedence, by use of the "information in the disc", in a music accompaniment playing apparatus using a plurality of discs including CD-I, in case of displaying the title name of the music to be reproduced on a picture plane of a monitor. In this case, the "information in the disc" used for the standard clock generation of the clock means to count the image display timing of the title name, differs depending upon the models of discs, and, according to it, the constitution of the music accompaniment playing apparatus differs slightly. Thus, hereinbelow, the explanation of the music accompaniment playing apparatus will be explained corresponding to the modes of combination of the discs. As the mode of the combination of the discs, it is divided roughly into the following three categories.

(1) The mode in which the discs in the CD changer **1** are constituted of a large number of the CD-Gs and a small number of CD-Is.

(2) The mode in which all of the discs in the CD changer **1** are constituted of the CD-Is.

(3) The mode in which the discs in the CD-changer **1** are random mixture of the CD-Gs and the CD-Is.

Third Embodiment

FIG. **5** shows a third embodiment of the present invention. The third embodiment is an example of the music accompaniment playing apparatus in the case that the disc constitution is the above mentioned (1) mode.

In FIG. 5, the same constitutional elements as those in FIG. 1 and FIG. 4, carry the same reference numerals, and the detailed explanations thereof are omitted. The third embodiment is provided with a graphics decoder **8**, a system controller **22c**, a display controller **26**, and a display device **27**.

In FIG. 5, the CD changer **1** accommodates a plurality of discs, and is constituted so that it selectively load each disc to the CD player **3** one by one by the driving unit **24**, according to an exchange command.

There are accommodated in the CD changer **1**, 300 discs in total, including, for example, 299 discs of CD-Gs **4** as an existing type disc, and just 1 disc of CD-I **2** as a newly added/changed disc.

The examples of the format of each disc, are shown in FIG. 3, as aforementioned.

The CD-G **4** has such a format by which the word data, which consists of the graphics data, is recorded in the sub code area (R—W) of each frame (not illustrated), other than the audio data, which is recorded in the program information area.

The format of the CD-I **2** is prescribed on the basis of the CD-ROM, which details are prescribed according to the mode 2 of the CD-ROM. Unlike the CD-ROM, the CD-I **2** is not restricted about the kind of data recorded in the data area, so that it is possible to record the character data etc. other than the audio data to it. There is a CD-I Ready as a modification of the CD-I. As shown in the format **100b** of FIG. **3**, the CD-I Ready has the CD-I area and the FILE TOC area through the gap area between the TOC area and the TNO1. The VD information is prepared in the CD-I area, and is the description word which appropriately expresses and describes the contents of the information stored in the disc. The information about all of the CD-Gs **4** accommodated in the CD changer **1** and the "title" and the address (the position and the track number on the disc) of the music of the music accompaniment play, which are recorded in the program area in each CD-G **4**, are recorded in the FILE TOC area, which is prepared as the application.

The format of CD-I **2** is defined by two formats of form **1**, **2**, such that a sub-header (8 bytes) is arranged next to the header portion of each format. File number (File No.) **S0**, channel number (Channel No.) **S1**, submode (Sub mode) S2, and coding information (Coding information) S3 are recorded in the sub-header, as shown in FIG. **6**. Data of these **S0** to S3 are recorded twice in a repeated manner. Repeating twice is for the countermeasure against the reading error.

Among these, as shown in FIG. **6**, the sub-mode **S2** (a total of 8 bits) includes a flag bit, which indicates the attribute of the pertinent block i.e. distinguishing audio data or video data, distinguishing form **1** or form **2** and so on. One bit code called as a "trigger bit" (Trigger bit), is recorded in the sub-mode **S2**. This trigger bit is built up once at the interval of 1/75 seconds. Thus, it is possible to utilize this trigger bit as a standard clock in a cycle of 1/75 seconds, and it is also possible to constitute a clock means by counting this standard clock by a software counter set in the CPU in the system controller **22c**. This point will be explained later in detail in the section of the image display controlling operation of the "title name".

On the other hand, as shown in the format **100c** of FIG. 3, as one form of the CD-I, for example, the **TNO1** may be used as the CD-I track, and the TNO 2 and the tracks thereafter may be used as the CD-DA tracks.

Both of the above CD-DA (CD-G) and the CD-I Ready have the TOC (Table of Contents) area. The flag bit, which indicates the kind of the disc (i.e. distinguishing CD-DA (CD-G) or CD-I) is stored in this predetermined position in this TOC area, so that the kind of the disc can be identified by this bit. The reproduction mode of the music accompaniment playing apparatus differs according to the result of reading this bit. Moreover, in the TOC, the address data in the musical movement is included as the address of the music of the music accompaniment play, so that this time information can be also used by the clock means in the same manner as the trigger bit of the CD-I. This point will be also mentioned later in the section of the image display controlling operation of the "title name".

Again, in FIG. **5**, the CD player **3** reproduces the record information of the CD-G **4** or the CD-I **2** selected from the CD changer **1**. Namely, for example, under the predetermined rotation by the spindle motor **5**, the loaded CD-I **2** is optically read by the pickup **6**. The RF signal converted into the electric signal by the pickup 6, is demodulated in the reproducing unit 7. The servo unit 9 performs the number of rotation control of the spindle motor 5, the focusing control and the tracking control of the pickup 6 and so on. Since the graphics data is included in the record information in case that the disc of reading object is the CD-G 4, the graphics decoder 8 is an apparatus for decoding the graphics data, and outputting the graphics data f as a result of decoding. The CD-I data processing unit 10 is connected to the output of the reproducing unit 7.

If the disc of the reading object is the CD-I 2, the CD-I data processing unit 10 outputs the peculiar data i.e. the title name data e and i of the music of music accompaniment play, for example, which are stored in the FILE TOC area. Even if the disc of the reading object is either the CD-G 4 or the CD-I 2, the CD-I data processing unit 10 extracts the audio signal, and sends the audio signal b to the amplifier 11. The example of the detailed constitution of the CD-I data processing unit 10 is shown in FIG. 2, as aforementioned.

The amplifier 11 combines the voice signal c from the microphone 13, and the audio signal b from the CD-I data processing unit 10, amplifies it to a predetermined power level, and outputs the composite signal d to the speaker 12.

The LD player 14 is an apparatus for supplying the BGV which runs short in the CD-G 4. The servo unit 25 controls the spindle motor 16 and the pickup 17 as well as the servo unit 9. These servo units 9 and 25 are subjected to the control by the system controller 22c.

The above title name data e from the CD-I data processing unit 10, the graphics data f from the graphics decoder 8 and the BGV data g from the reproducing unit 18, are inputted into the superimposer 19. The superimposer 19 superimposes these character information onto the BGV data g, generates the video signal h, and outputs it to the display device 20. The display device 20 displays thus given video signal h on the picture screen thereof. There are examples such as the display mode, that the graphics data f (words) is combined on the BGV, the title data e is overlapped on the BGV, and the title data e is superimposed on the background BGV of one color.

The system controller 22c performs the system control of the music accompaniment playing apparatus including the CD player 3 and the LD player 14. Namely, the system controller 22c receives the command from the commander 21, and, according to the command, supplies various control signals to each of the required unit. The memory 23 is connected to the system controller 22c.

The memory 23 is a memory for storing the title name data of the music of music accompaniment play with respect to all of the CD-Gs 4, which is stored in the CD-I 2, at the time of installing the music accompaniment playing apparatus.

The main operation of the apparatus will be explained in the above constitution.

Firstly, at the time of installing the music accompaniment playing apparatus, the operation enters an install mode, upon turning on a power supply. Though there are various items in the operation at the time of the install mode, only the operations directly related to the present invention, will be explained here.

When it enters an install mode, under the control of the system controller 22c, the CD-I 2 is selected from the CD changer 1, and is loaded to the CD player 3. The CD player 3 is controlled to read the FILE TOC from the CD-I 2 which is loaded. The RF signal of the title name data read by the pickup 6, is demodulated in the reproducing unit 7, is decoded through the CD-I data processing unit 10, and is taken into the system controller 22c as the title name data. The system controller 22c writes the inputted file name data i into the memory 23. The installing operation about the title name data is completed up to this point.

Nextly, at the time of the usual reproducing operation of the music accompaniment play, when the music selecting command is given to the system controller 22c by the commander 21, the system controller 22c gives a required control signal to each unit according to this command. Namely, the system controller 22c controls the exchanging operation of the selected discs with respect to the driving unit 24, and the reproducing operation of the CD player 3 and the LD player 14, and so on.

Now, it is assumed that a certain CD-G 4 is being reproduced. At this time, the system controller 22c calculates the ending time of the reproduced music of the CD-G 4 on the basis of the absolute time data (or progress time data) which is read from the TOC of the CD-G 4. The system controller 22c, reads out the "title name" data of the music of the music accompaniment play, which is to be nextly reproduced, from the memory 23, by the timing at a predetermined time before the ending time. The system controller 22c gives the title name data j to the display controller 26, which sends the video signal k to the display device 27, so that the display device 20 displays in the preliminary announcement manner the "title name" of the music to be nextly reproduced. Alternatively, the system controller 22c may be constituted such that the controller 22c sends the title data e to the superimposer 19 from the CD-I data processing unit 10, and superimposes it onto the BGV on the display device 20 of the music which is being reproduced. The above explained displaying operation is controlled on the basis of the absolute time data (or the progress time data) in the TOC of each CD-G 4, with respect to each CD-G 4 to be reproduced.

On the other hand, if the disc related to the reproduction is the CD-I 2, the "trigger bit" in the sub-mode S2 of the sub-header is used as the standard clock. By counting this standard clock by the software counter of the CPU in the system controller 22c, the system controller 22c reads the "title name" of the music of music accompaniment play, which is to be nextly reproduced, from the memory 23 in the same manner, and displays the title name of the next music at the predetermined time before the reproduction end of the music under the present reproduction.

In this manner, irrespective of the kind of the disc to be reproduced, the preliminary-announcement display "title name" of the music to be nextly reproduced, can be performed on the picture screen of the display device 20 or the display device 27 at the predetermined time before the end of the music under the present reproduction.

Fourth Embodiment

This fourth embodiment is the example of the music accompaniment playing apparatus in case where the disc constitution is the aforementioned (2) mode.

Since the basic constitution of the music accompaniment playing apparatus is the same as that of the third embodiment of FIG. 5, the explanation thereof is omitted.

The characteristic feature of the fourth embodiment is the point that the control of the display timing of the "title name" of the next music is performed by counting the above mentioned "trigger bit". This counting operation is realized by the software counter of the CPU in the system controller 22c, and the display timing etc. thereof, is the same as that of the third embodiment.

Fifth Embodiment

This third embodiment is the example of the music accompaniment playing apparatus in case that the disc constitution is the aforementioned (3) mode.

Since the basic constitution of the music accompaniment playing apparatus is the same as that of the third embodiment of FIG. 5, the explanation thereof is omitted.

The characteristic feature of the fifth embodiment is the point that the control of the display timing of the "title name" of the next music is performed by the absolute time data in the TOC if the disc is the CD-G 4, and the control is performed by the "trigger bits" if the disc is the CD-I, so that both operations can be performed and that the operations can be automatically switched over in accordance with the kind of the disc. The discrimination of the disc is performed by referring to the discrimination bit in the TOC.

In addition, in the above mentioned explanation, although the title name data e is transmitted to the display device 20 through the superimposer 19, it may be directly supplied to the display device 27 through the display controller 26. Moreover, the display device 27 in this case may be the TV monitor, which is, for example, set at the vicinity of the operator, and is different form the TV monitor set at the seat for the audience. The title name data may be supplied only to the operator's TV monitor in this case.

As explained above, according to the third to fifth embodiments of the present invention, since the information title name of other program information to be nextly reproduced after one program information at the specified time during the reproduction of one program information, is displayed on the display device, the information title name of the program information to be nextly reproduced can be announced beforehand.

The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiments are therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

What is claimed is:

1. A method of reproducing information of a disc player accommodating a plurality of first format discs, each of said first format discs storing program information by a first format, and accommodating at least one second format disc, said second format disc storing program information name data of each of the first format discs by a second format different from the first format, and reproducing the program information, said method comprising the steps of:

reading the program information name data from the second format disc and storing the program information name data into a memory;

selecting one of the first format discs to be reproduced; and

displaying the program information name data corresponding to a selected first format disc before reproducing the program information of the selected first format disc.

2. A method according to claim 1, wherein said disc player comprises a music accompaniment playing apparatus, and a title of music to be reproduced is displayed in the displaying step.

3. A method according to claim 1, wherein said first format disc comprises a CD-G, and said second format disc comprises a CD-I.

4. A disc player comprising:

a plurality of first format discs, each of said first format discs storing program information by a first format;

at least one second format disc, said second format disc storing program information name data of each of the first format discs by a second format different from the first format;

a reproducing unit for selecting one of the first format discs and reproducing the program information of a selected first format disc;

a memory for storing the program information name data;

a displaying means for displaying a program information name; and

a controlling means for reading the program information name data from the second format disc, a means for storing said information name data to said memory, and a means for transferring the program information name data corresponding to the selected first format disc to said displaying means before said reproducing unit reproduces the program information of the selected first format disc.

5. A disc player according to claim 4, wherein said second format disc comprises a disc having a data area for storing the program information name data on an inner circumferential side of an area corresponding to a first program area of said first format disc.

6. A disc player according to claim 4, further comprising a disc changer for accommodating said first and second format discs and selectively supplying one of the first and second format discs to said reproducing unit.

7. A disc player according to claim 4, further comprising:

an LD player for reproducing video data from an LD disc; and

a superimposer for superimposing the program information name data onto the video data, said displaying means displaying the superimposed data.

* * * * *

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平1－287872

⑤Int. Cl.⁴　識別記号　庁内整理番号　㊸公開　平成1年(1989)11月20日
G 11 B 20/12　8524－5D

審査請求　未請求　請求項の数 4　（全16頁）

㊹発明の名称　デイジタルデータ記録／再生装置

㉑特　願　昭63－118567
㉒出　願　昭63(1988)5月16日

⑫発明者　佐古　曜一郎　東京都品川区北品川6丁目7番35号　ソニー株式会社内
⑫発明者　小川　博司　東京都品川区北品川6丁目7番35号　ソニー株式会社内
⑪出願人　ソニー株式会社　東京都品川区北品川6丁目7番35号
⑭代理人　弁理士　杉浦　正知

明　細　書

1. 発明の名称

ディジタルデータ記録／再生装置

2. 特許請求の範囲

(1)入力データに対して所定のエラー訂正符号を付加する符号化手段と、上記符号化されたデータを記録媒体に記録する手段とを有するディジタルデータ記録装置において、

上記符号化手段は、上記入力データを一時的に蓄積するメモリと、

上記メモリに対するアドレスを制御するアドレス制御手段とを有し、

上記アドレス制御手段は、第1のアドレス制御と、第2のアドレス制御とが設定可能とされたディジタルデータ記録装置。

(2)上記第1のアドレス制御により記録されるデータと上記第2のアドレス制御により記録されるデータとを選択的に設定するための選択信号を上記記録媒体に記録するようにした請求項1記載のディジタルデータ記録装置。

(3)記録媒体の記録データを再生する手段と、上記再生されたデータを復号する復号化手段とを有するディジタルデータ再生装置において、

上記復号化手段は、上記再生データを一時的に蓄積するメモリと、

上記メモリに対するアドレスを制御するアドレス制御手段とを有し、

上記アドレス制御手段は、上記記録媒体の再生データが上記第1のアドレス制御により記録されたデータか上記第2のアドレス制御により記録されたデータかに応じて選択的にアドレス制御を設定可能とするようにしたディジタルデータ再生装置。

(4)上記光学的記録媒体に記録されている上記選択信号を再生して、上記光学的記録媒体の再生データが上記第1のアドレス制御により記録されたデータか上記第2のアドレス制御により記録されたデータかを判断するようにした請求項3記載のディジタルデータ再生装置。

3. 発明の詳細な説明

〔産業上の利用分野〕

この発明は、ディジタルデータ記録／再生装置に関するもので、例えば音楽用のコンパクトディスクに対応した形態で光学的記録媒体にデータを記録／再生するようにしたディジタルデータ記録／再生装置に係わる。

〔発明の概要〕

この発明は、コンパクトディスクに対応した形態でデータを記録媒体に記録／再生するようにしたディジタルデータ記録／再生装置において、コンパクトディスクと同様のインターリーブ処理と、1記録単位（セクタ）に対応する長さのインターリーブ処理とを行えるようにし、セクタ完結型の符号化及びその復号化を実行可能とすることにより、任意のセクタにデータを書き込む処理や任意のセクタのデータを書き換える処理を簡単に行えるようにしたものである。

〔従来の技術〕

オーディオ信号をディジタル化して光学的に記録したコンパクトディスクが広く知られている。このコンパクトディスクは、例えば直径12cmで、約500Mバイト以上ものデータを記録できる記録媒体である。したがって、このコンパクトディスクを大量のデータを記録するディジタルデータ記録媒体として用いることができる。

このことに着目し、コンパクトディスクの音楽記録領域にオーディオデータ以外のディジタルデータを記録できるようにしたCD－ROMが規格化されている。CD－ROMでは、音楽用のコンパクトディスクと同じ態様でディジタルデータが記録される。すなわち、CD－ROMでは、光ディスクとして音楽用のコンパクトディスクと同様の直径12cmでスパイラル状にトラックが形成されているものが用いられる。記録データは、CIRC（Cross Interleave Reed-Solomon Code）により2重に符号化され、EFM変調（8－14変調）されて光ディスクに光学的に記録される。

このようなCD－ROMにおいて、データの記録単位となる1セクタ（1ブロック）は、98フレームからなる1サブコードブロックから構成される。つまり、音楽用のコンパクトディスクでは、1フレーム当たりP～Wまでの8ビットのサブコード（R～Wはユーザーズビットとも称される）が用意されている。このサブコードに、98フレーム分で1つの情報単位（アドレス）となっていることから、98フレームがブロックと呼ばれる。

CD－ROMは、基本的に読み出し専用の記録媒体である。CD－ROMは、記憶容量が大きく、大量複製ができ、情報の劣化が少ない等の特徴がある。このような特徴を生かし、各種辞典類のデータや研究資料データを記録するのにCD－ROMが利用されている。

〔発明が解決しようとする課題〕

近年、追記型の光学記録媒体や、光磁気ディスクのように消去、再記録が可能な光学記録媒体が開発されている。このような追記型の光学記録媒体や消去、再記録可能な光学記録媒体を、コンパクトディスクと同様な態様で用い、ディジタルデータを記録することが提案されている。追記型のデータ記録媒体としてのコンパクトディスク（以下、CD－WOと称する）や、消去、再記録可能なデータ記録媒体としてのコンパクトディスク（以下、CD－イレーザブルと称する）では、データを追記したり、再記録したりすることができるので、再生専用のCD－ROM以上に、幅広い分野での利用が期待できる。

ところで、CD－WOやCD－イレーザブルでは、セクタ単位でデータの書き込み／読み出しがなされる。これに対して、音楽用のコンパクトディスクは再生専用で、データが時系列順に再生される。

音楽用のコンパクトディスクは、音楽データのようなシーケンシャルなデータに最適なように、記録データには最大108フレームのインターリーブが施されている。このため、音楽用のコンパクトディスクと全く同一の信号処理形態でCD－WOやCD－イレーザブルを実現した場合、任意

のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりする場合に、複雑な信号処理が必要になる。このような複雑な処理を必要としないために、CD－WOやCD－イレーザブルを、音楽用のコンパクトディスクと全くことなった信号フォーマットに変更すると、コンパクトディスク、CD－ROM、CD－WO、CD－イレーザブルの間のコンパティビリティを失うことになる。

すなわち、コンパクトディスクでは、前述したように、2次元配列上の各列毎のシンボルをリードソロモン符号を用いてC_2系列のパリティを生成付加し、インターリーブ遅延を行った後、リードソロモン符号を用いてC_1系列のパリティを生成付加するCIRCが用いられている。このようなCIRCでは、音楽データのようなシーケンシャルなデータに対して最適化するように、最大108フレームのインターリーブが施され、畳込み符号化がなされる。これに対して、CD－WOやCD－イレーザブルにおいてデータの書き込み、読み出しの単位となる1セクタは、98フレームからなるサブコードブロックから構成される。

したがって、任意の1セクタのデータを書き換えると、前2セクタと後2セクタとにその影響が及ぶ。すなわち、任意の1セクタのデータを書き換えると、その前後2セクタのデータに関するC_1系列のパリティが変わってくる。

このことから、任意の1セクタのデータの書き換えを行う場合には、そのセクタのデータとその前後2セクタのデータとを取り込み、C_1系列のパリティを求め直す処理が必要になってくる。

そこで、例えば特願昭62－244996号明細書に示されるように、2セクタ以上連続する全て「0」のデータのセクタをデータ記録用のセクタの間に挿入し、例えば3セクタ毎のセクタをデータ記録用のセクタとすることが提案されている。ところが、このようにすると、パリティを求め直す処理は必要なくなるが、データの記録容量が最悪½程度に減少してしまう。

したがって、この発明の目的は、任意のセクタにデータを記録する場合や、任意のセクタのデータを書き換える際に、複雑な処理を必要としないとともに、データ記録容量が削減されることのないディジタルデータ記録／再生装置を提供することにある。

また、従来では、このように畳込み符号が用いられているため、任意のセクタのデータを取り込むのに、少なくともこれに連続する2セクタのデータを取り込まなければならず、アクセス時間が長く必要になる。

この発明の更に他の目的は、アクセス時間の短縮化がはかれるディジタルデータ記録／再生装置を提供することにある。

〔課題を解決するための手段〕

この発明は、入力データに対して所定のエラー訂正符号を付加する符号化手段と、符号化されてデータを記録媒体に記録する手段とを有するディジタルデータ記録装置において、

符号化手段は、入力データを一時的に蓄積するメモリと、

メモリに対するアドレスを制御するアドレス制御手段とを有し、

アドレス制御手段は、第1のアドレス制御と、第2のアドレス制御とが設定可能とされたディジタルデータ記録装置である。

第1のアドレス制御により記録されるデータと第2のアドレス制御により記録されるデータとを選択的に設定するための選択信号が記録媒体に記録される。

また、この発明は、記録媒体の記録データを再生する手段と、再生されたデータを復号する復号化手段とを有するディジタルデータ再生装置において、

復号化手段は、再生データを一時的に蓄積するメモリと、

メモリに対するアドレスを制御するアドレス制御手段とを有し、

アドレス制御手段は、記録媒体の再生データが第1のアドレス制御により記録されたデータか第

2のアドレス制御により記録されたデータかに応じて選択的にアドレス制御を設定可能とするようにしたディジタルデータ再生装置である。

記録媒体に記録されている選択信号を再生して、記録媒体の再生データが第1のアドレス制御により記録されたデータか第2のアドレス制御により記録されたデータかを判断するようにしている。

〔作用〕

コンパクトディスクと同一形態の光ディスクを用いてディジタルデータの記録／再生が行われる。この際、第1及び第2の2つのインターリーブ処理が設定できる。第1のインターリーブ処理では、全体インターリーブ長108フレームとされる。この第1のインターリーブ処理は、音楽データのようなシーケンシャルなデータを扱う場合や、音楽用コンパクトディスクやＣＤ－ＲＯＭとのコンパチビリティを完全に保ちたい場合に好適である。

第2のインターリーブ処理では、全体インターリーブ長95フレームとされる。そして、モジュロ98の処理により98フレームで回りながらインターリーブが施される。これにより、エラー訂正符号化が98フレームからなる1セクタ内で完結される。この第2のインターリーブ処理は、1セクタ内でエラー訂正符号化が完結するため、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりする必要性が多い場合に用いて好適である。

〔実施例〕

この発明の実施例について以下の順序に従って説明する。

a．記録／再生装置の概要
b．フレーム構造及びセクタ構造
c．インターリーブ処理について
d．第1のインターリーブ処理の場合のエンコード及びデコードプロセス
e．第2のインターリーブ処理の場合のエンコード及びデコードプロセス
f．一実施例におけるエンコード・デコード処理

a．記録／再生装置の概要

第1図は、この発明が適用されたディジタルデータ記録／再生装置の概要を示すものである。第1図において、1はディジタルデータが光学的に記録／再生される光ディスクである。光ディスク1としては、追記型の光ディスクや消去．再記録可能な光ディスク、例えば光磁気ディスクを用いることができる。光ディスク1は、音楽用のコンパクトディスクと同様の形態とされている。すなわち、光ディスク1の直径は12cmであり、光ディスク1にはスパイラル状のトラックが形成される。そして、光ディスク1は、ＣＬＶ（線速度一定）でもって回転される。

記録時には、光ディスク1に記録すべきデータがデータ入力端子2に供給される。この記録データが符号化回路3に供給される。符号化回路3は、C_2エンコーダ4と、インターリーブ遅延回路5と、C_1エンコーダ6とから構成される。入力端子1からのデータは、所定フレーム構造に展開され、符号化回路3で、C_1系列とC_2系列とで2重に符号化される。インターリーブ遅延回路5には、端子7から選択信号が供給される。インターリーブ遅延回路5は、後に詳述するように、インターリーブ長の異なる第1、第2の2つのインターリーブ処理が施せる。第1のインターリーブ処理の場合には、最大108フレームのインターリーブが施される。そして、第1のインターリーブ処理の場合は、畳込み符号化がなされる。第2のインターリーブ処理の場合には、インターリーブ長が98になり、第2のインターリーブ処理では、エラー訂正符号化が1セクタ内で完結される。

符号化回路3で、2重にエラー訂正符号が付加されたデータは、ＥＦＭ変調回路8でＥＦＭ変調（8－14変調）され、光ディスク1に記録される。

光ディスク1の記録データを再生する際には、上述の記録時と逆のシーケンスで処理がなされる。

すなわち、光ディスク1の再生データがＥＦＭ

復調回路9に供給され、EFM復調される。EFM復調回路9の出力が復号化回路10に供給される。復号化回路10は、C_1デコーダ11と、ディインターリーブ遅延回路12と、C_2デコーダ13とから構成される。ディインターリーブ遅延回路12は、インターリーブ遅延回路5で設定できる2つのインターリーブ処理に対応して、第1及び第2の2つのディインターリーブ処理が行える。ディインターリーブ遅延回路12には、端子14から選択信号が供給され、この選択信号により2つのディインターリーブ処理が切換え可能とされている。

なお、記録時に、光ディスク1の一部、例えばディスク最内周のTOC (Table of Contents)に第1及び第2のインターリーブ処理のうちどちらのインターリーブ処理を行ったかのデータを書き込むようにし、再生時には、このデータに応じてディインターリーブ遅延回路12を切り換えるようにしても良い。

また、光ディスク1に第1のインターリーブ処理の信号と第2のインターリーブ処理の信号とを混在させて記録することも可能である。すなわち、例えば光ディスク1として、予め所定のデータが記録されるデータ領域と、ユーザーが自在に書き込みできるユーザー領域とを設けておき、予め所定のデータが記録される領域には第1のインターリーブ処理で信号を記録し、ユーザー領域には第2のインターリーブ処理で信号を記録するようにしても良い。この場合、データ領域は、CD-ROMと同様にスタンパを用いて大量複製することができる。

復号化回路10の出力が出力端子15から取り出され、出力端子15の出力から再生データが得られる。

b. フレーム構造及びセクタ構造

光ディスク1には、第2図Aに示すように、フレーム構造にデータが展開され、EFM変調されてデータが記録される。このフレーム構造は、音楽用のコンパクトディスクと同様である。すなわち、1フレームは、第2図Aに示すように、オーディオデータを16ビットでサンプリングした場合にL(左)、R(右)各6サンプル分に相当する24シンボル(1シンボルは8ビット、EFM変調されて14チャンネルビット)のデータビットと、8シンボルのパリティと、1シンボルのサブコードと、図示していない24チャンネルビットのフレームシンクと、直流分抑圧用のマージンビットからなる。したがって、1フレームの総チャンネルビット数は、

フレームシンク	24チャンネルビット
データビット	14×24=336チャンネルビット
サブコード	14チャンネルビット
パリティ	14×8=112チャンネルビット
マージンビット	3×34=102チャンネルビット
合　計	588チャンネルビット

となる。

各フレームの1シンボルのサブコードは、P～Wの8チャンネルある。第2図Bに示すように、各フレームのP～Wの8チャンネルのサブコードを98集めて、1サブコードブロックが構成される。このサブコードブロックが1セクタとされる。したがって、1セクタは、98フレームに相当する。サブコードフレームシンクS_0, S_1としては、データをEFM変調したとき256のパターンにない2つのパターンが選ばれている。

これらP～Wまでのサブコードのうち、Pチャンネルは先頭を示すフラグとされる。Qチャンネルは、コントロールビットとされる。すなわち、Qチャンネルには、データ/オーディオフラグ、アドレス、トラックナンバー、タイムコード等が記録される。

c. インターリーブ処理について

前述したように、この発明の一実施例では、C_1系列とC_2系列とで2重にエラー訂正符号が付加されてデータが記録される。そして、このような符号化を行う際に、第1及び第2の2つのインターリーブ処理が設定できる。

第1のインターリーブ処理では、最大108フレームのインターリーブが施される。そして、この場合には、連続するフレームとともに畳込み符号化がなされる。この第1のインターリーブ処理は、音楽用コンパクトディスクやCD－ROMと全く同様な処理である。したがって、音楽データのようなシーケンシャルなデータを扱う場合や、音楽用コンパクトディスクやCD－ROMとのコンパティビリティを完全に保ちたい場合に好適である。

第2のインターリーブ処理では、最大95フレームのインターリーブが施される。そして、モジュロ98の処理により98フレームで回りながらインターリーブが施される。これにより、エラー訂正符号化が98フレームからなる1セクタ内で完結される。この第2のインターリーブ処理は、1セクタ内でエラー訂正符号化が完結するため、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりする必要性が多い場合に用いて好適である。

なお、第1のインターリーブ処理の場合でも、第2のインターリーブ処理の場合でも、基本的なエンコードプロセス及びデコードプロセスは同様である。すなわち、データを2次元配列し、(28．24．5)リードソロモン符号によりパリティQが付加され、C_2系列の符号化がなされる。そして、インターリーブが施され、(32．28．5)リードソロモン符号によりパリティPが付加され、C_1系列の符号化がなされる。

第1のインターリーブ処理の場合には、最大108フレームのインターリーブが施されて畳込み符号化がなされる。このため、任意のセクタにデータを記録したり、任意のセクタのデータを書き換えたりする場合に、処理が非常に複雑になる。

すなわち、第1のインターリーブ処理を行った場合には、第3図に概念図で示すように、最大108フレームのインターリーブが行われ、連続するフレームのデータとともに畳込み符号化がなされる。例えば第3図においてセクタmのデータは、最大108フレームのインターリーブにより、第3図において斜線で示すような配置となる。この斜線で示す領域のセクタmのデータが書き換えられると、セクタ(m＋1)の全てのC_1系列とセクタ(m＋2)の一部のC_1系列にその影響が生じるとともに、セクタ(m－1)の全てのC_1系列とセクタ(m－2)一部のC_1系列にその影響が生じる。したがって、セクタmのデータを書き換える場合には、セクタ(m＋1)．(m＋2)．(m－1)．(m－2)のC_1系列のパリティをそれに応じて求め直す必要がある。

これに対して、第2のインターリーブ処理は、インターリーブ長が1セクタのフレーム数以下の95フレームとされ、モジュロ98で回りながらインターリーブがかけられている。このため、第4図に概念図で示すように、エラー訂正符号化が1セクタで完結する。したがって、例えばセクタnの書き換えを行っても、セクタ(n＋1)．(n＋2)．(n－1)．(n－2)には何ら影響を及ぼさない。

d．第1のインターリーブ処理の場合のエンコード及びデコードプロセス

第1のインターリーブ処理でデータを符号化する場合のエンコードプロセスについて、第5図を参照しながら説明する。

記録すべき16ビットの12個のデータL_{6n}．R_{6n}．L_{6n+1}．R_{6n+1}．…．L_{6n+5}．R_{6n+5}は、上位8ビット、下位8ビットのデータW_{12n}.A．W_{12n}.B．…．W_{12n+11}.A．W_{12n+11}.Bに分けられて遅延ブロック21に送られる。上位8ビットがA、下位8ビットがBで示されている。

遅延ブロック21で偶数番のデータL_{6n}．R_{6n}．L_{6n+2}．R_{6n+2}．……に対して、遅延素子D1～D12により、2フレーム分の遅延がかけられる。これとともに、遅延ブロック21でデータの並べ換えがなされる。

そして、遅延ブロック21から出力される24シンボルがC_2デコーダ22に送られる。C_2デコーダ22で、(28．24．5)リードソロモン符号により、4シンボルのパリティQ_{12n}．Q_{12n+1}．…．Q_{12n+3}が生成される。

遅延ブロック21の24シンボルの出力データの中央に、C_2デコーダ22で生成された4シンボルのパリティQ_{12n}, Q_{12n+1}, …, Q_{12n+3}が付加され、28シンボルとされる。

この28シンボルが遅延ブロック23に送られる。遅延ブロック23の遅延素子D21～D47により、この28シンボルのそれぞれに対して4の倍数フレームのインターリーブが施される。

遅延ブロック23から出力される28シンボルがC_1エンコーダ24に送られる。C_1エンコーダ24で(32, 28, 5)リードソロモン符号により4シンボルのパリティP_{12n}, P_{12n+1}, …, P_{12n+3}が生成される。

遅延ブロック23から出力される28シンボルの最後に、C_1エンコーダ24で生成された4シンボルのパリティP_{12n}, P_{12n+1}, …, P_{12n+3}が付加され、32シンボルとされる。

この32シンボルが遅延ブロック25に送られる。遅延ブロック25の遅延素子D51～D66により、この32シンボルが1シンボル毎に1フレーム遅延される。

そして、インバータI1～I4及びI5～I8により、パリティシンボルが反転され、エンコードプロセスが完了される。

デコードプロセスは、上述のエンコードプロセスと逆の処理となる。デコードプロセスについて、第6図を参照しながら説明する。

再生された32シンボル(データ24シンボルにパリティPが4シンボル、パリティQが4シンボルが付加されている)が遅延ブロック31に送られる。遅延ブロック31の遅延素子D71～D86により、1シンボル毎のシンボルが1フレーム遅延される。そして、インバータI11～I18によりパリティシンボルが反転される。この32シンボルがC_1デコーダ32に送られる。

C_1デコーダ32から出力される28シンボルが遅延ブロック33に送られる。遅延ブロック33の遅延素子D91～D116により4の倍数フレームずつのインターリーブが解かれる。そして、遅延ブロック33の出力がC_2デコーダ34に送られる。

C_1デコーダ32及びC_2デコーダ34でエラー訂正処理がなされる。C_2デコーダ34から出力される24シンボルは、遅延ブロック35に送られる。遅延ブロック35でデータが時系列順に戻される。そして遅延素子D121～D132により、奇数番のデータが2フレーム遅延され、デコードプロセスが完了される。

第1のインターリーブ処理で符号化した場合の各シンボルを2次元配列上のマップで示すと、第7図に示すようになる。第7図に示すように、第1のインターリーブ処理の場合、最大108フレームのインターリーブが施される。したがって、時系列順のシンボルに対応する1セクタ分のシンボルの座標は下表(表1)のようになる。なお、第8図に示すように、$M_{(i,\ j)}$はシンボルの配置される行番号iと列番号jを示している。

表 1

シンボル　　　座標

$$L_{6n}\begin{cases} W_{12n},A \dashrightarrow M_{(0,\ n)} \\ W_{12n},B \dashrightarrow M_{(1,\ n+3)} \end{cases}$$

$$R_{6n}\begin{cases} W_{12n+1},A \dashrightarrow M_{(6,\ n+24)} \\ W_{12n+1},B \dashrightarrow M_{(7,\ n+27)} \end{cases}$$

$$L_{6n+1}\begin{cases} W_{12n+2},A \dashrightarrow M_{(16,\ n+62)} \\ W_{12n+2},B \dashrightarrow M_{(17,\ n+65)} \end{cases}$$

$$R_{6n+1}\begin{cases} W_{12n+3},A \dashrightarrow M_{(22,\ n+86)} \\ W_{12n+3},B \dashrightarrow M_{(23,\ n+89)} \end{cases}$$

$$L_{6n+2}\begin{cases} W_{12n+4},A \dashrightarrow M_{(2,\ n+8)} \\ W_{12n+4},B \dashrightarrow M_{(3,\ n+11)} \end{cases}$$

$$R_{6n+2}\begin{cases} W_{12n+5},A \dashrightarrow M_{(8,\ n+32)} \\ W_{12n+5},B \dashrightarrow M_{(9,\ n+35)} \end{cases}$$

$$L_{6n+3}\begin{cases} W_{12n+6},A \dashrightarrow M_{(18,\ n+70)} \\ W_{12n+6},B \dashrightarrow M_{(19,\ n+73)} \end{cases}$$

$$R_{6n+3}\begin{cases} W_{12n+7},A \dashrightarrow M_{(24,\ n+94)} \\ W_{12n+7},B \dashrightarrow M_{(25,\ n+97)} \end{cases}$$

$$L_{6n+4}\begin{cases} W_{12n+8},A \dashrightarrow M_{(4,\ n+16)} \\ W_{12n+8},B \dashrightarrow M_{(5,\ n+19)} \end{cases}$$

$$R_{6n+4}\begin{cases} W_{12n+9},A \dashrightarrow M_{(10,\ n+40)} \\ W_{12n+9},B \dashrightarrow M_{(11,\ n+43)} \end{cases}$$

$$L_{6n+5}\begin{cases} W_{12n+10}A \dashrightarrow M_{(20,\ n+78)} \\ W_{12n+10}B \dashrightarrow M_{(21,\ n+81)} \end{cases}$$

$$R_{6n+5}\begin{cases} W_{12n+11}A \dashrightarrow M_{(26,\ n+102)} \\ W_{12n+11}B \dashrightarrow M_{(27,\ n+105)} \end{cases}$$

$$\overline{Q}_{12n} \dashrightarrow M_{(12,\ n+48)}$$

$$\overline{Q}_{12n+1} \dashrightarrow M_{(13,\ n+51)}$$

$$\overline{Q}_{12n+2} \dashrightarrow M_{(14,\ n+54)}$$

$$\overline{Q}_{12n+3} \dashrightarrow M_{(15,\ n+57)}$$

$$\overline{P}_{12n} \dashrightarrow M_{(28,\ n)}$$

$$\overline{P}_{12n+1} \dashrightarrow M_{(29,\ n-1)}$$

$$\overline{P}_{12n+2} \dashrightarrow M_{(30,\ n)}$$

$$\overline{P}_{12n+3} \dashrightarrow M_{(31,\ n-1)}$$

各シンボルが第7図に示すように2次元配列されている場合、第6図に示すデコードプロセスに対応したアドレス操作を行って列方向に読み出し／書き込みを行っていくことにより、インターリーブが解除され、データが復号できる。

すなわち、遅延ブロック31に対応して、偶数番目の行のシンボルを1フレーム遅延させて列方向にデータを読み出してライン81で示すように C_1 系列が復号できる。

その後、遅延ブロック33に対応して第1行に対して（27×4＝108）フレーム、第2行に対して（26×4＝104）フレーム、第3行に対して（25×4＝100）フレーム、…、それぞれ遅延させて読みだすことにより、ライン82で示すように C_2 系列が復号できる。

なお、遅延ブロック35に対応して奇数番のデータが2フレーム分遅延されるので、奇数番データは破線に四角で示すものが復号時に出力されることになる。

e．第2のインターリーブ処理の場合のエンコード及びデコードプロセス

第2のインターリーブ処理でデータを符号化する場合のエンコードプロセスについて、第9図を参照しながら説明する。

16ビットの12個のデータ L_{6n}, R_{6n}, L_{6n+1}, R_{6n+1}, …, L_{6n+5}, R_{6n+5} は、上位8ビット、下位8ビットのデータ $W_{12n}A$, $W_{12n}B$, …, W_{12n+11}, A, W_{12n+11}, Bに分けられて遅延ブロック41に送られる。

遅延ブロック41で偶数番のデータ L_{6n}, R_{6n}, L_{6n+2}, R_{6n+2}, …に対して、遅延素子D151～D162により、2フレーム分の遅延がかけられる（遅延ブロック41はモジュロ98で回っている）。これとともに、遅延ブロック41でデータの並べ換えがなされる。

そして、遅延ブロック41から出力される24シンボルが C_2 デコーダ42に送られる。C_2 デコーダ42で、（28，24，5）リードソロモン符号により、4シンボルのパリティ Q_{12n}, Q_{12n+1}, …, Q_{12n+3} が生成される。

遅延ブロック42の24シンボルの出力データの中央に、C_2 デコーダ42で生成された4シンボルのパリティ Q_{12n}, Q_{12n+1}, …, Q_{12n+3} が付加され、28シンボルとされる。

この28シンボルが遅延ブロック43に送られる。遅延ブロック43の遅延素子D171～D197により、4フレーム、3フレーム、4フレーム、3フレーム、…のインターリーブが施される（遅延ブロック43は、モジュロ98で回っている）。

遅延ブロック43から出力される28シンボルが C_1 エンコーダ44に送られる。C_1 エンコーダ44で（32，28，5）リードソロモン符号により4シンボルのパリティ P_{12n}, P_{12n+1}, …, P_{12n+3} が生成される。

遅延ブロック43から出力される28シンボルの最後に、C_1 エンコーダ44で生成された4シンボルのパリティ P_{12n}, P_{12n+1}, …, P_{12n+3} が付加され、32シンボルとされる。

この32シンボルが遅延ブロック45に送られる。遅延ブロック45の遅延素子D201～D216により、この32シンボルが1シンボル毎に1フレーム遅延される（遅延ブロック45はモジュロ98で回っている）。

そして、インバータ121～124及び125～128により、パリティシンボルが反転され、エンコードプロセスが完了される。

デコードプロセスは、上述のエンコードプロセスと逆の処理となる。デコードプロセスについて、第10図を参照しながら説明する。

再生された３２シンボルが遅延ブロック５１に送られる。遅延ブロック５１の遅延素子Ｄ２２１～Ｄ２３６により、１シンボル毎のシンボルが１フレーム遅延される（遅延ブロック５１はモジュロ９８で回っている）。そして、インバータＩ１３１～Ｉ１３８により再生されたパリティシンボルが反転される。この３２シンボルがC_1デコーダ５２に送られる。

C_1デコーダ５２から出力される２８シンボルが遅延ブロック５３に送られる。遅延ブロック５３の遅延素子Ｄ２４１～Ｄ２６７により４フレーム、３フレーム、４フレーム、３フレーム、…、のインターリーブが解かれる（遅延ブロック５３はモジュロ９８で回っている）。そして、遅延ブロック５３の出力がC_2デコーダ５４に送られる。

C_1デコーダ５２及びC_2デコーダ５４でエラー訂正処理がなされる。C_2デコーダ５４から出力される２４シンボルは、遅延ブロック５５に送られる。遅延ブロック５５でデータが時系列順に戻される。そして遅延素子Ｄ２７１～Ｄ２８２により、偶数番目のデータが２フレーム遅延され、デコードプロセスが完了される（遅延ブロック５５はモジュロ９８で回っている）。

第２のインターリーブ処理で符号化した場合の各シンボルを２次元配列上のマップで示すと、第１１図に示すようになる。第１１図に示すように、第２のインターリーブ処理の場合、最大９５フレームのインターリーブが施される。そして、モジュロ９８をとることにより、９７列目のフレームまで遅延されると０フレームに戻る。したがって、時系列順の１セクタ分のシンボルW_{12n}, A, W_{12n}, B, W_{12n+1}, A, W_{12n+1}, B…に対応する１セクタ分のシンボルの座標$M_{(i,j)}$は下表（表２）のようになる。$M_{(i,j)}$は第１２図に示すように各シンボルの座標を示し、ｉは行番号、ｊは列番号、ｎ＝０～９７である。

表　2

シンボル		座標
L_{6n}	W_{12n}, A	——→ $M_{(0,n)}$
	W_{12n}, B	——→ $M_{(1,n+3)}$
R_{6n}	W_{12n+1}, A	——→ $M_{(6,n+21)}$
	W_{12n+1}, B	——→ $M_{(7,n+24)}$
L_{6n+1}	W_{12n+2}, A	——→ $M_{(16,n+34)}$
	W_{12n+2}, B	——→ $M_{(17,n+37)}$
R_{6n+1}	W_{12n+3}, A	——→ $M_{(22,n+75)}$
	W_{12n+3}, B	——→ $M_{(23,n+78)}$
L_{6n+2}	W_{12n+4}, A	——→ $M_{(2,n+7)}$
	W_{12n+4}, B	——→ $M_{(3,n+10)}$
R_{6n+2}	W_{12n+5}, A	——→ $M_{(8,n+28)}$
	W_{12n+5}, B	——→ $M_{(9,n+31)}$
L_{6n+3}	W_{12n+6}, A	——→ $M_{(18,n+61)}$
	W_{12n+6}, B	——→ $M_{(19,n+64)}$
R_{6n+3}	W_{12n+7}, A	——→ $M_{(24,n+82)}$
	W_{12n+7}, B	——→ $M_{(25,n+85)}$
L_{6n+4}	W_{12n+8}, A	——→ $M_{(4,n+14)}$
	W_{12n+8}, B	——→ $M_{(5,n+17)}$
R_{6n+4}	W_{12n+9}, A	——→ $M_{(10,n+35)}$
	W_{12n+9}, B	——→ $M_{(11,n+38)}$
L_{6n+5}	W_{12n+10}, A	——— $M_{(20,n+68)}$
	W_{12n+10}, B	——→ $M_{(21,n+71)}$
R_{6n+5}	W_{12n+11}, A	——→ $M_{(26,n+89)}$
	W_{12n+11}, B	——→ $M_{(27,n+92)}$
	$\overline{Q}_{12n}$	——→ $M_{(12,n+42)}$
	$\overline{Q}_{12n+1}$	——→ $M_{(13,n+45)}$
	$\overline{Q}_{12n+2}$	——→ $M_{(14,n+49)}$
	$\overline{Q}_{12n+3}$	——→ $M_{(15,n+52)}$
	$\overline{P}_{12n}$	——→ $M_{(28,n)}$
	$\overline{P}_{12n+1}$	——→ $M_{(29,n+97)}$
	$\overline{P}_{12n+2}$	——→ $M_{(30,n)}$
	$\overline{P}_{12n+3}$	——→ $M_{(31,n+97)}$

各シンボルが第１１図に示すように２次元配列されている場合、第１０図に示すデコードプロセスに対応したアドレス操作を行って列方向に読み出し／書き込みを行っていくことにより、インターリーブが解除され、データが復号できる。

すなわち、遅延ブロック５１に対応して、偶数番目の行のシンボルを１フレーム遅延させて列方向にデータを読み出してC_1系列が復号できる。この際、モジュロ９８がとられる。

その後、遅延ブロック５３に対応して第１行に

対して95フレーム、第2行に対して91フレーム、第3行に対して88フレーム、…、それぞれ遅延させて読みだすことにより、C_2系列が復号できる。なお、この際にもモジュロ98がとられる。

第11図では、n＝0のときのデコード処理と、n＝60の場合のデコード処理が示されている。n＝0のときには、ライン91で示すようにC_1系列が復号される。この際、モジュロ98がとられるので、0列のフレームと97列のフレームのシンボルからC_1系列が復号される。またn＝0の場合には、ライン92で示すようにC_2系列が復号される。

n＝60の場合には、ライン93で示すようにC_1系列が復号される。また、n＝60の場合には、ライン94で示すようにC_2系列が復号される。この際、モジュロ98がとられるので、60フレームから38フレーム分遅延されるべき$W_{72,0}$.Bのシンボルは、$M_{(11,0)}$の位置に戻る。以下のシンボルは同様である。

f. 一実施例におけるエンコード・デコード処理

上述のエンコードプロセス及びデコードプロセスは、データをRAMに蓄えておき、アドレスを制御することにより実現されている。

すなわち、第1図における符号化回路3は、上述の第1のインターリーブ処理の場合のエンコードプロセス及び第2のインターリーブ処理の場合のエンコードプロセスとが選択可能とされていて、この符号化回路3は、第13図に示すように、RAM61と、エンコーダ62と、第1のインターリーブ処理に対応してアドレスを発生するアドレス発生回路63と、第2のインターリーブ処理に対応してアドレスを発生するアドレス発生回路64とからなる。そして、アドレス発生回路63から出力される第1のインターリーブ処理に対応するアドレスとアドレス発生回路64から出力される第2のインターリーブ処理に対応するアドレスとを、スイッチ手段65を介して選択的にRAM61に供給することにより、第1のインターリーブ処理の場合のエンコードプロセスと第2のインターリーブ処理の場合のエンコードプロセスとの両者が設定可能とされる。

また、第1図における復号化回路10は、上述の第1のインターリーブ処理の場合のデコードプロセス及び第2のインターリーブ処理の場合のデコードプロセスとが選択可能とされていて、この復号化回路10は、第14図に示すように、RAM71と、エンコーダ72と、第1のインターリーブ処理に対応してアドレスを発生するアドレス発生回路73と、第2のインターリーブ処理に対応してアドレスを発生するアドレス発生回路74とからなる。そして、アドレス発生回路73から出力される第1のインターリーブ処理に対応するアドレスと、アドレス発生回路74から出力される第2のインターリーブ処理に対応するアドレスとを、スイッチ手段75を介して選択的にRAM71に供給することにより、第1のインターリーブ処理の場合のデコードプロセスと第2のインターリーブ処理の場合のデコードプロセスとの両者が設定可能とされる。

〔発明の効果〕

この発明によれば、エラー訂正符号化が98フレームからなる1セクタで完結できるインターリーブ処理が設定できる。エラー訂正符号化が1セクタ内で完結できるため、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりした際、その影響が他のセクタに及ばず、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換える際に複雑な信号処理が必要とならないとともに、データの記録容量が低下しない。

更に、エラー訂正符号化が1セクタ内で完結しているので、データの読み出し／書き込みの際に複数のセクタのデータを取り込む必要がなく、アクセス時間が短縮できる。

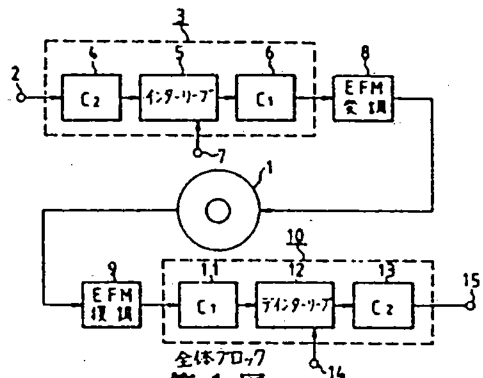
4.図面の簡単な説明

第1図はこの発明の一実施例の全体構成を示すブロック図、第2図はこの発明の一実施例の記録フォーマットの説明に用いる略線図、第3図及び

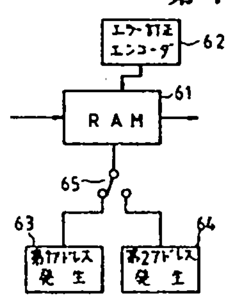
第4図はこの発明の一実施例の説明に用いる略線図、第5図及び第6図は第1のインターリーブ処理の説明に用いるブロック図、第7図及び第8図は第1のインターリーブ処理の説明に用いる略線図、第9図及び第10図は第2のインターリーブ処理の説明に用いるブロック図、第11図及び第12図は第2のインターリーブ処理の説明に用いる略線図、第13図は符号化装置の一例のブロック図、第14図は符号化装置の一例のブロック図である。

図面における主要な符号の説明

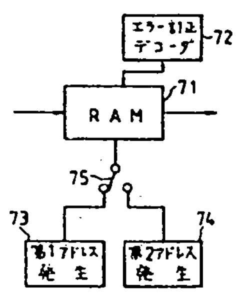
1：光ディスク、　2：データ入力端子、
3：符号化回路、　5：インターリーブ遅延回路、
8：ＥＦＭ変調回路、　9：ＥＦＭ復調回路、
10：復号化回路、
12：ディインターリーブ回路、
15：出力端子、　61，71：ＲＡＭ、
63，64，73，74：アドレス発生回路。



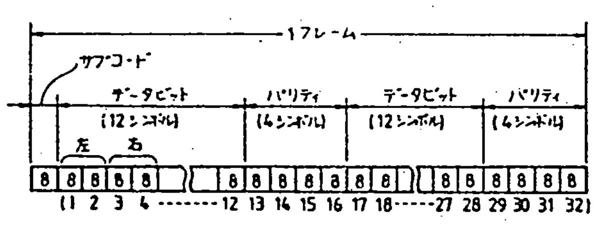
全体ブロック
第1図



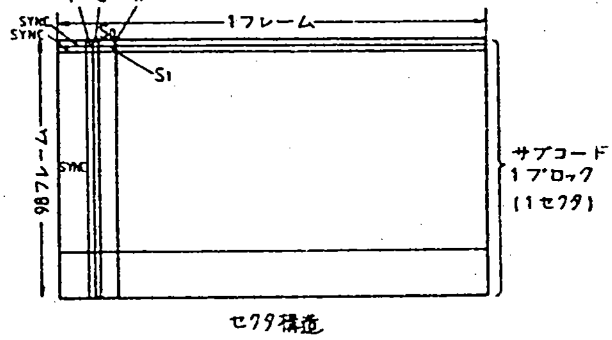
符号化装置
第13図



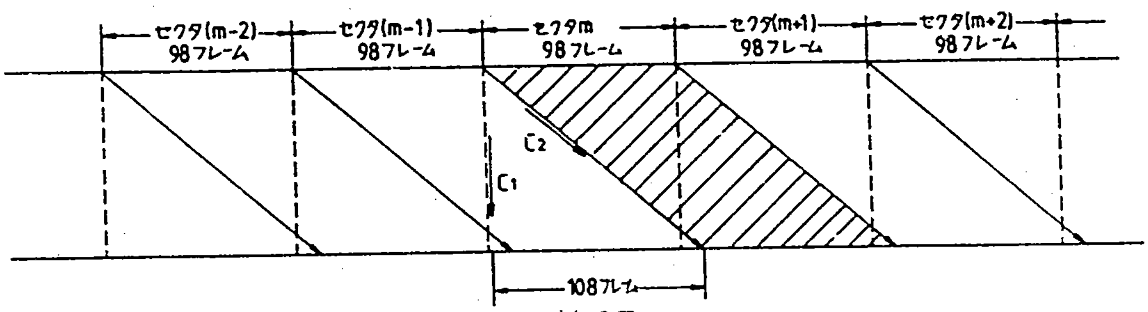
復号装置
第14図



フレーム構造
第2図A

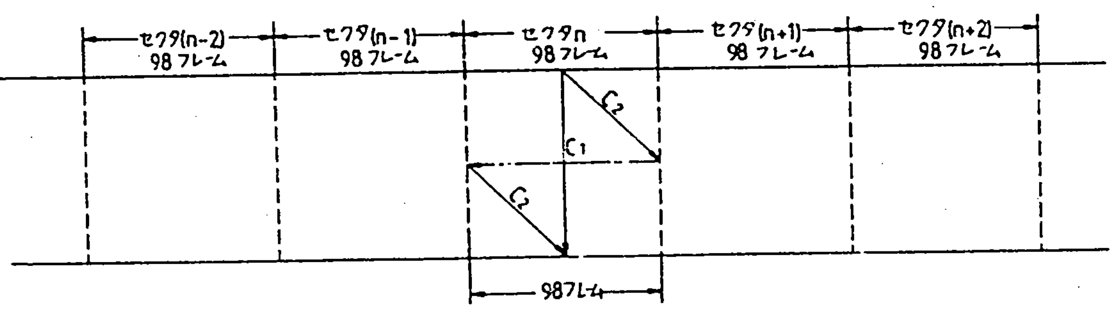


セクタ構造
第2図B



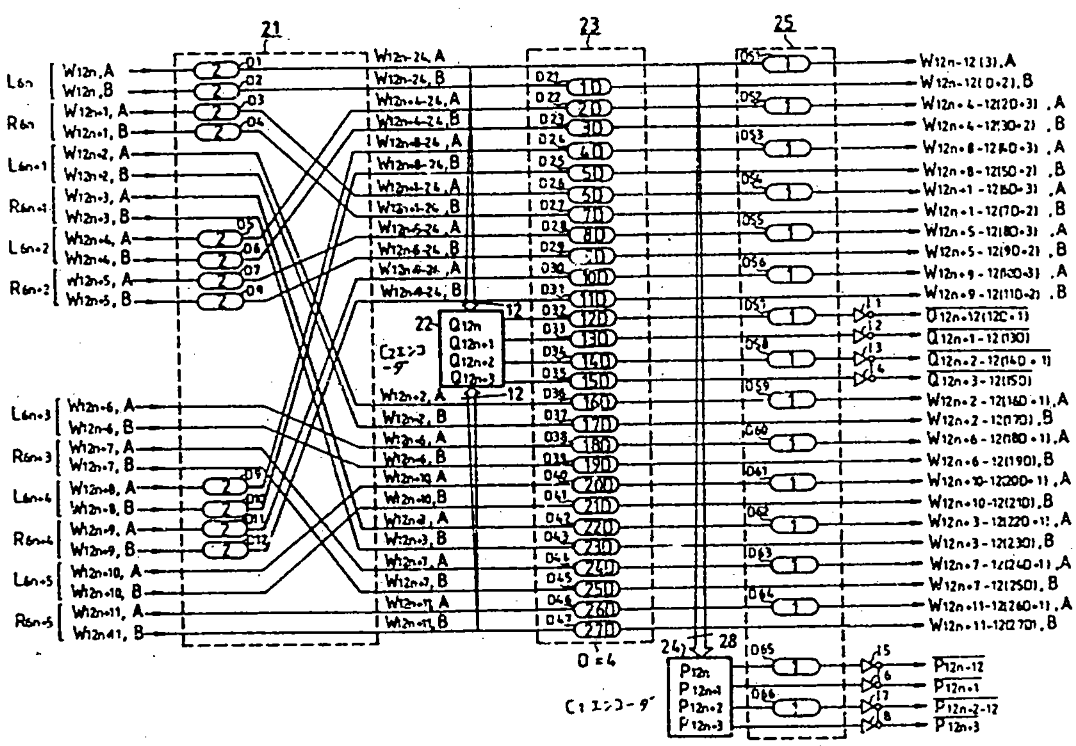
説明図

第 3 図



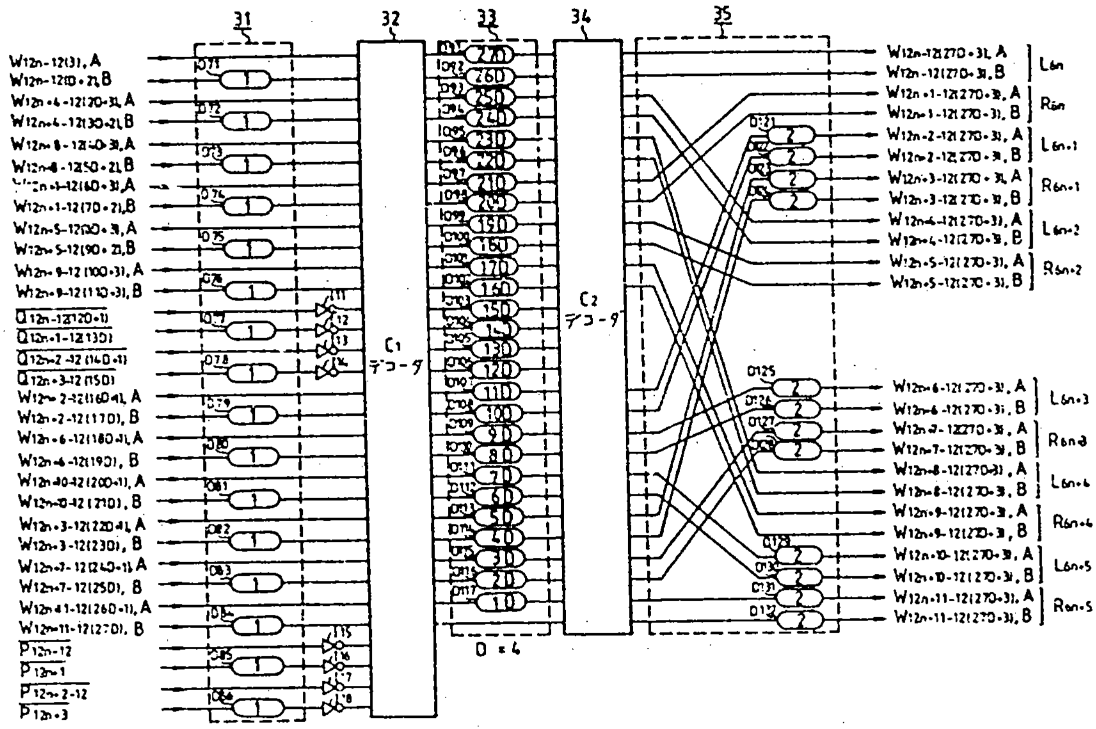
説明図

第 4 図



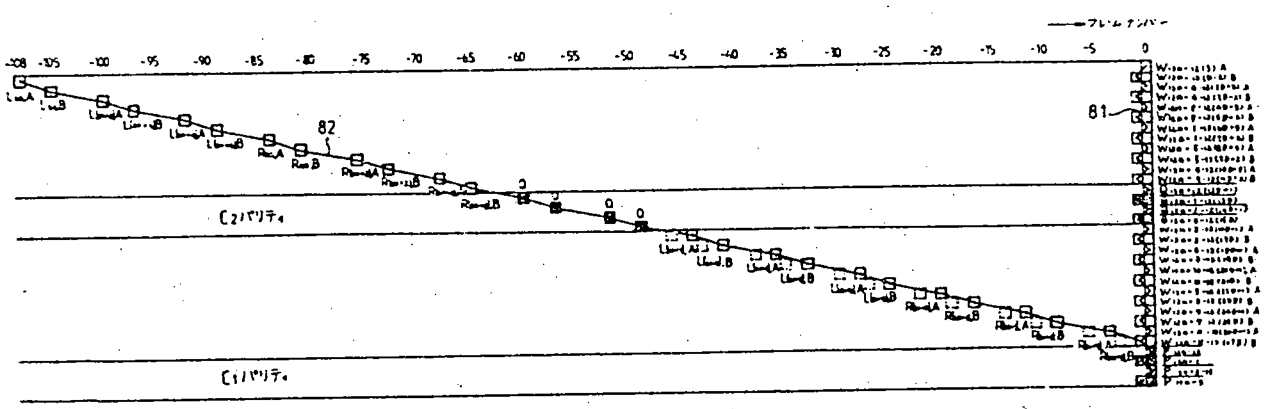
第1のインターリーブ処理の場合のエンコードプロセス

第 5 図



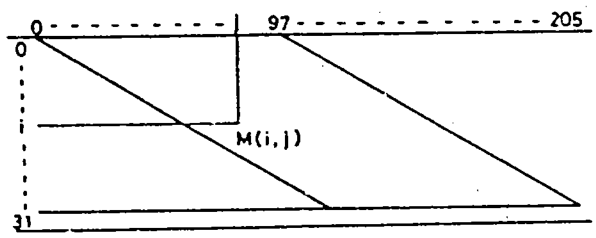
第1のインターリーブ処理の場合のデコードプロセス

第 6 図

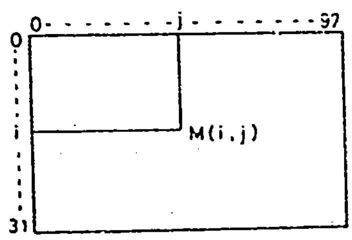


第1のエンコード処理の場合のマップ

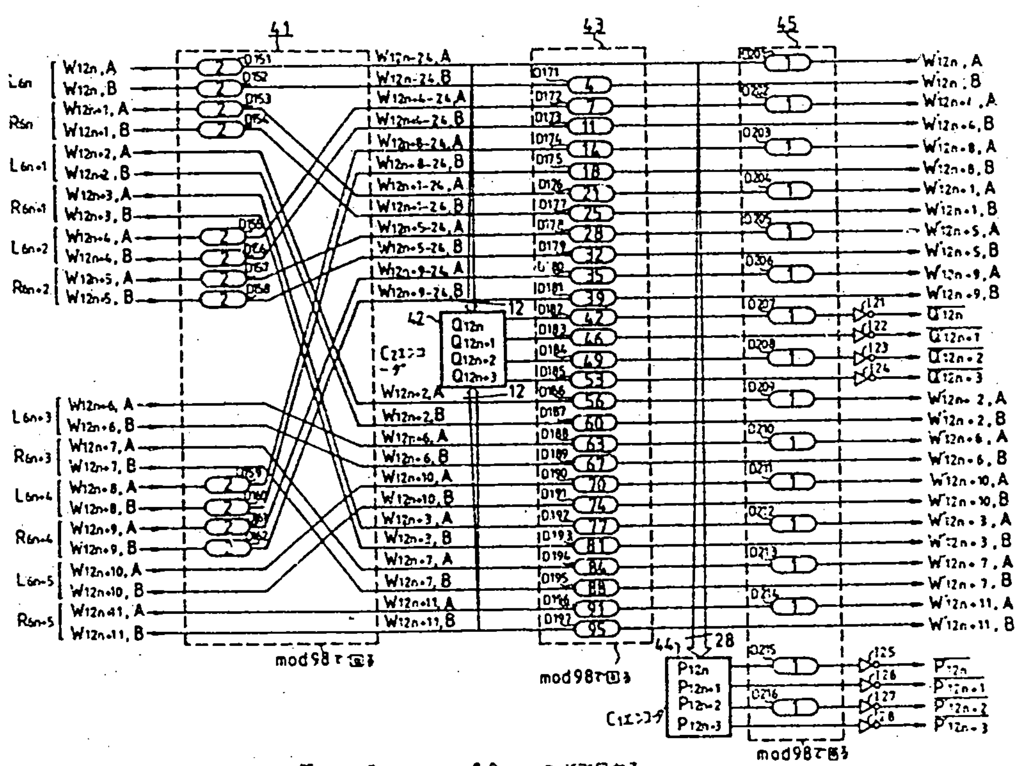
第 7 図



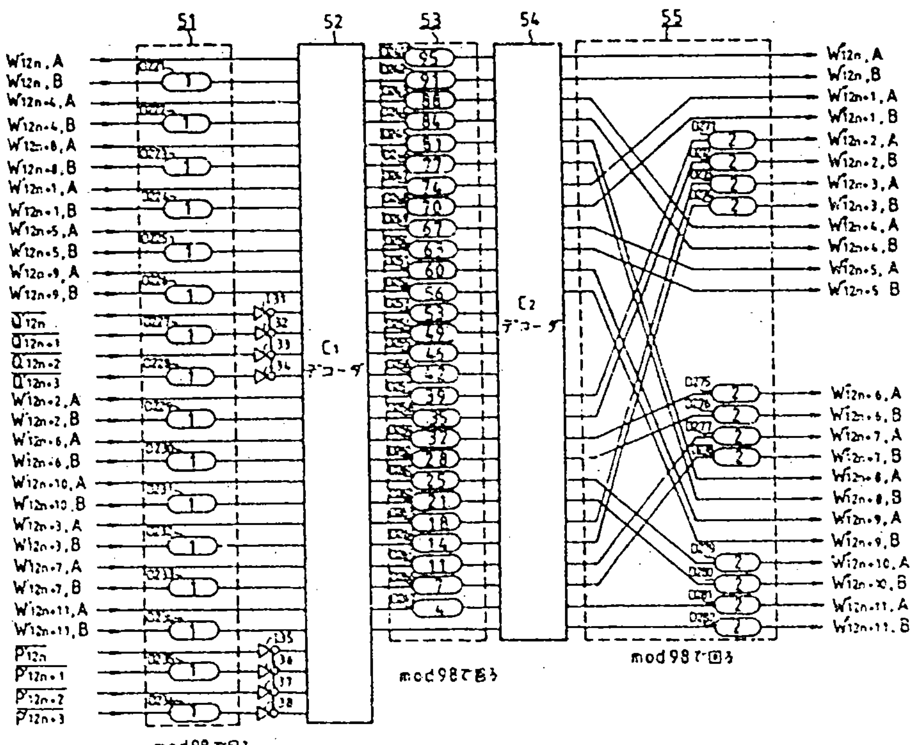
説明図
第8図



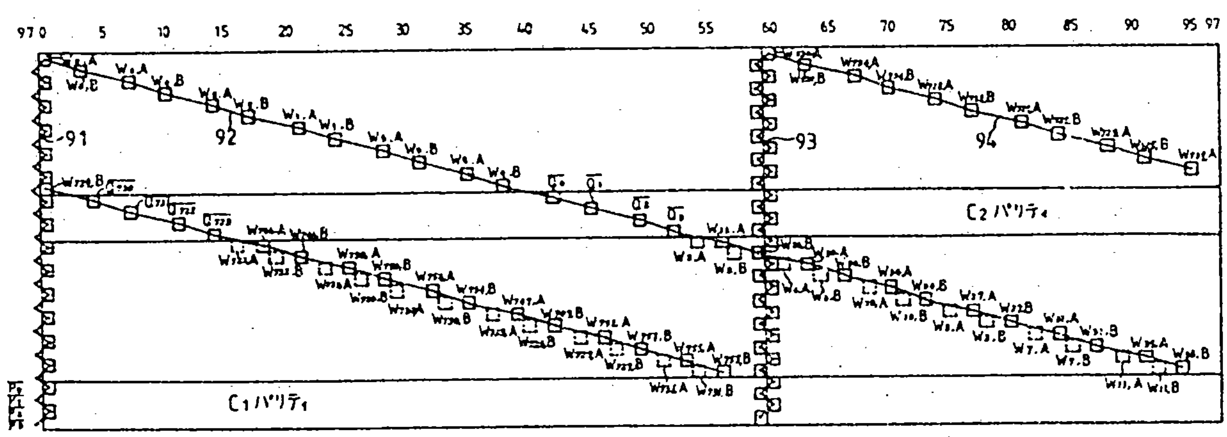
説明図
第12図



第2のインターリーブ処理の場合のエンコードプロセス
第9図



第2のインターリーブ処理の場合のデコードプロセス
第10図



第2のインターリーブ処理の場合のマップ
第11図

手続補正書

昭和63年　9月10日

特許庁長官　吉　田　文　毅　殿

1. 事件の表示

昭和63年特許願第118567号

2. 発明の名称

ディジタルデータ記録／再生装置

3. 補正をする者

事件との関係　　特許出願人

住所　東京都品川区北品川6丁目7番35号

名称　(218) ソ　ニ　ー　株　式　会　社

代表取締役　大　賀　典　雄

4. 代理人　〒170

住所　東京都豊島区東池袋1丁目48番10号

25山京ビル 420号　℡.(03)980-0339

氏名　(8276) 弁理士　杉　浦　正　知

5. 補正命令の日付　　自　　発

6. 補正の対象

明細書の発明の詳細な説明の欄

63. 9. 12

7. 補正の内容

(1)明細書中、第6頁第12行目に「なされる。」とあるを「できると有用である。」と補正する。

(2)同、第15頁第3行目及び第4行目、第15頁第5行目、第15頁第8行目、第15頁第9行目、第15頁第11行目、第15頁第18行目に、それぞれ「ディインターリーブ」とあるを「デインターリーブ」と補正する。

(3)同、第21頁第5行目及び第6行目に「全てのC，系列とセクタ（m−2）一部の」とあるを削除する。

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平1－287872

⑤Int. Cl.⁴　識別記号　庁内整理番号　④公開　平成1年(1989)11月20日
G 11 B 20/12　8524－5D

審査請求　未請求　請求項の数 4　（全16頁）

⑤発明の名称　デイジタルデータ記録／再生装置

㉑特　願　昭63－118567
㉒出　願　昭63(1988)5月16日

㉒発明者　佐古　曜一郎　東京都品川区北品川6丁目7番35号　ソニー株式会社内
㉒発明者　小川　博司　東京都品川区北品川6丁目7番35号　ソニー株式会社内
㉑出願人　ソニー株式会社　東京都品川区北品川6丁目7番35号
㉔代理人　弁理士　杉浦　正知

明　細　書

1. 発明の名称

ディジタルデータ記録／再生装置

2. 特許請求の範囲

(1)入力データに対して所定のエラー訂正符号を付加する符号化手段と、上記符号化されたデータを記録媒体に記録する手段とを有するディジタルデータ記録装置において、

上記符号化手段は、上記入力データを一時的に蓄積するメモリと、

上記メモリに対するアドレスを制御するアドレス制御手段とを有し、

上記アドレス制御手段は、第1のアドレス制御と、第2のアドレス制御とが設定可能とされたディジタルデータ記録装置。

(2)上記第1のアドレス制御により記録されるデータと上記第2のアドレス制御により記録されるデータとを選択的に設定するための選択信号を上記記録媒体に記録するようにした請求項1記載のディジタルデータ記録装置。

(3)記録媒体の記録データを再生する手段と、上記再生されたデータを復号する復号化手段とを有するディジタルデータ再生装置において、

上記復号化手段は、上記再生データを一時的に蓄積するメモリと、

上記メモリに対するアドレスを制御するアドレス制御手段とを有し、

上記アドレス制御手段は、上記記録媒体の再生データが上記第1のアドレス制御により記録されたデータか上記第2のアドレス制御により記録されたデータかに応じて選択的にアドレス制御を設定可能とするようにしたディジタルデータ再生装置。

(4)上記光学的記録媒体に記録されている上記選択信号を再生して、上記光学的記録媒体の再生データが上記第1のアドレス制御により記録されたデータか上記第2のアドレス制御により記録されたデータかを判断するようにした請求項3記載のディジタルデータ再生装置。

3. 発明の詳細な説明

〔産業上の利用分野〕

この発明は、ディジタルデータ記録／再生装置に関するもので、例えば音楽用のコンパクトディスクに対応した形態で光学的記録媒体にデータを記録／再生するようにしたディジタルデータ記録／再生装置に係わる。

〔発明の概要〕

この発明は、コンパクトディスクに対応した形態でデータを記録媒体に記録／再生するようにしたディジタルデータ記録／再生装置において、コンパクトディスクと同様のインターリーブ処理と、1記録単位（セクタ）に対応する長さのインターリーブ処理とを行えるようにし、セクタ完結型の符号化及びその復号化を実行可能とすることにより、任意のセクタにデータを書き込む処理や任意のセクタのデータを書き換える処理を簡単に行えるようにしたものである。

〔従来の技術〕

オーディオ信号をディジタル化して光学的に記録したコンパクトディスクが広く知られている。このコンパクトディスクは、例えば直径12cmで、約500Mバイト以上ものデータを記録できる記録媒体である。したがって、このコンパクトディスクを大量のデータを記録するディジタルデータ記録媒体として用いることができる。

このことに着目し、コンパクトディスクの音楽記録領域にオーディオデータ以外のディジタルデータを記録できるようにしたCD－ROMが規格化されている。CD－ROMでは、音楽用のコンパクトディスクと同じ態様でディジタルデータが記録される。すなわち、CD－ROMでは、光ディスクとして音楽用のコンパクトディスクと同様の直径12cmでスパイラル状にトラックが形成されているものが用いられる。記録データは、CIRC (Cross Interleave Reed-Solomon Code) により2重に符号化され、EFM変調（8－14変調）されて光ディスクに光学的に記録される。

このようなCD－ROMにおいて、データの記録単位となる1セクタ（1ブロック）は、98フレームからなる1サブコードブロックから構成される。つまり、音楽用のコンパクトディスクでは、1フレーム当たりP～Wまでの8ビットのサブコード（R～Wはユーザーズビットとも称される）が用意されている。このサブコードに、98フレーム分で1つの情報単位（アドレス）となっていることから、98フレームがブロックと呼ばれる。

CD－ROMは、基本的に読み出し専用の記録媒体である。CD－ROMは、記憶容量が大きく、大量複製ができ、情報の劣化が少ない等の特徴がある。このような特徴を生かし、各種辞典類のデータや研究資料データを記録するのにCD－ROMが利用されている。

〔発明が解決しようとする課題〕

近年、追記型の光学記録媒体や、光磁気ディスクのように消去、再記録が可能な光学記録媒体が開発されている。このような追記型の光学記録媒体や消去、再記録可能な光学記録媒体を、コンパクトディスクと同様な態様で用い、ディジタルデータを記録することが提案されている。追記型のデータ記録媒体としてのコンパクトディスク（以下、CD－WOと称する）や、消去、再記録可能なデータ記録媒体としてのコンパクトディスク（以下、CD－イレーザブルと称する）では、データを追記したり、再記録したりすることができるので、再生専用のCD－ROM以上に、幅広い分野での利用が期待できる。

ところで、CD－WOやCD－イレーザブルでは、セクタ単位でデータの書き込み／読み出しがなされる。これに対して、音楽用のコンパクトディスクは再生専用で、データが時系列順に再生される。

音楽用のコンパクトディスクは、音楽データのようなシーケンシャルなデータに最適なように、記録データには最大108フレームのインターリーブが施されている。このため、音楽用のコンパクトディスクと全く同一の信号処理形態でCD－WOやCD－イレーザブルを実現した場合、任意

のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりする場合に、複雑な信号処理が必要になる。このような複雑な処理を必要としないために、CD－WOやCD－イレーザブルを、音楽用のコンパクトディスクと全くことなった信号フォーマットに変更すると、コンパクトディスク、CD－ROM、CD－WO、CD－イレーザブルの間のコンパティビリティを失うことになる。

すなわち、コンパクトディスクでは、前述したように、2次元配列上の各列毎のシンボルをリードソロモン符号を用いてC_2系列のパリティを生成付加し、インターリーブ遅延を行った後、リードソロモン符号を用いてC_1系列のパリティを生成付加するCIRCが用いられている。このようなCIRCでは、音楽データのようなシーケンシャルなデータに対して最適化するように、最大108フレームのインターリーブが施され、畳込み符号化がなされる。これに対して、CD－WOやCD－イレーザブルにおいてデータの書き込み、読み出しの単位となる1セクタは、98フレームからなるサブコードブロックから構成される。

したがって、任意の1セクタのデータを書き換えると、前2セクタと後2セクタとにその影響が及ぶ。すなわち、任意の1セクタのデータを書き換えると、その前後2セクタのデータに関するC_1系列のパリティが変わってくる。

このことから、任意の1セクタのデータの書き換えを行う場合には、そのセクタのデータとその前後2セクタのデータとを取り込み、C_1系列のパリティを求め直す処理が必要になってくる。

そこで、例えば特願昭62－244996号明細書に示されるように、2セクタ以上連続する全て「0」のデータのセクタをデータ記録用のセクタの間に挿入し、例えば3セクタ毎のセクタをデータ記録用のセクタとすることが提案されている。ところが、このようにすると、パリティを求め直す処理は必要なくなるが、データの記録容量が最悪⅓程度に減少してしまう。

したがって、この発明の目的は、任意のセクタにデータを記録する場合や、任意のセクタのデータを書き換える際に、複雑な処理を必要としないとともに、データ記録容量が削減されることのないディジタルデータ記録／再生装置を提供することにある。

また、従来では、このように畳込み符号が用いられているため、任意のセクタのデータを取り込むのに、少なくともこれに連続する2セクタのデータを取り込まなければならず、アクセス時間が長く必要になる。

この発明の更に他の目的は、アクセス時間の短縮化がはかれるディジタルデータ記録／再生装置を提供することにある。

〔課題を解決するための手段〕

この発明は、入力データに対して所定のエラー訂正符号を付加する符号化手段と、符号化されてデータを記録媒体に記録する手段とを有するディジタルデータ記録装置において、

符号化手段は、入力データを一時的に蓄積するメモリと、

メモリに対するアドレスを制御するアドレス制御手段とを有し、

アドレス制御手段は、第1のアドレス制御と、第2のアドレス制御とが設定可能とされたディジタルデータ記録装置である。

第1のアドレス制御により記録されるデータと第2のアドレス制御により記録されるデータとを選択的に設定するための選択信号が記録媒体に記録される。

また、この発明は、記録媒体の記録データを再生する手段と、再生されたデータを復号する復号化手段とを有するディジタルデータ再生装置において、

復号化手段は、再生データを一時的に蓄積するメモリと、

メモリに対するアドレスを制御するアドレス制御手段とを有し、

アドレス制御手段は、記録媒体の再生データが第1のアドレス制御により記録されたデータか第

2のアドレス制御により記録されたデータかに応じて選択的にアドレス制御を設定可能とするようにしたディジタルデータ再生装置である。

記録媒体に記録されている選択信号を再生して、記録媒体の再生データが第1のアドレス制御により記録されたデータか第2のアドレス制御により記録されたデータかを判断するようにしている。

〔作用〕

コンパクトディスクと同一形態の光ディスクを用いてディジタルデータの記録／再生が行われる。この際、第1及び第2の2つのインターリーブ処理が設定できる。第1のインターリーブ処理では、全体インターリーブ長108フレームとされる。この第1のインターリーブ処理は、音楽データのようなシーケンシャルなデータを扱う場合や、音楽用コンパクトディスクやCD－ROMとのコンパティビリティを完全に保ちたい場合に好適である。

第2のインターリーブ処理では、全体インターリーブ長95フレームとされる。そして、モジュロ98の処理により98フレームで回りながらインターリーブが施される。これにより、エラー訂正符号化が98フレームからなる1セクタ内で完結される。この第2のインターリーブ処理は、1セクタ内でエラー訂正符号化が完結するため、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりする必要性が多い場合に用いて好適である。

〔実施例〕

この発明の実施例について以下の順序に従って説明する。

a．記録／再生装置の概要
b．フレーム構造及びセクタ構造
c．インターリーブ処理について
d．第1のインターリーブ処理の場合のエンコード及びデコードプロセス
e．第2のインターリーブ処理の場合のエンコード及びデコードプロセス
f．一実施例におけるエンコード・デコード処理

a．記録／再生装置の概要

第1図は、この発明が適用されたディジタルデータ記録／再生装置の概要を示すものである。第1図において、1はディジタルデータが光学的に記録／再生される光ディスクである。光ディスク1としては、追記型の光ディスクや消去．再記録可能な光ディスク、例えば光磁気ディスクを用いることができる。光ディスク1は、音楽用のコンパクトディスクと同様の形態とされている。すなわち、光ディスク1の直径は12cmであり、光ディスク1にはスパイラル状のトラックが形成される。そして、光ディスク1は、CLV（線速度一定）でもって回転される。

記録時には、光ディスク1に記録すべきデータがデータ入力端子2に供給される。この記録データが符号化回路3に供給される。符号化回路3は、C_2エンコーダ4と、インターリーブ遅延回路5と、C_1エンコーダ6とから構成される。入力端子1からのデータは、所定フレーム構造に展開され、符号化回路3で、C_1系列とC_2系列とで2重に符号化される。インターリーブ遅延回路5には、端子7から選択信号が供給される。インターリーブ遅延回路5は、後に詳述するように、インターリーブ長の異なる第1、第2の2つのインターリーブ処理が施せる。第1のインターリーブ処理の場合には、最大108フレームのインターリーブが施される。そして、第1のインターリーブ処理の場合は、畳込み符号化がなされる。第2のインターリーブ処理の場合には、インターリーブ長が98になり、第2のインターリーブ処理では、エラー訂正符号化が1セクタ内で完結される。

符号化回路3で、2重にエラー訂正符号が付加されたデータは、EFM変調回路8でEFM変調（8－14変調）され、光ディスク1に記録される。

光ディスク1の記録データを再生する際には、上述の記録時と逆のシーケンスで処理がなされる。すなわち、光ディスク1の再生データがEFM

復調回路9に供給され、EFM復調される。EFM復調回路9の出力が復号化回路10に供給される。復号化回路10は、C_1デコーダ11と、ディインターリーブ遅延回路12と、C_2デコーダ13とから構成される。ディインターリーブ遅延回路12は、インターリーブ遅延回路5で設定できる2つのインターリーブ処理に対応して、第1及び第2の2つのディインターリーブ処理が行える。ディインターリーブ遅延回路12には、端子14から選択信号が供給され、この選択信号により2つのディインターリーブ処理が切換え可能とされている。

なお、記録時に、光ディスク1の一部、例えばディスク最内周のTOC (Table of Contents)に第1及び第2のインターリーブ処理のうちどちらのインターリーブ処理を行ったかのデータを書き込むようにし、再生時には、このデータに応じてディインターリーブ遅延回路12を切り換えるようにしても良い。

また、光ディスク1に第1のインターリーブ処理の信号と第2のインターリーブ処理の信号とを混在させて記録することも可能である。すなわち、例えば光ディスク1として、予め所定のデータが記録されるデータ領域と、ユーザーが自在に書き込みできるユーザー領域とを設けておき、予め所定のデータが記録される領域には第1のインターリーブ処理で信号を記録し、ユーザー領域には第2のインターリーブ処理で信号を記録するようにしても良い。この場合、データ領域は、CD−ROMと同様にスタンパを用いて大量複製することができる。

復号化回路10の出力が出力端子15から取り出され、出力端子15の出力から再生データが得られる。

b. フレーム構造及びセクタ構造

光ディスク1には、第2図Aに示すように、フレーム構造にデータが展開され、EFM変調されてデータが記録される。このフレーム構造は、音楽用のコンパクトディスクと同様である。すなわち、1フレームは、第2図Aに示すように、オーディオデータを16ビットでサンプリングした場合にL(左)、R(右)各6サンプル分に相当する24シンボル(1シンボルは8ビット、EFM変調されて14チャンネルビット)のデータビットと、8シンボルのパリティと、1シンボルのサブコードと、図示していない24チャンネルビットのフレームシンクと、直流分抑圧用のマージンビットからなる。したがって、1フレームの総チャンネルビット数は、

フレームシンク	24チャンネルビット
データビット	14×24=336チャンネルビット
サブコード	14チャンネルビット
パリティ	14×8=112チャンネルビット
マージンビット	3×34=102チャンネルビット
合　計	588チャンネルビット

となる。

各フレームの1シンボルのサブコードは、P〜Wの8チャンネルある。第2図Bに示すように、各フレームのP〜Wの8チャンネルのサブコードを98集めて、1サブコードブロックが構成される。このサブコードブロックが1セクタとされる。したがって、1セクタは、98フレームに相当する。サブコードフレームシンクS_0、S_1としては、データをEFM変調したとき256のパターンにない2つのパターンが選ばれている。

これらP〜Wまでのサブコードのうち、Pチャンネルは先頭を示すフラグとされる。Qチャンネルは、コントロールビットとされる。すなわち、Qチャンネルには、データ/オーディオフラグ、アドレス、トラックナンバー、タイムコード等が記録される。

c. インターリーブ処理について

前述したように、この発明の一実施例では、C_1系列とC_2系列とで2重にエラー訂正符号が付加されてデータが記録される。そして、このような符号化を行う際に、第1及び第2の2つのインターリーブ処理が設定できる。

第1のインターリーブ処理では、最大108フレームのインターリーブが施される。そして、この場合には、連続するフレームとともに畳込み符号化がなされる。この第1のインターリーブ処理は、音楽用コンパクトディスクやCD－ROMと全く同様な処理できる。したがって、音楽データのようなシーケンシャルなデータを扱う場合や、音楽用コンパクトディスクやCD－ROMとのコンパティビリティを完全に保ちたい場合に好適である。

第2のインターリーブ処理では、最大95フレームのインターリーブが施される。そして、モジュロ98の処理により98フレームで回りながらインターリーブが施される。これにより、エラー訂正符号化が98フレームからなる1セクタ内で完結される。この第2のインターリーブ処理は、1セクタ内でエラー訂正符号化が完結するため、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりする必要性が多い場合に用いて好適である。

なお、第1のインターリーブ処理の場合でも、第2のインターリーブ処理の場合でも、基本的なエンコードプロセス及びデコードプロセスは同様である。すなわち、データを2次元配列し、(28，24，5)リードソロモン符号によりパリティQが付加され、C_2系列の符号化がなされる。そして、インターリーブが施され、(32，28，5)リードソロモン符号によりパリティPが付加され、C_1系列の符号化がなされる。

第1のインターリーブ処理の場合には、最大108フレームのインターリーブが施されて畳込み符号化がなされる。このため、任意のセクタにデータを記録したり、任意のセクタのデータを書き換えたりする場合に、処理が非常に複雑になる。

すなわち、第1のインターリーブ処理を行った場合には、第3図に概念図で示すように、最大108フレームのインターリーブが行われ、連続するフレームのデータとともに畳込み符号化がなされる。例えば第3図においてセクタmのデータは、最大108フレームのインターリーブにより、第3図において斜線で示すような配置となる。この斜線で示す領域のセクタmのデータが書き換えられると、セクタ(m＋1)の全てのC_1系列とセクタ(m＋2)の一部のC_1系列にその影響が生じるとともに、セクタ(m－1)の全てのC_1系列とセクタ(m－2)一部のC_1系列にその影響が生じる。したがって、セクタmのデータを書き換える場合には、セクタ(m＋1)，(m＋2)，(m－1)，(m－2)のC_1系列のパリティをそれに応じて求め直す必要がある。

これに対して、第2のインターリーブ処理は、インターリーブ長が1セクタのフレーム数以下の95フレームとされ、モジュロ98で回りながらインターリーブがかけられている。このため、第4図に概念図で示すように、エラー訂正符号化が1セクタで完結する。したがって、例えばセクタnの書き換えを行っても、セクタ(n＋1)，(n＋2)，(n－1)，(n－2)には何ら影響を及ぼさない。

d．第1のインターリーブ処理の場合のエンコード及びデコードプロセス

第1のインターリーブ処理でデータを符号化する場合のエンコードプロセスについて、第5図を参照しながら説明する。

記録すべき16ビットの12個のデータL_{6n}，R_{6n}，L_{6n+1}，R_{6n+1}，…，L_{6n+5}，R_{6n+5}は、上位8ビット、下位8ビットのデータW_{12n}，A，W_{12n}，B，…，W_{12n+11}，A，W_{12n+11}，Bに分けられて遅延ブロック21に送られる。上位8ビットがA、下位8ビットがBで示されている。

遅延ブロック21で偶数番のデータL_{6n}，R_{6n}，L_{6n+2}，R_{6n+2}……に対して、遅延素子D1～D12により、2フレーム分の遅延がかけられる。これとともに、遅延ブロック21でデータの並べ換えがなされる。

そして、遅延ブロック21から出力される24シンボルがC_2デコーダ22に送られる。C_2デコーダ22で、(28，24，5)リードソロモン符号により、4シンボルのパリティQ_{12n}，Q_{12n+1}，…，Q_{12n+3}が生成される。

遅延ブロック２１の２４シンボルの出力データの中央に、C_2デコーダ２２で生成された４シンボルのパリティQ_{12n}，Q_{12n+1}，…，Q_{12n+3}が付加され、２８シンボルとされる。

この２８シンボルが遅延ブロック２３に送られる。遅延ブロック２３の遅延素子Ｄ２１～Ｄ４７により、この２８シンボルのそれぞれに対して４の倍数フレームのインターリーブが施される。

遅延ブロック２３から出力される２８シンボルがC_1エンコーダ２４に送られる。C_1エンコーダ２４で（３２，２８，５）リードソロモン符号により４シンボルのパリティP_{12n}，P_{12n+1}，…，P_{12n+3}が生成される。

遅延ブロック２３から出力される２８シンボルの最後に、C_1エンコーダ２４で生成された４シンボルのパリティP_{12n}，P_{12n+1}，…，P_{12n+3}が付加され、３２シンボルとされる。

この３２シンボルが遅延ブロック２５に送られる。遅延ブロック２５の遅延素子Ｄ５１～Ｄ６６により、この３２シンボルが１シンボル毎に１フレーム遅延される。

そして、インバータ１１～１４及び１５～１８により、パリティシンボルが反転され、エンコードプロセスが完了される。

デコードプロセスは、上述のエンコードプロセスと逆の処理となる。デコードプロセスについて、第６図を参照しながら説明する。

再生された３２シンボル（データ２４シンボルにパリティＰが４シンボル、パリティＱが４シンボルが付加されている）が遅延ブロック３１に送られる。遅延ブロック３１の遅延素子Ｄ７１～Ｄ８６により、１シンボル毎のシンボルが１フレーム遅延される。そして、インバータ１１１～１１８によりパリティシンボルが反転される。この３２シンボルがC_1デコーダ３２に送られる。

C_1デコーダ３２から出力される２８シンボルが遅延ブロック３３に送られる。遅延ブロック３３の遅延素子Ｄ９１～Ｄ１１６により４の倍数フレームずつのインターリーブが解かれる。そして、遅延ブロック３３の出力がC_2デコーダ３４に送られる。

C_1デコーダ３２及びC_2デコーダ３４でエラー訂正処理がなされる。C_2デコーダ３４から出力される２４シンボルは、遅延ブロック３５に送られる。遅延ブロック３５でデータが時系列順に戻される。そして遅延素子Ｄ１２１～Ｄ１３２により、奇数番のデータが２フレーム遅延され、デコードプロセスが完了される。

第１のインターリーブ処理で符号化した場合の各シンボルを２次元配列上のマップで示すと、第７図に示すようになる。第７図に示すように、第１のインターリーブ処理の場合、最大１０６フレームのインターリーブが施される。したがって、時系列順のシンボルに対応する１セクタ分のシンボルの座標は下表（表１）のようになる。なお、第８図に示すように、$M_{(i,j)}$はシンボルの配置される行番号ｉと列番号ｊを示している。

表　1

シンボル		座標
L_{6n}	W_{12n},A ――――→	$M_{(0,n)}$
	W_{12n},B ――――→	$M_{(1,n+3)}$
R_{6n}	W_{12n+1},A ――――→	$M_{(6,n+24)}$
	W_{12n+1},B ――――→	$M_{(7,n+27)}$
L_{6n+1}	W_{12n+2},A ――――→	$M_{(16,n+62)}$
	W_{12n+2},B ――――→	$M_{(17,n+65)}$
R_{6n+1}	W_{12n+3},A ――――→	$M_{(22,n+86)}$
	W_{12n+3},B ――――→	$M_{(23,n+89)}$
L_{6n+2}	W_{12n+4},A ――――→	$M_{(2,n+8)}$
	W_{12n+4},B ――――→	$M_{(3,n+11)}$
R_{6n+2}	W_{12n+5},A ――――→	$M_{(8,n+32)}$
	W_{12n+5},B ――――→	$M_{(9,n+35)}$
L_{6n+3}	W_{12n+6},A ――――→	$M_{(18,n+70)}$
	W_{12n+6},B ――――→	$M_{(19,n+73)}$
R_{6n+3}	W_{12n+7},A ――――→	$M_{(24,n+94)}$
	W_{12n+7},B ――――→	$M_{(25,n+97)}$
L_{6n+4}	W_{12n+8},A ――――→	$M_{(4,n+16)}$
	W_{12n+8},B ――――→	$M_{(5,n+19)}$
R_{6n+4}	W_{12n+9},A ――――→	$M_{(10,n+40)}$
	W_{12n+9},B ――――→	$M_{(11,n+43)}$
L_{6n+5}	W_{12n+10},A ――――→	$M_{(20,n+78)}$
	W_{12n+10},B ――――→	$M_{(21,n+81)}$

$$R_{6n+5}\begin{cases} W_{12n+11}A \dashrightarrow M_{(26,\, n+108)} \\ W_{12n+11}B \dashrightarrow M_{(27,\, n+105)} \end{cases}$$

$$\bar{Q}_{12n} \dashrightarrow M_{(12,\, n+64)}$$
$$\bar{Q}_{12n+1} \dashrightarrow M_{(13,\, n+51)}$$
$$\bar{Q}_{12n+2} \dashrightarrow M_{(14,\, n+56)}$$
$$\bar{Q}_{12n+3} \dashrightarrow M_{(15,\, n+59)}$$
$$\bar{P}_{12n} \dashrightarrow M_{(28,\, n)}$$
$$\bar{P}_{12n+1} \dashrightarrow M_{(29,\, n-1)}$$
$$\bar{P}_{12n+2} \dashrightarrow M_{(30,\, n)}$$
$$\bar{P}_{12n+3} \dashrightarrow M_{(31,\, n-1)}$$

各シンボルが第7図に示すように2次元配列されている場合、第6図に示すデコードプロセスに対応したアドレス操作を行って列方向に読み出し／書き込みを行っていくことにより、インターリーブが解除され、データが復号できる。

すなわち、遅延ブロック31に対応して、偶数番目の行のシンボルを1フレーム遅延させて列方向にデータを読み出してライン81で示すようにC_1系列が復号できる。

その後、遅延ブロック33に対応して第1行に対して（27×4＝108）フレーム、第2行に対して（26×4＝104）フレーム、第3行に対して（25×4＝100）フレーム、…、それぞれ遅延させて読みだすことにより、ライン82で示すようにC_2系列が復号できる。

なお、遅延ブロック35に対応して奇数番のデータが2フレーム分遅延されるので、奇数番データは破線に四角で示すものが復号時に出力されることになる。

e．第2のインターリーブ処理の場合のエンコード及びデコードプロセス

第2のインターリーブ処理でデータを符号化する場合のエンコードプロセスについて、第9図を参照しながら説明する。

16ビットの12個のデータL_{6n}，R_{6n}，L_{6n+1}，R_{6n+1}，…，L_{6n+5}，R_{6n+5}は、上位8ビット、下位8ビットのデータW_{12n}. A，W_{12n}. B，…，W_{12n+11}．A，W_{12n+11}．Bに分けられて遅延ブロック41に送られる。

遅延ブロック41で偶数番のデータL_{6n}，R_{6n}，L_{6n+2}，R_{6n+2}，…に対して、遅延素子D151～D162により、2フレーム分の遅延がかけられる（遅延ブロック41はモジュロ98で回っている）。これとともに、遅延ブロック41でデータの並べ換えがなされる。

そして、遅延ブロック41から出力される24シンボルがC_2デコーダ42に送られる。C_2デコーダ42で、（28，24，5）リードソロモン符号により、4シンボルのパリティQ_{12n}, Q_{12n+1}, …，Q_{12n+3}が生成される。

遅延ブロック42の24シンボルの出力データの中央に、C_2デコーダ42で生成された4シンボルのパリティQ_{12n}, Q_{12n+1}, …，Q_{12n+3}が付加され、28シンボルとされる。

この28シンボルが遅延ブロック43に送られる。遅延ブロック43の遅延素子D171～D197により、4フレーム、3フレーム、4フレーム、3フレーム、…のインターリーブが施される（遅延ブロック43は、モジュロ98で回っている）。

遅延ブロック43から出力される28シンボルがC_1エンコーダ44に送られる。C_1エンコーダ44で（32，28，5）リードソロモン符号により4シンボルのパリティP_{12n}, P_{12n+1}, …，P_{12n+3}が生成される。

遅延ブロック43から出力される28シンボルの最後に、C_1エンコーダ44で生成された4シンボルのパリティP_{12n}, P_{12n+1}, …，P_{12n+3}が付加され、32シンボルとされる。

この32シンボルが遅延ブロック45に送られる。遅延ブロック45の遅延素子D201～D216により、この32シンボルが1シンボル毎に1フレーム遅延される（遅延ブロック45はモジュロ98で回っている）。

そして、インバータ121～124及び125～128により、パリティシンボルが反転され、エンコードプロセスが完了される。

デコードプロセスは、上述のエンコードプロセスと逆の処理となる。デコードプロセスについて、第10図を参照しながら説明する。

再生された32シンボルが遅延ブロック51に送られる。遅延ブロック51の遅延素子D221～D236により、1シンボル毎のシンボルが1フレーム遅延される（遅延ブロック51はモジュロ98で回っている）。そして、インバータI131～I138により再生されたパリティシンボルが反転される。この32シンボルがC_1デコーダ52に送られる。

C_1デコーダ52から出力される28シンボルが遅延ブロック53に送られる。遅延ブロック53の遅延素子D241～D267により4フレーム、3フレーム、4フレーム、3フレーム、…、のインターリーブが解かれる（遅延ブロック53はモジュロ98で回っている）。そして、遅延ブロック53の出力がC_2デコーダ54に送られる。

C_1デコーダ52及びC_2デコーダ54でエラー訂正処理がなされる。C_2デコーダ54から出力される24シンボルは、遅延ブロック55に送られる。遅延ブロック55でデータが時系列順に戻される。そして遅延素子D271～D282により、偶数番目のデータが2フレーム遅延され、デコードプロセスが完了される（遅延ブロック55はモジュロ98で回っている）。

第2のインターリーブ処理で符号化した場合の各シンボルを2次元配列上のマップで示すと、第11図に示すようになる。第11図に示すように、第2のインターリーブ処理の場合、最大95フレームのインターリーブが施される。そして、モジュロ98をとることにより、97列目のフレームまで遅延されると0フレームに戻る。したがって、時系列順の1セクタ分のシンボルW_{12n}.A, W_{12}, B, W_{12n+1}.A, W_{12n+1}.B…に対応する1セクタ分のシンボルの座標$M_{(i,j)}$は下表（表2）のようになる。$M_{(i,j)}$は第12図に示すように各シンボルの座標を示し、iは行番号、jは列番号、n＝0～97である。

表 2

	シンボル	座標
L_{6n}	W_{12n}.A	→ $M_{(0,n)}$
	W_{12n}.B	→ $M_{(1,n+3)}$
R_{6n}	W_{12n+1}.A	→ $M_{(6,n+21)}$
	W_{12n+1}.B	→ $M_{(7,n+24)}$
L_{6n+1}	W_{12n+2}.A	→ $M_{(16,n+54)}$
	W_{12n+2}.B	→ $M_{(17,n+57)}$
R_{6n+1}	W_{12n+3}.A	→ $M_{(22,n+75)}$
	W_{12n+3}.B	→ $M_{(23,n+78)}$
L_{6n+2}	W_{12n+4}.A	→ $M_{(2,n+7)}$
	W_{12n+4}.B	→ $M_{(3,n+10)}$
R_{6n+2}	W_{12n+5}.A	→ $M_{(8,n+28)}$
	W_{12n+5}.B	→ $M_{(9,n+31)}$
L_{6n+3}	W_{12n+6}.A	→ $M_{(18,n+61)}$
	W_{12n+6}.B	→ $M_{(19,n+64)}$
R_{6n+3}	W_{12n+7}.A	→ $M_{(24,n+82)}$
	W_{12n+7}.B	→ $M_{(25,n+85)}$
L_{6n+4}	W_{12n+8}.A	→ $M_{(4,n+14)}$
	W_{12n+8}.B	→ $M_{(5,n+17)}$
R_{6n+4}	W_{12n+9}.A	→ $M_{(10,n+35)}$
	W_{12n+9}.B	→ $M_{(11,n+38)}$
L_{6n+5}	W_{12n+10}.A	→ $M_{(20,n+68)}$
	W_{12n+10}.B	→ $M_{(21,n+71)}$
R_{6n+5}	W_{12n+11}.A	→ $M_{(26,n+89)}$
	W_{12n+11}.B	→ $M_{(27,n+92)}$
	$\overline{Q}_{12n}$	→ $M_{(12,n+42)}$
	$\overline{Q}_{12n+1}$	→ $M_{(13,n+45)}$
	$\overline{Q}_{12n+2}$	→ $M_{(14,n+49)}$
	$\overline{Q}_{12n+3}$	→ $M_{(15,n+52)}$
	$\overline{P}_{12n}$	→ $M_{(28,n)}$
	$\overline{P}_{12n+1}$	→ $M_{(29,n+97)}$
	$\overline{P}_{12n+2}$	→ $M_{(30,n)}$
	$\overline{P}_{12n+3}$	→ $M_{(31,n+97)}$

各シンボルが第11図に示すように2次元配列されている場合、第10図に示すデコードプロセスに対応したアドレス操作を行って列方向に読み出し／書き込みを行っていくことにより、インターリーブが解除され、データが復号できる。

すなわち、遅延ブロック51に対応して、偶数番目の行のシンボルを1フレーム遅延させて列方向にデータを読み出してC_1系列が復号できる。この際、モジュロ98がとられる。

その後、遅延ブロック53に対応して第1行に

対して95フレーム、第2行に対して91フレーム、第3行に対して88フレーム、…、それぞれ遅延させて読みだすことにより、C_2系列が復号できる。なお、この際にもモジュロ98がとられる。

第11図では、n＝0のときのデコード処理と、n＝60の場合のデコード処理が示されている。n＝0のときには、ライン91で示すようにC_1系列が復号される。この際、モジュロ98がとられるので、0列のフレームと97列のフレームのシンボルからC_1系列が復号される。またn＝0の場合には、ライン92で示すようにC_2系列が復号される。

n＝60の場合には、ライン93で示すようにC_1系列が復号される。また、n＝60の場合には、ライン94で示すようにC_2系列が復号される。この際、モジュロ98がとられるので、60フレームから38フレーム分遅延されるべき$W_{72,0}$,Bのシンボルは、$M_{(11,0)}$の位置に戻る。以下のシンボルは同様である。

f．一実施例におけるエンコード・デコード処理

上述のエンコードプロセス及びデコードプロセスは、データをRAMに蓄えておき、アドレスを制御することにより実現されている。

すなわち、第1図における符号化回路3は、上述の第1のインターリーブ処理の場合のエンコードプロセス及び第2のインターリーブ処理の場合のエンコードプロセスとが選択可能とされていて、この符号化回路3は、第13図に示すように、RAM61と、エンコーダ62と、第1のインターリーブ処理に対応してアドレスを発生するアドレス発生回路63と、第2のインターリーブ処理に対応してアドレスを発生するアドレス発生回路64とからなる。そして、アドレス発生回路63から出力される第1のインターリーブ処理に対応するアドレスとアドレス発生回路64から出力される第2のインターリーブ処理に対応するアドレスとを、スイッチ手段65を介して選択的にRAM61に供給することにより、第1のインターリーブ処理の場合のエンコードプロセスと第2のインターリーブ処理の場合のエンコードプロセスとの両者が設定可能とされる。

また、第1図における復号化回路10は、上述の第1のインターリーブ処理の場合のデコードプロセス及び第2のインターリーブ処理の場合のデコードプロセスとが選択可能とされていて、この復号化回路10は、第14図に示すように、RAM71と、エンコーダ72と、第1のインターリーブ処理に対応してアドレスを発生するアドレス発生回路73と、第2のインターリーブ処理に対応してアドレスを発生するアドレス発生回路74とからなる。そして、アドレス発生回路73から出力される第1のインターリーブ処理に対応するアドレスと、アドレス発生回路74から出力される第2のインターリーブ処理に対応するアドレスとを、スイッチ手段75を介して選択的にRAM71に供給することにより、第1のインターリーブ処理の場合のデコードプロセスと第2のインターリーブ処理の場合のデコードプロセスとの両者が設定可能とされる。

〔発明の効果〕

この発明によれば、エラー訂正符号化が98フレームからなる1セクタで完結できるインターリーブ処理が設定できる。エラー訂正符号化が1セクタ内で完結できるため、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換えたりした際、その影響が他のセクタに及ばず、任意のセクタにデータを書き込んだり、任意のセクタのデータを書き換える際に複雑な信号処理が必要とならないとともに、データの記録容量が低下しない。

更に、エラー訂正符号化が1セクタ内で完結しているので、データの読み出し／書き込みの際に複数のセクタのデータを取り込む必要がなく、アクセス時間が短縮できる。

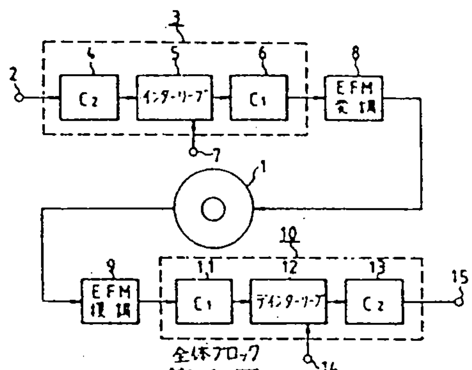
4.図面の簡単な説明

第1図はこの発明の一実施例の全体構成を示すブロック図、第2図はこの発明の一実施例の記録フォーマットの説明に用いる略線図、第3図及び

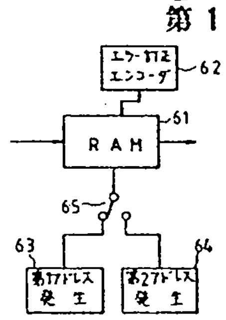
第4図はこの発明の一実施例の説明に用いる略線図、第5図及び第6図は第1のインターリーブ処理の説明に用いるブロック図、第7図及び第8図は第1のインターリーブ処理の説明に用いる略線図、第9図及び第10図は第2のインターリーブ処理の説明に用いるブロック図、第11図及び第12図は第2のインターリーブ処理の説明に用いる略線図、第13図は符号化装置の一例のブロック図、第14図は符号化装置の一例のブロック図である。

図面における主要な符号の説明

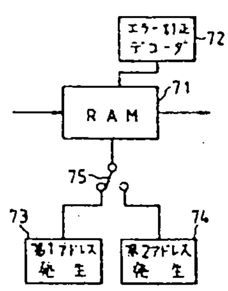
1：光ディスク、　2：データ入力端子、
3：符号化回路、　5：インターリーブ遅延回路、
8：ＥＦＭ変調回路、　9：ＥＦＭ復調回路、
10：復号化回路、
12：ディインターリーブ回路、
15：出力端子、　61，71：ＲＡＭ、
63，64，73，74：アドレス発生回路。



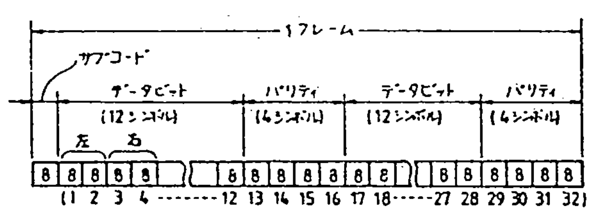
第1図



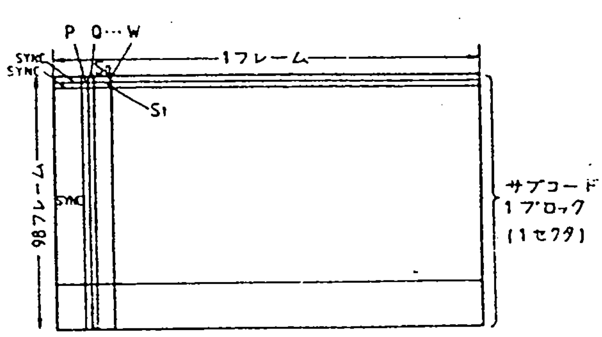
符号化装置
第13図



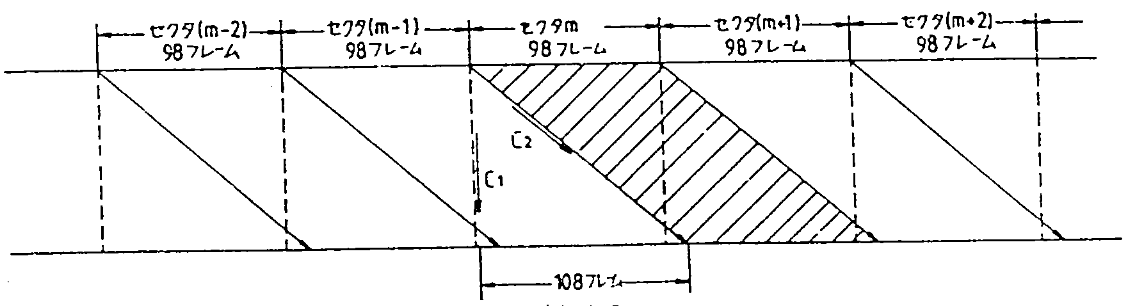
復号装置
第14図



フレーム構造
第2図A

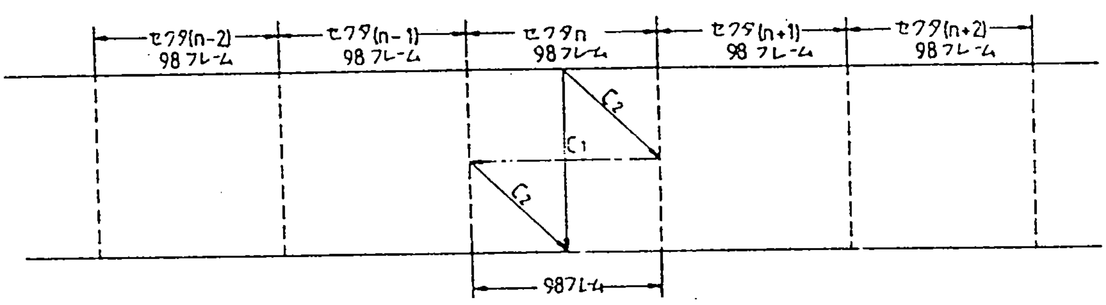


セクタ構造
第2図B



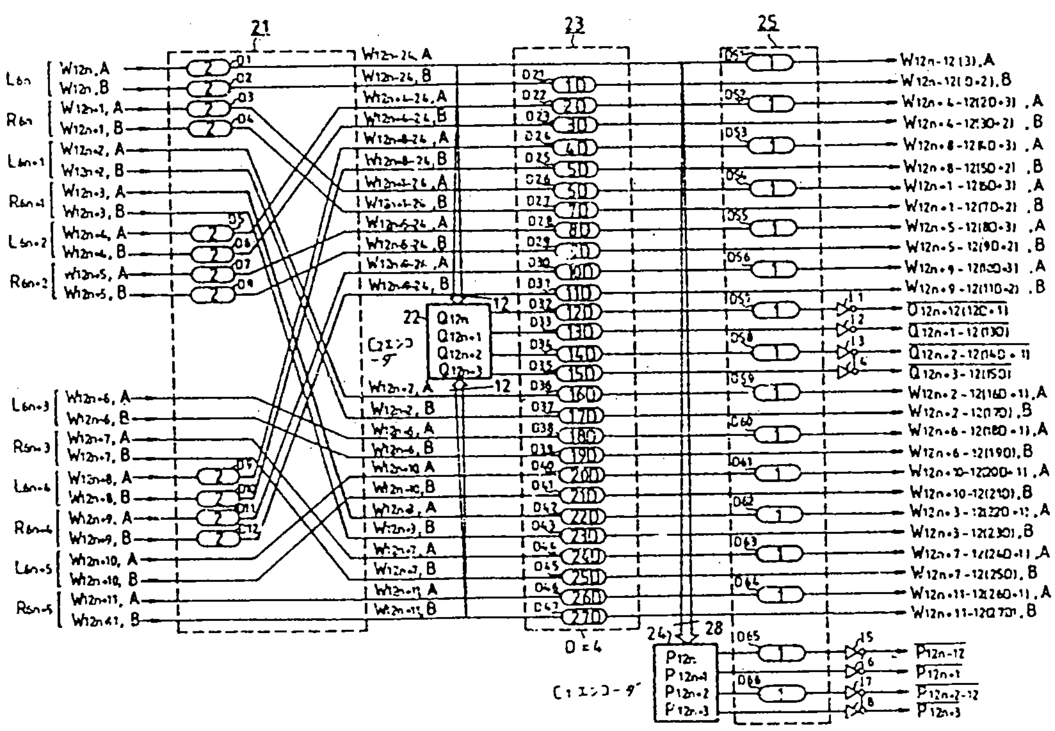
説明図

第3図



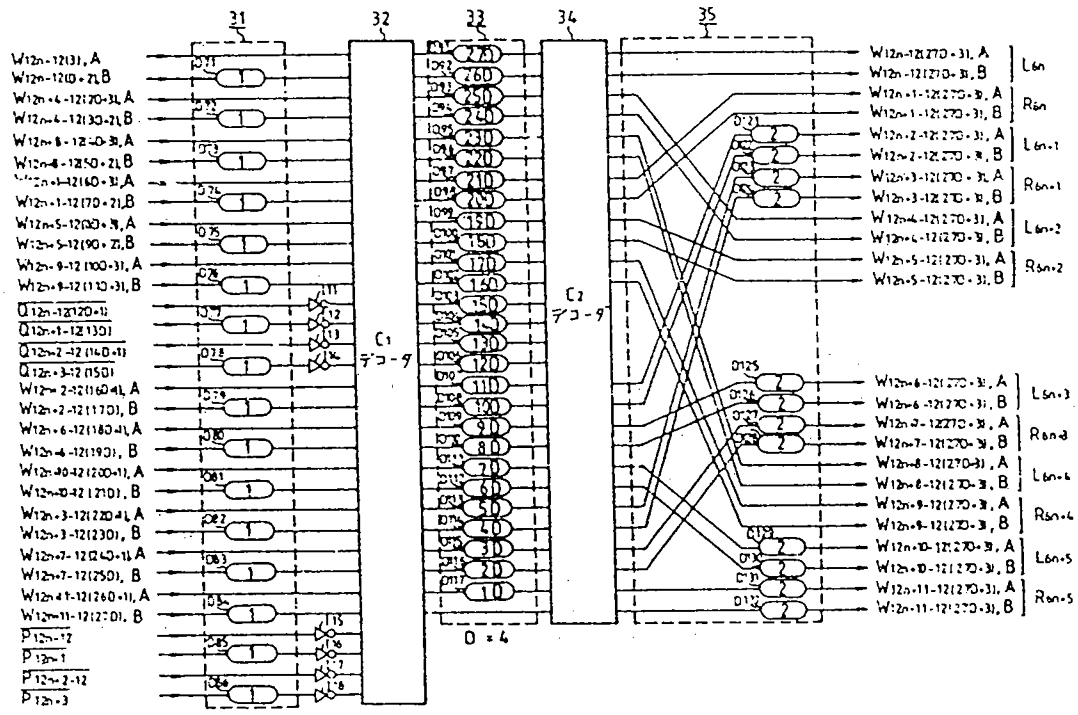
説明図

第4図



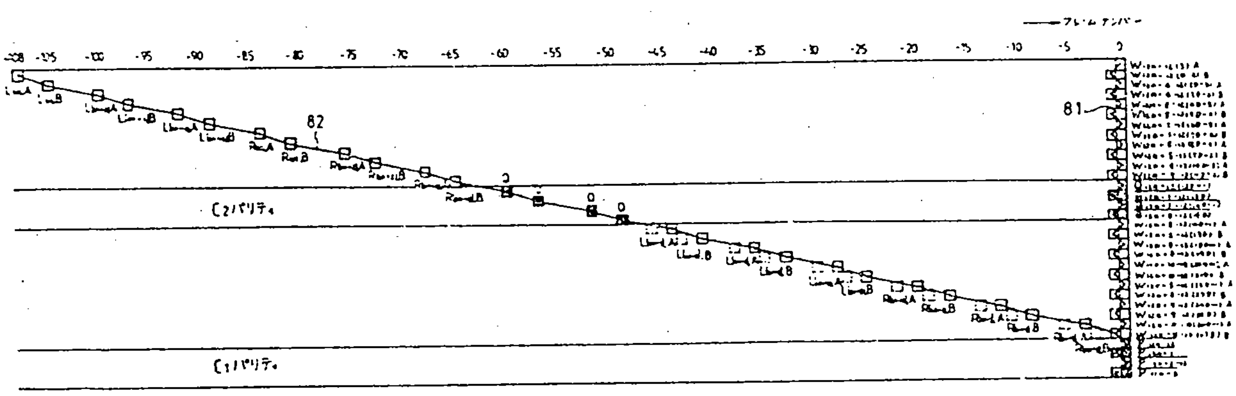
第1のインターリーブ処理の場合のエンコードプロセス

第5図



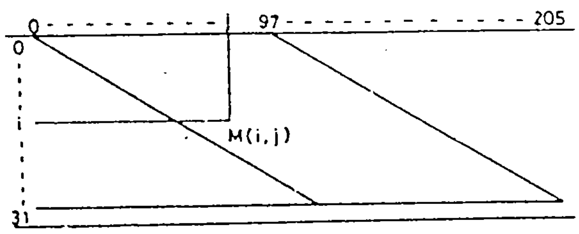
第1のインターリーブ処理の場合のデコードプロセス

第 6 図



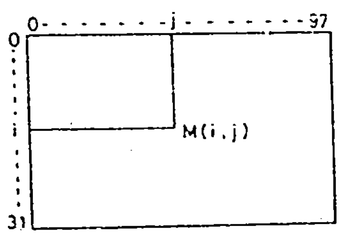
第1のエンコード処理の場合のマップ

第 7 図



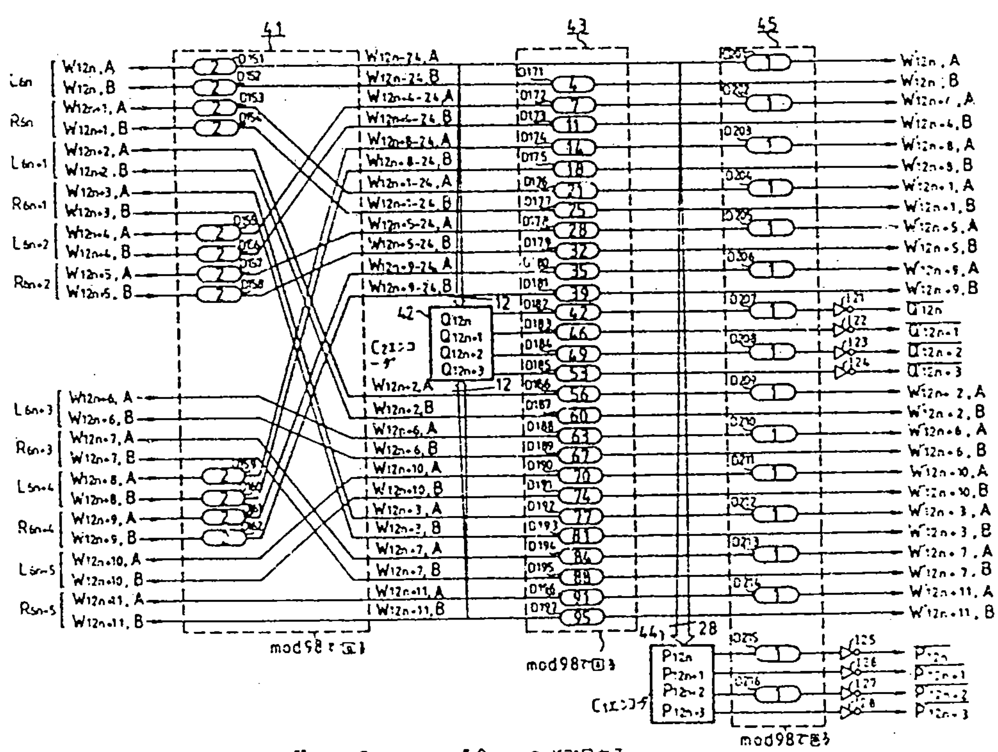
説明図

第8図



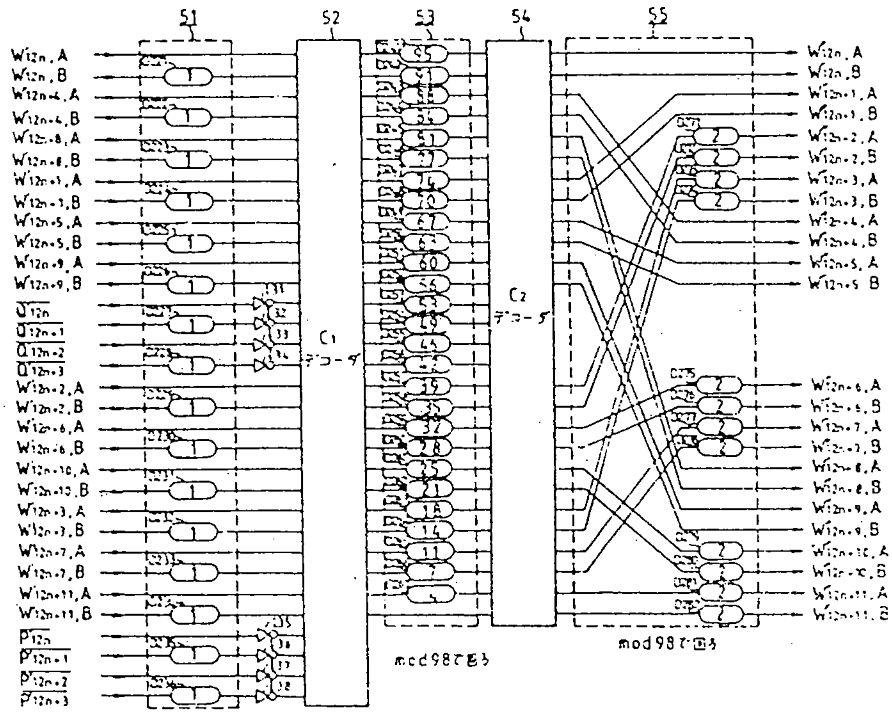
説明図

第12図



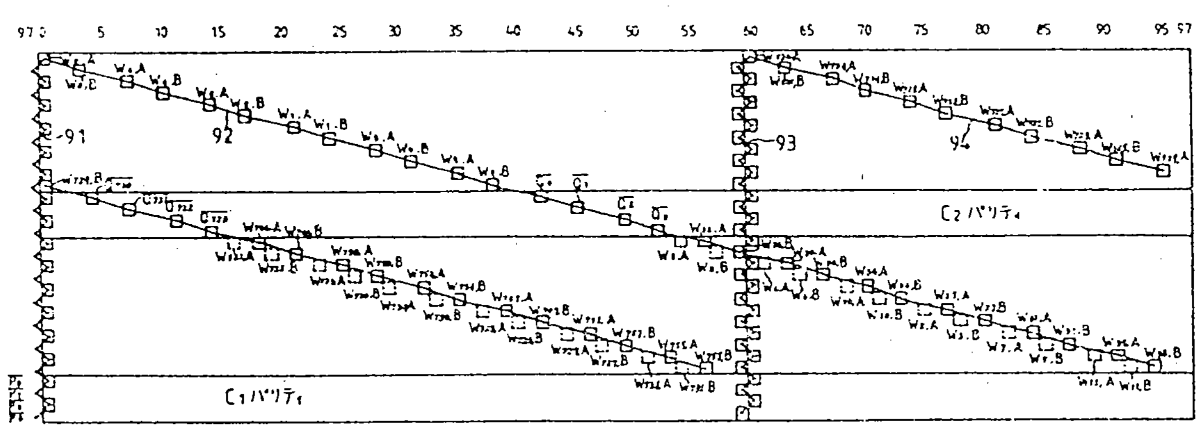
第2のインターリーブ処理の場合のエンコードプロセス

第9図



第2のインターリーブ処理の場合のデコードプロセス

第10図



第2のインターリーブ処理の場合のマップ

第11図

手続補正書

昭和63年　9月10日

特許庁長官　吉　田　文　毅　殿

1. 事件の表示

昭和63年特許願第118567号

2. 発明の名称

ディジタルデータ記録／再生装置

3. 補正をする者

事件との関係　　特許出願人

住所　東京都品川区北品川6丁目7番35号

名称　(218) ソ　ニ　ー　株　式　会　社

代表取締役　大　賀　典　雄

4. 代理人　〒170

住所　東京都豊島区東池袋1丁目48番10号

25山京ビル 420号　℡.(03)980-0339

氏名　(8276) 弁理士　杉　浦　正　知

5. 補正命令の日付　　自　　発

6. 補正の対象

明細書の発明の詳細な説明の欄

63. 9. 12

7. 補正の内容

(1)明細書中、第6頁第12行目に「なされる。」とあるを「できると有用である。」と補正する。

(2)同、第15頁第3行目及び第4行目、第15頁第5行目、第15頁第8行目、第15頁第9行目、第15頁第11行目、第15頁第18行目に、それぞれ「ディインターリーブ」とあるを「デインターリーブ」と補正する。

(3)同、第21頁第5行目及び第6行目に「全ての C_1 系列とセクタ (m−2) 一部の」とあるを削除する。

⑲日本国特許庁（JP）　⑪特許出願公開
⑫公開特許公報（A）　平2−121584
⑤Int. Cl.⁵　識別記号　庁内整理番号　⑬公開　平成2年(1990)5月9日
H 04 N 5/92　H　7734−5C
審査請求　未請求　請求項の数 1　（全4頁）

⑭発明の名称　映像信号記録装置

㉑特　願　昭63−275254
㉒出　願　昭63(1988)10月31日

⑫発明者　星　秀典　東京都大田区下丸子3丁目30番2号　キヤノン株式会社内
⑦出願人　キヤノン株式会社　東京都大田区下丸子3丁目30番2号
⑭代理人　弁理士　田中　常雄

明細書

1．発明の名称

映像信号記録装置

2．特許請求の範囲

映像信号を記録媒体にディジタル記録する映像信号記録装置であって、フレーム毎に異なるインターリーブ規則の下でフレーム内インターリーブを行う第1のインターリーブ手段と、当該第1のインターリーブ手段によるインターリーブの後に、フレーム間で映像データをインターリーブする第2のインターリーブ手段とを設けたことを特徴とする映像信号記録装置。

3．発明の詳細な説明

〔産業上の利用分野〕

本発明は、映像信号を記録媒体にディジタル記録する映像信号記録装置に関する。

〔従来の技術〕

映像信号を記録媒体にディジタル記録する従来の装置では、記録媒体へ記録する前に、エラー訂正のために、インターリーブなどの処理を施していたが、正しいデータの再現確率がきわて低い場合、例えば特殊再生の場合の画質、再生画像を考慮したものではない。即ち、従来のインターリーブ処理は、フレーム間でインターリーブ規則が変化しない、固定されたものである。

〔発明が解決しようとする課題〕

従来の映像信号記録装置でビデオ・テープに記録された映像信号を特殊再生する場合、ヘッドが走査しない部分が発生し、また、走査する部分でもヘッドのアジマスとトラックのアジマスとの相違から再生不可能な部分が発生する。この時、各フレーム間でインターリーブ規則が同じであれば、再生速度によっては、再生可能なライン・ブロックが各再生フレームで同一になってしまうという欠点がある。特に、所謂高速サーチを行う場合には、テープの走行速度を記録時の整数倍とすることが多いこと、再生されるデータ数が少なくなることもあって画面上で狭い領域の特性部分のみが繰り返し再生されることになり非常に見苦しい画面となってしまう。

そこで、本発明はこのような欠点の無い映像信号記録装置を提示することを目的とする。

〔課題を解決するための手段〕

本発明に係る映像信号記録装置は、映像信号を記録媒体にディジタル記録する映像信号記録装置であって、フレーム毎に異なるインターリーブ規則の下でフレーム内インターリーブを行う第1のインターリーブ手段と、当該第1のインターリーブ手段によるインターリーブの後に、フレーム間で映像データをインターリーブする第2のインターリーブ手段とを設けたことを特徴とする。

〔作用〕

上記手段により、正しいデータの再現確率が低い場合にも長期間再生不可能になる画面部分が発生することは無くなり、また、補間処理の際に時間的な相違が少なくなるので、自然な再生画像を得ることができる。

〔実施例〕

以下、図面を参照して本発明の実施例を説明する。

第1図は本発明の一実施例の構成ブロック図を示す。10はアナログ映像信号をディジタル化するA／D変換器、12はフレーム内インターリーブを行うインターリーブ回路、14は少なくとも1画面分のディジタル映像信号を記憶可能であり、インターリーブ回路12による作業用のメモリ、16はフレーム間、即ち時間軸方向でのインターリーブを行うインターリーブ回路、18はフレーム・メモリ18A及びフレーム・メモリ18Bを具備し、インターリーブ回路16による作業用のメモリ、20は誤り訂正符号（ECC）を付加するECC付加回路、22は同期コードSync及びID信号を付加する回路、24は変調回路、26は磁気ヘッド、28はビデオ・テープである。

ただし、メモリ14，18は本実施例の理解を容易にするために、区分して図示したのであり、実際の回路では、1つのメモリ回路を共用してもよいことは明らかである。

先ず、インターリーブ回路12によるインターリーブ処理を説明する。第2図は原画の1画面を示す。なお、本実施例では、この1画面を中央の縦線A－Bで2等分し、インターリーブを行うものとする。また、特殊再生時における最小再生単位としてのブロックを8画素×16画素とし、1画面の有効画素数とこの最小単位との関係から、1画面は、水平方向に88ブロック、垂直方向に132ブロックに分割されるとする。そして、第2図の垂直方向132分割の各々をライン・ブロックと呼ぶことにする。

このような1画面の画像情報をインターリーブし、第3図に示すように、(12n+1) 番目(n=0～10の整数) のライン・ブロックを抽出し、そのまとまりを2分割した左側のトラックを第1トラックとし、右側のトラックを第2トラックとする。同様に、(12n+2), (12n+3), ……, (12n+12) 番目のライン・ブロックを抽出し、同様に左側、右側でトラックを構成し、1フレーム（画面）当たり合計で24個のトラックを形成する。奇数番目のトラックは第2図のA－B線の左側になり、偶数番目のトラックは右側になる。

インターリーブ回路12はまた、上記のようなインターリーブ規則をフレーム毎に変化させる。第4図はフレーム間でインターリーブ規則が異なる様子を示す。第4図で、四角枠で囲んだ記号n－mは、nがフレーム番号、mが1フレーム内の132個のライン・ブロックの番号を示す。第1フレームでは、第1トラックに第1ライン・ブロックから始めて12ライン・ブロック毎のインターリーブしていたが、第2フレームでは第2ライン・ブロックからインターリーブを行う。この結果、第1フレームでの第1トラックに相当する情報は、第2フレームでは第23トラックに記録される。インターリーブ回路12は、第3フレーム以降も同様に、第1ライン・ブロックのデータが順次異なるトラックに配分されるように、インターリーブ規則を変化させる。インターリーブ回路12による処理の後では、データの並びと画面上の画素配列との対応関係がフレーム毎に変化している。但し、第4図は規則変化の一例であり、他の変化方法も採用できる。

次にインターリーブ回路１６による処理を説明する。インターリーブ回路１２によりインターリーブ処理をされた画像データは、第１フレームがフレーム・メモリ１８Ａに格納され、第２フレームがフレーム・メモリ１８Ｂに格納される。インターリーブ回路１６は、ビデオ・テープ２８における記録順序が第５図のようになるように、フレーム・メモリ１８Ａ及び同１８Ｂの記憶情報を交互に読み出して、ＥＣＣ付加回路２０に印加する。即ち、ビデオ・テープ２８の第１フレームの箇所には、記録しようとするビデオ信号の第１フレームの信号が50%、第２フレームの信号が50%含まれ、ビデオ・テープ２８の第２フレームの箇所には、記録しようとするビデオ信号の第１フレームの信号の残り50%と、第２フレームの信号の残り50%が含まれる。この２つのフレームの記録が終了した後、フレーム・メモリ１８Ａには第３フレームの情報を格納し、フレーム・メモリ１８Ｂには第４フレームの情報を格納し、第１及び第２フレームの場合と同様にビデオ・テープ２８に記録する。以下同様に、奇数フレームをフレーム・メモリ１８Ａに格納し、偶数フレームをフレーム・メモリ１８Ｂに格納して、順次フレーム間でインターリーブをかける。

つまり、本実施例では、インターリーブ回路１２によりフレーム毎にインターリーブ規則の変化するフレーム内インターリーブを行い、インターリーブ回路１６により、フレーム間でインターリーブを行う。これにより、特殊再生時における再生画像の補間による時間的相違が微小になり、画質劣化の極めて少ない再生画像を得ることができる。また、特定の再生速度で永久に再生されない画面部分も生じない。

〔発明の効果〕

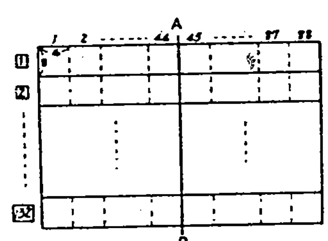
以上の説明から容易に理解できるように、本発明によれば、正しいデータの再現確率が低い場合にも画質劣化の極めて少ない再生画像を得ることができ、また、再生不能部分も生じない。

４．図面の簡単な説明

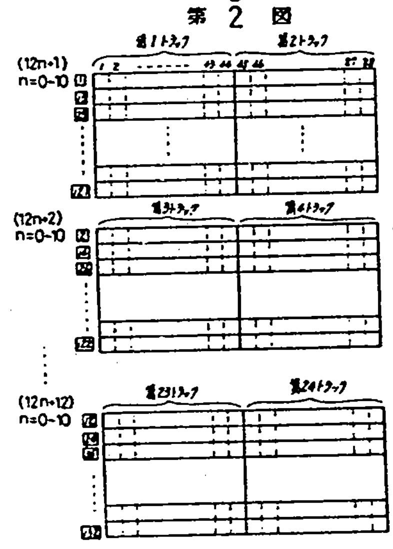
第１図は本発明の一実施例の構成ブロック図、第２図は画面分割の説明図、第３図はライン・ブロックと画素配列の説明図、第４図は第１図のインターリーブ回路１２によるインターリーブの説明図、第５図はインターリーブ回路１６によるインターリーブの説明図である。

１２，１６…インターリーブ回路　１８Ａ，１８Ｂ…フレーム・メモリ　２０…ＥＣＣ付加回路　２６…磁気ヘッド　２８…ビデオ・テープ

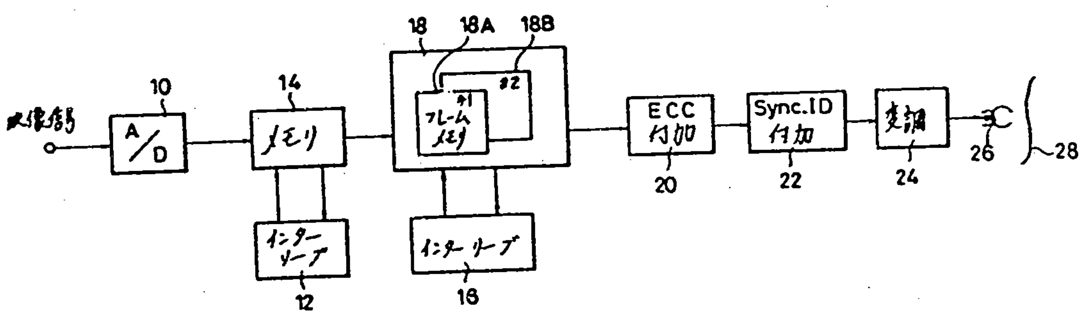
特許出願人　キヤノン株式会社
代理人弁理士　田中　常雄



第２図

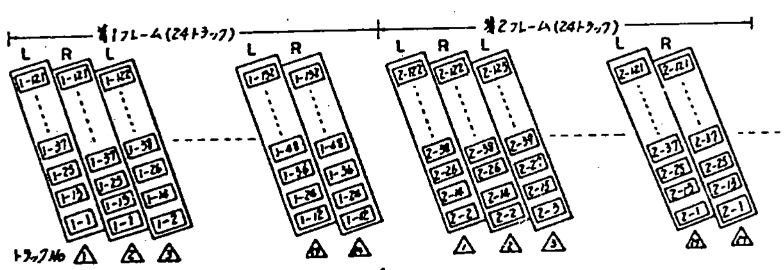


第３図



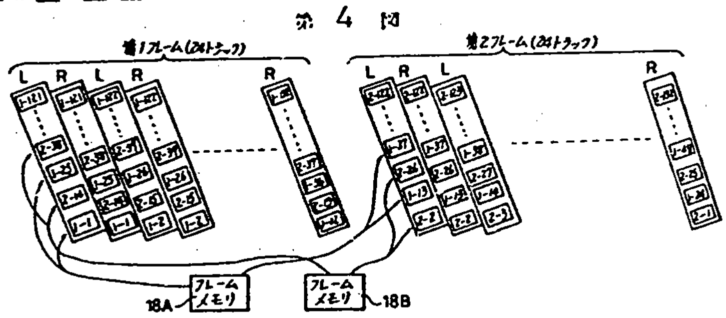
映像信号
10
A/D
14
メモリ
インターリーブ
12
18
18A
18B
#1
フレームメモリ
#2
インターリーブ
16
ECC付加
20
Sync.ID付加
22
変調
24
26
28

第 1 図



第1フレーム(24トラック)
第2フレーム(24トラック)
トラックNo

第 4 図



第1フレーム(24トラック)
第2フレーム(24トラック)
フレームメモリ
18A
フレームメモリ
18B

第 5 図

2 of 2

FORM PTO-892 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	Serial No.	GROUP ART UNIT	ATTACHMENT TO PAPER NUMBER	
NOTICE OF REFERENCES CITED	08/252,460	2317		2
	APPLICANT(S) Charles J. Kulas			

U.S. PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	NAME	CLASS	SUB-CLASS	FILING DATE IF APPROPRIATE
	A						
	B						
	C						
	D						
	E						
	F						
	G						
	H						
	I						
	J						
	K						

FOREIGN PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUB-CLASS	PERTINENT SHTS. DWG	PERTINENT PP. SPEC
	L								
	M								
	N								
	O								
	P								
	Q								

OTHER REFERENCES (Including Author, Title, Date, Pertinent Pages, Etc.)

	R	"A File System for Continuous Media", David P. Anderson et al., ACM Transactions on Computer Systems, Vol. 10, No. 4, Nov 1992, pp 311 - 337.
	S	"Designing File Systems for Digital Video and Audio", P. Venkat Rangan et al., Operating Systems Review, Vol. 25, No. 5, Oct 13 - 16, 1991, pp. 81 - 94.
	T	

EXAMINER	DATE	
Po Huang	09/27/95	Form 892BMR2107

* A copy of this reference is not being furnished with this office action.
See Manual of Patent Examining Procedure, section 707.05(a).

A File System for Continuous Media

(4)

DAVID P. ANDERSON
Sonic Solutions

YOSHITOMO OSAWA
Sony Corporation

and

RAMESH GOVINDAN
Bell Communications Research

p.311 - 337

The Continuous Media File System, CMFS, supports real-time storage and retrieval of continuous media data (digital audio and video) on disk. CMFS clients read or write files in "sessions," each with a guaranteed minimum data rate. Multiple sessions, perhaps with different rates, and non-real-time access can proceed concurrently. CMFS addresses several interrelated design issues: real-time semantics of sessions, disk layout, an acceptance test for new sessions, and disk scheduling policy. We use simulation to compare different design choices.

Categories and Subject Descriptors: D.4.3 [Operating Systems]: File Systems Management—*access methods; file organization*; D.4.7 [Operating Systems]: Organization and Design—*real-time systems*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Disk scheduling, multimedia

1. INTRODUCTION

Current disk drives have raw data rates of 5 to 10 million bits per second (Mbps) or more. Such rates suffice for many forms of digital audio and motion video (*continuous media*, or *CM*) data: Audio data rates are from 8.0 Kbps to

This work was performed at the University of California at Berkeley and was supported by NSF Infrastructure grant CDA-8722788, NSF PYI grant CCR-86-57529, the California MICRO program, and Sun Microsystems.
Authors' addresses: D. P. Anderson, Sonic Solutions, 1891 East Francisco Blvd., San Rafael, CA 94901; Y. Osawa, MO Business Development Division, Storage Systems Group, Sony Corporation, 2255 Okata, Atsugi, Kanagawa 243 Japan; R. Govindan, Bellcore, 445 South St., Morristown, NJ 07960.

1.4 Mbps, while compressed video ranges from one to several Mbps. However, when a disk is accessed via a general-purpose file system, the data rates seen by clients are generally lower and may vary unpredictably.

We have developed a *Continuous Media File System* (*CMFS*) whose clients read and write files in "sessions," each with a guaranteed minimum data rate. Multiple sessions, perhaps with different data rates, can coexist. CMFS can handle non-real-time traffic concurrently with these real-time sessions.

To provide data-rate guarantees, CMFS addresses the following interrelated issues:

—*Real-time semantics*. The CMFS client interface, described in Section 2, has flexible but well-defined real-time semantics.

—*Disk layout*. Section 4 gives the CMFS assumptions about disk layout.

—*Acceptance test*. Section 5 describes how CMFS determines if a new session can be accommodated.

—*Disk head scheduling*. Several alternative policies for ordering disk read and write operations are discussed in Sections 6 and 7.

Several broad classes of CM data servers can be envisioned: workstation file systems that handle voice mail messages as well as other data; network-accessible archives of data resources (lectures, hypermedia documents, etc.) for research and education; and, with the advent of B-ISDN networks, commercial information services offering movies, news, and music to hundreds or thousands of concurrent clients. High-level issues such as security, naming and indexing, and file structuring differ among these classes; we do not deal with these issues here. However, there is a common need to store and retrieve data streams with predictable real-time performance; thus, the concepts and techniques of CMFS apply to each class.

CMFS is meant to serve as part of a distributed system that handles integrated audio and video. End-to-end performance guarantees cannot, of course, be achieved by disk scheduling alone. CMFS conforms to the "meta-scheduling" model [2], which provides a mechanism for making such guarantees. The role of CMFS in this larger context is discussed in Section 3.

2. CLIENT INTERFACE

CMFS clients access real-time files in *sessions*. Each session has a FIFO buffer for data transfer between the client and CMFS. For concreteness, assume that CMFS is part of an OS kernel, a client is a kernel process, and FIFOs are circular buffers in physical memory. Alternatives will be discussed in Section 3.1.

2.1 The Semantics of Sessions

The flow of data in a session is not necessarily smooth or periodic. Instead, session semantics are defined in terms of a "logical clock" that runs at a fixed rate through the FIFO, stopping if it catches up to the client's position. CMFS promises to stay ahead of the logical clock by a given positive amount (the "cushion"). These semantics allow CMFS to handle variable-rate files and other nonuniform access in a simple way. Because CMFS is guided by client behavior, it need not know about data time stamps or file internals.

A session is created using the following:

```
ID = request_session (
   int direction,          /* READ or WRITE */
   File_ID name,
   int offset,
   FIFO* buffer,
   TIME cushion,
   int rate);

start_clock (ID);
```

If direction is READ, request_session ( ) requests a session in which the given file is read sequentially starting from the given offset (henceforth assumed to be zero). If the session cannot be accepted, an error code is returned. Otherwise, a session is established, and its ID is returned. Start_clock ( ) starts the session's logical clock; the client can remove up to cushion amount of data before calling this. The appropriate value for cushion depends on the client's maximum delay in handling data (see Section 3.2). The client is notified (via an RPC or exception) when the end of the file has been reached. CMFS provides a seek ( ) operation that flushes data currently in the FIFO and repositions the read or write point.

To describe the semantics of read sessions more formally, we use the following notation (see Figure 1a):

R: the rate argument to request_session ( );
$\overline{Y}$: the cushion argument to request_session ( );
t_{start}: the time when start_clock ( ) is called;
$P(t)$: the index of the next byte to be put into the FIFO by CMFS at time t;
$G(t)$: the index of the next byte to be removed fromthe FIFO by the client at time t;
$C(t)$: the value of the logical clock at time t; and
B: the size of the FIFO buffer, in bytes.

The logical clock $C(t)$ is zero for $t = t_{start}$, and $C(t)$ increases at rate R whenever $C(t) < G(t)$. The following "Read Session Axioms" must hold for all $t \geq t_{start}$:

$$P(t) - G(t) \leq B, \tag{1}$$

$$P(t) - C(t) \geq \overline{Y}, \tag{2}$$

$$G(t) \leq P(t). \tag{3}$$

These conditions say that CMFS does not overflow the FIFO, CMFS allows the client to read ahead of the logical clock by up to $\overline{Y}$ bytes, and the client does not read beyond the write point. CMFS therefore provides a guaranteed minimum data rate, but only as long as the client keeps up with its reading. There is no upper bound on the actual data rate; the client and CMFS can, in principle, work arbitrarily far ahead of the logical clock.

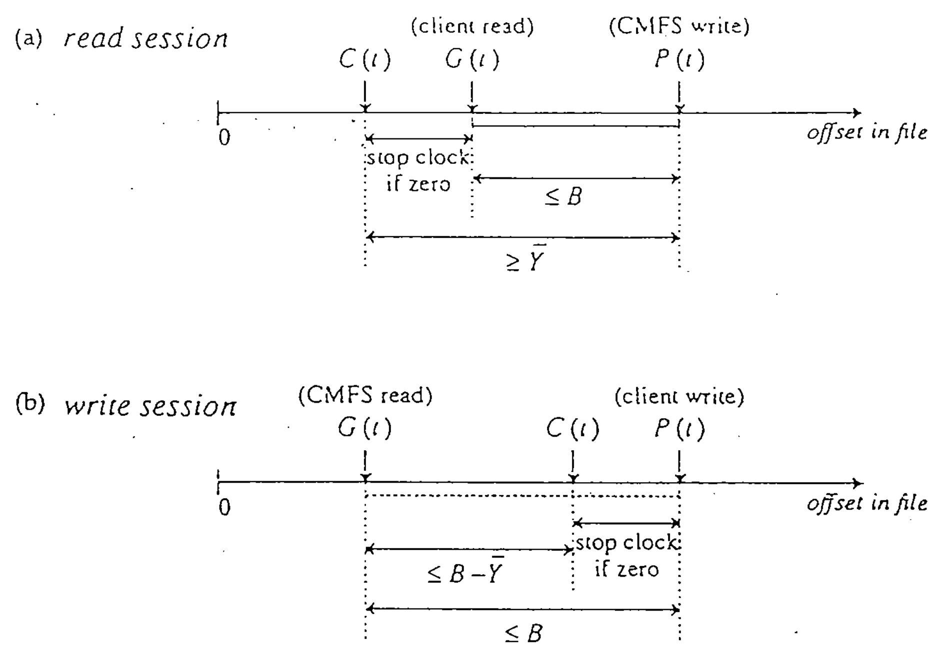


Fig. 1. The semantics of (a) read and (b) write sessions are described in terms of a "put pointer" $P(t)$, a "get pointer" $G(t)$, and a logical clock $C(t)$. The shaded rectangles represent data in the FIFO.

In a write session, the client transfers data into the FIFO buffer, and CMFS moves data from the buffer to the disk. We describe the semantics of write sessions using the same notation as above. In this case, $P(t)$ is the index of the next byte to be inserted in the FIFO by the client, and $G(t)$ is the index of the next byte to be removed by CMFS. The logical clock $C(t)$ increases at rate R whenever $C(t) < P(t)$. The following "Write Session Axioms" must hold for all $t \geq t_{start}$ (see Figure 1b):

$$P(t) - G(t) \leq B, \tag{4}$$

$$C(t) - G(t) \leq B - \overline{Y}, \tag{5}$$

$$G(t) \leq P(t). \tag{6}$$

These conditions say that the client does not overflow the FIFO, that CMFS removes data from the FIFO fast enough so that the client can always write ahead of the logical clock by at least $\overline{Y}$, and that CMFS does not read beyond the write point.

2.2 File Creation and Non-Real-Time Access

CMFS supports both *real-time* and *non-real-time* files. A real-time file is created using the following:

```
create_realtime_file (
  BOOLEAN expandable,
  int size,
  int max_rate);
```

expandable indicates whether the file can be dynamically expanded. If not, size gives its (fixed) size. max_rate is the maximum data rates (bytes per second) at which the file is to be read or written. CMFS rejects the creation request if it lacks disk space or if max_rate is too high.

Non-real-time operations may be performed on either type of file. CMFS provides two non-real-time service classes: *interactive* and *background*. Interactive access is optimized for fast response; background, for high throughput. There are no performance guarantees for non-real-time operations.

2.3 The Symmetry of Reading and Writing

In describing session acceptance and scheduling algorithms, the redundancy of treating read and write sessions separately can be avoided by observing the following symmetry between reading and writing: Suppose a write session S_W has fixed parameters t_{start}, B, $\overline{Y}$, and R, and time-varying parameters $C_W(t)$, $P_W(t)$, and $G_W(t)$. Consider an (imaginary) read session S_R having the same t_{start}, B, $\overline{Y}$, and R parameters, and for which

$$G_R(t) = P_W(t), \tag{7}$$

$$P_R(t) = G_W'(t) + B \tag{8}$$

(see Figure 2). Each disk block written by CMFS in S_W advances P_W, and thus corresponds to a disk block read by CMFS in S_R.

CLAIM 1. *Suppose S_W is a write session, and S_R is defined as above. Then $C_R(t) = C_W(t)$ for all $t \geq t_{start}$, and S_R satisfies the Read Session Axioms (eqs. (1)–(3)) if and only if (iff) S_W satisfies the Write Session Axioms (eq. (4)–(6)).*

PROOF. When $G(t)$ is substituted for $P(t)$ in the definition of $C_W(t)$, the result defines $C_R(t)$. Thus, the logical clock advances in S_R exactly when it advances in S_W and at the same rate, so $C_W(t) = C_R(t)$. The equivalence of the Read and Write axioms then follows from substituting eqs. (7) and (8) in eqs. (1) and (2). □

So, from the point of view of scheduling in CMFS, reading and writing are essentially equivalent. The main difference is the initial conditions: An empty buffer for a write session corresponds to a full buffer for a read session. In describing CMFS's algorithms for scheduling and session acceptance, we will refer only to read sessions.

2.4 Using the CMFS Interface

An idealized client might read one byte from the FIFO every $1/R$ s, beginning at the moment of session acceptance. In general, however, this uniformity is neither necessary nor desirable. For example, (1) data may be grouped into large chunks (e.g., video frames) that are needed at a single moment; (2) data may have long-term rate variation (perhaps because of variable-rate compression); (3) clients may delay initial I/O to synchronize multiple sessions; (4) clients may pause and resume I/O during sessions; and (5)

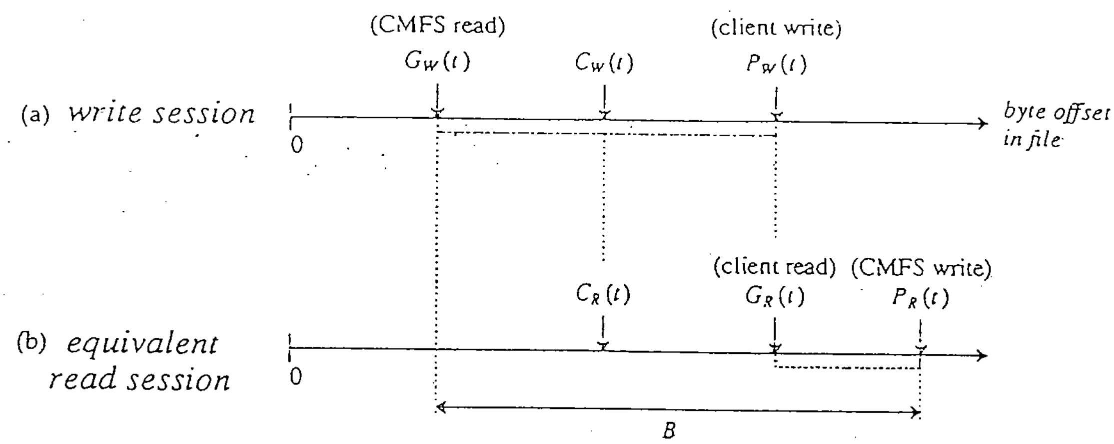


Fig. 2. By interchanging empty/full and read/write, a write session (a) is transformed into a read session (b) that is equivalent with respect to scheduling.

clients may "work ahead," filling up intermediate buffers to improve overall performance.

The CMFS interface accommodates these requirements. To show this, we start by defining a notion of temporal data with variable but bounded rate:

Definition. A *bounded-rate file* F has parameters R and E, and the ith byte of F has a time stamp $T(i) \geq 0$ such that $T(i) \leq T(j)$ for $i < j$, and

$$i - j \leq R(T(i) - T(j)) + E \qquad (9)$$

for all $i > j$ (see Figure 3). In other words, the amount of data in a time interval of length T is at most $RT + E$. (E.g., in a digital video file, R is the maximum long-term data rate, and E is the maximum number of bytes per frame.) The time stamps may be explicit (embedded in the data) or implicit.

Definition. Suppose a client reads a bounded-rate file F. We say that the file is *read in real time starting at* t_0 *with buffer size* N if (1) byte i is read (i.e., removed from the FIFO) before $t_0 + T(i)$, and (2) at any time t, no more than N bytes i such that $T(i) > t - t_0$ have been read.

Intuitively, this means that the client reads the file fast enough to get the data on time, but slow enough so that only N bytes of additional buffer space are needed. The CMFS interface allows a client to read a bounded-rate file in real time using limited buffer space:

CLAIM 2. *Suppose a client creates a session reading a bounded-rate file F with parameters R and E, and the session begins at time 0. Then it is possible for the client to read F in real time starting at E/R with buffer size E.*

The proof is given in the Appendix.

The implicit flow control provided by the CMFS interface can accomplish other goals as well:

—A client can pause a session by simply stopping the removal of data from

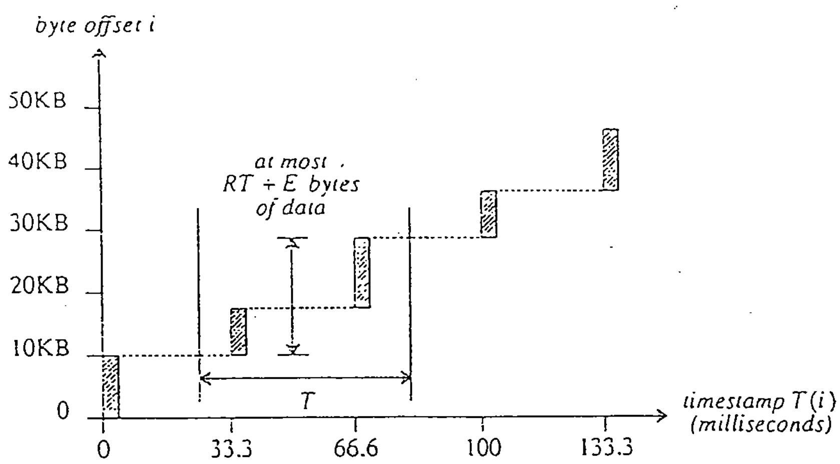


Fig. 3. A *bounded-rate file* contains time-stamped data. This example represents a file of 30-frames/s video data with a varying number of bytes per frame; each frame is shown as a vertical bar. The file has parameters R and E; the number of bytes with time stamps in an interval of length T cannot exceed $TR + E$, as shown.

the FIFO; the logical clock will stop soon thereafter. Data-rate and buffer-requirement guarantees will remain valid after the client resumes reading. (The pause/resume is equivalent to shifting the time stamps of the remaining data.)

—Suppose a client plays several files, transmitted via network connections from different CMFS servers, in synchrony on a single workstation. The I/O server on the workstation handles the synchronization; it begins output only when sufficient data have been received on each connection (e.g., as done by the ACME server [3]). CMFS handles this case with no client intervention: Logical clocks pause during the initial synchronization period and resume thereafter.

—The client can, if the hardware is fast enough, read arbitrarily far ahead of the logical clock. This "workahead" data can then be buffered (in distributed applications, the buffers may be spread across many nodes), protecting against playback glitches and allowing improved system response to transient work load.

3. SYSTEM INTEGRATION ISSUES

3.1 Implementation Alternatives

Our implementation of CMFS runs as a user-level process on UNIX™ accesses a SCSI disk via the UNIX raw disk interface, and communicates with clients via TCP connections. Other architectures, however, are possible (see Figure 4). For example, CMFS could be implemented at either kernel or user level. The client control interface could be provided either as system calls (for local clients only) or by remote procedure call.

™ UNIX is a registered trademark of AT&T Bell Laboratories.

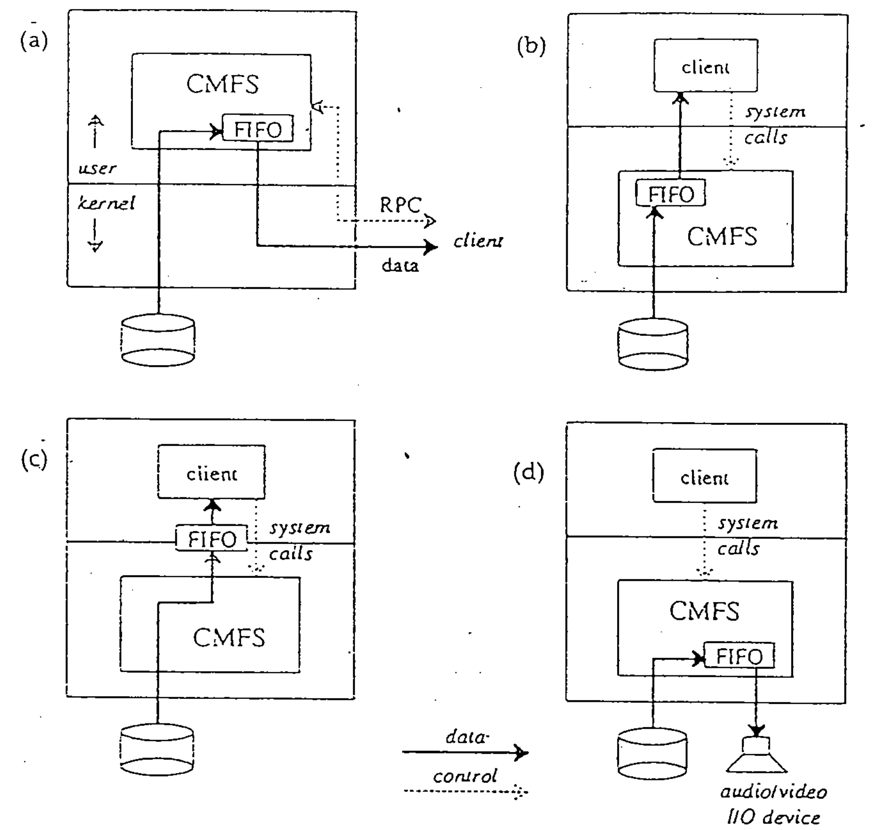


Fig. 4. The CMFS prototype is a user-level process that communicates with its clients over network connections (a). Alternatively, it could be implemented in an OS kernel, with client data access by system calls (b), memory-mapped streams (c), or device interrupts (d).

The client data interface (the mechanism by which data are removed from a read session's FIFO) can take several forms. A general approach is to have data sent out on a flow-controlled network connection, established as part of the request_session ( ) call. Data are removed from the FIFO whenever the protocol allows it. If the client is a user-level process on the same machine as CMFS, the FIFO could be a shared-memory buffer or "memory-mapped stream" [6]. Finally, if the data are consumed by an I/O device on the same machine as CMFS, the FIFO might reside in kernel memory, accessed directly by the I/O device interrupt handler.

These alternatives have different performance implications. For example, a kernel-level implementation might have lower CPU scheduling overhead (most work could be done at the interrupt level) and improved control of physical memory. However, the issues addressed by CMFS (disk scheduling, buffer allocation, etc.) apply regardless of the alternative chosen.

3.2 End-to-End Scheduling

In typical applications, CM data are handled by many shared hardware resources, such as disk, CPU, network, bus, and memory. To provide applicascheduling of all of these resources. CMFS is designed to serve as part of such a "meta-scheduling" scheme, the *CM-resource model* [2]. In this scheme, each resource can be reserved in "sessions" with fixed work load and delay bounds. The parameterization of work load and delay lets resources work ahead on real-time streams so they can respond quickly to non-real-time work load.

A CMFS session's "cushion" parameter $\bar{Y}$ allows the adjacent resource (typically the CPU) to have looser delay bounds. For example, suppose a client runs on the same host as CMFS (as in Figure 4c) and does CPU processing on the data before sending the data to the next resource (say, a network). The client determines an upper bound on the CPU time per unit of data it requires, and reserves a session with the CPU resource. This session has a delay bound, say, Y. It is now easy to show the following:

CLAIM 3. *Suppose, in the above situation, that the client successfully creates a CMFS session with cushion Y, starts the session when it has processed Y amount of data (say, at time 0), and thereafter processes data without (voluntarily) pausing. Then, at all times $t \geq 0$, the client has processed at least Rt bytes of data (R is the session data rate).*

(Note that this would *not* hold if the cushion were less than the CPU delay bound, because it would then be possible for the logical clock to stop.) Claim 3 can be generalized to bounded-rate files. The provides a more useful version of Claim 2, which makes the unrealistic assumption that client CPU processing is instantaneous and can be scheduled at precise instants.

4. DISK LAYOUT ASSUMPTIONS

We assume that the CMFS uses a single-spindle disk drive; disk operations are done sequentially. The disk is read and written in *blocks* of fixed size (a multiple of the hardware sector size). The CMFS reservation and scheduling algorithms do not mandate a particular disk layout. Instead, we assume that the layout allows the following "bounding functions" U_F and V_F to be obtained:

(1) For a given file F, $U_F(n)$ is an upper bound on the time to read n logically contiguous blocks of F (including all seek and rotation time), independent of the position of the disk head and the starting block number to be read.

(2) $V_F(i, n)$ is an upper bound on the time needed to read the n blocks of file F starting at block i.

The bounds need not be tight; slackness in the bounds may, however, cause sessions to be rejected unnecessarily.

The functions U and V should take into account sector interleaving, interrupt-handling latency, the CPU time used by CMFS itself, features (such as track buffering) of the disk controller, and bad sectors detected when the disk is initialized.

4.1 Examples of Disk Layouts

Our CMFS prototype uses contiguous allocation. Each file begins at some point within a cylinder, filling the remainder of that cylinder, zero or more adjacent cylinders, and part of a final cylinder. The number of sectors per block is a fixed parameter. Ignoring CPU overhead and other factors, bounds functions for this layout are easy to derive. We assume that $L_{seek-min}$ and $L_{seek-max}$ are bounds on the one-track seek time and the worst-case seek time, respectively; L_{block} is the time to read one block; $L_{rotation}$ is the rotation time; and N is the number of blocks per cylinder. Furthermore, we assume that the controller does track-buffering; it reads a track into a local buffer immediately after seeking to it. Hence, if an entire track is read, rotational latency is negligible regardless of the order in which the sectors are read. Possible bounds functions are then

$$U_F(n) = L_{seek-max} + nL_{block} + \left\lceil \frac{n}{N} \right\rceil L_{seek-min} + 2L_{rotation} \qquad (10)$$

and

$$V_F(i, n) = L_{seek-max} + nL_{block} + (k-1)L_{seek-min} + \frac{j}{N}L_{rotation},$$

where k is the number of cylinders storing the n blocks of F starting at offset i, and j is the number of blocks not in F in the first and last of these cylinders. To account for CPU overhead, it would be necessary to leave a gap between blocks (perhaps by interleaving them) and to modify U and V accordingly.

A contiguous layout policy is feasible for read-only file systems or if disk space is abundant. For more flexibility, a variant of the 4.2BSD UNIX file system layout [8] could be used. A real-time file might consist of clusters of n contiguous blocks, with every sequence of k clusters constrained to a single cylinder group. n and k are per-file parameters; they are related to the max_rate parameter of the file. Bounds functions U and V can be computed from n, k, the size of a cylinder group, the disk parameters, and the overhead of reading and writing allocation map and index blocks. Allocation and compaction strategies would pose a complex set of issues; we do not discuss them here.

5. ACCEPTANCE TEST

CMFS can accept a new session S only if its data-rate requirements, together with those of existing sessions, can be guaranteed. For this, a sufficient condition is the existence of a static schedule (that cyclically reads fixed numbers of blocks of each session) satisfying the rate requirements of all session under worst-case assumptions and for which enough buffer space is available.

In this section we give an algorithm for deciding whether such a schedule exists. The algorithm constructs the shortest such schedule; this minimal static schedule also plays an important role in dynamic scheduling (see Section 6).

5.1 Properties of Static Schedules

Suppose that sessions $S_1 \cdots S_n$ read files $F_1 \cdots F_n$ at rates $R_1 \cdots R_n$. An *operation set* ϕ assigns to each S_i a positive integer M_i. CMFS *performs* an operation set by seeking to the next block of file F_i, reading M_i blocks of the file, and doing this for every session S_i (the order of operations is not specified). From Section 4,

$$L(i) = U_{F_i}(M_i),$$

$$L(\phi) = \sum_{i=1}^{n} U_{F_i}(M_i) \tag{11}$$

are upper bounds on the elapsed time of S_i's operation and ϕ as a whole, respectively.

The data read in ϕ "sustain" session i for a period $M_i A/R_i$, where A is the block size in bytes; we denote this period $D_i(\phi)$. $D(\phi)$, the period for which the data read in ϕ sustains all of the sessions, is then $\min_{1 \le i \le n} D_i(\phi)$.

If the data read in ϕ "last longer" than the worst-case time it takes to perform ϕ, we call it a *workahead-augmenting set* (*WAS*). This hold if $L(\phi) \le D(\phi)$ or, equivalently,

$$M_i A > R_i L(\phi) \tag{12}$$

for all i.

If the amount of data read for each session in an operation set ϕ fits in the corresponding FIFO, we say that ϕ is *feasible*. This holds if, for all i,

$$(M_i + 1)A + \overline{Y}_i \le B_i, \tag{13}$$

where B_i is the size of the FIFO buffer used by S_i, and $\overline{Y}_i$ is the cushion parameter of S_i (Section 2.1). This inequality reflects the worst case in which a fractional block is already buffered and the client has used none of its cushion.

An *operation sequence* Φ is a pair (π, ϕ) where π is a permutation of $1 \cdots n$ and ϕ is an operation set. CMFS *performs* an operation sequence by doing the operations in ϕ in the order $\pi(1) \cdots \pi(n)$. Φ is called *workahead-augmenting* if ϕ is workahead-augmenting; likewise, Φ is *feasible* if ϕ is feasible.

At time t, the duration of data buffered for session S_i, above and beyond the client's cushion $\overline{Y}$, is given by $(P(t) - C(t) - \overline{Y})/R$, where P, C, $\overline{Y}$, and R are the parameters for S_i. We call this the "workahead" of S_i and denote it by $W_i(t)$. We say that S_i *starves* if W_i becomes negative. The *file system state* $W(t)$ at time t is the vector $\langle W_1(t) \cdots W_n(t)\rangle$.

Let Φ be an operation sequence. Suppose that enough data are buffered so that, if Φ is performed immediately, no session starves before its operation is completed. We then say that the state is *safe relative to* Φ. This holds if

$$W_{\pi(j)}(t) \ge \sum_{i=1}^{j} L(\pi(i))$$

for all j (recall that $L(j)$ is the worst-case time needed for S_j's operation).